

Chương 3: **HỆ TỔ HỢP**

I. NGUYÊN TẮC THIẾT KẾ

II. MỘT SỐ HỆ TỔ HỢP THÔNG DỤNG VÀ ỨNG DỤNG ĐỂ THỰC HIỆN HÀM BOOLE

I. NGUYÊN TẮC THIẾT KẾ

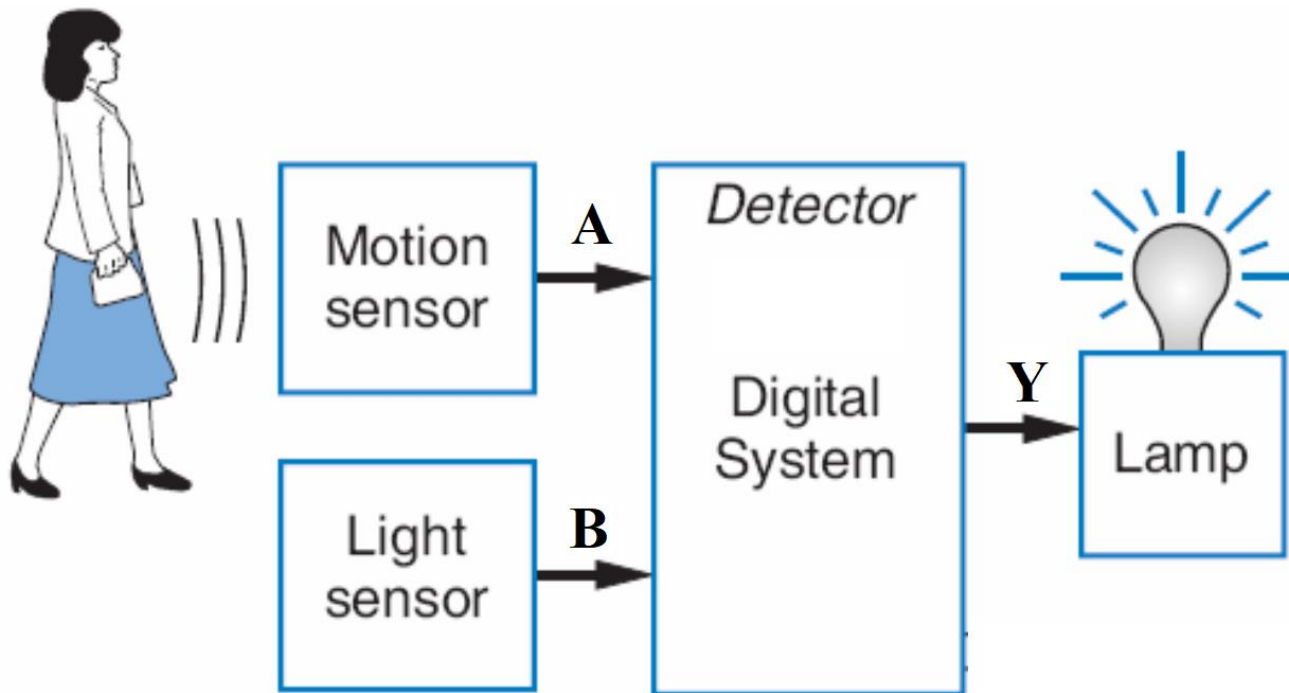
1. Phát biểu bài toán.
2. Xác định số biến ở ngõ vào và ngõ ra.
3. Thành lập bảng giá trị chỉ rõ mối quan hệ giữa ngõ vào và ngõ ra theo các điều kiện của bài toán.
4. Tìm biểu thức rút gọn của từng ngõ ra phụ thuộc vào các biến ở ngõ vào.
5. Thực hiện sơ đồ logic.

<i>Các ngõ vào</i> $X_{n-1} \dots X_0$	<i>Các ngõ ra</i> $Y_{m-1} \dots Y_0$
0.....00	
.....	
1.....11	

VD1

Thiết kế một mạch điều khiển đèn sáng khi phát hiện có người và trời tối.
Với trạng thái các ngõ vào như sau:

- Biến A=1 báo hiệu trạng thái có người nhận được từ cảm biến và ngược lại.
- Biến B=1 báo hiệu trạng thái trời sáng và ngược lại.
- Ngõ ra điều khiển đèn tích cực ở mức 1.



THỰC HIỆN THIẾT KẾ

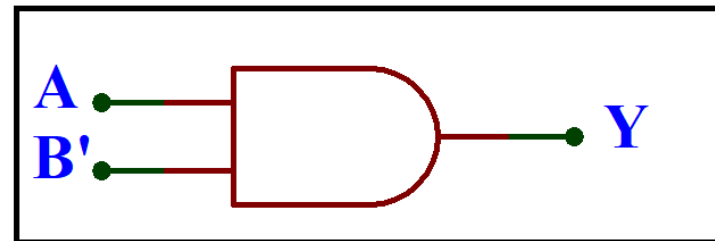
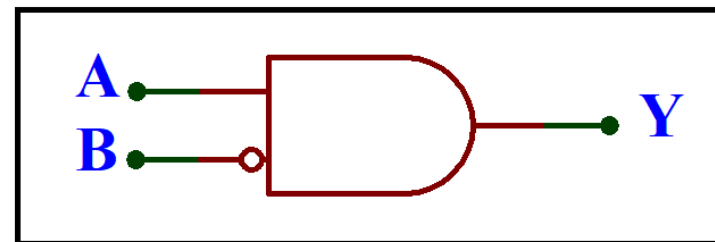
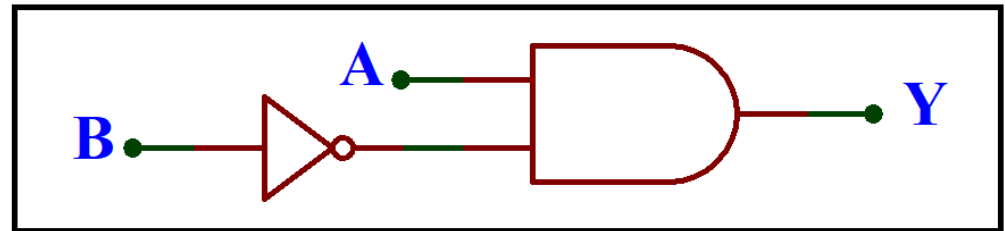
Lập bảng chân trị

Các ngõ vào		Ngõ ra
A	B	Y
0	0	0
0	1	0
1	0	1
1	1	0

Biểu thức ngõ ra

$$Y(A,B)=AB'$$

Thực hiện sơ đồ logic: $Y(A,B)=AB'$

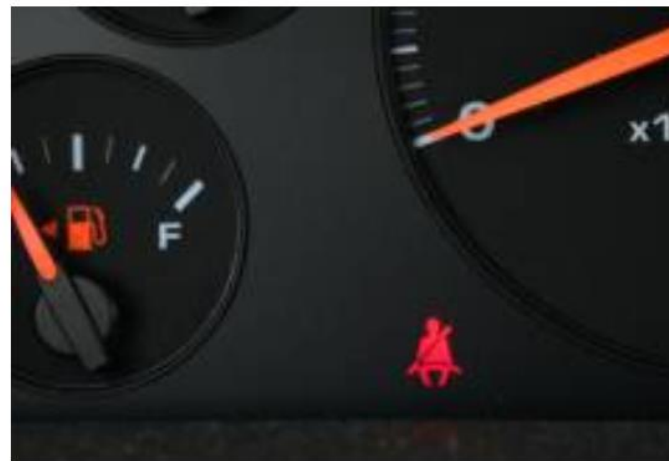


VD2

Thiết kế mạch đèn cảnh báo về trạng thái không thắt dây an toàn của người ngồi trên xe. Với các tín hiệu qui ước cho ngõ vào của mạch thiết kế nhận được từ các cảm biến như sau:

- **K = 1**: chìa khóa đã được cắm vào ổ khóa.
- **P = 1**: có người ở ghế ngồi.
- **S = 1**: dây an toàn đã được thắt.

Đèn cảnh báo tích cực cao (**W=1**) khi chìa khóa đã cắm vào ổ và người ngồi tại ghế không thắt dây an toàn.



THỰC HIỆN THIẾT KẾ

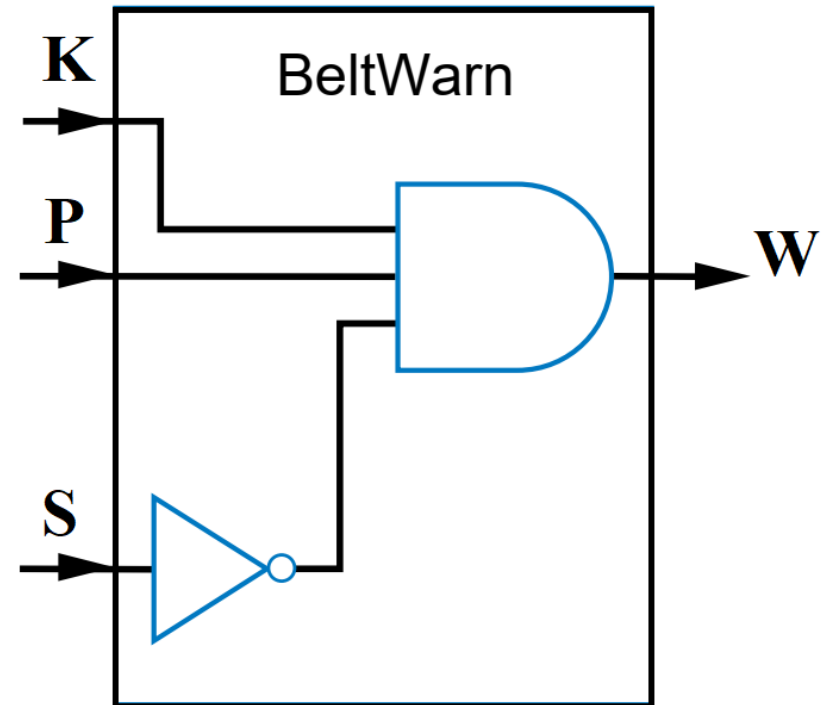
Lập bảng chân trị

Các ngõ vào			Ngõ ra
K	P	S	W
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

Biểu thức ngõ ra

$$W(K,P,S) = KPS'$$

Thực hiện sơ đồ logic

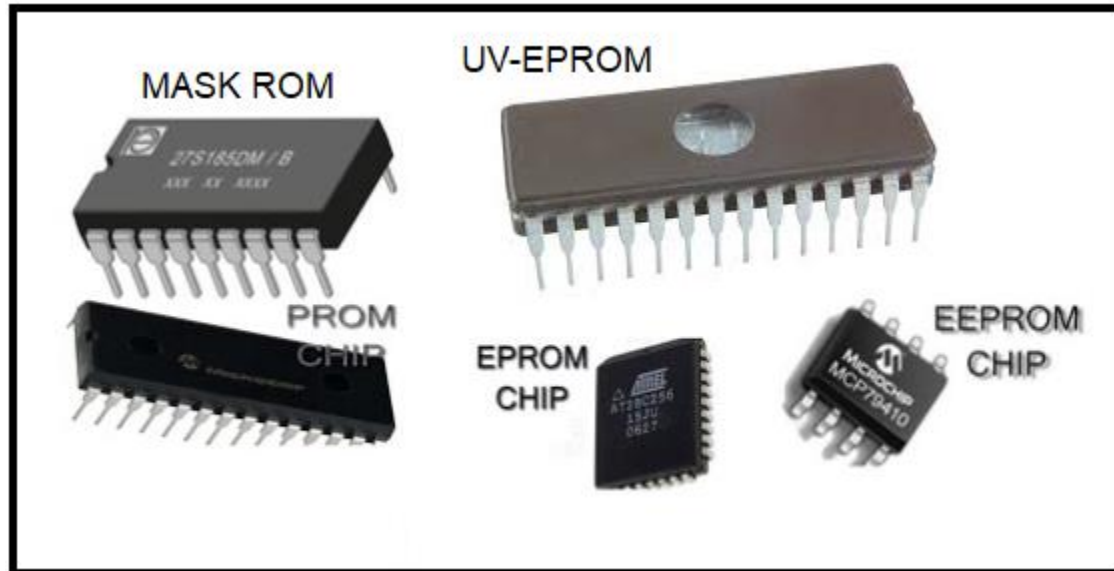


II. MỘT SỐ HỆ TỔ HỢP THÔNG DỤNG VÀ ỨNG DỤNG ĐỂ THỰC HIỆN HÀM BOOLE

- BỘ NHỚ ROM – PLA
- MẠCH CỘNG – TRỪ
- MẠCH GIẢI MÃ – MÃ HÓA
- MẠCH CHỌN KÊNH - PHÂN KÊNH

ROM (Read Only Memory)

- *Mask ROM*
- *PROM(One time Programmable)*
- *EPROM(UV-EPROM)*
- *EEPROM(Electronic Erasable PROM)*
- *FLASH ROM(EEPROM)*



BỘ NHỚ ROM- CÁC THÔNG SỐ LIÊN QUAN

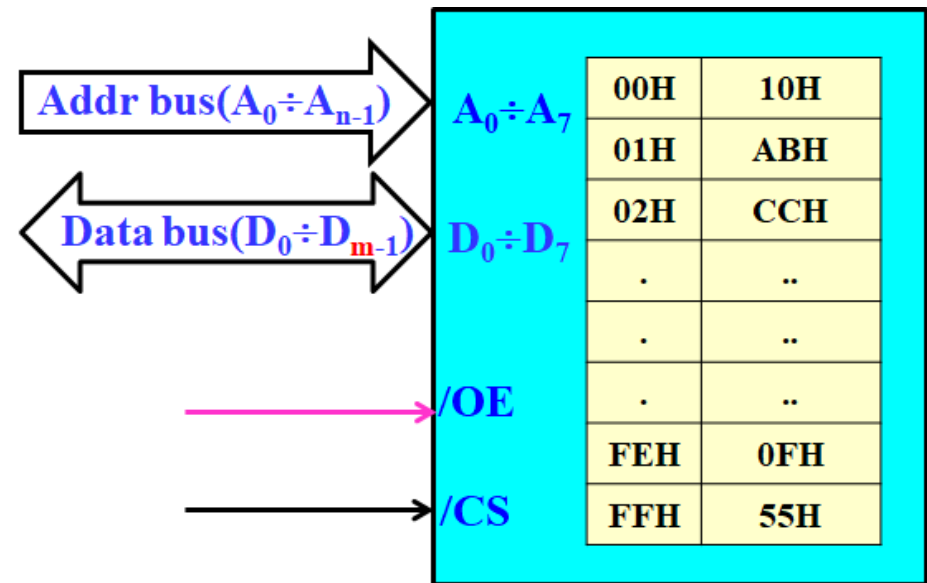
Dung lượng: số lượng bit mà bộ nhớ có thể lưu trữ được.

VD: 128 Kbits, 256Mbits

$$\text{Dung lượng(Kích thước)} = 2^n \cdot m \text{ (bit)}$$

Tổ chức: số lượng bit được lưu trữ tại 1 vị trí(địa chỉ) bộ nhớ.

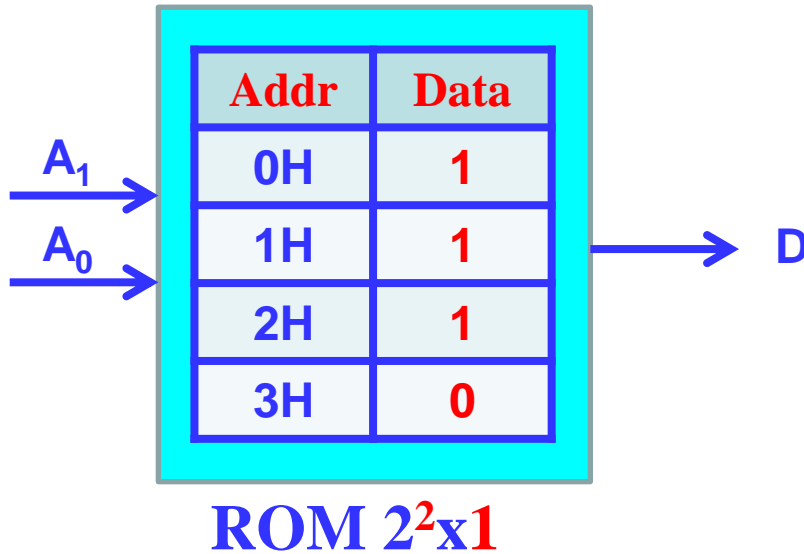
VD: $128(2^7) \times 8$ nghĩa là bộ nhớ có 128 địa chỉ ô nhớ(7 tín hiệu địa chỉ ở ngõ vào) và mỗi địa chỉ lưu 8 bit dữ liệu.



Thời gian truy xuất: khoảng thời gian để nhận được dữ liệu từ bộ nhớ.

BỘ NHỚ ROM- CÁC THÔNG SỐ LIÊN QUAN

Ví dụ:

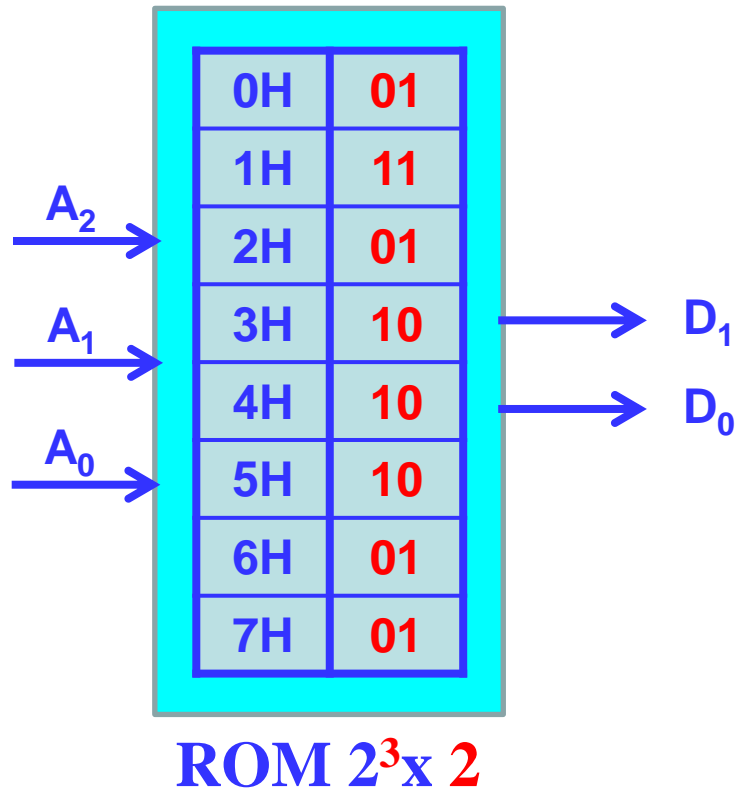


Inputs	Outputs
Address(A_1A_0)	Data(D)
00	1
01	1
10	1
11	0

Kích thước(dung lượng): $2^2 \times 1 = 4$ (bit)

BỘ NHỚ ROM- CÁC THÔNG SỐ LIÊN QUAN

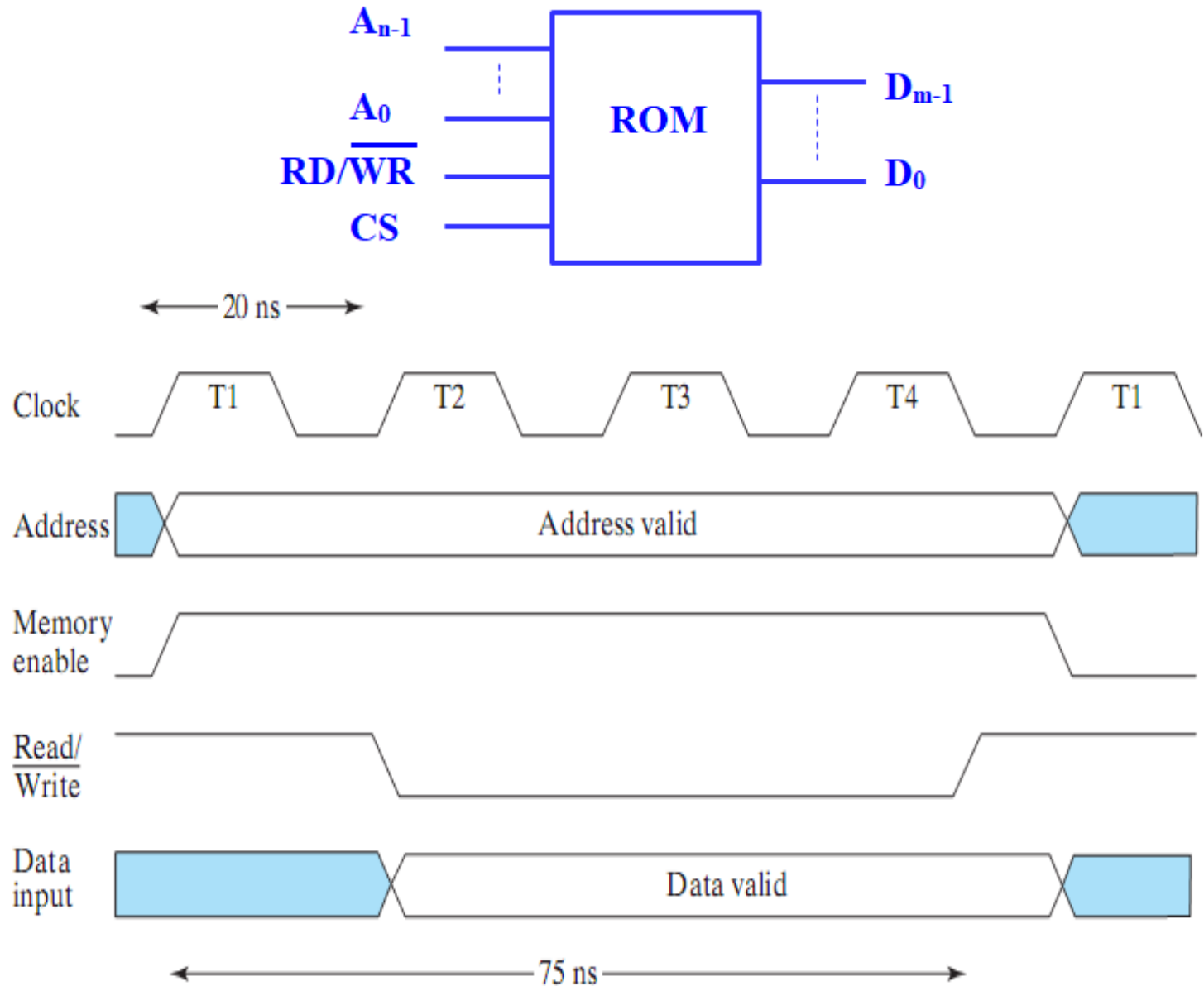
Ví dụ



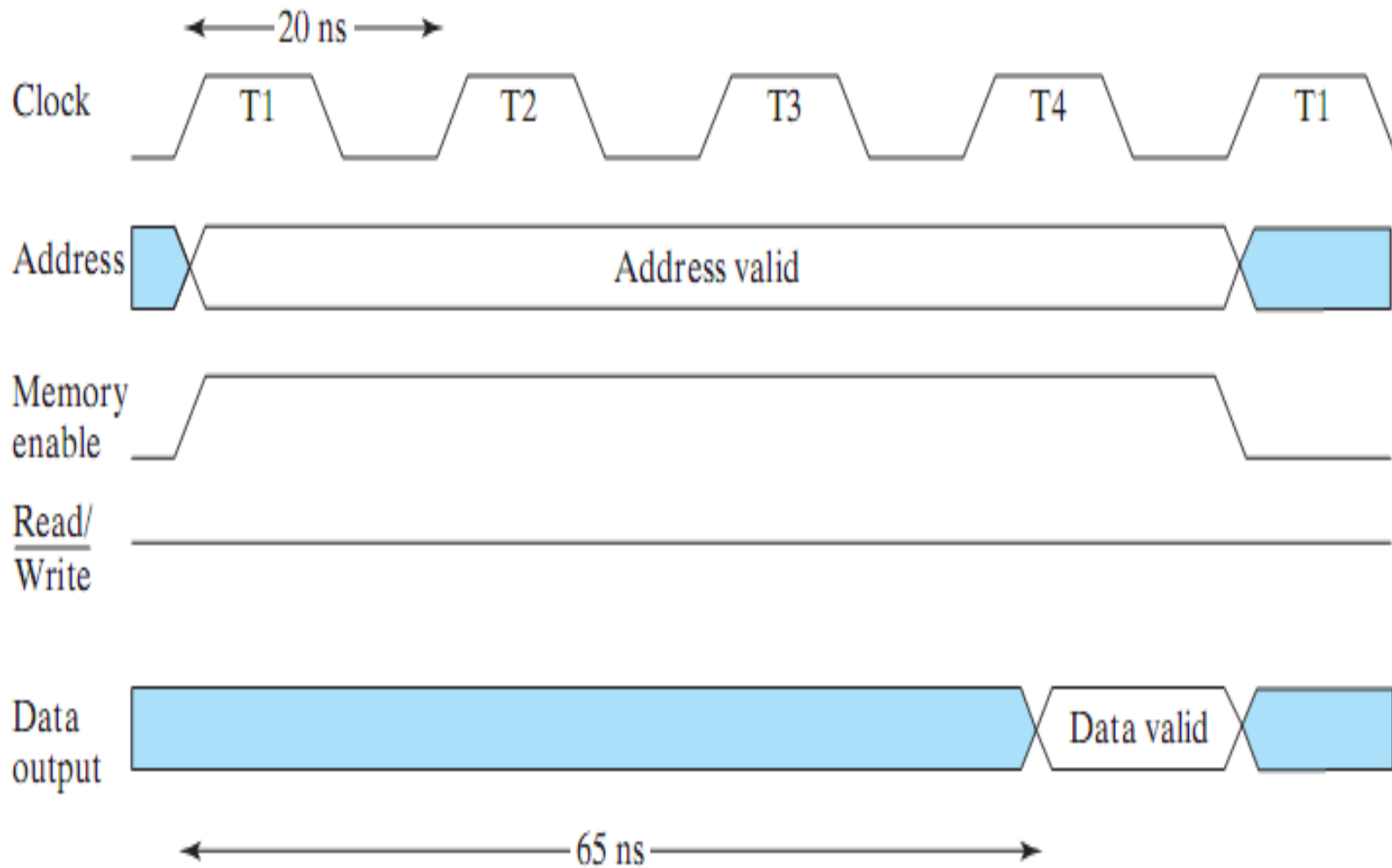
Inputs		Outputs	
Address($A_2 A_1 A_0$)		Data($D_1 D_0$)	
000		0	1
001		1	1
010		0	1
011		1	0
100		1	0
101		1	0
110		0	1
111		0	1

Kích thước(dung lượng): $2^3 \times 2=16$ (bit)=2BYTE

Giải đồ định thì cho một chu kỳ ghi bộ nhớ



Giải đồ định thì cho một chu kỳ đọc bộ nhớ



ỨNG DỤNG ROM TRONG THIẾT KẾ HÀM BOOLE

VD: Sử dụng **ROM** để thiết kế 2 hàm **Boole** sau:

$$F1(X,Y,Z)=\Sigma(0,1,4,7)$$

$$F2(A,B,C)=\Pi(1,2,5,7)$$

Yêu cầu thông số về **ROM**?

- Số tín hiệu địa chỉ ở ngõ vào (**Addr**)?
- Số đường dữ liệu ở ngõ ra (**Data**)?
- Kích thước?

ROM: $2^3 \times 2 \Rightarrow$ Dung lượng 16(bit)

Áp dụng

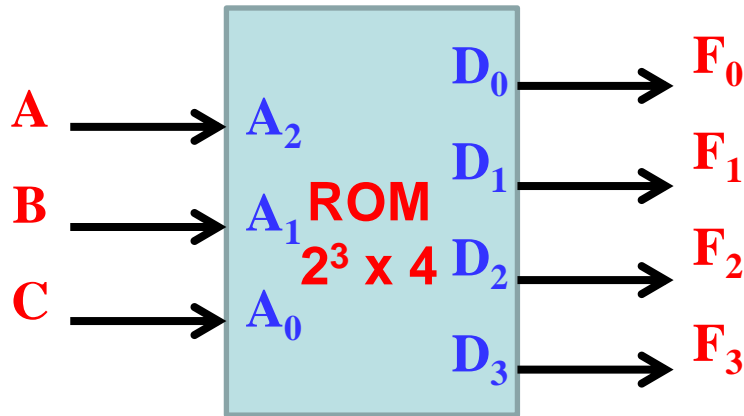
Thực hiện các hàm Boole được cho theo bảng chân trị sau bằng cách sử dụng ROM.

A	B	C	F3	F2	F1	F0
0	0	0	0	1	0	1
0	0	1	0	1	0	1
0	1	0	1	1	1	0
0	1	1	1	0	1	0
1	0	0	0	0	1	1
1	0	1	1	0	0	0
1	1	0	1	1	1	1
1	1	1	1	0	1	0

Phân tích

- Số biến chính là số đường địa chỉ (addr).
- Số hàm chính là số đường dữ liệu (data).

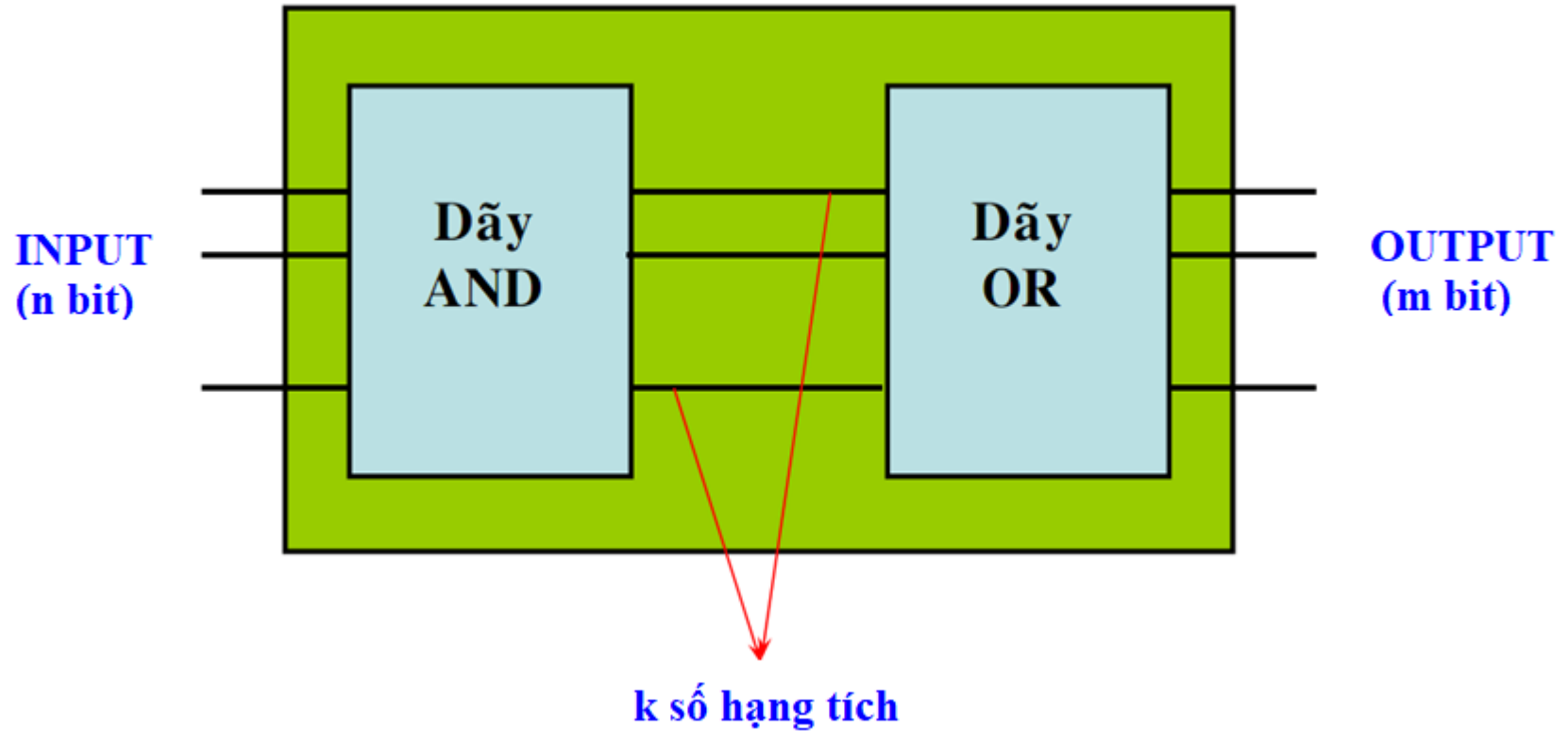
⇒ Cần ROM có dung lượng: $2^3 \times 4$ (bit)

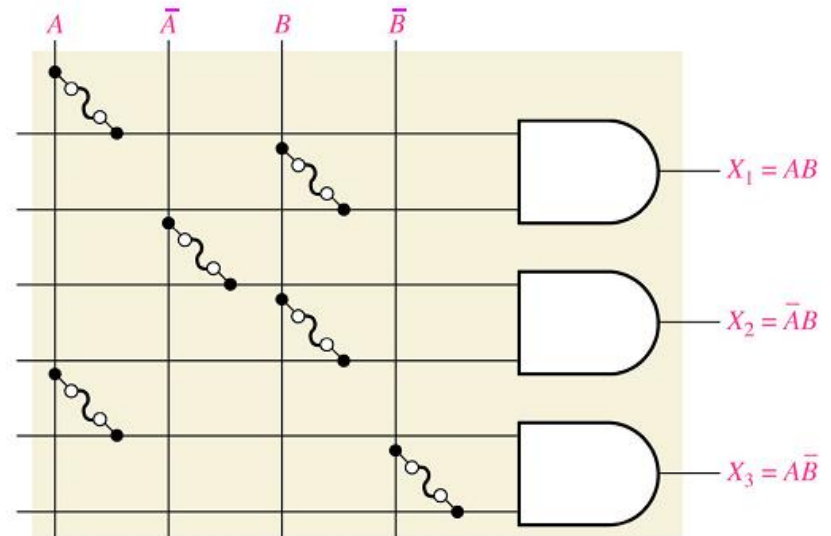
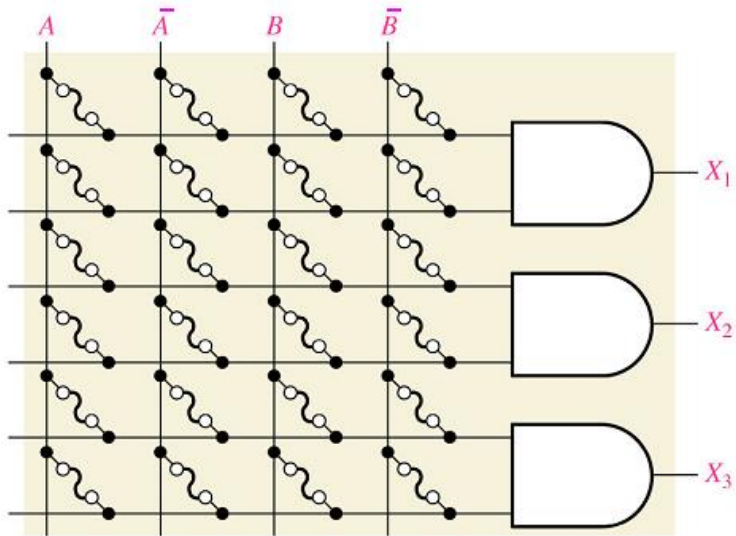


⇒ Nội dung nạp vào ROM?

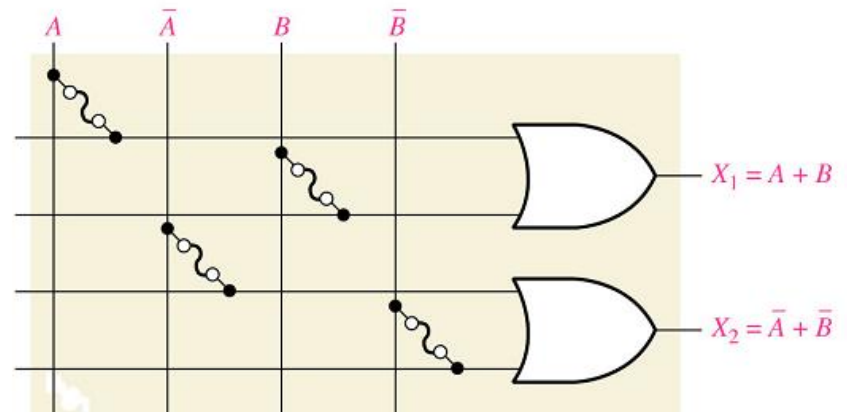
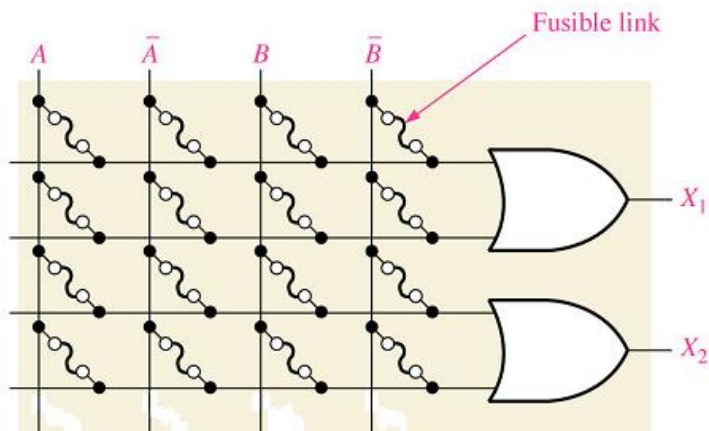
A	B	C	F₃	F₂	F₁	F₀
A_2	A_1	A_0	D_3	D_2	D_1	D_0
0	0	0	0	1	0	1
0	0	1	0	1	0	1
0	1	0	1	1	1	0
0	1	1	1	0	1	0
1	0	0	0	0	1	1
1	0	1	1	0	0	0
1	1	0	1	1	1	1
1	1	1	1	0	1	0

PLA (Programmable Logic Array)



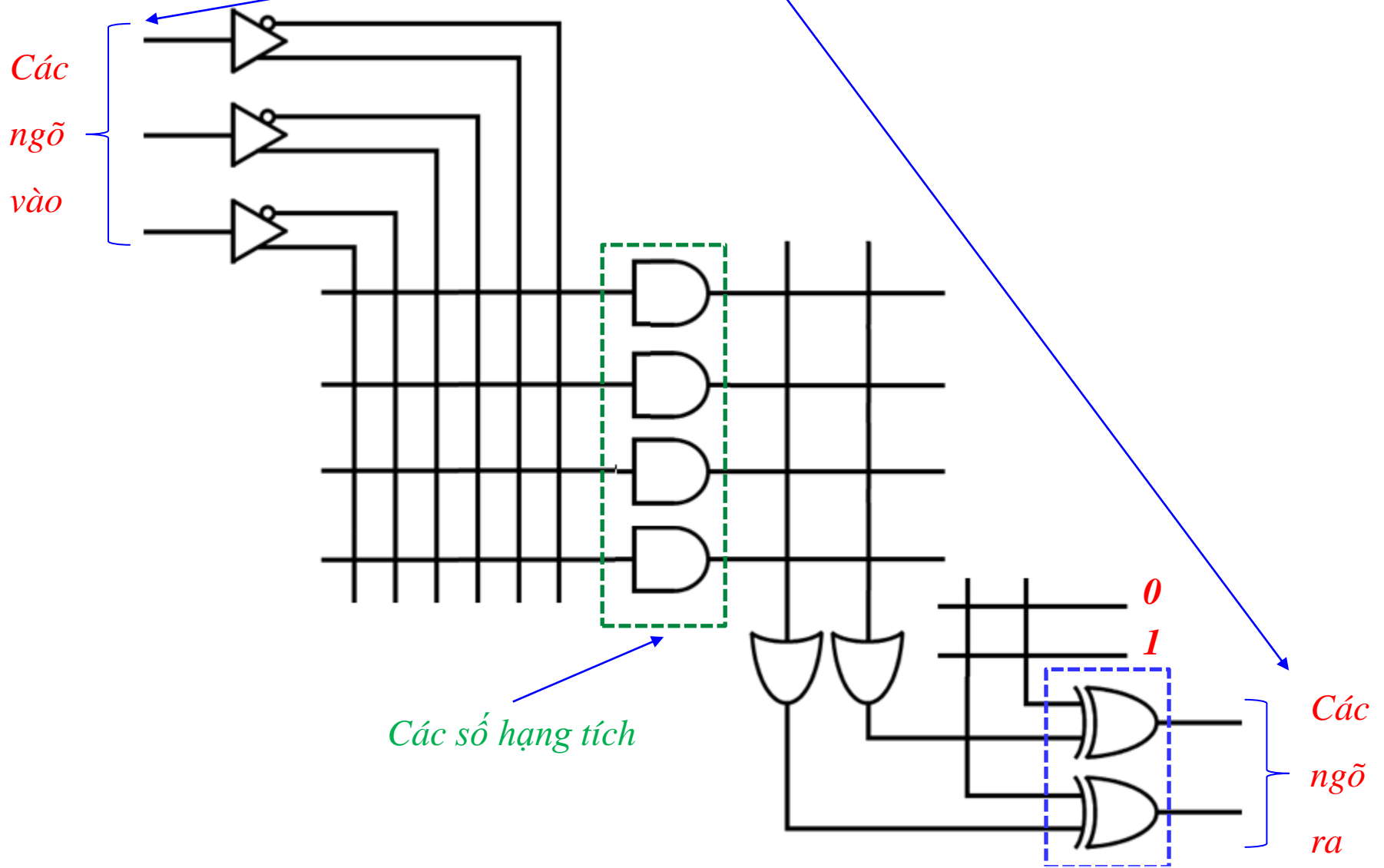


Dãy AND có thể lập trình



Dãy OR có thể lập trình

Giới thiệu vi mạch **PLA 3 x 2, 4** số hạng tích

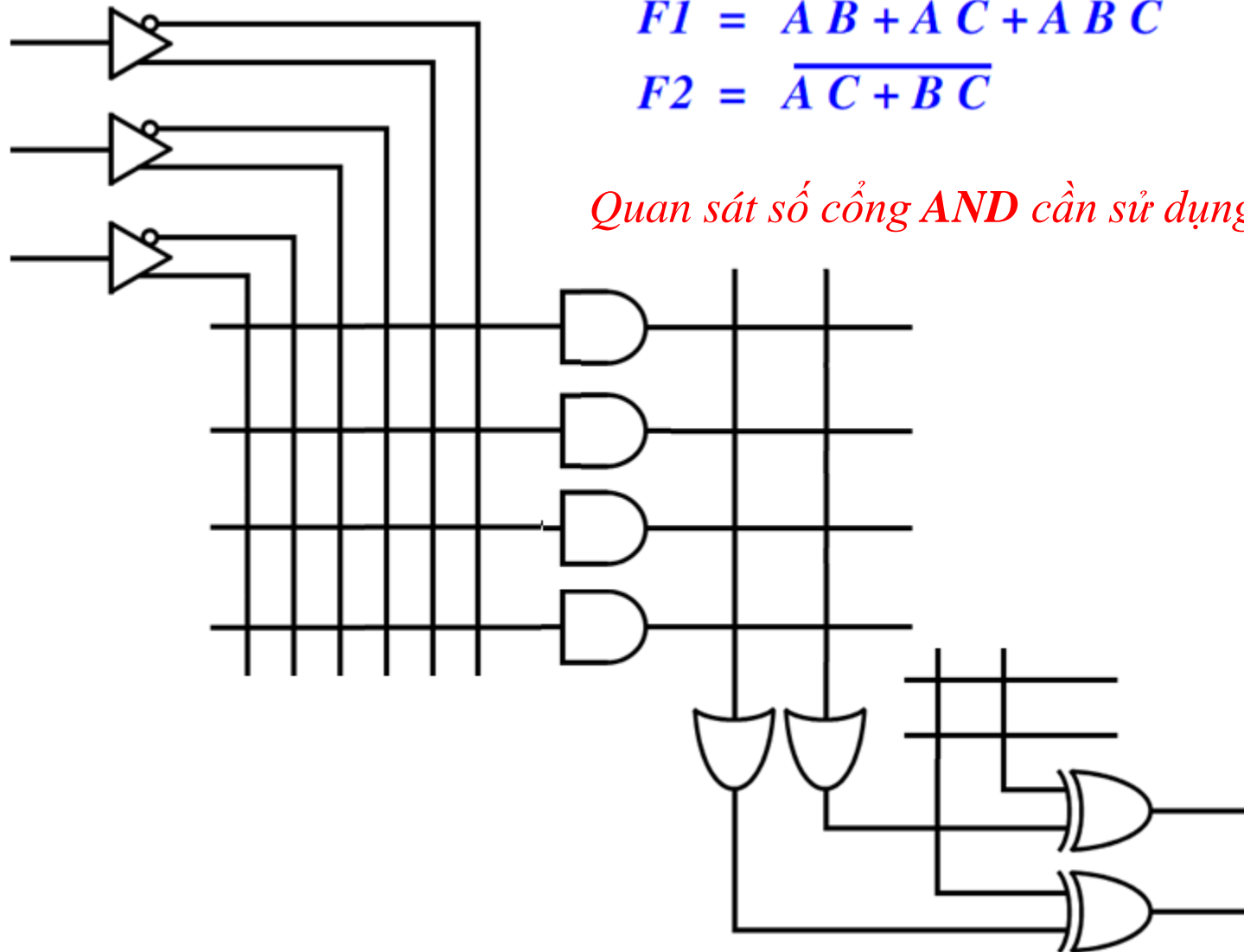


VD1: Sử dụng vi mạch **PLA 3 x 2**, 4 số hạng tích thực hiện 2 hàm sau:

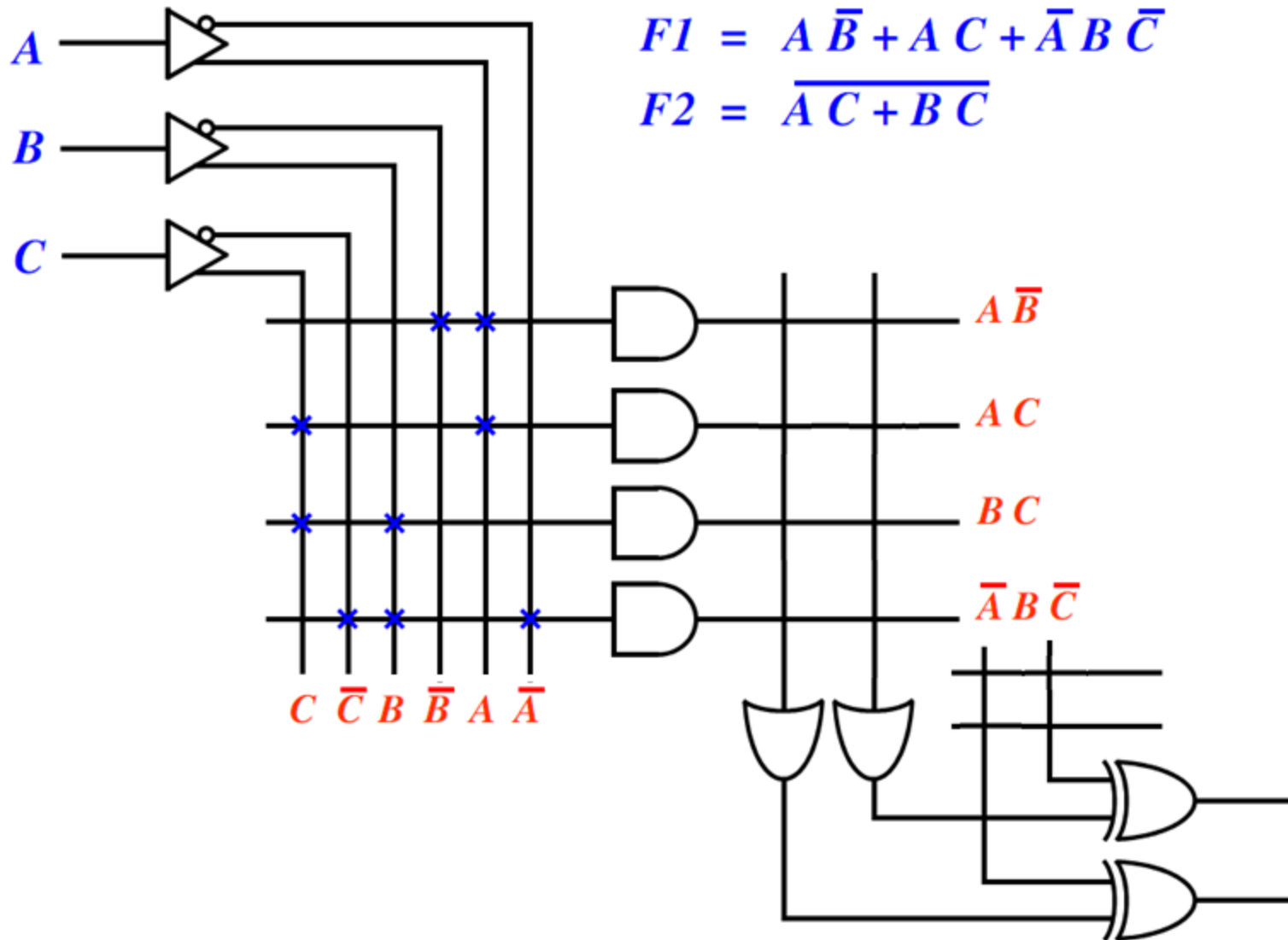
$$F1 = A \bar{B} + A C + \bar{A} B \bar{C}$$

$$F2 = \overline{A C + B C}$$

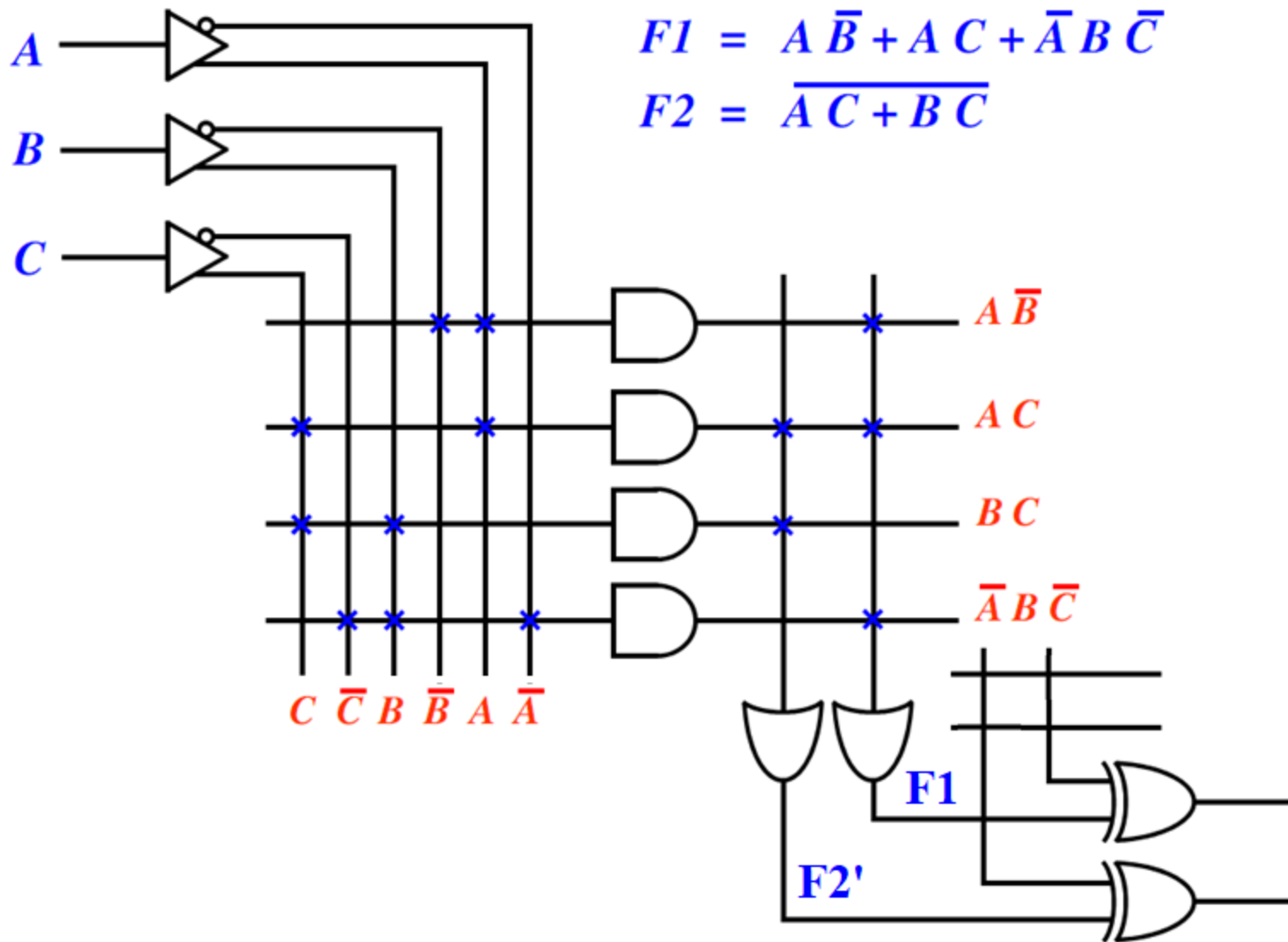
Quan sát số cổng AND cần sử dụng?



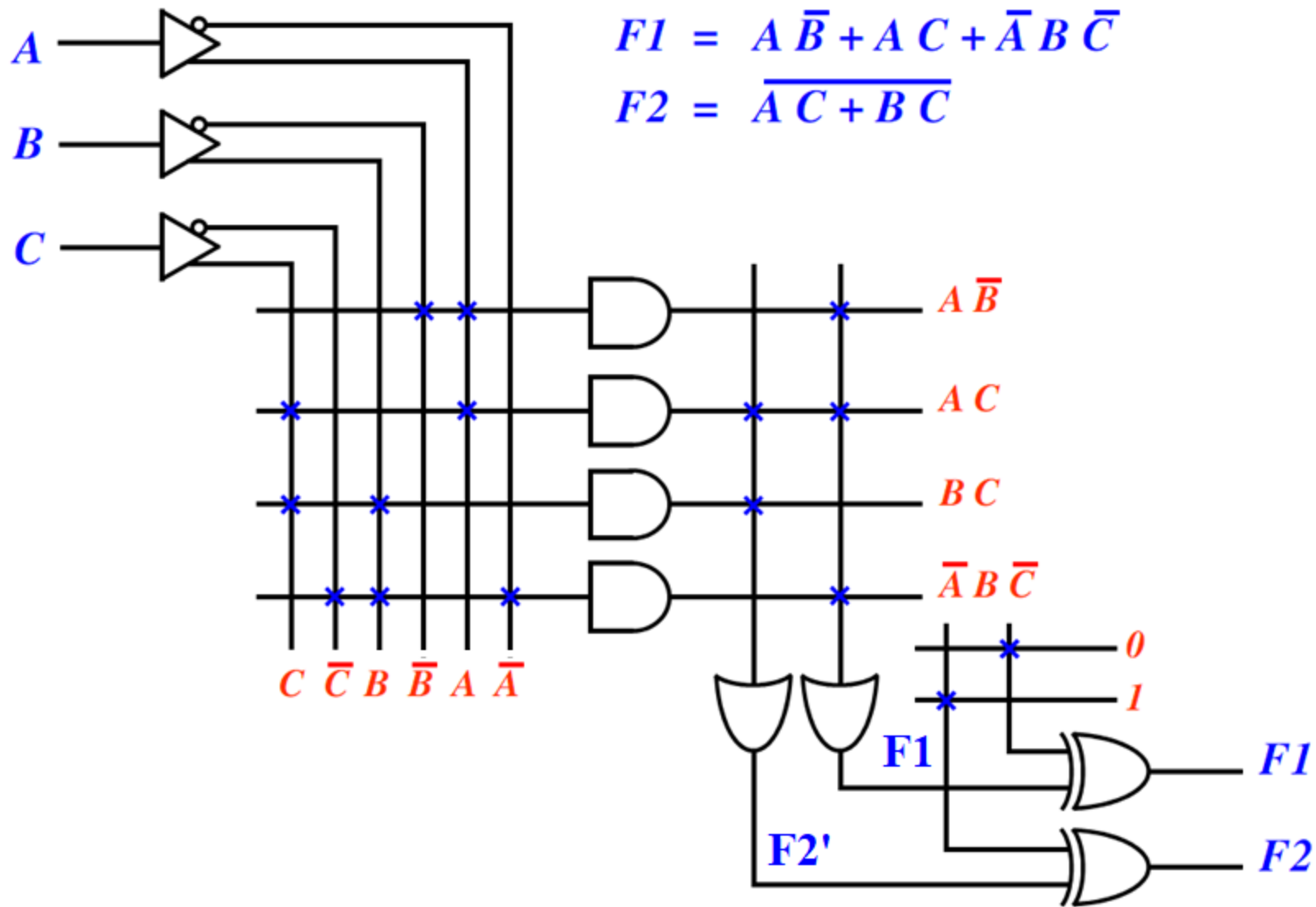
VD1: Sử dụng vi mạch **PLA 3 x 2**, **4 số hạng tích** thực hiện 2 hàm sau:



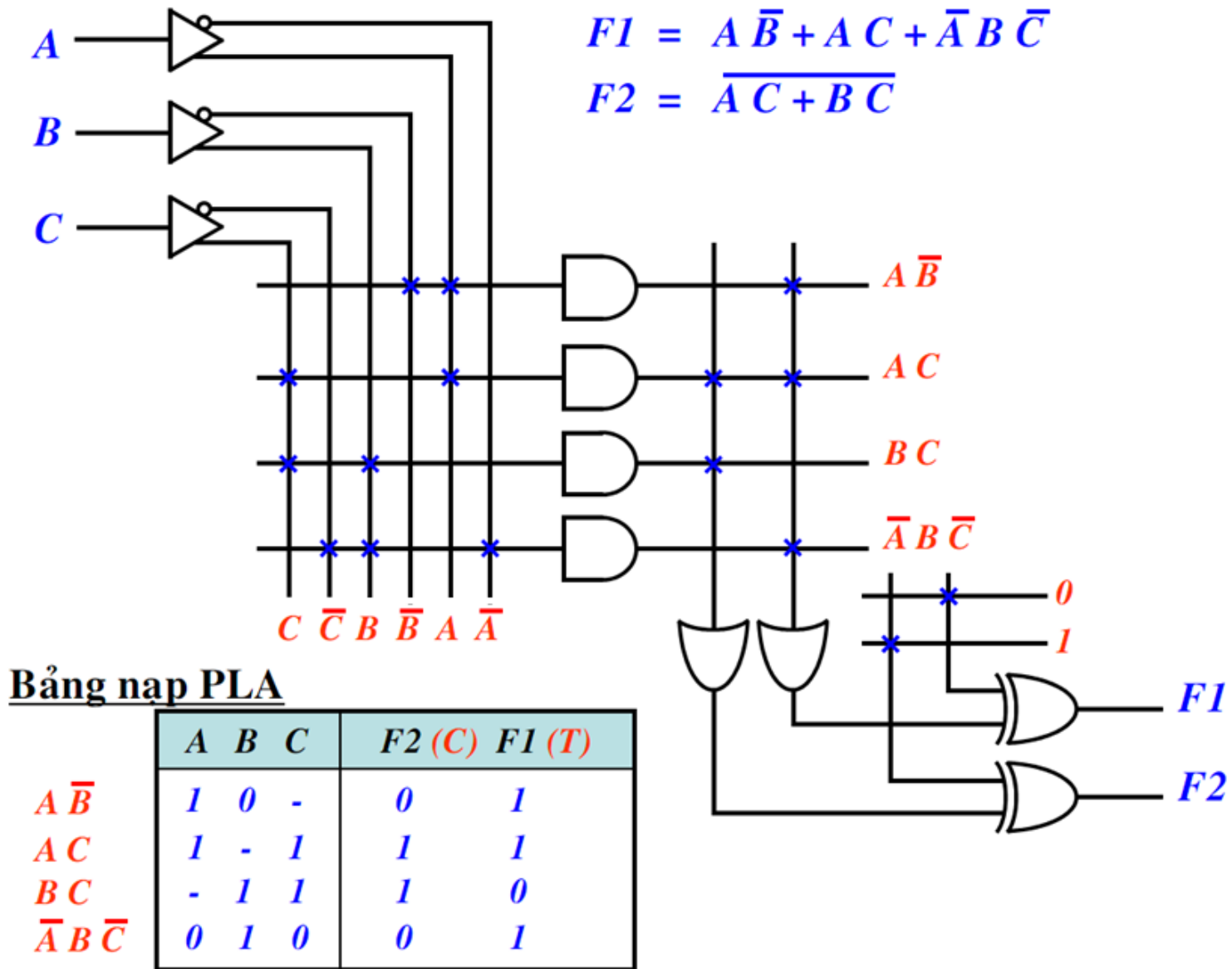
VD1: Sử dụng vi mạch **PLA 3 x 2**, 4 số hạng tích thực hiện 2 hàm sau:



VD1: Sử dụng vi mạch **PLA 3 x 2**, 4 số hạng tích thực hiện 2 hàm sau:



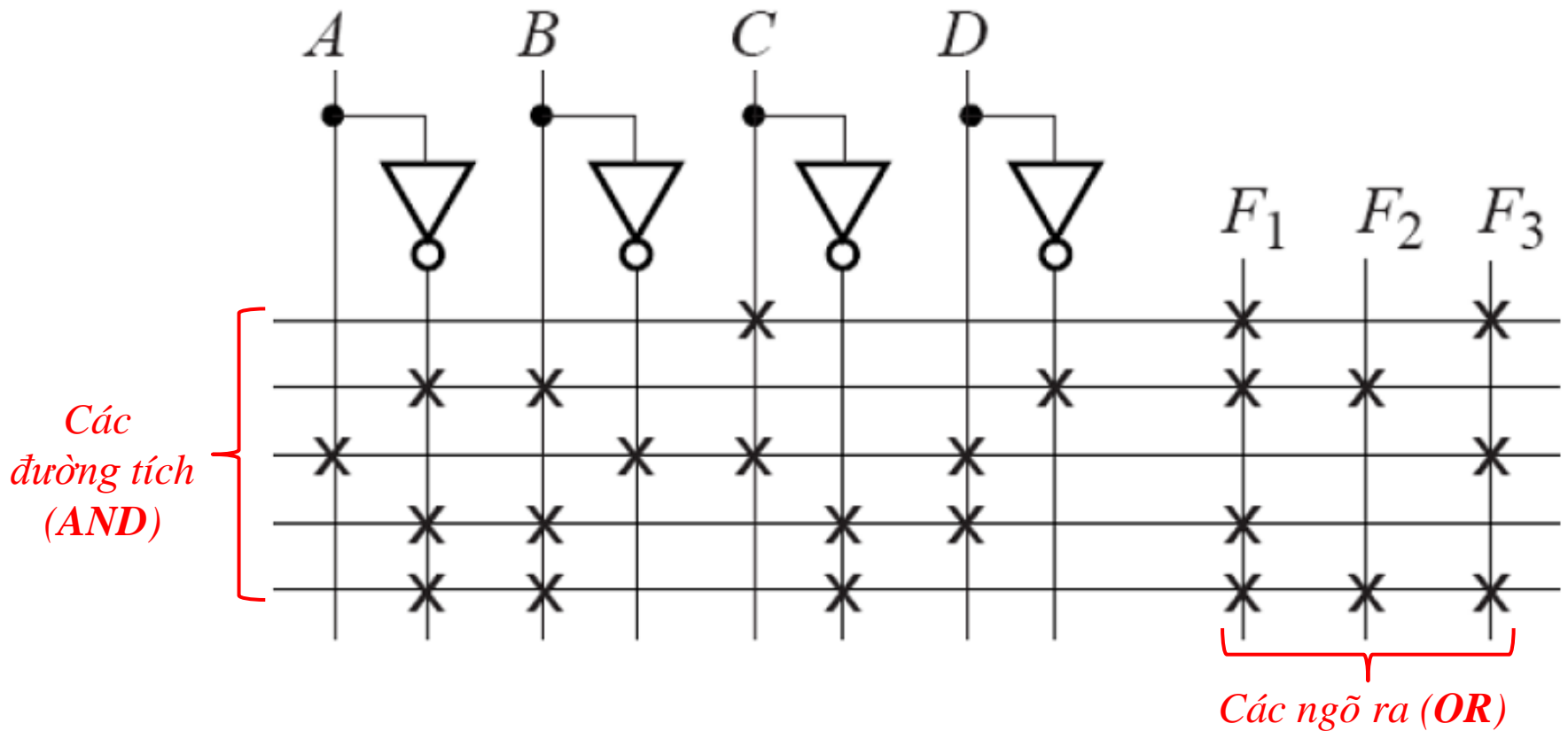
VD1: Sử dụng vi mạch **PLA 3 x 2**, **4 số hạng tích** thực hiện 2 hàm sau:



VD2

Cho sơ đồ kết nối PLA như hình.

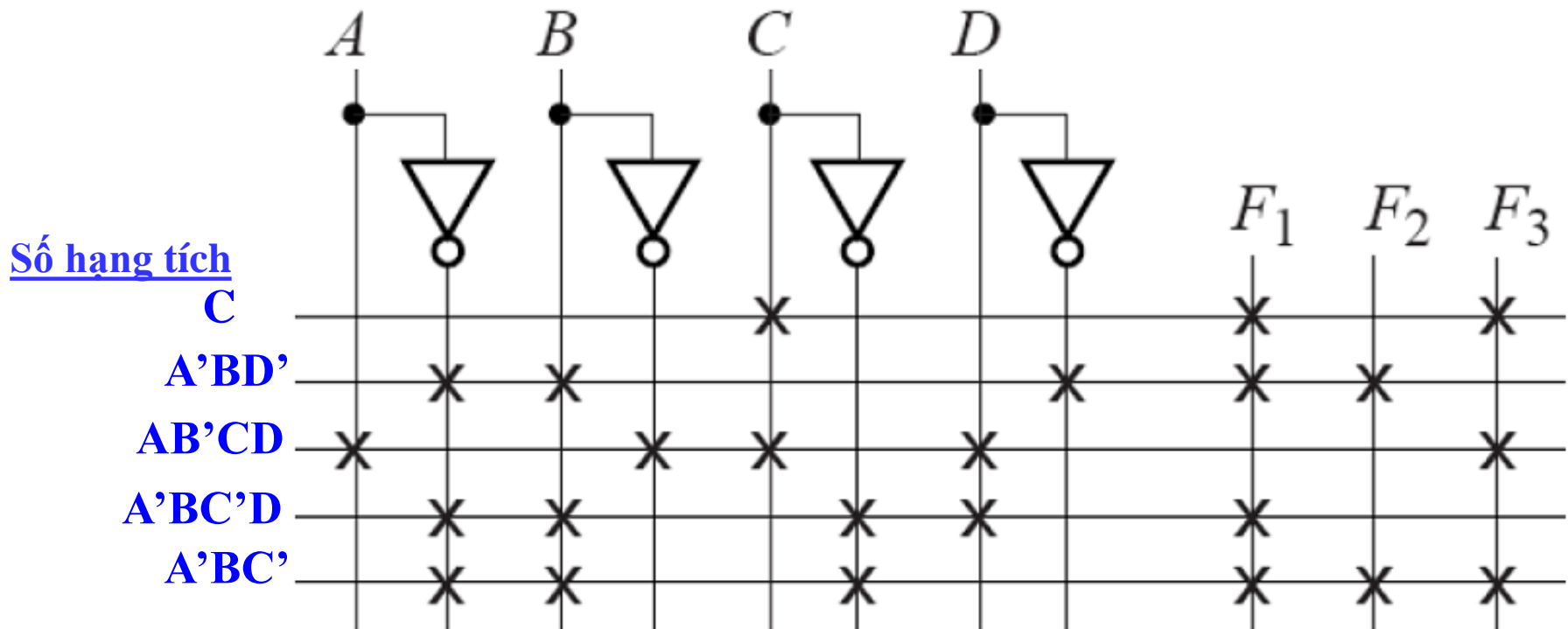
- Lập bảng PLA cho hệ thống này.
- Viết các phương trình của F_1, F_2 và F_3 .
- Nếu thay PLA bằng ROM thì cần ROM kích thước bao nhiêu?



VD2

Cho sơ đồ kết nối PLA như hình.

- Lập bảng PLA cho hệ thống này.
- Viết các phương trình của F_1, F_2 và F_3 .
- Nếu thay PLA bằng ROM thì cần ROM kích thước bao nhiêu?



Lập bảng PLA

Số hạng tích	Ngõ vào				Ngõ ra		
	A	B	C	D	F1	F2	F3
C	-	-	1	-	1	0	1
A'BD'	0	1	-	0	1	1	0
AB'CD	1	0	1	1	0	0	1
A'BC'D	0	1	0	1	1	0	0
A'BC'	0	1	0	-	1	1	1

Các phương trình

$$F_1 = C + A'BD' + A'BC'D + A'BC'$$

$$F_2 = A'BD' + A'BC'$$

$$F_3 = C + AB'CD + A'BC'$$

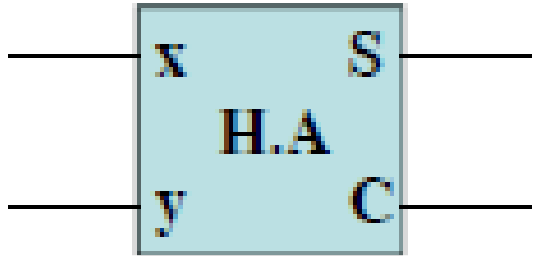
Nếu thay bằng **ROM** \Rightarrow *Kích thước ROM?*

ROM $2^4 \times 3$: 48(BIT) = 6 (BYTE)

MẠCH CỘNG

CỘNG BÁN PHẦN (HALF ADDER - HA)

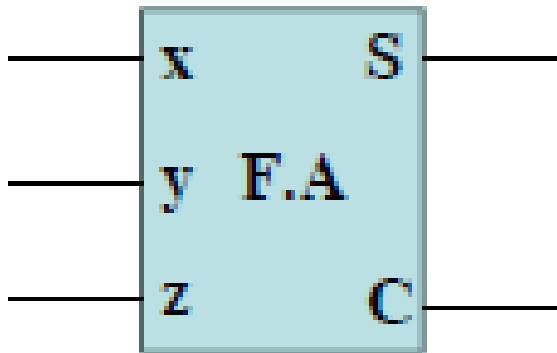
Mạch thực hiện phép cộng 2 số nhị phân 1 bit



$$S = x'y + xy' = x \oplus y$$
$$C = xy$$

CỘNG TOÀN PHẦN (FULL ADDER - FA)

Mạch thực hiện phép cộng 3 số nhị phân 1 bit

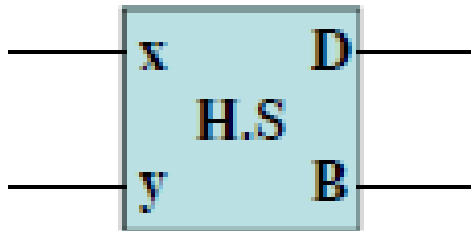


$$S = x \oplus y \oplus z$$
$$C = xy + xz + yz$$
$$= xy + z(x \oplus y)$$

MẠCH TRỪ

TRỪ BÁN PHẦN (HALF SUBTRACTOR – HS)

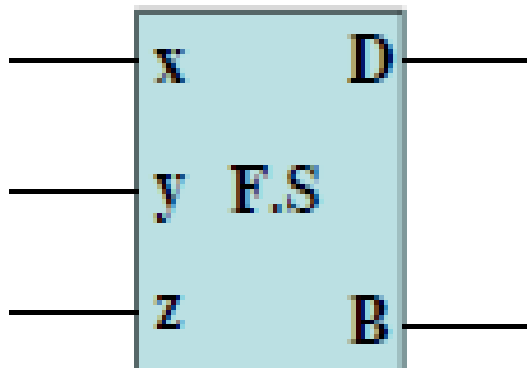
Mạch thực hiện phép trừ 2 số nhị phân 1 bit



$$D = x'y + xy' = x \oplus y$$
$$B = x'y$$

TRỪ TOÀN PHẦN (FULL SUBTRACTOR – FS)

Mạch thực hiện phép trừ 3 số nhị phân 1 bit

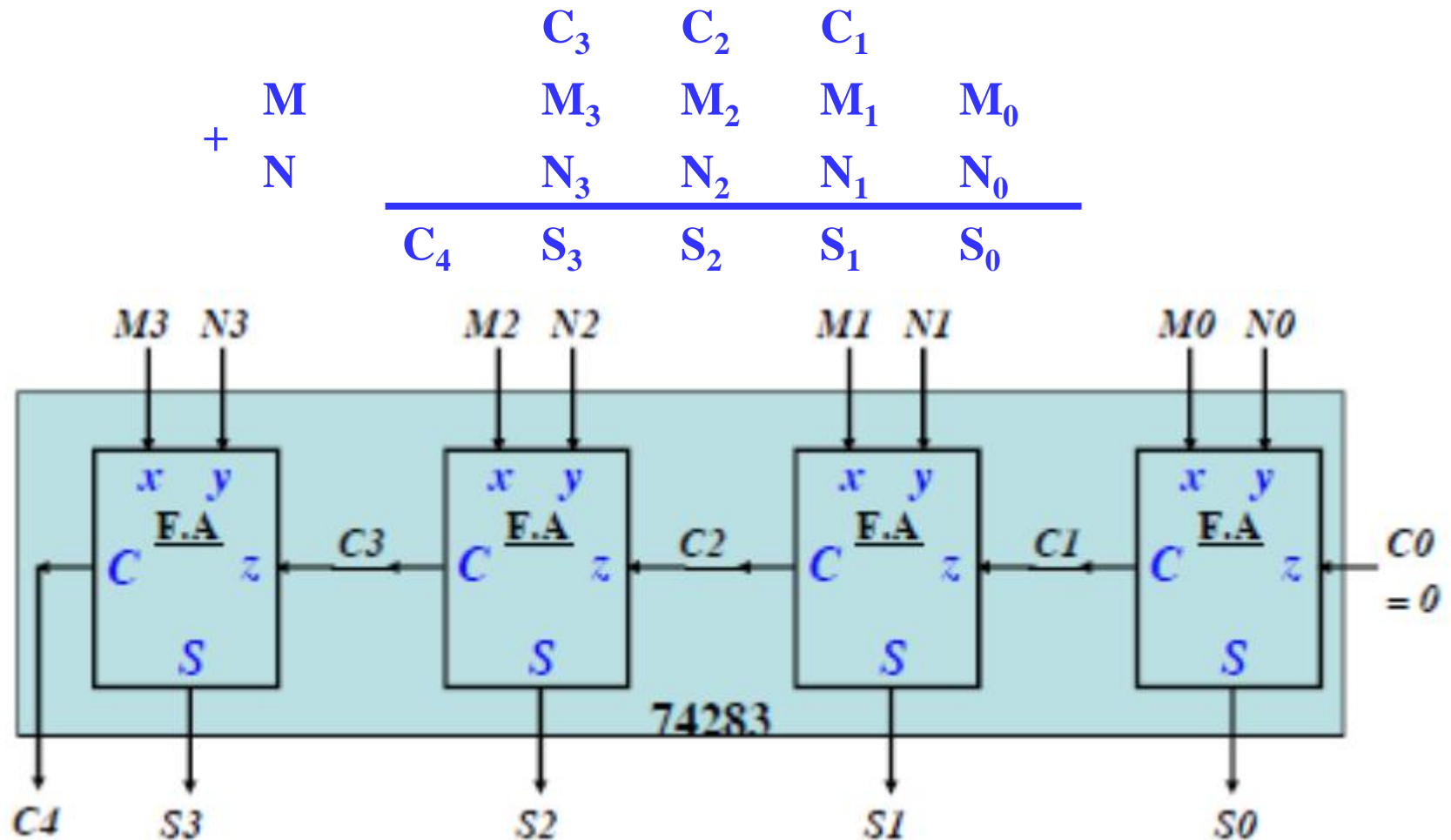


$$D = x \oplus y \oplus z$$
$$B = x'y + x'z + yz$$
$$= x'y + z(x \oplus y)'$$

MẠCH CỘNG / TRỪ SONG SONG

Mạch thực hiện phép cộng hoặc trừ 2 số nhị phân n bit

a. Bộ cộng nhị phân

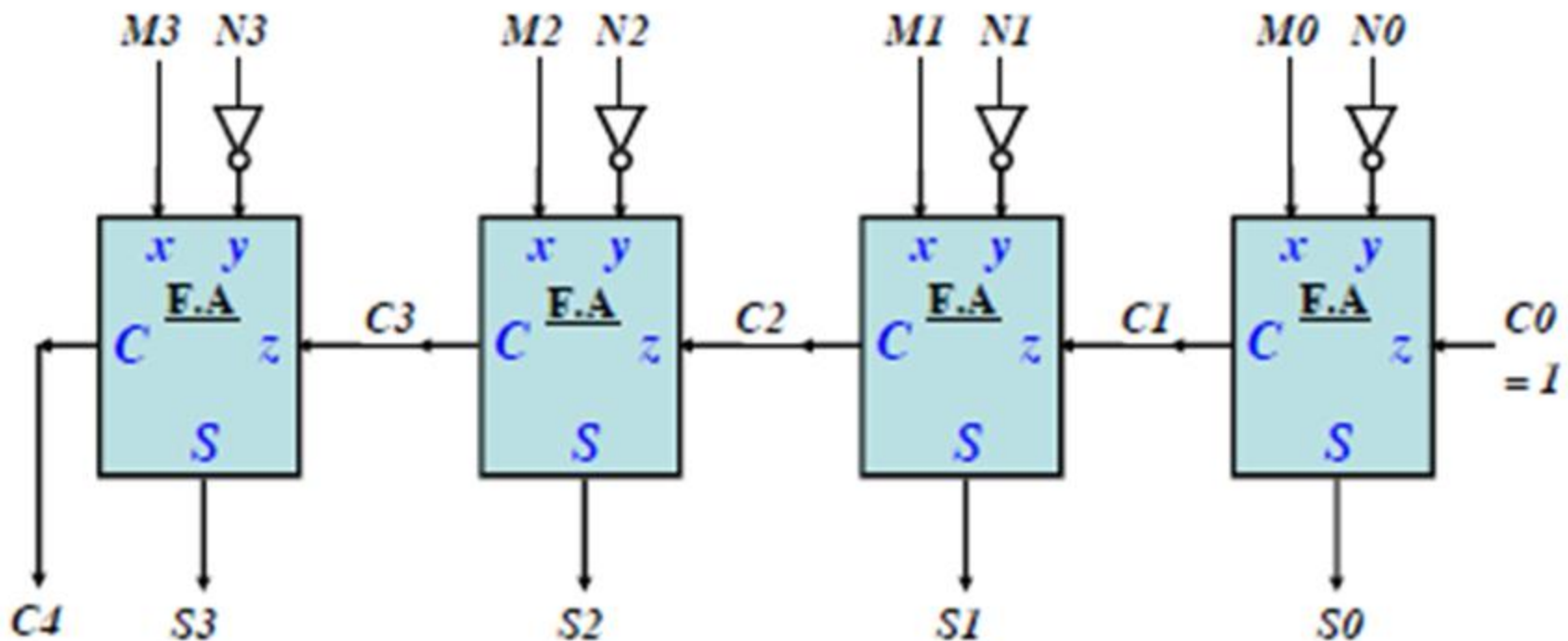


b. Bộ trừ nhị phân

- Sử dụng các bộ trừ toàn phần FS.
- Hoặc Thực hiện bằng phép cộng với bù 2 của số trừ.

$$M - N = M + \text{bù}_2(N) = M + \text{bù}_1(N) + 1$$

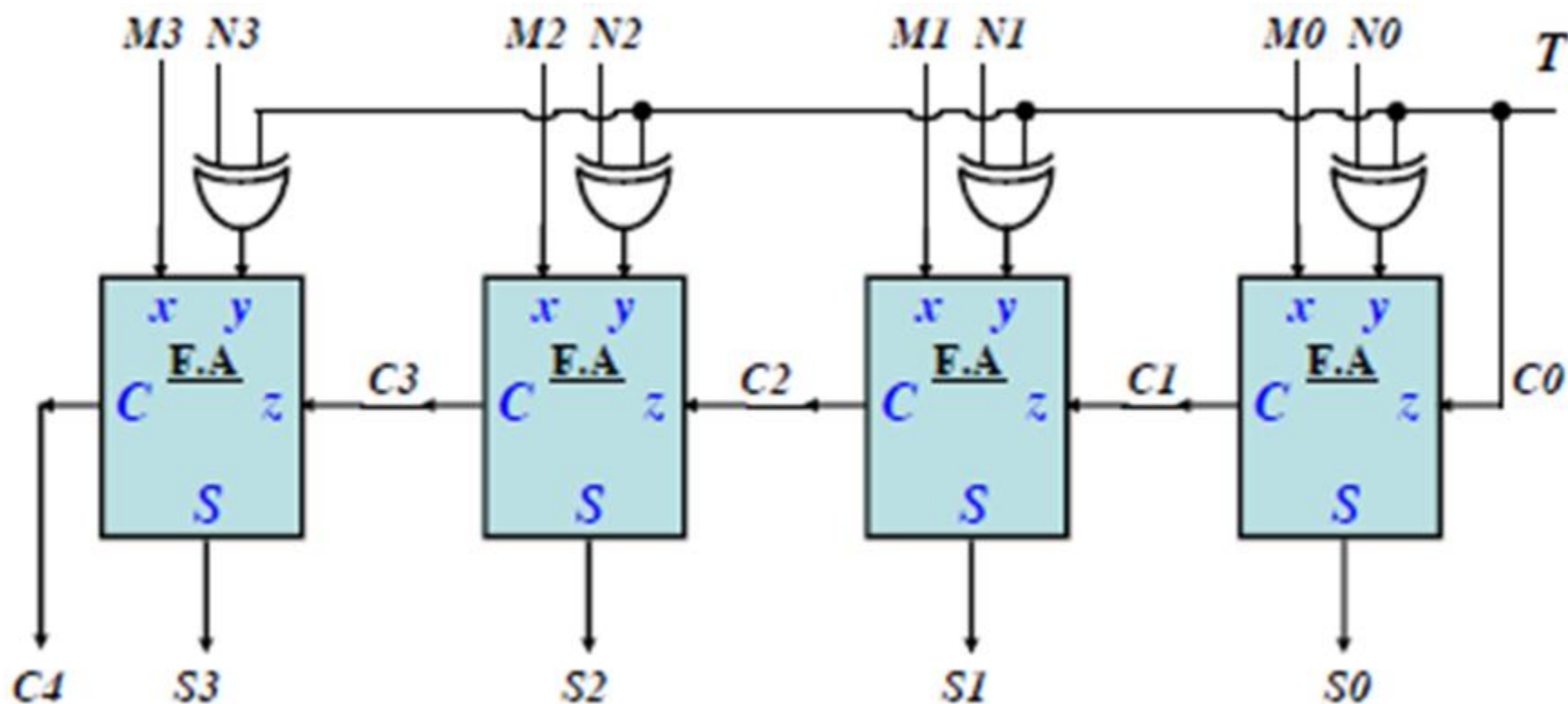
- Kết quả: $C4 = 1 \Rightarrow$ Kết quả là số dương.
 $C4 = 0 \Rightarrow$ Kết quả là số âm.



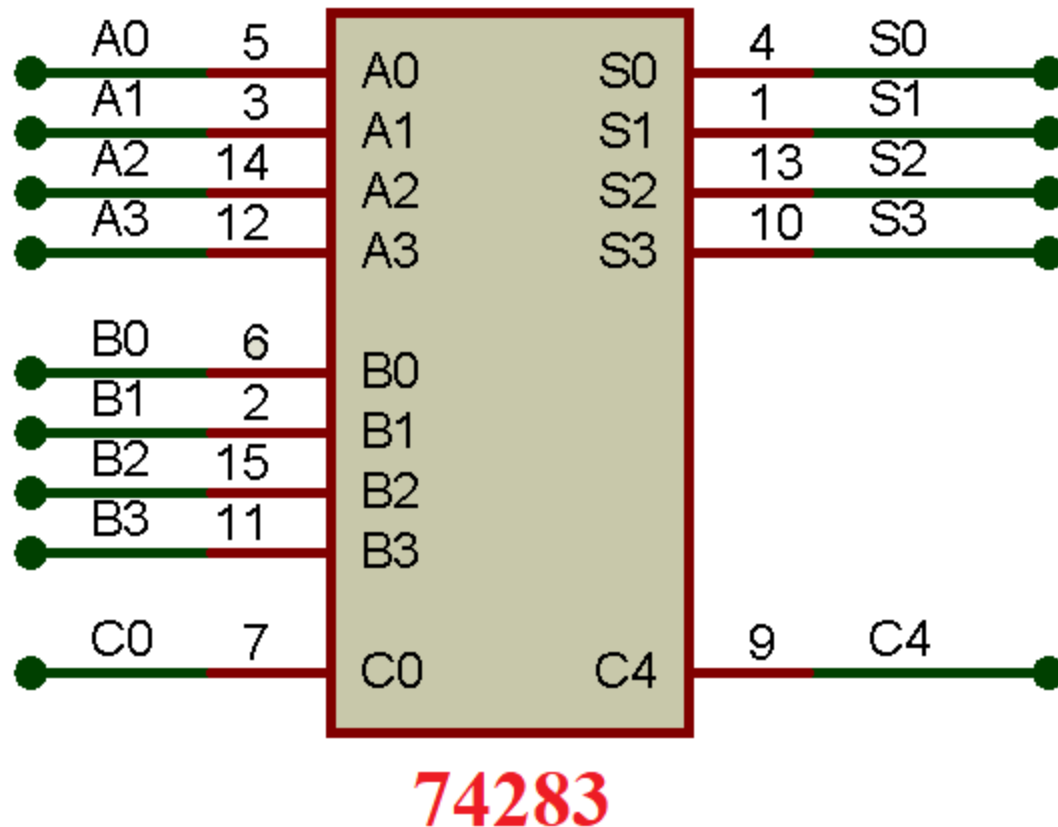
c. Thiết kế Bộ cộng/trừ nhị phân

Phép toán	C_0	y_i
CỘNG	0	N_i
TRỪ	1	$(N_i)'$

Ngõ vào điều khiển	
T=0: cộng	$C_0 = T$
T=1: trừ	$y_i = T \oplus N_i$



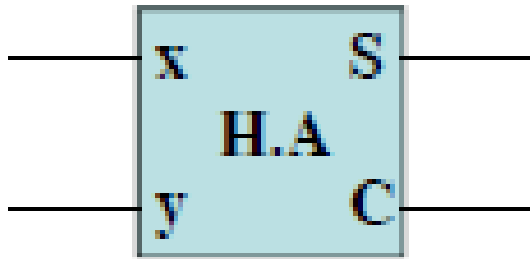
IC cộng 4 bit 74283(74LS283)



ỨNG DỤNG TRONG THIẾT KẾ HÀM BOOLE

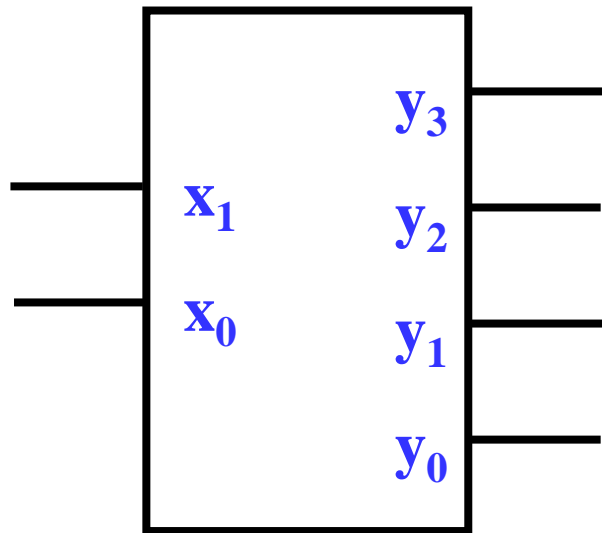
Sử dụng vi mạch cộng để thực hiện hàm

VD1 Chỉ sử dụng một H.A, hãy thiết kế hệ tổ hợp thực hiện hàm $Y = x^2 + x + 1$, với x là số nhị phân 2 bit.



$$S = x'y + xy' = x \oplus y$$
$$C = xy$$

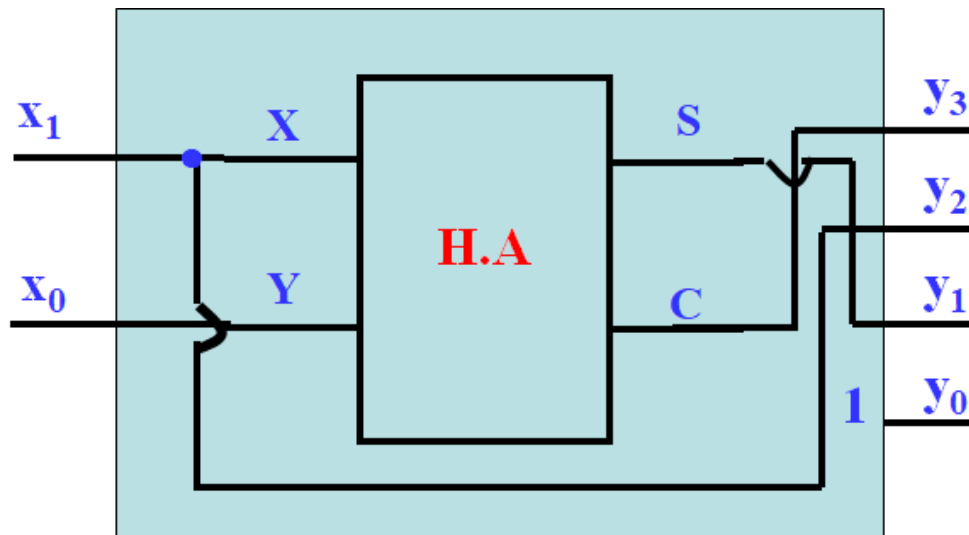
Sơ đồ khối: $Y = x^2 + x + 1$



Bảng chân trị

x_1	x_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	1
0	1	0	0	1	1
1	0	0	1	1	1
1	1	1	1	0	1

Mạch thiết kế



Các hàm ngõ ra

$$Y_0 = 1$$

$$Y_1 = x_1 \oplus x_0$$

$$Y_2 = x_1$$

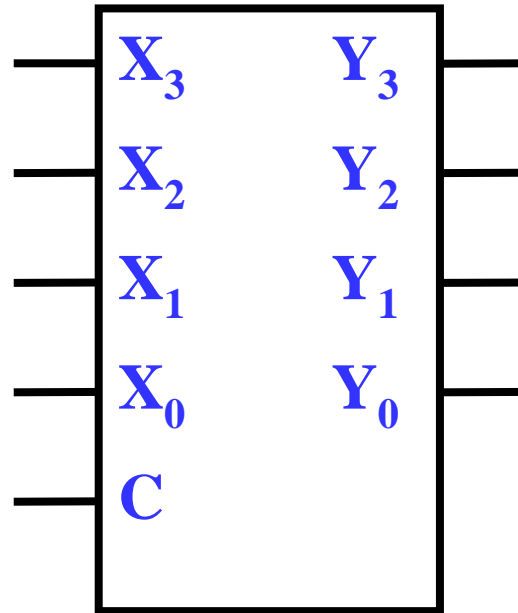
$$Y_3 = x_1 x_0$$

VD2 Dùng vi mạch 74283 (mạch cộng 4 bit) và các cổng logic (nếu cần) để thiết kế mạch tổ hợp có hoạt động như sau:

Nếu $C=0$ thì $Y_3Y_2Y_1Y_0 = X_3X_2X_1X_0$

Nếu $C=1$ thì $Y_3Y_2Y_1Y_0 = \text{bù 2 của } X_3X_2X_1X_0$

Sơ đồ khối



Phân tích yêu cầu

$$C = 0: \quad Y = X \quad = X \quad + 0 + 0$$

$$C = 1: \quad Y = \text{bù}_2(X) = \text{bù}_1(X) + 1 \quad = \text{bù}_1(X) \quad + 0 + 1$$

$$\text{IC74283} \quad S = A + B + C_0$$

Để có sự đồng nhất cho việc thiết kế: $Y = S$

$$\Rightarrow B = 0$$

C_0 giữ vai trò của C :

$$C_0 = 0 \Rightarrow A = X$$

$$C_0 = 1 \Rightarrow A = \text{bù}_1(X)$$

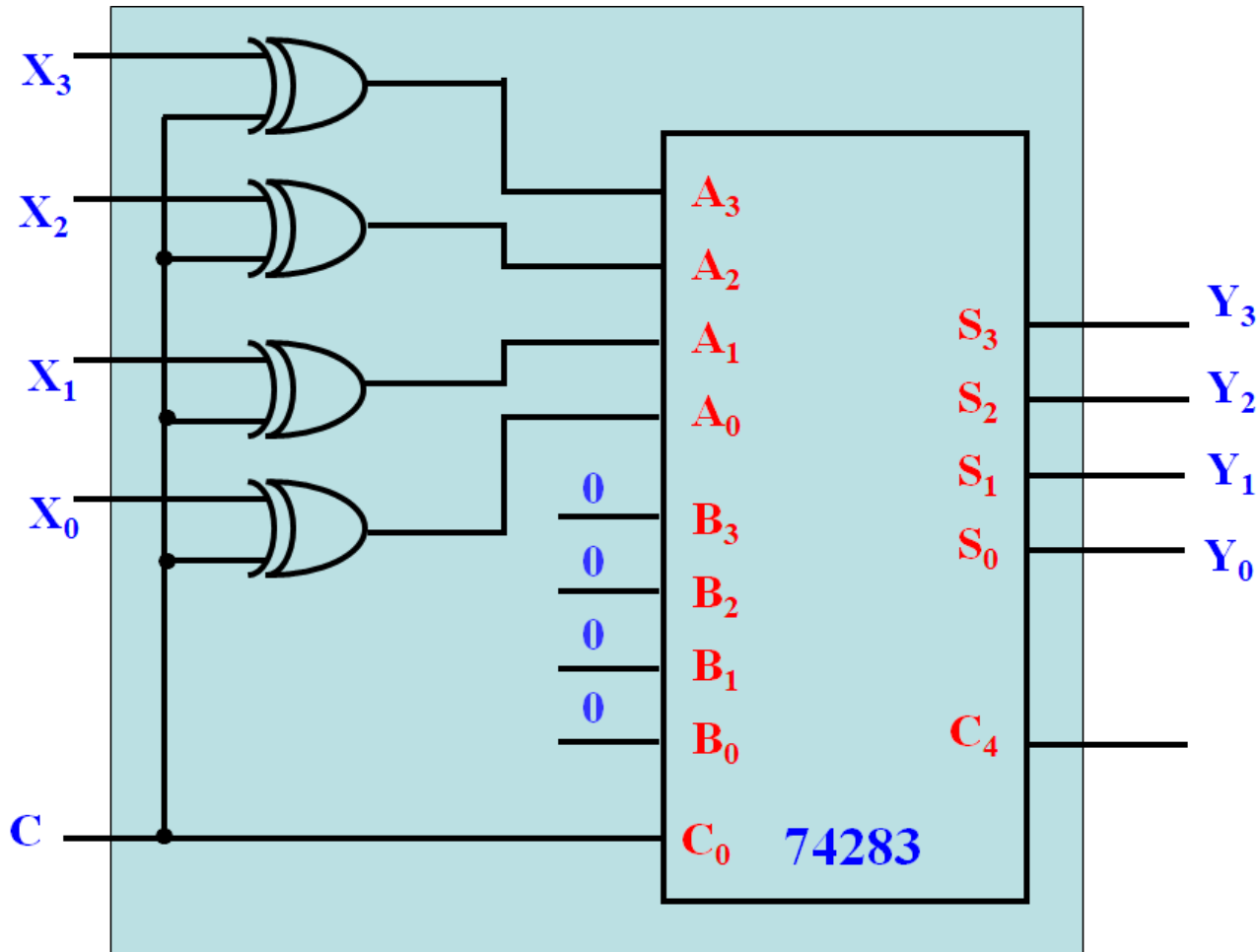
$$\Rightarrow A = X \oplus C$$

Kết luận:

$$A = X \oplus C; B = 0; C_0 = C; S = Y$$

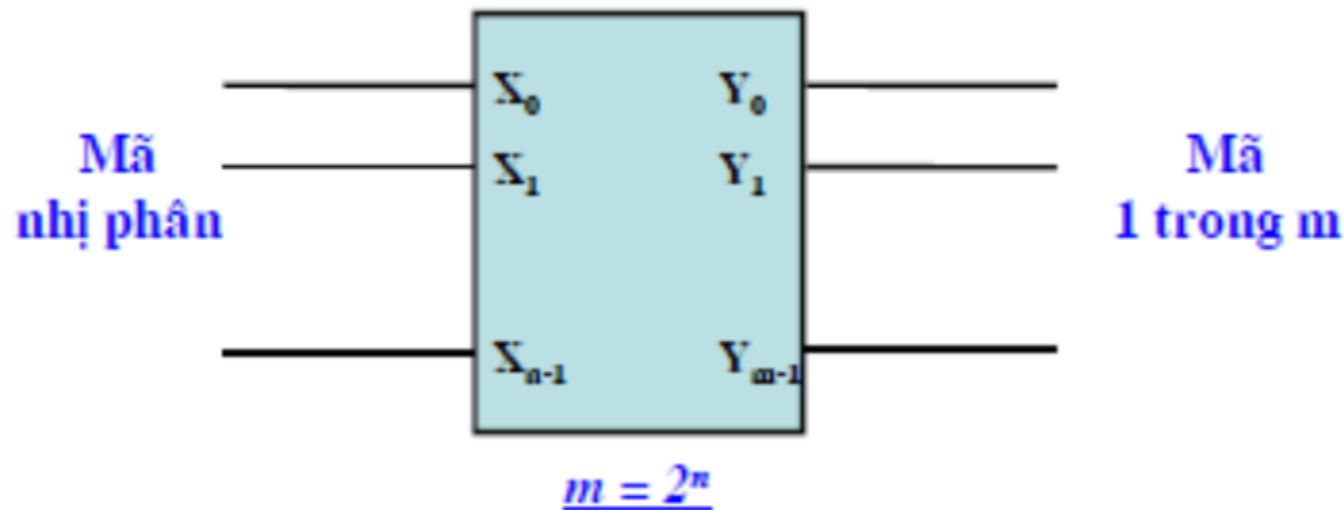
Mạch thiết kế

$$\mathbf{A} = \mathbf{X} \oplus \mathbf{C}; \mathbf{B} = \mathbf{0}; \mathbf{C}_0 = \mathbf{C}; \mathbf{S} = \mathbf{Y}$$

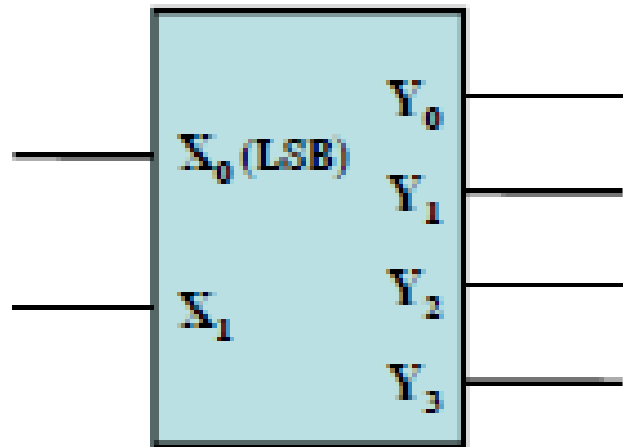


BỘ GIẢI MÃ (DECODER)

- Bộ giải mã là hệ chuyển mã có nhiệm vụ chuyển từ mã nhị phân cơ bản n bit ở ngõ vào thành mã nhị phân 1 trong m ở ngõ ra.
- Với giá trị i của tổ hợp nhị phân ở ngõ vào, thì ngõ ra Y_i sẽ tích cực và các ngõ ra còn lại sẽ không tích cực.
- Có 2 dạng: ngõ ra tích cực cao (mức 1) và ngõ ra tích cực thấp (mức 0).



Bộ giải mã ngõ ra tích cực cao



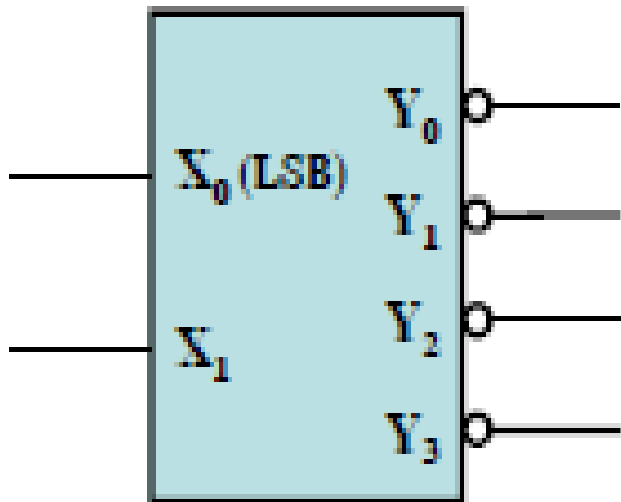
$$Y_0 = \overline{X_1} \overline{X_0} = m_0$$

$$Y_1 = \overline{X_1} X_0 = m_1$$

$$Y_2 = X_1 \overline{X_0} = m_2$$

$$Y_3 = X_1 X_0 = m_3$$

Bộ giải mã ngõ ra tích cực thấp



$$Y_0 = X_1 + X_0 = M_0 = \overline{m_0}$$

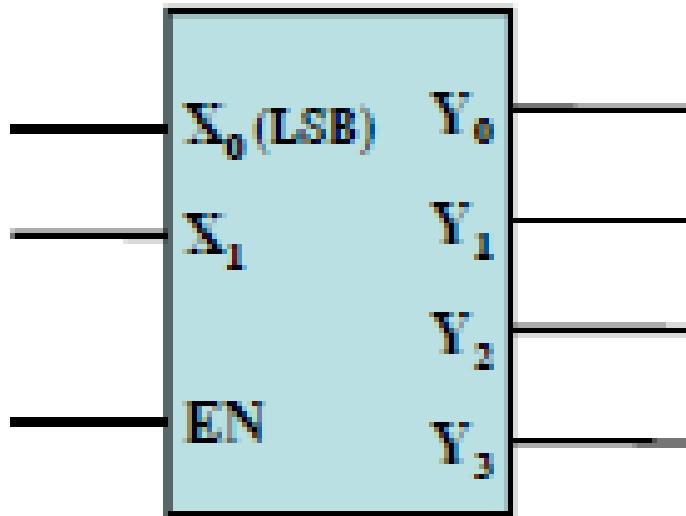
$$Y_1 = X_1 + \overline{X_0} = M_1 = \overline{m_1}$$

$$Y_2 = \overline{X_1} + X_0 = M_2 = \overline{m_2}$$

$$Y_3 = \overline{X_1} + \overline{X_0} = M_3 = \overline{m_3}$$

Bộ giải mã có ngõ vào cho phép

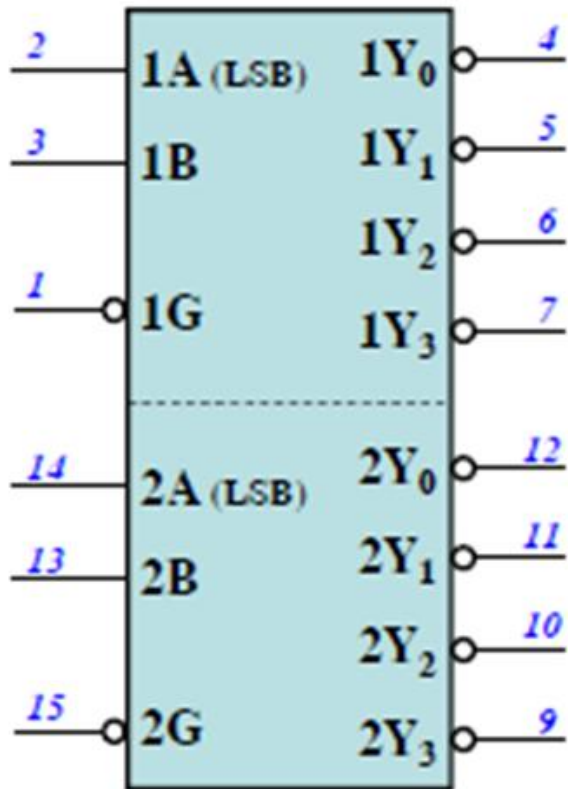
- Ngoài các ngõ vào dữ liệu, bộ giải mã có thể có 1 hay nhiều ngõ cho phép.
- Khi các ngõ vào cho phép ở trạng thái tích cực thì mạch giải mã mới được hoạt động. Ngược lại, mạch giải mã sẽ không hoạt động; khi đó các ngõ ra đều ở trạng thái không tích cực.



EN	X_1	X_0	Y_3	Y_2	Y_1	Y_0
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Giới thiệu một số IC giải mã

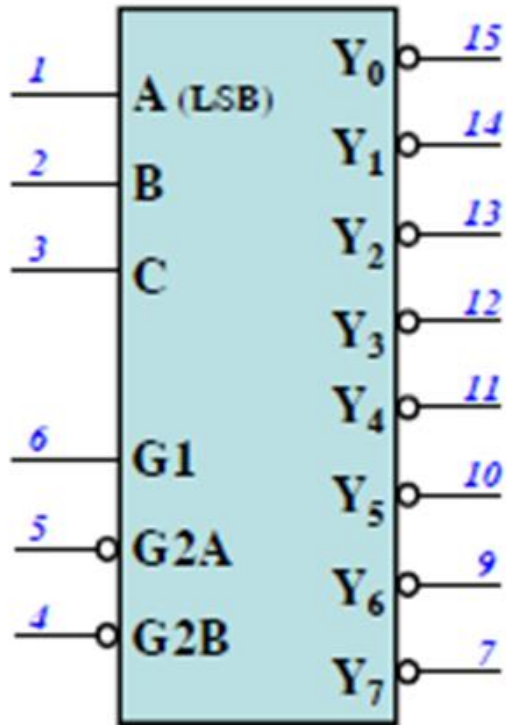
a. IC74139: gồm 2 bộ giải mã 2 sang 4, ngõ ra tích cực thấp



/G	B	A	/Y ₃	/Y ₂	/Y ₁	/Y ₀
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

Giới thiệu một số IC giải mã

b. IC74138: bộ giải mã 3 sang 8, ngõ ra tích cực thấp



G1	/G2A	/G2B	C	B	A	/Y ₇	/Y ₆	/Y ₅	/Y ₄	/Y ₃	/Y ₂	/Y ₁	/Y ₀
0	X	X	X	X	X	1	1	1	1	1	1	1	1
X	1	X	X	X	X	1	1	1	1	1	1	1	1
X	X	1	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1	0
1	0	0	0	0	1	1	1	1	1	1	1	0	1
1	0	0	0	1	0	1	1	1	1	1	0	1	1
1	0	0	0	1	1	1	1	1	1	0	1	1	1
1	0	0	1	0	0	1	1	1	0	1	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1	1	1
1	0	0	1	1	0	1	0	1	1	1	1	1	1
1	0	0	1	1	1	0	1	1	1	1	1	1	1

Sử dụng bộ giải mã để thực hiện hàm

- Mỗi ngõ ra của bộ giải mã tích cực cao là một tích chuẩn mi (minterm), ngược lại sẽ là một tổng chuẩn Mi(Maxterm). Do đó, ta có thể sử dụng bộ giải mã để thực hiện hàm Boole theo dạng chính tắc.

Nguyên tắc:

- Hàm cần thực hiện phải đưa về dạng chính tắc và kết hợp thêm công thức hợp để thiết kế.
- Các ngõ cho phép của bộ giải mã phải ở trạng thái tích cực.
- Các biến của hàm được nối đến các ngõ vào của bộ giải mã theo đúng vị trí trọng số.

VD

- Dùng IC74138 và 1 số cổng cần thiết để thực hiện 2 hàm Boole sau:
- $F1(x,y,z)=\Sigma(2,5,7)$ $F2(x,y,z)=\Pi(0,1,4)$

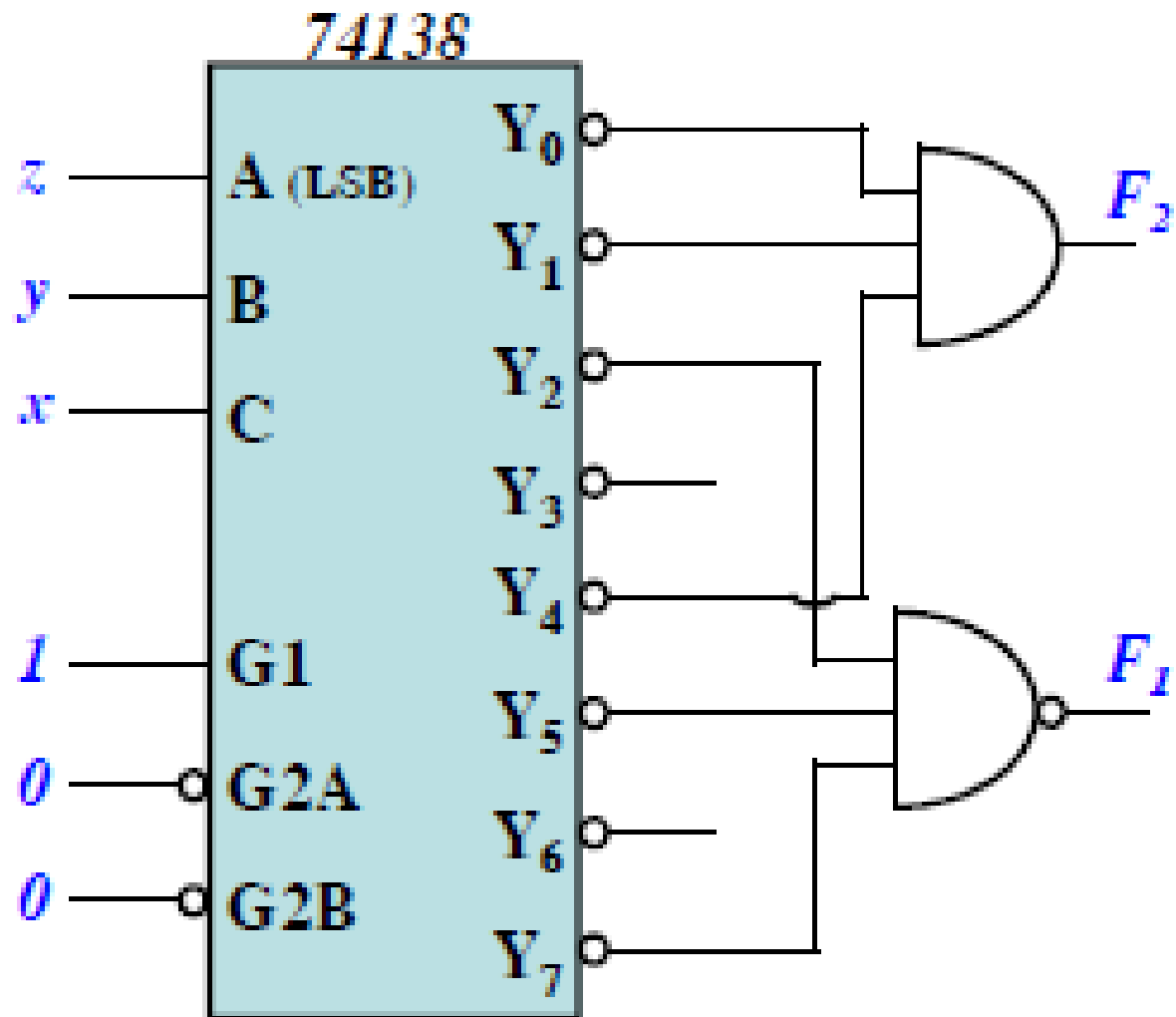
Phân tích:

Mỗi ngõ ra của IC74138 tương ứng với 1 tổng chuẩn (tích cực thấp), do đó hàm cần phân tích theo các tổng chuẩn.

$$F1(x,y,z) = \Sigma(2,5,7) = m_2 + m_5 + m_7 = \overline{M}_2 + \overline{M}_5 + \overline{M}_7 = \overline{M}_2 \cdot \overline{M}_5 \cdot \overline{M}_7$$

$$F2(x,y,z) = \Pi(0,1,4) = M_0 \cdot M_1 \cdot M_4$$

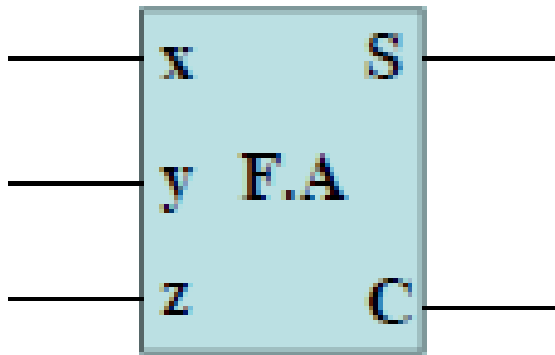
Mạch thiết kế



VD

Thiết kế FA dùng IC giải mã 74138 (74LS138) và 1 số cổng cần thiết.

Sơ đồ khối

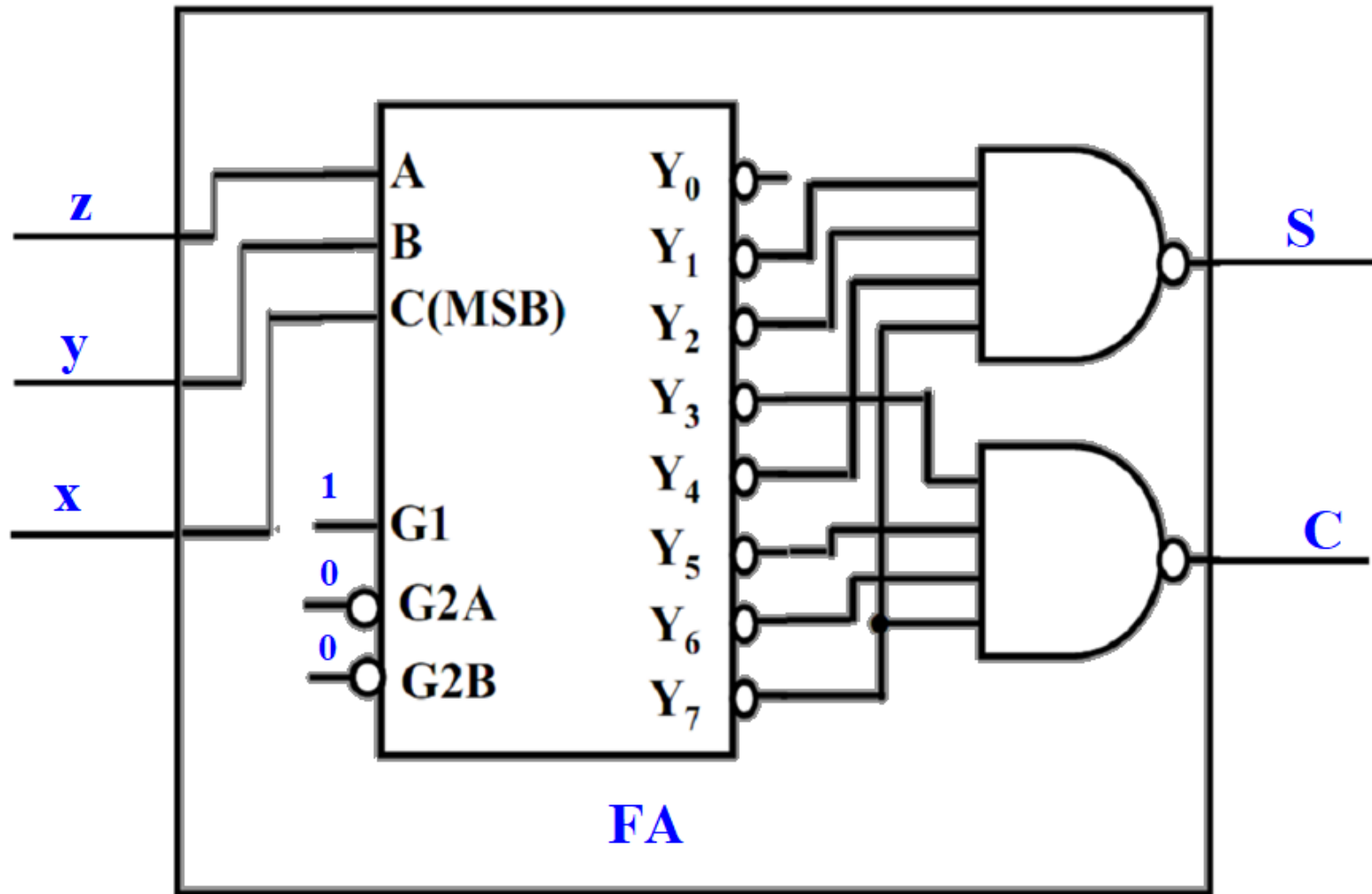


$$C(x,y,z) = \Sigma(3,5,6,7) = \Pi(0,1,2,4)$$

$$S(x,y,z) = \Sigma(1,2,4,7) = \Pi(0,3,5,6)$$

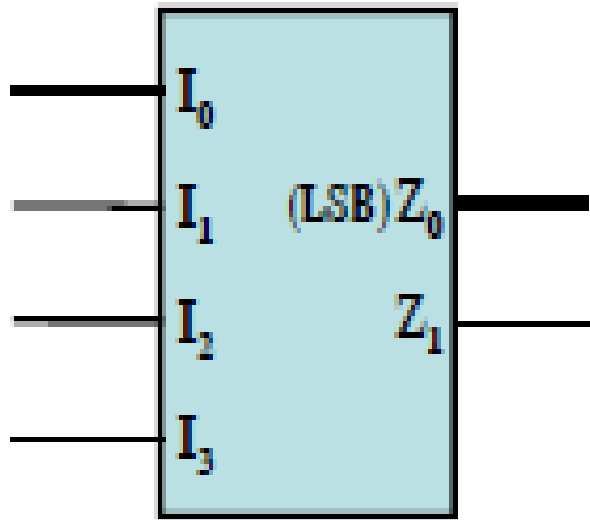
x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Mạch thiết kế



BỘ MÃ HÓA(ENCODER)

- Bộ mã hóa là hệ chuyển mã thực hiện hoạt động ngược lại với bộ giải mã. Nghĩa là bộ mã hóa có m ngõ vào theo mã nhị phân 1 trong m và n ngõ ra theo nhị phân cơ bản (với $m \leq 2^n$).
- Với ngõ vào I_i được tích cực thì ngõ ra chính là tổ hợp giá trị nhị phân i tương ứng.



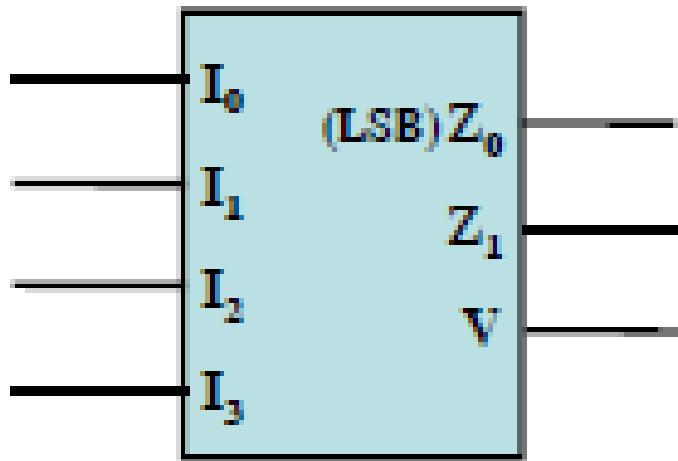
I_3	I_2	I_1	I_0	Z_1	Z_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$$Z_1 = I_3 + I_2$$

$$Z_0 = I_3 + I_1$$

Bộ mã hóa có ưu tiên

- Bộ mã hóa có ưu tiên là mạch mã hóa sao cho nếu có nhiều hơn 1 ngõ vào cùng tích cực thì ngõ ra sẽ là giá trị nhị phân của ngõ vào có ưu tiên cao nhất.



I_3	I_2	I_1	I_0	Z_1	Z_0	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
0	X	X	X	1	1	1

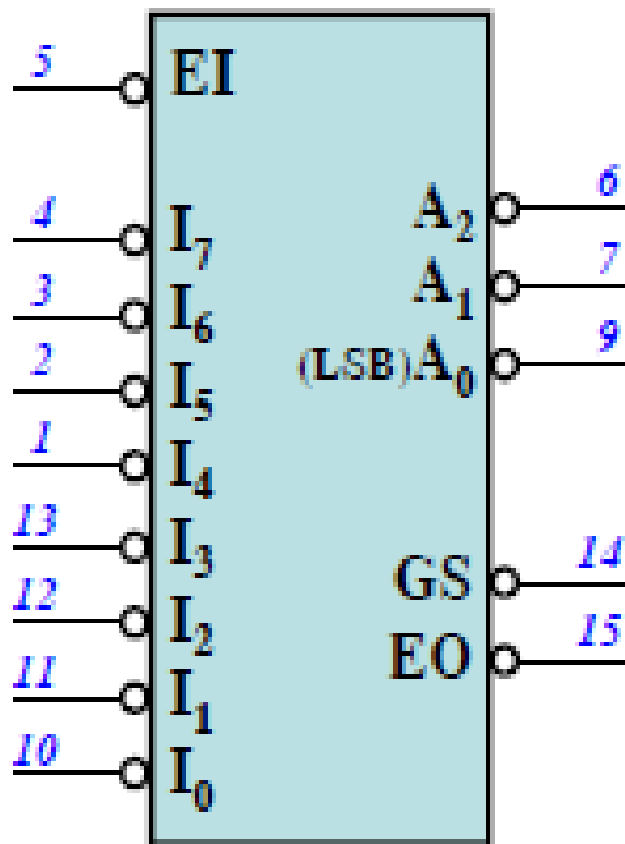
Thứ tự ưu tiên: $I_3 > I_2 > I_1 > I_0$

$$Z_1 = I_3 + I_2$$

$$Z_0 = I_3 + I_2' \cdot I_1$$

$$V = I_3 + I_2 + I_1 + I_0$$

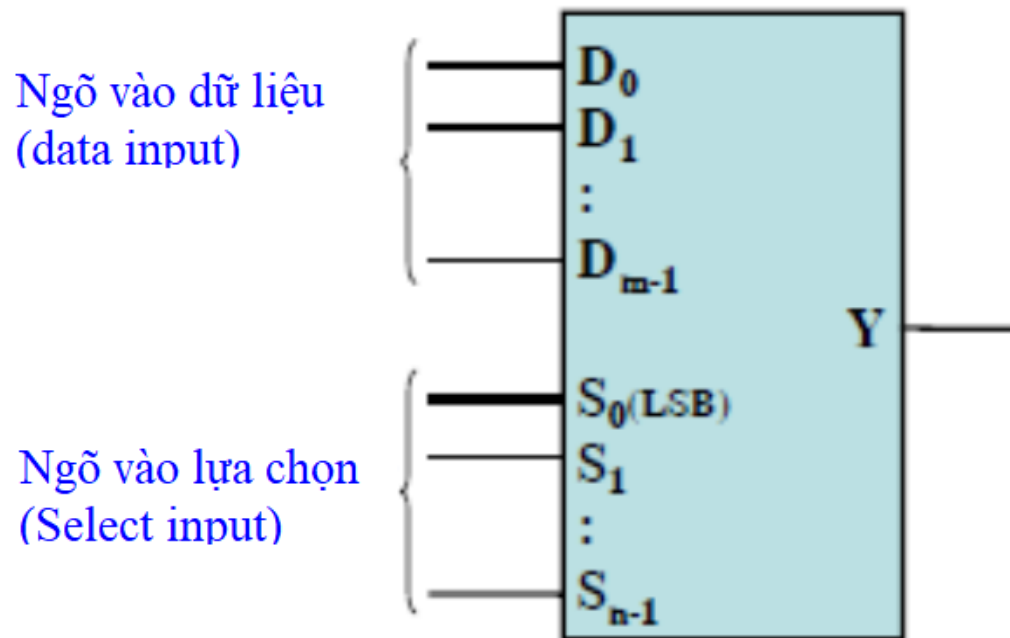
IC mã hóa ưu tiên 8 sang 3 – 74148(74LS148)



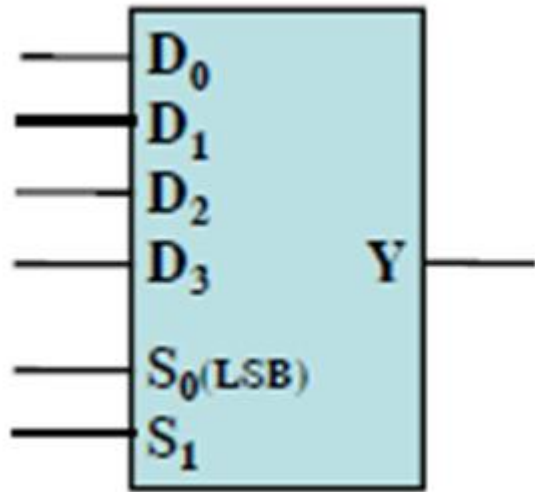
\overline{EI}	$\overline{I_7}$	$\overline{I_6}$	$\overline{I_5}$	$\overline{I_4}$	$\overline{I_3}$	$\overline{I_2}$	$\overline{I_1}$	$\overline{I_0}$	$\overline{A_2}$	$\overline{A_1}$	$\overline{A_0}$	\overline{GS}	\overline{EO}
1	X	X	X	X	X	X	X	X	1	1	1	1	1
0	0	X	X	X	X	X	X	X	0	0	0	0	1
0	1	0	X	X	X	X	X	X	0	0	1	0	1
0	1	1	0	X	X	X	X	X	0	1	0	0	1
0	1	1	1	0	X	X	X	X	0	1	1	0	1
0	1	1	1	1	0	X	X	X	1	0	0	0	1
0	1	1	1	1	1	0	X	X	1	0	1	0	1
0	1	1	1	1	1	1	0	X	1	1	0	0	1
0	1	1	1	1	1	1	1	0	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0

BỘ DÒNG KÊNH (MULTIPLEXER-MUX)

- MUX $2^n \rightarrow 1$ là hệ tổ hợp có nhiều ngõ vào nhưng chỉ có 1 ngõ ra.
- Ngõ vào gồm 2 nhóm: m ngõ vào dữ liệu (data input) và n ngõ vào lựa chọn (select input).
- Với 1 giá trị I của tổ hợp nhị phân của các ngõ vào lựa chọn, ngõ vào dữ liệu D_i sẽ được chọn đưa đến ngõ ra ($m = 2^n$)

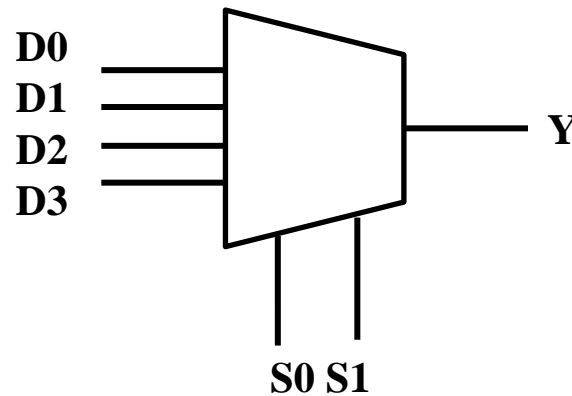


BỘ MUX 4 → 1



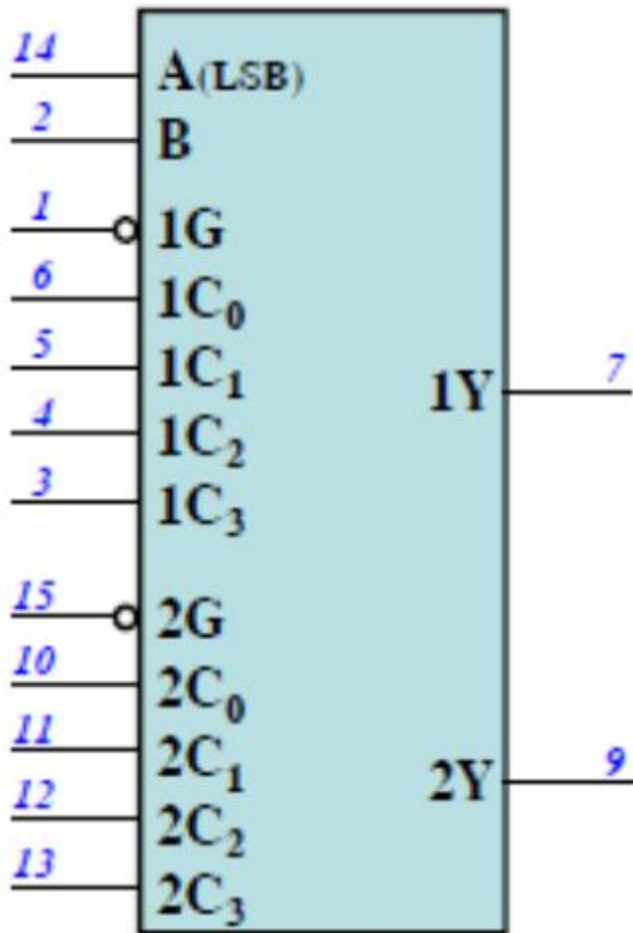
$$\begin{aligned} Y &= \overline{S_1} \overline{S_0} D_0 + \overline{S_1} S_0 D_1 + S_1 \overline{S_0} D_2 + S_1 S_0 D_3 \\ &= m_0 D_0 + m_1 D_1 + m_2 D_2 + m_3 D_3 \\ &= \sum m_i D_i \quad (i = 0, 1, 2, 3) \end{aligned}$$

Tổng quát: $Y = \sum m_i D_i$ ($i = 0, 1, \dots, 2^n - 1$)

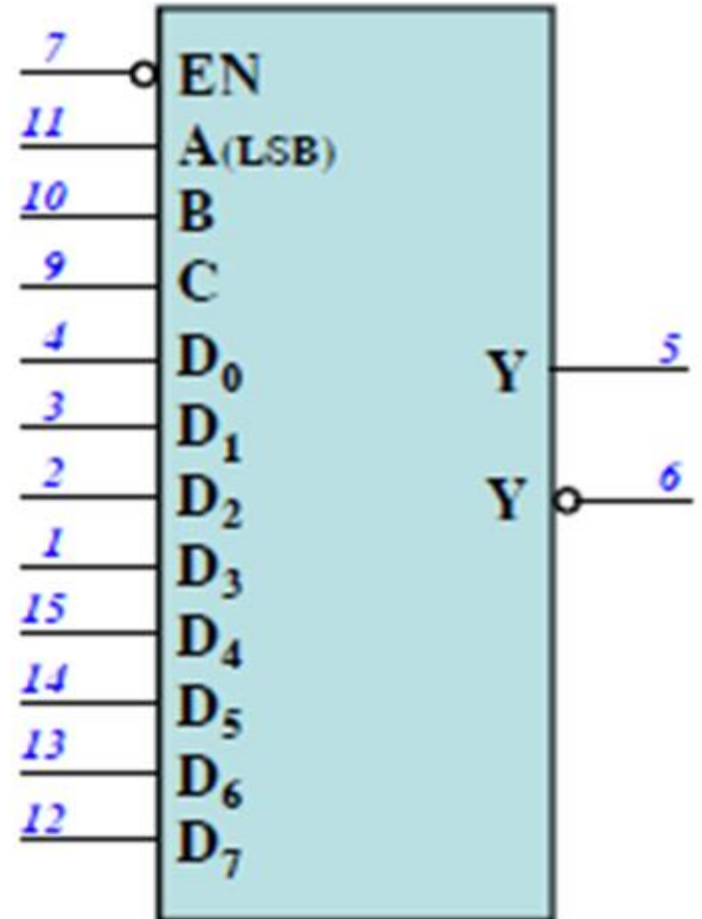


Ký hiệu khác của MUX

Giới thiệu IC dồn kênh



74153



74151

Sử dụng MUX để thực hiện hàm

Nguyên tắc:

- Hàm cần thực hiện phải đưa về dạng bảng sự thật hoặc chính tắc để dễ dàng đồng nhất tìm ra các Di, ngõ ra của MUX chính là hàm cần thiết kế.
- Các ngõ cho phép của MUX phải ở trạng thái tích cực.
- Các biến của hàm được nối đến các ngõ vào lựa chọn của MUX theo đúng vị trí trọng số.

a. Sử dụng bộ MUX $2^n \rightarrow 1$ thực hiện hàm Boole n biến.

$$F(x,y,z) = \Sigma(0,1,4,7)$$

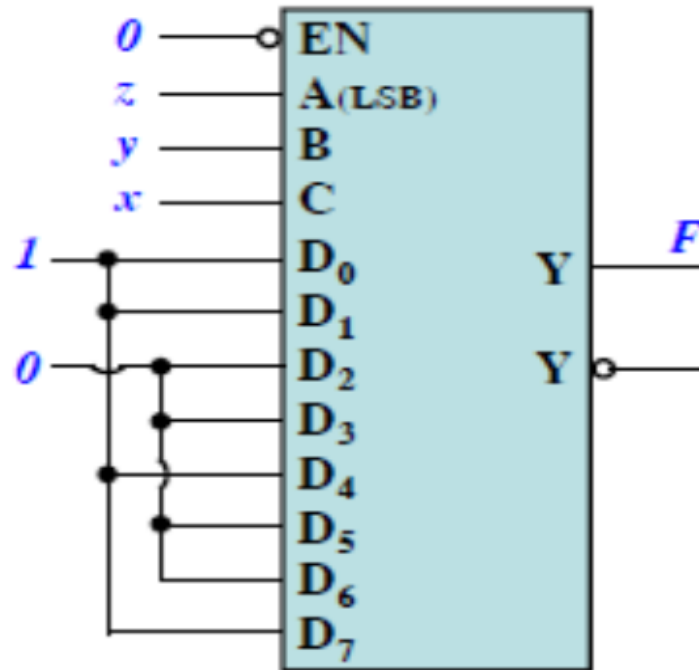
$$= m_0 + m_1 + m_4 + m_7$$

$$= m_0.1 + m_1.1 + m_2.0 + m_3.0 + m_4.1 + m_5.0 + m_6.0 + m_7.1$$

$$Y = \Sigma m_i D_i$$

$$= m_0.D_0 + m_1.D_1 + m_2.D_2 + m_3.D_3 + m_4.D_4 + m_5.D_5 + m_6.D_6 + m_7.D_7$$

$$\Rightarrow D_0=D_1=D_4=D_7=1; D_2=D_3=D_5=D_6=0$$



b. Sử dụng bộ MUX $2^n \rightarrow 1$ thực hiện hàm Boole $(n + 1)$ biến.

$$F(x,y,z) = \Sigma(0,1,4,7)$$

$$= x'y'z' + x'y'z + xy'z' + xyz$$

$$= x'y'.1 + x'y.0 + xy'.z' + xy.z$$

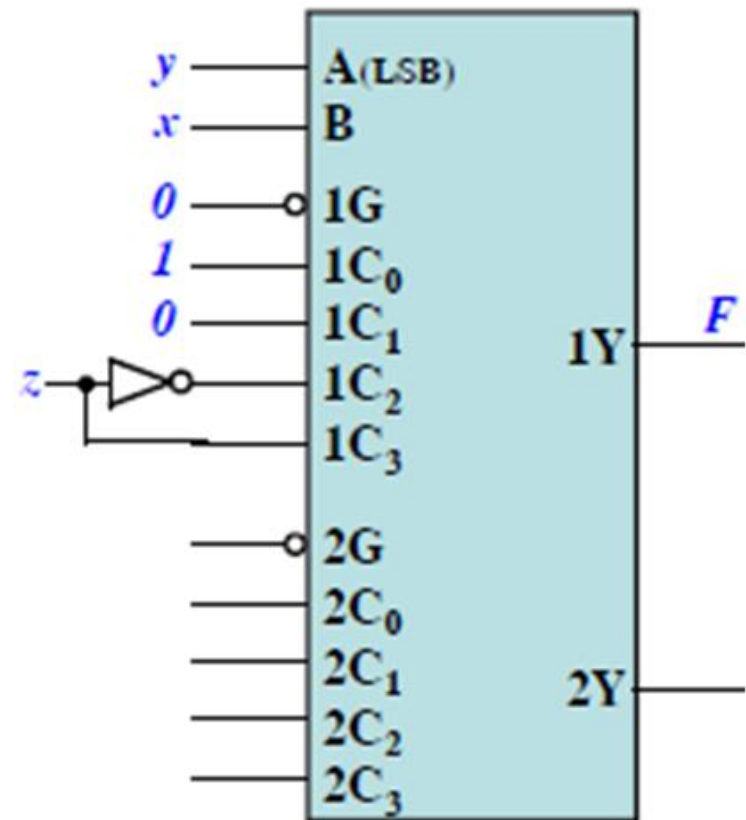
$$= m_0.1 + m_1.0 + m_2.1 + m_3.z$$

$$Y = \Sigma m_i D_i$$

$$= m_0.D_0 + m_1.D_1 + m_2.D_2 + m_3.D_3$$

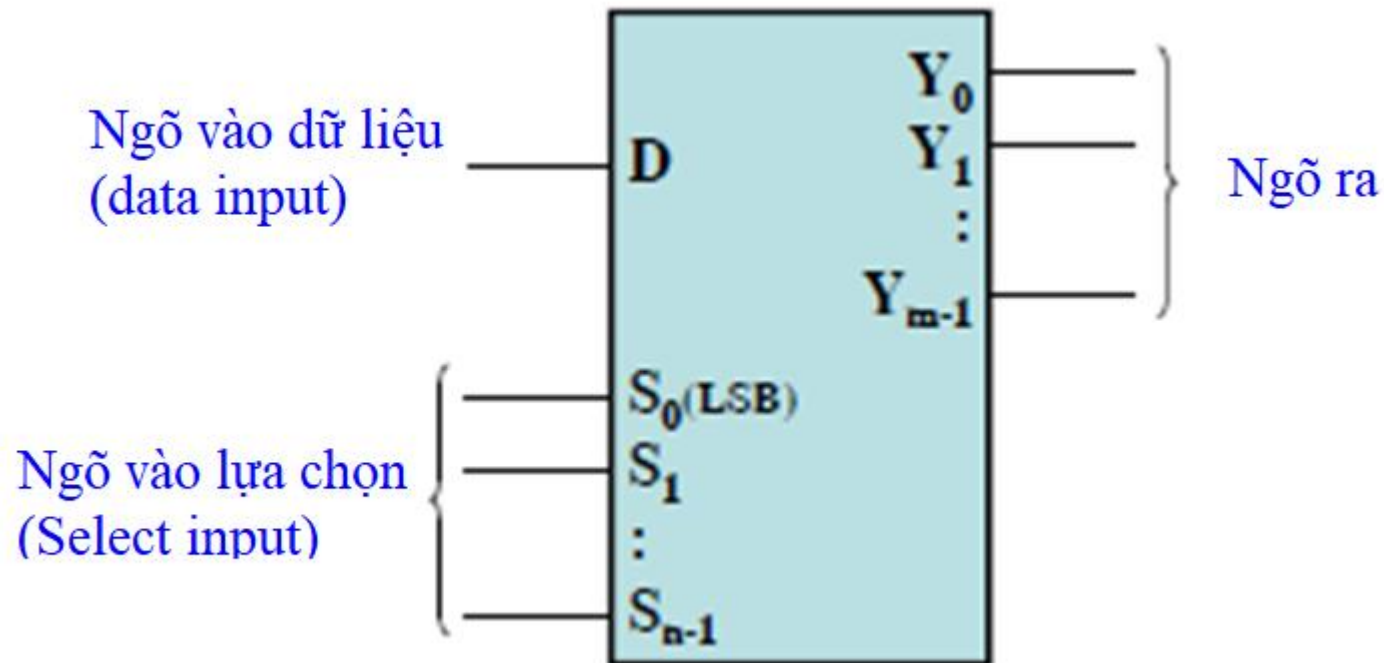
$$\Rightarrow D_0=1, D_1=0, D_2=z', D_3=z$$

x	y	z	F	D _i
0	0	0	1	D ₀ = 1
0	0	1	1	
0	1	0	0	D ₁ = 0
0	1	1	0	
1	0	0	1	D ₂ = z'
1	0	1	0	
1	1	0	0	D ₃ = z
1	1	1	1	



BỘ PHÂN KÊNH (DEMULTIPLEXER-DEMUX)

DEMUX $1 \rightarrow 2^n$ có chức năng thực hiện hoạt động ngược lại với bộ MUX. Mạch có 1 ngõ vào dữ liệu, n ngõ vào lựa chọn và 2^n ngõ ra.



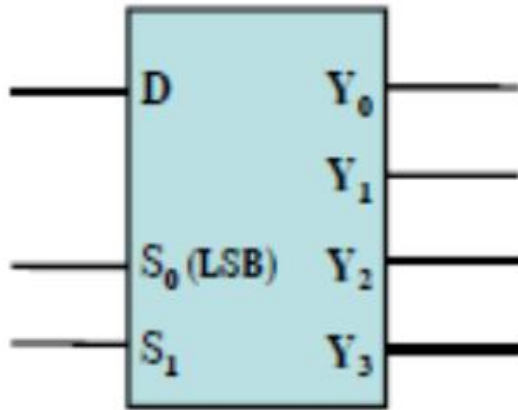
BỘ PHÂN KÊNH (DEMULTIPLEXER-DEMUX)

Hoạt động

- Với 1 giá trị **i** của tổ hợp nhị phân của các ngõ vào lựa chọn, ngõ vào dữ liệu **D** sẽ được đưa đến ngõ ra **Y_i**.

VD: TK bộ DEMUX 1 sang 4.

Sơ đồ khối



Bảng chân trị

S ₁	S ₀	Y ₃	Y ₂	Y ₁	Y ₀
0	0	0	0	0	D
0	1	0	0	D	0
1	0	0	D	0	0
1	1	D	0	0	0

Hàm ngõ ra

$$Y_0 = S_1' S_0' D; Y_1 = S_1' S_0 D; Y_2 = S_1 S_0' D; Y_3 = S_1 S_0 D$$

Tổng quát: $Y_i = m_i D$ ($i = 0, 1, \dots, 2^n - 1$)