

# Digital Systems Design and Laboratory

## [ 7. Multi-Level Gate Circuits ]

Chung-Wei Lin

[cwlin@csie.ntu.edu.tw](mailto: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

# Recap: Logic Design

## □ Steps

- Translate the word description into a switching function (Unit 4)
    - Truth table
    - Boolean expression
      - SOP/POS derived from minterm or maxterm expansion (Unit 4)
  - Simplify the function
    - Boolean algebra (Units 2 and 3)
    - Karnaugh map (Unit 5)
    - Quine-McCluskey (Unit 6)
    - Other methods
  - Realize it using available logic gates
    - AND-OR, OR-AND (Unit 2)
    - NAND-NAND, NOR-NOR (Unit 7)
    - Others
- 做出來了.*
- SOP*

# Outline

## ❑ **Multi-Level Gate Circuits**

- ❑ NAND and NOR Gates
- ❑ Two-Level NAND- and NOR-Gate Circuits
- ❑ Multi-Level NAND- and NOR-Gate Circuits
- ❑ Circuit Conversion Using Alternative Gate Symbols
- ❑ Design of Two-Level, Multiple-Output Circuits
- ❑ Multiple-Output NAND and NOR Circuits

# AND-OR & OR-AND Circuits

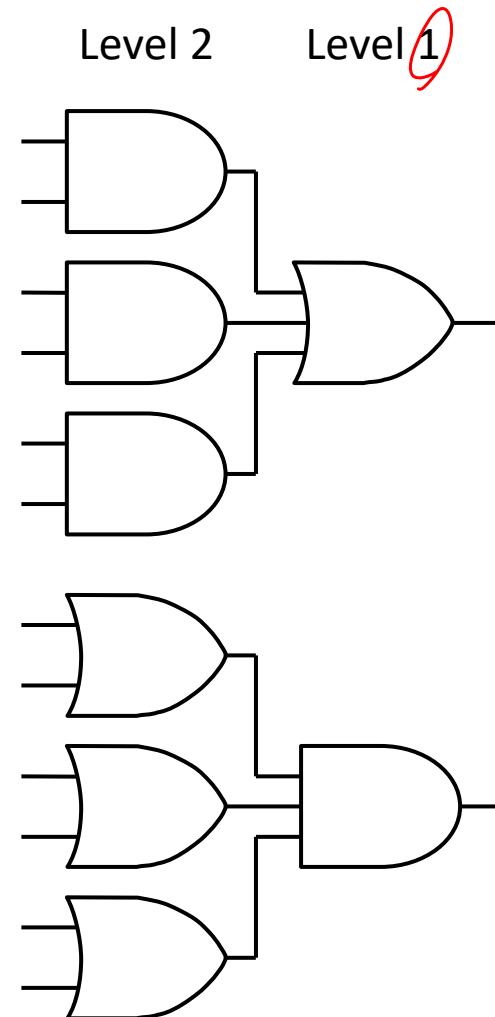
→ input 到 output 最多通過幾個 gate

- The # of levels in a circuit =  
the maximum # of gates cascaded  
in series between an input & the output

- Count from the output *從 output 往 input 算*.
- Do NOT count inverters at inputs *inverter 不算*.

- Two-level circuits learned so far

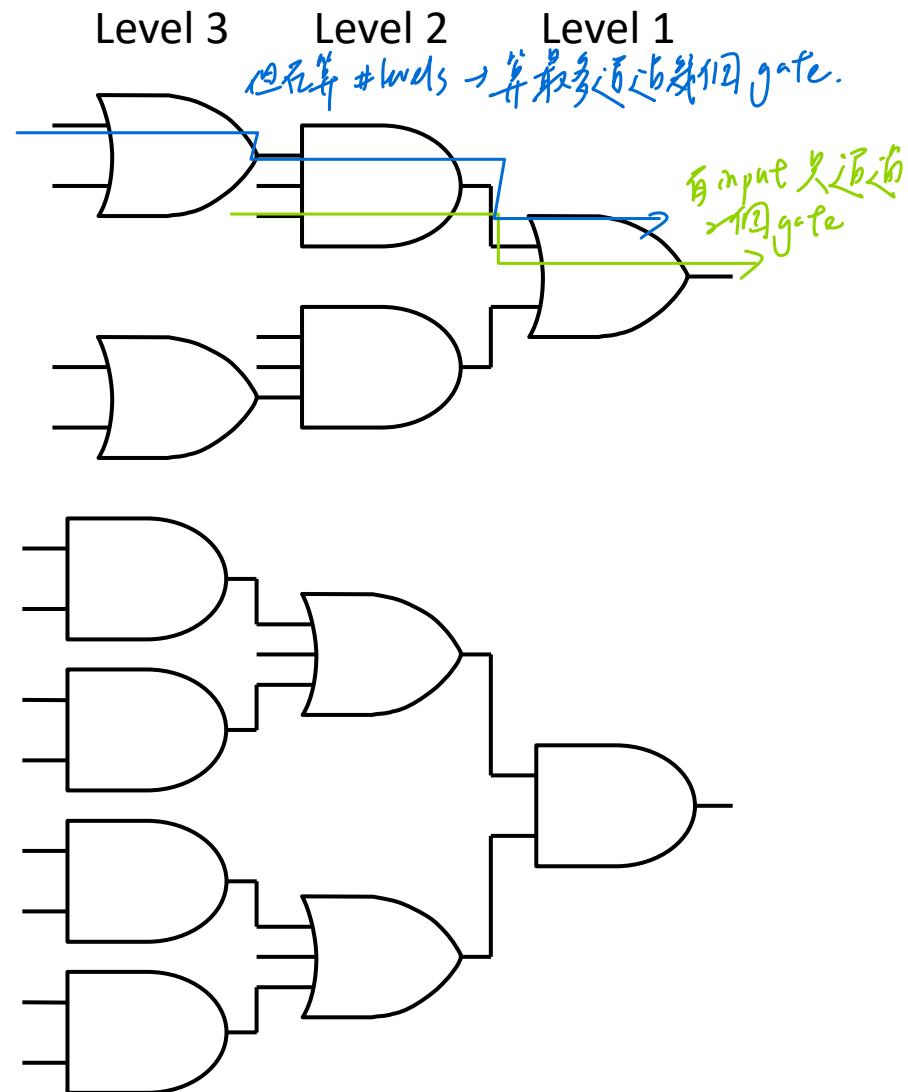
- AND-OR circuit = SOP form
  - One level of AND gates + one OR gate
- OR-AND circuit = POS form
  - One level of OR gates + one AND gate



# OR-AND-OR & AND-OR-AND Circuits

## ❑ Three-level circuits

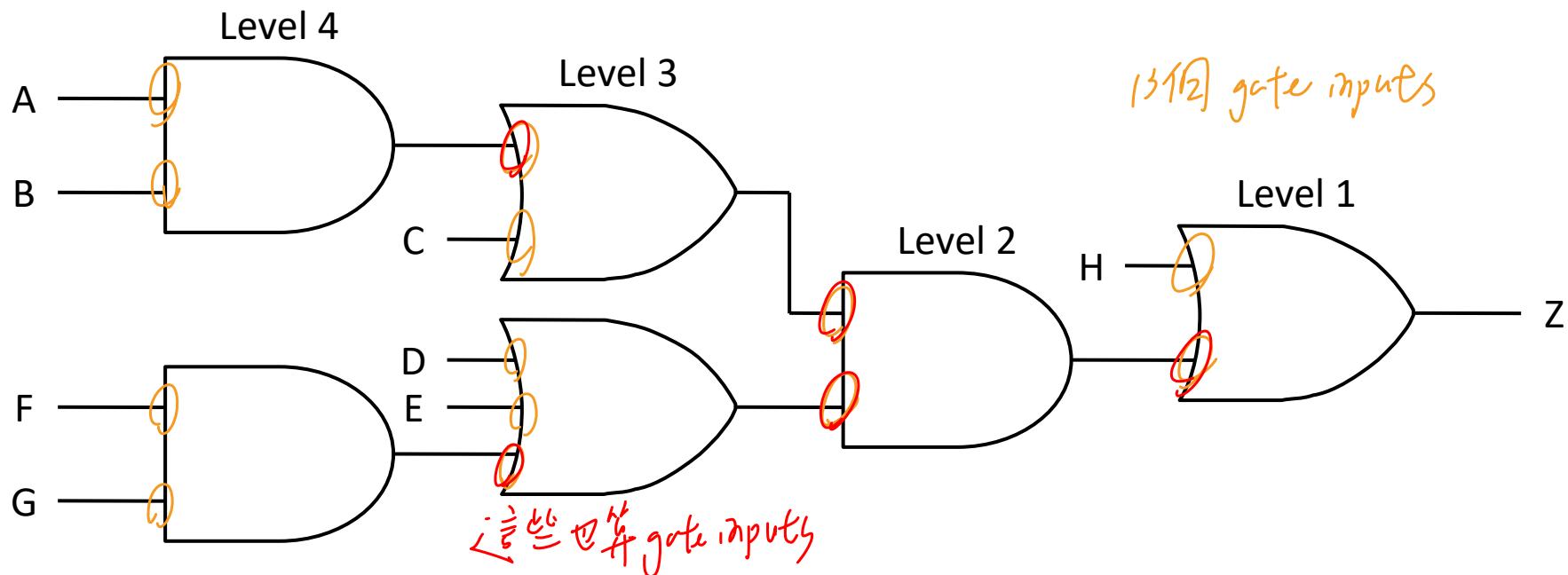
- OR-AND-OR circuit
- AND-OR-AND circuit



# One More Example (1/2)

## □ How to count the # of levels?

- By inspection
- Example:  $Z = (AB + C)(D + E + FG) + H$ 
  - 4 levels
  - 6 gates
  - 13 gate inputs



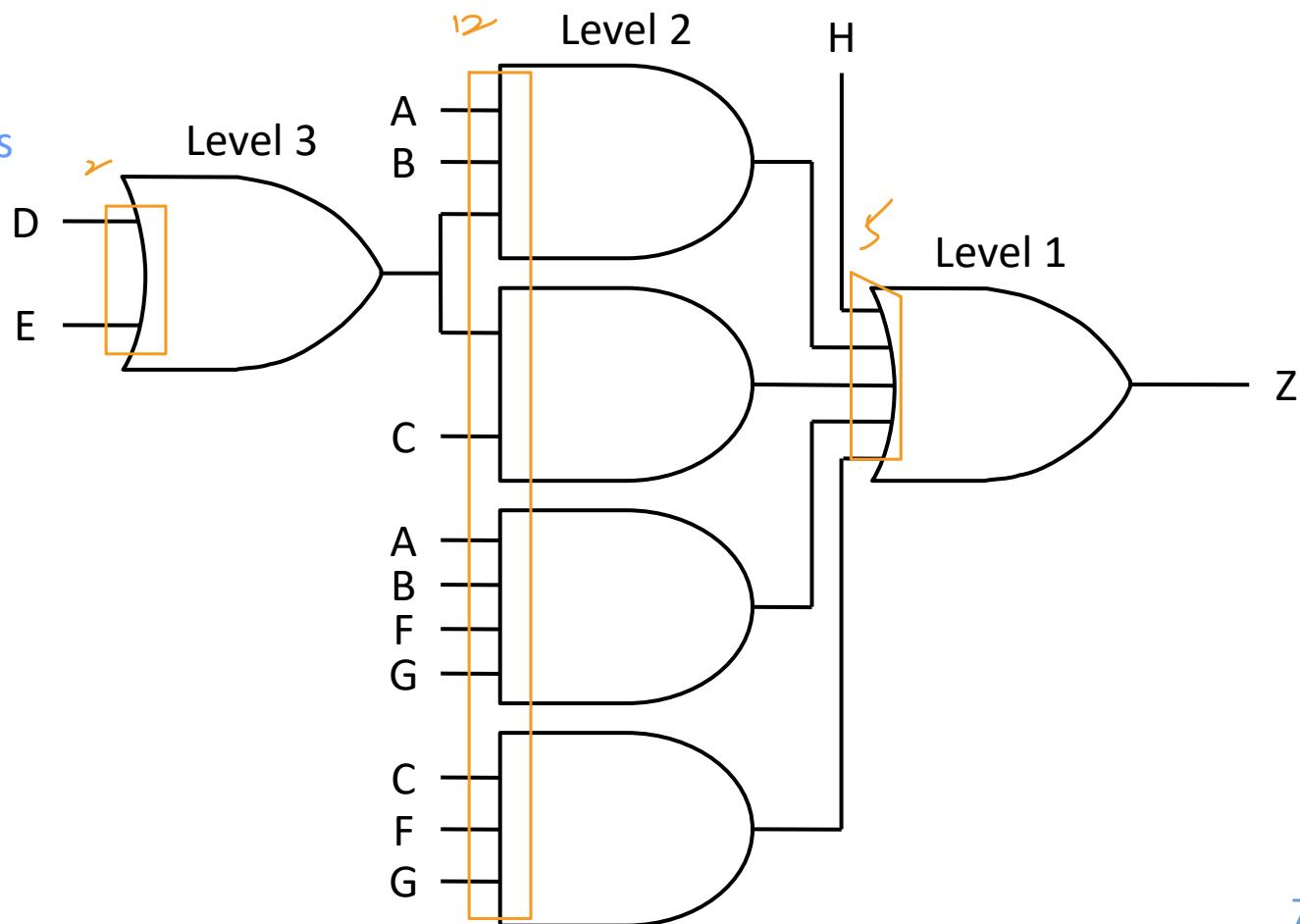
# One More Example (2/2)

## □ Same function in another realization

➤  $Z = AB(D + E) + C(D + E) + ABFG + CFG + H$

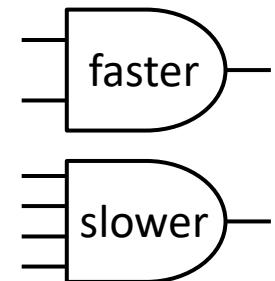
- 3 levels
- 6 gates
- 19 gate inputs

$\equiv$   
 $D \oplus E$



# Multi-Level Gate Circuits

- ❑ Q1: Why does the # of levels matter?
- ❑ A1: The # of levels affects cost and delay
  - Cost depends on # of gates & # of gate inputs
  - Delay depends on # of levels & # of inputs for a single gate
  - *trade-off* ➤ # of levels ↑: cost ↓ (maybe) and circuit delay ↑ (maybe)



- ❑ Q2: Any application for the level of each gate?
- ❑ A2: Evaluation according to the level in nonincreasing order
  - Level 4  $\rightarrow$  Level 3  $\rightarrow$  Level 2  $\rightarrow$  Level 1  
*(↑ delay, output)*
- ❑ Q3: How to change the # of levels?
- ❑ A3: Factoring or multiplying out

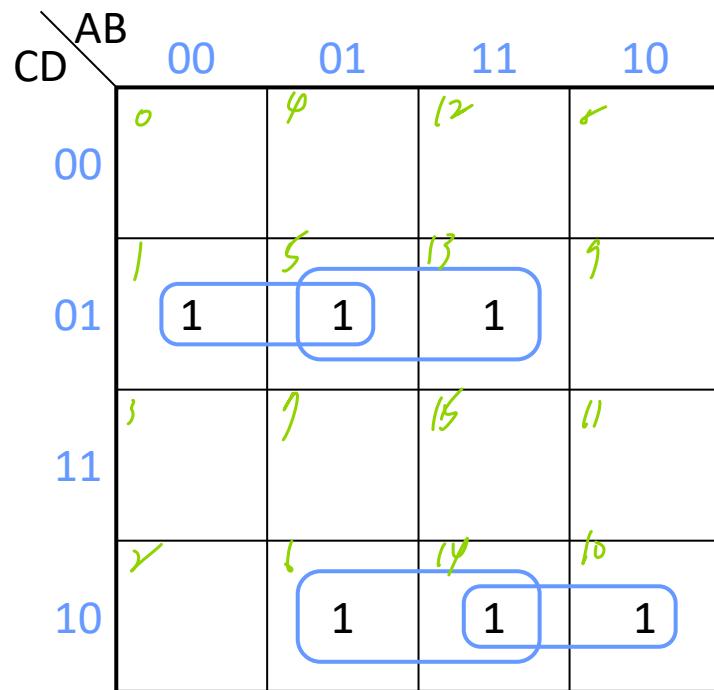
# Multi-Level Design with AND & OR (1/4)

- Find a circuit of AND and OR gates to realize

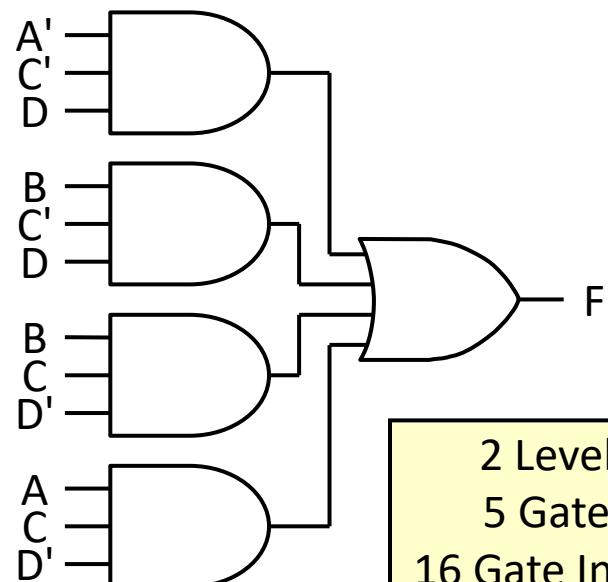
- $F(A,B,C,D) = \sum m(1, 5, 6, 10, 13, 14)$

- Solution 1: try two-level AND-OR circuit first

- Simplify F by K-map (circle 1's)
- Realize it



$$F = A'C'D + BC'D + BCD' + ACD'$$



# Multi-Level Design with AND & OR (2/4)

## ❑ From Solution 1

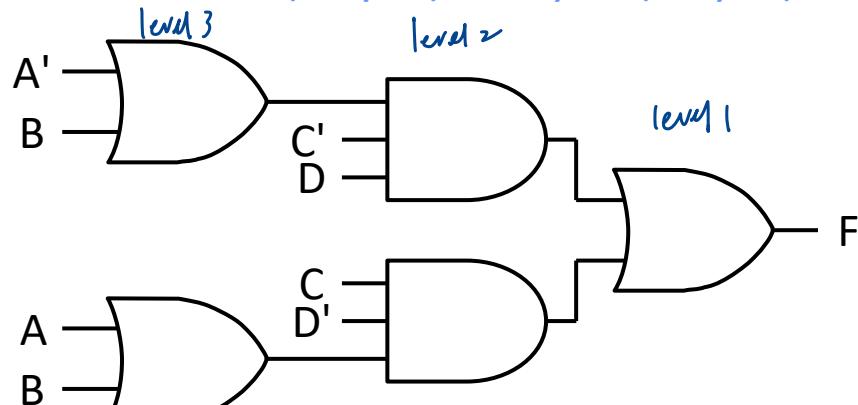
➤  $F = A'C'D + BC'D + BCD' + ACD'$

## ❑ Solution 2 *↙ solution 1 factoring, 4 inputs - 4 levels → levels > 3 levels*

➤ Factoring it yields  $F = (A' + B)C'D + (B + A)CD'$

➤ Lead to a three-level OR-AND-OR circuit

- # of gate inputs  $\downarrow$ , cost  $\downarrow$  (maybe), delay  $\uparrow$  (maybe)



3 Levels
5 Gates
12 Gate Inputs

*# Gates ↑* ↙ *# Gate Input ↓*

# Multi-Level Design with AND & OR (3/4)

- Find a circuit of AND and OR gates to realize

- $F(A,B,C,D) = \sum m(1, 5, 6, 10, 13, 14)$

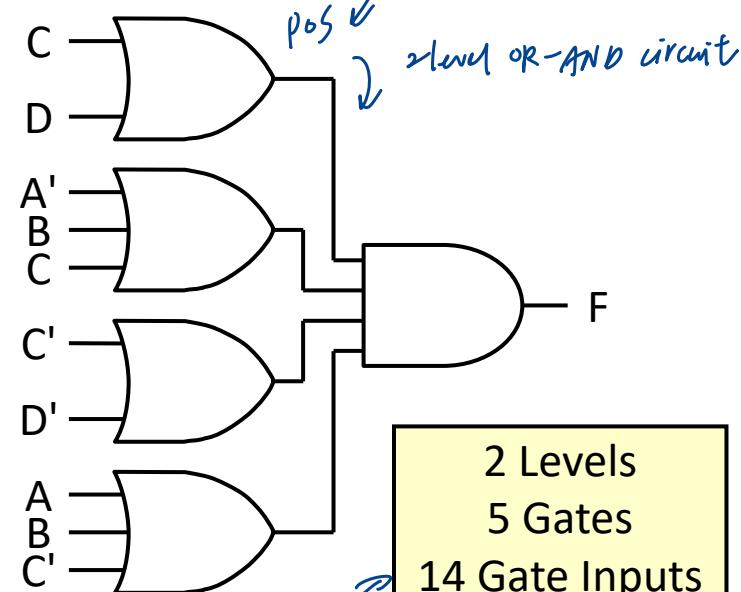
- Solution 3: try two-level OR-AND circuit first

- Simplify F by K-map (circle 0's)

- Realize it

CD \ AB	00	01	11	10
00	0	0	0	0
01	1	1	1	0
11	0	0	0	0
10	0	1	1	1

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



2 Levels  
5 Gates  
14 Gate Inputs

AND-OR = 16 # gate inputs 11

# Multi-Level Design with AND & OR (4/4)

## ❑ From Solution 3

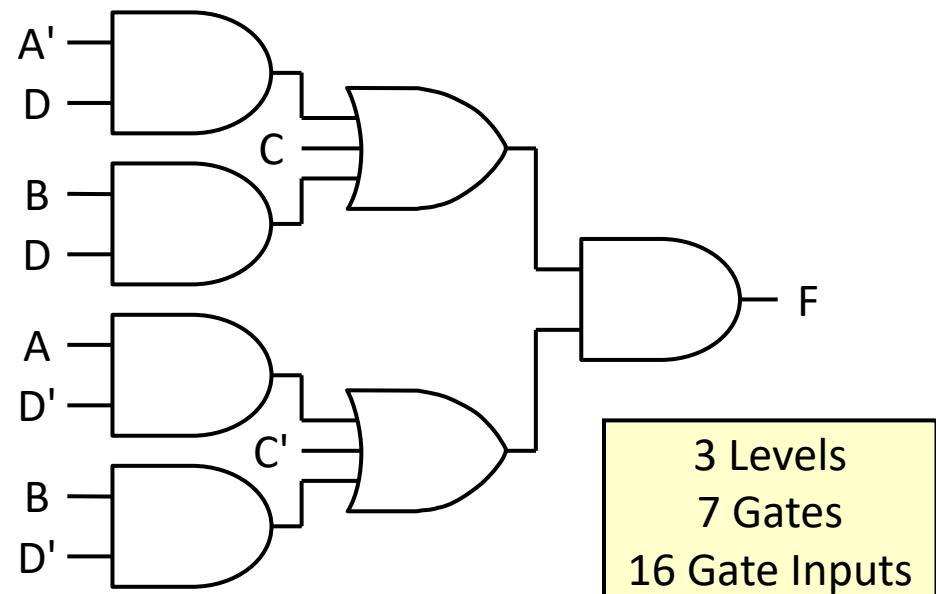
$$\triangleright F = (C + D)(A' + B + C)(C' + D')(A + B + C')$$

## ❑ Solution 4

- Multiplying it out yields  $F = [C + \underline{D(A' + B)}][C' + \underline{D'(A + B)}]$  (4-level) 未簡.
- Partially multiplying out yields  $F = (C + \underline{A'D + BD})(C' + \underline{AD' + BD'})$
- Lead to a three-level AND-OR-AND circuit

## ❑ Best solutions for F

- Two-level: OR-AND
  - (Solution 3) 3, 1, 1b
- Three-level: OR-AND-OR
  - (Solution 2) 3, 5, 12



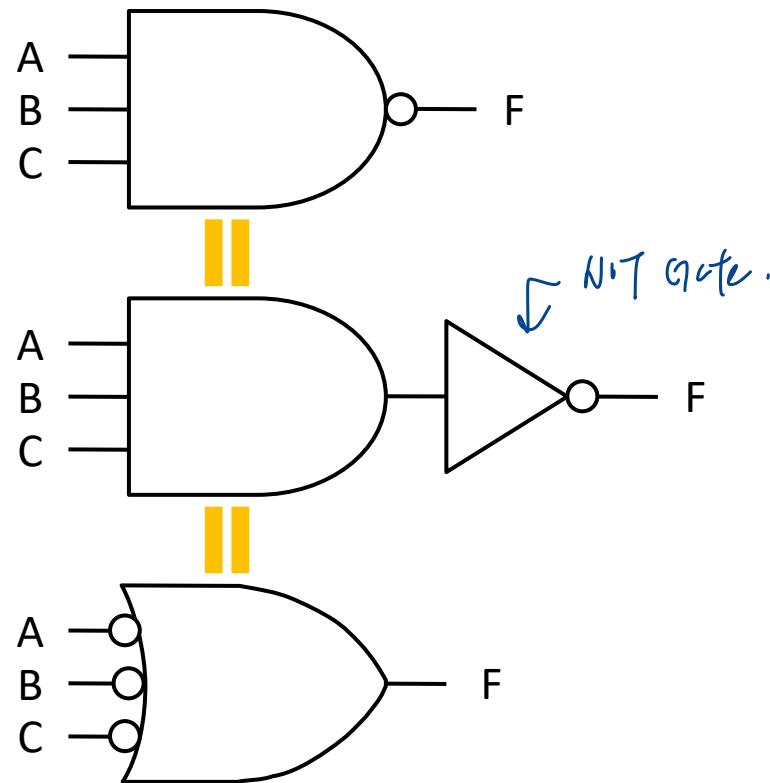
# Outline

- ❑ Multi-Level Gate Circuits
- ❑ **NAND and NOR Gates**
- ❑ Two-Level NAND- and NOR-Gate Circuits
- ❑ Multi-Level NAND- and NOR-Gate Circuits
- ❑ Circuit Conversion Using Alternative Gate Symbols
- ❑ Design of Two-Level, Multiple-Output Circuits
- ❑ Multiple-Output NAND and NOR Circuits

# NAND Gate

## □ NAND $\equiv$ AND-NOT

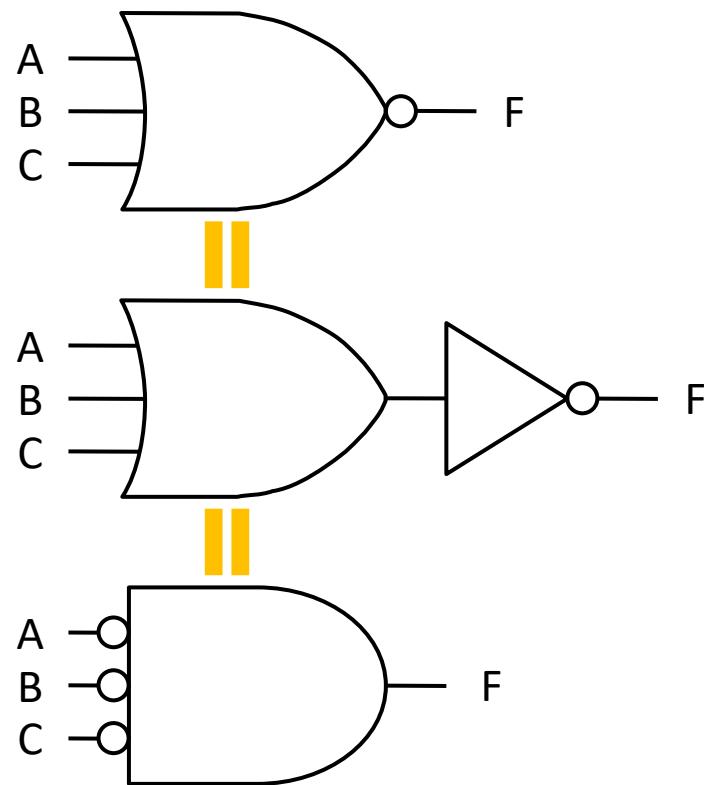
- DeMorgan's law:  $F = (ABC)' = A' + B' + C'$
- General:  $F = (X_1 X_2 \dots X_n)' = X_1' + X_2' + \dots + X_n'$
- Symbol: Bubble  $\equiv$  NOT



# NOR Gate

## □ NOR $\equiv$ OR-NOT

- DeMorgan's law:  $F = (A + B + C)' = A'B'C'$
- General:  $F = (X_1 + X_2 + \dots + X_n)' = X_1'X_2'\dots X_n'$
- Symbol: Bubble  $\equiv$  NOT



# NAND & NOR Gates

- ❑ Actually, NAND & NOR are implemented by fewer switch devices and operate faster than AND & OR gates

- Lower cost      *因为 NAND, NOR 实际上是通过较少的开关元件实现的，比 AND, OR 门快*
- Smaller delay

Gate Type	AND form	OR form
NAND gate	 AND symbol with a bubbled output	 OR symbol with bubbled inputs
NOR gate	 AND symbol with bubbled inputs	 OR symbol with a bubbled output
AND gate	 AND symbol	 All bubbled OR symbol
OR gate	 All bubbled AND symbol	 OR symbol

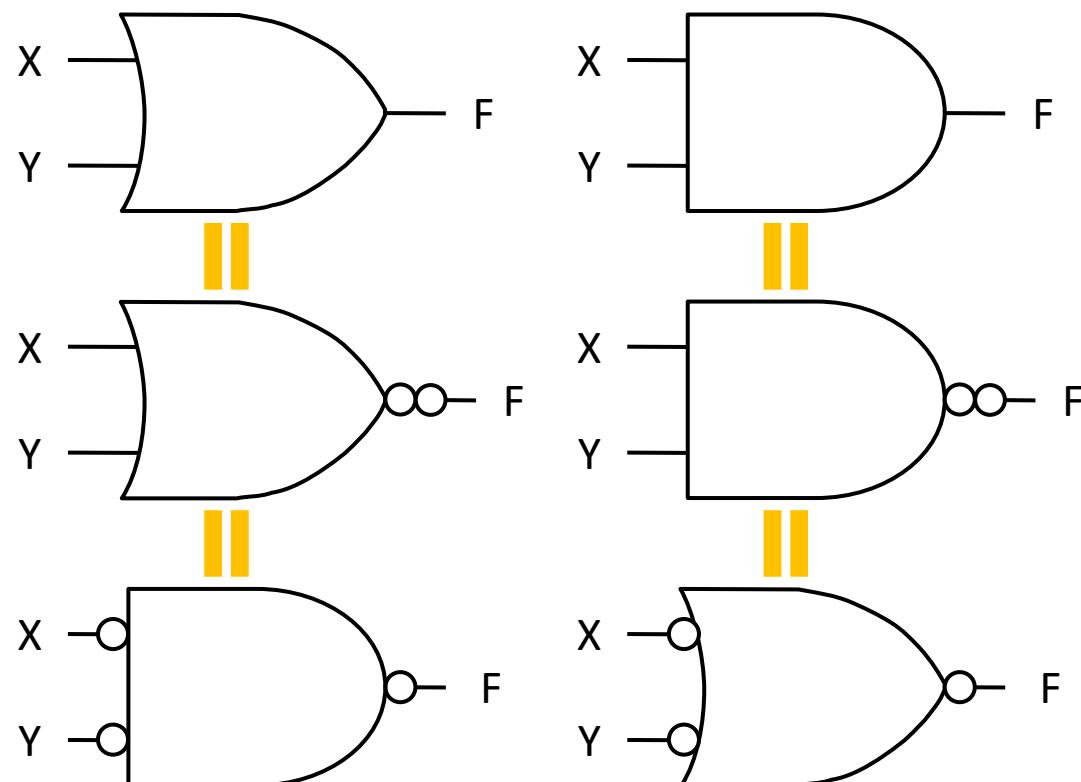
# Functionally Complete Set (1/2)

任何一個 Boolean function 都可以用這個 set 表示  $\Rightarrow$  functionally complete

- A set of logic operations is functionally complete if any Boolean function can be expressed by this set

## □ Examples

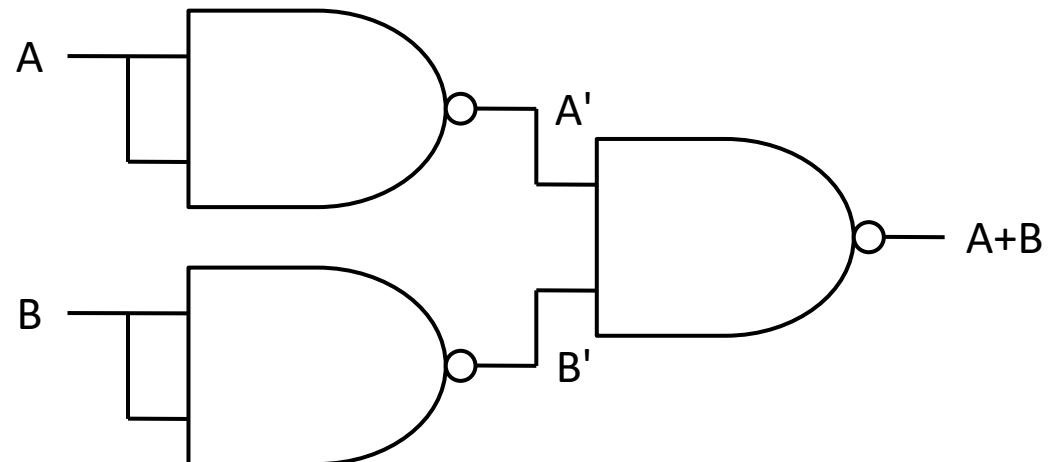
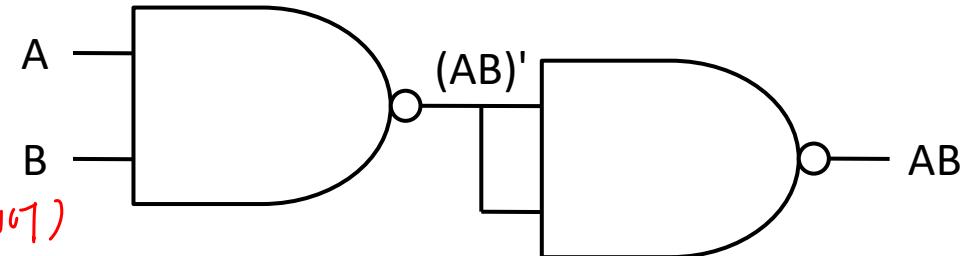
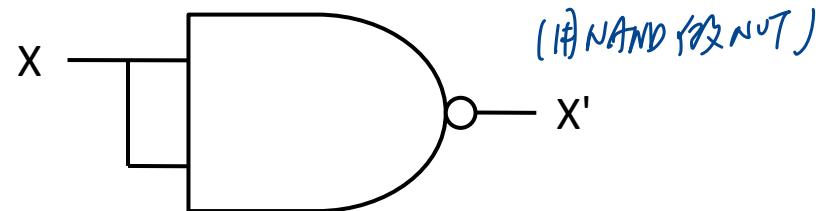
- Q1: {AND, OR, NOT}
- A1: Of course, yes!
- Q2: {AND, NOT}
- A2: Yes (check figure)
  - $X + Y = (X + Y)'' = (X'Y')$
- Q3: {OR, NOT}
- A3: Yes (check figure)
  - $XY = (XY)'' = (X' + Y')'$
- ✗ Q4: {AND}
- ✗ Q5: {OR}



# Functionally Complete Set (2/2)

## ❑ Examples

- Q6: {NAND}
- A6: Yes (check figure)
- Q7: {NOR}
- A7: Yes
- ✗ ➤ Q8: {AND, OR} *沒有 NOT.*  
*(NAND, NOR 也沒有 NOT.)*



# Outline

- ❑ Multi-Level Gate Circuits
- ❑ NAND and NOR Gates
- ❑ Two-Level NAND- and NOR-Gate Circuits**
- ❑ Multi-Level NAND- and NOR-Gate Circuits
- ❑ Circuit Conversion Using Alternative Gate Symbols
- ❑ Design of Two-Level, Multiple-Output Circuits
- ❑ Multiple-Output NAND and NOR Circuits

# Recap: DeMorgan's Law

## □ General

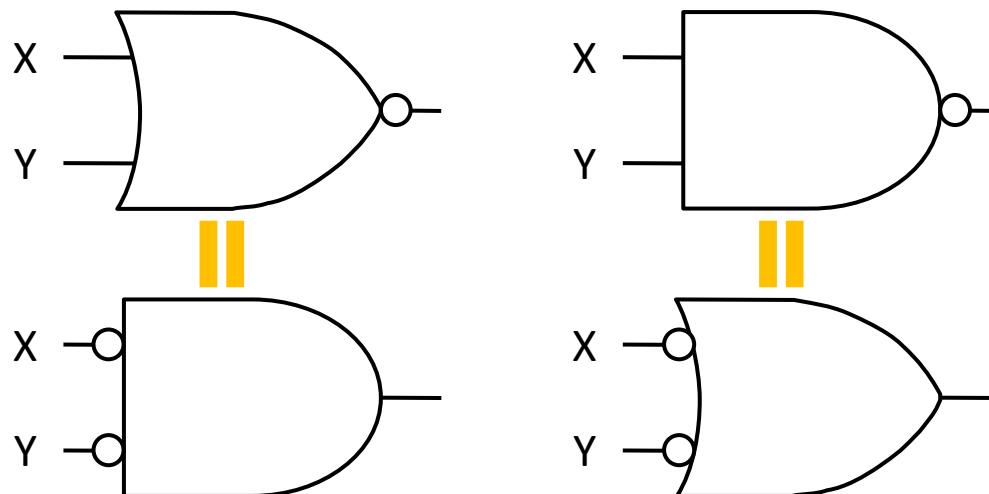
- $(X_1 + X_2 + \dots + X_n)' = X_1'X_2'\dots X_n'$
- $(X_1X_2\dots X_n)' = X_1' + X_2' + \dots + X_n'$

## □ One-step rule

- $[F(X_1, X_2, \dots, X_n, 0, 1, +, \bullet)]' = F(X_1', X_2', \dots, X_n', 1, 0, \bullet, +)$ 
  - $X \leftrightarrow X'; + \leftrightarrow \bullet; 0 \leftrightarrow 1$

## □ Idea    *NAND $\leftrightarrow$ NOR : dual*

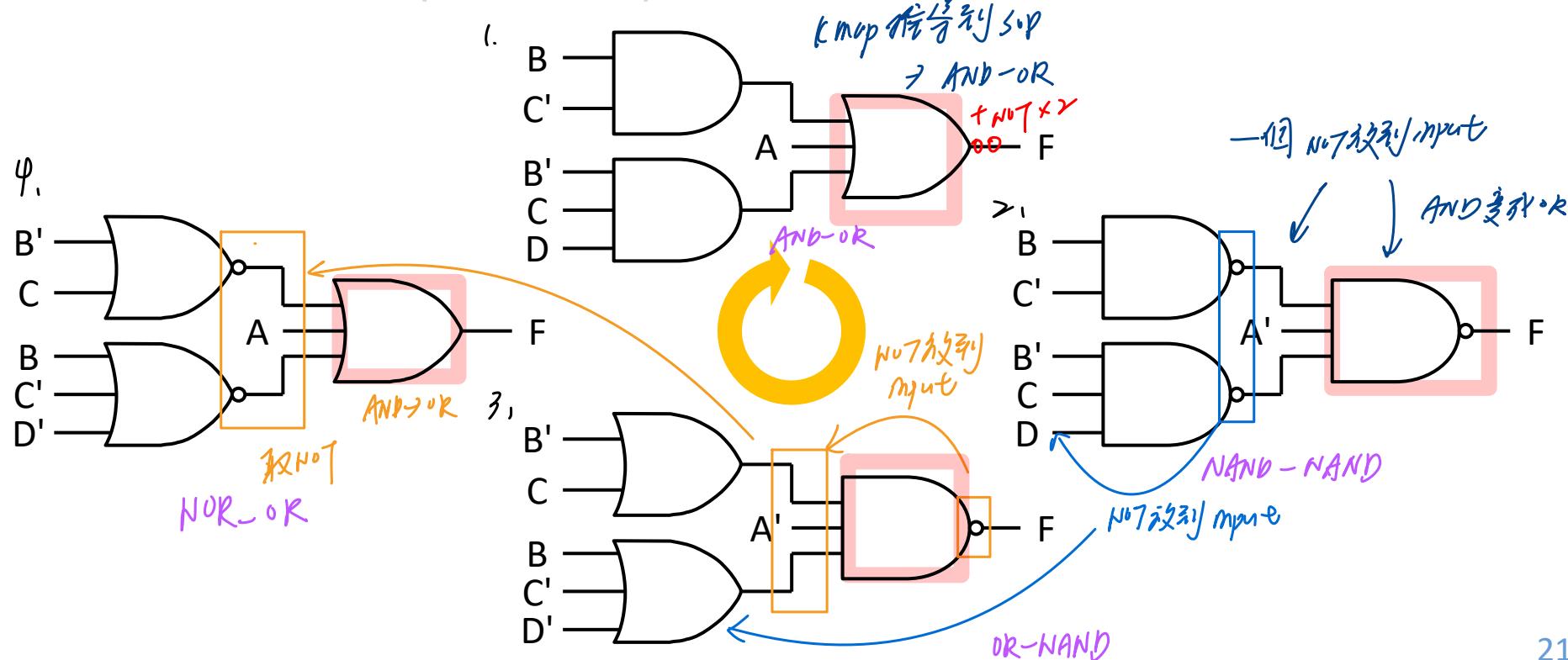
- Move "bubbles" and exchange AND gates and or gates



# 8 Basic Forms for 2-Level Circuits (1/2)

- AND-OR  $\leftrightarrow$  AND-(NOT-NAND)  $\leftrightarrow$  NAND-NAND
- NAND-NAND  $\leftrightarrow$  (NOT-OR)-NAND  $\leftrightarrow$  OR-NAND
- OR-NAND  $\leftrightarrow$  OR-(NOT-OR)  $\leftrightarrow$  NOR-OR
- NOR-OR  $\leftrightarrow$  (NOT-AND)-OR  $\leftrightarrow$  AND-OR

(S.o.P)  
Level 1  
OR, NAND



題目要求做 2-level circuit  $\Rightarrow$  有 level 1 之程度  
手寫字從 POS / SOP 計算

# 8 Basic Forms for 2-Level Circuits (2/2)

OR-AND  $\leftrightarrow$

OR-(NOT-NOR)  $\leftrightarrow$

NOR-NOR  $\leftrightarrow$

(NOT-AND)-NOR  $\leftrightarrow$

AND-NOR  $\leftrightarrow$

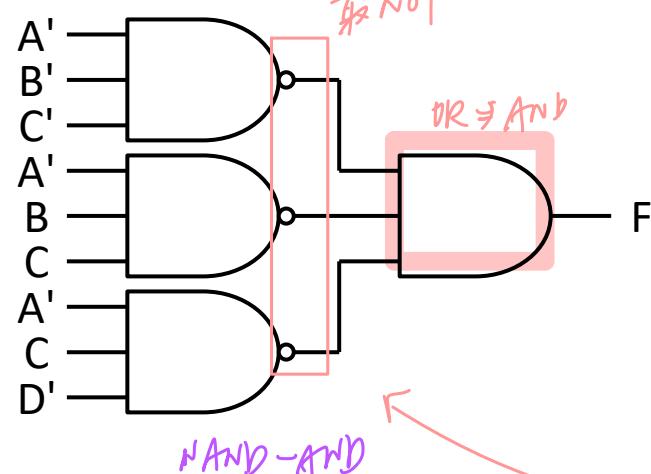
AND-(NOT-AND)  $\leftrightarrow$

NAND-AND  $\leftrightarrow$

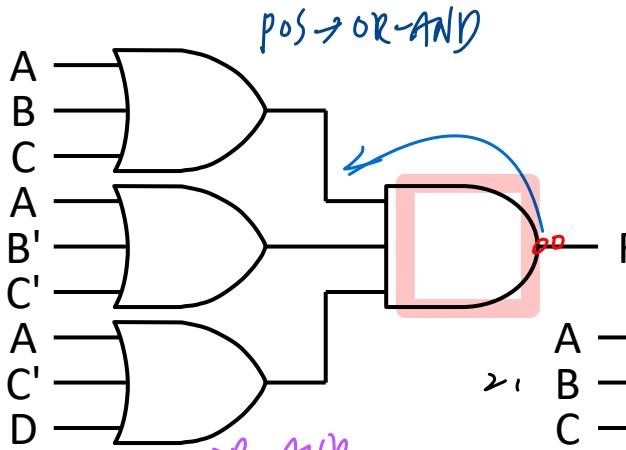
(NOT-OR)-AND  $\leftrightarrow$

OR-AND

4.

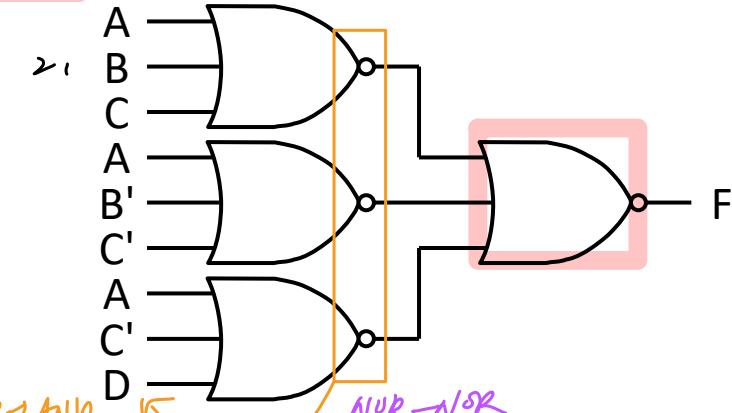


1.

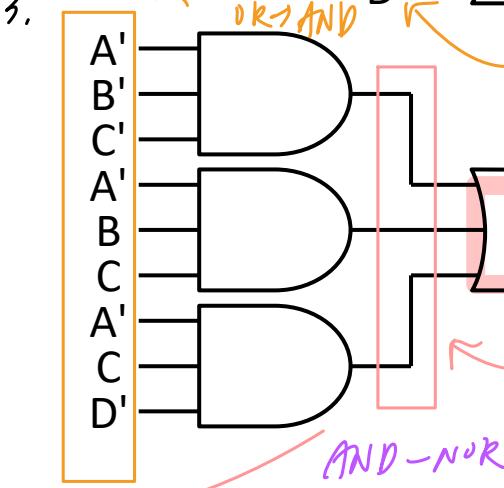


OR-AND

2.



3.



# Other Forms

2nd. 用 AND, OR, NAND, NOR → 可以組合出 16 種，但前面只講了 8 個 basic forms

- Other 8 forms are degenerated, i.e., cannot realize all  
switching functions
- AND-AND, AND-NAND  
    ➤ OR-OR, OR-NOR  
    ➤ NAND-OR, NAND-NOR  
    ➤ NOR-AND, NOR-NAND
- 剩下 8 種  
 $\Rightarrow$  degenerated

- Example

- NAND-NOR  $\equiv$  AND-AND...

# Example: NAND-NAND & NOR-NOR

## ❑ Procedure for designing a min 2-level NAND-NAND circuit

- Find a minimum SOP
- Draw AND-OR (2-level) circuit
- Convert into NAND-NAND circuit
  - Introduce 2 bubbles into each intermediate line or the output

## ❑ Procedure for designing a min 2-level NOR-NOR circuit

- Find a minimum POS
- Draw OR-AND (2-level) circuit
- Convert into NOR-NOR circuit
  - Introduce 2 bubbles into each intermediate line or the output

## ❑ Examples

- $F = A + BC' + B'CD$
- $F = (A + B + C)(A + B' + C')D$

# Summary: 2-Level Logic Minimization

- ❑ Start with minimum SOP (AND-OR) or minimum POS (OR-AND)
  - Simplified: cost = (# of terms, # of literals) (Units 2--5)
  - Real: cost = (# of gates, # of gate inputs)
    - $\# \text{ of gates} \leq \# \text{ of terms} + 1$
    - $\# \text{ of gate inputs} \leq \# \text{ of literals} + \# \text{ of terms}$
    - " $<$ " holds if a term is degenerated to a single variable
- ❑ Convert the minimum SOP (POS) to the target form
  - 8 basic forms can convert each other
    - DeMorgan's law
    - Use bubbles
- ❑ NAND/NOR gates: low cost & high speed

# Outline

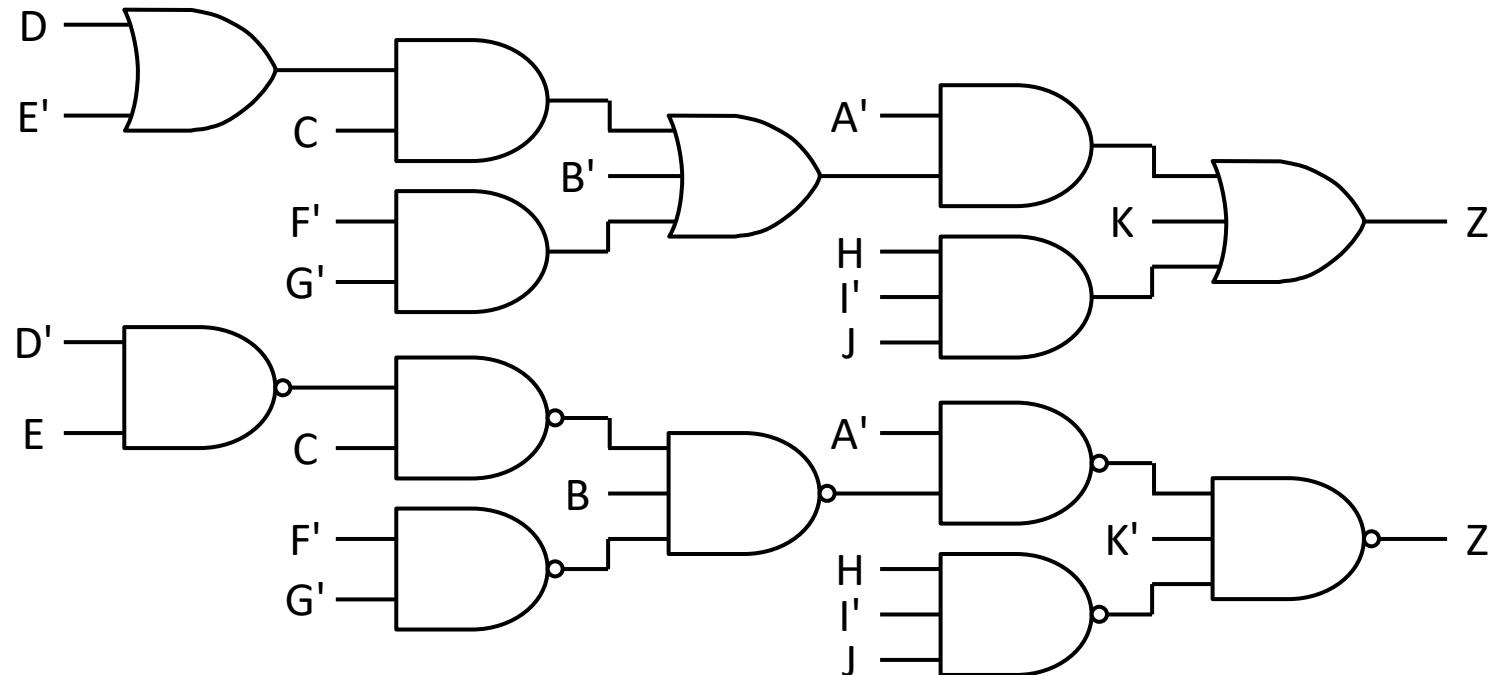
- ❑ Multi-Level Gate Circuits
- ❑ NAND and NOR Gates
- ❑ Two-Level NAND- and NOR-Gate Circuits
- ❑ Multi-Level NAND- and NOR-Gate Circuits**
- ❑ Circuit Conversion Using Alternative Gate Symbols
- ❑ Design of Two-Level, Multiple-Output Circuits
- ❑ Multiple-Output NAND and NOR Circuits

# Multi-Level NAND-Gate Circuits

## ❑ Procedure for designing a multi-level NAND circuit

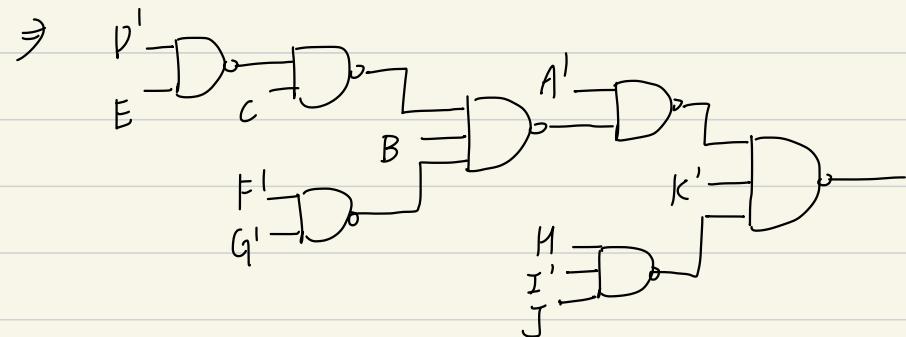
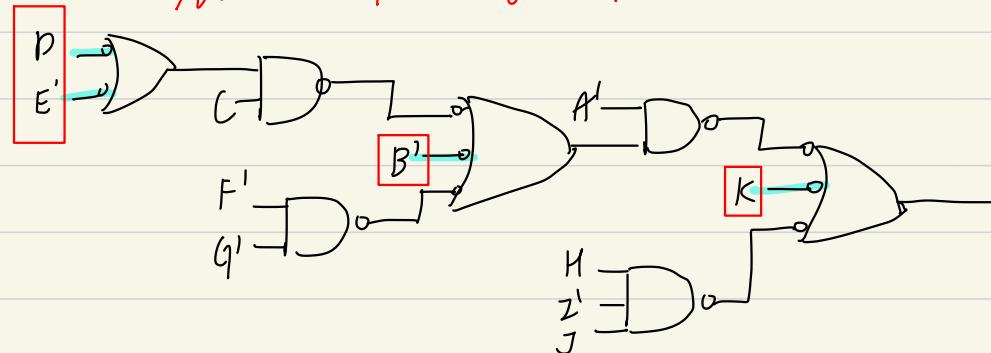
- Simplify the switching function
- Draw its multi-level AND-OR circuit
- Replace all gates with NANDs and number all levels (from the output)
- Invert inputs to levels 1, 3, 5, ...

## ❑ Example: $Z = A'[B' + C(D + E') + F'G'] + HI'J + K$



若有 inverted input  $\Rightarrow$  記得取 complement!

詳 p. 12 ~ 13

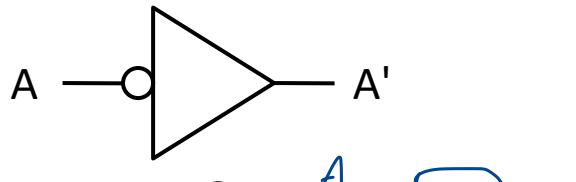


# Outline

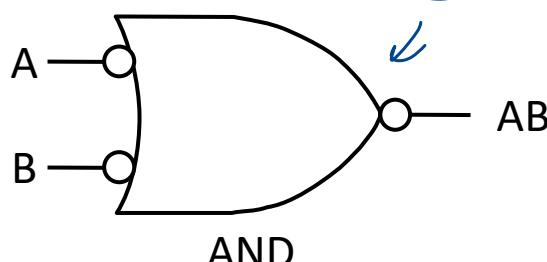
- ❑ Multi-Level Gate Circuits
- ❑ NAND and NOR Gates
- ❑ Two-Level NAND- and NOR-Gate Circuits
- ❑ Multi-Level NAND- and NOR-Gate Circuits
- ❑ Circuit Conversion Using Alternative Gate Symbols**
- ❑ Design of Two-Level, Multiple-Output Circuits
- ❑ Multiple-Output NAND and NOR Circuits

# Alternative Gate Symbols

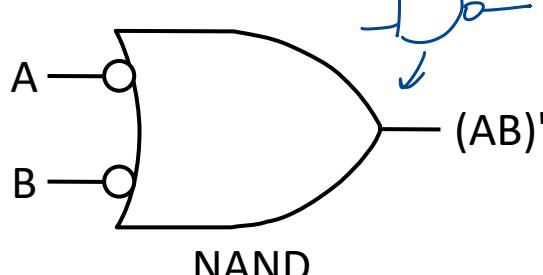
- ❑ Do NOT memorize these gate symbols
- ❑ DeMorgan's law is the key of conversion
  - Use bubbles adequately



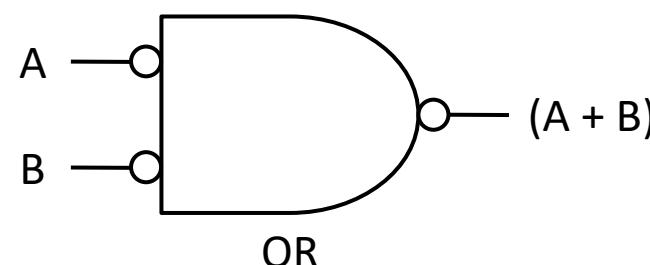
NOT



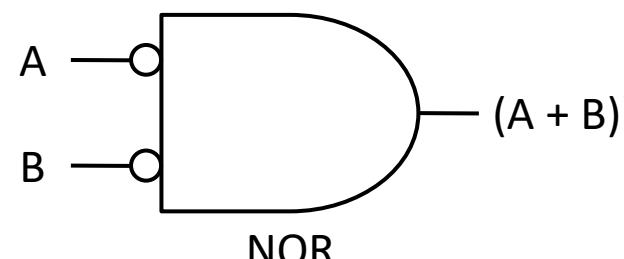
AND



NAND

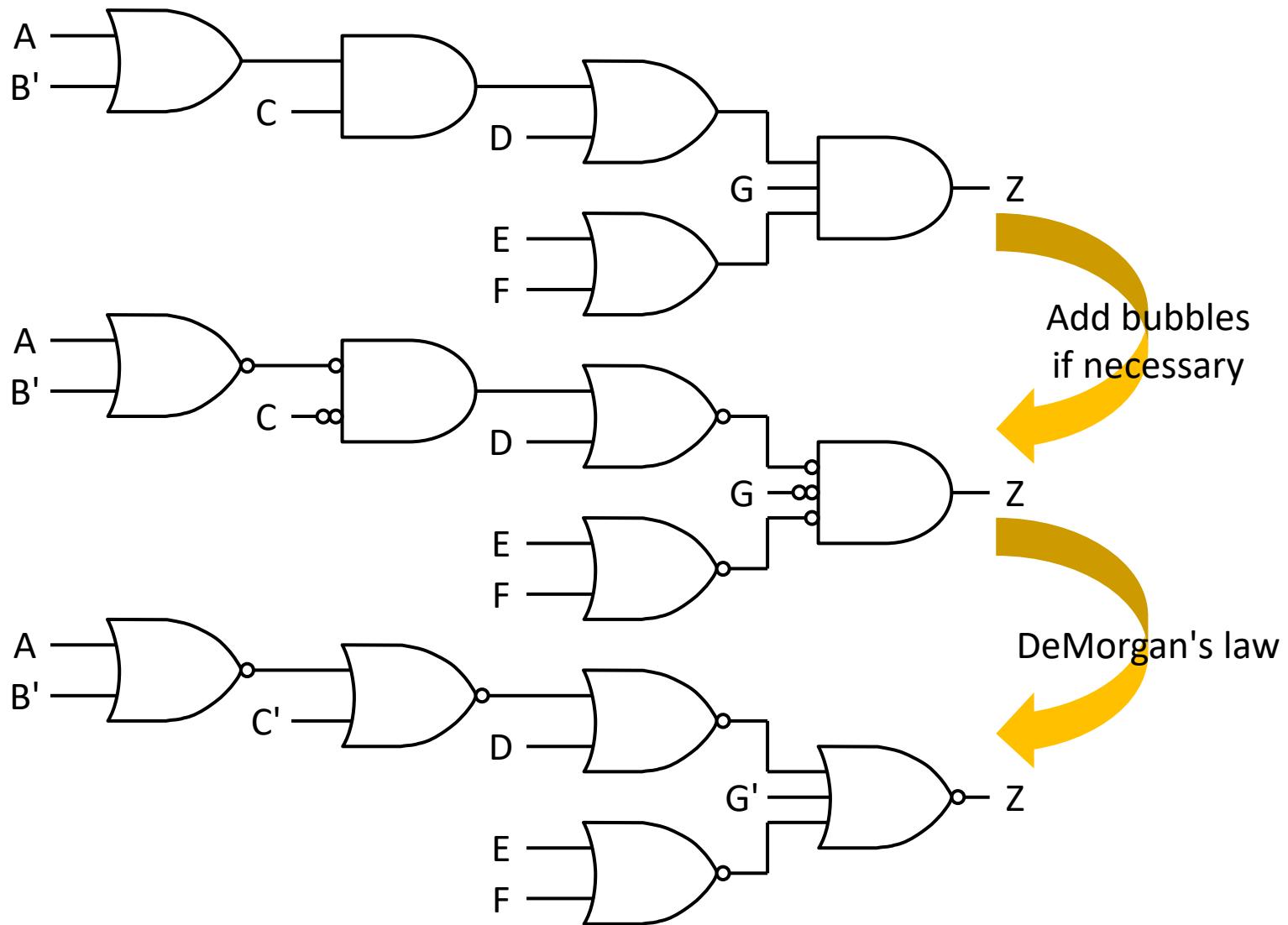


OR

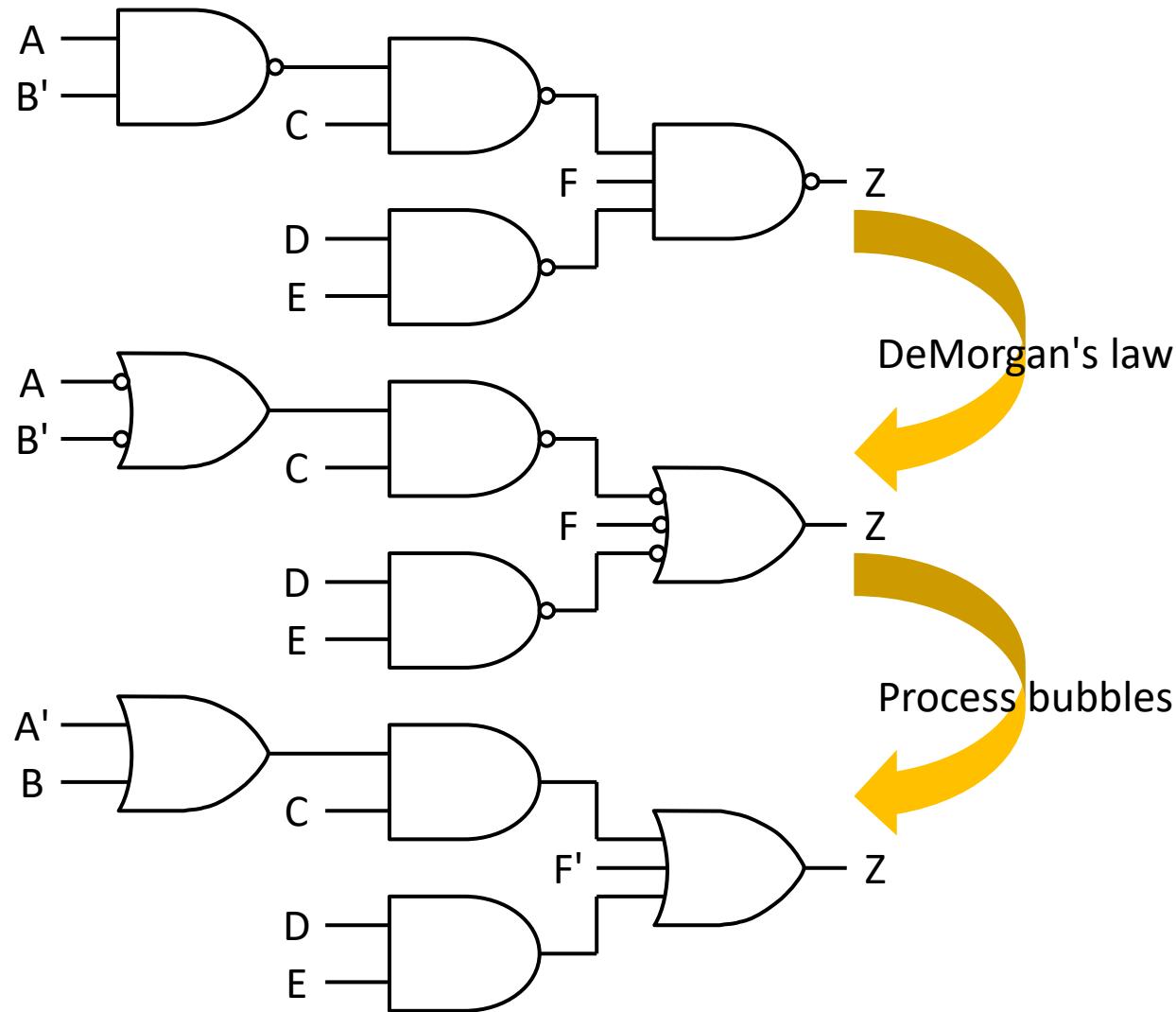


NOR

# Example from OR-AND to NOR Gates



# Example from NAND to AND-OR Gates



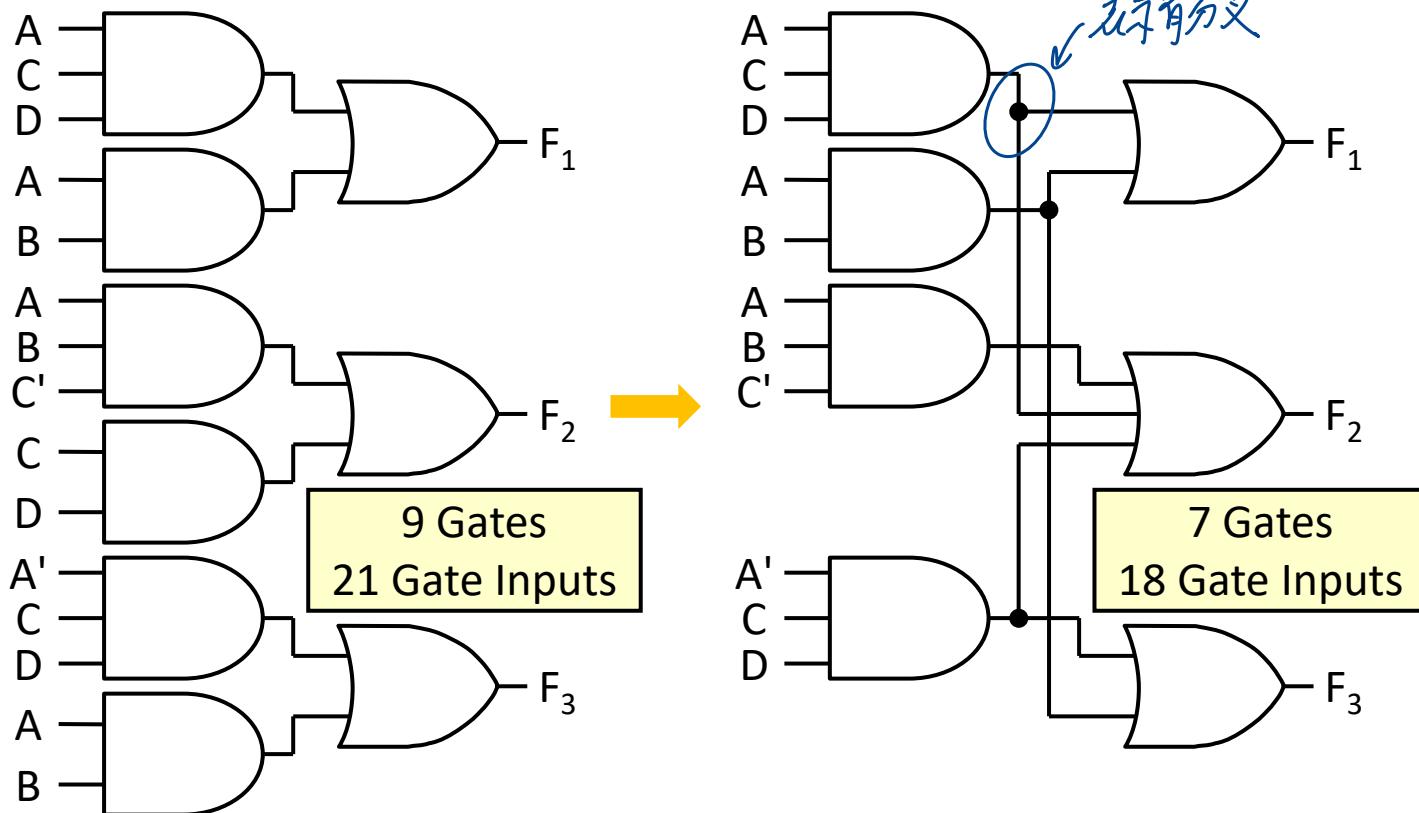
# Outline

- ❑ Multi-Level Gate Circuits
- ❑ NAND and NOR Gates
- ❑ Two-Level NAND- and NOR-Gate Circuits
- ❑ Multi-Level NAND- and NOR-Gate Circuits
- ❑ Circuit Conversion Using Alternative Gate Symbols
- ❑ Design of Two-Level, Multiple-Output Circuits**
- ❑ Multiple-Output NAND and NOR Circuits

# Case 1: 4 Inputs and 3 Outputs

## □ Example

- $F_1(A, B, C, D) = \sum m(11, 12, 13, 14, 15) = ACD + AB$
- $F_2(A, B, C, D) = \sum m(3, 7, 11, 12, 13, 15) = ABC' + \underline{CD} = ABC' + \boxed{ACD + A'CD}$
- $F_3(A, B, C, D) = \sum m(3, 7, 12, 13, 14, 15) = \underline{A'CD} + AB$



# Case 2: 4 Inputs and 3 Outputs (1/2)

## □ Example

- $F_1(A,B,C,D) = \sum m(2, 3, 5, 7, 8, 9, 10, 11, 13, 15) = BD + B'C + AB'$
- $F_2(A,B,C,D) = \sum m(2, 3, 5, 6, 7, 10, 11, 14, 15) = C + A'BD$
- $F_3(A,B,C,D) = \sum m(6, 7, 8, 9, 13, 14, 15) = BC + AB'C' + ABD$  (or  $AC'D$ )

AB	CD	00	01	11	10
CD	00				1
00					
01		1	1		1
11	1	1	1		1
10	1				1

AB	CD	00	01	11	10
CD	00				
00					
01			1		
11		1	1	1	1
10		1	1	1	1

AB	CD	00	01	11	10
CD	00				1
00					
01				1	
11		1	1	1	1
10		1	1	1	1

10 Gates  
25 Gate Inputs

# Case 2: 4 Inputs and 3 Outputs (2/2)

## □ Find common minterms from K-map

- $F_1(A,B,C,D) = \underline{BD} + \underline{B'C} + \underline{AB'} = \underline{\text{ABD}} + \underline{A'BD} + B'C + \underline{AB'C'}$
- $F_2(A,B,C,D) = C + \underline{A'BD}$
- $F_3(A,B,C,D) = BC + \underline{AB'C'} + \underline{ABD}$

AB		00	01	11	10
CD	00				1
	01	1	1	1	1
CD	11	1	1	1	1
	10	1			1

AB		00	01	11	10
CD	00				
	01		1		
CD	11	1	1	1	1
	10	1	1	1	1

AB		00	01	11	10
CD	00				1
	01			1	1
CD	11	1	1	1	1
	10	1	1	1	1

非 minimum SOP  
(没有圈最大的圆圈)

# Gates  
# Gate Inputs

8 Gates  
22 Gate Inputs

# Case 3: Essential Prime Implicants

multi-output

- ❑ Essential for multi-output

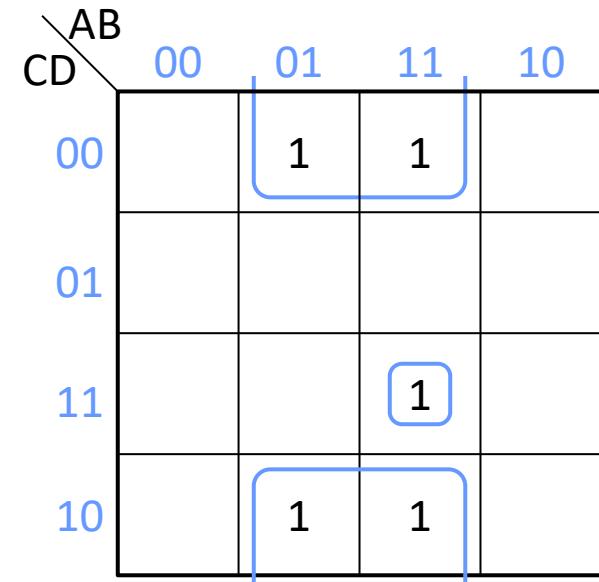
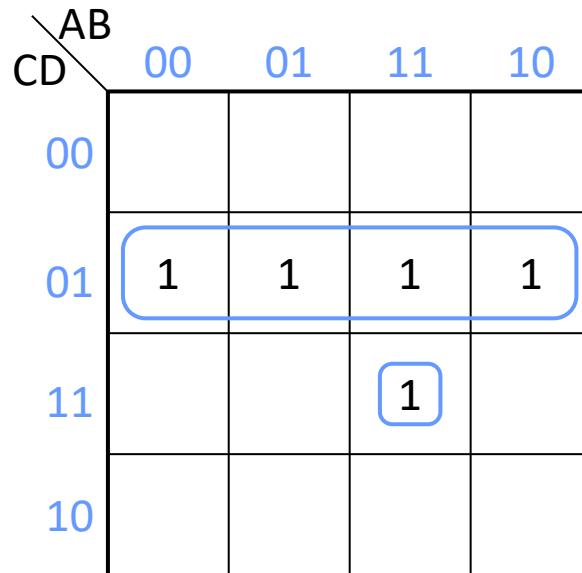
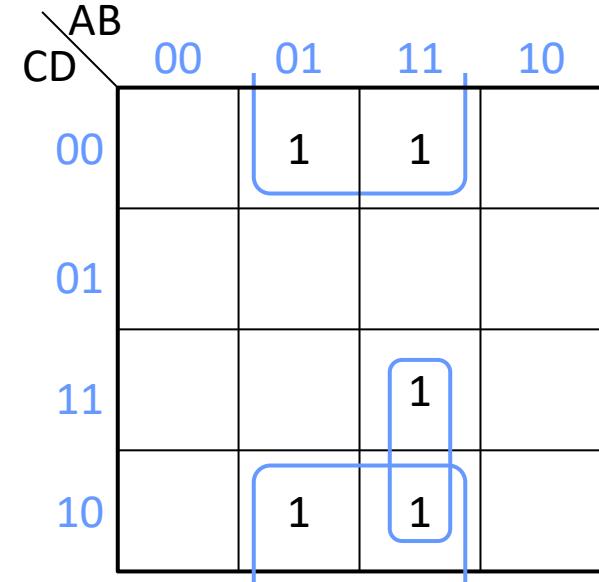
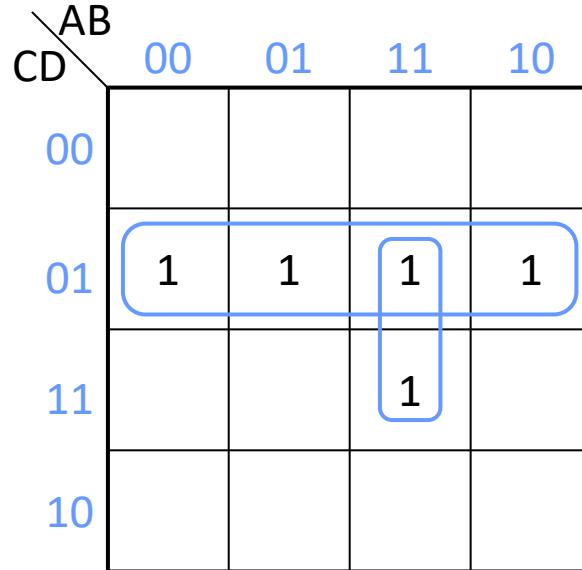
只找只  
essential  
系数

Only check 1's which do not appear in other maps

## ➤ Example

- $C'D$  and  $BD'$  are essential
- $ABD$  and  $ABC$  are not

Reduce one gate by sharing ABCD



有些 case 用原方法取最短的圖較好  
有些 case 用原方法取最短的圖較好

## Case 4



- Having more common terms is not always better

8 Gates  
26 Gate Inputs

AB	CD	00	01	11	10
00	1	1			
01		1			
11					
10	1	1	1		

AB	CD	00	01	11	10
00	1	1	1		
01		1			
11					
10		1	1		

AND OR  
~ AND ~ OR  
 $\therefore 3+2+1+1$

NOT這些 AND  
(結果是 15 來)

7 Gates  
18 Gate Inputs

AB	CD	00	01	11	10
00	1	1			
01		1			
11					
10	1	1	1		

AB	CD	00	01	11	10
00	1				
01		1	1		
11					
10		1	1		

$$\begin{array}{l} \textcircled{1} 2+3+3+3=11 \\ \textcircled{2} 3+6 \text{ 先 OR } \end{array}$$

$$\begin{array}{l} \textcircled{3} 3+2+2=7 \\ \textcircled{4} 2+4 \text{ 先 OR } \end{array}$$

# Summary: 2-Level Multiple-Output Circuits

## ❑ Cost depends on

- The total # of gates
- The total # of gate inputs

## ❑ Cost minimization

- A global view may help
  - A minimum SOP for each cannot guarantee minimum cost for all
- Common and essential terms may help
  - Sometimes, we do not need large circles
  - "Essential" is defined in a different way → 有时候多层共用反相形.
- Very difficult to find global optimal solution
  - Boolean algebra is very useful in more sophisticated techniques (advanced courses)

# Outline

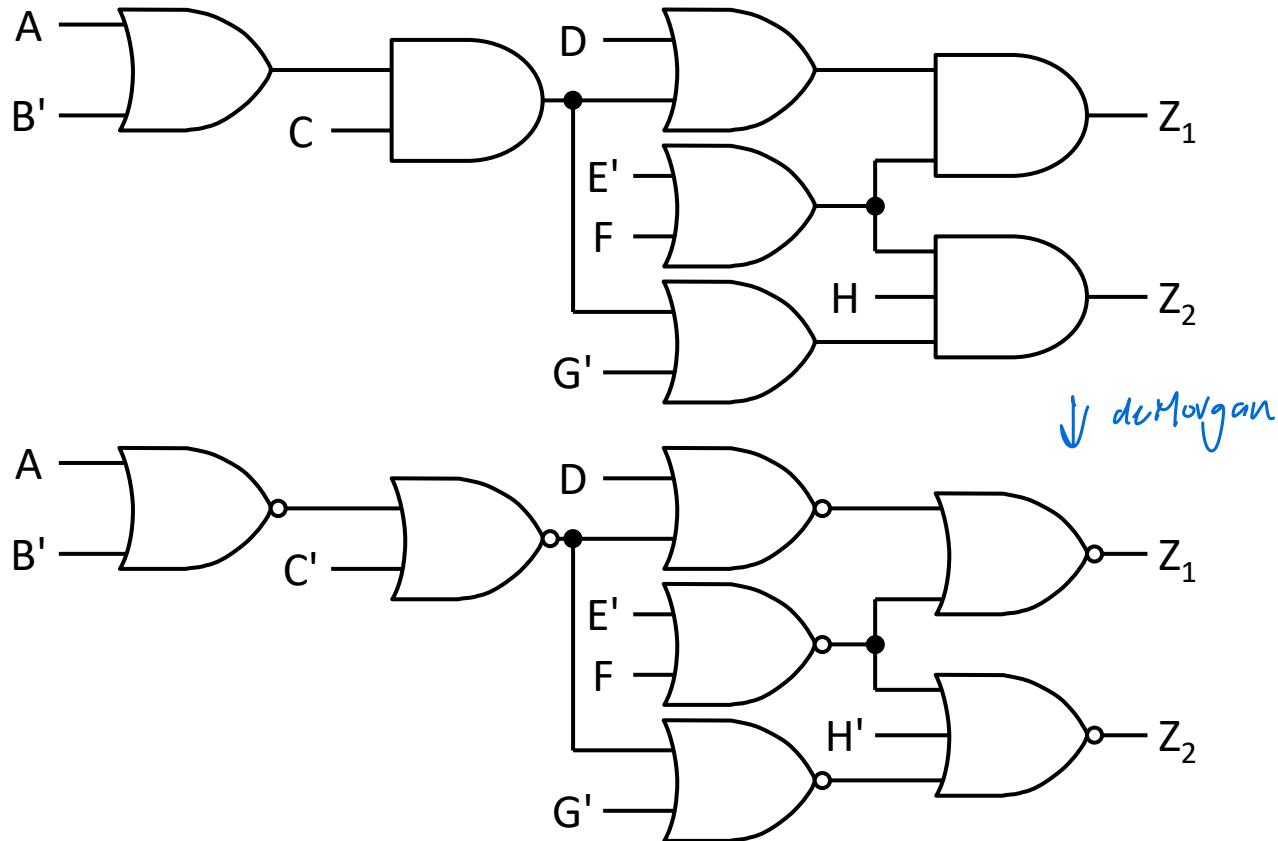
- ❑ Multi-Level Gate Circuits
- ❑ NAND and NOR Gates
- ❑ Two-Level NAND- and NOR-Gate Circuits
- ❑ Multi-Level NAND- and NOR-Gate Circuits
- ❑ Circuit Conversion Using Alternative Gate Symbols
- ❑ Design of Two-Level, Multiple-Output Circuits
- ❑ **Multiple-Output NAND and NOR Circuits**

# Multiple-Output NAND & NOR Circuits

{NAND}, {NOR} 算是功能上完全互补的  
且可以互换

- If all of the output gates are OR/AND gates, direct conversion to a NAND-/NOR-gate circuit is possible

➤ Example:  $Z_1 = [(A + B')C + D](E' + F)$  and  $Z_2 = [(A + B')C + G'](E' + F)H$



# Q&A