



UNIVERSITI  
TEKNOLOGI  
MALAYSIA  
[www.utm.my](http://www.utm.my)

## MODULE 5: COMBINATIONAL LOGIC CIRCUITS

SECR1013 DIGITAL LOGIC

FACULTY OF COMPUTING



## Objectives:

1. To introduce **AND-OR** and **AND-OR-Invert** logic
2. To illustrate the conversion between different digital circuit representations
3. To introduce the universality of gate
4. To introduce the dual symbol in digital circuit
5. To explain the design steps for combinational circuit

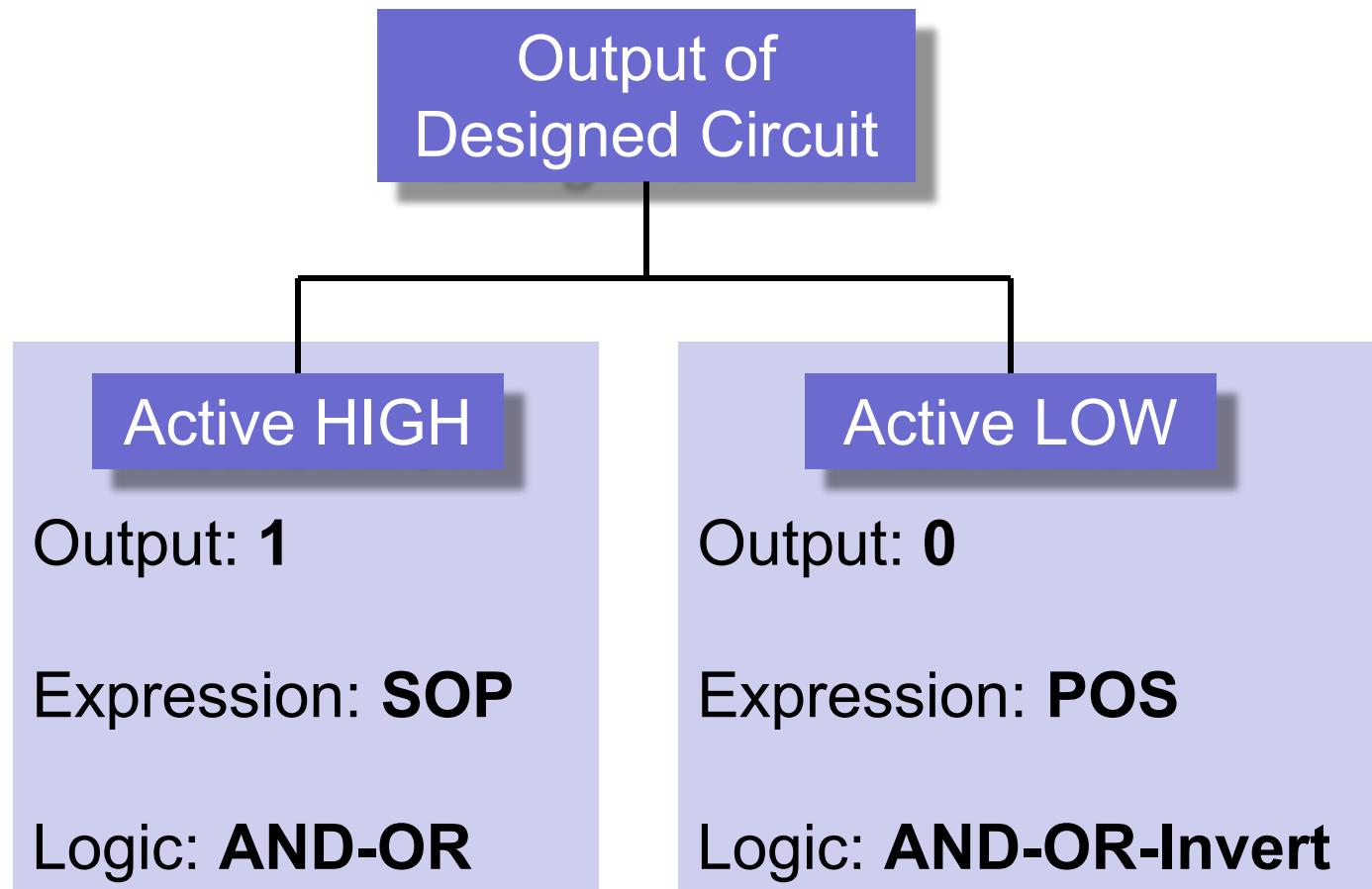


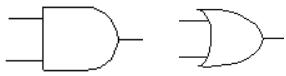
# Basic Combinational Gates

# Basic Combinational Logic

- **AND-OR** Logic
- **AND-OR-Invert** Logic
- Exclusive-OR (XOR) Logic
- Exclusive-NOR (XNOR) Logic

In designing a circuit, we can make the output to become **0** (LOW) or **1** (HIGH)

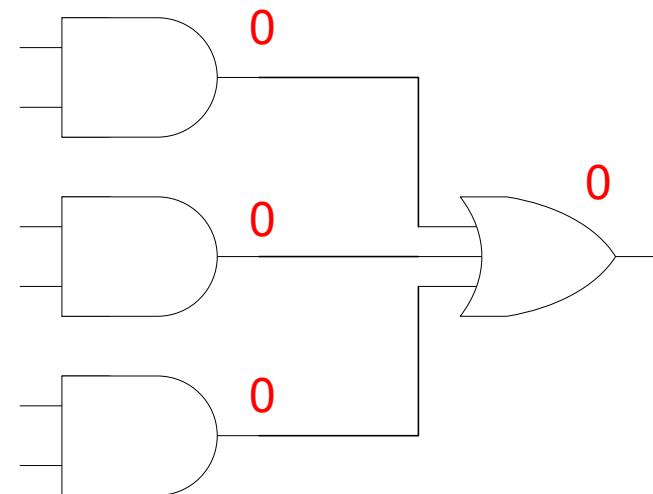




## AND-OR Logic

- A circuit consisting of any number of **AND** gates and an **OR** gate.
- Example: **SOP** expressions (e.g.,  $AB + CD + EF$ )

- If **any** of the **AND** gates output are **HIGH**, the output in the **OR** gate is **HIGH**
- If **all** of the **AND** gates outputs are **LOW**, the output in the **OR** gate is **LOW**



## Example: AND-OR Logic

Based on the truth table:

- derive the expression
- simplify using K-Map
- Implement using AND-OR logic

INPUTS						OUTPUT
A	B	C	D	AB	CD	X
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	0	1	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	0	1	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	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	1

## Solution #1:

- Write the Boolean expression or map the 1's to the K-Map

$$X = \overline{A}\overline{B}CD + \overline{A}\overline{B}CD + A\overline{B}CD + ABC\overline{D} + A\overline{B}CD + ABC\overline{D} + ABCD$$

- Simplify the output expression by using Boolean algebra or K-Map

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

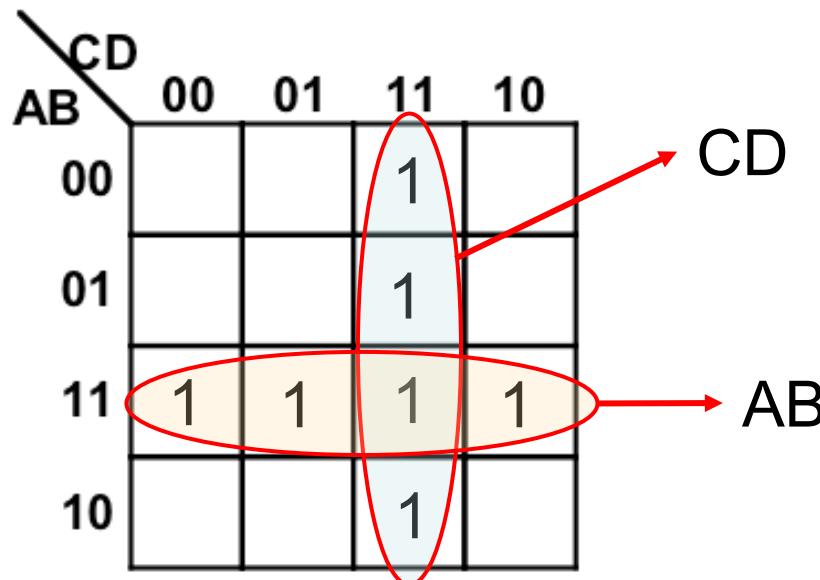
INPUTS				OUTPUT		X
A	B	C	D	AB	CD	
0	0	0	0	0	0	0
0	0	0	1	0	0	0
0	0	1	0	0	0	0
0	0	1	1	0	1	1
0	1	0	0	0	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	1

## Solution #1:

- Write the Boolean expression or map the 1's to the K-Map

$$X = \overline{A}\overline{B}CD + \overline{A}BC\overline{D} + A\overline{B}CD + AB\overline{C}\overline{D} + A\overline{B}CD + ABC\overline{D} + ABCD$$

- Simplify the output expression by using Boolean algebra or K -Map



INPUTS				OUTPUT		
A	B	C	D	AB	CD	X
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	0	1	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	0	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	1	0	1	1
1	1	0	0	1	0	1
1	1	0	1	1	0	1
1	1	1	0	1	0	1
1	1	1	1	1	1	1

## Solution #2:

Simplify the expression using Boolean algebra.

$$X = \overline{A}\overline{B}CD + \overline{A}BCD + A\overline{B}CD + AB\overline{C}\overline{D} + AB\overline{C}D + ABC\overline{D} + ABCD$$

**Solution #2:**

$$\begin{aligned} &= (\overline{A} + A)\overline{B}CD + \overline{A}BCD + ABC(\overline{C}(D + \overline{D}) + D(D + \overline{D})) \\ &= \overline{B}CD + \overline{A}BCD + ABC\overline{C} + ABC \\ &= \overline{B}CD + \overline{A}BCD + AB(\overline{C} + C) \\ &= \overline{B}CD + \overline{A}BCD + AB \\ &= \overline{B}CD + B(\overline{A}CD + A) \\ &= \overline{B}CD + B(CD + A) \\ &= \overline{B}CD + BCD + AB \\ &= (\overline{B} + B)CD + AB \\ &= AB + CD \end{aligned}$$

Rule 6

Rule 11

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

Simplify the expression using Boolean algebra.

$$X = \overline{A}\overline{B}CD + \overline{A}BCD + A\overline{B}CD + AB\overline{C}\overline{D} + ABC\overline{D} + ABCD + A\overline{B}CD$$

**Solution #2:**  $= (\overline{A} + A)\overline{B}CD + \overline{A}BCD + AB\overline{C}(D + \overline{D}) + ABC(D + \overline{D})$

$$= \overline{B}CD + \overline{A}BCD + AB\overline{C} + ABC$$

$$= \overline{B}CD + \overline{A}BCD + AB(\overline{C} + C)$$

$$= \overline{B}CD + \overline{A}BCD + AB$$

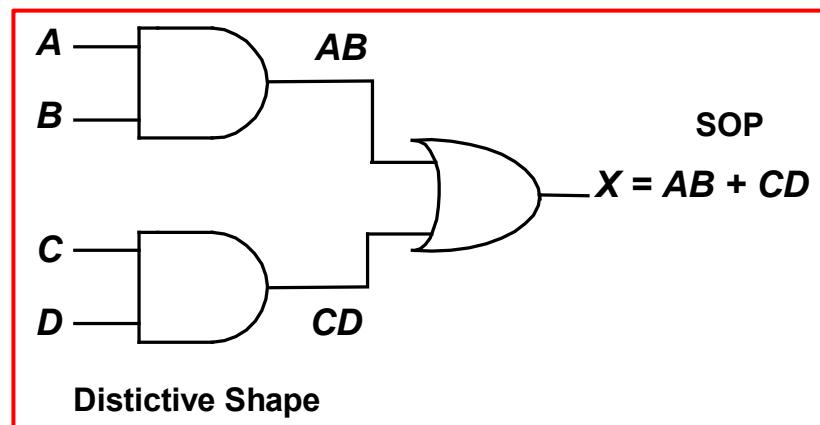
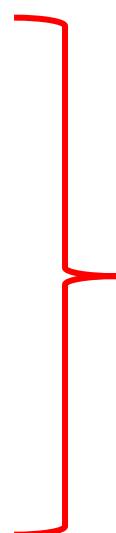
$$= \overline{B}CD + B(\overline{A}CD + A)$$

$$= \overline{B}CD + B(CD + A)$$

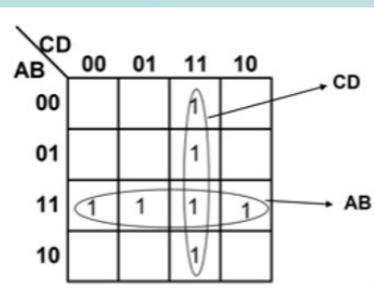
$$= \overline{B}CD + BCD + AB$$

$$= (\overline{B} + B)CD + AB$$

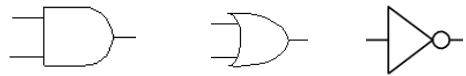
$$= AB + CD$$



**Solution #1**



**AND-OR Logic**



## AND-OR-Invert Logic

- It is the **complemented** AND-OR circuit.
- Example:

$$X = AB + CD$$

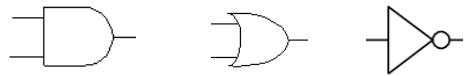
$$\bar{X} = \overline{AB + CD}$$

$$= \overline{AB} \bullet \overline{CD}$$

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

$$X = AB + CD = (\bar{\bar{A}} + \bar{\bar{B}})(\bar{\bar{C}} + \bar{\bar{D}})$$

$$\bar{X} = \overline{AB + CD} = (\overline{\overline{\bar{A}} + \overline{\bar{B}}})(\overline{\overline{\bar{C}} + \overline{\bar{D}}})$$

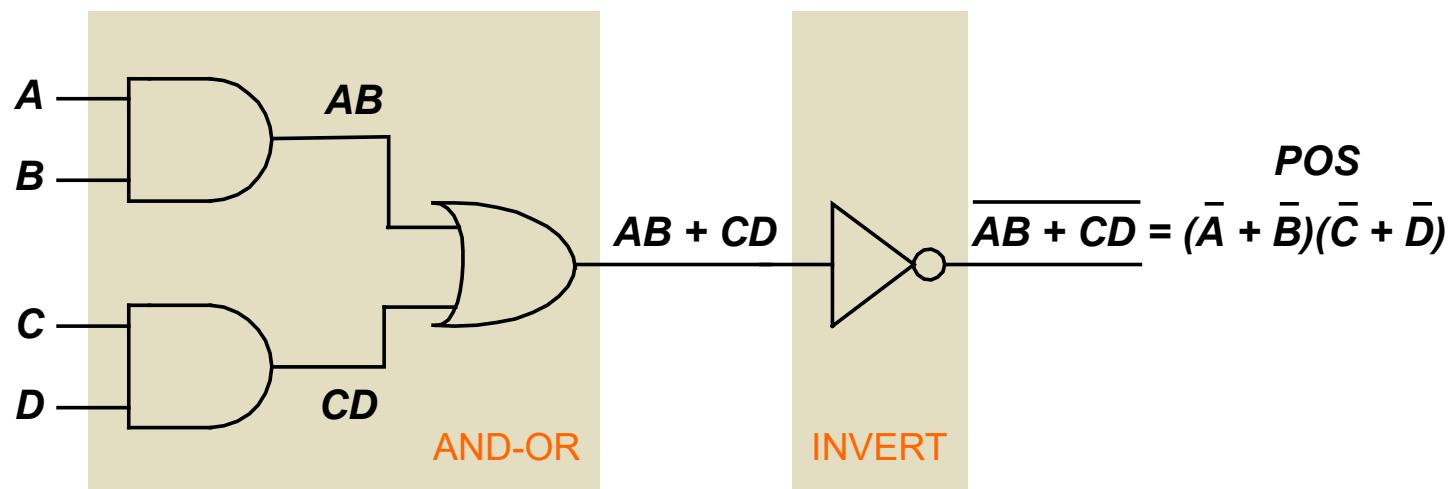


## AND-OR-Invert Logic

- POS expressions  $(\bar{A} + \bar{B})(\bar{C} + \bar{D})$

is actually

$$\overline{(\bar{A} + \bar{B})(\bar{C} + \bar{D})} = \overline{\overline{AB + CD}}$$



**Exercise 5.1:** Refer to previous example (slide 8), solve the problem for an **active low output** using AND-OR-Invert logic.

**Solution 5.1:**

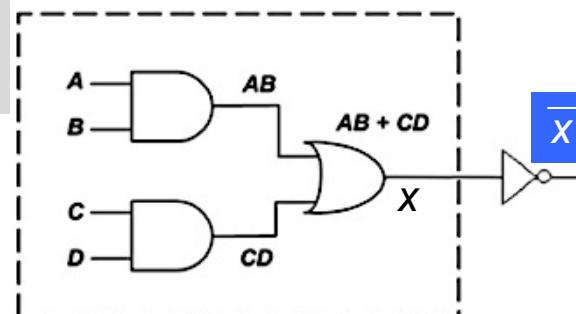
To implement an active low output, take the *complement of X* to produce a POS for  $\bar{X}$

(not POS for X)

Based on the truth table:

- derive the expression
- Simplify using K-Map
- Implement using **AND-OR-Invert** logic.

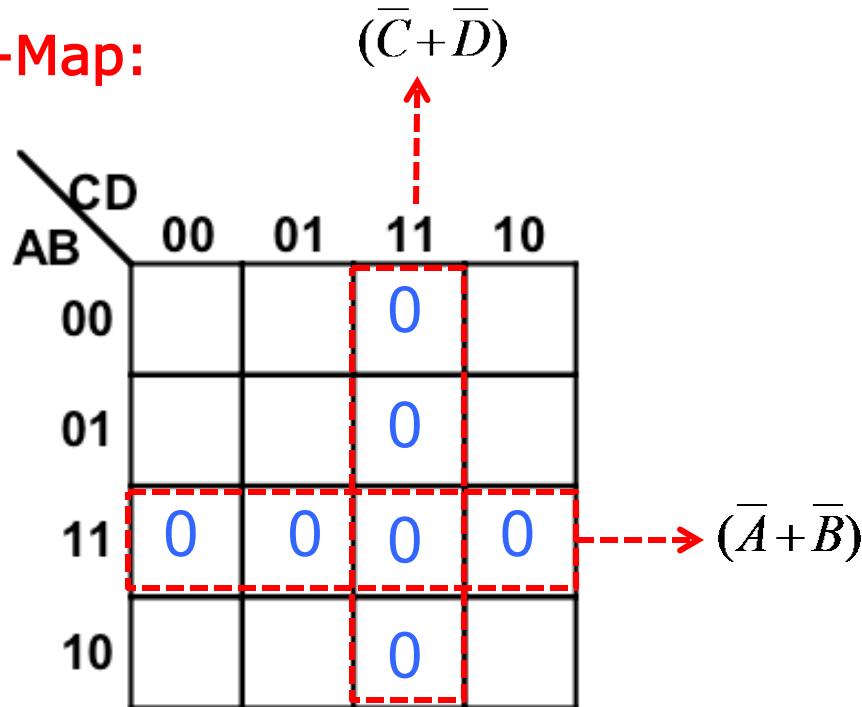
INPUTS				OUTPUT		Output $\bar{X}$
A	B	C	D	AB	CD	
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	0	0
0	1	0	1	0	1	1
0	1	1	0	0	0	0
0	1	1	1	0	1	1
1	0	0	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	0	0	0
1	0	1	1	0	1	1
1	1	0	0	0	0	0
1	1	0	1	0	1	1
1	1	1	0	0	0	0
1	1	1	1	0	1	1



continue...

## Truth table:

K-Map:



INPUTS						Output $\bar{X}$
A	B	C	D	AB	CD	
0	0	0	0	0	0	1
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	0	1
0	1	1	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	1	1	0

Expression (Truth table):

$$\begin{aligned}\bar{X} &= (A+B+\bar{C}+\bar{D})(A+\bar{B}+\bar{C}+\bar{D})(\bar{A}+B+\bar{C}+\bar{D})(\bar{A}+\bar{B}+C+D) \\ &\quad (\bar{A}+\bar{B}+C+\bar{D})(\bar{A}+\bar{B}+\bar{C}+D)(\bar{A}+\bar{B}+\bar{C}+\bar{D})\end{aligned}$$

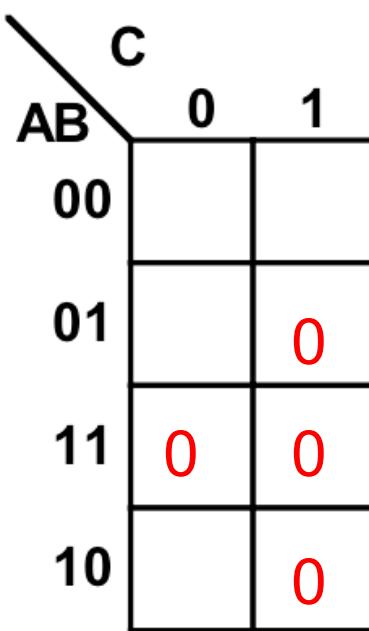
Expression (K-Map):  $\bar{X} = (\bar{A} + \bar{B})(\bar{C} + \bar{D})$

## Self-Test:

Based on the truth table below, implement the active low output by applying the AND-OR-Invert logic.

sensor inputs

A	B	C	Output	Output $\bar{X}$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0



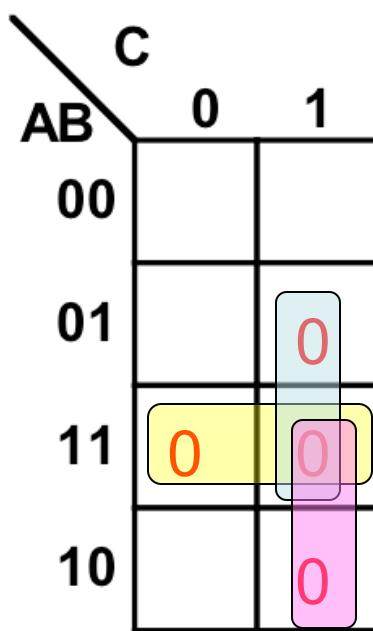
Expression (K-Map): POS

## Self-Test:

Based on the truth table below, implement the active low output by applying the AND-OR-Invert logic.

sensor inputs

A	B	C	Output	Output $\bar{X}$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0



Expression (K-Map): POS

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



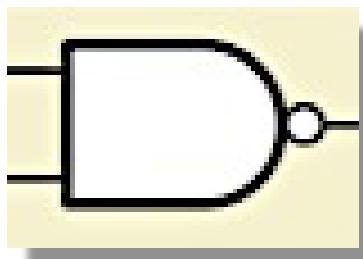
---

# The Universal Property of NAND and NOR Gates

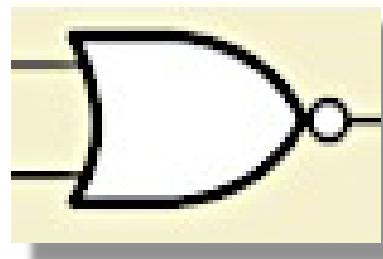
---

## The Universal Property of NAND and NOR Gates

- If a gate can be converted and perform the function of a basic gates, then it can be used to implement any logic circuits.
- Therefore, they are called universal gate.
- There are 2 common universal gates:

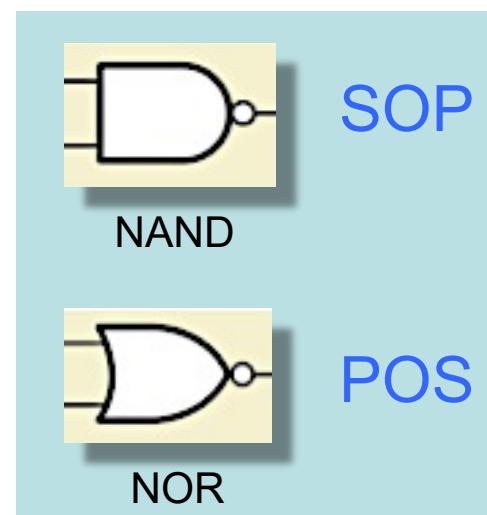


NAND

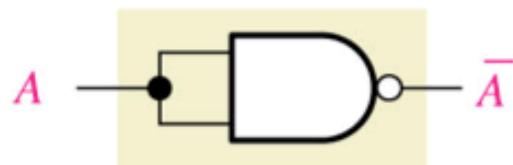


NOR

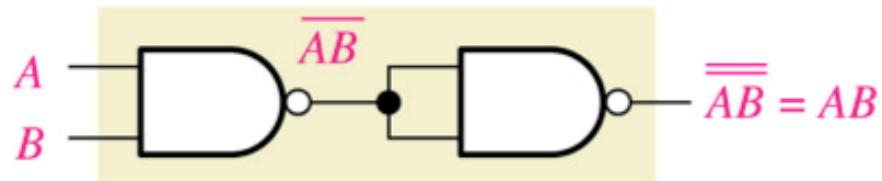
- It's a good thing to implement the final circuit using only a single type of gate. The advantages are
  - Have to stock only one type of gates
  - Buy only one type of gates in a volume so that, it can reduce cost
  - Can reuse the extra of unused gates to implement other gates
- This can be done by using a universal gate -- NAND or NOR
  - Usually SOP is implemented using NAND only
  - and POS implemented as NOR only



# The NAND Gate as Universal Logic Element



(a) A NAND gate used as an inverter

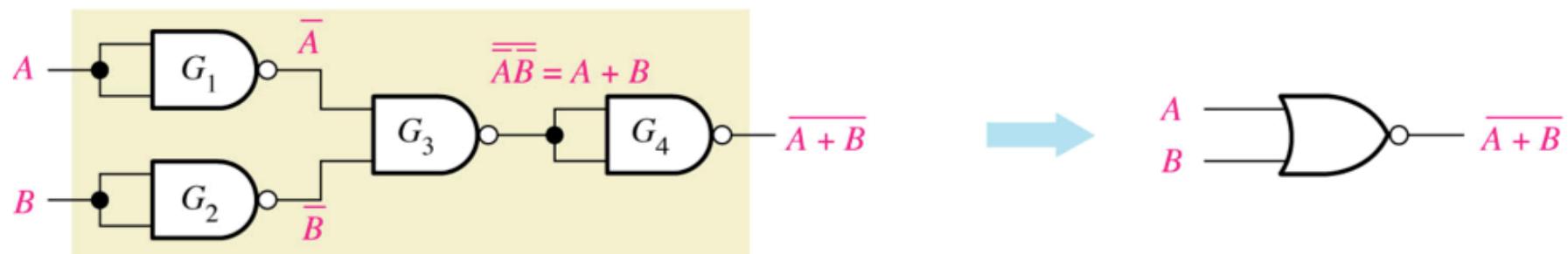


(b) Two NAND gates used as an AND gate

continue...



(c) Three NAND gates used as an OR gate



(d) Four NAND gates used as a NOR gate

## Exercise 5.2:

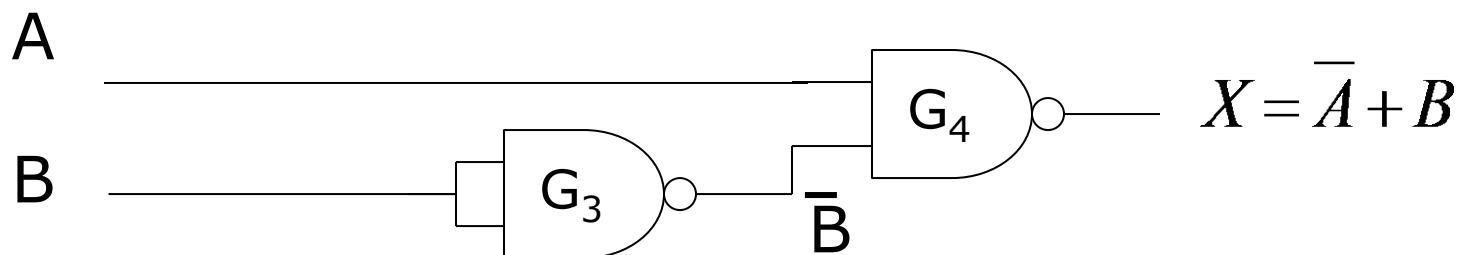
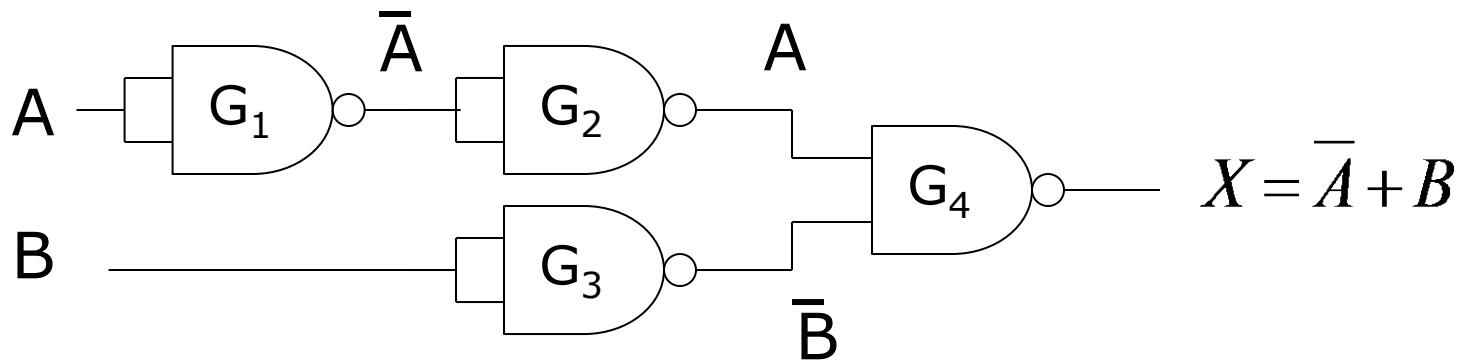
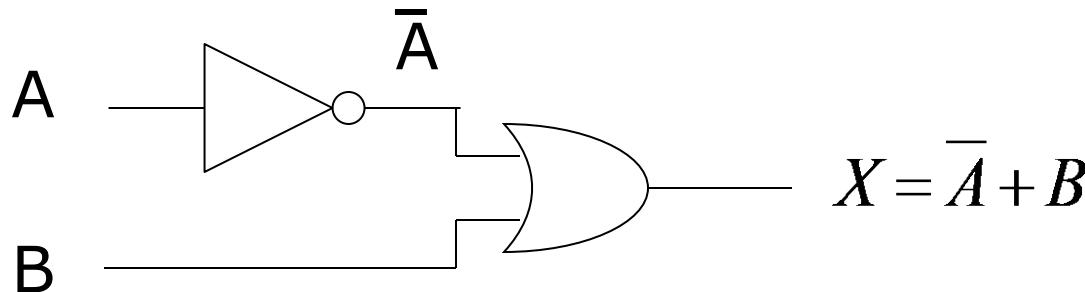
Use NAND gates to implement each expression.

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

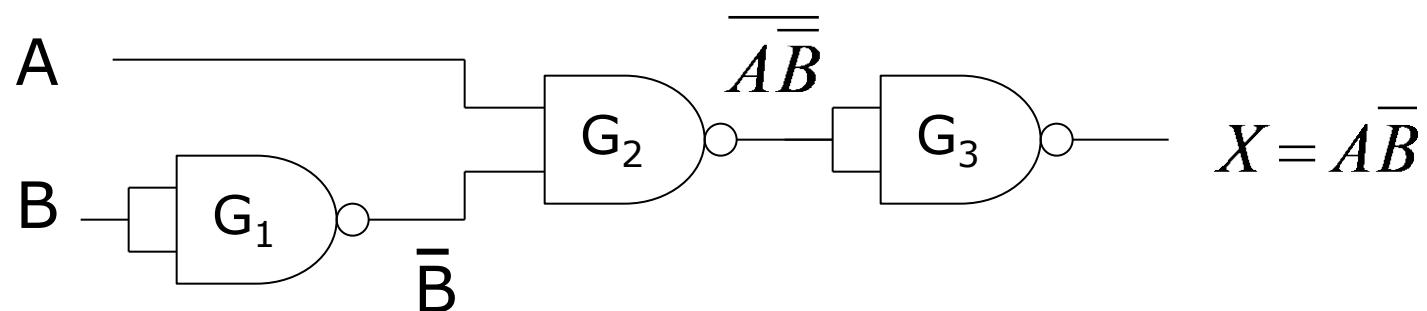
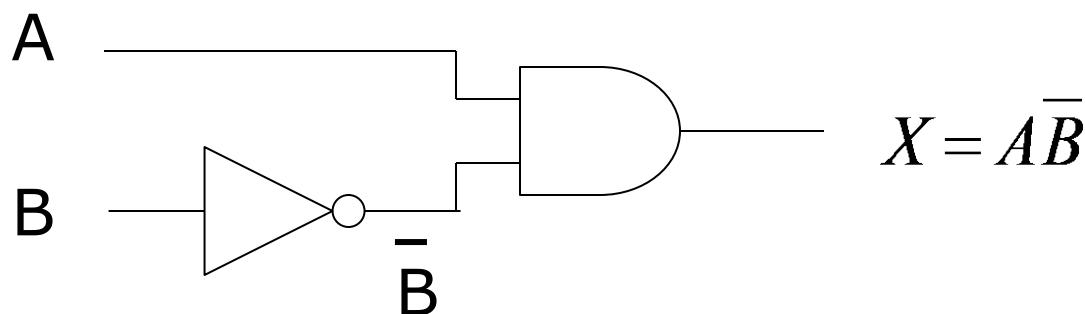
b.  $X = A\overline{B}$

**Solution:**

**4(a)**



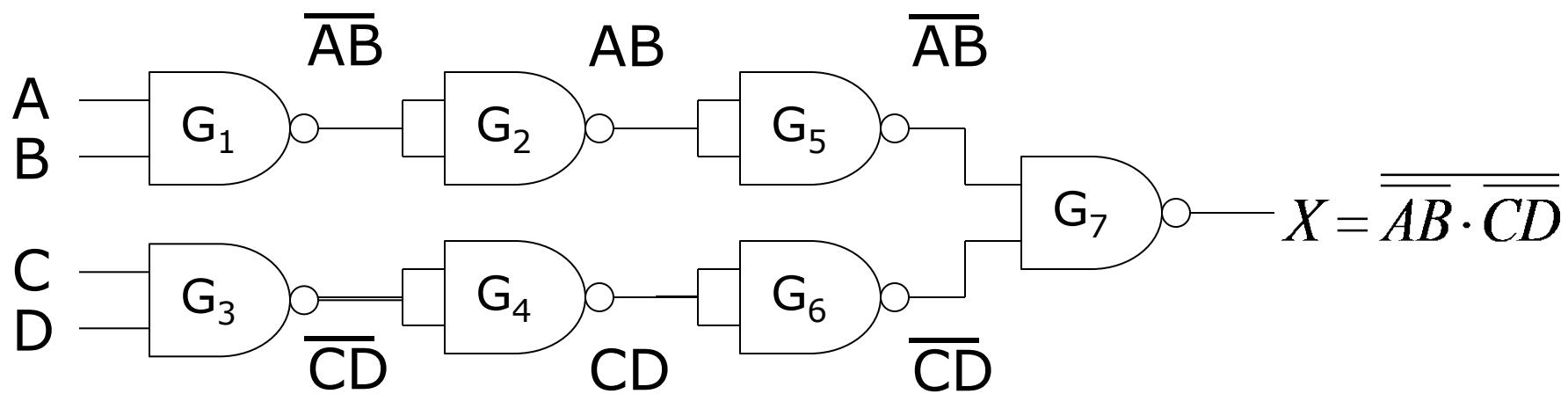
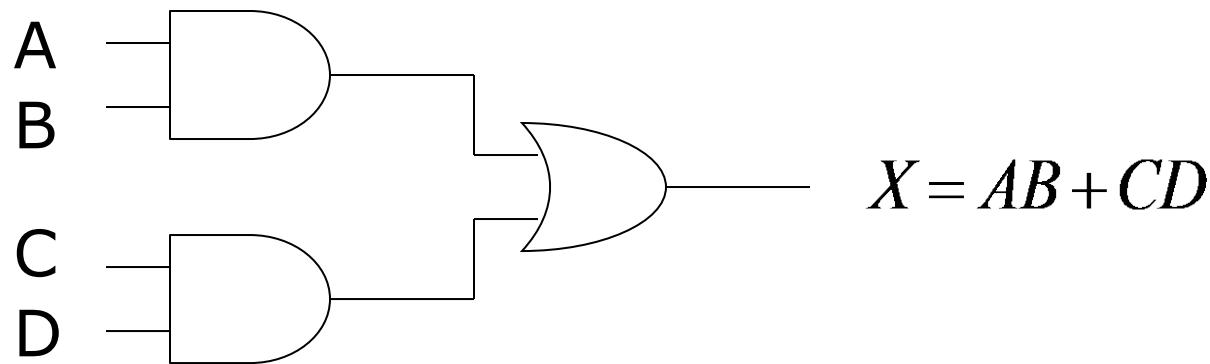
4(b)



**Exercise 5.3:** Use NAND gates to implement this expression.

$$X = AB + CD$$

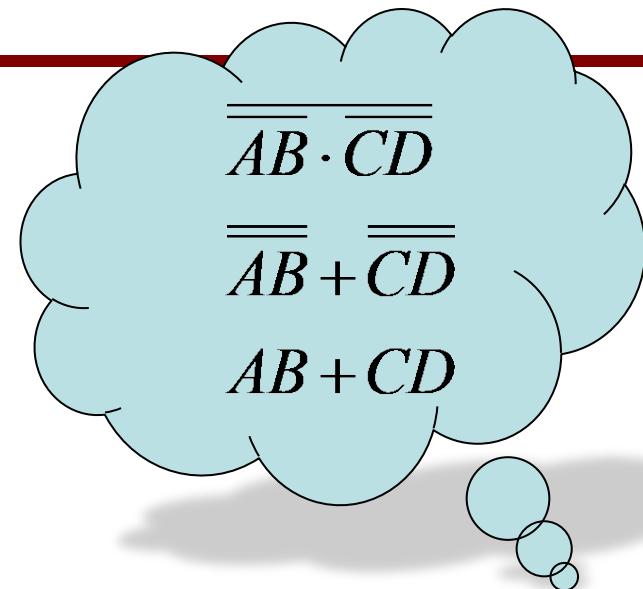
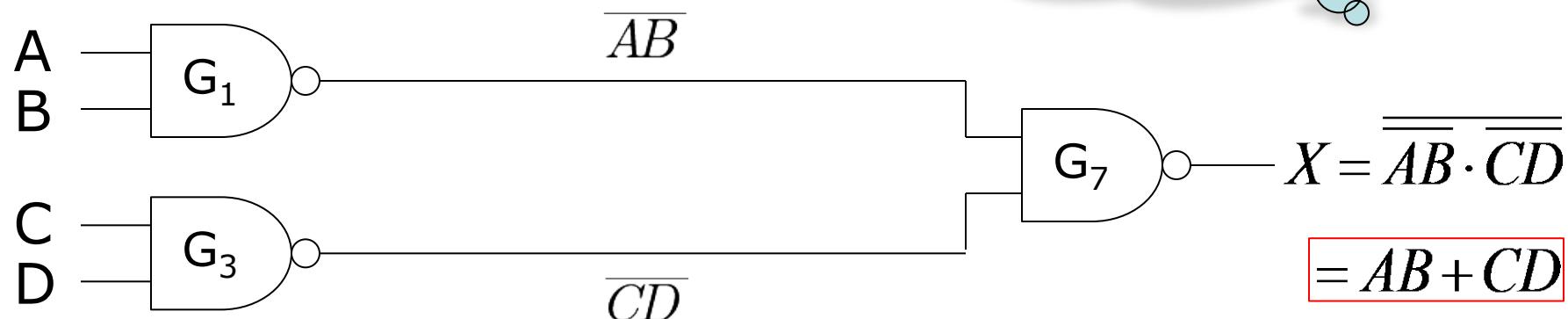
**Solution:**



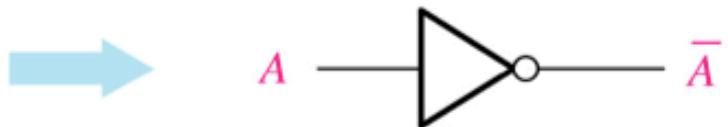
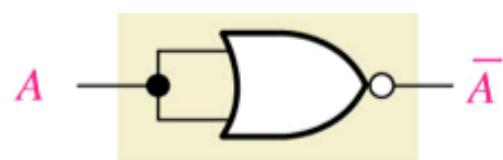
**Exercise 5.3:** Use NAND gates to implement this expression.

$$X = AB + CD$$

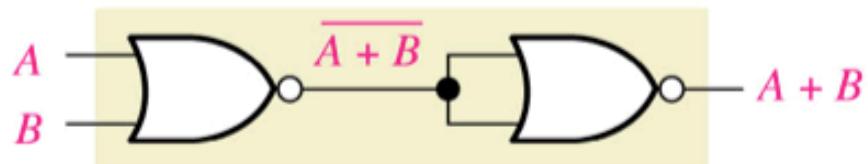
Can discard  $G_2$ ,  $G_4$ ,  $G_5$  and  $G_6$



# The NOR Gate as Universal Logic Element

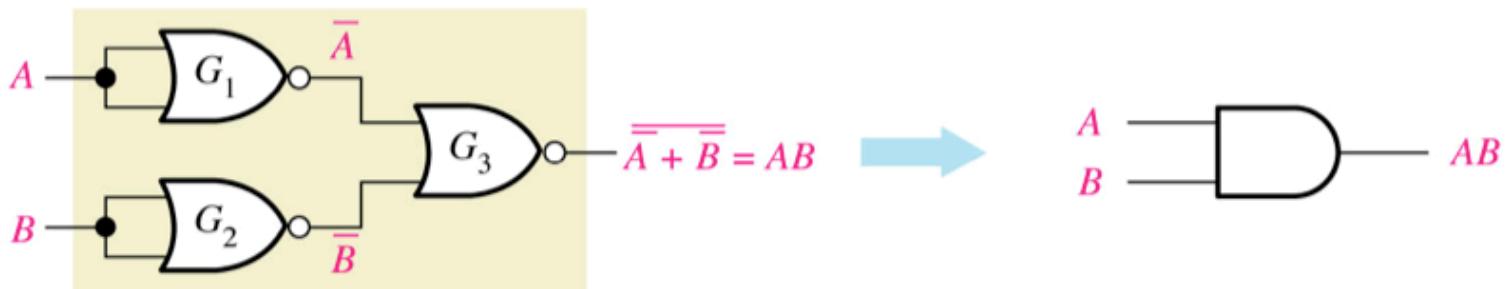


(a) A NOR gate used as an inverter

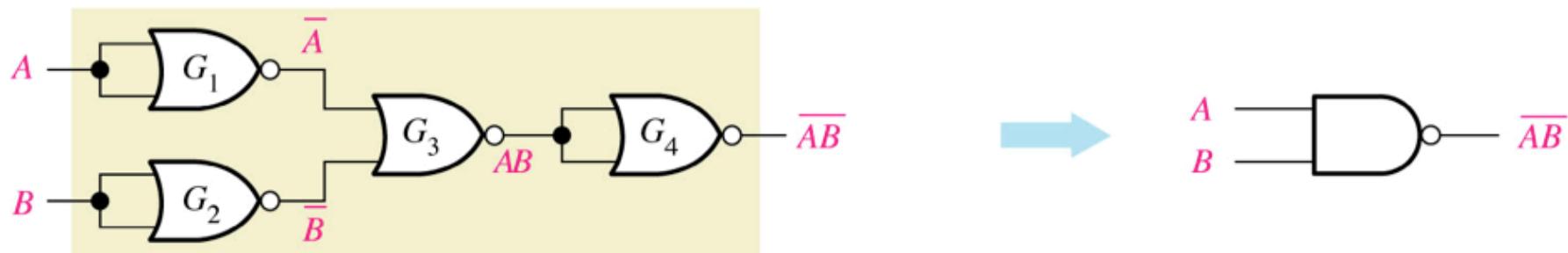


(b) Two NOR gates used as an OR gate

continue...



(c) Three NOR gates used as an AND gate



(d) Four NOR gates used as a NAND gate

## Exercise 5.4:

Use NOR gates to implement each expression.

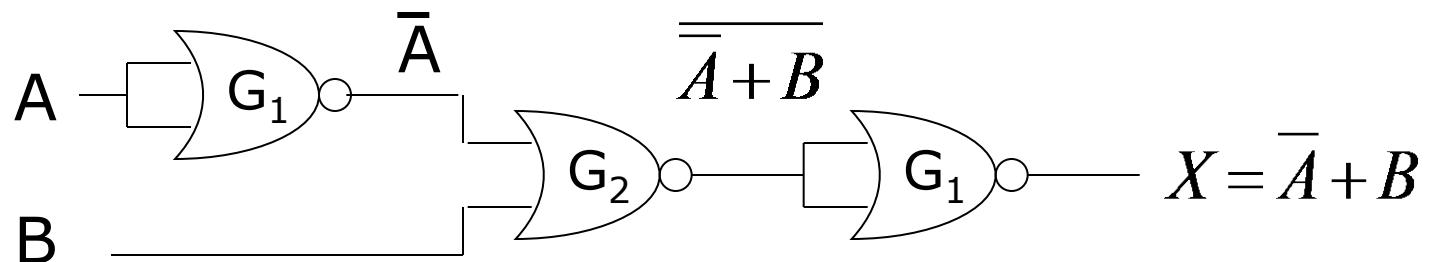
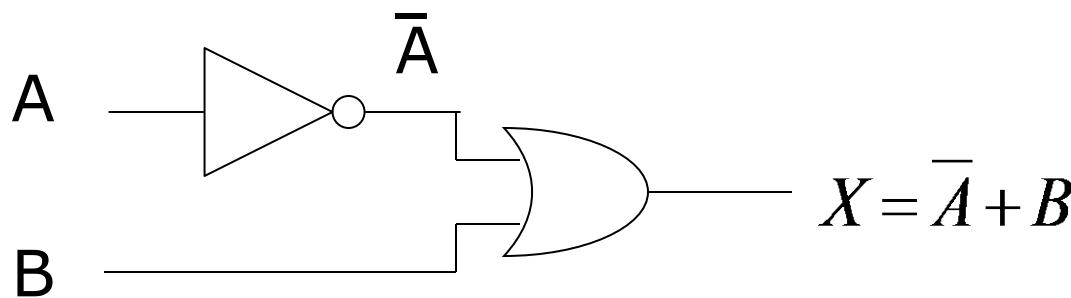
a)  $X = \overline{A} + B$

b)  $Y = A\overline{B}$

- a)  $X = \overline{A} + B$   
 b)  $Y = A\overline{B}$

**Solution:**

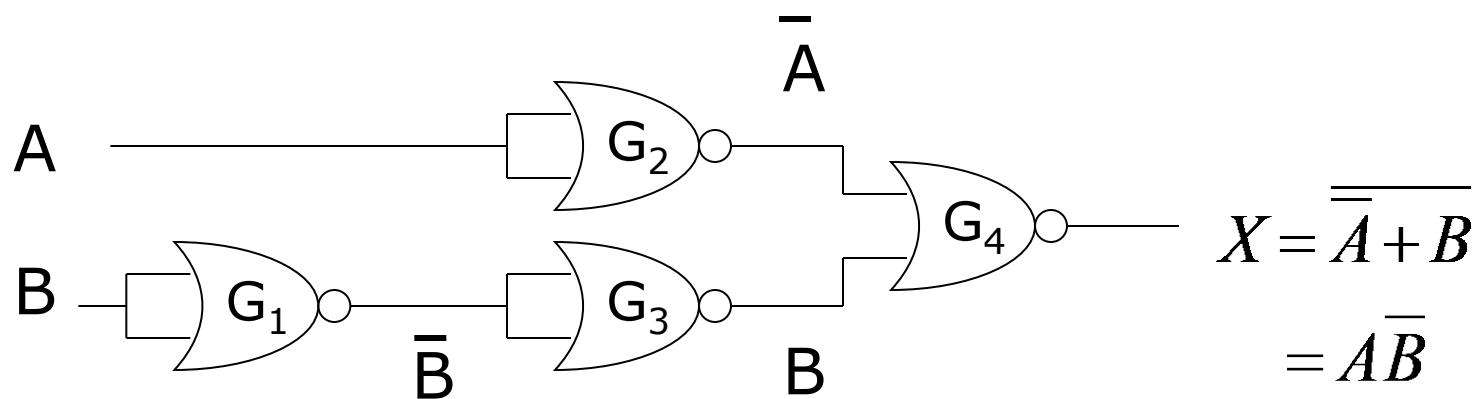
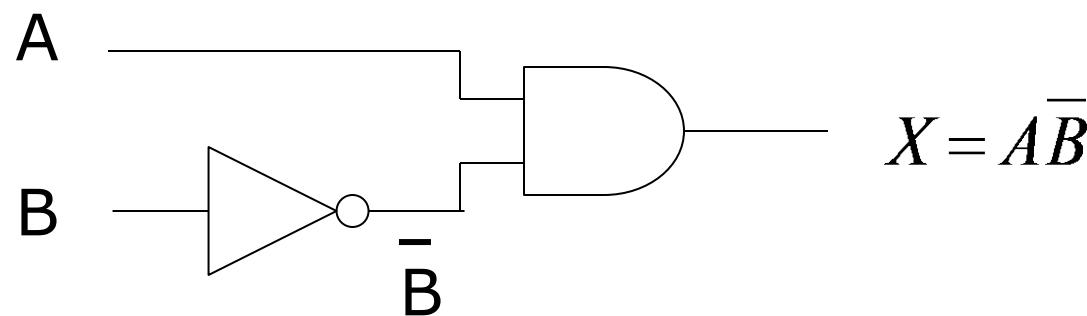
**6(a)**



**continue...**

- a)  $X = \overline{A} + B$   
 b)  $Y = A\overline{B}$

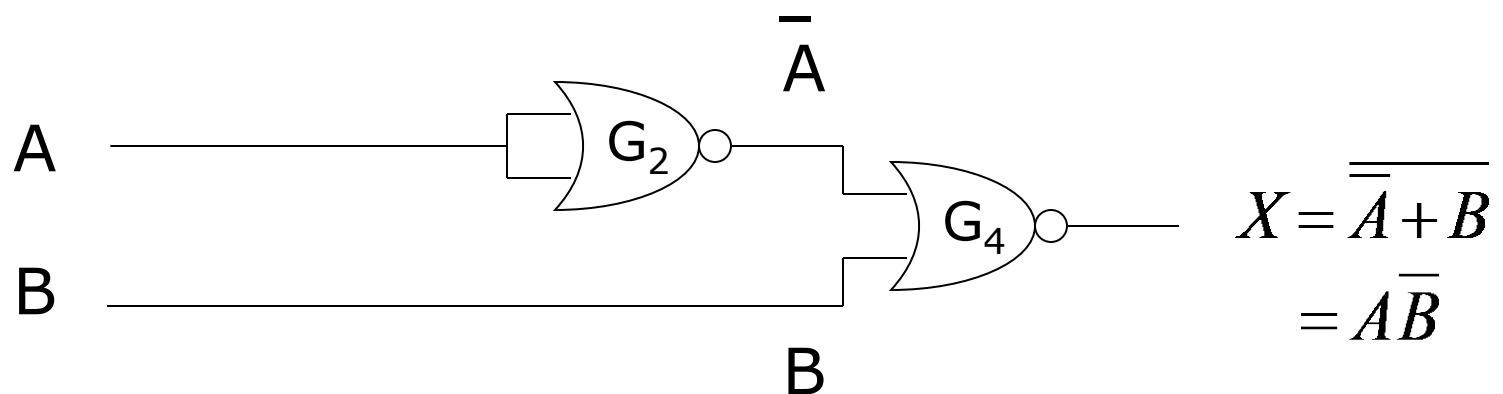
6(b)



continue...

- a)  $X = \overline{A} + B$   
b)  $Y = A\overline{B}$

Can discard  $G_1$ , and  $G_3$





Tips

**Table:** Number of universal logic element for NAND and NOR gate

	NAND	NOR
NOT	1	1
AND	2	3
OR	3	2
NAND	-	4
NOR	4	-



## Dual Symbol

# Dual Symbol

NAND



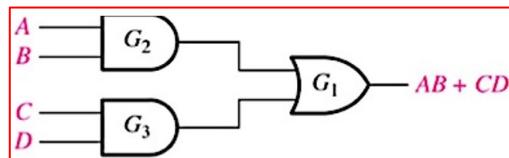
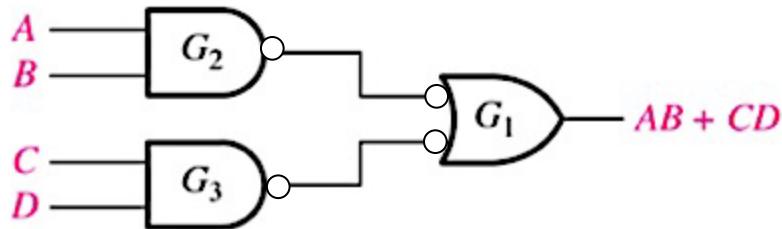
Alternate symbol for NAND – Negative OR

NOR



Alternate symbol for NOR – Negative AND

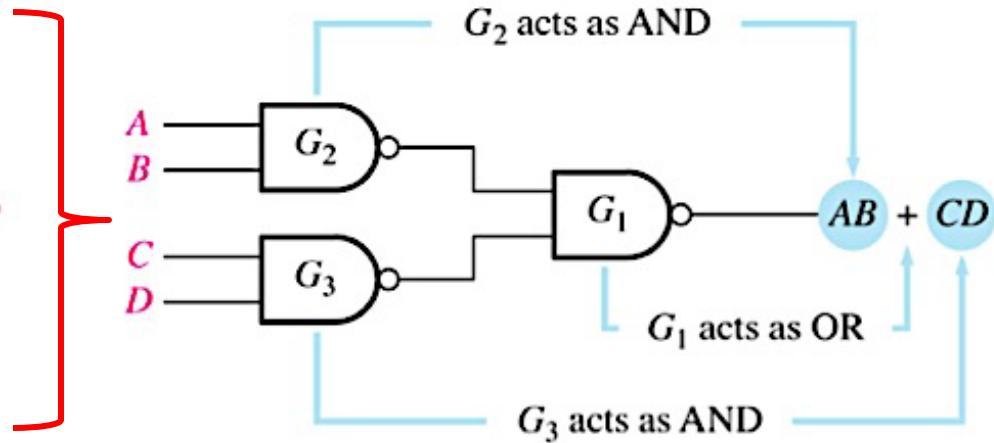
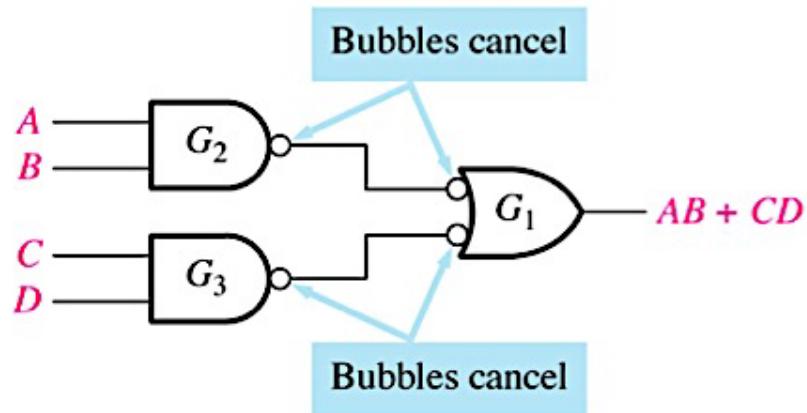
## Converting SOP Circuit to NAND only



**Step 1:** Put the bubbles at the input of right most OR gate.

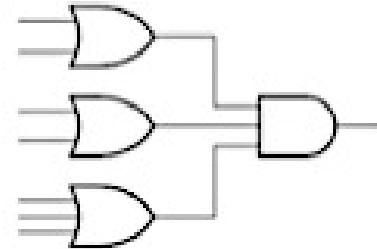
**Step 2:** To offset the effect of adding those bubbles, place additional bubbles at the output of the AND gate

**Step 3:** Replace the **negative-OR** symbol with **NAND** symbol.

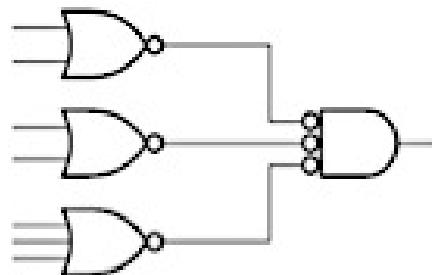


## Converting POS Circuit to NOR only

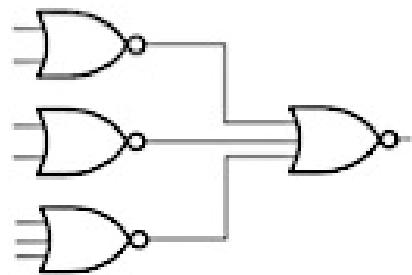
Original POS circuit



Step1: Put the bubble at the input of the right most AND



Step2: to compensate the effect of putting that bubble, add another bubble at the output of the OR gate



Step3: Replace the negative-AND symbol with NOR symbol

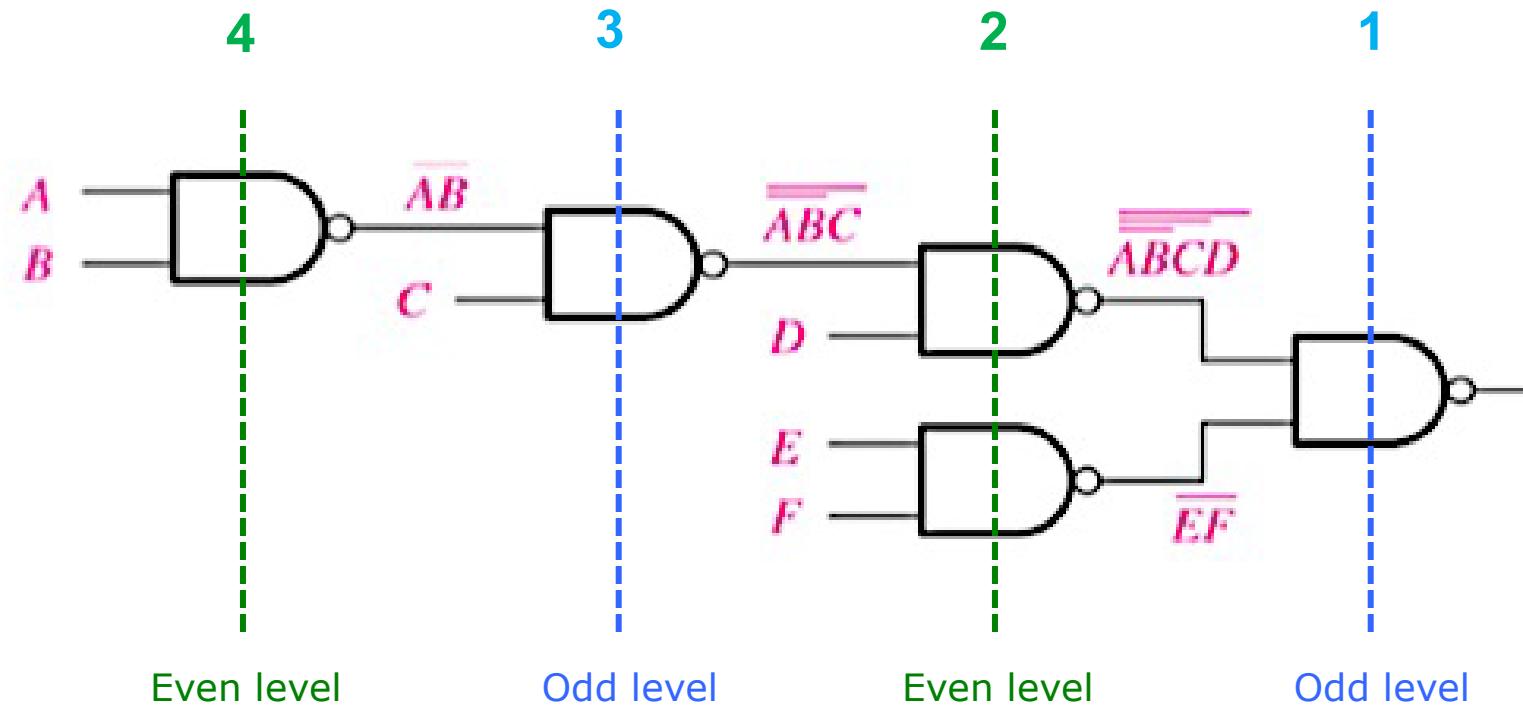
**Exercise 5.5:** Convert the logic circuit of the following expression to NOR or NAND gates only.

a)  $X = AB + CD + EF$

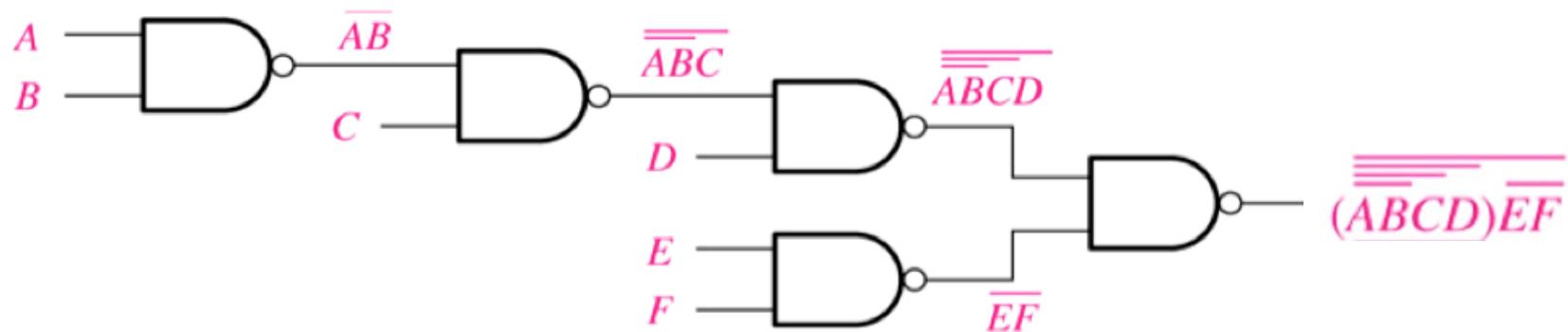
b)  $Y = (A + B)(C + D)(E + F)$



- Dual symbol is needed to simplify the “reading” of the schematic
  - can determine how the circuits works – very fast
- Replace the NAND gate with its dual symbol which is Negative-OR
- Choose which level that you want to replace the alternate symbol - odd or even
  - If you choose odd – only replace the odd level with an alternate symbol



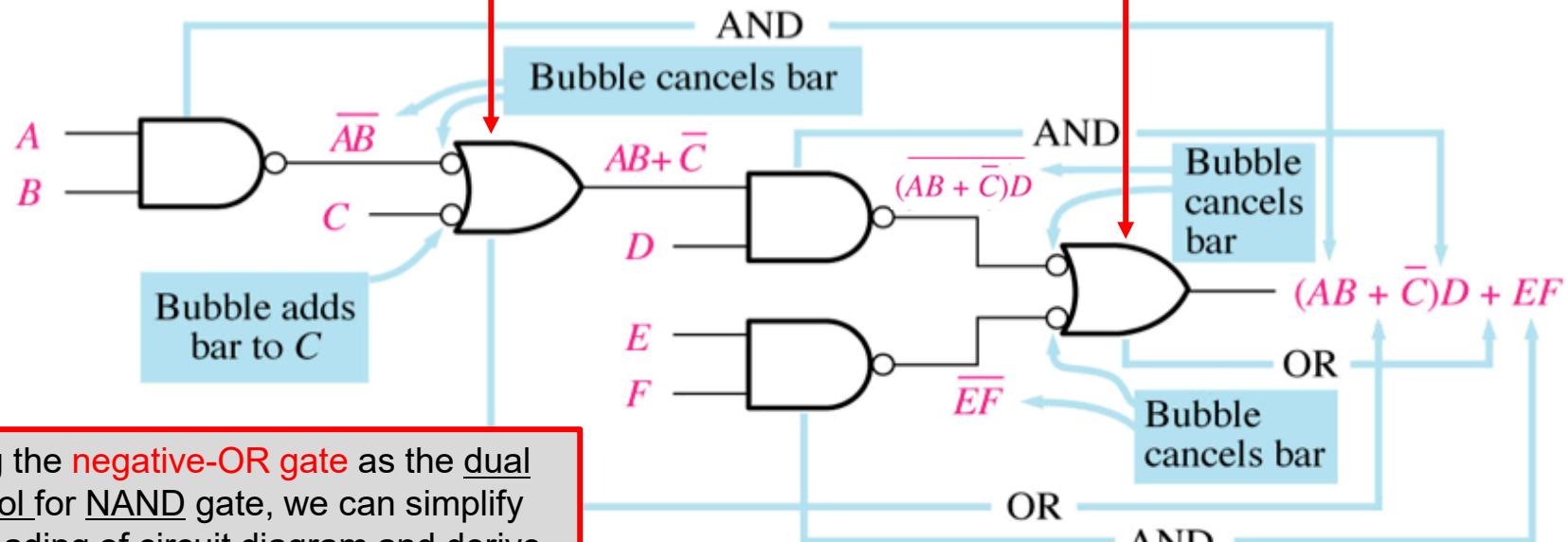
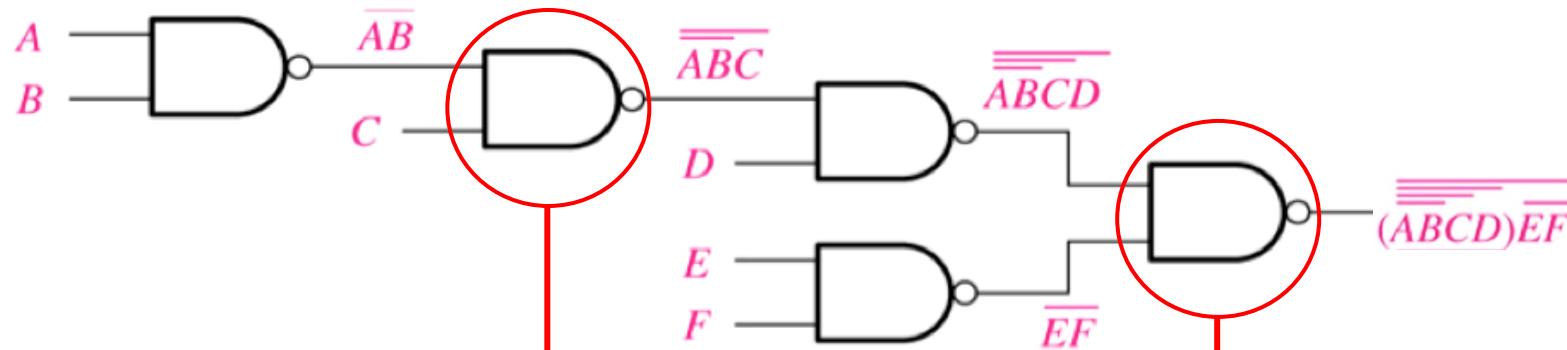
## Dual Symbols: NAND Only



(a) Several Boolean steps are required to arrive at final output expression.

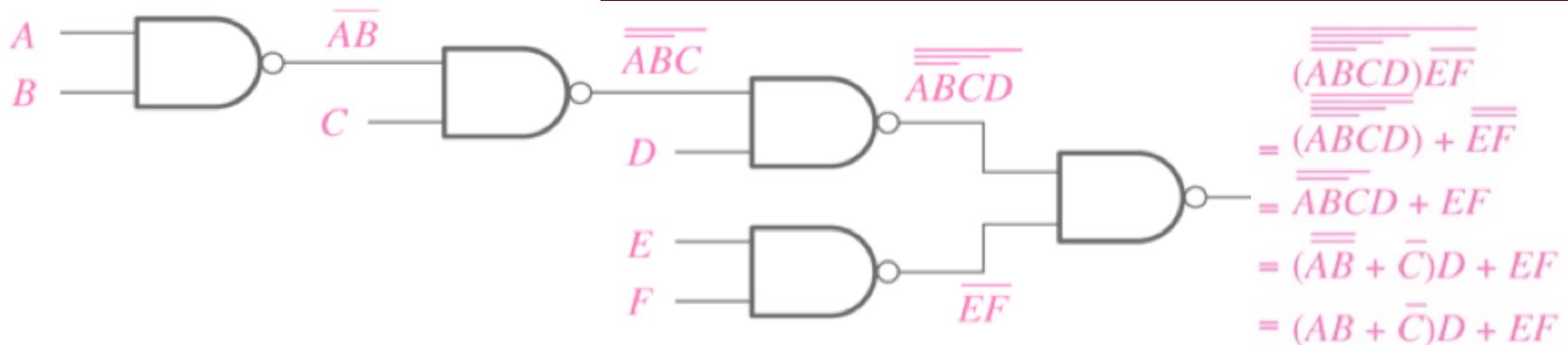
$$\begin{aligned} \overline{\overline{\overline{ABCD})\overline{EF}}} &= \overline{\overline{\overline{ABCD}}} + \overline{\overline{EF}} \\ &= \overline{\overline{ABCD}} + EF \\ &= (\overline{\overline{AB}} + \overline{\overline{C}})D + EF \\ &= (AB + \overline{C})D + EF \end{aligned}$$

(b) Output expression can be obtained directly from the function of each gate symbols in the diagram with dual symbol gate inserted



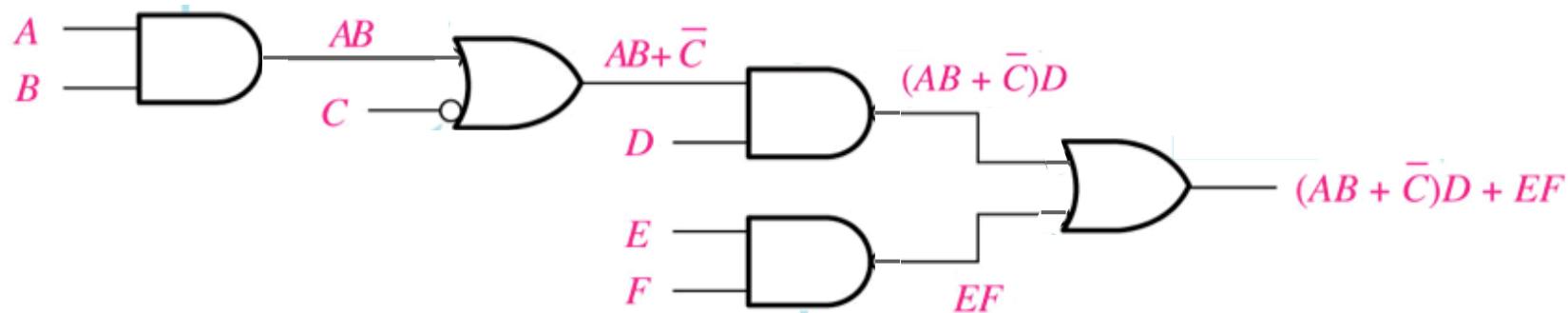
Using the **negative-OR gate** as the dual symbol for NAND gate, we can simplify the reading of circuit diagram and derive the Boolean expression faster  
(assume using dual symbol at odd level)

# Dual Symbols: NAND Only



(a) Several Boolean steps are required to arrive at final output expression.

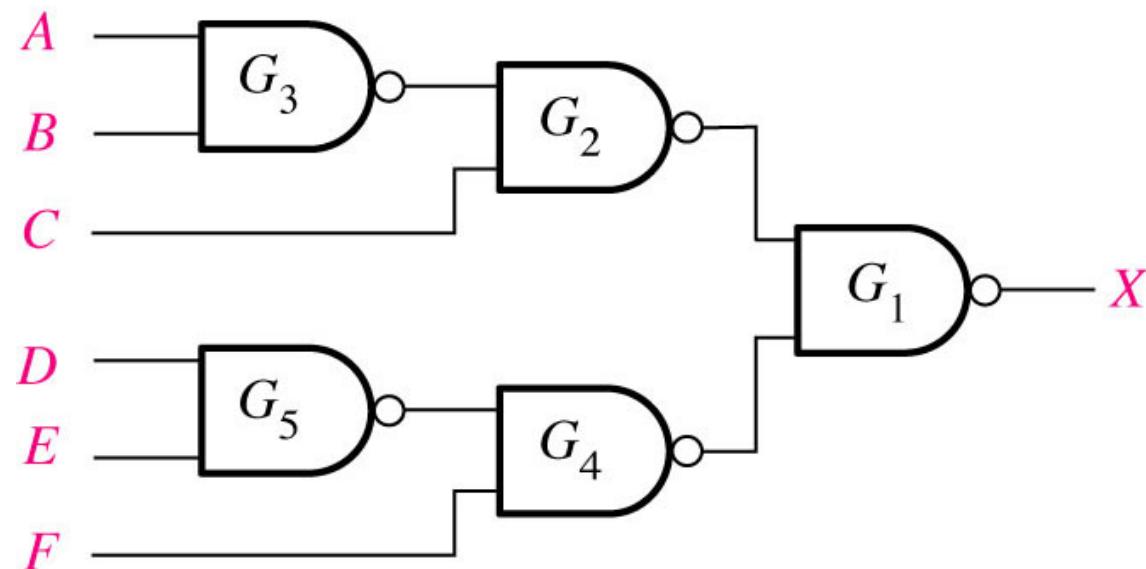
(b) Output expression can be obtained directly from the function of each gate symbols in the diagram with dual gate symbols inserted



Dual symbol can simplify the reading of the circuit

**Example:** a) Write the expression of X.

b) Redraw the logic diagram and develop the output expression for the circuit using the appropriate dual symbols (**odd level**).

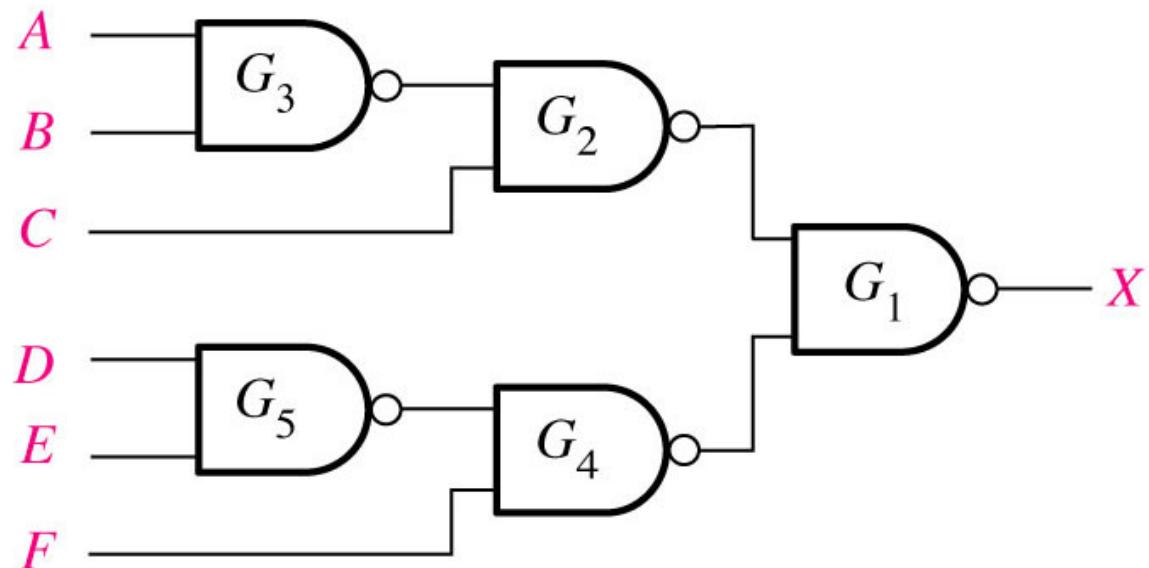


continue...

**Solution:**

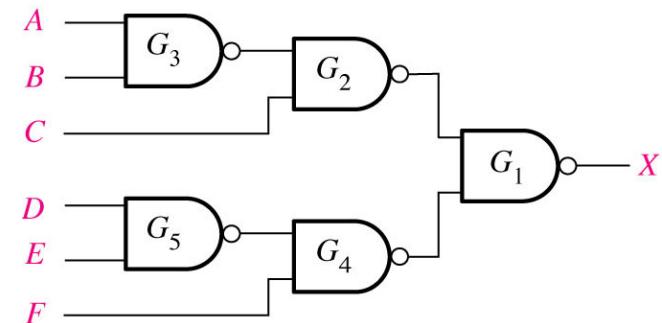
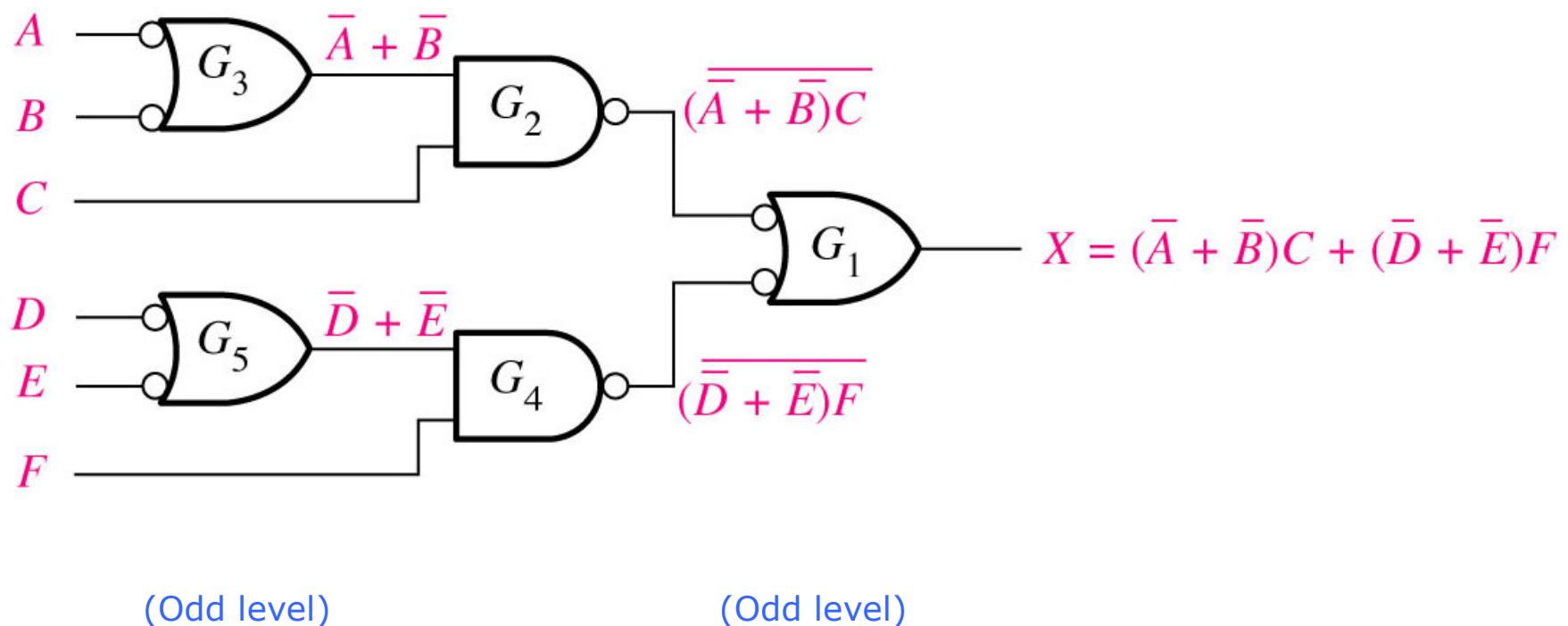
a) Expression of X

$$\begin{aligned}X &= (\overline{\overline{AB}} \cdot C) \cdot (\overline{\overline{DE}} \cdot F) \\&= (\overline{\overline{AB}} \cdot C) + (\overline{\overline{DE}} \cdot F) \\&= (\overline{AB} \cdot C) + (\overline{DE} \cdot F) \\&= (\overline{A} + \overline{B})C + (\overline{D} + \overline{E})F\end{aligned}$$



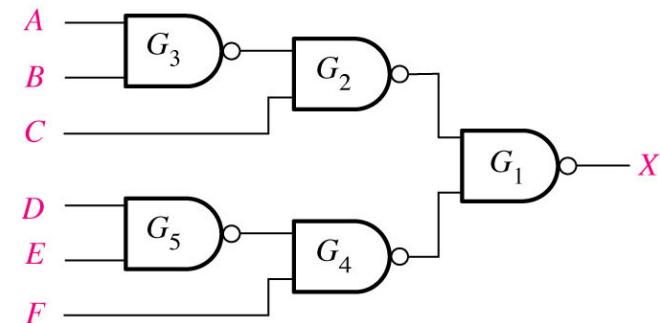
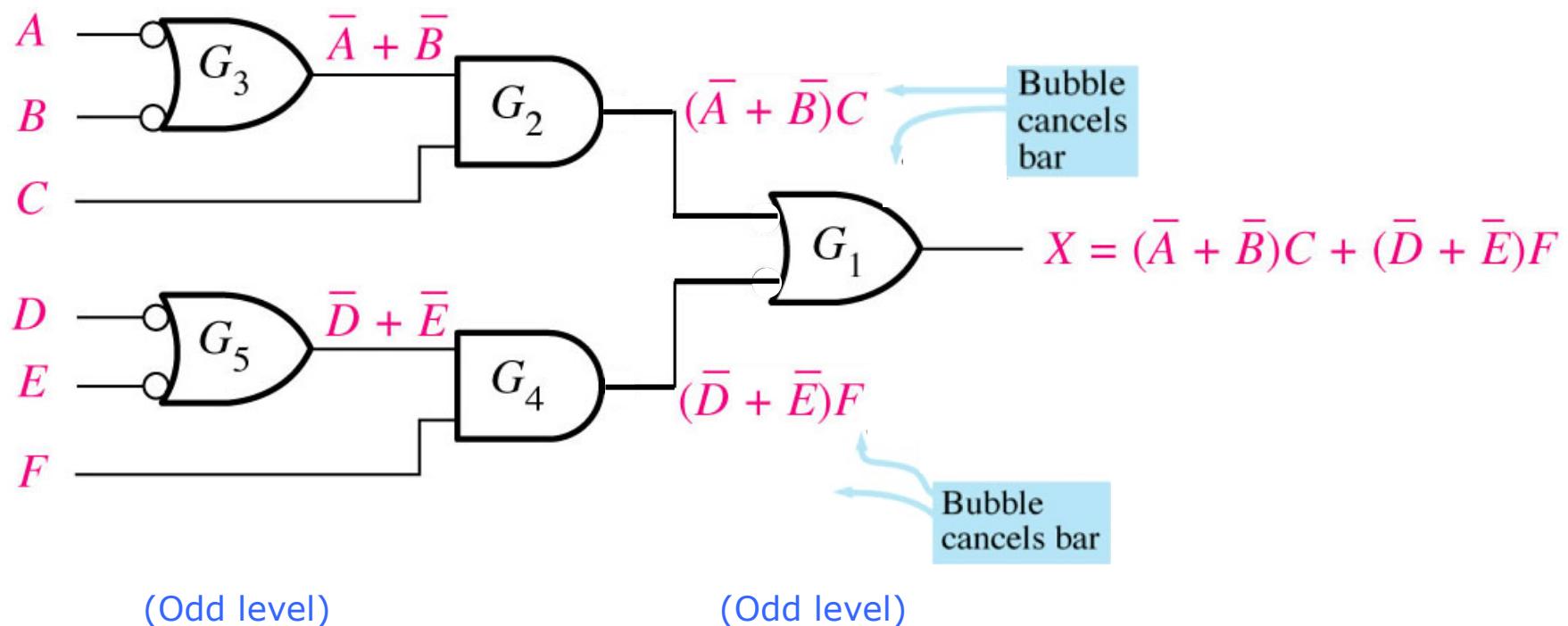
**Solution:**

b) (odd level)



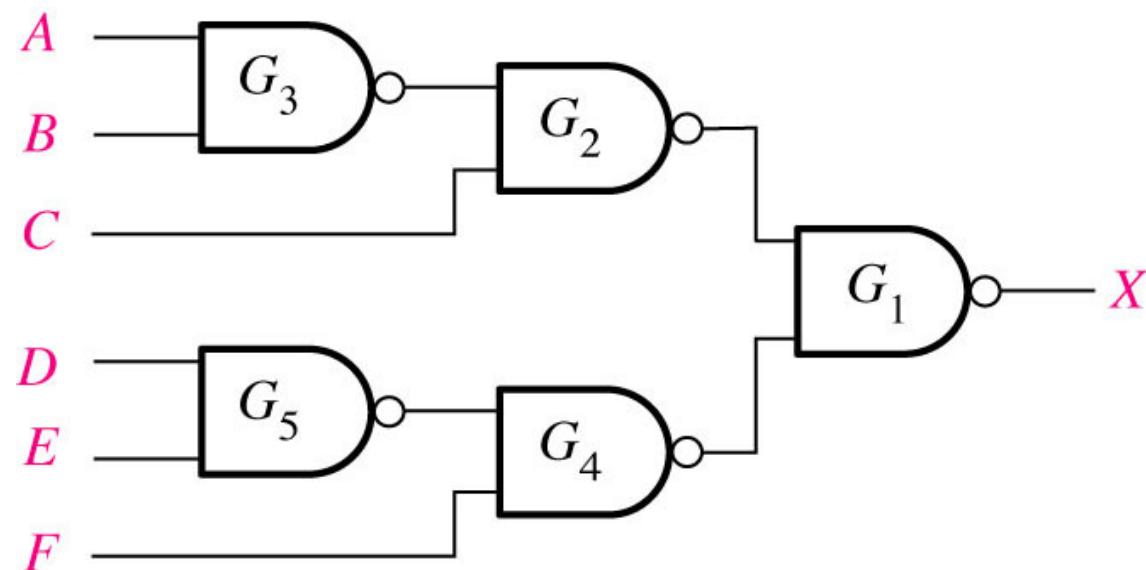
**Solution:**

b) (odd level)



Homework

**Exercise 5.6:** Redraw the logic diagram and develop the output expression for the circuit using the appropriate dual symbols (even level).



**Example:**

Implement each expression with NAND logic using appropriate dual symbol.

(a)  $ABC + DE$

(b)  $ABC + \overline{D} + \overline{E}$

Rule #9:  $\bar{\bar{A}} = A$

Example:

Implement each expression with NAND logic using appropriate dual symbol.

(a)  $ABC + DE$



$$= \overline{\overline{ABC} + \overline{DE}}$$

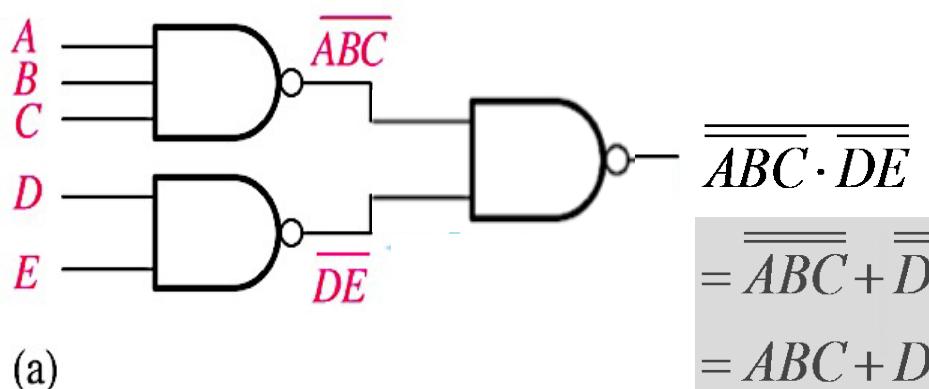


(b)  $ABC + \overline{D} + \overline{E}$

$$= \overline{\overline{(ABC)}} \cdot \overline{\overline{(DE)}}$$

Solution: (Before using dual symbol)

→ next, draw the circuit with three NAND gates (... count no of 'bars')

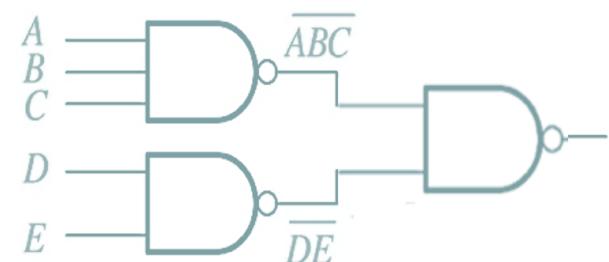


**Example:**

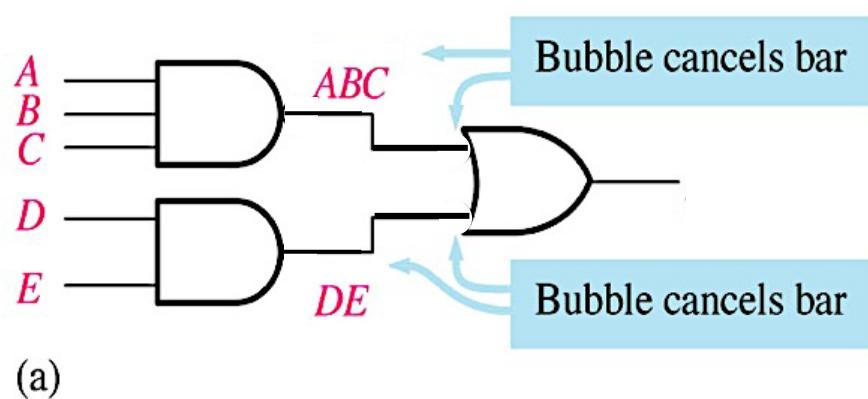
Implement each expression with NAND logic using appropriate dual symbol.

(a)  $ABC + DE$

(b)  $ABC + \bar{D} + \bar{E}$



**Solution:** (After using dual symbol at odd level)



Rule #9:  $\bar{\bar{A}} = A$

Example:

Implement each expression with NAND logic using appropriate dual symbol.

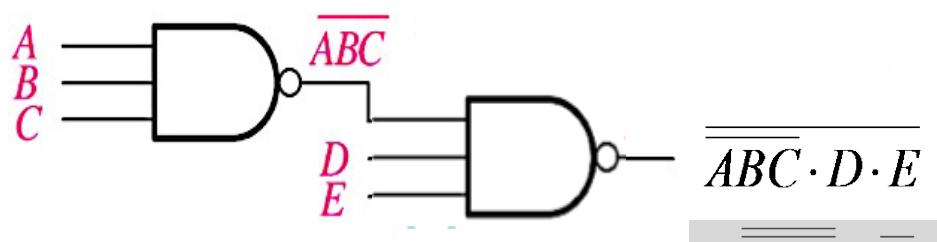
(a)  $ABC + DE$

(b)  $ABC + \bar{D} + \bar{E}$  

$$\begin{aligned}&= \overline{\overline{ABC} + \bar{D} + \bar{E}} \\&= \overline{\overline{(ABC)}} \cdot \overline{\bar{D}} \cdot \overline{\bar{E}} \\&= \overline{\overline{(ABC)}} \cdot D \cdot E\end{aligned}$$

→ next, draw the circuit with two NAND gates (... count no of 'bars')

Solution: (Before using dual symbol)



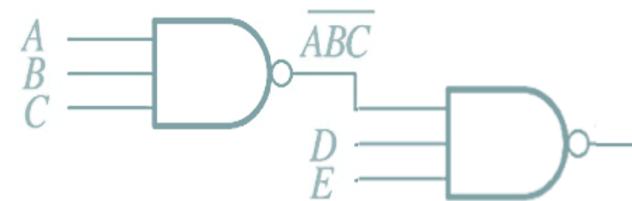
$$\begin{aligned}&= \overline{\overline{ABC} + \bar{D} + \bar{E}} \\&= ABC + \bar{D} + \bar{E}\end{aligned}$$

## Example:

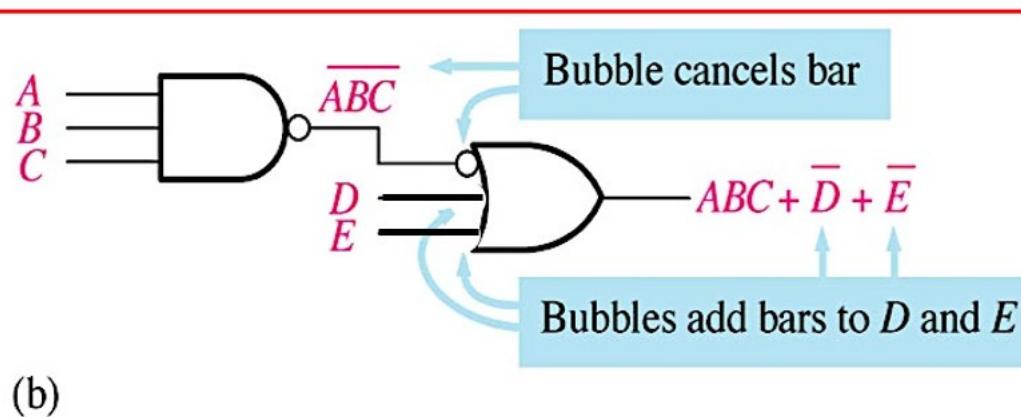
Implement each expression with NAND logic using appropriate dual symbol.

(a)  $ABC + DE$

(b)  $ABC + \bar{D} + \bar{E}$



Solution: (After using dual symbol at odd level)

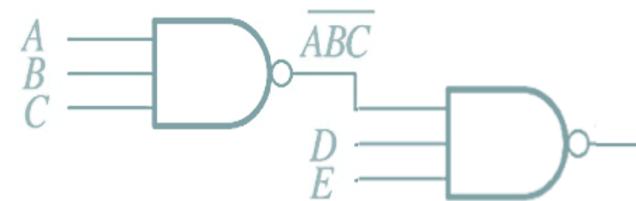


**Example:**

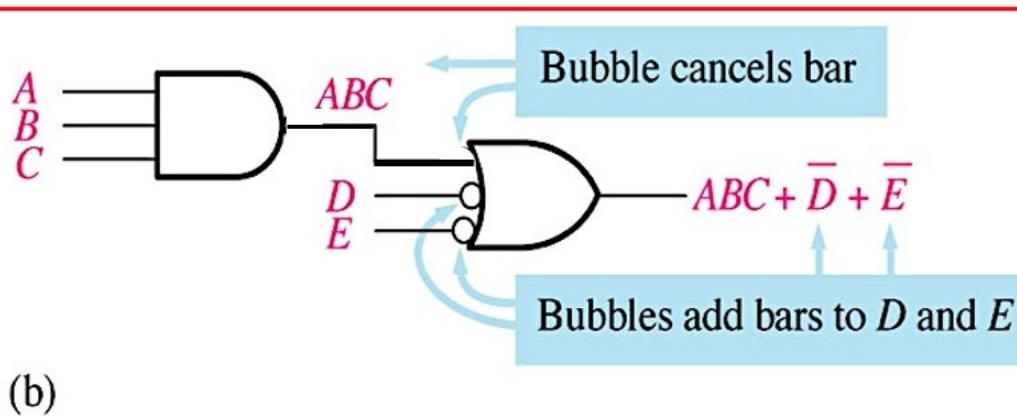
Implement each expression with NAND logic using appropriate dual symbol.

(a)  $ABC + DE$

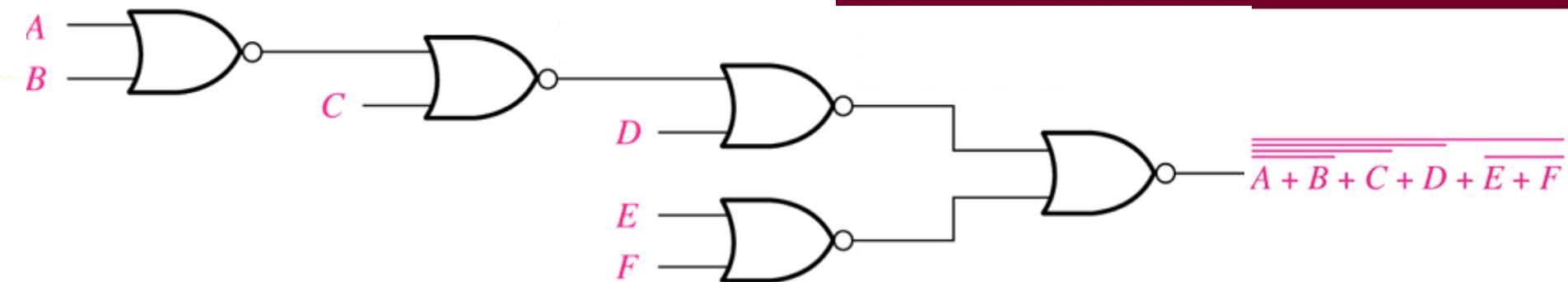
(b)  $ABC + \bar{D} + \bar{E}$



**Solution: (After using dual symbol at odd level)**



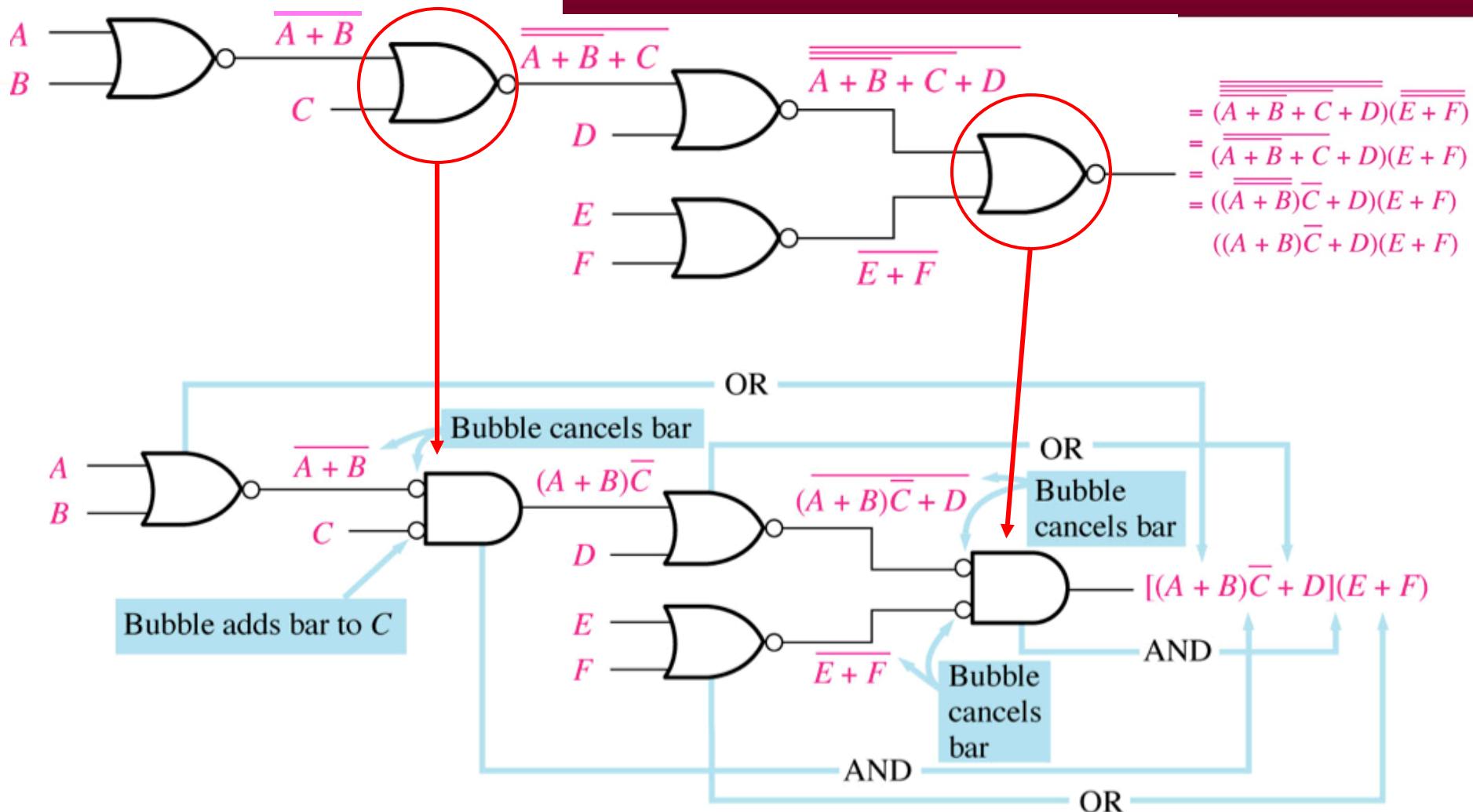
## Dual Symbols: NOR Only



(a) Final output expression is obtained after several Boolean steps.

$$\begin{aligned}\overline{\overline{A} + \overline{B} + \overline{C} + \overline{D} + \overline{E} + \overline{F}} &= (\overline{A} + \overline{B} + \overline{C} + \overline{D})(\overline{E} + \overline{F}) \\ &= (\overline{\overline{A} + \overline{B} + \overline{C} + \overline{D}})(E + F) \\ &= ((\overline{A} + \overline{B})\overline{C} + D)(E + F) \\ &\underline{((A + B)\overline{C} + D)(E + F)}\end{aligned}$$

## Dual Symbols: NOR Only



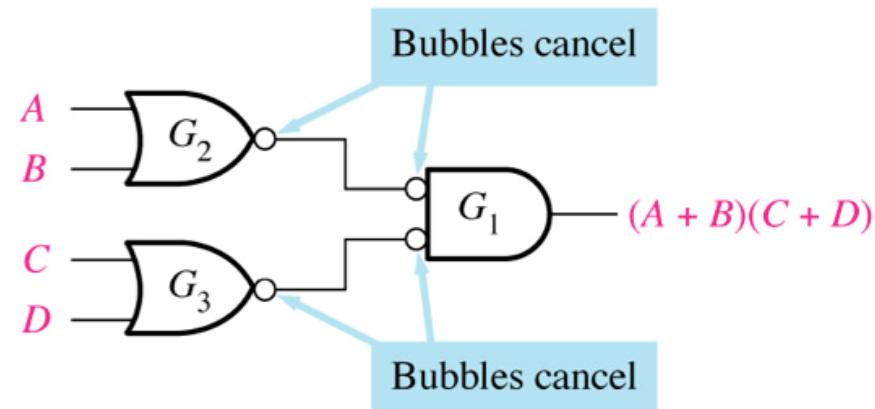
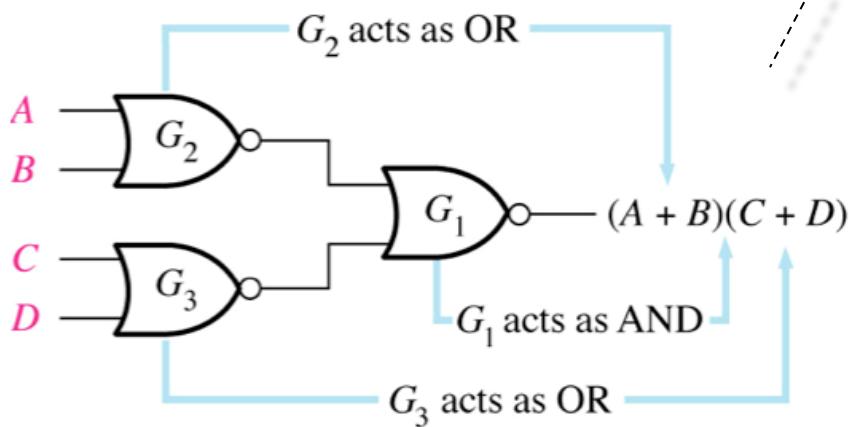
(b) Output expression can be obtained directly from the function of each gate symbol in the diagram.

## Example:

Implement the expression with NOR logic using appropriate dual symbol.  
 $(A + B)(C + D)$

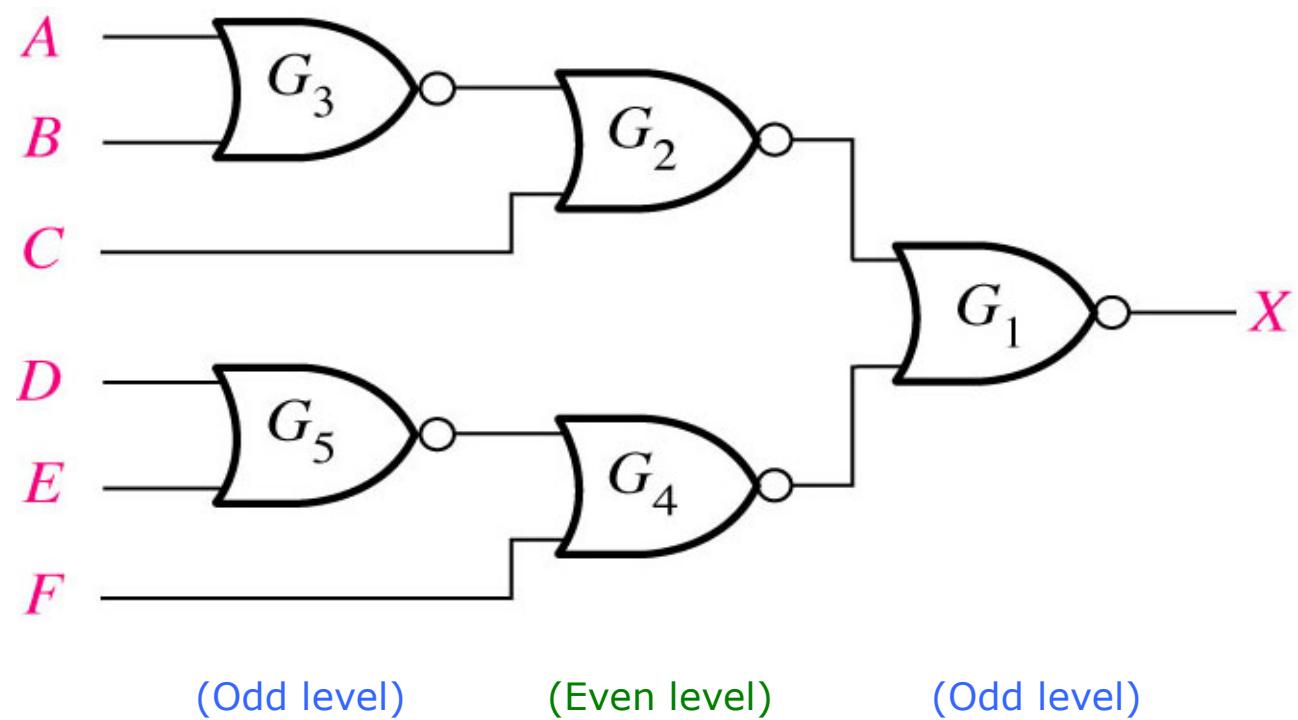
$$\begin{aligned}\overline{\overline{A}+\overline{B}+\overline{C}+\overline{D}} \\ = (\overline{\overline{A}+\overline{B}})(\overline{\overline{C}+\overline{D}}) \\ = (A+B)(C+D)\end{aligned}$$

## Solution:

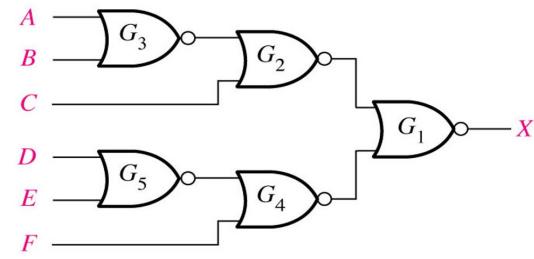
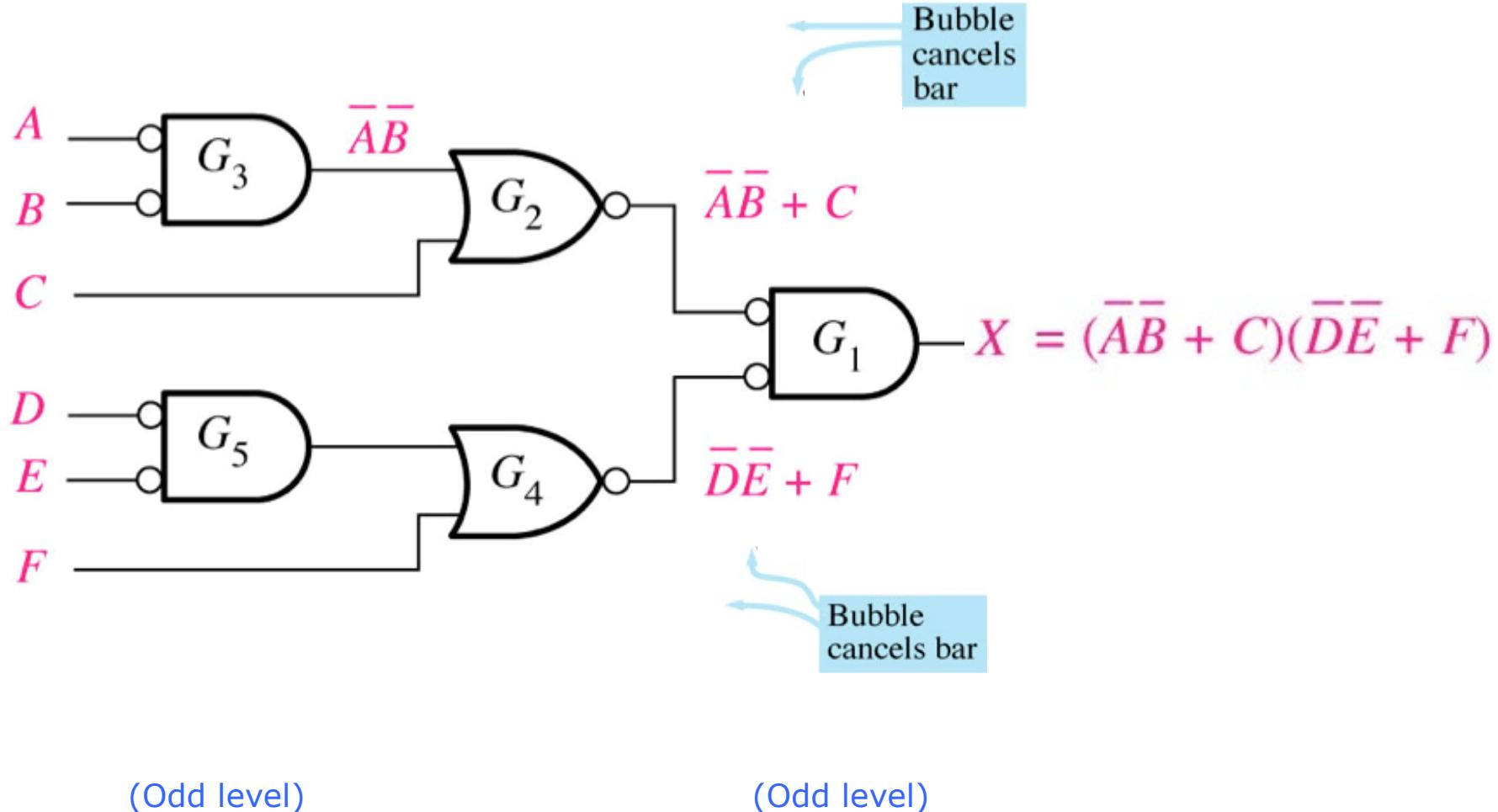


**Example:** Convert the NOR only circuit to an equivalent circuit using dual symbol.

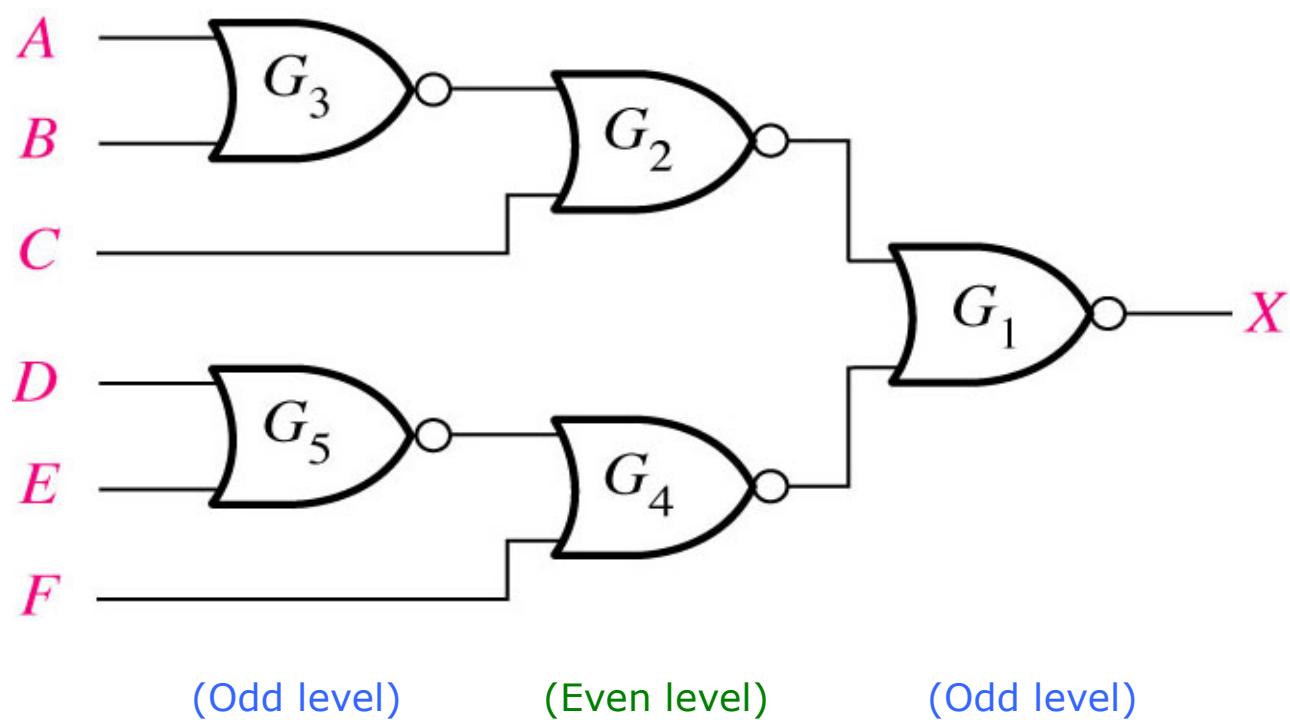
(Odd level)

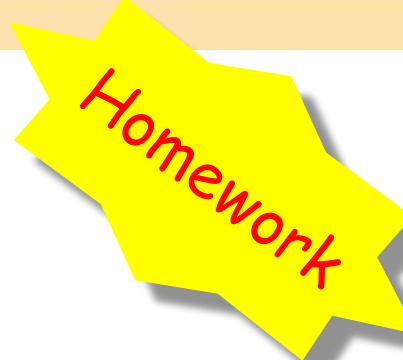


## Solution:



**Exercise 5.7:** Redraw the logic diagram and develop the output expression for the circuit using the appropriate dual symbols (**even level**).





## Homework

### Question 1

Question 1: Implement the expression using NAND logic and produce the answer using dual symbol.

$$X = \overline{\overline{A} + \overline{B} + \overline{C}}DE$$

### Question 2

Question 2: Implement the expression using NOR logic and produce the answer using dual symbol.

$$X = \overline{\overline{ABC}} + (D + E)$$



## Problem Statements

Designing a Combinational circuit  
Deriving a Boolean Expression

# Steps in designing a combinational circuits

1. Understand the problem to be solved
2. Convert the problem statement to a Boolean expression or truth table, whichever is most natural for the given problem
3. Simplify the output by using Boolean algebra, K- Map or other technique
4. Draw the circuit from the simplified output
5. Convert the simplified circuit to universal gates only
  - SOP to NAND only
  - POS to NOR only
6. Redraw the final circuit using dual symbol

continue...

The most critical stage is step 1. Understand

Determine 3 things

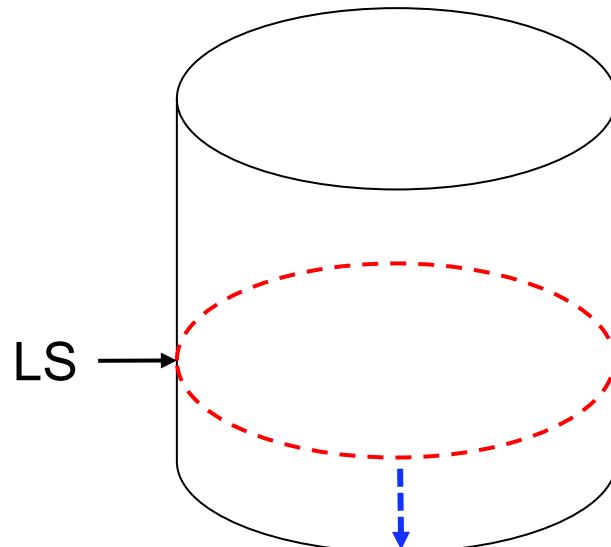
1. How many inputs and give it a variable name
2. How many outputs and give it a variable name
  - If more than one output treat each output separately, design a circuit for each output and finally combine them together
3. What is the relationship between output(s) and the inputs

## Example: (Read and try to understand)

- In a certain chemical-processing plant, a liquid chemical is used in a manufacturing process.
- The chemical is stored in **three different tanks**.
- A level sensor in each tank produces a **HIGH** voltage when the level of chemical in the tank drops **below** a specified point.
- Design a circuit that monitors the chemical level in each tank and indicates when the level in **any two of the tanks** drops below the specified point.

## 3 Tanks A, B, C

Solution:



Step1:

**Input** – 3 tanks – call it A,B,C - if drop below specified point it will input a HIGH (1) else LOW (0)

**Output** – 1 – call it X -- produces HIGH (1) if active

**Relationship** – any two tanks drop below specified point (has logic 1) the output active (HIGH)

continue...

**Relationship** – any two tanks drop below specified point (has logic 1) the output active (HIGH)

## Step2:

Convert to standard form :

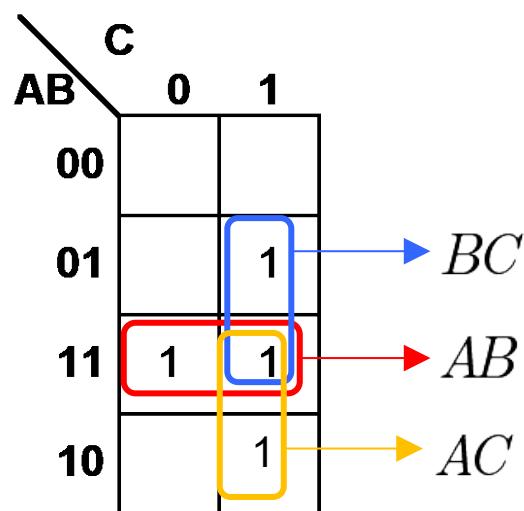
Truth table – the easiest in this case

Truth table:

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

## Step3:

Map to K-Map then simplify

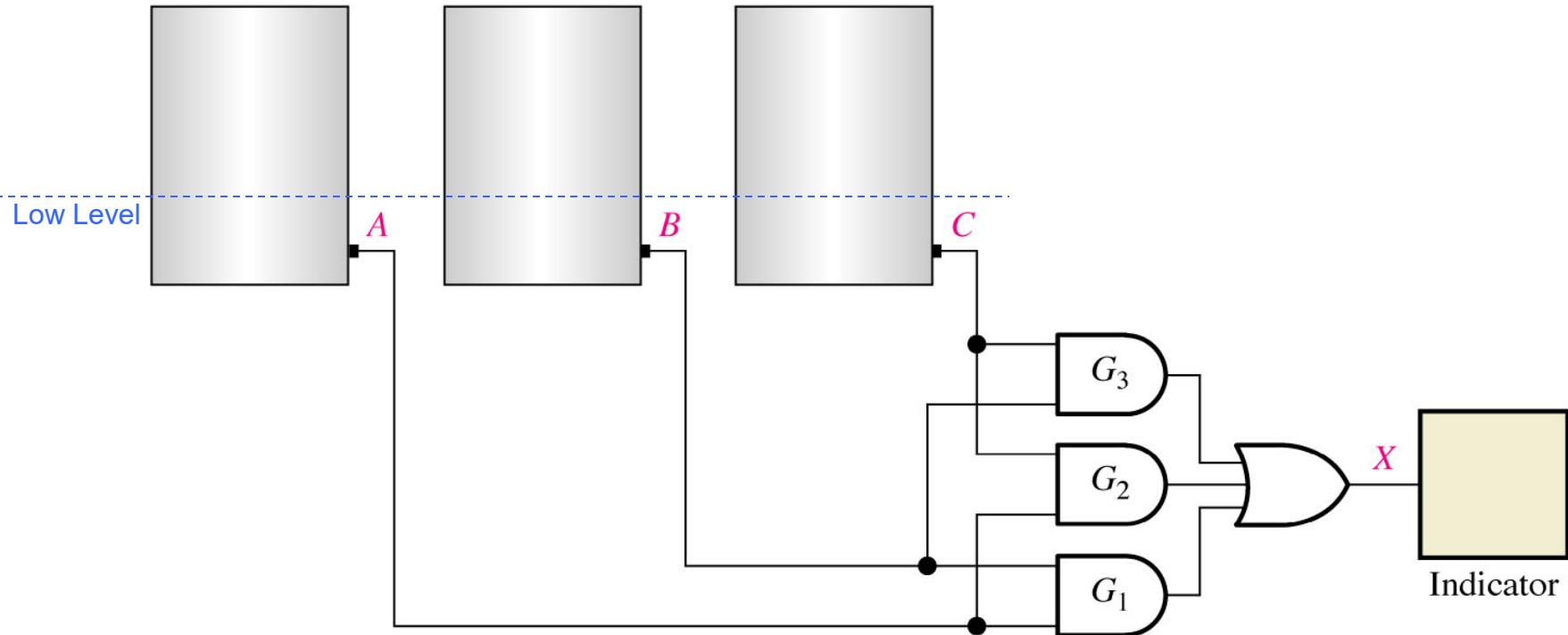


Simplified Equation:  $X = AB + AC + BC$

continue...

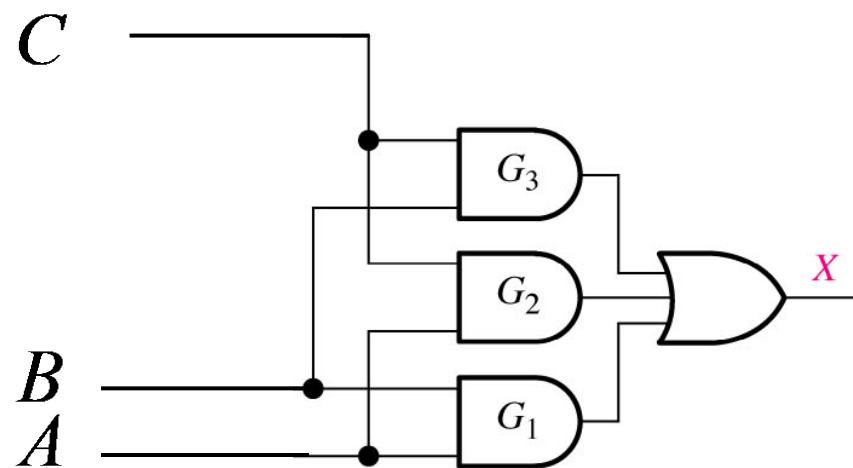
(The three different tanks with a level sensor each)

**Step4:** Draw the circuit for the simplified output  $X = AB + AC + BC$



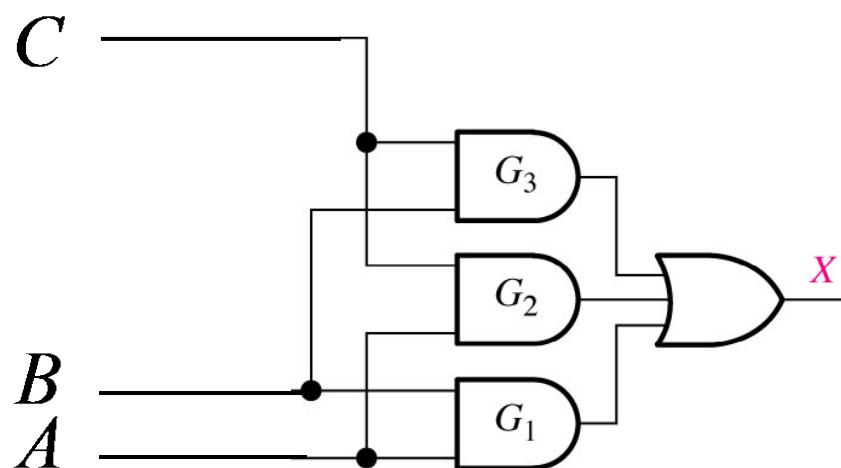
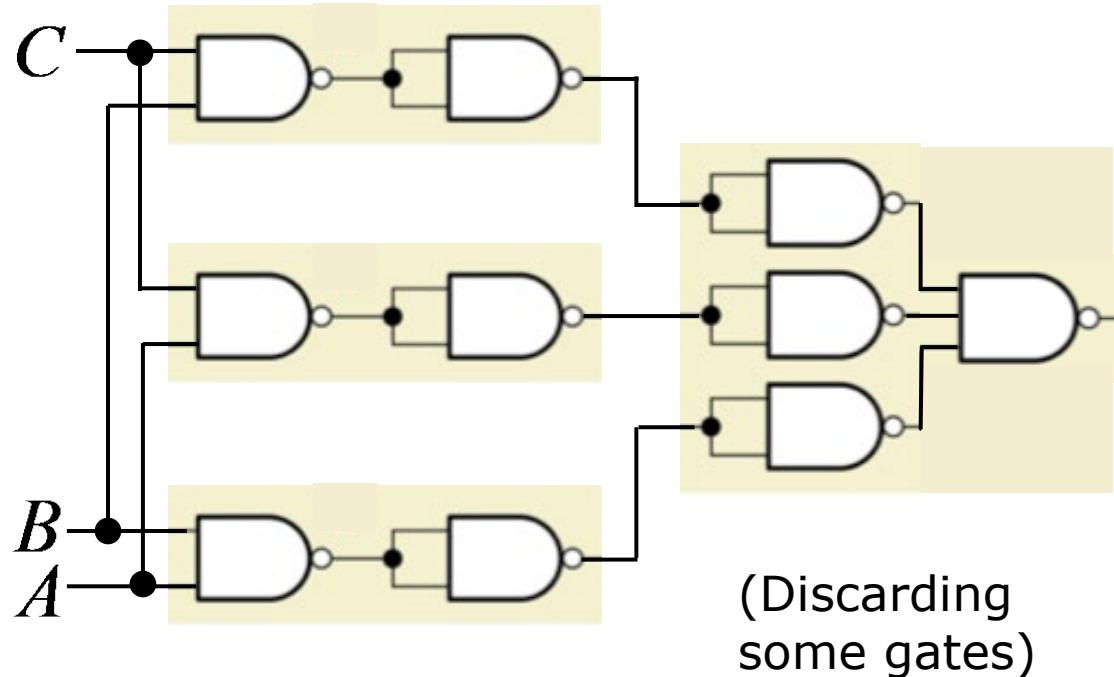
## Exercise 5.10a: (Step5)

Implement the circuit using universal gates only.



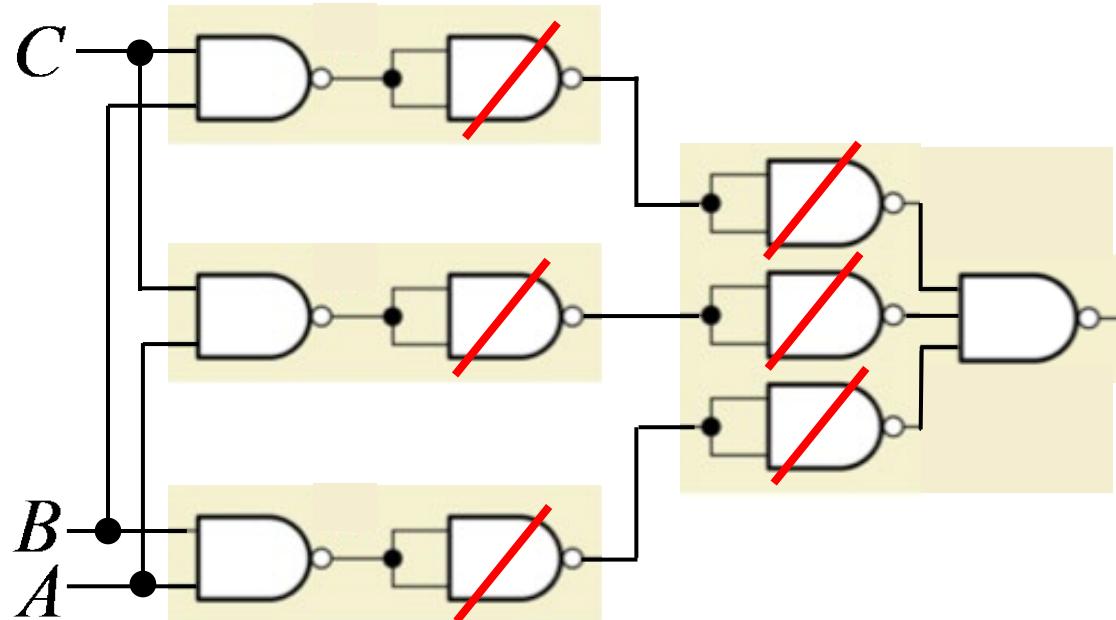
## Solution:

Using NAND gates for SOP expression

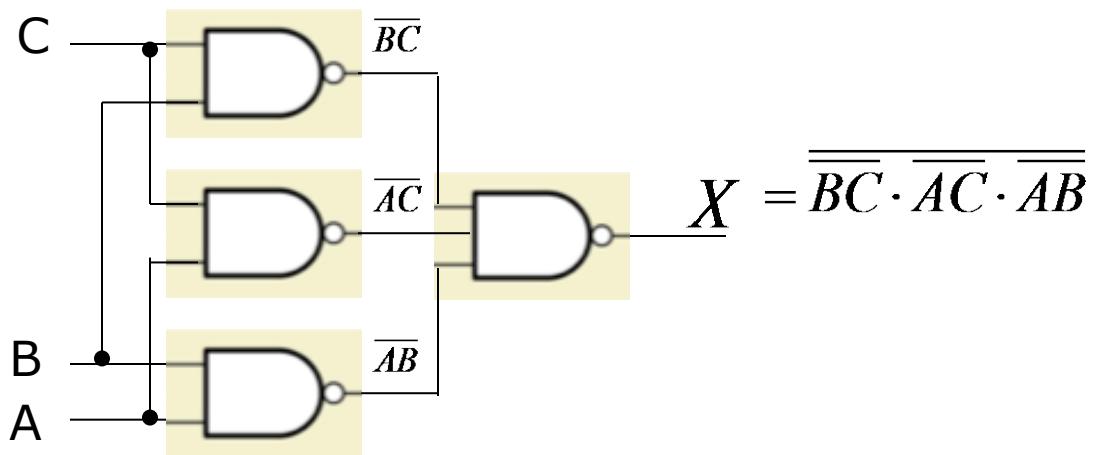


## Solution:

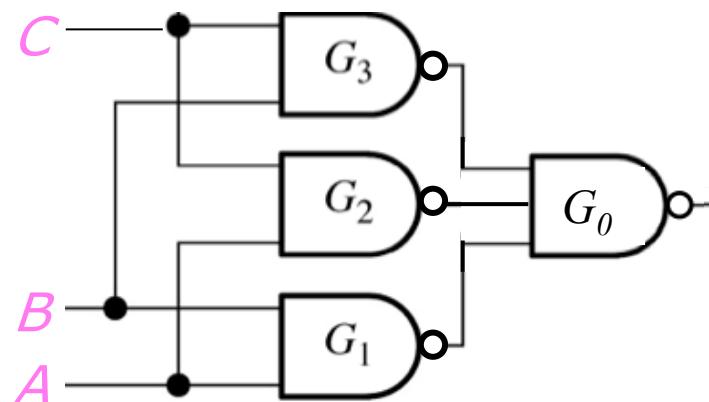
Using NAND gates for SOP expression



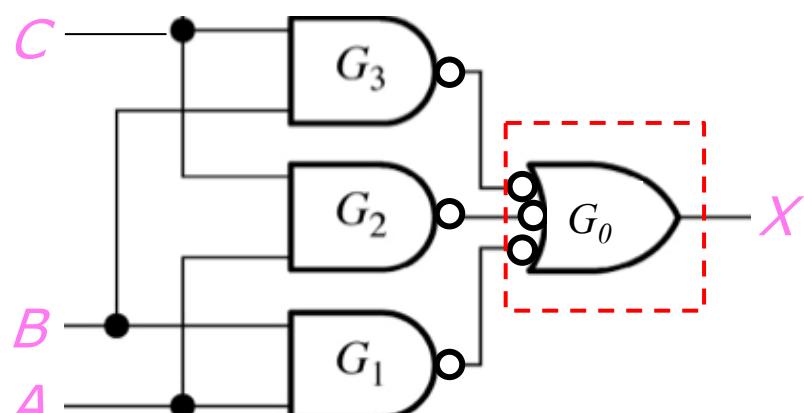
(After discarding  
some gates)



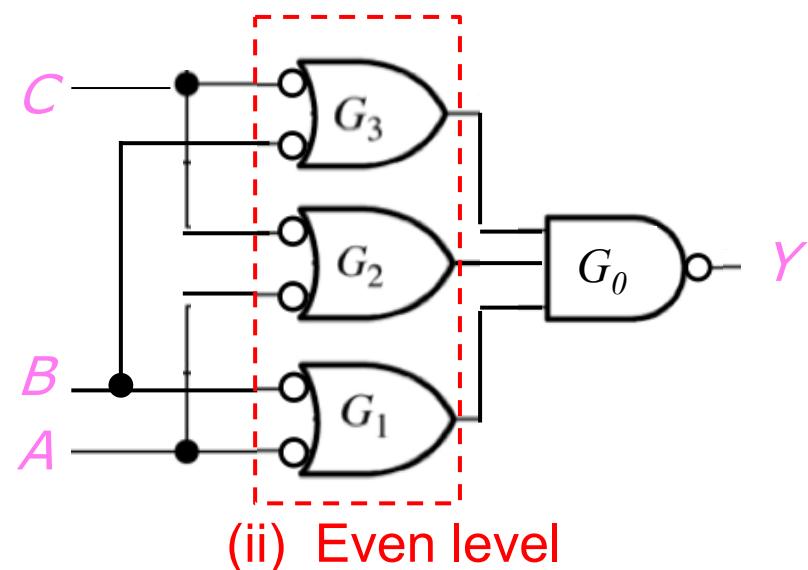
## Exercise 5.10b: (Step6) Draw the final circuit using dual symbol.



Solution:



(i) Odd level



(ii) Even level

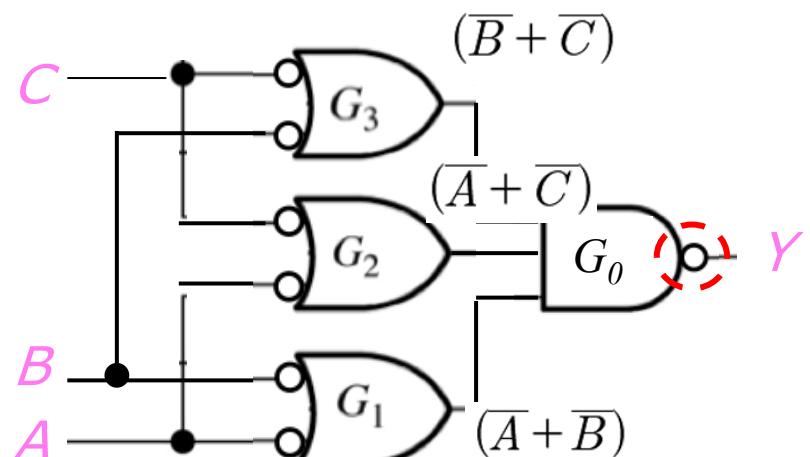
$$X = Y = AB + AC + BC$$

**Exercise 5.10c:** Proof that  $Y = AB + AC + BC$ .

$$\overline{(A+B)} \bullet \overline{(A+C)} \bullet \overline{(B+C)}$$

**Solution:**

$$\begin{aligned} Y &= \overline{(A+B)} \overline{(A+C)} \overline{(B+C)} \\ &= \overline{(A+B)} + \overline{(A+C)} + \overline{(B+C)} \\ &= \overline{\overline{A} \overline{B}} + \overline{\overline{A} \overline{C}} + \overline{\overline{B} \overline{C}} \\ &= AB + AC + BC \end{aligned}$$



(ii) Even level

## **Example: multiple output..**

Design a combinational circuit with 3 inputs A, B and C and 2 outputs X and Y.

When the binary input is 0,1, or 2 the binary output is one greater than the input.

When the binary input is 3, 4, 5, or 6 the binary output is three less than the input.

The binary input will never be 7.

When the binary input is 0,1, or 2 the binary output is one greater than the input.

When the binary input is 3, 4, 5, or 6 the binary output is three less than the input.

The binary input will never be 7.

## Solution:

### Step1:

**Input – 3** – which is A, B AND C

**Output – 2** – which is X and Y

**Note:** for N number of output, treat each output separately, go through the normal design process, it will produce N different circuits, finally combine all N circuits into one

### Relationship

for 0,1,2 the output will be 1,2,3

for 3,4,5,6 the output will be 0,1,2,3

for 7, it will never exist therefore treat it as don't care

**Note:** input A, B, and C will be combined and treated as 3 bits binary number (ABC), X and Y combined together and treated as a binary number (XY)

## Step2:

Use Truth table – the easiest in this case

Inputs			Outputs	
ABC			XY	
	000		01	
	001		10	
	010		11	

for 0,1,2 the output will be 1,2,3

for 3,4,5,6 the output will be 0,1,2,3

for 7, it will never exist therefore treat it as don't care

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

**Note:** input A, B, and C will be combined and treated as 3 bits binary number (ABC), X and Y combined together and treated as a binary number (XY)

## Step2:

Use Truth table – the easiest in this case

Inputs	Outputs
ABC	ABC

011	00
100	01
101	10
110	11

for 0,1,2 the output will be 1,2,3

for 3,4,5,6 the output will be 0,1,2,3

for 7, it will never exist therefore treat it as don't care

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



**Note:** input A, B, and C will be combined and treated as 3 bits binary number (ABC), X and Y combined together and treated as a binary number (XY)

## Step2:

Use Truth table – the easiest in this case

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

for 0,1,2 the output will be 1,2,3

for 3,4,5,6 the output will be 0,1,2,3

for 7, it will never exist therefore treat it as don't care -----

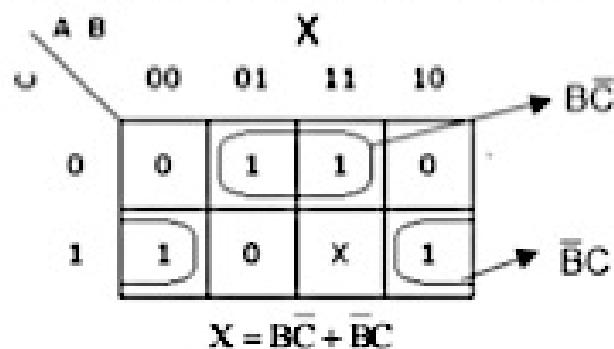
Note: input A, B, and C will be combined and treated as 3 bits binary number (ABC), X and Y combined together and treated as a binary number (XY)

## Step2:

Use Truth table – the easiest in this case

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

## Step 3 : Map the truth table to K-Map and simplify

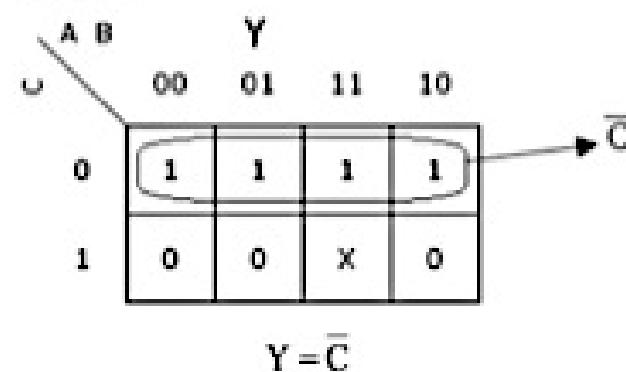
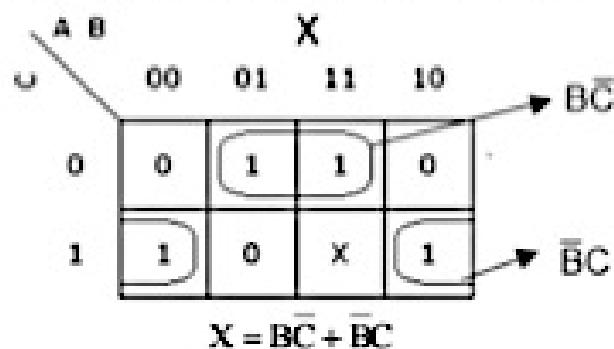


## Step2:

Use Truth table – the easiest in this case

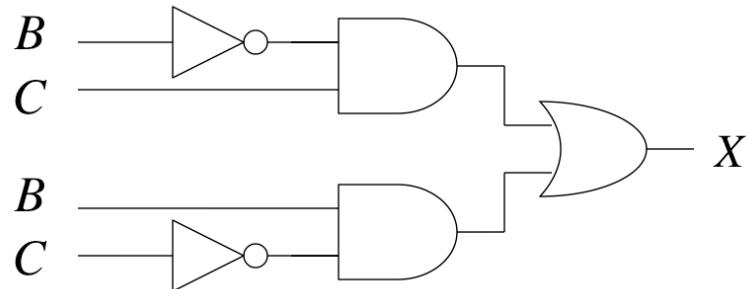
A	B	C	X	Y
0	0	0	0	1
0	0	1	1	0
0	1	0	1	1
0	1	1	0	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	1
1	1	1	X	X

## Step 3 : Map the truth table to K-Map and simplify

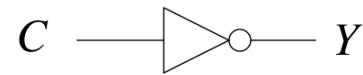


#### Step4: Draw the simplified circuit

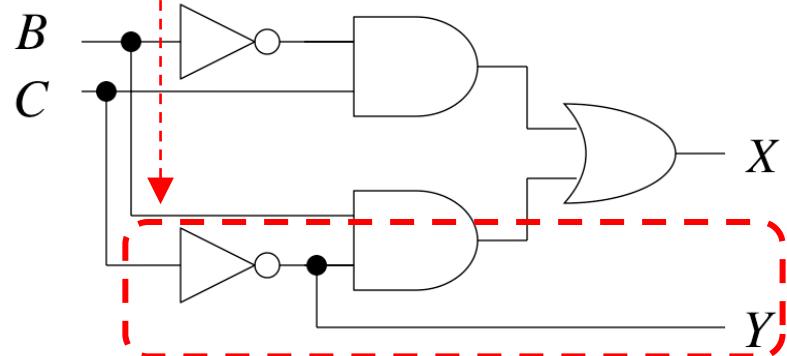
circuit for  $X = B\bar{C} + \bar{B}C$



circuit for  $Y = \bar{C}$



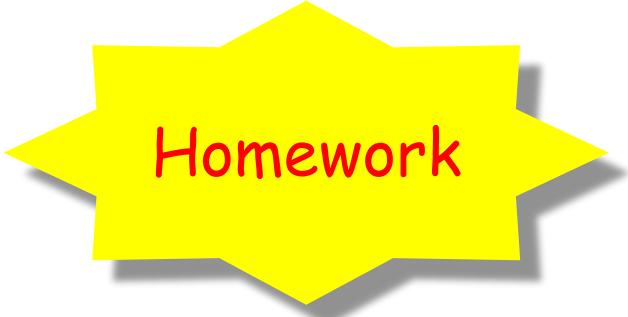
combined circuit



**Exercise 5.11:** **Step 5:** Implement the circuit using universal gates

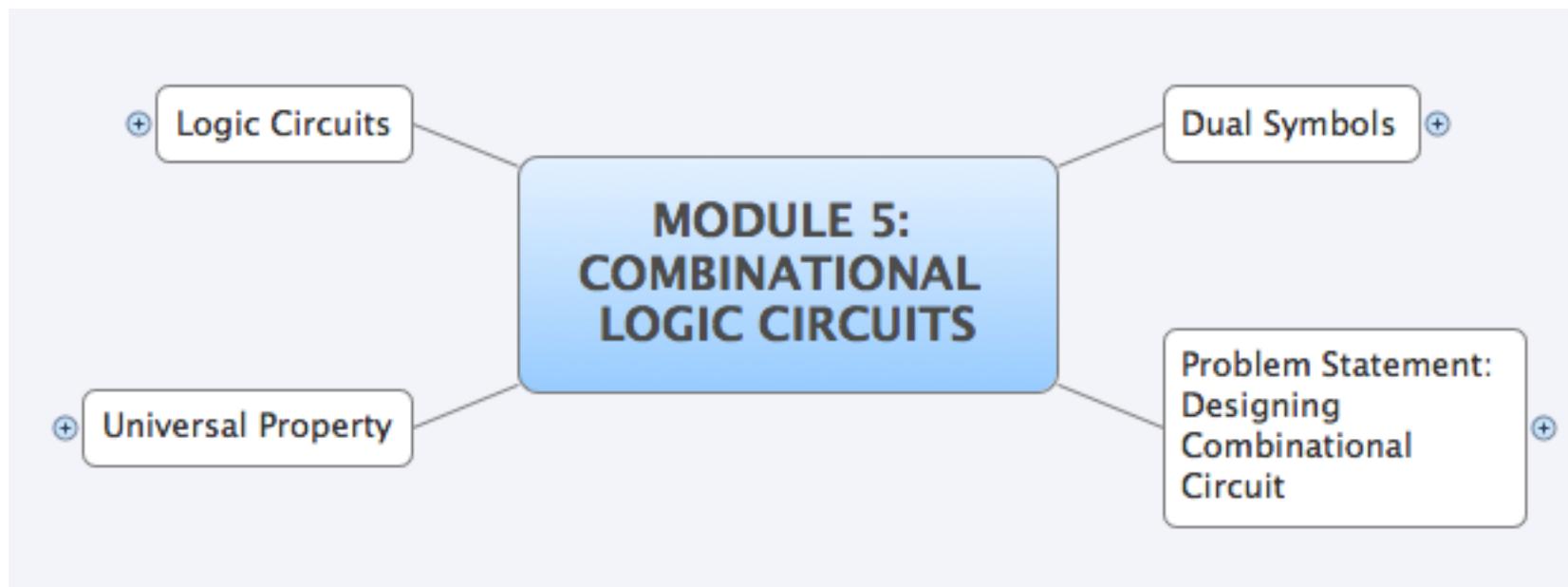
**Step 6:** Draw the final circuit using dual symbol

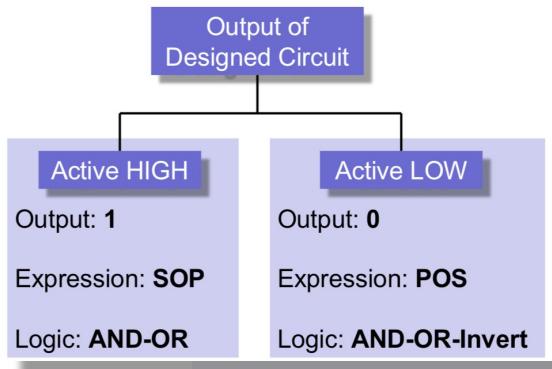
---

A yellow starburst or speech bubble shape with a black outline and a slight shadow. The word "Homework" is written in red inside it.

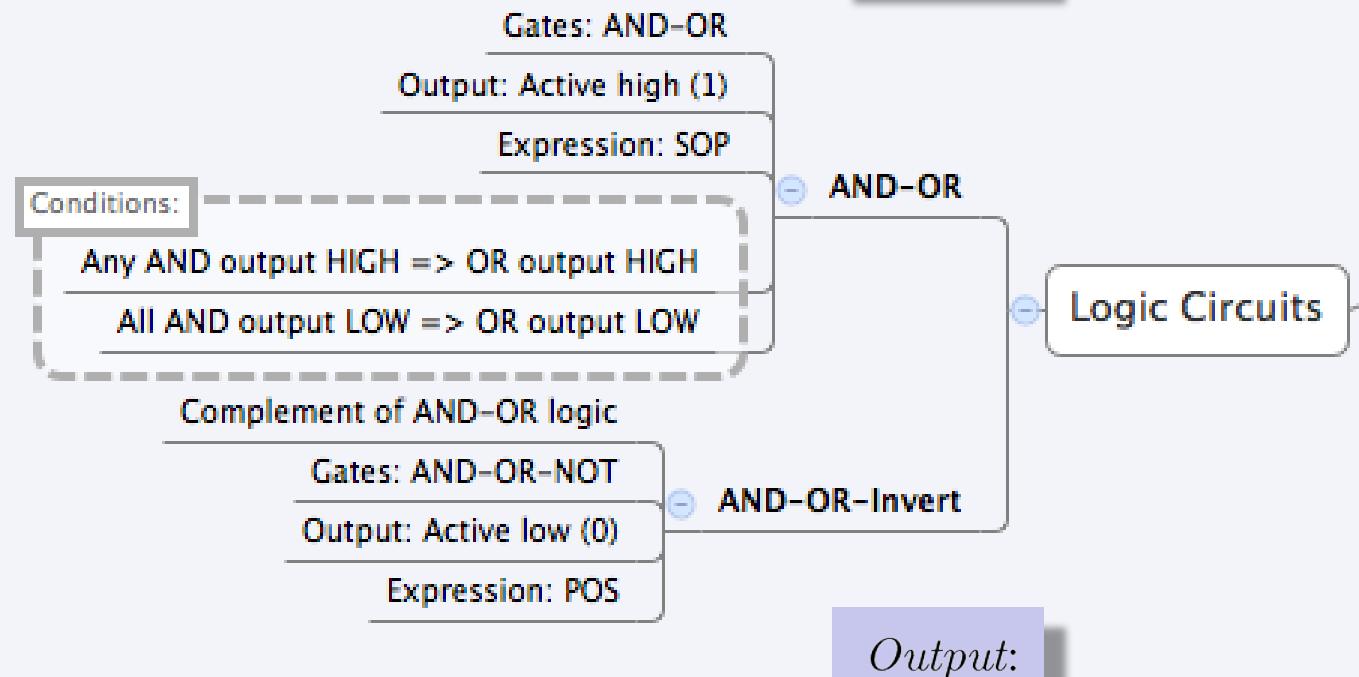
Homework

# Summary

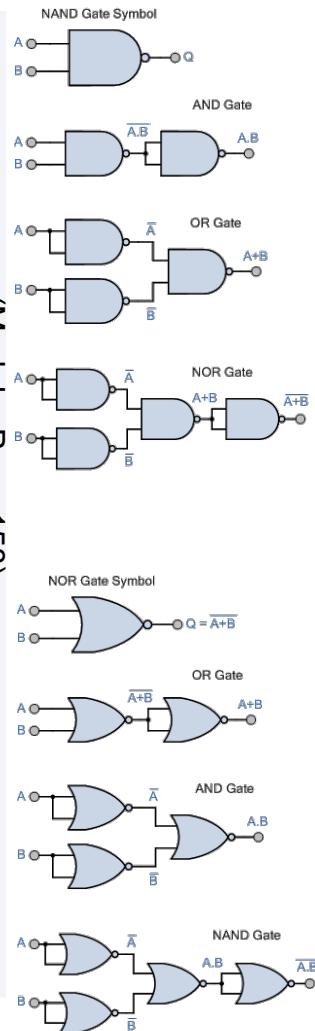
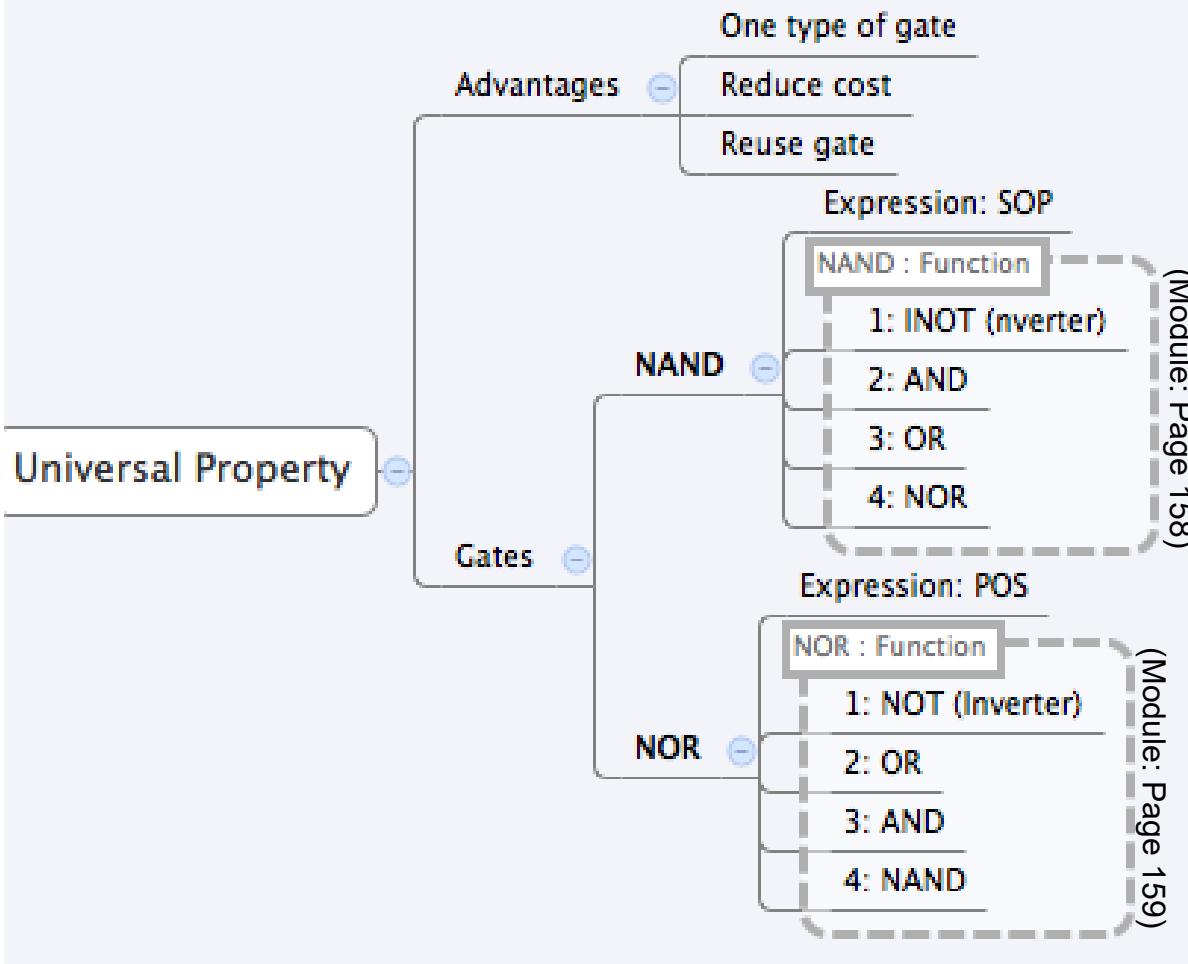


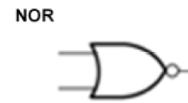
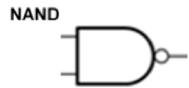


*Output:*  $X \rightarrow 1$  (Module: Page 153)



*Output:*  $\overline{X} \rightarrow 0$  (Module: Page 155)



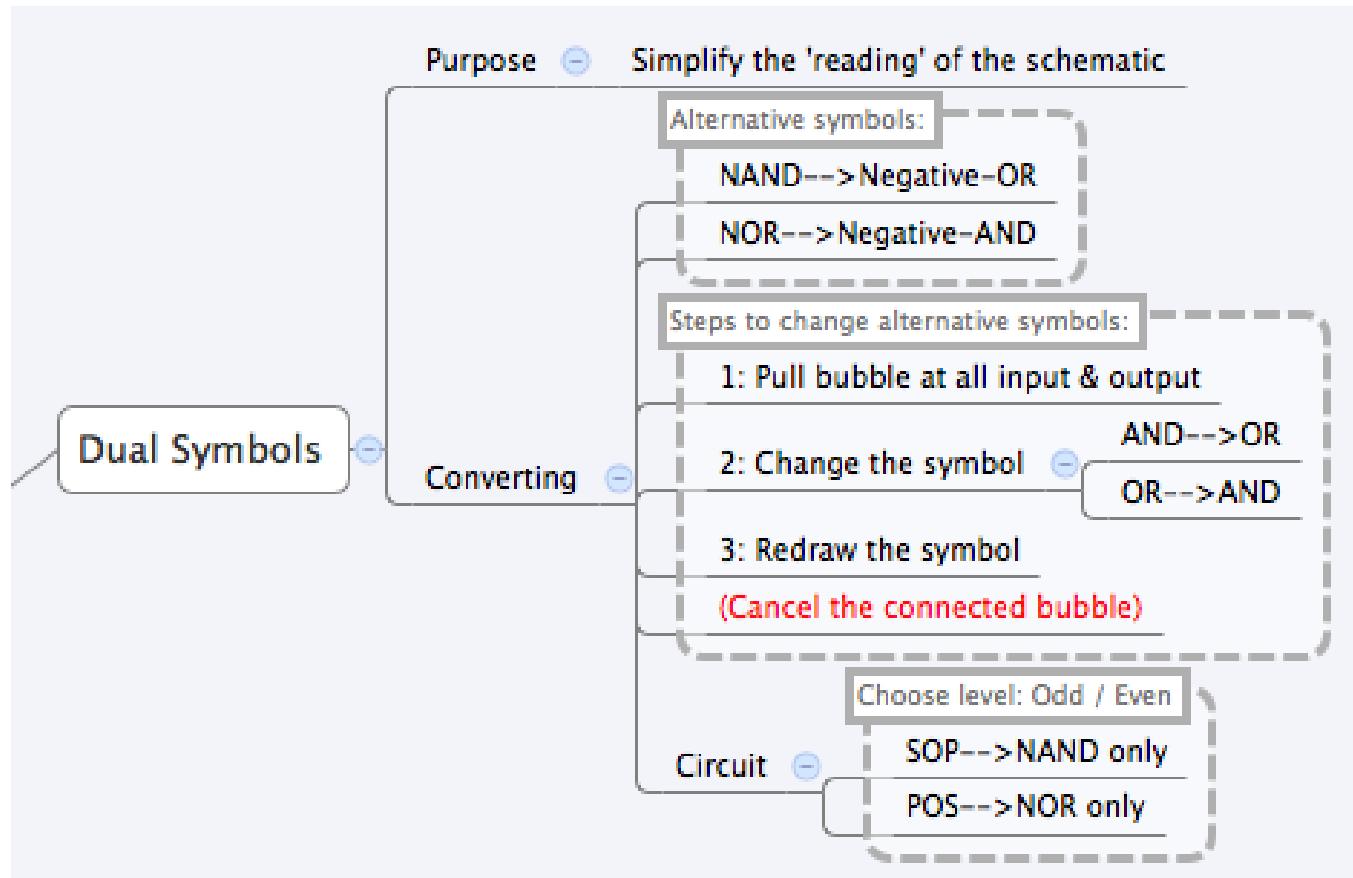


Alternate symbol for NAND – Negative OR



Alternate symbol for NOR – Negative AND

(Module: Page 160)



**Problem Statement:**  
**Designing**  
**Combinational**  
**Circuit**

(Module: Page 167)

- Steps:
- 1: Understand
  - 2: Convert problem to
  - 3: Simplify
  - 4: Draw simplified circuit
  - 5: Convert to universal gate only
  - 6: Redraw final circuit using Dual symbol