

NHẬP MÔN MẠCH SỐ

CHƯƠNG 3



Đại Số Boolean và Các Cổng Logic

Tổng quan



Chương này sẽ học về:

- Đại số Boolean: với đặc điểm là chỉ thực hiện trên hai giá trị/trạng thái 0(OFF) và 1(ON) nên rất phù hợp với việc biểu diễn và tính toán trong các mạch logic Số
- Các cổng logic cơ bản, từ đó có thể xây dựng nên các mạch logic hoặc các hệ thống số phức tạp trong những chương sau.

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**
- **Đại số Boolean**

Tổng Quát

- 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**



Cổng Logic Cơ Bản

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

- Các giá trị ngõ ra tương ứng với số ngõ vào
 - Một bảng có 2 ngõ vào sẽ có

?

 giá trị ngõ ra tương ứng
 - Một bảng có 3 ngõ vào sẽ có

?

 giá trị ngõ ra tương ứng

Cổng Logic OR

- Biểu thức Boolean cho cổng logic **OR**:

– $X = A + B$ — Đọc là “X bằng A OR B”

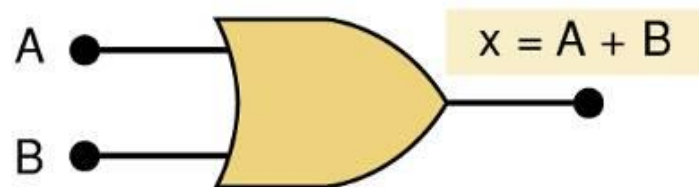
Dấu + không có nghĩa là phép cộng thông thường, mà là ký hiệu cho cổng logic OR

- Bảng sự thật và ký hiệu mạch của cổng 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”

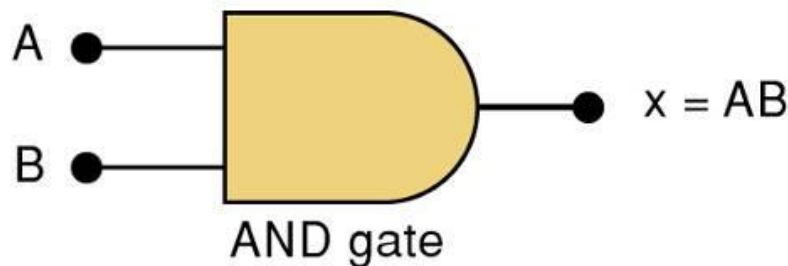
Dấu \bullet không có nghĩa là phép nhân thông thường ,
mà là ký hiệu cho cổng logic AND

- Bảng sự thật và ký hiệu mạch cổng AND có 2 inputs:

AND

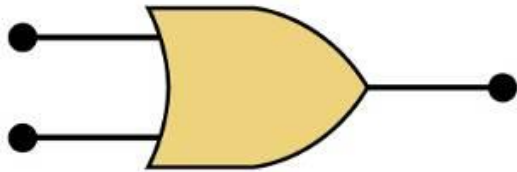
A	B	$x = A \bullet B$
0	0	0
0	1	0
1	0	0
1	1	1

(a)



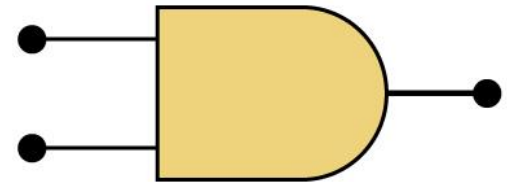
(b)

OR vs AND



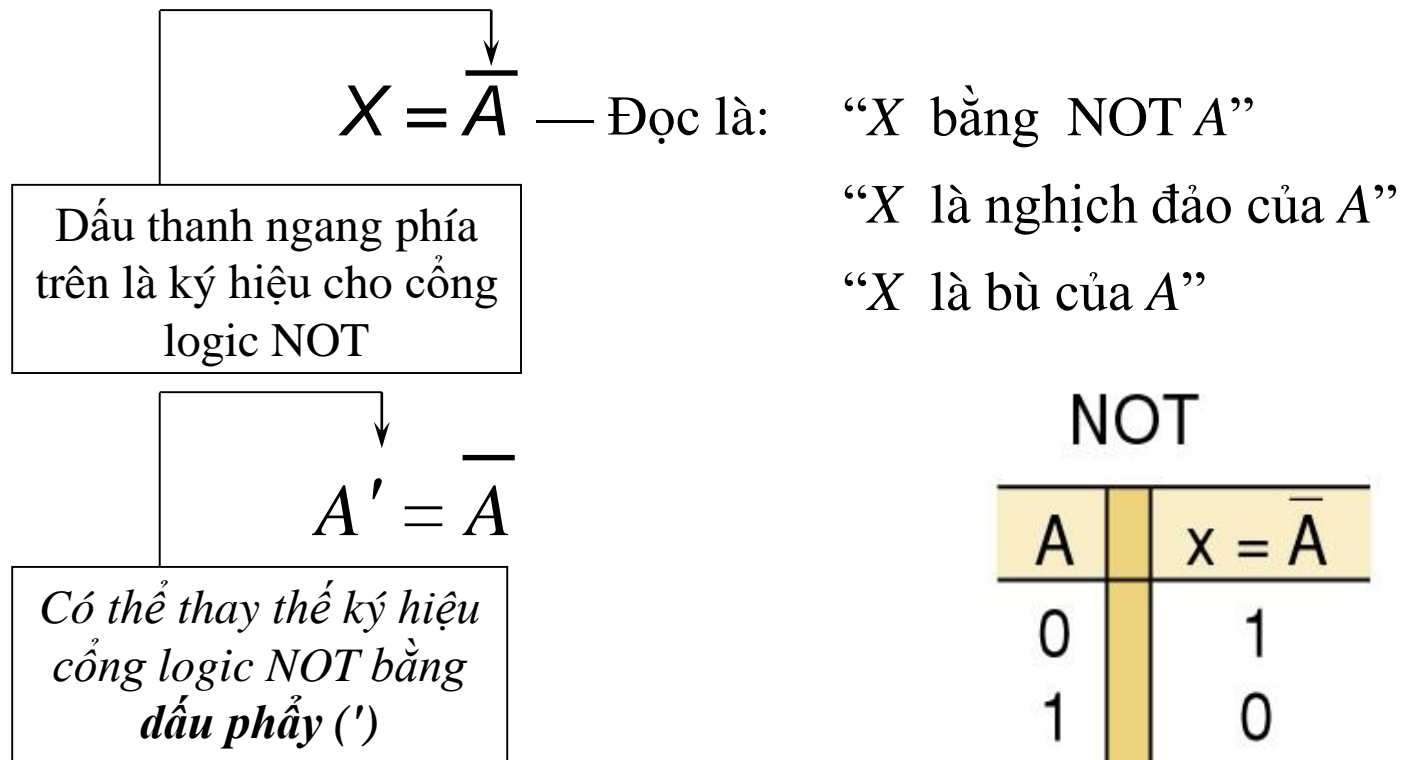
Ký hiệu của cổng logic OR có nghĩa là output sẽ có trạng thái là HIGH khi có bất kỳ input nào có trạng thái là HIGH

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**

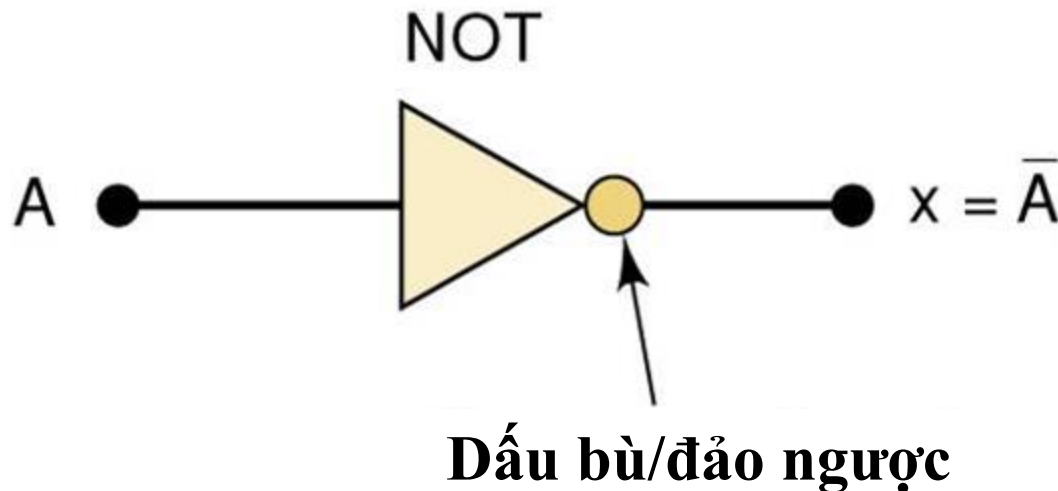


NOT	
A	$x = \overline{A}$
0	1
1	0

Bảng sự thật cổng
Logic NOT

Cổng Logic NOT

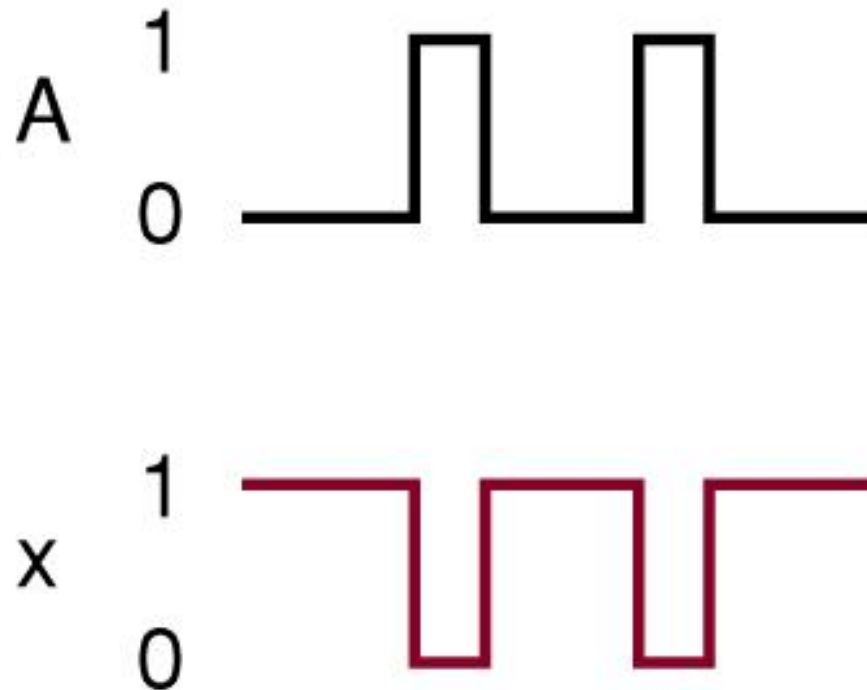
- Cổng logic NOT có thể gọi chung là cổng *INVERTER*



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 (lấy bù) tín hiệu ngõ vào tại tất cả các thời điểm để tạo ra tín hiệu ngõ ra tương ứng



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

$$\overline{0} = 1$$

$$\overline{1} = 0$$

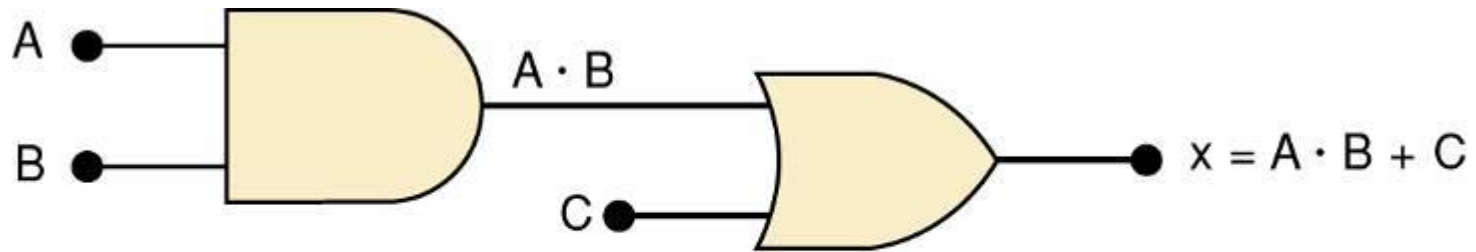
Ba cổng logic Boolean cơ bản có thể mô tả được bất kỳ mạch logic nào



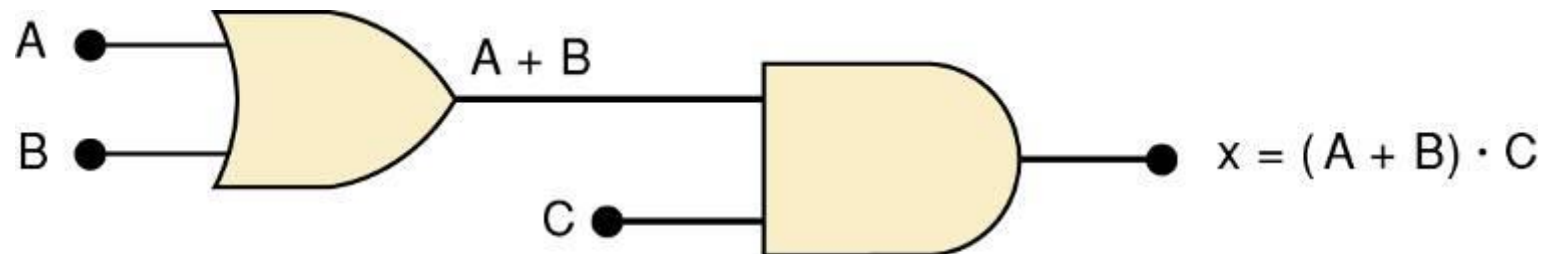
Mạch Logic => Biểu thức đại số

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 :

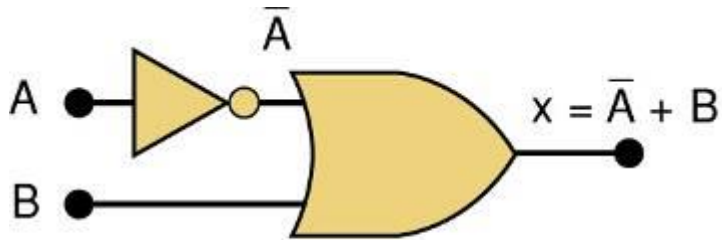


- Trừ khi có một dấu ngoặc trong biểu thức

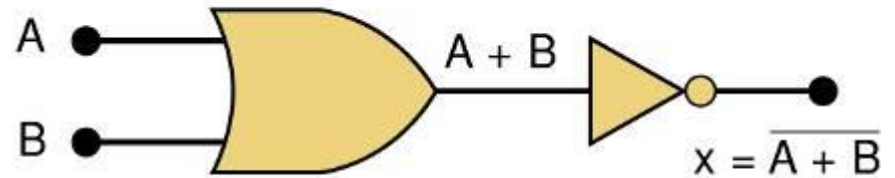


Mô tả mạch logic đại số

Input A qua một inverter sẽ có output là \bar{A}

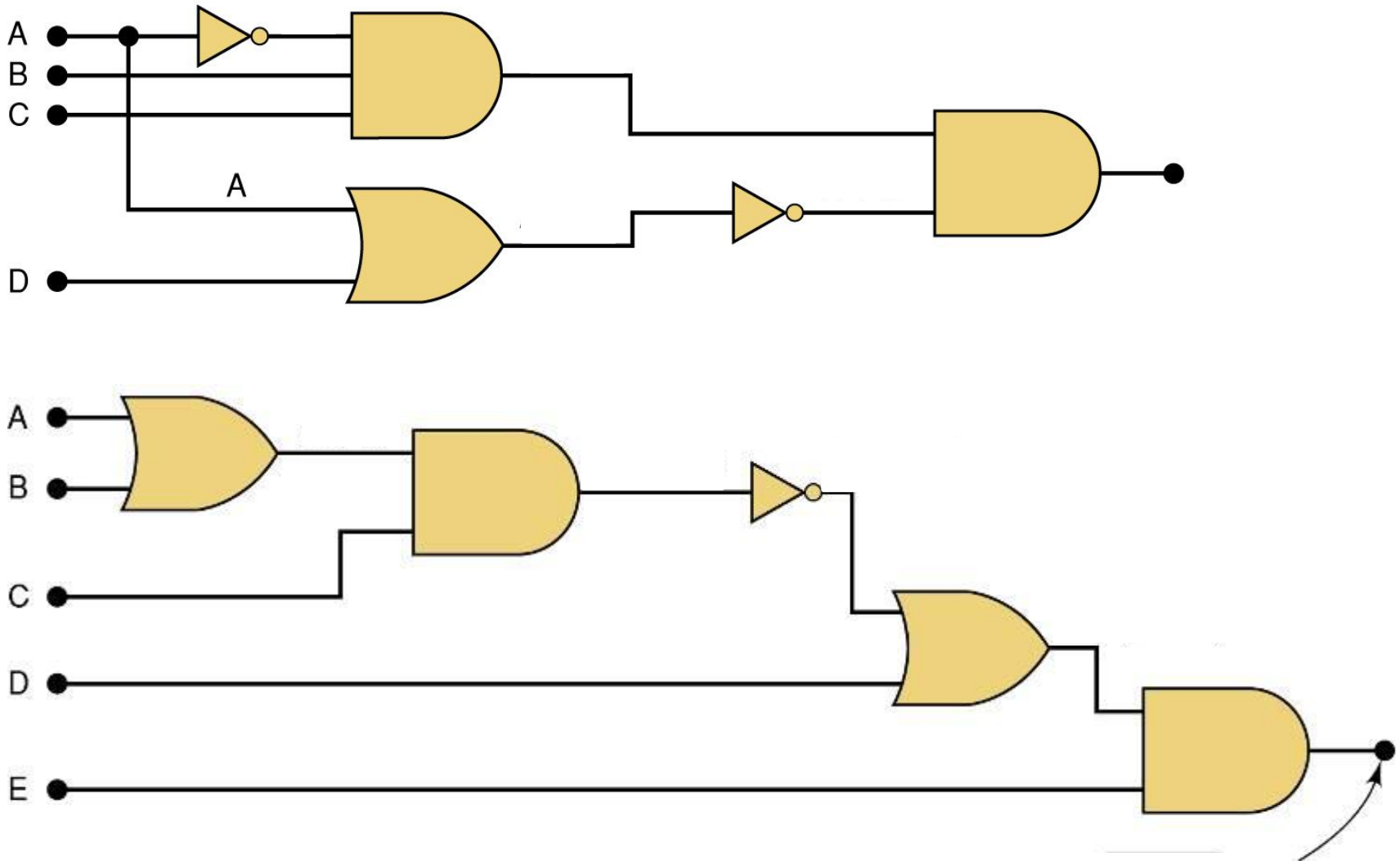


(a)



(b)

Ví Dụ



Đánh giá OUTPUT của mạch logic

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

Quy tắc đánh giá một biểu thức Boolean theo trình tự sau:

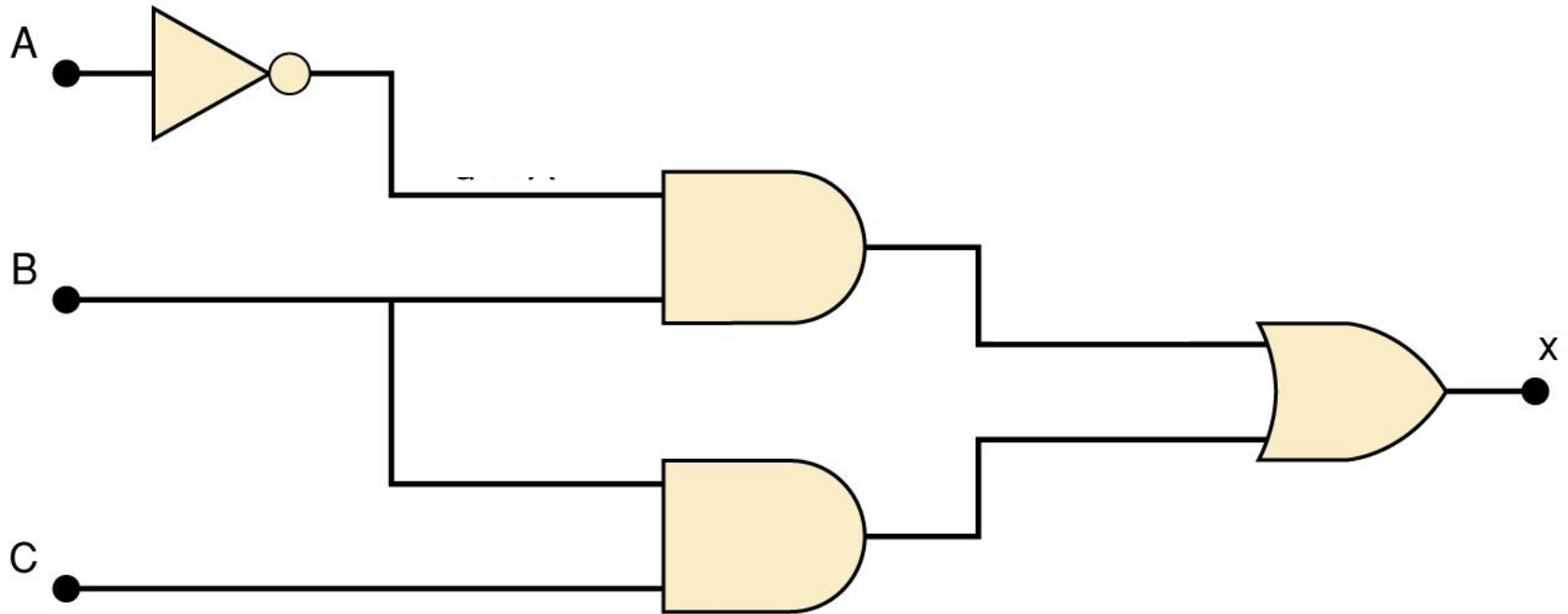
- Tính giá trị ngõ ra của các *cổng đảo có một thành phần*
- Tính giá trị biểu thức trong dấu ngoặc đơn
- Tính giá trị biểu thức cổng AND trước biểu thức cổng OR (nếu biểu thức cổng OR không có dấu ngoặc đơn)
- Nếu cả một biểu thức có thanh ngang trên đầu, thực hiện các phép tính bên trong biểu thức trước, và sau đó đảo ngược kết quả lại

Đánh giá OUTPUT 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 một cách chính xác nhất
 - Bảng sự thật giúp ích trong việc phát hiện và xử lý lỗi hay sự cố xuất hiện trong mạch logic

Đánh giá OUTPUT của mạch logic

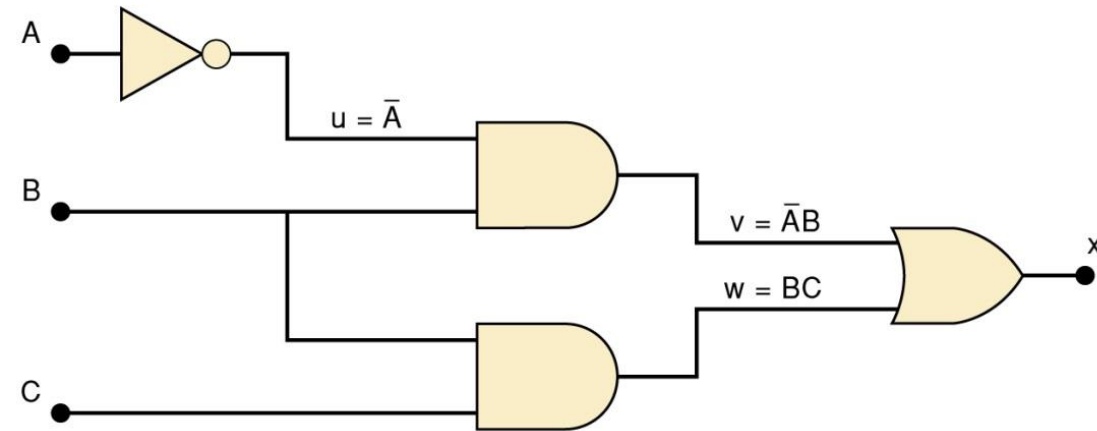
- Đánh giá outputs của mạch logic sau:



Đánh giá OUTPUT 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)

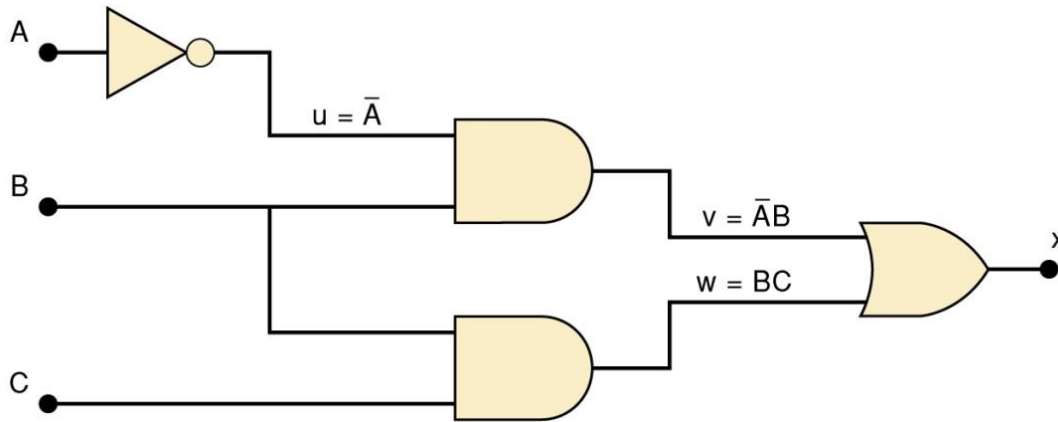


A	B	C	$u = \bar{A}$	$v = \bar{A}B$	$w = BC$	$x = v + w$
0	0	0				
0	0	1				
0	1	0				
0	1	1				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

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

Đánh giá OUTPUT của mạch logic

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



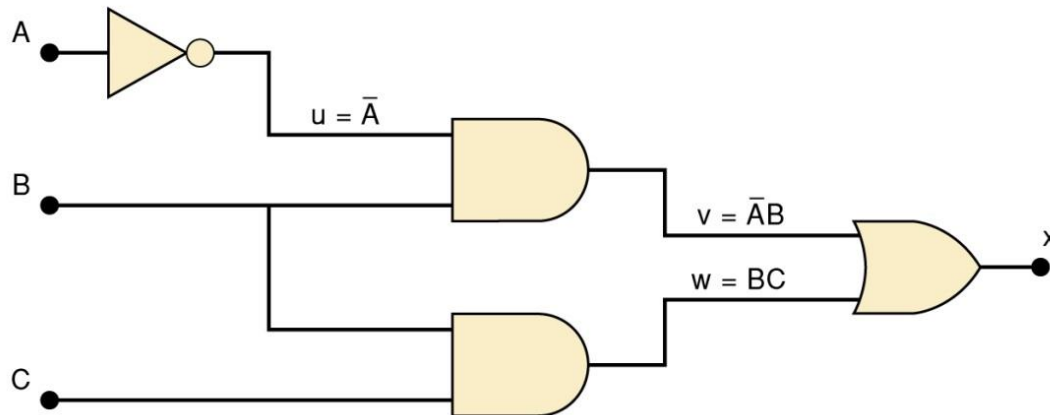
A	B	C	$u = \bar{A}$	$v = \bar{A}B$	$w = BC$	$x = v + w$
0	0	0	1			
0	0	1	1			
0	1	0	1			
0	1	1	1			
1	0	0	0			
1	0	1	0			
1	1	0	0			
1	1	1	0			

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

Khi A (node u) là HIGH và B là HIGH

Đánh giá OUTPUT của mạch logic

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

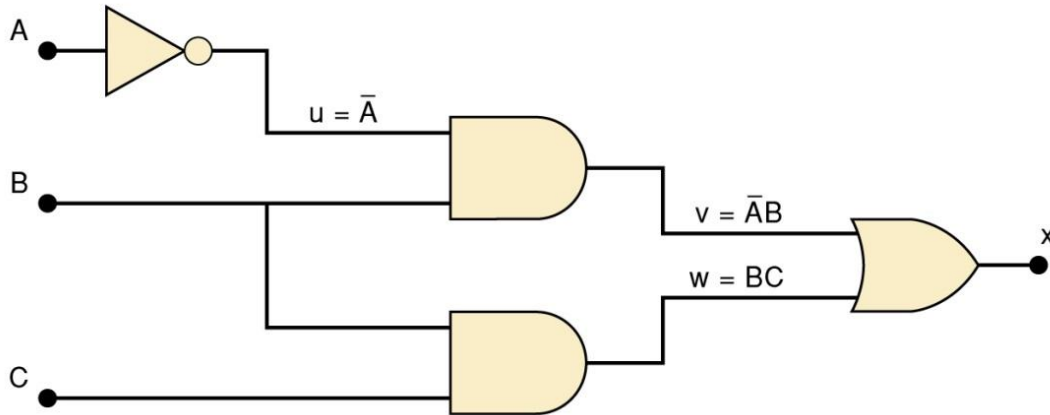


A	B	C	$u = \bar{A}$	$v = \bar{A}B$	$w = BC$	$x = v + w$
0	0	0	1	0		
0	0	1	1	0		
0	1	0	1	1		
0	1	1	1	1		
1	0	0	0	0		
1	0	1	0	0		
1	1	0	0	0		
1	1	1	0	0		

Node w là HIGH khi và chỉ khi B là HIGH và cả C là HIGH

Đánh giá OUTPUT của mạch logic

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

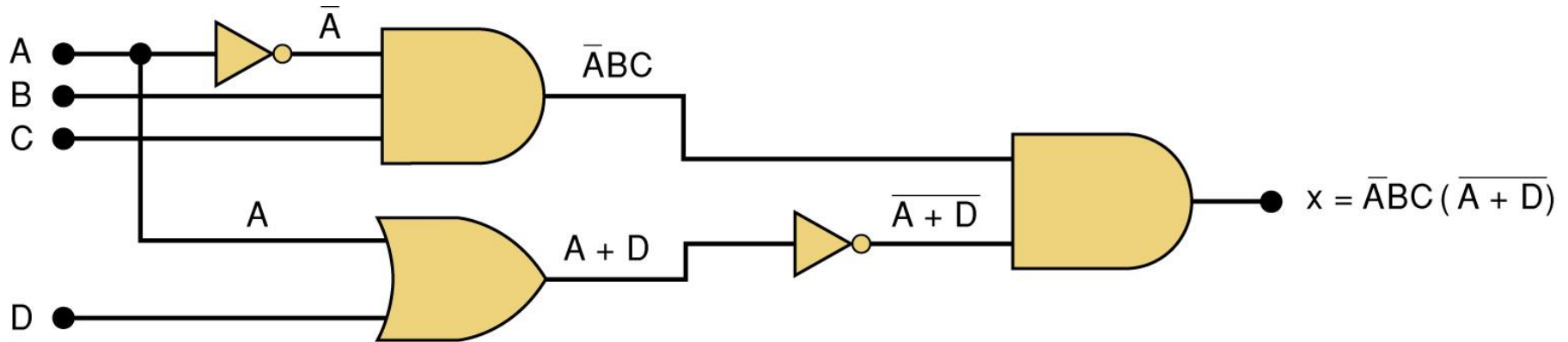


A	B	C	$u = \bar{A}$	$v = \bar{A}B$	$w = BC$	$x = v + w$
0	0	0	1	0	0	
0	0	1	1	0	0	
0	1	0	1	1	0	
0	1	1	1	1	1	
1	0	0	0	0	0	
1	0	1	0	0	0	
1	1	0	0	0	0	
1	1	1	0	0	1	

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

Đánh giá OUTPUT 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 thực hiện bởi một cổng AND có 3 ngõ vào.
- Một 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 cổng INVERTER kèm theo.

Ví Dụ

Vẽ sơ đồ mạch logic với output
như sau:

$$y = AC + B\bar{C} + \bar{A}BC$$

Ví Dụ (tt)

- 
- Mỗi ngõ vào của cổng **OR** là ngõ ra của các cổng **AND** trước đó

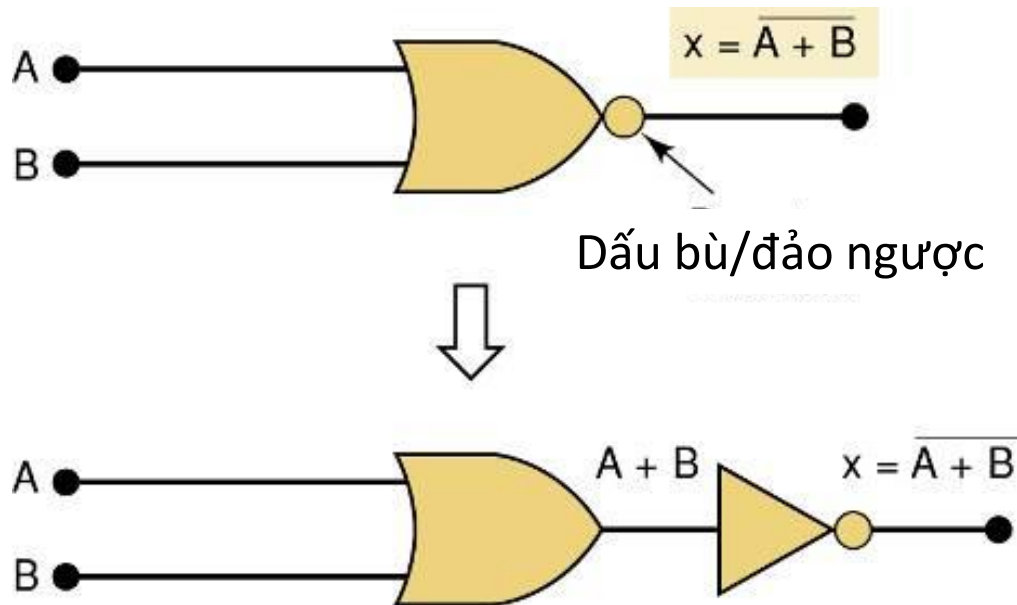


Cổng Logic NOR và NAND

Cổng Logic NOR

- NOR = NOT OR

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

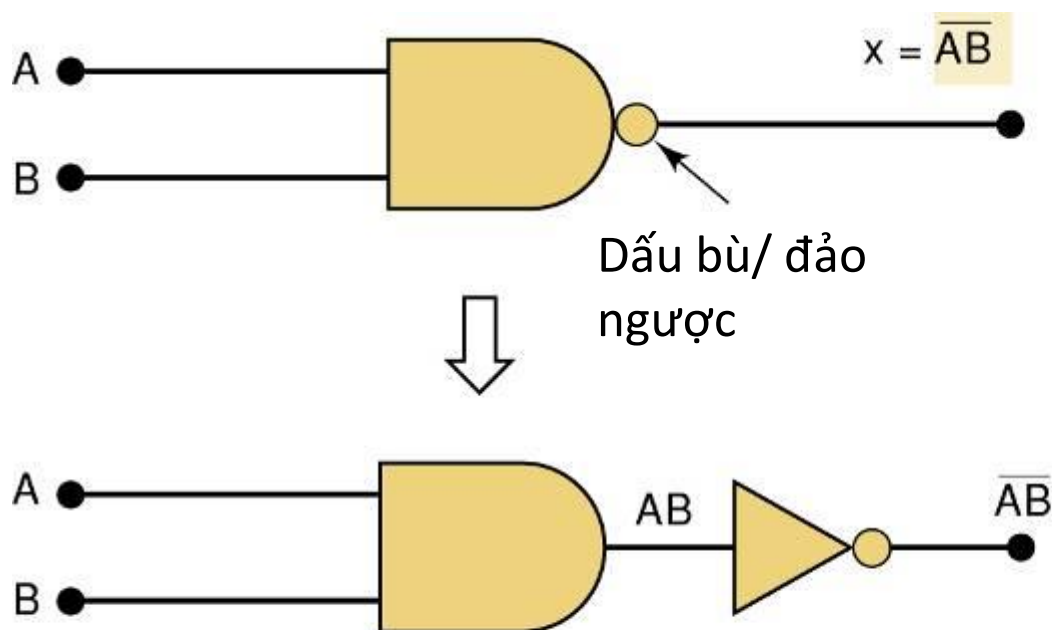


		OR		NOR	
A	B	$A + B$		$\overline{A + B}$	
0	0	0		1	
0	1	1		0	
1	0	1		0	
1	1	1		0	

Cổng Logic NAND

- NAND = NOT AND

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



		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

Vẽ sơ đồ mạch thực hiện
biểu thức logic:

$$X = \overline{AB \bullet (\overline{C + D})}$$

- Chỉ sử dụng cổng logic OR, AND, NOT
- Chỉ sử dụng cổng logic NOR và NAND

Lưu ý: Nếu đề không yêu cầu cổng logic sử dụng có bao nhiêu ngõ vào, thì người thiết kế có thể chọn cổng logic có bao nhiêu ngõ vào cũng được.

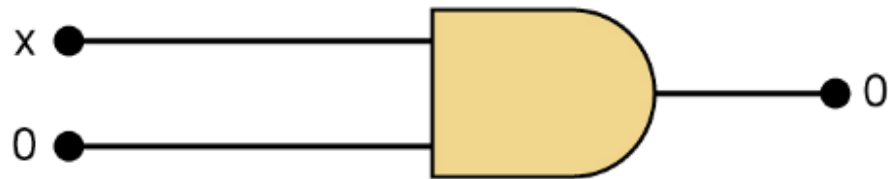


Đại Số Boolean

Đại Số Boolean

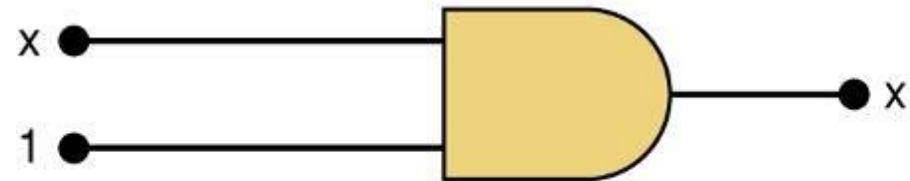
- Máy tính kỹ thuật số là tổng hợp các mạch logic được thực hiện dựa trên những biểu thức của đại số Boolean (biểu thức Boolean)
- Biểu thức Boolean càng đơn giản, thì mạch thực hiện càng nhỏ
→ giá thành rẻ hơn, tiêu tốn ít công suất hơn, và thực hiện nhanh hơn mạch phức tạp
- Dựa vào các **định luật Boolean** sẽ giúp ta đơn giản được các biểu thức Boolean về dạng đơn giản nhất

Định Luật Boolean I



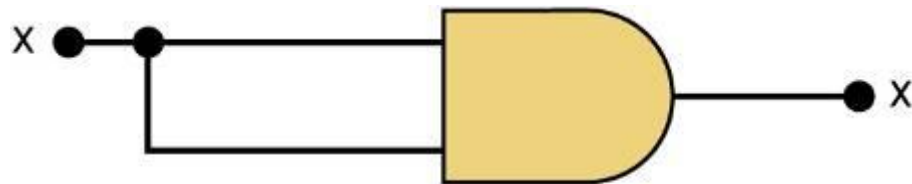
(1) $x \cdot 0 = 0$

Định Luật 1 nếu một cổng AND-2 có 1 ngõ vào bằng 0, thì ngõ ra sẽ bằng 0 bất kể giá trị ngõ vào còn lại.



(2) $x \cdot 1 = x$

Định Luật 2 nếu một cổng AND-2 có 1 ngõ vào bằng 1, thì ngõ ra sẽ bằng giá trị với ngõ vào còn lại.



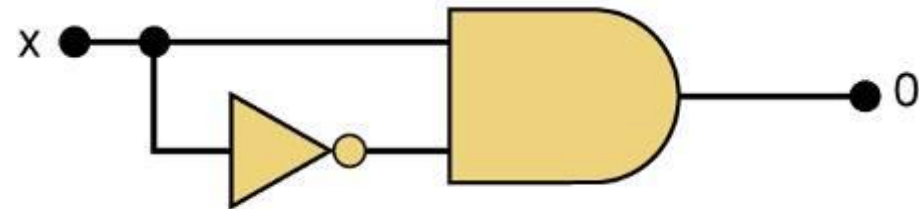
(3) $x \cdot x = x$

Định Luật 3 xét từng trường hợp

Nếu $x = 0$, thì $0 \cdot 0 = 0$

Nếu $x = 1$, thì $1 \cdot 1 = 1$

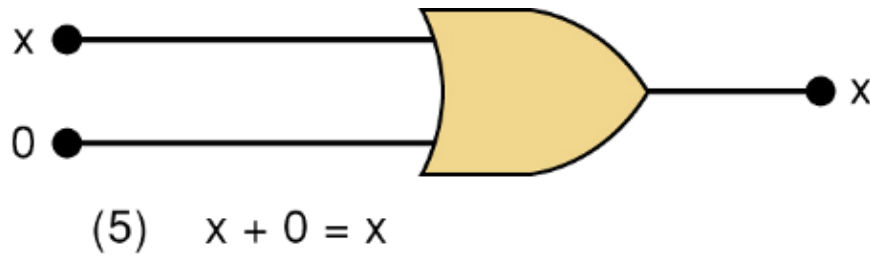
Do đó, $x \cdot x = x$



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

Định Luật 4 có thể chứng minh bằng cách tương tự

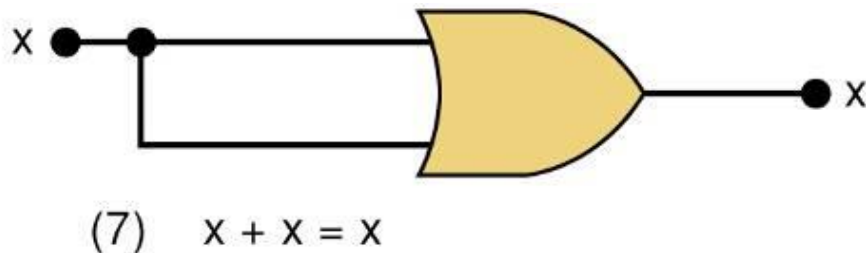
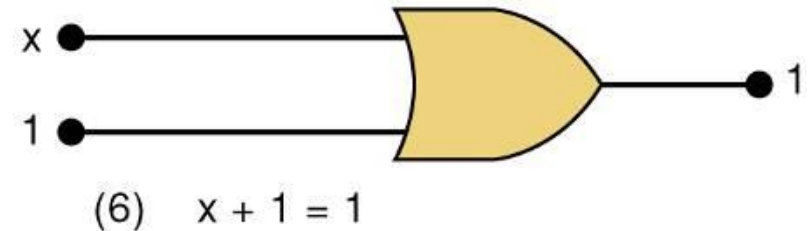
Định Luật Boolean II



Định Luật 5 nếu một cổng OR-2 có 1 ngõ vào bằng 0, thì ngõ ra sẽ bằng giá trị với ngõ vào còn lại

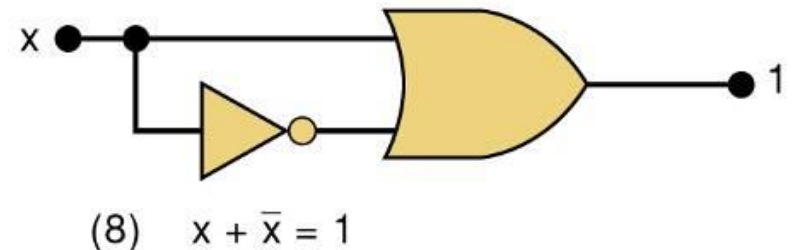
Định Luật 6

nếu một cổng OR-2 có 1 ngõ vào bằng 1, thì ngõ ra sẽ bằng 1 bất kể giá trị ngõ vào còn lại



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**

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

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

- **PHÉP LIÊN KẾT / KẾT HỢP**

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

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

- **PHÉP PHÂN PHỐI**

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

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

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

Định Luật Boolean IV

- Định Luật Đa Biến
- Định Luật (14) và (15) *không gặp trong đại số thông thường.*

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

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

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

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

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

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

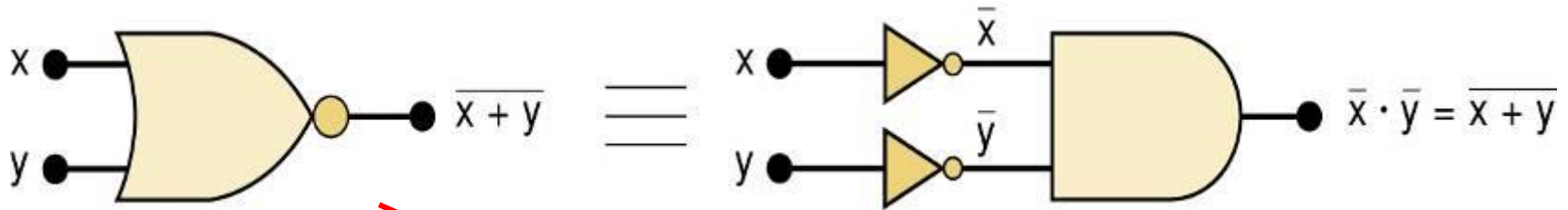
$$(16) \quad \overline{(x + y)} = \bar{x} \cdot \bar{y}$$

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

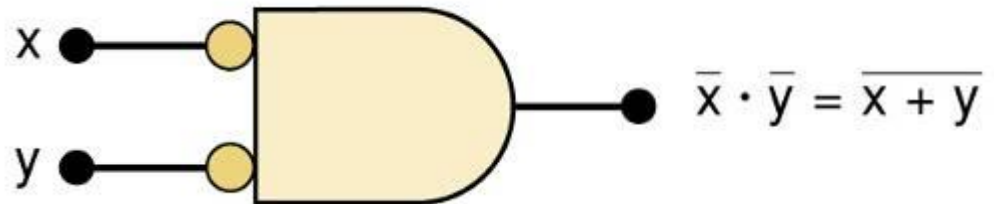
Định Luật DeMorgan's

- Mạch tương đương với negation của Định Luật (16)

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



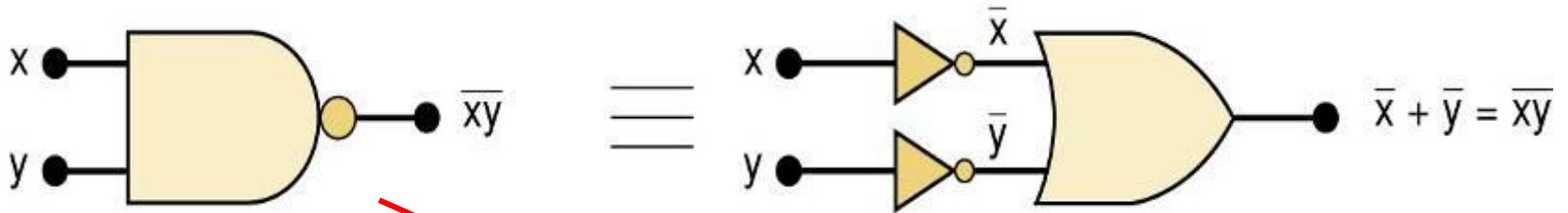
Mạch logic khác tương đương với hàm **NOR**



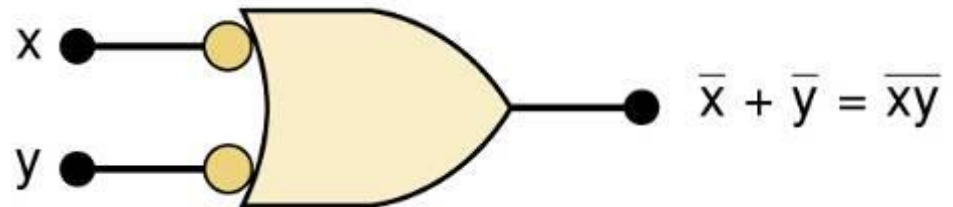
Định Luật DeMorgan's

- Mạch tương đương với negation của Định Luật (17)

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



Mạch logic khác tương đương với hàm NAND



Ví Dụ #1

- Áp dụng các định luật Boolean để đơn giản biểu thức sau đây:
$$F(X, Y, Z) = (X + Y)(X + \bar{Y})(\overline{XZ})$$

Ví Dụ #2

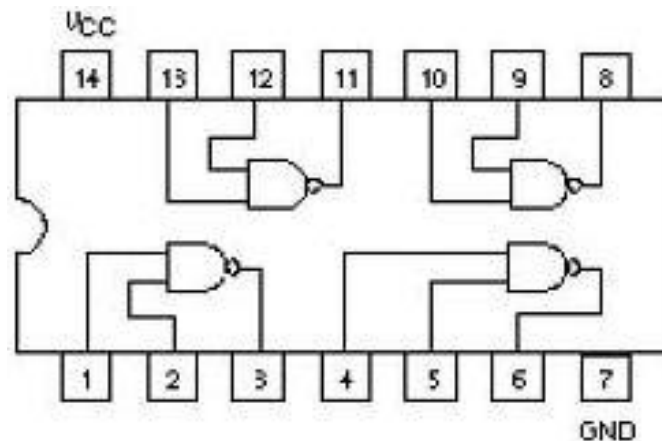
- Áp dụng định luật DeMorgan để đơn giản các biểu thức sau

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

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

Ví dụ #3

Chỉ sử dụng cổng NAND để thực hiện các cổng logic cơ bản (NOT, OR, AND)



74LS00 chip

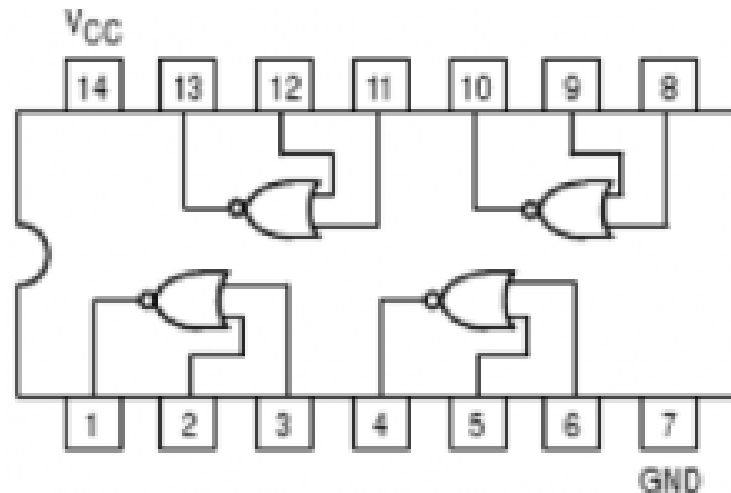
Tính phổ biến của cổng NAND

- Chỉ sử dụng cổng NAND để thực hiện 3 cổng logic cơ bản (AND, OR, NOT)

Có thể thực hiện được bất cứ biểu thức logic nào chỉ sử dụng duy nhất 1 loại cổng logic NAND

Example #4

Chỉ sử dụng cổng NOR để thực hiện các cổng logic cơ bản (NOT, OR, AND)



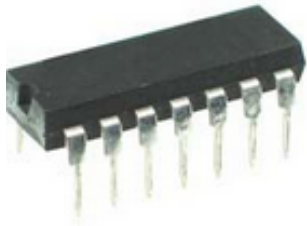
74LS02 chip

Tính phổ biến của cổng NOR

- Chỉ sử dụng cổng NOR để thực hiện 3 cổng logic cơ bản (AND, OR, NOT)

Có thể thực hiện được bất cứ biểu thức logic nào chỉ sử dụng duy nhất 1 loại cổng logic NOR

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

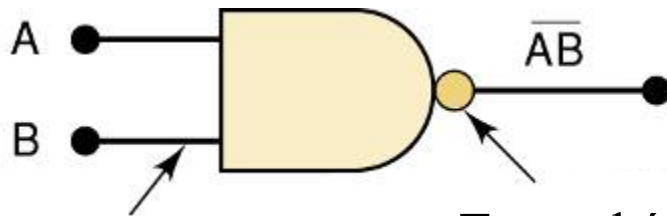
74LS Series

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>

Biểu diễn cổng logic (mở rộng)

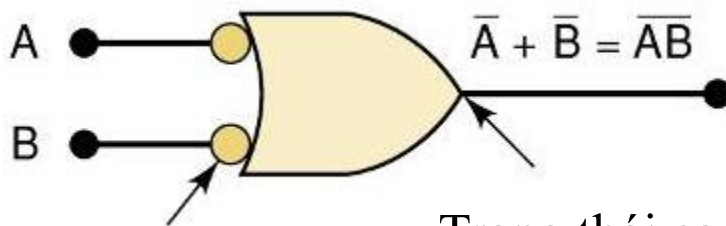
- Ý nghĩa của 2 loại cổng logic **NAND**



Tích cực mức cao

Trạng thái thấp là
trạng thái tích cực

Output là LOW khi
tất cả inputs là HIGH



Tích cực mức thấp

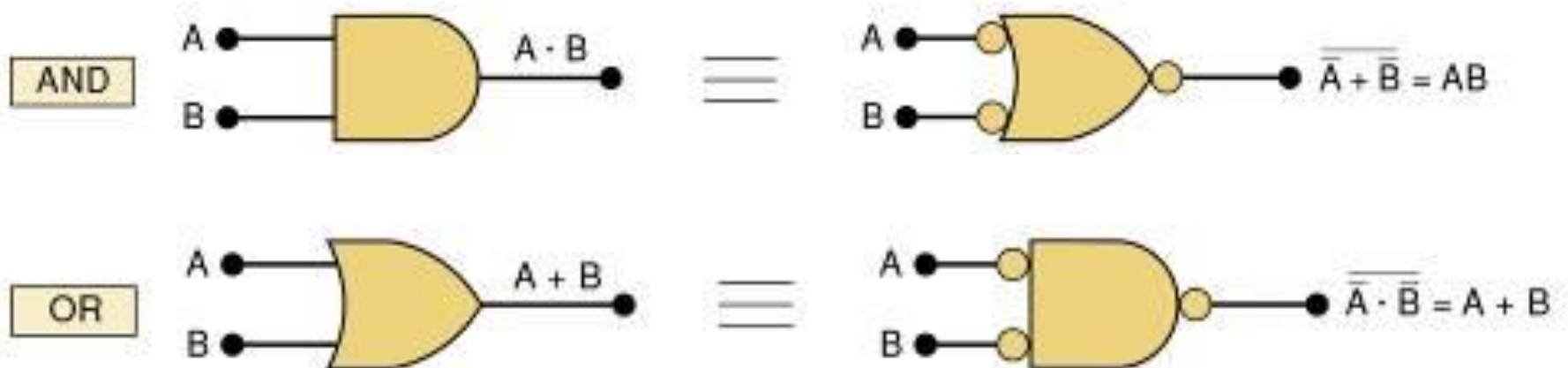
Trạng thái cao là
trạng thái tích cực

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

Biểu diễn cổng logic (mở rộng)

Để biến đổi một cổng logic cơ bản sang một cổng logic tương đương, phải thực hiện các bước sau:

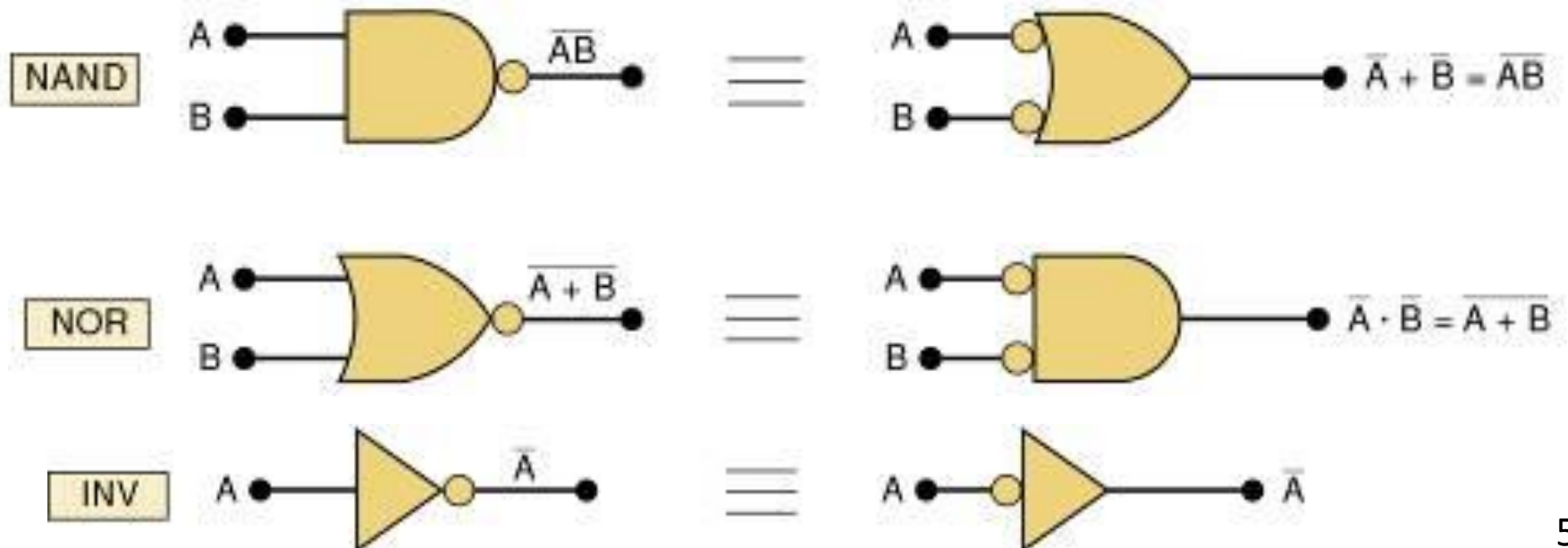
- Chuyển đổi **OR sang AND** hoặc **AND sang OR**
- Nghịch đảo tất cả input và output trong cổng logic cơ bản
 - Thêm ký hiệu dấu bù (bong bóng) tại ngõ vào/ngõ ra không có
 - Xóa ký hiệu dấu bù (bong bóng) tại ngõ vào/ngõ ra có sẵn



Biểu diễn cổng logic (mở rộng)

Để biến đổi một cổng logic cơ bản sang một cổng logic tương đương, phải thực hiện các bước sau:

- Chuyển đổi **OR** sang **AND** hoặc **AND** sang **OR**
- Nghịch đảo tất cả input và output trong cổng logic cơ bản
 - Thêm ký hiệu dấu bù (bong bóng) tại ngõ vào/ngõ ra không có
 - Xóa ký hiệu dấu bù (bong bóng) tại ngõ vào/ngõ ra có sẵn



Thuật ngữ kỹ thuật số

Inverter một cổng logic thực hiện việc đảo ngược/lấy bù ngõ vào của nó

Truth table một bảng chỉ ra giá trị ngõ vào và ngõ ra tương ứng của một mạch logic

Timing diagram một giản đồ chỉ ra dạng sóng của các tín hiệu trong mạch logic tại từng thời điểm

Boolean algebra Phương pháp toán học cho mạch logic

AND gate 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