

# Chương 4

## HỆ TỔ HỢP

### 4.1. KHÁI NIỆM CHUNG

Các phần tử logic AND, OR, NOR, NAND là các phần tử logic cơ bản còn được gọi là hệ tổ hợp đơn giản. Như vậy, hệ tổ hợp là hệ có các ngõ ra là các hàm logic theo ngõ vào, điều này nghĩa là khi một trong các ngõ vào thay đổi trạng thái lập tức làm cho ngõ ra thay đổi trạng thái ngay (nếu bỏ qua thời gian trễ của các phần tử logic) mà không chịu ảnh hưởng của trạng thái ngõ ra trước đó.

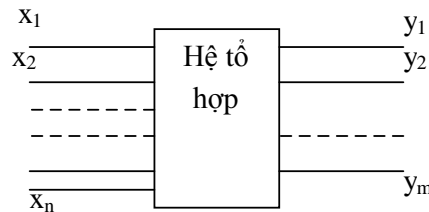
Xét một hệ tổ hợp có  $n$  ngõ vào và có  $m$  ngõ ra (hình 4.1), ta có:

$$y_1 = f(x_1, x_2, \dots, x_n)$$

$$y_2 = f(x_1, x_2, \dots, x_n)$$

.....

$$y_m = f(x_1, x_2, \dots, x_n)$$



Hình 4.1

Như vậy, sự thay đổi của ngõ ra  $y_j$  ( $j = 1 \div m$ ) theo các biến vào  $x_i$  ( $i = 1 \div n$ ) là tùy thuộc vào bảng trạng thái mô tả hoạt động của hệ tổ hợp.

Đặc điểm cơ bản của hệ tổ hợp là tín hiệu ra tại mỗi thời điểm chỉ phụ thuộc vào giá trị các tín hiệu vào ở thời điểm đó mà không phụ thuộc vào giá trị các tín hiệu ngõ ra ở thời điểm trước đó.

#### **Trình tự để thiết kế hệ tổ hợp theo các bước sau:**

1. Từ yêu cầu thực tế ta lập bảng trạng thái mô tả hoạt động của mạch (hệ tổ hợp).
2. Dùng các phương pháp tối thiểu để tối thiểu hoá các hàm logic.
3. Thành lập sơ đồ logic (Dựa vào phương trình logic đã tối giản).
4. Thành lập sơ đồ hệ tổ hợp.

Các mạch tổ hợp thông dụng:

- Mạch mã hoá - giải mã
- Mạch chọn kênh - phân đường
- Mạch so sánh
- Mạch số học ....v....v....

### 4.2. MẠCH MÃ HOÁ & MẠCH GIẢI MÃ

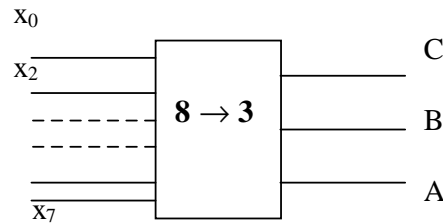
#### 4.2.1. Khái niệm:

Mạch mã hoá (ENCODER) là mạch có nhiệm vụ biến đổi những ký hiệu quen thuộc với con người sang những ký hiệu không quen thuộc con người. Ngược lại, mạch giải mã (DECODER) là mạch làm nhiệm vụ biến đổi những ký hiệu không quen thuộc với con người sang những ký hiệu quen thuộc với con người.

## 4.2.2. Mạch mã hoá (Encoder)

### 1. Mạch mã hoá nhị phân

Xét mạch mã hóa nhị phân từ 8 sang 3 (8 ngõ vào và 3 ngõ ra). Sơ đồ khối của mạch được cho trên hình 4.2.



Hình 4.2 Sơ đồ khối mạch mã hóa nhị phân từ 8 sang 3

Trong đó:

- $x_0, x_1, \dots, x_7$  là 8 đường tín hiệu vào
- A, B, C là 3 ngõ ra.

Mạch mã hóa nhị phân thực hiện biến đổi tín hiệu ngõ vào thành một từ mã nhị phân tương ứng ở ngõ ra, cụ thể như sau:

$0 \rightarrow 000$        $3 \rightarrow 011$        $6 \rightarrow 100$   
 $1 \rightarrow 001$        $4 \rightarrow 100$        $7 \rightarrow 111$   
 $2 \rightarrow 010$        $5 \rightarrow 101$

Chọn mức tác động (tích cực) ở ngõ vào là mức logic 1, ta có bảng trạng thái mô tả hoạt động của mạch :

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

**Giải thích bảng trạng thái:** Khi một ngõ vào ở trạng thái tích cực (mức logic 1) và các ngõ vào còn lại không được tích cực (mức logic 0) thì ngõ ra xuất hiện từ mã tương ứng. Cụ thể là: khi ngõ vào  $x_0=1$  và các ngõ vào còn lại bằng 0 thì từ mã ở ngõ ra là 000, khi ngõ vào  $x_1=1$  và các ngõ vào còn lại bằng 0 thì từ mã nhị phân ở ngõ ra là 001, ..v..v..

Phương trình logic tối giản:

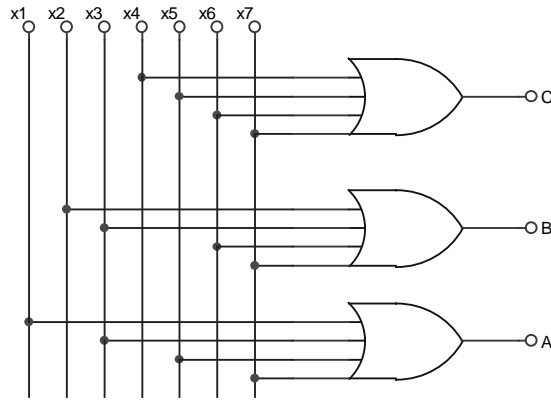
$$A = x_1 + x_3 + x_5 + x_7$$

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

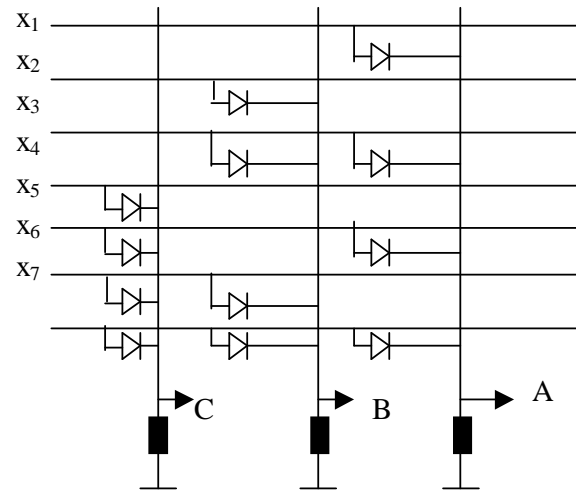
$$C = x_4 + x_5 + x_6 + x_7$$

Sơ đồ logic thực hiện mạch mã hóa nhị phân từ 8 sang 3 (hình 4.3):

Biểu diễn bằng công logic dùng Diode (hình 4.4):



Hình 4.3 Mạch mã hóa nhị phân từ 8 sang 3



Hình 4.4 Mạch mã hóa nhị phân từ 8 sang 3 sử dụng diode

Nếu chọn mức tác động tích cực ở ngõ vào là mức logic 0, bảng trạng thái mô tả hoạt động của mạch lúc này như sau:

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

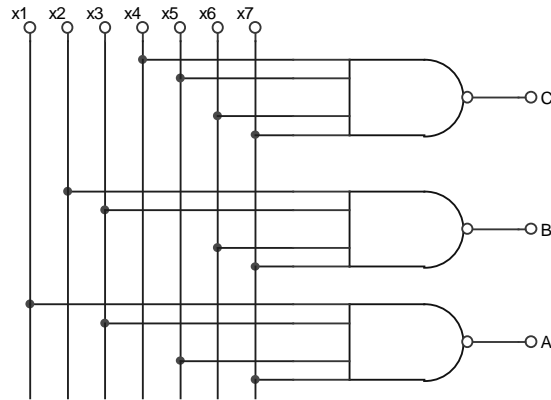
Phương trình logic tối giản :

$$A = \bar{x}_1 + \bar{x}_3 + \bar{x}_5 + \bar{x}_7 = \overline{x_1 x_3 x_5 x_7}$$

$$B = \bar{x}_2 + \bar{x}_3 + \bar{x}_6 + \bar{x}_7 = \overline{x_2 x_3 x_6 x_7}$$

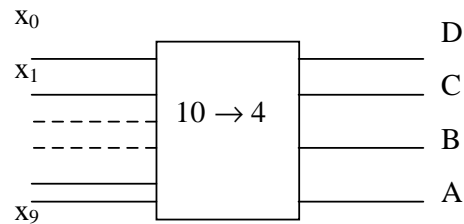
$$C = \bar{x}_4 + \bar{x}_5 + \bar{x}_6 + \bar{x}_7 = \overline{x_4 x_5 x_6 x_7}$$

Sơ đồ mạch thực hiện cho trên hình 4.5



Hình 4.5 Mạch mã hóa nhị phân 8 sang 3 ngõ vào tích cực mức 0

## 2. Mạch mã hoá thập phân



Hình 4.6 Sơ đồ khối mạch mã hóa từ 10 sang 4

Bảng trạng thái mô tả hoạt động của mạch :

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

Phương trình logic đã tối giản:

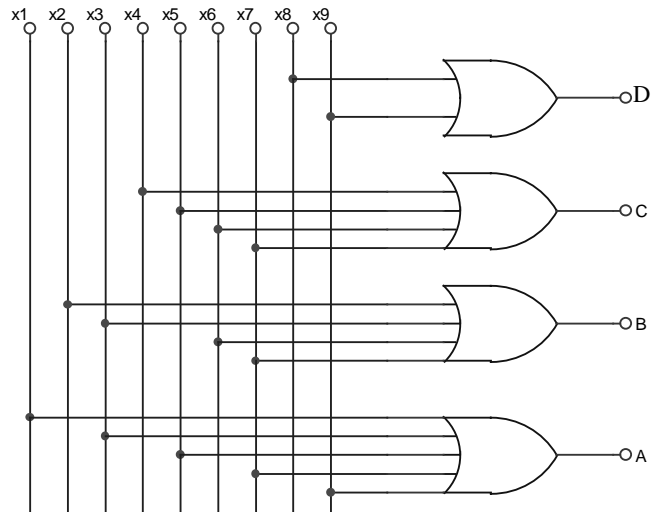
$$A = x_1 + x_3 + x_5 + x_7 + x_9$$

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

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

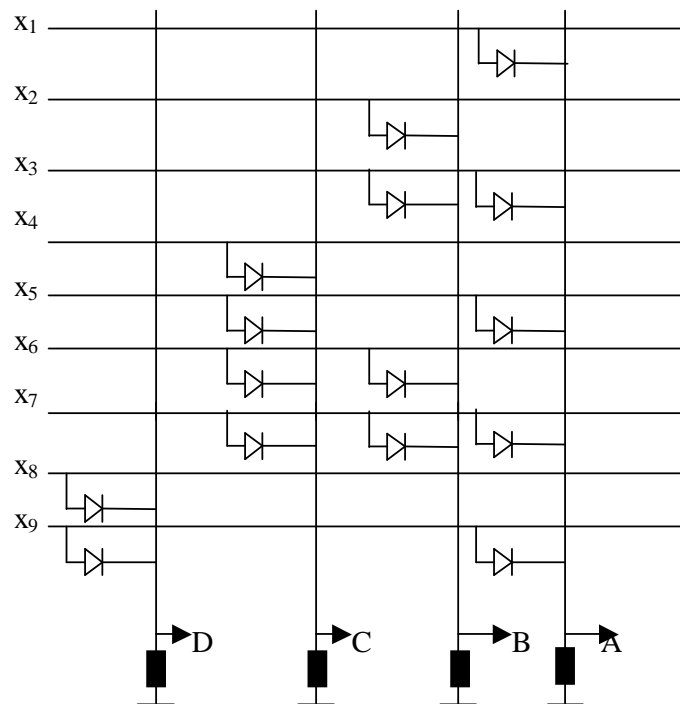
$$D = x_8 + x_9$$

Biểu diễn bằng sơ đồ logic (hình 4.7)



Hình 4.7 Sơ đồ mạch mã hóa thập phân từ  $10 \rightarrow 4$

Biểu diễn sơ đồ này bằng công logic sử dụng Diode được cho trên hình 4.8



Hình 4.8

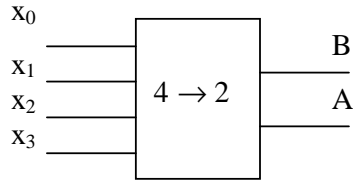
### 3. Mạch mã hoá ưu tiên

Trong hai mạch mã hoá đã xét ở trên, tín hiệu đầu vào tồn tại độc lập tức là không có tình huống có 2 tín hiệu trở lên đồng thời tác động ở mức logic 1 (nếu ta chọn mức tích cực ở ngõ vào là mức logic 1), thực tế đây là tình huống hoàn toàn có thể xảy ra, do đó cần phải đặt ra vấn đề ưu tiên.

Vấn đề ưu tiên: Khi có nhiều tín hiệu vào đồng thời tác động, tín hiệu nào có mức ưu tiên cao hơn ở thời điểm đang xét sẽ được ưu tiên tác động, tức là nếu ngõ vào có độ ưu tiên cao hơn bằng 1

trong khi những ngõ vào có độ ưu tiên thấp hơn nếu bằng 1 thì mạch sẽ tạo ra từ mã nhị phân ứng với ngõ vào có độ ưu tiên cao nhất.

Xét mạch mã hoá ưu tiên  $4 \rightarrow 2$  (4 ngõ vào, 2 ngõ ra) (hình 4.9).



Hình 4.9

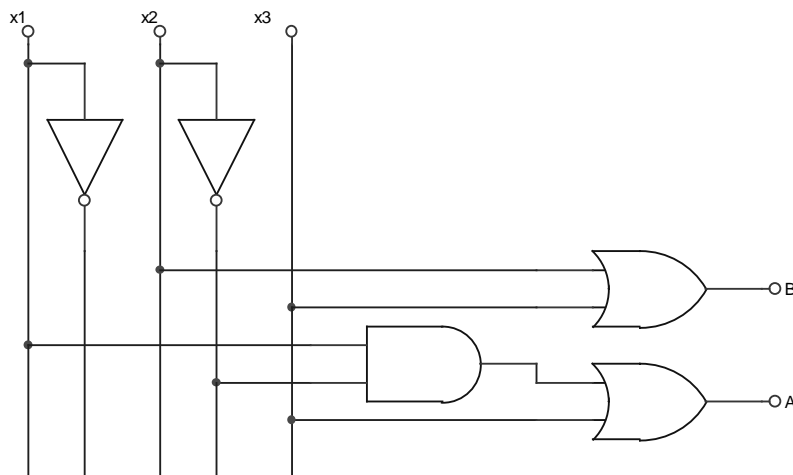
Bảng trạng thái

$x_0$	$x_1$	$x_2$	$x_3$	B	A
1	0	0	0	0	0
x	1	0	0	0	1
x	x	1	0	1	0
x	x	x	1	1	1

Từ bảng trạng thái có thể viết được phương trình logic các ngõ ra A và B:

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

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



Hình 4.10 Sơ đồ logic mạch mã hóa ưu tiên  $4 \rightarrow 2$

Sơ đồ logic: hình 4.10.

Một số vi mạch mã hóa ưu tiên thông dụng: 74LS147, 74LS148.

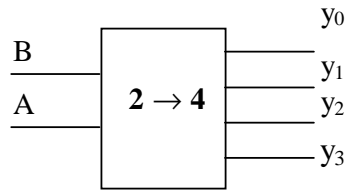
### 4.2.3. Mạch giải mã (Decoder)

#### 1. Mạch giải mã nhị phân

Xét mạch giải mã nhị phân  $2 \rightarrow 4$  (2 ngõ vào, 4 ngõ ra) như trên hình 4.11

Chọn mức tích cực ở ngõ ra là mức logic 1.

Bảng trạng thái mã tái hoạt  
đầu của mạch



Hình 4.11 Mạch giải mã 2 sang 4

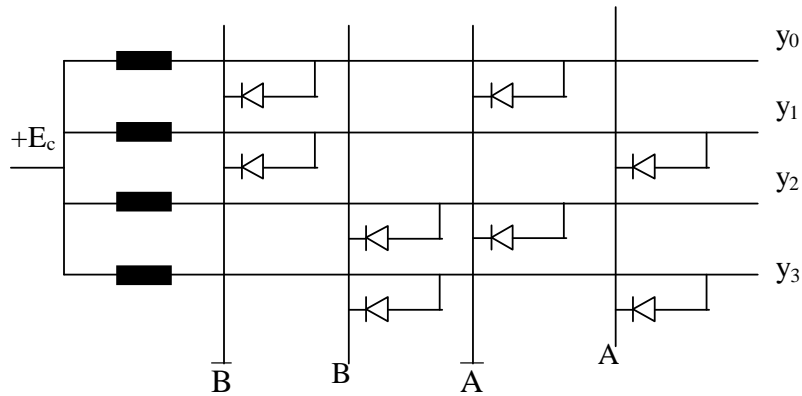
B	A	y <sub>0</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Phương trình logic tối giản và sơ đồ mạch thực hiện

$$y_0 = \overline{B}.\overline{A} \quad y_1 = \overline{B}.A$$

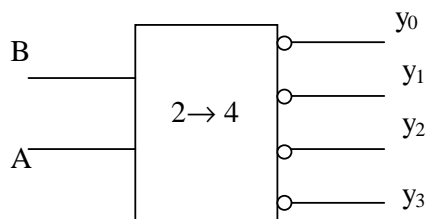
$$y_2 = B.\overline{A} \quad y_3 = A.B$$

Biểu diễn bằng công logic dùng Diode.



Hình 4.13. Mạch giải mã 2 → 4 dùng diode

Trường hợp chọn mức tích cực ở ngõ ra là mức logic 0 (mức logic thấp) ta có sơ đồ khối mạch giải mã được cho trên hình 4.14.



Hình 4.14. Mức tích cực ngõ ra là mức thấp

Bảng trạng thái

B	A	y <sub>0</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

Phương trình logic:

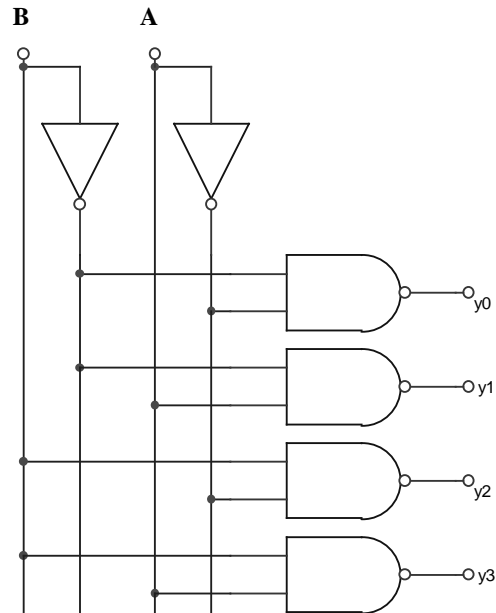
$$y_0 = B + A = \overline{\overline{B}.\overline{A}}$$

$$y_1 = B + \overline{A} = \overline{\overline{B}.A}$$

$$y_2 = \overline{B} + A = \overline{B.A}$$

$$y_3 = \overline{B} + \overline{A} = \overline{B.A}$$

Sơ đồ mạch thực hiện:



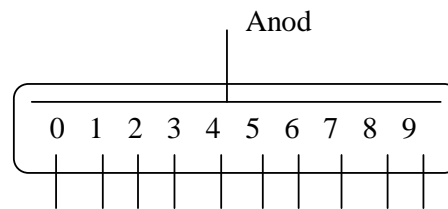
Hình 4.15. Mạch giải mã  $2 \rightarrow 4$  với ngõ ra mức tích cực thấp

## 2. Mạch giải mã thập phân

### a. Giải mã đèn NIXIE

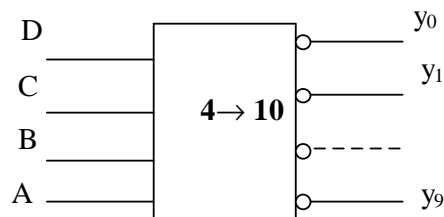
Đèn NIXIE là loại đèn điện tử loại Katod lạnh (Katod không được nung nóng bởi tim đèn), có cấu tạo gồm một Anod và 10 Katod mang hình các số từ 0 đến 9.

Sơ đồ khai triển của đèn được cho trên hình 4.16:



Hình 4.16. Sơ đồ khai triển của đèn NIXIE

Sơ đồ khối của mạch giải mã đèn NIXIE



Hình 4.17. Sơ đồ khối mạch giải mã đèn NIXIE

Chọn mức tích cực ở ngõ ra là mức logic 1, lúc đó bảng trạng thái hoạt động của mạch như sau:

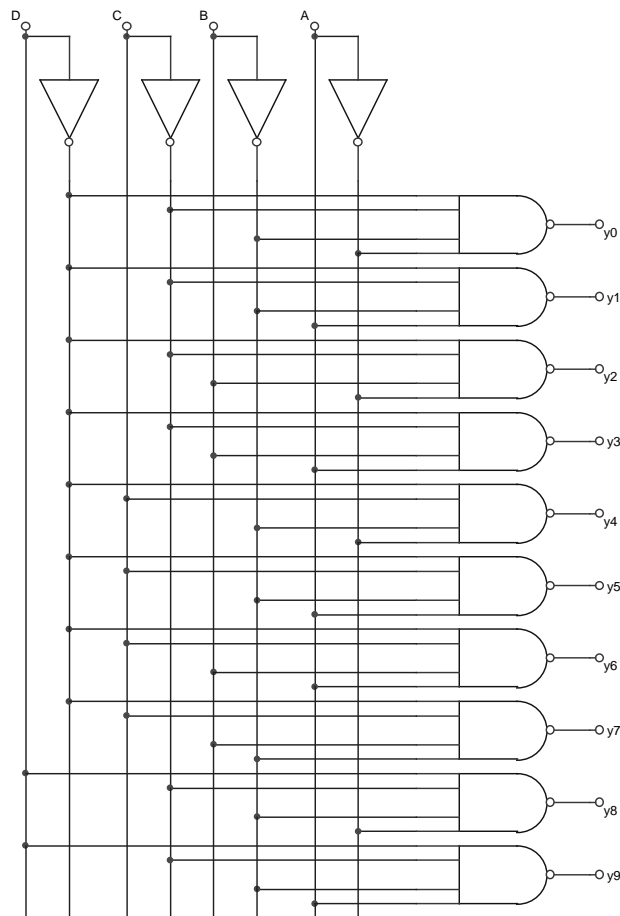


D	C	B	A	y <sub>0</sub>	y <sub>1</sub>	y <sub>2</sub>	y <sub>3</sub>	y <sub>4</sub>	y <sub>5</sub>	y <sub>6</sub>	y <sub>7</sub>	y <sub>8</sub>	y <sub>9</sub>
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1

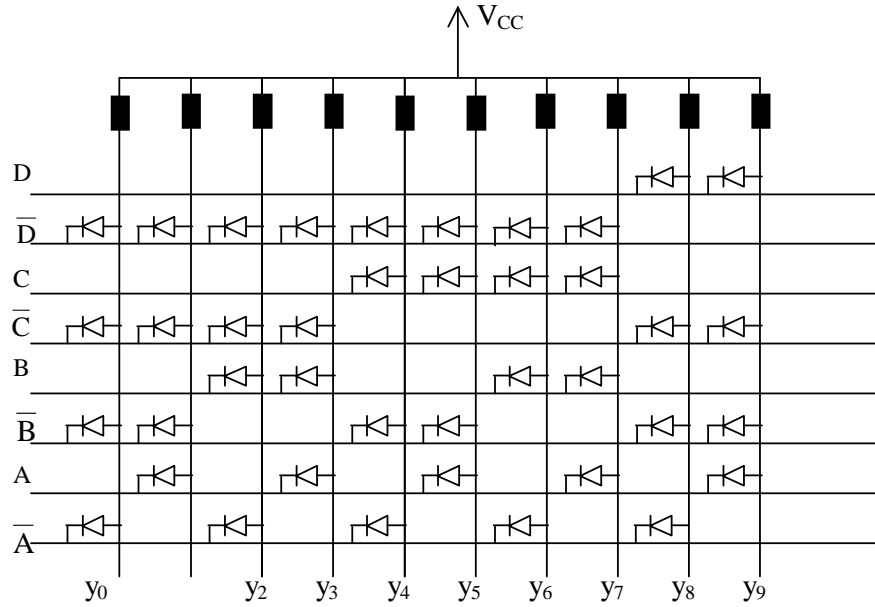
Phương trình logic:

$$\begin{aligned}
 y_0 &= \overline{D}\overline{C}\overline{B}\overline{A} & y_1 &= \overline{D}\overline{C}\overline{B}A & y_2 &= \overline{D}\overline{C}B\overline{A} & y_3 &= \overline{D}\overline{C}BA \\
 y_4 &= \overline{D}C\overline{B}\overline{A} & y_5 &= \overline{D}C\overline{B}A & y_6 &= \overline{D}CB\overline{A} & y_7 &= \overline{D}CBA \\
 y_8 &= D\overline{C}\overline{B}\overline{A} & y_9 &= D\overline{C}\overline{B}A
 \end{aligned}$$

Sơ đồ thực hiện mạch giải mã đèn NIXIE được cho trên hình 4.18 và 4.19:



Hình 4.18. Sơ đồ thực hiện bằng cổng logic

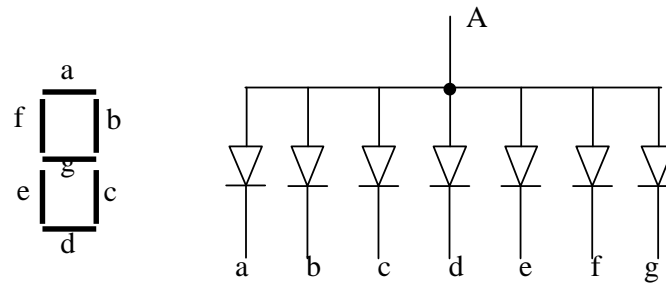


Hình 4.19. Sơ đồ thực hiện dùng diode

**b. Giải mã đèn LED 7 đoạn**

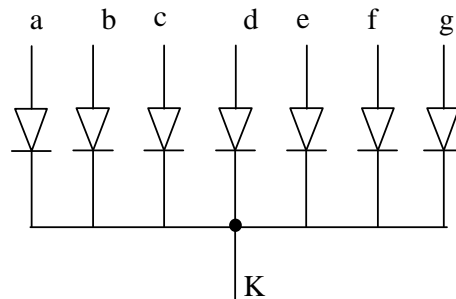
Đèn LED 7 đoạn có cấu tạo gồm 7 đoạn, mỗi đoạn là 1 đèn LED. Tùy theo cách nối các Kathode (Catốt) hoặc các Anode (Anốt) của các LED trong đèn, mà người ta phân thành hai loại:

LED 7 đoạn loại Anode chung:



Hình 4.20. LED 7 đoạn loại Anode chung

LED 7 đoạn loại Kathode chung :



Hình 4.21. LED 7 đoạn loại Kathode chung

Ứng với mỗi loại LED khác nhau ta có một mạch giải mã riêng. Sơ đồ khối của mạch giải mã LED 7 đoạn như sau:



Hình 4.22. Sơ đồ khối mạch giải mã LED 7 đoạn

#### **Giải mã LED 7 đoạn loại Anode chung:**

Đối với LED bảy đoạn loại anode chung, vì các anode của các đoạn led được nối chung với nhau và đưa lên mức logic 1 (5V), nên muốn đoạn led nào tắt ta nối cathode tương ứng lên mức logic 1 (5V) và ngược lại muốn đoạn led nào sáng ta nối cathode tương ứng xuống mass (mức logic 0).

Ví dụ: Để hiển thị số 0 ta nối cathode của đèn g lên mức logic 1 để đèn g tắt, và nối các cathode của đèn a, b, c, d, e, f xuống mass nên ta thấy số 0.

Lúc đó bảng trạng thái mô tả hoạt động của mạch giải mã LED bảy đoạn loại Anode chung như sau:

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

Dùng bảng Karnaugh để tối thiểu hóa mạch trên. Phương trình tối thiểu hóa có thể viết ở dạng chính tắc 1 (tổng của các tích số) hoặc dạng chính tắc 2 (tích của các tổng số):

Phương trình logic của ngõ ra a:

Dạng chính tắc 2:

$$a = \overline{B}.D.(\overline{C} + \overline{A})(C + A) = \overline{B}C\overline{D}A + \overline{B}DCA$$

Dạng chính tắc 1:

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

Lưu ý: Trên bảng Karnaugh chúng ta đã thực hiện tối thiểu hóa theo dạng chính tắc 2.

**a**

	DC			
BA	00	01	11	10
00	0	1	x	0
01	1	0	x	0
11	0	0	x	x
10	0	0	x	x

Phương trình logic của ngõ ra b:

Dạng chính tắc 2:

$$b = .C(A + B)(\overline{A} + \overline{B}) = C(\overline{A}B + A\overline{B})$$

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

Dạng chính tắc 1:

$$b = C\overline{B}A + CBA\overline{A} = C(A \oplus B)$$

**b**

	DC			
BA	00	01	11	10
00	0	0	x	0
01	0	1	x	0
11	0	0	x	x
10	0	1	x	x

Phương trình logic của ngõ ra c:

Dạng chính tắc 2:

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

Dạng chính tắc 1:

$$c = \overline{D}CBA$$

**c**

	DC			
BA	00	01	11	10
00	0	0	x	0
01	0	0	x	0
11	0	0	x	x
10	1	0	x	x

Phương trình logic của ngõ ra d:

Dạng chính tắc 2:

$$d = \overline{D}(\overline{A} + B + \overline{C})(\overline{B} + C + D)(A + \overline{B})(A + C)$$

$$= \overline{A}BCD + ABCD + \overline{A}BCD$$

Dạng chính tắc 1:

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

**d**

	DC			
BA	00	01	11	10
00	0	1	x	0
01	1	0	x	0
11	0	1	x	x
10	0	0	x	x

Phương trình logic của ngõ ra e:

Dạng chính tắc 2:

$$e = .(\overline{B} + A)(C + A)$$

Dạng chính tắc 1:

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

**e**

	DC			
BA	00	01	11	10
00	0	1	x	0
01	1	1	x	1
11	1	1	x	x
10	0	0	x	x

Phương trình logic của ngõ ra f:

Dạng chính tắc 2:

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

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

Dạng chính tắc 1:

$$f = BA + \overline{D}CA + \overline{D}CB$$

f	DC				
		00	01	11	10
BA	00	0	0	x	0
	01	1	0	x	0
	11	1	1	x	x
	10	1	0	x	x

Phương trình logic của ngõ ra g:

Dạng chính tắc 2:

$$g = \overline{D}(A + \overline{B})(\overline{C} + B)(\overline{B} + C)$$

$$= \overline{B}\overline{C}\overline{D} + \overline{D}CBA$$

Dạng chính tắc 1:

$$g = \overline{D}CBA + \overline{D}CB\overline{B}$$

g	DC				
		00	01	11	10
BA	00	1	0	x	0
	01	1	0	x	0
	11	0	1	x	x
	10	0	0	x	x

### Xét mạch giải mã đèn led 7 đoạn loại Kathode chung:

Chọn mức tích cực ở ngõ ra là mức logic 1. Vì Kathode của các đoạn led được nối chung và được nối xuống mức logic 0 (0V-mass) nên muốn đoạn led nào tắt ta đưa Anode tương ứng xuống mức logic 0 (0V-mass).

Ví dụ: Để hiển thị số 0 ta nối Anode của đoạn led g xuống mức logic 0 để đoạn g tắt, đồng thời các cathode của đoạn a, b, c, d, e, f được nối lên nguồn nên các đoạn này sẽ sáng do đó ta thấy số 0.

Lúc đó bảng trạng thái mô tả hoạt động của mạch như sau:

D	B	C	A	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
1	0	1	0	X	X	X	X	X	X	X
1	0	1	1	X	X	X	X	X	X	X
1	1	0	0	X	X	X	X	X	X	X
1	1	0	1	X	X	X	X	X	X	X
1	1	1	0	X	X	X	X	X	X	X
1	1	1	1	X	X	X	X	X	X	X

Tương tự như trường hợp trên, ta cũng dùng bảng Karnaugh để tối thiểu hóa hàm mạch và đi tìm phương trình logic tối giản các ngõ ra của các đoạn led: (Lưu ý trong những bảng đồ Karnaugh sau ta thực hiện tối thiểu hóa theo dạng chính tắc 1)

Phương trình logic của ngõ ra a:

Dạng chính tắc 1:

$$a = D + B + \overline{AC} + AC$$

Dạng chính tắc 2:

$$\begin{aligned} a &= (\overline{A} + B + C + D)(A + B + \overline{C}) \\ &= AD + B + AC + \overline{AC} \end{aligned}$$

a

DC	00	01	11	10
BA	00	01	11	10
00	1	0	x	1
01	0	1	x	1
11	1	1	x	x
10	1	1	x	x

Phương trình logic của ngõ ra b:

Dạng chính tắc 1:

$$b = \overline{C} + BA + \overline{B} \overline{A} = \overline{C} + A \oplus B$$

Dạng chính tắc 2:

$$\begin{aligned} b &= (\overline{C} + B + \overline{A})(\overline{C} + \overline{B} + A) \\ &= \overline{C} + AB + \overline{AB} = \overline{C} + A \oplus B \end{aligned}$$

b

DC	00	01	11	10
BA	00	01	11	10
00	1	1	x	1
01	1	0	x	1
11	1	1	x	x
10	1	0	x	x

Phương trình logic của ngõ ra c:

Dạng chính tắc 1:

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

Dạng chính tắc 2:

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

c

DC	00	01	11	10
BA	00	01	11	10
00	1	1	x	1
01	1	1	x	1
11	1	1	x	x
10	0	1	x	x

Phương trình logic của ngõ ra d:

Dạng chính tắc 1:

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

Dạng chính tắc 2:

$$\begin{aligned} d &= (A + B + \overline{C})(\overline{A} + \overline{B} + \overline{C})(\overline{A} + B + C + D) \\ &= (\overline{C} + A \overline{B} + \overline{AB})(\overline{A} + B + C + D) \\ &= (C + A \oplus B)(\overline{A} + B + C + D) \end{aligned}$$

d

DC	00	01	11	10
BA	00	01	11	10
00	1	0	x	1
01	0	1	x	1
11	1	0	x	x
10	1	1	x	x

Phương trình logic của ngõ ra e:

Dạng chính tắc 1:

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

Dạng chính tắc 2:

$$e = \overline{A} (\overline{C} + B) = \overline{A} \overline{C} + \overline{A} . B$$

e

DC	00	01	11	10
BA	00	01	11	10
00	1	0	x	1
01	0	0	x	0
11	0	0	x	x
10	1	1	x	x

Phương trình logic của ngõ ra f:

Dạng chính tắc 1:

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

Dạng chính tắc 2:

$$\begin{aligned} f &= (\bar{B} + \bar{A})(D + C + \bar{A})(C + \bar{B}) \\ &= D + \bar{B}C + \bar{A}C + \bar{A}\bar{B} \end{aligned}$$

Phương trình logic của ngõ ra g:

Dạng chính tắc 1:

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

Dạng chính tắc 2:

$$g = (\bar{C} + \bar{B} + \bar{A})(B + C + D)$$

f		DC			
BA		00	01	11	10
		00	01	11	10
00		(1)	(1)	(x)	(1)
01		0	(1)	(x)	1
11		0	0	x	x
10		0	(1)	(x)	x

g		DC			
BA		00	01	11	10
		00	01	11	10
00		0	1	(x)	1
01		0	(1)	(x)	1
11		(1)	0	x	(x)
10		(1)	(1)	(x)	(x)

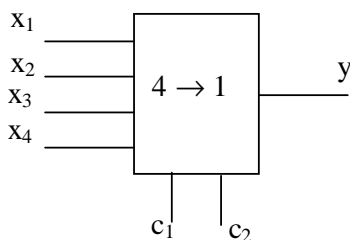
## 4.3. MẠCH CHỌN KÊNH - PHÂN ĐƯỜNG

### 4.3.1. Đại cương

Mạch chọn kênh còn gọi là mạch hợp kênh (ghép kênh) là mạch có chức năng chọn lần lượt 1 trong N kênh vào để đưa đến ngõ ra duy nhất (ngõ ra duy nhất đó gọi là đường truyền chung). Do đó, mạch chọn kênh còn gọi là mạch chuyển dữ liệu song song ở ngõ vào thành dữ liệu nối tiếp ở ngõ ra, được gọi là Multiplex (viết tắt là MUX).

Mạch chọn kênh thực hiện chức năng ở đầu phát còn mạch phân đường thực hiện chức năng ở đầu thu. Mạch phân đường còn gọi là mạch tách kênh (phân kênh, giải đa hợp), mạch này có nhiệm vụ tách N nguồn dữ liệu khác nhau ở cùng một đầu vào để rẽ ra N ngõ ra khác nhau. Do đó, mạch phân đường còn gọi là mạch chuyển dữ liệu nối tiếp ở ngõ vào thành dữ liệu song song ở ngõ ra, được gọi là Demultiplex (viết tắt là DEMUX).

### 4.3.2. Mạch chọn kênh



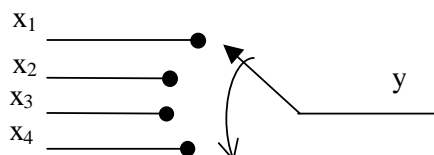
Hình 4.23a. Mạch chọn kênh

Xét mạch chọn kênh đơn giản có 4 ngõ vào và 1 ngõ ra như hình 4.23a.

Trong đó:

- +  $x_1, x_2, x_3, x_4$  : Các kênh dữ liệu vào.
- + Ngõ ra  $y$  : Đường truyền chung.
- +  $c_1, c_2$  : Các ngõ vào điều khiển

Vậy mạch này giống như 1 chuyển mạch (hình 4.23b):



Hình 4.23b

Để thay đổi lần lượt từ  $x_1 \rightarrow x_4$  phải có điều khiển do đó đối với mạch chọn kênh để chọn lần lượt từ 1 trong 4 kênh vào cần có các ngõ vào điều khiển  $c_1, c_2$ . Nếu có N kênh vào thì cần có n ngõ vào điều khiển thỏa mãn quan hệ:  $N=2^n$ . Nói cách khác: Số tổ hợp ngõ vào điều  **khiển bằng số lượng các kênh vào**.

Việc chọn dữ liệu từ 1 trong 4 ngõ vào để đưa đến đường truyền chung là tùy thuộc vào tổ hợp tín hiệu điều khiển tác động đến hai ngõ vào điều khiển  $c_1, c_2$ .

- +  $c_1 = 0, c_2 = 0 \rightarrow y = x_1$  ( $x_1$  được nối tới ngõ ra y).
- +  $c_1 = 0, c_2 = 1 \rightarrow y = x_2$  ( $x_2$  được nối tới ngõ ra y).
- +  $c_1 = 1, c_2 = 0 \rightarrow y = x_3$  ( $x_3$  được nối tới ngõ ra y).
- +  $c_1 = 1, c_2 = 1 \rightarrow y = x_4$  ( $x_4$  được nối tới ngõ ra y).

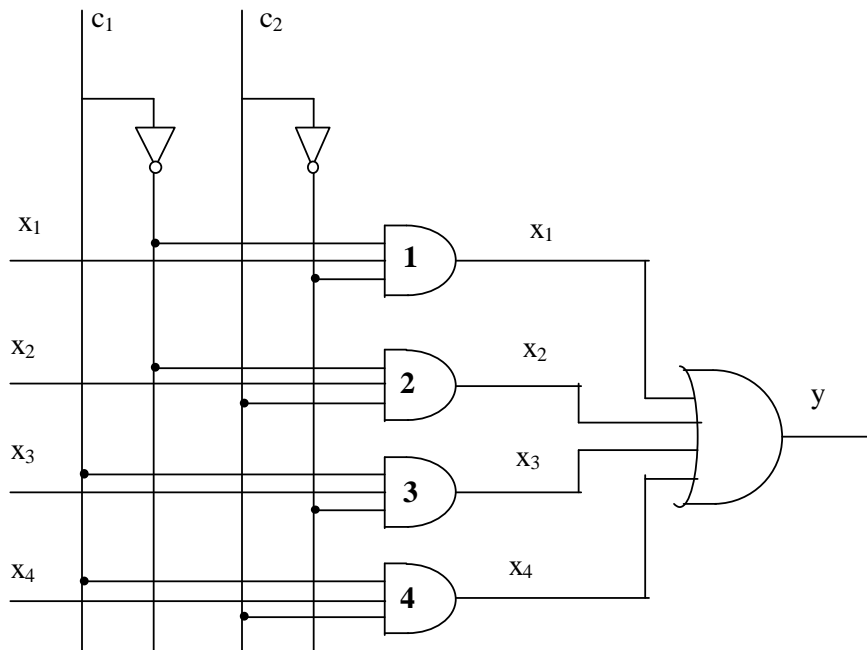
Vậy tín hiệu điều khiển phải liên tục để dữ liệu từ các kênh được liên tục đưa đến ngõ ra. Từ đó ta lập được bảng trạng thái mô tả hoạt động của mạch chọn kênh.

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

Phương trình logic mô tả hoạt động của mạch :

$$y = \overline{c_1} \overline{c_2} . x_1 + \overline{c_1} c_2 . x_2 + c_1 \overline{c_2} . x_3 + c_1 . c_2 . x_4$$

Sơ đồ logic của mạch:

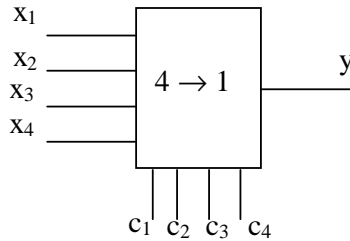


Hình 4.24. Sơ đồ logic mạch chọn kênh từ 4  $\rightarrow$  1

Bây giờ, xét mạch chọn kênh có 4 ngõ vào và 1 ngõ ra, nhưng lại có 4 ngõ điều khiển. Lúc này, ta không dựa vào tổ hợp tín hiệu tác động lên ngõ vào điều khiển, mà chỉ xét đến mức tích cực ở ngõ vào điều khiển. Ta sẽ chọn một trong hai mức logic 1 hoặc mức logic 0 làm mức tích cực, nếu 1 ngõ vào trong số 4 ngõ vào điều khiển tồn tại mức logic tích cực (mức 1 hoặc mức 0) thì kênh dữ liệu vào có cùng chỉ số với ngõ vào điều khiển đó sẽ được kết nối với ngõ ra. Trên hình 4.25 biểu diễn mạch chọn kênh với số lượng ngõ vào điều khiển bằng số lượng kênh vào.



Nếu chọn mức tích cực của các ngõ vào điều khiển là mức logic 1, ta có bảng trạng thái mô tả hoạt động của mạch như sau:



Hình 4.25. Mạch chọn kênh với số lượng ngõ vào điều khiển bằng số kênh vào

$c_1$	$c_2$	$c_3$	$c_4$	$y$
1	0	0	0	$x_1$
0	1	0	0	$x_2$
0	0	1	0	$x_3$
0	0	0	1	$x_4$

Phương trình logic:

$$y = c_1 \cdot x_1 + c_2 \cdot x_2 + c_3 \cdot x_3 + c_4 \cdot x_4$$

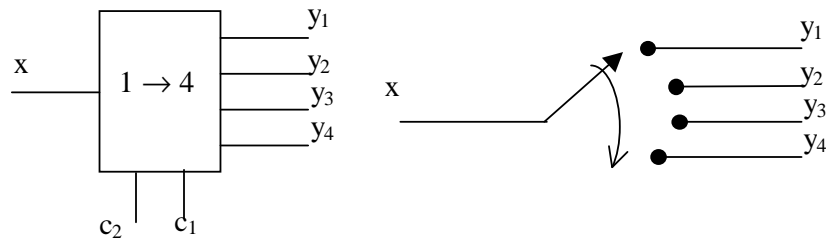
Ý nghĩa trong thực tế của mạch:

+  $c_1, c_2, c_3, c_4$  : Có thể hiểu là các địa chỉ (nguồn và đích).

+  $x_1, x_2, x_3, x_4$  : Thông tin cần truyền đi.

### 4.3.3. Mạch phân đường

Xét mạch phân đường đơn giản có 1 ngõ vào và 4 ngõ ra ký hiệu như sau :



Hình 4.26. Mạch phân đường đơn giản từ 1 → 4

Trong đó:

+  $x$  là kênh dữ liệu vào.

+  $y_1, y_2, y_3, y_4$  các ngõ ra dữ liệu;  $c_1, c_2$  các ngõ vào điều khiển.

Ta có thể thấy mạch này thực hiện chức năng như 1 chuyển mạch (hình vẽ 4.26).

Tùy thuộc vào tổ hợp tín hiệu điều khiển tác dụng vào mạch mà lần lượt tín hiệu từ ngõ vào  $x$  sẽ chuyển đến ngõ ra  $y_1, y_2, y_3, y_4$  một cách tương ứng.

Lúc đó bảng trạng thái mô tả hoạt động của mạch :

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

Phương trình logic các ngõ ra:

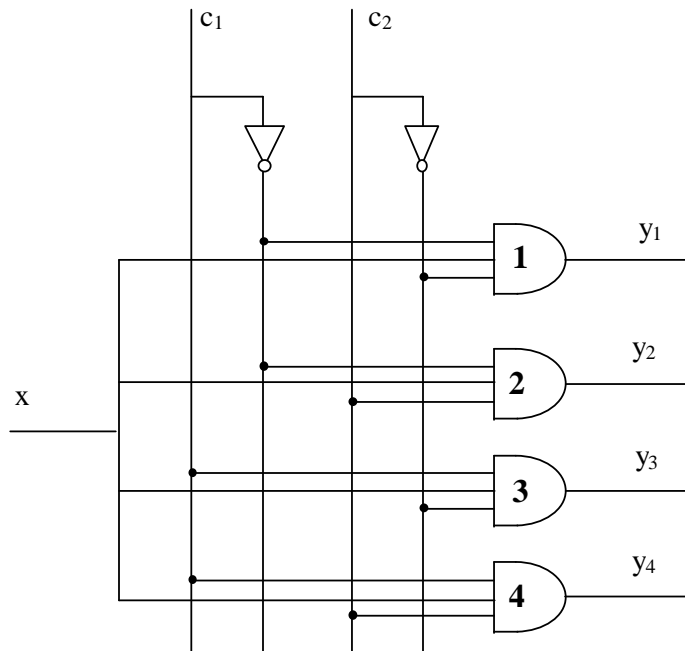
$$y_1 = \overline{c_1} \overline{c_2} . x$$

$$y_2 = \overline{c_1} c_2 . x$$

$$y_3 = c_1 \overline{c_2} . x$$

$$y_4 = c_1 c_2 . x$$

Sơ đồ logic được cho trên hình 4.27:



Hình 4.27. Sơ đồ logic thực hiện mạch phân đường

Nếu  $x = 1$  và hoán đổi ngõ vào điều khiển thành ngõ vào dữ liệu thì mạch phân đường chuyển thành mạch giải mã nhị phân. Vì vậy, nhà sản xuất đã chế tạo IC đảm bảo cả hai chức năng: giải mã và giải đa hợp (Decode/Demultilex). Ví dụ: các IC 74138, 74139, 74154: giải mã và phân đường tùy thuộc vào cách nối chân.

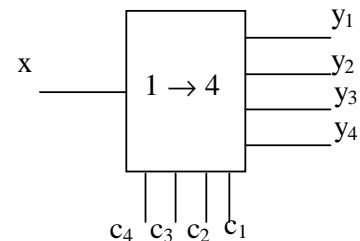
**Trong trường hợp tổng quát, mạch phân đường có 1 ngõ vào và  $2^n$  ngõ ra: để tách  $N=2^n$  nguồn dữ liệu khác nhau cần có  $n$  ngõ vào điều khiển, lúc đó số tổ hợp ngõ vào điều khiển bằng số lượng ngõ ra.**

Tuy nhiên trong thực tế, ta còn gặp mạch phân đường có số lượng ngõ vào điều khiển bằng số ngõ ra (hình 4.28). Lúc đó chỉ xét đến mức tích cực ở ngõ vào điều khiển, người ta chọn một trong hai mức logic 1 hoặc mức logic 0 làm mức tích cực. Giả sử chọn mức logic 1 là mức tích cực: nếu 1 ngõ vào trong số 4 ngõ vào điều khiển tồn tại mức logic 1 (mức tích cực), thì ngõ ra dữ liệu tương ứng có cùng chỉ số với ngõ vào điều khiển đó sẽ được nối với ngõ vào dữ liệu chung  $x$ .

Ví dụ:

$$c_1 = 1 \rightarrow x = y_1 \quad c_2 = 1 \rightarrow x = y_2$$

$$c_3 = 1 \rightarrow x = y_3 \quad c_4 = 1 \rightarrow x = y_4$$



Hình 4.28

Lúc đó bảng trạng thái hoạt động của mạch:

$c_1$	$c_2$	$c_3$	$c_4$	$y_1$	$y_2$	$y_3$	$y_4$
1	0	0	0	X	0	0	0
0	1	0	0	0	X	0	0
0	0	1	0	0	0	X	0
0	0	0	1	0	0	0	X

Phương trình logic và sơ đồ logic được cho trên hình 4.29:

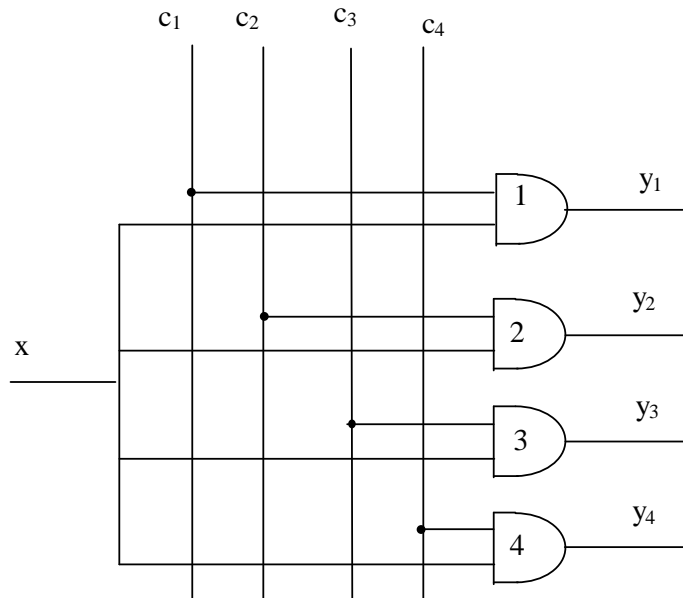
$$y_1 = c_1 x \quad y_2 = c_2 x$$

$$y_3 = c_3 x \quad y_4 = c_4 x$$

Giải thích hoạt động của mạch:

- + Khi  $c_1=1, c_2=c_3=c_4=0$  chỉ có cổng AND(1) thông cho dữ liệu từ x nối đến đầu ra  $y_1$ .
- + Khi  $c_2=1, c_1=c_3=c_4=0$  chỉ có cổng AND(2) thông cho dữ liệu từ x nối đến đầu ra  $y_2$ .
- + Khi  $c_3=1, c_2=c_1=c_4=0$  chỉ có cổng AND(3) thông cho dữ liệu từ x nối đến đầu ra  $y_3$ .
- + Khi  $c_4=1, c_2=c_3=c_1=0$  chỉ có cổng AND(4) thông cho dữ liệu từ x nối đến đầu ra  $y_4$ .

Vì mạch chọn kênh được thực hiện ở đầu phát và mạch phân đường được thực hiện ở đầu thu nên để đảm bảo dữ liệu được chuyển đúng kênh thì mạch chọn kênh và mạch phân đường phải đồng bộ với nhau.



Hình 4.29. Mạch phân đường số lượng ngõ vào điều khiển bằng số ngõ ra

## 4.4. MẠCH SO SÁNH

### 4.4.1. Đại cương

- Mạch so sánh dùng để so sánh các số nhị phân về mặt độ lớn.  
Ví dụ: So sánh a và b:  $a = 0, b = 1$  ( $a < b$ ).
- Có hai mạch so sánh:
  - + So sánh hai số nhị phân 1 bit.
  - + So sánh hai số nhị phân nhiều bit.

#### 4.4.2. Mạch so sánh 1 bit

Là mạch thực hiện chức năng so sánh hai số nhị phân 1 bit.

Xét hai số nhị phân 1 bit a và b. Có các trường hợp sau đây:

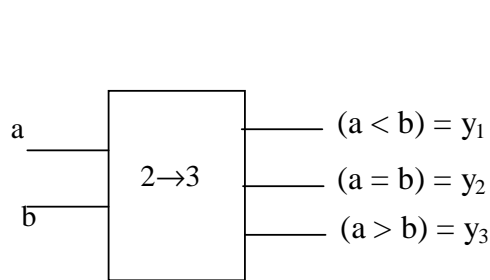
$$+ a = 0, b = 0 \Rightarrow a = b.$$

$$+ a = 1, b = 1 \Rightarrow a = b.$$

$$+ a = 0, b = 1 \Rightarrow a < b.$$

$$+ a = 1, b = 0 \Rightarrow a > b.$$

Về phương diện mạch điện, mạch so sánh 1 bit có 2 ngõ vào và 3 ngõ ra. Các ngõ vào a, b là các bit cần so sánh; các ngõ ra thể hiện kết quả so sánh:  $y_1$  ( $a < b$ ),  $y_2$  ( $a = b$ ) và  $y_3$  ( $a > b$ ). Sơ đồ khối mạch so sánh trên hình 4.30.



Hình 4.30. Mạch so sánh 1 bit

Bảng trạng thái

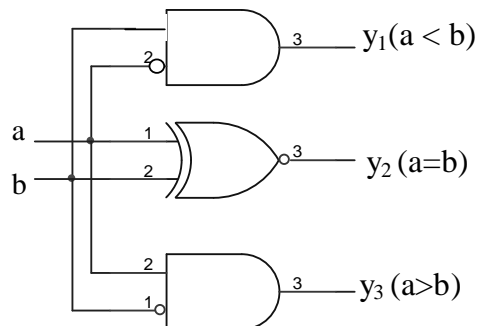
a	b	$y_1$	$y_2$	$y_3$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Chọn mức tích cực ở ngõ ra là mức logic 1. Ta lập được bảng trạng thái mô tả hoạt động của mạch. Từ bảng trạng thái, ta có phương trình logic:

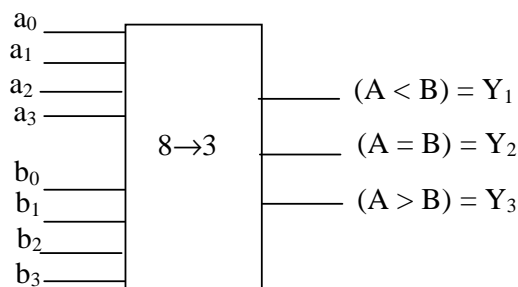
$$y_1 = \bar{a} \cdot b$$

$$y_2 = \bar{a} \cdot \bar{b} + a \cdot b = \overline{a \oplus b}$$

$$y_3 = a \cdot \bar{b}$$



Hình 4.31. Sơ đồ mạch so sánh 1 bit



Hình 4.32. Sơ đồ khối mạch so sánh nhiều bit

#### 4.4.3. Mạch so sánh nhiều bit

Mạch có 8 ngõ vào và 3 ngõ ra, thực hiện so sánh 2 số nhị phân 4 bit A ( $a_3a_2a_1a_0$ ) và B ( $b_3b_2b_1b_0$ ). Có hai phương pháp thực hiện mạch so sánh nhiều bit:

- Thực hiện trực tiếp.
- Thực hiện mạch so sánh nhiều bit trên cơ sở mạch so sánh 1 bit.

Chúng ta lần lượt xét từng phương pháp.

### 1. Phương pháp trực tiếp

Ta có bảng trạng thái hoạt động của mạch

INPUT				OUTPUT		
$a_3$ và $b_3$	$a_2$ và $b_2$	$a_1$ và $b_1$	$a_0$ và $b$	$A < B$	$A = B$	$A > B$
<	x	x	x	1	0	0
>	x	x	x	0	0	1
=	<	x	x	1	0	0
=	>	x	x	0	0	1
=	=	<	x	1	0	0
=	=	>	x	0	0	1
=	=	=	<	1	0	0
=	=	=	>	0	0	1
=	=	=	=	0	1	0

Phương trình logic của mạch:

$$Y_1 = (A < B)$$

$$= (a_3 < b_3) + (a_3 = b_3)(a_2 < b_2) + (a_3 = b_3)(a_2 = b_2)(a_1 < b_1) \\ + (a_3 = b_3)(a_2 = b_2)(a_1 = b_1)(a_0 < b_0)$$

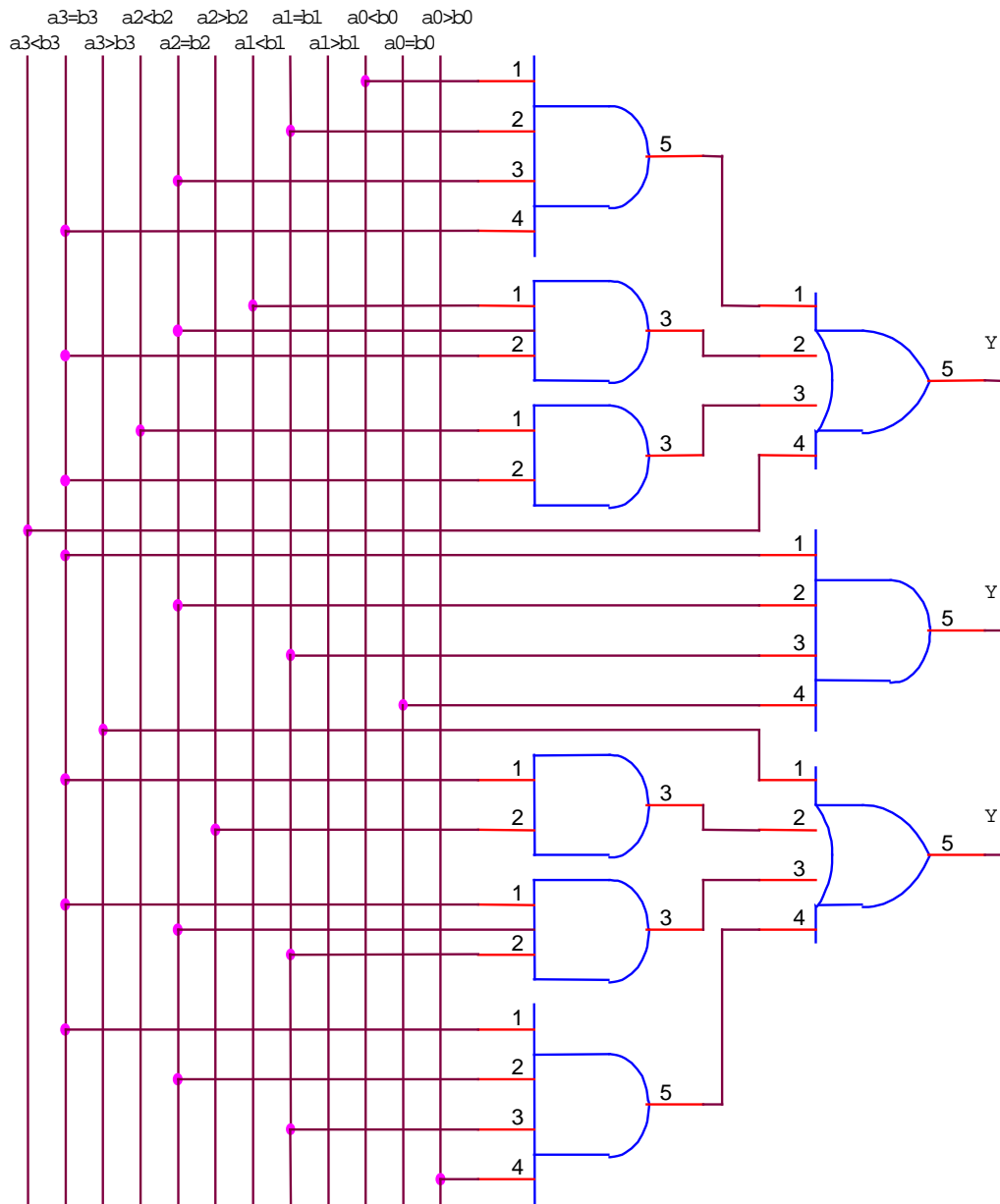
$$Y_2 = (A = B)$$

$$= (a_3 = b_3)(a_2 = b_2)(a_1 = b_1)(a_0 = b_0)$$

$$Y_3 = (A > B)$$

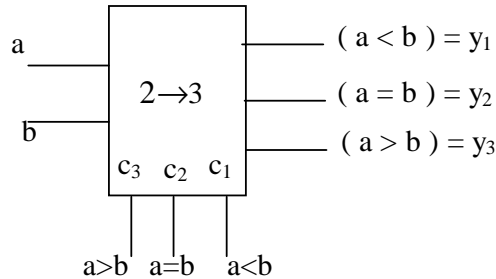
$$= (a_3 > b_3) + (a_3 = b_3)(a_2 > b_2) + (a_3 = b_3)(a_2 = b_2)(a_1 > b_1) \\ + (a_3 = b_3)(a_2 = b_2)(a_1 = b_1)(a_0 > b_0).$$

Sơ đồ mạch thực hiện trên hình 4.33.



Hình 4.33. Thực hiện mạch so sánh nhiều bit theo cách trực tiếp

## 2. Phương pháp xây dựng trên cơ sở mạch so sánh 1 bit



Hình 4.34. Mạch so sánh 1 bit cải tiến

Để mạch so sánh hai số nhị phân 1 bit có thể thực hiện công việc xây dựng mạch so sánh hai số nhị phân nhiều bit ta cải tiến lại mạch so sánh 1 bit như sau: ngoài các ngõ vào và ngõ ra giống như mạch so sánh 1 bit ta đã khảo sát ở trên, còn có các ngõ vào điều khiển  $a < b$ ,  $a > b$ ,  $a = b$ , với sơ đồ mạch như sau :

Bảng trạng thái mô tả hoạt động của mạch so sánh nhị phân 1 bit đầy đủ như sau:

Ngõ vào điều khiển			Ngõ vào DATA		Ngõ ra		
$a < b$	$a = b$	$a > b$	a	b	$(a < b)$	$(a = b)$	$(a > b)$
1	0	0	x	x	1	0	0
0	0	1	x	x	0	0	1
0	1	0	0	0	0	1	0
0	1	0	0	1	1	0	0
0	1	0	1	0	0	0	1
0	1	0	1	1	0	1	0

Phương trình logic:

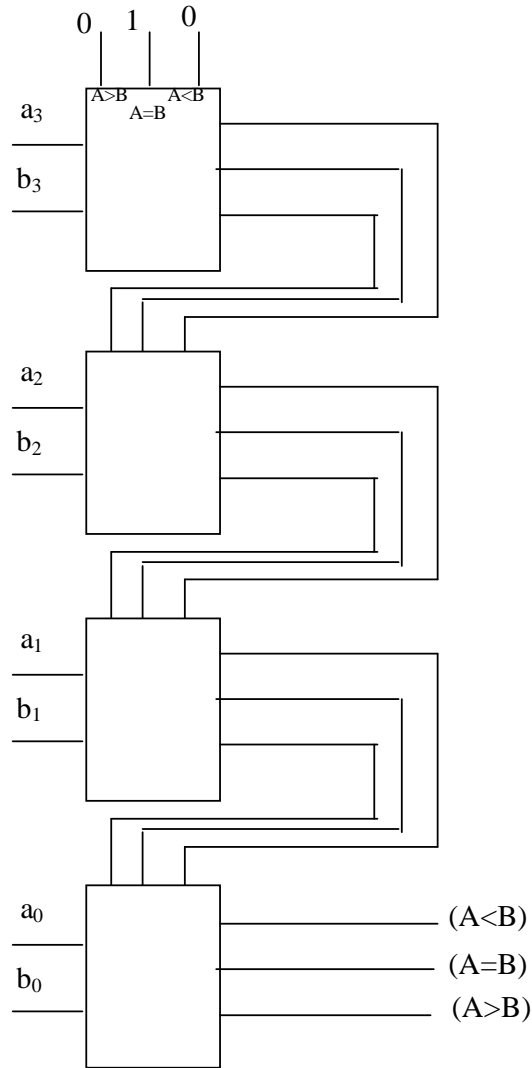
$$y_1 = (a < b) = c_1 + c_2(\bar{a} b).$$

$$y_2 = (a = b) = c_2(a \oplus b).$$

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

Dựa vào vi mạch so sánh đầy đủ này, người ta thực hiện mạch so sánh hai số nhị phân 4 bit bằng cách sử dụng các vi mạch so sánh 1 bit đầy đủ này giữa  $a_3$  với  $b_3$ ,  $a_2$  với  $b_2$ ,  $a_1$  với  $b_1$ ,  $a_0$  với  $b_0$  với cách nối theo sơ đồ như trên hình 4.35.

Lưu ý đối với mạch trên hình 4.35: mạch có 3 ngõ vào điều khiển  $(A > B)$ ,  $(A = B)$ ,  $(A < B)$  nên để mạch làm việc được thì bắt buộc cho ngõ vào điều khiển  $(A = B) = 1$  (tức là xem như  $a_4$ ,  $a_4$  trở về trước bằng nhau, nếu  $a_4 > a_4$  thì ngõ ra  $A > B$ ).



Hình 4.35. Mạch so sánh nhiều bit

## 4.5. MẠCH SỐ HỌC

### 4.5.1. Đại cương

Mạch số học là mạch có chức năng thực hiện các phép toán số học  $+$ ,  $-$ ,  $\times$ ,  $/$  các số nhị phân. Đây là cơ sở để xây dựng đơn vị luận lý và số học (ALU) trong  $\mu$ p (micro Processor) hoặc CPU (Centre Processing Unit).

### 4.5.2. Bộ cộng (Adder)

#### 1. Bộ bán tổng (HA-Half Adder)

Bộ bán tổng thực hiện cộng 2 số nhị phân một bit.

Quy tắc cộng như sau:

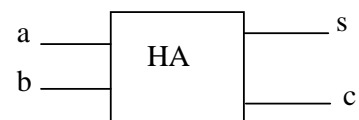
$$0 + 0 = 0 \text{ nhớ } 0$$

$$0 + 1 = 1 \text{ nhớ } 0$$

$$1 + 0 = 1 \text{ nhớ } 0$$

$$1 + 1 = 0 \text{ nhớ } 1$$

$$(a) \quad (b) \quad (s) \quad (c)$$



Hình 4.36. Mạch cộng 1 bit



Trong đó a, b là số cộng, s là tổng, c là số nhớ.

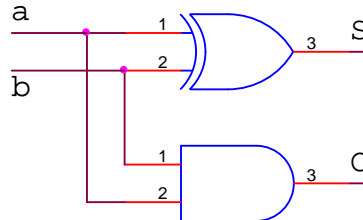
Bảng trạng thái mô tả hoạt động của mạch và phương trình logic:

$$s = a \cdot \bar{b} + \bar{a} \cdot b = a \oplus b$$

$$c = a \cdot b$$

Mạch cộng này chỉ cho phép cộng hai số nhị phân 1 bit mà không thực hiện cộng hai số nhị phân nhiều bit.

a	b	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

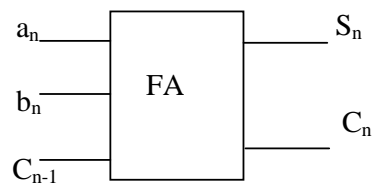


Hình 4.37. Sơ đồ mạch cộng bán phần

## 2. Bộ tổng (Bộ cộng toàn phần - FA: Full Adder)

Về phương diện mạch có sơ đồ khối như sau:

Trong đó:



$a_n$	$b_n$	$C_{n-1}$	$S_n$	$C_n$
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

Hình 4.38. Bộ cộng toàn phần

+  $C_{n-1}$  : Số nhớ của lần cộng trước đó.

+  $C_n$  : Số nhớ của lần cộng hiện tại.

+  $S_n$  : Tổng hiện tại.

Từ bảng trạng thái mô tả hoạt động của mạch ta viết được phương trình logic:

$$S_n = f(a_n, b_n, C_{n-1})$$

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

Lập bảng Karnaugh và tối thiểu hóa, ta có:

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

$$S_n = \bar{a}_n \bar{b}_n C_{n-1} + \bar{a}_n b_n \bar{C}_{n-1} + a_n \bar{b}_n \bar{C}_{n-1} + a_n b_n C_{n-1}$$

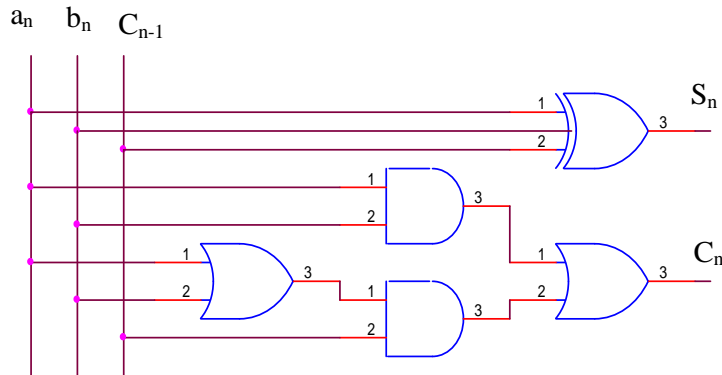
$$S_n = a_n \oplus b_n \oplus C_{n-1}$$

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

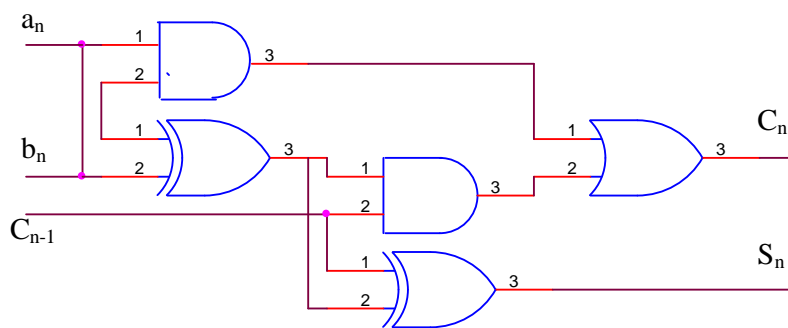
$$C_n = \bar{a}_n \bar{C}_{n-1} + b_n C_{n-1} + a_n b_n$$

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

Có thể thực hiện trực tiếp (sơ đồ 4.39) hoặc sử dụng HA để thực hiện FA (sơ đồ 4.40):



Hình 4.39. Mạch cộng toàn phần trực tiếp



Hình 4.40. Thực hiện mạch cộng toàn phần từ bộ bán tổng

### 4.5.3. Bộ trừ (Subtractor)

#### 1. Bộ bán trừ (Bộ trừ bán phần - HS: Half subtractor)

Bộ bán trừ thực hiện trừ 2 số nhị phân 1 bit.

Quy tắc trừ như sau:

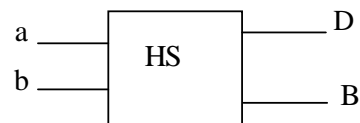
$$0 - 0 = 0 \text{ mượn } 0$$

$$0 - 1 = 1 \text{ mượn } 1$$

$$1 - 0 = 1 \text{ mượn } 0$$

$$1 - 1 = 0 \text{ mượn } 0$$

$$(a) \quad (b) \quad (D) \quad (B)$$

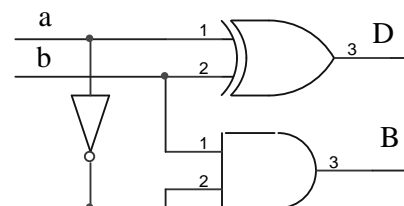


Hình 4.41 Mạch trừ bán phần

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

Bảng trạng thái mô tả hoạt động :

a	b	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0



Hình 4.42. Sơ đồ logic

Phương trình logic :

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

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

Mạch trừ này chỉ cho phép trừ hai số nhị phân 1 bit mà không thực hiện việc trừ hai số nhị phân nhiều bit.

## 2. Bộ trừ toàn phần (FS - Full Subtractor)

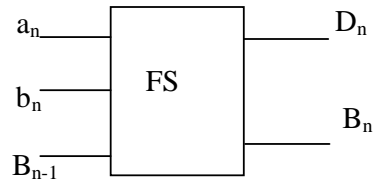
Mạch có sơ đồ khối và bảng trạng thái mô tả hoạt động như sau:

Trong đó:  $B_{n-1}$  : Số mượn của lần trừ trước đó.

$B_n$  : Số mượn của lần trừ hiện tại.

$D_n$  : Hiệu số hiện tại.

$a_n$	$b_n$	$B_{n-1}$	$D_n$	$B_n$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	0
1	1	0	0	0
0	0	1	1	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	1



Hình 4.43. Mạch trừ toàn phần

Lập bảng Karnaugh và tối thiểu hóa, ta có:

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

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

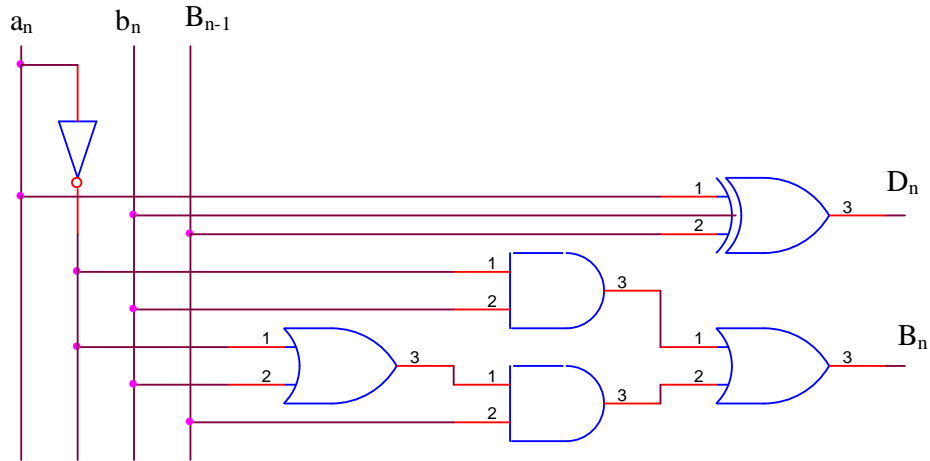
$$D_n = a_n \oplus b_n \oplus B_{n-1}$$

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

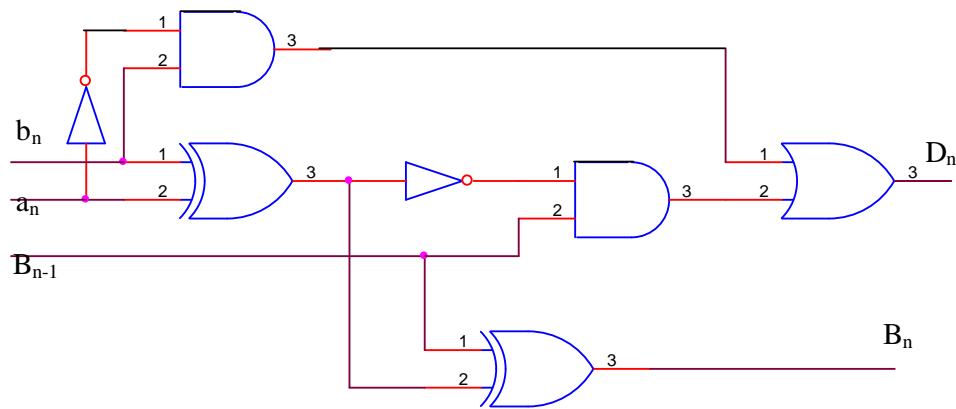
$$B_n = \overline{a_n} B_{n-1} + b_n B_{n-1} + \overline{a_n} b_n$$

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

Có 2 cách thực hiện bộ trừ toàn phần theo biểu thức logic đã tìm được: hoặc thực hiện trực tiếp (hình 4.44) hoặc sử dụng HS để thực hiện FS (hình 4.45).



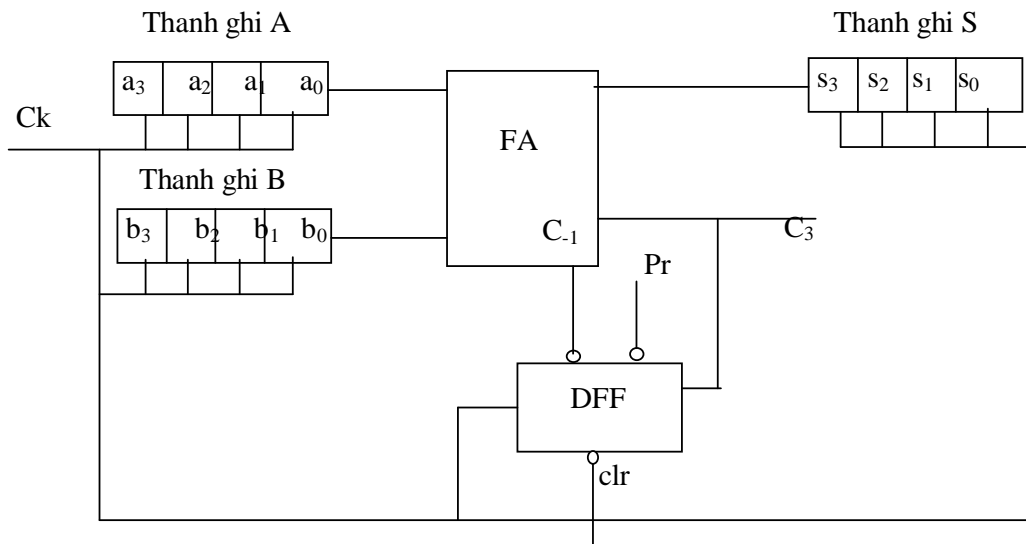
Hình 4.44. Thực hiện mạch trừ toàn phần trực tiếp



Hình 4.45. Thực hiện FS trên cơ sở HS

Từ bộ cộng toàn phần, ta xây dựng mạch cộng hai số nhị phân nhiều bit bằng 2 phương pháp: Nối tiếp và Song Song.

**Phương pháp nối tiếp:**



Hình 4.46. Mạch cộng 2 số nhị phân nhiều bit theo kiểu nối tiếp

Thanh ghi A chứa số A :  $a_3, a_2, a_1, a_0$

Thanh ghi B chứa số B :  $b_3, b_2, b_1, b_0$

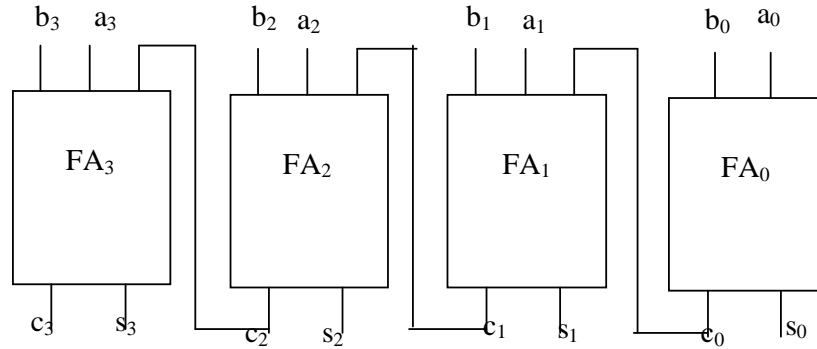
Thanh ghi S chứa số S :  $s_3, s_2, s_1, s_0$

Nhược điểm của phương pháp này là thời gian thực hiện lâu.

#### **Phương pháp song song:**

Để khắc phục nhược điểm đó, người ta dùng phương pháp cộng song song (hình 4.47).

Do tín hiệu điều khiển Ck (điều khiển cộng) đồng thời nên thời gian thực hiện phép cộng nhanh hơn phương pháp nối tiếp, song do số nhớ vẫn phải chuyển nối tiếp nên ảnh hưởng tốc độ xử lý.



Hình 4.47. Mạch cộng song song, số nhớ chuyển nối tiếp

#### **Mạch cộng nhớ nhanh - Mạch cộng với số nhớ nhìn thấy trước:**

Người ta cải tiến mạch trên thành mạch cộng song song với số nhớ nhìn thấy trước còn gọi là mạch cộng nhớ nhanh (Fast Carry, Carry Look Ahead). Bằng cách dựa vào sự phân tích mạch cộng toàn phần như sau:

Ta có:

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

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

Ta ââu:

$$P_n = a_n \oplus b_n$$

$$G_n = a_n \cdot b_n$$

Suy ra:

$$S_n = P_n \oplus C_{n-1}$$

$$C_n = G_n + P_n \cdot C_{n-1}$$

Khi  $n=0$  (LSB):

$$S_0 = P_0 \oplus C_{-1}$$

$$C_0 = G_0 + P_0 \cdot C_{-1}$$

Khi  $n=1$ :

$$S_1 = P_1 \oplus C_0 = P_1 \oplus (G_0 + P_0 \cdot C_{-1})$$

$$C_1 = G_1 + P_1 \cdot C_0 = G_1 + P_1 \cdot (G_0 + P_0 \cdot C_{-1})$$

Khi  $n=2$ :

$$S_2 = P_2 \oplus C_1 = P_2 \oplus [G_1 + P_1 \cdot (G_0 + P_0 \cdot C_{-1})]$$

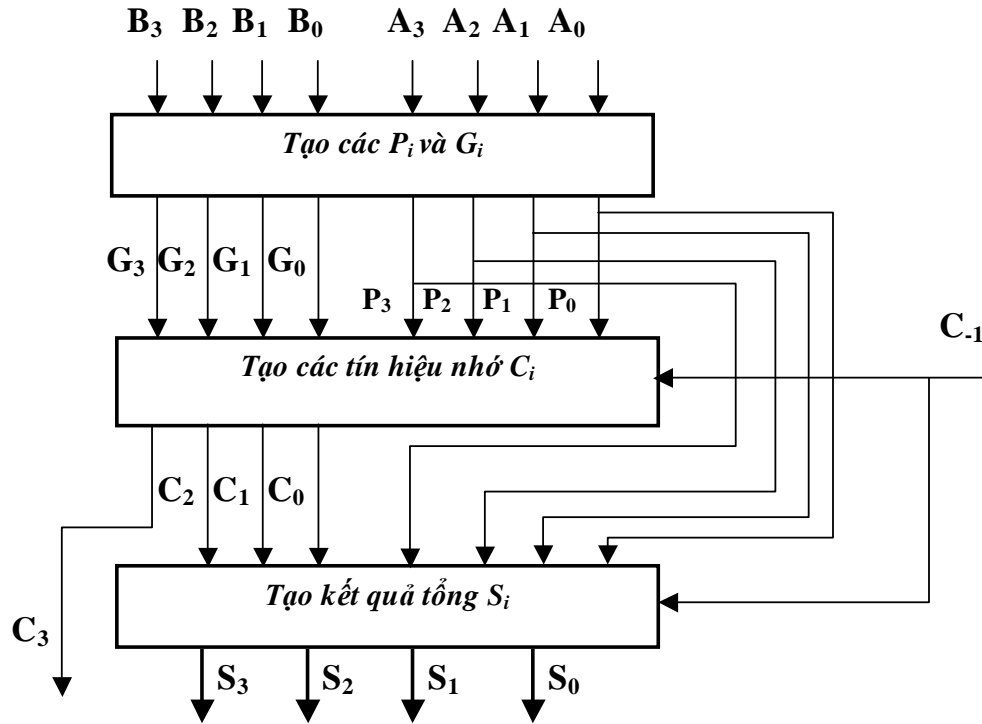
$$C_2 = G_2 + P_2 \cdot C_1 = G_2 + P_2 \cdot [G_1 + P_1 \cdot (G_0 + P_0 \cdot C_{-1})]$$

Khi  $n=3$ :

$$S_3 = P_3 \oplus C_2 = P_3 \oplus \{G_2 + P_2 \cdot [G_1 + P_1 \cdot (G_0 + P_0 \cdot C_{-1})]\}$$

$$C_3 = G_3 + P_3 . C_2 = G_3 + P_3 . \{ G_2 + P_2 . [ G_1 + P_1 . ( G_0 + P_0 . C_{-1} ) ] \}$$

Đây chính là cơ sở tính toán để tạo ra số nhớ C1, C2, C3 và S3 tùy thuộc vào an, bn. Sơ đồ khối mạch cộng song song 4 bit nhớ nhanh được cho trên hình 4.48



Hình 4.48. Sơ đồ mạch cộng song song 4 bit nhớ nhanh

Trên thực tế người ta đã chế tạo ra các vi mạch cộng nhớ nhanh, ví dụ: IC 7483.

