

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

* Các bước thiết kế:

- Phát biểu bài toán.
- Xác định số biến ngõ vào và số biến ngõ ra.
- Thành lập bảng giá trị chỉ rõ mối quan hệ giữa ngõ vào và ngõ ra.

<i>Ngõ vào</i>	<i>Ngõ ra</i>
$X_{n-1} \dots X_1 X_0$	$Y_{m-1} \dots Y_1 Y_0$
0 ... 0 0	
1 ... 1 1	

- Tìm biểu thức rút gọn của từng ngõ ra phụ thuộc vào các biến ngõ vào.
- Thực hiện sơ đồ logic.

2

II. MỘT SỐ HỆ TỔ HỢP THÔNG DỤNG

1. BỘ CỘNG - TRỪ NHỊ PHÂN (SV TỰ ĐỌC)

- * CỘNG:
 - BÁN PHẦN (HALF ADDER- HA)
 - TOÀN PHẦN (FULL ADDER- FA)
- * TRỪ:
 - BÁN PHẦN (HALF SUBTRACTOR- HS)
 - TOÀN PHẦN (FULL SUBTRACTOR- FS)
- * CỘNG – TRỪ NHỊ PHÂN SONG SONG

2. BỘ CHUYỂN MÃ (CODE CONVERSION)

3. BỘ GIẢI MÃ (DECODER)

4. BỘ MÃ HÓA (ENCODER)

5. BỘ DÒNG KÊNH (MULTIPLEXER-MUX)

6. BỘ PHÂN KÊNH (DEMULTIPLEXER- DEMUX)

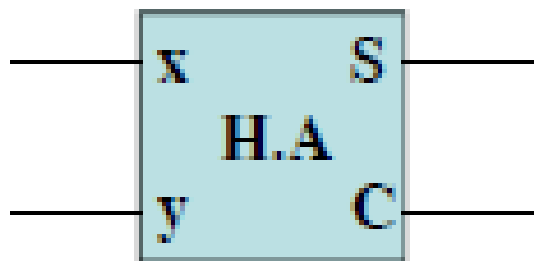
7. BỘ SO SÁNH ĐỘ LỚN (COMPARATOR)

8. PLD

BỘ CỘNG

CỘNG BÁN PHẦN (HALF ADDER - HA)

Mạch thực hiện phép cộng 2 số nhị phân 1 bit

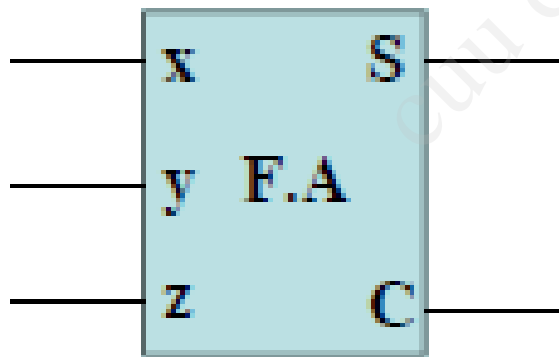


$$S = \bar{x}y + x\bar{y} = x \oplus y$$

$$C = xy$$

CỘNG TOÀN PHẦN (FULL ADDER - FA)

Mạch thực hiện phép cộng 3 số nhị phân 1 bit



$$S = x \oplus y \oplus z$$

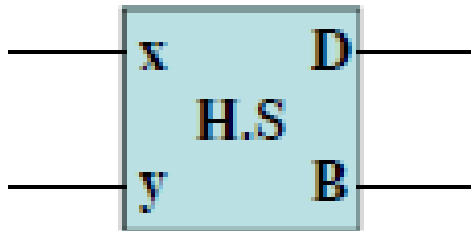
$$C = xy + xz + yz$$

$$= xy + z(x \oplus y)$$

MẠCH TRỪ

TRỪ BÁN PHẦN (HALF SUBTRACTOR – HS)

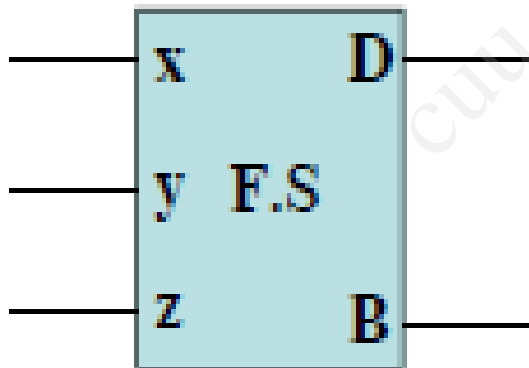
Mạch thực hiện phép trừ 2 số nhị phân 1 bit



$$D = \bar{x}y + x\bar{y} = x \oplus y$$
$$B = \bar{x}y$$

TRỪ TOÀN PHẦN (FULL SUBTRACTOR – FS)

Mạch thực hiện phép trừ 3 số nhị phân 1 bit



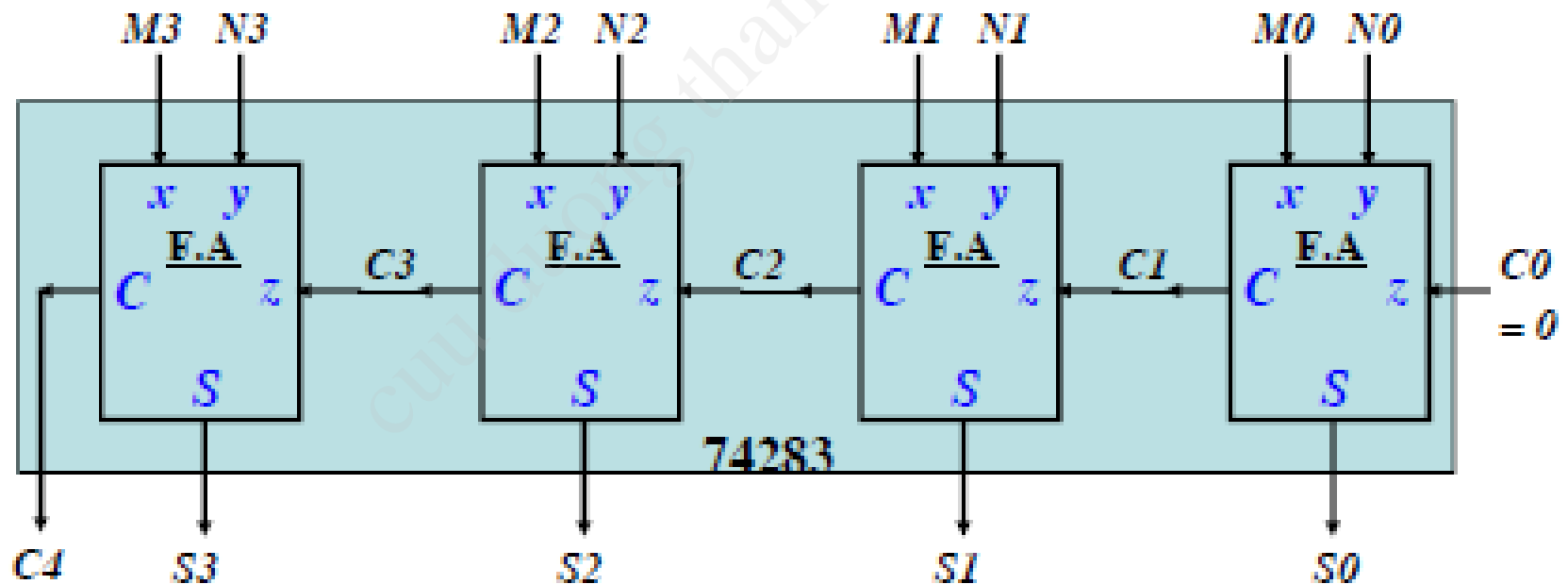
$$D = x \oplus y \oplus z$$
$$B = \bar{x}y + \bar{x}z + yz$$
$$= \bar{x}y + z(x \oplus y)$$

MẠCH CỘNG TRỪ SONG SONG

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

a. Bộ cộng nhị phân:

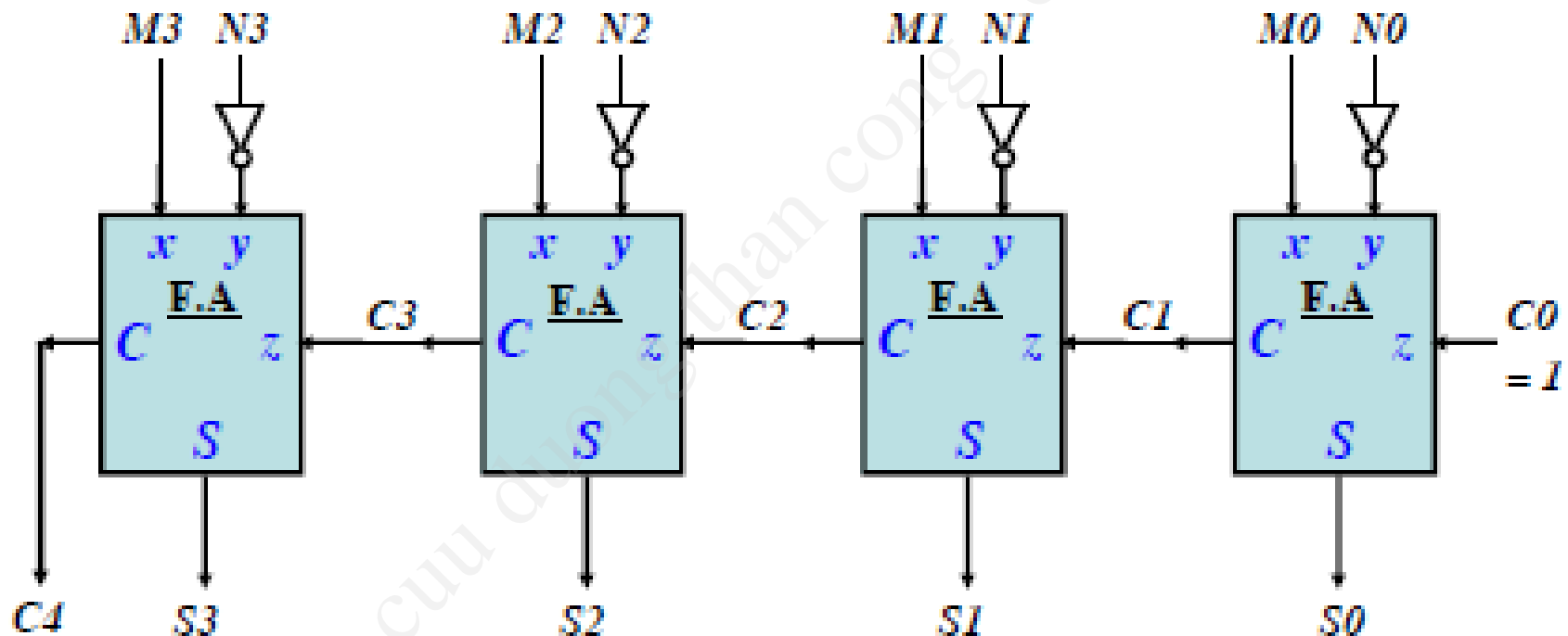
$$\begin{array}{rcccccc} & & C3 & C2 & C1 & & \\ M: & M3 & M2 & M1 & M0 & & \\ + & & & & & & \\ N: & N3 & N2 & N1 & N0 & & \\ \hline & C4 & S3 & S2 & S1 & S0 & \end{array}$$



b. Bộ trừ nhị phân:

- Sử dụng các bộ trừ toàn phần F.S
- 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:

- $C4 = 1$ kết quả là số dương
- $C4 = 0$ kết quả là số âm

c. Bộ cộng/trừ nhị phân:

Phép toán	C_0	y_i
CỘNG	0	N_i
TRỪ	1	$\overline{N_i}$

Ngõ vào điều khiển

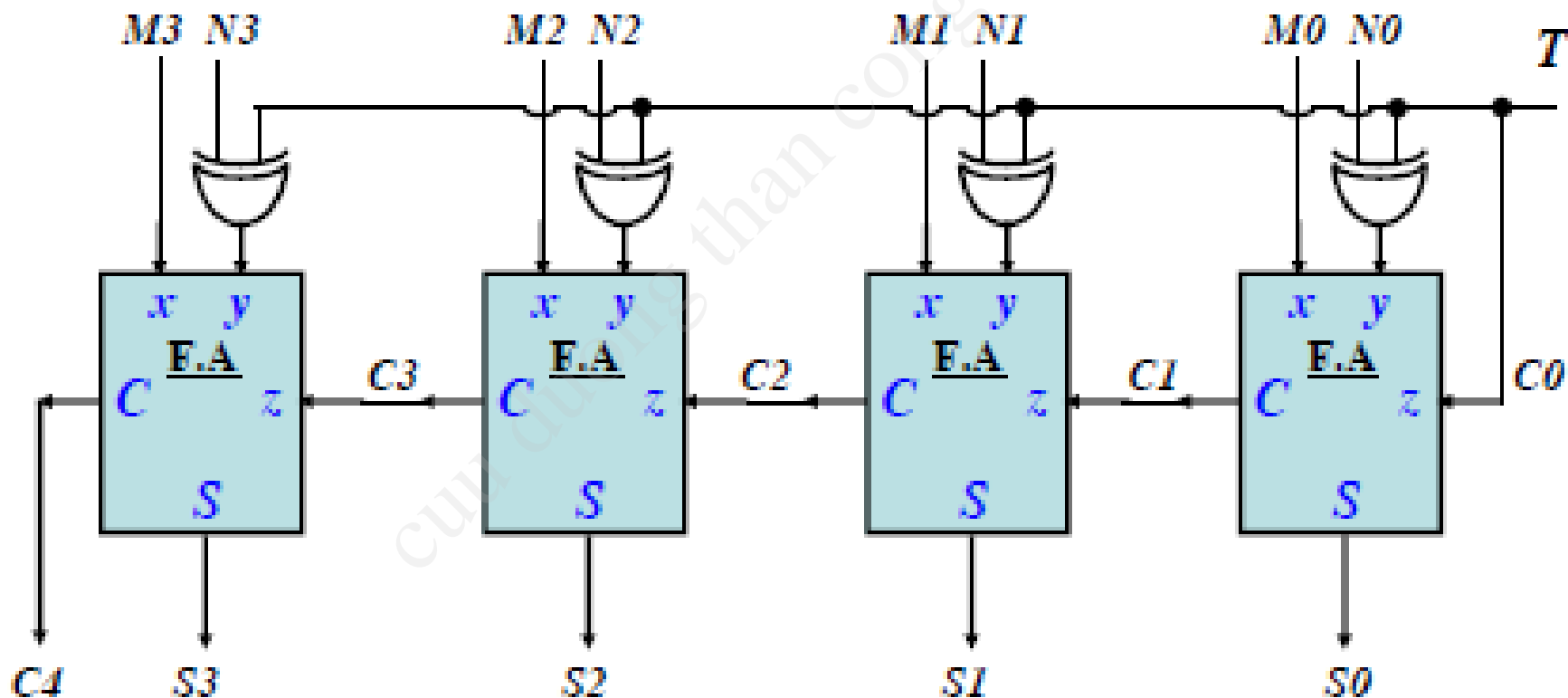
$T = 0$: Cộng

$T = 1$: Trừ

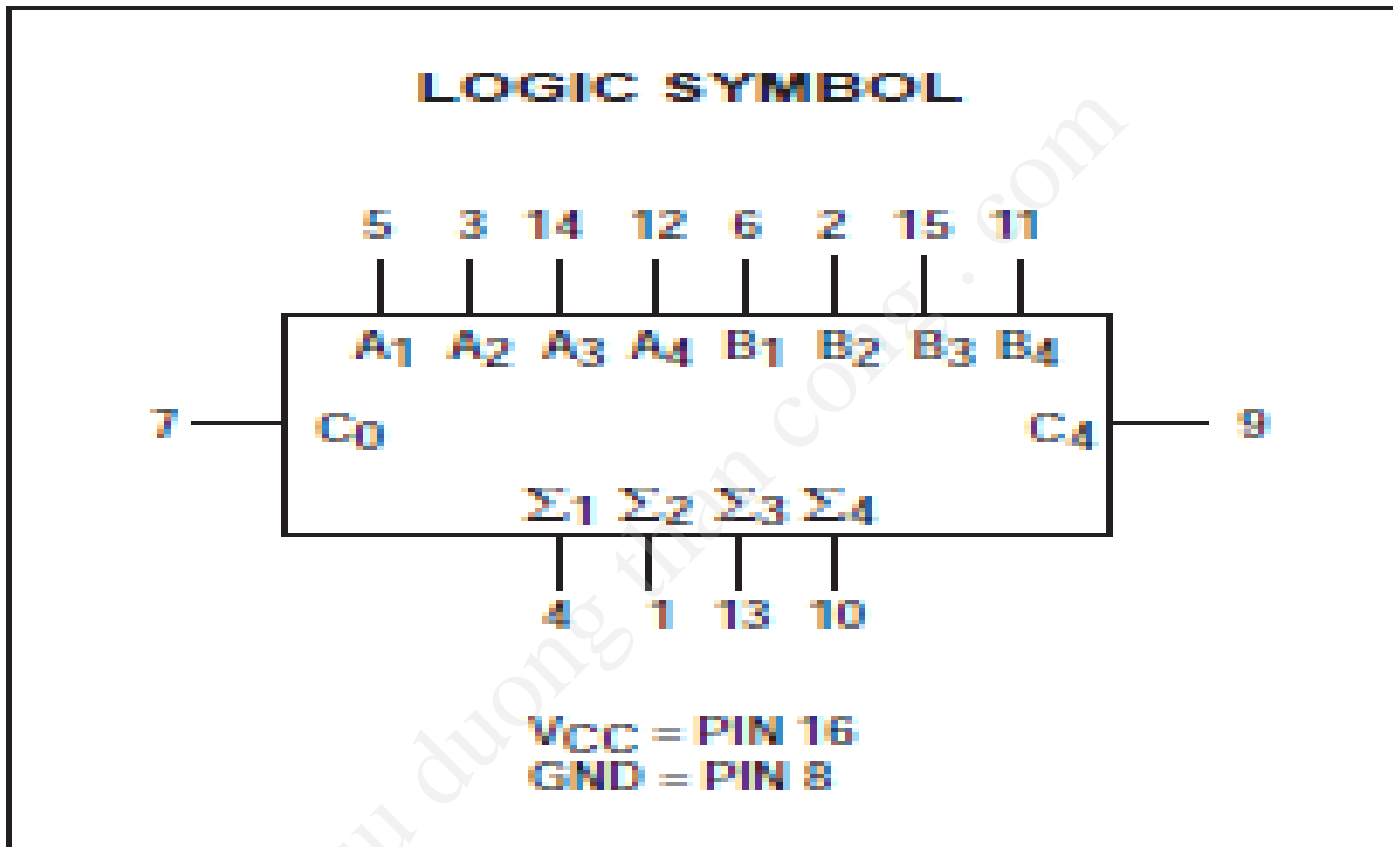


$$C_0 = T$$

$$y_i = T \oplus N_i$$

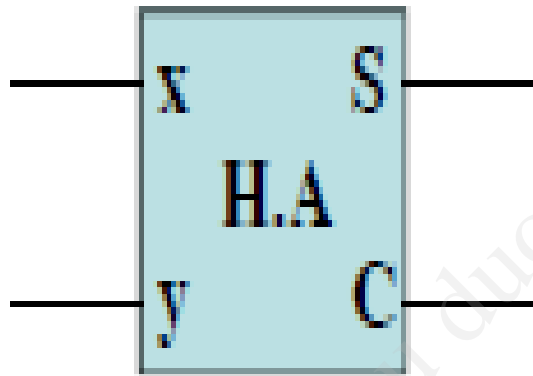


IC cộng 4 bit 74283(74LS283)



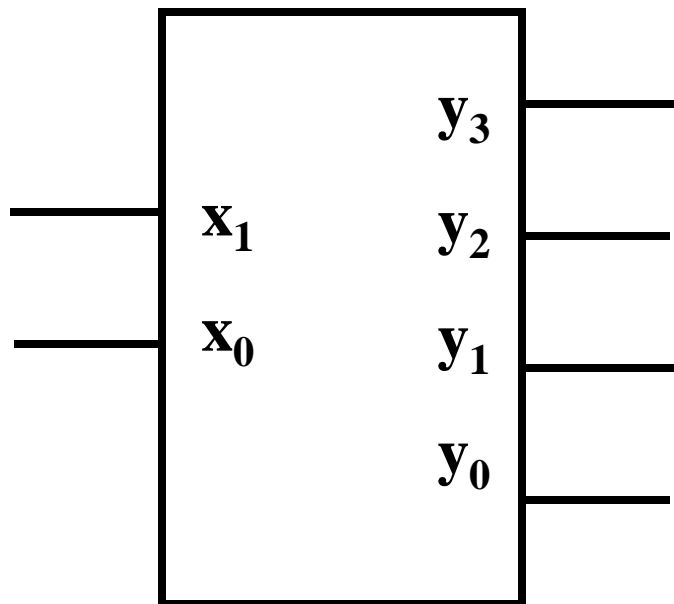
Sử dụng vi mạch cộng để thực hiện hàm

Ví dụ 1 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.



$$S = \bar{x}y + x\bar{y} = x \oplus y$$
$$C = xy$$

Sơ đồ khối



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

Các hàm ngõ ra

$$y_0 = 1$$

$$y_1 = x_1 \oplus x_0$$

$$y_2 = x_1$$

$$y_3 = x_1 x_0$$

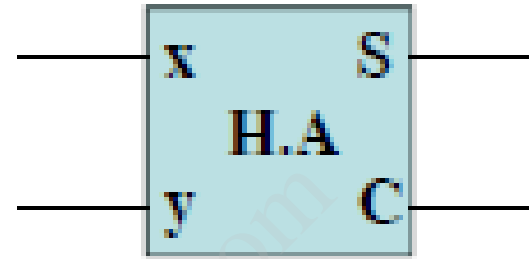
Các hàm ngõ ra

$$y_0 = 1$$

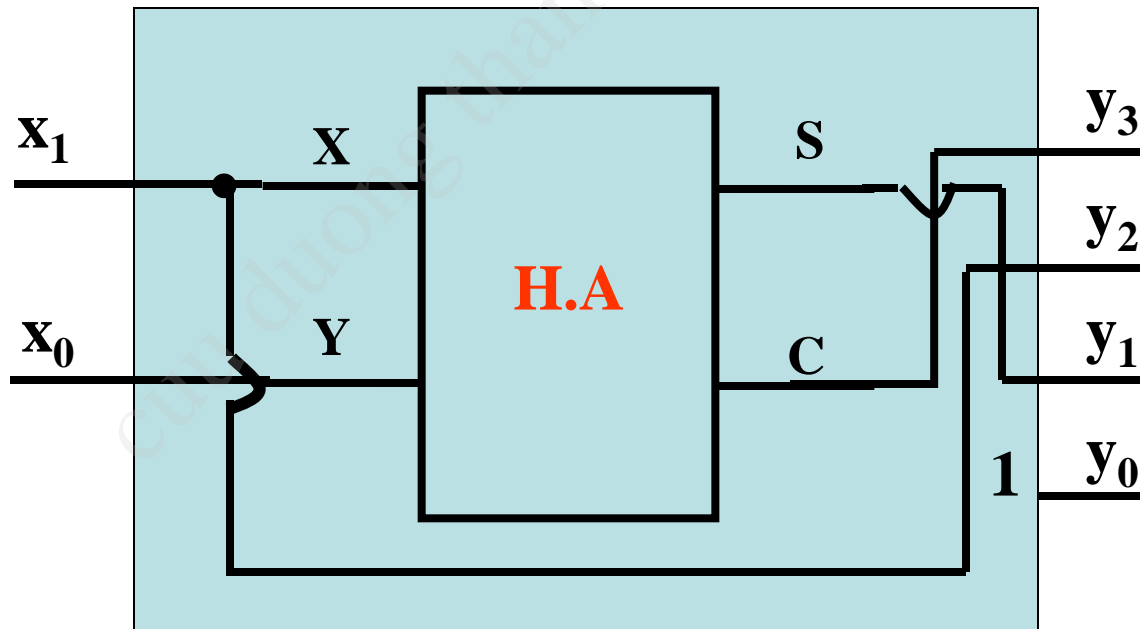
$$y_1 = x_1 \oplus x_0$$

$$y_2 = x_1$$

$$y_3 = x_1 x_0$$



$$S = \bar{x}y + x\bar{y} = x \oplus y$$
$$C = xy$$

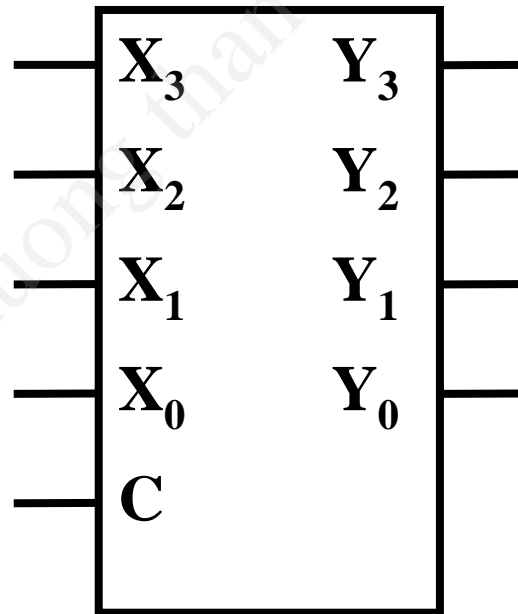


Ví dụ 2 Dùng vi mạch 74283 (mạch cộng 4 bit) và các cổng logic (nếu cần) để thiết kế mạch tổ hợp có hoạt động như sau:

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



IC74283:

$$S = A + B + C_0$$

Theo đề bài:

$$\begin{cases} C = 0: & Y = X & = X & + 0 + 0 \\ C = 1: & Y = \text{bù}_2(X) = \text{bù}_1(X) + 1 & = \text{bù}_1(X) + 0 + 1 \end{cases}$$

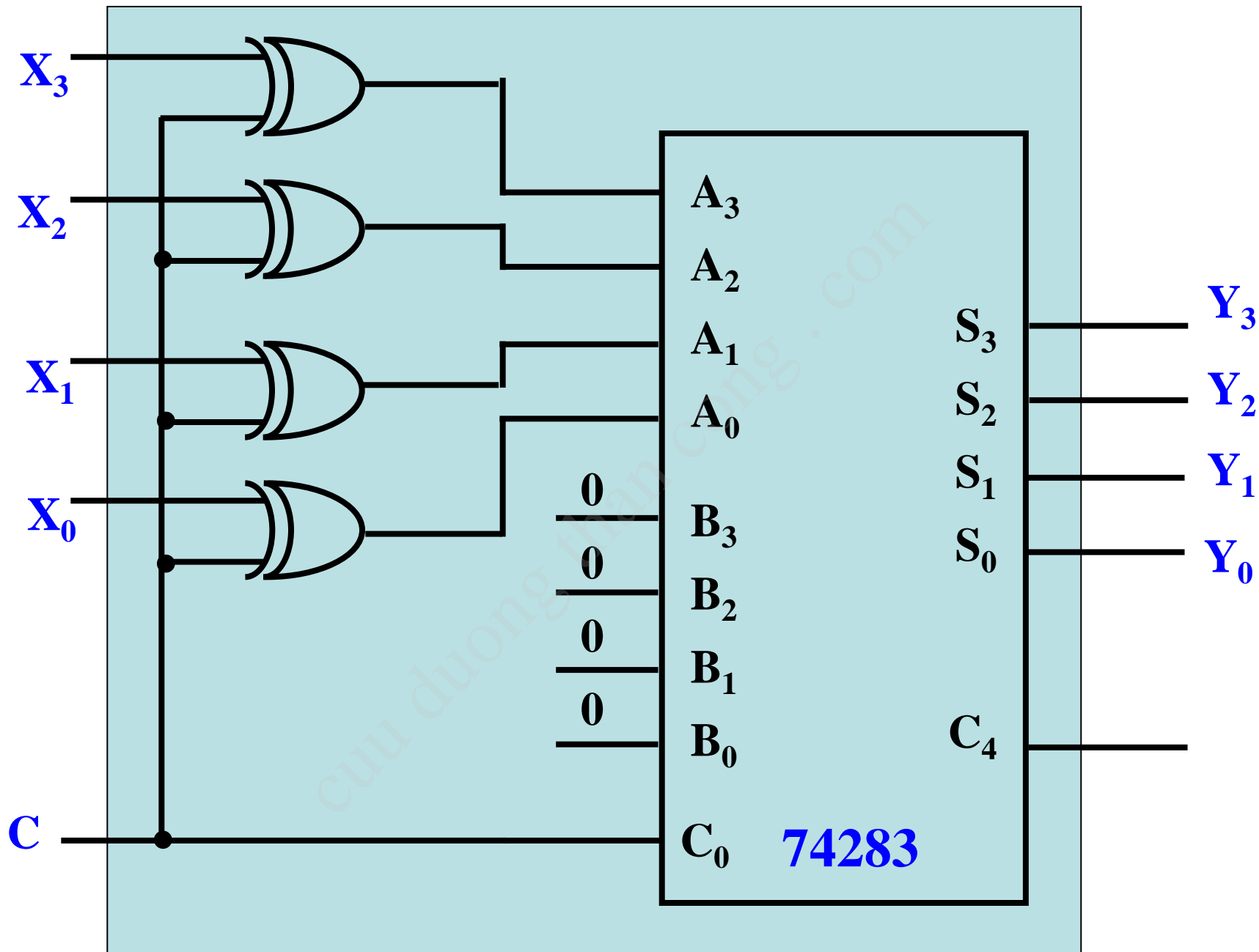
\Rightarrow Chọn $B = 0$; $C_0 = C$

\Rightarrow A phụ thuộc vào C: $\begin{cases} C = 0 \Rightarrow A = X \\ C = 1 \Rightarrow A = \text{bù}_1(X) \end{cases}$

$\Rightarrow A = X \oplus C$ (xem A là hàm của C và X)

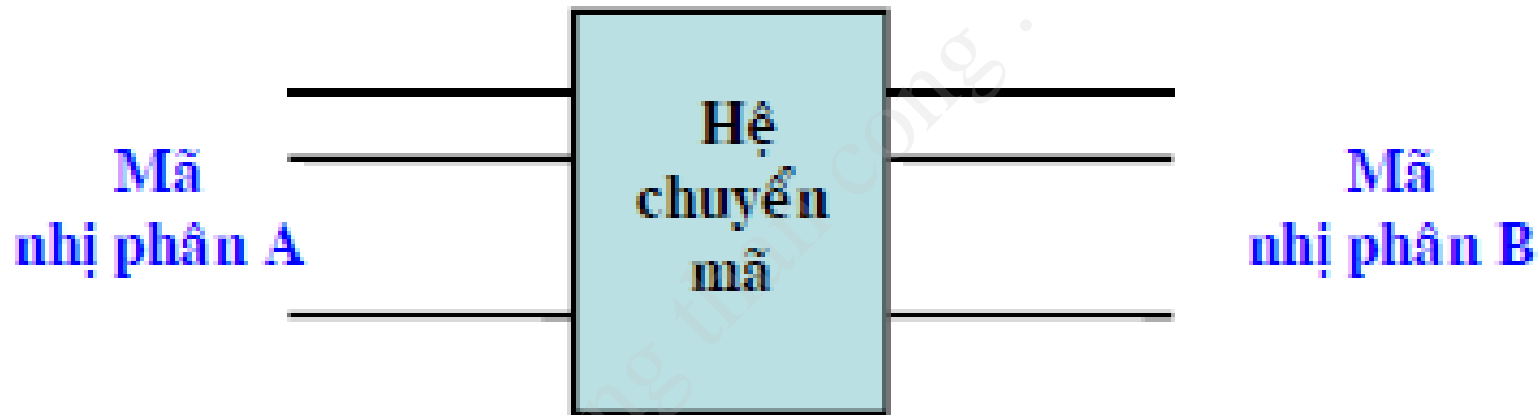
Kết luận:

$$\begin{cases} C_0 = C \\ B = 0 \\ A = X \oplus C \end{cases}$$



BỘ CHUYỂN MÃ

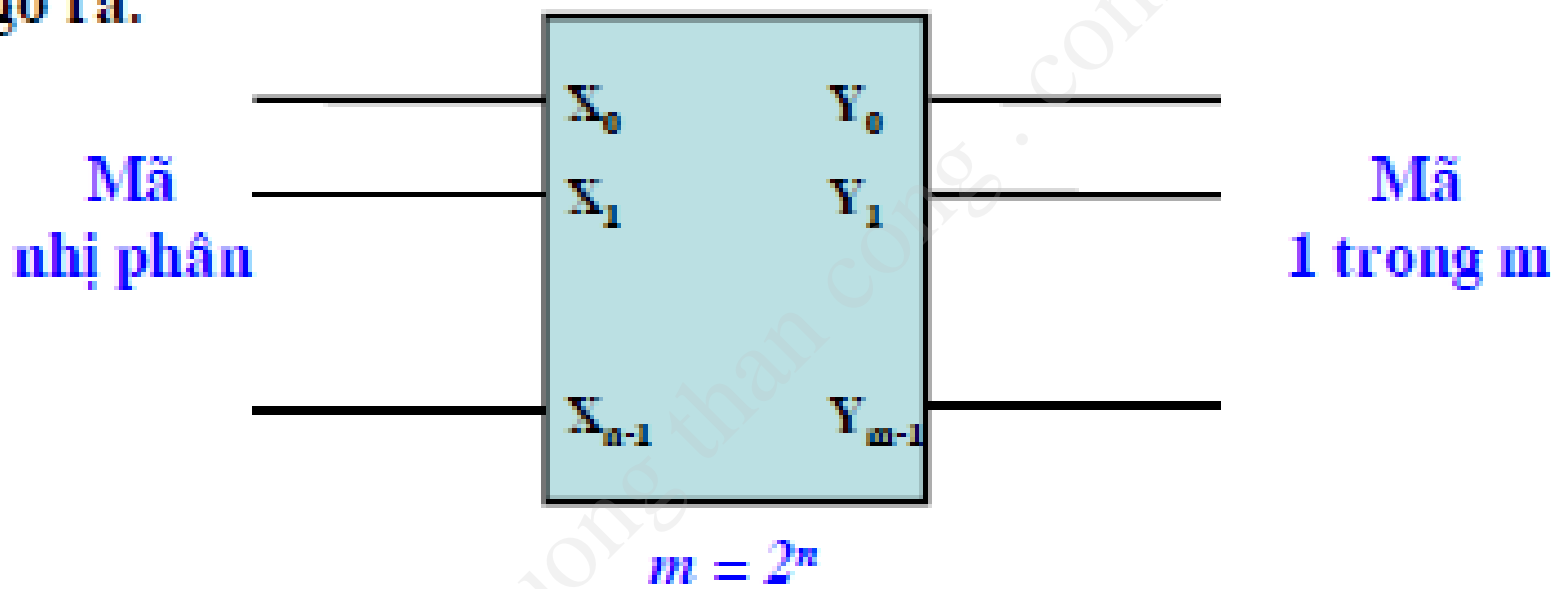
- Hệ chuyển mã là hệ tổ hợp có nhiệm vụ làm cho 2 hệ thống tương thích với nhau, mặc dù mỗi hệ thống dùng mã nhị phân khác nhau.



- Hệ chuyển mã có ngõ vào cung cấp các tổ hợp mã nhị phân A và các ngõ ra tạo ra các tổ hợp mã nhị phân B. Như vậy, ngõ vào và ngõ ra phải có số lượng từ mã bằng nhau.

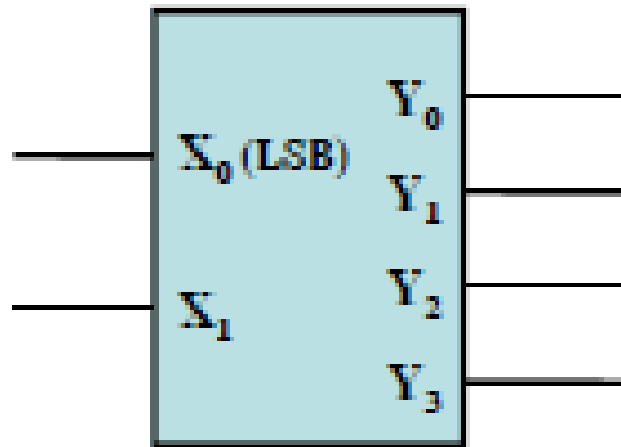
BỘ GIẢI MÃ (DECODER)

- 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



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

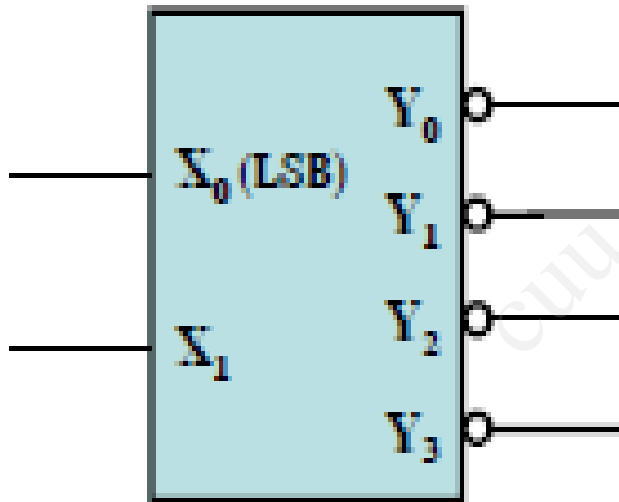
$$Y_1 = \overline{X_1} X_0 = m_1$$

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

$$Y_3 = X_1 X_0 = m_3$$

Ngõ ra: $Y_i = m_i$
($i = 0, 1, \dots, 2^n - 1$)

Bộ giải mã ngõ ra tích cực thấp



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

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

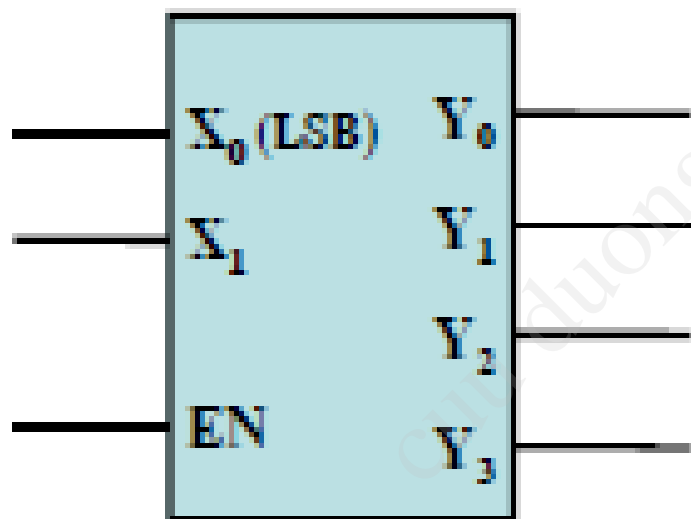
$$Y_2 = \overline{X_1} + X_0 = M_2 = \overline{m_2}$$

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

Ngõ ra: $Y_i = M_i$
($i = 0, 1, \dots, 2^n - 1$)

Bộ giải mã có ngõ vào cho phép

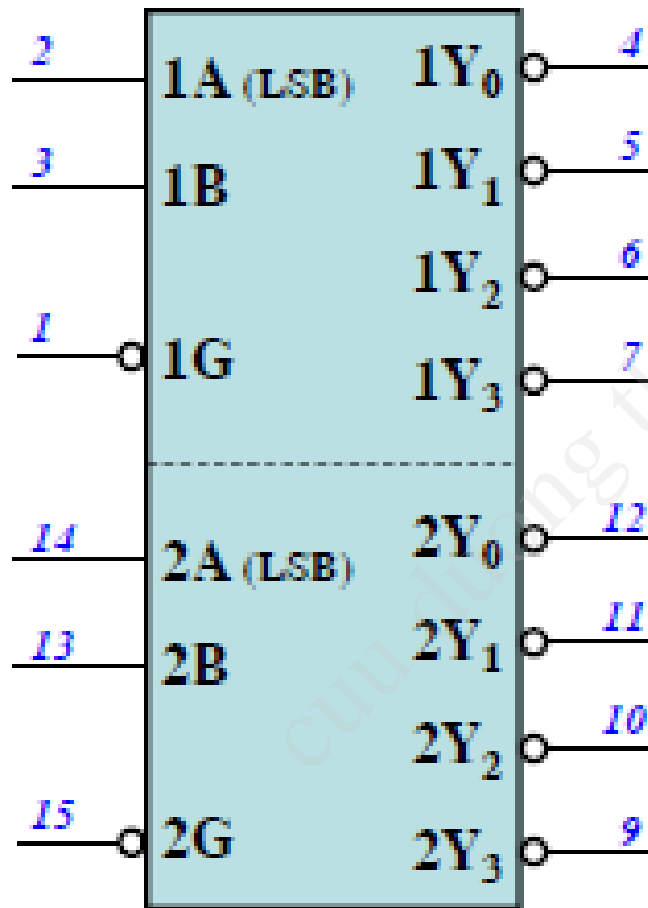
- Ngoài các ngõ vào dữ liệu, bộ giải mã có thể có 1 hay nhiều ngõ vào cho phép.
- Khi các ngõ vào cho phép ở trạng thái tích cực thì mạch giải mã mới được hoạt động. Ngược lại, mạch giải mã sẽ không hoạt động; khi đó các ngõ ra đều ở trạng thái không tích cực.



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

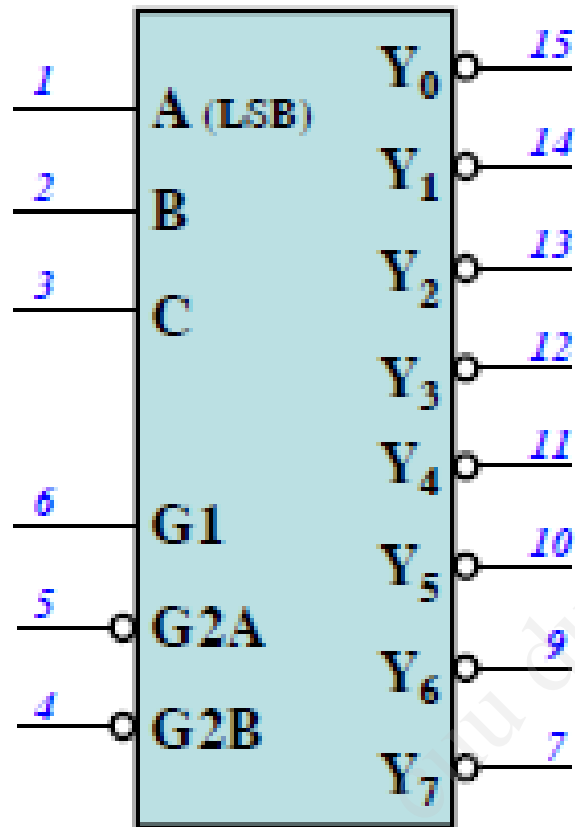
Giới thiệu một số IC giải mã 74139(74LS139)

a. IC 74139: gồm 2 bộ giải mã 2 sang 4 ngõ ra tích cực thấp



\overline{G}	B	A	\overline{Y}_3	\overline{Y}_2	\overline{Y}_1	\overline{Y}_0
1	X	X	1	1	1	1
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1

b. IC 74138: bộ giải mã 3 sang 8 ngõ ra tích cực thấp



$G1$	$\overline{G2A}$	$\overline{G2B}$	C	B	A	$\overline{Y_7}$	$\overline{Y_6}$	$\overline{Y_5}$	$\overline{Y_4}$	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
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

Ngõ ra của bộ giải mã là minterm (ngõ ra tích cực cao) hoặc maxterm (ngõ ra tích cực thấp) của n biến ngõ vào. Do đó, ta có thể sử dụng bộ giải mã thực hiện hàm Boole theo dạng chính tắc.

Nguyên tắc:

- *Hàm cần thực hiện phải đưa về dạng chính tắc và kết hợp thêm cổng thích 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.*

Ví dụ 3 Dùng IC giải mã 74138 (74LS138) và 1 số cổng cần thiết để thực hiện 2 hàm Boole sau:

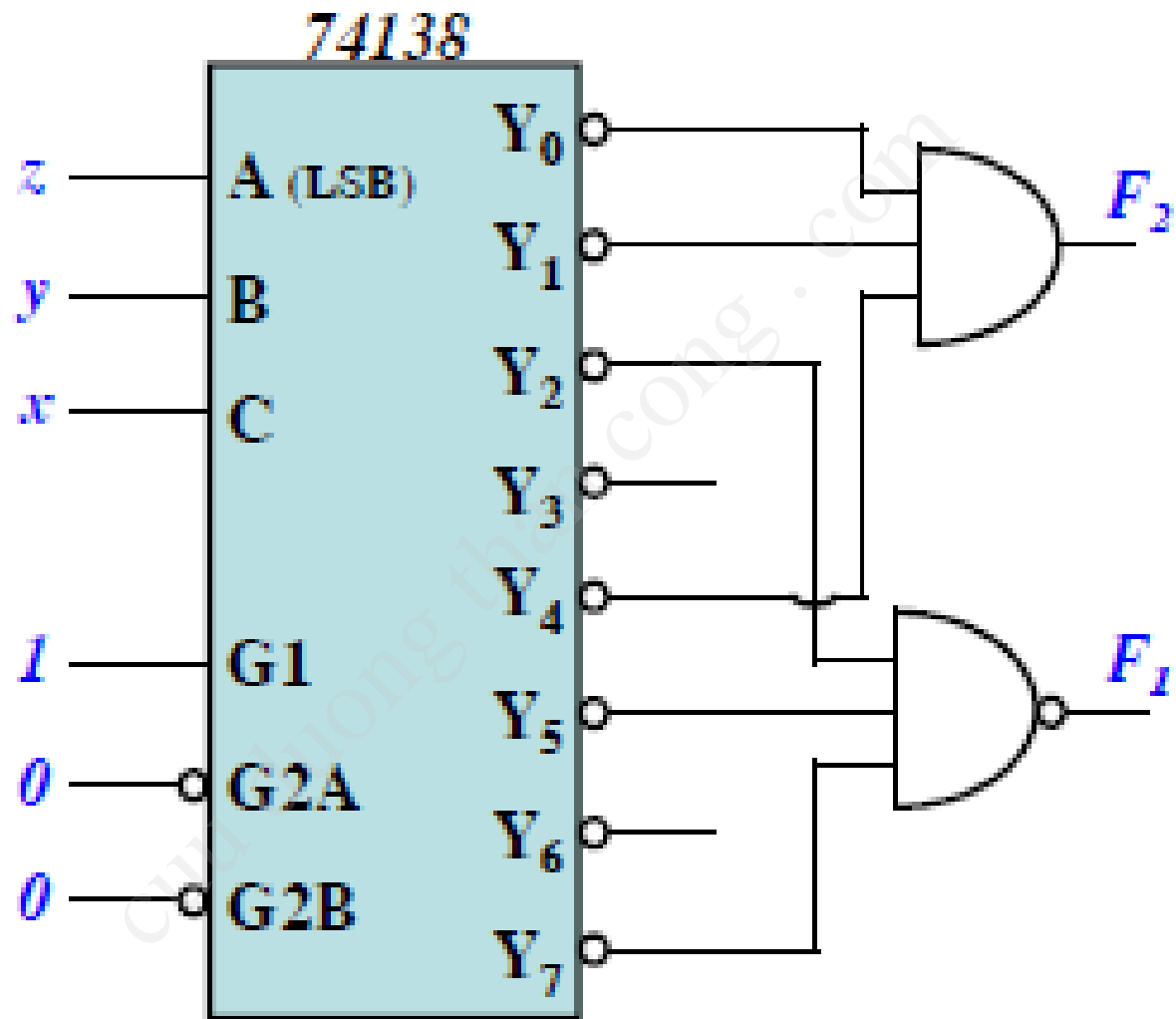
$$F1(x,y,z)=\Sigma(2,5,7) \quad F2(x,y,z)=\Pi(0,1,4)$$

74138: Ngõ ra: $Y_i = M_i$

$$(i = 0, 1, \dots, 2^n-1)$$

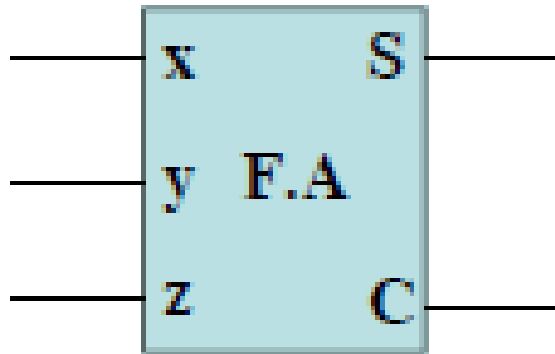
$$\begin{aligned} F1(x,y,z) &= m_2 + m_5 + m_7 = \overline{M}_2 + \overline{M}_5 + \overline{M}_7 = \overline{M_2 \cdot M_5 \cdot M_7} \\ &= M_0 \cdot M_1 \cdot M_3 \cdot M_4 \cdot M_6 \end{aligned}$$

$$F2(x,y,z) = M_0 \cdot M_1 \cdot M_4$$



Ví dụ 4 Thiết kế FA dùng IC giải mã 74138 (74LS138) và 1 số cổng cần thiết.

Sơ đồ khối



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

74138

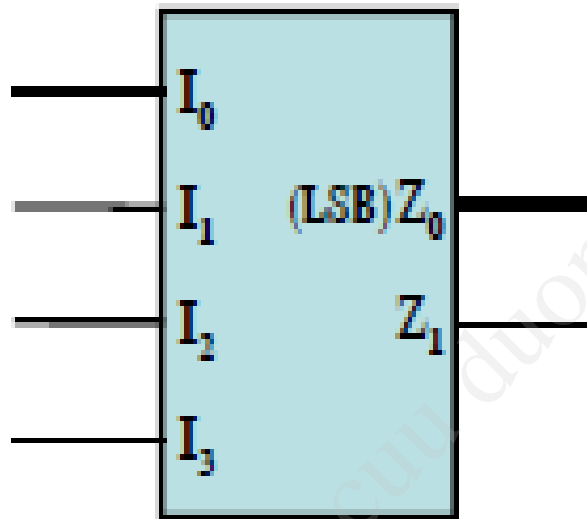
Ngõ ra: $Y_i = M_i$
 $(i = 0, 1, \dots, 2^n - 1)$

$$C = \prod (0, 1, 2, 4) \\ = M_0.M_1.M_2.M_4$$

$$S = \prod (0, 3, 5, 6) \\ = M_0.M_3.M_5.M_6$$

BỘ MÃ HÓA (ENCODER)

- Encoder là hệ chuyển mã thực hiện hoạt động ngược lại với decoder. Nghĩa là encoder có m ngõ vào theo mã nhị phân 1 trong m và n ngõ ra theo mã 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.



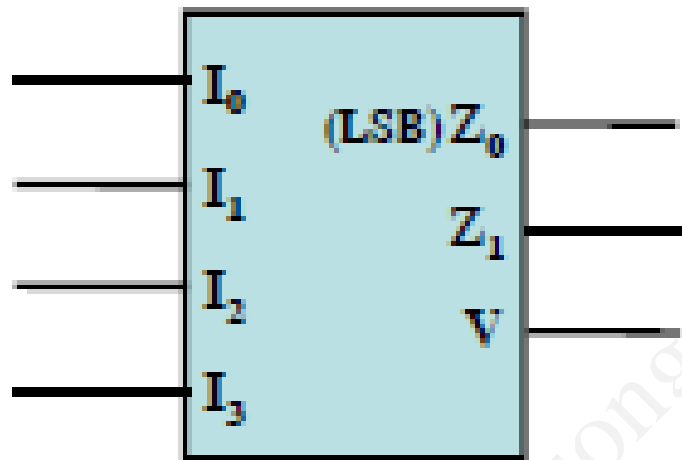
I_3	I_2	I_1	I_0	Z_1	Z_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$$Z_1 = I_3 + I_2$$

$$Z_0 = I_3 + I_1$$

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
1	X	X	X	1	1	1

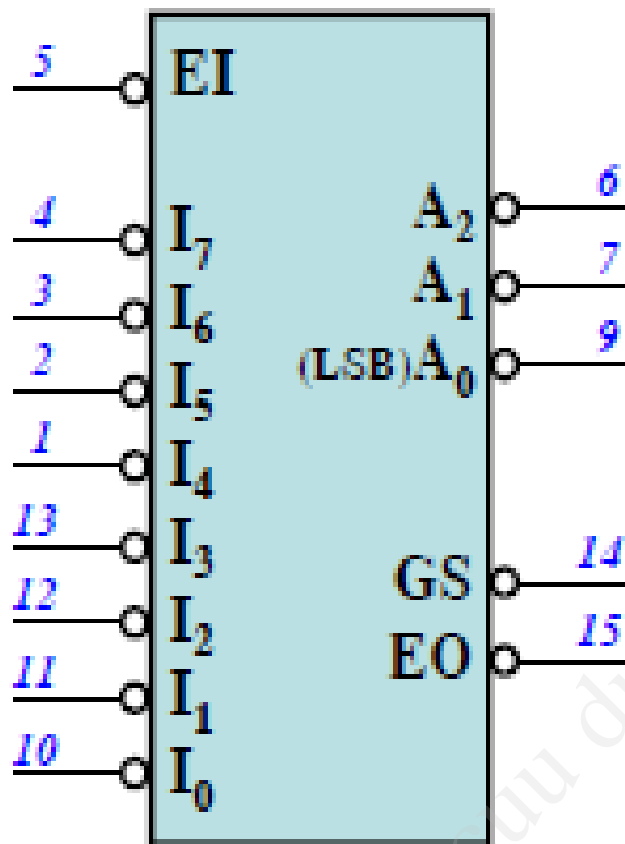
Thứ tự ưu tiên: $I_3 \succ I_2 \succ I_1 \succ I_0$

$$Z_1 = I_3 + I_2$$

$$Z_0 = I_3 + I_2 I_1$$

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

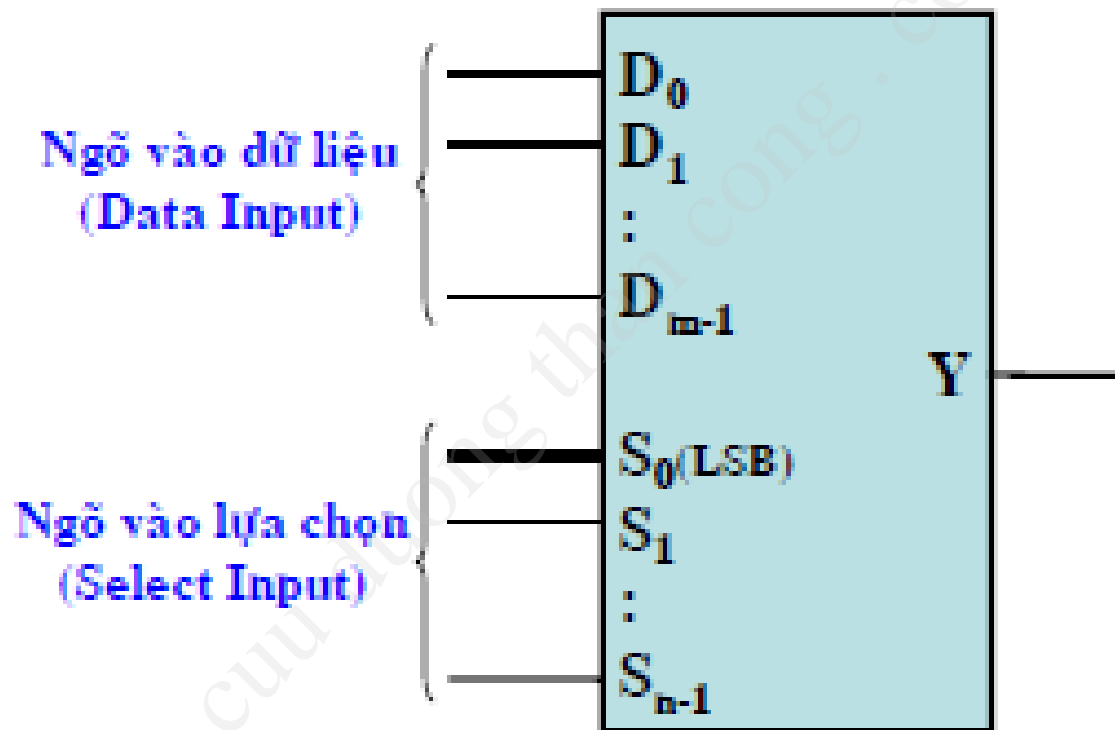
IC mã hóa ưu tiên 8 sang 3– 74148(74LS148)



\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

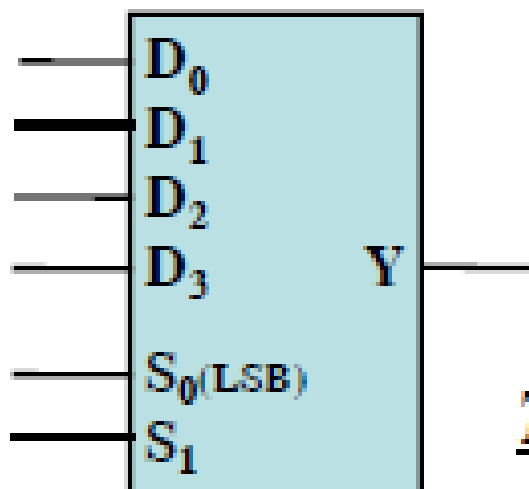
BỘ DÒNG KÊNH (MULTIPLEXER-MUX)

- 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 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:

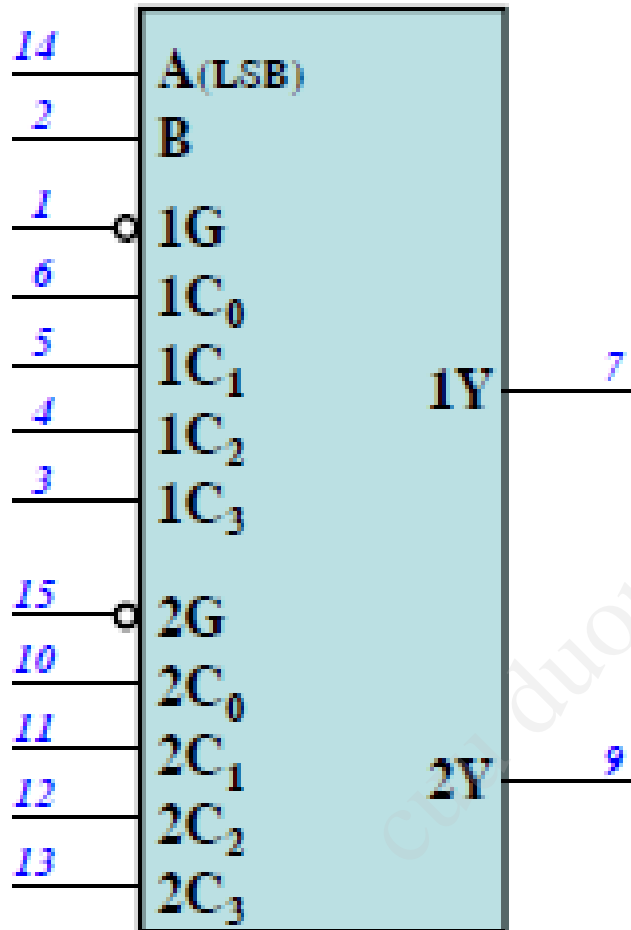


$$\begin{aligned} Y &= \overline{S_1} \overline{S_0} D_0 + \overline{S_1} S_0 D_1 + S_1 \overline{S_0} D_2 + S_1 S_0 D_3 \\ &= m_0 D_0 + m_1 D_1 + m_2 D_2 + m_3 D_3 \\ &= \sum m_i D_i \quad (i = 0, 1, 2, 3) \end{aligned}$$

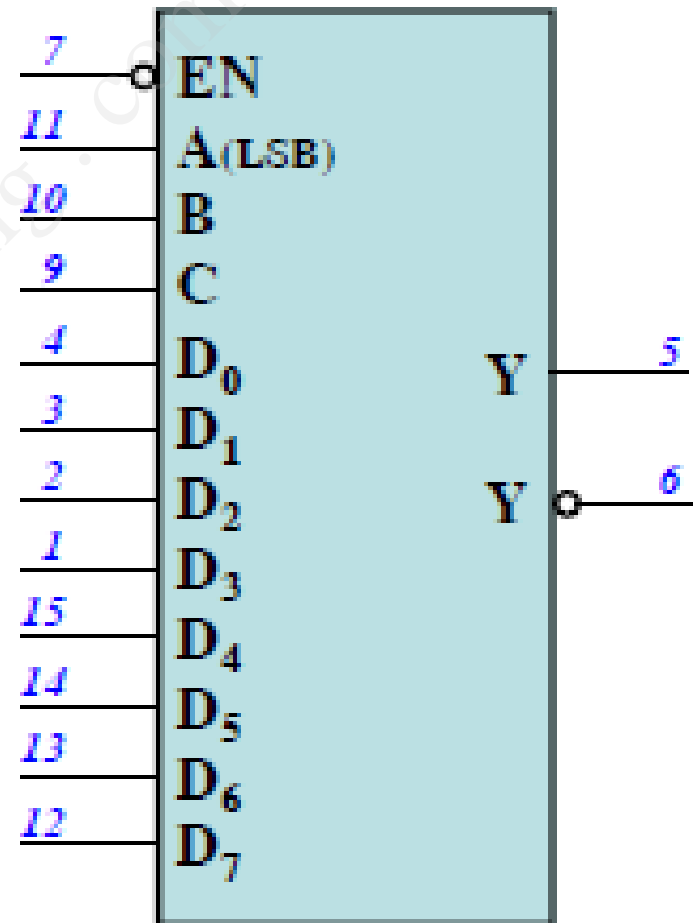
Tổng quát: $Y = \sum m_i D_i$ (với $i = 0, 1, \dots, 2^n - 1$)

IC dồn kênh

a. 74LS153: gồm 2 bộ MUX 4 \rightarrow 1



b. 74151: bộ MUX 8 \rightarrow 1



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. Bộ MUX 2ⁿ thực hiện hàm Boole n biến:

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

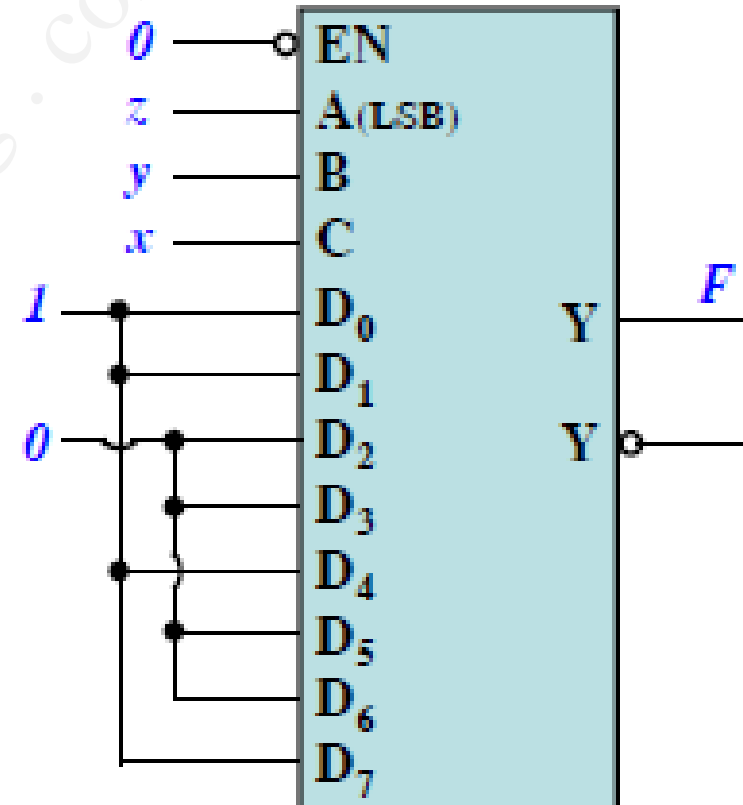
$$= m_0 + m_1 + m_4 + m_7$$

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

$$Y = \sum m_i D_i$$

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

$$\Rightarrow \begin{aligned} D_0 &= D_1 = D_4 = D_7 = 1 \\ D_2 &= D_3 = D_5 = D_6 = 0 \end{aligned}$$



b. Bộ MUX 2ⁿ thực hiện hàm Boole n+1 biến:

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

$$= \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z + x\bar{y}\bar{z} + xyz$$

$$= \bar{x}\bar{y}.1 + \bar{x}\bar{y}.0 + x\bar{y}.\bar{z} + xy.z$$

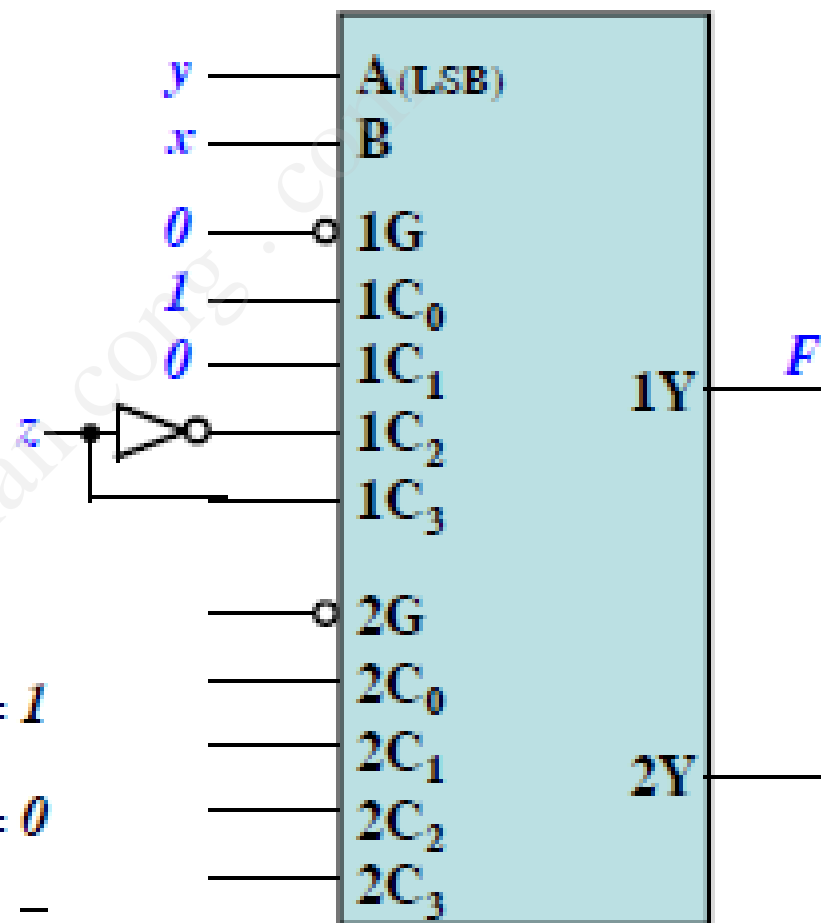
$$= m_0.1 + m_1.0 + m_2.\bar{z} + m_3.z$$

$$Y = m_0D_0 + m_1D_1 + m_2D_2 + m_3D_3$$

$$\Rightarrow D_0 = 1; D_1 = 0; D_2 = \bar{z}; D_3 = z$$

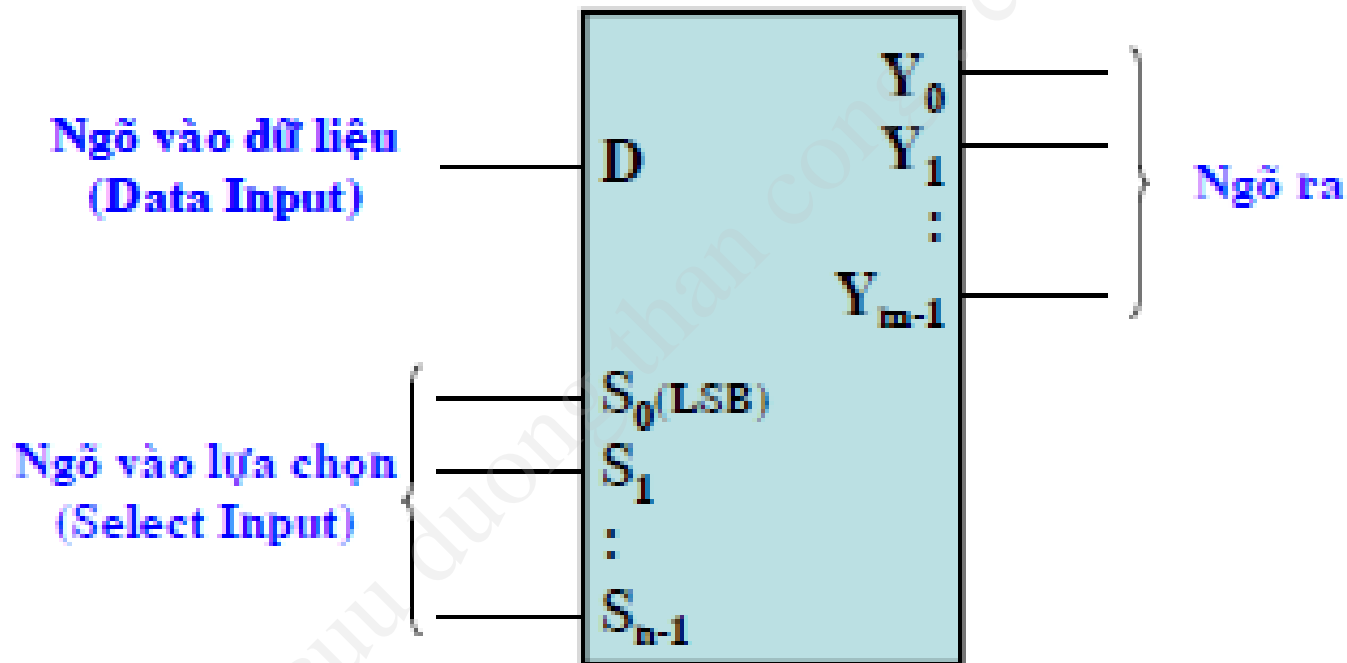
x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$\left. \begin{matrix} 1 \\ 1 \end{matrix} \right\} D_0 = 1$
 $\left. \begin{matrix} 0 \\ 0 \end{matrix} \right\} D_1 = 0$
 $\left. \begin{matrix} 1 \\ 0 \end{matrix} \right\} D_2 = \bar{z}$
 $\left. \begin{matrix} 0 \\ 1 \end{matrix} \right\} D_3 = z$



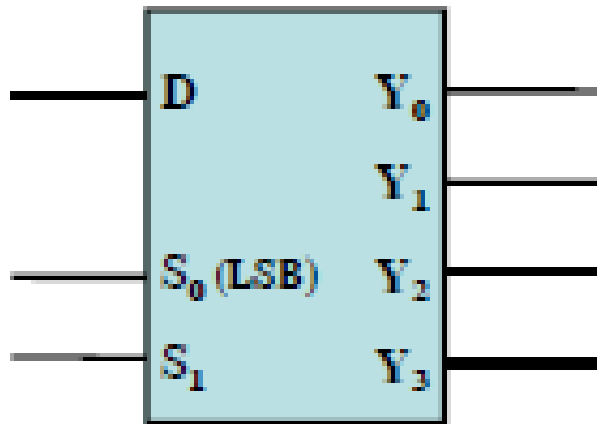
BỘ PHÂN KÊNH (DEMULTIPLEXER-DEMUX)

- Bộ 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.



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

*** Bộ DEMUX 1 → 4:**



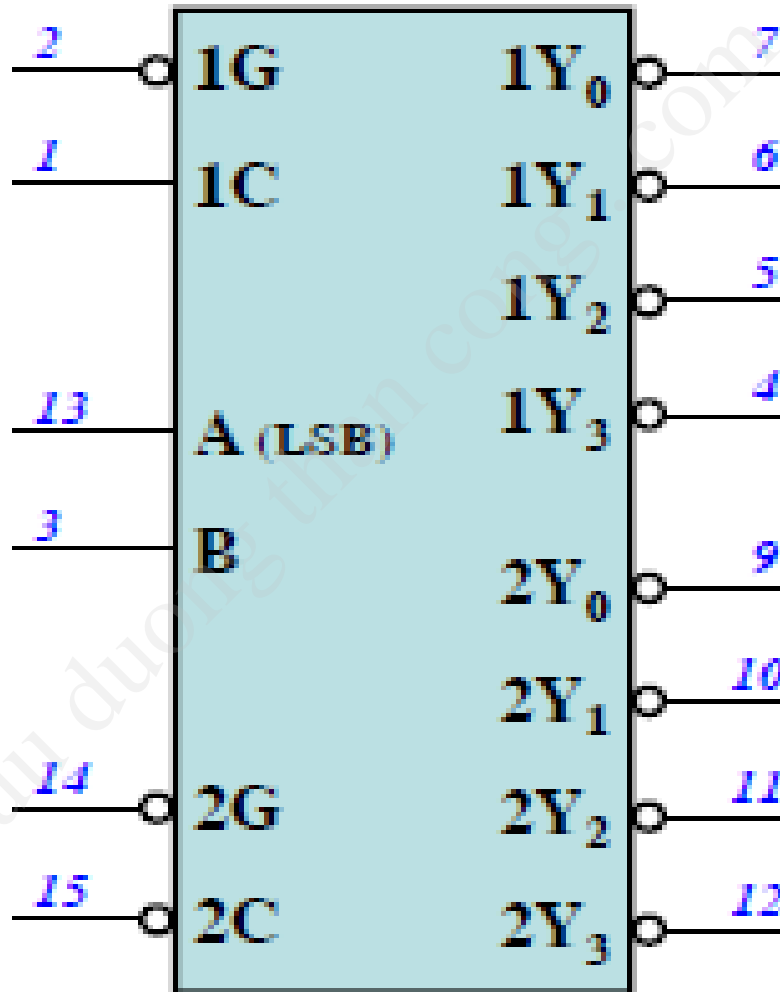
$$Y_0 = \overline{S_1} \overline{S_0} D = m_0 D$$

$$Y_1 = \overline{S_1} S_0 D = m_1 D$$

$$Y_2 = S_1 \overline{S_0} D = m_2 D$$

$$Y_3 = S_1 S_0 D = m_3 D$$

IC phân kênh 74155 (74LS155)



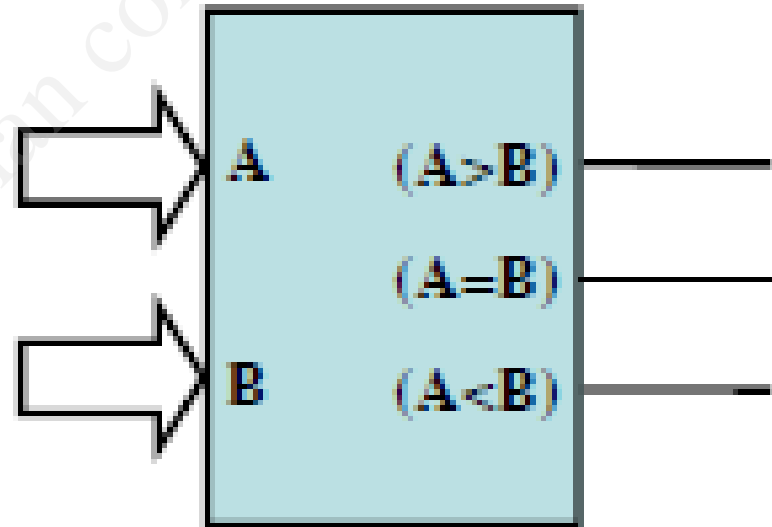
BỘ SO SÁNH ĐỘ LỚN (COMPARATOR)

- Bộ so sánh là hệ tổ hợp có nhiệm vụ so sánh 2 số nhị phân không dấu A và B (mỗi số n bit).
- Bộ so sánh có 3 ngõ ra ($A > B$), ($A = B$) và ($A < B$); chỉ có 1 ngõ ra tích cực theo kết quả so sánh.

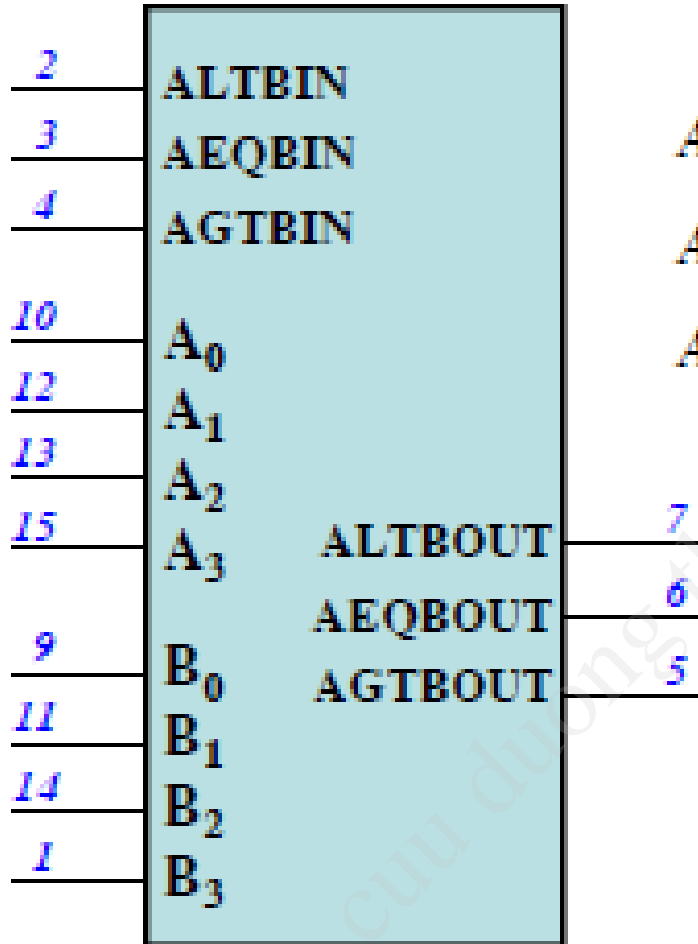
* Bộ so sánh 3 bit:

A: A_2 A_1 A_0

B: B_2 B_1 B_0



IC SO SÁNH 7485 (74LS85)



$$AGTBOUT = (A > B) + (A = B)AGTBIN$$

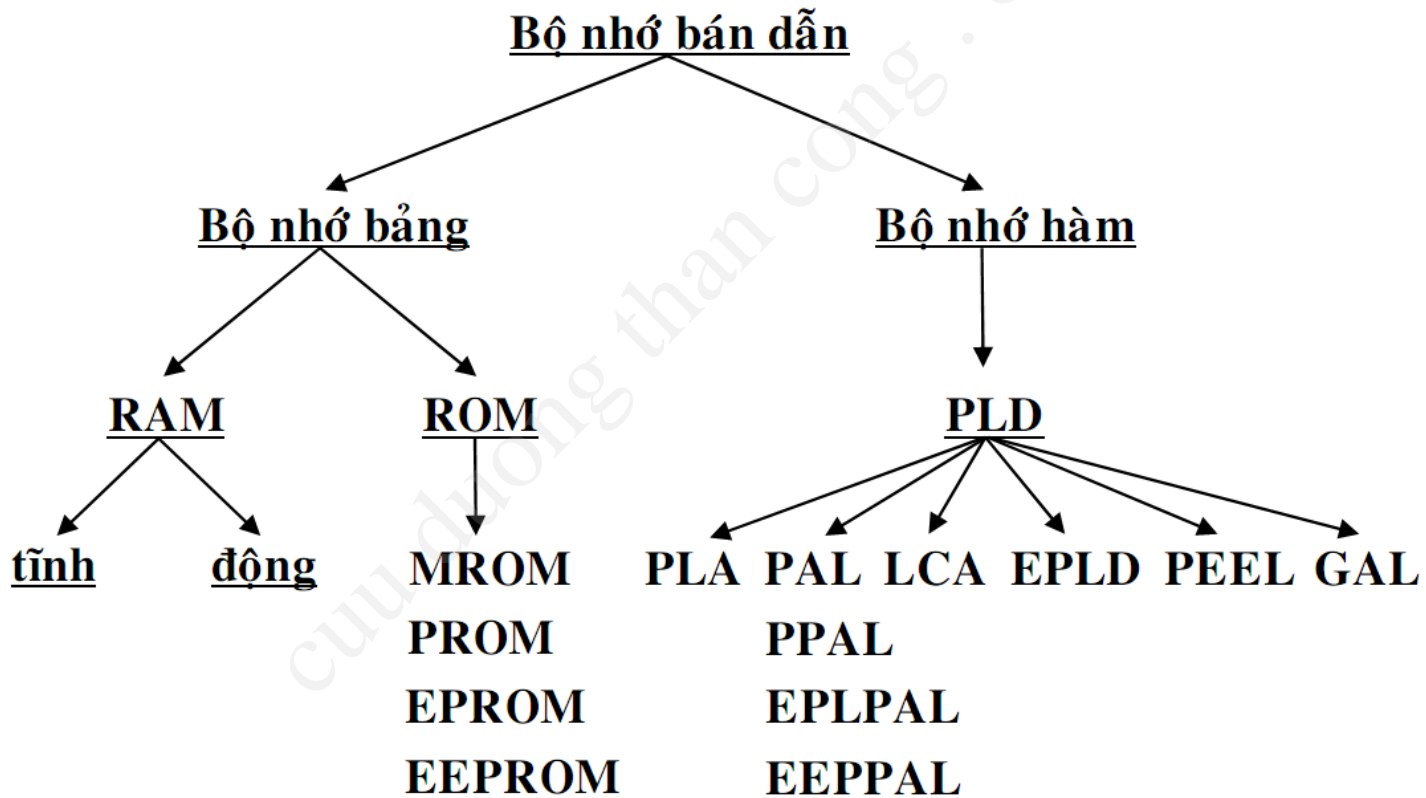
$$AEQBOUT = (A = B) AEQBIN$$

$$ALTBOUT = (A < B) + (A = B)ALTBIN$$

PLD (PROGRAMMABLE LOGIC DEVICE)

Thiết bị logic lập trình được

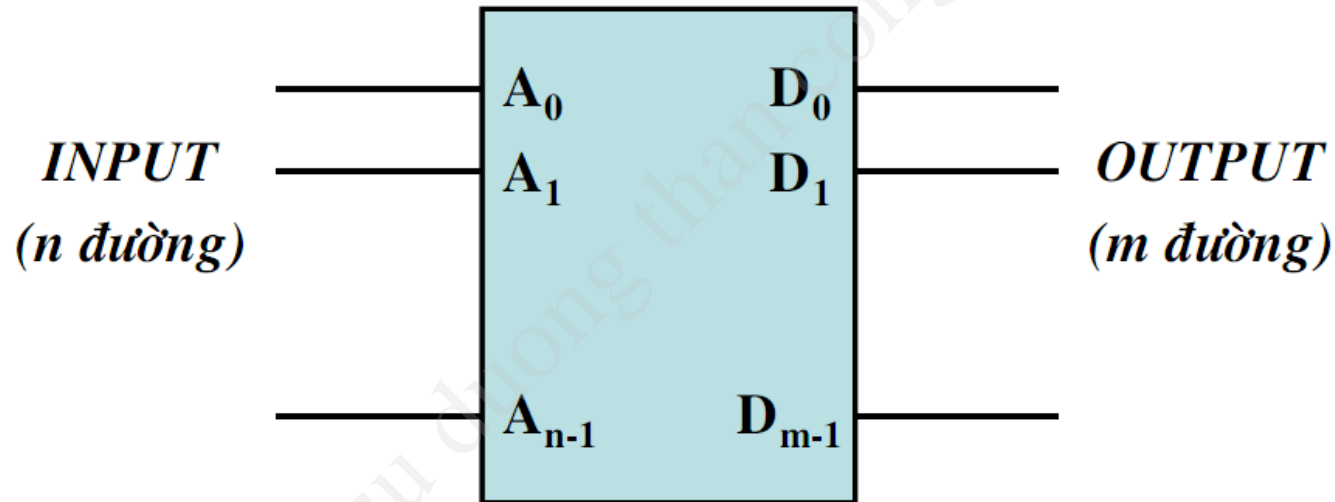
BỘ NHỚ BÁN DẪN



BỘ NHỚ ROM

Input: các tín hiệu địa chỉ (Address)

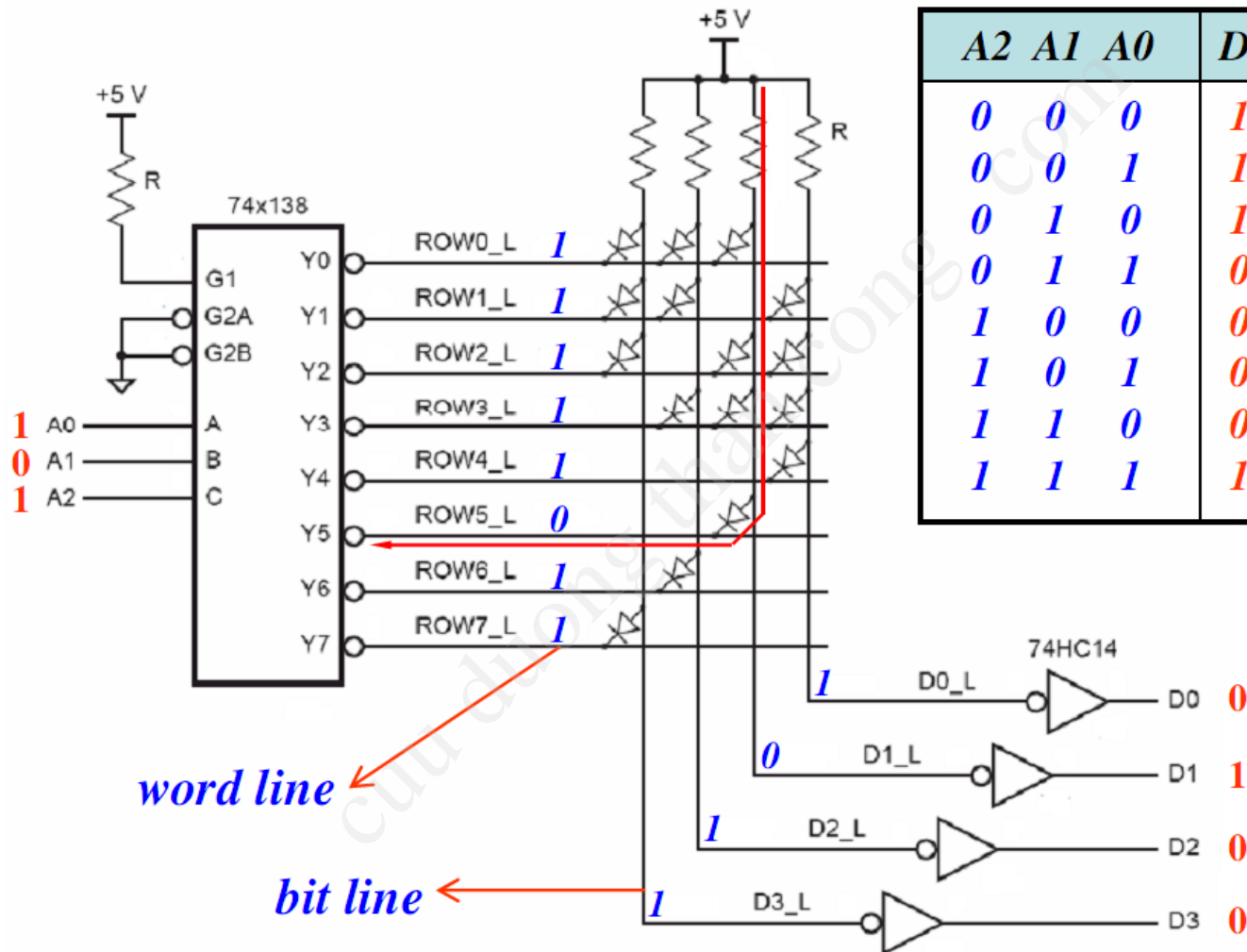
Output: các tín hiệu dữ liệu (Data)



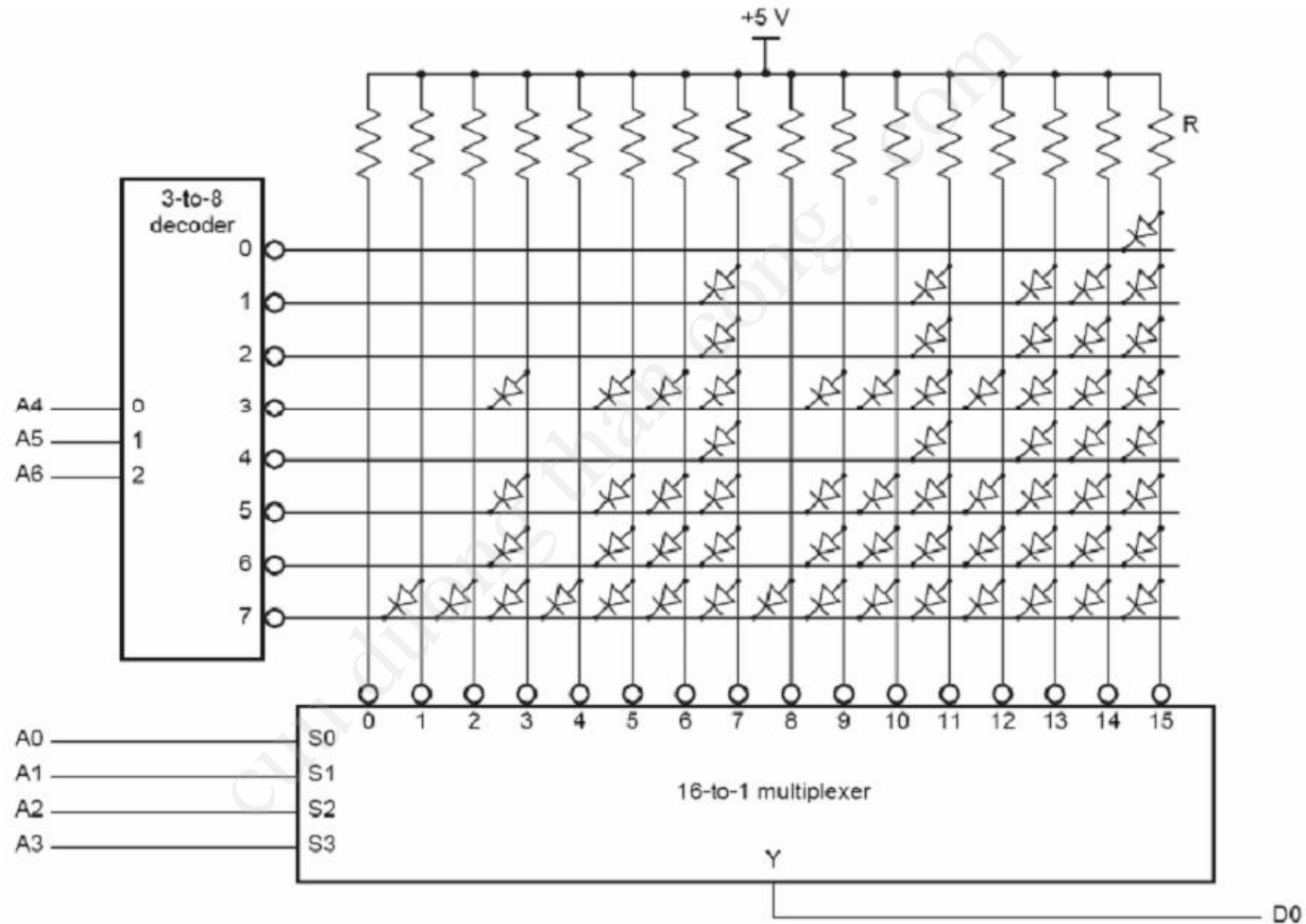
Kích thước ROM: $2^n \times m \text{ (bit)}$

Cấu trúc nội ROM 8 x 4 (bit)

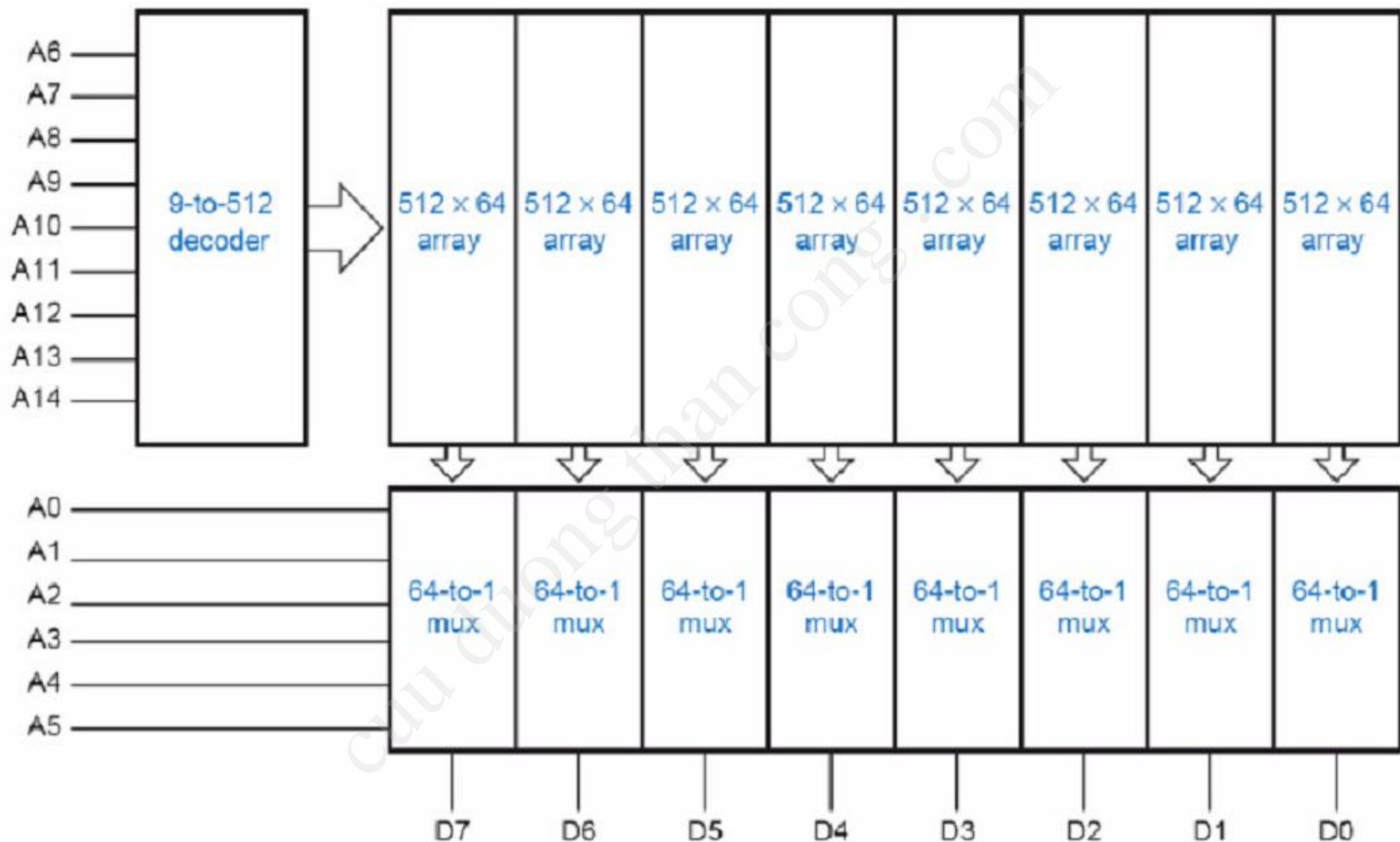
Bảng nạp ROM



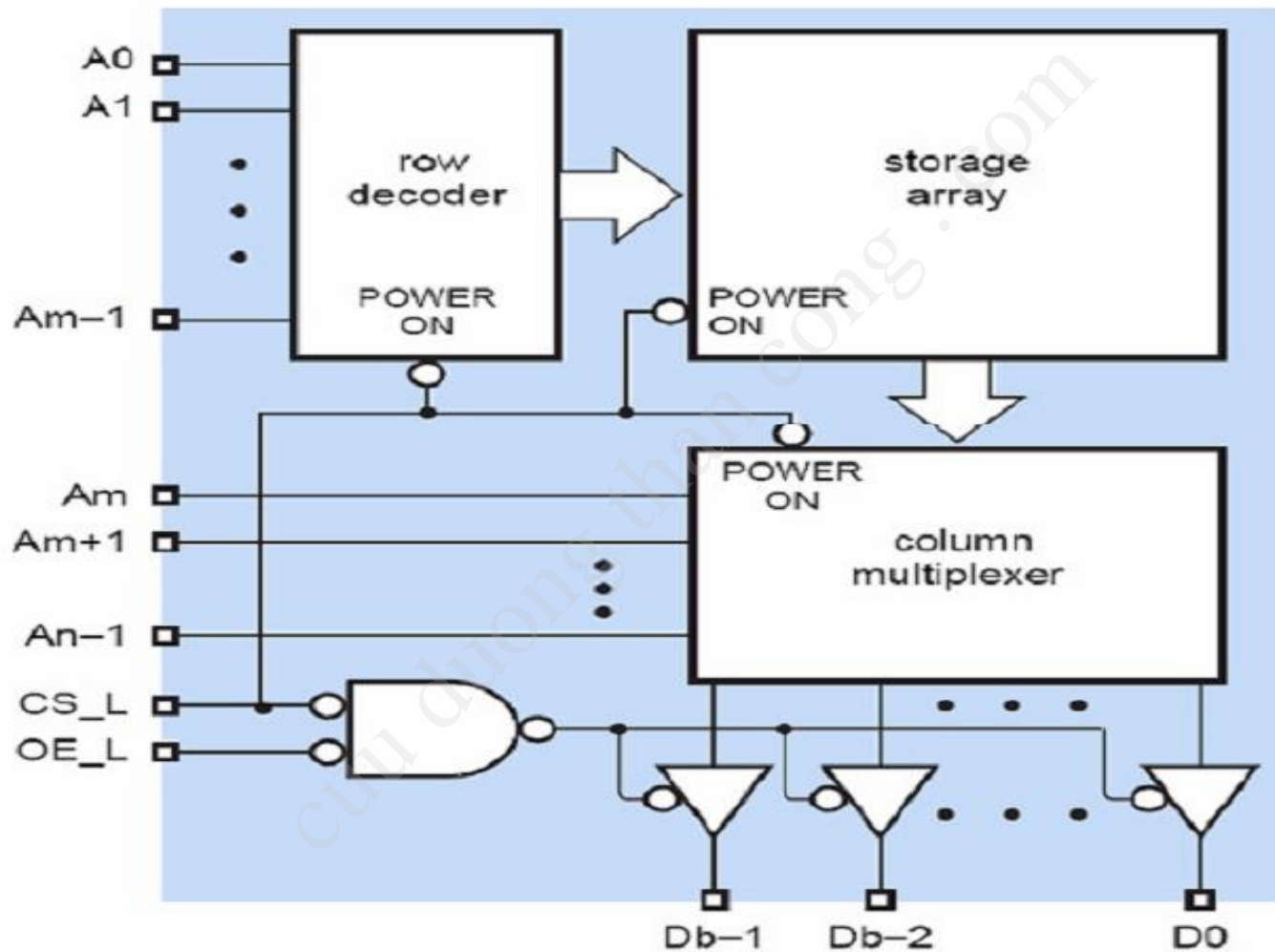
ROM 128 x 1 (bit) giải mã 2 chiều



ROM 32K x 8 (bit) = 32KB

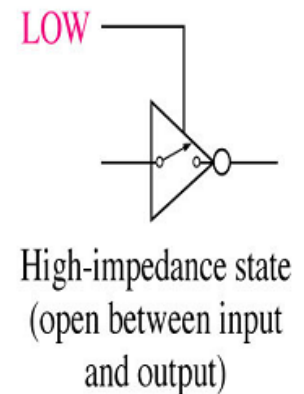
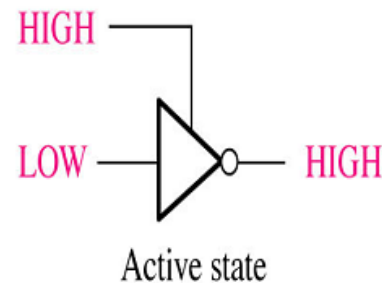
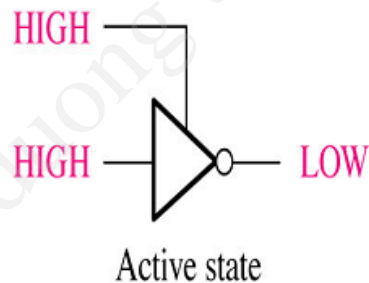
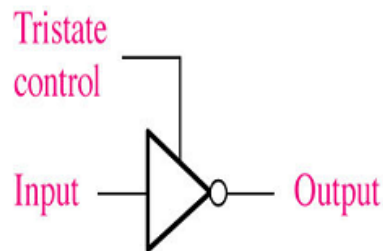


Cấu trúc ROM có ngõ vào điều khiển

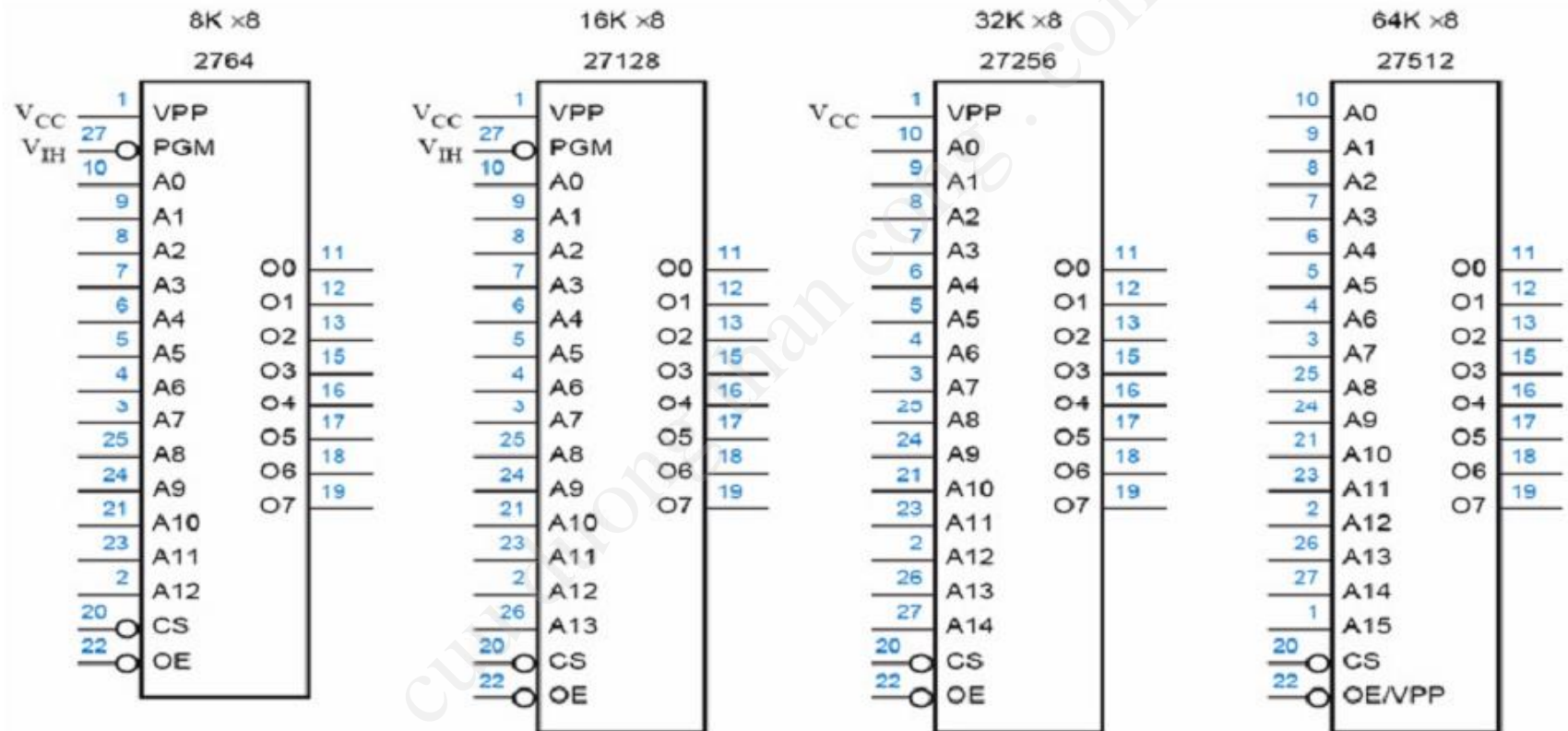


Cổng đệm ba trạng thái (*Tristate Output Buffer*):

- 3 trạng thái (tristate): **LOW / HIGH / HIGH impedance**
- Trạng thái tổng trở cao (**HIGH impedance**): ngõ ra hở mạch
- Ngõ điều khiển 3 trạng thái:
 - * **HIGH: The buffer is Active**
 - * **LOW: HIGH impedance**

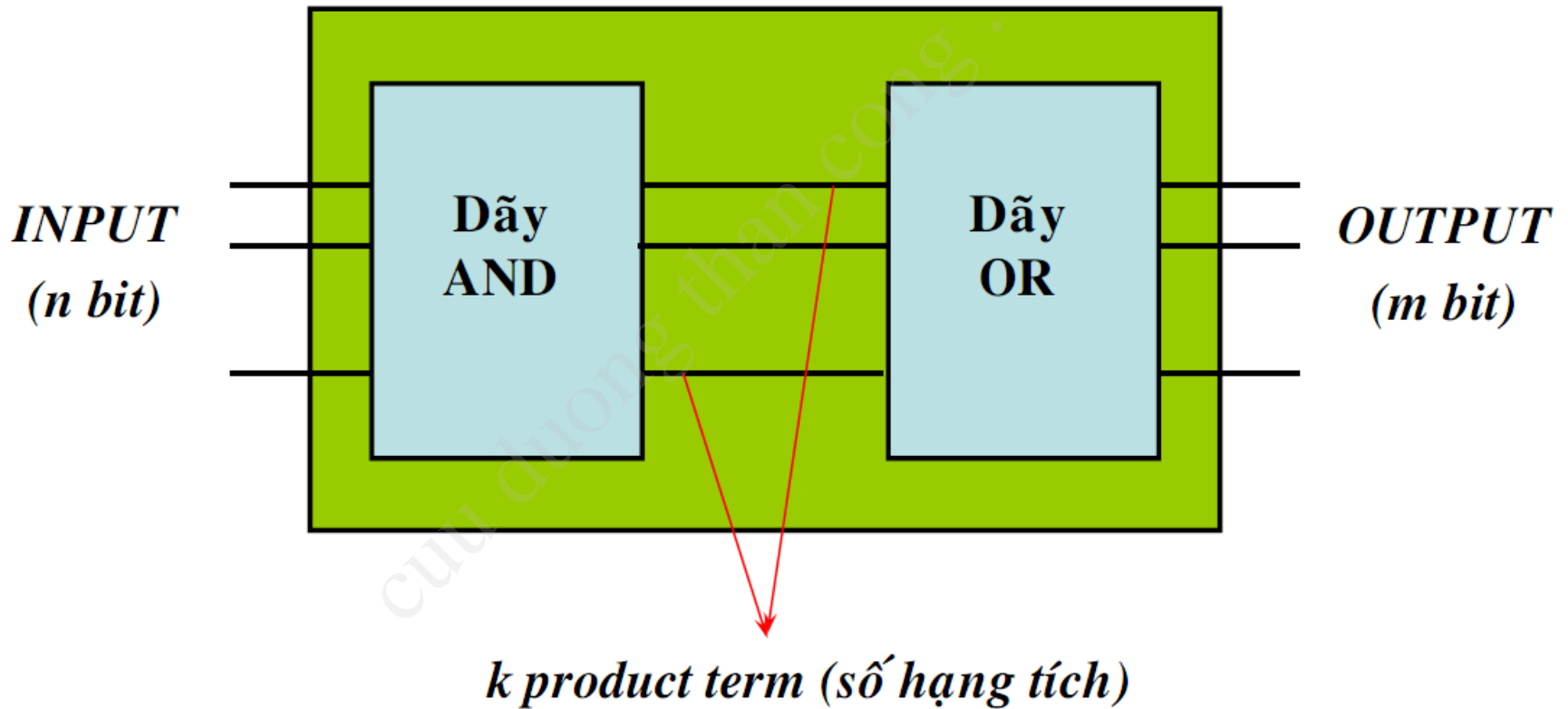


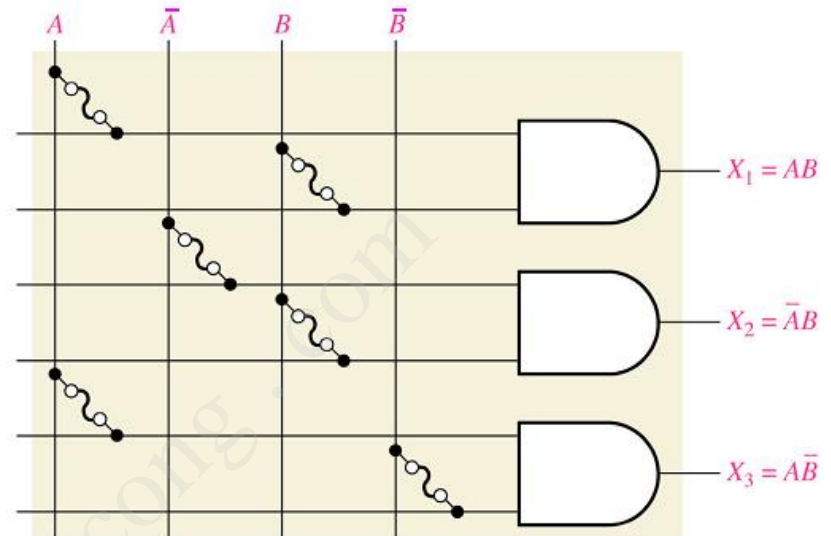
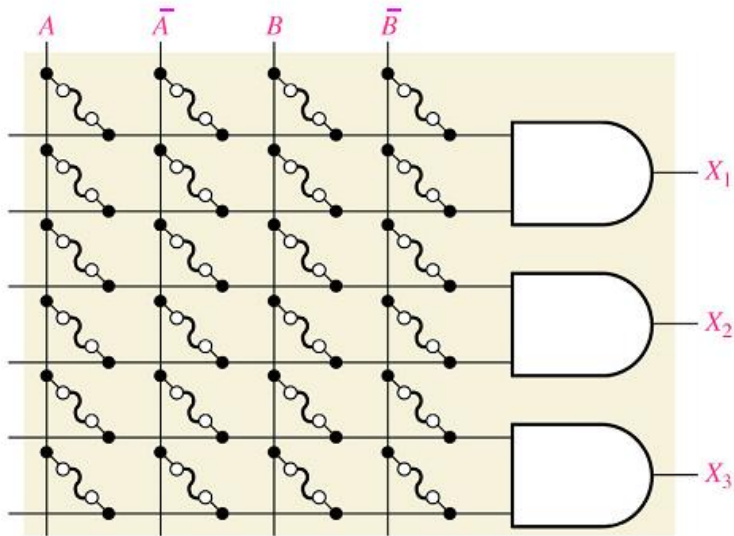
Các EPROM thông dụng



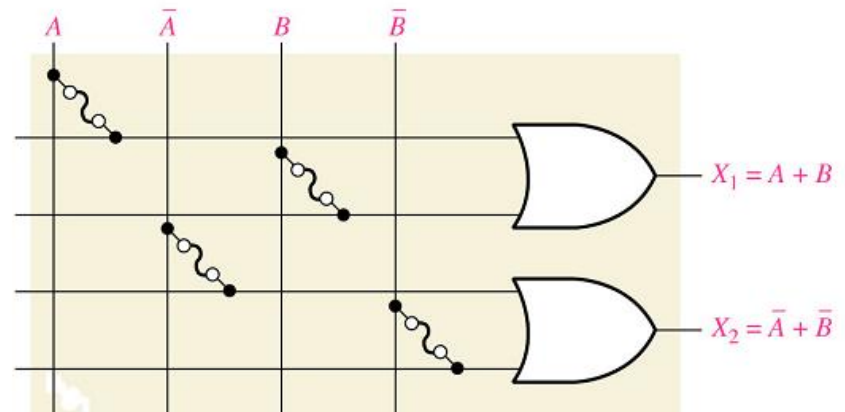
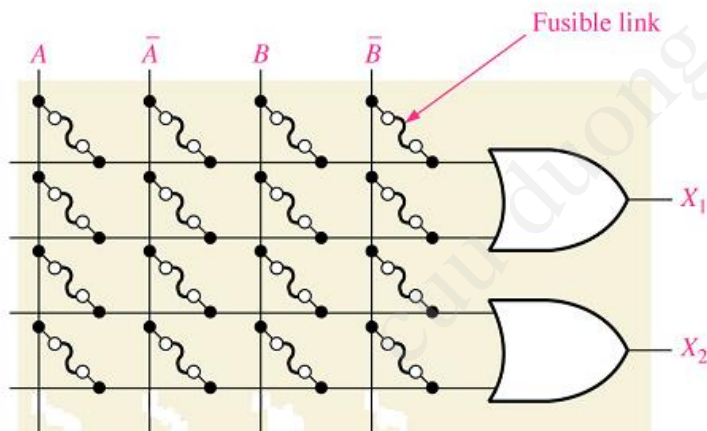
PLA

(PROGRAMMABLE LOGIC ARRAY)





Dãy AND có thể lập trình

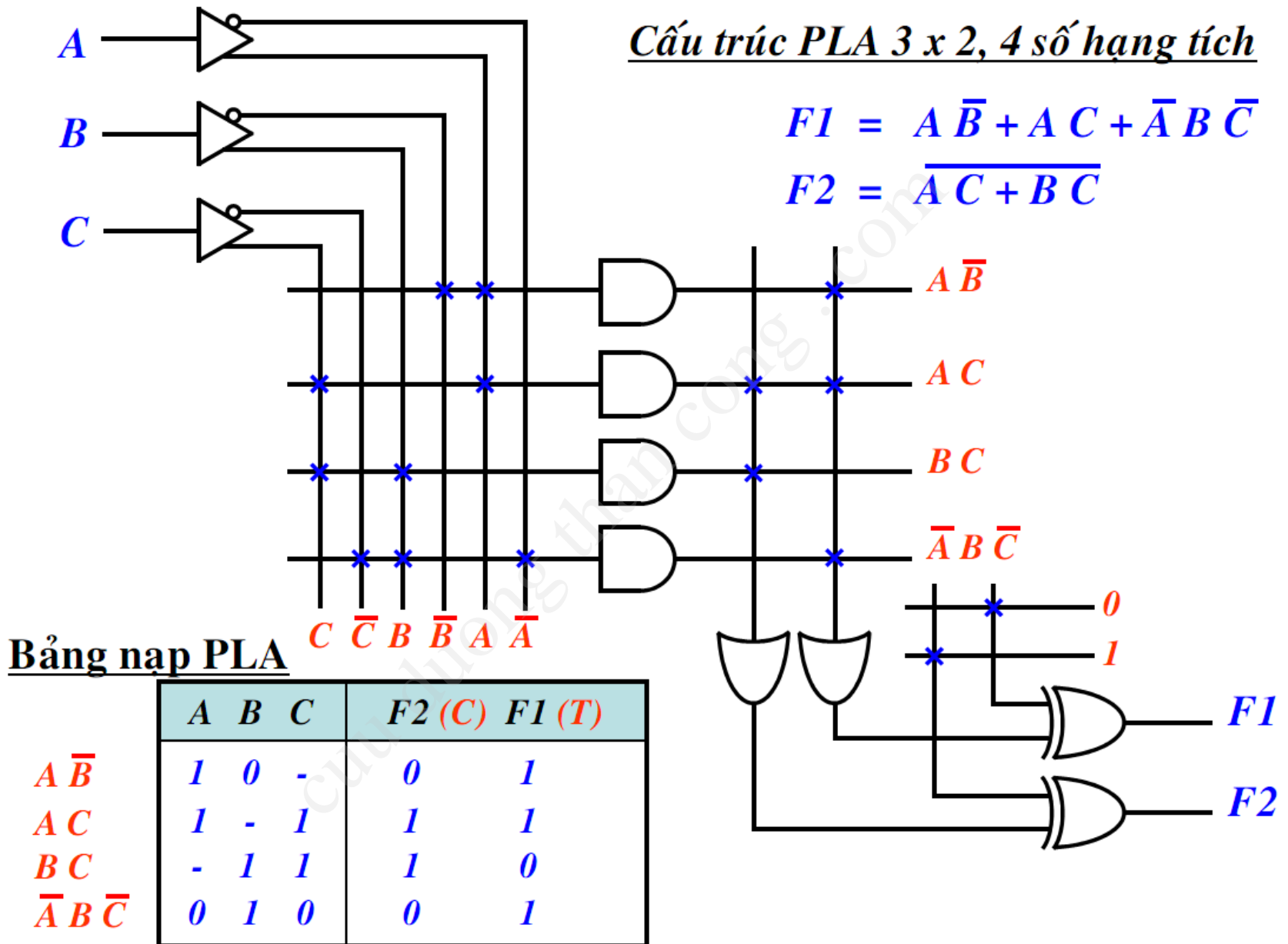


Dãy OR có thể lập trình

Cấu trúc PLA 3 x 2, 4 số hạng tích

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

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

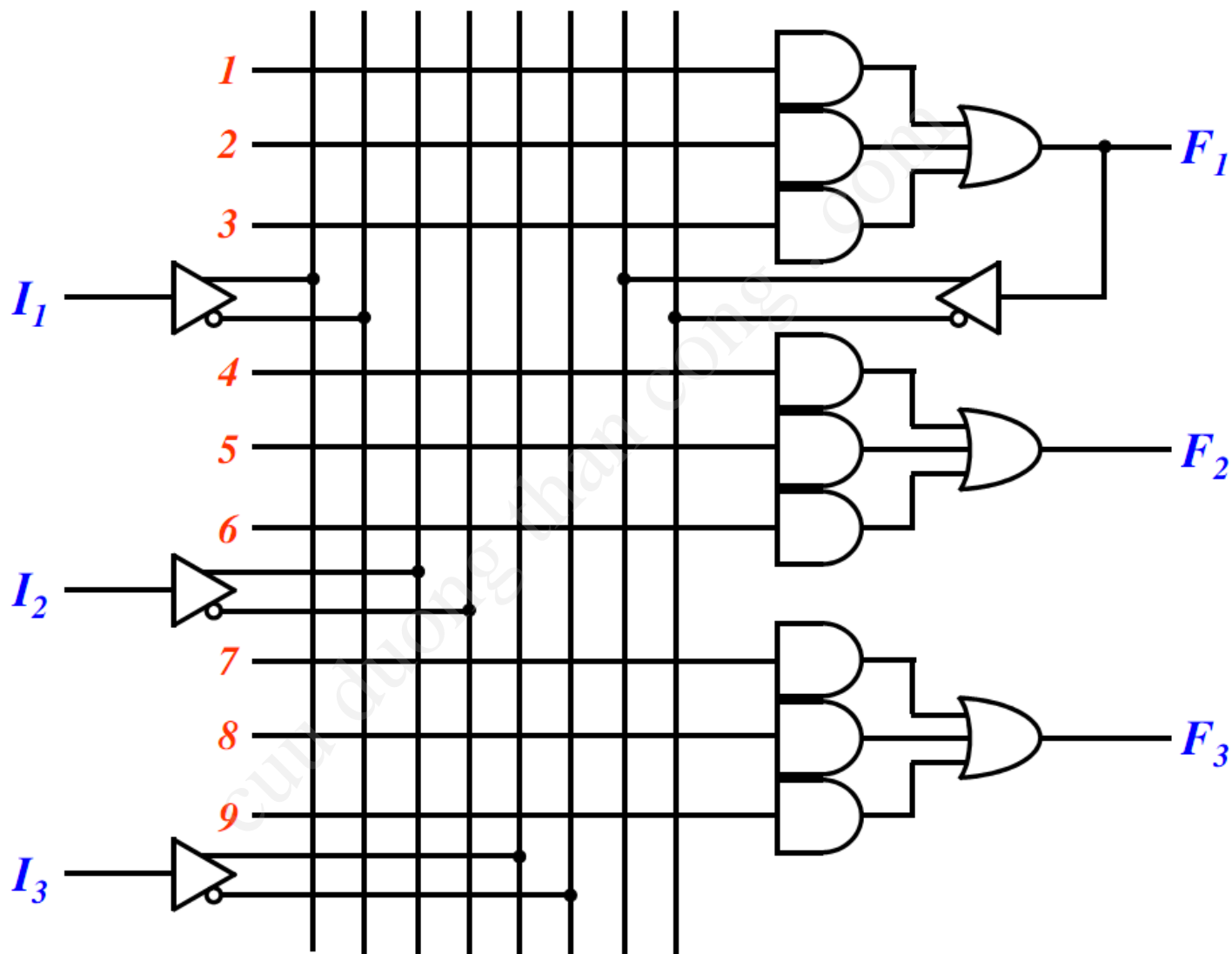


PAL

(PROGRAMMABLE ARRAY LOGIC)

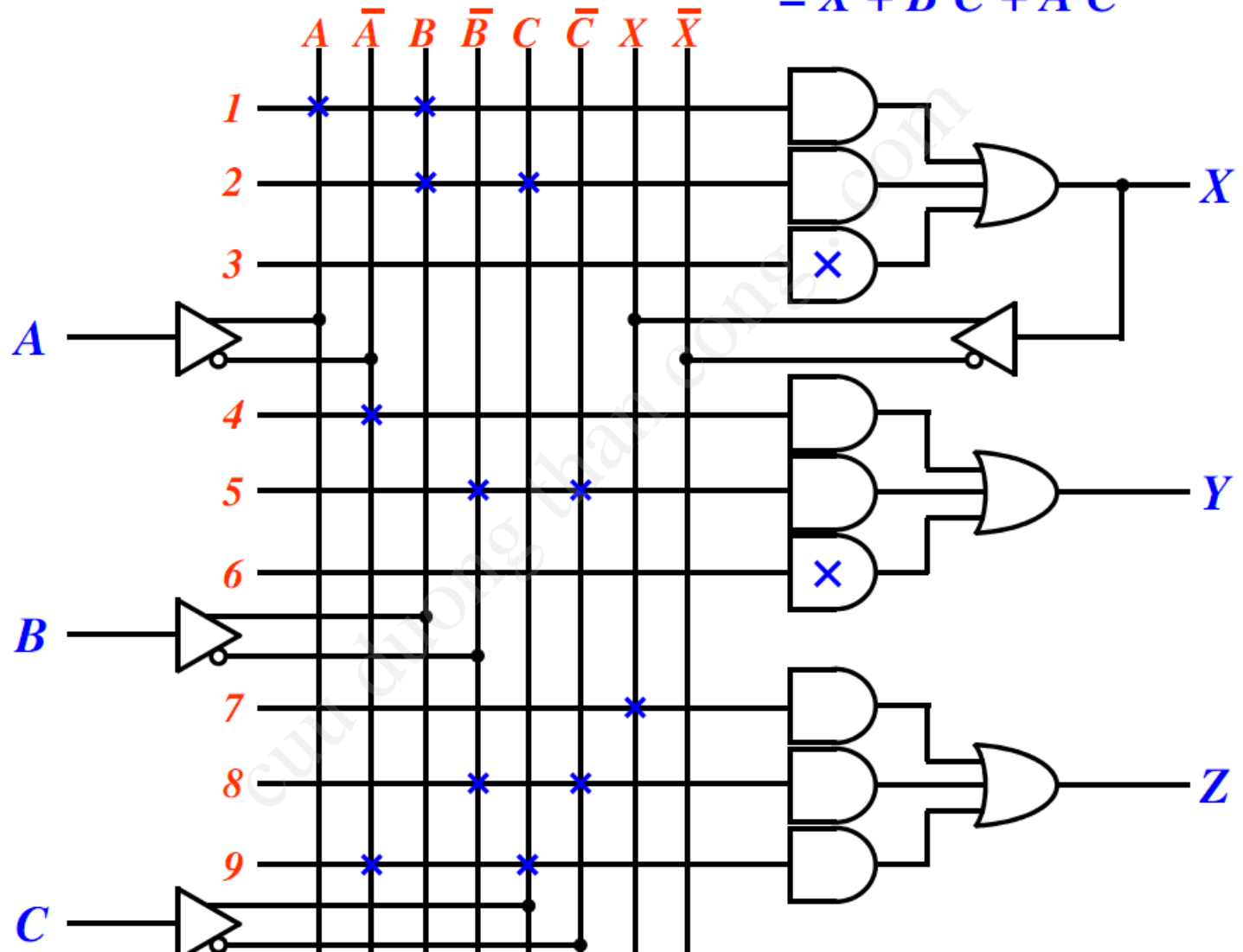
- **Dãy AND lập trình, dãy OR cố định**
- **Mỗi ngõ ra là cổng OR có số ngõ vào cố định**
- **Số hạng tích không sử dụng chung cho các ngõ ra**
- **Cấu trúc PLA: số ngõ vào, số ngõ ra và số cổng AND trên 1 cổng OR**

Cấu trúc PAL 3 ngõ vào, 3 ngõ ra, 3 cổng AND / OR



$$X = AB + BC \quad Y = \bar{A} + \bar{B}\bar{C} \quad Z = AB + BC + \bar{B}\bar{C} + \bar{A}C$$

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



$$X = A B + B C \quad Y = \bar{A} + \bar{B} \bar{C} \quad Z = A B + B C + \bar{B} \bar{C} + \bar{A} C$$

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

Bảng nạp PAL

	<i>A</i>	<i>B</i>	<i>C</i>	<i>X</i>	<i>OUTPUT</i>
<i>1</i>	<i>1</i>	<i>1</i>	—	—	<i>X</i> = <i>A B</i> + <i>B C</i>
<i>2</i>	—	<i>1</i>	<i>1</i>	—	
<i>3</i>	—	—	—	—	
<i>4</i>	<i>0</i>	—	—	—	<i>Y</i> = \bar{A} + $\bar{B} \bar{C}$
<i>5</i>	—	<i>0</i>	<i>0</i>	—	
<i>6</i>	—	—	—	—	
<i>7</i>	—	—	—	<i>1</i>	<i>Z</i> = <i>X</i> + $\bar{B} \bar{C}$ + $\bar{A} C$
<i>8</i>	—	<i>0</i>	<i>0</i>	—	
<i>9</i>	<i>0</i>	—	<i>1</i>	—	