

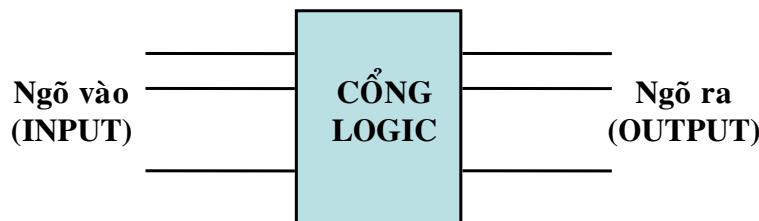
## Chương 3: HỆ TỔ HỢP

### I. Giới thiệu – Cách thiết kế hệ tổ hợp:

Mạch logic được chia làm 2 loại:

- Hệ tổ hợp (Combinational Circuit)
- Hệ tuần tự (Sequential Circuit).

Hệ 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.



1

### \* 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.

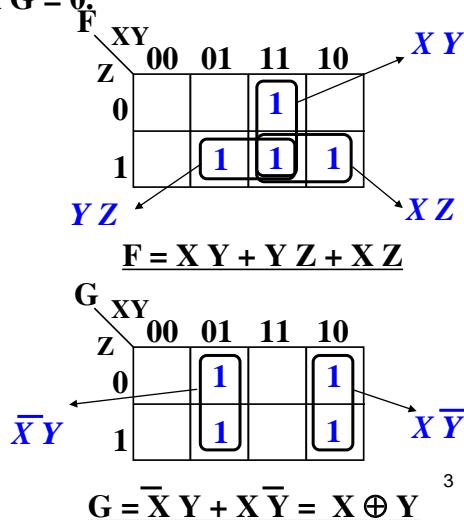
2

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.

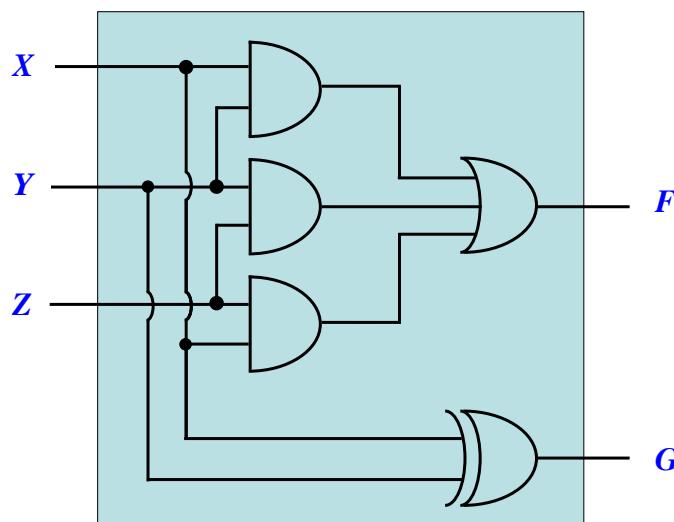
- Ngõ ra G là 1 nếu như giá trị nhị phân của 3 ngõ vào lớn hơn 1 và nhỏ hơn 6; ngược lại G = 0.

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



$$F = XY + YZ + XZ$$

$$G = \overline{X}Y + X\overline{Y} = X \oplus Y$$



4

*Trường hợp 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.

$$F_2 = A + BCD + \bar{B}C\bar{D}$$

$$F_1 = AD + B\bar{C}D + BC\bar{D}$$

$$F_0 = A\bar{D} + B\bar{D} + \bar{A}\bar{B}\bar{C}D$$

A	B	C	D	$F_2$	$F_1$	$F_0$
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

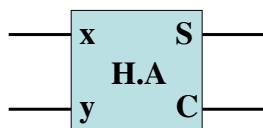
5

## II. Bộ cộng - trừ nhị phân:

### 1. Bộ cộng (Adder):

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

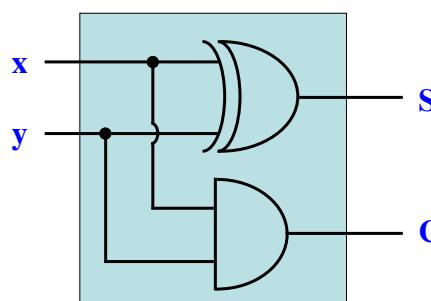
Bộ cộng bán phần là hệ tổ hợp có nhiệm vụ thực hiện phép cộng số học  $x + y$  ( $x, y$  là 2 bit nhị phân ngõ vào); hệ có 2 ngõ ra: bit tổng  $S$  (Sum) và bit nhớ  $C$  (Carry).



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

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

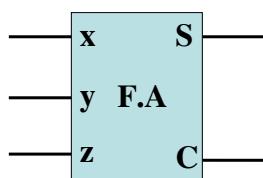
$$C = xy$$



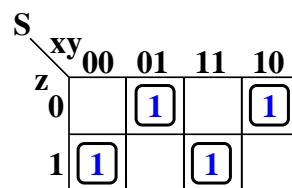
6

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

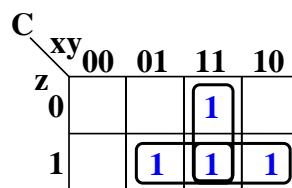
Bộ cộng toàn phần thực hiện phép cộng số học 3 bit  $x + y + z$   
( $z$  biểu diễn cho bit nhớ từ vị trí có trọng số nhỏ hơn gởi tới)



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



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



$$C = xy + xz + yz$$

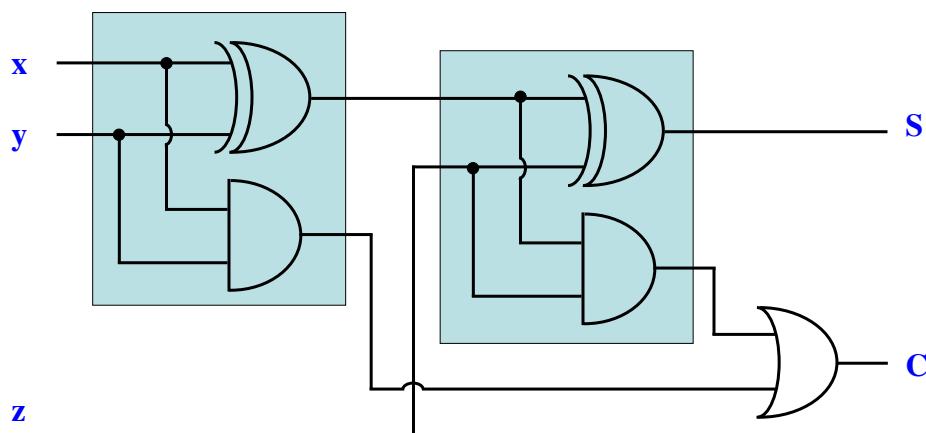
7

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

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

$$\begin{aligned} C &= xy + xz + yz \\ &= xy + x\bar{y}z + x y z + \bar{x} y z \\ &= xy(I + z) + z(\bar{x}y + x\bar{y}) \end{aligned}$$

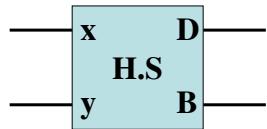
$$C = xy + z(x \oplus y)$$



8

**2. Bộ trừ (Subtractor):****a. Bộ trừ bán phần (Half Subtractor – H.S):**

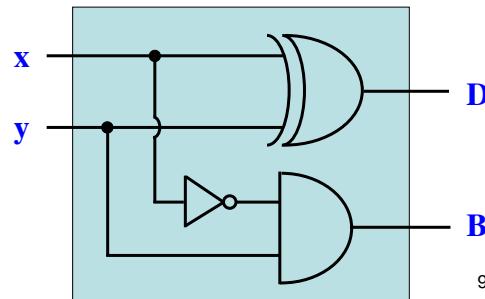
Bộ trừ bán phần có nhiệm vụ thực hiện phép trừ số học  $x - y$  ( $x, y$  là 2 bit nhị phân ngõ vào); hệ có 2 ngõ ra: bit hiệu  $D$  (Difference) và bit mượn  $B$  (Borrow).



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

$$B = \bar{x}y$$

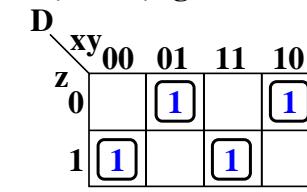
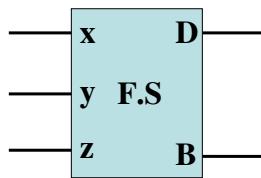
x	y	B	D
0	0	0	0
0	1	1	1
1	0	0	1
1	1	0	0



9

**b. Bộ trừ toàn phần (Full Subtractor – F.S):**

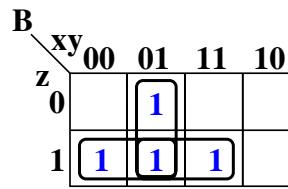
Bộ trừ toàn phần thực hiện phép trừ số học 3 bit  $x - y - z$  ( $z$  biểu diễn cho bit mượn từ ví trí có trọng số nhỏ hơn)



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

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

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



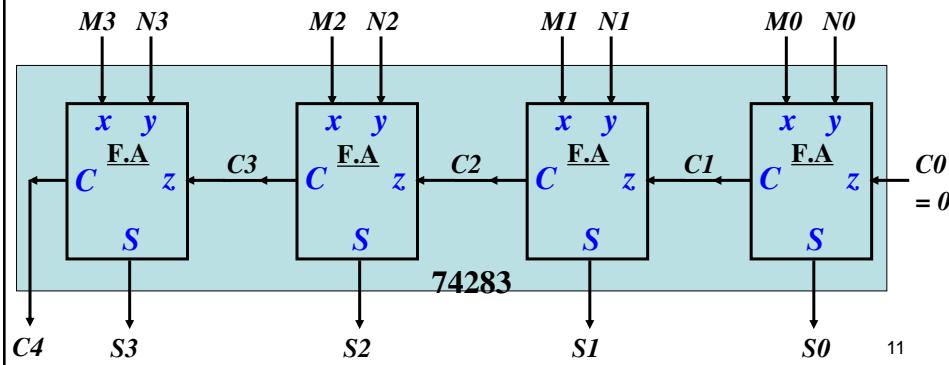
$$C = \bar{x}y + \bar{x}z + yz$$

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

10

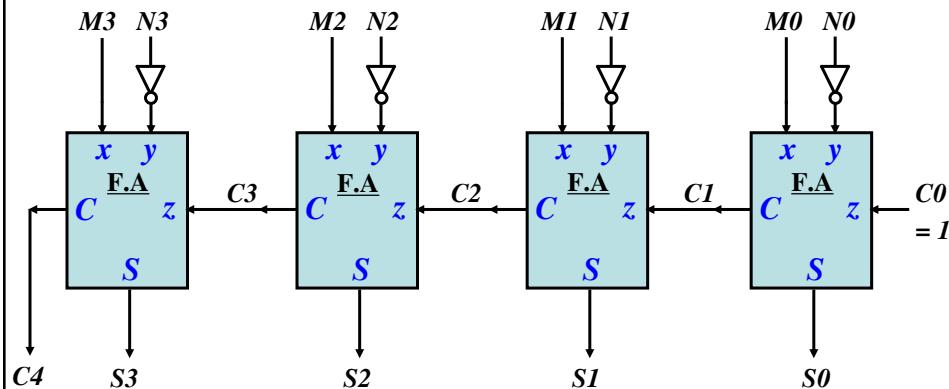
**3. Bộ cộng/trừ nhị phân song song:****a. Bộ cộng nhị phân:**

$$\begin{array}{r}
 & C_3 & C_2 & C_1 \\
 M: & M_3 & M_2 & M_1 & M_0 \\
 + N: & N_3 & N_2 & N_1 & N_0 \\
 \hline
 C_4 & S_3 & S_2 & S_1 & S_0
 \end{array}$$

**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

12

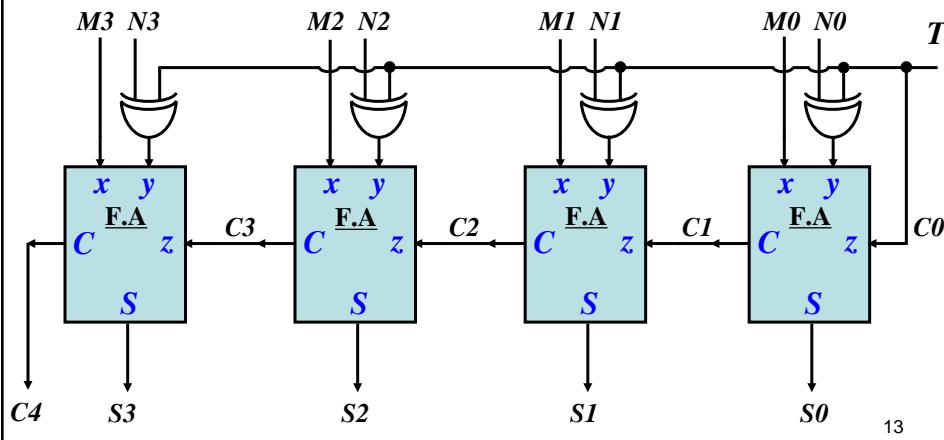
**c. Bộ cộng/trừ nhị phân:**

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

Ngõ vào điều khiển

$$\begin{aligned} T = 0: & \text{ Cộng} \\ T = 1: & \text{ Trừ} \end{aligned}$$

$$\begin{aligned} C_0 &= T \\ y_i &= T \oplus N_i \end{aligned}$$

**III. Hệ chuyển mã (Code Conversion):**

- 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.

14

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

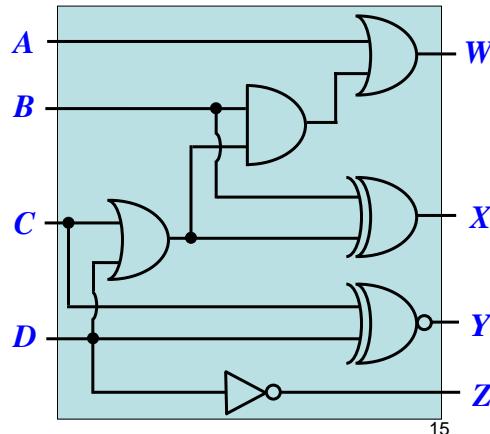
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	X	X	X	X
1	0	1	1	X	X	X	X
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

$$W = A + B(C + D)$$

$$X = B \oplus (C + D)$$

$$Y = \overline{C} \oplus D$$

$$Z = \overline{D}$$

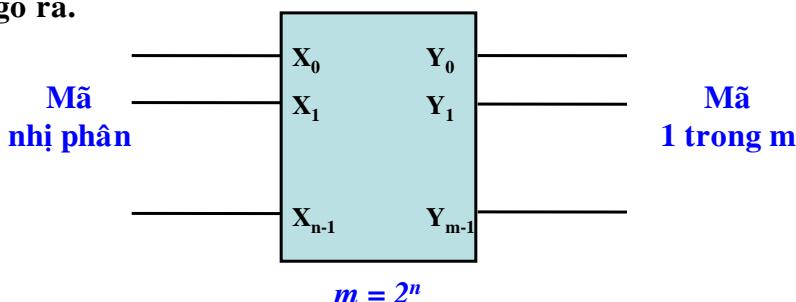


15

#### IV. Bộ giải mã (DECODER):

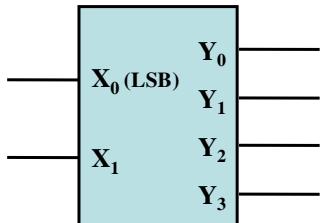
##### 1. Giới thiệu:

- 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).

**a. Bộ giải mã ngõ ra tích cực cao:**

X <sub>1</sub>	X <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

$$Y_0 = \overline{X}_1 \overline{X}_0 = m_0$$

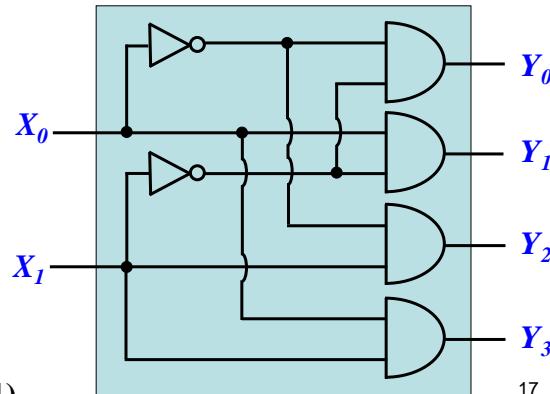
$$Y_1 = \overline{X}_1 X_0 = m_1$$

$$Y_2 = X_1 \overline{X}_0 = m_2$$

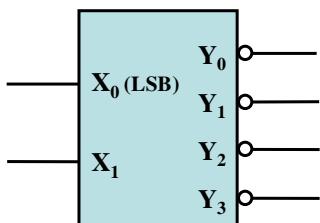
$$Y_3 = X_1 X_0 = m_3$$

Ngõ ra:  $Y_i = m_i$

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



17

**b. Bộ giải mã ngõ ra tích cực thấp:**

X <sub>1</sub>	X <sub>0</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

$$Y_0 = X_1 + X_0 = M_0 = \overline{m}_0$$

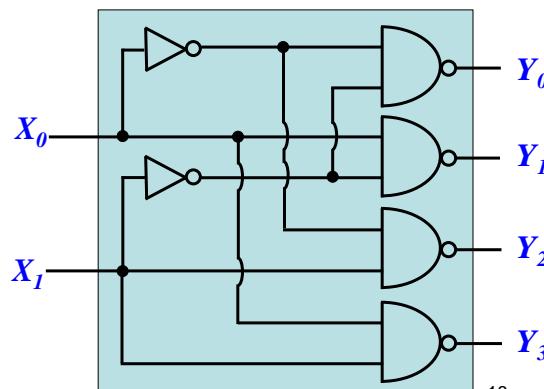
$$Y_1 = X_1 + \overline{X}_0 = M_1 = \overline{m}_1$$

$$Y_2 = \overline{X}_1 + X_0 = M_2 = \overline{m}_2$$

$$Y_3 = \overline{X}_1 + \overline{X}_0 = M_3 = \overline{m}_3$$

Ngõ ra:  $Y_i = M_i$

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

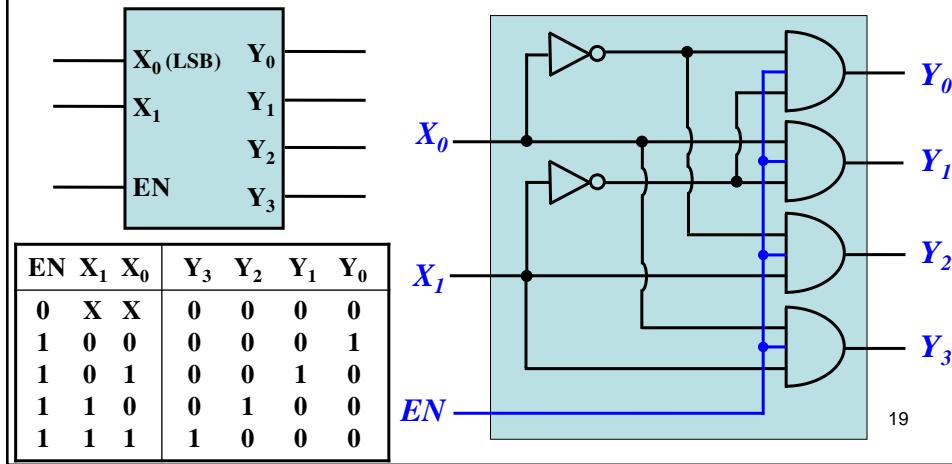


18

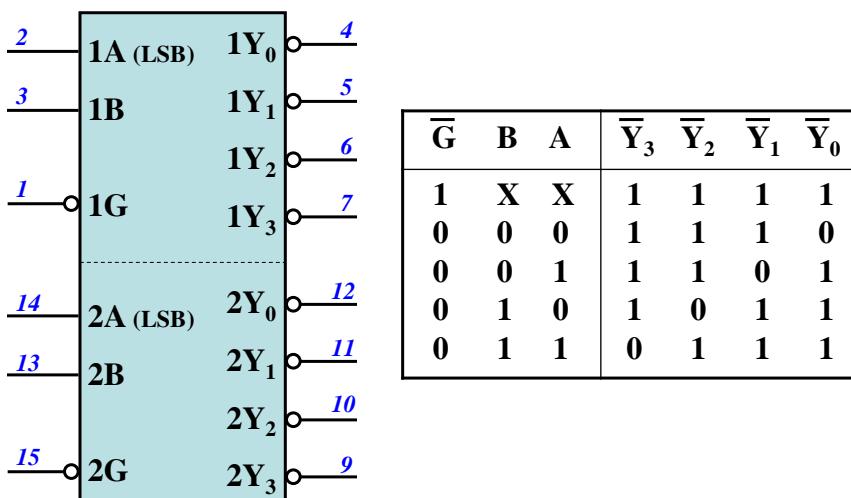
**c. Bộ giải mã có ngõ vào cho phép:**

- Ngoài các ngõ vào dữ liệu, bộ giải mã có thể có 1 hay nhiều ngõ vào cho phép.

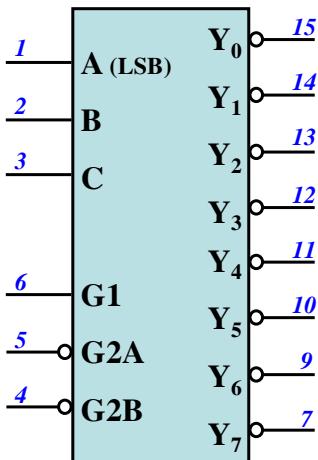
- Khi các ngõ vào cho phép ở trạng thái tích cực thì mạch giải mã mới được hoạt động. Ngược lại, mạch giải mã sẽ không hoạt động; khi đó các ngõ ra đều ở trạng thái không tích cực.



19

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

20

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

$G_1$	$\bar{G}_{2A}$	$\bar{G}_{2B}$	C	B	A	$\bar{Y}_7$	$\bar{Y}_6$	$\bar{Y}_5$	$\bar{Y}_4$	$\bar{Y}_3$	$\bar{Y}_2$	$\bar{Y}_1$	$\bar{Y}_0$
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

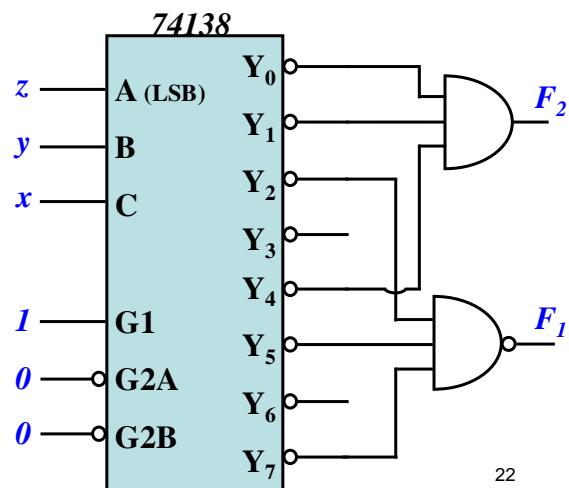
21

**3. Sử dụng bộ giải mã thực hiện hàm Boolean:**

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 Boolean theo dạng chính tắc.

$$\begin{aligned}
 F_1(x, y, z) &= \sum(2, 5, 7) \\
 &= m_2 + m_5 + m_7 \\
 &= \overline{M_2} + \overline{M_5} + \overline{M_7} \\
 &= \overline{M_2 M_5 M_7}
 \end{aligned}$$

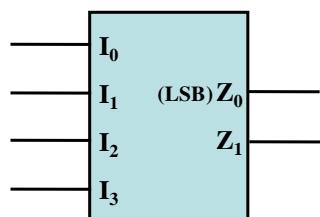
$$\begin{aligned}
 F_2(x, y, z) &= \prod(0, 1, 4) \\
 &= M_0 M_1 M_4
 \end{aligned}$$



22

**V. Bộ mã hóa (ENCODER):****1. Giới thiệu:**

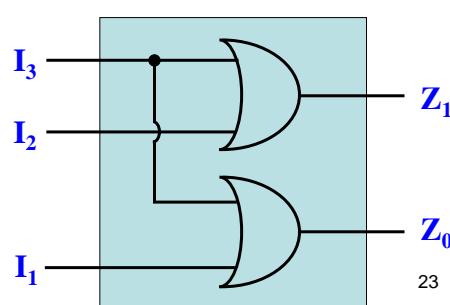
- Encoder là hệ chuyển mã thực hiện hoạt động ngược lại với decoder. Nghĩa là encoder có  $m$  ngõ vào theo mã nhị phân 1 trong  $m$  và  $n$  ngõ ra theo mã nhị phân cơ bản (với  $m \leq 2^n$ ).
- Với ngõ vào  $I_i$  được tích cực thì ngõ ra chính là tổ hợp giá trị nhị phân  $i$  tương ứng.



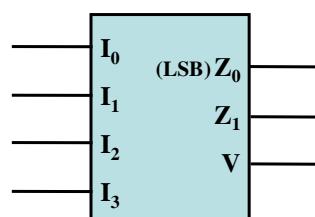
$I_3$	$I_2$	$I_1$	$I_0$	$Z_1$	$Z_0$
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

$$Z_1 = I_3 + I_2$$

$$Z_0 = I_3 + I_1$$

**\* Bộ mã hóa có ưu tiên (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.

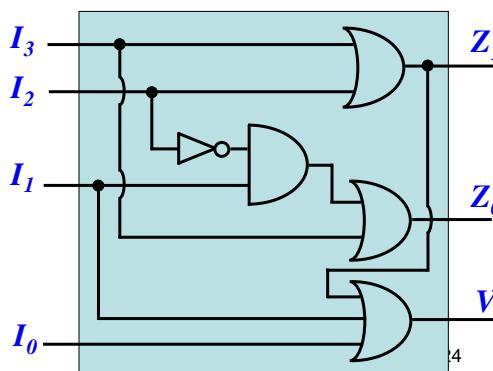


$I_3$	$I_2$	$I_1$	$I_0$	$Z_1$	$Z_0$	$V$
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

$$Z_1 = I_3 + I_2$$

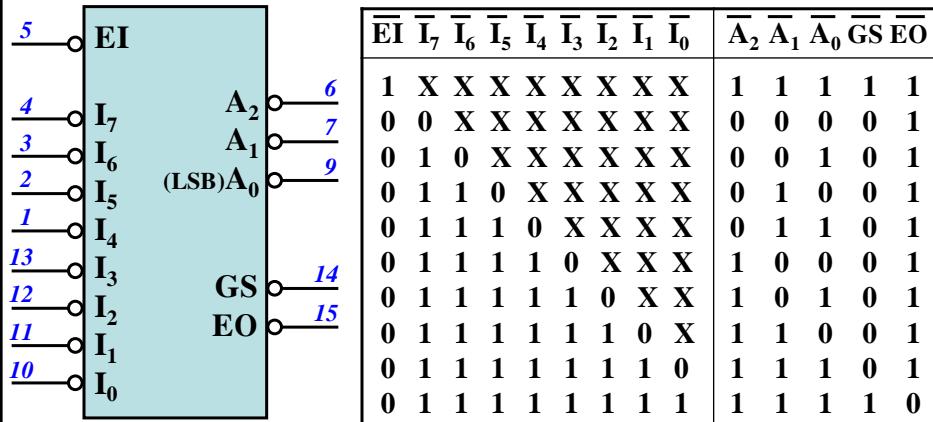
$$Z_0 = I_3 + \overline{I}_2 I_1$$

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



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

## 2. IC mã hóa ưu tiên 8 → 3 (74148):

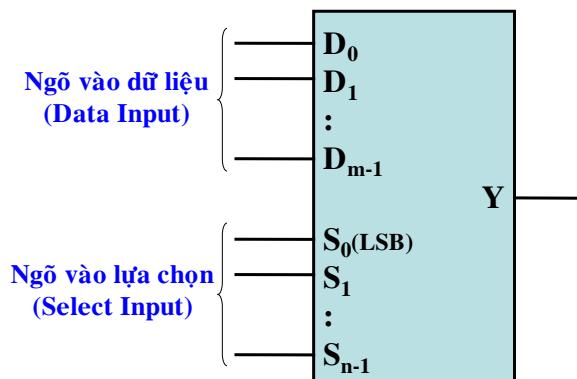


25

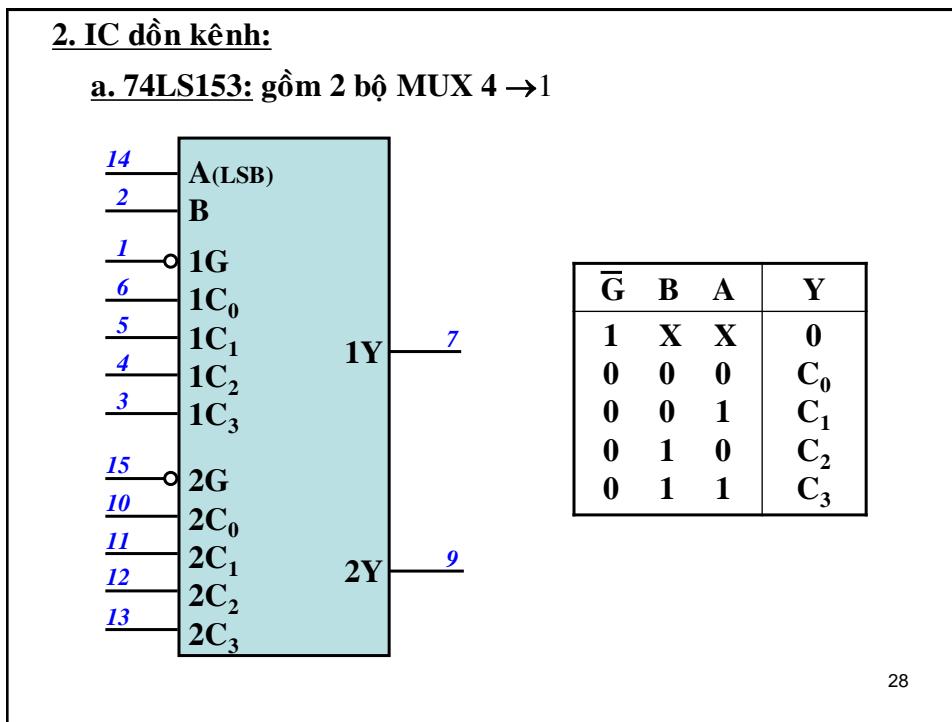
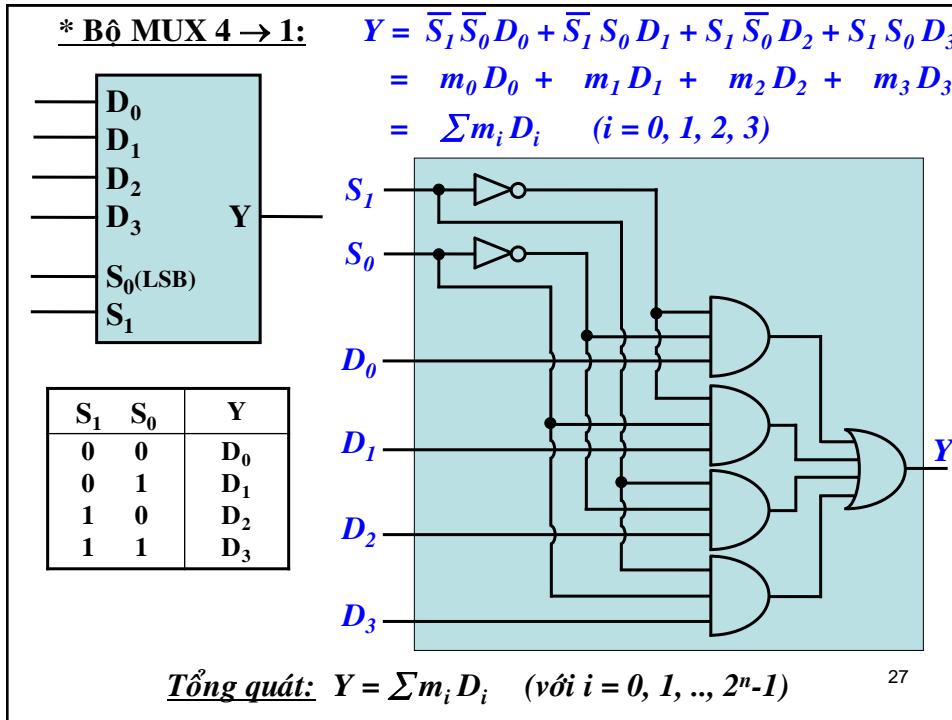
## VI. Bộ dồn kênh (Multiplexer - MUX):

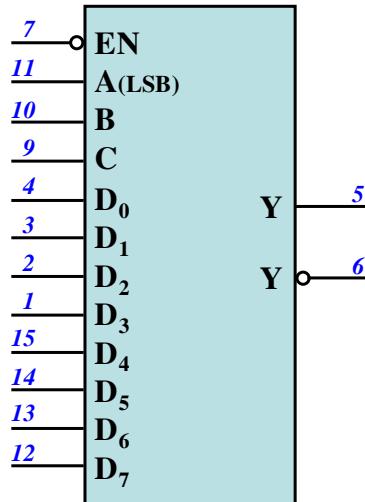
### 1. Giới thiệu:

- 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  $D_i$  sẽ được chọn đưa đến ngõ ra. ( $m = 2^n$ ) 26



**b. 74151: bộ MUX 8 → 1**

$\overline{EN}$	C	B	A	Y
1	X	X	X	0
0	0	0	0	$D_0$
0	0	0	1	$D_1$
0	0	1	0	$D_2$
0	0	1	1	$D_3$
0	1	0	0	$D_4$
0	1	0	1	$D_5$
0	1	1	0	$D_6$
0	1	1	1	$D_7$

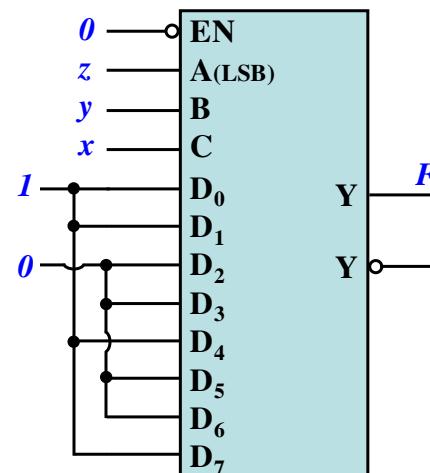
29

**3. Sử dụng bộ MUX thực hiện hàm Boolean:****a. Bộ MUX  $2^n$  thực hiện hàm Boolean n biến:**

$$\begin{aligned}
 F(x, y, z) &= \sum(0, 1, 4, 7) \\
 &= m_0 + m_1 + m_4 + m_7 \\
 &= m_0 1 + m_1 1 + m_2 0 + m_3 0 \\
 &\quad + m_4 1 + m_5 0 + m_6 0 + m_7 1
 \end{aligned}$$

$$\begin{aligned}
 Y &= \sum m_i D_i \\
 &= m_0 D_0 + m_1 D_1 + m_2 D_2 + m_3 D_3 \\
 &\quad + m_4 D_4 + m_5 D_5 + m_6 D_6 + m_7 D_7
 \end{aligned}$$

$$\begin{array}{l}
 \Rightarrow \boxed{D_0 = D_1 = D_4 = D_7 = 1} \\
 \boxed{D_2 = D_3 = D_5 = D_6 = 0}
 \end{array}$$



30

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

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

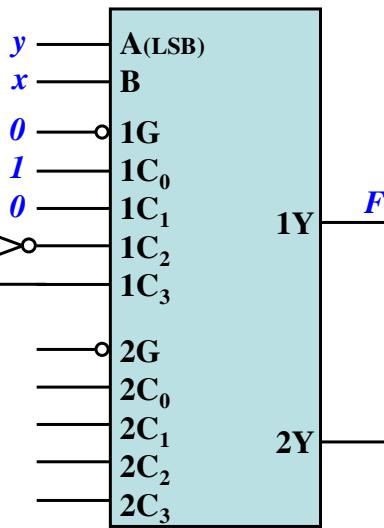
$$= \bar{x}\bar{y}\bar{z} + \bar{x}\bar{y}z + x\bar{y}\bar{z} + xy\bar{z}$$

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

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

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

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

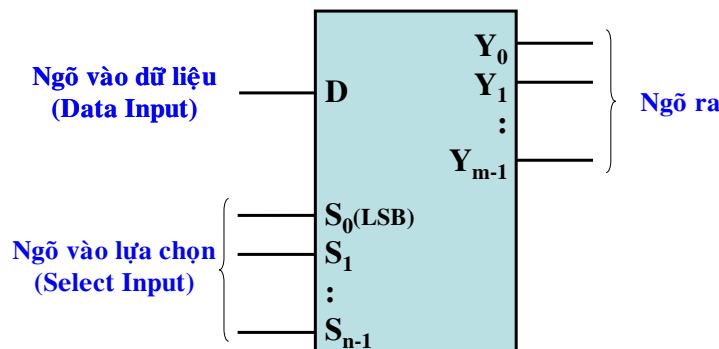


x	y	z	F
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

31

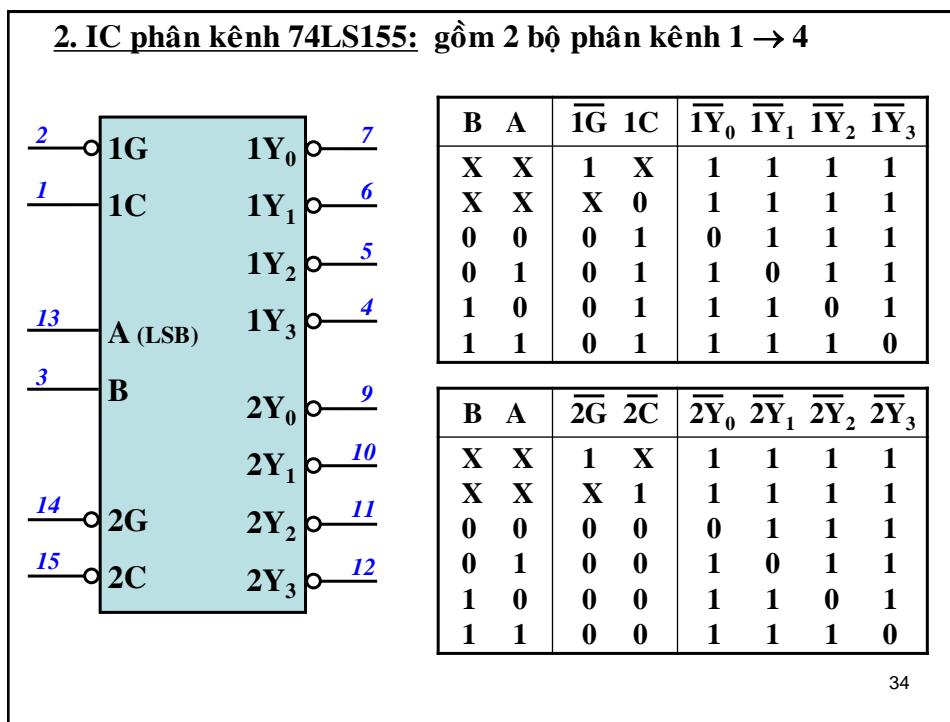
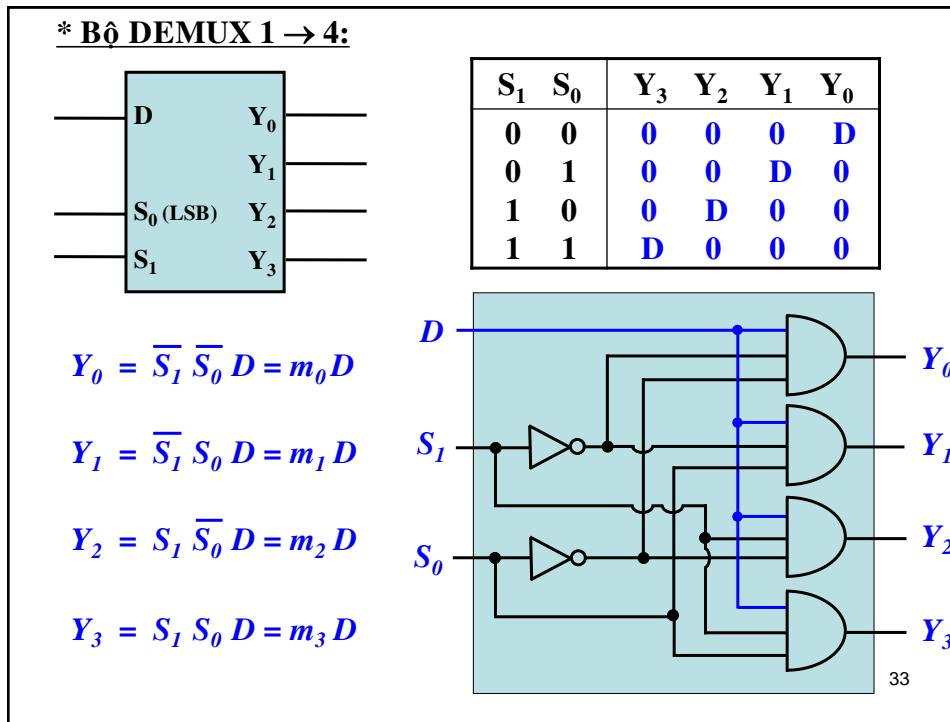
**VII. Bộ phân kênh (DEMUX):****1. Giới thiệu:**

- 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  $Y_i$ .

32



### VIII. Bộ so sánh ô ô lớn (Comparator):

#### 1. Giới thiệu:

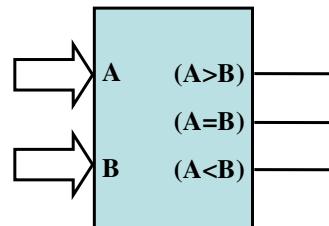
- Bộ so sánh là hệ tổ hợp có nhiệm vụ so sánh 2 số nhị phân không dấu A và B (mỗi số n bit).
- Bộ so sánh có 3 ngõ ra ( $A > B$ ), ( $A = B$ ) và ( $A < B$ ); chỉ có 1 ngõ ra tích cực theo kết quả so sánh.

\* Bộ so sánh 3 bit:

$$\begin{array}{ll} \textbf{A:} & A_2 \ A_1 \ A_0 \\ \textbf{B:} & B_2 \ B_1 \ B_0 \end{array}$$

Sử dụng biến trung gian:

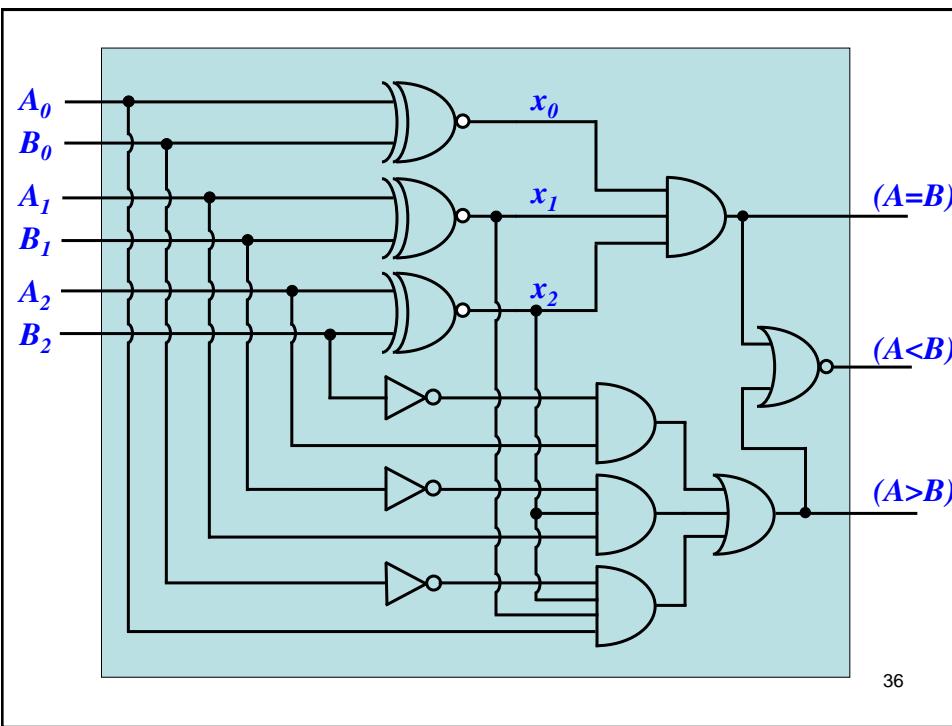
$$x_i = \overline{A_i} \oplus \overline{B_i} \quad (i = 0, 1, 2)$$



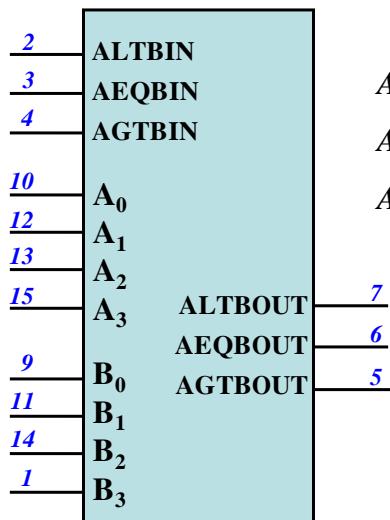
$$(A = B) = x_2 x_1 x_0$$

$$(A > B) = A_2 \overline{B}_2 + x_2 A_1 \overline{B}_1 + x_2 x_1 A_0 \overline{B}_0$$

$$(A < B) = \overline{A}_2 B_2 + x_2 \overline{A}_1 B_1 + x_2 x_1 \overline{A}_0 B_0 = \overline{(A = B) + (A > B)}^{35}$$



36

**2. IC so sánh 74LS85:**

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

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

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

37