Digital Systems Design and Laboratory
[2. Boolean Algebra]

Chung-Wei Lin

cwlin@csie.ntu.edu.tw

CSIE Department

National Taiwan University

We would like to thank Prof. Hui-Ru Jiang (NTU) for contributing the lecture material

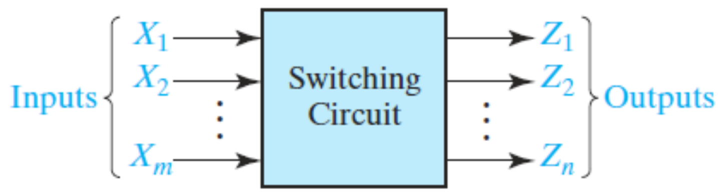
Outline

- **Introduction**
- Basic Operation
- Boolean Expressions and Truth Tables
- Basic Theorems
- Commutative, Associative, Distributive, and DeMorgan's Laws
- Simplification Theorems
- Multiplying Out and Factoring
- Complementing Boolean Expressions

Introduction

- **Boolean algebra**
  - Is the basic mathematics for logic design of digital systems
- **History**
  - George Boole developed Boolean algebra in 1847 and used it to solve problems in mathematical logic
  - Claude Shannon first applied Boolean algebra to the design of switching circuits in 1939
    - Master's thesis (21 years old)
- **Switching devices we will use are essentially two-state devices**
  - Represent an input or output by a Boolean variable
  - 1/0 for High/Low or True/False or Yes/No or Closed/Open
    - Just symbols
    - No numeric value

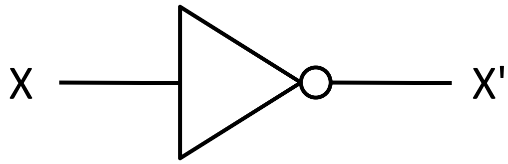
FIGURE 1-1
Switching Circuit



Outline

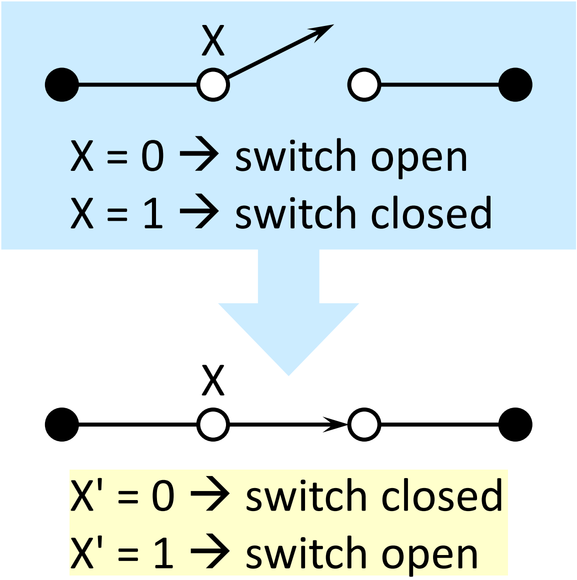
Logic NOT

- Complement = Inverse = Negate = NOT (' ; ¯ ; ~ ; ¬)
  - 0' = 1, 1' = 0
  - Symbol (NOT gate, inverter)



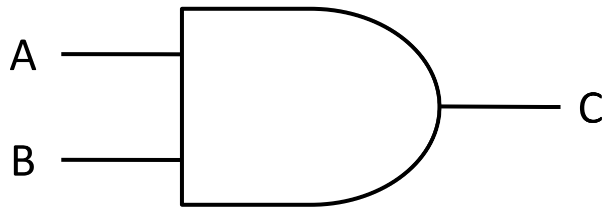
  - Truth table

X (Input)	X' (Output)
0	1
1	0



Logic AND

- AND (• ; ∧ ; sometimes omitted)
  - 0 • 0 = 0, 0 • 1 = 0, 1 • 0 = 0, 1 • 1 = 1
  - Symbol (AND gate)



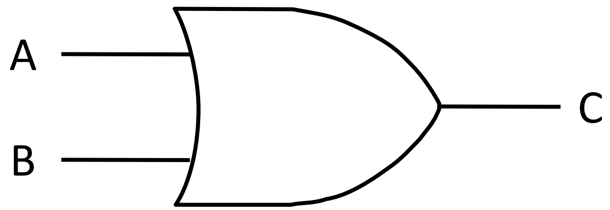
C = 0 → open circuit between 1 and 2
C = 1 → closed circuit between 1 and 2

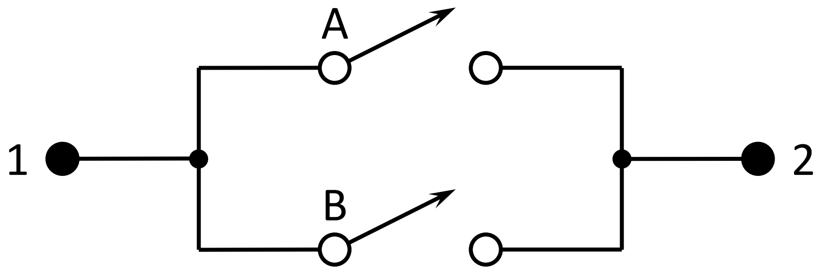
  - Truth table

A	B	C = A • B
0	0	0
0	1	0
1	0	0
1	1	1

Logic OR

- OR (+ ; ∨)
  - 0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, 1 + 1 = 1
  - Symbol (OR gate)





C = 0 → open circuit between 1 and 2
C = 1 → closed circuit between 1 and 2

  - Truth table

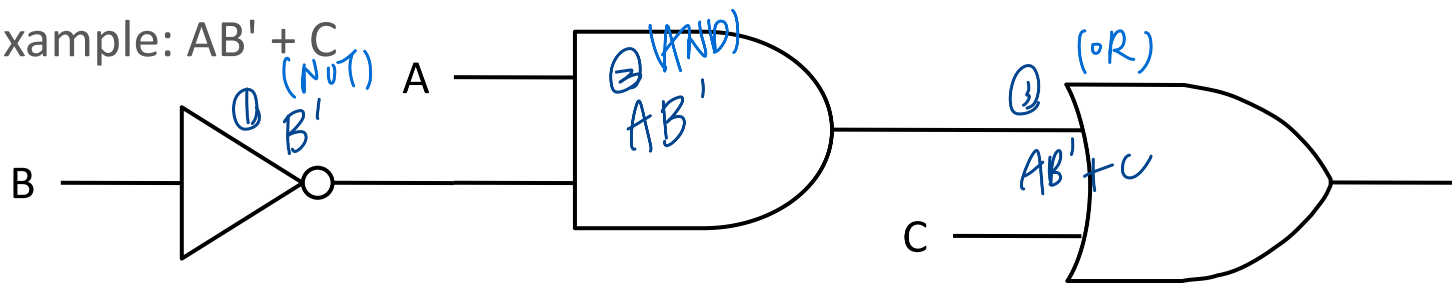
A	B	C = A + B
0	0	0
0	1	1
1	0	1
1	1	1

Outline

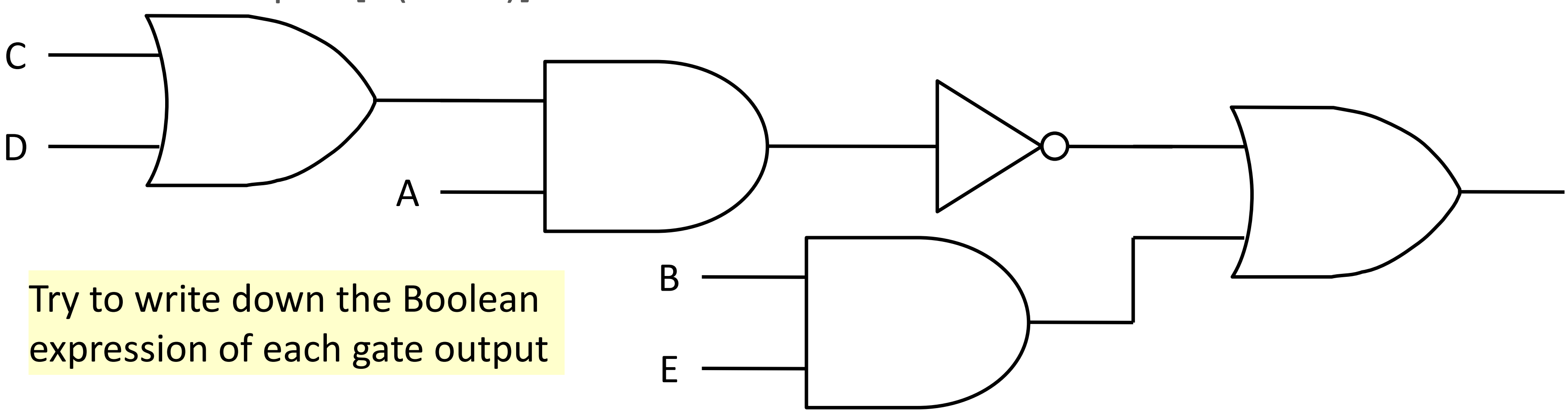
- Introduction
- Basic Operation
- **Boolean Expressions and Truth Tables**
- Basic Theorems
- Commutative, Associative, Distributive, and DeMorgan's Laws
- Simplification Theorems
- Multiplying Out and Factoring
- Complementing Boolean Expressions

Boolean Expressions vs. Logic Gates

- A Boolean expression is formed by basic operations on constants or variables, e.g., 0, 1, X, Y'
- Realize a Boolean expression by a circuit of logic gates
  - Perform operations in order: parentheses → NOT → AND → OR
  - Example: AB' + C



  - Example: [A(C + D)]' + BE



Try to write down the Boolean expression of each gate output

Boolean Expressions vs. Truth Tables

↳ 檢查 equivalence

- ❑ A truth table specifies the output values of a Boolean expression for all possible combinations of input values
  - ➢ How to check the equivalence between two expressions?
  - ➢ Example: AB' + C = (A + C)(B'+ C)

A	B	C	B'	AB'	LHS	A+C	B'+C	RHS
0	0	0	1	0	0	0	1	0
0	0	1	1	0	1	1	1	1
0	1	0	0	0	0	0	0	0
0	1	1	0	0	1	1	1	1
1	0	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1
1	1	0	0	0	0	1	0	0
1	1	1	0	0	1	1	1	1

相同 → 代表 AB'+C (A+C)(B'+C) equivalent

Outline

- Introduction
- Basic Operation
- Boolean Expressions and Truth Tables
- **Basic Theorems**
- Commutative, Associative, Distributive, and DeMorgan's Laws
- Simplification Theorems
- Multiplying Out and Factoring
- Complementing Boolean Expressions

Basic Theorems

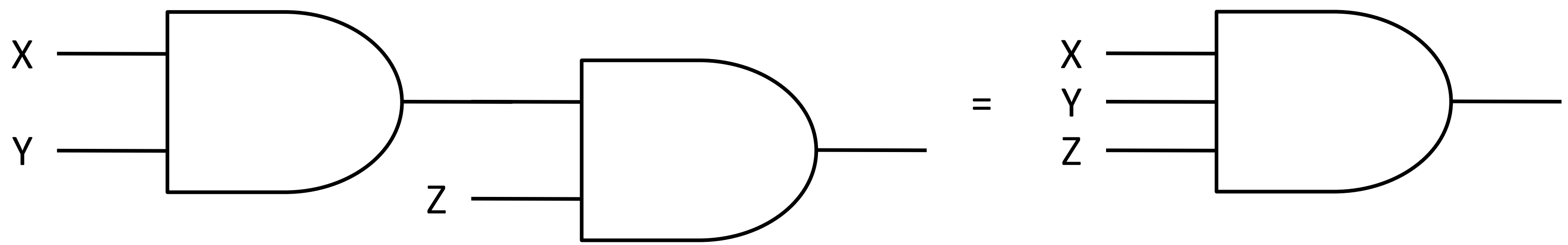
- Operations with 0 and 1
  - X + 0 = X
  - X • 1 = X
  - X + 1 = 1
  - X • 0 = 0
- Idempotent laws
  - X + X = X　$\begin{cases} 0+0=0 \\ 1+1=1 \end{cases}$
  - X • X = X　$\begin{cases} 0\cdot 0=0 \\ 1\cdot 1=1 \end{cases}$
- Involution law
  - (X')' = X　$\begin{cases} (1')' = (0)' = 1 \\ (0')' = (1)' = 0 \end{cases}$
- Laws of complementarity
  - X + X' = 1　∵ X和X'必有一為1
  - X • X' = 0　∵ X和X'必有一為0

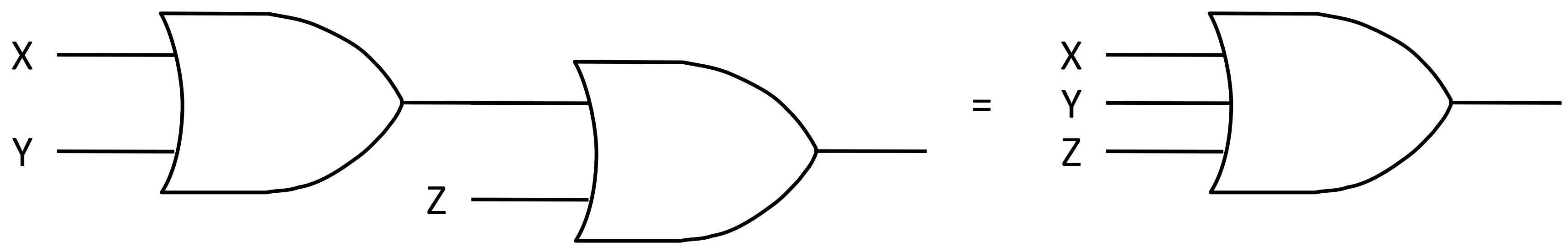
Outline

- ❑ Introduction
- ❑ Basic Operation
- ❑ Boolean Expressions and Truth Tables
- ❑ Basic Theorems
- ❑ **Commutative, Associative, Distributive, and DeMorgan's Laws**
- ❑ Simplification Theorems
- ❑ Multiplying Out and Factoring
- ❑ Complementing Boolean Expressions

Commutative and Associative Laws

- Commutative laws for AND and OR
  - $XY = YX$
  - $X + Y = Y + X$
- Associative laws for AND and OR
  - $(XY)Z = X(YZ) = XYZ$
  - $(X + Y) + Z = X + (Y + Z) = X + Y + Z$





Distributive and DeMorgan's Laws

- Distributive laws
  - Ordinary one : $X(Y + Z) = XY + XZ$
  - Second one: $\mathbf{X + YZ = (X + Y)(X + Z)}$
    - $X + YZ = X \bullet 1 + YZ = X\,\underline{(1 + Y + Z)} + YZ = X + XY + XZ + YZ = XX + XY + XZ + YZ$
      $= XX + XZ + YX + YZ = X\,(X + Z) + Y\,(X + Z) = (X + Y)(X + Z)$

      $1 = 1 + Y + Z$
    - You can also use a truth table to prove it
- DeMorgan's laws
  - $(X + Y)' = X'Y'$
  - $(XY)' = X' + Y'$

Duality (1/2)

- The dual of a Boolean expression is obtained by
  - Interchanging the constants 0 and 1
  - Interchanging the operations of AND and OR
  - Leaving variables and complements unchanged
- Given a Boolean identity, another identity can be obtained by taking the dual of both sides of the identity

Duality (2/2)

- Laws of Boolean algebra

Dual: ① 0, 1 对調 ② AND ⇔ OR 交換 ③ variable complement 不变.

➢ Operations with 0 and 1	[1]	$X + 0 = X$	[1D]	$X \bullet 1 = X$
	[2]	$X + 1 = 1$	[2D]	$X \bullet 0 = 0$
➢ Idempotent laws	[3]	$X + X = X$	[3D]	$X \bullet X = X$
➢ Involution law	[4]	$(X')' = X$		
➢ Laws of complementarity	[5]	$X + X' = 1$	[5D]	$X \bullet X' = 0$
➢ Commutative laws	[6]	$X + Y = Y + X$	[6D]	$XY = YX$
➢ Associative laws	[7]	$(X + Y) + Z = X + (Y + Z) = X + Y + Z$		
	[7D]	$(XY)Z = X(YZ) = XYZ$		
➢ Distributive laws	[8]	$X(Y + Z) = XY + XZ$		
	[8D]	$X + YZ = (X + Y)(X + Z)$		
➢ DeMorgan's laws	[9]	$(X + Y)' = X'Y'$		
	[9D]	$(XY)' = X' + Y'$		

Outline

- ❑ Introduction
- ❑ Basic Operation
- ❑ Boolean Expressions and Truth Tables
- ❑ Basic Theorems
- ❑ Commutative, Associative, Distributive, and DeMorgan's Laws
- ❑ **Simplification Theorems**
- ❑ Multiplying Out and Factoring
- ❑ Complementing Boolean Expressions

Simplification Theorems

Uniting

dual

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

$X\underline{(Y+Y')} = X \cdot 1 = X$, $Y+Y' = 1$

Absorption

dual

- $X + XY = X$
- $X (X + Y) = X$

$X \cdot 1 + X \cdot Y = X \cdot (1+Y) = X \cdot 1 = X$

Elimination

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

$X=0$: $X+X'Y = 0+1\cdot Y = Y = 0+Y = X+Y$

$X=1$: $X+X'Y = 1+0\cdot Y = 1 = 1+Y = X+Y$

Consensus

- $\underline{X}\boxed{Y} + \underline{X}'\boxed{Z} + \boxed{YZ} = XY + X'Z$
- $(X + Y)(X' + Z)\boxed{(Y + Z)} = (X + Y)(X' + Z)$

可直接拿掉

Absorption

$Z=1$: $Z+YZ = 1+Y = 1$

$Z=0$: $Z+YZ = 0+0=0$

对$X=0$:

(左) $XY+X'Z+YZ = Z+YZ = Z$

(右) $XY+X'Z = Z$

对$X=1$

(左) $XY+X'Z+YZ = Y+YZ = Y$

(右) $XY+X'Z = Y$

Simplification Practices

- Simplify Z = A'BC + A'

$A'BC + A' = A'BC + A' \cdot 1 = A'(BC+1) = A'$

- Simplify Z = [A + B'C + $\underbrace{\text{D + EF}}_{X}$][A + B'C + $\underbrace{\text{(D + EF)'}}_{X'}$]

$= [A+B'C+D+EF][A+B'C+D'(EF)'] = [A+B'C+D+EF][A+B'C+D'(E'+F')]$

- Simplify Z = $\underbrace{\text{(AB + C)}}_{X'}$ $\underbrace{\text{(B'D + C'E')}}_{Y}$ + $\underbrace{\text{(AB + C)'}}_{=X}$

$= (AB+C)' + (B'D+C'E')$

$X + X'Y = X + Y$ (elimination)

$$= [\underbrace{A+B'C}_{X} + \underbrace{D+EF}_{Y}][\underbrace{A+B'C}_{X} + \underbrace{D'E'+D'F'}_{Z}]$$

$(X+Y)(X+Z) = X+YZ$

$$= (A+B'C) + (D+EF)\underbrace{(D'E'+D'F')}_{\shortparallel} = A+B'C \quad \square$$

$D'(E'+F') = (D+EF)'$

Outline

- ❑ Introduction
- ❑ Basic Operation
- ❑ Boolean Expressions and Truth Tables
- ❑ Basic Theorems
- ❑ Commutative, Associative, Distributive, and DeMorgan's Laws
- ❑ Simplification Theorems
- ❑ **Multiplying Out and Factoring**
- ❑ Complementing Boolean Expressions

Multiplying Out

- Use the distributive laws to multiply out an expression to obtain a **sum-of-products** (SOP) form
  - Ordinary distributive law: $X(Y + Z) = XY + XZ$
  - Second distributive law: $X + YZ = (X + Y)(X + Z)$
- Example: multiply out $(\underbrace{A}_{X} + \underbrace{BC}_{Y})(\underbrace{A}_{X} + \underbrace{D + E}_{Z})$

  $(X+Y)(X+Z)$
  $= X + YZ$

  - Use the ordinary distributive law
    - $(A + BC)(A + D + E) = A + AD + AE + ABC + BCD + BCE$
      $= A(1 + D + E + BC) + BCD + BCE$
      $= \underline{A + BCD + BCE}$

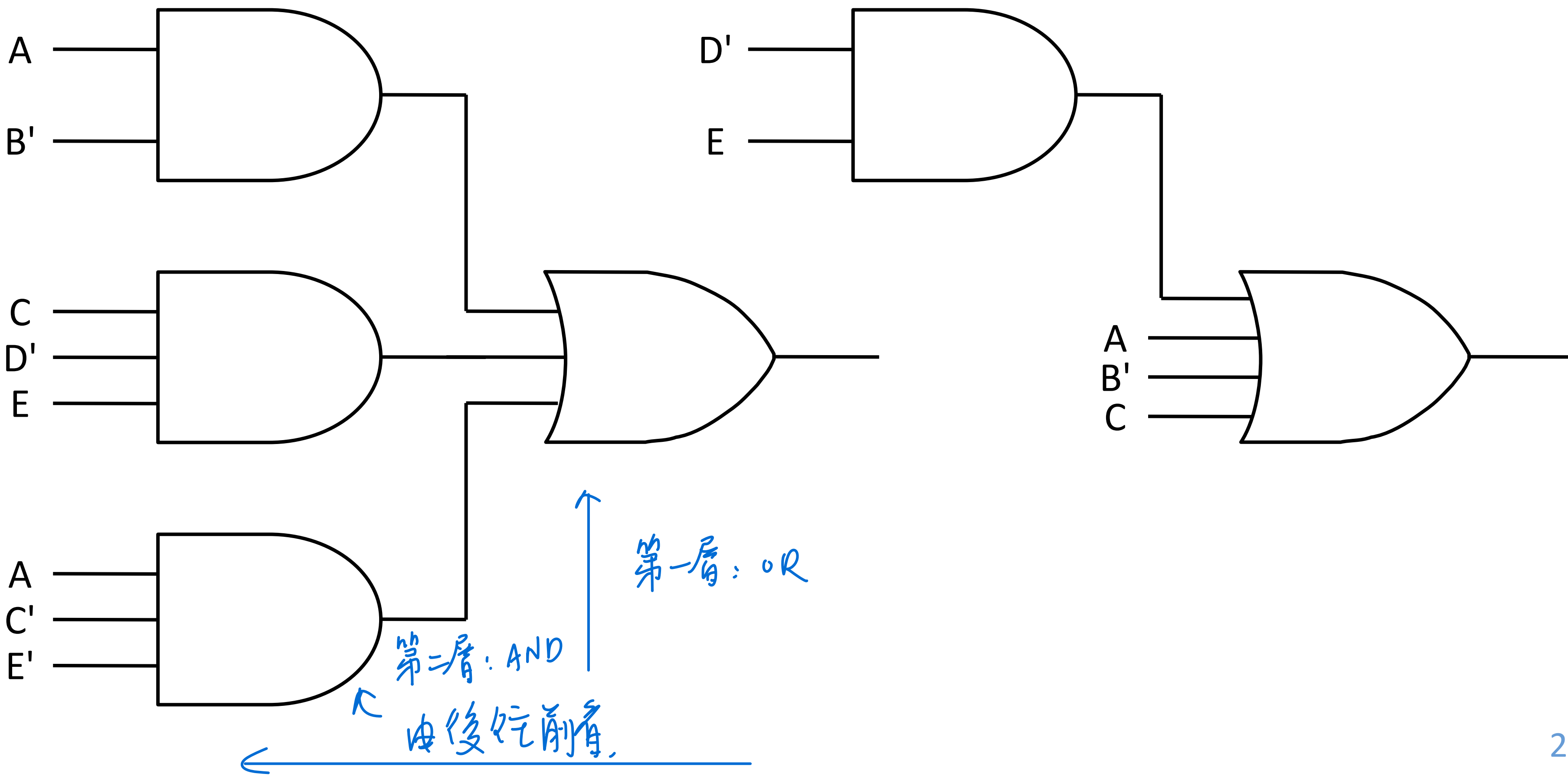
    $A + BC \cdot (D+E)$
    $= A + BCD + BCE$

  - Use the second distributive law sum of product
    - $(A + BC)(A + D + E) = A + BC(D + E) = \underline{A + BCD + BCE}$

SOP vs. Logic Gates

* SOP 對應一個兩層的 AND-OR circuit

- Realize SOPs by two-level circuits (AND-OR)
  - AB' + CD'E + AC'E'
  - A + B' + C + D'E



Factoring

- Use the second distributive law to factor an expression to obtain a **product-of-sums** (POS) form
  - $\boxed{X + YZ = (X + Y)(X + Z)}$
- Example: factor $A + \underbrace{B'}_{Y}\underbrace{CD}_{Z}$

$= (A+B')(A+CD) = (A+B')(A+C)(A+D)$

- Example: factor $AB' + C'D$

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

- Example: factor $C'D + C'E' + G'H$

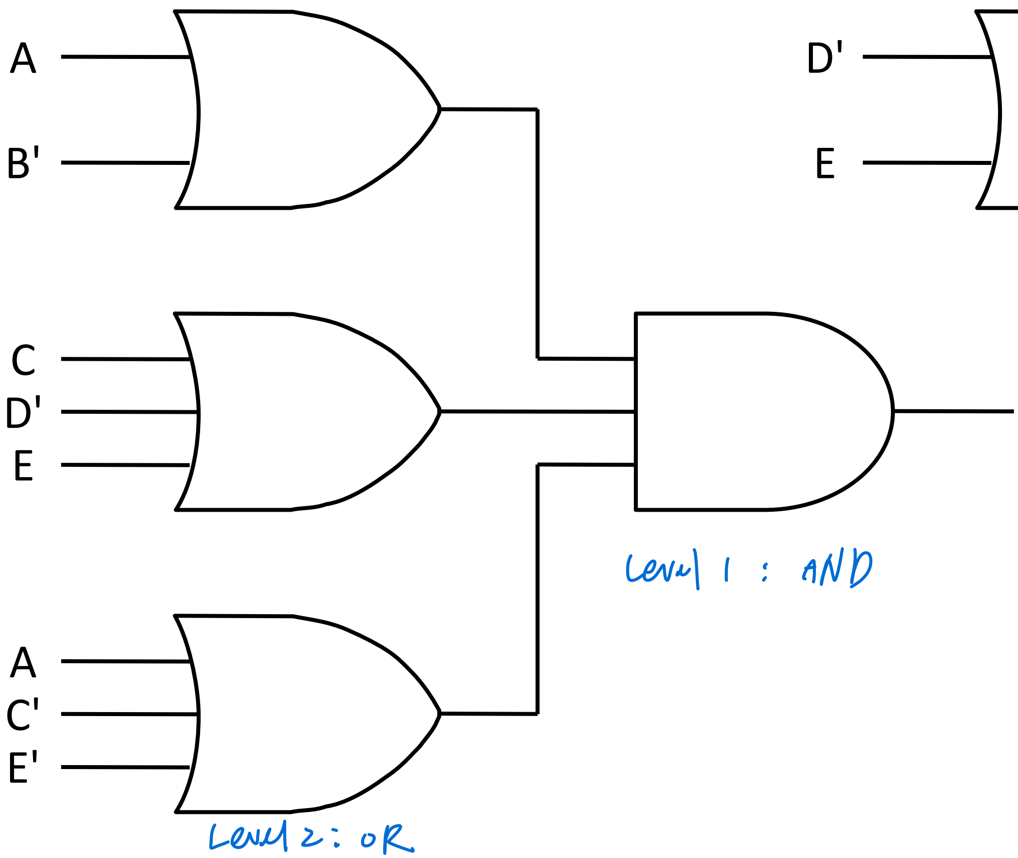
$= C'(D+E') + G'H = [C'(D+E') + G'][C'(D+E') + H]$

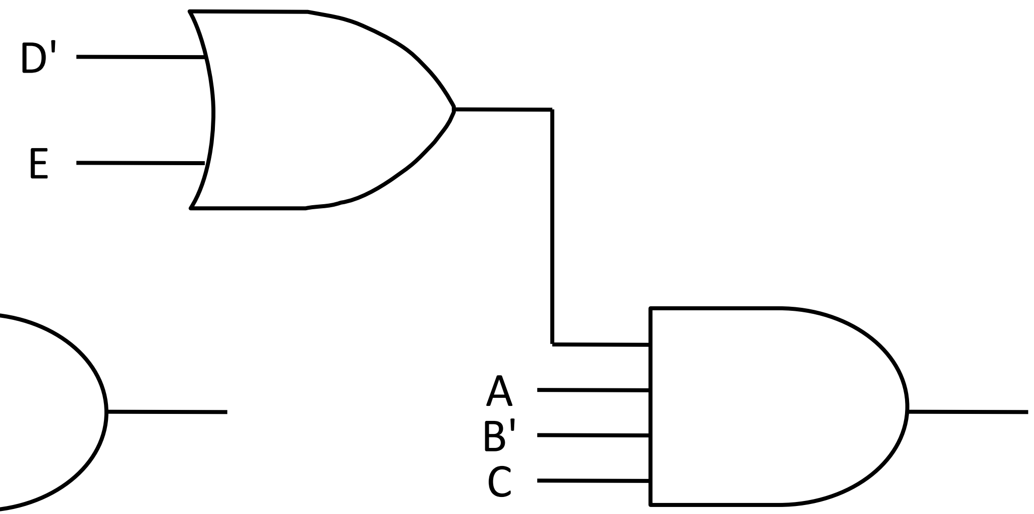
$= (C'+G')[(D+E')+G'](C'+H)[(D+E')+H]$

$= (C'+G')(D+E'+G')(C'+H)(D+E'+H)$

POS vs. Logic Gates

- Realize POSs by two-level circuits (OR-AND)
  - (A + B')(C + D' + E)(A + C' + E')
  - AB'C(D' + E)





Outline

- ❑ Introduction
- ❑ Basic Operation
- ❑ Boolean Expressions and Truth Tables
- ❑ Basic Theorems
- ❑ Commutative, Associative, Distributive, and DeMorgan's Laws
- ❑ Simplification Theorems
- ❑ Multiplying Out and Factoring
- ❑ **Complementing Boolean Expressions**

Complementing Boolean Expressions

- DeMorgan's laws with n variables
  - $(X_1 + X_2 + \ldots + X_n)' = X_1'X_2' \ldots X_n'$
  - $(X_1X_2 \ldots X_n)' = X_1' + X_2' + \ldots + X_n'$
- Complement an expression by iteratively applying DeMorgan's laws
  - Example: complement (AB' + C)D' + E so that NOT is applied only to single variables
    - $[(A \bullet B' + C) \bullet D' + E]' = [(A \bullet B' + C) \bullet D']' \bullet E'$
      $= [(A \bullet B' + C)' + D] \bullet E'$
      $= [(A \bullet B')' \bullet C' + D] \bullet E'$
      $= [(A'+B) \bullet C' + D] \bullet E'$

Q&A