

Chương 2

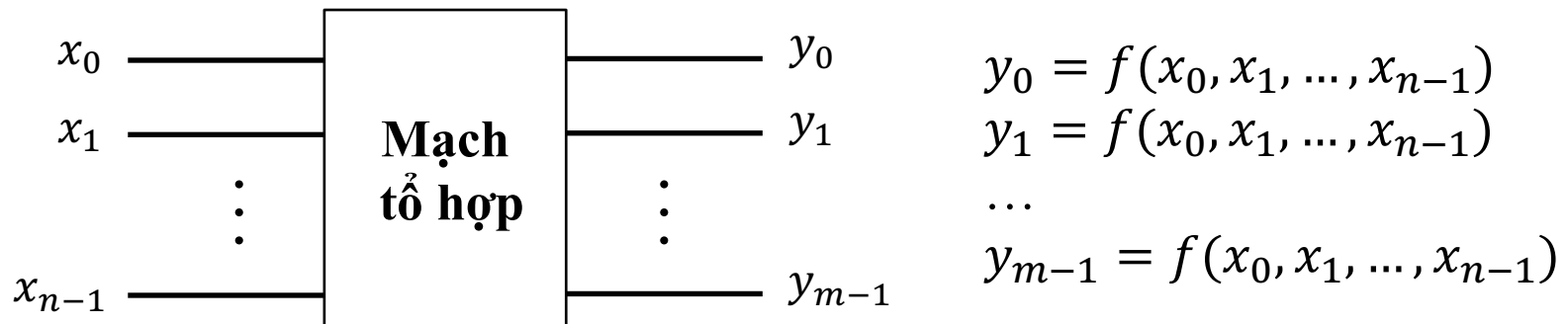
Mạch tổ hợp

Nội dung chương 2

- Khái niệm chung
- Mạch mã hóa - giải mã
- Mạch ghép kênh - phân kênh
- Mạch so sánh
- Mạch số học (cộng, trừ)

Khái niệm chung

- Mạch tổ hợp: ngõ ra là các hàm logic theo ngõ vào
- Các ngõ ra thay đổi trạng thái ngay khi các ngõ vào thay đổi trạng thái (nếu bỏ qua thời gian trễ của các phần tử logic)
- Tín hiệu ra tại mỗi thời điểm chỉ phụ thuộc vào giá trị các tín hiệu vào tại thời điểm đó
- Xét mạch tổ hợp có n ngõ vào và m ngõ ra:

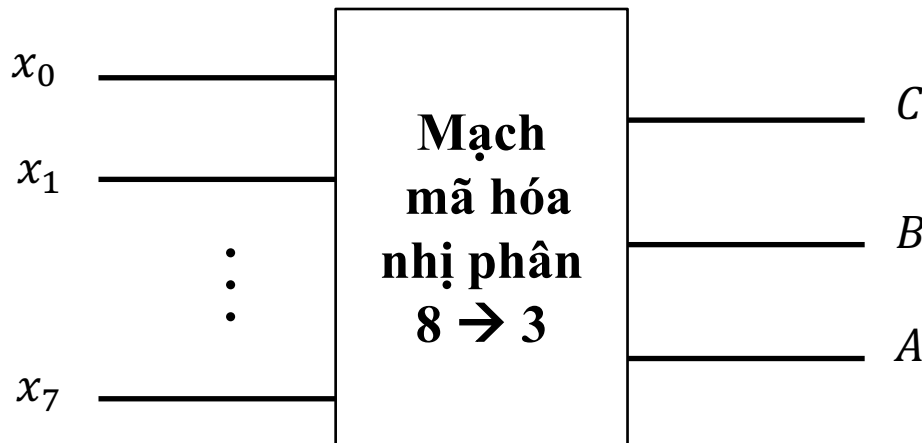


Khái niệm chung (tt)

- Thiết kế mạch tổ hợp:
 - Lập bảng trạng mô tả hoạt động của mạch
 - Lập hàm logic ngõ ra theo ngõ vào
 - Lập sơ đồ logic hệ tổ hợp
- Một số mạch tổ hợp cụ thể:
 - Mạch mã hóa – giải mã
 - Mạch ghép kênh – phân kênh
 - Mạch so sánh
 - Mạch số học

Mạch mã hóa nhị phân

- Mạch mã hóa nhị phân biến đổi tín hiệu ngõ vào thành một từ mã nhị phân tương ứng ở ngõ ra
- Xét mạch mã hóa nhị phân từ 8 sang 3 (8 ngõ vào, 3 ngõ ra):



Mạch mã hóa nhị phân

- Mạch mã hóa nhị phân từ 8 sang 3:
 - Bảng trạng thái (khi ngõ vào tác động mức 1):

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	C	B	A
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

- Phương trình logic:

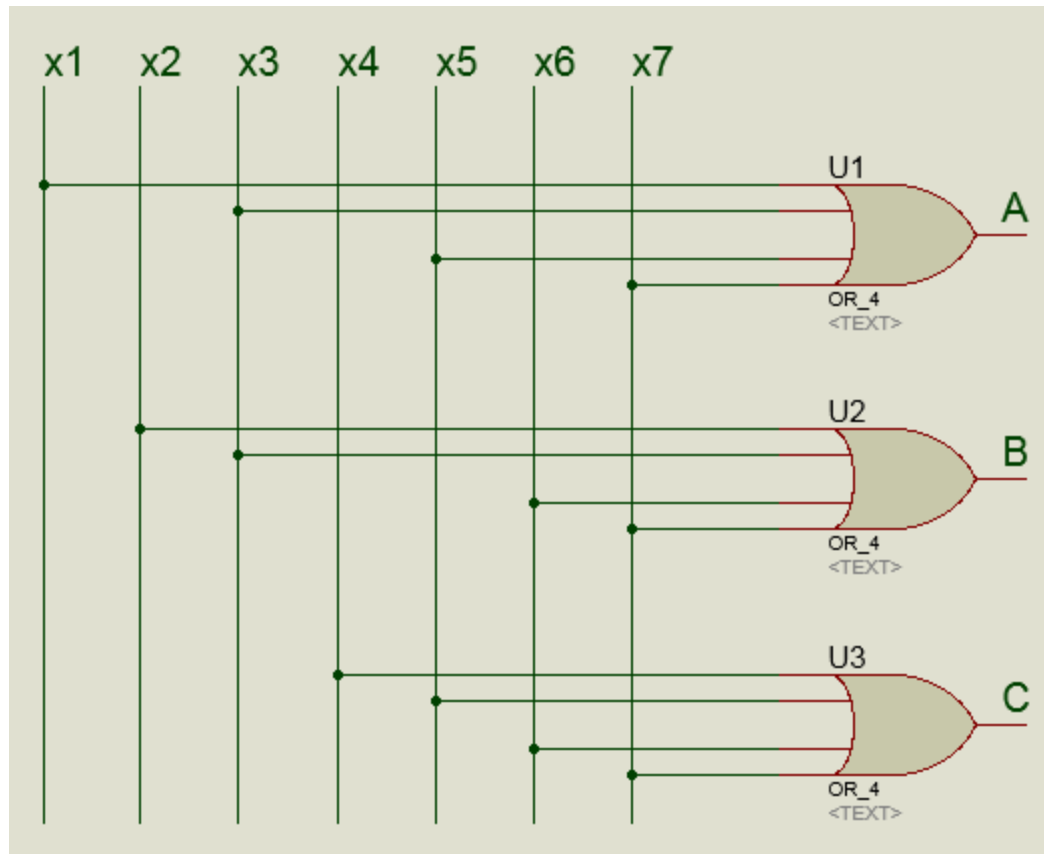
$$A = x_1 + x_3 + x_5 + x_7$$

$$B = x_2 + x_3 + x_6 + x_7$$

$$C = x_4 + x_5 + x_6 + x_7$$

Mạch mã hóa nhị phân (tt)

- Mạch mã hóa nhị phân từ 8 sang 3:
 - Sơ đồ logic (khi ngõ vào tác động mức 1):



Mạch mã hóa nhị phân (tt)

- Mạch mã hóa nhị phân từ 8 sang 3:
 - Bảng trạng thái (khi ngõ vào tác động mức 0):

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	C	B	A
0	1	1	1	1	1	1	1	0	0	0
1	0	1	1	1	1	1	1	0	0	1
1	1	0	1	1	1	1	1	0	1	0
1	1	1	0	1	1	1	1	0	1	1
1	1	1	1	0	1	1	1	1	0	0
1	1	1	1	1	0	1	1	1	0	1
1	1	1	1	1	1	0	1	1	1	0
1	1	1	1	1	1	1	0	1	1	1

- Phương trình logic:

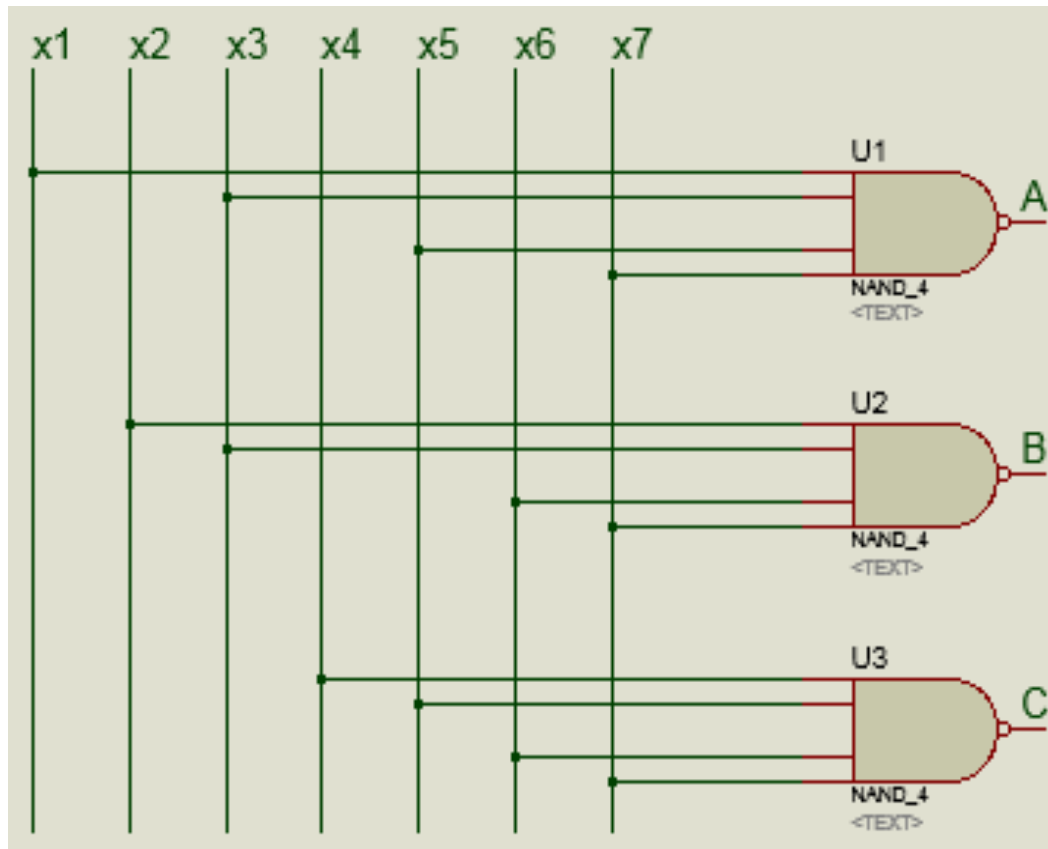
$$A = \overline{x_1} + \overline{x_3} + \overline{x_5} + \overline{x_7} = \overline{x_1 x_3 x_5 x_7}$$

$$B = \overline{x_2} + \overline{x_3} + \overline{x_6} + \overline{x_7} = \overline{x_2 x_3 x_6 x_7}$$

$$C = \overline{x_4} + \overline{x_5} + \overline{x_6} + \overline{x_7} = \overline{x_4 x_5 x_6 x_7}$$

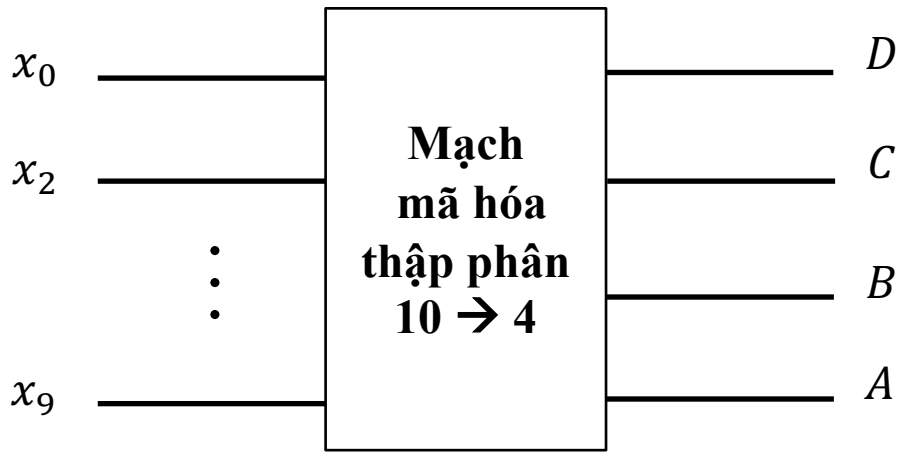
Mạch mã hóa nhị phân (tt)

- Mạch mã hóa nhị phân từ 8 sang 3:
 - Sơ đồ logic (khi ngõ vào tác động mức 0):



Mạch mã hóa thập phân

- Mạch mã hóa từ 10 sang 4



Mạch mã hóa thập phân (tt)

- Mạch mã hóa từ 10 sang 4 (10 ngõ vào, 4 ngõ ra)
 - Bảng trạng thái (khi ngõ vào tác động mức 1):

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	D	C	B	A
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

Mạch mã hóa thập phân (tt)

- Mạch mã hóa từ 10 sang 4

- Phương trình logic:

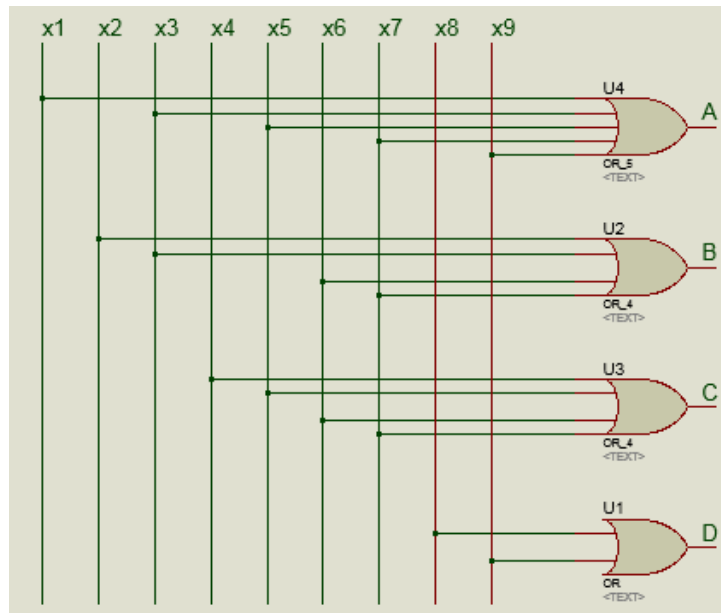
$$A = x_1 + x_3 + x_5 + x_7 + x_9$$

$$B = x_2 + x_3 + x_6 + x_7$$

$$C = x_4 + x_5 + x_6 + x_7$$

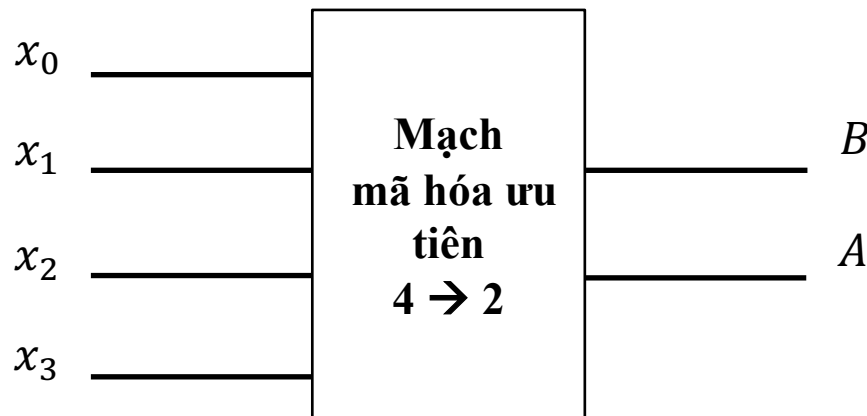
$$D = x_8 + x_9$$

- Sơ đồ logic:



Mạch mã hóa ưu tiên

- Khi có nhiều tín hiệu cùng tác động, mạch sẽ tạo ra từ mã nhị phân ứng với ngõ vào có mức độ ưu tiên cao nhất
- Xét mạch mã hóa ưu tiên từ 4 sang 2 (4 ngõ vào, 2 ngõ ra):



Mạch mã hóa ưu tiên (tt)

- Mạch mã hóa ưu tiên từ 4 sang 2
 - Bảng trạng thái (khi ngõ vào tác động mức 1, độ ưu tiên $x_0 < x_1 < x_2 < x_3$):

x_0	x_1	x_2	x_3	B	A
1	0	0	0	0	0
X	1	0	0	0	1
X	X	1	0	1	0
X	X	X	1	1	1

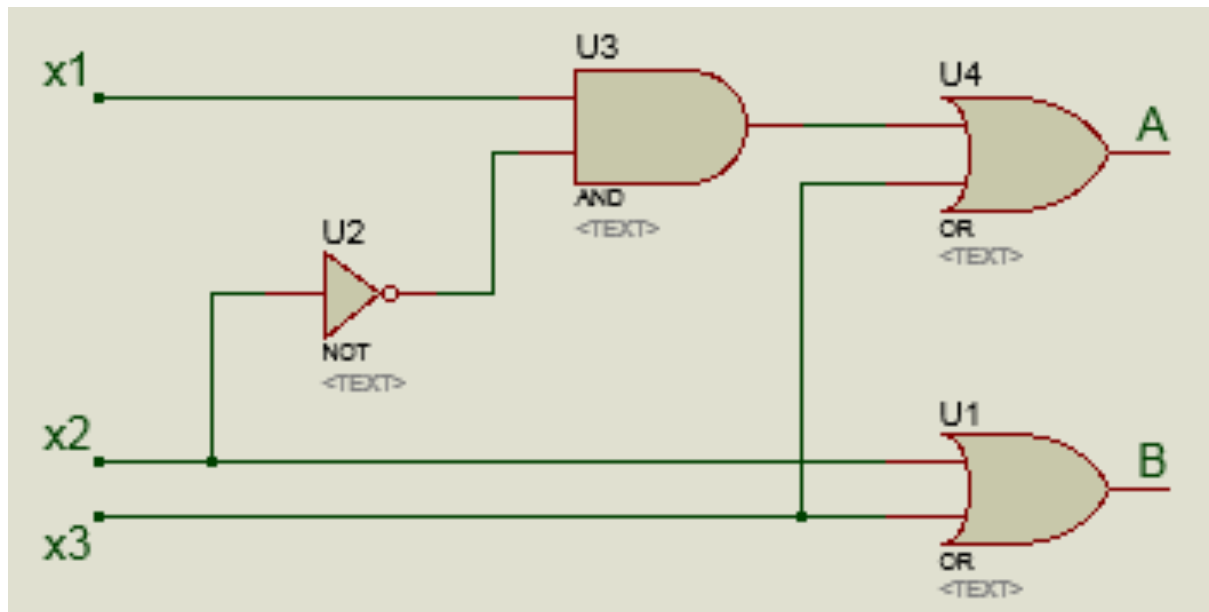
- Phương trình logic:

$$A = x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_3 = x_1 \cdot \overline{x_2} + x_3$$

$$B = x_2 \cdot \overline{x_3} + x_3 = x_2 + x_3$$

Mạch mã hóa ưu tiên (tt)

- Mạch mã hóa ưu tiên từ 4 sang 2:
 - Sơ đồ logic:

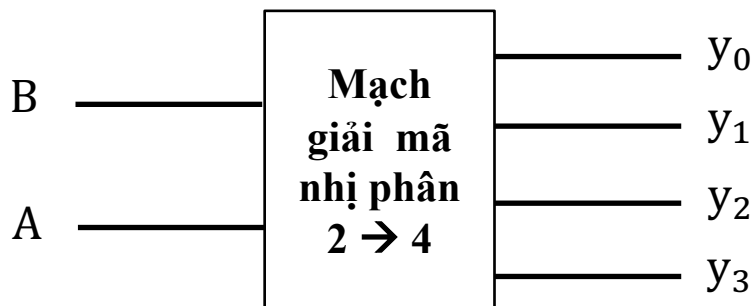


Mạch giải mã

- Ứng với mỗi tổ hợp nhị phân của ngõ vào, chỉ có một ngõ ra ở trạng thái tích cực, các ngõ ra còn lại ở trạng thái không tích cực (trừ một số mạch như BCD – LED 7 đoạn)
- Có 2 dạng tích cực:
 - Tích cực mức cao (mức 1)
 - Tích cực mức thấp (mức 0)
- Mạch giải mã thường gặp:
 - n sang 2^n (2 sang 4, 3 sang 8,...)
 - BCD sang thập phân, BCD sang LED 7 đoạn

Mạch giải mã nhị phân 2 sang 4

- Sơ đồ khối:



- Phương trình logic:

$$y_0 = \bar{B} \cdot \bar{A}$$

$$y_1 = \bar{B} \cdot A$$

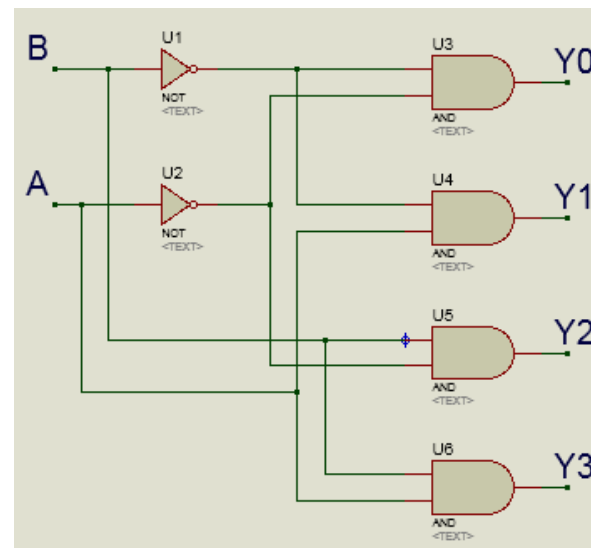
$$y_2 = B \cdot \bar{A}$$

$$y_3 = B \cdot A$$

- Bảng trạng thái:
(Ngõ ra tích cực mức 1)

B	A	y ₀	y ₁	y ₂	y ₃
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

- Sơ đồ logic:



Mạch giải mã nhị phân 2 sang 4 (tt)

- Bảng trạng thái:

(Ngõ ra tích cực mức 0)

B	A	y ₀	y ₁	y ₂	y ₃
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

- Phương trình logic:

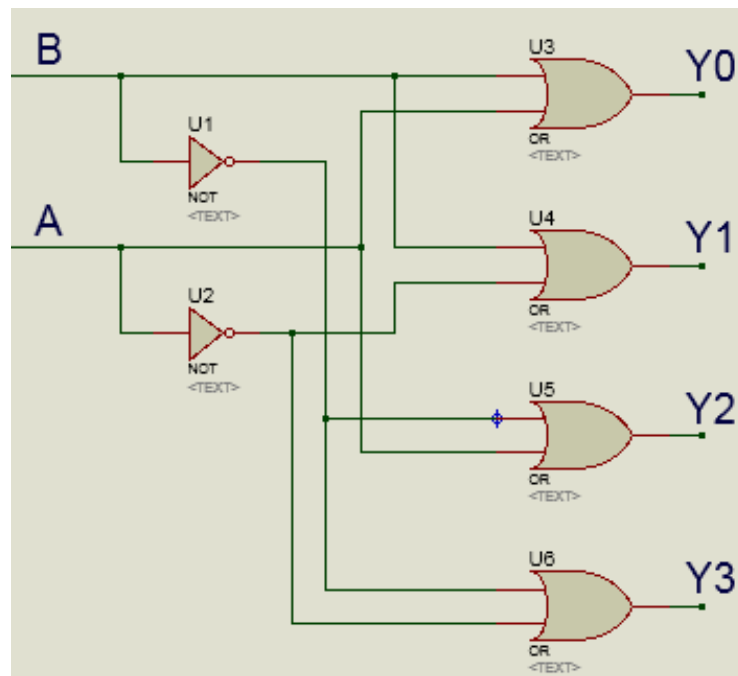
$$y_0 = B + A$$

$$y_1 = B + \bar{A}$$

$$y_2 = \bar{B} + A$$

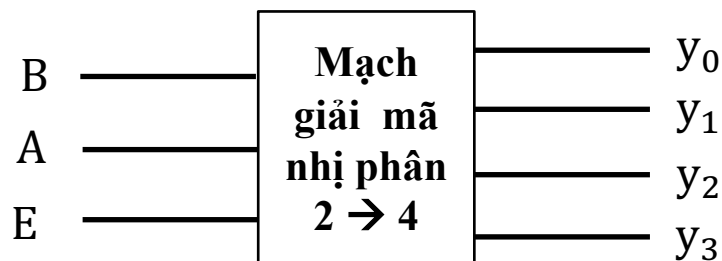
$$y_3 = \bar{B} + \bar{A}$$

- Sơ đồ logic:



Mạch giải mã nhị phân 2 sang 4 (tt)

- Sơ đồ khối: Thêm Enable (E)



- Bảng trạng thái:

(E và ngõ ra tích cực mức 1)

B	A	E	y ₀	y ₁	y ₂	y ₃
X	X	0	0	0	0	0
0	0	1	1	0	0	0
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	1

- Phương trình logic:

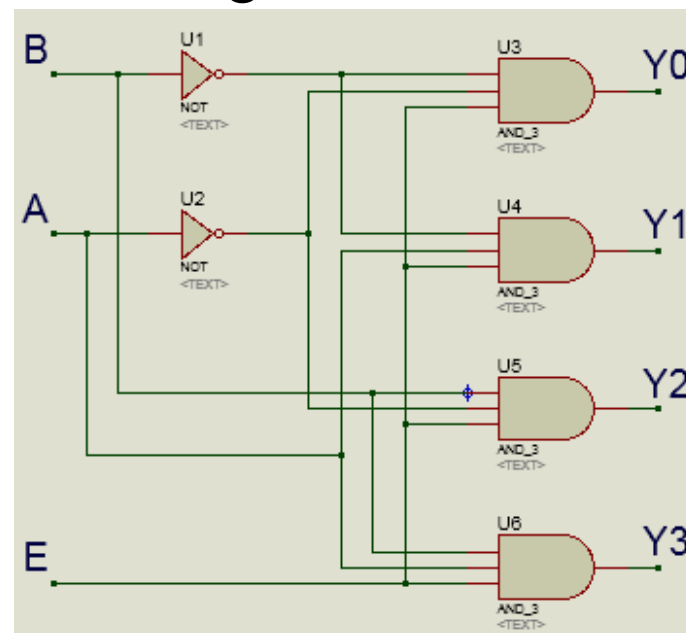
$$y_0 = E \cdot \bar{B} \cdot \bar{A}$$

$$y_1 = E \cdot \bar{B} \cdot A$$

$$y_2 = E \cdot B \cdot \bar{A}$$

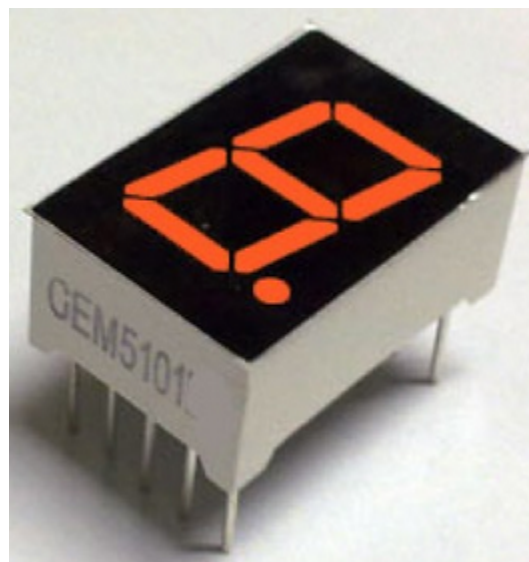
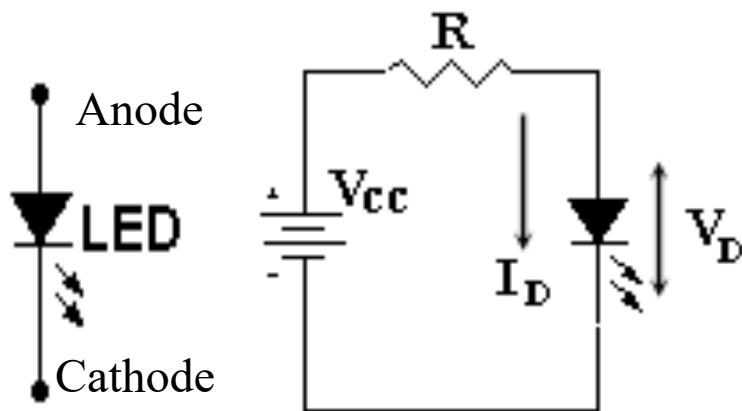
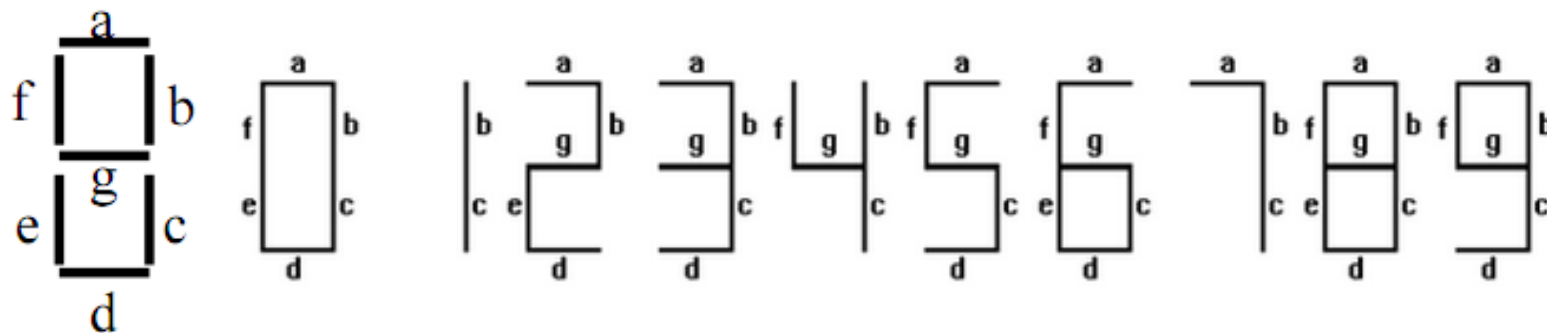
$$y_3 = E \cdot B \cdot A$$

- Sơ đồ logic:



Mạch giải mã BCD – LED 7 đoạn

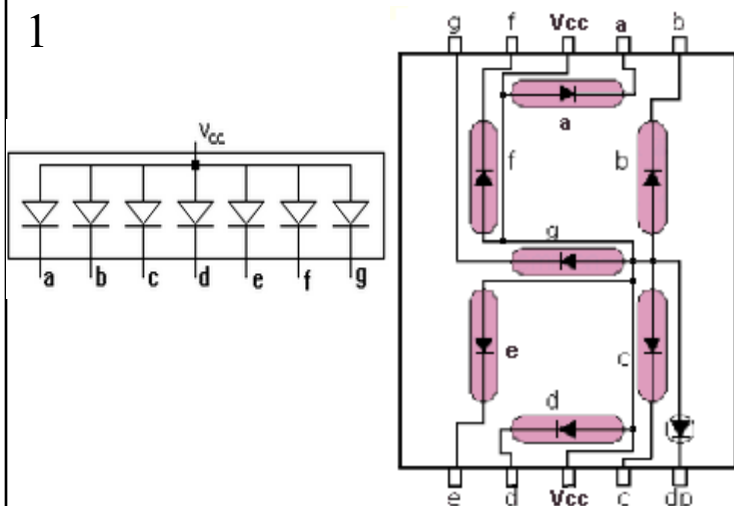
- LED 7 đoạn: mỗi đoạn là một đèn LED



Mạch giải mã BCD – LED 7 đoạn (tt)

LED 7 đoạn Anode chung

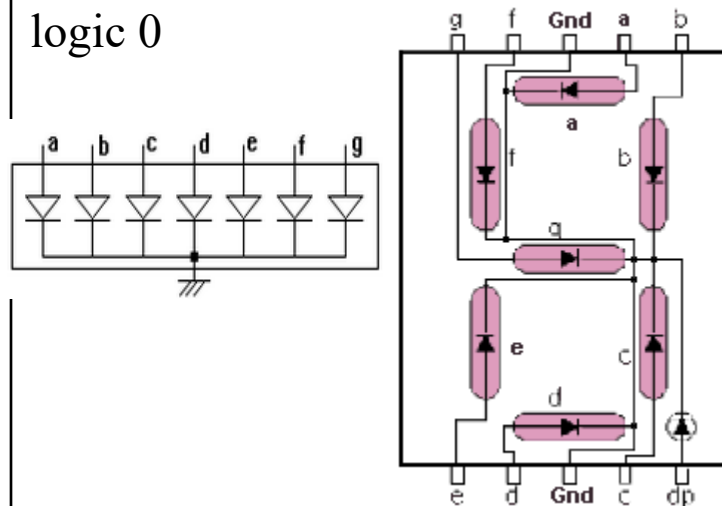
Anode của các đoạn LED được nối chung với nhau và đưa lên mức logic 1



- muốn đoạn LED nào tắt, nối Cathode tương ứng lên mức logic 1
- muốn đoạn LED nào sáng, nối Cathode tương ứng xuống mức logic 0

LED 7 đoạn Cathode chung

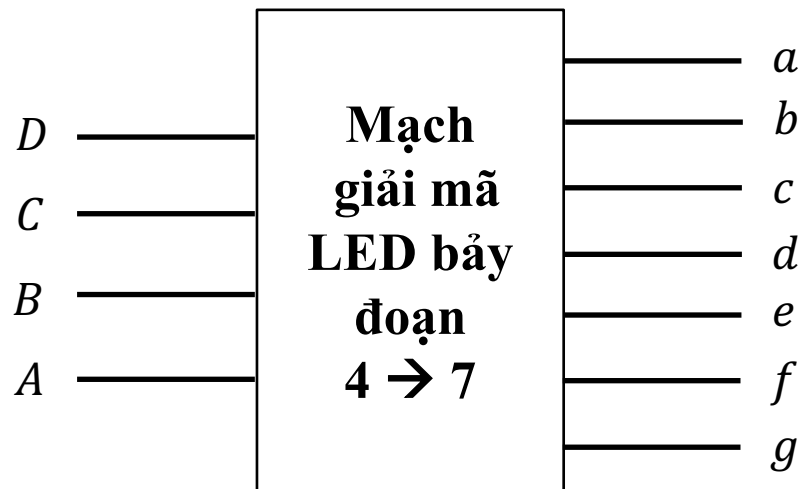
Cathode của các đoạn LED được nối chung với nhau và đưa xuống mức logic 0



- muốn đoạn LED nào tắt, nối Anode tương ứng xuống mức logic 0
- muốn đoạn LED nào sáng, nối Anode tương ứng lên mức logic 1

Mạch giải mã BCD – LED 7 đoạn (tt)

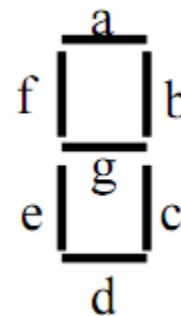
- 4 ngõ vào: từ mã BCD 4 bit
- 7 ngõ ra: nối đến 7 Anode/Cathode của LED 7 đoạn để biểu diễn số thập phân tương ứng
- Sơ đồ khối:



Mạch giải mã BCD – LED 7 đoạn (tt)

- Thiết kế mạch giải mã BCD 8421 – LED 7 đoạn Anode chung
 - Bảng trạng thái:

D	C	B	A	a	b	c	d	e	f	g	Số hiển thị
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	0	1	0	2
0	0	1	1	0	0	0	0	1	1	0	3
0	1	0	0	1	0	0	1	1	0	0	4
0	1	0	1	0	1	0	0	1	0	0	5
0	1	1	0	0	1	0	0	0	0	0	6
0	1	1	1	0	0	0	1	1	1	1	7
1	0	0	0	0	0	0	0	0	0	0	8
1	0	0	1	0	0	0	0	1	0	0	9
1	0	1	0	X	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X	X



Mạch giải mã BCD – LED 7 đoạn (tt)

- Phương trình logic:

a

BA \ DC				
	00	01	11	10
00	0	1	X	0
01	1	0	X	0
11	0	0	X	X
10	0	0	X	X

$$a = C\bar{B}\bar{A} + \bar{D}\bar{C}\bar{B}A$$

b

BA \ DC				
	00	01	11	10
00	0	0	X	0
01	0	1	X	0
11	0	0	X	X
10	0	1	X	X

$$b = C\bar{B}A + CB\bar{A} = C(B \oplus A)$$

c

BA \ DC				
	00	01	11	10
00	0	0	X	0
01	0	0	X	0
11	0	0	X	X
10	1	0	X	X

$$c = \bar{C}B\bar{A}$$

Mạch giải mã BCD – LED 7 đoạn (tt)

- Phương trình logic:

d

DC \ BA	00	01	11	10
00	0	1	X	0
01	1	0	X	0
11	0	1	X	X
10	0	0	X	X

$$d = C\bar{B}\bar{A} + CBA + \bar{D}\bar{C}\bar{B}A$$

f

DC \ BA	00	01	11	10
00	0	0	X	0
01	1	0	X	0
11	1	1	X	X
10	1	0	X	X

$$f = BA + \bar{D}\bar{C}A + \bar{C}B$$

e

DC \ BA	00	01	11	10
00	0	1	X	0
01	1	1	X	1
11	1	1	X	X
10	0	0	X	X

$$e = C\bar{B} + A$$

g

DC \ BA	00	01	11	10
00	1	0	X	0
01	1	0	X	0
11	0	1	X	X
10	0	0	X	X

$$g = CBA + \bar{D}\bar{C}\bar{B}$$

Mạch giải mã BCD – LED 7 đoạn (tt)

- Lập sơ đồ logic

Phương trình logic:

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

$$b = C(B \oplus A)$$

$$c = \overline{C}\overline{B}\overline{A}$$

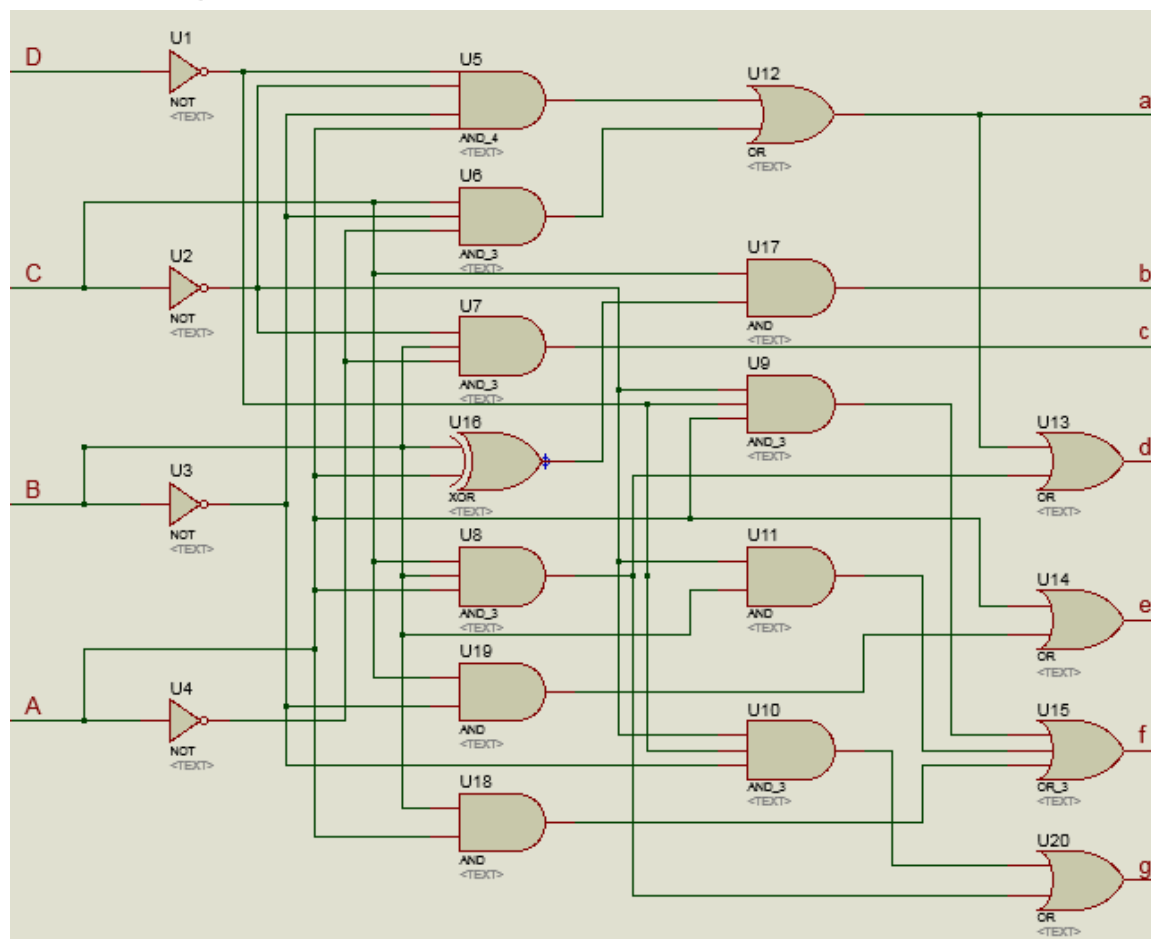
$$d = \overline{C}BA + \overline{C}\overline{B}\overline{A} + \overline{D}\overline{C}\overline{B}\overline{A}$$
$$= \overline{C}BA + a$$

$$e = \overline{C}\overline{B} + A$$

$$f = BA + \overline{D}\overline{C}A + \overline{C}\overline{B}$$

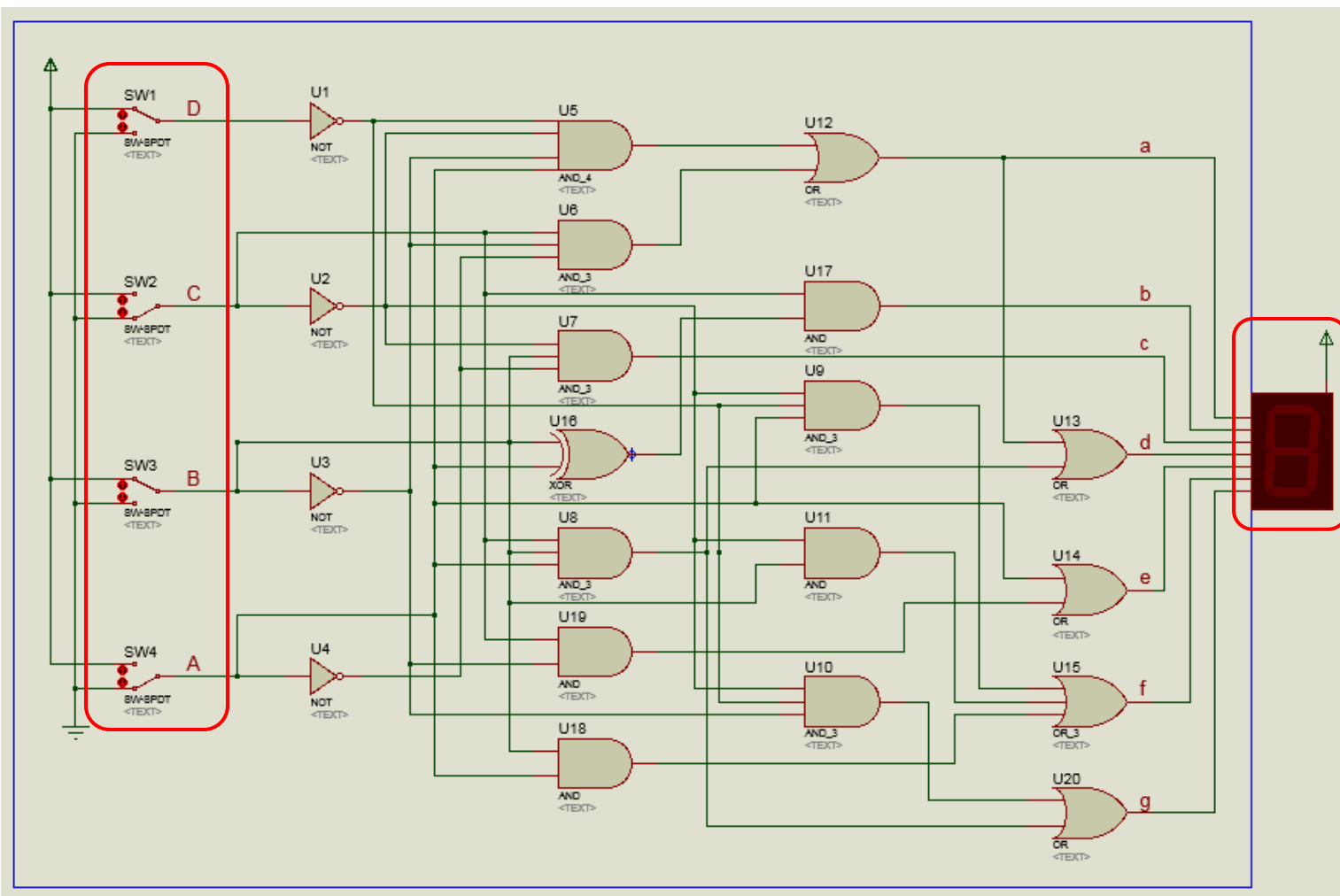
$$g = \overline{C}BA + \overline{D}\overline{C}\overline{B}$$

Sơ đồ logic:



Mạch giải mã BCD – LED 7 đoạn (tt)

- Mạch mô phỏng trên Proteus:

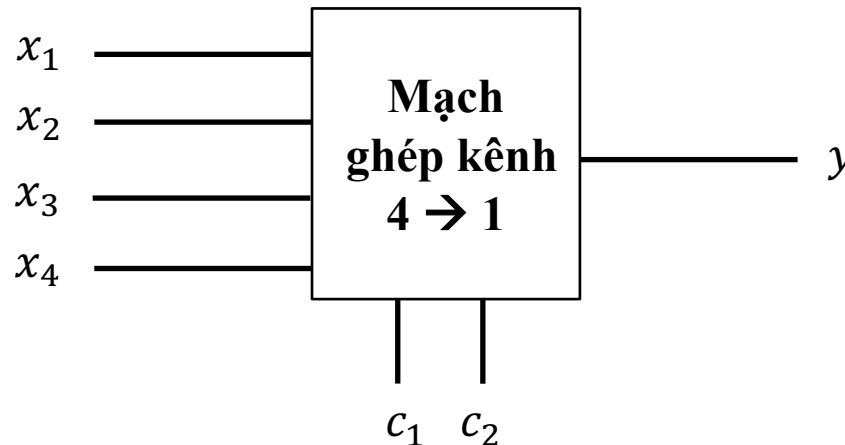


Mạch ghép kênh – phân kênh

- Mạch ghép kênh (chọn kênh): chọn lần lượt 1 trong N kênh vào để đưa đến ngõ ra duy nhất
 - Mạch ghép kênh còn gọi là mạch chuyển dữ liệu song song ở ngõ vào thành dữ liệu nối tiếp ở ngõ ra
- Mạch phân kênh (tách kênh): tách nguồn dữ liệu cùng một đầu vào thành N ngõ ra khác nhau
 - Mạch phân kênh còn gọi là mạch chuyển dữ liệu nối tiếp ở ngõ vào thành dữ liệu song song ở ngõ ra

Mạch ghép kênh

- Xét mạch ghép kênh có 4 ngõ vào và 1 ngõ ra:



- c_1, c_2 là các ngõ vào điều khiển mạch ghép kênh chọn lần lượt 1 trong 4 kênh vào
- Nếu có N kênh vào thì có n ngõ vào điều khiển sao cho:

$$N = 2^n$$

Mạch ghép kênh (tt)

- Mạch ghép kênh có 4 ngõ vào và 1 ngõ ra:
 - Bảng trạng thái:

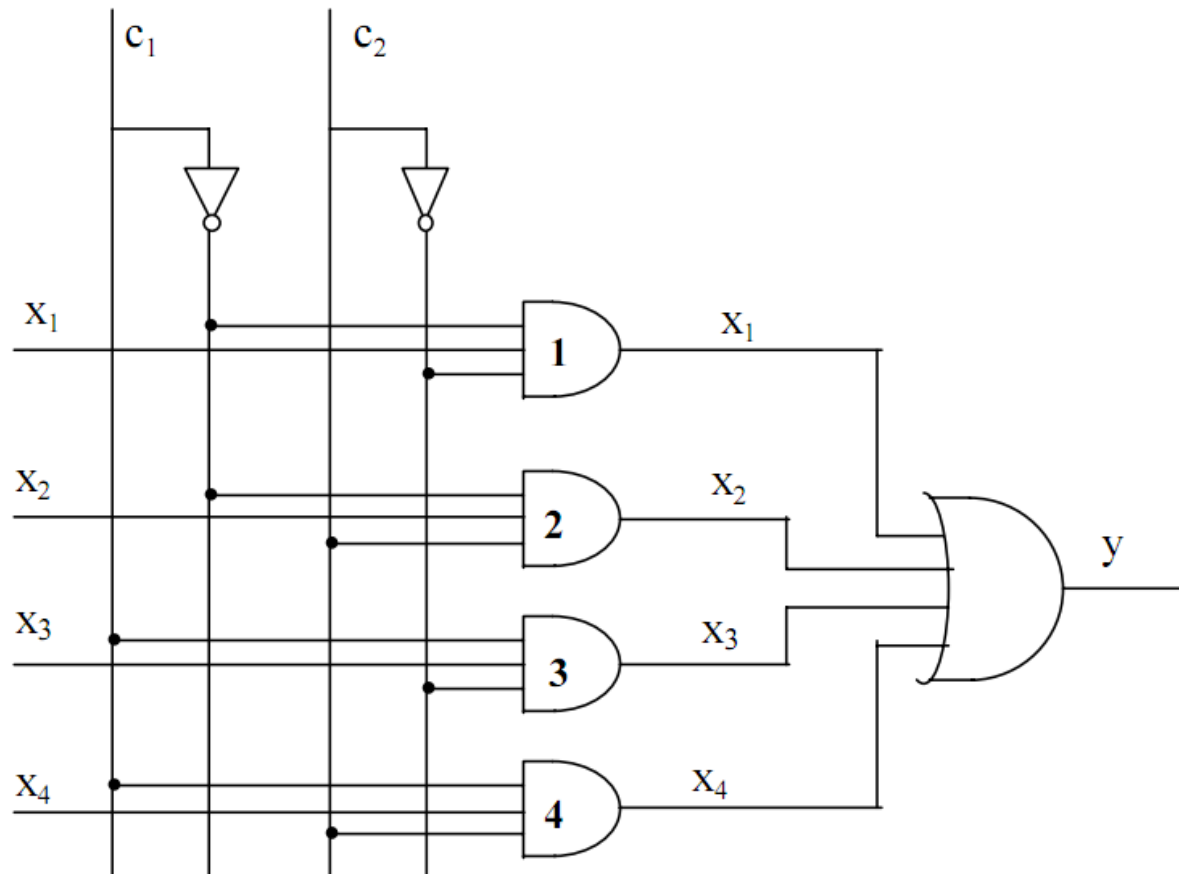
c_1	c_2	y
0	0	x_1
0	1	x_2
1	0	x_3
1	1	x_4

- Phương trình logic:

$$y = \bar{c}_1\bar{c}_2x_1 + \bar{c}_1c_2x_2 + c_1\bar{c}_2x_3 + c_1c_2x_4$$

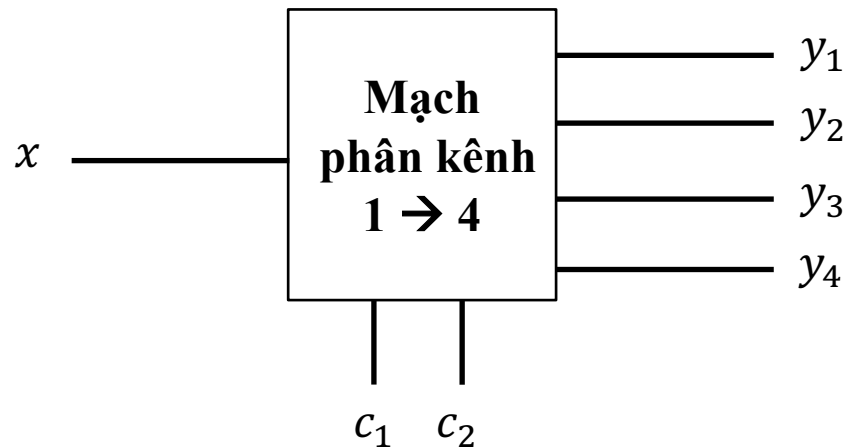
Mạch ghép kênh (tt)

- Mạch ghép kênh có 4 ngõ vào và 1 ngõ ra:
 - Sơ đồ logic:



Mạch phân kênh

- Xét mạch phân kênh có 1 ngõ vào và 4 ngõ ra:



- Bảng trạng thái:

c_1	c_2	y_1	y_2	y_3	y_4
0	0	x	0	0	0
0	1	0	x	0	0
1	0	0	0	x	0
1	1	0	0	0	x

Mạch phân kênh (tt)

- Xét mạch phân kênh có 1 ngõ vào và 4 ngõ ra:

- Phương trình logic:

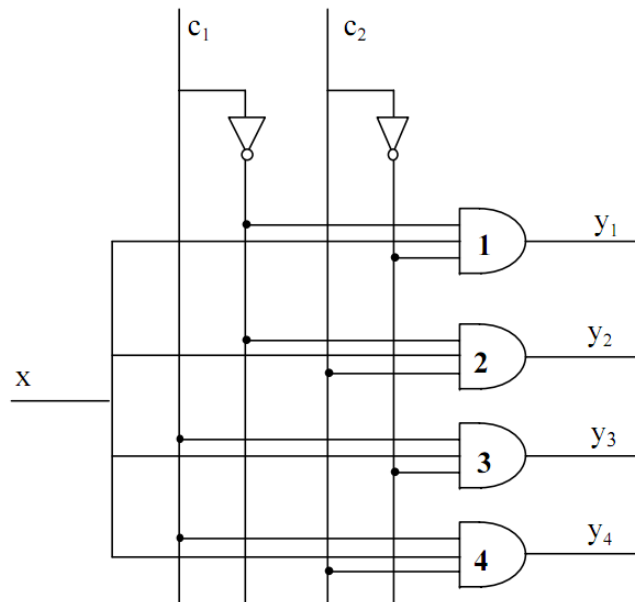
$$y_1 = \bar{c}_1 \bar{c}_2 x$$

$$y_2 = \bar{c}_1 c_2 x$$

$$y_3 = c_1 \bar{c}_2 x$$

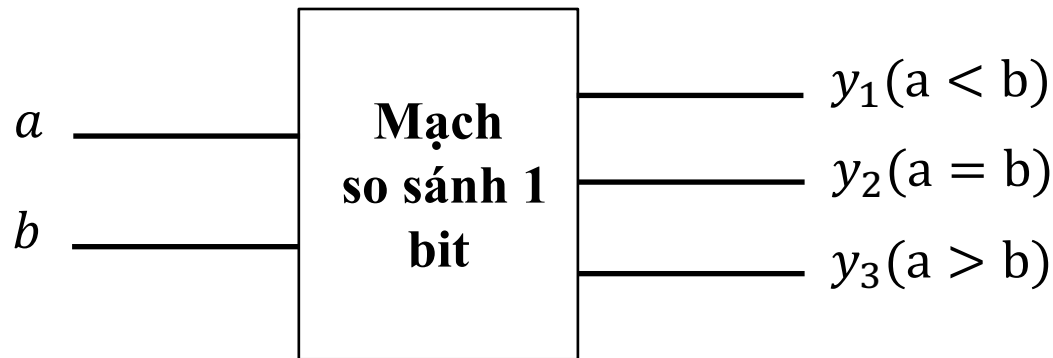
$$y_4 = c_1 c_2 x$$

- Sơ đồ logic:



Mạch so sánh

- Mạch so sánh dùng để so sánh các số nhị phân về mặt độ lớn
- Mạch so sánh 1 bit:
 - Sơ đồ khối:



Mạch so sánh (tt)

- Mạch so sánh 1 bit:
 - Bảng trạng thái:

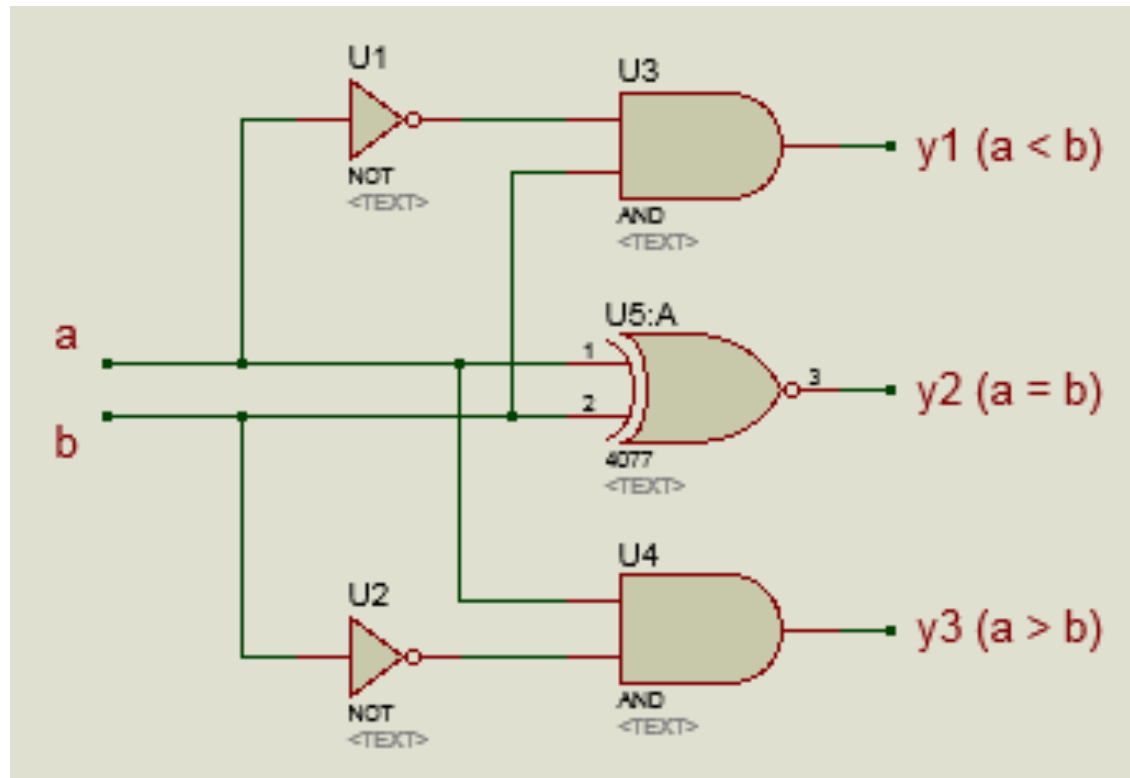
a	b	y_1 ($a < b$)	y_2 ($a = b$)	y_3 ($a > b$)
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

- Phương trình logic:

$$\begin{aligned}y_1 &= \bar{a}b \\y_2 &= \bar{a}\bar{b} + ab = \overline{a \oplus b} \\y_3 &= a\bar{b}\end{aligned}$$

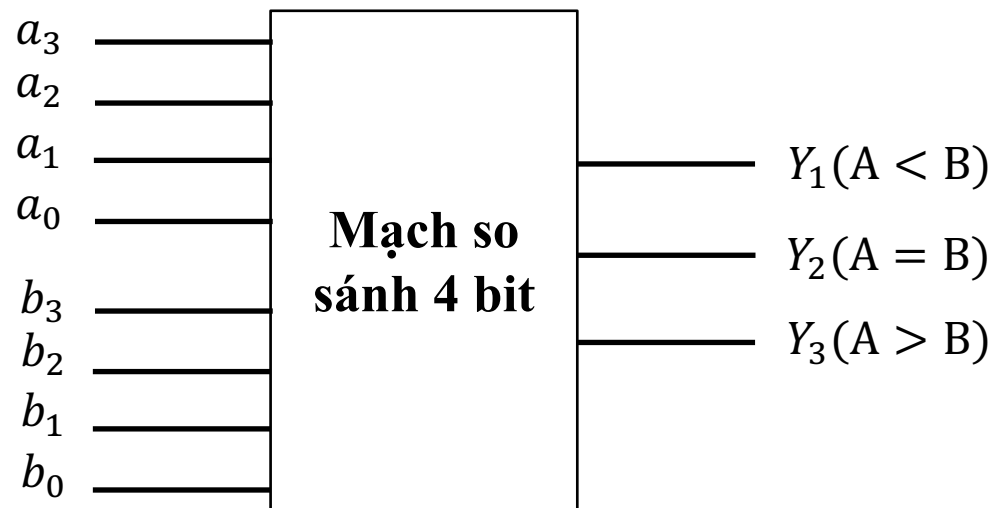
Mạch so sánh (tt)

- Mạch so sánh 1 bit:
 - Sơ đồ logic:



Mạch so sánh (tt)

- Mạch so sánh nhiều bit:
 - Xét mạch so sánh 2 số nhị phân 4 bit $A (a_3a_2a_1a_0)$ và $B (b_3b_2b_1b_0)$:



Mạch so sánh (tt)

- Mạch so sánh 4 bit:
 - Thực hiện trên cơ sở mạch so sánh 1 bit:
 - Bảng trạng thái:

Ngõ vào				Ngõ ra		
a_3 và b_3	a_2 và b_2	a_1 và b_1	a_0 và b_0	$Y_1 (A < B)$	$Y_2 (A = B)$	$Y_3 (A > B)$
<	x	x	x	1	0	0
>	x	x	x	0	0	1
=	<	x	x	1	0	0
=	>	x	x	0	0	1
=	=	<	x	1	0	0
=	=	>	x	0	0	1
=	=	=	<	1	0	0
=	=	=	>	0	0	1
=	=	=	=	0	1	0

Mạch so sánh (tt)

- Mạch so sánh 4 bit:
 - Thực hiện trên cơ sở mạch so sánh 1 bit:
 - Phương trình logic:

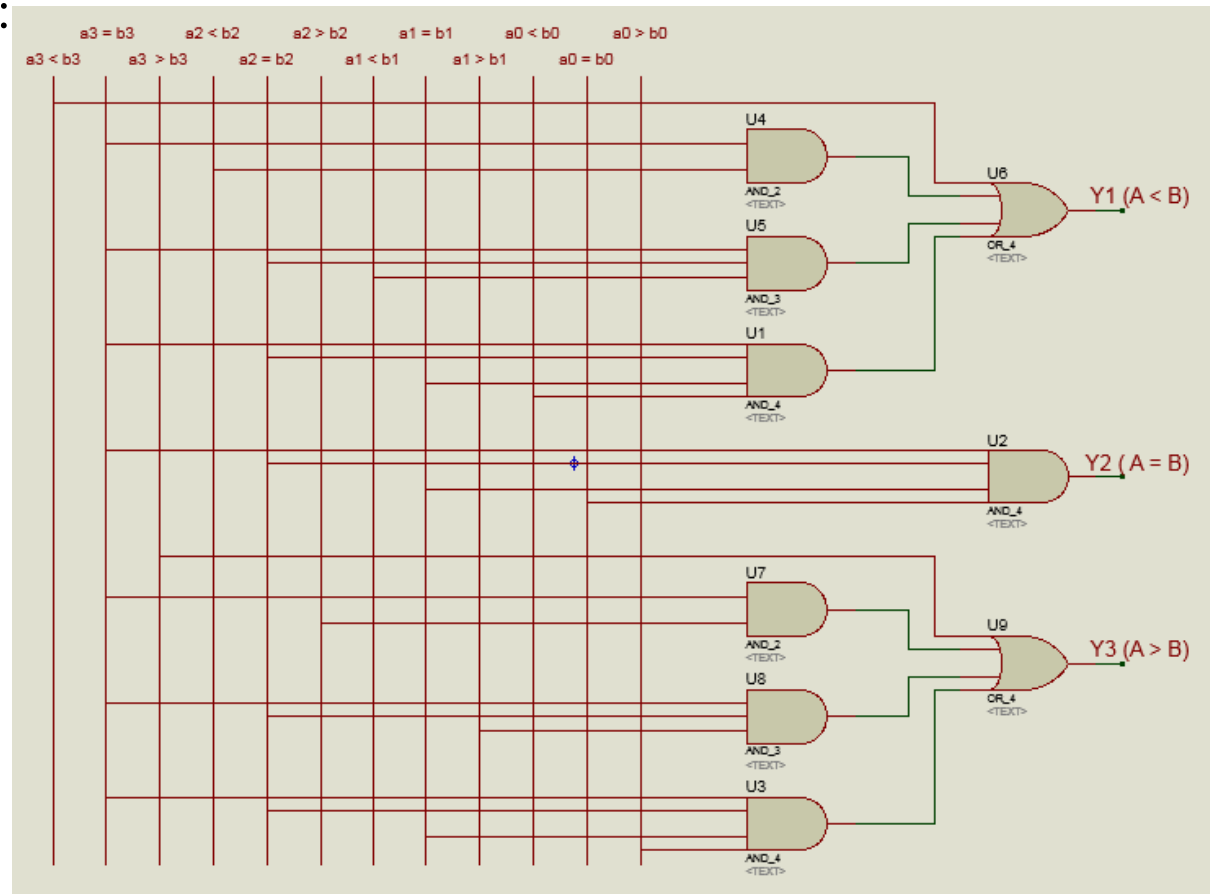
$$\begin{aligned} Y_1 &= (A < B) \\ &= (a_3 < b_3) + (a_3 = b_3)(a_2 < b_2) + (a_3 = b_3)(a_2 = b_2)(a_1 < b_1) \\ &\quad + (a_3 = b_3)(a_2 = b_2)(a_1 = b_1)(a_0 < b_0) \end{aligned}$$

$$\begin{aligned} Y_2 &= (A = B) \\ &= (a_3 = b_3)(a_2 = b_2)(a_1 = b_1)(a_0 = b_0) \end{aligned}$$

$$\begin{aligned} Y_3 &= (A > B) \\ &= (a_3 > b_3) + (a_3 = b_3)(a_2 > b_2) + (a_3 = b_3)(a_2 = b_2)(a_1 > b_1) \\ &\quad + (a_3 = b_3)(a_2 = b_2)(a_1 = b_1)(a_0 > b_0). \end{aligned}$$

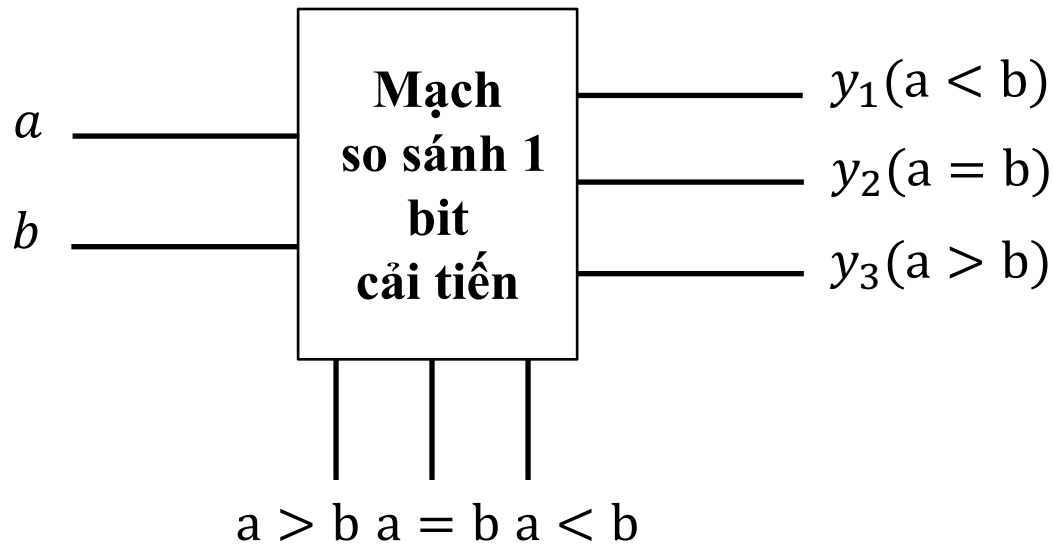
Mạch so sánh (tt)

- Mạch so sánh 4 bit:
 - Thực hiện trên cơ sở mạch so sánh 1 bit:
 - Sơ đồ logic:



Mạch so sánh (tt)

- Mạch so sánh 1 bit cải tiến:



Mạch so sánh (tt)

- Mạch so sánh 1 bit cải tiến:
 - Bảng trạng thái:

Ngõ vào điều khiển			Ngõ vào dữ liệu		Ngõ ra		
$c_1 (a < b)$	$c_2 (a = b)$	$c_3 (a > b)$	a	b	$y_1 (a < b)$	$y_2 (a = b)$	$y_3 (a > b)$
1	0	0	x	x	1	0	0
0	0	1	x	x	0	0	1
0	1	0	0	0	0	1	0
0	1	0	0	1	1	0	0
0	1	0	1	0	0	0	1
0	1	0	1	1	0	1	0

- Phương trình logic :

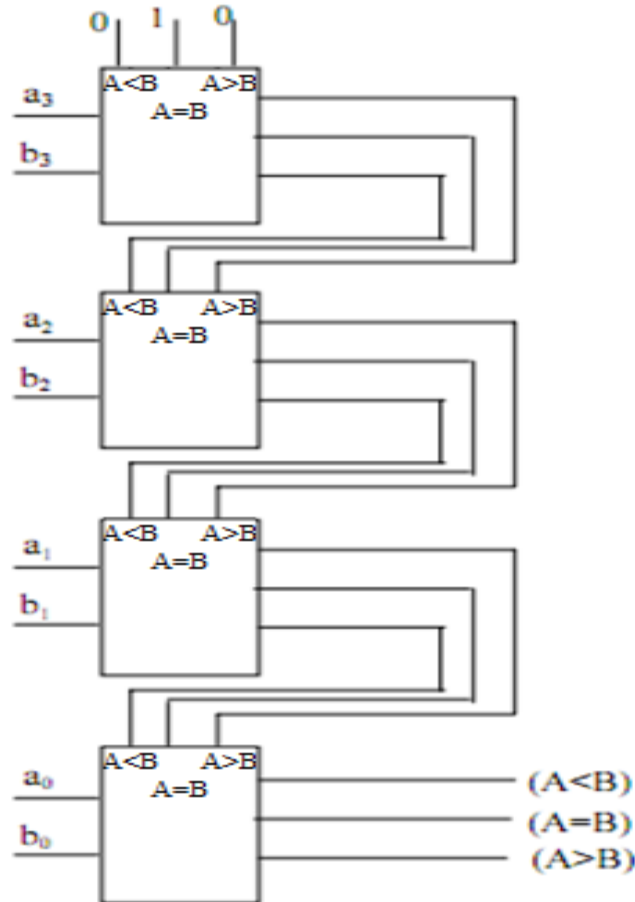
$$y_1 = (a < b) = c_1 + c_2 \bar{a}b$$

$$y_2 = (a = b) = c_2(\bar{a} \oplus \bar{b})$$

$$y_3 = (a > b) = c_3 + c_2 a \bar{b}$$

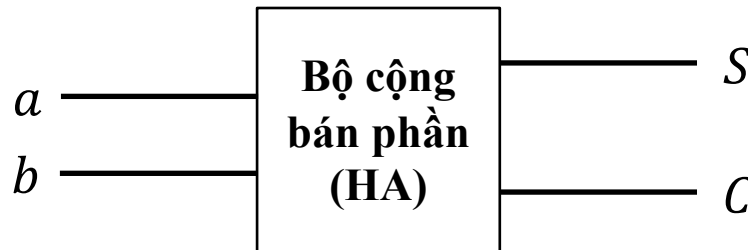
Mạch so sánh (tt)

- Mạch so sánh 4 bit:
 - Thực hiện trên cơ sở mạch so sánh 1 bit cải tiến:
 - Sơ đồ logic:



Mạch số học

- Mạch số học có chức năng thực hiện các phép toán số học cộng, trừ, nhân, chia
- Bộ cộng bán phần (Half Adder):
 - Bộ cộng bán phần thực hiện cộng hai số nhị phân 1 bit
 - Sơ đồ khối:



Trong đó a , b là số cộng, S là tổng, C là số nhớ

Mạch số học (tt)

- Bộ cộng bán phần:

- Bảng trạng thái:

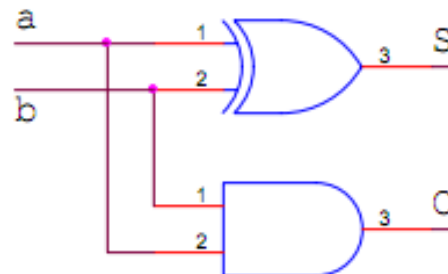
<i>a</i>	<i>b</i>	<i>S</i>	<i>C</i>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

- Phương trình logic:

$$S = a\bar{b} + \bar{a}b = a \oplus b$$

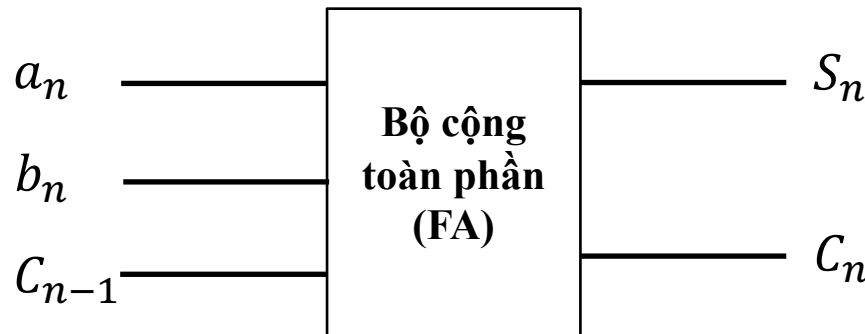
$$C = a.b$$

- Sơ đồ logic:



Mạch số học (tt)

- Bộ cộng toàn phần (Full Adder):
 - Sơ đồ khối:



Trong đó: C_{n-1} là số nhớ của lần cộng trước đó
 C_n là số nhớ của lần cộng hiện tại
 S_n là tổng hiện tại

Mạch số học (tt)

- Bộ cộng toàn phần:
 - Bảng trạng thái:

a_n	b_n	c_{n-1}	s_n	c_n
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Mạch số học (tt)

- Bộ cộng toàn phần:
 - Phương trình logic: tối thiểu hóa dùng bảng Karnaugh

		$a_n b_n$			
		00	01	11	10
C_{n-1}	0	0	1	0	1
	1	1	0	1	0

		$a_n b_n$			
		00	01	11	10
C_{n-1}	0	0	0	1	0
	1	0	1	1	1

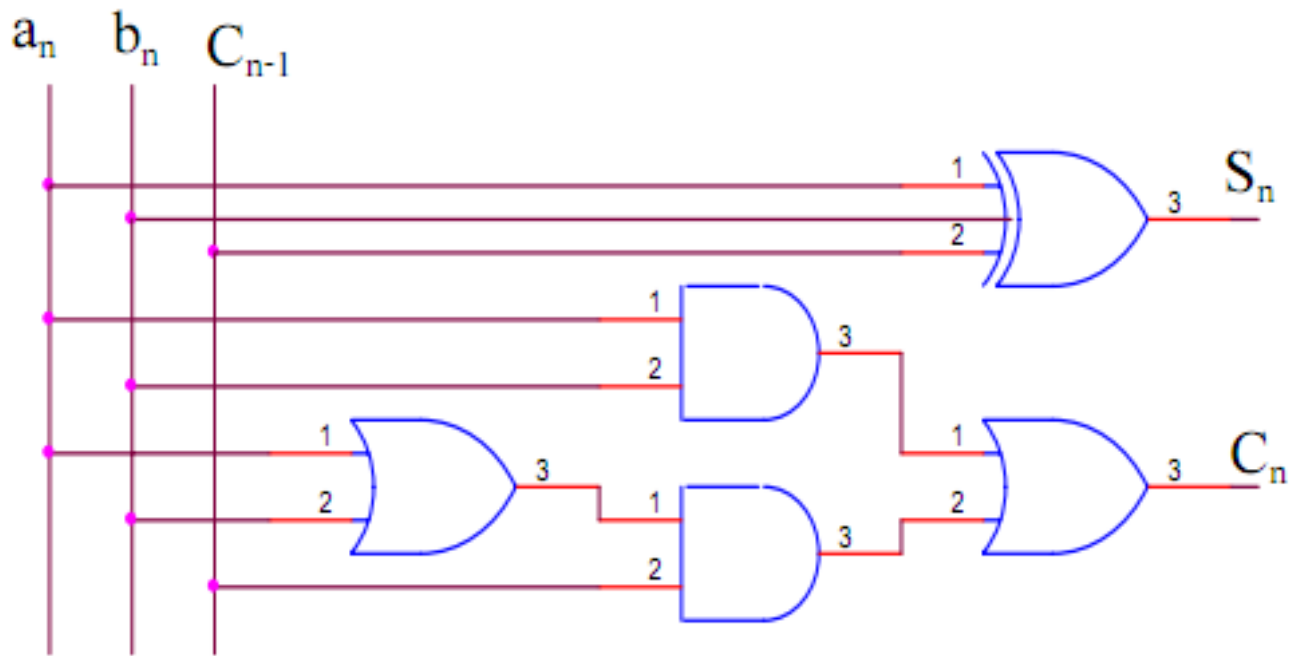
$$S_n = \overline{a_n} \overline{b_n} C_{n-1} + a_n b_n C_{n-1} + \overline{a_n} b_n \overline{C_{n-1}} + a_n \overline{b_n} \overline{C_{n-1}}$$

$$S_n = a_n \oplus b_n \oplus C_{n-1}$$

$$C_n = a_n C_{n-1} + b_n C_{n-1} + a_n b_n = a_n b_n + C_{n-1}(a_n + b_n)$$

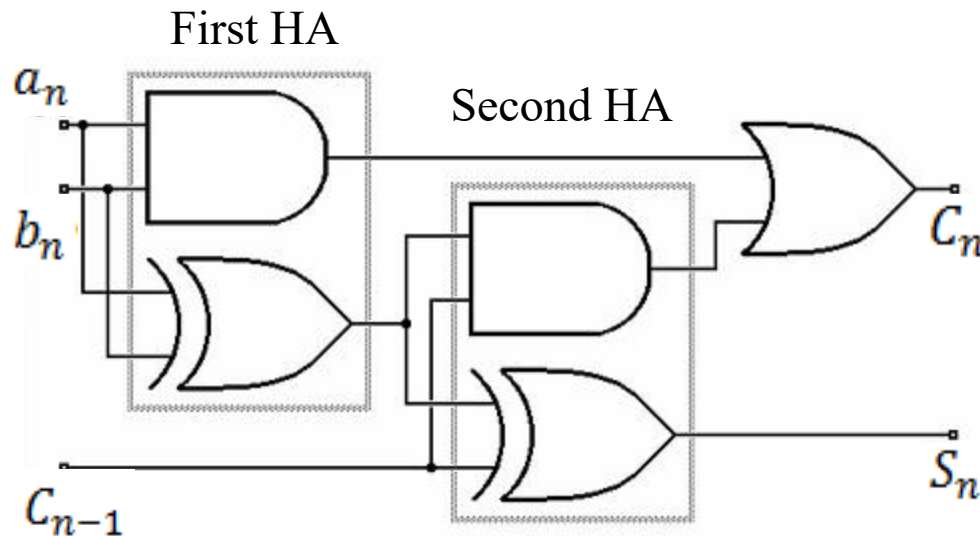
Mạch số học (tt)

- Bộ cộng toàn phần:
 - Sơ đồ logic:



Mạch số học (tt)

- Bộ cộng toàn phần:
 - Sử dụng HA để thực hiện FA:



Half adder:

$$S = a \oplus b$$

$$C = a.b$$

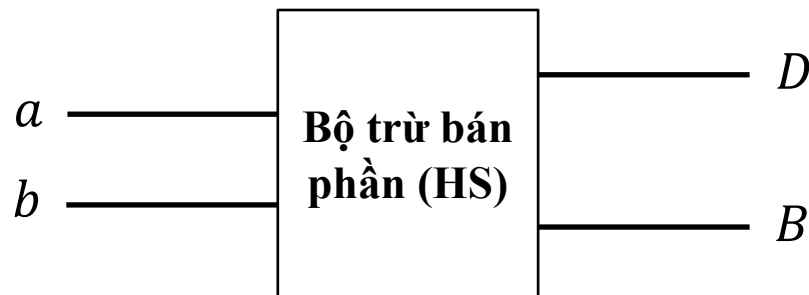
Full adder:

$$S_n = a_n \oplus b_n \oplus C_{n-1}$$

$$C_n = a_n b_n + C_{n-1}(a_n + b_n)$$

Mạch số học (tt)

- Bộ trừ bán phần (Half Subtractor):
 - Bộ trừ bán phần thực hiện trừ hai số nhị phân 1 bit
 - Sơ đồ khối:



Trong đó a là số bị trừ, b là số trừ, D là hiệu, B là số mượn

Mạch số học (tt)

- Bộ trừ bán phần:

- Bảng trạng thái:

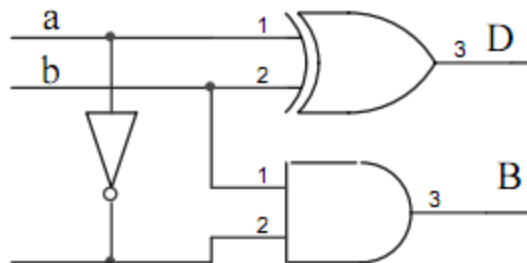
<i>a</i>	<i>b</i>	<i>D</i>	<i>B</i>
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

- Phương trình logic:

$$D = a\bar{b} + \bar{a}b = a \oplus b$$

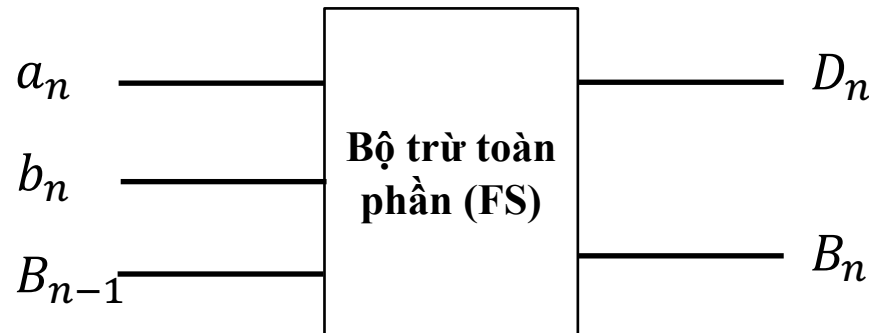
$$B = \bar{a}.b$$

- Sơ đồ logic:



Mạch số học (tt)

- Bộ trừ toàn phần (Full Subtractor):
 - Sơ đồ khối:



Trong đó: B_{n-1} là số mượn của lần trừ trước đó
 B_n là số mượn của lần trừ hiện tại
 D_n là hiệu hiện tại

Mạch số học (tt)

- Bộ trừ toàn phần:
 - Bảng trạng thái:

a_n	b_n	c_{n-1}	s_n	c_n
0	0	0	0	0
0	1	0	1	1
1	0	0	1	0
1	1	0	0	0
0	0	1	1	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	1

Mạch số học (tt)

- Bộ trừ toàn phần:
 - Phương trình logic: tối thiểu hóa dùng bảng Karnaugh

		$a_n b_n$			
		00	01	11	10
B_{n-1}	0	0	1	0	1
	1	1	0	1	0

		$a_n b_n$			
		00	01	11	10
B_{n-1}	0	0	1	0	0
	1	1	1	1	0

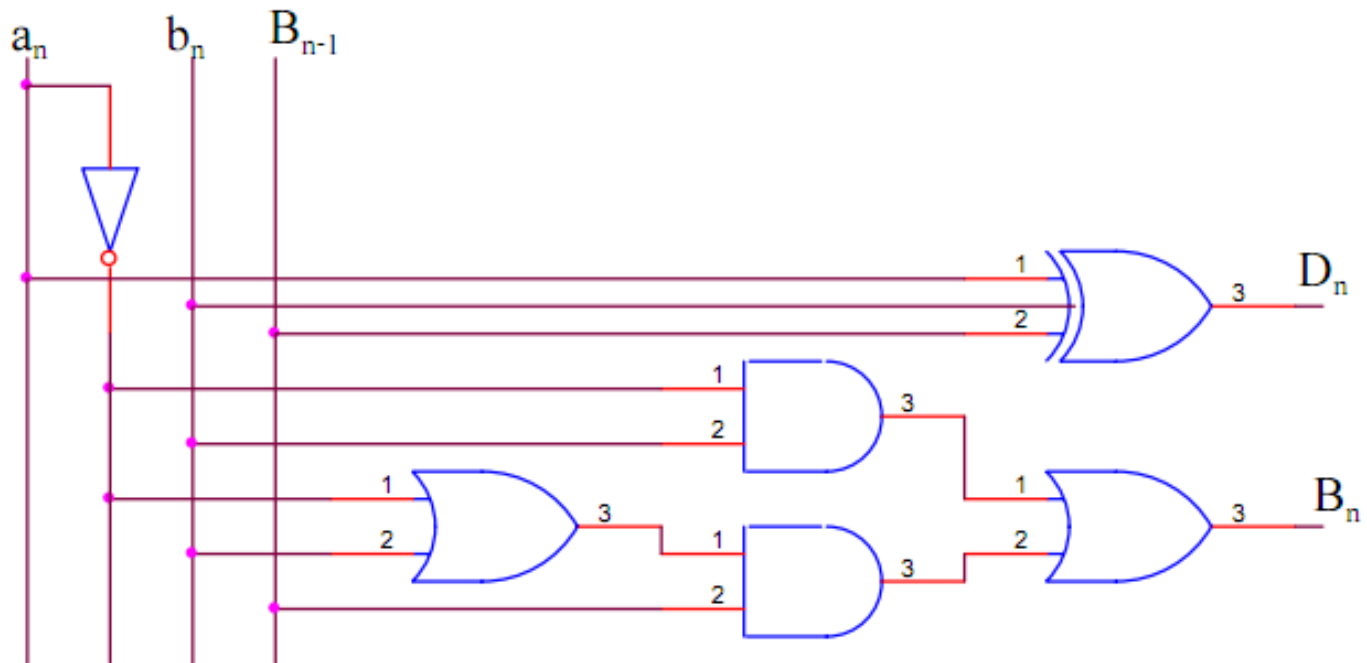
$$D_n = \overline{a_n} \overline{b_n} B_{n-1} + a_n b_n B_{n-1} + \overline{a_n} b_n \overline{B_{n-1}} + a_n \overline{b_n} \overline{B_{n-1}}$$

$$D_n = a_n \oplus b_n \oplus B_{n-1}$$

$$B_n = \overline{a_n} B_{n-1} + b_n B_{n-1} + \overline{a_n} b_n = \overline{a_n} b_n + B_{n-1}(\overline{a_n} + b_n)$$

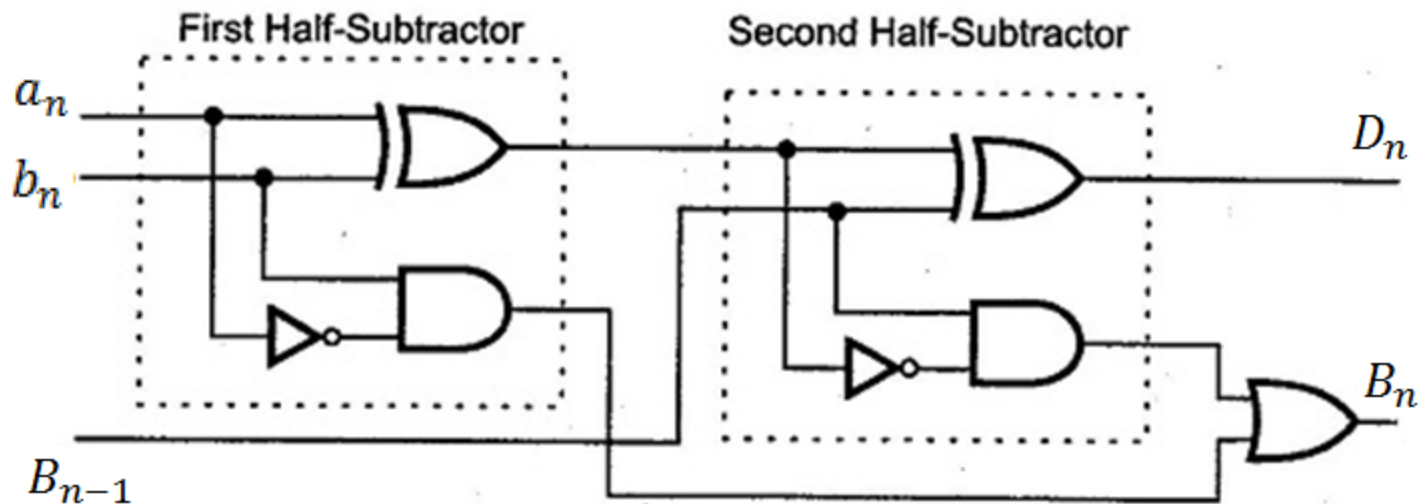
Mạch số học (tt)

- Bộ trừ toàn phần:
 - Sơ đồ logic:



Mạch số học (tt)

- Bộ trừ toàn phần:
 - Sử dụng HS để thực hiện FS:



Half subtractor:

$$D = a \oplus b$$

$$B = \bar{a} \cdot b$$

Full subtractor :

$$D_n = a_n \oplus b_n \oplus B_{n-1}$$

$$B_n = \bar{a}_n b_n + B_{n-1}(\bar{a}_n + b_n)$$