

1. MẠCH LOGIC TỔ HỢP

1.1 CƠ SỞ LOGIC CỦA KỸ THUẬT SỐ.

1.2 PHÂN TÍCH MẠCH TỔ HỢP.

1.3 THIẾT KẾ MẠCH TỔ HỢP.

1.4 MỘT SỐ MẠCH TỔ HỢP THƯỜNG GẶP.

1.5 CÁC VI MẠCH TỔ HỢP VÀ LƯU Ý KHI SỬ DỤNG.

1.1 CƠ SỞ LOGIC CỦA KTS

1.1.1 BIẾN LOGIC VÀ HÀM LOGIC

- Biến logic: $x \in B = \{0;1\}$
- Tổ hợp biến logic: $X = x_1, x_2, \dots, x_n \in B^n$
- Hàm logic: $f(x_1, x_2, \dots, x_n) \in B = \{0;1\}$
- Bảng chân lý:

Ví dụ: Bảng chân lý của hàm logic

Tổ hợp biến	x_1	x_2	x_3	f_1	f_2
0	0	0	0	1	0
1	0	0	1	1	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	0
5	1	0	1	0	0
6	1	1	0	0	1
7	1	1	1	0	1

Tập hợp các giá trị của tổ hợp biến logic

- $B^1 = B = \{0;1\}$ *Số phần tử* $= 2^1 = 2$
- $B^2 = \{00;01;10;11\}$ *Số phần tử* $= 2^2 = 4$
- $B^3 = \{000;001;010;011;100;101;110;111\}$
Số phần tử $= 2^3 = 8$
- $B^n = \{0..0;00..01;...;11..1\}$
Số phần tử $= 2^n$

Mỗi phần tử là một tổ hợp các giá trị của n biến nhị phân.

Các hàm logic một biến $f(x)$

x	f_1	f_2	f_3	f_4
0	0	1	0	1
1	0	0	1	1

$f_1 = 0$ Hàm hằng 0

$f_2 = \bar{x}$ Hàm phủ định

$f_3 = x$ Hàm lặp lại

$f_4 = 1$ Hàm hằng 1

Số tổ hợp biến: $2^1 = 2$

Số hàm logic: $2^{2^1} = 4$

Các hàm logic 2 biến $f(x_1, x_0)$

x_1	x_0	f_0	f_1	f_2	...	f_{14}	f_{15}
0	0	0	1	0		0	1
0	1	0	0	1		1	1
1	0	0	0	0		1	1
1	1	0	0	0		1	1

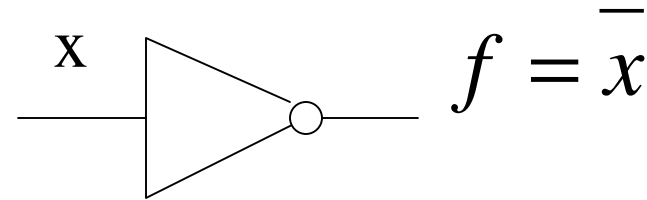
Số tổ hợp biến: $2^2 = 4$ $f_0 = 0$ $f_1 = \overline{x_1 x_0}$

Số hàm logic: $2^{2^2} = 2^4 = 16$ $f_2 = \overline{x_1} x_0$

$f_{14} = \overline{f_1}$ $f_{15} = 1 = \overline{f_0}$

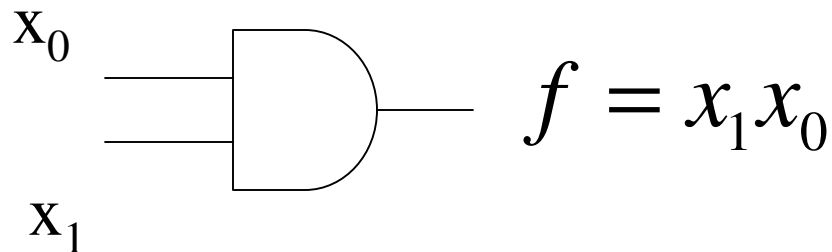
1.1.2 MỘT SỐ PHẦN TỬ LOGIC CƠ BẢN

- Hàm "Phủ định" (NOT)



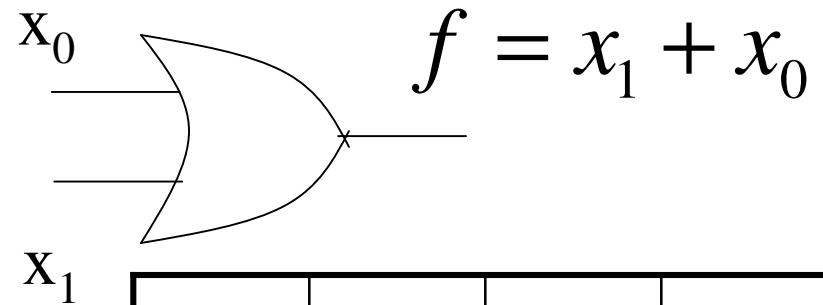
tt	x	f
0	0	1
1	1	0

- Hàm "Và" (AND)



tt	x_1	x_0	f
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

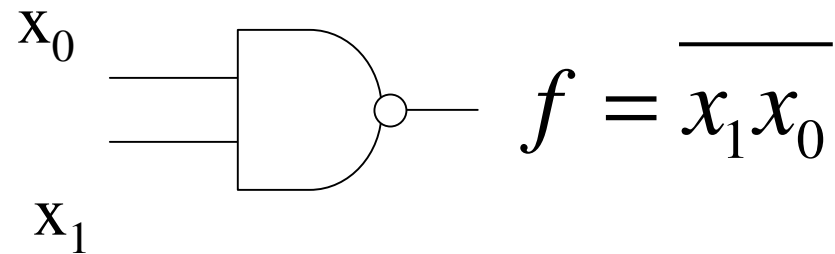
- Hàm "Hoặc" (OR)



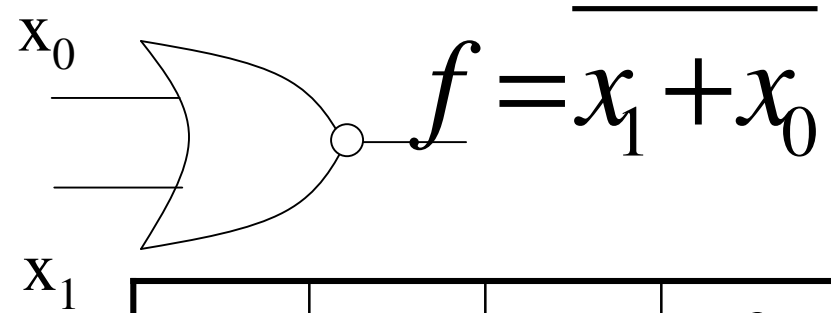
- Hàm "Và-phủ định" (NAND)

tt	x_1	x_0	f
0	0	0	1
1	0	1	1
2	1	0	1
3	1	1	0

tt	x_1	x_0	f
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1



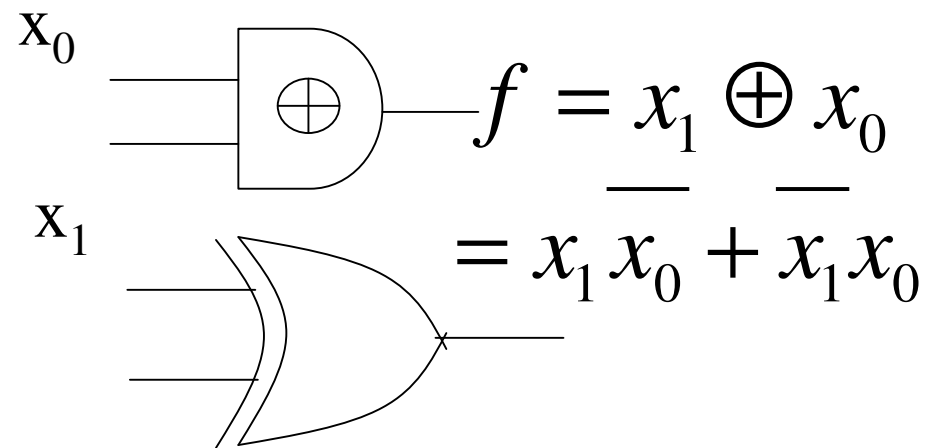
- Hàm "Hoặc-phủ định" (NOR)



- Hàm cộng modul 2 (XOR-Exclusive OR)

tt	x_1	x_0	f
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

tt	x_1	x_0	f
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	0



1.1.3 CÁC TÍNH CHẤT VÀ QUY TẮC CƠ BẢN CỦA ĐẠI SỐ BOOL

- Tính chất giao hoán:
$$x_1 + x_2 = x_2 + x_1$$
$$x_1 x_2 = x_2 x_1$$
- Tính chất kết hợp:
$$x_1 + x_2 + x_3 = x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3$$
$$x_1 x_2 x_3 = x_1 (x_2 x_3) = (x_1 x_2) x_3$$
- Tính chất phân phối:
$$x_1 + x_2 x_3 = (x_1 + x_2)(x_1 + x_3)$$
$$x_1 (x_2 + x_3) = x_1 x_2 + x_1 x_3$$

Một số qui tắc cơ bản

- Qui tắc phủ định (qui tắc De Moorgan):

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

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

- Qui tắc luôn đúng:

$$x + 1 = 1 \quad x + \overline{x} = 1$$

- Qui tắc luôn sai:

$$x \cdot 0 = 0 \quad x \overline{x} = 0$$

- Qui tắc không đổi:

$$x + 0 = x \quad x \cdot 1 = x$$

- Qui tắc phủ định 2 lần:

$$= \\ x = x$$

- Qui tắc lặp:

$$xxx\dots x = x$$

$$x + x + \dots + x = x$$

- Qui tắc dán:

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

$$(x_1 + x_2)(x_1 + \overline{x_2}) = x_1$$

- Qui tắc nuốt (hấp thụ):

$$x_1 + x_1 x_2 = x_1$$

$$x_1 (x_1 + x_2) = x_1$$

- Hệ quả:

$$a + \overline{a}b = a + b$$

$$a(\overline{a} + b) = ab$$

1.1.4 CÁC DẠNG BIỂU THỨC HÀM LOGIC. HỆ HÀM ĐỦ

- **Biểu thức dạng chuẩn tắc tuyến (CTT).**

✓ *Hội cơ bản* là tích logic của một số hữu hạn không lặp các biến logic, mỗi biến có thể không hoặc có phủ định.

$$\overline{x_1} x_2 x_3$$

$$x_1 \cdot x_3 \cdot x_4$$

✓ *Đỉnh* là tổ hợp các giá trị của đủ n biến của hàm logic $f(x_1, x_2, \dots, x_n)$.

$$x_1 x_4$$

✓ *Đỉnh 1* là đỉnh, tại đó, hàm logic có giá trị 1.

✓ *Đỉnh 0* là đỉnh, tại đó, hàm logic có giá trị 0.

✓ Biểu thức dạng chuẩn tắc tuyến (CTT) là tổng của các hội cơ bản.

✓ Biểu thức dạng chuẩn tắc tuyến đủ (CTTĐ) là tổng tất cả các hội cơ bản đủ n biến tại các đỉnh 1. Biến có giá trị 0 đánh dấu phủ định.

✓ Dạng CTT rút gọn là tổng các tích cực tiểu (các tích không thể dán lẫn nhau).

$$f = \overline{x_1} \overline{x_2} \cdot x_3 x_4 + \overline{x_1} x_4 + \overline{x_1}$$

tt	x_1	x_2	f
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	0

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

- **Biểu thức dạng chuẩn tắc hội (CTH).**

- ✓ *Tuyến cơ bản* là tổng logic của một số hữu hạn không lặp các biến logic, mỗi biến có thể không hoặc có phủ định.

- ✓ *Biểu thức dạng chuẩn tắc hội (CTH)* là tích của các tuyến cơ bản.

$$\overline{x_1} + x_2 + \overline{x_3}$$

$$\overline{x_1} + x_4$$

$$f = (\overline{x_1} + \overline{x_2} + \overline{x_3})(\overline{x_1} + x_2)x_4$$

✓ *Biểu thức dạng chuẩn tắc hội đủ (CTHĐ)* là tích tất cả các tuyến cơ bản đủ n biến tại các đỉnh 0. Biến có giá trị 1 đánh dấu phủ định.

✓ *Dạng CTH rút gọn* là tích các tuyến cực tiểu (các tuyến không thể dán lẫn nhau).

tt	x_1	x_2	x_3	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

$$\begin{aligned}
 f &= (x_1 + x_2 + x_3)(\overline{x_1} + \overline{x_2} + \overline{x_3})(x_1 + x_2 + x_3)(\overline{x_1} + \overline{x_2} + \overline{x_3}) \\
 &= (x_1 + x_2)(\overline{x_1} + \overline{x_2})
 \end{aligned}$$

- **Hệ hàm đủ.**

Hệ hàm đủ là một bộ các hàm logic cơ bản mà nhờ chúng có thể viết bất kỳ các hàm logic phức tạp nào.

✓ *Các hệ hàm đủ:*

- Hệ gồm các hàm: Và, Hoặc, Phủ định.
- Hệ gồm hàm: Và-P phủ định (hàm Sheffer).
- Hệ gồm hàm: Hoặc-P phủ định (hàm Pirse).

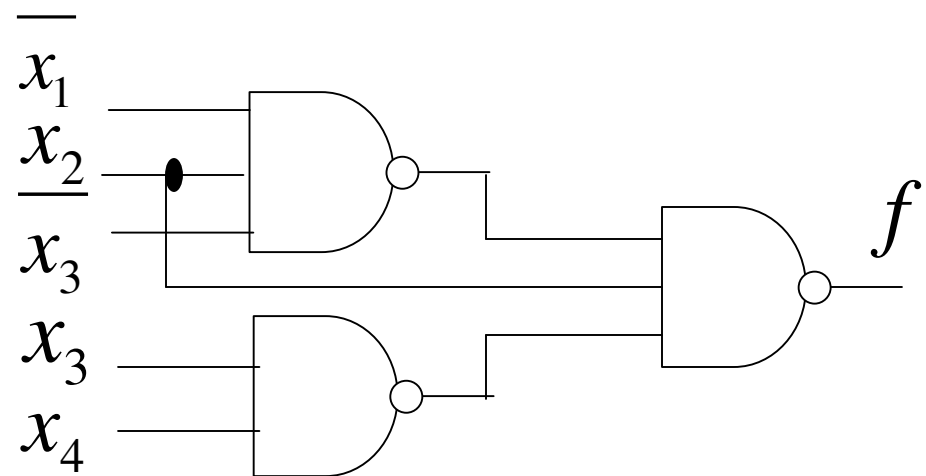
✓ Xây dựng sơ đồ mạch logic trên cơ sở phần tử "Và-phủ định" (NAND).

- Viết hàm logic ở dạng CTT.

- Thực hiện phủ định 2 lần về phải và áp dụng qui tắc De Moorgan biến về phải thành dạng dễ dàng thực hiện bằng phần tử NAND

$$f = \overline{x_1 x_2 x_3} + x_3 x_4 + \overline{x_2} = \overline{\overline{\overline{x_1 x_2 x_3}} + x_3 x_4 + \overline{x_2}}$$

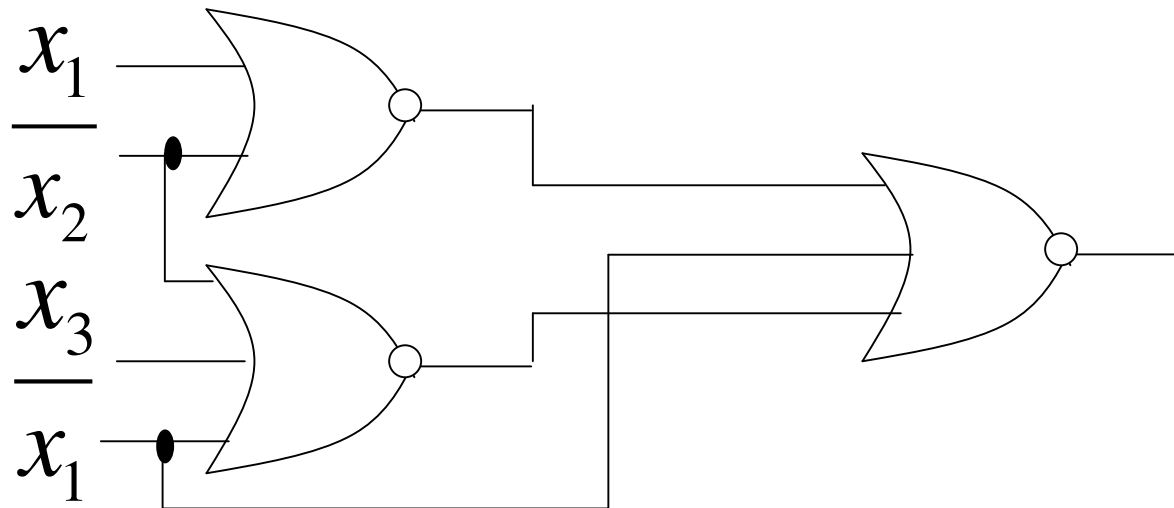
$$= \overline{\overline{\overline{x_1 x_2 x_3}} \cdot \overline{x_3 x_4} \cdot \overline{\overline{x_2}}}$$



- ✓ Xây dựng sơ đồ mạch logic trên cơ sở phần tử "Hoặc-phủ định" (NOR).
 - Viết hàm logic ở dạng CTH.
 - Thực hiện phủ định 2 lần vế phải và áp dụng qui tắc De Moorgan biến vế phải thành dạng dễ dàng thực hiện bằng phần tử NOR.

$$f = (x_1 + \overline{x_2})(\overline{x_1} + \overline{x_2} + \overline{x_3})x_1 = \overline{\overline{(x_1 + \overline{x_2})(\overline{x_1} + \overline{x_2} + \overline{x_3})x_1}}$$

$$= \overline{\overline{(x_1 + \overline{x_2})} + \overline{\overline{(x_1 + \overline{x_2} + x_3)} + \overline{x_1}}}$$



1.1.5 CÁC PHƯƠNG PHÁP TỐI THIỂU HÓA HÀM LOGIC

- **Phương pháp Quine.** Thực hiện 2 bước.

Bước 1. Chuyển hàm logic từ dạng CTTĐ (bao gồm tất cả các đỉnh 1 và đỉnh không xác định) sang dạng CTT rút gọn:

- ✓ Mỗi tích được lần lượt so sánh đôi một với từng tích còn lại. Mỗi lần so sánh mà thấy có thể dán được thì thực hiện phép dán. Viết lại hàm logic bao gồm các tích cực tiểu (TCT) n biến (không dán được) và các tích $n-1$ biến vừa nhận được từ các phép dán.

✓ Thực hiện lặp lại phép so sánh và dán như trên với các tích n-1 biến, rồi n-2 biến... cho đến khi được các tích r biến mà không thể dán lẫn nhau được. Kết quả nhận được hàm CTT rút gọn bao gồm các TCT n, n-1, ..., r biến.

VD: Tối thiểu hóa hàm logic 4 biến sau:

$$f = \overline{x_1} \cdot \overline{x_2} x_3 x_4 + \overline{x_1} x_2 \overline{x_3} \cdot x_4 + \overline{x_1} x_2 \overline{x_3} x_4 + \overline{x_1} x_2 x_3 \overline{x_4} + \overline{x_1} x_2 \cdot x_3 x_4 + \overline{x_1} x_2 x_3 x_4 + x_1 \overline{x_2} \overline{x_3} \cdot x_4 + x_1 \overline{x_2} \overline{x_3} x_4$$

Sẽ dán các tích: 1-4; 1-6; 2-3; 2-7; 3-4; 3-8; 5-6; 5-8; 7-8; không có TCT 4 biến:

$$f = \overline{x_1}x_3x_4 + \overline{x_2}x_3x_4 + \overline{x_1}x_2\overline{x_3} + \overline{x_2}x_3\overline{x_4} + \overline{x_1}x_2x_4 +$$

$$+ \overline{x_2}x_3x_4 + \overline{x_1}x_2x_4 + \overline{x_1}x_3x_4 + \overline{x_1}x_2x_3$$

Tại đây, dán các tích: 3-9; 4-6; Các TCT 3 biến: 1; 2; 5; 7; 8; Hàm CTT rút gọn nhận được:

$$f = \overline{x_1}x_3x_4 + \overline{x_2}x_3x_4 + \overline{x_1}x_2x_4 + \overline{x_1}x_2x_4 + \overline{x_1}x_3x_4 + \overline{x_2}x_3$$

Bước 2. Tối thiểu hóa hàm CTT rút gọn nhận được bằng cách loại bỏ các TCT thừa (là các TCT mà việc loại bỏ không làm thay đổi giá trị hàm logic):

- ✓ Lập bảng với các đầu hàng là các TCT, các đầu cột là các đỉnh 1. Trên từng dòng, ứng với đỉnh nào mà tích cực tiêu nhận giá trị 1 thì đánh dấu x.
- ✓ Chọn một bộ tối thiểu các TCT mà phủ tất cả các đỉnh 1. Bắt đầu là chọn các TCT quan trọng (chỉ chúng phủ những đỉnh nhất định). Trong số các TCT không quan trọng (cùng phủ những đỉnh nhất định), chọn một số ít nhất các TCT mà phủ hết các đỉnh 1 còn lại.

Các đỉnh 1

TCT	$\overline{\overline{x_1 x_2 x_3 x_4}}$	$\overline{x_1 x_2} \overline{x_3 x_4}$	$\overline{x_1 x_2} \overline{x_3 x_4}$	$\overline{x_1 x_2 x_3 x_4}$	$\overline{\overline{x_1 x_2 x_3 x_4}}$	$\overline{x_1 x_2 x_3 x_4}$	$\overline{x_1 x_2 x_3 x_4}$	$\overline{x_1 x_2 x_3 x_4}$
$\overline{x_1 x_3 x_4}$	X			X				
$\overline{x_2 x_3 x_4}$	X					X		
$\overline{x_1 x_2 x_4}$			X	X				
$\overline{x_1 x_2 x_4}$					X	X		
$\overline{x_1 x_3 x_4}$					X			X
$\overline{x_2 x_3}$		X	X				X	X

Nhận xét: TCT quan trọng:

$$\overline{x_2 x_3}$$

2 tích phủ nốt 4 đỉnh còn lại:

$$\overline{x_1 x_3 x_4} \quad \overline{x_1 x_2 x_4}$$

Hàm tối thiểu hóa:

$$f = \overline{x_2 x_3} + \overline{x_1 x_3 x_4} + \overline{x_1 x_2 x_4}$$

- **Phương pháp Quine-Mc Cluskey.** Thực hiện hai bước như pp Quine, chỉ khác ở bước 1: nhằm giảm số lần so sánh từng cặp các tích, trước khi so sánh, chia các tích thành từng nhóm với cùng số biến không có dấu phủ định. Việc so sánh đôi một các tích chỉ cần thực hiện giữa hai nhóm cạnh nhau.
- **Phương pháp bảng Karnaugh.**
 - ✓ *Lập bảng Karnaugh* cho hàm logic theo nguyên tắc: các bộ biến được phân bố theo hàng và theo cột sao cho mỗi ô là một đỉnh, hai đỉnh cạnh nhau theo hàng cũng như theo cột chỉ khác nhau bởi giá trị của 1 biến

Bảng Karnaugh cho hàm 2 biến $f(x_1, x_2)$

$x_1 \backslash x_2$	0	1
0	0	1
1	2	3

Bảng Karnaugh cho hàm 3 biến $f(x_1, x_2, x_3)$

$x_1 \backslash x_2 x_3$	00	01	11	10
0	0	1	3	2
1	4	5	7	6

Bảng Karnaugh cho hàm 4 biến $f(x_1, x_2, x_3, x_4)$

$x_1x_2 \backslash x_3x_4$	00	01	11	10
00	0	1	3	2
01	4	5	7	6
11	12	13	15	14
10	8	9	11	10

Bảng Karnaugh cho hàm 5 biến $f(x_1, x_2, x_3, x_4, x_5)$

		$x_3x_4x_5$						
x_1x_2	000	001	011	010	110	111	101	100
00	0	1	3	2	6	7	5	4
01	8	9	11	10	14	15	13	12
11	24	25	27	26	30	31	29	28
10	16	17	19	18	22	23	21	20

✓ *Trình tự tối thiểu hóa hàm logic dạng CTT.*

- Ghi 1 vào các đỉnh 1, ghi x vào các đỉnh kxđ của hàm logic trong bảng Karnaugh.

- Thực hiện dán các đỉnh 1 với nhau và với các đỉnh kxđ bằng các hình chữ nhật phủ qua. Chỉ các đỉnh 1 và kxđ liền kề nhau (theo hàng, theo cột) mới được dán với nhau. Số đỉnh trong một phép dán phải là lũy thừa của 2: 2, 4, 8, ...

Hàng trên cùng và hàng dưới cùng, cột cạnh trái và cột cạnh phải của bảng Karnaugh được xem là liền kề nhau.

- Trong các ô được dán bằng một hình chữ nhật, biến nào thay đổi giá trị sẽ bị loại khi viết kết quả phép dán. Hàm logic tối thiểu dạng CTT là kết quả của tất cả các phép dán và những đỉnh 1 không dán được.

VD: Tối thiểu hóa hàm logic 4 biến sau:

$$f = \overline{x_1} \overline{x_2} \overline{x_3} \overline{x_4} + \overline{x_1} \overline{x_2} \overline{x_3} x_4 + \overline{x_1} \overline{x_2} x_3 \overline{x_4} + \overline{x_1} \overline{x_2} x_3 x_4 +$$

$$+ \overline{x_1} x_2 \overline{x_3} \overline{x_4} + \overline{x_1} x_2 \overline{x_3} x_4 + \overline{x_1} x_2 x_3 \overline{x_4} + \overline{x_1} x_2 x_3 x_4$$

Lập bảng Karnaugh cho hàm logic, đánh dấu các đỉnh 1, dán bằng các hình chữ nhật, viết kết quả các phép dán:

$x_1x_2 \backslash x_3x_4$	00	01	11	10
00			1	
01	1	1	1	
11	1	1		
10		1	1	

$$f = \overline{x_2} \overline{x_3} + \overline{x_1} x_3 x_4 + x_1 \overline{x_2} x_4$$

✓ *Trình tự tối thiểu hóa hàm logic dạng CTH.*

- Ghi 0 vào các đỉnh 0, ghi x vào các đỉnh kxđ của hàm logic trong bảng Karnaugh.

- Thực hiện dán các đỉnh 0 với nhau và với các đỉnh kxđ bằng các hình chữ nhật phủ qua. Chỉ các đỉnh 0 và kxđ liền kề nhau (theo hàng, theo cột) mới được dán với nhau. Số đỉnh trong một phép dán phải là lũy thừa của 2: 2, 4, 8, ... Hàng trên cùng và hàng dưới cùng, cột cạnh trái và cột cạnh phải của bảng Karnaugh được xem là liền kề nhau.

- Trong các ô được dán bằng một hình chữ nhật, biến nào thay đổi giá trị sẽ bị loại khi viết kết quả phép dán. Hàm logic tối thiểu dạng CTH là kết quả của tất cả các phép dán và những đỉnh 0 không dán được.

VD: Tối thiểu hóa hàm logic sau:

$$f = \prod(0,1,3,5,7,11,16,20,23,29)$$

Các đỉnh kxđ: N=(4,8,15,24,28,31)

Lập bảng Karnaugh cho hàm logic, đánh dấu các đỉnh 0, các đỉnh kxđ, dán bằng các hình chữ nhật, viết kết quả các phép dán:

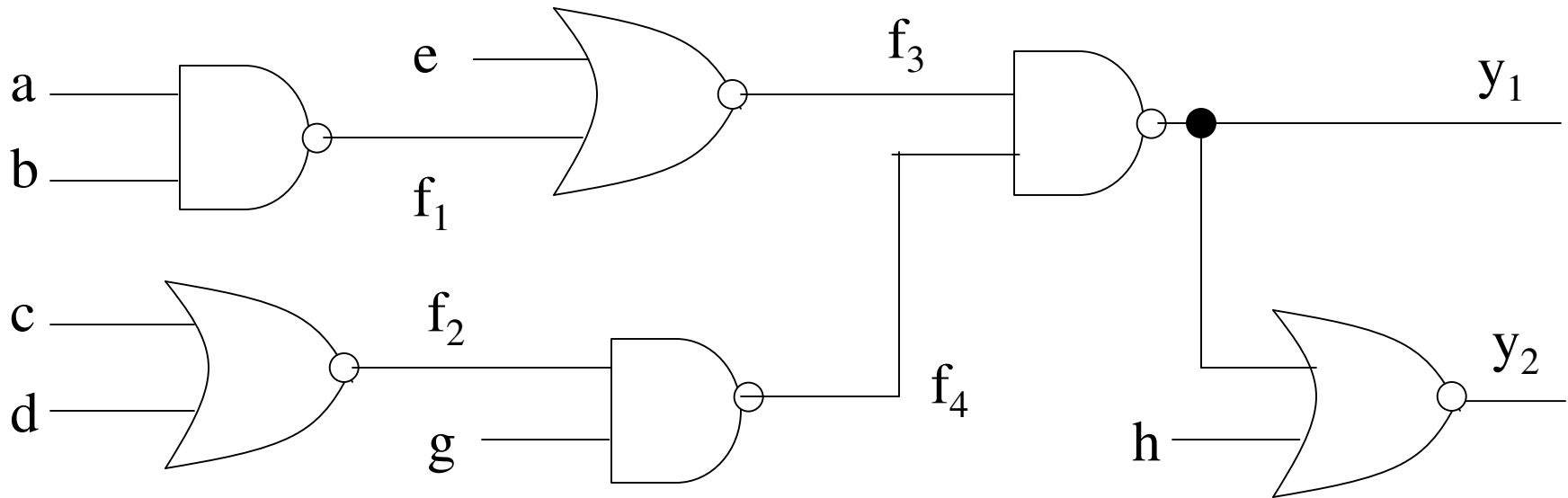
		$x_3x_4x_5$							
x_1x_2		000	001	011	010	110	111	101	100
00		0	0	0			0	0	x
01		x		0			x		
11		x					x	0	x
10		0					0		0

$$f = (x_2 + x_4 + x_5)(x_1 + x_2 + \overline{x_5})(x_1 + \overline{x_4} + \overline{x_5})$$

$$(\overline{x_3} + \overline{x_4} + \overline{x_5})(\overline{x_1} + \overline{x_2} + \overline{x_3} + x_4)$$

1.2 PHÂN TÍCH MẠCH TỔ HỢP.

- **Phân tích mạch tổ hợp** là từ sơ đồ mạch logic tổ hợp cho trước, viết hàm logic các đầu ra phụ thuộc các biến đầu vào.
- **Các bước phân tích một mạch tổ hợp:**
 - ✓ Đặt các biến phụ tại đầu ra của các mạch (phần tử) logic cơ bản;
 - ✓ Viết biểu thức của các biến phụ đó, là hàm của các đầu vào của chúng;
 - ✓ Thay lần lượt biểu thức của các biến phụ vào biểu thức các hàm đầu ra của mạch tổ hợp đã cho.



$$f_1 = \overline{ab} \quad f_2 = \overline{c+d} \quad f_3 = \overline{e+f_1} \quad f_4 = \overline{gf_2}$$

$$\begin{aligned} y_1 &= \overline{f_3 f_4} = \overline{f_3} + \overline{f_4} = (e + f_1) + gf_2 = \\ &= (e + \overline{ab}) + g(\overline{c+d}) = e + \overline{a} + \overline{b} + g\overline{c}\overline{d} \\ y_2 &= \overline{y_1 + h} = \overline{y_1} \cdot \overline{h} = \overline{e.a.b.(g+c+d).h} = \\ &= \overline{e.a.b.h(g+c+d)} \end{aligned}$$

1.3 THIẾT KẾ MẠCH TỔ HỢP.

- **Thiết kế mạch tổ hợp** là xây dựng sơ đồ mạch logic thực hiện chức năng của hàm logic cho trước trên cơ sở những phần tử logic cơ bản.

1.3.1 Các bước thiết kế mạch tổ hợp:

- ✓ Tối thiểu hóa hàm logic đã cho;
- ✓ Biến đổi hàm logic đã tối thiểu hóa về dạng dễ dàng thực hiện bằng các phần tử logic cơ bản cho trước;
- ✓ Vẽ sơ đồ nguyên lý mạch tổ hợp.

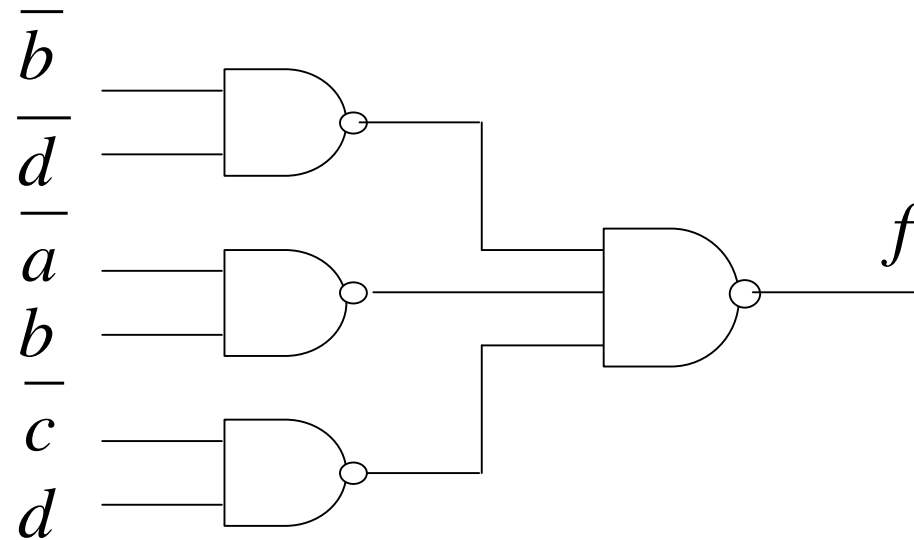
VD: thiết kế mạch logic tổ hợp dùng các phần tử NAND cho hàm sau:

$$f = \Sigma 2,4,5,7,8,13$$

$$N = 0,1,6,9,10,15$$

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

$$f = \overline{b}.d + \overline{a}.b + \overline{c}.d = \overline{\overline{\overline{b}.d.a.b.c.d}}$$



* **Chú ý:** - Nếu tối thiểu hóa bằng PP Quine Mc Cluskey, tùy việc lựa chọn các tích cực tiểu mà ta có các dạng biểu diễn khác nhau của hàm tối thiểu hóa, tuy nhiên, các đỉnh 1 và các đỉnh 0 không thay đổi.

- Nếu tối thiểu hóa bằng PP bảng Karnaugh, tùy cách dán các đỉnh 1 và các đỉnh không xác định, ta có các dạng biểu diễn khác nhau của hàm tối thiểu hóa, tuy nhiên, các đỉnh 1 và các đỉnh 0 không thay đổi.

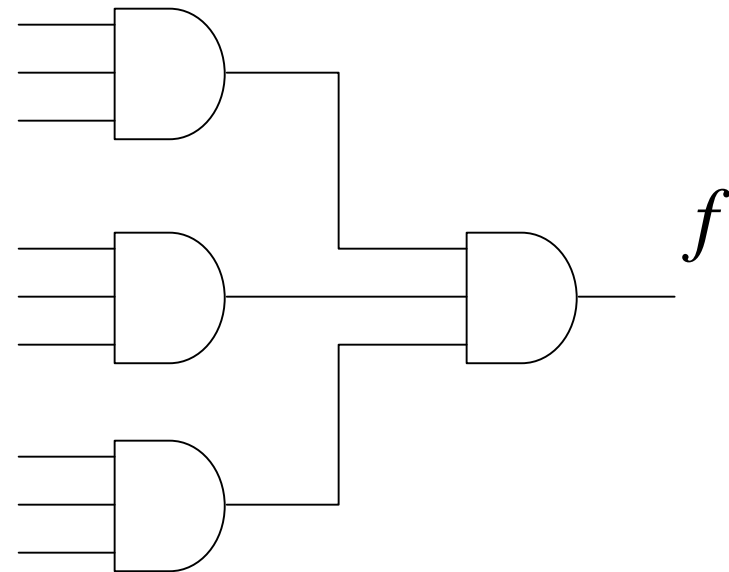
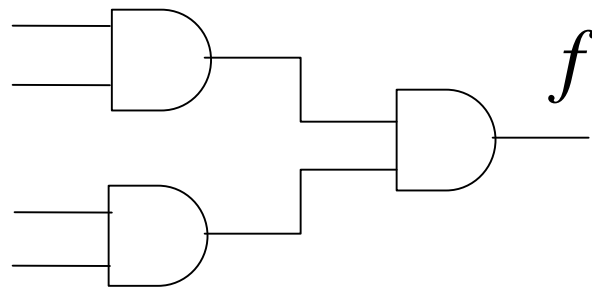
1.3.2 Thiết kế mạch tổ hợp 2 tầng và nhiều tầng.

✓ Tầng một là AND, tầng hai là AND.

Hàm logic là một hội (tích) n biến: $f = x_1 x_2 \dots x_n$

Số đầu vào của một phần tử AND là m;

$n > m$

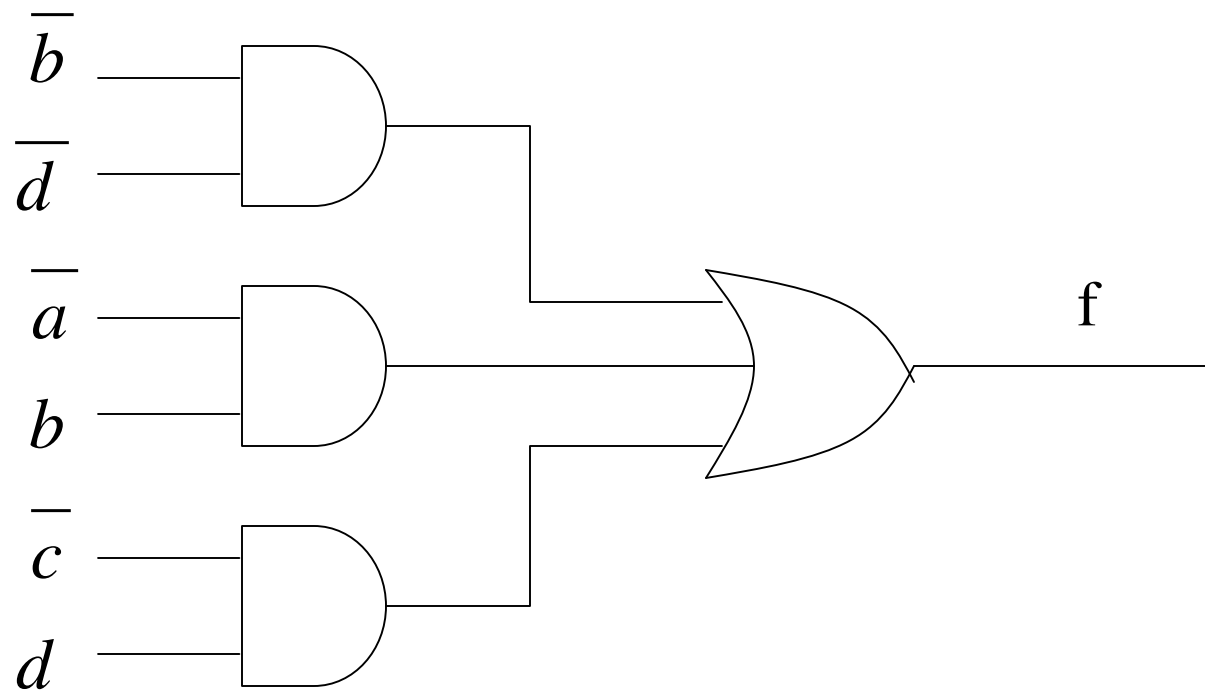


✓ Tầng một là AND, tầng hai là OR.

Hàm logic được viết ở dạng CTT:

VD:

$$f = \bar{b}.\bar{d} + \bar{a}.b + \bar{c}.d$$

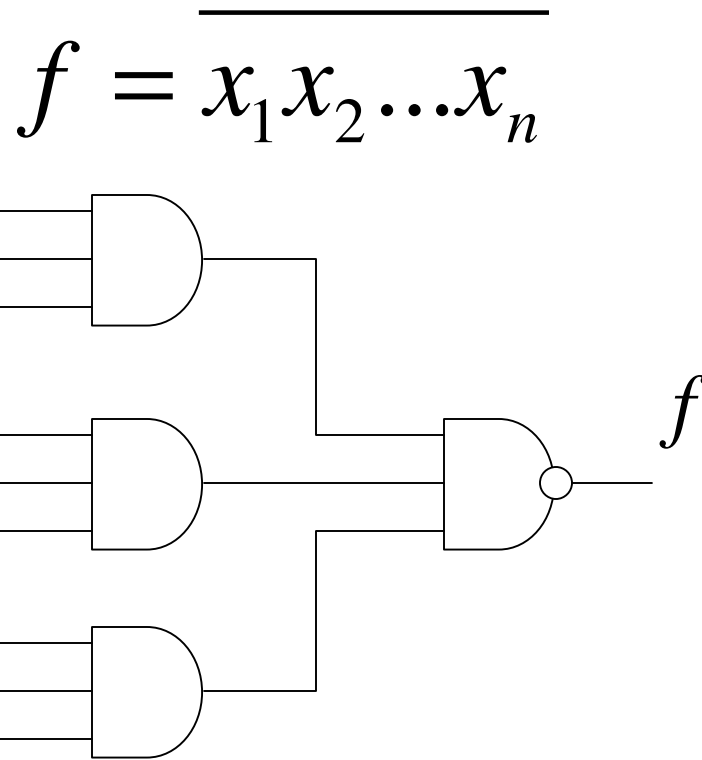


✓ Tầng một là AND, tầng hai là NAND.

Hàm logic là phủ định của một hội n biến.

Số đầu vào của phân tử NAND là m;

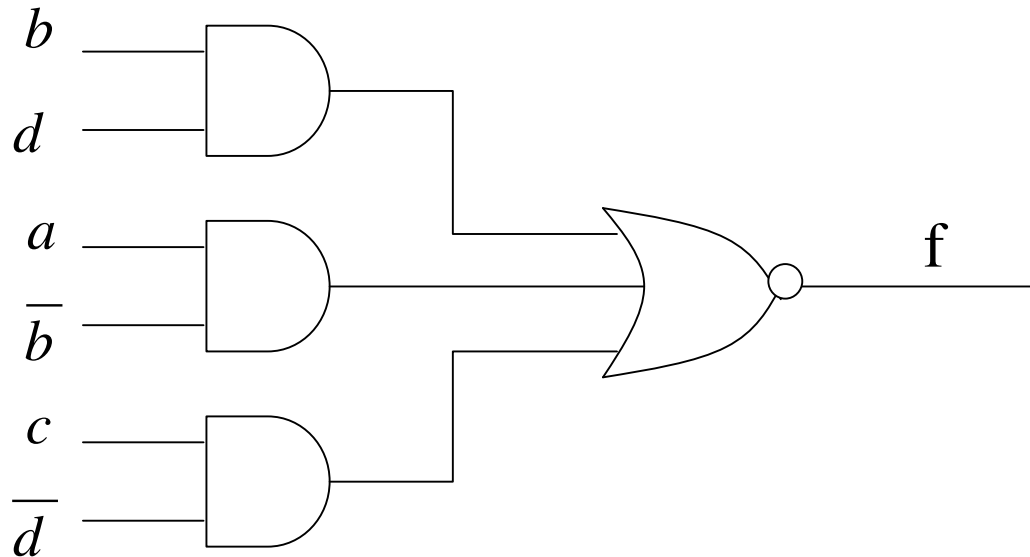
$n > m$



✓ Tầng một là *AND*, tầng hai là *NOR*.

Hàm logic được viết ở dạng CTH, phủ định hai lần và áp dụng qui tắc De Moorgan hai lần:

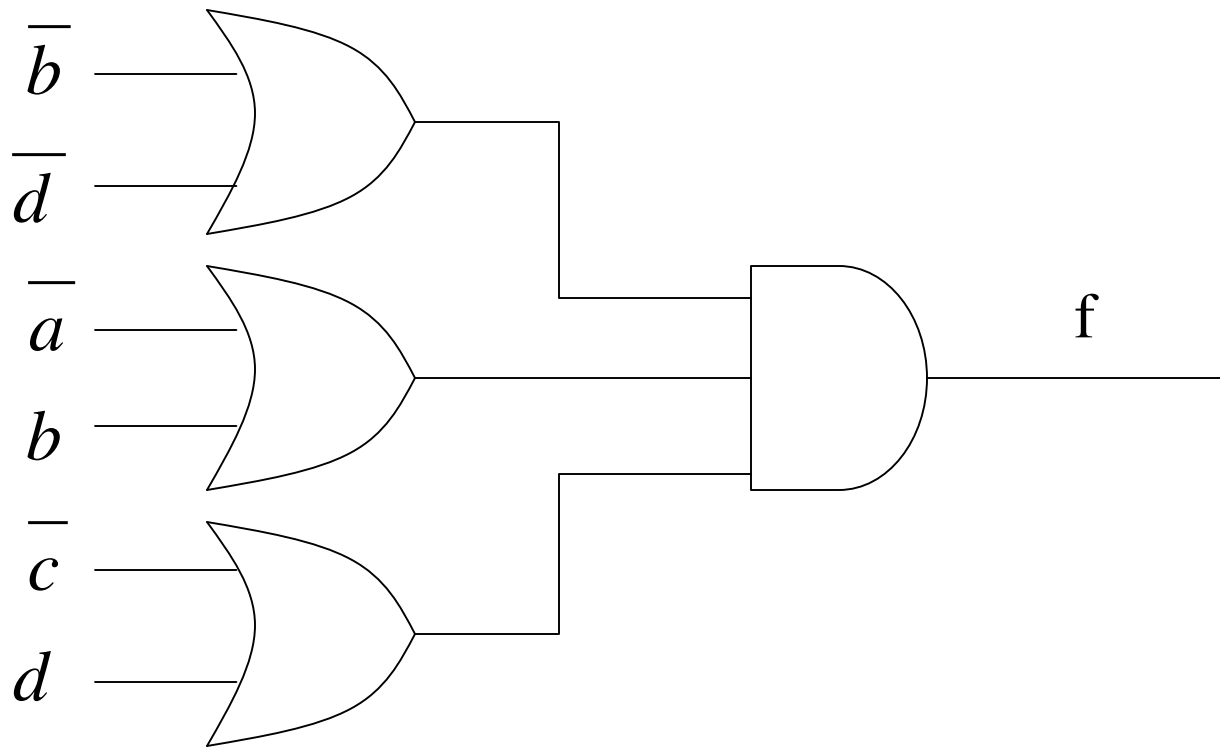
$$f = (\bar{b} + \bar{d})(\bar{a} + b)(\bar{c} + d) = \overline{\overline{(\bar{b} + \bar{d})(\bar{a} + b)(\bar{c} + d)}} \\ = \overline{\overline{(\bar{b} + \bar{d})} + \overline{\overline{(\bar{a} + b)}} + \overline{\overline{(\bar{c} + d)}}} = \overline{bd + ab + cd}$$



✓ Tầng một là OR, tầng hai là AND.

Hàm logic được viết ở dạng CTH:

$$f = (\bar{b} + \bar{d})(\bar{a} + b)(\bar{c} + d)$$

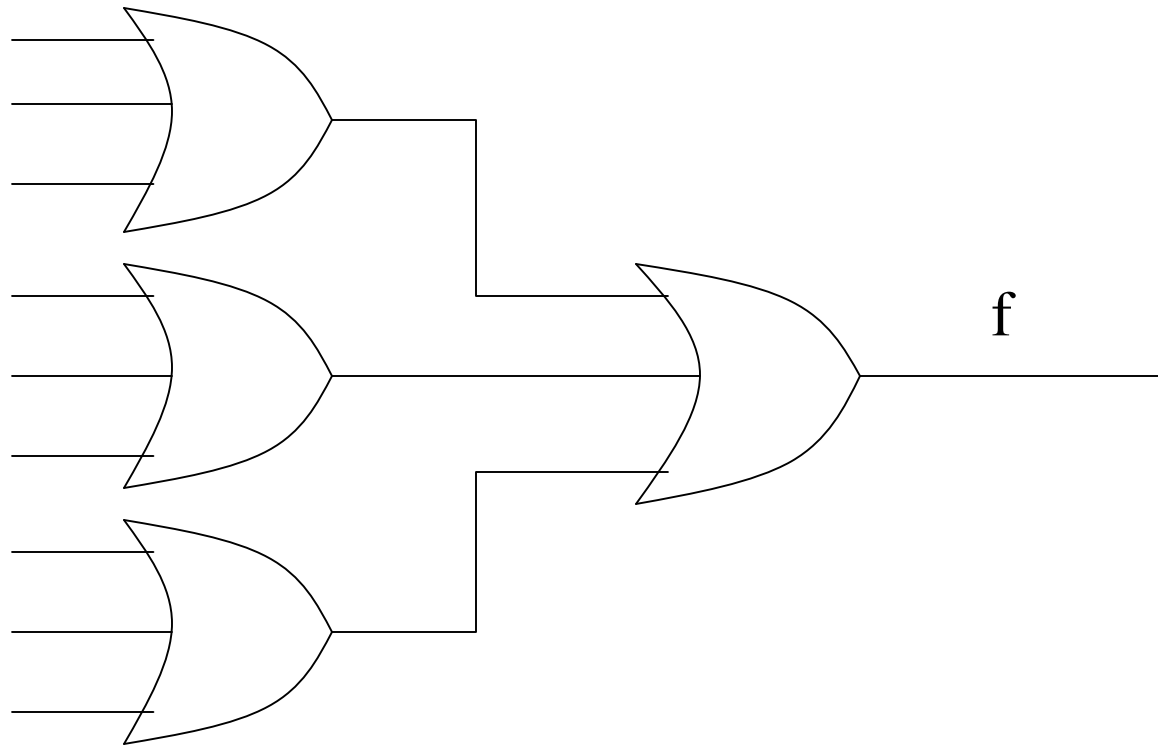


✓ Tầng một là OR, tầng hai là OR.

Hàm logic là một tuyến n biến $f = x_1 + x_2 + \dots + x_n$.

Số đầu vào của phần tử OR là m;

$n > m$

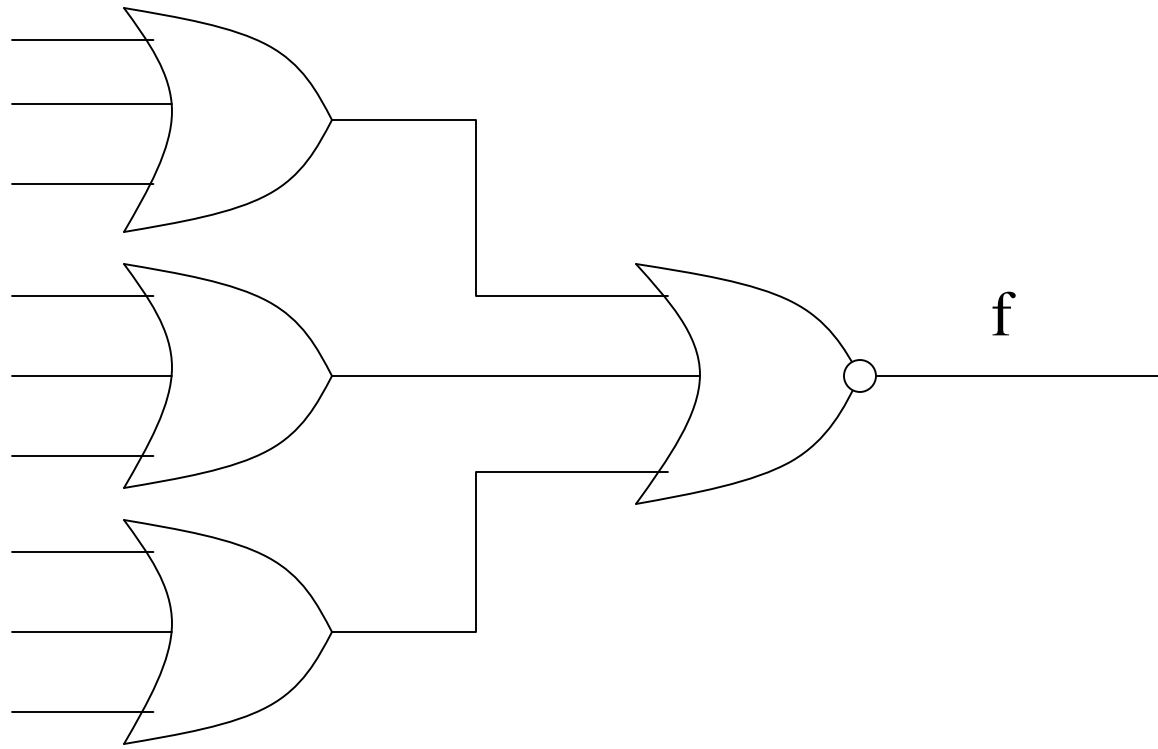


✓ Tầng một là *OR*, tầng hai là *NOR*.

Hàm logic là phủ định của một tuyến n biến.

Số đầu vào của phân tử *NOR* là m ;

$n > m$

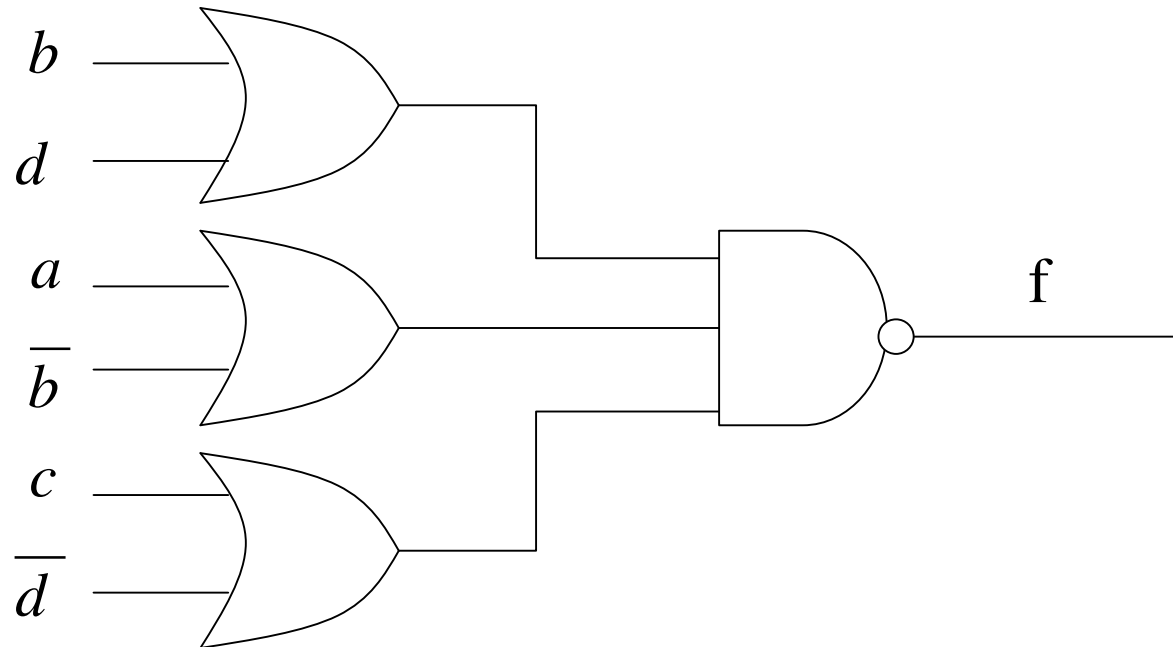


✓ Tầng một là OR, tầng hai là NAND.

Hàm logic được viết ở dạng CTT, phủ định hai lần và áp dụng qui tắc De Moorgan hai lần:

$$f = \overline{\overline{b}}.\overline{\overline{d}} + \overline{\overline{a}}.\overline{\overline{b}} + \overline{\overline{c}}.\overline{\overline{d}} = \overline{\overline{\overline{\overline{b.d}}}} + \overline{\overline{\overline{\overline{a.b}}}} + \overline{\overline{\overline{\overline{c.d}}}}$$

$$= \overline{\overline{\overline{\overline{b.d.a.b.c.d}}}} = \overline{(b + d)(a + \overline{b})(c + \overline{d})}$$

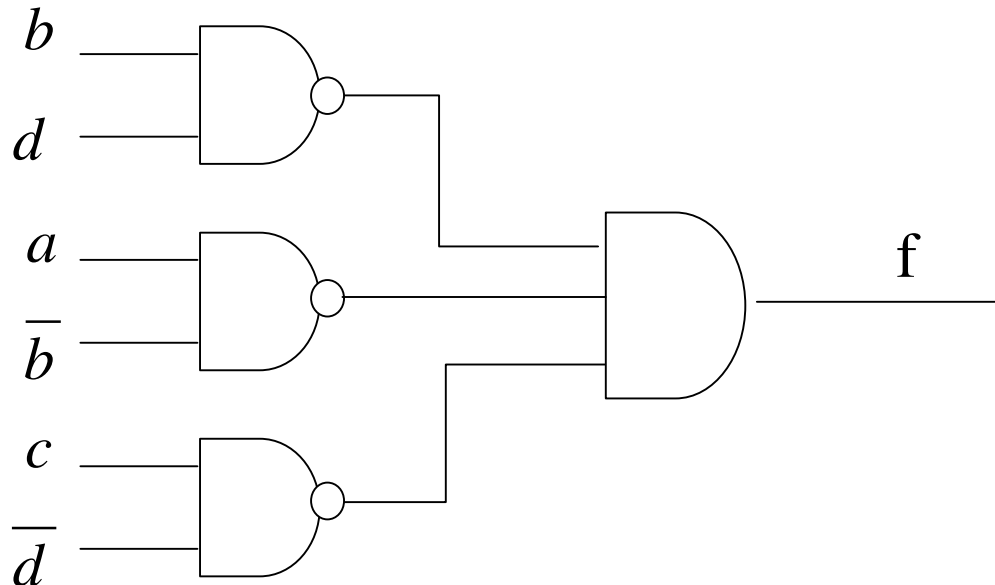


✓ Tầng một là NAND, tầng hai là AND.

Hàm logic được viết ở dạng CTH, phủ định hai lần từng tuyến và áp dụng qui tắc De Moorgan:

$$f = (\bar{b} + \bar{d})(\bar{a} + b)(\bar{c} + d) =$$

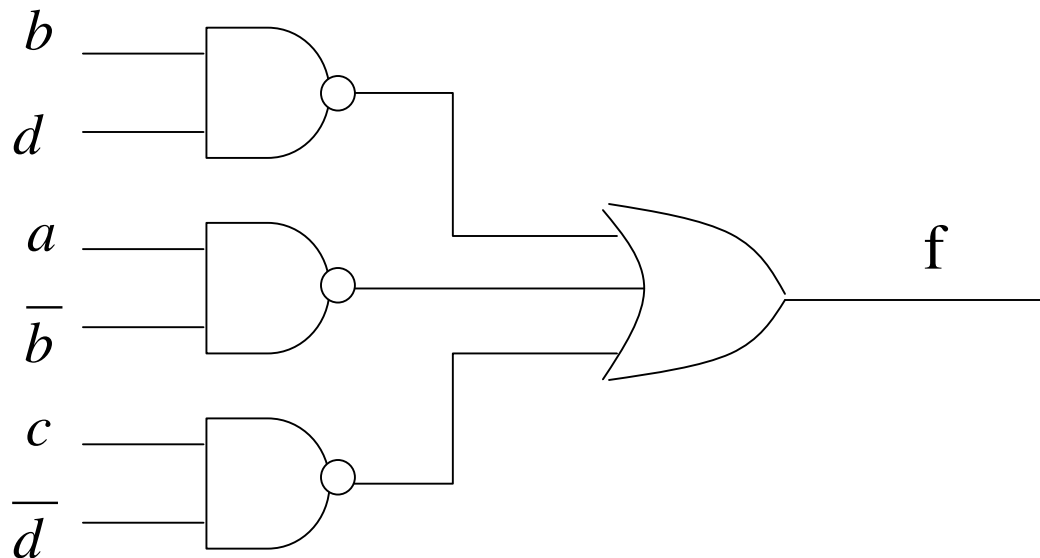
$$= \overline{\overline{\bar{b} + \bar{d}}}. \overline{\overline{\bar{a} + b}}. \overline{\overline{\bar{c} + d}} = \overline{\overline{b} \overline{\overline{d}}}. \overline{\overline{a} \overline{\overline{b}}}. \overline{\overline{c} \overline{\overline{d}}} = \overline{\overline{b} d}. \overline{\overline{a} \bar{b}}. \overline{\overline{c} d}$$



✓ Tầng một là NAND, tầng hai là OR.

Hàm logic là một tuyến n biến. Số đầu vào của phần tử OR là m. $n > m$. Viết hàm ở dạng tổng của các tuyến, mỗi tuyến phủ định hai lần và áp dụng qui tắc De Moorgan

$$f = \overline{\overline{b}} + \overline{\overline{d}} + \overline{\overline{a}} + \overline{\overline{b}} + \overline{\overline{c}} + \overline{\overline{d}} = (\overline{\overline{b + d}}) + (\overline{\overline{a + b}}) + (\overline{\overline{c + d}}) = \overline{\overline{bd}} + \overline{\overline{ab}} + \overline{\overline{cd}}$$

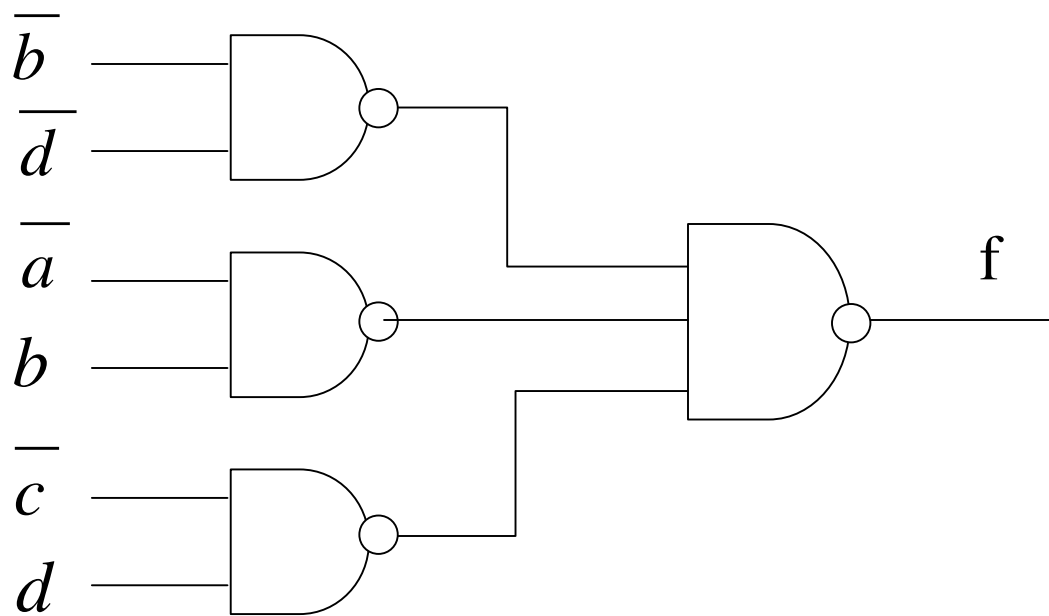


✓ Tầng một là NAND, tầng hai là NAND.

Hàm logic được viết ở dạng CTT, phủ định hai lần và áp dụng qui tắc De Morgan một lần:

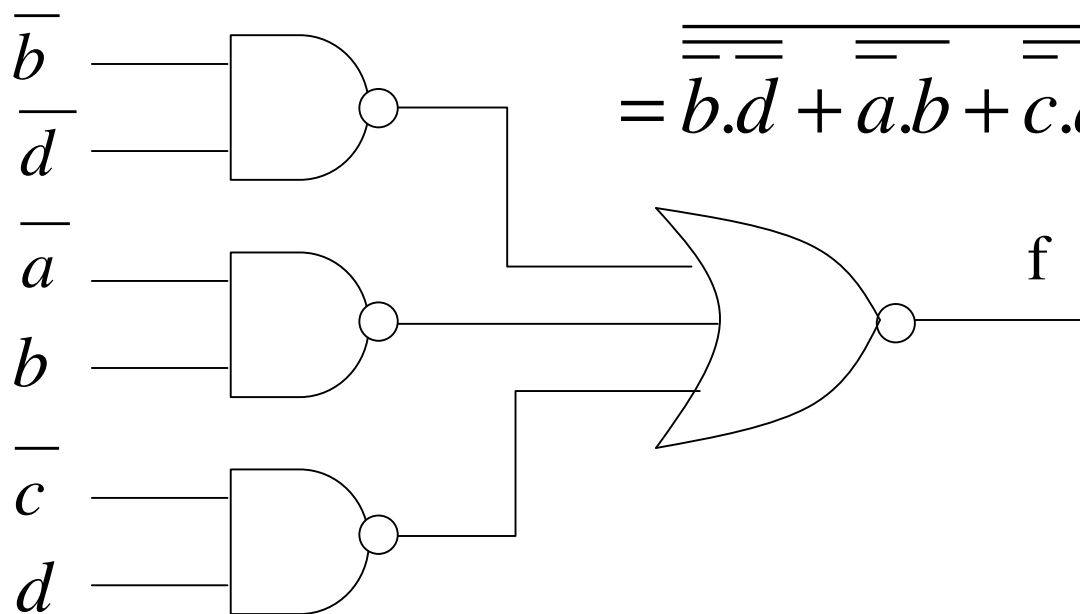
$$f = \overline{\overline{b} \cdot \overline{d}} + \overline{a \cdot b} + \overline{c \cdot d} = \overline{\overline{\overline{b \cdot d}} + \overline{a \cdot b} + \overline{c \cdot d}}$$

$$= \overline{\overline{\overline{b \cdot d \cdot a \cdot b \cdot c \cdot d}}}$$



✓ Tầng một là NAND, tầng hai là NOR.

Hàm logic là một hội n biến. Số đầu vào của phần tử NOR là m. $n > m$. Viết hàm ở dạng tích của các hội, phủ định hai lần và áp dụng qui tắc De Moorgan



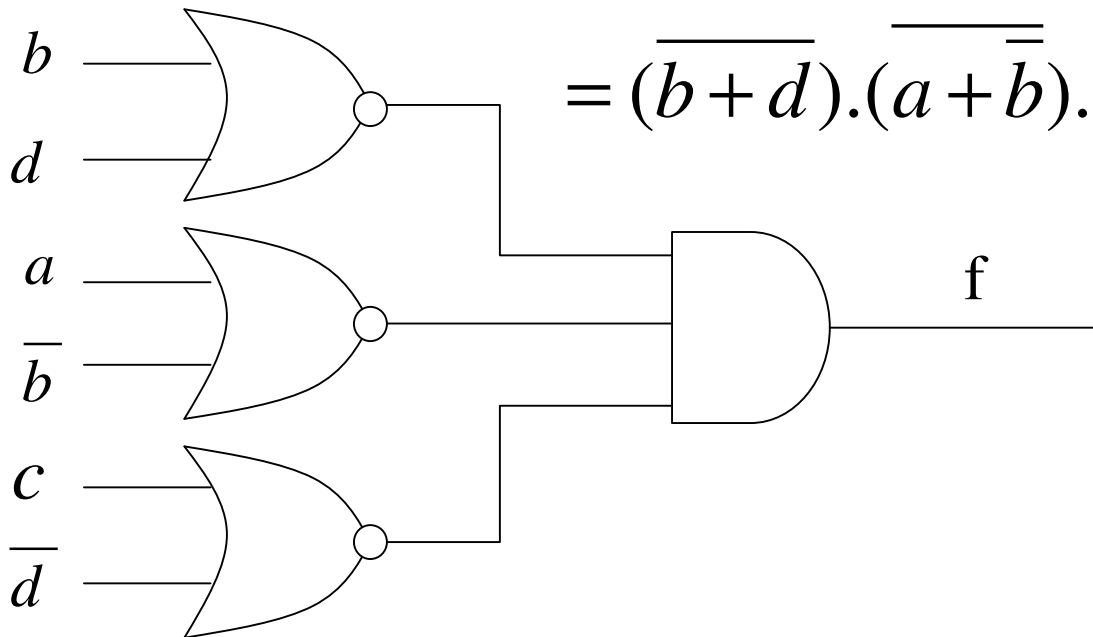
$$f = \overline{\overline{b} \cdot \overline{d} \cdot \overline{a} \cdot b \cdot \overline{c} \cdot d} = \overline{\overline{\overline{b \cdot d}} \cdot \overline{\overline{a \cdot b}} \cdot \overline{\overline{c \cdot d}}}$$

$$= \overline{\overline{\overline{b \cdot d}} + \overline{\overline{a \cdot b}} + \overline{\overline{c \cdot d}}}$$

✓ Tầng một là NOR, tầng hai là AND.

Hàm logic là một hội n biến. Số đầu vào của phần tử AND là m. $n > m$. Viết hàm ở dạng tích của các hội, phủ định hai lần mỗi hội và áp dụng qui tắc De Moorgan.

$$f = \overline{\overline{b} \cdot \overline{d}} \cdot \overline{\overline{a} \cdot \overline{b}} \cdot \overline{\overline{c} \cdot \overline{d}} = (\overline{\overline{b \cdot d}}) \cdot (\overline{\overline{a \cdot b}}) \cdot (\overline{\overline{c \cdot d}}) = (\overline{b + d}) \cdot (\overline{a + b}) \cdot (\overline{c + d})$$

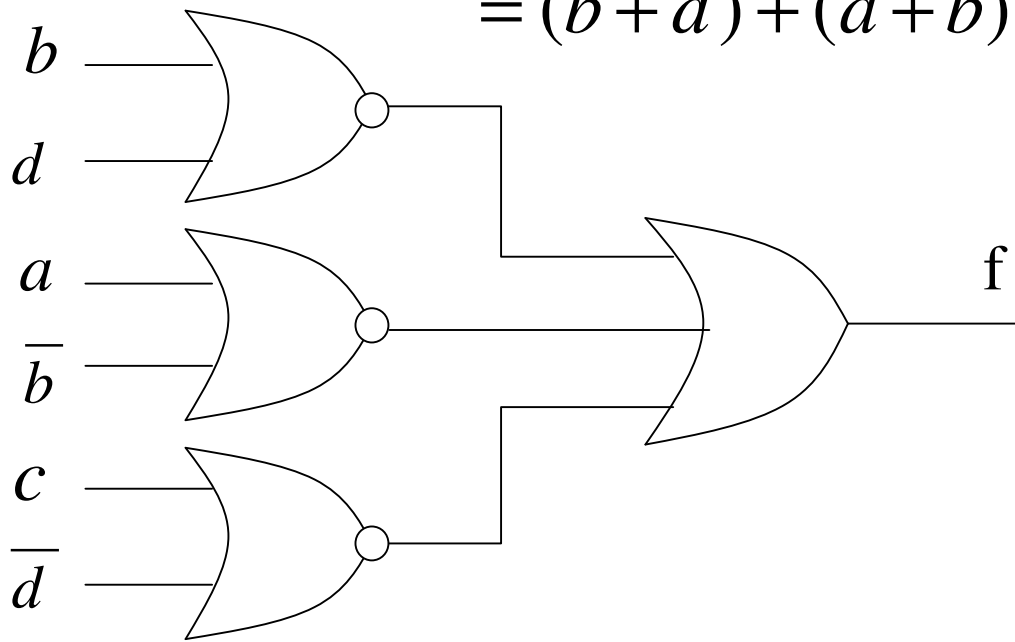


✓ Tầng một là *NOR*, tầng hai là *OR*.

Hàm logic được viết ở dạng CTT, phủ định hai lần từng hội và áp dụng qui tắc De Moorgan:

$$f = \overline{\overline{b}}.\overline{\overline{d}} + \overline{\overline{a}}.\overline{\overline{b}} + \overline{\overline{c}}.\overline{\overline{d}} = \overline{\overline{\overline{\overline{b}}}}.\overline{\overline{\overline{\overline{d}}}} + \overline{\overline{\overline{\overline{a}}}}.\overline{\overline{\overline{\overline{b}}}} + \overline{\overline{\overline{\overline{c}}}}.\overline{\overline{\overline{\overline{d}}}} =$$

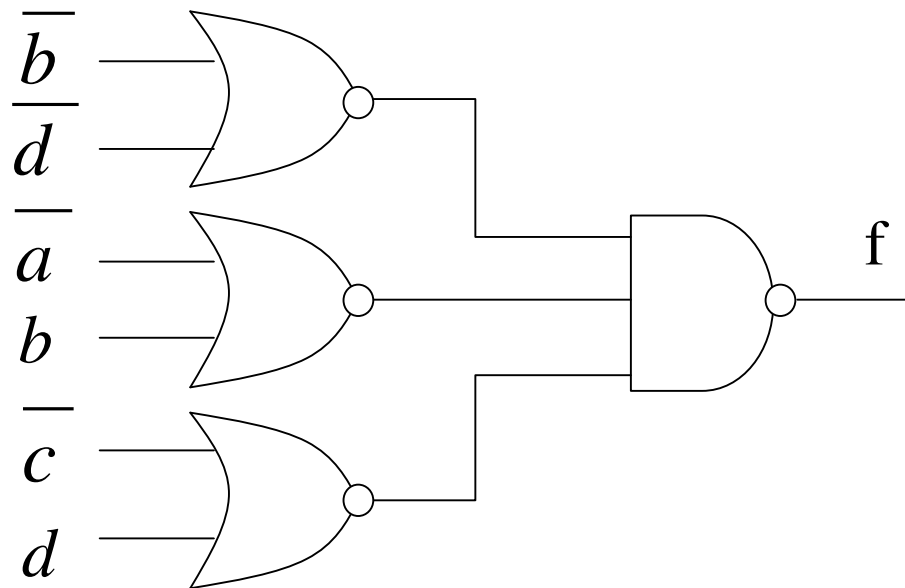
$$= \overline{\overline{b+d}} + \overline{\overline{a+b}} + \overline{\overline{c+d}}$$



✓ Tầng một là NOR, tầng hai là NAND.

Hàm logic là một tuyến n biến. Số đầu vào của phần tử NAND là m. $n > m$. Viết hàm ở dạng tổng của các tuyến, phủ định hai lần và áp dụng qui tắc De Moorgan.

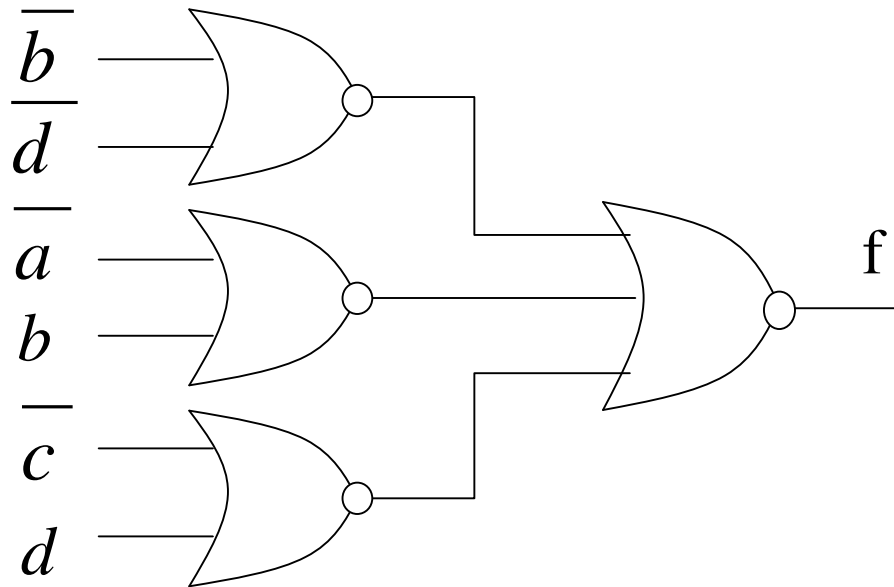
$$f = \overline{\overline{b + d} + \overline{a + b} + \overline{c + d}} = \overline{\overline{b + d}} \cdot \overline{\overline{a + b}} \cdot \overline{\overline{c + d}} = (b + d) \cdot (a + b) \cdot (c + d)$$



✓ Tầng một là NOR, tầng hai là NOR.

Hàm logic được viết ở dạng CTH, phủ định hai lần và áp dụng qui tắc De Moorgan một lần:

$$f = (\bar{b} + \bar{d})(\bar{a} + b)(\bar{c} + d) = \overline{\overline{(\bar{b} + \bar{d})(\bar{a} + b)(\bar{c} + d)}} \\ = \overline{(\bar{b} + \bar{d}) + (\bar{a} + b) + (\bar{c} + d)}$$



Bảng tóm tắt thiết kế mạch tổ hợp 2 tầng

Tầng 1 \ Tầng 2	AND	OR	NAND	NOR
AND	Hội n biến	CTT	Phủ định hội n biến	CTH, phủ định 2 lần, DM 2 lần
OR	CTH	Tuyển n biến	CTT, phủ định 2 lần, DM 2 lần	Phủ định tuyển n biến
NAND	CTH, phủ định 2 lần tuyển, DM	Tổng các tuyển, phủ định 2 lần tuyển, DM	CTT, phủ định 2 lần, DM 1 lần	Tích các hội, phủ định 2 lần, DM
NOR	Tích các hội, phủ định 2 lần hội, DM	CTT, phủ định 2 lần hội, DM	Tổng các tuyển, phủ định 2 lần, DM	CTH, phủ định 2 lần, DM 1 lần

1.3.3 Thiết kế hệ các hàm logic.

- ✓ *Thiết kế riêng từng hàm:* như trình bày ở những mục trên.
- ✓ *Sử dụng những phần chung của các hàm* cho phép giảm độ phức tạp của sơ đồ cả hệ.
 - Lập bảng Karnaugh cho từng hàm, đánh dấu các đỉnh 1 của chúng;
 - Khoanh các đỉnh 1 chung của từ 2 hàm trở lên;
 - Thực hiện dán các đỉnh 1 riêng, rồi dán các đỉnh 1 chung của các hàm theo cách giống nhau

- VD: Thiết kế hệ 3 hàm f , g , h phụ thuộc 4 biến a , b , c , d có các đỉnh 1 như sau:

f

ab \ cd	00	01	11	10
00			1	
01	1			
11	1	1		
10			1	

g

ab \ cd	00	01	11	10
00			1	1
01		1		
11				1
10			1	1

h

ab \ cd	00	01	11	10
00			1	1
01	1	1		
11		1		
10			1	

Thiết kế riêng rẽ từng hàm:

f

ab \ cd	00	01	11	10
00			1	
01	1			
11	1		1	
10			1	

g

ab \ cd	00	01	11	10
00			1	1
01		1		
11				1
10			1	1

h

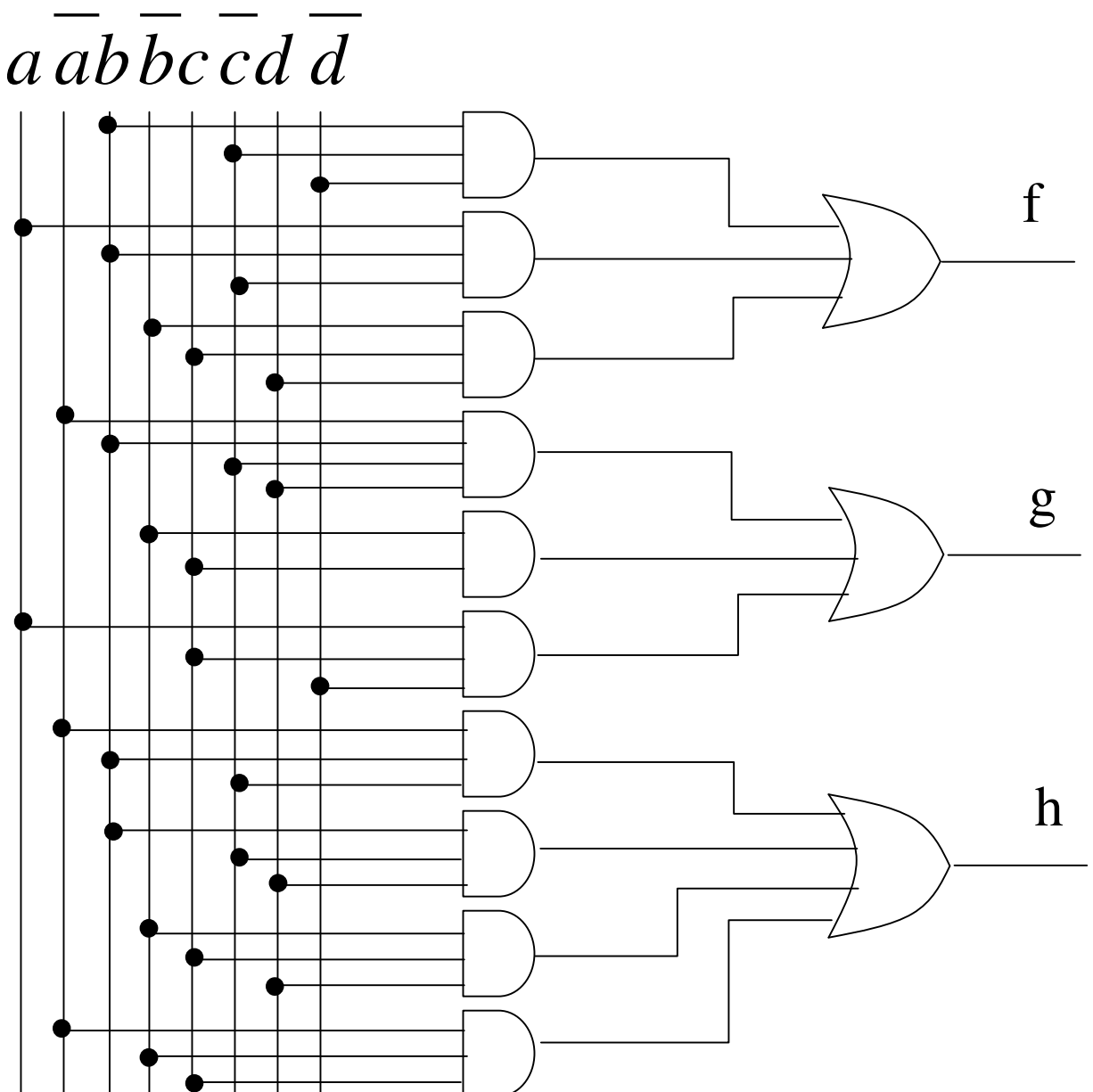
ab \ cd	00	01	11	10
00			1	1
01	1	1		
11		1		
10			1	

$$f = \bar{b}\bar{c}.d + ab\bar{c} + \bar{b}cd$$

$$g = \bar{a}bcd + \bar{b}c + ac\bar{d}$$

$$h = \bar{a}bc + bcd + \bar{b}cd + \bar{a}.\bar{b}c$$

30 đầu vào, 10 AND, 3 OR



Thiết kế sử dụng các phần chung:

f

ab \ cd	00	01	11	10
00			(1)	
01	(1)			
11	1	(1)		
10			(1)	

g

ab \ cd	00	01	11	10
00			(1)	(1)
01		(1)		
11				1
10			(1)	1

h

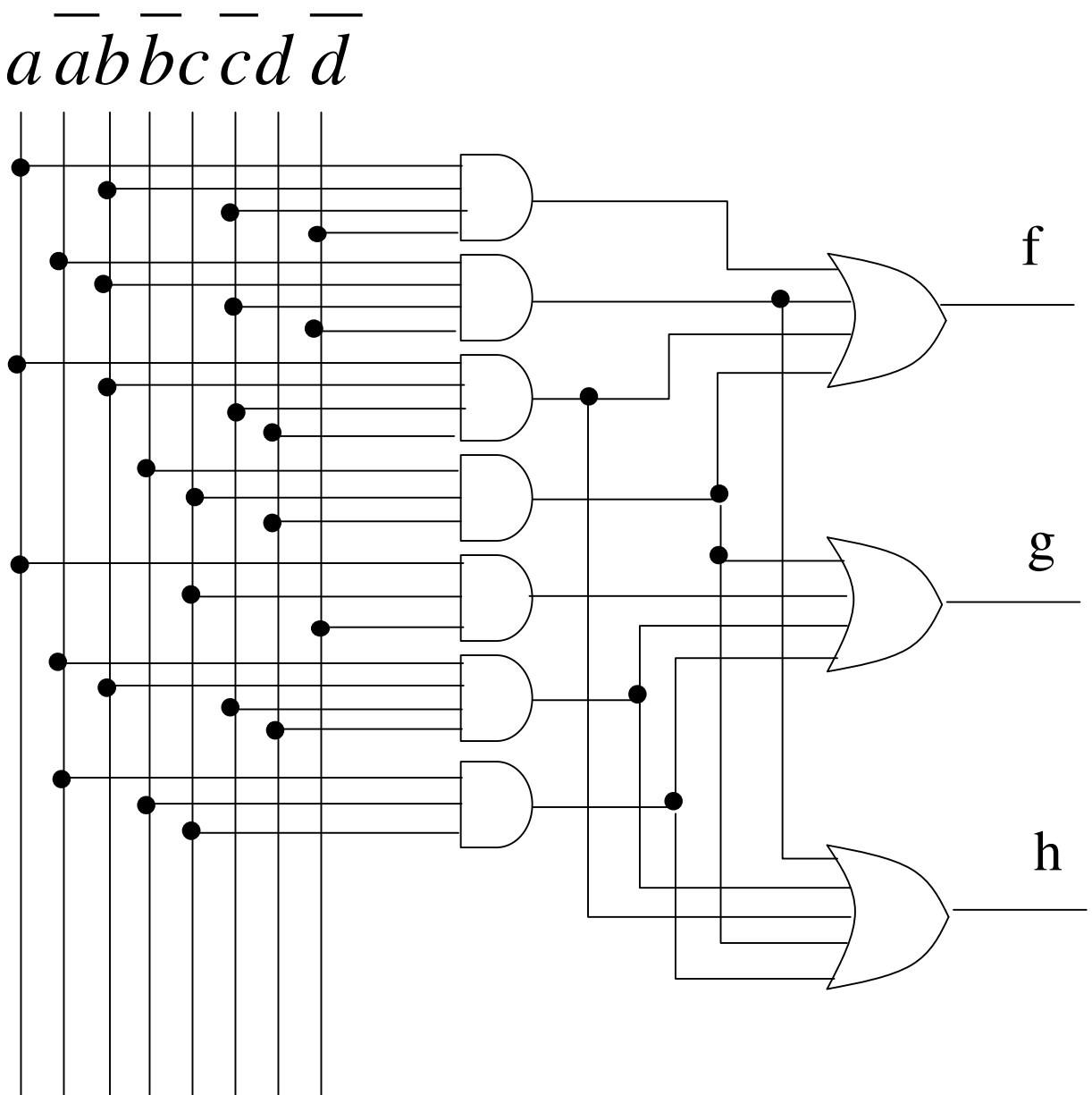
ab \ cd	00	01	11	10
00			(1)	(1)
01	(1)	(1)		
11		(1)		
10			(1)	

$$f = (12) + (4) + (13) + \bar{b}cd$$

$$g = ac\bar{d} + (5) + \bar{b}cd + \bar{a}\bar{b}c$$

$$h = (4) + (5) + (13) + \bar{b}cd + \bar{a}\bar{b}c$$

25 đầu vào, 7 AND, 3 OR



1.4 MỘT SỐ MẠCH TỔ HỢP THƯỜNG GẶP.

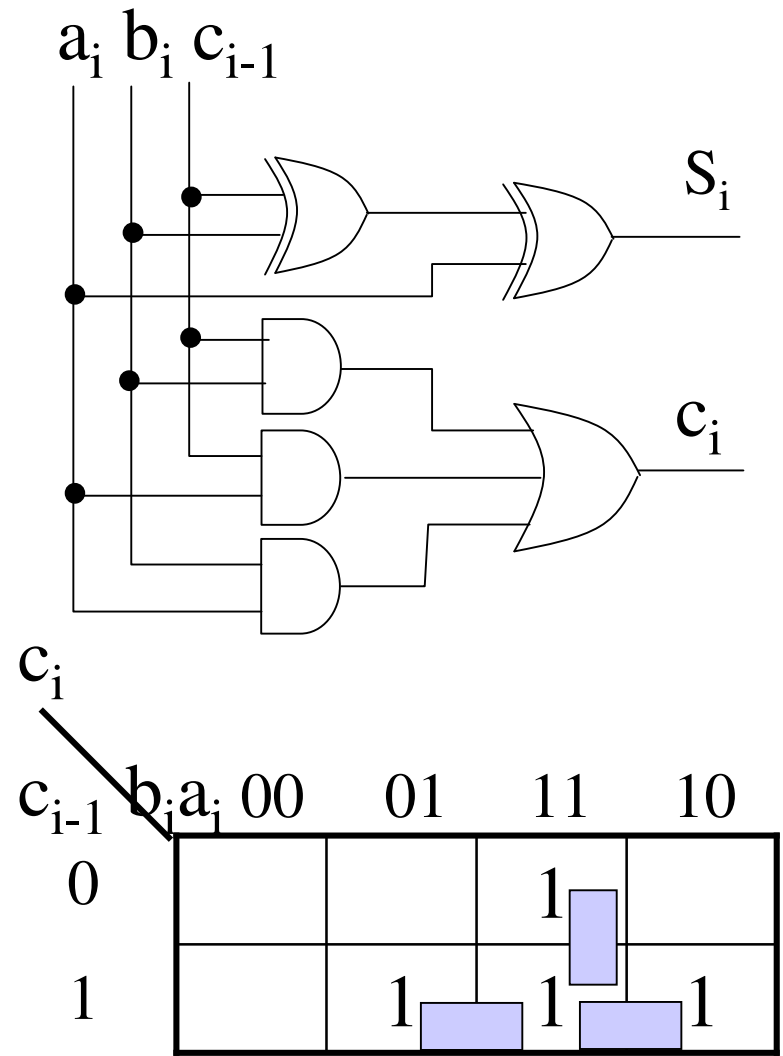
(Combinational Circuits)

1.4.1 CÁC BỘ CỘNG NHỊ PHÂN 1 BIT

- **Bộ cộng 1 bit** thực hiện cộng 2 biến nhị phân a_i và b_i đồng thời với biến nhớ c_{i-1} (nhớ từ phép cộng các bit có trọng số nhỏ hơn a_{i-1} và b_{i-1}); đầu ra cho kết quả là bit tổng S_i và bit nhớ c_i .

Bảng chân lý:

Đầu vào			Đầu ra	
c_{i-1}	b_i	a_i	S_i	c_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



$$S_i = c_{i-1} \oplus b_i \oplus a_i$$

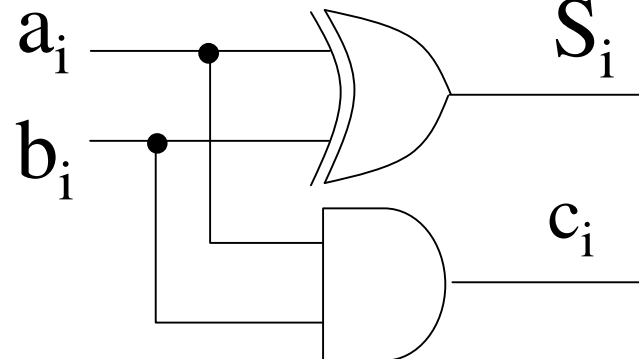
$$c_i = b_i a_i + c_{i-1} b_i + c_{i-1} a_i$$

- **Bộ bán tổng (Half-adder).**

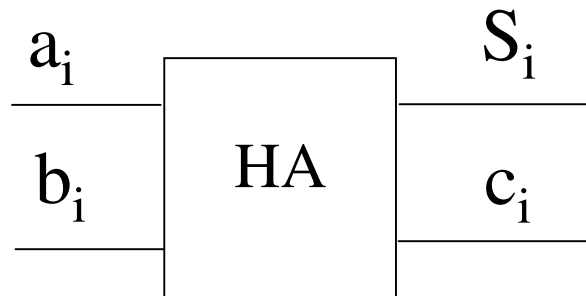
Bảng chân lý:

b_i	a_i	S_i	c_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

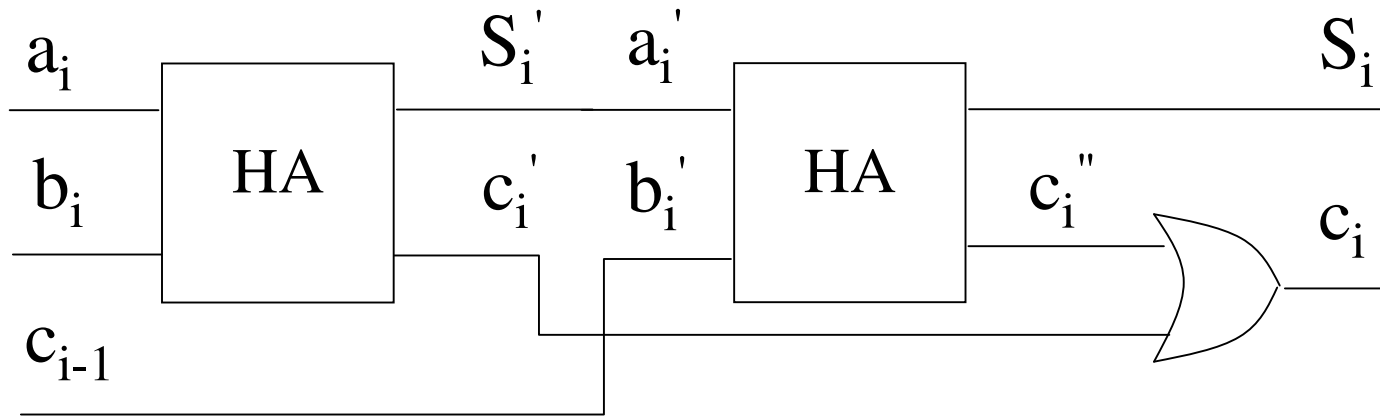
$$S_i = b_i \oplus a_i \qquad c_i = b_i a_i$$



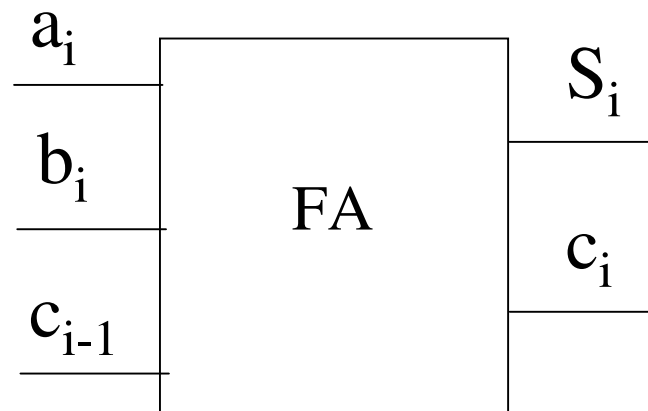
Ký hiệu:



- Bộ cộng 1 bit đầy đủ (Full-adder) từ 2 bộ bán tổng.



Ký hiệu:



Phân tích mạch:

$$S_i = a_i' \oplus b_i' = S_i' \oplus c_{i-1} = a_i \oplus b_i \oplus c_{i-1}$$

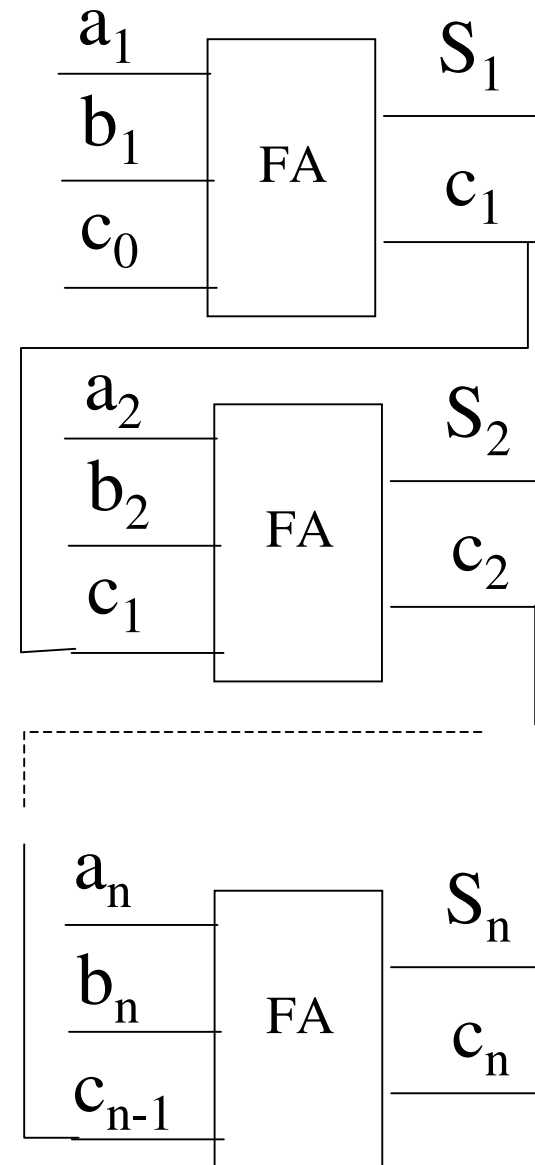
$$\begin{aligned} c_i &= c_i'' + c_i' = a_i' b_i' + a_i b_i = S_i' c_{i-1} + a_i b_i = \\ &= (a_i \oplus b_i) c_{i-1} + a_i b_i = (\overline{a_i} b_i + a_i \overline{b_i}) c_{i-1} + a_i b_i = \\ &= \overline{a_i} b_i c_{i-1} + a_i \overline{b_i} c_{i-1} + a_i b_i = \overline{a_i} b_i c_{i-1} + a_i (c_{i-1} + b_i) = \\ &= a_i c_{i-1} + a_i b_i + \overline{a_i} b_i c_{i-1} = a_i c_{i-1} + b_i (a_i + c_{i-1}) = \\ &= a_i b_i + a_i c_{i-1} + b_i c_{i-1} \end{aligned}$$

1.4.2 CÁC BỘ CỘNG NHỊ PHÂN n BIT

- **Bộ cộng n bit chuyển nhớ tuần tự.**

- T/h nhớ c_1 bị giữ chậm tối thiểu là $2t_0$, với t_0 là thời gian giữ chậm trung bình của 1 cổng logic.

- T/h nhớ c_n bị giữ chậm $2nt_0$.



- **Bộ cộng n bit chuyển nhớ nhanh (Fast Carry, Carry Look Ahead).**

Xét bộ cộng 1 bit FA, ta có:

$$S_i = a_i \oplus b_i \oplus c_{i-1}$$

$$c_i = (a_i \oplus b_i)c_{i-1} + a_i b_i$$

$$\text{Đặt: } (a_i \oplus b_i) = p_i \quad a_i b_i = g_i$$

$$\text{Ta có: } S_i = p_i \oplus c_{i-1} \quad c_i = p_i c_{i-1} + g_i$$

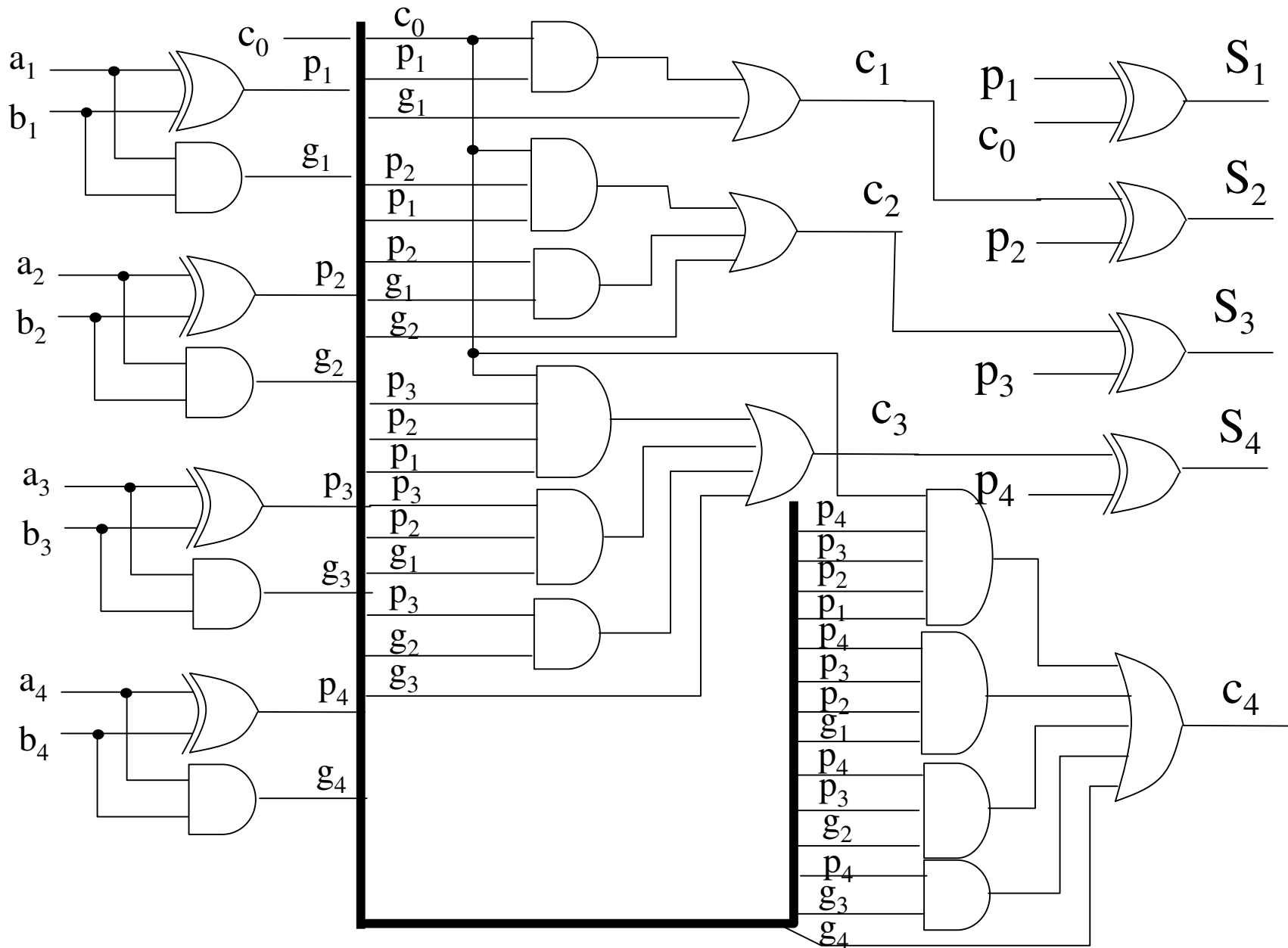
$$\text{Triển khai các bit nhớ: } c_1 = p_1 c_0 + g_1$$

$$c_2 = p_2 c_1 + g_2 = p_2 (p_1 c_0 + g_1) + g_2 = p_2 p_1 c_0 + p_2 g_1 + g_2$$

$$c_3 = p_3 p_2 p_1 c_0 + p_3 p_2 g_1 + p_3 g_2 + g_3$$

$$c_4 = p_4 p_3 p_2 p_1 c_0 + p_4 p_3 p_2 g_1 + p_4 p_3 g_2 + p_4 g_3 + g_4$$

Sơ đồ bộ cộng 4 bit chuyển nhớ nhanh:



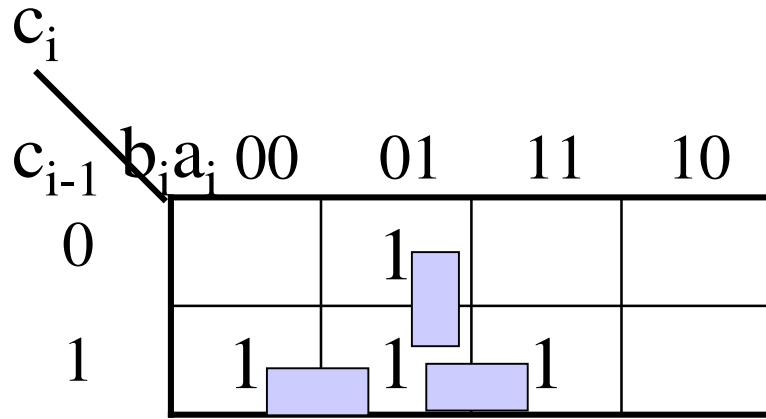
1.4.3 CÁC BỘ TRỪ NHỊ PHÂN 1 BIT

- **Bộ trừ 1 bit đầy đủ** (FS-Full Subtractor) thực hiện phép tính: $b_i - (a_i + c_{i-1})$, đầu ra là bit hiệu H_i và bit nhớ c_i sang cột có trọng số lớn hơn tiếp theo.

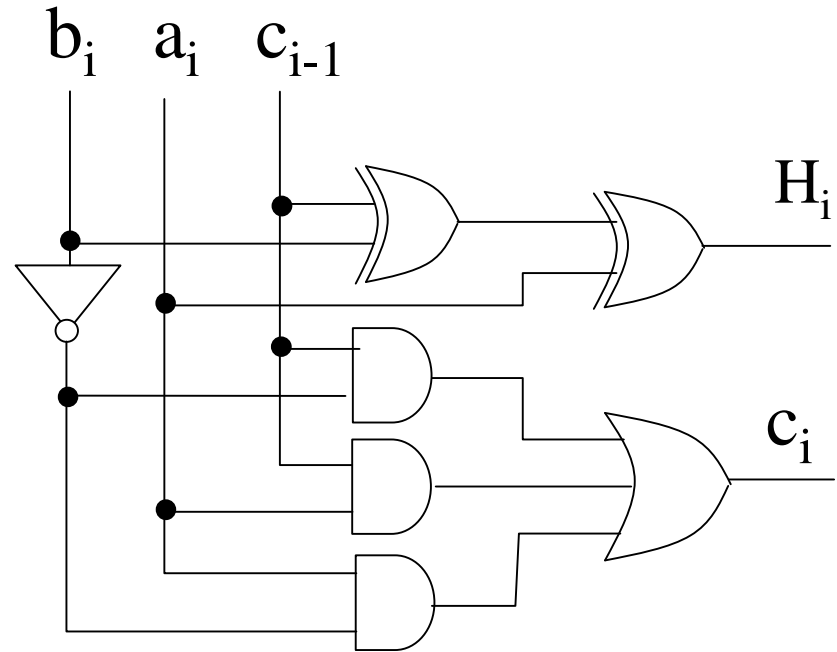
Bảng chân lý:

Đầu vào			Đầu ra	
b_i	a_i	c_{i-1}	H_i	c_i
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

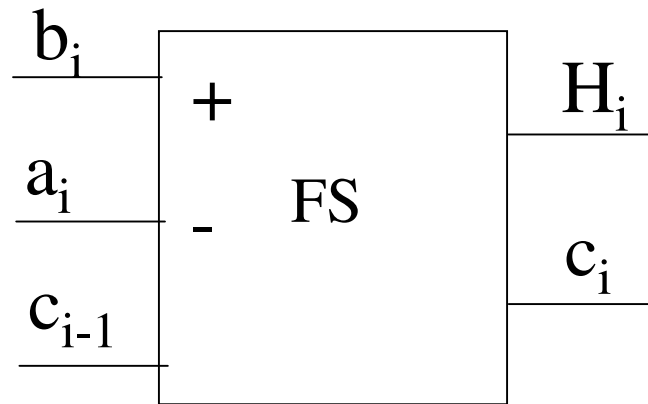
$$H_i = c_{i-1} \oplus b_i \oplus a_i$$



$$c_i = \bar{b}_i a_i + c_{i-1} \bar{b}_i + c_{i-1} a_i$$



Ký hiệu:

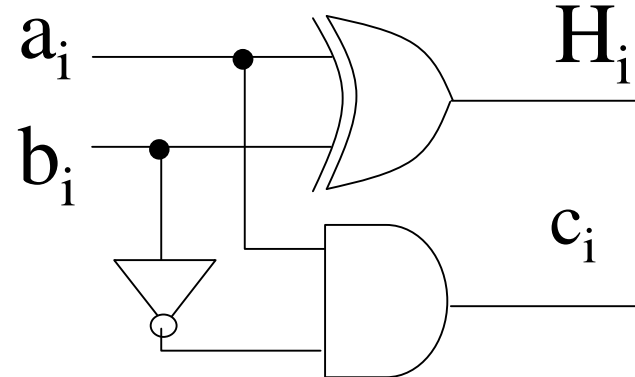


- **Bộ bán trừ (Half-Subtractor).**

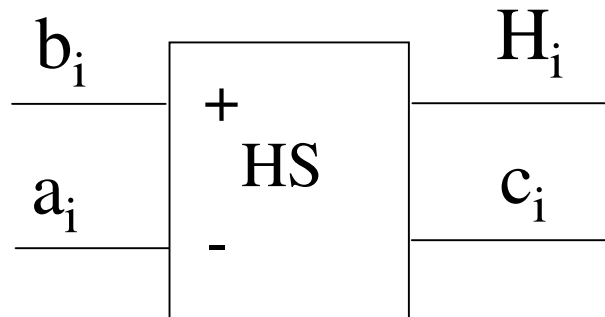
Bảng chân lý:

b_i	a_i	H_i	c_i
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

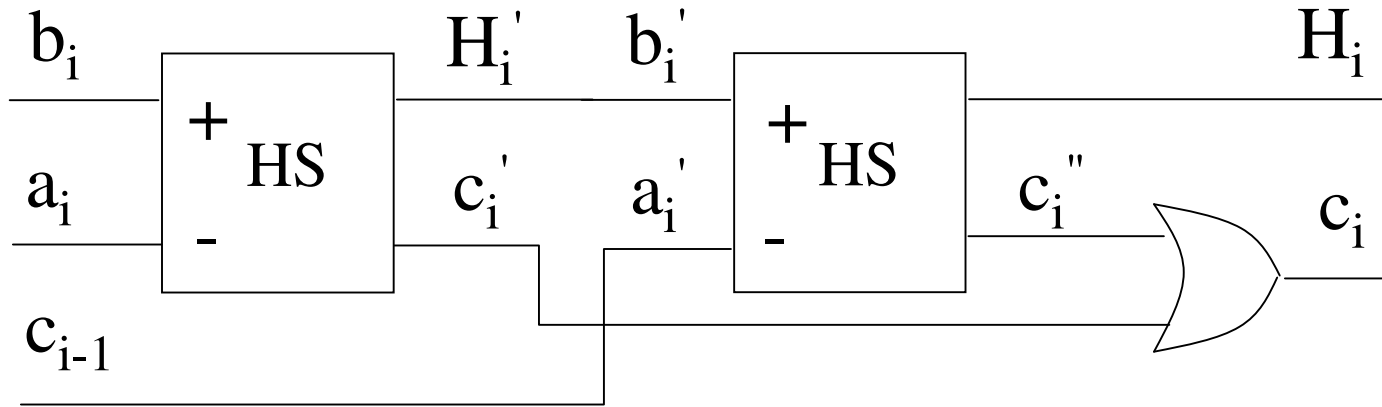
$$H_i = b_i \oplus a_i \quad c_i = \overline{b_i} a_i$$



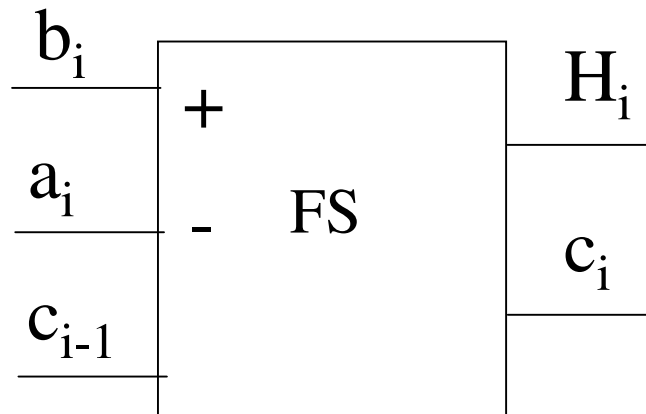
Ký hiệu:



- Bộ trừ 1 bit đầy đủ (Full-Subtractor) từ 2 bộ bán trừ.



Ký hiệu:



Phân tích mạch:

$$H_i = b_i' \oplus a_i' = H_i' \oplus c_{i-1} = b_i \oplus a_i \oplus c_{i-1}$$

$$\begin{aligned} c_i &= c_i' + c_i'' = \overline{b_i}a_i + \overline{b_i'}a_i' = \overline{b_i}a_i + \overline{(b_i \oplus a_i)}c_{i-1} = \\ &= \overline{b_i}a_i + \overline{(\overline{b_i}a_i + b_i\overline{a_i})}c_{i-1} = \overline{b_i}a_i + \overline{\overline{b_i}a_i} \cdot \overline{b_i\overline{a_i}} \cdot c_{i-1} = \\ &= \overline{b_i}a_i + (b_i + \overline{a_i})(\overline{b_i} + a_i)c_{i-1} = \overline{b_i}a_i + b_i\overline{a_i}c_{i-1} + \overline{a_i}\overline{b_i}c_{i-1} = \\ &= a_i(\overline{b_i} + b_i\overline{c_{i-1}}) + \overline{a_i}\overline{b_i}c_{i-1} = a_i\overline{b_i} + a_i\overline{c_{i-1}} + \overline{a_i}\overline{b_i}c_{i-1} = \\ &= a_i\overline{c_{i-1}} + \overline{b_i}c_{i-1} + a_i\overline{b_i} \end{aligned}$$

1.4.4 CÁC BỘ SO SÁNH 2 SỐ NHỊ PHÂN

- Bộ so sánh 2 số nhị phân 1 bit.

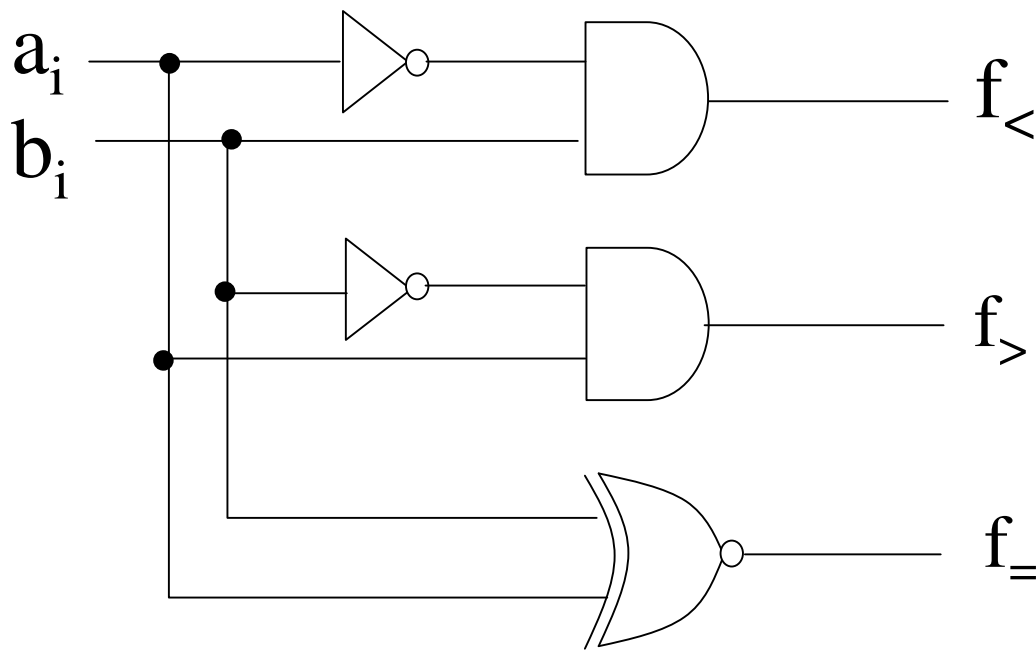
Bảng chân lý:

a_i	b_i	$f_{<}$	$f_{>}$	$f_{=}$
0	0	0	0	1
0	1	1	0	0
1	0	0	1	0
1	1	0	0	1

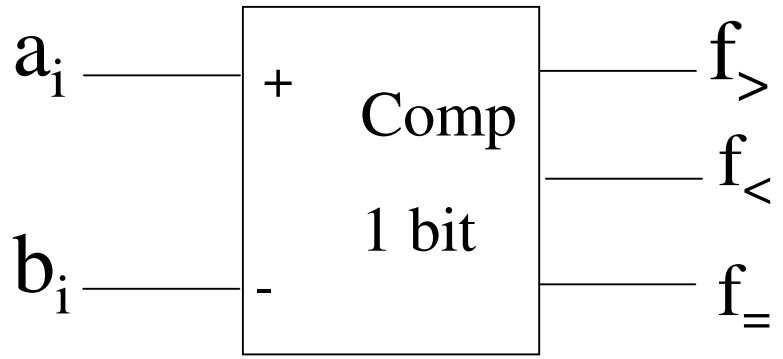
$$f_{<} = \overline{a_i} b_i \quad f_{>} = a_i \overline{b_i}$$

$$f_{=} = \overline{a_i} \overline{b_i} + a_i b_i = a_i \approx b_i = \overline{a_i} \oplus b_i$$

Sơ đồ logic:



Ký hiệu:



- **Bộ so sánh 2 số nhị phân 2 bit.**

$$A = A_2A_1 \quad B = B_2B_1$$

Nếu: $A_2 > B_2 \rightarrow A > B$

$$A_2 < B_2 \rightarrow A < B$$

$$A_2 = B_2$$

Nếu: $A_1 > B_1 \rightarrow A > B$

$$A_1 < B_1 \rightarrow A < B$$

$$A_1 = B_1 \rightarrow A = B$$

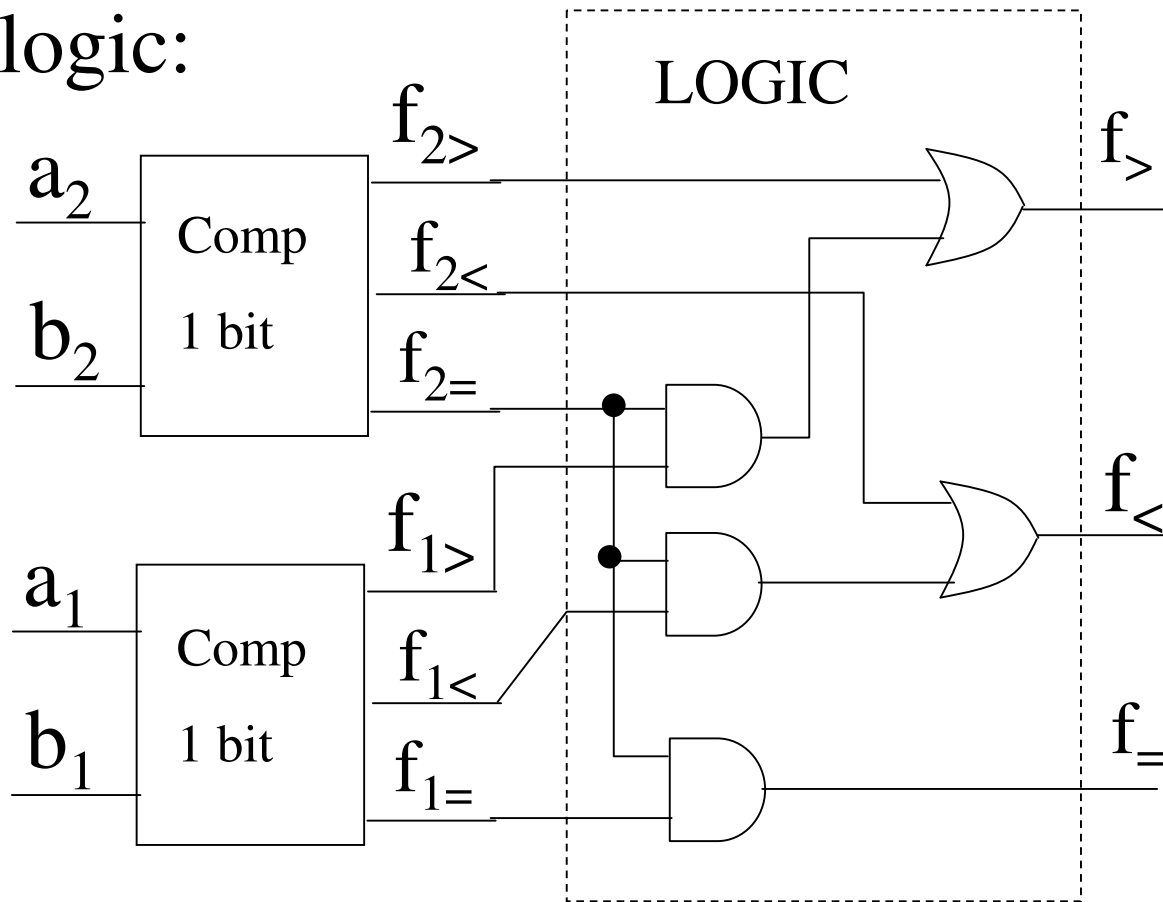
Xác định các hàm logic:

$$f_{>} = f_{2>} + f_{2=} f_{1>}$$

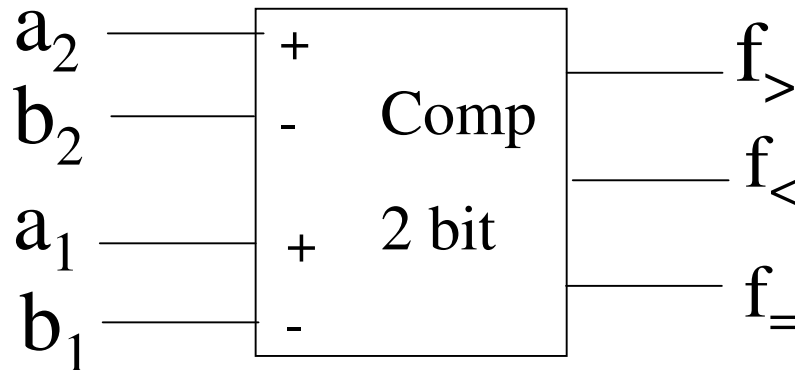
$$f_{<} = f_{2<} + f_{2=} f_{1<}$$

$$f_{=} = f_{2=} f_{1=}$$

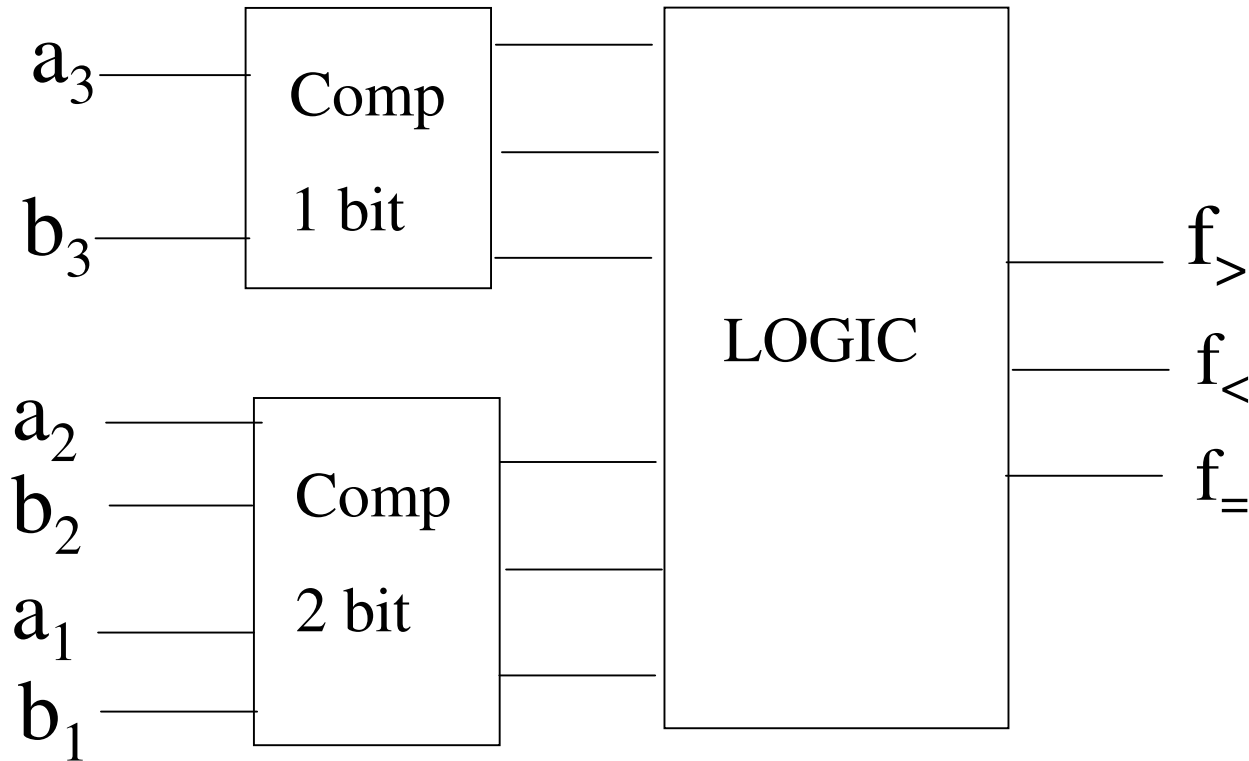
Sơ đồ logic:



Ký hiệu:



- Bộ so sánh 2 số nhị phân 3 bit.



1.4.5 MẠCH TẠO VÀ KIỂM TRA CHẴN LẺ

- **Mạch tạo bit chẵn lẻ.**

Mạch tạo ra 2 tín hiệu: X_e – bit chẵn (Even bit), X_o – bit lẻ (Odd bit). Trong n bit dữ liệu đầu vào, nếu số các chữ số 1 là lẻ:

$$b_n \oplus b_{n-1} \oplus \dots \oplus b_1 = 1$$

Thì $X_e = 1$ và $X_o = 0$

Nếu số các chữ số 1 là chẵn:

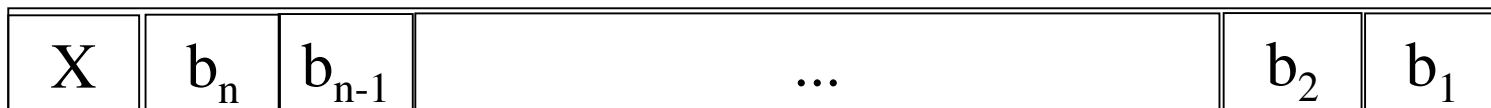
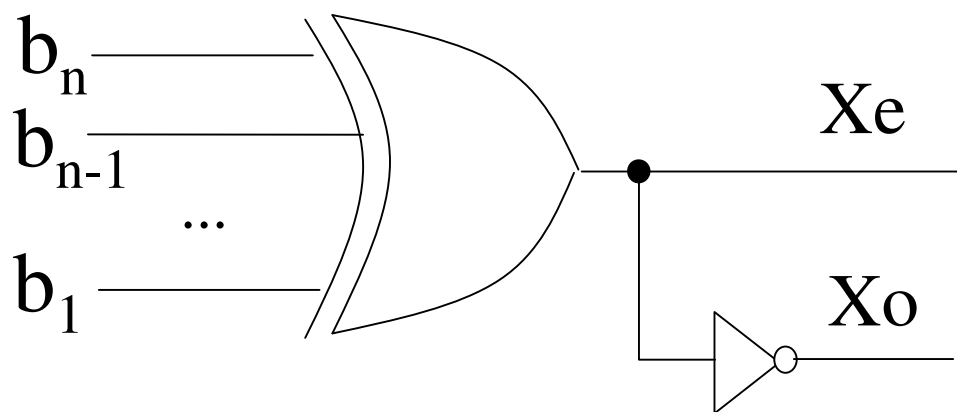
$$b_n \oplus b_{n-1} \oplus \dots \oplus b_1 = 0$$

Thì $X_e = 0$ và $X_o = 1$

Nếu hệ là **hệ chẵn**, thì dãy n bit dữ liệu được thêm 1 bit $X = X_e$, nếu là **hệ lẻ** thì bit thêm vào là $X = X_o$. Hàm logic và sơ đồ mạch tạo bit chẵn và bit lẻ như sau:

$$X_e = b_n \oplus b_{n-1} \oplus \dots \oplus b_1$$

$$X_o = \overline{X_e}$$

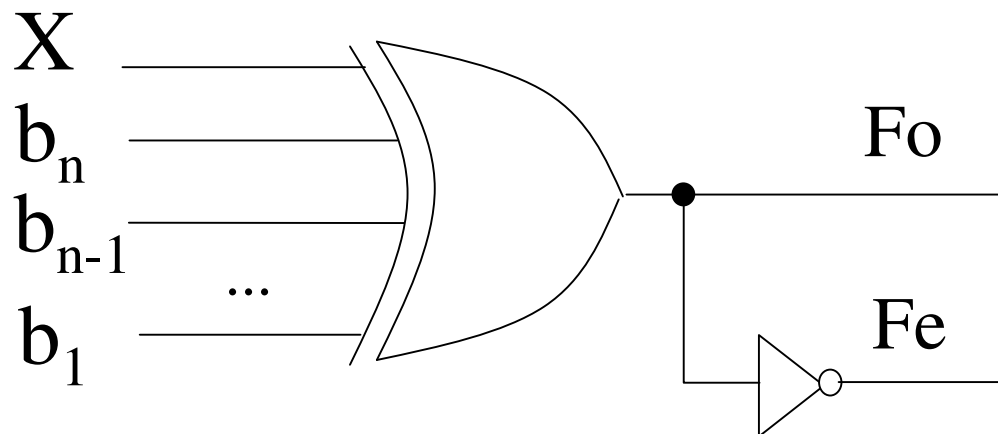


- **Mạch kiểm tra chẵn lẻ.**

Mạch tạo ra 2 tín hiệu thông báo (Flag): Fe – báo chẵn (Even), Fo – báo lẻ (Odd). Fe = 1 và Fo = 0 nếu hệ là chẵn. Ngược lại, Fe = 0 và Fo = 1 nếu hệ là lẻ.

$$F_o = X \oplus b_n \oplus b_{n-1} \oplus \dots \oplus b_1$$

$$F_e = \overline{F_o}$$

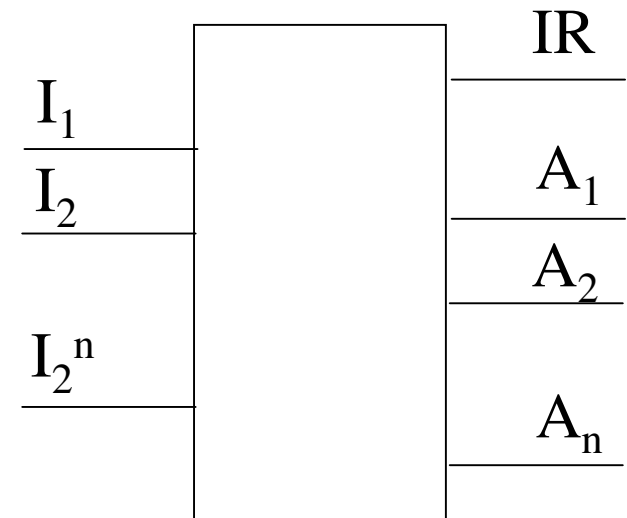


1.4.6 MẠCH PHÂN LOẠI NGẮT

- **Chức năng:**

Đầu vào: I_1, I_2, \dots, I_{2^n} là các t/h yêu cầu ngắt của 2^n thiết bị ngoại vi.

Đầu ra: $IR=1$ báo hiệu có yêu cầu ngắt; A_1, A_2, \dots, A_n xác định TB ngoại vi sẽ được phục vụ. Giá trị của số $A_n A_{n-1} \dots A_1$ là $11\dots 1$ ứng với mức ưu tiên cao nhất (khi có $I_1 = 1$), là $00\dots 0$ ứng với mức ưu tiên thấp nhất (khi chỉ có $I_{2^n} = 1$).



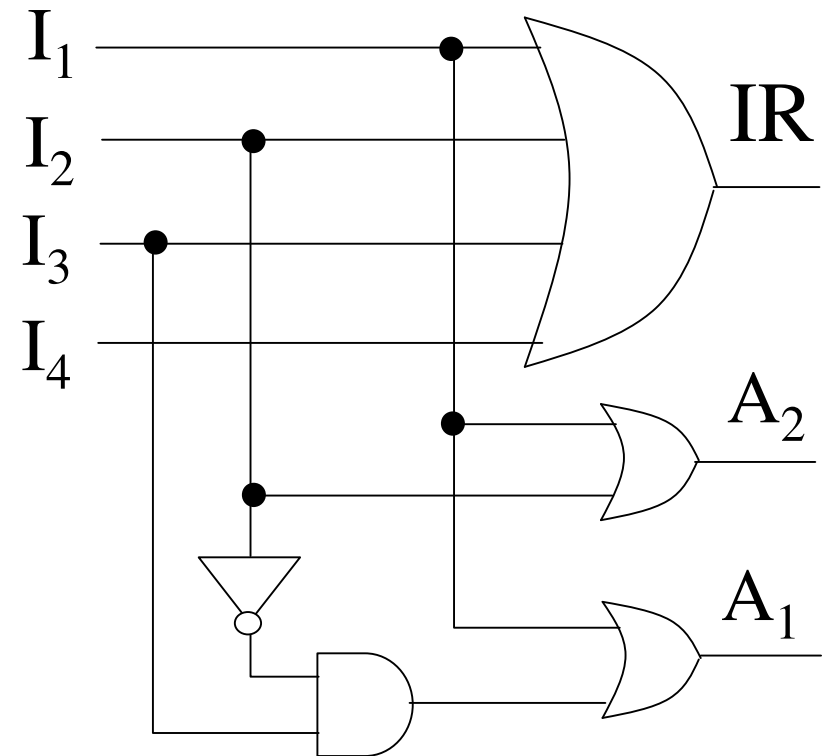
- Trường hợp $n = 2$ ($2^2 = 4$ TB ngoại vi).

I_1	I_2	I_3	I_4	IR	A_2	A_1
0	0	0	0	0	0	0
1	x	x	x	1	1	1
0	1	x	x	1	1	0
0	0	1	x	1	0	1
0	0	0	1	1	0	0

$$IR = I_1 + I_2 + I_3 + I_4$$

$$A_2 = I_1 + \bar{I}_1 I_2 = I_1 + I_2$$

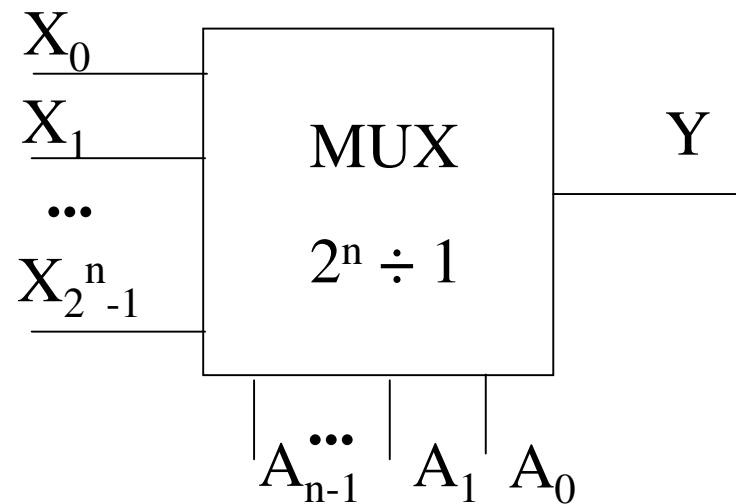
$$A_1 = I_1 + \bar{I}_1 \cdot \bar{I}_2 I_3 = I_1 + \bar{I}_2 I_3$$



1.4.7 BỘ DỒN KÊNH (MUX-Multiplexer/ Data Selector)

- **Chức năng.**

MUX có n đầu vào điều khiển A_{n-1}, \dots, A_1, A_0 cho phép chọn 1 trong 2^n đầu vào dữ liệu $X_0, X_1, \dots, X_{2^n-1}$ để đưa tới đầu ra Y .

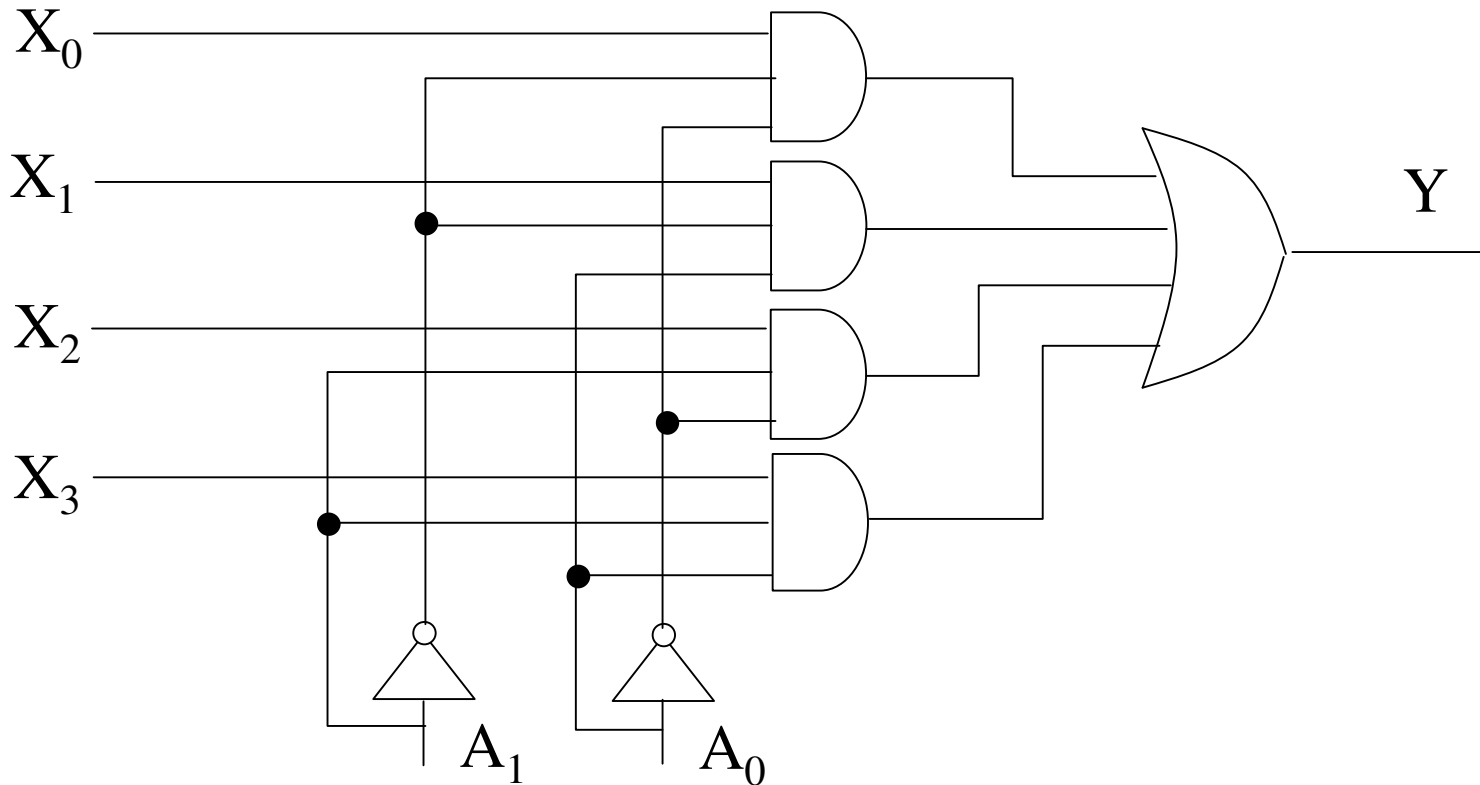


Hàm đầu ra:

$$Y = X_0(\overline{A_{n-1}} \dots \overline{A_1} \overline{A_0}) + X_1(\overline{A_{n-1}} \dots \overline{A_1} A_0) + \dots + X_{2^n-1}(A_{n-1} A_{n-2} \dots A_0)$$

Trường hợp n = 2

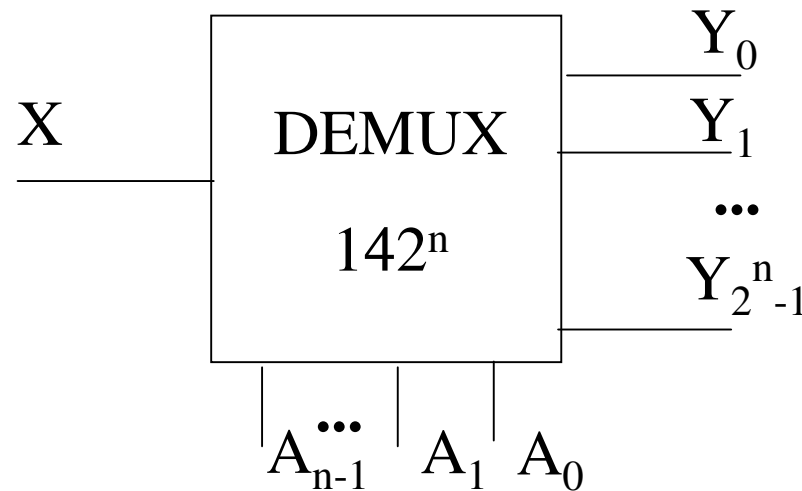
$$Y = X_0(\overline{A_1} \overline{A_0}) + X_1(\overline{A_1} A_0) + X_2(A_1 \overline{A_0}) + X_3(A_1 A_0)$$



1.4.8 BỘ PHÂN KÊNH (DEMUX-Demultiplexer)

- **Chức năng.**

DEMUX có n đầu vào điều khiển A_{n-1}, \dots, A_1, A_0 cho phép chọn 1 trong 2^n đầu ra $Y_0, Y_1, \dots, Y_{2^n-1}$ để nhận dữ liệu từ đầu vào X .



Các hàm đầu ra:

$$Y_0 = X(\overline{A_{n-1}} \dots \overline{A_1} \cdot \overline{A_0})$$

$$Y_1 = X(\overline{A_{n-1}} \dots \overline{A_1} \cdot A_0)$$

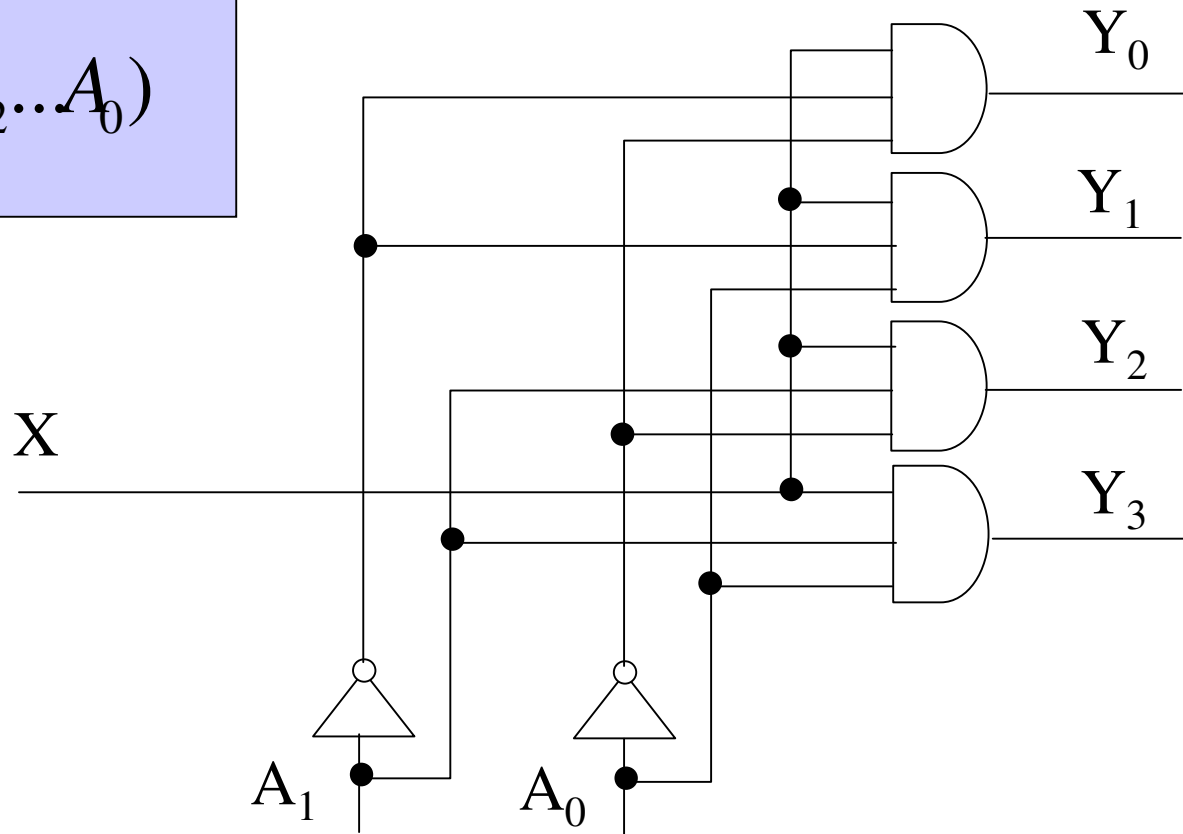
.....

$$Y_{2^n-1} = X(A_{n-1} A_{n-2} \dots A_0)$$

Trường hợp n = 2

$$Y_0 = X(\overline{A_1} \cdot \overline{A_0}) \quad Y_1 = X(\overline{A_1} \cdot A_0)$$

$$Y_2 = X(A_1 \cdot \overline{A_0}) \quad Y_3 = X(A_1 \cdot A_0)$$

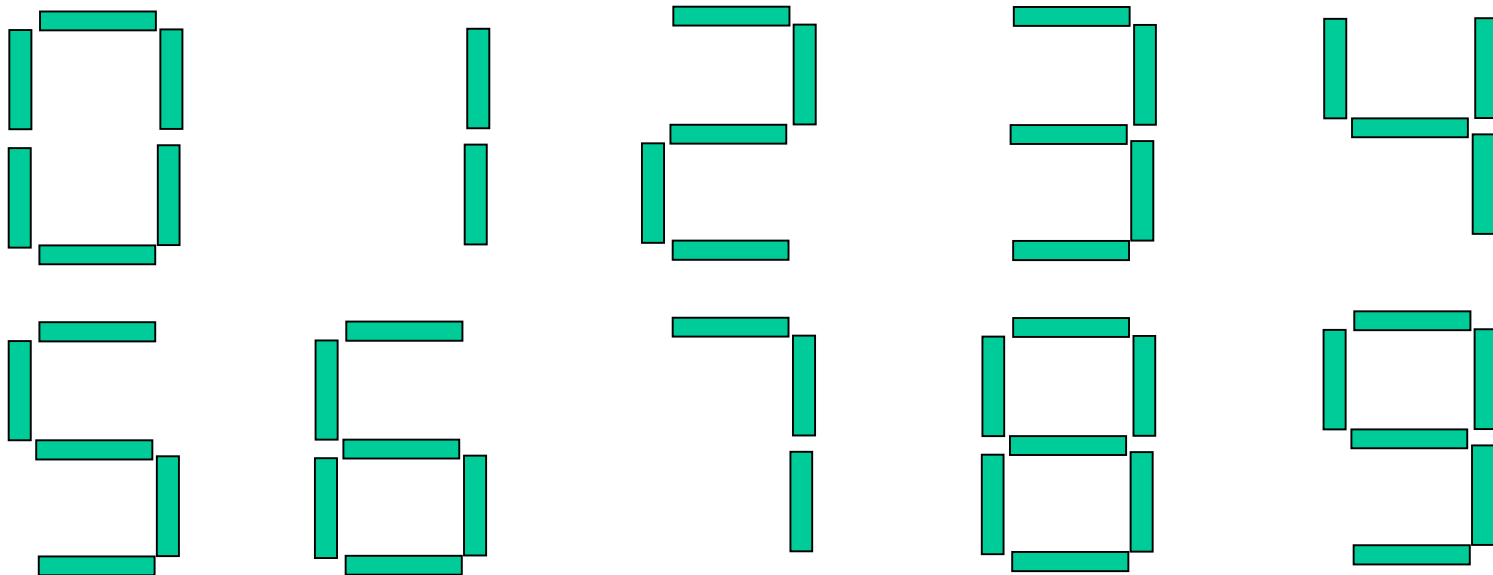


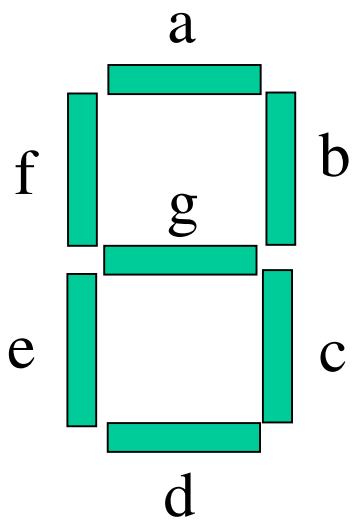
1.4.9 CÁC BỘ CHUYỂN MÃ

- **Các bước thiết kế:**
 - ✓ Lập bảng chân lý tất cả các tổ hợp biến đầu vào (mã đầu vào) và các tổ hợp hàm đầu ra tương ứng (mã đầu ra);
 - ✓ Lập bảng Karnaugh cho từng hàm đầu ra và tối thiểu hóa chúng;
 - ✓ Xây dựng sơ đồ bộ chuyển mã.

- **Mạch chuyển mã NBCD sang mã 7 vạch.**

Từ sự bố trí các vạch sáng của đèn chỉ thị 7 vạch, ta lập bảng chân lý, từ đó lập bảng Karnaugh cho từng hàm ra ứng với từng vạch và tối thiểu hóa:





Số	ABCD	a	b	c	d	e	f	g
0	0000	1	1	1	1	1	1	0
1	0001	0	1	1	0	0	0	0
2	0010	1	1	0	1	1	0	1
3	0011	1	1	1	1	0	0	1
4	0100	0	1	1	0	0	1	1
5	0101	1	0	1	1	0	1	1
6	0110	1	0	1	1	1	1	1
7	0111	1	1	1	0	0	0	0
8	1000	1	1	1	1	1	1	1
9	1001	1	1	1	1	0	1	1

a

CD \ AB	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	1	1	1	1
10	1	1	1	1

b

CD \ AB	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	1	1	1	1
10	1	1	1	1

c

CD \ AB	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$a = A + C + BD + \overline{B.D} = \overline{\overline{A.C.BD.B.D}}$$

$$b = A + \overline{B} + CD + \overline{\overline{C.D}} = \overline{\overline{\overline{A.B.CD.C.D}}}$$

$$c = A + B + \overline{C} + D = \overline{\overline{\overline{A.B.C.D}}}$$

d

CD \ AB	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	1	1	1	1
10	1	1	1	1

e

CD \ AB	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	1	1	1	1
10	1	1	1	0

f

CD \ AB	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	1	1	1	1
10	1	1	1	1

g

CD \ AB	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	1	1	1	1
10	1	1	1	1

1.5 CÁC VI MẠCH TỔ HỢP VÀ LƯU Ý KHI SỬ DỤNG.

1.5.1 MỘT SỐ KHÁI NIỆM CẦN BIẾT.

1.5.2 CÁC THÔNG SỐ KỸ THUẬT CỦA IC SỐ.

1.5.3 CÁC HỌ IC SỐ.

1.5.4 GIAO TIẾP GIỮA CÁC HỌ IC SỐ.

1.5.1 MỘT SỐ KHÁI NIỆM CẦN BIẾT

- **Tín hiệu tương tự và tín hiệu số**
 - Tín hiệu tương tự là tín hiệu có biên độ biến thiên liên tục theo thời gian.
 - Tín hiệu số là tín hiệu có biên độ gián đoạn về thời gian và được lượng tử hóa về mức. Trong các mạch điện, t/h số thường là điện áp hay dòng điện có một trong hai mức "cao" hoặc "thấp" trong từng khoảng thời gian nhất định.

- **Mạch tương tự và mạch số**

- Các mạch điện tử tạo hoặc xử lý tín hiệu tương tự được gọi là mạch tương tự.

- Các mạch điện tử tạo hoặc xử lý tín hiệu số được gọi là mạch số.

Các mạch số có thể được sản xuất với độ tích hợp ngày càng cao. Tùy theo số cổng logic trong một vi mạch, người ta phân loại vi mạch số như sau:

- + SSI-Small Scale Integrated, số cổng < 10 ;

+ MSI-Medium Scale Integrated

$10 < \text{số công} < 100;$

+ LSI-Large Scale Integrated

$100 < \text{số công} < 1000;$

+ VLSI-Very Large Scale Integrated

$1000 < \text{số công} < 10.000;$

+ ULSI-Ultra Large Scale Integrated

$\text{số công} > 10.000.$

- **Biểu diễn các trạng thái Logic 1 và 0**

Các mạch số có thể được quy ước logic dương hoặc logic âm. Trong các mạch với logic dương, điện thế cao biểu diễn logic 1, điện thế thấp biểu diễn logic 0. Trong các mạch với logic âm thì ngược lại. Ví dụ, các mạch logic họ TTL, mức điện thế từ 2,4 đến 5 V là mức logic 1, mức điện thế từ 0 đến 0,4 V là mức logic 0. Khoảng nằm giữa 0,4 đến 2,4 V là khoảng không xác định.

1.5.2 CÁC THÔNG SỐ KT CỦA IC SỐ

- Các đại lượng điện đặc trưng
- Công suất tiêu tán (Power requirement)
- Khả năng chịu tải (Fan-Out)
- Thời trễ truyền (Propagation delays)
- Tích số công suất-vận tốc (speed- power product)
- Tính miễn nhiễm (noise immunity)
- Logic cấp dòng và logic nhận dòng

- **Các đại lượng điện đặc trưng.**

- V_{CC} : Điện thế nguồn (power supply). Thí dụ, với IC số họ TTL, $V_{CC}=5\pm 0,5$ V , họ CMOS $V_{DD}=3-15$ V (Người ta thường dùng ký hiệu V_{DD} và V_{SS} để chỉ nguồn và mass của IC họ MOS) –

- $V_{IH}(\text{min})$: Điện thế đầu vào mức cao (High level input voltage) nhỏ nhất.

- $V_{IL}(\text{max})$: Điện thế đầu vào mức thấp (Low level input voltage) lớn nhất.

- $V_{OH}(\text{min})$: Điện thế đầu ra mức cao (High level output voltage) nhỏ nhất.

- $V_{OL}(\max)$: Điện thế đầu ra mức thấp (Low level output voltage) lớn nhất.
- I_{IH} : Dòng điện đầu vào mức cao (High level input current).
- I_{IL} : Dòng điện đầu vào mức thấp (Low level input current).
- I_{OH} : Dòng điện đầu ra mức cao (High level output current).
- I_{OL} : Dòng điện đầu ra mức thấp (Low level output current).
- I_{CCH}, I_{CCL} : Dòng điện tiêu thụ (chạy qua) của IC khi đầu ra lần lượt ở mức cao và thấp.

- **Công suất tiêu tán (Power requirement).**

Mỗi IC tiêu thụ một công suất từ nguồn V_{CC} (hay V_{DD}). Công suất tiêu tán này xác định bởi điện thế nguồn và dòng điện tiêu thụ của IC. Do dòng tiêu thụ thay đổi giữa hai trạng thái cao và thấp nên công suất tiêu tán sẽ được tính từ dòng trung bình và là công suất tiêu tán trung bình:

$$P(\text{avg}) = I_{CC}(\text{avg}) \cdot V_{CC} .$$

Trong đó: $I_{CC}(\text{avg}) = (I_{CCH} + I_{CCL}) : 2$.

Các cổng logic họ TTL có công suất tiêu tán ở hàng mW, còn họ MOS thì chỉ ở hàng \star W.

- **Khả năng chịu tải (Fan-Out).**

Khả năng này chỉ ra số đầu vào lớn nhất có thể nối với một đầu ra của mạch logic cùng loại, nói lên khả năng chịu tải của một mạch logic.

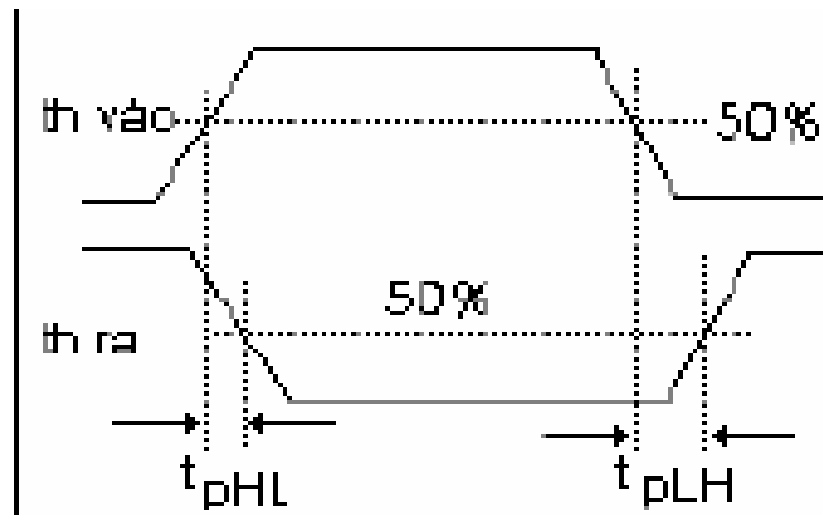
$$\text{Fan-Out}_H = I_{OH} / I_{IH} \text{ (UL-Unit Load);}$$

$$\text{Fan-Out}_L = I_{OL} / I_{IL} \text{ (UL);}$$

Khi sử dụng các vi mạch số, người ta dùng giá trị nhỏ nhất trong hai giá trị trên.

- **Thời trễ truyền (Propagation delays).**

Thời trễ truyền là khoảng thời gian giữ chậm của tín hiệu đầu ra so với tín hiệu đầu vào của một mạch logic. Có hai loại thời trễ truyền: thời trễ truyền từ thấp lên cao t_{PLH} và thời trễ truyền từ cao xuống thấp t_{PHL} . Tùy theo họ IC, thời trễ truyền thay đổi từ vài ns đến vài trăm ns. Thời trễ truyền càng lớn thì tốc độ làm việc của IC càng nhỏ.



- **Tích số công suất-vận tốc (speed- power product).**

Để đánh giá chất lượng IC, người ta dùng đại lượng tích số công suất-vận tốc đó là tích số công suất tiêu tán và thời trễ truyền. Thí dụ họ IC có thời trễ truyền là 10 ns và công suất tiêu tán trung bình là 50 mW thì tích số công suất-vận tốc là: $10 \text{ ns} \times 5 \text{ mW} = 10 \cdot 10^{-9} \times 5 \cdot 10^{-3} = 50 \times 10^{-12} \text{ watt-sec} = 50 \text{ picojoules (pj)}$. Một IC có chất lượng càng tốt khi tích số công suất-vận tốc càng nhỏ.

- **Tính miễn nhiễu (noise immunity)**

Tính miễn nhiễu của một mạch logic là khả năng chống nhiễu của mạch và được xác định bởi lề nhiễu. Lề nhiễu là sự chênh lệch của các điện thế $V_{OH}(\min)$ với $V_{IH}(\min)$ và $V_{OL}(\max)$ với $V_{IL}(\max)$ nên ta có 2 giá trị lề nhiễu:

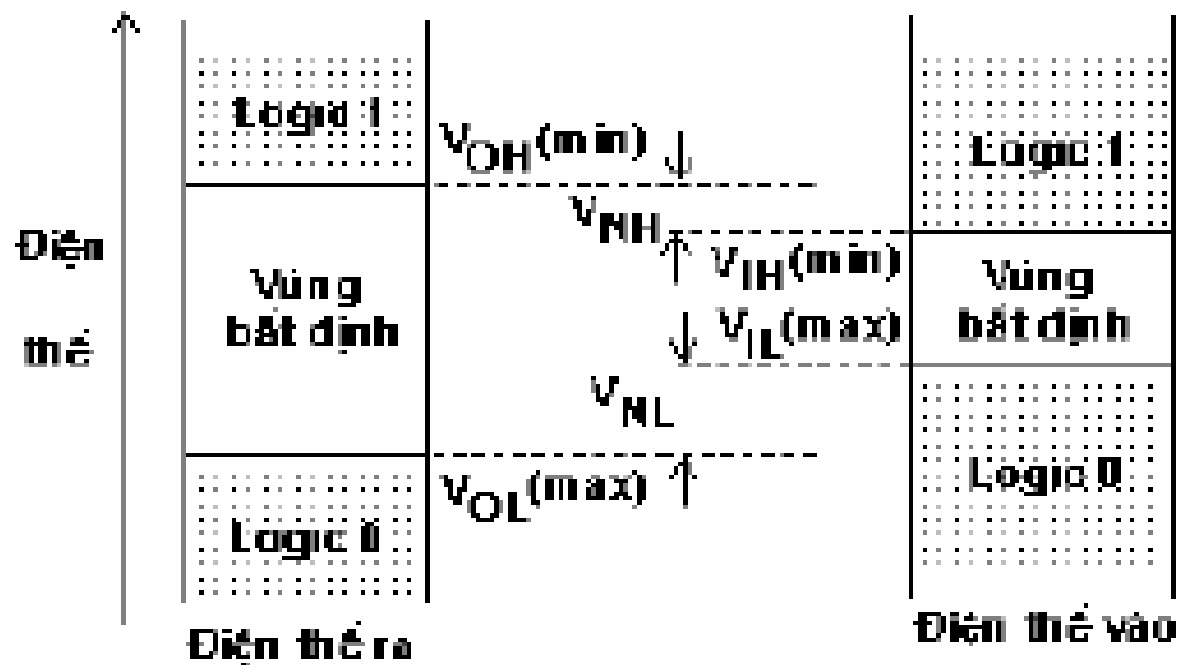
- Lề nhiễu mức cao:

$$V_{NH} = V_{OH}(\min) - V_{IH}(\min).$$

- Lề nhiễu mức thấp:

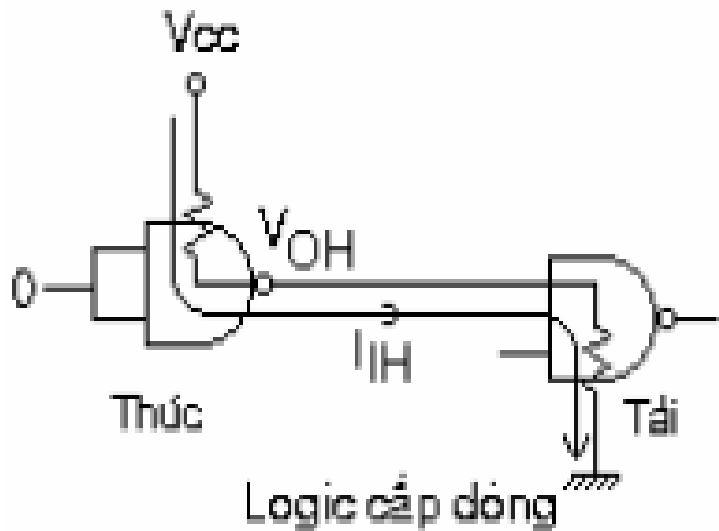
$$V_{NL} = V_{IL}(\max) - V_{OL}(\max).$$

Khi đầu vào có mức cao, nếu t/h nhiễu có giá trị âm và biên độ $>V_{NH}$ sẽ làm cho điện thế đầu vào rơi vào vùng bất định và mạch không nhận ra được t/h vào thuộc mức logic nào. Khi đầu vào ở mức thấp, tín hiệu nhiễu có trị dương và biên độ $>V_{NL}$ sẽ đưa mạch vào trạng thái bất định.

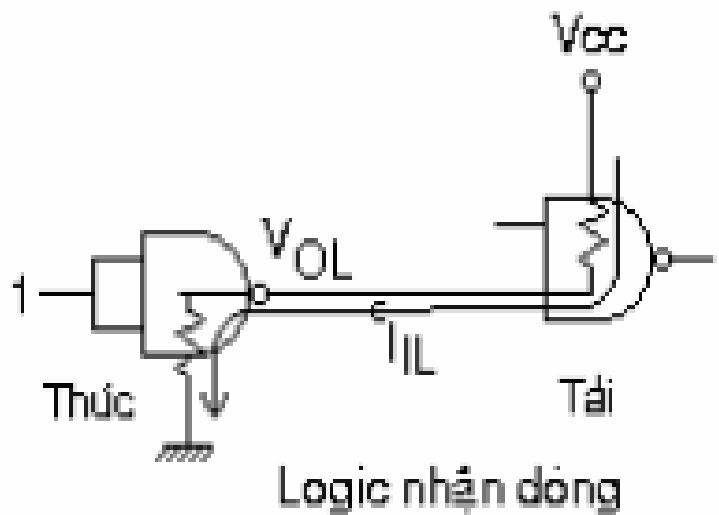


- **Logic cấp dòng và logic nhận dòng.**

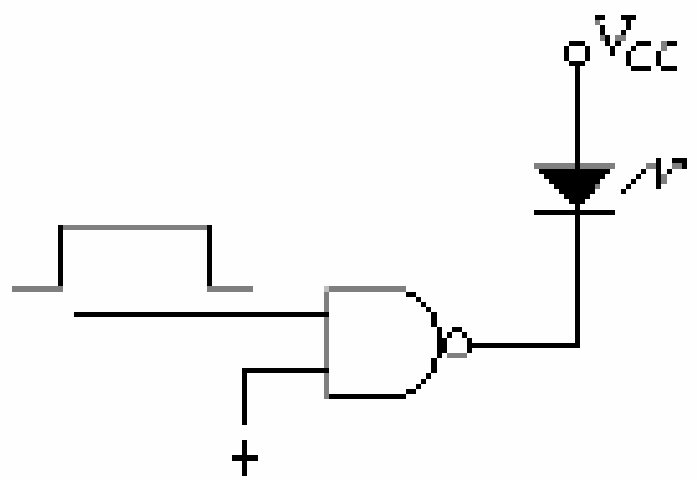
Một mạch logic thường gồm nhiều tầng kết nối với nhau. Sự trao đổi dòng điện giữa hai tầng: tầng cấp tín hiệu và tầng nhận tín hiệu (tầng tải) thể hiện bởi logic cấp dòng và logic nhận dòng. Khi đầu ra mạch logic 1 ở mức cao, nó cấp dòng I_{IH} cho đầu vào của mạch logic 2, vai trò như một tải nối mass nhận dòng. Khi đầu ra mạch logic 1 ở mức thấp, nó nhận dòng I_{IL} từ đầu vào của mạch logic 2 xem như nối với nguồn V_{CC}



(a)



(b)



1.5.3 CÁC HỌ IC SỐ

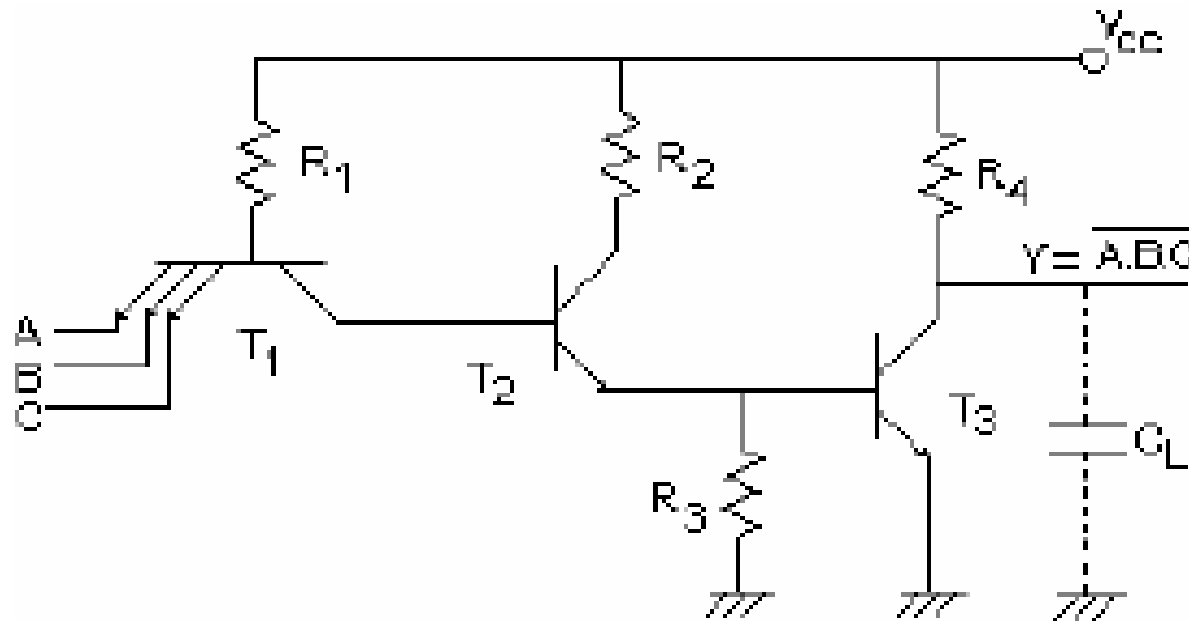
- Công nghệ chế tạo mạch số có các họ: RTL (Resistor-transistor logic), DCTL (Direct couple-transistor logic), RCTL (Resistor-Capacitor-transistor logic), DTL (Diod-transistor logic), ECL (Emitter- couple logic) v.v.....
- Hiện nay tồn tại hai họ có nhiều tính năng kỹ thuật cao như thời trễ truyền nhỏ, tiêu hao công suất ít là: TTL (transistor-transistor logic) và họ MOS (metal-oxyde-semiconductor).

HỘ TTL

- **Công cơ bản hộ TTL**
- **Các kiểu mạch ra**
- **Đặc tính các loạt TTL**

- **Cổng cơ bản họ TTL**

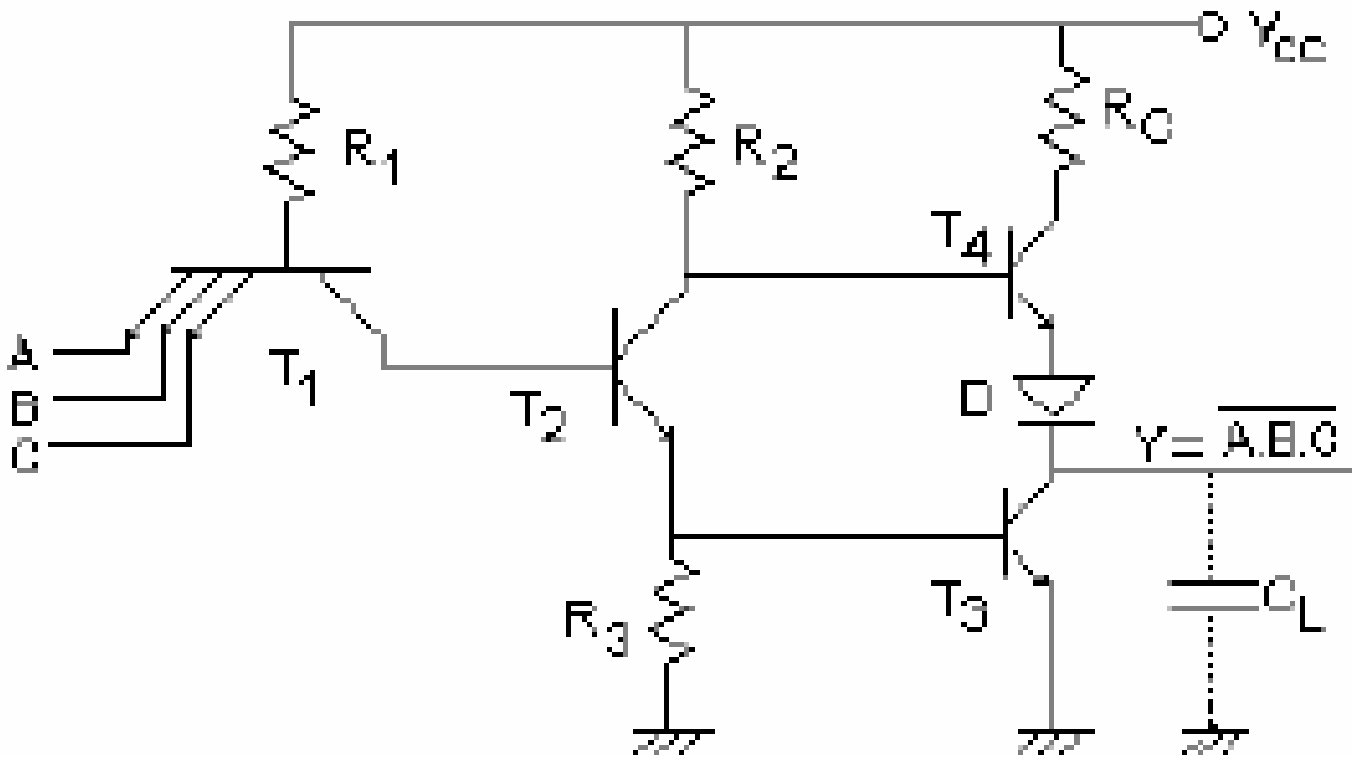
Xét cổng NAND 3 đầu vào. Khi một trong các đầu vào A, B, C xuống mức 0, T1 dẫn đưa đến T2 ngưng, T3 ngưng, đầu ra Y lên cao; khi cả 3 đầu vào lên cao, T1 ngưng, T2 dẫn, T3 dẫn, đầu ra Y xuống thấp.



Tụ C_L là tụ ký sinh tạo bởi sự kết hợp giữa đầu ra của tầng cấp với đầu vào của tầng tải, khi T3 ngưng, tụ nạp điện qua R4 và khi T3 dẫn, tụ phóng qua T3 do đó thời trễ truyền của mạch quyết định bởi R4 và C_L . Khi R4 nhỏ mạch hoạt động nhanh nhưng công suất tiêu thụ lớn, muốn giảm công suất phải tăng R4 nhưng như vậy thời trễ truyền sẽ lớn. Để giải quyết khuyết điểm này đồng thời thỏa mãn một số yêu cầu khác, người ta đã chế tạo các cổng logic với các kiểu mạch ra khác nhau.

- **Các kiểu mạch ra.**

- *Mạch ra Totempole*: R4 được thay thế bởi cụm T4, R_C và diod D, trong đó R_C có trị rất nhỏ, không đáng kể. T2 bây giờ giữ vai trò mạch đảo pha: khi T2 dẫn thì T3 dẫn và T4 ngưng, đầu ra Y xuống thấp; khi T2 ngưng thì T3 ngưng và T4 dẫn, đầu ra Y lên cao. Tụ C_L nạp điện qua T4 khi T4 dẫn và phóng qua T3, thời hằng mạch rất nhỏ và kết quả là thời trễ truyền nhỏ. Do T3 & T4 luân phiên ngưng tương ứng với 2 trạng thái của đầu ra nên công suất tiêu thụ giảm đáng kể. Diod D có tác dụng nâng điện thế cực B của T4 lên để bảo đảm khi T3 dẫn thì T4 ngưng.

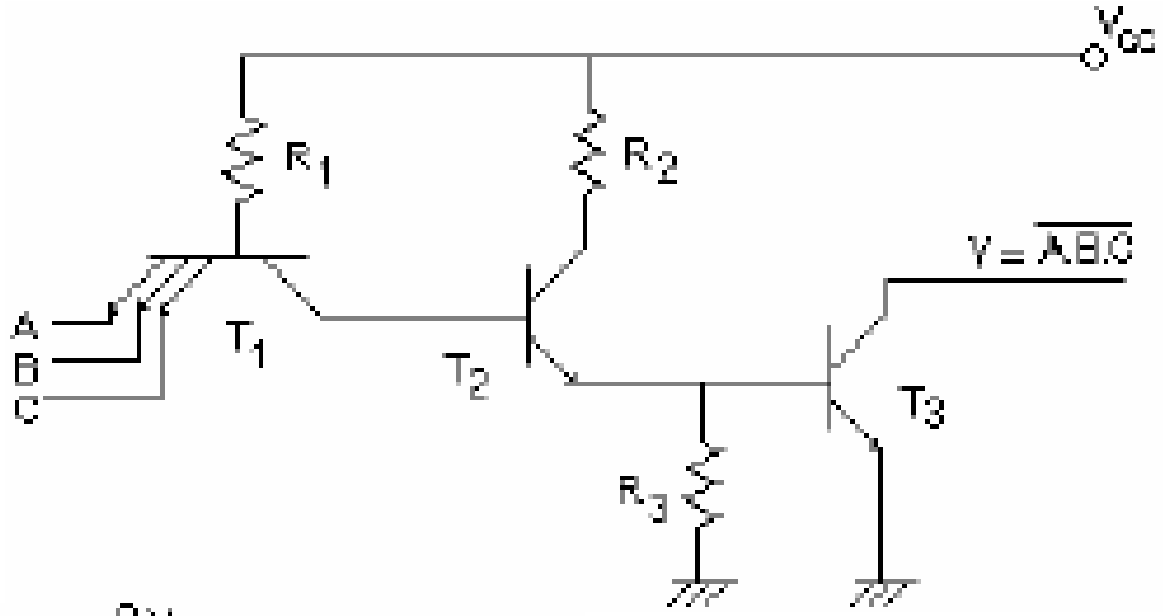
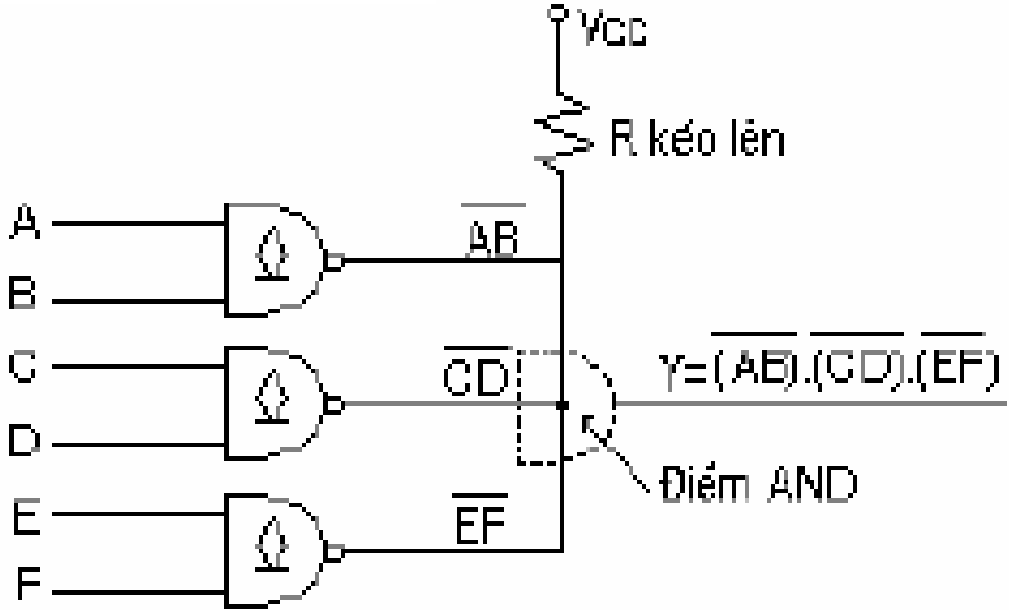


- *Mạch ra cực góp hở:*

+ Cho phép kết nối các đầu ra của nhiều cổng khác nhau, nhưng phải mắc một điện trở từ đầu ra lên nguồn Vcc, gọi là **điện trở kéo lên**, trị số của điện trở này có thể lớn hay nhỏ tùy theo yêu cầu có lợi về mặt công suất hay tốc độ làm việc.

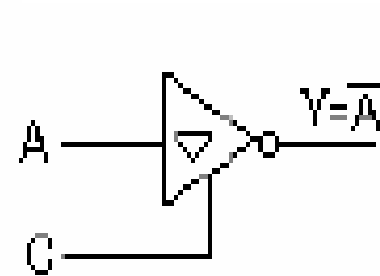
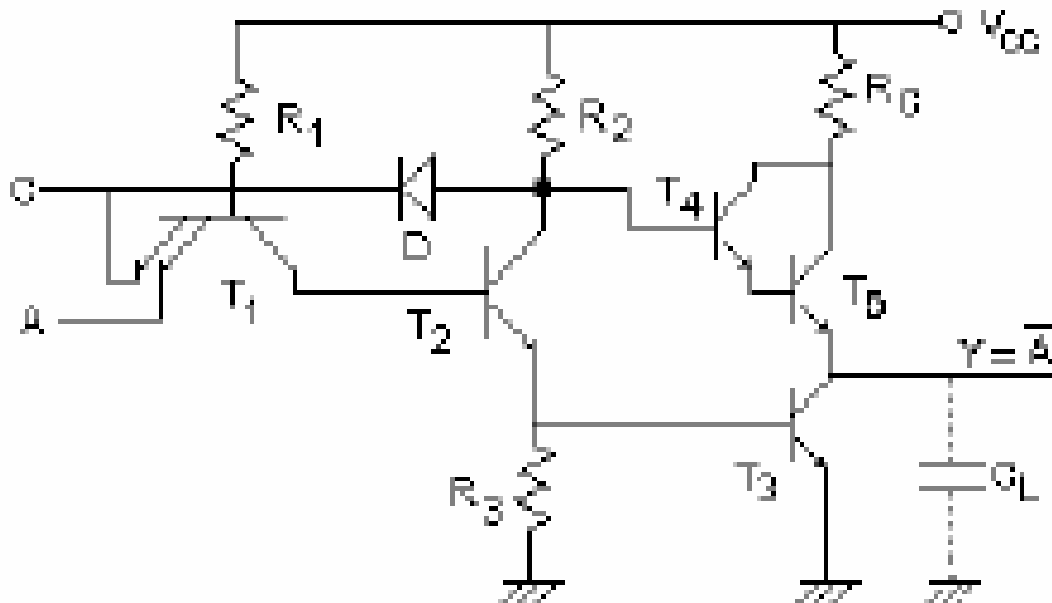
+ Điểm nối chung của các đầu ra có tác dụng như một cổng AND nên ta gọi là điểm AND.

+ Cho phép điện trở kéo lên mắc vào nguồn điện thế cao, dùng cho các tải đặc biệt hoặc dùng tạo sự giao tiếp giữa họ TTL với CMOS dùng nguồn cao.



- Mạch ra ba trạng thái:

Xét công đảo có đầu ra 3 trạng thái, trong đó T4 & T5 được mắc Darlington để cấp dòng ra lớn cho tải. Diode D để điều khiển: khi C=1, diode D ngưng dẫn, mạch hoạt động như một công đảo; khi C=0, diode D dẫn, cực góp T2 bị ghim áp ở mức thấp nên T3, T4 & T5 đều ngắt, đầu ra mạch ở trạng thái trở kháng cao.



C	A	Y
1	0	1
1	1	0
0	X	Z cao

Bảng sự thật

- **Đặc tính các loạt TTL.**

- Ký hiệu chung: 74XXXX và 54XXXX.

- 74: $V_{CC}=5 \pm 0,5$ V, nhiệt độ 0° C đến 70° C

- 54: $V_{CC}=5 \pm 0,25$ V, nhiệt độ -55° C đến 125° C

- Các ký hiệu riêng: sau 74 (54) không có chữ gì là IC chuẩn, L: Low power, H: High speed, S: Schottky, LS: Low power Schottky, AS: Advance Schottky, 74ALS (Advance Low power Schottky), 74F (Fast)

Thông số kỹ thuật	74	74L	74H	74S	74LS	74AS	74ALS	74F
Thời trễ truyền (ns)	9	33	6	3	9,5	1,7	4	3
Công suất tiêu tán (mW)	10	1	23	20	2	8	1,2	6
Tích số công suất vận tốc (pJ)	90	33	138	60	19	13,6	4,8	18
Tần số xung C_K max (MHz)	35	3	50	125	45	200	70	100
Fan Out (cùng loạt)	10	20	10	20	20	40	20	33
Điện thế								
V_{OH} (min)	2,4	2,4	2,4	2,7	2,7	2,5	2,5	2,5
V_{OL} (max)	0,4	0,4	0,4	0,5	0,5	0,5	0,4	0,5
V_{IH} (min)	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0
V_{IL} (max)	0,8	0,7	0,8	0,8	0,8	0,8	0,8	0,8

HỌ MOS

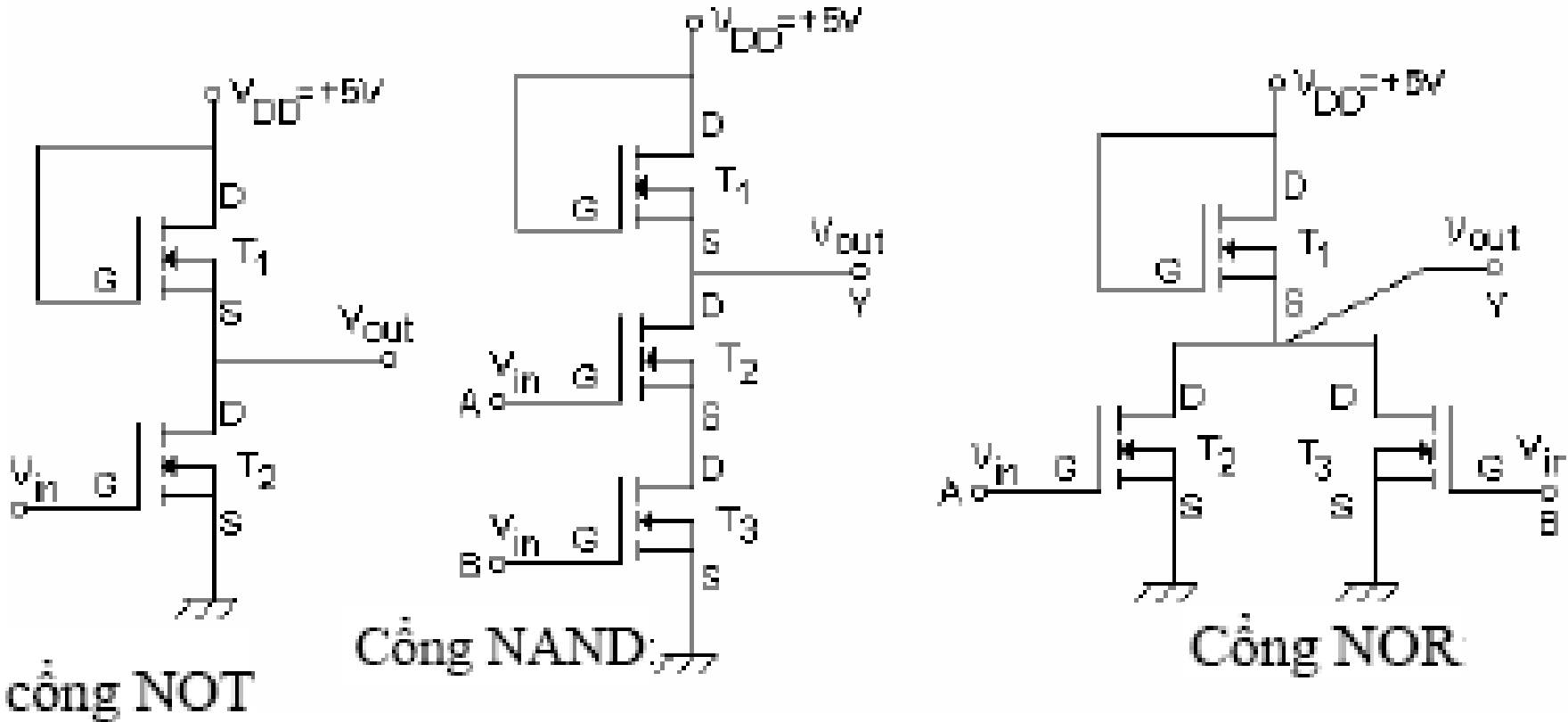
- **Cổng cơ bản NMOS**
- **Cổng cơ bản CMOS (complementary MOS)**
- **Các cổng CMOS khác**
- **Đặc tính của họ MOS**
- **Các loại CMOS**

Công nghệ MOSFET, kênh N (NMOS) và kênh P (PMOS) hoặc cả hai loại kênh P&N (CMOS). NMOS và PMOS chỉ khác nhau là ngược chiều nguồn cấp.

Các transistor MOS:

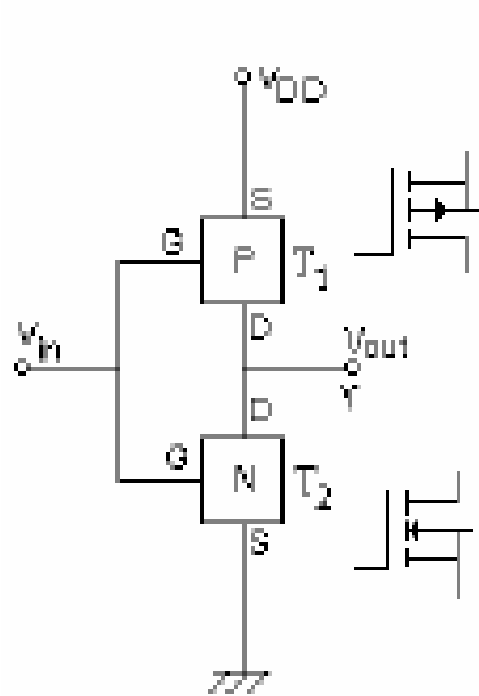
- Khi dẫn, nội trở nhỏ (vài chục Ω đến trăm $k\Omega$).
- Khi ngưng, nội trở rất lớn (hàng $10^{10}\Omega$), tương đương với một khóa hở.

• Cổng cơ bản NMOS

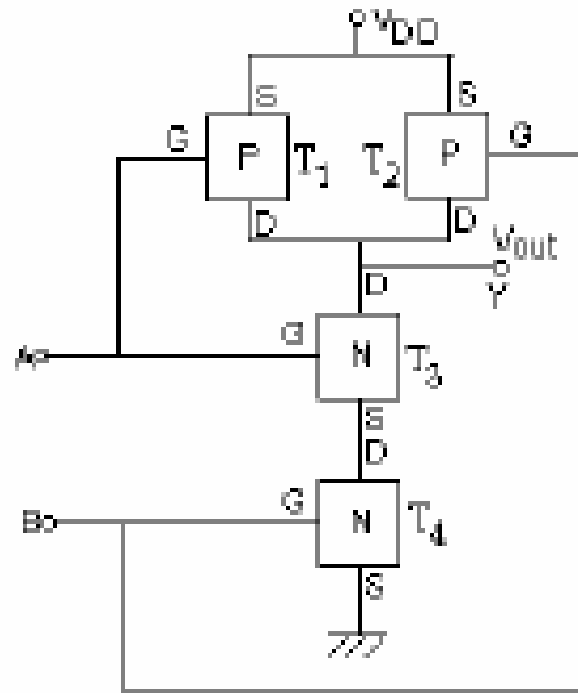


V_{in}	T_1	T_2	V_{out}
0V (logic 0)	$R_{ON} = 100K\Omega$	$R_{OFF} = 10^{10}\Omega$	+5V (logic 1)
+5V (logic 1)	$R_{ON} = 100K\Omega$	$R_{ON} = 1K\Omega$	0,05V (logic 0)

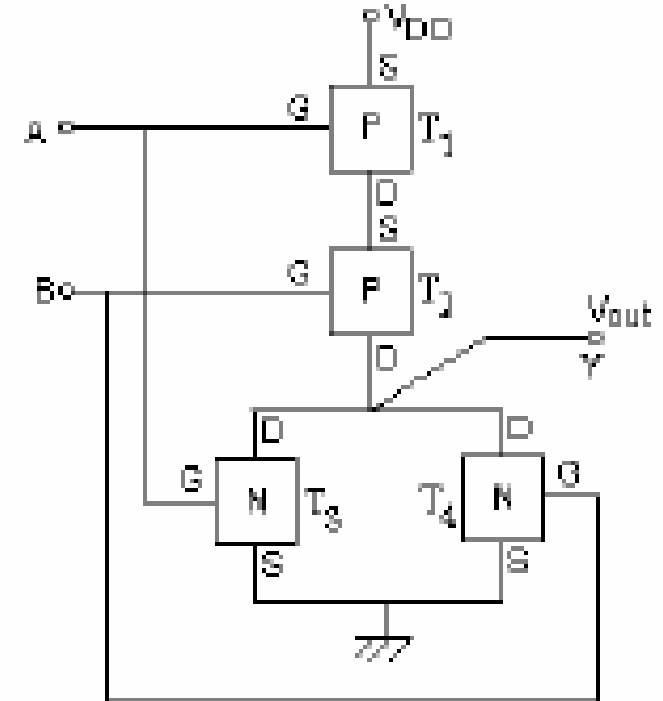
• Cổng cơ bản CMOS



cổng NOT



Cổng NAND

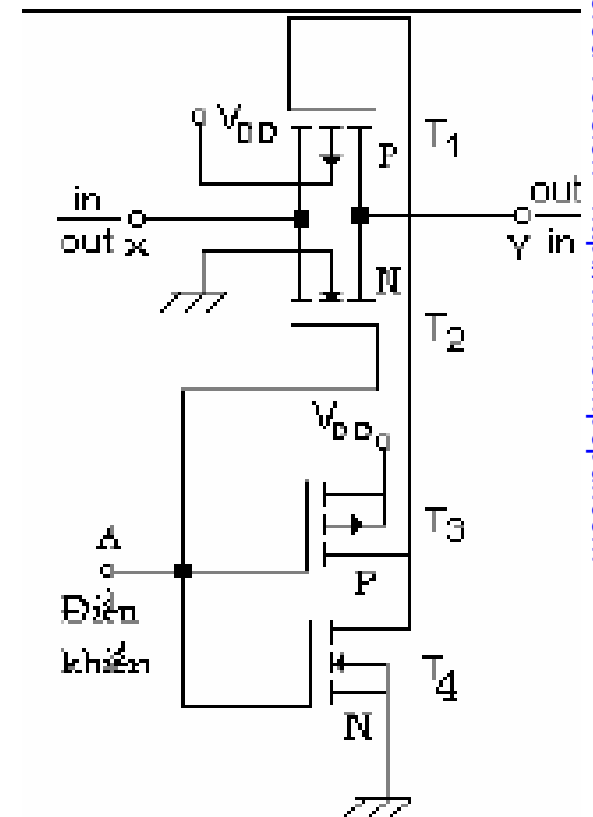
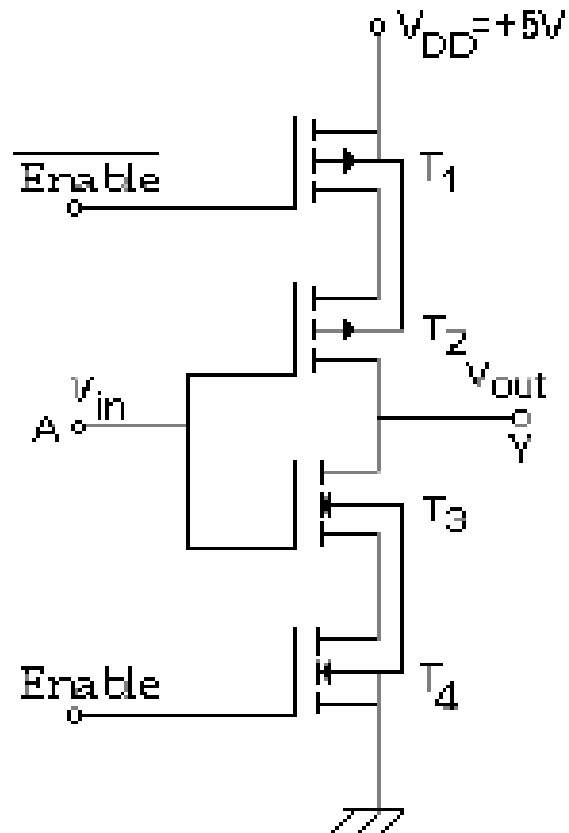
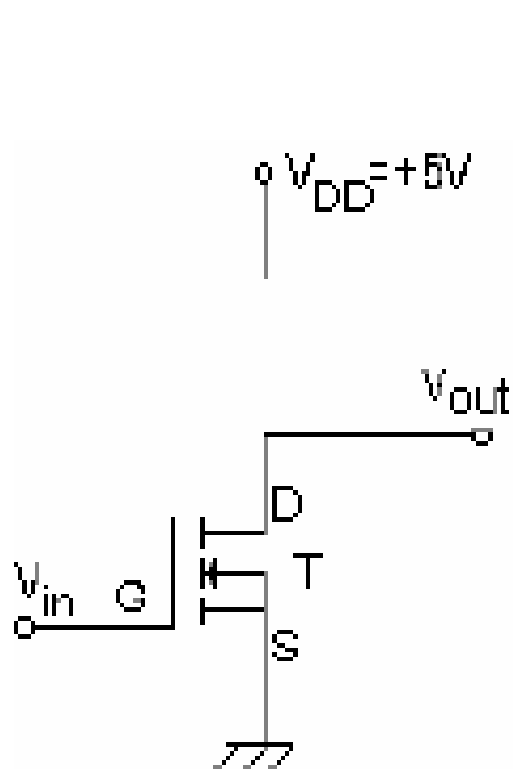


Cổng NOR

V_{in}	T_1	T_2	V_{out}
V_{DD} (logic 1)	$R_{OFF} = 10^{10} \Omega$	$R_{ON} = 1K\Omega$	0V (logic 0)
0V (logic 0)	$R_{ON} = 1K\Omega$	$R_{OFF} = 10^{10} \Omega$	V_{DD} (logic 1)

- **Các cổng CMOS khác**

- Cổng CMOS với cực máng (drain) hở;
- Cổng CMOS với đầu ra 3 trạng thái;
- Khóa-mở t/h 2 chiều.



- **Đặc tính của họ MOS**

- Nguồn: $V_{DD} = 3 - 15V$

- Mức logic: $V_{OL}(\max) = 0V$ $V_{OH}(\min) = V_{DD}$

- $V_{IL}(\max) = 30\% V_{DD}$ $V_{IH}(\min) = 70\% V_{DD}$

- Lệ nhiễu: $V_{NH} = 30\% V_{DD}$ $V_{NL} = 30\% V_{DD}$

- Thời trễ truyền: vài chục ns

- Công suất tiêu tán: hàng nW

- Fan Out: 50 UL

- Mật độ tích hợp cao, thích hợp cho LSI, VLSI

- **Các loạt CMOS**

CMOS có hai ký hiệu: 4XXX do hãng RCA chế tạo và 14XXX của hãng MOTOROLA, có hai loạt 4XXXA (14XXXA) và 4XXXB (14XXXB), loạt B ra đời sau có cải thiện dòng ra. Ngoài ra còn có các loạt :

- 74CXXX: có sơ đồ chân, chức năng và các thông số như TTL.
- 74HC (High speed CMOS), 74HCT: loạt cải tiến của 74C, tốc độ như 74LS, riêng 74HCT thì hoàn toàn tương thích với TTL, được dùng rộng rãi.
- 74AC và 74ACT (Advance CMOS) cải tiến của 74HC và HCT về mặt nhiều bằng cách sắp xếp lại thứ tự các chân, do đó không tương thích với TTL về sơ đồ chân.

1.5.4 GIAO TIẾP GIỮA CÁC HỌ IC SỐ

- **TTL tải CMOS**
- **CMOS tải TTL**

Giao tiếp là thực hiện kết nối đầu ra của một mạch hay hệ thống với đầu vào của mạch hay hệ thống khác. Do tính chất về điện khác nhau giữa hai họ TTL và CMOS nên việc giao tiếp giữa chúng trong nhiều trường hợp không thể nối đơn giản được mà phải nhờ mạch phụ sao cho điện thế tín hiệu ra ở tầng cấp phù hợp với tín hiệu vào của tầng tải và dòng điện tầng cấp phải lớn hơn hoặc bằng dòng vào của tầng tải.

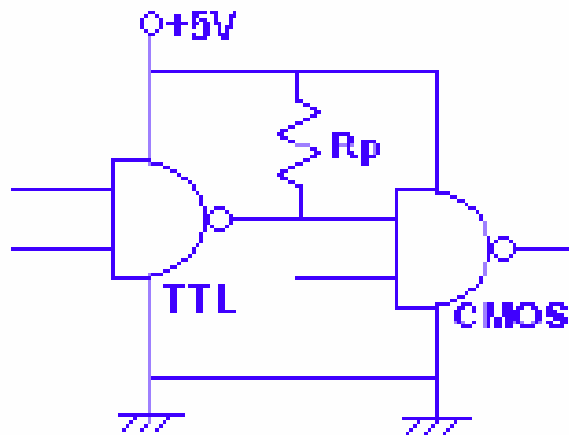
Bảng kê các thông số của 2 họ IC:

CMOS ($V_{DD} = 5V$)				TTL			
Thông số	4000B	74HC	74HCT	74	74LS	74AS	74ALS
$V_{IH}(\text{min})$	3,5V	3,5V	2,0V	2,0V	2,0V	2,0V	2,0V
$V_{IL}(\text{max})$	1,5V	1,0V	0,8V	0,8V	0,8V	0,8V	0,8V
$V_{OH}(\text{min})$	4,95V	4,9V	4,9V	2,4V	2,7V	2,7V	2,7V
$V_{OL}(\text{max})$	0,05V	0,1V	0,1V	0,4V	0,5V	0,5V	0,4V
$I_{IH}(\text{max})$	1 μ A	1 μ A	1 μ A	40 μ A	20 μ A	200 μ A	20 μ A
$I_{IL}(\text{max})$	1 μ A	1 μ A	1 μ A	1,6 mA	0,4 mA	2 mA	100 μ A
$I_{OH}(\text{max})$	0,4 mA	4 mA	4 mA	0,4 mA	0,4 mA	2 mA	0,4 mA
$I_{OL}(\text{max})$	0,4 mA	4 mA	4 mA	16 mA	8 mA	20 mA	8 mA

- **TTL tải CMOS**

- *TTL tải CMOS dùng $V_{DD} = 5V$.*

Từ bảng, dòng điện vào của CMOS có trị rất nhỏ so với dòng ra của TTL, vậy về dòng điện không có vấn đề. Điện thế ra của TTL $V_{OH}(\text{max})$ thấp so với $V_{IH}(\text{min})$ của CMOS, như vậy phải nâng điện thế ra của TTL lên bằng cách mắc thêm một điện trở kéo lên ở đầu ra của IC TTL.

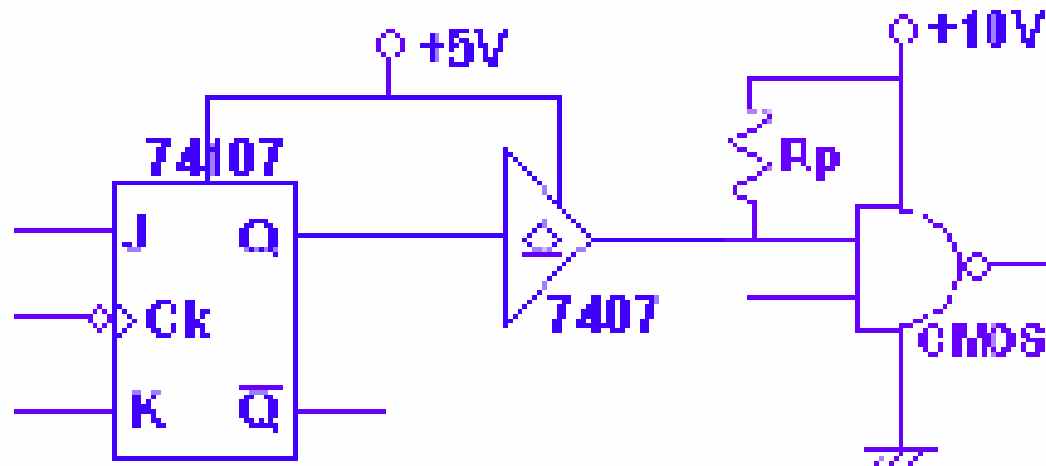


- *TTL tải CMOS 74HCT.*

74HCT là loại CMOS tương thích với TTL, có thể kết nối trực tiếp không cần điện trở kéo lên.

- *TTL tải CMOS dùng nguồn cao ($V_{DD} > +5V$)*

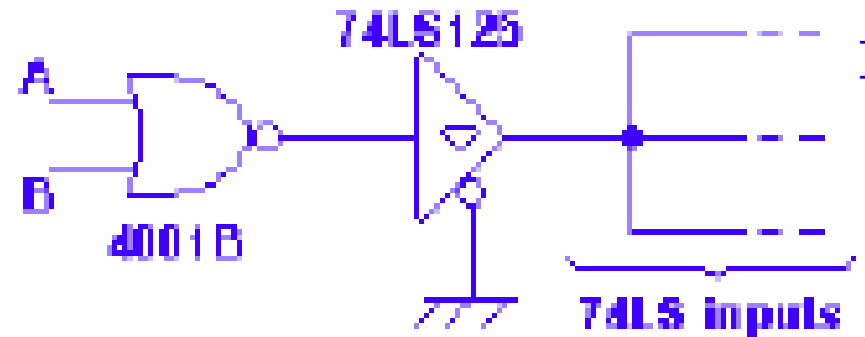
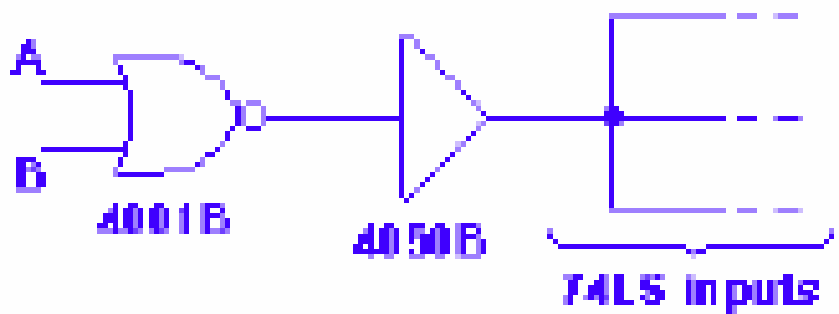
Dùng một cổng đệm có đầu ra hở mạch (7407), với điện trở kéo lên nguồn cao để thực hiện sự giao tiếp:



- **CMOS tải TTL**

- *CMOS (74HC, 74HCT) dùng nguồn 5V và tương thích TTL: có thể cấp dòng (4mA) cho 2 cổng TTL.*

- *CMOS loạt 4000B dùng nguồn 5V: dòng ra nhỏ (0,4mA), phải nối qua cổng đệm (4050B, 74LS125) để nâng dòng.*



- *CMOS dùng nguồn cao tải TTL:*

Dùng cổng đệm để hạ điện thế ra xuống, đồng thời cấp đủ dòng cho phù hợp với IC TTL.

