

# CHƯƠNG 3



## Đại Số Boolean và Các Cổng Logics

Giảng viên: ThS. Phan Như Minh

# NỘI DUNG

- **Cổng Logic cơ bản AND, OR, NOT**
  - Mạch Logic  $\Rightarrow$  Biểu thức Đại Số
  - Biểu thức Đại Số  $\Rightarrow$  Mạch Logic
- **Cổng Logic NAND và NOR**
- **Luận lý đại số Boolean**

# 1. Biểu diễn logic số (Đại số Boolean)

- ❖ Đại số Boolean (đại số logic) là một tập hợp các đối tượng có hai trạng thái: có hoặc không.
- ❖ **Biến logic:** đại lượng biểu diễn bằng ký hiệu nào đó, lấy giá trị 0 hoặc 1
- ❖ **Hàm logic:** nhóm các biến logic liên hệ với nhau qua các phép toán logic, lấy giá trị 0 hoặc 1.

# 1. Biểu diễn logic số (Đại số Boolean)

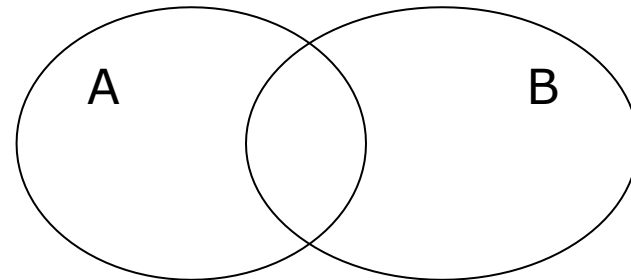
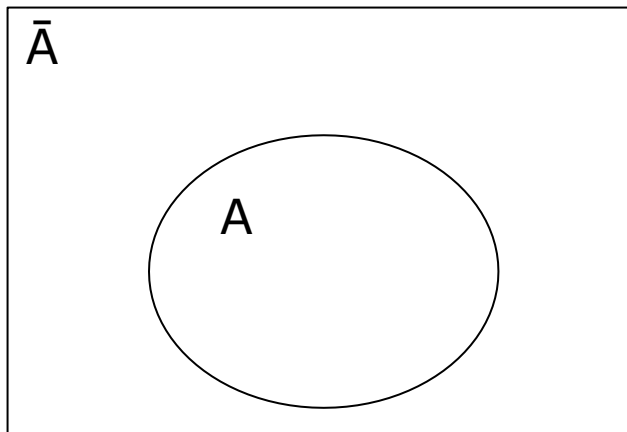
- Đại Số Boolean chỉ xử lý 2 giá trị duy nhất (2 trạng thái logic): **0** và **1**

Logic 0	Logic 1
False	True
Off	On
LOW	HIGH
No	Yes
Open switch	Closed switch

# Các phép toán cơ bản tập hợp

- Phép giao: ký hiệu  $\cap$
- Phép hợp: ký hiệu  $\cup$
- Phép hiệu: ký hiệu  $\setminus$
- Phép bù: ký hiệu  $^c$

$$A, \bar{A}, A \cup B, A \cap B, A \setminus B$$



# Các tiên đề của đại số Boole

Tiên đề	Tính chất	Minh họa
1	Giao hoán	$x + y = y + x$ $x.y = y.x$
2	Phân bố	$x(y + z) = (x.y) + (x.z)$ $x + yz = (x + y)(x + z)$
3	Tồn tại	$x + 0 = x$ $x.1 = x$
4	Tồn tại phần tử bù của x	$x + x = 1$ $x.x = 0$
5	Kết quả các phép toán giữa các tập hợp là duy nhất	Phép cộng logic (+): phép hợp Phép nhân nhân (.): phép giao

# Tổng hợp các định lý cơ bản

Stt	Tên gọi	Dạng tích	Dạng tổng
1	Đồng nhất	$X.1 = X$	$X + 0 = X$
2	Phần tử 0, 1	$X.0 = 0$	$X + 1 = 1$
3	Bù	$X.\bar{X} = 0$	$X + \bar{X} = 1$
4	Bất biến	$X.X = X$	$X + X = X$
5	Hấp thụ	$X + X.Y = X$	$X.(X + Y) = X$
6	Phủ định kép	$\bar{\bar{X}} = X$	
7	Định lý DeMorgan	$\overline{(X.Y.Z...)} = \bar{X} + \bar{Y} + \bar{Z} + ...$	$\overline{(X + Y + Z + ...)} = \bar{X}.\bar{Y}.\bar{Z}...$

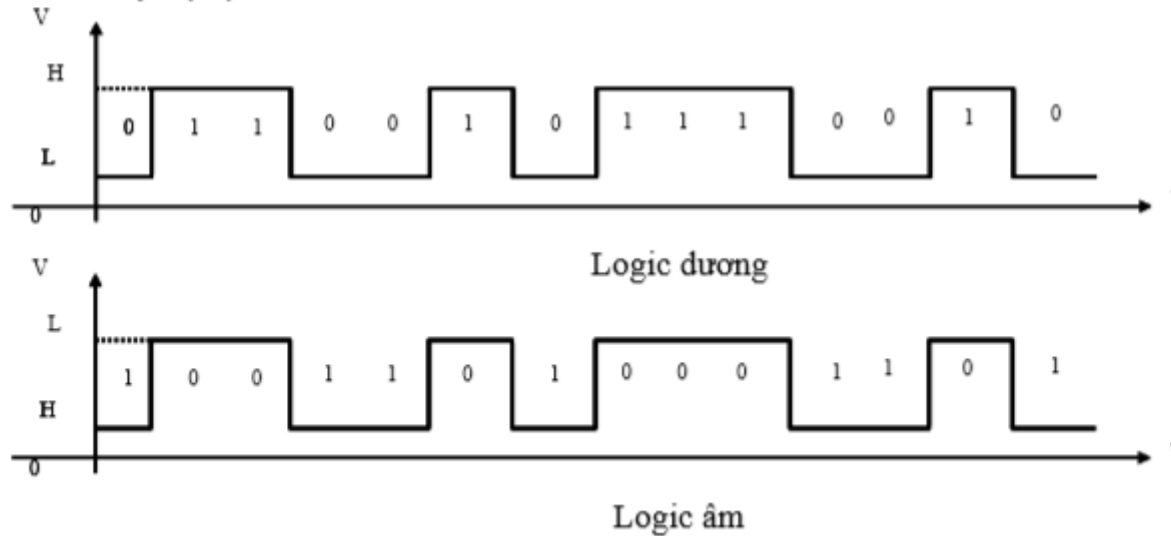
# Các phần tử của hàm Boole

- ❖ Để nhận biết và biểu diễn các giá trị 0 và 1, trong các thiết bị điện tử giá trị này được biểu diễn dưới dạng các mức điện thế.
- ❖ Các mức điện thế này phụ thuộc bản chất từng loại linh kiện bán dẫn khác nhau như:
  - TTL (Transistor Transistor Logic)
  - ECL (Emitter Coupled Logic)
  - CMOS (Complementary Metal Oxide Semiconductor)



# Logic dương và logic âm

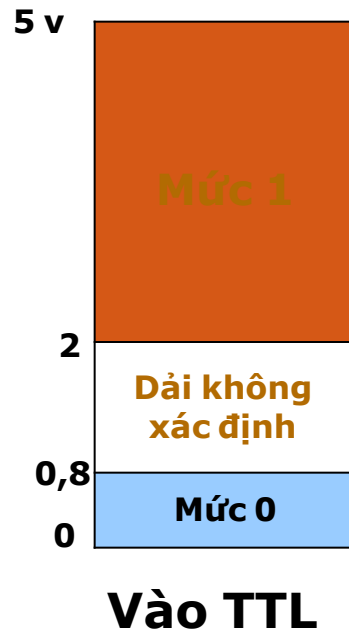
- *Logic dương*: điện thế mức cao (H) luôn lớn hơn điện thế mức thấp (L).
- *Logic âm*: điện thế mức cao (H) luôn nhỏ hơn điện thế mức thấp (L).



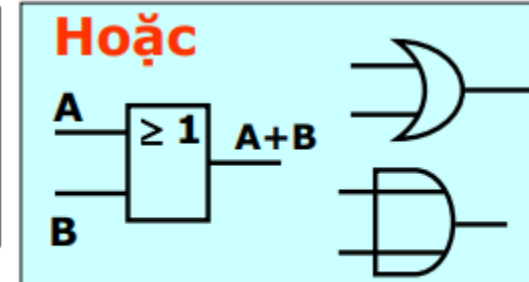
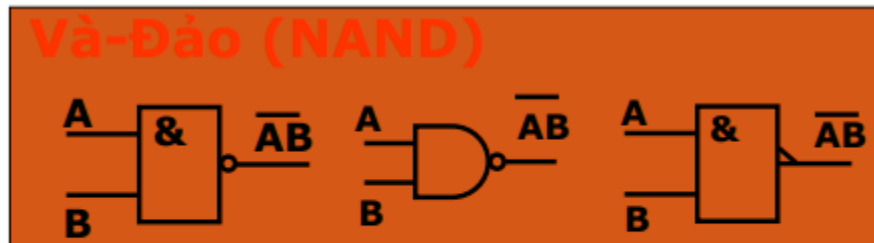
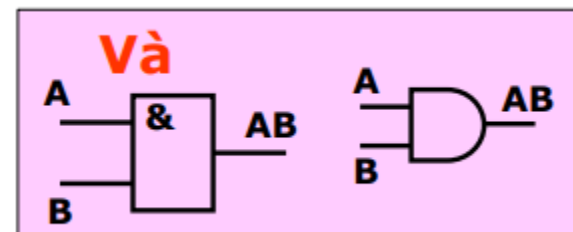
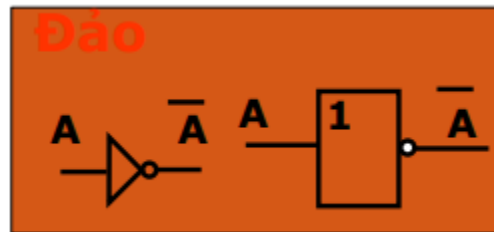
# Minh họa

## ❖ Đặc tính điện

- Các mức logic của họ TTL

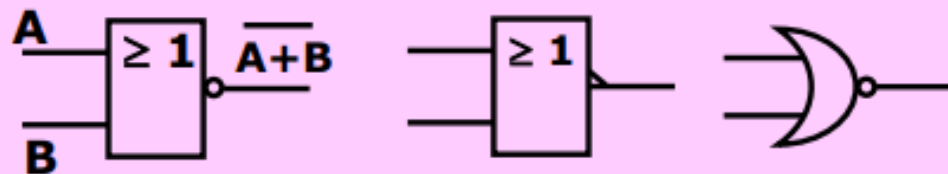


# Ký hiệu các phần tử logic cơ bản

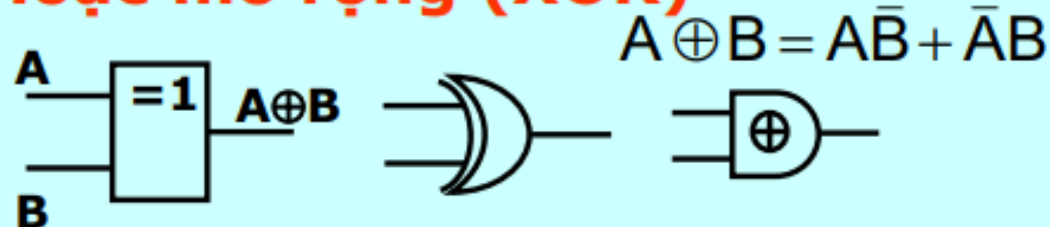


# Ký hiệu các phần tử logic cơ bản

## Hoặc-Đảo (NOR)



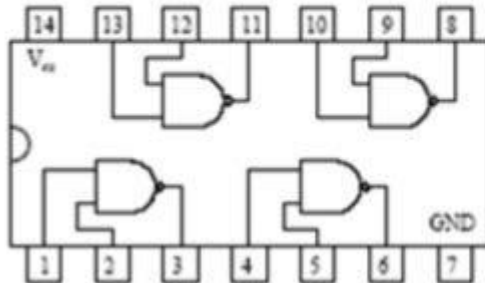
## Hoặc mở rộng (XOR)



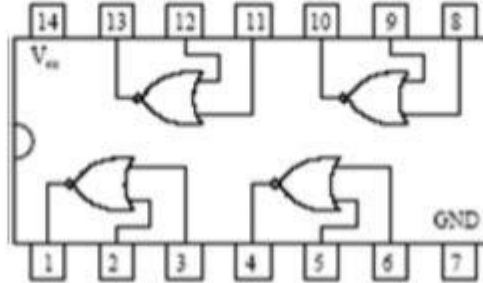
AB	F
00	0
01	1
10	1
11	0

# Sơ đồ chân của một số IC họ TTL

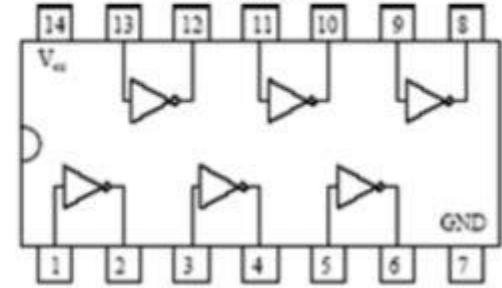
**74LS00**  
NAND



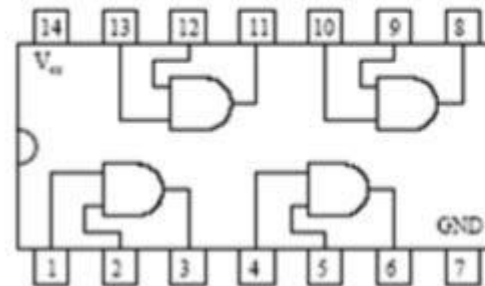
**74LS02**  
NOR



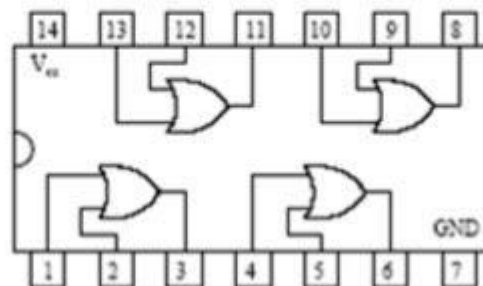
**74LS04**  
NOT



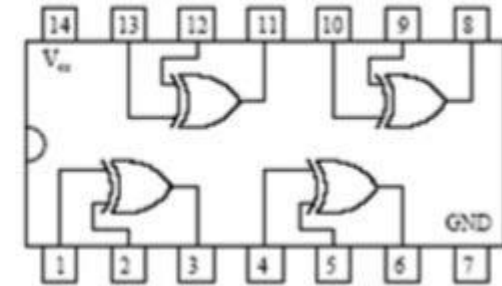
**74LS08**  
AND



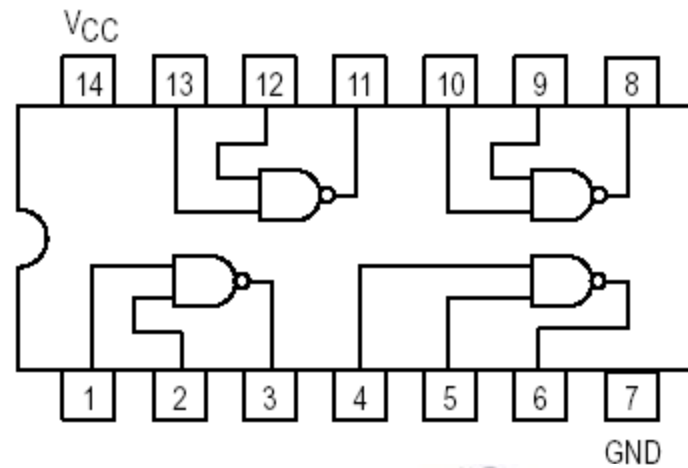
**74LS32**  
OR



**74LS86**  
XOR



# Minh họa



## 2. Cổng Logic Cơ Bản

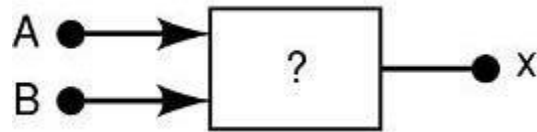
- Đại Số Boolean chỉ xử lý 2 giá trị duy nhất (2 trạng thái logic): **0** và **1**

Logic 0	Logic 1
False	True
Off	On
LOW	HIGH
No	Yes
Open switch	Closed switch

- 3 cổng logic cơ bản:
  - **OR**, **AND** và **NOT**

# Bảng Sự thật / Chân trị

- Mô tả các mối quan hệ giữa inputs và outputs của một mạch logic



Inputs		Output
A	B	x
0	0	1
0	1	0
1	0	1
1	1	0

- Số lượng các mục tương ứng với số inputs
  - A 2-input bảng sẽ có ? mục
  - A 3-input bảng sẽ có ? mục



# Cổng Logic OR

- Biểu thức Boolean cho cổng logic **OR** có hoạt động:  
 –  $X = A + B$  — Đọc là “X bằng A OR B”

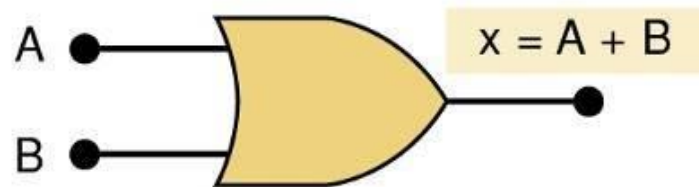
Dấu + không có nghĩa là phép cộng thông thường, mà là ký hiệu cho cổng logic OR

- Bảng sự thật và biểu diễn cổng logic OR có 2 inputs:

OR

A	B	$x = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

(a)



OR Gate

(b)

# Cổng Logic AND

- **Cổng logic AND** thực hiện tương tự như phép nhân:

–  $X = A \bullet B$  — Đọc là “X bằng A AND B”

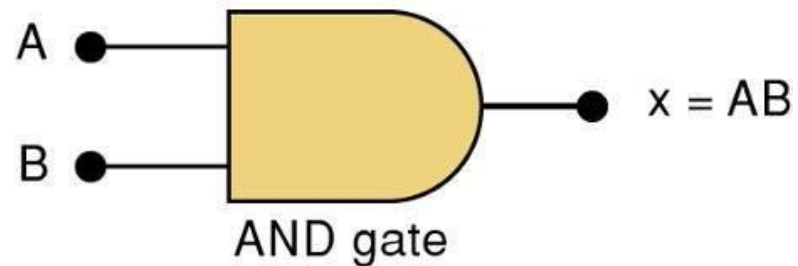
Dấu  $\bullet$  không có nghĩa là phép nhân thông thường, mà là ký hiệu cho cổng logic AND

- Bảng sự thật và biểu diễn cổng logic AND có 2 inputs:

AND

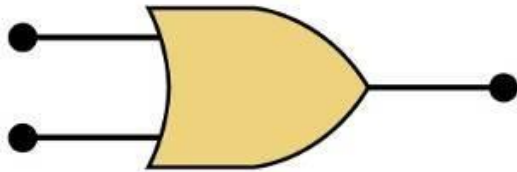
A	B	$x = A \bullet B$
0	0	0
0	1	0
1	0	0
1	1	1

(a)



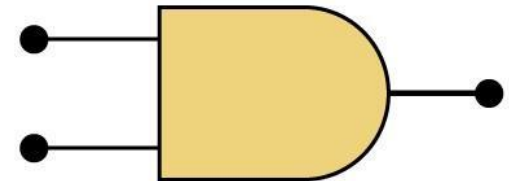
(b)

# OR vs AND



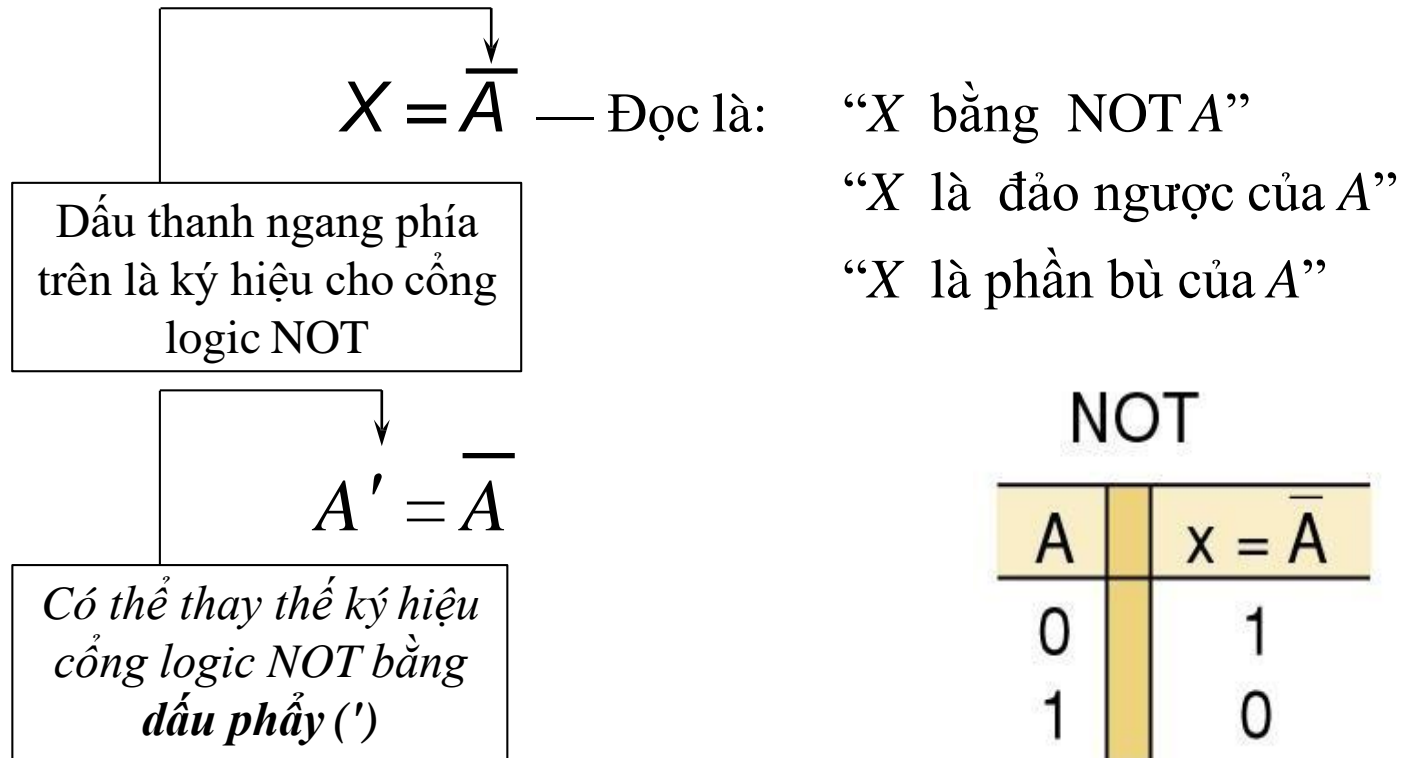
Ký hiệu của cổng logic OR có nghĩa là output sẽ có trạng thái là HIGH khi có bất kỳ input nào có trạng thái là HIGH(CAO)

Ký hiệu của cổng logic AND có nghĩa là output sẽ có trạng thái là HIGH khi tất cả các input đều có trạng thái là HIGH



# Cổng Logic NOT

- Biểu thức Boolean đối với cổng logic NOT



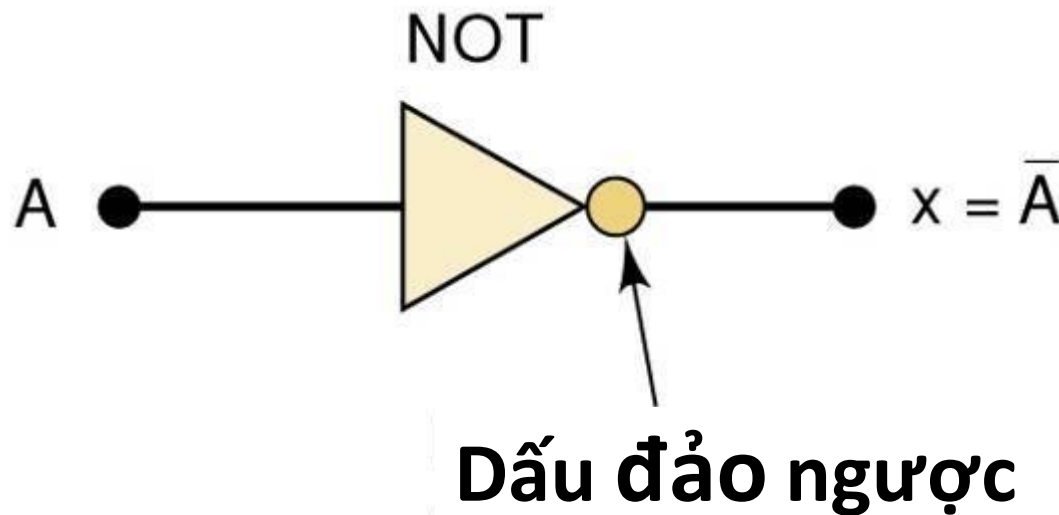
NOT

A	$x = \overline{A}$
0	1
1	0

Bảng sự thật cổng  
Logic NOT

# Cổng Logic NOT

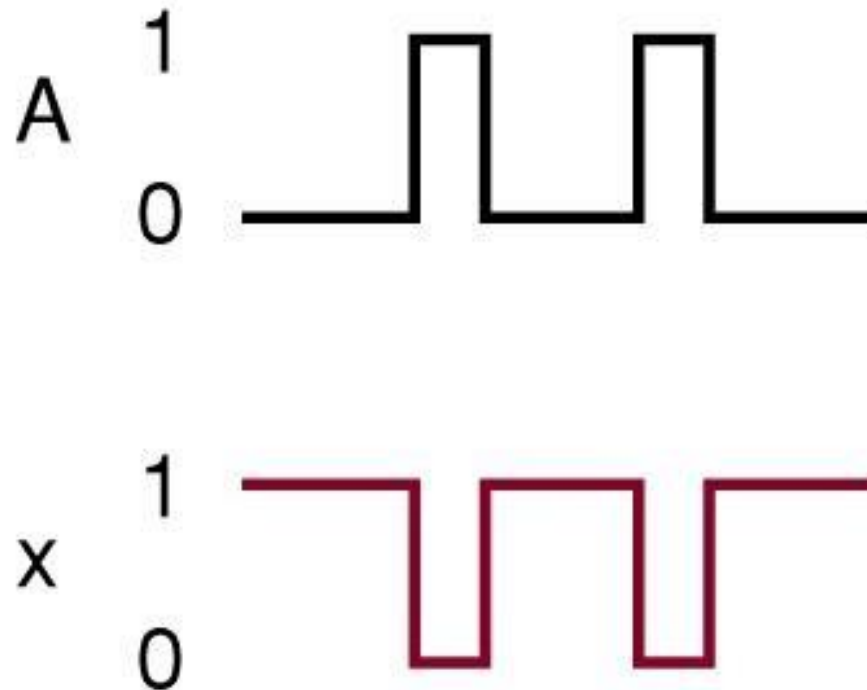
- Cổng logic NOT có thể gọi chung là **INVERTER**



Cổng logic này luôn luôn chỉ có duy nhất 1 input, và trạng thái của output sẽ đối nghịch với trạng thái của input

# Cổng Logic NOT

Cổng INVERTER nghịch đảo (*phản bù*) trạng thái tín hiệu của các inputs tại các điểm trong cùng bước sóng



Bất cứ khi nào có: input = 0, output = 1, và ngược lại

# Cổng Logic Cơ Bản

***OR***

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$

***AND***

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

***NOT***

$$\overline{0} = 1$$

$$\overline{1} = 0$$

Ba cổng logic Boolean cơ bản có thể mô tả được bất kỳ mạch logic nào

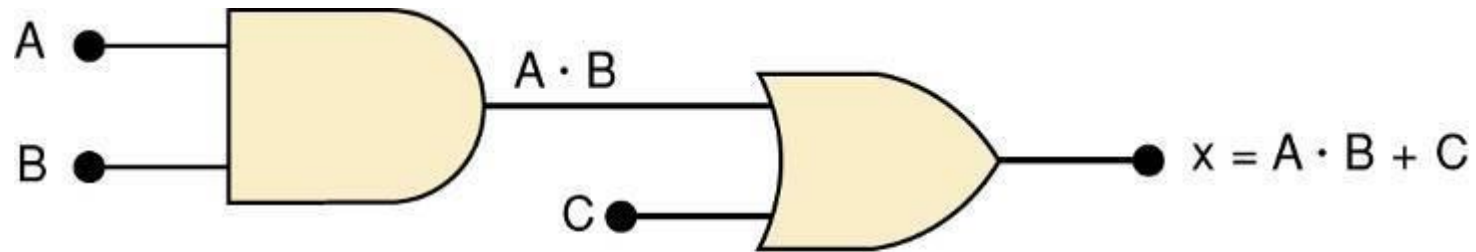


**Mạch Logic  $\Rightarrow$  Biểu thức đại số**

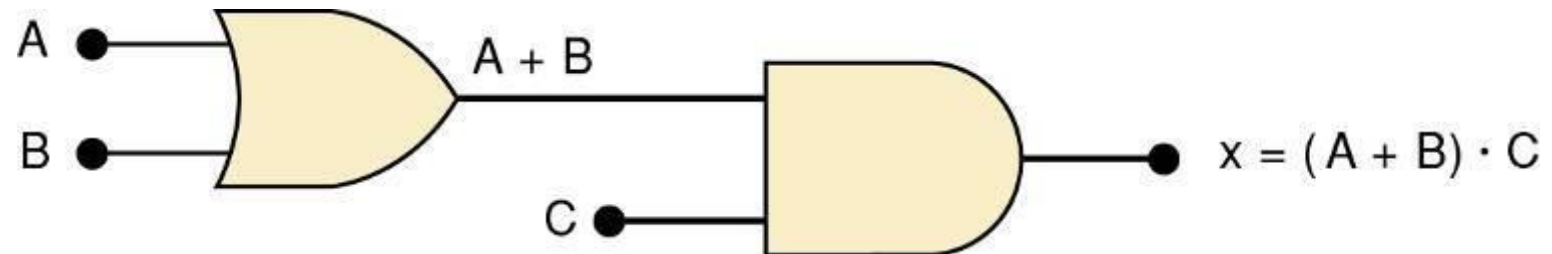


# Mô tả mạch logic đại số

- Nếu một biểu thức có chứa cả hai cổng Logic **AND** và **OR**, thì cổng logic **AND** sẽ được thực hiện trước :



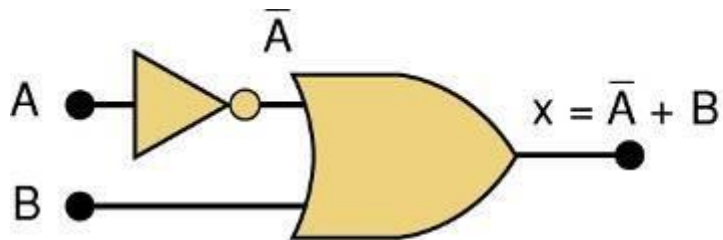
- Trừ khi có một dấu ngoặc trong biểu thức



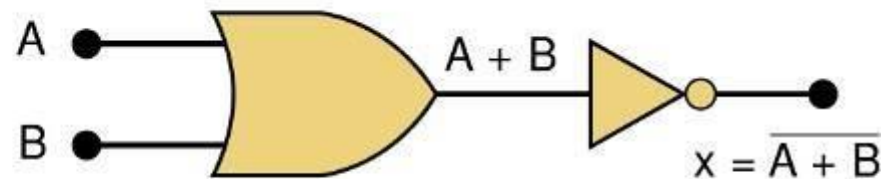
(b)

# Mô tả mạch logic đại số

- Bất cứ khi nào có sự xuất hiện của cổng logic INVERTER trong mạch, output sẽ có giá trị tương đương với input, kèm theo dấu thanh ngang — trên đầu của output
  - Input A qua một inverter sẽ có output là  $\bar{A}$

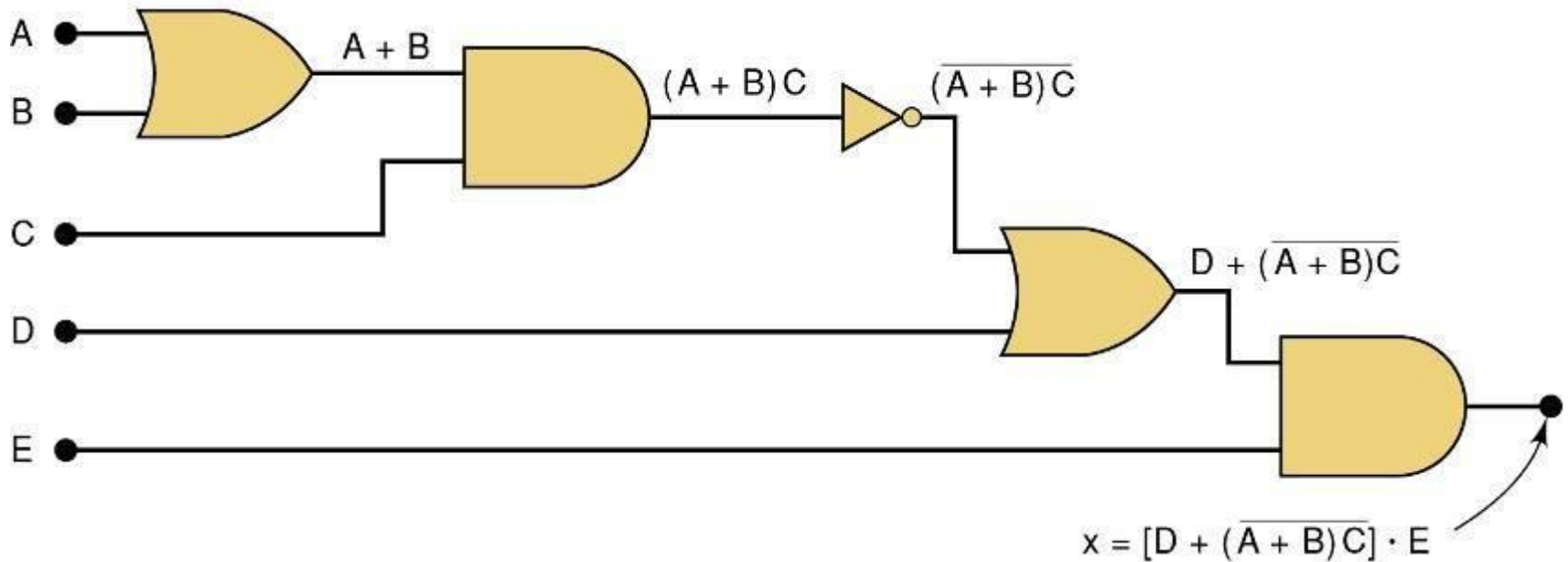
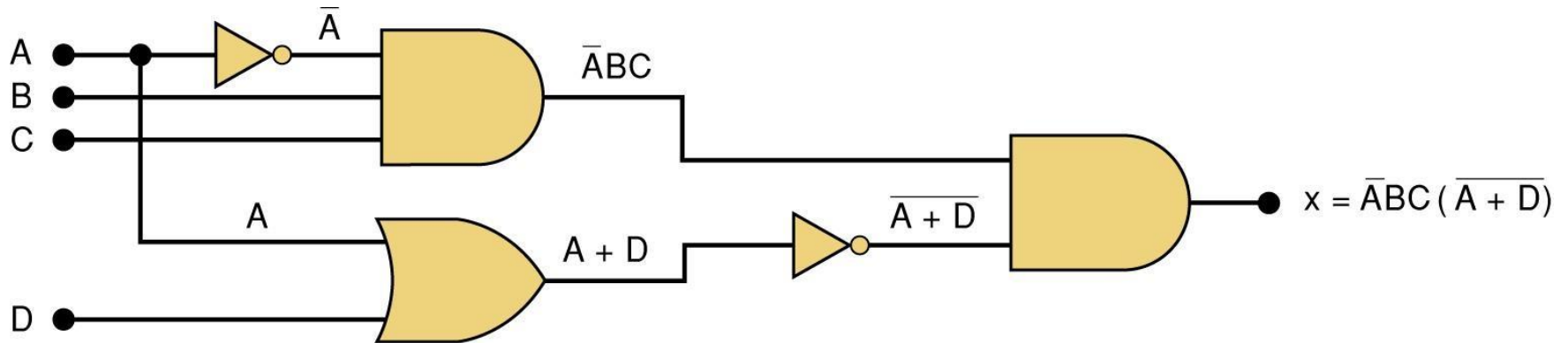


(a)



(b)

# Ví Dụ



# Đánh giá OUTPUTs của mạch logic

• Ex: 
$$X = \overline{A\bar{B}C(D + \bar{E}) + FG}$$

Quy tắc đánh giá một biểu thức Boolean:

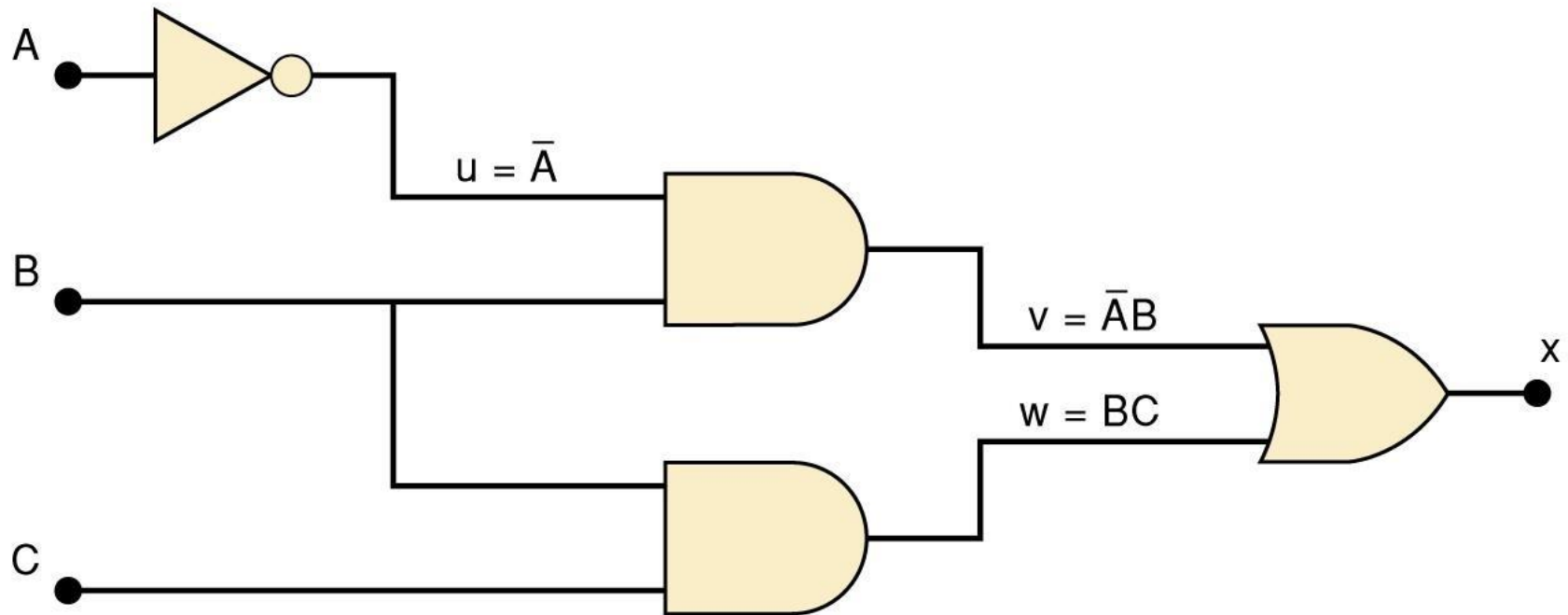
- Thực hiện tất cả đảo ngược đối với các inputs đơn trước
- Thực hiện xử lý tất cả các phép tính trong ngoặc trước
- Thực hiện xử lý cổng logic AND trước rồi mới đến cổng logic OR, trừ khi trường hợp cổng logic OR ở trong ngoặc trước
- Nếu cả một biểu thức có thanh ngang trên đầu, thực hiện các phép tính bên trong biểu thức trước, và sau đó đảo ngược kết quả lại

# Đánh giá OUTPUTs của mạch logic

- Cách tốt nhất để phân tích một mạch gồm có nhiều cổng logic khác nhau là sử dụng **bảng sự thật**
  - Cho phép chúng ta có thể phân tích một cổng hoặc một tổ hợp các cổng logic có trong mạch **cùng một lúc**
  - Cho phép chúng ta dễ dàng kiểm tra lại hoạt động của mạch logic tổ hợp một cách chính xác nhất
  - Bảng sự thật giúp ích trong việc phát hiện và xử lý lỗi hay sự cố xuất hiện có trong mạch logic tổ hợp

# Evaluating Logic Circuit Outputs

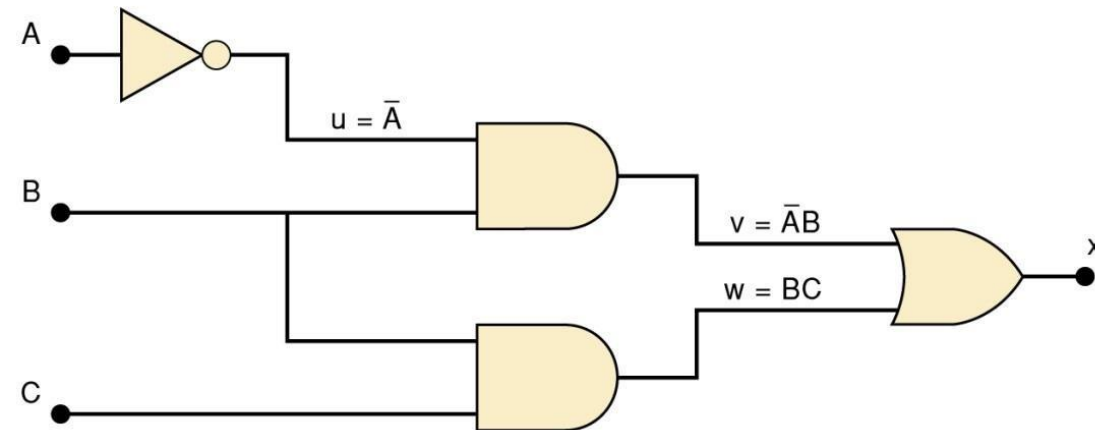
- Đánh giá outputs của mạch logic sau:



# Đánh giá OUTPUTs của mạch logic

- Bước 1: Liệt kê tất cả các inputs có trong mạch logic tổ hợp
- Bước 2: Tạo ra một cột trong bảng sự thật cho mỗi tín hiệu

trung gian (node)

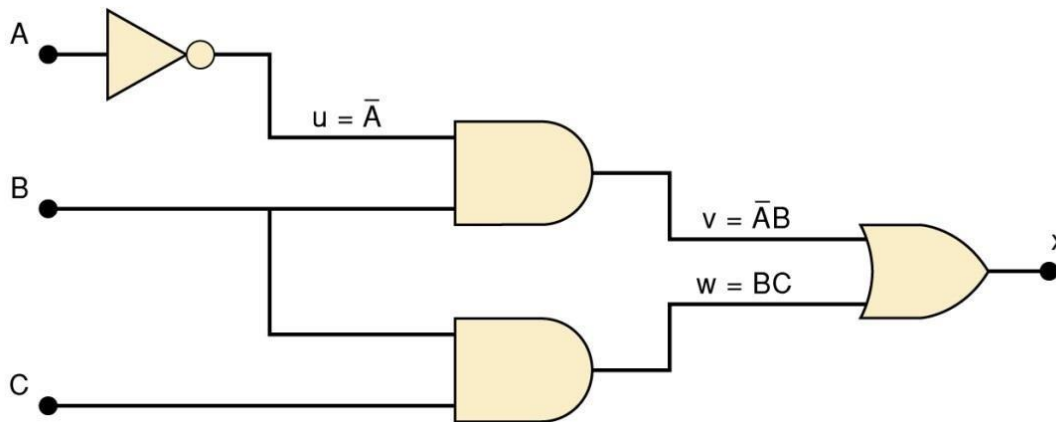


A	B	C	$u = \bar{A}$	$v = \bar{A}B$	$w = BC$	$x = v + w$
0	0	0	1			
0	0	1	1			
0	1	0	1			
0	1	1	1			
1	0	0	0			
1	0	1	0			
1	1	0	0			
1	1	1	0			

Node  $u$  đã được điền vào như là kết quả của phần bù của tín hiệu input  $A$

# Đánh giá OUTPUTs của mạch logic

- Bước 3: điền vào các giá trị tín hiệu của cột node  $v$



A	B	C	$u = \bar{A}$	$v = \bar{A}B$	$w = BC$	$x = v + w$
0	0	0	1	0		
0	0	1	1	0		
0	1	0	1	1		
0	1	1	1	1		
1	0	0	0	0		
1	0	1	0	0		
1	1	0	0	0		
1	1	1	0	0		

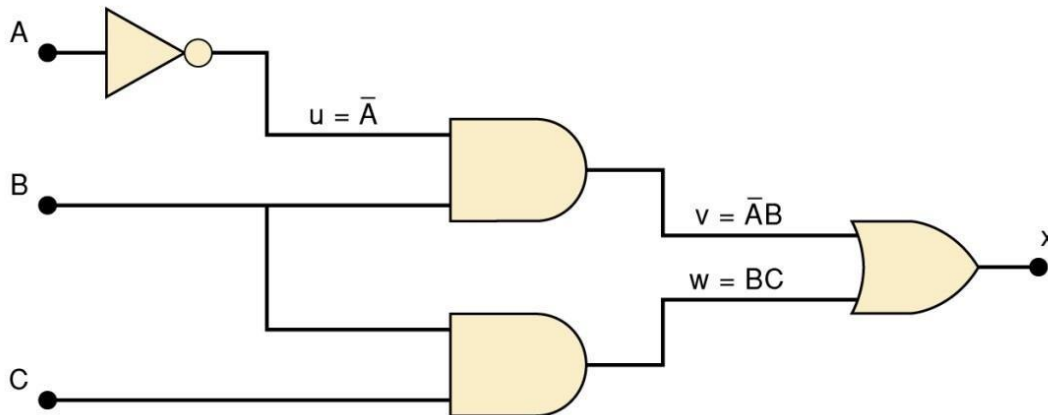
$v = \bar{A}B$  — Node  $v$  sẽ có giá trị HIGH

Khi  $\bar{A}$  (node  $u$ ) là HIGH và B là HIGH



# Đánh giá OUTPUTs của mạch logic

- Bước 4: Dự đoán trước giá trị tín hiệu của node w là outputs của cổng logic BC



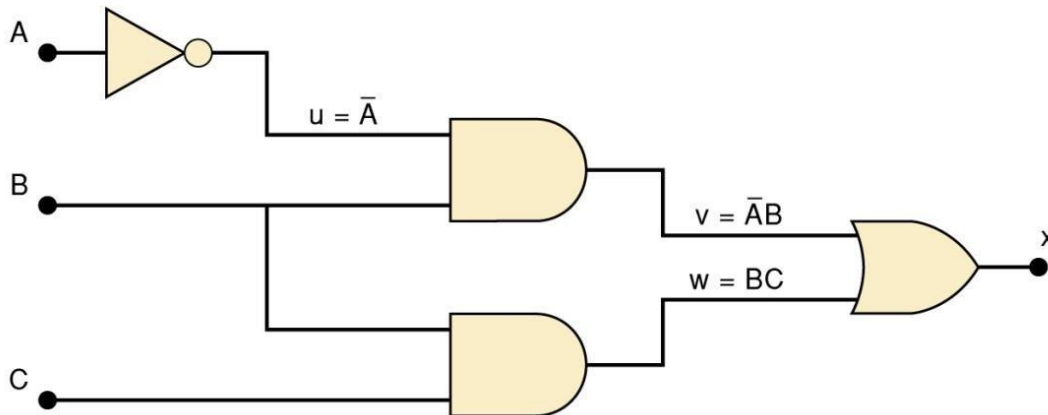
A	B	C	$u = \bar{A}$	$v = \bar{A}B$	$w = BC$	$x = v + w$
0	0	0	1	0	0	
0	0	1	1	0	0	
0	1	0	1	1	0	
0	1	1	1	1	1	
1	0	0	0	0	0	
1	0	1	0	0	0	
1	1	0	0	0	0	
1	1	1	0	0	1	

Cột này là HIGH khi và chỉ khi *B* là HIGH và cả *C* là HIGH



# Đánh giá OUTPUTs của mạch logic

- Bước cuối cùng: kết hợp một cách logic 2 cột  $\underline{v}$  và  $\underline{w}$  để dự đoán cho output  $\underline{x}$

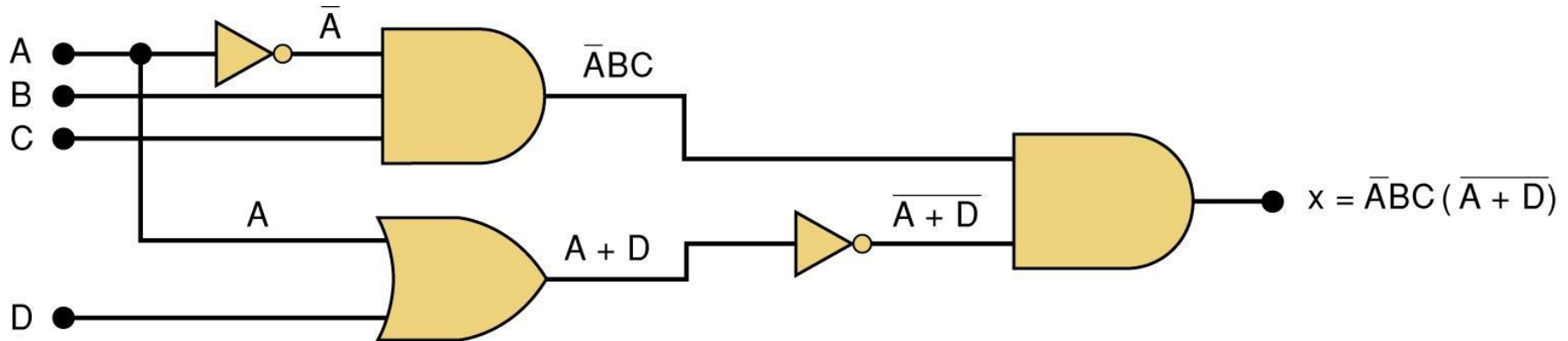


A	B	C	$\underline{u} = \bar{A}$	$\underline{v} = \bar{A}B$	$\underline{w} = BC$	$\underline{x} = v + w$
0	0	0	1	0	0	0
0	0	1	1	0	0	0
0	1	0	1	1	0	1
0	1	1	1	1	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	1	1

Từ biểu thức  $x = v + w$ , thì  $x$  output sẽ là HIGH khi  $v$  OR  $w$  là HIGH

# Đánh giá OUTPUTs của mạch logic

- Ví dụ:





**Biểu thức đại số  $\Rightarrow$  Mạch Logic**

# Thiết kế mạch logic từ biểu thức Boolean

- Biểu thức  $x = A.B.C$  có thể được thiết kế và vẽ nên bởi 3 inputs là 3 cổng logic AND .
- 1 mạch logic có biểu thức  $x = \overline{A} + B$  sẽ sử dụng 1 cổng logic OR gồm có 2 inputs, trong đó có 1 input sẽ có INVERTER kèm theo.

# Ví Dụ

- Vẽ sơ đồ mạch logic với output như sau:  $y = AC + B\bar{C} + \bar{A}BC$

# Thiết kế mạch logic từ biểu thức Boolean

- Mỗi cổng logic **OR** sẽ là một thành phần input của cổng logic chính **AND**



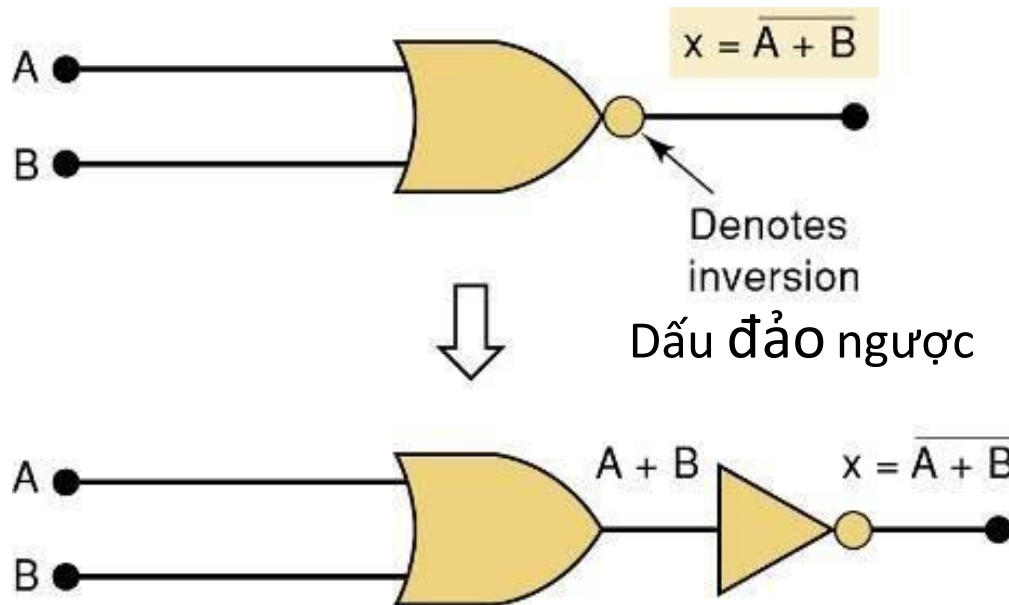
# Cổng Logic NOR và NAND



# Cổng Logic NOR

- NOR = NOT OR

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

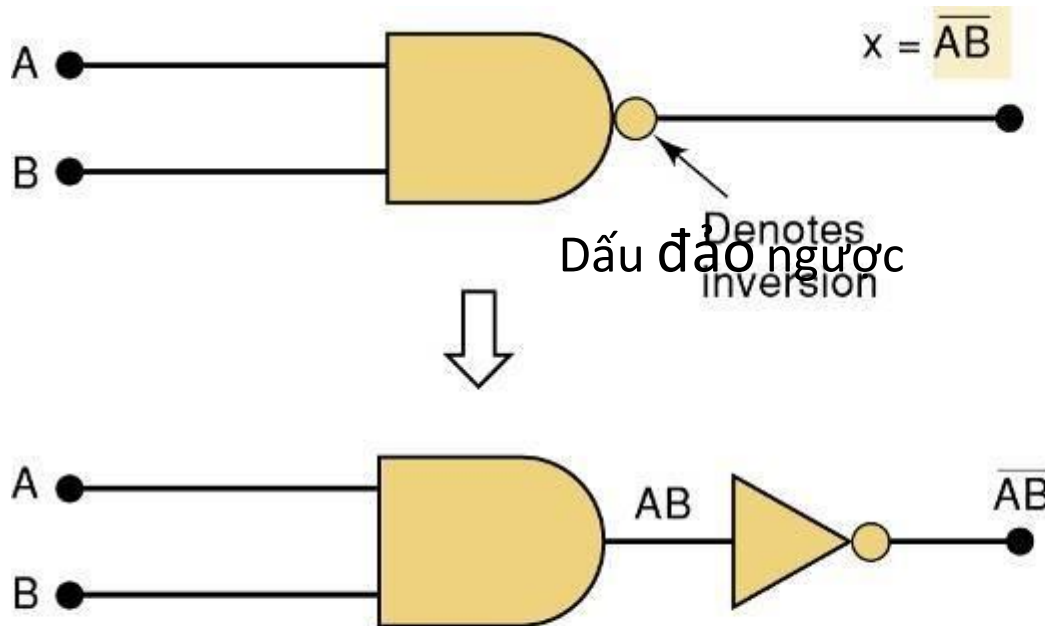


		OR		NOR	
A	B		$A + B$		$\overline{A + B}$
0	0		0		1
0	1		1		0
1	0		1		0
1	1		1		0

# Cổng Logic NAND

- NAND = NOT AND

$$- X = \overline{A \bullet B}$$



		AND		NAND	
A	B	AB		$\overline{AB}$	
0	0	0		1	
0	1	0		1	
1	0	0		1	
1	1	1		0	

# Ví Dụ NAND/NOR

- Thực hiện vẽ sơ đồ mạch logic  $X = AB \bullet \overline{(C + D)}$ 
  - Chỉ sử dụng cổng logic OR, AND, NOT
  - Chỉ sử dụng cổng logic NOR và NAND

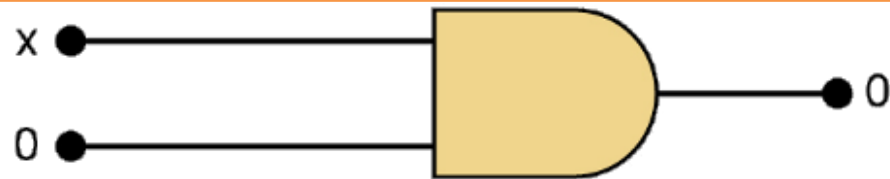


# Các Định Lý Đại Số Boolean

# Đại Số Boolean

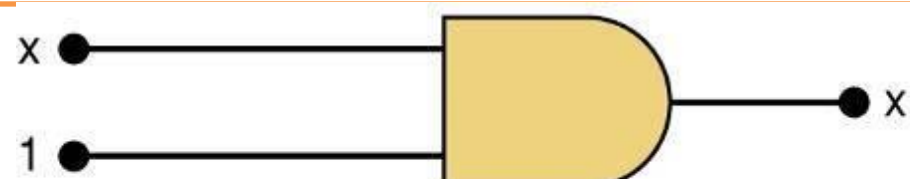
- Máy tính kỹ thuật số là tổng hợp các mạch logic được thực hiện dựa trên những hàm số của Boolean
- Khi chúng ta tạo nên một hàm số dựa trên Định Luật của Boolean, thì sẽ tạo nên mạch logic nhỏ hơn và đơn giản hơn
  - Giá thành rẻ hơn, tiêu tốn ít điện năng hơn, và đặc biệt và sẽ hoạt động xử lý nhanh hơn là mạch phức hợp.
- Do đó, dựa vào Định Luật của Boolean sẽ giúp chúng ta thực hiện xử lý những hàm số Boolean thành những dạng đơn giản nhất có thể

# Định Luật Boolean I



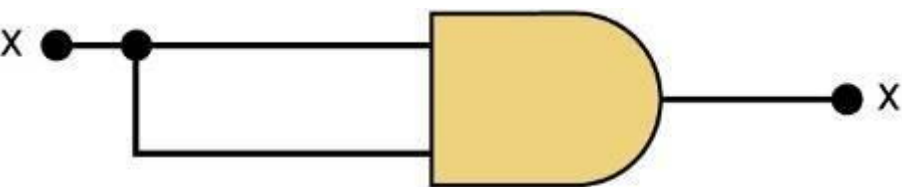
(1)  $x \cdot 0 = 0$

**Định Luật 1** nếu có bất kỳ input nào có giá trị tín hiệu là 0 trong cổng logic AND, thì kết quả của output sẽ là 0



(2)  $x \cdot 1 = x$

**Định Luật 2** khi một input có giá trị tín hiệu là 1 trong cổng logic AND thì không ảnh hưởng đến giá trị của tín hiệu output



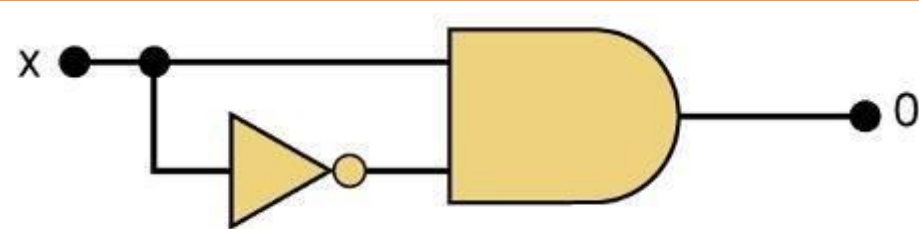
(3)  $x \cdot x = x$

**Định Luật 3** xét từng trường hợp

Nếu  $x = 0$ , thì  $0 \cdot 0 = 0$

Nếu  $x = 1$ , thì  $1 \cdot 1 = 1$

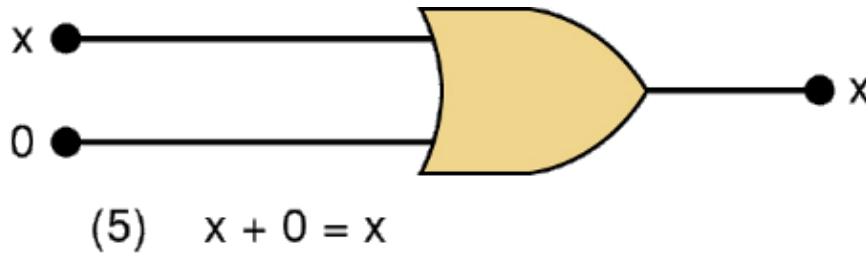
Do đó,  $x \cdot x = x$



(4)  $x \cdot \bar{x} = 0$

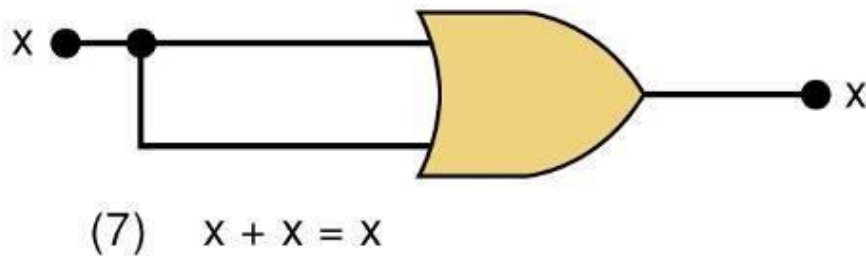
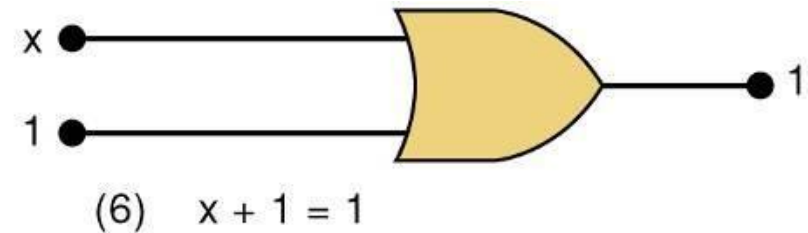
**Định Luật 4** có thể chứng minh bằng cách tương tự

# Định Luật Boolean II



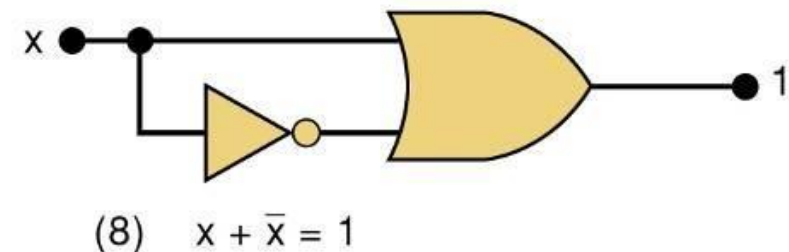
**Định Luật 5** nếu có 1 input có giá trị tín hiệu là 0 thì sẽ không gây ảnh hưởng đến giá trị tín hiệu của output

**Định Luật 6** nếu có 1 input là 1 thì output của cổng logic OR luôn là 1. Kiểm tra giá trị:  $0 + 1 = 1$  và  $1 + 1 = 1$



**Định Luật 7** có thể chứng minh bằng cách kiểm tra cả hai giá trị của x:  
 $0 + 0 = 0$  and  $1 + 1 = 1$

**Định Luật 8** có thể chứng minh một cách tương tự



# Định Luật Boolean III

## - PHÉP GIAO HOÁN

### - Commutative laws

$$(9) \quad x + y = y + x$$

$$(10) \quad x \cdot y = y \cdot x$$

## PHÉP LIÊN KẾT / KẾT HỢP

### - Associative laws

$$(11) \quad x + (y + z) = (x + y) + z = x + y + z$$

$$(12) \quad x \cdot (y \cdot z) = (x \cdot y) \cdot z = x \cdot y \cdot z$$

## PHÉP PHÂN PHỐI

### - Distributive laws

$$(13a) \quad x \cdot (y + z) = x \cdot y + x \cdot z$$

$$(13b) \quad (w + x) \cdot (y + z) = w \cdot y + x \cdot y + w \cdot z + x \cdot z$$

$$(13c) \quad x + yz = (x + y)(x + z)$$



# Định Luật Boolean IV

- Định Luật Đa Biến
- Định Luật (14) và (15) *không có thành phần đếm như trong phép tính số học thông thường.*

$$(14) \quad x + \overline{xy} = x$$

$$(15a) \quad \overline{x} + \overline{xy} = \overline{x} + y$$

$$(15b) \quad \overline{x} + xy = \overline{x} + y$$

$$\begin{aligned} x + xy &= x(1 + y) \\ &= x \cdot 1 && \text{[using theorem (6)]} \\ &= x && \text{[using theorem (2)]} \end{aligned}$$

x	y	xy	x + xy
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

# Định Luật Boolean V

## Tính đối ngẫu (Duality):

*Hai biểu thức được gọi là đối ngẫu của nhau khi ta thay phép toán AND bằng OR, phép toán OR bằng AND, 0 thành 1 và 1 thành 0*

### Ví Dụ

- $\bar{A} + B = B + \bar{A} \rightarrow A.B = B.A$
- $A + \bar{A}.B = A + B \rightarrow A.(\bar{A} + B) = A.B$
- $A + 0 = A \rightarrow A.1 = A$
- $A + 1 = 1 \rightarrow A.0 = 0$
- $A + \bar{A} = 1 \rightarrow A.\bar{A} = 0$

# Định Luật DeMorgan's

- Định Luật **DeMorgan's** là phương pháp cực kỳ hữu ích trong việc đơn giản hóa các biểu thức trong đó một tích hay tổng của các biến được đảo ngược

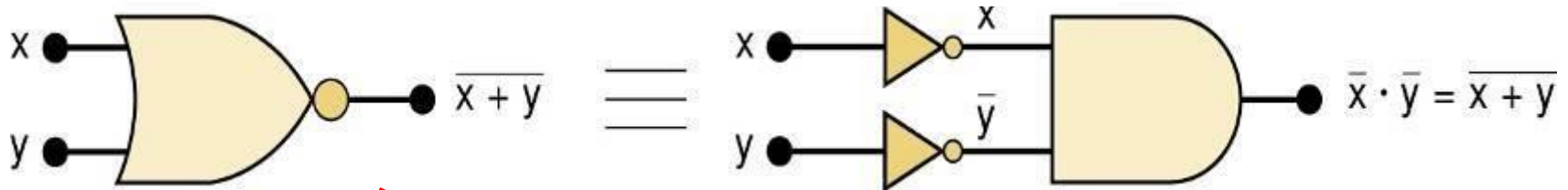
$$(16) \quad \overline{(x + y)} = \bar{x} \cdot \bar{y}$$

$$(17) \quad \overline{(x \cdot y)} = \bar{x} + \bar{y}$$

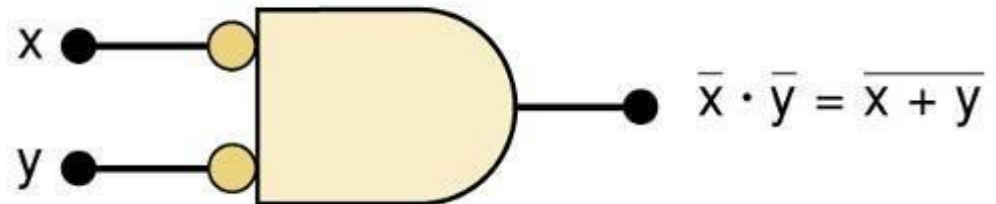
# Định Luật DeMorgan's

- Mạch tương đương với ngụ ý của Định Luật (16)

$$(16) \quad \overline{(x + y)} = \bar{x} \cdot \bar{y}$$



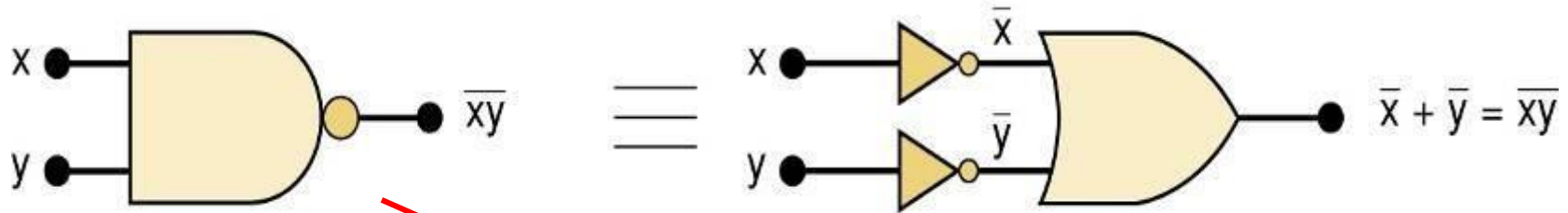
Mạch logic tương đương  
với hàm NOR



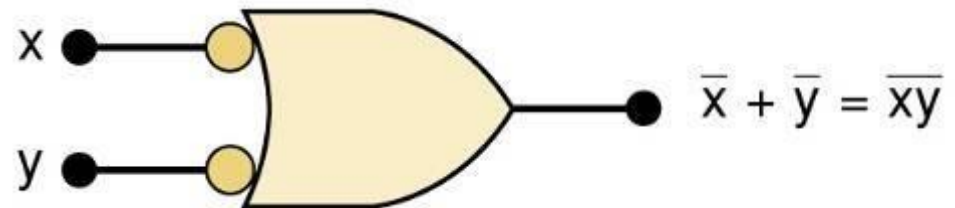
# Định Luật DeMorgan's

- Mạch tương đương với ngụ ý của Định Luật (17)

$$(17) \quad \overline{(x \cdot y)} = \bar{x} + \bar{y}$$



Mạch logic tương đương  
với hàm NAND



# Ví Dụ #1

- Áp dụng các định luật Boolean để đơn giản biểu thức sau đây:

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

$$\begin{aligned} & (X + Y)(X + \bar{Y})(\overline{XZ}) \\ & (X + Y)(X + \bar{Y})(\bar{X} + Z) \\ & (XX + X\bar{Y} + XY + Y\bar{Y})(\bar{X} + Z) \\ & ((X + Y\bar{Y}) + X(Y + \bar{Y}))(\bar{X} + Z) \\ & ((X + 0) + X(1))(\bar{X} + Z) \\ & X(\bar{X} + Z) \\ & X\bar{X} + XZ \\ & 0 + XZ \\ & XZ \end{aligned}$$

Idempotent Law (Rewriting)

DeMorgan's Law

Distributive Law

Commutative & Distributive Laws

Inverse Law

Idempotent Law

Distributive Law

Inverse Law

Idempotent Law

# Ví Dụ #2

- Áp dụng định luật DeMorgan để đơn giản các biểu thức sau

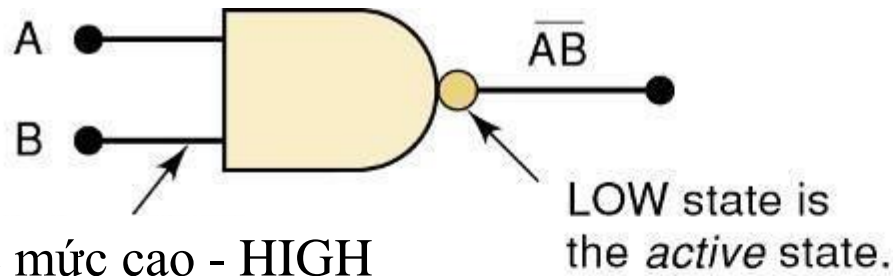
$$(i) \overline{(M + \bar{N})(\bar{M} + N)}$$

$$(ii) \overline{(\bar{A} + \bar{C} + \bar{D})}$$

- Đánh giá , nhận xét xem có bao nhiêu thiết bị linh kiện transistors có thể tiết kiệm được bằng phương pháp DeMorgan

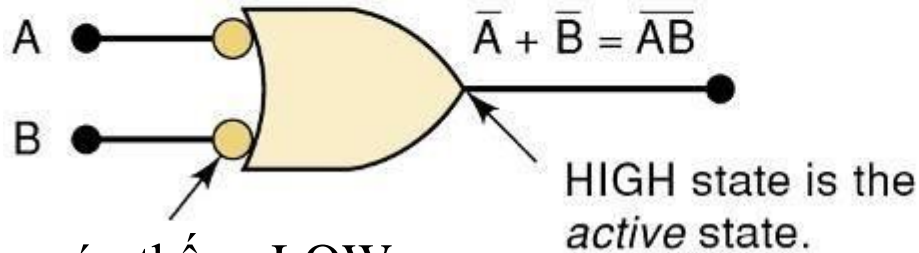
# Biểu diễn cổng logic (mở rộng)

- Ý nghĩa của 2 loại cổng logic NAND



(a)

Output là LOW khi tất cả inputs là HIGH



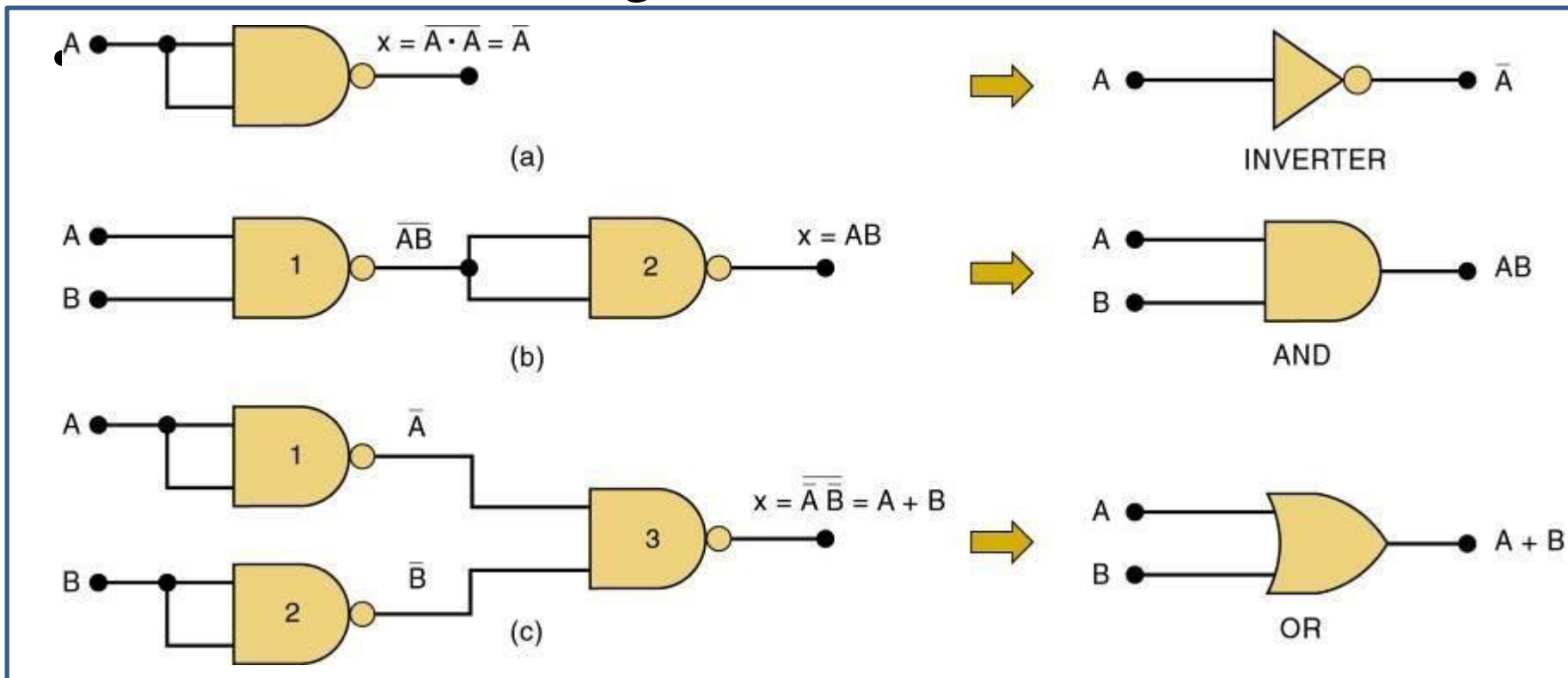
(b)

Output là HIGH khi có ít nhất 1 input có trạng thái là LOW



# Tính chất chung của cổng logic NAND

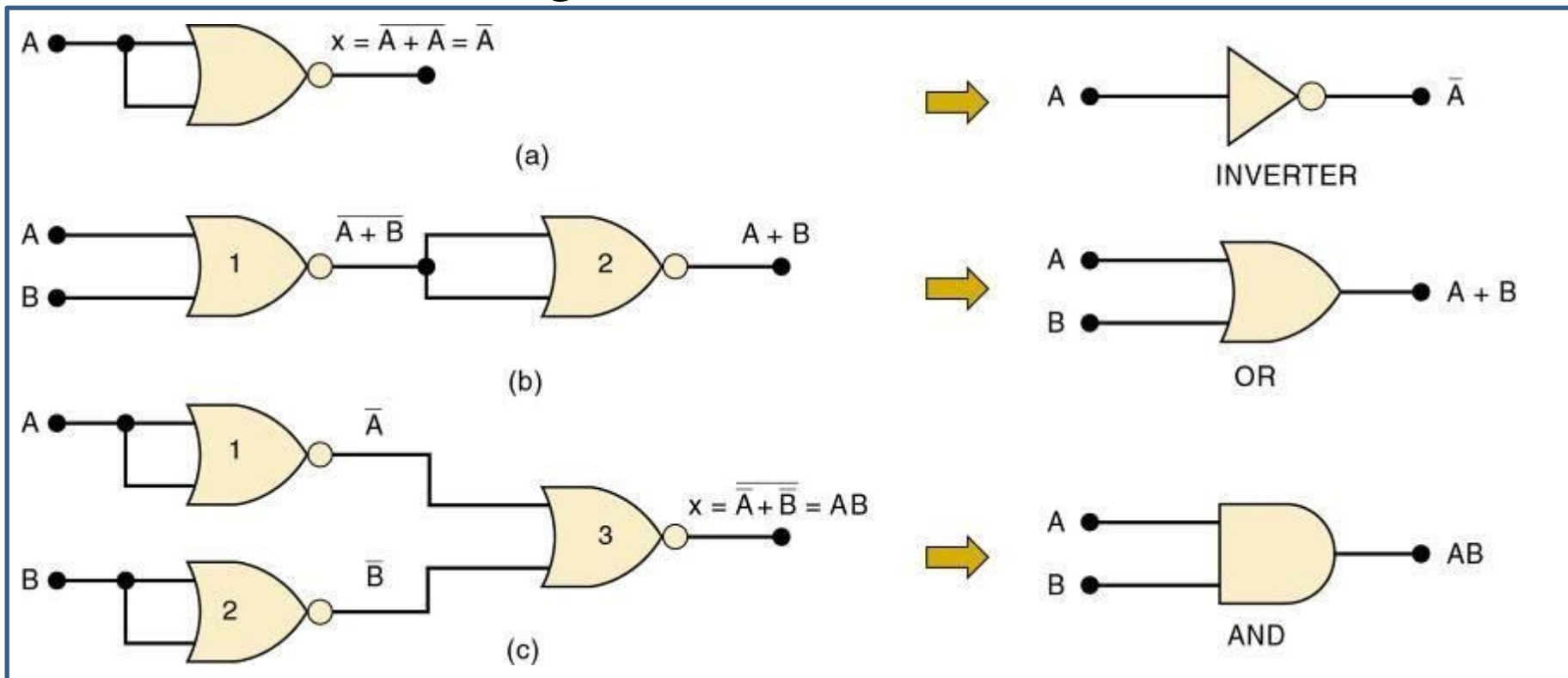
- Làm sao sử dụng một tổ hợp các cổng logic NANDs để tạo ra các hàm logic



Điều đó hoàn toàn có thể để thực hiện được bất cứ biểu thức logic nào mà chỉ sử dụng duy nhất 1 loại cổng logic NANDs

# Tính chất chung của cổng logic NOR

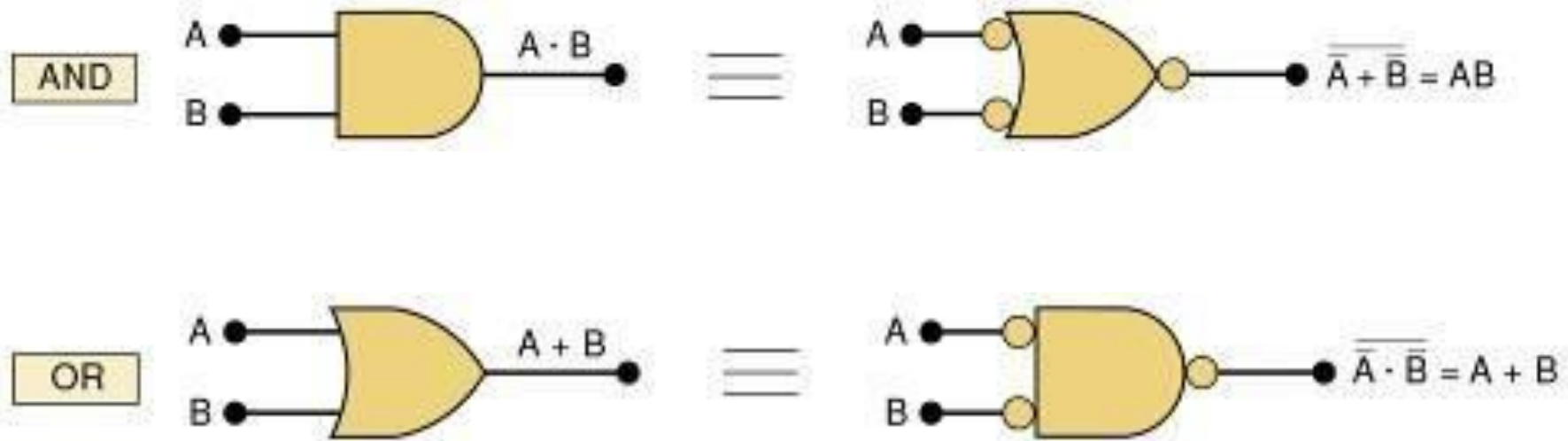
- Làm sao sử dụng một tổ hợp các cổng logic NORs để tạo ra các hàm logic



Điều đó hoàn toàn có thể để thực hiện được bất cứ biểu thức logic nào mà chỉ sử dụng duy nhất 1 loại cổng logic NORs

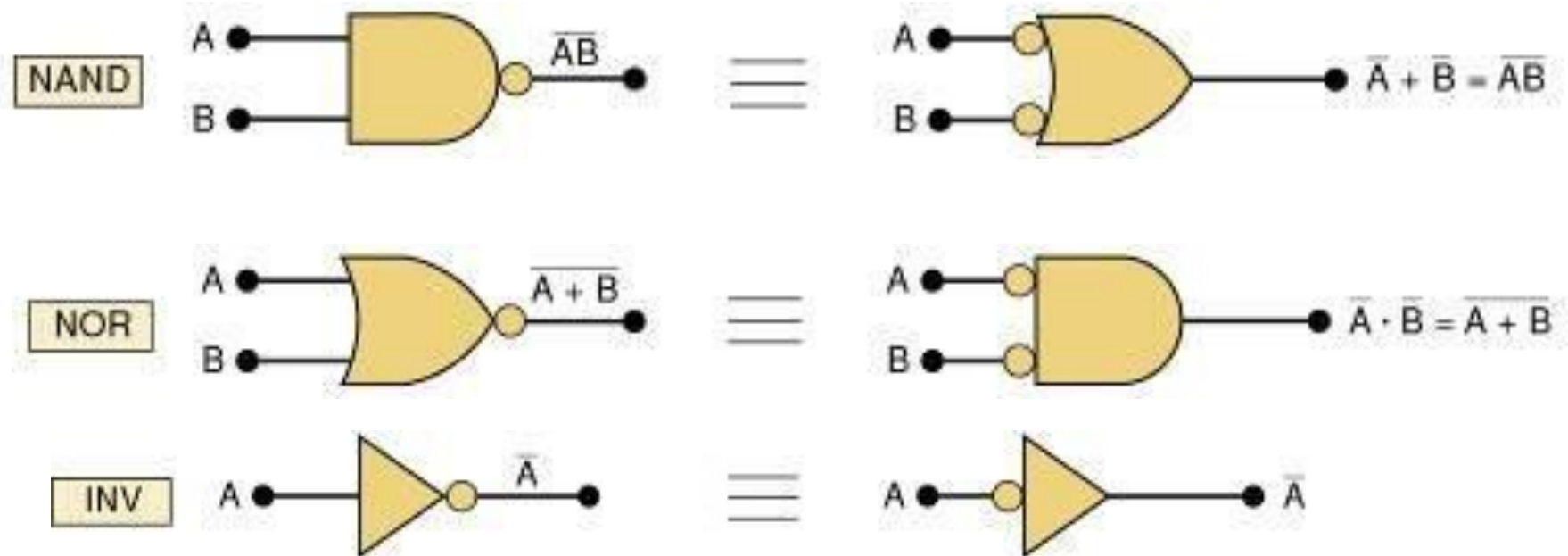
# Biểu diễn cổng logic (mở rộng)

- Để biến đổi một cổng logic cơ bản sang một cổng logic khác, có các cách như sau :
  - Nghịch đảo **OR** sang **AND** hoặc **AND** sang **OR**
  - Nghịch đảo mỗi input và output trong cùng một cổng logic



# Biểu diễn cổng logic (mở rộng)

- Thêm vào 1 bong bóng (bubble) nghịch đảo khi bạn đầu không có
- Loại bỏ bong bóng khi đã có tín hiệu output xuất hiện



# Danh sách chip IC thuộc họ 74LS...




## 74LS Series

### Features

- Standard 74LS Family in DIP Package
- Low Power and High Speed Schottky Family
- Large Quantity Discounts Available

### 74LS Series

Part No.	Description	More Info	In Stock	Package
74LS00	74LS00 Quad 2-input NAND Gate		Yes	PDIP14
74LS01	74LS01 Quad 2-input NAND Gate (OC)		Yes	PDIP14
74LS02	74LS02 Quad 2-input NOR Gate		Yes	PDIP14
74LS03	74LS03 Quad 2-input NAND Gate		Yes	PDIP14
74LS04	74LS04 Hex Inverter		Yes	PDIP14
74LS05	74LS05 Hex Inverter (Open Collector)		Yes	PDIP14
74LS06	74LS06 Hex Inverter Buffer/Driver (Open Coll.)		Yes	PDIP14
74LS07	74LS07 Hex Buffer/Driver (Open Collector)		Yes	PDIP14
74LS08	74LS08 Quad 2-input AND Gate		Yes	PDIP14
74LS09	74LS09 Quad 2-input AND Gate (Open Collector)		Yes	PDIP14
74LS10	74LS10 Triple 3-input NAND Gate		Yes	PDIP14
74LS11	74LS11 Triple 3-input AND Gate		Yes	PDIP14

Source: <http://www.futurlec.com/IC74LS00Series.shtml>

# Thuật ngữ kỹ thuật số

***Inverter*** Mạch logic thực hiện việc đảo ngược hoặc sẽ tạo ra phân bù đối với giá trị của inputs

***Truth table*** Bảng sự thật/ chân trị biểu diễn/thể hiện giá trị trạng thái của tín hiệu inputs cũng như outputs tương ứng

***Timing diagram*** Một sơ đồ dạng sóng cho thấy mối quan hệ thời điểm thích hợp của tất cả các dạng sóng

***Boolean algebra***

***AND gate*** Các phương pháp toán học đại số Boolean dành cho mạch logic

Với cổng AND có nhiều ngõ vào,  
ngõ ra sẽ là 1 nếu tất cả các ngõ vào đều là 1

# Thuật ngữ kỹ thuật số

***OR gate*** Với cổng OR có nhiều ngõ vào,  
ngõ ra sẽ là 0 nếu tất cả các ngõ vào đều là 0

***NAND gate*** Với cổng NAND có nhiều ngõ vào,  
ngõ ra sẽ là 0 nếu tất cả các ngõ vào đều là 1

***NOR gate*** Với cổng NOR có nhiều ngõ vào,  
ngõ ra sẽ là 1 nếu tất cả các ngõ vào đều là 0

***Exclusive-OR gate*** Với cổng XOR có nhiều ngõ vào, ngõ ra sẽ là 1  
nếu tổng số bit 1 ở các ngõ vào là số lẻ

***Exclusive-NOR gate*** Với cổng XNOR có nhiều ngõ vào, ngõ ra sẽ là 1  
nếu tổng số bit 1 ở các ngõ vào là số chẵn