

Chương 3

Cổng logic

Nội dung chương 3

- Mức logic 0, 1
- Cổng logic cơ bản
- Chuyển đổi giữa các cổng logic
- Thực hiện hàm Boole bằng cổng logic

Mức logic 0, 1

- Nhiều đại lượng vật lý tồn tại ở một trong hai giá trị đối nghịch
- Ví dụ:
 - Đèn: sáng <> tắt
 - Khóa: mở <> đóng
 - Điện áp: cao <> thấp
 - Biểu thức: đúng <> sai
- Dùng hai mức logic 0, 1 để biểu diễn hai giá trị đối nghịch

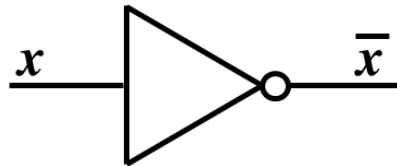
Cổng logic

- Cổng logic là thành phần cơ bản để xây dựng mạch số
- Cổng logic được thiết kế trên cơ sở các phần tử linh kiện bán dẫn như diode, BJT, FET
- Phân loại:
 - Theo chức năng
 - Theo phương pháp chế tạo
 - Ngõ ra

Cổng logic (tt)

- Cổng đảo (NOT)

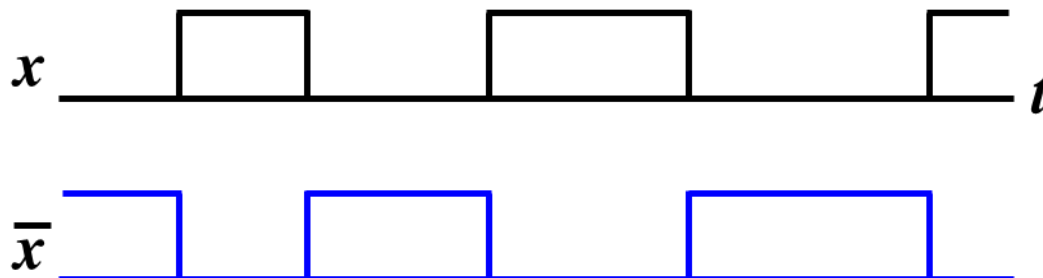
- Ký hiệu (ANSI):



- Phương trình logic: $y = \bar{x}$
- Bảng trạng thái:

x	$y = \bar{x}$
0	1
1	0

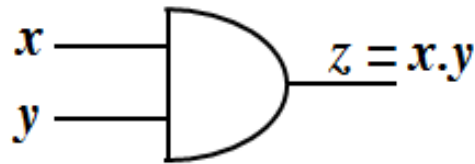
- Dạng sóng:



Cổng logic (tt)

- Cổng AND

- Ký hiệu (ANSI):

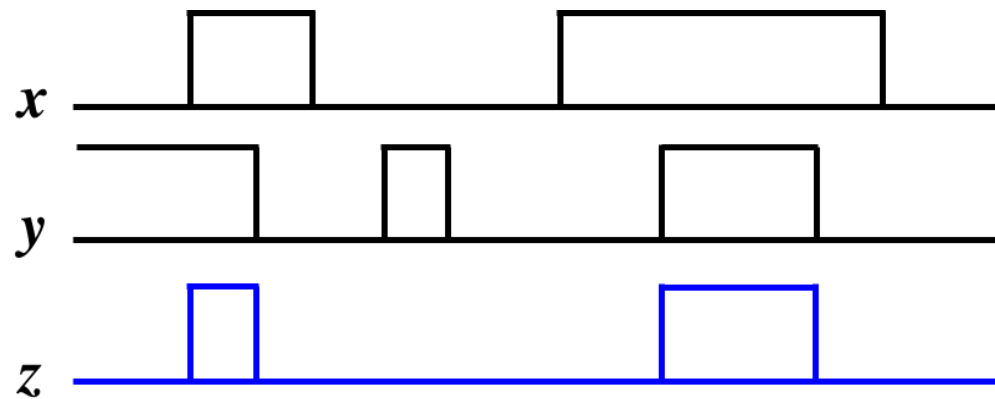


- Phương trình logic: $z = x.y$

- Bảng trạng thái:

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

Dạng sóng:

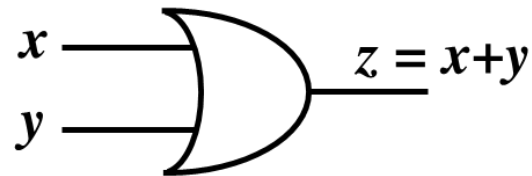


$$y = x_1 x_2 \dots x_n = \begin{cases} 1 & \text{nếu } x_1 = x_2 = \dots = x_n = 1 \\ 0 & \text{nếu } \exists x_i = 0 \ (i = 1, \dots, n) \end{cases}$$

Cổng logic (tt)

- Cổng OR

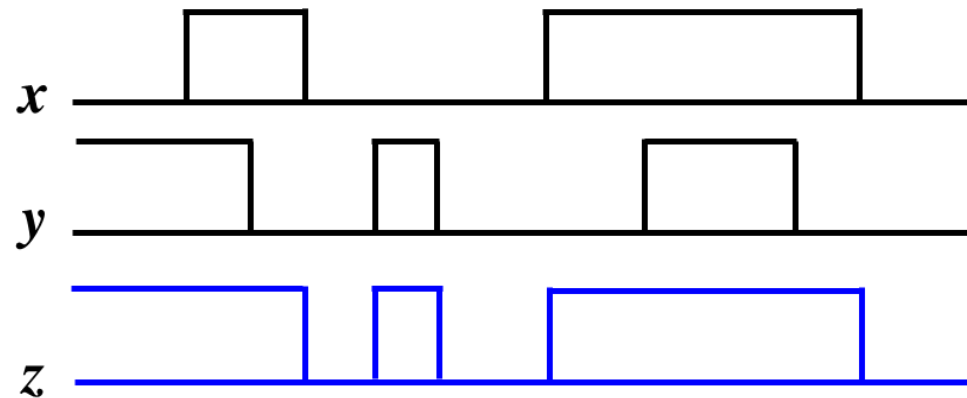
- Ký hiệu (ANSI):



- Phương trình logic: $z = x + y$
 - Bảng trạng thái:

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

Dạng sóng:

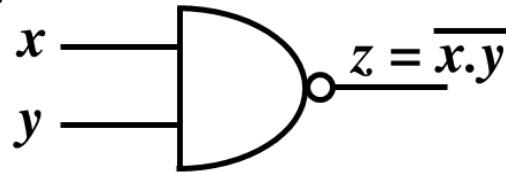


$$y = x_1 + x_2 + \dots + x_n = \begin{cases} 0 & \text{nếu } x_1 = x_2 = \dots = x_n = 0 \\ 1 & \text{nếu } \exists x_i = 1 \ (i = 1, \dots, n) \end{cases}$$

Cổng logic (tt)

- Cổng NAND

- Ký hiệu (ANSI):

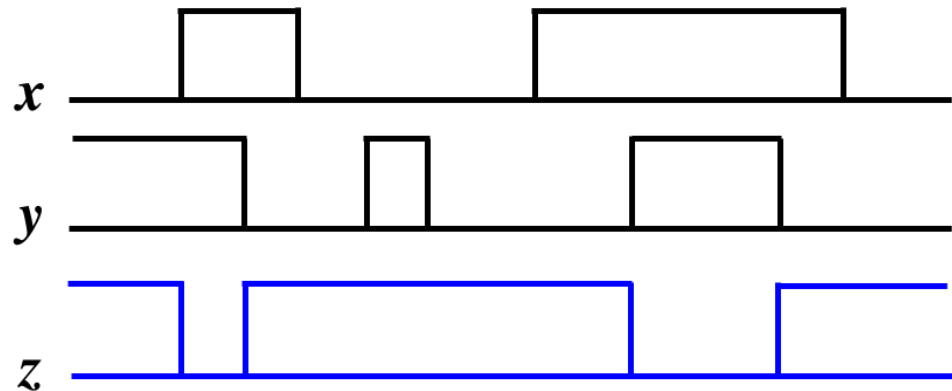


- Phương trình logic: $z = \overline{x \cdot y}$

- Bảng trạng thái:

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

Dạng sóng:

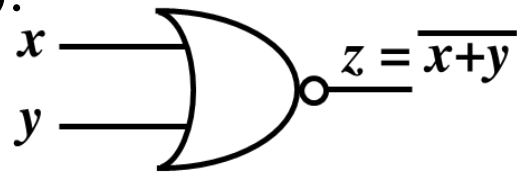


$$y = \overline{x_1 x_2 \dots x_n} = \begin{cases} 0 & \text{nếu } x_1 = x_2 = \dots = x_n = 1 \\ 1 & \text{nếu } \exists x_i = 0 \ (i = 1, \dots, n) \end{cases}$$

Cổng logic (tt)

- Cổng NOR

- Ký hiệu (ANSI):

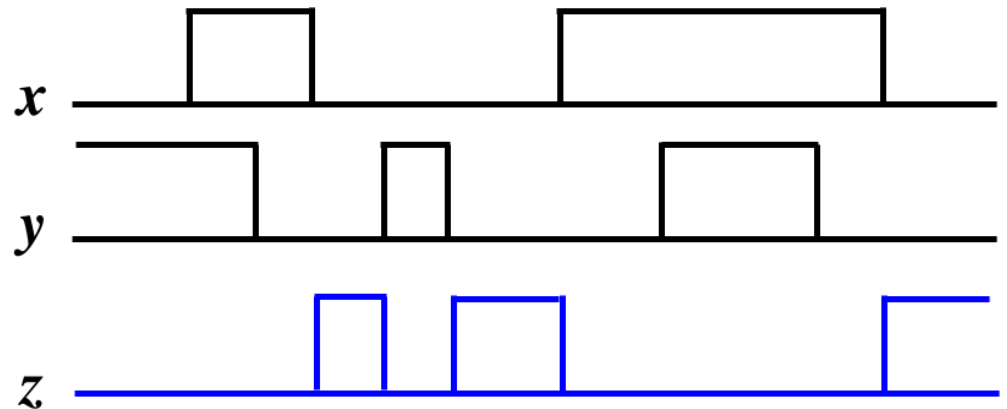


- Phương trình logic: $z = \overline{x + y}$

- Bảng trạng thái:

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

Dạng sóng:

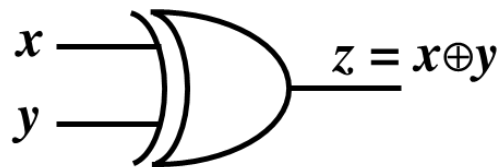


$$y = \overline{x_1 + x_2 + \dots + x_n} = \begin{cases} 1 & \text{nếu } x_1 = x_2 = \dots = x_n = 0 \\ 0 & \text{nếu } \exists x_i = 1 \ (i = 1, \dots, n) \end{cases}$$

Cổng logic (tt)

- Cổng XOR (Exclusive_OR)

- Ký hiệu:

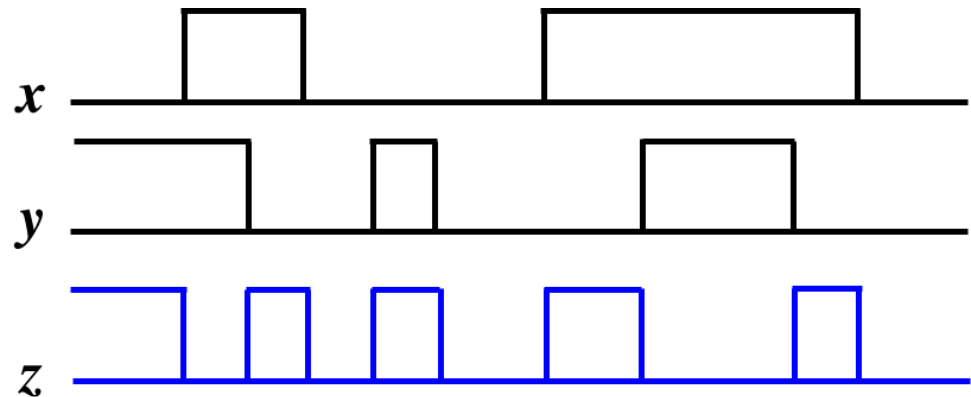


- Phương trình logic: $z = x \oplus y = \bar{x}y + x\bar{y}$

- Bảng trạng thái:

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

Dạng sóng:



$$y = x_1 \oplus x_2 \oplus \dots \oplus x_n = \begin{cases} 1 & \text{nếu số đầu vào bằng 1 là số lẻ} \\ 0 & \text{nếu số đầu vào bằng 1 là số chẵn} \end{cases}$$

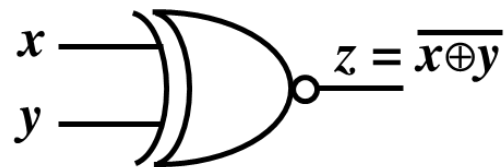
Cổng logic (tt)

- Các tính chất của phép XOR
 - $x \oplus y = y \oplus x$
 - $x \cdot (y \oplus z) = (x \cdot y) \oplus (x \cdot z)$
 - $x \oplus 0 = x$
 - $x \oplus 1 = \bar{x}$
 - $x \oplus x = 0$
 - $x \oplus \bar{x} = 1$

Cổng logic (tt)

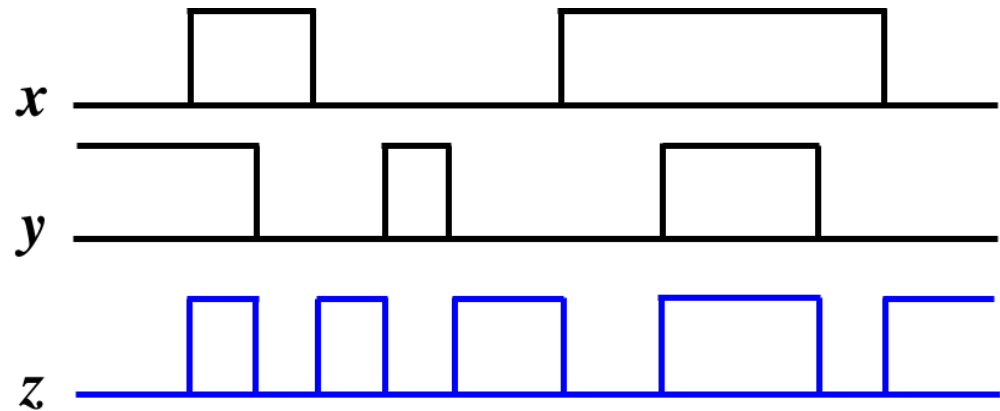
- Cổng XNOR (Exclusive_NOR)

- Ký hiệu:



- Phương trình logic: $z = \overline{x \oplus y} = (\bar{x} + y)(x + \bar{y}) = \bar{x}\bar{y} + xy$
 - Bảng trạng thái: Dạng sóng:

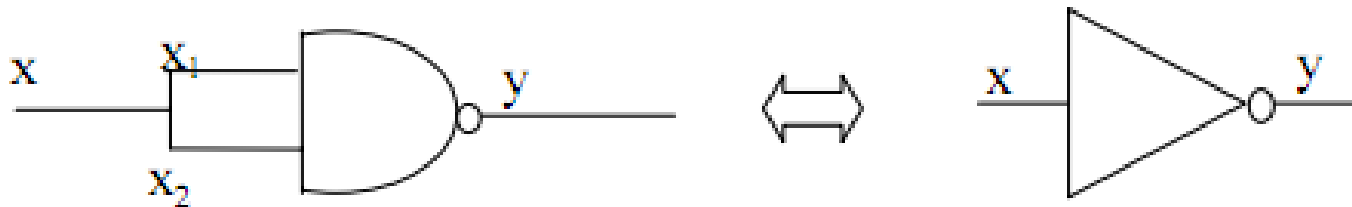
x	y	z
0	0	1
0	1	0
1	0	0
1	1	1



$$y = \overline{x_1 \oplus x_2 \oplus \dots \oplus x_n} = \begin{cases} 0 & \text{nếu số đầu vào bằng 1 là số lẻ} \\ 1 & \text{nếu số đầu vào bằng 1 là số chẵn} \end{cases}$$

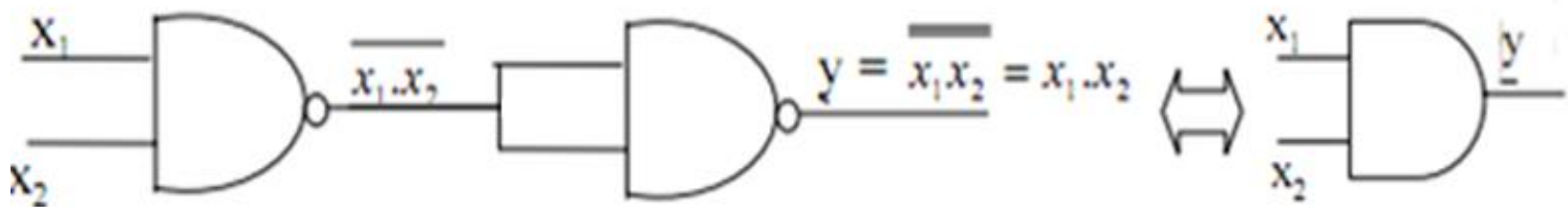
Chuyển đổi giữa các cổng logic

- Sử dụng cổng NAND tạo các cổng logic khác
 - Dùng cổng NAND tạo cổng NOT:



$$y = \overline{x \cdot x} = \bar{x}$$

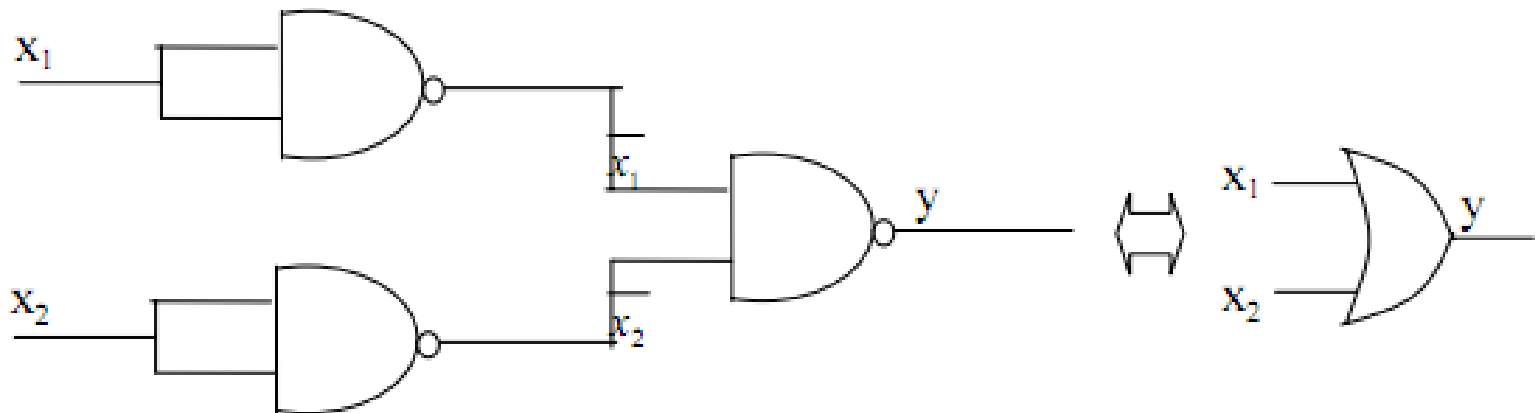
- Dùng cổng NAND tạo cổng AND:



$$y = \overline{\overline{x_1 \cdot x_2}} = x_1 x_2$$

Chuyển đổi giữa các cổng logic (tt)

- Sử dụng cổng NAND tạo các cổng logic khác
 - Dùng cổng NAND tạo cổng OR:



$$y = \overline{\overline{x_1} \cdot \overline{x_2}} = \overline{\overline{x_1 + x_2}} = x_1 + x_2$$

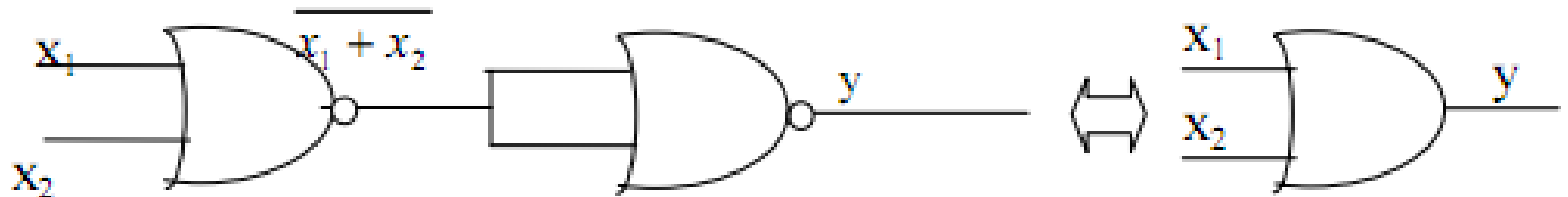
Chuyển đổi giữa các cổng logic (tt)

- Sử dụng cổng NOR tạo các cổng logic khác
 - Dùng cổng NOR tạo cổng NOT:



$$y = \overline{x + x} = \bar{x}$$

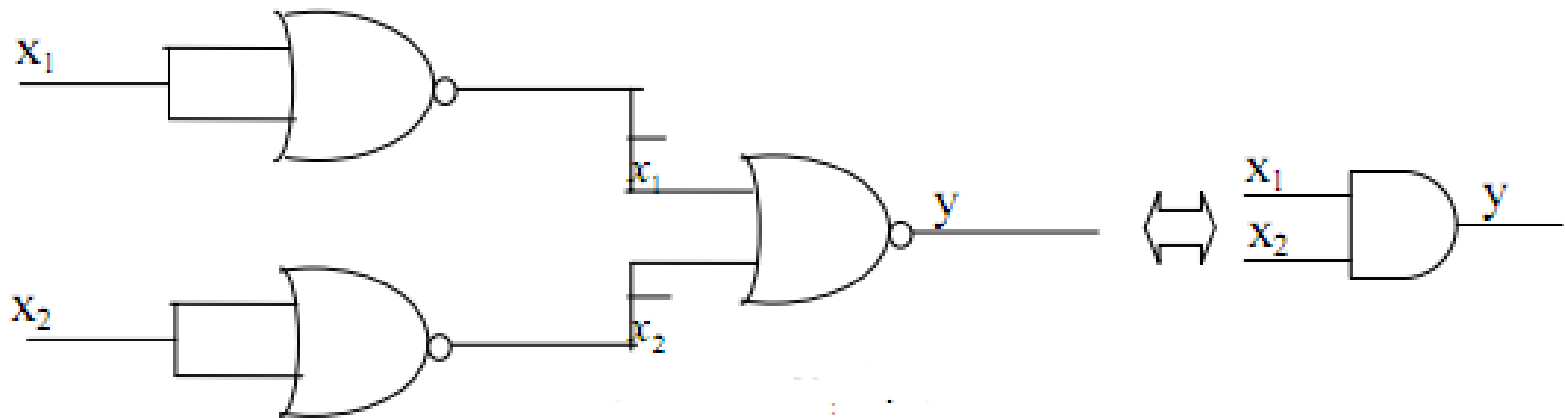
- Dùng cổng NOR tạo cổng OR:



$$y = \overline{\overline{x_1 + x_2}} = x_1 + x_2$$

Chuyển đổi giữa các cổng logic (tt)

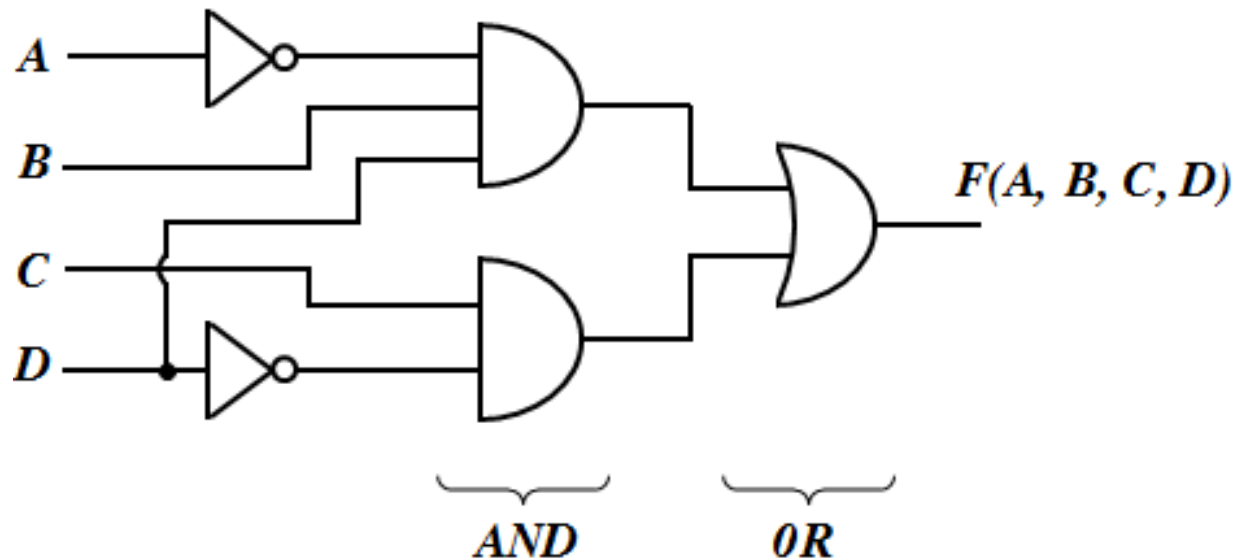
- Sử dụng cổng NOR tạo các cổng logic khác
 - Dùng cổng NOR tạo cổng AND:



$$y = \overline{\overline{x_1} + \overline{x_2}} = \overline{\overline{x_1} \cdot \overline{x_2}} = x_1 \cdot x_2$$

Thực hiện hàm Boole bằng cổng logic

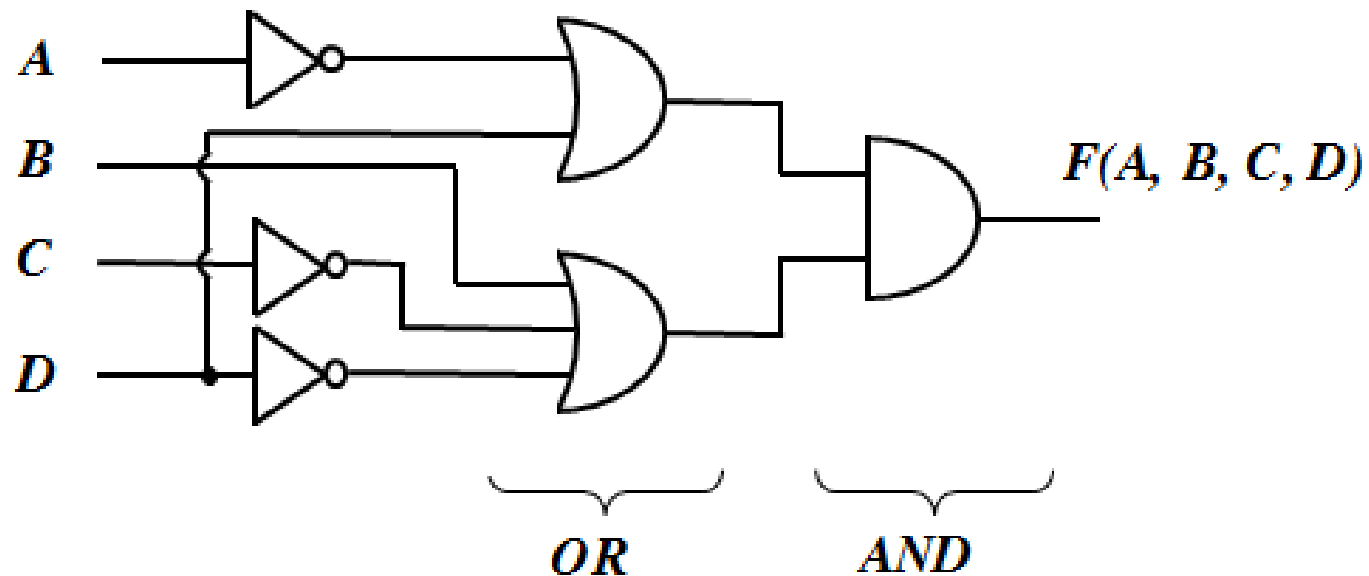
- Cấu trúc cổng AND – OR:
 - Thực hiện hàm Boole biểu diễn theo dạng chính tắc 1 (tổng các tích)
 - Ví dụ:



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

Thực hiện hàm Boole bằng cổng logic (tt)

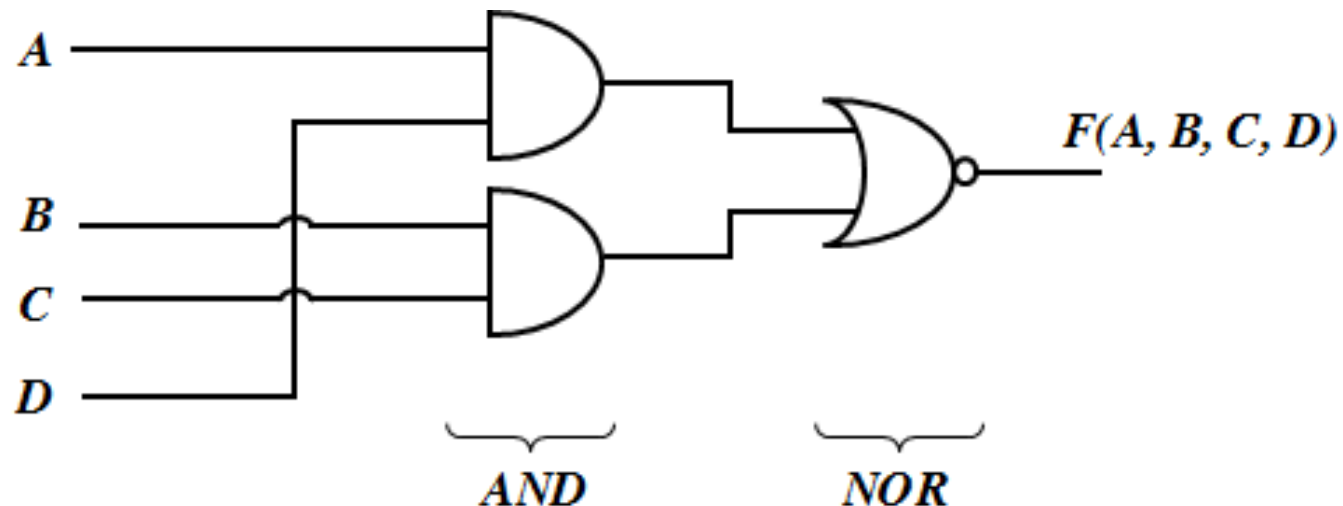
- Cấu trúc cổng OR – AND:
 - Thực hiện hàm Boole biểu diễn theo dạng chính tắc 2 (tích các tổng)
 - Ví dụ:



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

Thực hiện hàm Boole bằng cổng logic (tt)

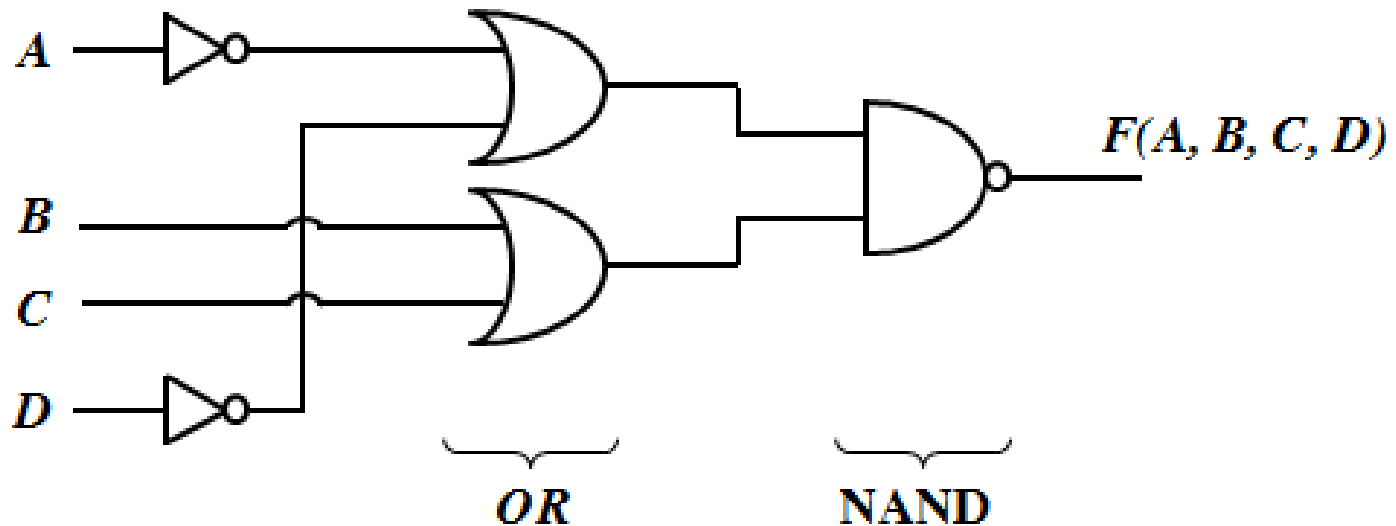
- Cấu trúc cổng AND – OR – INVERTER (AOI):
 - Thực hiện hàm Boole biểu diễn theo dạng bù của tổng các tích
 - Ví dụ:



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

Thực hiện hàm Boole bằng cổng logic (tt)

- Cấu trúc cổng OR – AND – INVERTER (OAI):
 - Thực hiện hàm Boole biểu diễn theo dạng bù của tích các tổng
 - Ví dụ:



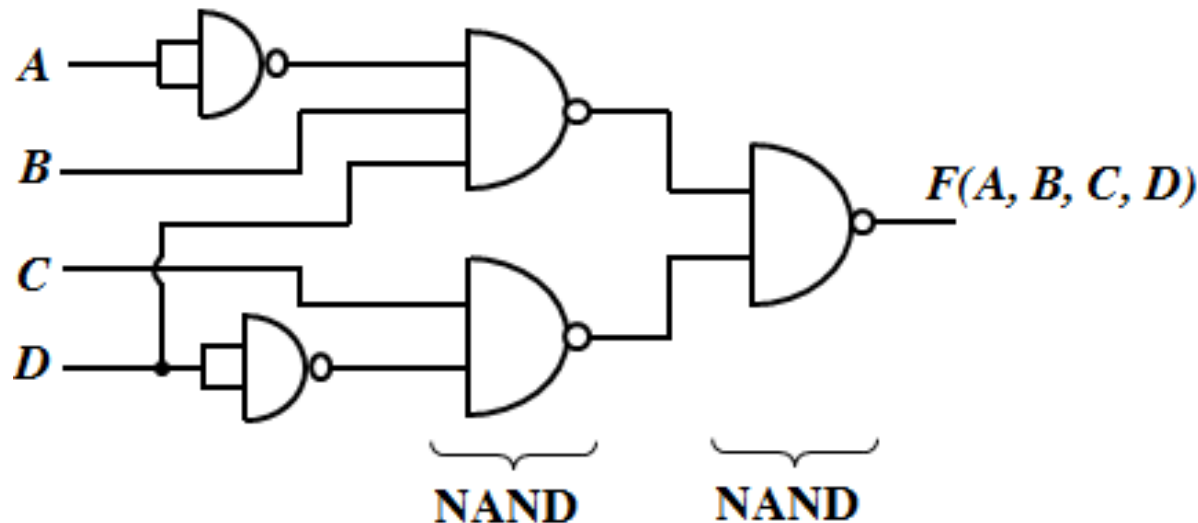
$$F(A, B, C, D) = \overline{(\bar{A} + \bar{D})(B + C)}$$

Thực hiện hàm Boole bằng cổng logic (tt)

- Cấu trúc toàn cổng NAND:

- Thực hiện hàm Boole có dạng bù của một số hạng tích
- Dùng định lý De-Morgan để biến đổi số hạng tổng thành tích
- Cổng NOT cũng được thực hiện bằng cổng NAND
- Ví dụ:

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

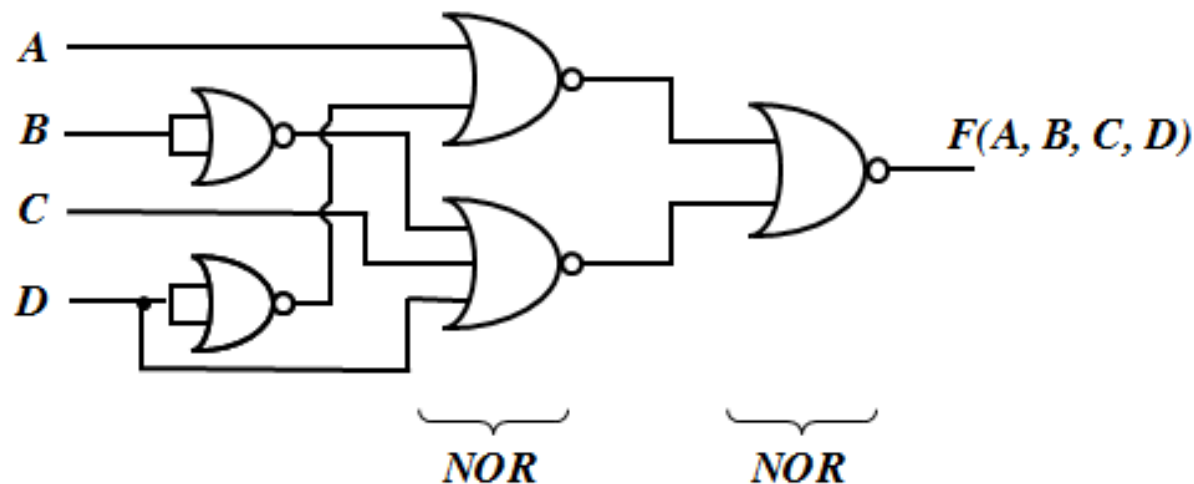


Thực hiện hàm Boole bằng cổng logic (tt)

- Cấu trúc toàn cổng NOR:

- Thực hiện hàm Boole có dạng bù của một số hạng tổng
- Dùng định lý De-Morgan để biến đổi số hạng tích thành tổng
- Cổng NOT cũng được thực hiện bằng cổng NOR
- Ví dụ:

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



Các bước thiết kế mạch logic

1. Lập bảng trạng thái mô tả hoạt động của mạch
2. Lập hàm logic ngõ ra theo ngõ vào và tối thiểu hóa các hàm logic
3. Lập sơ đồ logic

Các bước thiết kế mạch logic (tt)

- Ví dụ: Thiết kế mạch giải mã BCD 8421 sang Gray

1. Lập bảng trạng thái mô tả hoạt động của mạch

Ngõ vào				Ngõ ra			
a	b	c	d	x	y	z	t
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

Các bước thiết kế mạch logic (tt)

- Ví dụ: Thiết kế mạch giải mã BCD 8421 sang Gray
2. Lập hàm logic ngõ ra theo ngõ vào và tối thiểu hóa các hàm logic

x	ab		cd	
	00	01	11	10
00			X	1
01			X	1
11			X	X
10			X	X

$$x = a$$

y	ab		cd	
	00	01	11	10
00		1	X	1
01		1	X	1
11		1	X	X
10		1	X	X

$$y = a + b$$

Các bước thiết kế mạch logic (tt)

- Ví dụ: Thiết kế mạch giải mã BCD 8421 sang Gray
2. Lập hàm logic ngõ ra theo ngõ vào và tối thiểu hóa các hàm logic

$\begin{matrix} z \\ \swarrow \\ ab \\ \searrow \\ cd \end{matrix}$		ab			
		00	01	11	10
00			1	X	
01			1	X	
11		1		X	X
10		1		X	X

$$z = b\bar{c} + \bar{b}c = b \oplus c$$

$\begin{matrix} t \\ \swarrow \\ ab \\ \searrow \\ cd \end{matrix}$		ab			
		00	01	11	10
00				X	
01		1	1	X	1
11				X	X
10		1	1	X	X

$$t = \bar{c}d + c\bar{d} = c \oplus d$$

Các bước thiết kế mạch logic (tt)

- Ví dụ: Thiết kế mạch giải mã BCD 8421 sang Gray
3. Lập sơ đồ logic

