Boolean Algebra

Summer 21

Asst. Prof. Nattapol Aunsri, Ph.D.
IATE Research Unit
CCC Research Group
School of Information Technology, MFU

Prepared for Digital Industry Integration: DII, CAMT, CMU

Outline

- Boolean Algebra
  - Basic Boolean Equations
  - Multiple Level Logic Representation
  - Basic Identities
  - Algebraic Manipulation
  - Complements and Duals

Basic Boolean Equations

Basic gates/functions

- AND
  - $Z = A\ B$
  - $X = C\ D\ E$ 3 input gate
  - $Y = F\ G\ H\ K$ 4 input gate
- OR
  - $Z = A + B$
  - $Y = F + G + H + K$ 4 input gate
- NOT
  - $Z = \overline{A}$
  - $Y = \overline{(F\ G\ H\ K)}$ actually 2 level logic

Level Logic

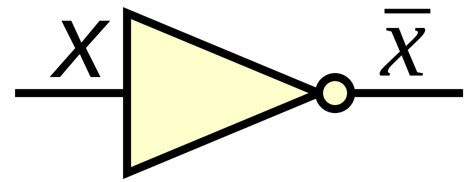
- Consider the following logic equation
  - Z(A,B,C,D) = A B + C D
  - The Z(A,B,C,D) means that the output is a function of the four variables within the ().
  - The AB and CD are terms of the expression.
  - This form of representing the function is an algebraic expression.
  - For this function to be True, either both A AND B are True OR both C AND D are True.

Logic Gates

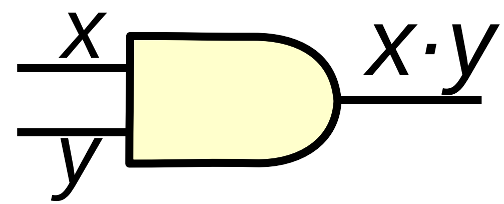
- Inverter, Or, And, XOR gate symbols, etc.
- Multi–input gates.
- Logic circuits and examples.
- Adders, “half,” “full,” and *n*–bit.

Logic Gate Symbols

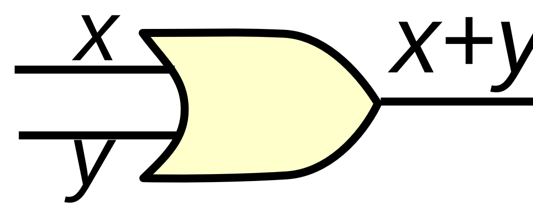
- Inverter (logical NOT, Boolean complement).



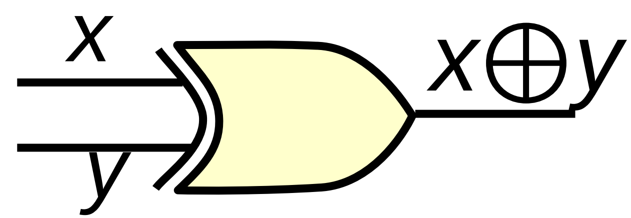
- AND gate (Boolean product).



- OR gate (Boolean sum).

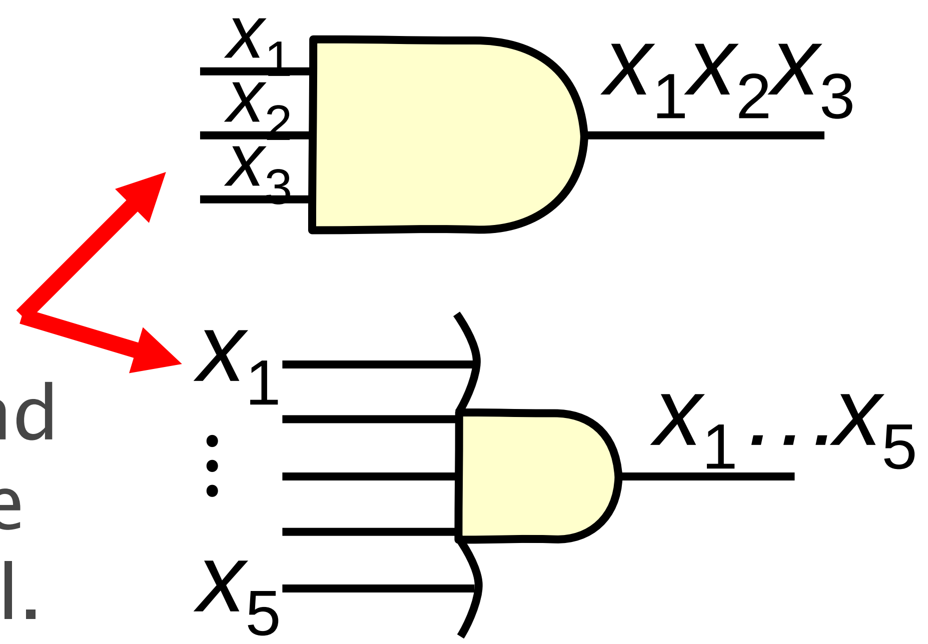


- XOR gate (exclusive-OR, sum mod 2).



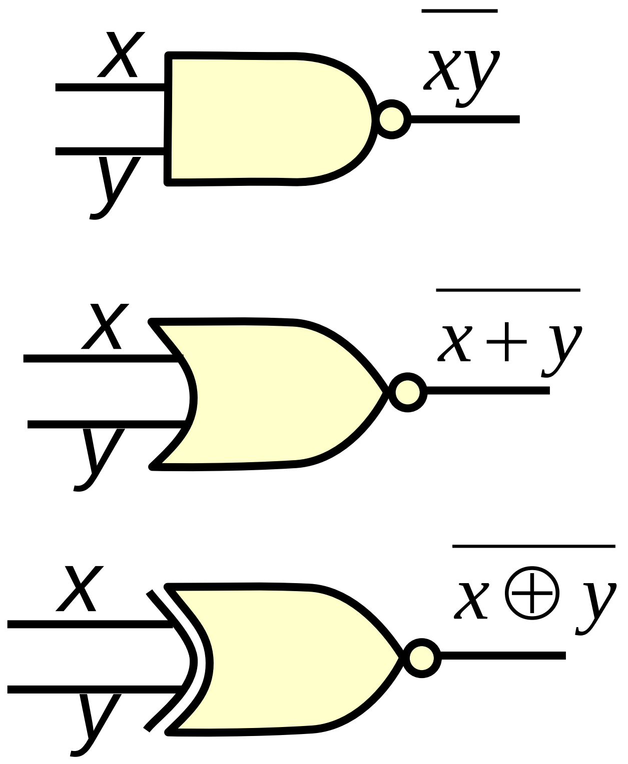
Multi-input

- Can extend these gates to arbitrarily many inputs.
- Two commonly seen drawing styles:
  - Note that the second style keeps the gate icon relatively small.



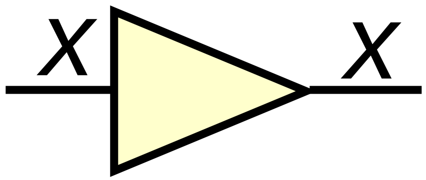
NAND, NOR, XNOR

- Just like the earlier icons, but with a small circle on the gate's output.
  - Denotes that output is complemented.
- The circles can also be placed on inputs.
  - Means, input is complemented before being used.



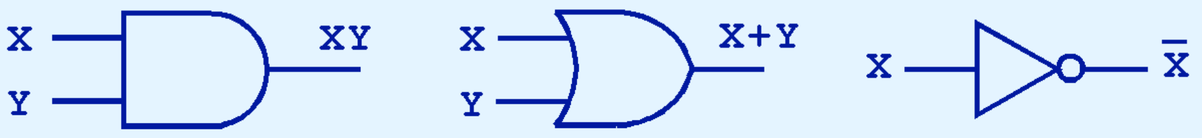
Buffer

- What about an inverter symbol *without* a circle?
- This is called a *buffer*.
- It is the identity function.



Logic Gates

- The three simplest gates are the AND, OR, and NOT gates.



X AND Y

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X OR Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT X

X	$\overline{X}$
0	1
1	0

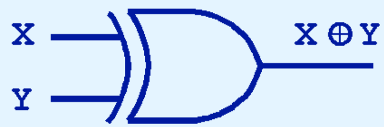
- They correspond directly to their respective Boolean operations, as you can see by their truth tables.
- And these representations map exactly into the transistors on the last two slides.

10

Logic Gates

X XOR Y

X	Y	$X \oplus Y$
0	0	0
0	1	1
1	0	1
1	1	0

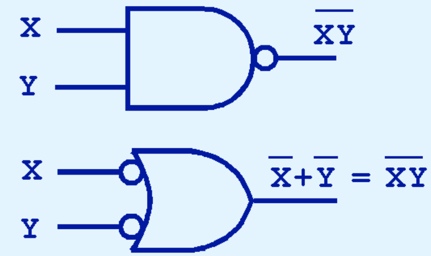


- The output of the XOR operation is true only when the values of the inputs differ.

Note the special symbol ⊕ for the XOR operation.

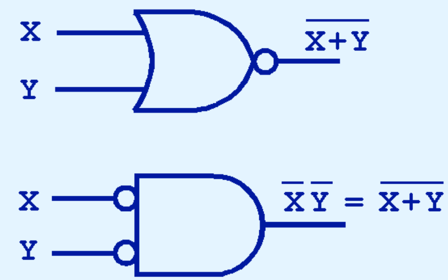
X NAND Y

X	Y	X NAND Y
0	0	1
0	1	1
1	0	1
1	1	0



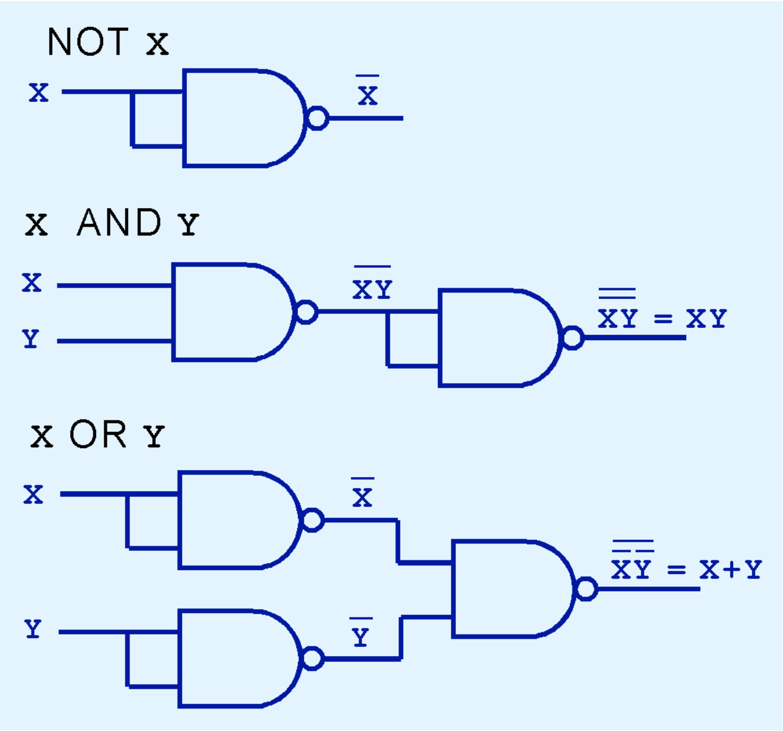
X NOR Y

X	Y	X NOR Y
0	0	1
0	1	0
1	0	0
1	1	0



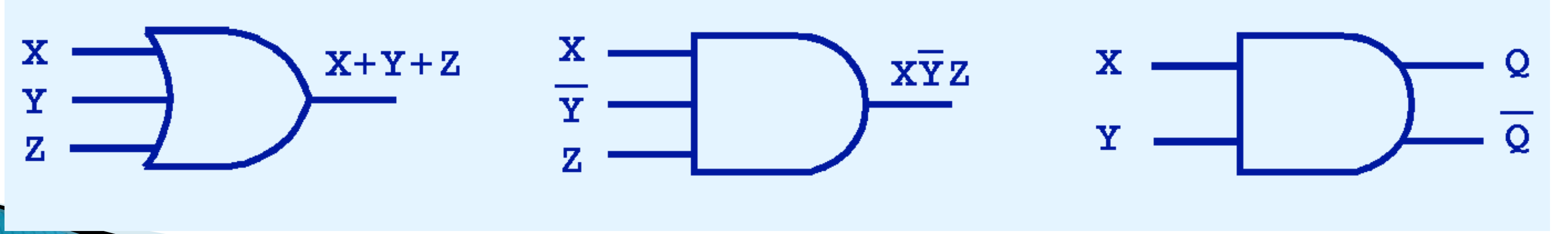
- Symbols for NAND and NOR, and truth tables are shown at the right.

Logic Gates



- NAND and NOR are known as *universal gates* because they are inexpensive to manufacture and any Boolean function can be constructed using only NAND or only NOR gates.

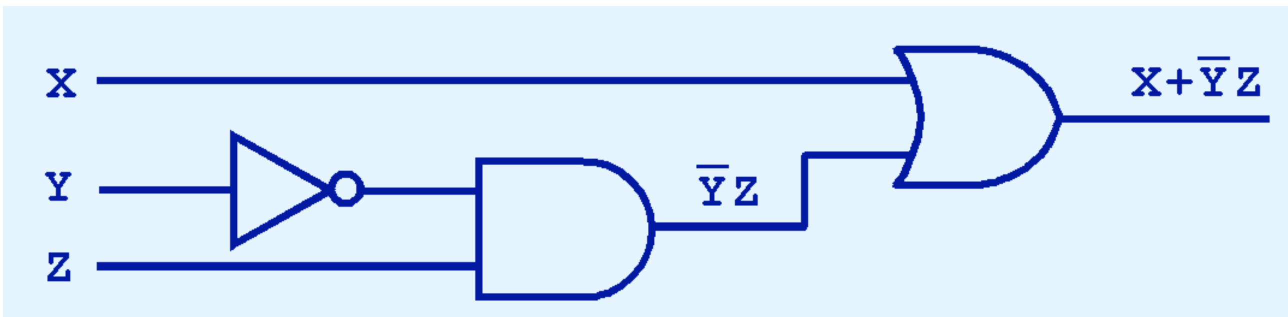
- Gates can have multiple inputs and more than one output.
 A second output can be provided for the complement of the operation. We'll see more of this later.



Digital Components

- Combinations of gates implement Boolean functions.
- The circuit below implements the function:

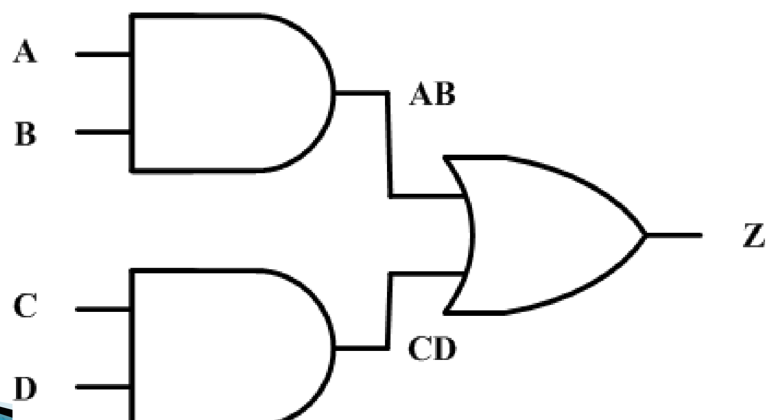
$$F(X, Y, Z) = X + \overline{Y}Z$$



- This is an example of a *combinational logic* circuit.
- Combinational logic circuits produce a specified output (almost) at the instant when input values are applied. Later we'll explore circuits where this is not the case.

Truth table expression

- Just like we had the truth tables for the basic functions, we can also construct truth tables for any function.



A	B	C	D	Z	AB	CD
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	1	0	1
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	1	0	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	1	0	1
1	1	0	0	1	1	0
1	1	0	1	1	1	0
1	1	1	0	1	1	0
1	1	1	1	1	1	1

Examples of Boolean Equations

- Some examples
  - F = AB + CD + BD’
  - Y = CD + A’B’
  - SUM = AB + A C + B C
  - $P = A_0A_1A_2A_3A_4B_0B_1B_2B_3B_4 + \ldots$

  - Equations can be very complex
  - Usually desire a minimal expression

Basic Identities of Boolean Algebra

- 1. $X + 0 = X$

OR +

A	0	RESULT
0	0	0
1	0	1

- 2. $X \cdot 1 = X$

AND •

A	1	RESULT
0	1	0
1	1	1

- 3. $X + 1 = 1$
- 4. $X \cdot 0 = 0$
- 5. $X + X = X$
- 6. $X \cdot X = X$

Basic Identities (2)

- 7. $X + X' = 1$

X	X'	RES
0	1	1
1	0	1

- 8. $X \cdot X' = 0$

X	X'	RES
0	1	0
1	0	0

- 9. $(X')' = X$

Basic Properties (Laws)

- Commutative
  - 10. $X + Y = Y + X$
- Associative
  - 12. $X+(Y+Z)=(X+Y)+Z$
- Distributive
  - 14. $X(Y+Z) = XY+XZ$
  - AND distributes over OR

- Commutative
  - 11. $X \cdot Y = Y \cdot X$
- Associative
  - 13. $X(YZ) = (XY)Z$
- Distributive
  - 15. $X+YZ=(X+Y)(X+Z)$
  - OR distributes over AND

Basic Properties (2)

- DeMorgan's Theorem
- Very important in simplifying equations
  - 16. (X + Y)' = X' · Y'
  - 17. (XY)' = X' + Y'

X	Y	X+Y	$\overline{X+Y}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

X	Y	$\overline{X}$	$\overline{Y}$	$\overline{X}\bullet\overline{Y}$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	0

Simplification

- These properties (Laws and Theorems) can be used to simplify equations to their simplest form.
  - Simplify F=X'YZ+X'YZ'+XZ

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

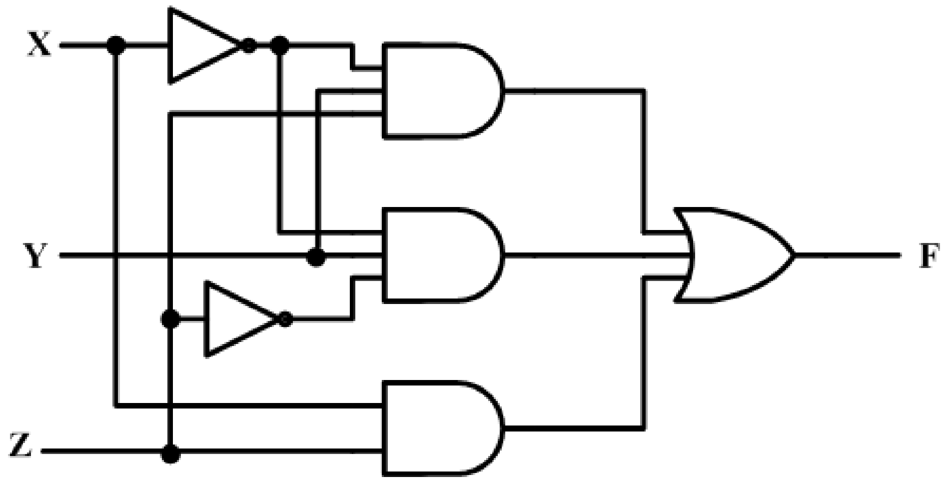
$$= \overline{X}Y(Z + \overline{Z}) + XZ \qquad \text{by identity 14}$$

$$= \overline{X}Y \bullet 1 + XZ \qquad \text{by identity 7}$$

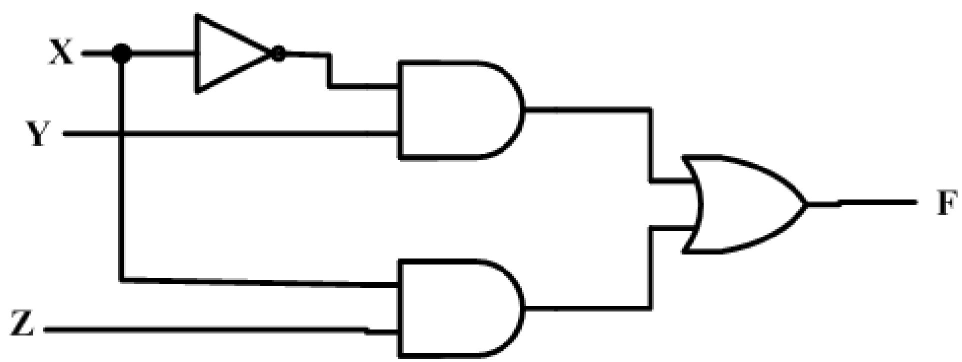
$$= \overline{X}Y + XZ \qquad \text{by identity 2}$$

Affect on implementation

- F = X'YZ + X'YZ' + XZ



- Reduces to F = X'Y + XZ



More examples

- Examples of simplification
  - 1. $X + XY = X \cdot 1 + XY = X(1+Y) = X \cdot 1 = X$
    - Use 2 14 3 2
  - 2. $XY+XY' = X(Y + Y') = X \cdot 1 = X$
    - Use 14 7 2
  - 3. $X+X'Y = (X+X')(X+Y) = 1 \cdot (X+Y) = X+Y$
    - Use 15 7 2

Further Examples

- Examples from the text
  - 4. $X\cdot(X+Y)=X\cdot X+X\cdot Y= X+XY=X(1+Y)=X\cdot 1=X$
    - Use 14 6 14 3 2
  - 5. $(X+Y)\cdot(X+Y')=XX+XY'+XY+YY'=$
  - $X+XY'+XY+0=X(1+Y'+Y)=X\cdot 1=X$
  - by a slightly different reduction
  - 6. $X(X'+Y) = XX'+XY = 0 + XY = XY$

Consensus Theorem

- The Consensus Theorem states that

$$XY + X'Z + YZ = XY + X'Z$$

- This can be considered as redundancy trick.

Application of Consensus Theorem

- Consider
  - $(A+B)(A'+C) = AA' + AC + A'B + BC$
    $= AC + A'B + BC$
    $= AC + A'B$

- Use consensus theorem to simplify the followings:
  - $AB'+BC+AC$
  - $(A+B)(A'+C)(B+C)$
  - $A'B'+B'C'+AC'$

Application of Consensus Theorem

- Consider
  - $(A+B)(A'+C) = AA' + AC + A'B + BC$
    $= AC + A'B + BC$
    $= AC + A'B$

- Use consensus theorem to simplify the followings:
  - $AB'+BC+AC = AB'+BC$
  - $(A+B)(A'+C)(B+C) = (A+B)(A'+C)$
  - $A'B'+B'C'+AC' = A'B'+AC'$

Complement of a function

- In real implementation sometimes the complement of a function is needed.
  - Have F=X'YZ'+X'Y'Z

$$F = \bar{X}Y\bar{Z} + \bar{X}\bar{Y}Z$$

$$\bar{F} = \overline{\bar{X}Y\bar{Z} + \bar{X}\bar{Y}Z}$$

$$= (\overline{\bar{X}Y\bar{Z}})\bullet(\overline{\bar{X}\bar{Y}Z})$$

$$= (X+\bar{Y}+Z)\bullet(X+Y+\bar{Z})$$

Duals

- What is meant by the dual of a function?
  - The *dual* of a function is obtained by interchanging OR and AND operations and replacing 1s and 0s with 0s and 1s.
- Shortcut to getting function complement
  - Starting with the equation on the previous slide
  - Generate the dual $F=(X'+Y+Z')(X'+Y'+Z)$
  - Complement each literal to get:
  - $F'=(XY'Z)+(XYZ')$

Thanks

Q & A