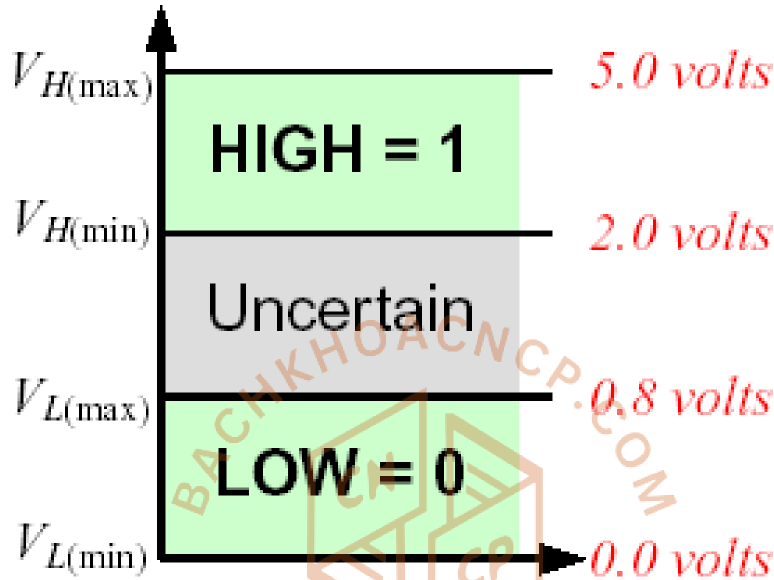
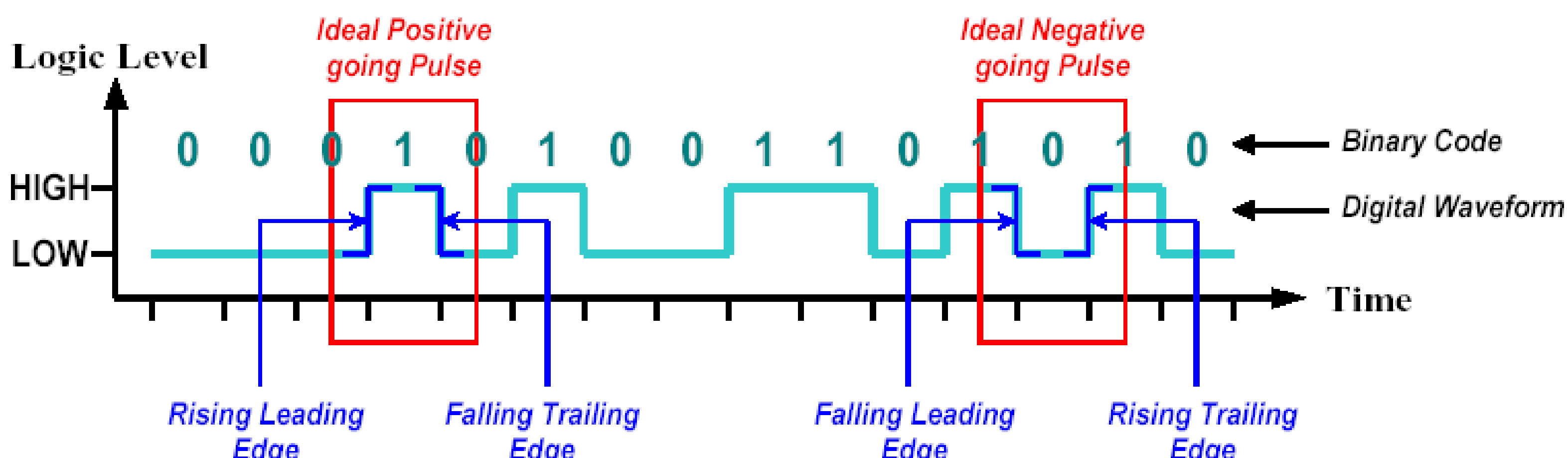
Trạng thái logic của tín hiệu số (Digital Signal):



Giản đồ xung (Waveform) của tín hiệu số:



Chương 2: ĐẠI SỐ BOOLE – CỔNG LOGIC

I. Cấu trúc đại số Boole:

Là cấu trúc đại số được định nghĩa trên 1 tập phần tử nhị phân B = {0, 1} và các phép toán nhị phân: AND (.), OR (+), NOT (’).

x	y	x . y (x AND y)
0	0	0
0	1	0
1	0	0
1	1	1

x	y	x + y (x OR y)
0	0	0
0	1	1
1	0	1
1	1	1

x	x’ (NOT x, $\overline{x}$)
0	1
1	0

*** Thứ tự phép toán:** theo thứ tự dấu ngoặc (), NOT, AND, OR

1. Các tiên đề (Axioms):

a. Tính kín (Closure Property)

b. Phần tử đồng nhất (Identity Element):

$$x \cdot 1 = 1 \cdot x = x$$

$$x + 0 = 0 + x = x$$

c. Tính giao hoán (Commutative Property):

$$x \cdot y = y \cdot x$$

$$x + y = y + x$$

d. Tính phân bố (Distributive Property):

$$x \cdot (y + z) = x \cdot y + x \cdot z$$

$$x + (y \cdot z) = (x + y) \cdot (x + z)$$

e. Phần tử bù (Complement Element):

$$x + \overline{x} = 1 \qquad x \cdot \overline{x} = 0$$

2. Các định lý cơ bản (Basic Theorems):

a. Định lý 1: $\overline{\overline{x}} = x$

b. Định lý 2: $x + x = x$ $\quad$ $x \cdot x = x$

c. Định lý 3: $x + 1 = 1$ $\quad$ $x \cdot 0 = 0$

d. Định lý 4: định lý hấp thu (Absorption)

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

e. Định lý 5: định lý kết hợp (Associative)

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

f. Định lý 6: định lý De Morgan

$\overline{x + y} = \overline{x} \cdot \overline{y}$ $\quad$ $\overline{x \cdot y} = \overline{x} + \overline{y}$

Mở rộng:

$$\overline{x_1 + x_2 + .. + x_n} = \overline{x_1} \cdot \overline{x_2} \,..\, \overline{x_n}$$

$$\overline{x_1 \cdot x_2 \,..\, x_n} = \overline{x_1} + \overline{x_2} + .. + \overline{x_n}$$

II. Hàm Boole (Boolean Function):

1. Định nghĩa:

* Hàm Boole là 1 biểu thức được tạo bởi các biến nhị phân và các phép toán nhị phân NOT, AND, OR.

$$F(x, y, z) = x \cdot y + \overline{x} \cdot \overline{y} \cdot z$$

* Với giá trị cho trước của các biến, hàm Boole sẽ có giá trị là 0 hoặc 1.

* Bảng giá trị:

x	y	z	*F*
0	0	0	*0*
0	0	1	*1*
0	1	0	*0*
0	1	1	*0*
1	0	0	*0*
1	0	1	*0*
1	1	0	*1*
1	1	1	*1*

2. Bù của 1 hàm:

- Sử dụng định lý De Morgan:

$$F = x \cdot y + \overline{x} \cdot \overline{y} \cdot z$$

$$\overline{F} = \overline{x \cdot y + \overline{x} \cdot \overline{y} \cdot z}$$

$$= \overline{(x \cdot y)} \cdot \overline{(\overline{x} \cdot \overline{y} \cdot z)}$$

$$\overline{F} = (\overline{x} + \overline{y}) \cdot (x + y + \overline{z})$$

- Lấy biểu thức đối ngẫu và lấy bù các biến:

<u>* Tính đối ngẫu (Duality):</u> 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.

$$F = x \cdot y + \overline{x} \cdot \overline{y} \cdot z$$

Lấy đối ngẫu: $(x + y) \cdot (\overline{x} + \overline{y} + z)$

Bù các biến: $\overline{F} = (\overline{x} + \overline{y}) \cdot (x + y + \overline{z})$

III. Dạng chính tắc và dạng chuẩn của hàm Boole:

1. Các tích chuẩn (minterm) và tổng chuẩn (Maxterm):

- ***Tích chuẩn (minterm):*** m_i $(0 \le i \le 2^n-1)$ là các số hạng tích (AND) của n biến mà hàm Boole phụ thuộc với quy ước biến đó có bù nếu nó là 0 và không bù nếu là 1.
- ***Tổng chuẩn (Maxterm):*** M_i $(0 \le i \le 2^n-1)$ là các số hạng tổng (OR) của n biến mà hàm Boole phụ thuộc với quy ước biến đó có bù nếu nó là 1 và không bù nếu là 0.

x y z	*minterm*	*Maxterm*
0 0 0	$m_0 = \overline{x}\,\overline{y}\,\overline{z}$	$M_0 = x + y + z$
0 0 1	$m_1 = \overline{x}\,\overline{y}\,z$	$M_1 = x + y + \overline{z}$
0 1 0	$m_2 = \overline{x}\,y\,\overline{z}$	$M_2 = x + \overline{y} + z$
0 1 1	$m_3 = \overline{x}\,y\,z$	$M_3 = x + \overline{y} + \overline{z}$
1 0 0	$m_4 = x\,\overline{y}\,\overline{z}$	$M_4 = \overline{x} + y + z$
1 0 1	$m_5 = x\,\overline{y}\,z$	$M_5 = \overline{x} + y + \overline{z}$
1 1 0	$m_6 = x\,y\,\overline{z}$	$M_6 = \overline{x} + \overline{y} + z$
1 1 1	$m_7 = x\,y\,z$	$M_7 = \overline{x} + \overline{y} + \overline{z}$

$$m_i = \overline{M_i}$$

2. Dạng chính tắc (Canonical Form):

a. Dạng chính tắc 1:

là dạng tổng của các tích chuẩn (minterm) làm cho hàm Boole có giá trị 1

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$$F(x, y, z) = \bar{x}\,\bar{y}\,z + \bar{x}\,y\,\bar{z} + x\,\bar{y}\,z + x\,y\,\bar{z} + x\,y\,z$$
$$= m_1 + m_2 + m_5 + m_6 + m_7$$
$$= \Sigma m(1, 2, 5, 6, 7)$$
$$= \Sigma(1, 2, 5, 6, 7)$$

$$F(x, y, z) = (x + y + z)(x + \bar{y} + \bar{z})(\bar{x} + y + z)$$
$$= M_0 \;.\; M_3 \;.\; M_4$$
$$= \Pi M(0, 3, 4) = \Pi(0, 3, 4)$$

b. Dạng chính tắc 2:

là dạng tích của các tổng chuẩn (Maxterm) làm cho hàm Boole có giá trị 0

* Trường hợp hàm Boole tùy định (don’t care):

Hàm Boole n biến có thể không được định nghĩa hết tất cả 2^n tổ hợp của n biến phụ thuộc. Khi đó tại các tổ hợp không sử dụng này, hàm Boole sẽ nhận giá trị tùy định (don’t care), nghĩa là hàm Boole có thể nhận giá tri 0 hoặc 1.

x y z	F
0 0 0	X
0 0 1	1
0 1 0	1
0 1 1	0
1 0 0	0
1 0 1	1
1 1 0	1
1 1 1	X

$$F(x, y, z) = \Sigma(1, 2, 5, 6) + d(0, 7)$$

$$= \Pi(3, 4) \cdot D(0, 7)$$

3. Dạng chuẩn (Standard Form):

a. Dạng chuẩn 1:

là dạng tổng các tích (S.O.P – Sum of Product)

$$F(x, y, z) = x\,y + z$$

* $F(x, y, z) = x\,y + z$

$= x\,y\,(\overline{z} + z) + (\overline{x} + x)\,(\overline{y} + y)\,z$

$= x\,y\,\overline{z} + x\,y\,z + \overline{x}\,\overline{y}\,z + \overline{x}\,y\,z + x\,\overline{y}\,z + \cancel{x\,y\,z}$

$= m_6 + m_7 + m_1 + m_5 + m_3$

$= \Sigma(1, 3, 5, 6, 7)$

* $F(x, y, z) = x\,y + z$

$= (x + z)\,(y + z)$

$= (x + \overline{y}\,y + z)\,(\overline{x}\,x + y + z)$

$= (x + \overline{y} + z)\,(x + y + z)\,(\overline{x} + y + z)\,\cancel{(x + y + z)}$

$= M_2 \,.\, M_0 \,.\, M_4$

$= \Pi(0, 2, 4)$

b. Dạng chuẩn 2:

là dạng tích các tổng (P.O.S – Product of Sum)

$$F(x, y, z) = (x + \overline{z})\,\overline{y}$$

* $F(x, y, z) = (x + \overline{z})\,\overline{y} = x\overline{y} + \overline{y}\,\overline{z}$

$= x\overline{y}(\overline{z} + z) + (\overline{x} + x)\overline{y}\,\overline{z}$

$= x\overline{y}\,\overline{z} + x\overline{y}z + \overline{x}\,\overline{y}\,\overline{z} + \cancel{x\overline{y}\,\overline{z}}$

$= m_4 + m_5 + m_0$

$= \Sigma(0, 4, 5)$

* $F(x, y, z) = (x + \overline{z})\,\overline{y}$

$= (x + \overline{y}y + \overline{z})(\overline{x}x + \overline{y} + \overline{z}z)$

$= (x + \overline{y} + \overline{z})(x + y + \overline{z})$

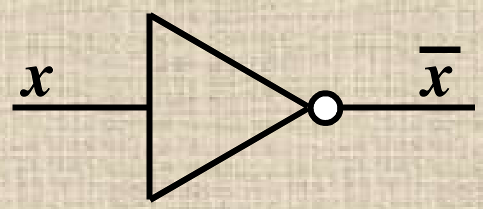
$(\overline{x} + \overline{y} + \overline{z})(\overline{x} + \overline{y} + z)\cancel{(x + \overline{y} + \overline{z})}(x + \overline{y} + z)$

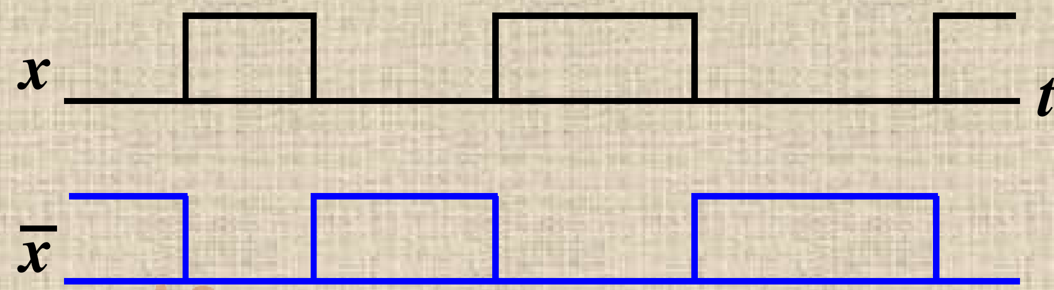
$= M_3 . M_1 . M_7 . M_6 . M_2$

$= \Pi(1, 2, 3, 6, 7)$

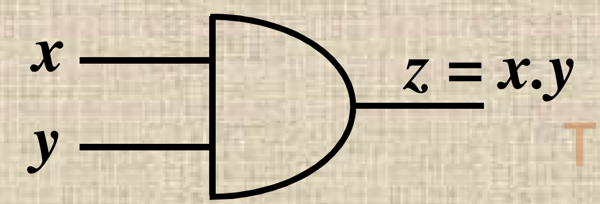
IV. Cổng logic:

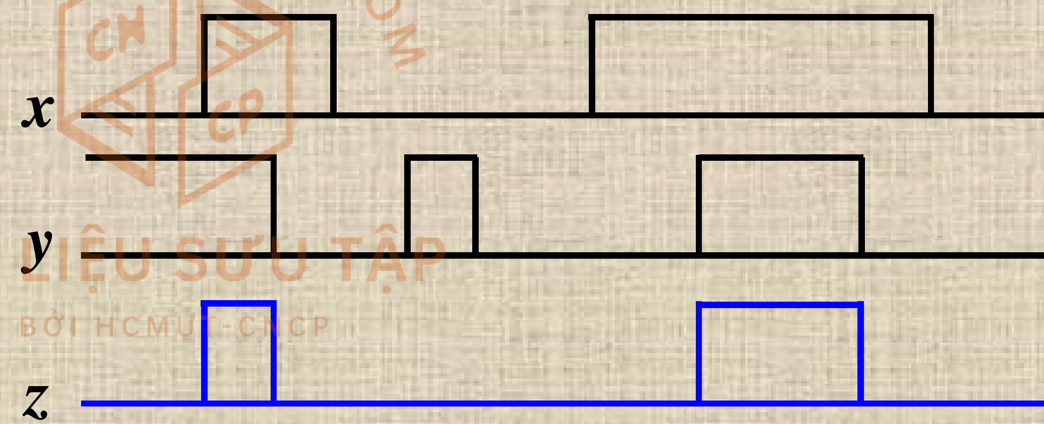
1. Cổng NOT:





2. Cổng AND:

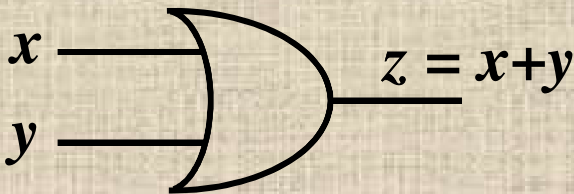




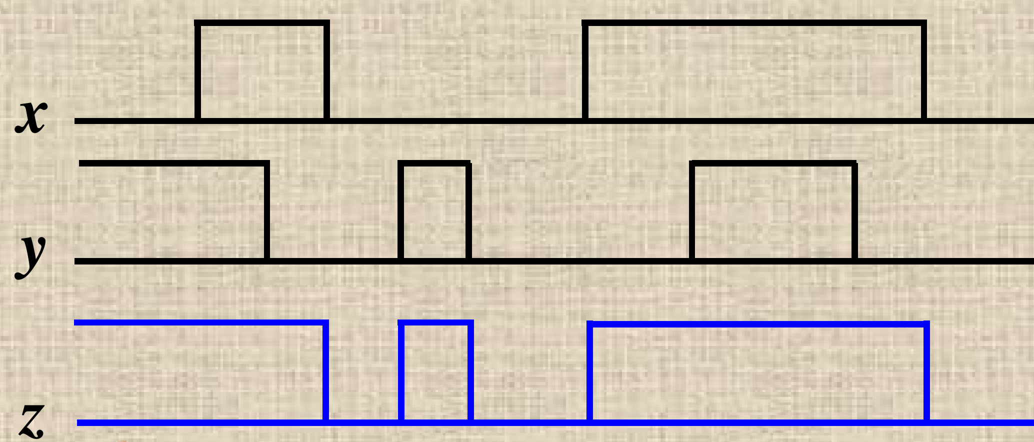
x	*y*	*z*
0	*0*	*0*
0	*1*	*0*
1	*0*	*0*
1	*1*	*1*

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

3. Cổng OR:

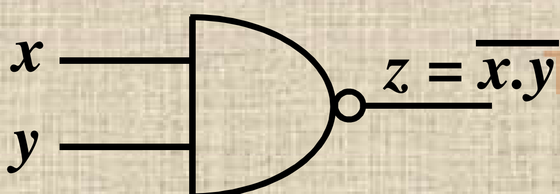


x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

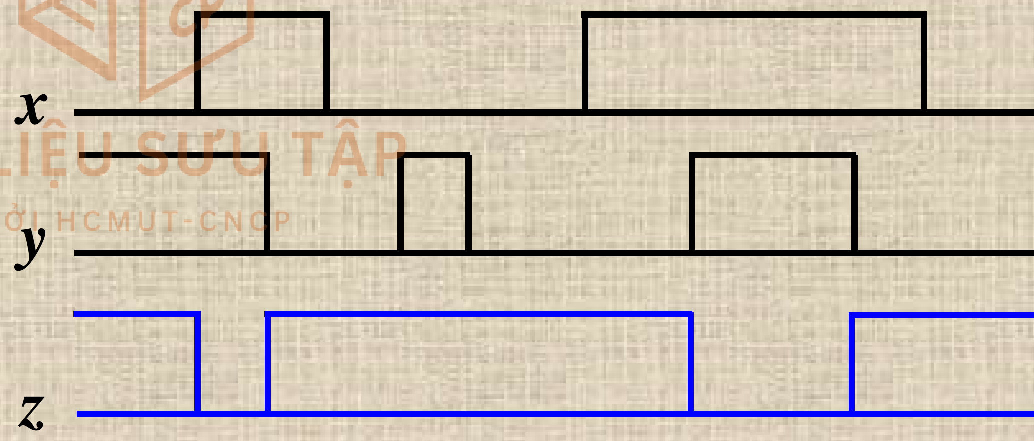


Với cổng OR có nhiều ngõ vào, ngõ ra sẽ là 0 nếu tất cả các ngõ vào đều là 0

4. Cổng NAND:

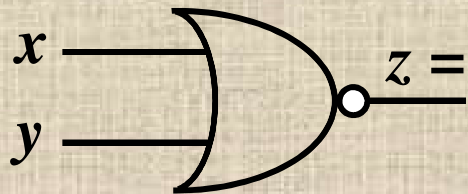


x	y	z
0	0	1
0	1	1
1	0	1
1	1	0

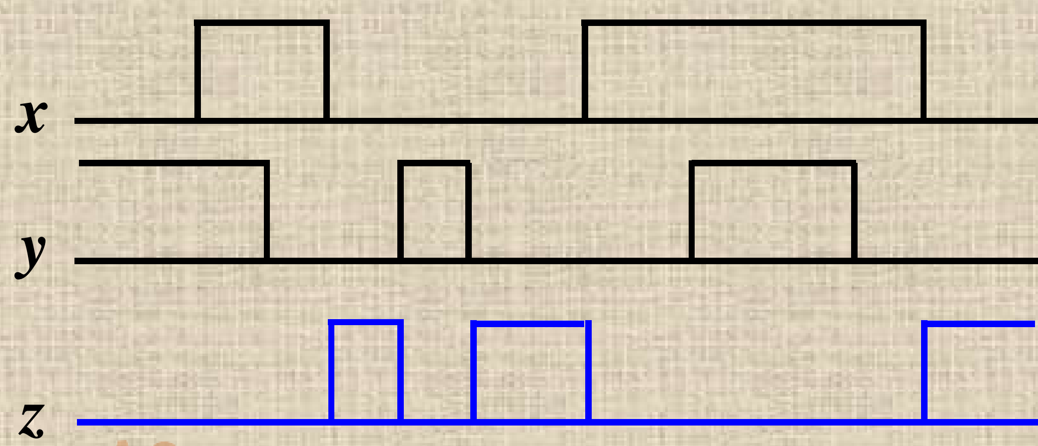


Với cổng NAND có nhiều ngõ vào, ngõ ra sẽ là 0 nếu tất cả các ngõ vào đều là 1

5. Cổng NOR:



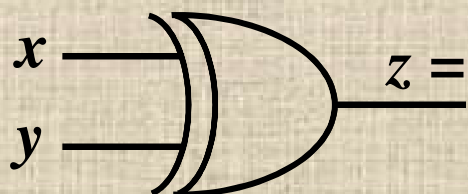
$z = \overline{x+y}$



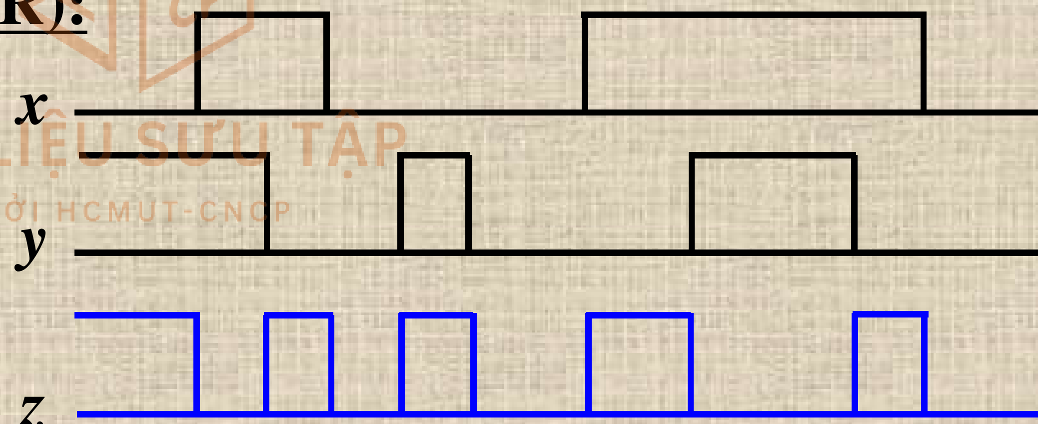
x	y	z
0	0	1
0	1	0
1	0	0
1	1	0

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

6. Cổng XOR (Exclusive_OR):



$z = x \oplus y$

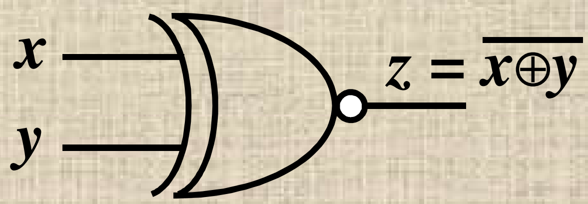


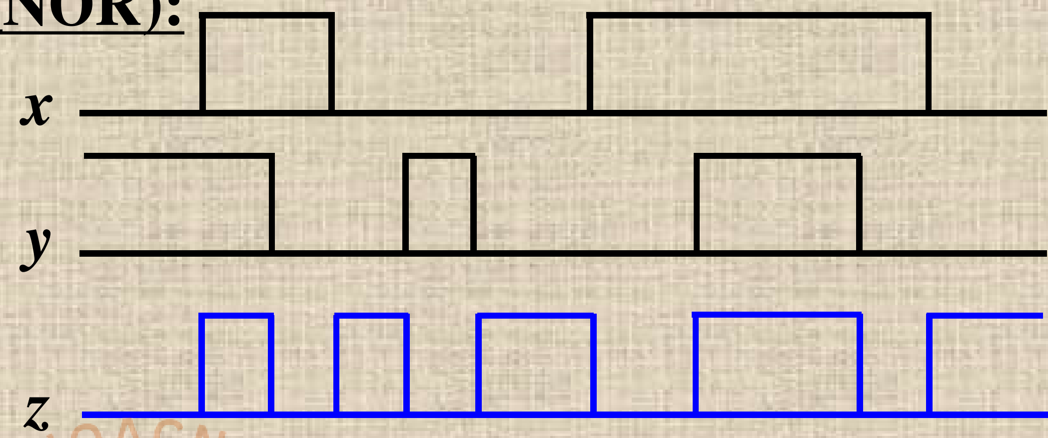
x	y	z
0	0	0
0	1	1
1	0	1
1	1	0

Với cổng XOR có nhiều ngõ vào, ngõ ra sẽ là 1 nếu tổng số bit 1 ở các ngõ vào là số lẻ

$$z = x \oplus y = \overline{x}\, y + x\, \overline{y} = (x + y)(\overline{x} + \overline{y})$$

7. Cổng XNOR (Exclusive_NOR):





x	y	z
0	0	1
0	1	0
1	0	0
1	1	1

Với cổng XNOR có nhiều ngõ vào, ngõ ra sẽ là 1 nếu tổng số bit 1 ở các ngõ vào là số chẵn

$$z = \overline{x \oplus y} = \overline{x}\,\overline{y} + x\,y = (\overline{x} + y)(x + \overline{y})$$

V. Rút gọn hàm Boole:

Rút gọn (tối thiểu hóa) hàm Boole nghĩa là đưa hàm Boole về dạng biểu diễn đơn giản nhất, sao cho:

- Biểu thức có chứa ít nhất các thừa số và mỗi thừa số chứa ít nhất các biến.
- Mạch logic thực hiện có chứa ít nhất các vi mạch số.

1. Phương pháp đại số:

Dùng các định lý và tiên đề để rút gọn hàm.

$$
\begin{aligned}
F(A, B, C) &= \Sigma(2, 3, 5, 6, 7) \\
&= \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC \\
&= \overline{A}B(\overline{C} + C) + AC(\overline{B} + B) + AB(\overline{C} + C) \\
&= \overline{A}B + AC + AB \\
&= (\overline{A} + A)B + AC \\
&= B + AC
\end{aligned}
$$

2. Phương pháp bìa KARNAUGH:

a. Cách biểu diễn:

- Bìa K gồm các ô vuông, mỗi ô vuông biểu diễn cho tổ hợp n biến. Như vậy bìa K cho n biến sẽ có 2^n ô.

- Hai ô được gọi là kề cận nhau khi tổ hợp biến mà chúng biểu diễn chỉ khác nhau 1 biến.

- Trong ô sẽ ghi giá trị tương ứng của hàm Boole tại tổ hợp đó. Ở dạng chính tắc 1 thì đưa các giá trị 1 và X lên các ô, không đưa các giá trị 0. Ngược lại, dạng chính tắc 2 thì chỉ đưa giá trị 0 và X.

*** Bìa 2 biến:**

$F(A, B) = \Sigma(0, 2) + d(3) = \Pi(1) . D(3)$

F: B \ A	0	1
0	0	2
1	1	3

F: B \ A	0	1
0	1	1
1		X

F: B \ A	0	1
0		
1	0	X

*** Bìa 3 biến:**

F: C \ AB	00	01	11	10
0	0	2	6	4
1	1	3	7	5

$$F(A, B, C) = \Sigma(2, 4, 7) + d(0, 1) = \Pi(3, 5, 6) \,.\, D(0, 1)$$

F: C \ AB	00	01	11	10
0	X	1		1
1	X		1	

F: C \ AB	00	01	11	10
0	X		0	
1	X	0		0

* Bìa 4 biến:

F — CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

* Bìa 5 biến:

F — A	0				1			
DE \ BC	00	01	11	10	10	11	01	00
00	0	4	12	8	24	28	20	16
01	1	5	13	9	25	29	21	17
11	3	7	15	11	27	31	23	19
10	2	6	14	10	26	30	22	18

b. Rút gọn bìa Karnaugh:

* Nguyên tắc:

- Liên kết đôi: Khi liên kết (OR) hai ô có giá trị 1 (Ô_1) kề cận với nhau trên bìa K, ta sẽ được 1 số hạng tích mất đi 1 biến so với tích chuẩn (biến mất đi là biến khác nhau giữa 2 ô). Hoặc khi liên kết (AND) hai ô có giá trị 0 (Ô_0) kề cận với nhau trên bìa K, ta sẽ được 1 số hạng tổng mất đi 1 biến so với tổng chuẩn (biến mất đi là biến khác nhau giữa 2 ô).

F

C \ AB	00	01	11	10
0		1	1	
1				

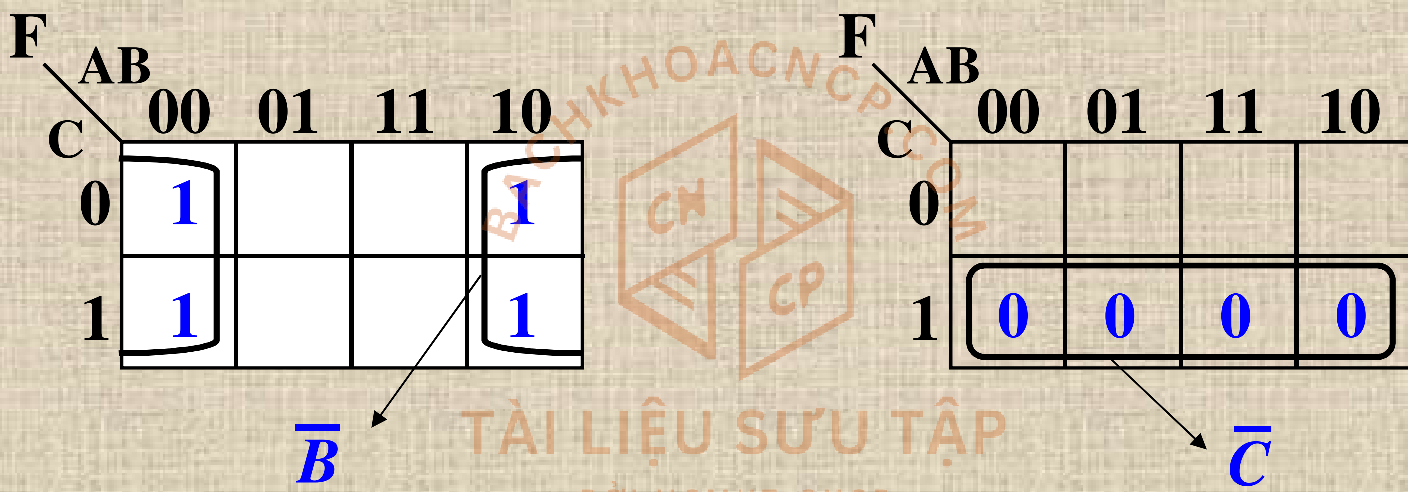
$B\overline{C}$

F

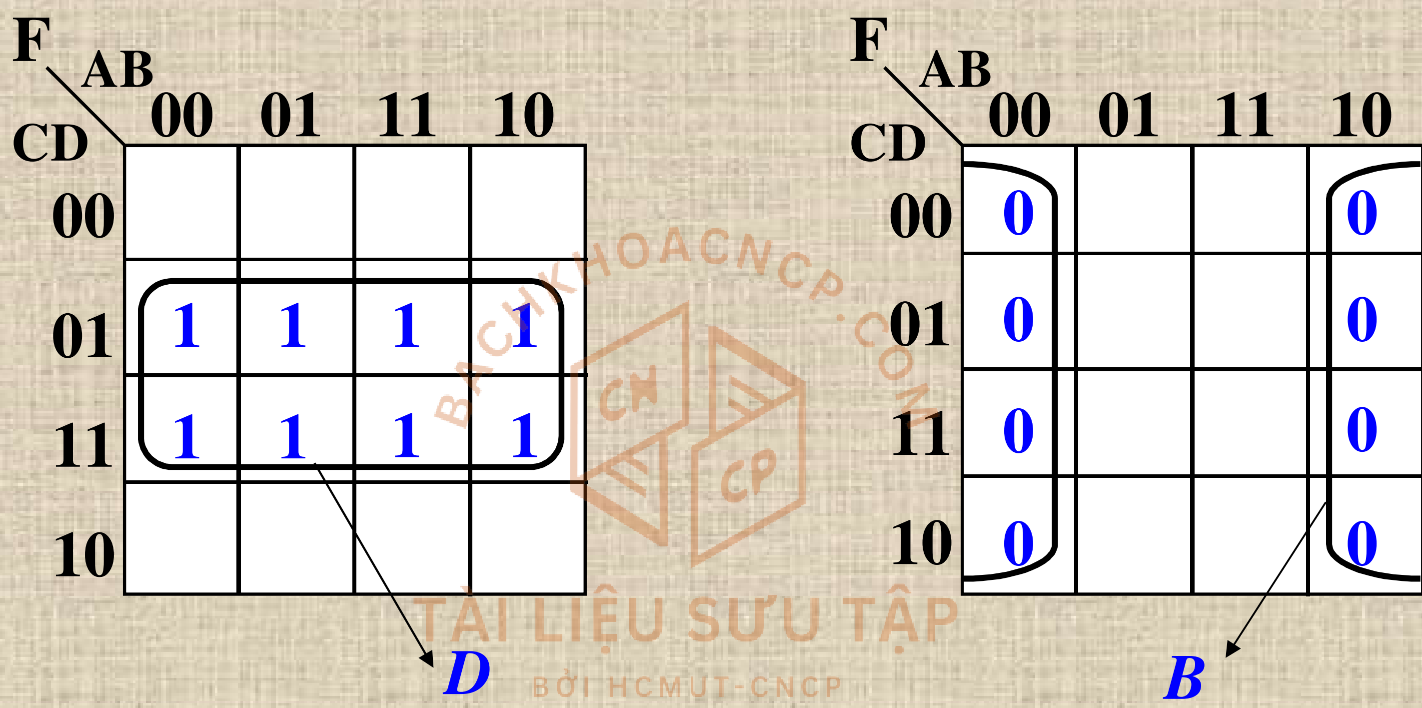
C \ AB	00	01	11	10
0				0
1				0

$\overline{A}+B$

- Liên kết 4: Tương tự như liên kết đôi khi liên kết 4 Ô_1 hoặc 4 Ô_ 0 kề cận với nhau, ta sẽ loại đi được 2 biến (2 biến khác nhau giữa 4 ô)



- Liên kết 8: liên kết 8 ô kề cận với nhau, ta sẽ loại đi được 3 biến (3 biến khác nhau giữa 8 ô)

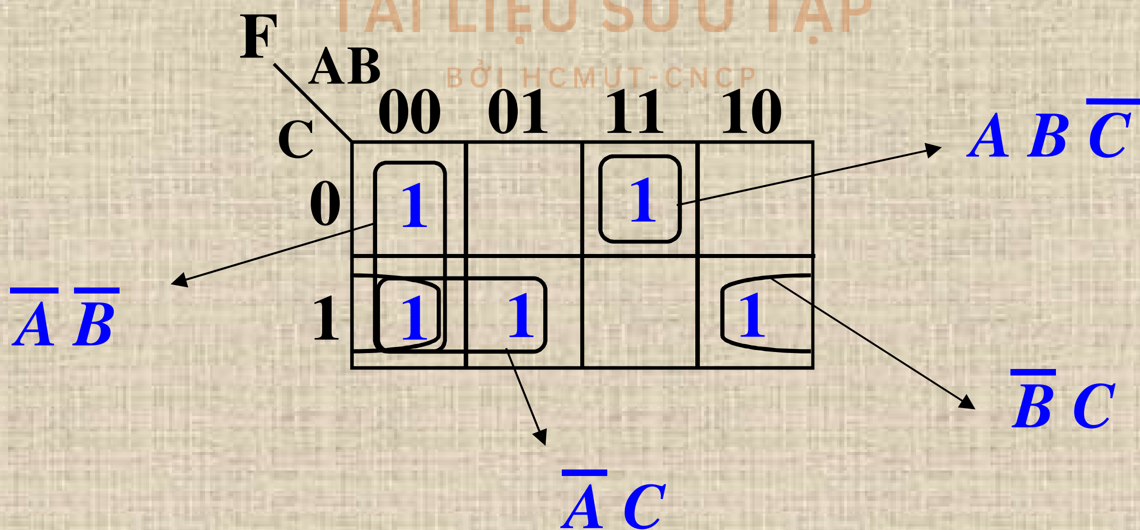


- Liên kết 2^k: khi ta liên kết 2^k Ô_1 hoặc 2^k Ô_0 kề cận với nhau ta sẽ loại đi được k biến (k biến khác nhau giữa 2^k ô)

* Các bước thực hiện rút gọn theo dạng S.O.P:

- Biểu diễn các Ô_1 lên bìa Karnaugh
- Thực hiện các liên kết có thể có sao cho các Ô_1 được liên kết ít nhất 1 lần; mỗi liên kết cho ta 1 số hạng tích. (Nếu Ô_1 không có kề cận với các Ô_1 khác thì ta có liên kết 1: số hạng tích chính bằng minterm của ô đó).
- Biểu thức rút gọn có được bằng cách lấy tổng (OR) của các số hạng tích liên kết trên.

$$F(A, B, C) = \Sigma(0, 1, 3, 5, 6) = \overline{A}\,\overline{B} + \overline{A}\,C + \overline{B}\,C + A\,B\,\overline{C}$$



* Các bước thực hiện rút gọn theo dạng P.O.S:

- Biểu diễn các Ô_0 lên bìa Karnaugh
- Thực hiện các liên kết có thể có sao cho các Ô_0 được liên kết ít nhất 1 lần; mỗi liên kết cho ta 1 số hạng tổng.
- Biểu thức rút gọn có được bằng cách lấy tích (AND) của các số hạng tổng liên kết trên.

$$F(A, B, C, D) = \Pi(0, 4, 8, 9, 12, 13, 15)$$

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

F

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

$(C + D)$

$(\overline{A} + C)$

$(\overline{A} + \overline{B} + \overline{D})$

<u>* Trường hợp rút gọn hàm Boole có tùy định:</u> **thì ta có thể coi các Ô tùy định này là Ô_1 hoặc Ô_0 sao cho có lợi khi liên kết (nghĩa là có được liên kết nhiều Ô kề cận nhất)**

$$F(A, B, C, D) = \Sigma(0, 4, 8, 10) + d\,(2, 12, 15)$$

$$= \overline{B}\,\overline{D} + \overline{C}\,\overline{D}$$

F

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

$\overline{C}\,\overline{D}$

$\overline{B}\,\overline{D}$

$$F(A, B, C, D) = \Pi(0, 2, 3, 4, 6, 10, 14) \,.\, D(8, 9, 11, 12, 13)$$

$$= D\,(B + \overline{C})$$

F

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

→ D

→ $(B + \overline{C})$

*** Chú ý:**

- Ưu tiên liên kết cho các ô chỉ có 1 kiểu liên kết (phải là liên kết có nhiều ô nhất).

- Khi liên kết phải đảm bảo có chứa ít nhất 1 ô chưa được liên kết lần nào.

- Có thể có nhiều cách liên kết có kết quả tương đương nhau

- Ta coi các tùy định như là những ô đã liên kết rồi.

Vd: Rút gọn các hàm

$$F_1(A, B, C, D) = \Sigma(1, 3, 5, 12, 13, 14, 15) + d(7, 8, 9)$$

$$F_2(A, B, C, D) = \Pi(1, 3, 7, 11, 15) \cdot D(0, 2, 5)$$

$$F_1(A, B, C, D, E) = \Sigma(1, 3, 5, 7, 12, 14, 29, 31) + d(13, 15, 17, 19, 20, 21, 22, 23)$$

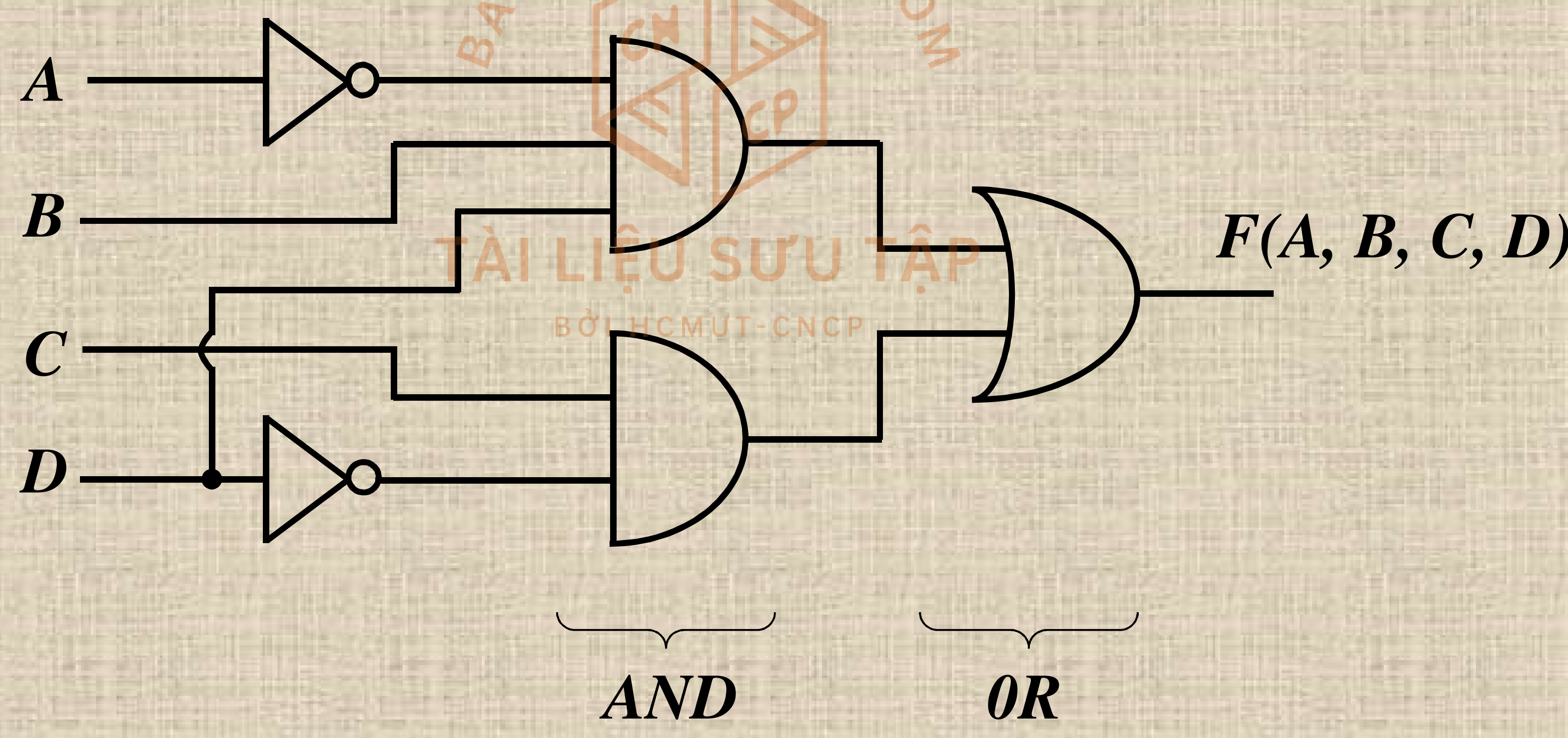
$$F_2(A, B, C, D, E) = \Pi(0, 8, 12, 13, 16, 18, 28, 30) \cdot D(2, 6, 10, 14, 15, 24, 26)$$

VI. Thực hiện hàm Boole bằng cổng logic:

1. Cấu trúc cổng AND _ OR:

Cấu trúc AND_OR là sơ đồ logic thực hiện cho hàm Boole biểu diễn theo dạng tổng các tích (S.O.P)

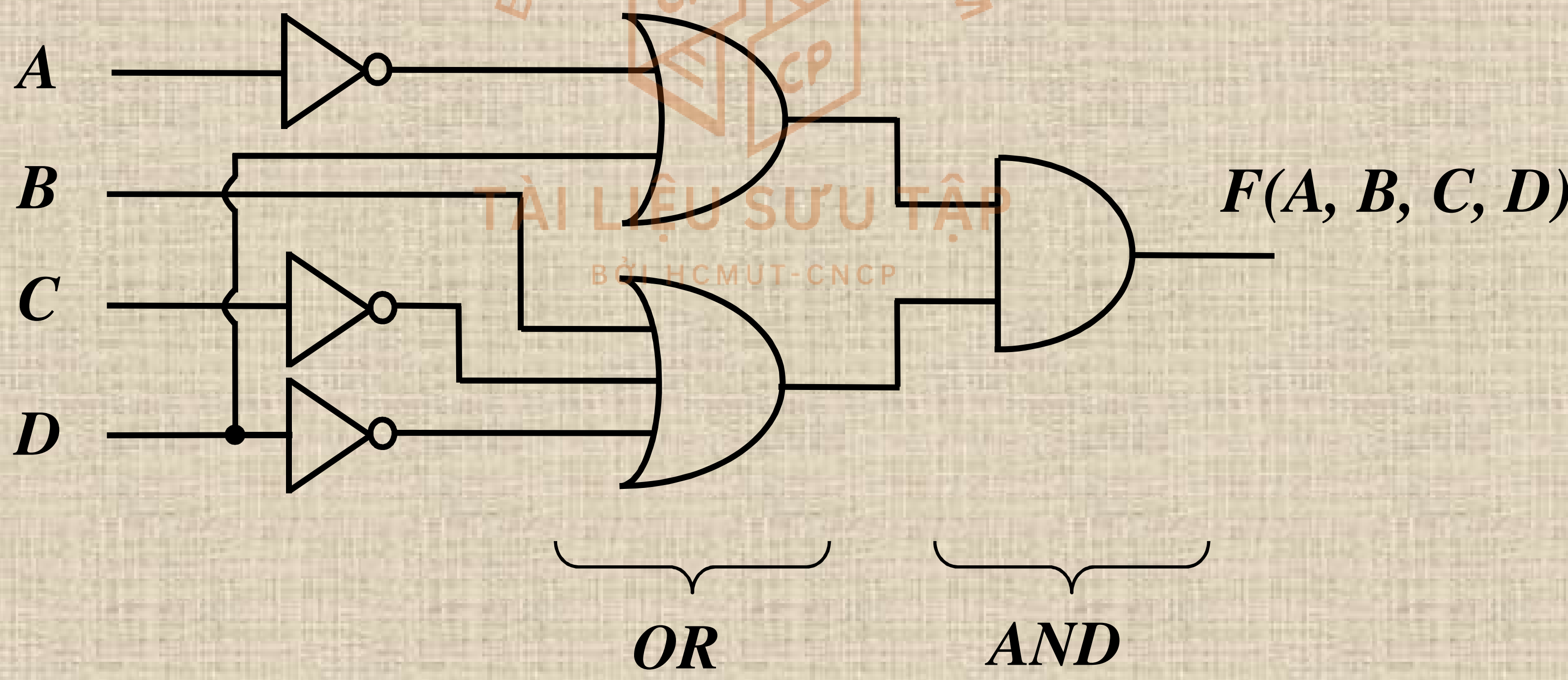
$$F(A, B, C, D) = \overline{A}\,B\,D + C\,\overline{D}$$



2. Cấu trúc cổng OR _ AND :

Cấu trúc OR_AND là sơ đồ logic thực hiện cho hàm Boole biểu diễn theo dạng tích các tổng (P.O.S).

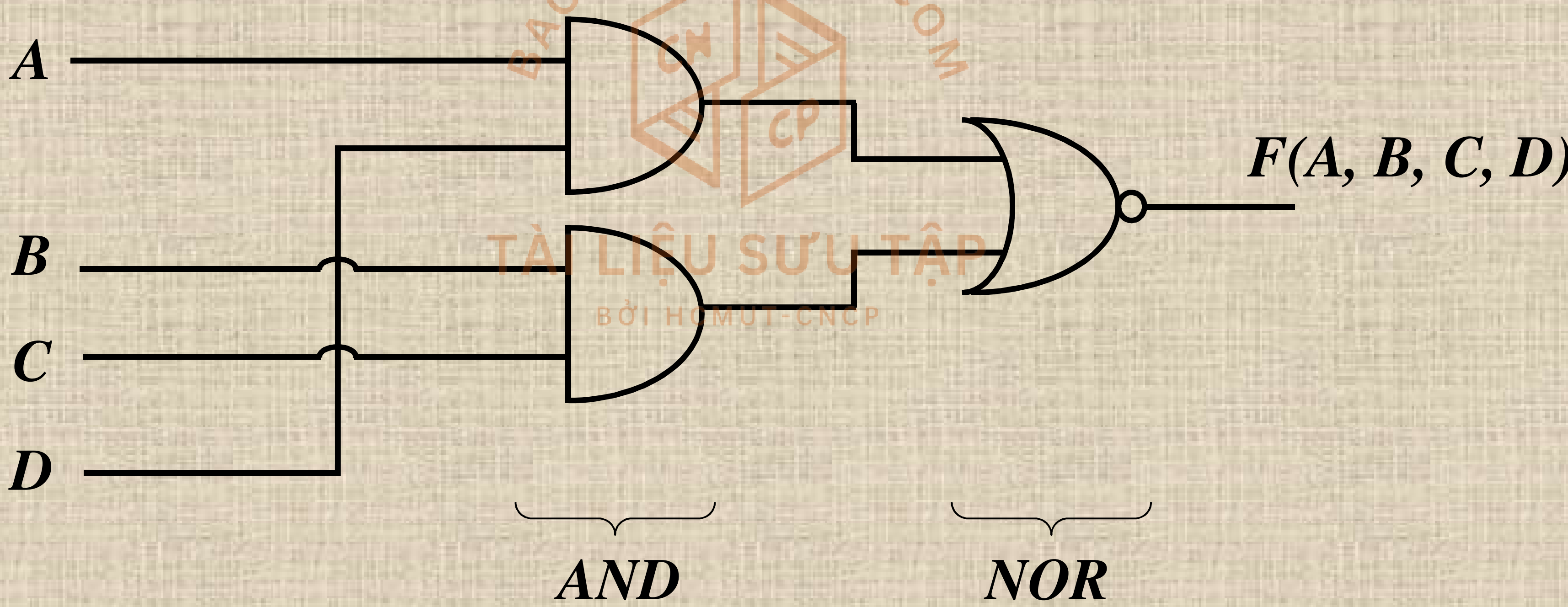
$$F(A, B, C, D) = (\overline{A} + D)\ (B + \overline{C} + \overline{D})$$



3. Cấu trúc cổng AND _ OR _ INVERTER (AOI):

Cấu trúc AOI là sơ đồ logic thực hiện cho hàm Boole biểu diễn theo dạng bù (INVERTER = NOT) của tổng các tích.

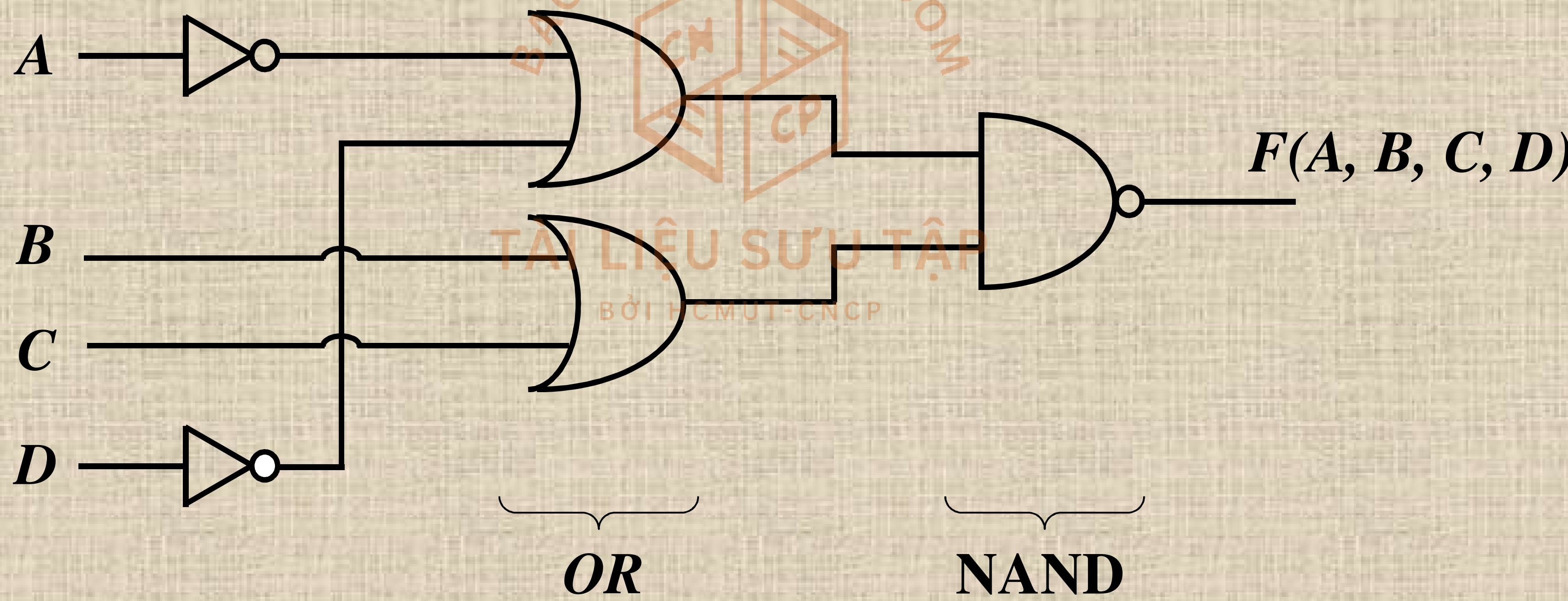
$$F(A, B, C, D) = \overline{A\,D + B\,C}$$



<u>**4. Cấu trúc cổng OR _ AND _ INVERTER (OAI):**</u>

Cấu trúc OAI là sơ đồ logic thực hiện cho hàm Boole biểu diễn theo dạng bù của tích các tổng.

$$F(A, B, C, D) = \overline{(\overline{A} + \overline{D})(B + C)}$$



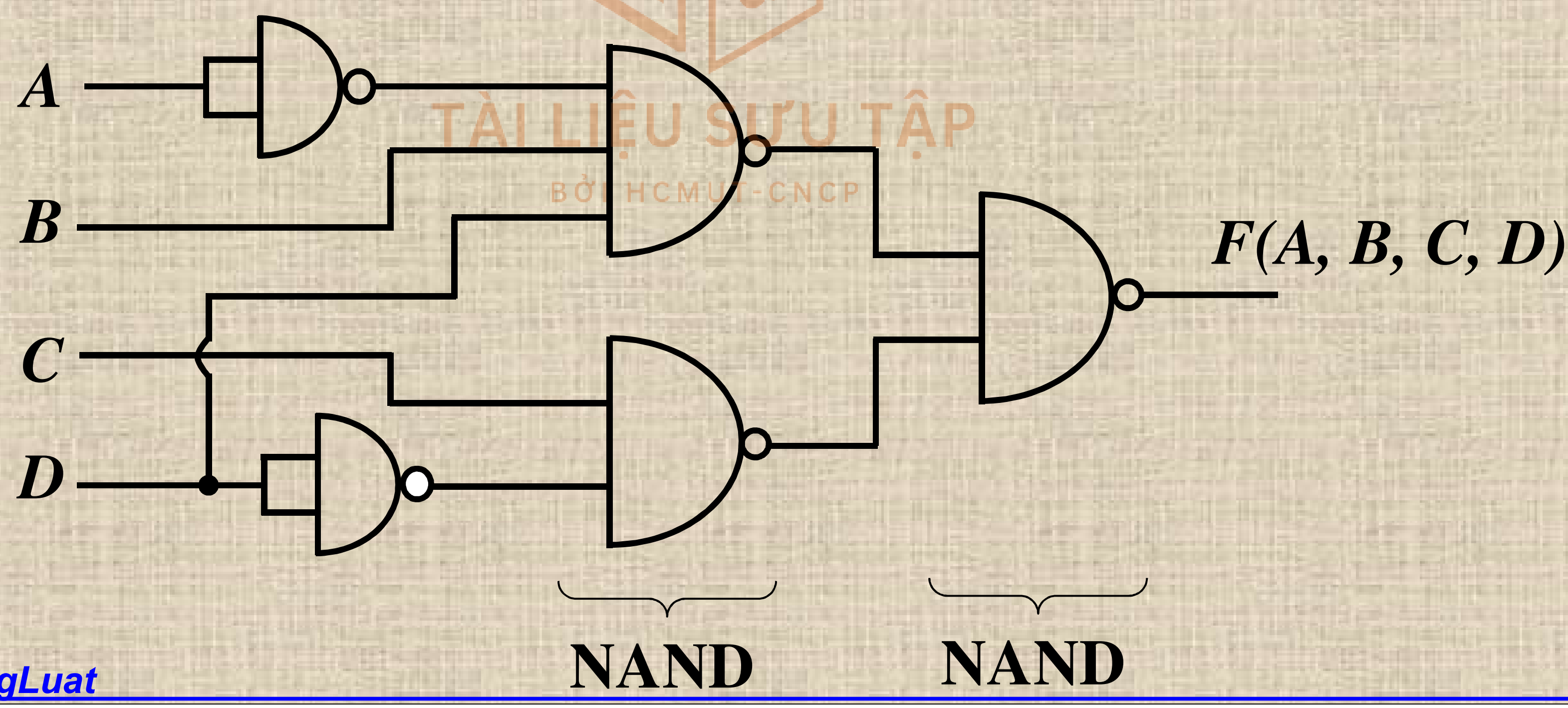
5. Cấu trúc toàn cổng NAND:

Cấu trúc NAND là sơ đồ logic thực hiện cho hàm Boole có biểu thức là dạng bù của 1 số hạng tích.

- **Dùng định lý De-Morgan để biến đổi số hạng tổng thành tích.**
- **Cổng NOT cũng được thay thế bằng cổng NAND**

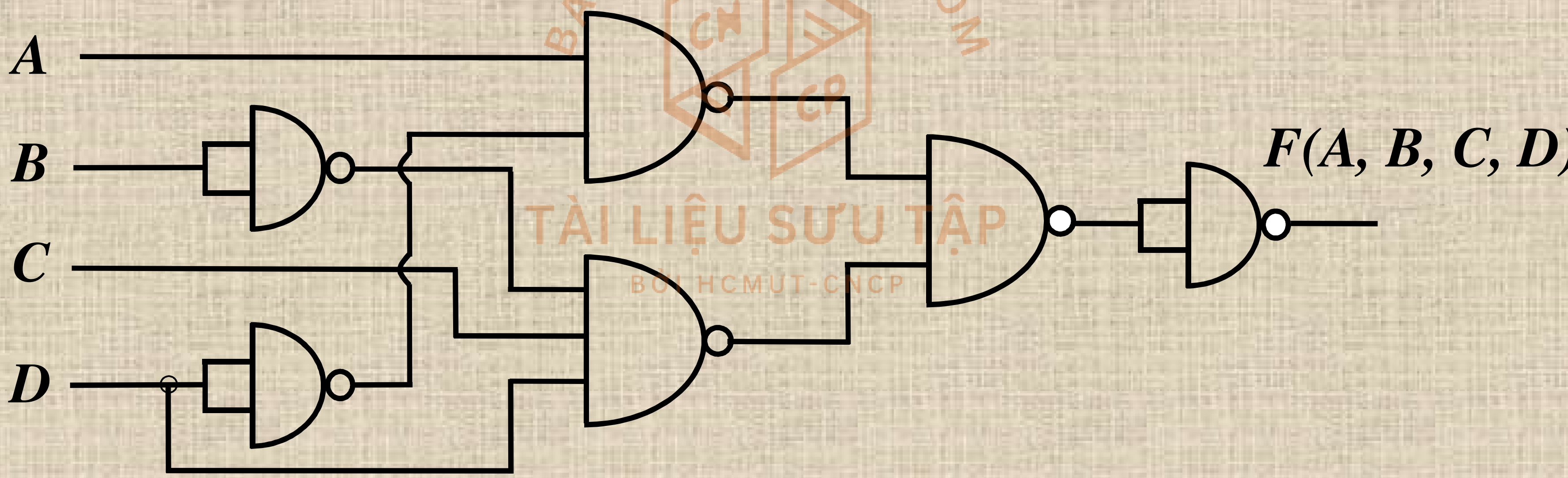
$$F(A, B, C, D) = \overline{\overline{\bar{A}\, B\, D + C\, \bar{D}}}$$

$$= \overline{\overline{\bar{A}\, B\, D} \;.\; \overline{C\, \bar{D}}}$$



$$F(A, B, C, D) = \overline{\overline{(\overline{A} + D)}}\ \overline{\overline{(B + \overline{C} + \overline{D})}}$$

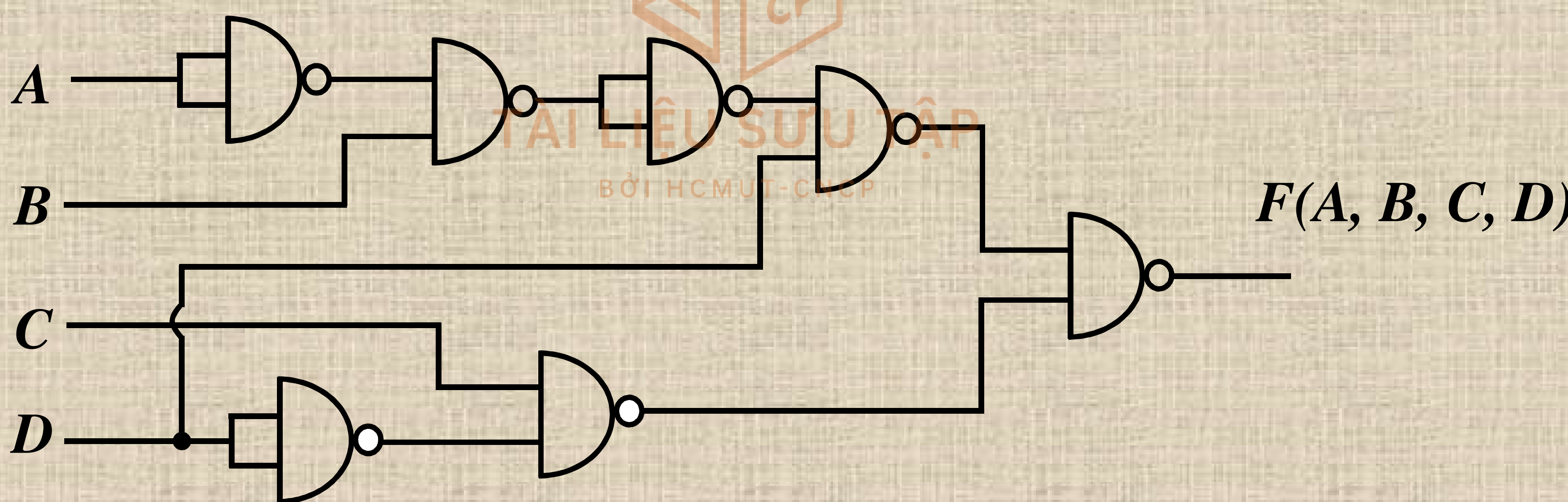
$$= \overline{\overline{\overline{A\,\overline{D}}\ .\ \overline{\overline{B}\,C\,D}}}$$



- Trong thực tế người ta chỉ sử dụng 1 loại cổng NAND 2 ngõ vào; khi đó ta phải biến đổi biểu thức sao cho chỉ có dạng bù trên 1 số hạng tích chỉ có 2 biến

$$F(A, B, C, D) = \overline{\overline{\overline{A}\,B\,D} \,.\, \overline{C\,\overline{D}}}$$

$$= \overline{\overline{\overline{\overline{A}\,B}\,D} \,.\, \overline{C\,\overline{D}}}$$



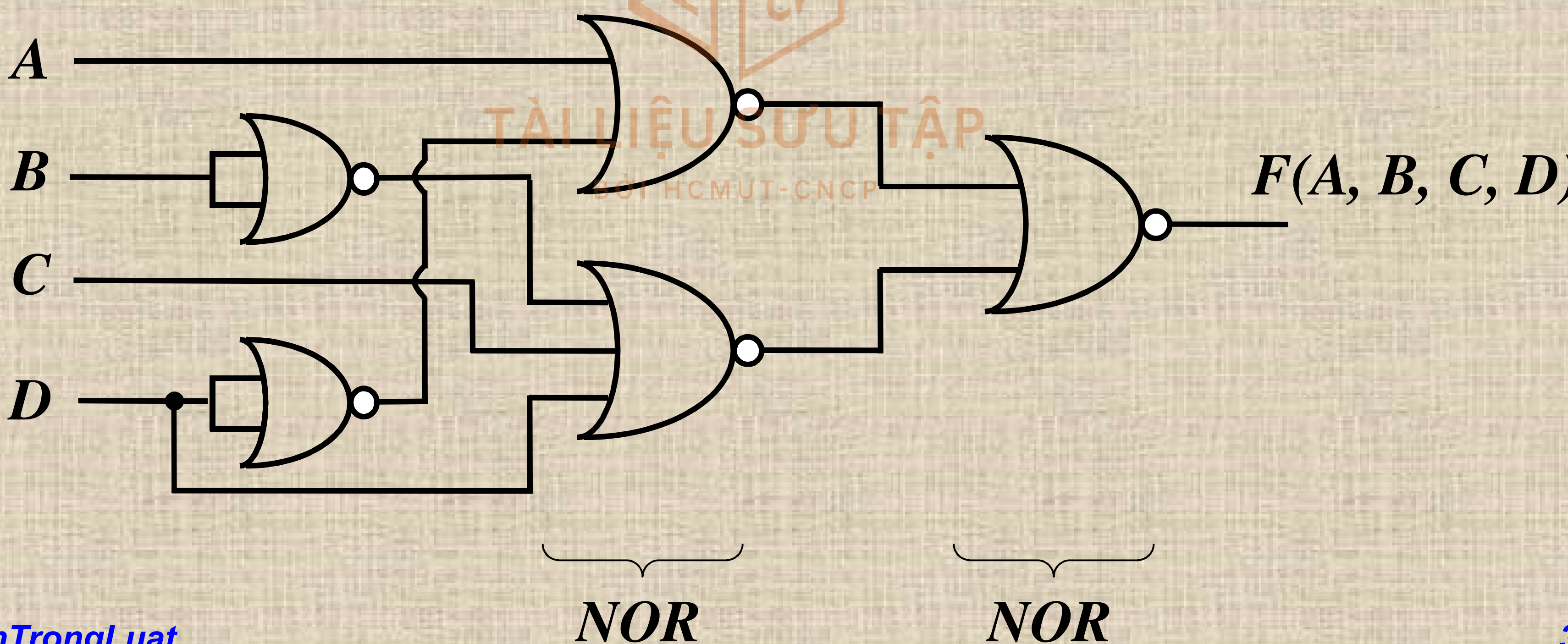
6. Cấu trúc toàn cổng NOR:

Cấu trúc NOR là sơ đồ logic thực hiện cho hàm Boole có biểu thức là dạng bù của 1 số hạng tổng.

- **Dùng định lý De-Morgan để biến đổi số hạng tích thành tổng**
- **Cổng NOT cũng được thay thế bằng cổng NOR**

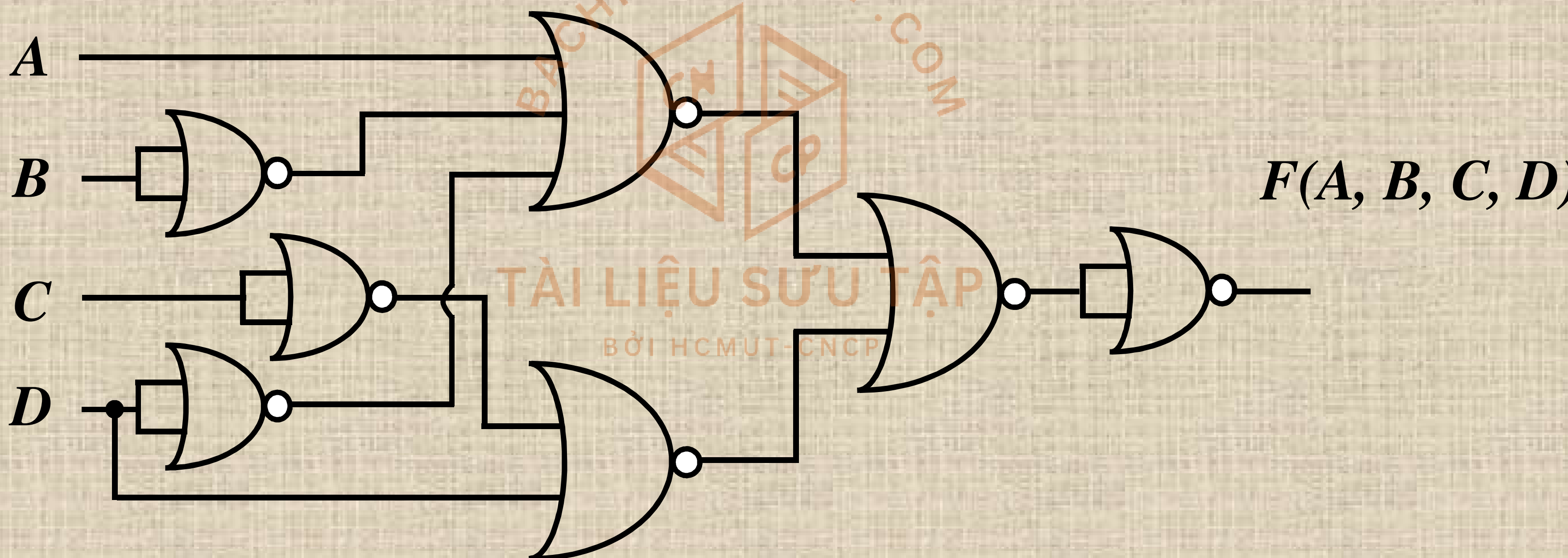
$$F(A,B,C,D) = \overline{\overline{(A+\overline{D})\,(\overline{B}+C+D)}}$$

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



$$F(A, B, C, D) = \overline{\overline{\overline{A}\,B\,D}} + \overline{\overline{C\,\overline{D}}}$$

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



$$F(A, B, C, D) = \overline{\overline{(\overline{A} + D)(B + \overline{C})(\overline{C} + D)}}$$

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

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