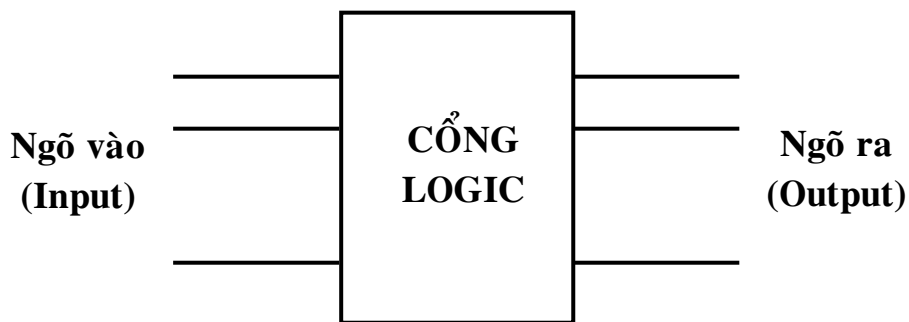


Chương 3: HỆ TỔ HỢP

I. GIỚI THIỆU – CÁCH THIẾT KẾ HỆ TỔ HỢP:

Mạch logic được chia làm 2 loại: mạch tổ hợp (Combinational Circuit) và mạch tuần tự (Sequential Circuit).

Mạch tổ hợp là mạch mà các ngõ ra chỉ phụ thuộc vào giá trị của các ngõ vào. Mọi sự thay đổi của ngõ vào sẽ làm ngõ ra thay đổi theo.



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

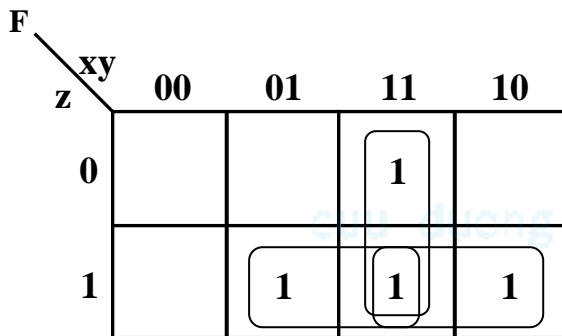
Vd: Thiết kế hệ tổ hợp có 3 ngõ vào x, y, z ; và 2 ngõ ra F, G . Ngõ ra F là 1 nếu như 3 ngõ vào có số bit 1 nhiều hơn số bit 0; ngược lại $F = 0$. Ngõ ra G là 1 nếu như giá trị nhị phân của ngõ vào lớn hơn 1 và nhỏ hơn 6; ngược lại $G = 0$.

- Hệ có 3 ngõ vào: x, y, z và 2 ngõ ra: F, G

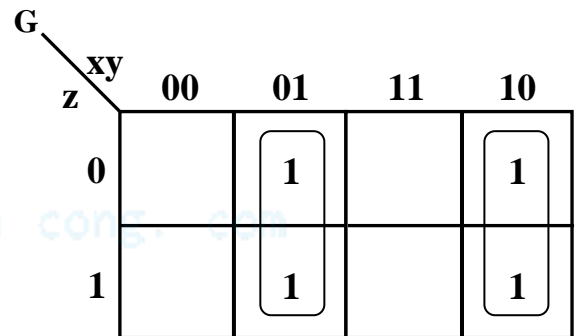
- Bảng giá trị:

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

- Tìm biểu thức rút gọn:

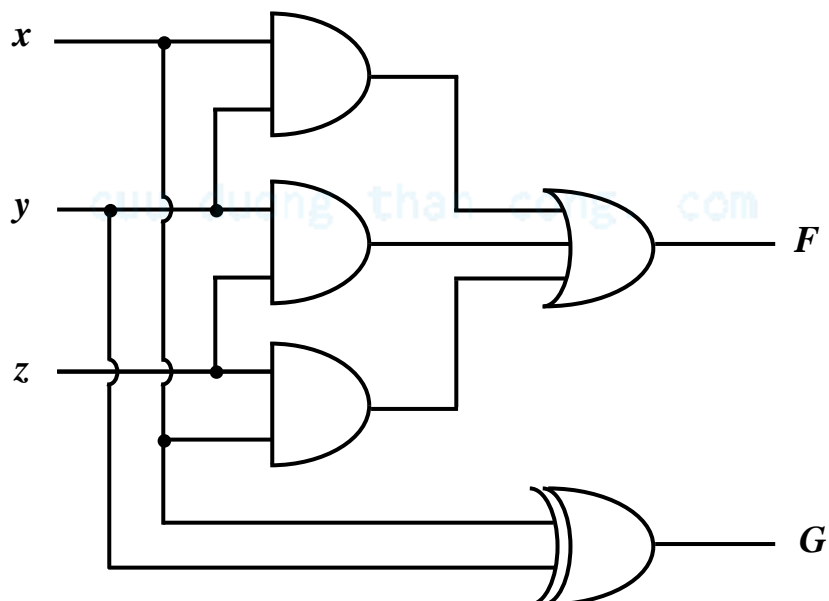


$$F = x y + y z + x z$$



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

- Thực hiện sơ đồ logic:



Nếu hệ tổ hợp không sử dụng tất cả 2^n tổ hợp của ngõ vào, thì tại các tổ hợp không sử dụng đó ngõ ra có giá trị tùy định.

Vd: Thiết kế hệ tổ hợp có ngõ vào biểu diễn cho 1 số mã BCD. Nếu giá trị ngõ vào nhỏ hơn 3 thì ngõ ra có giá trị bằng bình phương giá trị ngõ vào; ngược lại giá trị ngõ ra bằng giá trị ngõ vào trừ đi 3.

- Hệ có 4 ngõ vào và 3 ngõ ra.
- Bảng giá trị của hệ

A	B	C	D	F2	F1	F0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	1	0	0
0	0	1	1	0	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	1	0	1
1	0	0	1	1	1	0
1	0	1	0	x	x	x
1	0	1	1	x	x	x
1	1	0	0	x	x	x
1	1	0	1	x	x	x
1	1	1	0	x	x	x
1	1	1	1	x	x	x

F2	AB			
	00	01	11	10
00			X	1
01			X	1
11		1	X	X
10	1		X	X

F1	AB			
	00	01	11	10
00			X	
01		1	X	1
11			X	X
10		1	X	X

F0	AB			
	00	01	11	10
00		1	X	1
01	1		X	
11			X	X
10		1	X	X

$$F2 = A + B C D + \bar{B} C \bar{D}$$

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

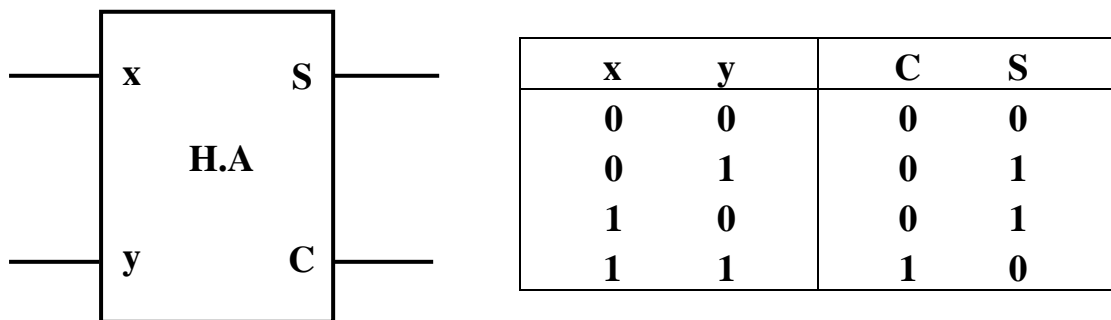
$$F0 = A \bar{D} + B \bar{D} + \bar{A} \bar{B} \bar{C} D$$

II. BỘ CỘNG - TRỪ NHỊ PHÂN:

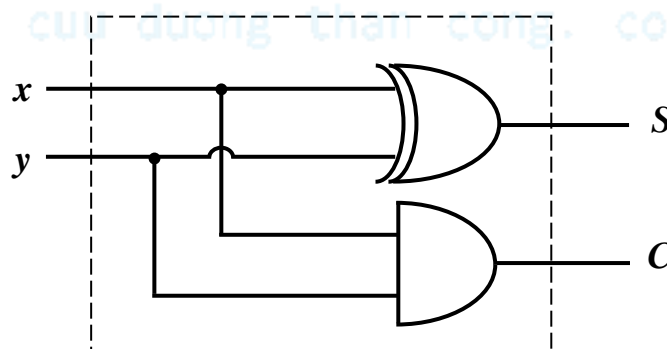
1. Bộ cộng (Adder):

a. Bộ cộng bán phần (Half Adder – H.A):

Bộ cộng bán phần là hệ tổ hợp có 2 ngõ vào x, y; 2 ngõ ra S (Sum) và C (Carry). Hệ có nhiệm vụ thực hiện phép cộng số học 2 bit nhị phân $x + y$.

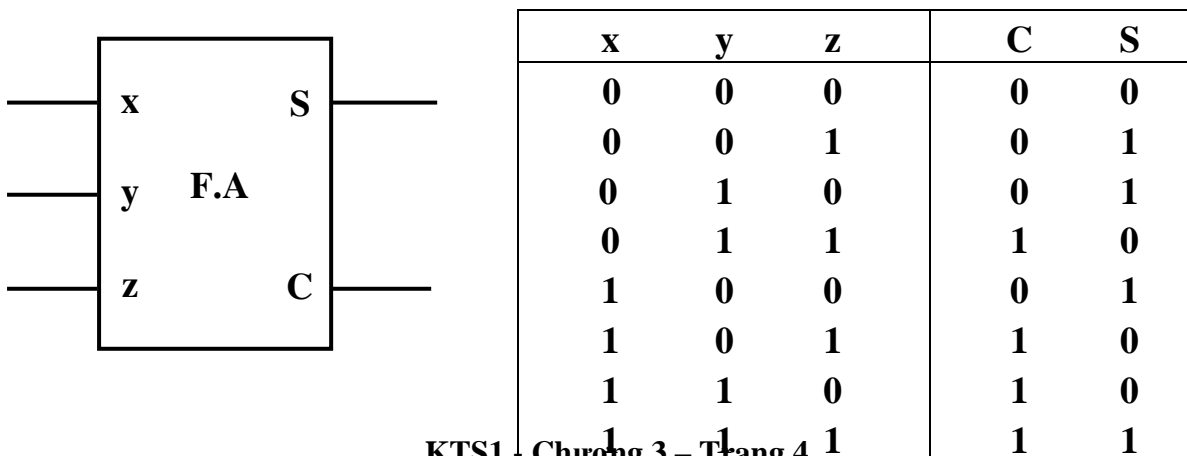


$$S = x \oplus y \text{ và } C = x y$$



b. Bộ cộng toàn phần (Full Adder – F.A):

Hệ có nhiệm vụ cộng số học 3 bit $x + y + z$ (z biểu diễn cho bit nhớ có được từ ví trị có trọng số nhỏ hơn gởi tới)

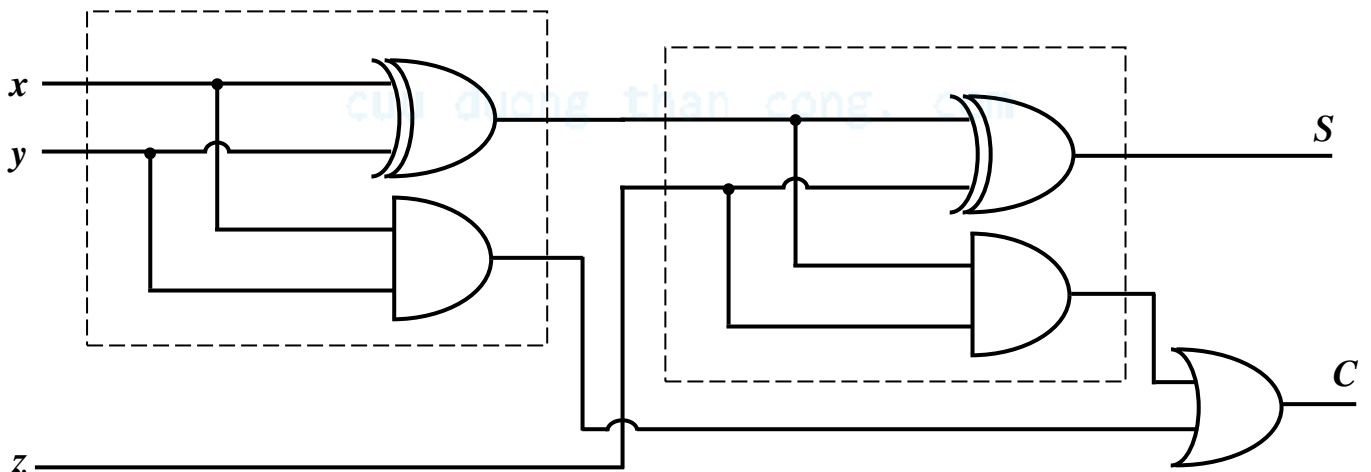


S z \ x y				
	00	01	11	10
0		1		1
1	1		1	

$$\begin{aligned}
 S &= \bar{x}\bar{y}z + \bar{x}y\bar{z} + x y z + x \bar{y}\bar{z} \\
 &= z(\bar{x}\bar{y} + x y) + \bar{z}(\bar{x}y + x \bar{y}) \\
 &= z(\overline{x \oplus y}) + \bar{z}(x \oplus y) \\
 &= z \oplus (x \oplus y)
 \end{aligned}$$

C z \ x y				
	00	01	11	10
0			1	
1		1	1	1

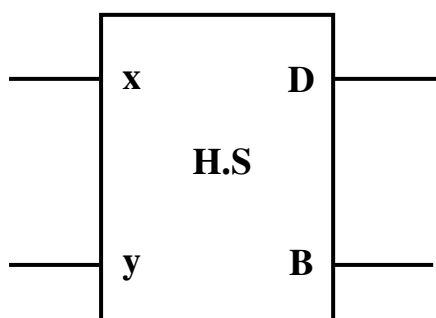
$$\begin{aligned}
 C &= x y + x z + y z \\
 &= x y + x \bar{y}z + x y z + \bar{x} y z \\
 &= x y + z(x \bar{y} + \bar{x} y) \\
 &= x y + z(x \oplus y)
 \end{aligned}$$



2. Bộ trừ (Subtractor):

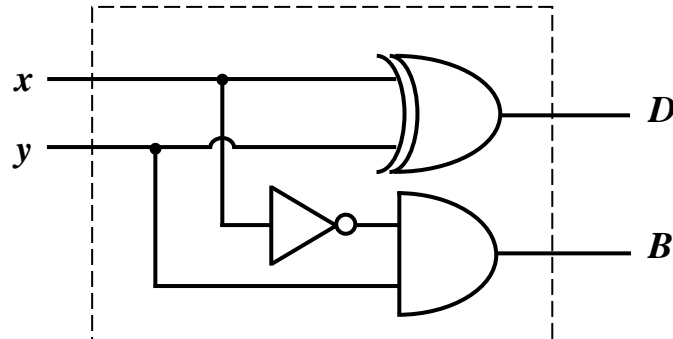
a. Bộ trừ bán phần (H.S):

Bộ trừ bán phần là hệ tổ hợp có 2 ngõ vào x, y; 2 ngõ ra D (Difference) và B (Borrow). Hệ có nhiệm vụ thực hiện phép trừ số học 2 bit nhị phân x - y.



x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0

$$D = x \oplus y \text{ và } B = \bar{x}y$$



b. Bộ trừ toàn phần (F.S):

Hệ có nhiệm vụ thực hiện phép trừ số học 3 bit $x - y - z$ (z biểu diễn cho bit mượn từ ví trị có trọng số nhỏ hơn gởi tới)

	x	y	z	B	D
	0	0	0	0	0
	0	0	1	1	1
	0	1	0	1	1
	0	1	1	1	0
	1	0	0	0	1
	1	0	1	0	0
	1	1	0	0	0
	1	1	1	1	1

D	z	x y	00	01	11	10
0				1		1
1			1		1	

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

$$= z(\bar{x}\bar{y} + x y) + \bar{z}(\bar{x}y + x \bar{y})$$

B	z	x y	00	01	11	10
0				1		
1			1	1	1	

$$B = \bar{x}y + \bar{x}z + yz$$

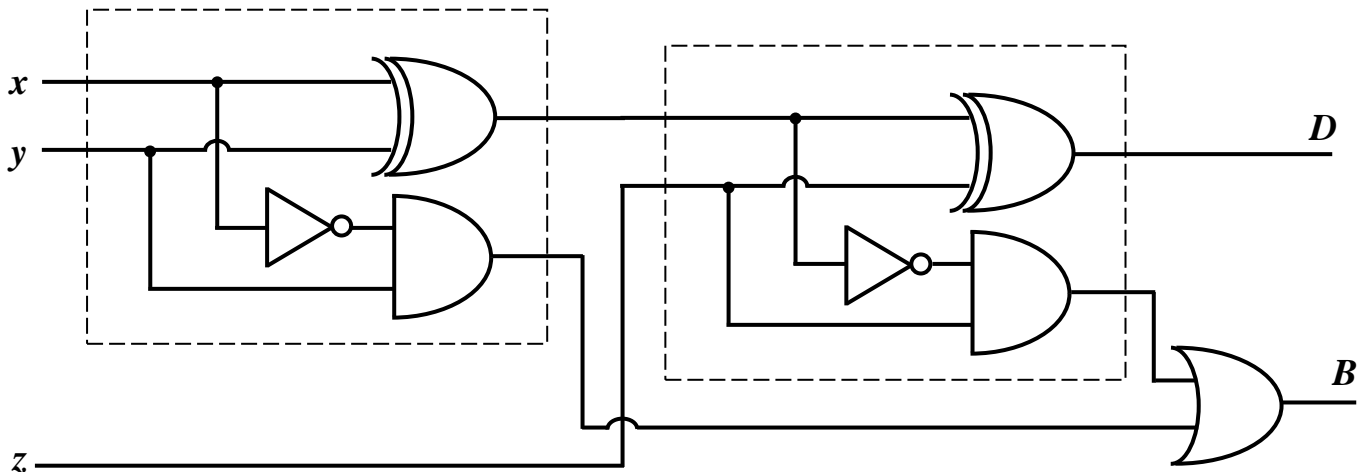
$$= \bar{x}y + \bar{x}\bar{y}z + \bar{x}y z + x y z$$

$$= z(\overline{x \oplus y}) + \overline{z}(x \oplus y)$$

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

$$= \overline{x}y + z(\overline{x} \overline{y} + xy)$$

$$= \overline{x}y + z(\overline{x \oplus y})$$



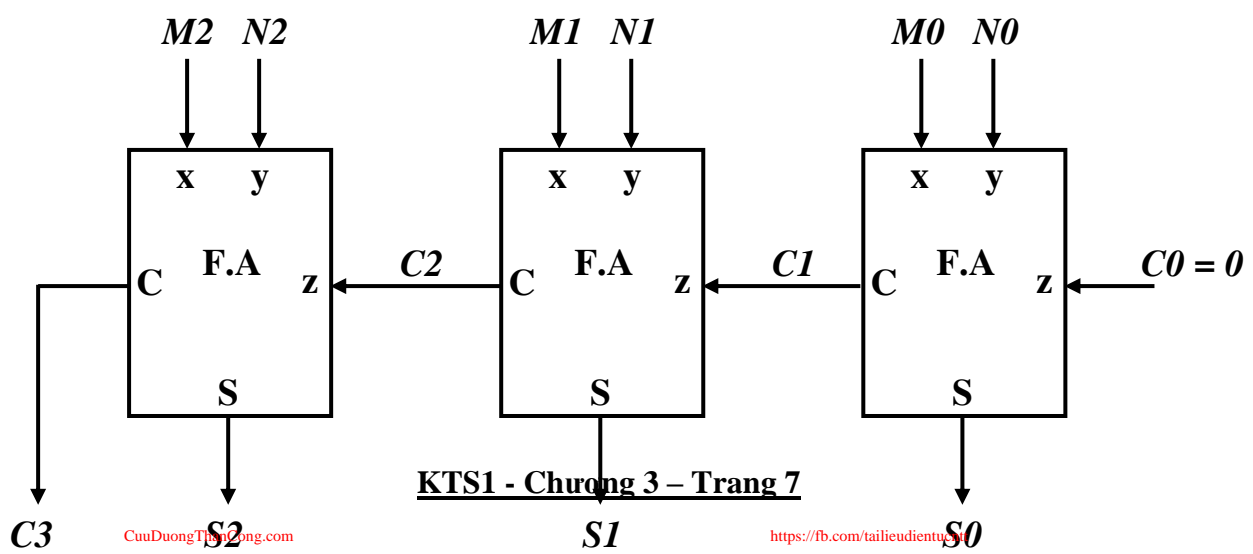
3. Bộ cộng/trừ nhị phân song song:

a. Bộ cộng nhị phân:

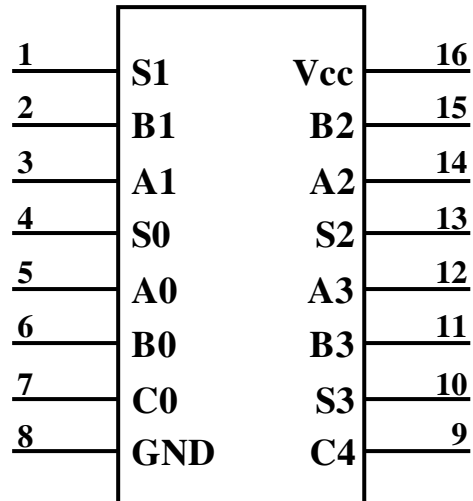
Cộng 2 số nhị phân M và N, mỗi số 3 bit

<i>bit nhớ:</i>		<i>C2</i>	<i>C1</i>		
<i>M</i>	:	<i>M2</i>	<i>M1</i>	<i>M0</i>	
<i>N</i>	:	<i>N2</i>	<i>N1</i>	<i>N0</i>	
	+				
<i>kết quả:</i>		<i>C3</i>	<i>S2</i>	<i>S1</i>	<i>S0</i>

Ta thực hiện bộ cộng nhị phân 3 bit bằng cách ghép 3 bộ cộng toàn phần



* IC cộng nhị phân song song 4 bit 74LS283

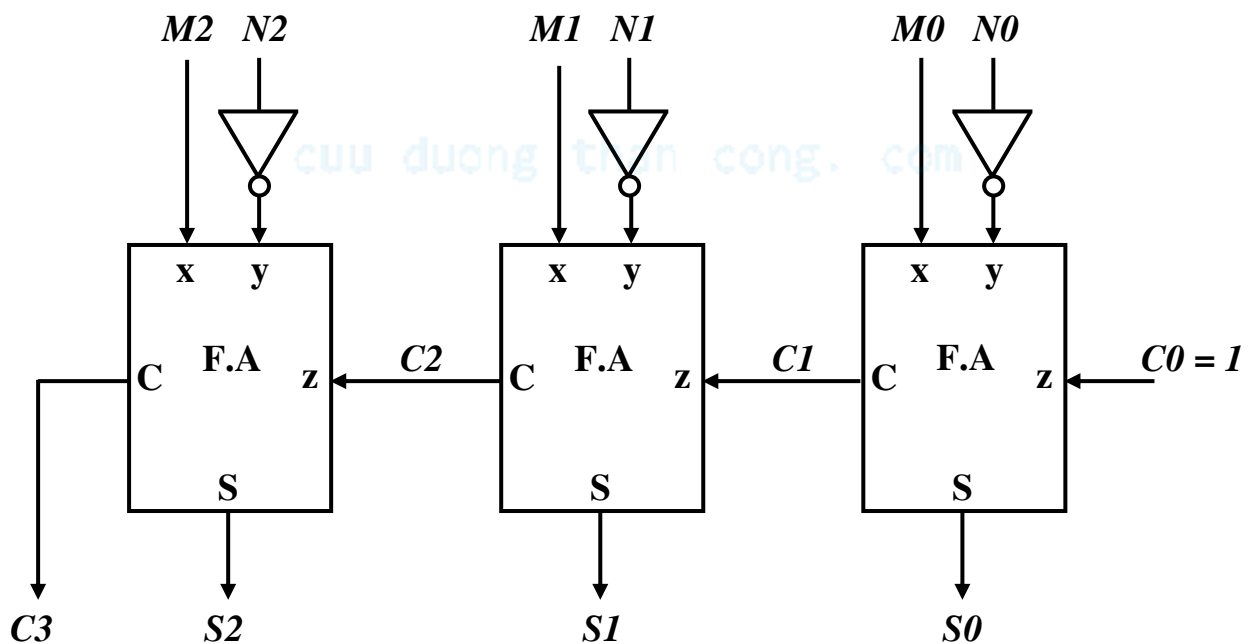


b. Bộ trừ nhị phân:

Có 2 cách :

- Ghép n bộ trừ toàn phần F.S.
- Thực hiện phép cộng với bù 2 của số trừ.

$$M - N = M + BÙ_2(N) = M + BÙ_1(N) + 1$$



Kết quả: - $C3 = 1$ kết quả là số dương

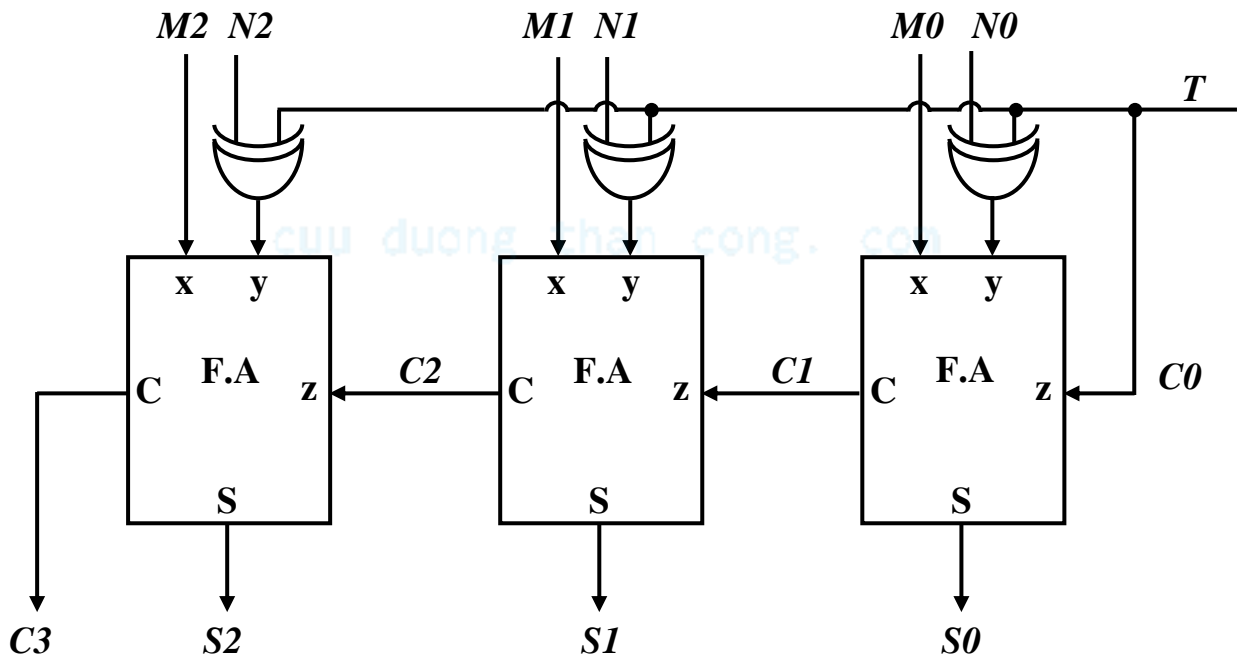
- $C3 = 0$ kết quả là số âm

c. Bộ cộng/trừ nhị phân:

Ta có thể kết hợp phép cộng và trừ trên cùng một bộ cộng nhị phân. Ta sử dụng thêm 1 biến điều khiển T : $T = 0$ thực hiện phép cộng, $T = 1$ thực hiện phép trừ. Ta thấy phép cộng và phép trừ khác nhau ở ngõ vào y_i và C_0 .

	T	y_i	C_0
Phép cộng:	0	N_i	0
Phép trừ:	1	$\overline{N_i}$	1

$C_0 = T$
 $y_i = T \oplus N_i$



III. HỆ CHUYỂN MÃ (CODE CONVERSION):

Hệ chuyển mã là hệ tổ hợp có nhiệm vụ là 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.

Vd: Thiết kế hệ chuyển mã từ mã BCD thành mã BCD quá 3.

A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

W \ CD \ AB	00	01	11	10
00			X	1
01		1	X	1
11		1	X	X
10		1	X	X

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

Y \ CD \ AB	00	01	11	10
00	1	1	X	1
01			X	
11	1	1	X	X
10			X	X

Z \ CD \ AB	00	01	11	10
00	1	1	X	1
01			X	
11			X	X
10	1	1	X	X

$$Z = \bar{D}$$

$$Y = CD + \bar{C}\bar{D}$$

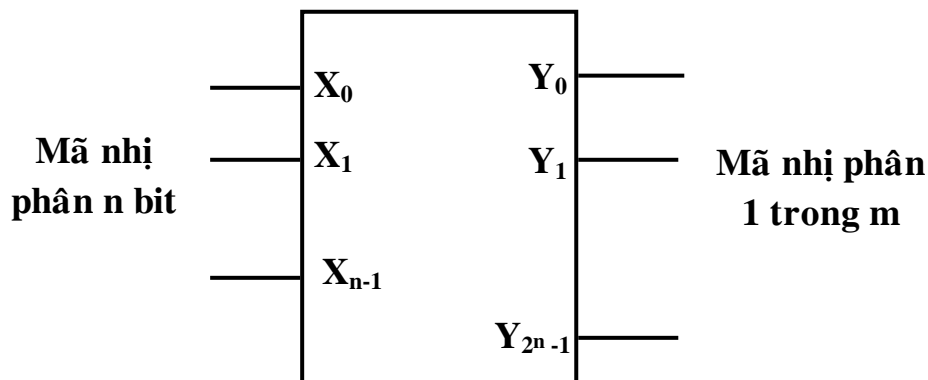
$$X = \bar{B}C + \bar{B}D + B\bar{C}\bar{D}$$

$$W = A + BC + BD$$

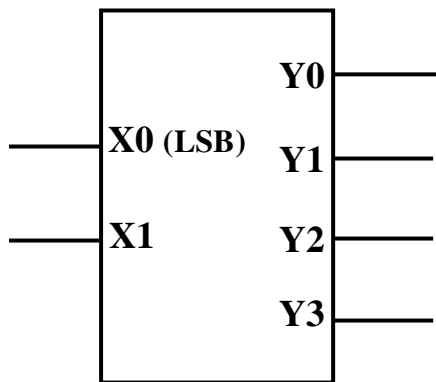
IV. BỘ GIẢI MÃ (DECODER):

1. Giới thiệu:

Bộ giải mã là hệ chuyển mã có nhiệm vụ chuyển từ mã nhị phân thuần túy n bit ở ngõ vào thành mã nhị phân 1 trong m ở ngõ ra ($m = 2^n$). Nghĩa là với giá trị i của tổ hợp nhị phân ở ngõ vào, thì ngõ ra thứ 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).



a. Bộ giải mã ngõ ra tích cực cao: $Y_i = m_i$ ($i = 0, 1, \dots, 2^n - 1$)



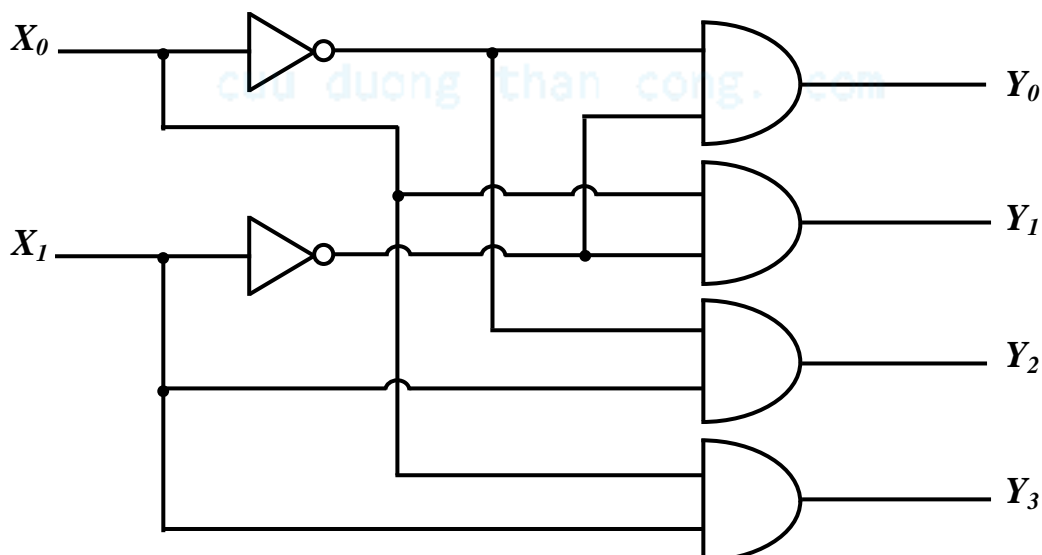
X1	X0	Y3	Y2	Y1	Y0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

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

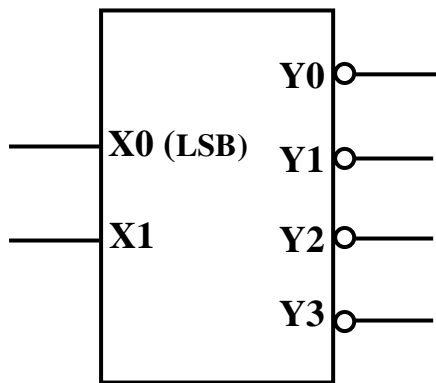
$$Y_2 = X_1 \overline{X_0} = m_2$$

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

$$Y_3 = X_1 X_0 = m_3$$



b. Bộ giải mã ngõ ra tích cực thấp: $Y_i = M_i$



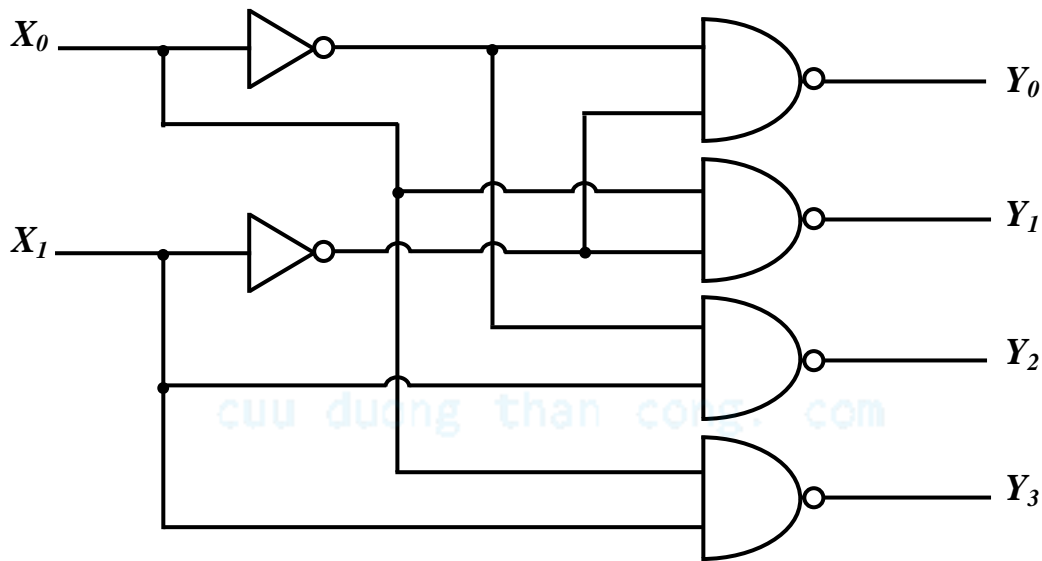
X1	X0	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	0
1	1	0	1	1	1

$$Y_0 = X_1 + X_0 = M_0$$

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

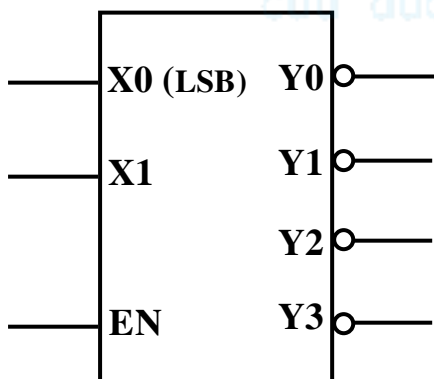
$$Y_1 = X_1 + \overline{X_0} = M_1$$

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



2. Bộ giải mã có ngõ vào cho phép:

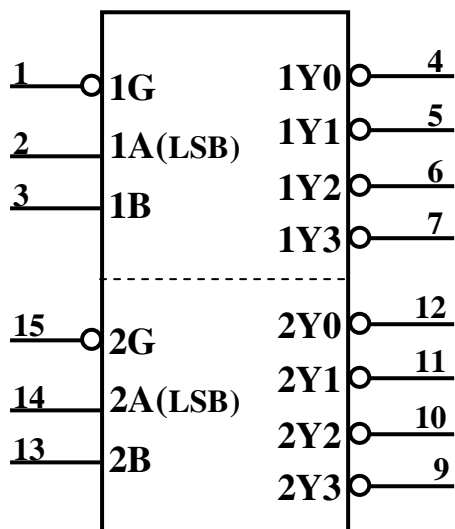
Ngoài các ngõ vào dữ liệu, bộ giải mã có thể có 1 hay nhiều ngõ vào cho phép. Muốn mạch giải mã hoạt động, các ngõ vào cho phép phải ở trạng thái tích cực. Ngược lại, mạch giải mã sẽ không hoạt động được; khi đó các ngõ ra đều ở trạng thái không tích cực.



EN	X1	X0	$\overline{Y_3}$	$\overline{Y_2}$	$\overline{Y_1}$	$\overline{Y_0}$
0	x	x	1	1	1	1
1	0	0	1	1	1	0
1	0	1	1	1	0	1
1	1	0	1	0	1	0
1	1	1	0	1	1	1

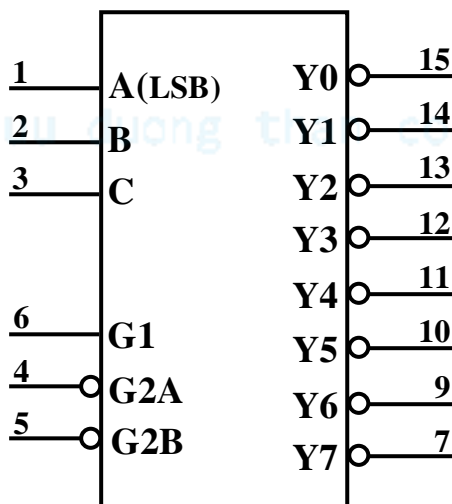
3. IC giải mã:

a. IC 74LS139: là vi mạch có 2 bộ giải mã 2 sang 4 ngõ ra tích cực thấp



\overline{G}	B	A	$\overline{Y3}$	$\overline{Y2}$	$\overline{Y1}$	$\overline{Y0}$
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	0
0	1	1	0	1	1	1
1	x	x	1	1	1	1

b. IC 74LS138: là vi mạch giải mã 3 sang 8 ngõ ra tích cực thấp



G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	$\overline{Y7}$	$\overline{Y6}$	$\overline{Y5}$	$\overline{Y4}$	$\overline{Y3}$	$\overline{Y2}$	$\overline{Y1}$	$\overline{Y0}$
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	0	1	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

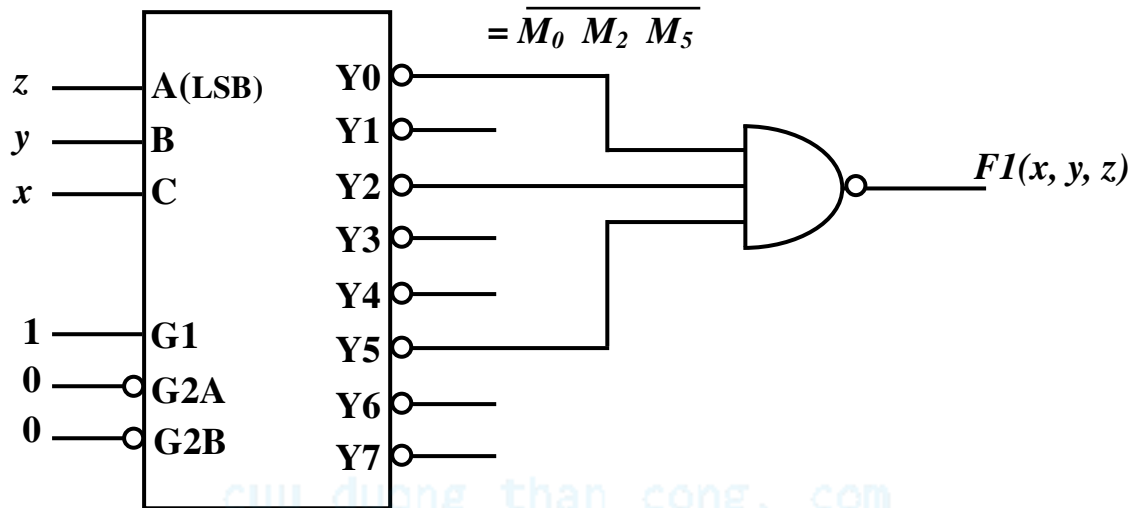
4. Sử dụng bộ giải mã thực hiện hàm Boole:

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 trực tiếp hàm Boole có dạng chính tắc.

Vd: Cho hàm $F1(x, y, z) = \sum(0, 2, 5)$

$$= m_0 + m_2 + m_5 = \overline{M}_0 + \overline{M}_2 + \overline{M}_5$$

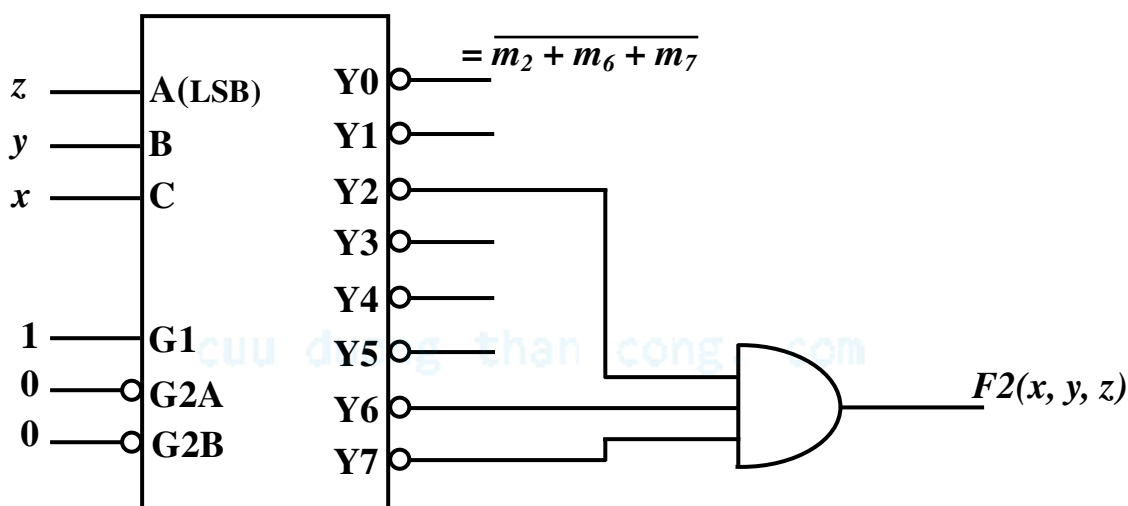
$$= \overline{M_0 M_2 M_5}$$



Vd: Cho hàm $F2(x, y, z) = \prod(2, 6, 7)$

$$= M_2 M_6 M_7 = \overline{m}_2 \overline{m}_6 \overline{m}_7$$

$$= \overline{m_2 + m_6 + m_7}$$

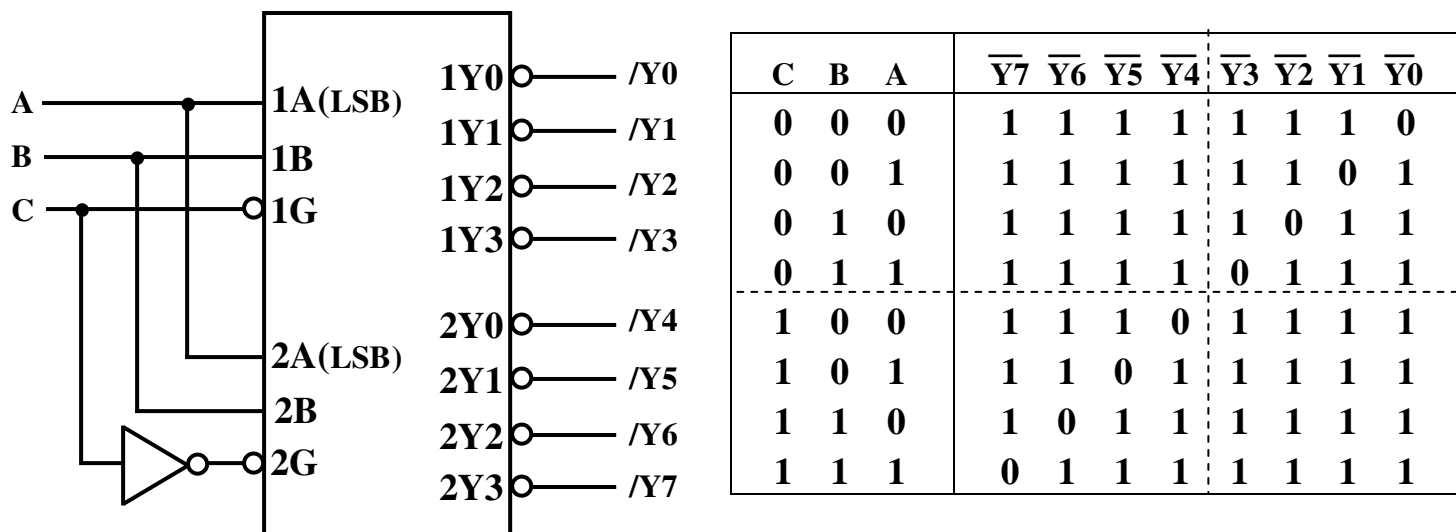


- Ta có thể thực hiện nhiều hàm trên 1 bộ giải mã.

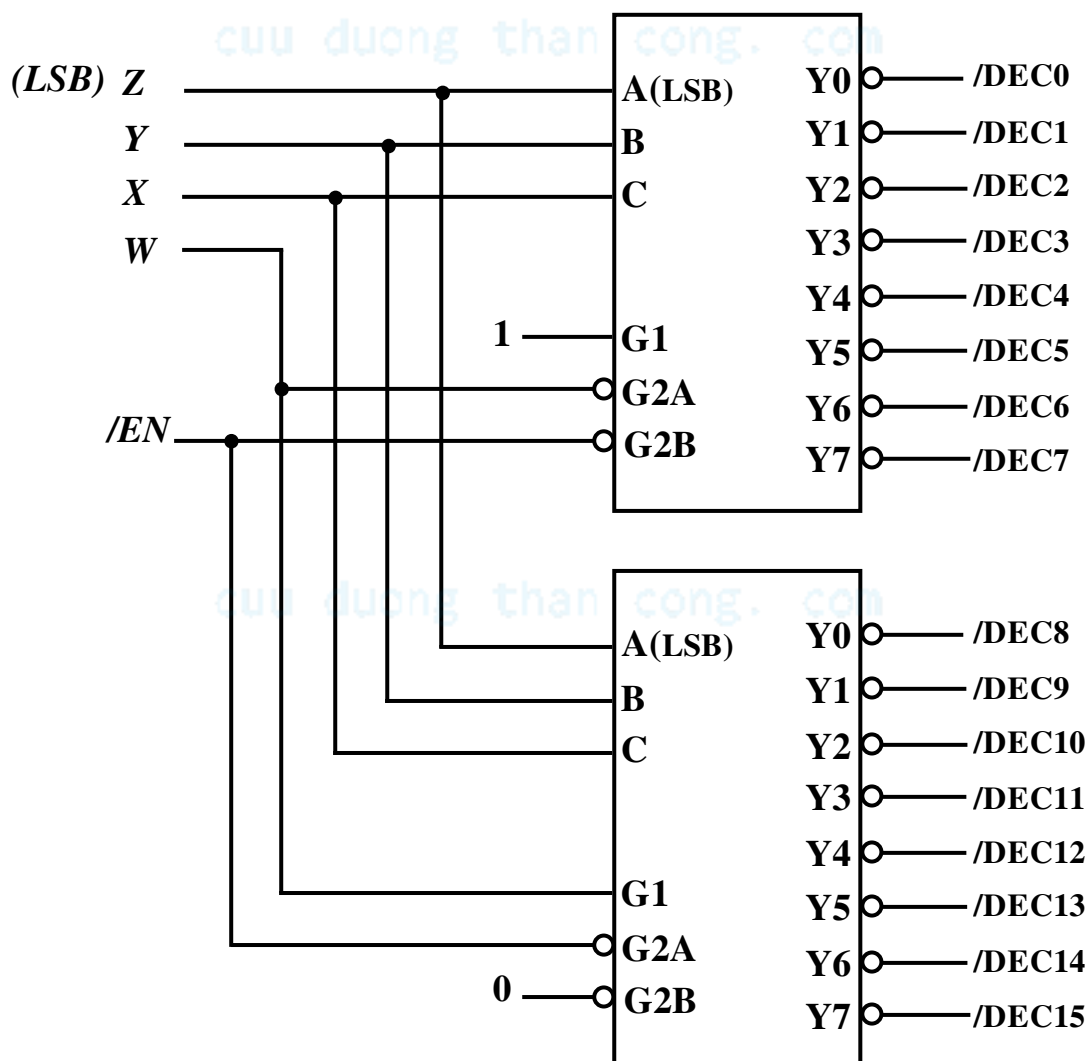
- Decoder ngõ ra tích cực cao ta thực hiện hàm F bằng cổng OR (chính tắc 1) hoặc NOR (chính tắc 2)

5. Ghép bộ giải mã:

Sử dụng ngõ vào cho phép để ghép bộ giải mã



Ghép 2 IC 74138 ta có được decoder 4 sang 16, có ngõ vào cho phép

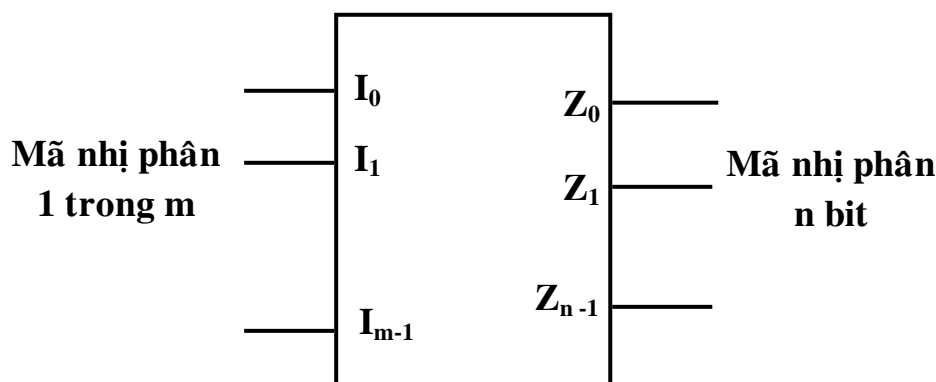


V. BỘ MÃ HÓA (ENCODER):

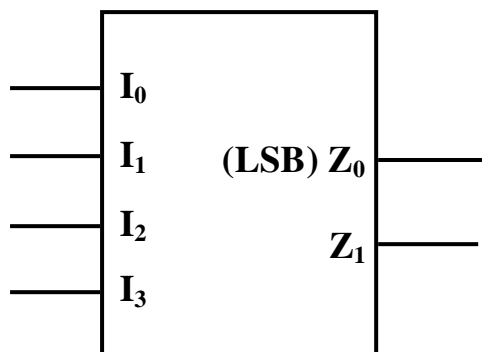
1. Giới thiệu:

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 thuần túy (với $m \leq 2^n$).

Với ngõ vào thứ i được tích cực thì ngõ ra chính là tổ hợp giá trị nhị phân i tương ứng.



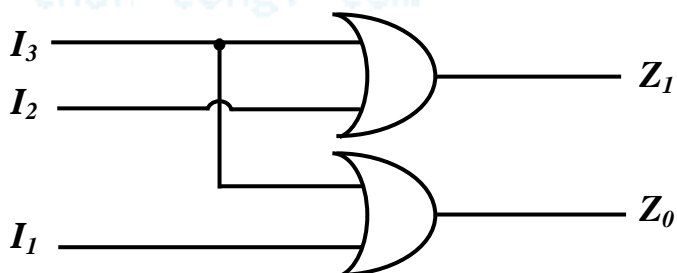
Vd: Encoder 4 sang 2



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

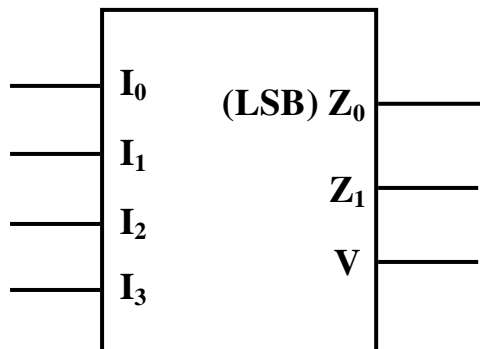
$$Z_1 = I_2 + I_3$$

$$Z_0 = I_1 + I_3$$



2. Bộ mã hóa có ưu tiên (Priority Encoder):

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.



I0	I1	I2	I3	Z1	Z0	V
0	0	0	0	x	x	0
1	0	0	0	0	0	1
x	1	0	0	0	1	1
x	x	1	0	1	0	1
x	x	x	1	1	1	1

Thứ tự ưu tiên $I3 > I2 > I1 > I0$.

Ngõ ra V có chức năng chỉ thị: $V = 1$ nếu có ít nhất 1 ngõ vào tích cực; ngược lại $V = 0$ nếu không có ngõ vào nào tích cực.

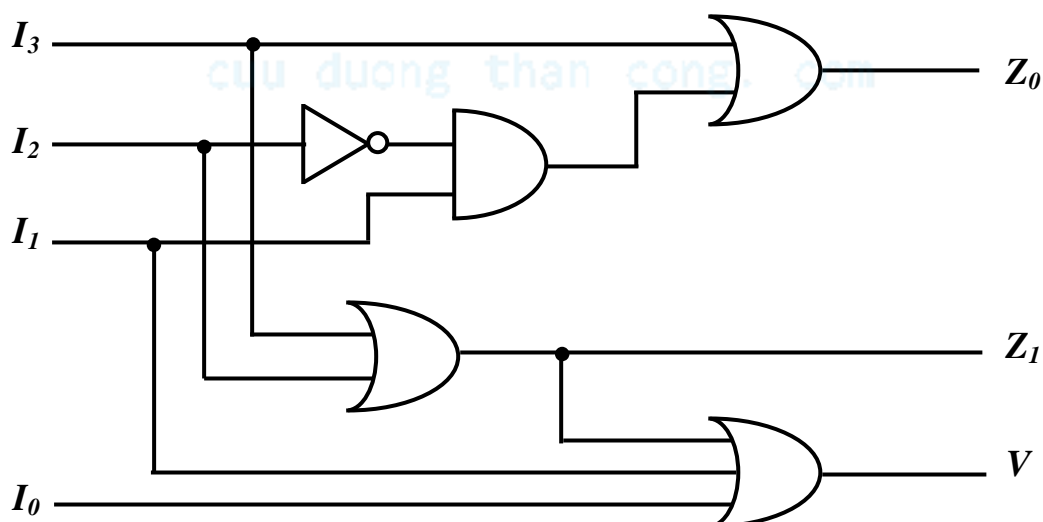
$$V = I0 + I1 + I2 + I3$$

I2I3 \ I0I1	00	01	11	10
	Z1	Z0	V	
00	X			
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$$Z1 = I2 + I3$$

I2I3 \ I0I1	00	01	11	10
	Z1	Z0	V	
00	X	1	1	
01	1	1	1	1
11	1	1	1	1
10				

$$Z0 = I3 + I1 \bar{I2}$$



3. IC mã hóa ưu tiên 8 → 3 (74LS148):

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

- Thứ tự ưu tiên: $I_7 > I_6 > I_5 > I_4 > I_3 > I_2 > I_1 > I_0$.

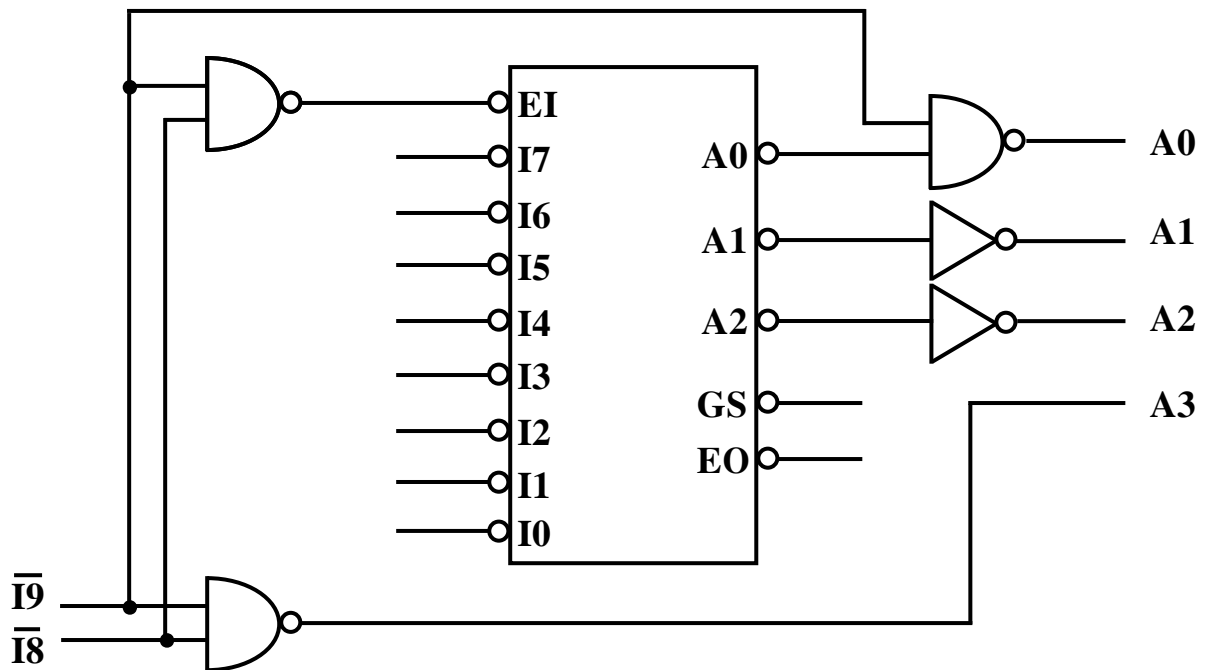
- EI (Enable Input): ngõ vào cho phép, tích cực thấp. Nếu không tích cực các ngõ ra đều bằng 0 (không tích cực).

- EO (Enable Output): ngõ ra tích cực thấp. Ngõ ra này chỉ tích cực khi ngõ vào EI tích cực nhưng không có ngõ vào I_i nào tích cực. Thường được nối vào EI của 1 Encoder 74148 khác có ưu tiên thấp hơn.

- GS (Group Select): ngõ ra tích cực thấp. Ngõ ra này tích cực khi ngõ vào EI tích cực và có ít nhất 1 ngõ vào I_i tích cực.

- A2, A1, A0: các ngõ ra tổ hợp nhị phân tương ứng với ngõ vào I_i tích cực khi EI tích cực. Tuy nhiên giá trị này bị lấy bù.

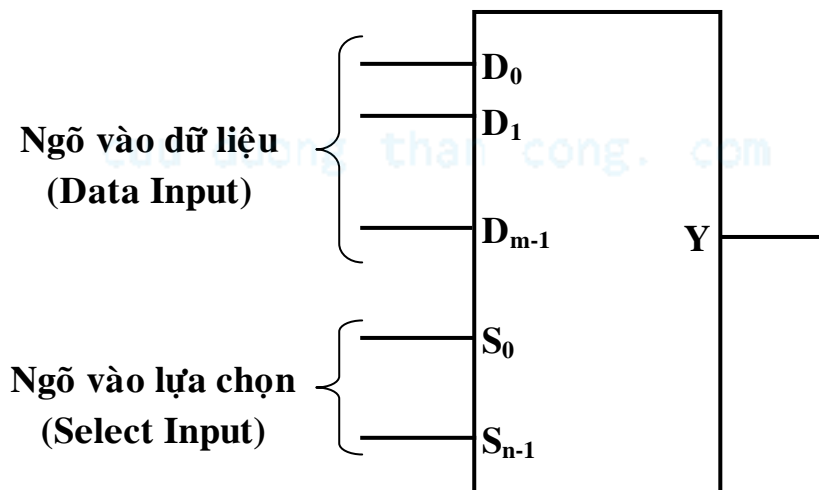
* Mạch ứng dụng mở rộng Encoder 8 → 3 thành Encoder 10 → 4:



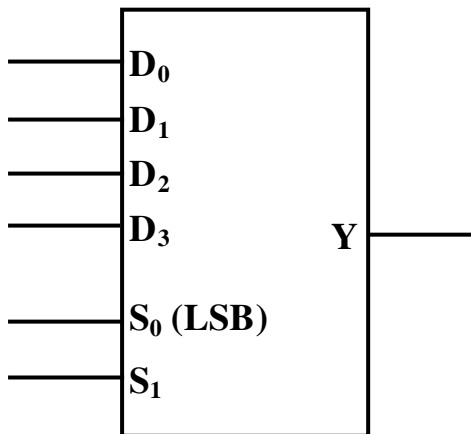
VI. BỘ ĐỒNG KÊNH (MULTIPLEXER - MUX):

1. Giới thiệu:

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 thứ i sẽ được chọn đưa đến ngõ ra. ($m = 2^n$)

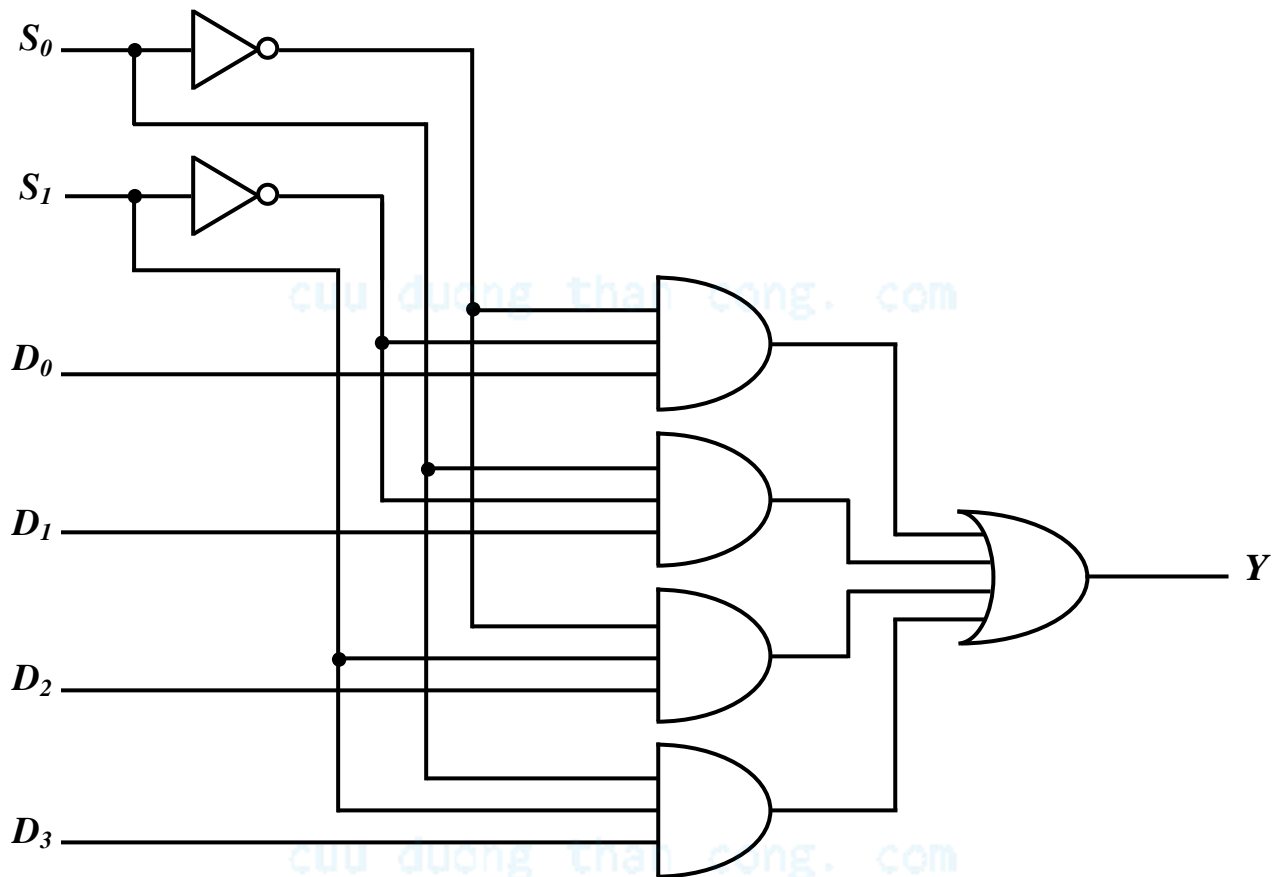


Thiết kế bộ MUX 4 → 1



S1	S0	Y
0	0	D0
0	1	D1
1	0	D2
1	1	D3

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



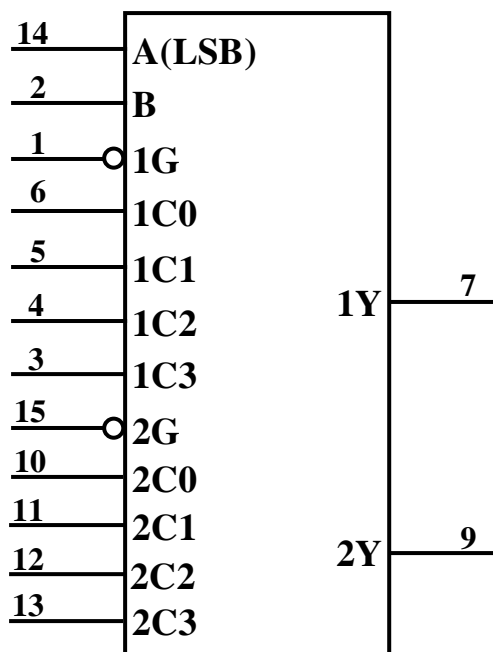
Tổng quát: $Y = \sum m_i D_i$ (với $i = 0, 1, \dots, 2^n - 1$)

m_i là minterm thứ i của n ngõ vào lựa chọn

D_i là các ngõ vào dữ liệu

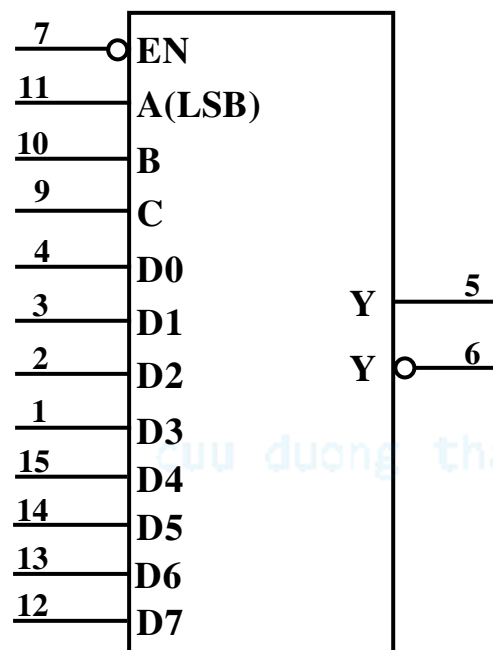
2. IC dồn kênh:

a. **74LS153**: gồm 2 bộ MUX 4 → 1



\overline{G}	B	A	Y
1	x	x	0
0	0	0	C0
0	0	1	C1
0	1	0	C2
0	1	1	C3

b. **74LS151**: bộ MUX 8 → 1



\overline{EN}	C	B	A	Y
1	x	x	x	0
0	0	0	0	D0
0	0	0	1	D1
0	0	1	0	D2
0	0	1	1	D3
0	1	0	0	D4
0	1	0	1	D5
0	1	1	0	D6
0	1	1	1	D7

2. Sử dụng bộ MUX thực hiện hàm Boole:

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

Cho hàm Boole:

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

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

$$= m_0.1 + m_1.1 + m_2.0 + m_3.0 + m_4.1 + m_5.0 + m_6.0 + m_7.1 \quad (1)$$

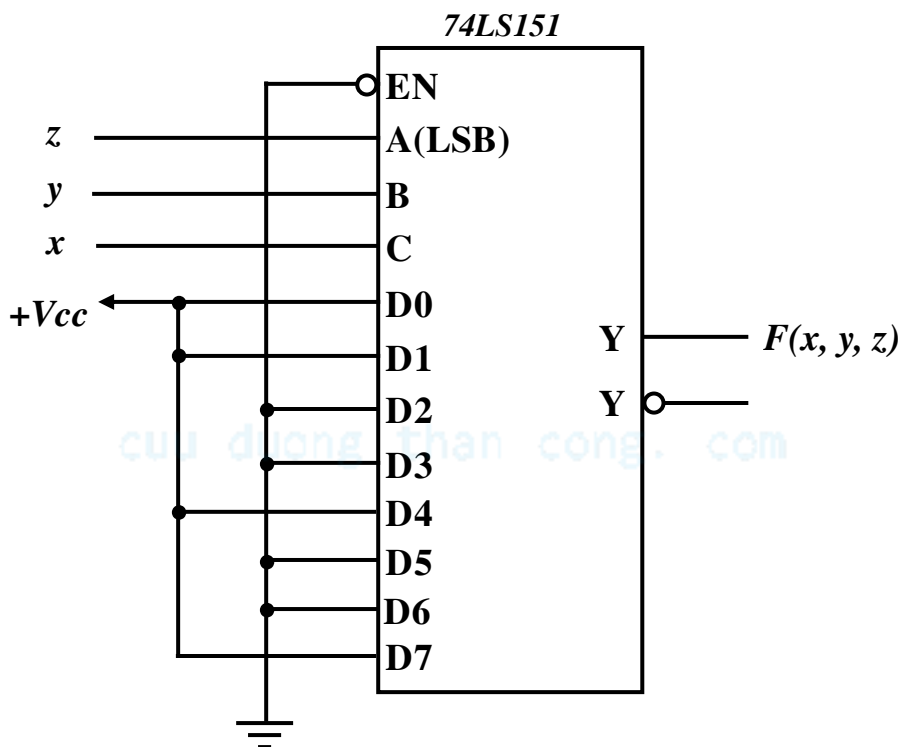
Ngõ ra của bộ MUX $8 \rightarrow 1$ có dạng:

$$Y = \sum m_i D_i$$

$$= m_0 D_0 + m_1 D_1 + m_2 D_2 + m_3 D_3 + m_4 D_4 + m_5 D_5 + m_6 D_6 + m_7 D_7 \quad (2)$$

Để đồng nhất (1) và (2) ta có:

- Đưa các biến x, y, z vào ngõ vào lựa chọn C, B, A (đúng theo trọng số)
- Cho các ngõ vào $D_0 = D_1 = D_4 = D_7 = 1$ và $D_2 = D_3 = D_5 = D_6 = 0$



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

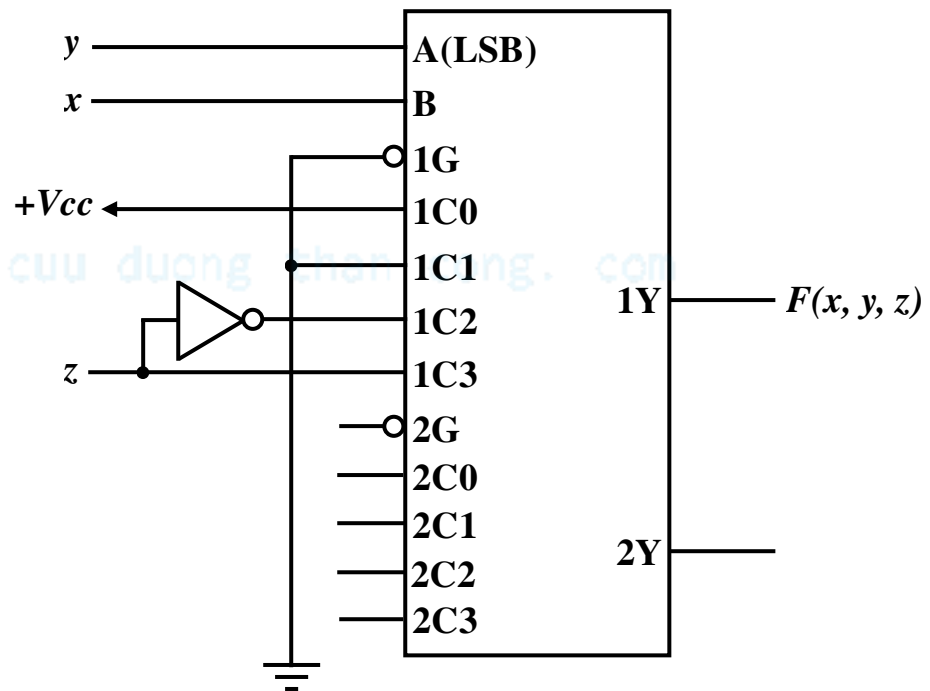
$$\begin{aligned}
 F(x, y, z) &= \sum(0, 1, 4, 7) = \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z + x\bar{y}\bar{z} + xyz \\
 &= \bar{x}\bar{y}.1 + \bar{x}y.0 + x\bar{y}.\bar{z} + x y.z \\
 &= m_0.1 + m_1.0 + m_2.\bar{z} + m_3.z
 \end{aligned} \quad (1)$$

Ngõ ra của bộ MUX $4 \rightarrow 1$ có dạng:

$$Y = m_0 D_0 + m_1 D_1 + m_2 D_2 + m_3 D_3 \quad (2)$$

Để đồng nhất (1) và (2) ta có:

- Đưa các biến x, y vào ngõ vào lựa chọn B, A (đúng theo trọng số)
- Cho các ngõ vào $D_0 = 1, D_1 = 0, D_2 = \bar{z}, D_3 = z$



Cách khác:

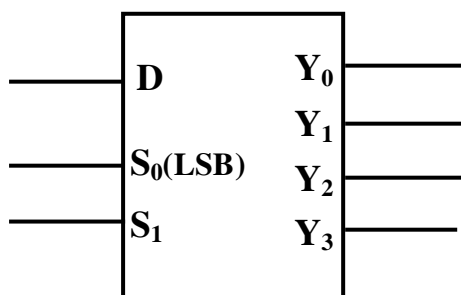
	x	y	z	F	
m_0	0	0	0	1	$D_0 = 1$
	0	0	1	1	
m_1	0	1	0	0	$D_1 = 0$
	0	1	1	0	
m_2	1	0	0	1	$D_2 = \bar{z}$
	1	0	1	0	
m_3	1	1	0	0	$D_3 = z$
	1	1	1	1	

VII. BỘ PHÂN KÊNH (DEMUX):

1. Giới thiệu:

Bộ DEMUX 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 sẽ được chọn đưa đến ngõ ra thứ i.

Thiết kế bộ DEMUX 1 → 4:



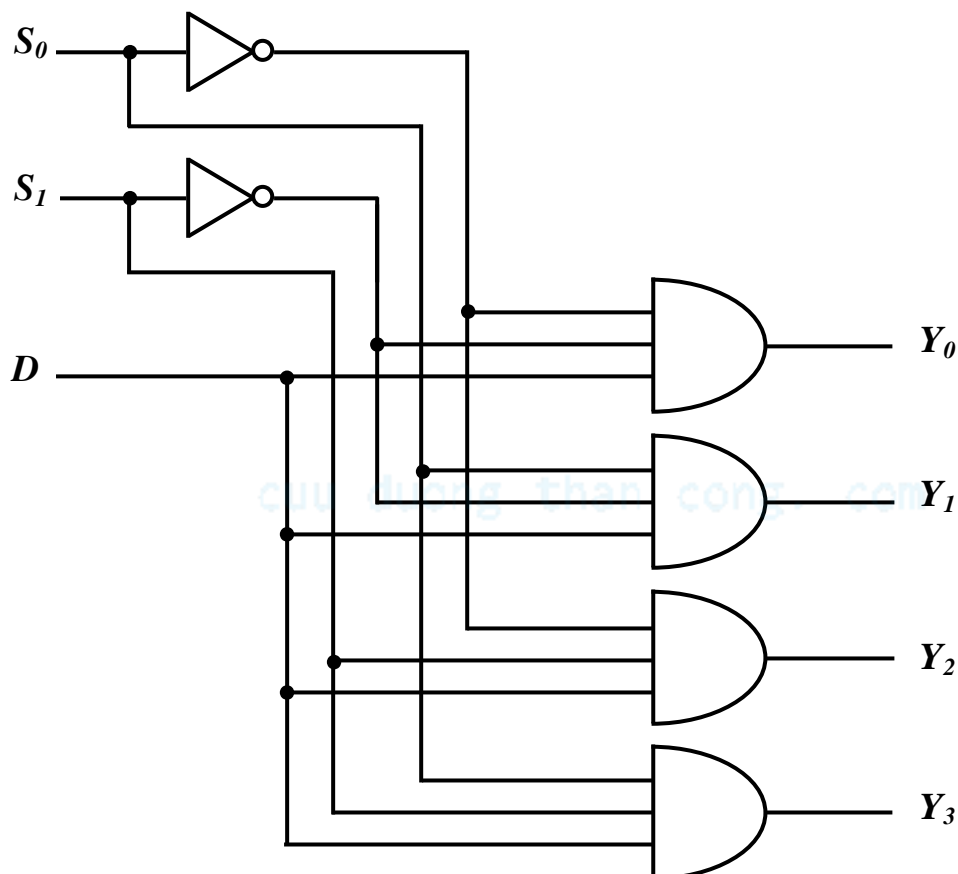
S1	S0	Y0	Y1	Y2	Y3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

$$Y_0 = \bar{S}_1 \bar{S}_0 D$$

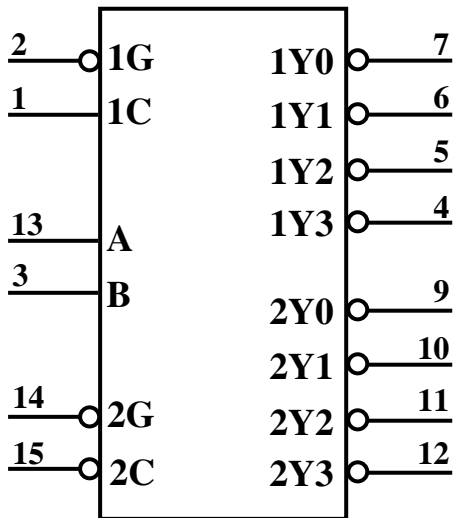
$$Y_2 = S_1 \bar{S}_0 D$$

$$Y_1 = \bar{S}_1 S_0 D$$

$$Y_3 = S_1 S_0 D$$



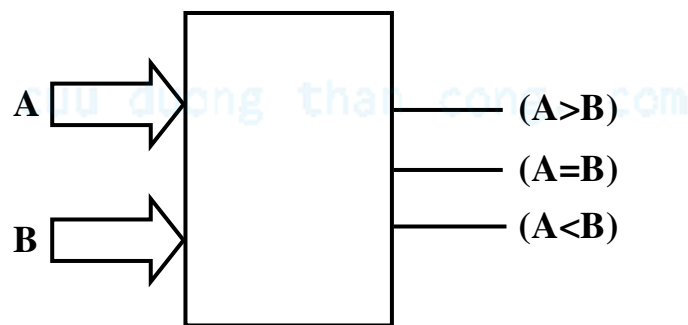
2. IC phân kênh 74LS155: gồm 2 bộ phân kênh 1 → 4



B	A	$\overline{1G}$	1C	$\overline{1Y_0}$	$\overline{1Y_1}$	$\overline{1Y_2}$	$\overline{1Y_3}$	$\overline{2G}$	$\overline{2C}$	$\overline{2Y_0}$	$\overline{2Y_1}$	$\overline{2Y_2}$	$\overline{2Y_3}$
x	x	1	x	1	1	1	1	1	x	1	1	1	1
x	x	x	0	1	1	1	1	x	1	1	1	1	1
0	0	0	1	0	1	1	1	0	0	0	1	1	1
0	1	0	1	1	0	1	1	0	0	1	0	1	1
1	0	0	1	1	1	0	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0	0	0	1	1	1	0

VIII. BỘ SO SÁNH ĐỘ LỚN (COMPARATOR):

1. Giới thiệu:



Bộ so sánh là hệ tổ hợp có nhiệm vụ so sánh 2 số A và B (mỗi số n bit).

Mạch có 3 ngõ ra (A>B), (A=B) và (A<B) chỉ thị cho độ tương đối của chúng.

Thiết kế bộ so sánh:

A và B có chiều dài bit là n, nên mạch so sánh có 2n ngõ vào; do vậy ta không thể thiết kế theo kiểu thông thường là lập bảng chân trị. Ta có cách khác:

$$\text{Vd: } n = 4 \quad A = A_3 A_2 A_1 A_0$$

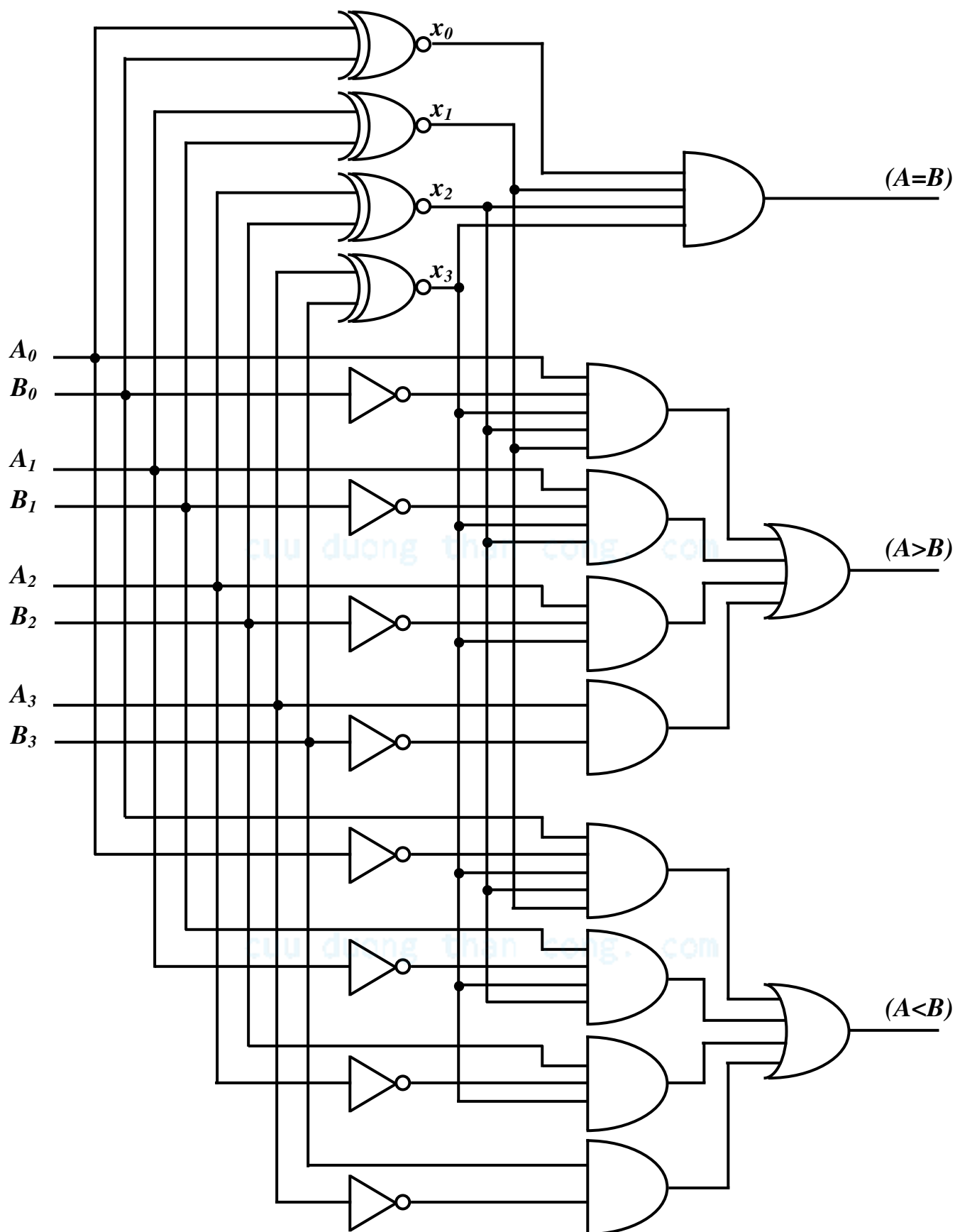
$$B = B_3 B_2 B_1 B_0$$

Ta đặt biến trung gian $x_i = \overline{A_i \oplus B_i}$ ($i = 0, 1, 2, 3$)

$$(A = B) = x_3 x_2 x_1 x_0$$

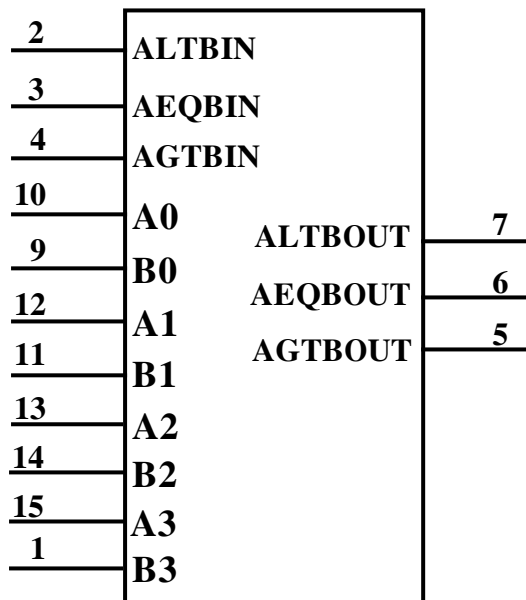
$$(A > B) = A_3 \overline{B_3} + x_3 A_2 \overline{B_2} + x_3 x_2 A_1 \overline{B_1} + x_3 x_2 x_2 A_0 \overline{B_0}$$

$$(A < B) = \overline{A_3} B_3 + x_3 \overline{A_2} B_2 + x_3 x_2 \overline{A_1} B_1 + x_3 x_2 x_1 \overline{A_0} B_0$$



Ta có thể thực hiện $(A < B) = \overline{(A > B) + (A = B)}$

2. IC so sánh 74LS85:



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

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

$$ALTBOU = (A < B) + (A = B).ALTBOU$$

cuu duong than cong. com

cuu duong than cong. com