

## **CHƯƠNG 5**

# **CÁC HỆ LOGIC TỔ HỢP**

### **5.1. KHÁI NIỆM**

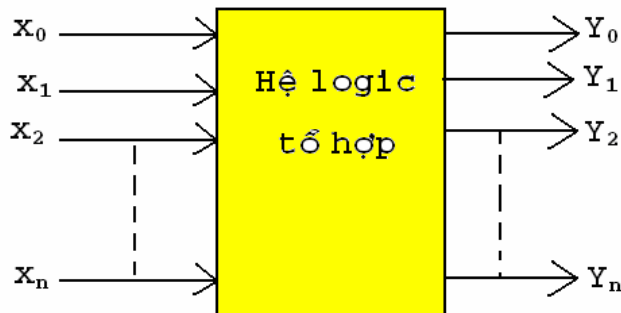
Trong các thiết bị số, sự gia công và biến đổi thông tin được thực hiện nhờ các hệ thống mạch logic tổ hợp (combinational logic) mà một số chức năng chính của chúng như sau:

Trong các khâu biến đổi thông tin, các hệ logic tổ hợp có thể thực hiện chức năng chuyển đổi từ mã số này sang mã số khác hoặc chuyển đổi mã từ dạng song song sang dạng nối tiếp và ngược lại...

Trong các khâu gia công thông tin, các hệ logic tổ hợp có thể thực hiện nhiều chức năng khác nhau như giải mã, chọn kênh, phân kênh, hoặc thực hiện các phép tính số học trên cơ sở các mã số khác nhau...

Trong thực tế kỹ thuật, các hệ logic tổ hợp thường được sử dụng phối hợp với các thiết bị số khác, hoặc các thiết bị tương tự để tạo ra các thiết bị hệ lai nhiều chức năng. Các hệ logic tổ hợp chỉ sử dụng chủ yếu các cổng logic. Trong khâu biến đổi thông tin, các hệ logic tổ hợp có thể chuyển đổi từ mã hệ này sang mã số hệ khác.

Nó có sơ đồ khối tổng quát như hình 5.1



**Hình 5.1:** Sơ đồ khối tổng quát hệ logic tổ hợp

Hệ logic tổ hợp có  $n$  lối vào  $m$  lối ra. Mỗi lối ra là một hàm của các biến đầu vào theo phương trình tổng quát:

$$Y_0 = f_1(x_0, x_1, \dots, x_n)$$

$$Y_1 = f_2(x_0, x_1, \dots, x_n)$$

.....

$$Y_2 = f_3(x_0, x_1, \dots, x_n)$$

Như vậy, hàm ra của hệ logic tổ hợp chỉ phụ thuộc các biến đầu vào mà không phụ thuộc vào các trạng thái trước đó của hệ, nghĩa là trạng thái đầu ra được thiết lập ngay sau khi có tác động ở đầu vào.

Hệ logic tổ hợp được sử dụng nhiều trong hệ thống số. Nó sẽ thực hiện những nhiệm vụ mà kết quả đầu ra chỉ là tổ hợp của các trạng thái đầu vào, như lập mã, giải mã, chuyển đổi mã... Trong chương này ta phân tích và nghiên cứu hoạt động của các ứng dụng đó.

## 5.2. MÃ HOÁ

### 5.2.1 Khái niệm:

Để xử lý, chế biến các thông tin trong hệ thống số, ta chỉ có được các bit 1 và 0. Điều này làm cho chúng ta gặp nhiều khó khăn khi đọc và hiểu một chuỗi dài và rất dài các con số 1,0 đó. Vì vậy, thông tin đưa đến từ các dụng cụ biến đổi (Bàn phím, chuột ...) phải được mã hóa. Sau khi máy xử lý xong ta lại phải giải mã trước khi đưa ra chỉ thị. Việc này gọi là biến đổi mã, nó là phương tiện để chuyển đổi giữa ngôn ngữ thường ngày ra ngôn ngữ máy.

Ta lấy ví dụ với sơ đồ khối một máy tính cầm tay (Hình 5.2)



**Hình 5.2:** Sơ đồ khối một máy tính cầm tay

Các số thập phân (từ 0 đến 9) được đánh vào từ bàn phím sẽ được bộ mã hóa biến thành các số nhị phân. Khối đơn vị tính toán tiến hành xử lý để cho một kết quả nhị phân. Khối giải mã chuyển số nhị phân thành số thập phân và đưa đến bộ chỉ thị hiện số bảy đoạn.

Các bộ lập mã và giải mã là những bộ dịch mã điện tử, bộ mã hóa dịch ngôn ngữ thường ngày ra ngôn ngữ máy, bộ giải mã thì ngược lại.

Để thực hiện việc mã hóa và giải mã ta dùng các hệ thống mạch logic tổ hợp.

Chúng ta biết rằng, có thể sử dụng một nhóm mã hệ nhị phân có 4 chữ số để biểu diễn các con số hệ thập phân từ 0 đến 9, quá trình này được gọi là mã hóa. Ta cũng có thể mã hóa các chữ cái (A,B,C ..), các ký hiệu đặc biệt (&,%,#...) hoặc cả một tập lệnh của máy tính số ..

Với một chuỗi các số hệ nhị phân có k chữ số ta sẽ có  $2^k$  bộ giá trị khác nhau có thể đại diện cho  $2^k$  ký hiệu hoặc lệnh cần mã hóa. Như vậy, nếu số ký hiệu hoặc số lệnh cần được mã hóa là N thì ta phải có :

$$N \leq 2^k$$

Có 2 trường hợp xảy ra :

\* $N = 2^k$ : Số bộ giá trị của nhóm mã gồm k chữ số vừa đủ để biểu diễn N ký hiệu .

\* $N < 2^k$ : Số bộ giá trị của nhóm mã gồm k chữ số nhiều hơn số ký hiệu hoặc số lệnh cần biểu diễn. Như vậy sẽ thừa một số giá trị không dùng đến gọi là số tổ hợp thừa, các tổ hợp thừa này có thể được dùng vào các mục đích khác nhau như phát hiện và sửa sai trong quá trình truyền thông tin. Trong một số trường hợp ta phải có biện pháp sửa sai khi các tổ hợp thừa này xuất hiện trong các kết quả tính toán.

### 5.2.2. Bộ mã hóa thập phân- BCD77

Bộ mã hóa thập phân ra mã BCD có nhiệm vụ biến đổi (dịch) tín hiệu hệ đếm cơ số 10 sang mã BCD 8421.

Dùng 4 chữ số hệ nhị phân (ký hiệu là  $A_3A_2A_1A_0$ ) để mã hóa các số từ 0 đến 9 của hệ

mười theo mã BCD ta có :

Số hệ 10	$A_3A_2A_1A_0$
0	0 0 0 0
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1

Như vậy, một chữ số của mã BCD có thể xuất hiện nhiều lần dưới dạng 1 (hoặc 0) tại cùng một vị trí ứng với nhiều số khác nhau. Ta thấy chữ số ở vị trí B bằng 1 đối với các số 2,3,6,7 hệ 10. Nhận xét này cho ta thấy rằng mạch mã hóa này có đặc tính của hàm OR và ta có thể viết (với dấu + là ký hiệu phép cộng logic):

$$A_1 = 2 + 3 + 6 + 7$$

Vậy, muốn biến đổi các số thập phân từ 0 đến 9 sang mã BCD 8421 ta dùng 4 phân tử OR như sau :

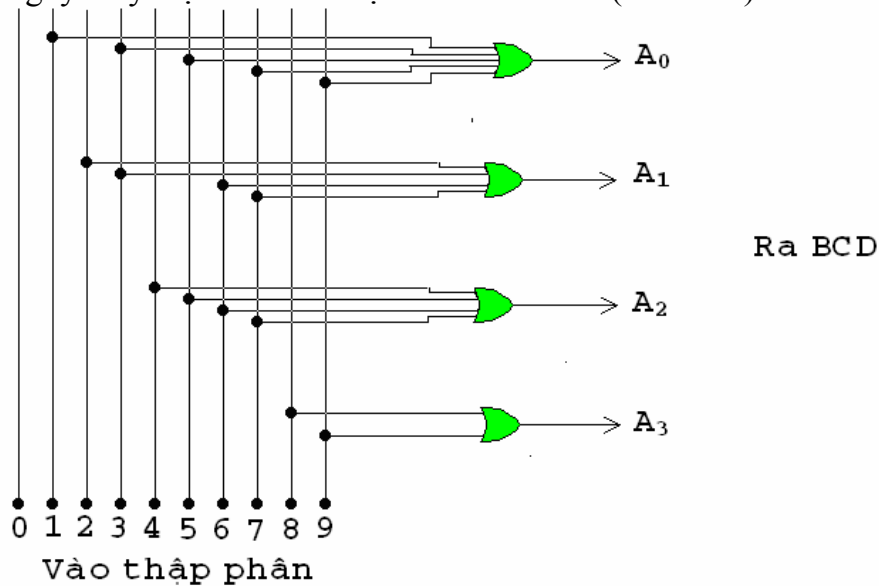
$$\text{Bit : } A_0 = 1 + 3 + 5 + 7 + 9$$

$$A_1 = 2 + 3 + 6 + 7$$

$$A_2 = 4 + 5 + 6 + 7$$

$$A_3 = 8 + 9$$

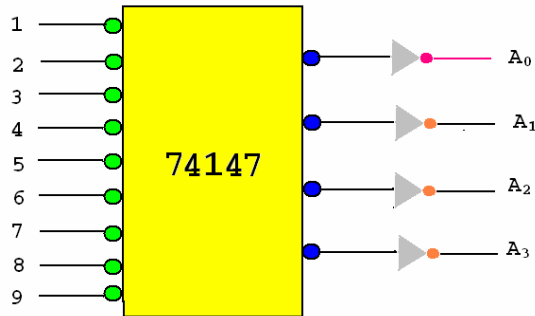
Ta có sơ đồ nguyên lý mạch mã hóa hệ 10 - BCD 8421 (Hình 5.3)



Hình 5.3: Sơ đồ nguyên lý bộ mã hóa thập phân sang BCD

Giả sử cần mã hoá số 3. Ta ấn phím số 3, một điện thế mức cao (+U) được cung cấp cho đường số 3 còn các đường khác không được cung cấp nên ở mức thấp (0V). Các đường ra A và B có điện thế ở mức cao còn các đường C và D vẫn ở mức điện thế thấp, nghĩa là ta có 0011 ở ngõ ra.

Thực tế, mạch mã hóa này được chế tạo thành một vi mạch, nó có sơ đồ khối tổng quát hoạt động như hình 5.4.



**Hình 5.4:** Sơ đồ khối tổng quát bộ lập mã dùng vi mạch

Khi có một đầu vào tác động, bộ lập mã tạo ra một từ nhị phân tương ứng ở lối ra và chỉ một mà thôi. Đối với bộ mã hóa dạng ưu tiên, khi có nhiều đầu vào tác động đồng thời thì từ logic ở lối ra sẽ ứng với đầu vào có số hạng cao nhất. Bộ mã hóa trên chỉ có 9 đầu vào từ 1 đến 9 (không có số 0), tác dụng ở mức thấp tức là mức 0, biểu thị ở các vòng tròn ở lối vào. Bình thường chúng ở mức cao vì lối vào để lơ lửng. Lúc đó ngõ ra sẽ ứng với số 0 trong hệ đếm 10.

Ngõ ra bình thường ở mức cao (hoặc 1 logic), khi hoạt động thì ở mức thấp (hoặc 0 logic). Các bộ đảo ở lối ra để đưa các ngõ ra trở lại dạng thông thường. Như vậy, khi cả 9 đầu vào không có tín hiệu gì cả (ứng với tất cả ngõ vào đều là 1), các ngõ ra ở mức cao  $A_3A_2A_1A_0$  đều ở mức 1. Nghĩa là số 0 ứng với ngõ ra là 1111 nhưng khi qua các bộ đảo sẽ thành 0000 phù hợp với dạng thông thường. Ta có bảng trạng thái của bộ mã hóa trên.

Vào									Ra			
1	2	3	4	5	6	7	8	9	$A_3$	$A_2$	$A_1$	$A_0$
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	B	B	H	H	B
X	X	X	X	X	X	X	B	H	B	H	H	H
X	X	X	X	X	X	B	H	H	H	B	B	B
X	X	X	X	X	B	H	H	H	H	B	B	H
X	X	X	X	B	H	H	H	H	H	B	H	B
X	X	X	B	H	H	H	H	H	H	H	B	B
X	X	B	H	H	H	H	H	H	H	H	B	H
B	H	H	H	H	H	H	H	H	H	H	H	B

*Chú thích:* B: mức thấp (Bass), H: mức cao (Haut), X: Chỉ tín hiệu vào không tương thích

Để mạch hoạt động ta đưa vào đầu vào các xung vuông. Bộ mã hóa có đặc điểm là được tác động của số lớn nhất có tín hiệu vào thấp (B). Ví dụ nếu có một B ở 5 và 9 tín hiệu ra sẽ là 1001 tương ứng với số thập phân 9.

### 5.2.3. Bộ mã hóa bát phân – nhị phân

Với cách làm tương tự ta có thể thiết kế bộ mã hóa bát phân – nhị phân

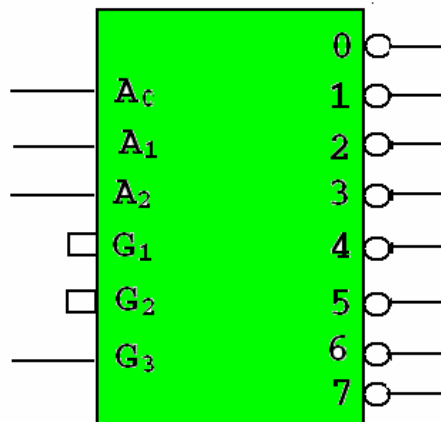
## 5.3. GIẢI MÃ

### 5.3.1 Khái niệm

Giải mã là quá trình ngược lại quá trình mã hóa. Nghĩa là từ một bộ giá trị của các nhóm mã k chữ số hệ 2 ta tìm lại được 1 trong N ký hiệu hoặc lệnh tương ứng.

Trong bộ giải mã chỉ có một đầu ra có giá trị logic 1 ứng với một tổ hợp biến đầu vào nào đó, còn các đầu ra khác có giá trị logic 0. Đầu ra có giá trị 1 này sẽ kích hoạt bộ hiển thị hoạt động, báo cho chúng ta biết kết quả ra.

Đầu ra bộ giải mã là 2, 3 hoặc 4 bit. Thí dụ bộ giải mã nhị - bát phân, có mã vào 3 bit và  $2^3 = 8$  tổ hợp ở đầu ra cho 8 đầu ra từ 0 đến 7 (Hình 5.5)



**Hình 5.5:** Sơ đồ khối tổng quát bộ giải mã nhị - bát phân 3 bit

**Thí dụ:** Đầu vào: 100 (số 4) thì đầu ra 4 sẽ có trạng thái 0 (trạng thái tác dụng), mọi đầu ra khác đều ở trạng thái 1.

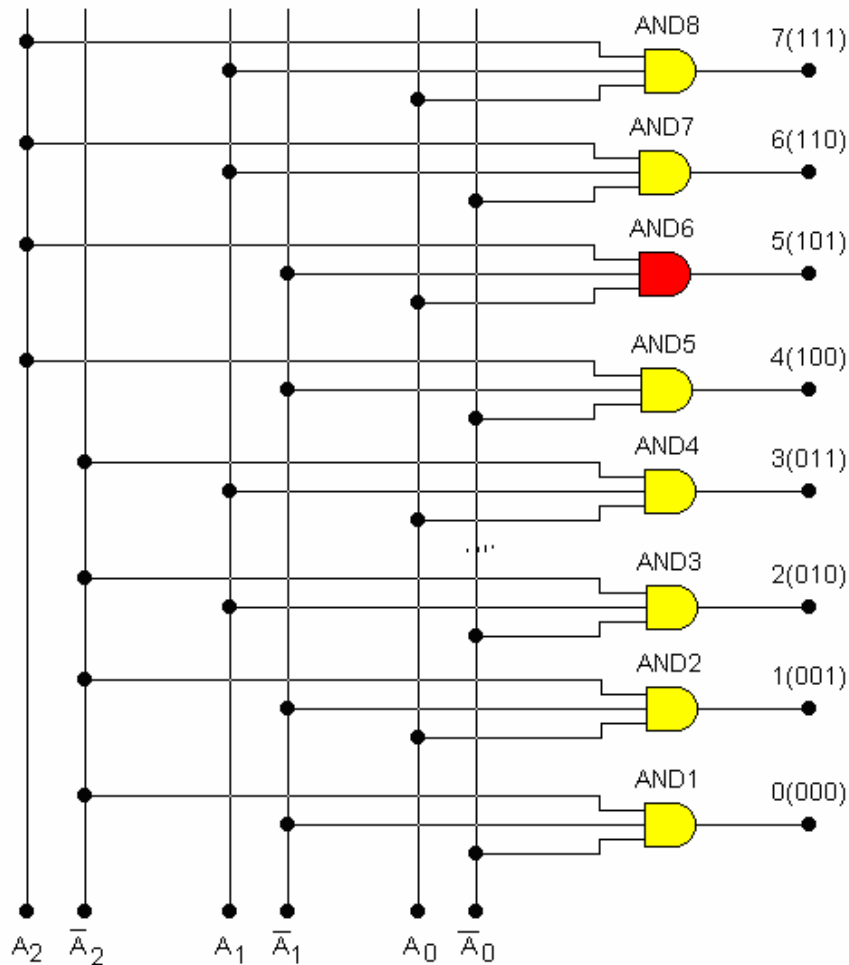
### 5.3.2. Bộ giải mã nhị-bát phân

Bộ giải mã này có 3 biến đầu vào, 8 đầu ra và có bảng trạng thái:

Đầu vào	Đầu ra
$A_2 A_1 A_0$	0 1 2 3 4 5 6 7
0 0 0	1 0 0 0 0 0 0 0
0 0 1	0 1 0 0 0 0 0 0
0 1 0	0 0 1 0 0 0 0 0
0 1 1	0 0 0 1 0 0 0 0
1 0 0	0 0 0 0 1 0 0 0

1 0 1	0 0 0 0 0 1 0 0
1 1 0	0 0 0 0 0 0 1 0
1 1 1	0 0 0 0 0 0 0 1

8 đầu ra, vào mỗi thời điểm chỉ có một đầu ở mức 1, còn các đầu ra còn lại ở mức 0.  
 Từ bảng trạng thái ta thiết lập sơ đồ logic của bộ giải mã trên như sau (Hình 5.6)



**Hình 5.6:** Sơ đồ logic bộ giải mã nhị - bát phân

Khi sử dụng ta đưa điện áp vào đầu vào tương ứng để có mức 1.

*Thí dụ:* Máy tính thực hiện xong phép tính và đưa kết quả vào bộ giải mã, giả sử điện áp vào  $A_2 \bar{A}_1 A_0$ , lúc đó đầu ra của cổng AND 6 ở mức 1 còn các đầu ra các cổng AND khác ở mức 0, mức 1 này sẽ kích hoạt bộ chỉ thị làm hiển thị số 5.

### 5.3.3. Bộ giải mã BCD ra thập phân

Bộ giải mã BCD ra thập phân có nhiệm vụ chuyển mã BCD ở ngõ vào thành số thập phân tương ứng ở ngõ ra. Trong mỗi thời điểm chỉ có một đầu ra duy nhất hoạt động. Các đầu ra có

lắp các thiết bị chỉ thị (LCD , LED ...) để chỉ thị các kết quả ở ngõ ra. Các đầu ra đều hoạt động ở mức thấp, đầu ra ở trạng thái 0 là đầu ra có tác dụng (bình thường ở mức cao) vì vậy khi dùng ta phải đấu thêm các bộ đảo ở ngõ ra để điều khiển các chỉ thị thập phân thấp sáng .

$N = 4$  nên bảng trạng thái có 16 hàng, chỉ sử dụng 10 hàng, 6 hàng còn lại không hiệu lực. Ta có bảng trạng thái sau:

	Đầu vào BCD	Đầu ra thập phân
Số	C D B A	0 1 2 3 4 5 6 7 8 9
0	B B B B	B H H H H H H H H H
1	B B B H	H B H H H H H H H H
2	B B H B	H H B H H H H H H H
3	B B H H	H H H B H H H H H H
4	B H B B	H H H H B H H H H H
5	B H B H	H H H H H B H H H H
6	B H H B	H H H H H H B H H H
7	B H H H	H H H H H H H B H H
8	H B B B	H H H H H H H H B H
9	H B B H	H H H H H H H H H B
Không hiệu lực	H B H B	H H H H H H H H H H
	H B H H	H H H H H H H H H H
	H H B B	H H H H H H H H H H
	H H B H	H H H H H H H H H H
	H H H B	H H H H H H H H H H
	H H H H	H H H H H H H H H H

#### 5.4. CÁC HỆ CHUYỂN ĐỔI MÃ

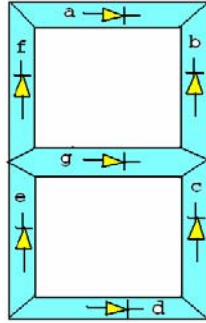
Các hệ chuyển đổi mã có nhiệm vụ chuyển đổi từ một mã này sang một mã khác. Đầu vào là các mã nhị phân, đầu ra cũng là các mã nhị phân thuộc một mã khác.

*Thí dụ:* bộ chuyển đổi nhị phân sang Gray, đầu vào là các số nhị phân, đầu ra cũng là các số nhị phân. Trong thực tế ta sử dụng nhiều bộ chuyển đổi mã khác nhau, sau đây là các bộ chuyển đổi mã thường gặp nhất .

##### 5.4.1. Bộ chuyển đổi mã BCD ra mã 7 thanh (seven segment)

Sau khi tính toán các mã nhị phân BCD cần phải được biến đổi (dịch) ra số thập phân. Thiết bị dịch ngôn ngữ máy ra số thập phân là một thiết bị rất thông dụng trong mạch số. Trong kỹ thuật để thực hiện việc hiển thị các số từ 0 đến 9 người ta dùng một tập hợp gồm 7 thanh phát sáng. (Hình 5.7).

Mỗi thanh phát sáng này có thể được điều khiển để được bật tắt một cách độc lập. Tùy theo số đoạn được bật sáng mà ta có được các chữ số từ 0 đến 9.



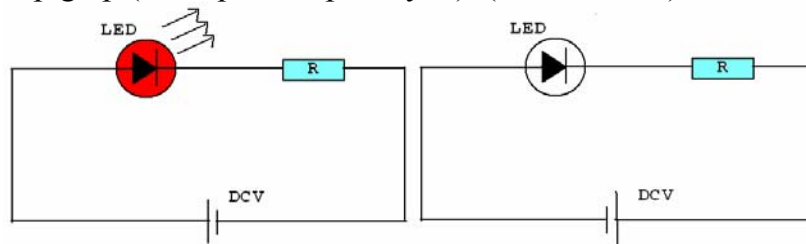
**Hình 5.7: Đèn hiển thị 7 thanh dùng đèn LED**

Để chế tạo đèn 7 thanh ta có thể dùng nhiều phương pháp khác nhau, trong đó có 2 phương pháp được sử dụng rộng rãi nhất:

**a/Phương pháp dùng L.E.D ( Light Emetting Diode ) :**

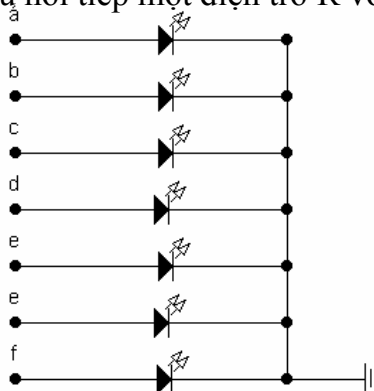
LED hay còn gọi là Diode phát quang là một loại diode được chế tạo từ hợp chất bán dẫn Gali-Axênic hoặc Gali-Phospho. LED sẽ phát sáng khi phân cực thuận.

Tùy theo chất chế tạo LED phát ra các màu khác nhau (đỏ , xanh lá cây ...). Ánh sáng phát ra, do hiện tượng kết hợp của điện tử và lỗ trống, có cường độ sáng thay đổi theo cường độ dòng điện qua tiếp giáp (theo quan hệ phi tuyến). (Hình 5.8 a,b).



**Hình 5.8 a: Phân cực thuận LED sáng Hình 5.8b: Phân cực nghịch LED tắt**

Vì kích thước rất nhỏ nên với phương pháp này ta có thể chế tạo các đèn 7 thanh (là 7 diode) với kích thước rất nhỏ. Các diode trong đèn thường có anode chung hoặc cathode chung để giảm nhỏ số dây nối ra ngoài. Hình 5.9 cho ta sơ đồ một đèn 7 thanh dùng LED có các cathode đấu chung và nối đất, vì vậy chân nào của LED có mức logic dương LED đó sẽ sáng lên. Để bảo vệ các diode, ta nên đấu nối tiếp một điện trở R với mỗi LED.



**Hình 5.9: Đèn 7 thanh dùng LED cathode chung**

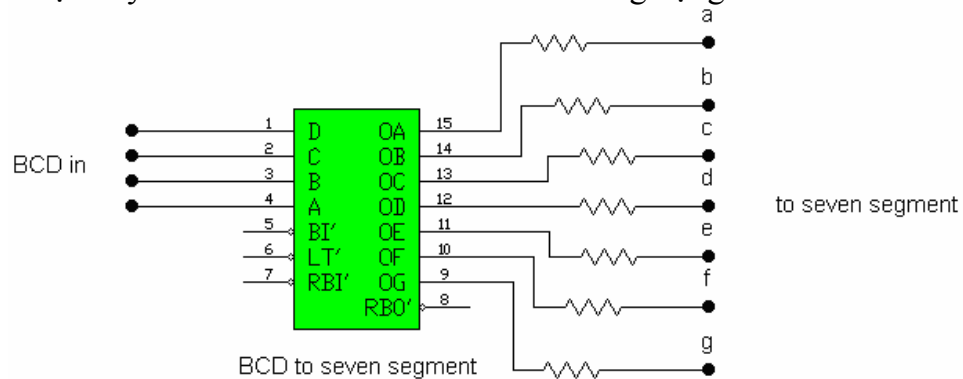


***b/Phương pháp dùng LCD ( Liquid Crystal Display )***

Dùng LED tuy tiêu hao năng lượng đã nhỏ nhưng vẫn còn khá cao (gần 20 mA cho mỗi LED), nên hiện nay người ta dùng loại tinh thể lỏng LCD. Tinh thể lỏng LCD gồm 2 miếng thủy tinh xếp song song với nhau, ở giữa là dây dẫn và hỗn hợp tinh thể lỏng được tạo thành hình dáng cần thiết (dạng điểm hoặc dạng thanh).

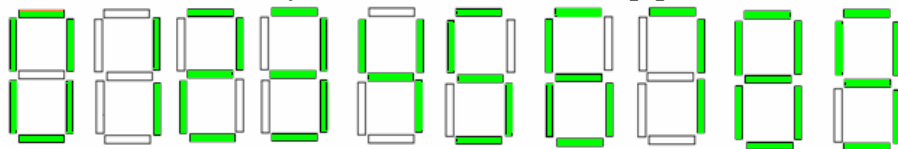
Khi có tác dụng của điện trường, hỗn hợp tinh thể lỏng sẽ đổi màu khiến cho mắt ta có thể trông thấy được, Hiện thị kiểu này tiêu thụ rất ít năng lượng cho nên được dùng rất phổ biến. Chúng có thể được sử dụng để chế tạo thành những màn hình lớn chứa hàng triệu điểm cũng như các đèn 7 thanh như đèn LED. Khác với đèn LED khi sử dụng LCD cần có một tín hiệu “mặt sau” (backplane) dạng sóng vuông với tần số từ 30 Hz đến 200 Hz .

Hình 5.10 là bộ chuyển đổi mã BCD ra mã 7 thanh thông dụng .



***Hình 5.10 : Bộ chuyển đổi mã BCD - 7 thanh***

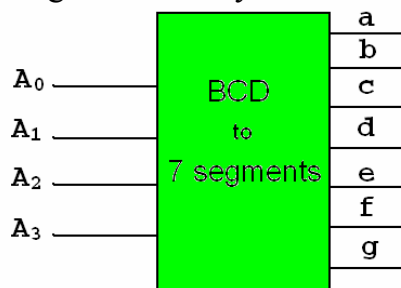
Bộ chuyển đổi mã trên là vi mạch 7447A. Đầu vào là BCD 4 bit và 3 đầu vào phụ, trong đó LT (Lamp Test) là đầu vào thử đèn, LT sẽ cho các thanh vào mạch làm việc để xem sự hoạt động của đèn. Hình 5.11 cho ta thấy việc hiển thị các số thập phân từ 0 đến 9 dùng 7 đoạn.



***Hình 5.11: Hiển thị các số thập phân từ 0 – 9 dùng dụng cụ 7 đoạn***

***c/Bộ chuyển đổi mã BCD sang 7 đoạn:***

Bộ chuyển đổi mã BCD sang 7 đoạn có ký hiệu như hình 5.12



***Hình 5.12: Ký hiệu bộ chuyển đổi mã BCD – 7 đoạn***

Theo hình 5.12, ta thấy một đoạn sẽ hiển thị cho nhiều chữ số.

*Thí dụ:* đoạn a sẽ hiển thị cho các số 0,2,3,5,7,8 hoặc 9. vì vậy, ta có thể viết cho cả 7 đoạn như sau:

$$a = \Sigma (0,2,3,5,6,7,8,9)$$

$$b = \Sigma (0,1,2,3,4,7,8,9)$$

$$c = \Sigma (0,1,3,4,5,6,7,8,9)$$

$$d = \Sigma(0,2,3,5,6,8,9)$$

$$e = \Sigma(0,2,6,8)$$

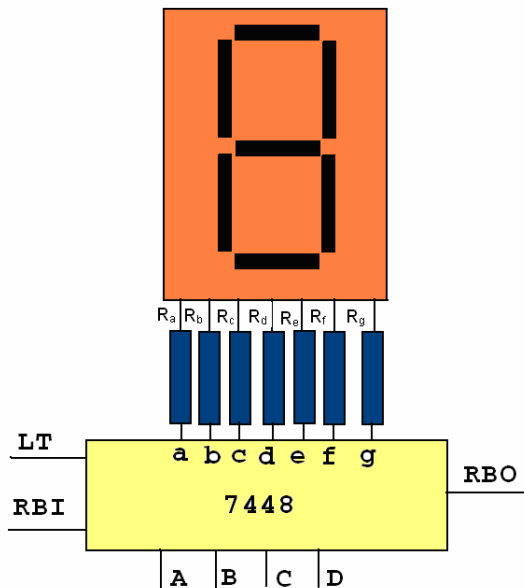
$$f = \Sigma(0,4,5,6,8,9)$$

$$g = \Sigma(2,3,4,5,6,8,9)$$

Để thực hiện việc chuyển đổi mã BCD sang 7 đoạn, ta dùng các vi mạch 7447, 74247 các LED được mắc anode chung, hoặc 7448, 4511 các LED được mắc Cathode chung. Mỗi IC sẽ điều khiển để LED hiển thị từ 0 đến 9. Các điện trở được thêm vào để nhằm bảo vệ các đoạn. Hình 5.13 cho ta một sơ đồ sử dụng vi mạch chuyển đổi 7448 .

Đầu vào RBI (Ripple Blanking Input): Đầu vào xóa nối tiếp.

Đầu vào LT (Light Test): Dùng để thử hoạt động của các đoạn



**Hình 5.13:** Mạch chuyển đổi mã BCD – 7 đoạn dùng IC 7448

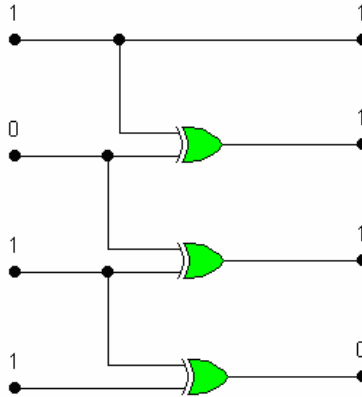
#### 5.4.2. Chuyển đổi mã hệ 2 – Gray

Chương 2 ta đã biết phương pháp lý thuyết chuyển đổi nhị phân sang mã Gray. *Thí dụ* chuyển số nhị phân 1011 sang mã Gray

$$\begin{array}{r} + 101 \\ 1011 \\ \hline 1110 \end{array}$$

Theo yêu cầu của phép chuyển đổi, ta thấy chỉ có cổng XOR mới đáp ứng việc chuyển đổi. Từ các cổng logic Exclusive OR (XOR), ta có thể xây dựng mạch chuyển đổi mã nhị phân sang mã GRAY. Nếu số nhị phân có n bit thì số cổng XOR sử dụng: n - 1.

Sơ đồ nguyên lý bộ chuyển đổi số nhị phân 1011 sang mã Gray ở hình 5.14



**Hình 5.14:** Mạch logic chuyển đổi nhị phân – Gray

Số nhị phân cần chuyển có 4 bit nên ta dùng 3 cổng XOR. Đầu vào cổng XOR trước nối vào đầu vào cổng XOR tiếp theo.

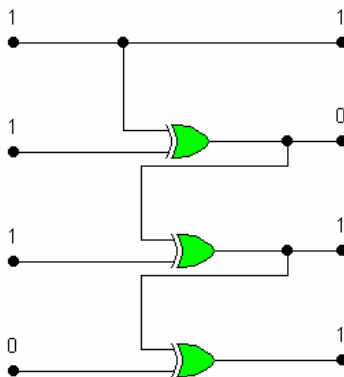
#### 5.4.3. Chuyển đổi mã Gray sang nhị phân

Từ phương pháp lý thuyết chuyển đổi mã Gray sang nhị phân. *Thí dụ:* chuyển mã Gray 1110 sang số nhị phân

$$\begin{array}{r} + 101 \\ \underline{1110} \\ 1011 \end{array}$$

Ta thấy cổng logic cần sử dụng chính là cổng XOR. Số lượng cổng XOR sử dụng: n - 1. Trong đó n là số bit sử dụng.

Hình 5.15 là sơ đồ nguyên lý bộ chuyển đổi số mã Gray 1110 sang số nhị phân



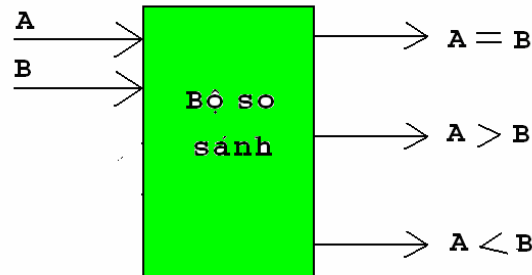
**Hình 5.15:** Mạch logic chuyển đổi Gray - nhị phân

Số nhị phân cần chuyển có 4 bit nên ta dùng 3 cổng XOR. Đầu ra cổng XOR trước nối vào đầu vào cổng XOR tiếp theo.

## 5.5. BỘ SO SÁNH:

### 5.5.1: Khái niệm:

Đây là một chức năng logic cho phép so sánh hai số nhị phân. Thí dụ A và B. Nó sẽ chỉ rõ ở đầu ra  $A = B$ ,  $A < B$  hoặc  $A > B$ . Hình 5.16 cho ta ký hiệu một bộ so sánh.



**Hình 5.16:** Ký hiệu bộ so sánh.

Một trong 3 lối ra sẽ lên 1 để kích hoạt bộ hiển thị hiển thị kết quả so sánh, hai đầu ra còn lại ở mức 0.

### 5.5.2. Bộ so sánh hai số nhị phân 1 bit

Muốn so sánh  $A = a_0$  và  $B = b_0$ , trước hết ta thiết lập bảng trạng thái cho bộ so sánh. Bảng trạng thái là cơ sở giúp ta thiết lập sơ đồ logic.

$b_0$	$a_0$	$S (A=B)$	$S_a (A>B)$	$S_b (A<B)$
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
1	1	1	0	0

Từ bảng trạng thái ta viết được các phương trình sau:

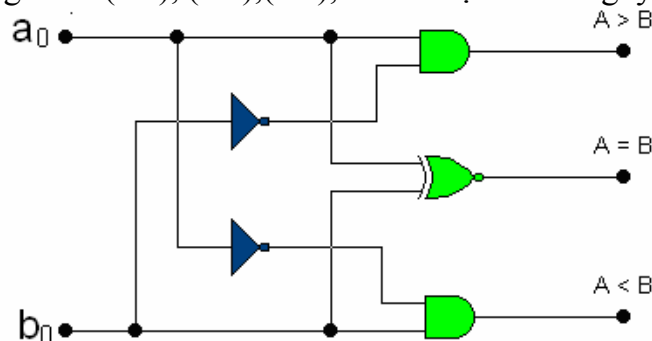
$$S = \bar{a}_0 \bar{b}_0 + a_0 b_0 = a_0 \oplus b_0 \quad (5.1)$$

$$S_a = a_0 \bar{b}_0 \quad (5.2)$$

$$S_b = \bar{a}_0 b_0 \quad (5.3)$$

$S = 1$  khi  $A = B$ ,  $S_a = 1$  khi  $A > B$ ,  $S_b = 1$  khi  $A < B$

Theo các phương trình (5.1), (5.2), (5.3), ta vẽ được sơ đồ nguyên lý của bộ so sánh 1 bit như hình 5.17



**Hình 5.17:** Sơ đồ logic bộ so sánh một bit

### 5.5.3 Bộ so sánh hai số nhị phân 2 bit

Trong trường hợp A và B mỗi số có 2 bit sẽ có 4 khả năng:

$$A = 00, 01, 10, 11$$

$$B = 00, 01, 10, 11$$

*a/ So sánh bằng:*

*Bảng trạng thái nếu  $A = B$*

$a_1$	$a_0$	$b_1$	$b_0$	Phương trình	Số thứ tự
0	0	0	0	$\bar{a}_1 \bar{a}_0 \bar{b}_1 \bar{b}_0$	1
0	1	0	1	$\bar{a}_1 a_0 \bar{b}_1 b_0$	2
1	0	1	0	$a_1 \bar{a}_0 b_1 \bar{b}_0$	3
1	1	1	1	$a_1 a_0 b_1 b_0$	4

*Phương trình nếu  $A = B$*

Gọi S là kết quả ra khi  $A = B$ , từ bảng trạng thái ta thiết lập các phương trình đầu ra:

$$\begin{aligned} S &= (\bar{a}_1 \bar{a}_0 \bar{b}_1 \bar{b}_0) + (\bar{a}_1 a_0 \bar{b}_1 b_0) + (a_1 \bar{a}_0 b_1 \bar{b}_0) + (a_1 a_0 b_1 b_0) \\ &= \bar{a}_1 \bar{b}_1 (\bar{a}_0 \bar{b}_0 + a_0 b_0) + a_1 b_1 (\bar{a}_0 \bar{b}_0 + a_0 b_0) \\ &= (a_1 b_1 + \bar{a}_1 \bar{b}_1) (a_0 b_0 + \bar{a}_0 \bar{b}_0) \end{aligned}$$

$$\text{Vậy } S = (\overline{a_1 \oplus b_1}) (\overline{a_0 \oplus b_0}) \quad (5.4)$$

*b/ So sánh  $A > B$ :*

*Bảng trạng thái nếu  $A > B$*

Ta có 6 trường hợp đánh thứ tự từ 1 đến 6

$a_1$	$a_0$	$b_1$	$b_0$	Phương trình	Số thứ tự
0	1	0	0	$\bar{a}_1 a_0 \bar{b}_1 \bar{b}_0$	1
1	0	0	0	$a_1 \bar{a}_0 \bar{b}_1 \bar{b}_0$	2
1	0	0	1	$a_1 \bar{a}_0 \bar{b}_1 b_0$	3
1	1	0	0	$a_1 a_0 \bar{b}_1 \bar{b}_0$	4
1	1	0	1	$a_1 a_0 \bar{b}_1 b_0$	5
1	1	1	0	$a_1 a_0 b_1 \bar{b}_0$	6

*Phương trình nếu  $A > B$*

Gọi  $S_a$  là kết quả ra khi  $A > B$ , các phương trình có số thứ tự 1 và 6 cho ta:

$$\begin{aligned} \bar{a}_1 a_0 \bar{b}_1 \bar{b}_0 + a_1 a_0 b_1 \bar{b}_0 &= a_0 \bar{b}_0 (\bar{a}_1 \bar{b}_1 + a_1 b_1) \\ &= a_0 \bar{b}_0 (\overline{a_1 \oplus b_1}) \end{aligned}$$

Các phương trình có số thứ tự 2, 3, 4, 5 cho ta:

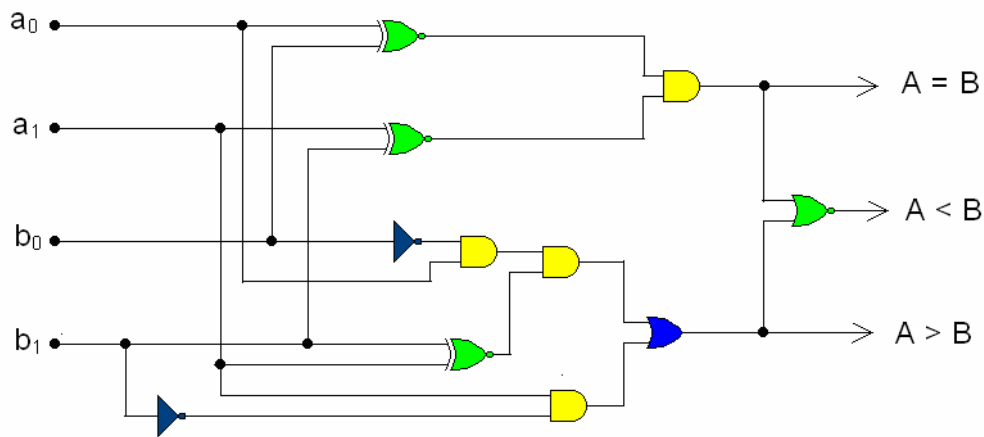
$$\begin{aligned} & a_1 \bar{a}_0 \bar{b}_1 \bar{b}_0 + a_1 \bar{a}_0 \bar{b}_1 b_0 + a_1 a_0 \bar{b}_1 \bar{b}_0 + a_1 a_0 \bar{b}_1 b_0 \\ &= a_1 \bar{b}_1 (\bar{a}_0 \bar{b}_0 + \bar{a}_0 b_0 + a_0 \bar{b}_0 + a_0 b_0) \\ &= a_1 \bar{b}_1 \left[ \overline{(a_0 \oplus b_0)} + (a_0 \oplus b_0) \right] \\ &= a_1 \bar{b}_1 \end{aligned}$$

$$\text{Vậy: } S_a = a_0 \bar{b}_0 (\overline{a_1 \oplus b_1}) + a_1 \bar{b}_1 \quad (5.5)$$

**c/ So sánh  $A < B$ :**

**Từ nhận xét:** Mỗi lần  $A = B$  hoặc  $A > B$  thì không thể có  $A < B$ , ta đi đến kết luận:  $A < B$  nếu không phải  $A \geq B$ .

Từ các phương trình (5.4) và (5.5) ta vẽ sơ đồ nguyên lý của bộ so sánh 2 bit như hình 5.18



**Hình 5.18:** Sơ đồ logic bộ so sánh hai bit

## 5.6 MẠCH SỐ HỌC:

Mạch số học giúp ta thực hiện các phép tính số học nhị phân. Ở chương 1 ta biết rằng các phép tính số học đều được thực hiện từ mạch cộng, trong kỹ thuật số có hai mạch cộng là mạch bán cộng và mạch cộng toàn phần.

### 5.6.1. Mạch bán cộng

Mạch bán cộng giúp ta cộng hai số nhị phân, trong đó mỗi số nhị phân chỉ có một bit. Từ các tính chất của phép cộng nhị phân ta có:

$$0 + 0 = 0$$

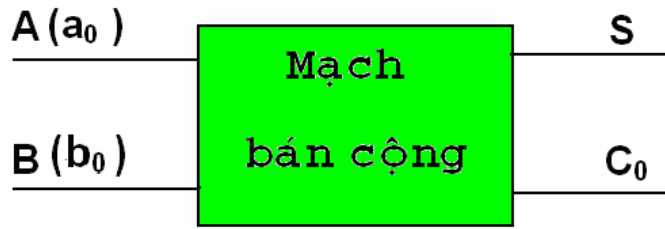
$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \text{ nhớ 1 cho vị trí trên}$$

Như vậy mạch bán cộng có hai đầu vào cho hai số nhị phân 1 bit A và B, hai đầu ra cho

tổng số S và nhớ C<sub>0</sub> (Hình 5.19)



**Hình 5.19:** Sơ đồ khối tổng quát mạch bán cộng

Bảng trạng thái bộ bán cộng

a	b	Tổng số	Nhớ (C <sub>0</sub> )	Phương trình	
				S	C <sub>0</sub>
0	0	0	0		
0	1	1	0	$\bar{a}b$	
1	0	1	0	$a\bar{b}$	
1	1	0	1		ab

Các phương trình

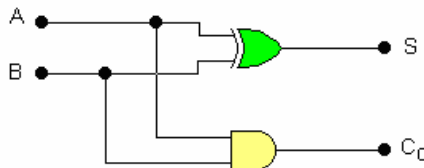
Từ bảng trạng thái ta viết được các phương trình sau của mạch bán cộng

$$S = \bar{a}b + a\bar{b} = a \oplus b$$

$$C_0 = ab$$

Sơ đồ nguyên lý mạch bán cộng

Từ các phương trình thu được ta vẽ sơ đồ sơ đồ logic mạch bán cộng như hình 5.20



**Hình 5.20:** Sơ đồ logic mạch bán cộng

### 5.6.2. Mạch cộng toàn phần (full Adder)

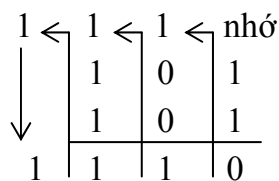
Mạch cộng toàn phần giúp ta cộng hai số nhị phân, trong đó mỗi số nhị phân có nhiều bit.

Đây là trường hợp thường gặp nhất.

Thí dụ:  $a_2a_1a_0 + b_2b_1b_0$

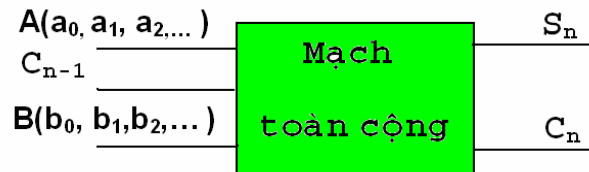
Ta phải cộng các hàng đơn vị (trọng số 0), trọng số 1, trọng số 2 cho đến hết.

Thí dụ:



Ta thấy ngay con số có trọng số 1  $A_1$  đã có thể có một con số 1 nhớ từ phép cộng các con số có trọng số 0  $A_0$  đưa lên. Nghĩa là ta cần phải đưa thêm vào kết quả ở hàng trên số nhớ mà phép cộng ở hàng trọng số dưới  $C_{n-1}$  đưa lên.

Như vậy mạch cộng toàn phần có ba đầu vào: Hai đầu vào cho hai số nhị phân nhiều bit A và B và một cho bit nhớ  $C_{n-1}$  từ phép cộng hai bit đứng trước. Hai đầu ra cho tổng số  $S_n$  và nhớ  $C_n$  (Hình 5.21)



**Hình 5.21:** Sơ đồ khối tổng quát mạch cộng toàn phần

*Bảng trạng thái bộ toàn cộng*

$C_{n-1}$	a	b	$S_n$	$C_n$	Phương trình	
					$S_n$	$C_n$
0	0	0	0	0		
0	0	1	1	0	$\bar{a} b \bar{c}$	
0	1	0	1	0	$a \bar{b} \bar{c}$	
0	1	1	0	1		$a b \bar{c}$
1	0	0	1	0	$\bar{a} \bar{b} c$	
1	0	1	0	1		$\bar{a} b c$
1	1	0	0	1		$a \bar{b} c$
1	1	1	1	1	$a b c$	$a b c$

*Các phương trình*

Từ bảng trạng thái ta viết được các phương trình sau của mạch toàn cộng

**Phương trình của tổng:**  $S_n = \bar{a} b \bar{c} + a \bar{b} \bar{c} + \bar{a} \bar{b} c + a b c$

$$= (\bar{a} b + a \bar{b}) \bar{c} + (\bar{a} \bar{b} + a b) c$$

mà  $\bar{a} b + a \bar{b} = a \oplus b$

và  $\bar{a} \bar{b} + a b = \overline{a \oplus b}$

Nên  $S_n = (a \oplus b) \bar{c} + \overline{(a \oplus b)} c$

Nó có dạng:  $A\bar{B} + \bar{A}B = A \oplus B$

Nghĩa là:  $S_n = (a \oplus b) \oplus c$  hay  $S_n = (a \oplus b) \oplus C_{n-1}$  (5.6)

**Phương trình của nhớ:**  $C_n = a b \bar{c} + a \bar{b} c + \bar{a} b c + a b c$



$$= a b (\bar{c} + c) + (\bar{a} b + a \bar{b}) c$$

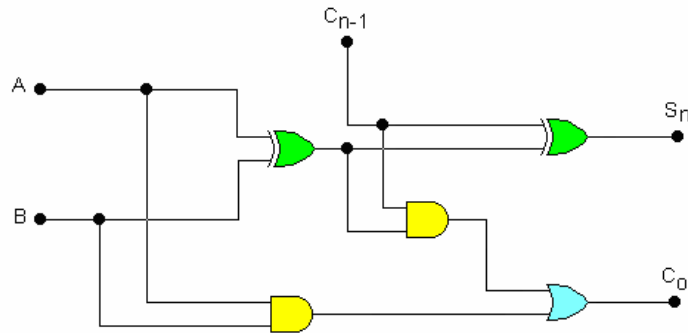
$$C_0 = a b + (\bar{a} b + a \bar{b}) c$$

$$\text{Vậy: } C_0 = a b + (a \oplus b) c$$

$$\text{Hay: } C_0 = a b + (a \oplus b) C_{n-1} \quad (5.7)$$

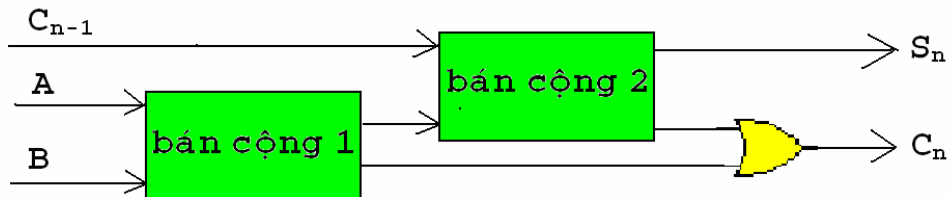
*Sơ đồ logic mạch cộng toàn phần:*

Từ các phương trình (5.6), (5.7) ta vẽ sơ đồ nguyên lý của mạch cộng toàn phần (hình 5.22)



**Hình 5.22:** Sơ đồ logic mạch cộng toàn phần

Ta thấy mạch toàn cộng chính là kết hợp hai mạch bán cộng (Hình 5.23)



**Hình 5.23:** Sơ đồ khối tổng quát mạch cộng toàn phần

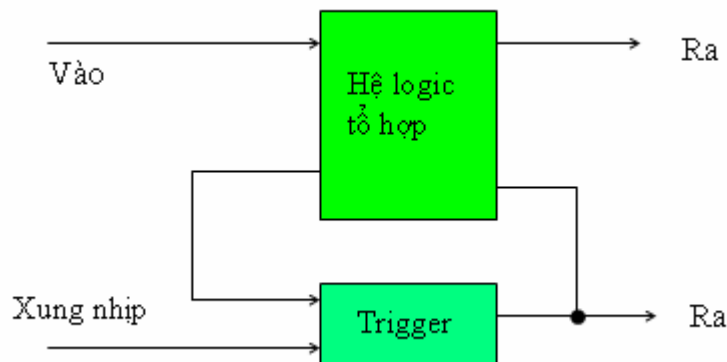
## CHƯƠNG 6

## HỆ LOGIC TUẦN TỰ

### 6.1.KHÁI NIỆM

Ngoài hệ logic tổ hợp, trong điện tử số còn có các hệ logic tuần tự (sequential logic). Đó là các hệ thống logic mà hàm ra tại mỗi thời điểm không những phụ thuộc vào các biến (hệ 2) đầu vào ở chính thời điểm đó, mà còn phụ thuộc vào các biến đầu vào ở các thời điểm trước đó. Các biến này được “*nhớ*” lại trong các bộ nhớ của hệ (vì vậy hệ logic tuần tự còn được gọi là các hệ logic có nhớ). Nói cách khác, hệ logic tuần tự là các hệ logic phụ thuộc vào các trạng thái của hệ. Các phần tử tham gia vào hệ sẽ hoạt động theo một trật tự nhất định. Khi trật tự này được hoàn tất thì trạng thái trên đầu ra mới được thiết lập.

Một cách tổng quát, ta có thể biểu diễn cấu trúc một hệ logic tuần tự như hình 6.1



**Hình 6.1:** Sơ đồ khối tổng quát hệ logic tuần tự

Từ hình 6.1 ta thấy một hệ logic tuần tự bao gồm hai khối:

*a/ Khối logic tổ hợp:* Có nhiệm vụ điều khiển việc thu nhận và đưa ra các thông tin vào ra, cũng như điều hành mọi hoạt động của các Trigger.

*b/ Khối các Trigger:* Đóng vai trò các bộ nhớ của hệ, chúng có nhiệm vụ nhớ các thông tin trung gian. Như vậy, tổ hợp các trạng thái của của các Trigger ở một thời điểm nào đó chính là trạng thái của hệ. Khi số Trigger của hệ tăng lên thì số trạng thái của hệ cũng tăng theo. Từ các Trigger ta cũng có thể lấy các thông tin ra.

Đối với việc thiết kế các hệ logic tuần tự, cần phải chú ý vòng hồi tiếp giữa phần hệ tổ hợp điều khiển và các Trigger, hoặc giữa các Trigger với nhau.

Việc đưa một xung nhịp có độ ổn định cao vào để đồng bộ các Trigger có ý nghĩa rất lớn, nó hình thành hai loại kết cấu thuộc hệ logic tuần tự là đồng bộ và không đồng bộ.

Tóm lại, các hệ logic tuần tự sử dụng chủ yếu các Trigger, chúng đều là các mạch logic kế tiếp, Trong các hệ logic tuần tự có hai hệ thống thông dụng nhất là các bộ đếm và mạch ghi.

### 6.2. BỘ ĐẾM

Trong quá trình tính toán, điều khiển, tự động hoá... thường cần phải ghi lại một số trạng

thái xảy ra theo một thứ tự nhất định nào đó, như số xung nhận được theo thứ tự thời gian, sau đó trạng thái này được ghi vào bộ nhớ. Việc đếm được thực hiện nhờ bộ đếm (counter).

Bộ đếm còn được dùng trong máy tính để ghi nhận thứ tự các lệnh của một chương trình, sau đó máy tính sẽ thực hiện tự động. Được dùng làm bộ chia tần số điều chỉnh được (điều chỉnh hệ số chia) nhằm thực hiện phép tính một cách đơn giản.

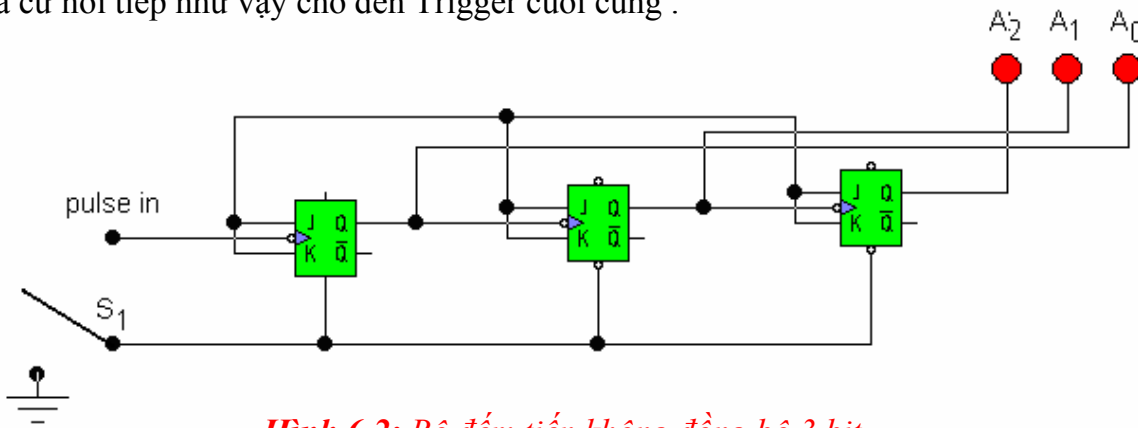
Bộ đếm là một trong những mạch điện tử số quan trọng sử dụng các trigger, thuộc hệ logic tuần tự, trong đó số lượng trigger có trong bộ đếm quyết định dung lượng của bộ đếm.

### 6.2.1: Bộ đếm không đồng bộ

Còn gọi là bộ đếm nối tiếp. Các xung đếm chỉ đưa vào đầu vào của Trigger thứ nhất, Trigger này lại điều khiển Trigger tiếp sau nó.

#### a/ Bộ đếm tiến nhị phân không đồng bộ 3 bit

Bộ đếm tiến nhị phân không đồng bộ 3 bit có sơ đồ logic như hình 6.2. Đây là bộ đếm nhị phân cấu tạo từ JKFF, đầu ra của Trigger thứ nhất được nối vào lối vào đồng bộ của Trigger thứ 2 và cứ nối tiếp như vậy cho đến Trigger cuối cùng .



**Hình 6.2: Bộ đếm tiến không đồng bộ 3 bit**

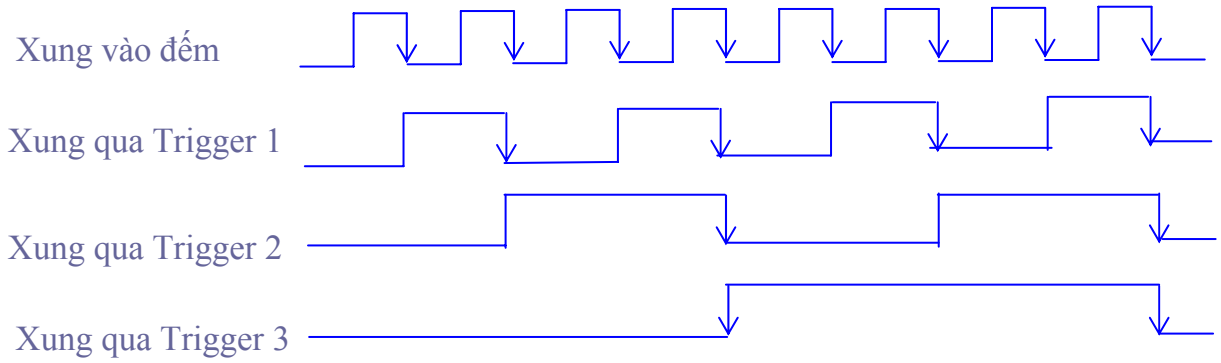
Bộ đếm có một đầu vào cho xung đếm và nhiều đầu ra. Những đầu ra thường là những đầu ra Q của Trigger. Vì Q chỉ có thể có một trong hai trạng thái 1 và 0 cho nên sự sắp xếp các đầu ra này cho phép ta biểu diễn kết quả dưới dạng một số hệ nhị phân có số bit bằng số Trigger dùng trong bộ đếm.

Bộ đếm tiến nhị phân không đồng bộ 3 bit (còn gọi là bộ đếm mod 8) có dung lượng tối đa là 7 (một chục) sau đó quay lại đếm chục khác. Nó không tự dừng lại khi đếm hết dung lượng của nó.

Các lối vào JK của Trigger đấu chung với nhau và để lơ lửng (luôn luôn ở trạng thái 1). Các đầu vào Cl đấu chung với nhau và đầu vào chuyển mạch  $S_1$ . Trước khi đếm, ta đóng  $S_1$  để nối đất Cl, nhằm điều khiển các đèn chỉ thị về 0 trước khi đếm.

Nó có biểu đồ xung như Hình 6.3

Các Trigger đều có đầu vào JK luôn luôn ở mức 1 nên sẽ chuyển trạng thái mỗi khi có xung đến. Ta thấy :



**Hình 6.3:** Biểu đồ xung đưa vào đầu vào đồng bộ

\*Trigger thứ nhất thay đổi trạng thái mỗi khi có xung đến .

\*Các Trigger sau chỉ thay đổi trạng thái khi lối ra Q của Trigger trước nó chuyển từ 1 về 0

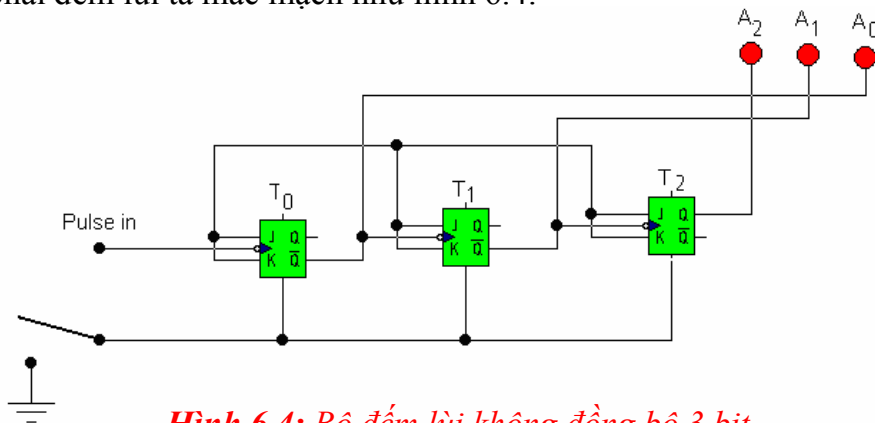
\*Với 3 Trigger mắc nối tiếp, sau 8 xung bộ đếm trở lại trạng thái đầu. Vì vậy với n Trigger nối tiếp có thể đếm đến  $2^n$ .

**Bảng 6.1** Bảng trạng thái của bộ đếm không đồng bộ 3 bit

Xung đếm	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
0	0	0	0

**b/Bộ đếm lùi nhị phân không đồng bộ ba bit**

Khi cần phải đếm lùi ta mắc mạch như hình 6.4.



**Hình 6.4:** Bộ đếm lùi không đồng bộ 3 bit

Khác với bộ đếm tiến, bộ đếm lùi có các đầu ra lấy ở đầu ra bù ( $\bar{Q} = 1$ )

Lúc mới bắt đầu đếm các đèn đều sáng nên ta có chỉ thị 111 (số 7), sau đó số đếm lùi dần theo số xung vào. Bảng trạng thái ngược với bảng trạng thái của bộ đếm tiến.

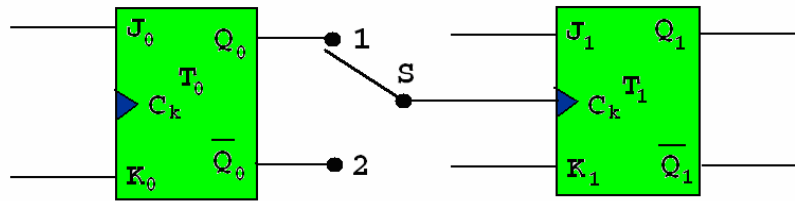
### c/ Bộ đếm thuận nghịch không đồng bộ

Bộ đếm thuận nghịch còn gọi là bộ đếm hỗn hợp, nghĩa là đếm theo cả 2 chiều:

*Chiều thuận:* Mỗi xung vào làm tăng số đếm lên 1.

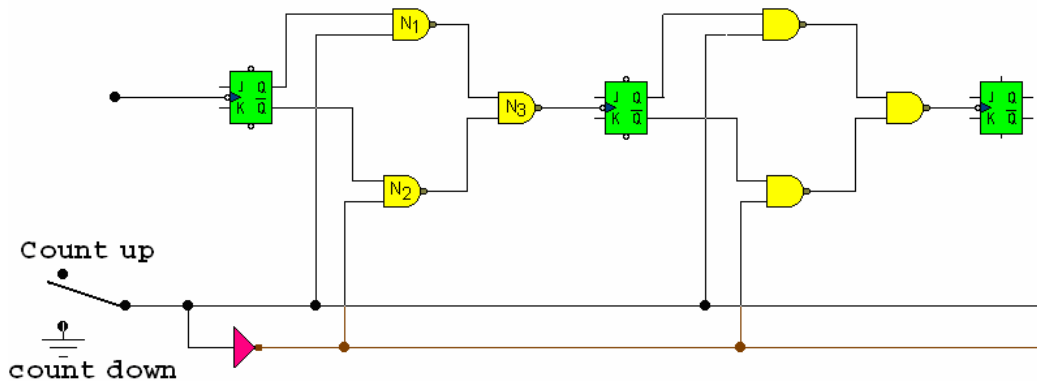
*Chiều nghịch:* Mỗi xung vào làm giảm số đếm 1

Như trên ta đã biết, muốn đếm theo chiều nghịch ta chỉ việc nối lối vào của mỗi Trigger với lối ra  $\bar{Q}$  của Trigger trước nó. Khi một Trigger nào đó có lối ra Q chuyển trạng thái từ 1 về 0 thì lối ra  $\bar{Q}$  của nó chuyển trạng thái từ 0 lên 1 và do đó làm Trigger kế tiếp (được nối với  $\bar{Q}$ ) chuyển trạng thái. Một chuyển mạch (Hình 6.5) sẽ làm nhiệm vụ đảo mạch khi cần đếm tiến hoặc đếm lùi.



**Hình 6.5:** Chuyển mạch cơ khí của bộ đếm thuận nghịch

Chuyển mạch S được chuyển vào Q (vị trí 1) khi đếm thuận và  $\bar{Q}$  (vị trí 2) khi đếm nghịch. Chuyển mạch S trong thực tế được thay thế bằng bộ chuyển mạch logic (Hình 6.6)



**Hình 6.6:** Chuyển mạch Logic bộ đếm thuận nghịch

Nếu  $X = 1$  (để X lơ lửng) ta thực hiện đếm tiến. Lúc đó cổng NAND  $N_1$  có một đầu ở mức 1, từ bảng trạng thái của NAND ta có: Nếu một đầu vào của cổng NAND ở trạng thái 1 thì đầu ra sẽ là đảo của đầu vào kia. Vì vậy, đầu ra của  $N_1$  là  $\bar{Q}$ .

Cổng NAND của  $N_2$  có một đầu vào là 0 nên đầu ra luôn luôn là 1, dù đầu vào kia ở bất kỳ trạng thái nào.

Nếu  $X = 0$  (nối đất) ta thực hiện đếm lùi. Lúc đó  $N_1$  có một đầu vào ở mức 0 nên đầu ra luôn luôn ở mức 1. Đầu ra của  $N_1$  nhận trạng thái 1.  $N_2$  có một đầu vào ở mức 1 nên đầu ra là

ngịch đảo của đầu kia (lúc đó là  $\bar{Q}$ ) nên đầu ra  $N_2$  là  $Q$ .

Tương tự,  $N_3$  có một đầu vào ở mức 1 nên đầu ra là  $\bar{Q}$ .

**Kết quả:** Đầu ra  $\bar{Q}$  của Trigger thứ nhất được đấu với đầu vào đếm của Trigger tiếp theo nên bộ đếm sẽ thực hiện việc đếm lùi.

#### ***d/Bộ đếm không đồng bộ N phân***

Khi cần đếm trên một hệ đếm cơ số  $N$  khác 2, nghĩa là cần một bộ đếm đến  $N$  bất kỳ, ta có thể dùng một chuỗi  $n$  Trigger liên tiếp, trong đó  $n$  là số nhỏ nhất thỏa mãn điều kiện  $2^n > N$ , sau đó dùng phương pháp hồi tiếp để xóa đi các trạng thái không cần thiết.

**Thí dụ:** Với bộ đếm đến 10 ( $N = 10$ ) phải dùng 4 Trigger ( $n = 4$ ), vì số  $n$  nhỏ nhất thỏa mãn bất đẳng thức  $2^n > 10$  là 4. Tiếp đó ta dùng mạch hồi tiếp sao cho khi đếm đến  $N$  (trong trường hợp này là đếm đến 10) thì tất cả Trigger phải trở về trạng thái 0. Như vậy bộ đếm 10 chỉ lấy 10 trạng thái trong số 16 trạng thái.

Mạch hồi tiếp có thể dùng cổng NAND, lối ra của nó được nối đồng thời với tất cả các lối vào xóa (Cl) của các Trigger.

Ta có bảng tương đương của 2 hệ đếm thập phân và nhị phân (Bảng 6.2)

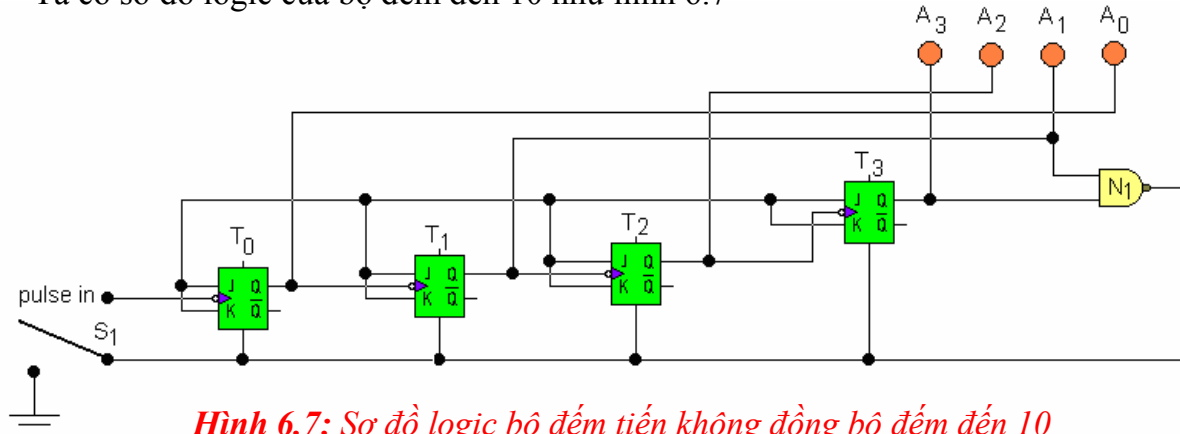
***Bảng 6.2: Bảng tương đương giữa thập phân và nhị phân.***

Đếm thập phân	Đếm nhị phân			
	$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

Như vậy số 10 viết trong hệ đếm cơ số hai là 1010. Khi xung thứ 10 đến, đầu ra của bộ

đếm phải là  $Q_0 = 0, Q_1 = 0, Q_2 = 0, Q_3 = 0$ , thay vì  $Q_0 = 1, Q_1 = 0, Q_2 = 1, Q_3 = 0$ . Muốn vậy, ta đưa  $Q_1$  và  $Q_3$  vào mạch hồi tiếp do cổng NAND đảm nhận, mạch này sẽ đưa đầu ra của  $Q_1$  và  $Q_3$  về lại đầu vào  $C_1$  để xóa trạng thái của Trigger 1 và 3 về 0. Muốn đếm đến một số khác 10, ta cũng thực hiện như trên.

Ta có sơ đồ logic của bộ đếm đến 10 như hình 6.7



**Hình 6.7:** Sơ đồ logic bộ đếm tiến không đồng bộ đếm đến 10

### 6.2.2. bộ đếm đồng bộ

Bộ đếm không đồng bộ có khuyết điểm là gây ra một sự trễ giữa các lần chuyển trạng thái liên tiếp, vì sự chuyển trạng thái của Trigger thứ 2 trở đi phụ thuộc vào sự chuyển trạng thái về 0 của Trigger trước nó. Để sự chuyển trạng thái của các Trigger sau không còn phụ thuộc Trigger thứ nhất người ta sử dụng bộ đếm đồng bộ (còn gọi là bộ đếm song song).

Đặc điểm của bộ đếm đồng bộ là sự thay đổi trạng thái của bộ đếm chỉ phụ thuộc vào xung đếm hay xung đồng bộ, nghĩa là xung đồng bộ được đưa vào đồng thời tất cả các Trigger của bộ đếm. Do đó quá trình chuyển trạng thái chỉ phụ thuộc vào sự truyền lan tín hiệu chuyển (chuyển từ cột số thấp sang cột số cao).

Như vậy, bộ đếm đồng bộ sẽ :

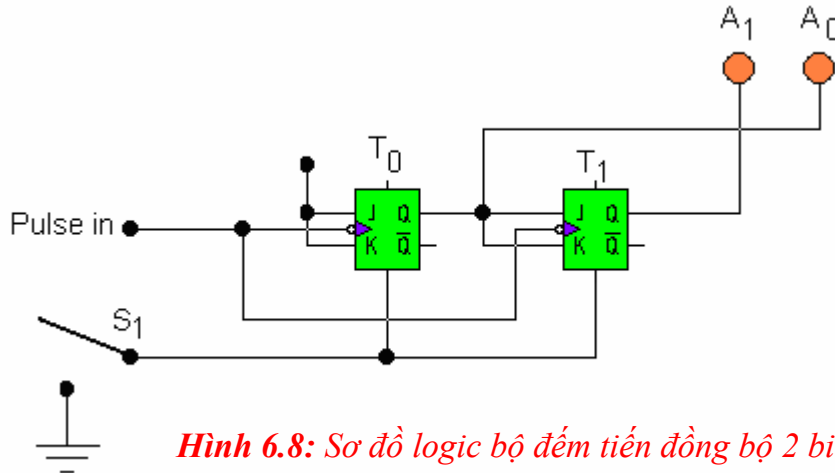
- \* Thực hiện việc chuyển trạng thái của các tầng ở cùng 1 thời điểm.
- \* Các Trigger có thể thay đổi trạng thái của nó hoặc không (tùy theo trạng thái trước đó của nó) nhưng cùng một thời điểm.
- \* Xung đếm được đưa vào đồng thời tất cả các Trigger.

#### a/Bộ đếm đồng bộ 2 bit

Bộ đếm đồng bộ 2 bit có sơ đồ logic như hình 6.8

Các đầu vào JK của Trigger thứ nhất đấu với nhau và để lơ lửng ( $J_0 = K_0 = 1$ ), đầu ra  $Q_0$  của Trigger thứ nhất đấu vào đầu vào JK của Trigger thứ 2. Các đầu vào đồng bộ (đầu vào đếm) đấu với nhau. Như vậy các xung đồng bộ tác động đồng thời đến các Trigger. Các đầu  $C_1$  đấu với nhau và đầu vào công tắc  $S_1$ , trước khi đếm ta ấn công tắc  $S_1$  xuống đất để đưa các Trigger về trạng thái 0 ( $Q_0 = Q_1 = 0$ ).

Khi sườn âm của xung thứ nhất tác động vào đầu vào đếm, Trigger thứ nhất  $T_0$  chuyển



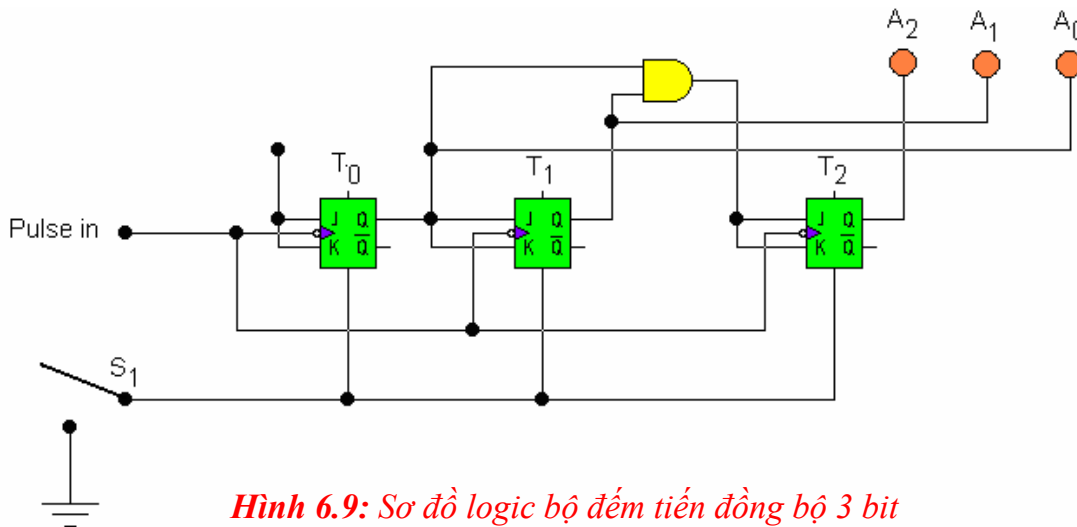
**Hình 6.8:** Sơ đồ logic bộ đếm tiến đồng bộ 2 bit

trạng thái  $Q_0 = J_1 = K_1 = 1$ . Trigger thứ hai  $T_1$  giữ nguyên trạng thái vì trước đó  $Q_0 = 0$ . Xung tiếp theo Trigger thứ nhất chuyển trạng thái, lúc đó Trigger thứ hai cũng chuyển trạng thái (vì trước đó  $J_1 = K_1 = Q_0 = 1$ ), nghĩa là cả hai đều chuyển trạng thái.

**Vậy:** Trigger thứ nhất luôn luôn có đầu vào JK ở trạng thái 1 (lơ lửng trên cao) nên luôn luôn thay đổi trạng thái mỗi khi có xung đến, các Trigger tiếp sau thay đổi trạng thái chỉ khi nào Trigger trước nó ở trạng thái 1.

### **b/Bộ đếm đồng bộ 3 bit**

Còn gọi là bộ đếm đồng bộ Mod- 8. Có sơ đồ nguyên lý như hình 6.9



**Hình 6.9:** Sơ đồ logic bộ đếm tiến đồng bộ 3 bit

Trigger thứ nhất  $T_0$  có đầu vào JK luôn luôn ở trạng thái 1 vì vậy sẽ chuyển trạng thái mỗi khi có xung đến. Lỗi ra  $Q_0$  của nó đầu vào  $J_1 K_1$  của trigger thứ hai và một đầu vào của cổng AND. Đầu vào còn lại của cổng AND nối vào lỗi ra  $Q_1$  của Trigger thứ hai, đầu ra của cổng AND đầu vào đầu vào  $J_2 K_2$  của trigger thứ ba  $T_3$ . Cổng AND có nhiệm vụ sửa sai trong quá trình đếm.

Xung đếm được đưa đồng thời vào các đầu vào đồng bộ  $C_k$  của các trigger. Các đầu vào xóa  $Cl$  đầu chung với nhau để chuyển trạng thái các trigger về 0 trước khi đếm.



Khi sườn sau của xung đếm thứ nhất đưa đến  $C_k$  của trigger thứ nhất. Lỗi ra  $Q$  của nó chuyển trạng thái lên 1, các trigger còn lại giữ nguyên trạng thái 0 (vì  $J_1 = K_1 = J_2 = K_2 = 0$ ). Bộ chỉ thị hiện số 001.

Khi sườn sau của xung thứ hai đến, lỗi ra  $Q_0$  chuyển trạng thái về 0,  $Q_1$  chuyển trạng thái lên 1,  $Q_2$  vẫn ở trạng thái 0. Bộ chỉ thị hiện số 010.

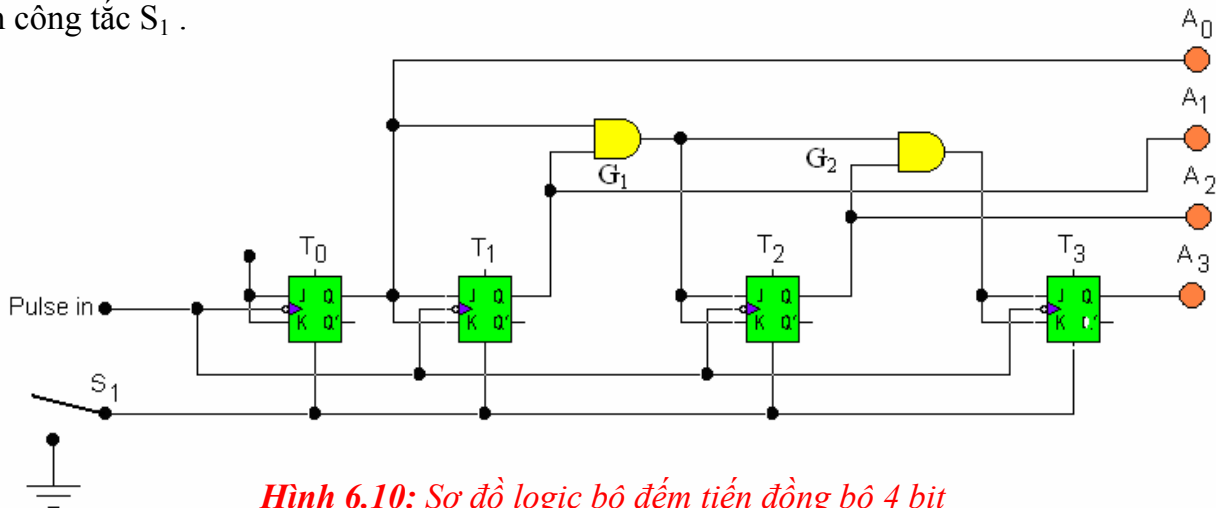
Khi sườn sau của xung đếm thứ ba đến, lỗi ra  $Q_0$  lại chuyển trạng thái lên 1,  $Q_1$  vẫn giữ nguyên trạng thái. Do cổng AND có một lỗi vào bằng 0 nên lỗi ra bằng 0 vì vậy  $Q_1$  không chuyển trạng thái. Bộ chỉ thị hiện số 011.

Khi sườn sau của xung đếm thứ tư đến, lỗi ra  $Q_0$  chuyển trạng thái về 0,  $Q_1$  cũng chuyển trạng thái về 0. Do cổng AND có hai lỗi vào đều bằng 1 nên lỗi ra bằng 1 vì vậy  $Q_2$  cũng chuyển trạng thái từ 0 lên 1. Bộ chỉ thị hiện số 100.

Quá trình tiếp diễn như vậy cho đến khi đếm đến 7 bộ chỉ thị quay về trạng thái đầu  
*c/Bộ đếm đồng bộ 4 bit*

Còn gọi là bộ đếm đồng bộ Mod 16. Nó có sơ đồ nguyên lý như hình 6.11 và giản đồ xung như hình 6.10.

Trước khi đếm ta nối các đầu ra  $C1$  xuống đất để tất cả Trigger đều chỉ thị 0 bằng cách nhấn công tắc  $S_1$ .



**Hình 6.10:** Sơ đồ logic bộ đếm tiến đồng bộ 4 bit

Khi sườn sau của xung thứ nhất tác động vào,  $Q_0$  sẽ chuyển trạng thái từ 0 lên 1 (do  $J_0 K_0$  luôn luôn ở trạng thái bập bênh 1).  $Q_1, Q_2, Q_3$ , vẫn giữ nguyên trạng thái. Bộ chỉ thị hiện số 0001.

Khi sườn sau của xung thứ hai tác động vào,  $Q_0$  chuyển trạng thái về 0,  $Q_1$  chuyển trạng thái lên 1 vì trước đó  $Q_0$  đang ở trạng thái 1.  $Q_2, Q_3$  giữ nguyên trạng thái 0. Bộ chỉ thị hiện số 0010.

Khi sườn sau của xung thứ ba tác động vào,  $Q_0$  chuyển trạng thái lên 1,  $Q_1$  vẫn ở trạng thái 1 vì trước đó  $Q_0$  ở trạng thái 0.  $Q_2, Q_3$  vẫn ở trạng thái 0. Bộ chỉ thị hiện số 0011.

Khi sườn sau của xung thứ tư tác động vào,  $Q_0$  chuyển trạng thái về 0,  $Q_1$  cũng chuyển

trạng thái về 0 vì trước đó  $Q_0$  ở trạng thái 1,  $Q_2$  chuyển trạng thái lên 1 vì  $G_1$  có hai lối vào đều ở trạng thái 1 nên lối ra là 1 nên  $Q_2$  chuyển trạng thái,  $Q_3$  vẫn ở 0. Bộ chỉ thị hiện số 0100.

Khi sườn sau của xung thứ 11 tác động vào,  $Q_0$  chuyển trạng thái từ 0 lên 1,  $Q_1$  vẫn giữ nguyên trạng thái vì trước đó  $Q_0$  ở trạng thái 0,  $Q_2$  giữ nguyên trạng thái 0. Trước đó cổng  $G_1$  có một đầu vào là 0 một đầu vào là 1 nên đầu ra là 0,  $Q_2$  giữ nguyên trạng thái.  $Q_3$  cũng giữ nguyên trạng thái 1 vì  $G_2$  có hai lối vào đều ở trạng thái 0 nên lối ra bằng 0,  $Q_3$  không chuyển trạng thái. Bộ chỉ thị hiện số 1011.

### 6.3. BỘ GHI DỊCH (Shift Register)

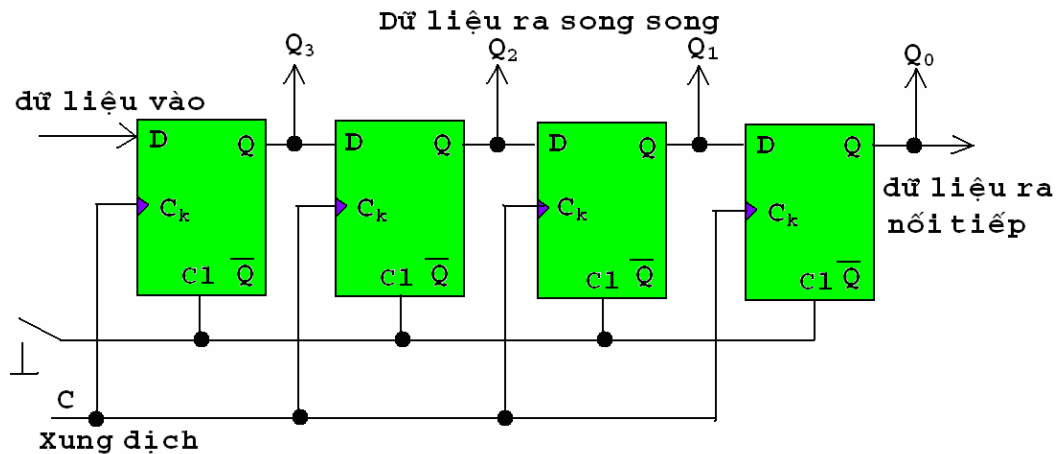
Bộ ghi dịch còn gọi là thanh ghi dịch, có hai nhiệm vụ là lưu giữ tạm thời dữ liệu và dịch từng bit theo xung đồng hồ. Dữ liệu lưu trữ trong bộ ghi dịch, dưới tác dụng của xung dịch (shift) có thể tuần tự dịch trái hoặc dịch phải. Phần tử cấu thành bộ ghi dịch là các trigger D. Vì việc dịch được tiến hành đồng thời đối với mỗi phần tử, nên các trigger này hoạt động theo chế độ đồng bộ. Bộ ghi dịch được phân loại như sau:

**Theo cách dịch:** Bộ ghi dịch một hướng (trái hoặc phải), bộ ghi dịch hai hướng (trái và phải), bộ ghi dịch vòng.

**Theo cách đưa tín hiệu vào và lấy tín hiệu ra:** Bộ ghi dịch vào nối tiếp ra nối tiếp hoặc ra song song, bộ ghi dịch vào song song ra nối tiếp hoặc song song, bộ ghi dịch đa năng.

#### 6.3.1. Bộ ghi dịch một hướng:

Hình 6.11 là sơ đồ nguyên lý một bộ ghi dịch 4 bit. Dữ liệu đưa vào bên trái, dịch phải, lấy ra song song hoặc nối tiếp.



**Hình 6.11:** Sơ đồ nguyên lý bộ ghi dịch một hướng

Bộ ghi dịch một hướng trên dùng trigger D. Đầu ra Q của mỗi trigger được nối với đầu vào của trigger tiếp theo. Đầu tiên ta nối đất C1 để đưa các đầu ra Q về 0. Dữ liệu được đưa vào D của trigger thứ nhất. Khi sườn dương của xung đồng hồ xuất hiện, tín hiệu mã hoá được dịch vào trigger 1, đồng thời trạng thái của mỗi trigger cũng được dịch đến trigger tiếp theo.

Giả sử ở đầu vào ta đưa từ mã 1101. Dưới tác dụng của xung dịch, quá trình dịch được

ghi ở bảng 6.3

**Bảng 6.3: Quá trình ghi dịch từ mã 1101 trong bộ ghi dịch.**

$C_k$	Từ mã trong bộ ghi dịch			
Số thứ tự	Trigger 1	Trigger 2	Trigger 3	Trigger 4
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	0	1	1	0
4	1	0	1	1

Như vậy, sau 4 xung đồng hồ từ mã 1101 đã dịch vừa hết vào bộ ghi dịch. Lúc này ta có thể lấy ra 4 bit song song của từ mã 1101 ở 4 đầu ra Q của các trigger. Đầu ra  $Q_0$  của trigger cuối cùng có thể làm đầu ra nối tiếp của 1101, vì chỉ cần qua một thời gian 4 xung đồng hồ nữa thì ở đầu ra Q của trigger cuối sẽ có 4 bit nối tiếp dịch ra. Ta thấy bộ ghi dịch này có đầu vào nối tiếp, đầu ra nối tiếp và song song. Một cách tổng quát ta có bảng trạng thái bộ ghi dịch 4 bit sau (bảng 6.4).

**Bảng 6.4: Bảng trạng thái bộ ghi dịch 4 bit**

$C_k$	$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	0	0	0	0
1	$D_0$	0	0	0
2	$D_1$	$D_0$	0	0
3	$D_2$	$D_1$	$D_0$	0
4	$D_3$	$D_2$	$D_1$	$D_0$
5	0	$D_3$	$D_2$	$D_1$
6	0	0	$D_3$	$D_2$
7	0	0	0	$D_3$

Từ bảng trạng thái ta thấy:

Nếu lấy ra song song thì sau 4 xung đồng hồ (tổng quát là n xung cho bộ ghi dịch n bit) dữ liệu đã được ghi vào bộ ghi.

Nếu lấy ra nối tiếp ta cần 7 xung đồng hồ (tổng quát là n - 1 xung cho bộ ghi dịch n bit).

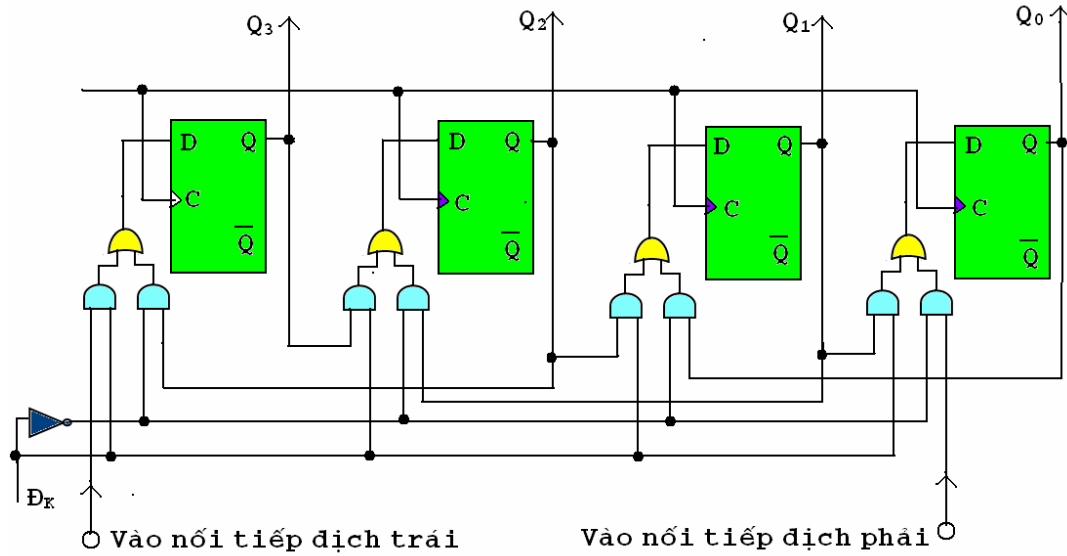
Như vậy bộ ghi dịch một hướng trên có thể đồng thời chuyển chuỗi tín hiệu từ nối tiếp sang song song hoặc làm trễ chuỗi này đi n - 1 chu kỳ xung đồng bộ.

### 6.3.2. Bộ ghi dịch hai hướng:

Bộ ghi dịch hai hướng có thể dịch phải hoặc dịch trái tùy theo tín hiệu điều khiển, bộ ghi dịch hai hướng có thêm một số cổng logic để thay đổi các mối liên lạc giữa các trigger, nhằm thay đổi hướng dịch. Hình 6.12 cho ta sơ đồ nguyên lý bộ ghi dịch hai hướng phải và trái.

Việc thay đổi hướng dịch được tiến hành nhờ đầu vào điều khiển  $D_K$ . Khi  $D_K = 1$  bộ ghi

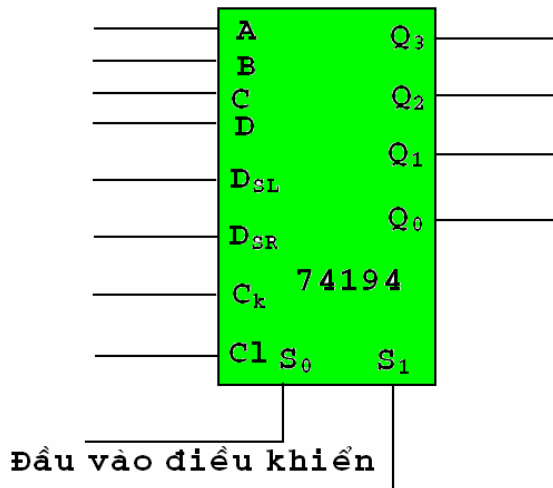
dịch sẽ thực hiện dịch phải. Khi  $D_K = 0$  bộ ghi dịch sẽ thực hiện dịch trái.



**Hình 6.12:** Sơ đồ nguyên lý bộ ghi dịch hai hướng

**6.3.3. Bộ ghi dịch dùng vi mạch:**

Để thực hiện bộ ghi dịch, các nhà chế tạo vi mạch đã sản xuất vi mạch 74194. Bộ ghi dịch này có 10 cửa vào và 4 cửa ra. (hình 6.13), ABCD là bốn đầu vào nạp song song.  $D_{SR}$  và  $D_{SL}$  là các đầu vào nạp nối tiếp. Đầu vào nối tiếp dịch phải  $D_{SR}$  nạp số liệu vào vị trí  $A = Q_3$  rồi dịch



**Hình 6.13:** Vi mạch ghi dịch 74194

phải. Đầu vào nối tiếp dịch trái  $D_{SL}$  nạp số liệu vào vị trí  $D = Q_0$  rồi dịch trái. Đầu vào  $Cl$  xóa các trạng thái của trigger và cho về 0. Đầu vào xung nhịp  $C_K$  sẽ khởi động cả 4 trigger khi xung nhịp chuyển từ thấp lên cao. Các đầu vào điều khiển sẽ điều khiển bộ ghi dịch dịch phải hoặc dịch trái.

**6.3.4. Bộ ghi dịch làm bộ đếm vòng**

Bộ ghi dịch có nhiều ứng dụng trong mạch số. Một trong những ứng dụng của nó là bộ

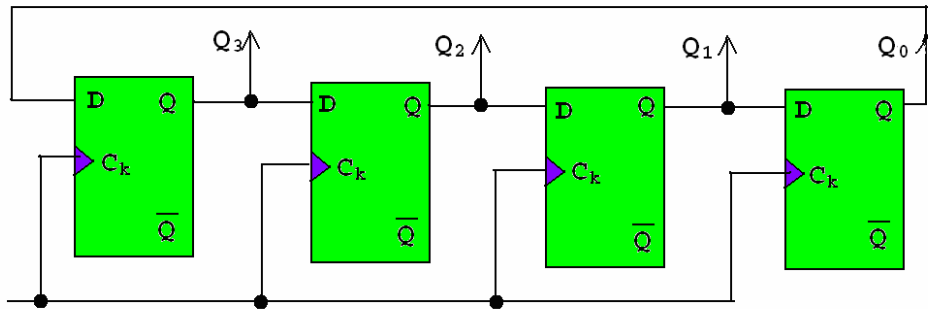
đếm vòng. Mã đếm vòng là loại mã đơn giản về phương pháp mã hoá cũng như giải mã. Bảng 6.5 cho ta mã đếm vòng 10 bit.

**Bảng 6.5: Mã đếm vòng 10 bit**

Thập phân	Mã đếm vòng
0	0000000001
1	0000000010
2	0000000100
3	0000001000
4	0000010000
5	0000100000
6	0001000000
7	0010000000
8	0100000000
9	1000000000

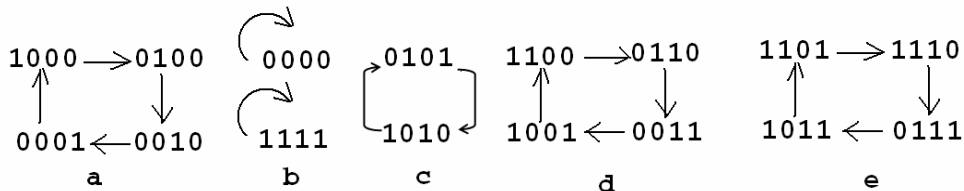
Từ bảng trạng thái ta thấy mỗi tổ hợp mã chỉ có một bit 1, các bit còn lại là bit 0. Như vậy khi đếm từ 0 đến 9, ta thấy ở đầu ra lần lượt lên 1, do đó việc đếm (giải mã) được tiến hành bằng cách dịch liên tiếp bit 1 từ phải qua trái.

Hình 6.14 là sơ đồ nguyên lý bộ đếm vòng 4 bit



**Hình 6.14: Sơ đồ nguyên lý bộ đếm vòng 4 bit**

Dùng 4 bit nên có 16 tổ hợp xuất hiện:



Trong đó đồ hình a được dùng vì phù hợp với mã vòng. Lấy  $D_n = Q_0$  bằng cách nối đầu ra  $Q_0$  vào D. Xung nhịp được đưa vào đồng thời đầu vào  $C_k$  của các trigger.

## **CHƯƠNG 7**

## **CHUYỂN ĐỔI TÍN HIỆU**

### **7.1. TÍN HIỆU TƯƠNG TỰ VÀ TÍN HIỆU SỐ**

Tín hiệu tương tự (analog) thường là những điện thế hoặc dòng điện có dạng biến thiên liên tục theo thời gian. Thông tin mà các tín hiệu tương tự muốn truyền đạt cho con người là biên độ của nó. Biên độ này là một hàm theo thời gian.

$$u = f(t) \qquad i = g(t)$$

Tùy theo dạng của  $f$  và  $g$  mà ta có thể kết luận được về bản chất của tín hiệu  $u$ ,  $i$  trên.

Tín hiệu liên tục có thể biến đổi một cách liên tục giữa các giá trị cực đại và cực tiểu của điện áp hay dòng điện.

Tín hiệu số (digital) chứa thông tin ở vị trí các xung hay sự thay đổi đột ngột của biên độ (so với tín hiệu thời gian chuẩn) còn trị số tuyệt đối của biên độ xung không quan trọng.

Tín hiệu số chỉ có hai mức gián đoạn (ứng với trạng thái của một mạch lật) của điện áp.

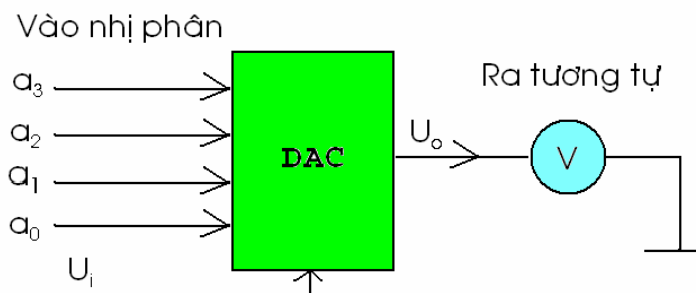
Các thông tin cần xử lý có thể ở dạng số hoặc dạng tương tự, nhưng khi tính toán hoặc xử lý thông tin máy tính lại chỉ hoạt động theo cơ chế nhị phân. Vì vậy ta phải chuyển các tín hiệu không phải số thành số (ADC), sau khi tính toán, xử lý xong các kết quả cần phải hiển thị theo tín hiệu tương tự nhờ bộ chuyển đổi số tương tự (DAC).

Như vậy, bản chất của hai loại tín hiệu tương tự và số hoàn toàn khác nhau. Tín hiệu tương tự thường gặp trong thực tế do tính chất liên tục theo thời gian của nó phù hợp với các hiện tượng vật lý thông thường. Còn tín hiệu số thì chỉ xuất hiện trong các thiết bị số, thường được dùng như một hình thức trung gian rất thuận tiện trong việc đo lường, tính toán và xử lý tín hiệu, nó thường không xuất hiện trực tiếp trong các hiện tượng thực tế. Để thực hiện việc ghép nối các hệ thống số với các thiết bị tương tự, chúng ta phải thực hiện việc chuyển đổi giữa hai loại tín hiệu này.

### **7.2. BỘ BIẾN ĐỔI SỐ - TƯƠNG TỰ**

#### **7.2.1. Khái niệm**

Bộ biến đổi Số - Tương tự DAC (Digital Analog Converter) có sơ đồ khối tổng quát như hình 7.1



**Hình 7.1:** Sơ đồ khối tổng quát bộ DAC

DAC tiếp nhận một mã số  $n$  bit nhị phân song song ở lối vào và biến đổi thành điện áp hoặc dòng điện tương ứng ở lối ra. Dòng điện hoặc điện áp ra từ DAC là hàm của mã số ở lối vào và có biến thiên phù hợp với mã số này.

Điện áp  $U_i$  còn gọi là  $U$  chuẩn, đây là một điện áp một chiều (DCV) rất ổn định nằm trong DAC. Điện áp ra  $U_0$  có được là do sự phân chia điện áp một chiều này.

Ta có bảng trạng thái của một DAC 4 bit (bảng 7.1).

**Bảng 7. 1: Bảng trạng thái DAC 4 bit**

Dòng	Vào nhị phân				Ra tương tự (V)
	A <sub>3</sub> 8	A <sub>2</sub> 4	A <sub>1</sub> 2	A <sub>0</sub> 1	
1	0	0	0	0	0
2	0	0	0	1	1
3	0	0	1	0	2
4	0	0	1	1	3
5	0	1	0	0	4
6	0	1	0	1	5
7	0	1	1	0	6
8	0	1	1	1	7
9	1	0	0	0	8
10	1	0	0	1	9
11	1	0	1	0	10
12	1	0	1	1	11
13	1	1	0	0	12
14	1	1	0	1	13
15	1	1	1	0	14
16	1	1	1	1	15

Từ bảng trạng thái ta có:

**Dòng 1:** Tất cả các đầu vào đều là 0, đầu ra là 0 Volt .

**Dòng 2:** Ứng với đầu vào A<sub>0</sub> hoạt động ở mức 1, lúc đó đầu vào là A<sub>3</sub> A<sub>2</sub> A<sub>1</sub> A<sub>0</sub> là 0001 đầu ra cho ta 1 Volt .

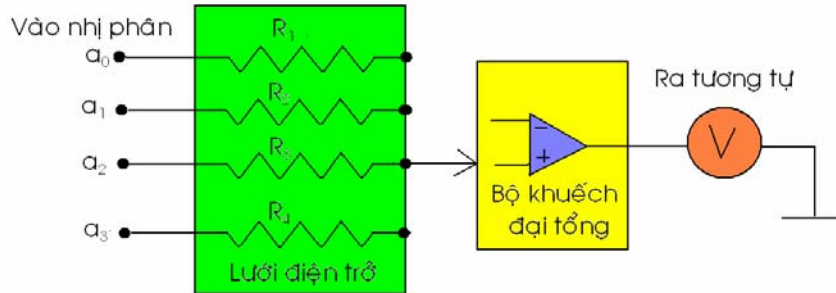
**Dòng 3:** Ứng với đầu vào A<sub>1</sub> hoạt động ở mức 1, lúc đó đầu vào A<sub>3</sub> A<sub>2</sub> A<sub>1</sub> A<sub>0</sub> là 0010 đầu ra cho ta điện áp 2 volt.

**Dòng 5:** Ứng với đầu vào A<sub>2</sub> hoạt động ở mức 1, lúc đó đầu vào A<sub>3</sub> A<sub>2</sub> A<sub>1</sub> A<sub>0</sub> là 0100 đầu ra cho ta điện áp 4 volt.

**Dòng 9:** Ứng với đầu vào A<sub>3</sub> hoạt động ở mức 1, lúc đó đầu vào A<sub>3</sub> A<sub>2</sub> A<sub>1</sub> A<sub>0</sub> là 1000 đầu ra cho ta điện áp 8 volt .

Như vậy cần có một sự cân bằng nào đó giữa các đầu vào  $A_3 A_2 A_1 A_0$ . Sự cân bằng này là 8 cho  $A_3$ , 4 cho  $A_2$ , 2 cho  $A_1$  và 1 cho  $A_0$ .

Từ đó ta thấy một DAC phải có sơ đồ khối như hình 7.2.



**Hình 7.2:** Sơ đồ khối tổng quát một DAC

DAC cấu tạo từ 3 phần chức năng:

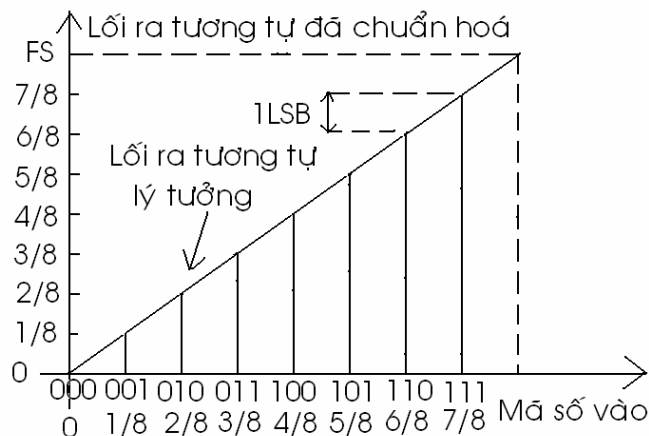
**a/ Điện áp một chiều DCV chuẩn rất ổn định:** DAC sẽ phân chia điện áp này theo các số nhị phân ở đầu vào, để có điện áp ra.

**b/ Lưới điện trở:** Lưới được cân bằng về trị số thế nào đó cho thích hợp với các đầu vào nhị phân 8421.

**c/ Bộ khuếch đại tổng:** Đưa từng điện áp ra theo bảng chân lý. Thông thường bộ khuếch đại tổng là một vi mạch thuật toán. Lưới điện trở sẽ làm thay đổi hệ số khuếch đại của mạch khuếch đại tổng, nhờ vậy ta có điện áp ra tương ứng.

Như vậy, đầu vào DAC là những chuyển mạch tương tự, được điều khiển bởi các số nhị phân mà ta muốn chuyển đổi sang tương tự, khi số nhị phân ở mức 1 chuyển mạch tương ứng đóng lại, ở mức 0 thì mở ra. Các chuyển mạch này sẽ nối các điện trở tương ứng với vi mạch làm thay đổi hệ số khuếch đại của vi mạch và tạo nên dòng điện hoặc điện áp ở đầu ra tương ứng với mã số đầu vào.

Đặc tuyến của một DAC lý tưởng với mã nhị phân 3 bit được mô tả ở hình 7.3.



**Hình 7.3:** Đặc tuyến một DAC lý tưởng 3 bit



Đại lượng ra tương tự không liên tục mà rời rạc, bằng một trong tám giá trị có thể có. Mỗi giá trị ứng với một trong 8 tổ hợp của mã nhị phân 3 bit ở đầu vào.

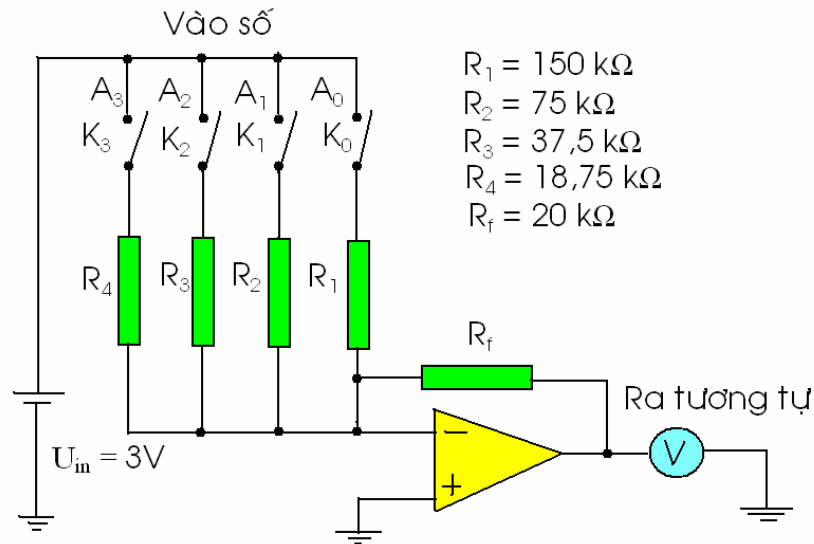
Với DAC 3 bit, điện áp ra từ 0 đến 10V (FS = 10V) điện áp ra cực đại có thể có được là 8,75V. Độ lớn của mỗi nấc điện áp ra ứng với sự chuyển dịch của một bit ở đầu vào là  $\frac{FS}{2^n}$ , mỗi nấc điện áp ra là  $\frac{10V}{2^3} = 1,25V$ .

Tùy từng ứng dụng ta dùng DAC với số bit khác nhau. DAC có số bit càng lớn thì khoảng cách điện áp mỗi nấc càng nhỏ, độ phân giải càng cao. Cũng với FS = 10V DAC 8 bit sẽ có 256 mức điện áp khác nhau, mỗi nấc điện áp là: 3,906 mV.

### 7.2.2. Bộ biến đổi số - tương tự loại điện trở trọng lượng (weighted resistor)

Đây là phương pháp đơn giản nhất để xây dựng một bộ DAC. Mạch DAC loại điện trở trọng lượng gồm có một nguồn điện áp chuẩn, các khóa logic, các điện trở với trị số có tỷ lệ như sau:  $R, R/2, R/4, R/8 \dots R/2^{n-1}$ , một vi mạch tuyến tính làm nhiệm vụ khuếch đại tổng.

Xét một bộ DAC 4 bit loại điện trở trọng lượng với sơ đồ nguyên lý như hình 7.4



**Hình 7.4:** Sơ đồ nguyên lý DAC 4 bit loại điện trở trọng lượng.

$U_{in}$ : Điện áp chuẩn một chiều rất ổn định.

$R_1, R_2, R_3, R_4$ : Lưới điện trở

IC: Bộ khuếch đại tổng, đây là một vi mạch thuật toán

$R_f$ : điện trở hồi tiếp.

$K_0, K_1, K_2, K_3$  Các khóa logic, khi tiếp nhận bit 0 khóa sẽ mở ra làm cho mạch hở, khi tiếp nhận bit 1 khóa đóng lại làm cho điện áp chuẩn  $U_{in}$  được nối với điện trở tương ứng  $R_1, R_2, R_3$  hoặc  $R_4$  Thông qua các khóa ta có thể thay đổi hệ số khuếch đại điện áp của vi mạch, nhờ vậy thay đổi được điện áp ra ( $U_o$ ). Điện áp  $U_o$  này chính là tín hiệu tương tự tương ứng với tín hiệu

số ở đầu vào.

Từ sơ đồ nguyên lý ta thiết lập bảng trạng thái cho DAC trên.

**Dòng 1 (0000):** Tất cả khóa K đều tắt. Điện áp vào bằng 0 do đó điện áp ra bằng 0.

**Dòng 2 (0001):** Khóa  $K_0$  nhận bit 1 nên đóng lại, ta tính điện áp ra của DAC :

Hệ số khuếch đại điện áp của sơ đồ :

$$K_u = \frac{R_f}{R_{in}} = \frac{20000}{150000} = 0,133$$

Từ công thức tính điện áp ra của DAC :

$$U_0 = U_{in} \times K_u = 3 \times 0,133 \approx 0,4 \text{ V}$$

**Dòng 3 (0010):** Khóa  $K_1$  nhận bit 1 nên đóng lại. Tương tự như trên, ta có điện áp ra tương ứng:

$$K_u = \frac{20000}{75000} = 0,266$$

$$V_{out} = 3 \times 0,266 = 0,8 \text{ V}$$

**Dòng 5 (0100):** Khóa  $K_2$  nhận bit 1 nên đóng lại. Tương tự như trên, ta có điện áp ra tương ứng:

$$K_u = \frac{20000}{37500} = 0,533$$

$$V_{out} = 3 \times 0,533 = 1,6 \text{ V}$$

**Dòng 7 (0110):** 2 khóa  $K_1$  và  $K_2$  cùng nhận bit 1 nên cùng đóng. Trường hợp này ta phải tính điện trở vào tương đương:

$$R_{in} = \frac{R_3 R_2}{R_2 + R_3} = 25000 \Omega$$

$$K_u = \frac{20000}{25000} = 0,8$$

$$V_{out} = 3 \times 0,8 = 2,4 \text{ V}$$

Với phương pháp tính tương tự, ta có thể tính được điện áp ra cho tất cả các dòng còn lại.

Từ các kết quả nhận được ở trên, ta viết bảng trạng thái của DAC này như bảng 7.2

**Bảng 7.2: Bảng trạng thái DAC 4 bit**

Dòng	Vào nhị phân				Ra tương tự (V)
	$A_3$	$A_2$	$A_1$	$A_0$	
1	0	0	0	0	0
2	0	0	0	1	0,4
3	0	0	1	0	0,8
4	0	0	1	1	1,2

5	0	1	0	0	1,6
6	0	1	0	1	2,0
7	0	1	1	0	2,4
8	0	1	1	1	2,8
9	1	0	0	0	3,2
10	1	0	0	1	3,6
11	1	0	1	0	4,0
12	1	0	1	1	4,4
13	1	1	0	0	4,8
14	1	1	0	1	5,2
15	1	1	1	0	5,6
16	1	1	1	1	6,0

Ta cũng có thể dùng phương pháp sau để tính điện áp ra cho mỗi dòng:

Từ sơ đồ nguyên lý ta thấy: Khi khóa K nào đó được nối với điện áp nguồn chuẩn  $U_{in}$  thì sẽ cung cấp cho vi mạch tuyến tính một dòng điện có cường độ:

$$U_{ra} = \frac{U_{in}R_f}{R} (a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + a_{n-3}2^{n-3} + \dots + a_02^0) \quad (7.1)$$

trong đó  $A_0 \div A_{n-1}$  có giá trị 0 hoặc 1

**Thí dụ:** Ở đầu vào tiếp nhận một thông tin số 4 bit 1011 (số 11), dưới tác dụng của các bit nhị phân này khóa  $K_0, K_1, K_3$  đóng lại, một dòng điện tổng  $I_{tg}$  chảy qua vi mạch tuyến tính. Ta được :

$$I_{tg} = \frac{U_{in}}{R} + \frac{U_{in}}{\frac{R}{2}} + \frac{U_{ch}}{\frac{R}{8}}$$

$$I_{tg} = U_{in} \left( \frac{1}{R} + \frac{2}{R} + \frac{8}{R} \right) = 11 \cdot \frac{U_{in}}{R}$$

$$\text{Vậy } U_{ra} = I_{tg}R_f = 11 \frac{U_{in}}{R} R_f$$

Thử lại bằng công thức tổng quát (7.1) ta có:

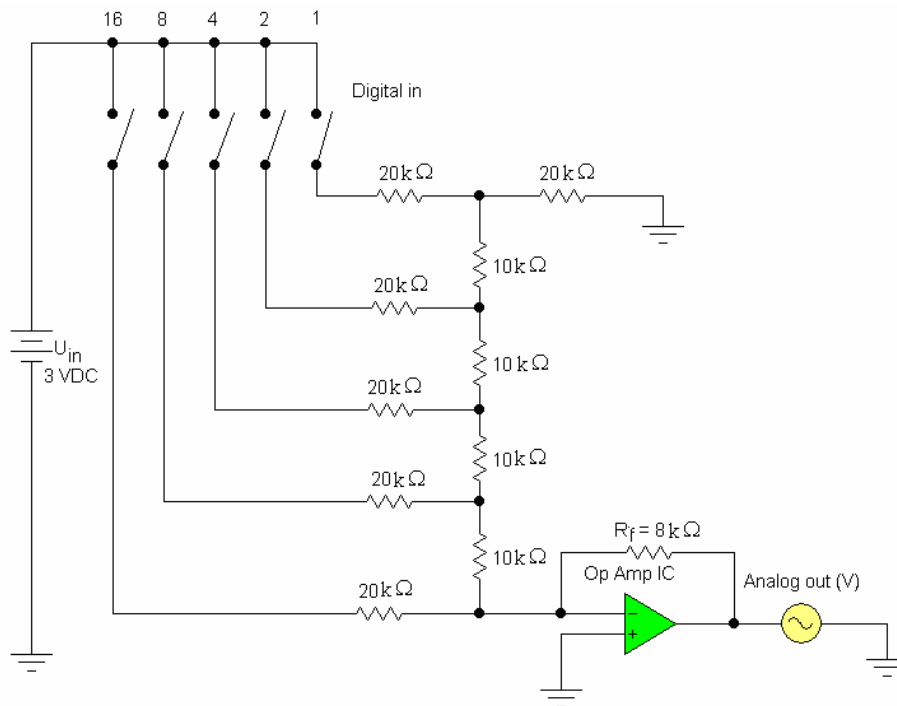
$$U_{ra} = U_{in} \frac{R_f}{R} (2^3 + 2^1 + 2^0) = 11 \frac{U_{in}}{R} R_f$$

Áp dụng công thức tổng quát (7.1) ta dễ dàng viết được bảng trạng thái của các DAC điện trở trọng lượng. Loại DAC điện trở trọng lượng có thể mở rộng cho các thông tin số có số bit cao hơn.

### 7.2.3. Bộ biến đổi số tương tự loại điện trở R - 2R

Bộ DAC loại điện trở trọng lượng có ưu điểm là đơn giản, nhưng có khuyết điểm là độ chính xác và tính ổn định của kết quả phụ thuộc nhiều vào độ chính xác của các điện trở. Khi số bit thông tin số tăng lên thì trị số điện trở của lưới điện trở cũng tăng lên, trị số của các điện trở này phải có tỷ lệ nhất định, tuân theo quy luật 8 4 2 1 cho 4 bit. Chế tạo các điện trở có tỉ lệ đúng như trên rất khó khăn và tốn kém. Ngoài ra, điện áp ra còn phụ thuộc vào cả độ ổn định của điện áp vào.

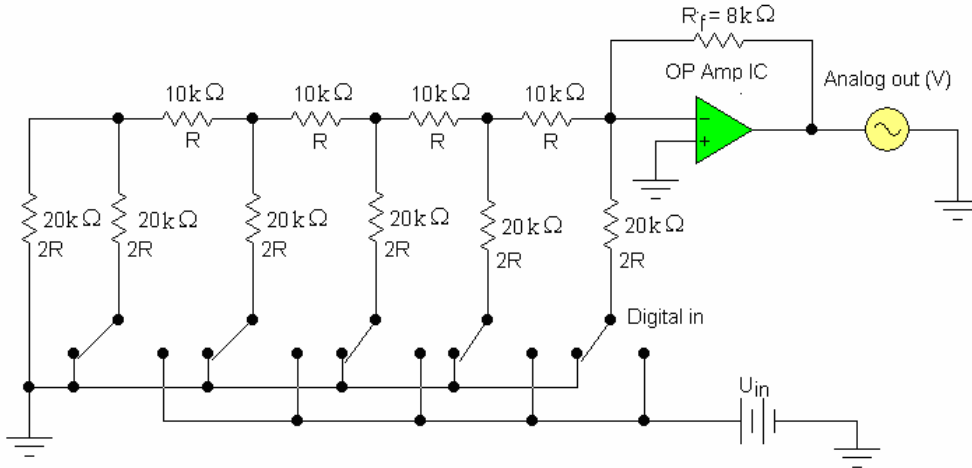
Giải quyết các khuyết điểm của DAC loại điện trở trọng lượng, người ta chế tạo loại DAC R - 2 R. Lưới điện trở của loại này không đòi hỏi tỷ lệ trị số của các điện trở, Các điện trở của lưới điện trở chỉ có 2 trị số là R và 2R (ví dụ 10 k $\Omega$  và 20 k $\Omega$ ) và được gọi là lưới chia điện thế. Độ chính xác của DAC chỉ còn phụ thuộc vào độ ổn định điện áp của nguồn điện vào. Khuyết điểm của loại này là số lượng điện trở nhiều hơn so với DAC loại điện trở trọng lượng với cùng số bit. Hình 7.5 là sơ đồ nguyên lý một DAC 5 bit loại R- 2R.



**Hình 7.5:** Sơ đồ nguyên lý DAC 5 bit loại R-2R

Từ sơ đồ nguyên lý ta thấy đối với DAC R – 2R, các khóa K luôn luôn nối đất dù thông tin số có bit 1 hay là bit 0. Khi ở bit 0 khóa K nối đất thật sự, còn khi ở bit 1 khóa K nối đất qua đầu vào của bộ khuếch đại tổng. Như vậy, dòng điện qua các điện trở nối với khóa K có giá trị cố định đối với mỗi điện trở, không phụ thuộc vào trạng thái của các khóa K. Cũng như DAC điện trở trọng lượng, DAC R – 2R sử dụng nhiều khóa K, thông tin số có bao nhiêu bit ta phải dùng bấy nhiêu khóa K. Khi nhận trạng thái 1 khóa K đầu vào  $U_{in}$ , khi nhận trạng thái 0 khóa K

nối đất. Lưới điện trở của DAC R- 2R có thể vẽ theo mạng điện trở hình T. Sơ đồ nguyên lý của DAC trên có thể vẽ lại như hình 7.6



**Hình 7.6:** Sơ đồ nguyên lý DAC 5 bit R- 2R

Gọi  $I_{tg}$  là dòng điện tổng chảy qua vi mạch khi các khóa K tương ứng với số nhị phân đầu vào đóng lại, Ta có:  $U_{ra} = I_{tg}R_f$

Một cách tổng quát với một DAC R – 2R n bit (từ  $A_0$  cho đến  $A_{n-1}$ ) ta có thể tính theo công thức sau:

$$U_{ra} = U_{in} \frac{R_f}{2^n R} (a^{n-1} 2^{n-1} + a^{n-2} 2^{n-2} + \dots + a^0 2^0) \quad (7.2)$$

Trong đó  $A_0 \div A_{n-1}$  có giá trị 0 hoặc 1

**Thí dụ:** Ở đầu vào DAC tiếp nhận số nhị phân 5 bit 10101 (số 21).

Ta có bit có trọng lượng nhỏ nhất là 1 nên chuyển mạch  $2^0$  nối với  $U_{in}$  và dòng điện do nó cung cấp cho vi mạch tuyến tính:

$$I(2^0) = \frac{U_{in}}{16} \cdot \frac{1}{2R} = \frac{U_{in}}{32R}$$

$$I(2^2) = \frac{U_{in}}{4} \cdot \frac{1}{2R} = \frac{U_{in}}{8R}$$

$$I(2^4) = \frac{U_{in}}{1} \cdot \frac{1}{2R} = \frac{U_{in}}{2R}$$

$$I_{tg} = I(2^0) + I(2^2) + I(2^4) = U_{in} \left( \frac{1}{32} + \frac{1}{8R} + \frac{1}{2R} \right)$$

$$\text{Mà } U_{ra} = R_f I_{tg}$$

$$\text{Nên } U_{ra} = U_{in} R_f \left( \frac{1}{32} + \frac{1}{8R} + \frac{1}{2R} \right) = U_{in} R_f \left( \frac{21}{32R} \right)$$

Nghiệm lại bằng công thức tổng quát 7.2

$$U_{ra} = U_{in} \frac{R_f}{2^5 R} (2^4 + 2^2 + 2^0) = R_f U_{in} \left( \frac{21}{32R} \right)$$

**7.2.4. Độ phân giải**

Độ phân giải của 1 DAC là một đặc trưng quan trọng, được đo bằng số bit tín hiệu vào hay đo bằng của bước nhảy đầu ra so với toàn thang.

**Thí dụ:** DAC 4 bit thì bước nhảy cực tiểu của điện áp đầu ra là 1/15 của toàn thang .

Ta có biểu thức xác định độ phân giải :

$$R_e \% = \frac{1}{2^n - 1} \times 100 \quad \text{với } n \text{ là số bit của tín hiệu vào .}$$

Đối với DAC 4 bit ta có

$$R_e \% = \frac{1}{2^4 - 1} \times 100 = 6,7\%$$

Nghĩa là đối với mỗi một giá trị nhị phân, điện áp ra của DAC biến đổi 6,7% điện áp cực đại đầu ra .

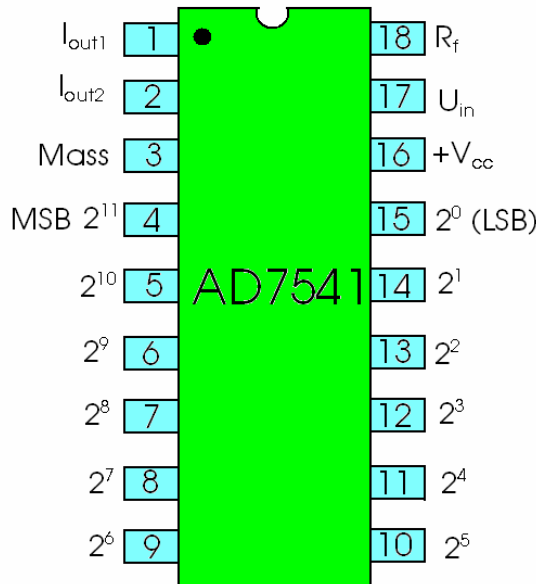
Tương tự độ phân giải của DAC 5 bit :

$$R_e \% = \frac{1}{2^5 - 1} \times 100 = 3,2\%$$

Như vậy so với DAC 4 bit, DAC 5 bit có độ phân giải cao hơn. Một DAC còn được đánh giá bằng độ chính xác, độ tuyến tính và tốc độ hoạt động.

**7.2.5: Vi mạch chuyển đổi số - tương tự.**

Để thực hiện việc chuyển đổi số - tương tự, ta có thể dùng vi mạch AD75421. Đây là vi mạch 12 bit CMOS loại R – 2R, nó có sơ đồ chân như hình 7.7

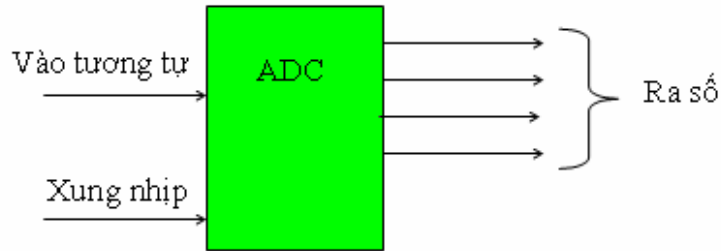


**Hình 7.7:** Sơ đồ chân vi mạch ADC AD7541

### 7.3. BỘ BIẾN ĐỔI TƯƠNG TỰ - SỐ

#### 7.3.1 Khái niệm

Các thông tin (điện áp, dòng điện ...) thường ở dưới dạng tương tự, muốn tiện cho việc xử lý ta phải biến đổi sang dạng số, việc biến đổi này thực hiện nhờ bộ biến đổi tương tự - số ADC (Analog Digital Converter), nghĩa là một điện áp liên tục bất kỳ ở đầu vào sẽ cho ta tín hiệu số tương ứng ở đầu ra (Hình 7.8)

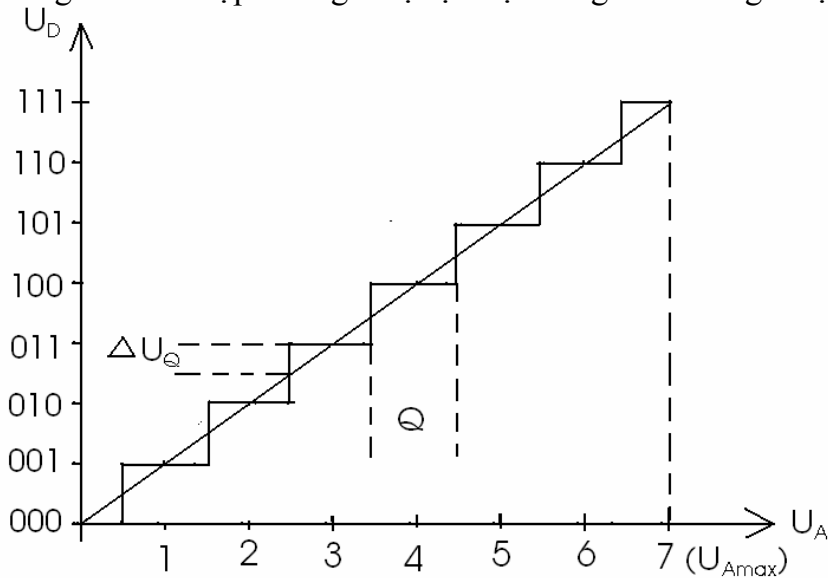


**Hình 7.8:** Bộ biến đổi tương tự - số

**Xung nhịp:** Do một máy phát xung có tần số không đổi tạo ra, các xung này sẽ đưa vào cổng G.

**Lối vào tương tự:** Nơi nhận điện áp tương tự cần biến đổi thành số.

Quá trình biến đổi tín hiệu số sang tương tự được minh họa bởi đặc tuyến truyền đạt của một DAC 3 bit như hình 7.9. Tín hiệu tương tự  $U_A$  được chuyển thành một tín hiệu có dạng bậc thang đều. Với đặc tuyến truyền đạt như vậy, một phạm vi giá trị của  $U_A$  được biểu diễn bằng một giá trị đại diện bằng số thích hợp. Các giá trị đại diện bằng số là các giá trị rời rạc. Với một



**Hình 7.9:** Đặc tuyến truyền đạt của bộ chuyển đổi tương tự - số

ADC N bit thì mỗi nút trên hình thang chiếm một giá trị:

$$Q = U_{LSB} = \frac{U_{Amax}}{2^{N-1}}$$

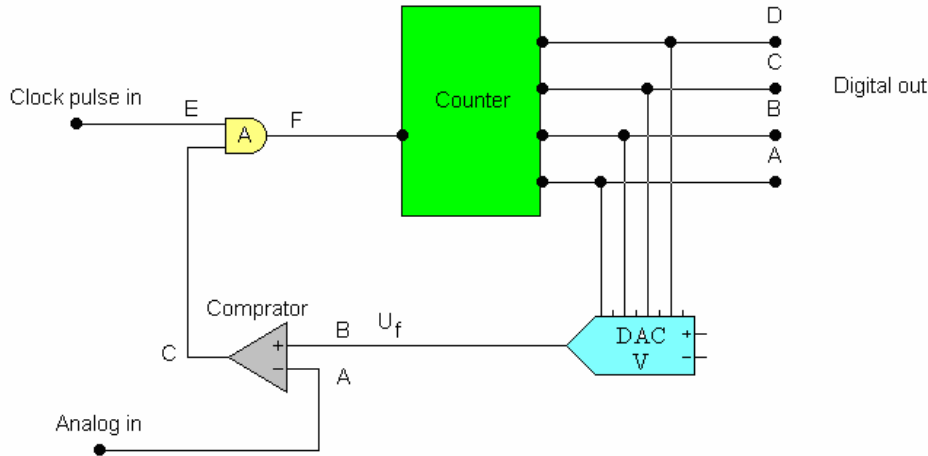
Với  $U_{A_{max}}$  là giá trị cực đại của điện áp tương tự ở đầu vào bộ chuyển đổi.

Giá trị của  $U_{LSB}$  hoặc  $Q$  gọi là mức lượng tử (một nấc của hình thang). Tín hiệu số nhận được ở đầu ra là tín hiệu rời rạc, nên trong quá trình chuyển đổi sẽ xuất hiện một sai số gọi là sai số lượng tử  $\Delta U_Q$ .

$$\Delta U_Q = \frac{Q}{2}$$

### 7.3.2 Bộ biến đổi Tương tự - Số ADC:

Bộ chuyển đổi tương tự - số ADC thường có sơ đồ nguyên lý như sau (Hình 7.10)



**Hình 7.11:** Sơ đồ khối bộ chuyển đổi tương tự - số ADC

Một máy phát xung có tần số không đổi, cung cấp các xung cho cổng G. Cổng này chỉ mở khi lối ra bộ so sánh điện áp là 1. Số xung sau khi qua cổng sẽ được đếm bởi bộ đếm cơ số 2, ngõ ra của bộ đếm cũng là lối ra của ADC, đây cũng chính là tín hiệu số tương ứng với tín hiệu tương tự cần chuyển đổi. Số nhị phân ở lối ra của bộ đếm còn được đưa vào bộ biến đổi Số - Tương tự DAC. Lối ra của DAC sẽ là một điện áp hồi tiếp tỉ lệ với số xung qua cổng. Chừng nào mà điện áp hồi tiếp  $U_f$  còn nhỏ hơn điện áp tương tự  $U_A$  cần biến đổi thì cổng còn mở cho xung vào bộ đếm. Khi điện áp hồi tiếp lớn hơn hoặc bằng điện áp tương tự cần biến đổi thì lối ra của bộ so sánh bằng 0 và cổng G đóng lại không cho xung nhịp vào bộ đếm. Do đó số đếm ghi được ở máy đếm là một số nhị phân biểu diễn điện áp tương tự ở lối vào.

Bảng 7.3 cho ta bảng trạng thái của một ADC 4 bit

**Bảng 7.3:** Bảng trạng thái của ADC 4 bit

Dòng	Vào tương tự (V)	Ra nhị phân			
		$A_3$	$A_2$	$A_1$	$A_0$
1	0	0	0	0	0
2	0,2	0	0	0	1
3	0,4	0	0	1	0
4	0,6	0	0	1	1



5	0,8	0	1	0	0
6	1,0	0	1	0	1
7	1,2	0	1	1	0
8	1,4	0	1	1	1
9	1,6	1	0	0	0
10	1,8	1	0	0	1
11	2,0	1	0	1	0
12	2,2	1	0	1	1
13	2,4	1	1	0	0
14	2,6	1	1	0	1
15	2,8	1	1	1	0
16	3,0	1	1	1	1

Để biết được hoạt động của ADC trên, ta giả thiết đưa vào đầu vào biến đổi một điện áp 0,75 Volt. Theo bảng chân lý đầu ra sẽ là 0100.

Giả sử bảng trạng thái của DAC có trong cấu trúc của ADC như bảng 7.4, ta phân tích các chu kỳ xảy ra trong ADC đang khảo sát. Các chu kỳ này thực tế xảy ra rất nhanh (Bảng 7.5).

**Bảng 7.4:** Bảng trạng thái của DAC trong cấu trúc của ADC đang khảo sát.

Dòng	Vào nhị phân				Ra tương tự (V)
	A <sub>3</sub> 8	A <sub>2</sub> 4	A <sub>1</sub> 2	A <sub>0</sub> 1	
1	0	0	0	0	0
2	0	0	0	1	0,2
3	0	0	1	0	0,4
4	0	0	1	1	0,6
5	0	1	0	0	0,8
6	0	1	0	1	1,0
7	0	1	1	0	1,2
8	0	1	1	1	1,4
9	1	0	0	0	1,6
10	1	0	0	1	1,8
11	1	0	1	0	2,0
12	1	0	1	1	2,2
13	1	1	0	0	2,4
14	1	1	0	1	2,6

15	1	1	1	0	2, 8
16	1	1	1	1	3, 0

**Bảng 7.5: Mô tả hoạt động của một ADC**

Chu kỳ	Đơn vị	Trạng thái hiện tại của các đầu vào	Kết quả ở đầu ra
1	Bộ so sánh	$A = 0,75 \text{ V}$ , $B = 0 \text{ V}$ $A > B$	$C = 1$
	Cổng VÀ Bộ đếm Bộ hiện số ADC	$E = 1$ , $C = 1$ Nhận xung nhịp thứ nhất Đầu ra A hoạt động Nhận số nhị phân 0001	$F = 1$ cổng VÀ mở Đếm xung nhịp thứ nhất Hiển thị 0001 Điện áp ra: 0,2V
2	Bộ so sánh	$A = 0,75 \text{ V}$ , $B = 0,2 \text{ V}$ $A > B$	$C = 1$
	Cổng VÀ Bộ đếm Bộ hiện số DAC	$E = 1$ , $C = 1$ Nhận xung nhịp thứ hai Đầu B hoạt động 0010	$F = 1$ cổng VÀ mở Đếm xung nhịp thứ hai Hiển thị 0010 Điện áp ra: 0,4V
3	Bộ so sánh	$A = 0,75 \text{ V}$ , $C = 0,4 \text{ V}$ $A > B$	$C = 1$
	Cổng VÀ Bộ đếm Bộ hiện số DAC	$E = 1$ , $B = 1$ Nhận xung nhịp thứ ba Đầu A và B hoạt động Nhận số nhị phân 0011	$F = 1$ cổng VÀ mở Đếm xung nhịp thứ ba Hiển thị 0011 Điện áp ra: 0,6V
4	Bộ so sánh	$A = 0,75 \text{ V}$ , $C = 0,6 \text{ V}$ $A > B$	$C = 1$
	Cổng VÀ Bộ đếm Bộ hiện số DAC	$E = 1$ , $B = 1$ Nhận xung nhịp thứ tư Đầu C hoạt động Nhận số nhị phân 0100	$F = 1$ cổng VÀ mở Đếm xung nhịp thứ tư Hiển thị 0100 Điện áp ra: 0,8V
5	Bộ so sánh	$A = 0,75 \text{ V}$ , $B = 0,8 \text{ V}$ $A < B$	$C = 0$
	Cổng VÀ Bộ đếm	$E = 1$ , $C = 0$ Không nhận được xung nhịp	$F = 0$ cổng VÀ đóng lại Bộ đếm dừng đếm

	Bộ hiện số DAC	Đầu C vẫn hoạt động Vẫn nhận số nhị phân 0100	Vẫn hiển thị 0100 Điện áp ra : 0,8V
--	-------------------	--	--

Ở chu kỳ 1, bộ so sánh có  $A = 0,75 \text{ V}$ ,  $B = 0\text{V}$ , đầu ra của bộ so sánh ở mức 1, theo tính chất của cổng VÀ, ta có lối ra của nó ở mức 1 (cả 2 lối vào đều bằng 1). Cổng VÀ cho xung đếm thứ nhất đi vào bộ đếm. Bộ đếm tăng lên 1 (đếm đến 0001). Bộ chỉ thị số hiện số 0001. Bộ biến đổi số tương tự (DAC) nhận số 0001 ở đầu vào cho một điện áp tương ứng 0,2 Volt ở lối ra (xem bảng 7.5). Điện áp này là điện áp hồi tiếp được đưa về đầu vào B của bộ so sánh.

Các chu kỳ tiếp theo cũng tương tự như vậy, cho đến chu kỳ thứ 4 thì điện áp hồi tiếp lấy ra từ bộ biến đổi DAC là 0,8 V. Chu kỳ thứ 5 như sau: Bộ so sánh có đầu  $A = 0,75 \text{ V}$  và đầu  $B = 0,8 \text{ V}$ . Vậy  $A < B$  nên lối ra của bộ so sánh  $C = 0$ . Cổng VÀ có một lối vào nhận mức thấp, đầu ra ở mức 0, cổng VÀ đóng lại, bộ đếm nhị phân không nhận được xung đếm, nó dừng đếm ở 0100. Như vậy khi cho ADC nhận một điện áp tương tự 0,75 V ở lối vào, ta sẽ nhận được số nhị phân 0100 ở lối ra.

Bộ biến đổi ADC trên là 4 bit, cũng như DAC ta cũng có ADC có độ phân giải là 8 bit và 12 bit được dùng rất phổ biến.

## TÀI LIỆU THAM KHẢO

- Võ Tri An.1996. Kỹ thuật điện tử số ứng dụng. NXB Khoa học Kỹ thuật, Hà nội.
- Đặng Văn Chuyét. 2002. Điện tử số. NXB Giáo dục, Hà nội
- Nguyễn Khang Cường. 1987. Kỹ thuật mạch vi. NXB Khoa học Kỹ thuật, Hà nội.
- Phan văn Đường. 2001 Vi mạch - Điện tử số . ĐHSP Huế
- Lương Ngọc Hải. 2004. Kỹ thuật Xung - số . NXB Giáo dục, Hà nội.
- Huỳnh Đắc Thắng.1986. Kỹ thuật số thực hành . NXB Khoa học Kỹ thuật, Hà nội.
- Nguyễn Thuỳ Vân. 2001. Kỹ thuật số. NXB Khoa học kỹ thuật, Hà nội.
- Phạm Minh Việt - Trần Công Nhượng.1999. Kỹ thuật mạch điện tử phi tuyến. NXB Giáo dục, Hà nội
- V.T.Frolkin.1982. Pulse Circuits. Mir Puplishers, Moscou
- Roger L.Tokheim. 1995. Digital Electronic. Mc Graw Hill book, NewYork
- Victor H. Grinch. 1977. Introduction to Intergrated Circuits. Mc Graw Hill book, NewYork
- Motorola. 1992. Digital IC data book
- Microsoft Encarta Reference Library 2002
- Lạc Việt tự điển 2002
- <http://Wikipedia.org>

## MỤC LỤC

<b>CHƯƠNG 1</b>	<b>VI MẠCH (I.C.)</b>	<b>2</b>
1.1.	KHÁI NIỆM MỞ ĐẦU	2
1.2.	ĐẠI CƯƠNG VỀ VI MẠCH	3
1.2.1.	Cấu tạo	3
1.2.2.	Lịch sử vi mạch	5
1.2.3.	Vỏ ngoài của vi mạch	8
1.3.	VI MẠCH TUYẾN TÍNH	10
1.3.1.	Ký hiệu : Vi mạch tuyến tính có ký hiệu như hình 1.10	10
1.3.2.	Mạch khuếch đại vi sai	11
1.3.3.	Sơ đồ nguyên lý một vi mạch tuyến tính:	12
1.3.4.	Các cách mắc cơ bản của vi mạch tuyến tính:	14
1.3.5.	Ứng dụng của Vi mạch thuật toán để thực hiện các phép tính cơ bản:	15
1.3.6.	Ứng dụng vào các bộ khuếch đại	17
1.4.	VI MẠCH LOGIC	22
1.4.1.	Tổng quan:	22
1.4.3.	Một số IC số thông dụng	24
<b>CHƯƠNG 2</b>	<b>CƠ SỞ TOÁN HỌC CỦA ĐIỆN TỬ SỐ</b>	<b>26</b>
2.1	KHÁI NIỆM VỀ THÔNG TIN VÀ MÃ	26
2.1.1	Thông tin :	26
2.1.2.	Phân loại thông tin	26
2.1.3.	Mã ( code )	27
2.2.	CÁC HỆ THỐNG ĐẾM SỐ :	27
2.2.1.	Định nghĩa :	27
2.2.2.	Nguyên lý chung của các hệ đếm :	29
2.2.3.	Phương pháp chuyển đổi giữa các hệ đếm :	30
2.3.	CÁC MÃ NHỊ PHÂN ĐẶC BIỆT	32
2.3.1.	Mã BCD (Binary Coded Decimal)	33
2.3.2.	Mã Gray	34
2.3.3.	Mã đếm vòng	35
2.3.4.	Mã ký tự	35
2.4.	CÁC PHÉP TÍNH SỐ HỌC TRONG HỆ NHỊ PHÂN	37
2.4.1.	Phép cộng nhị phân	37
2.4.2.	Phép trừ nhị phân	37
2.4.3.	Phép nhân nhị phân	39
2.4.4.	Phép chia nhị phân	40
2.5.	PHÉP CỘNG BCD	40
2.5.1:	Cộng hai mã BCD có kết quả nhỏ hơn 10:	40
2.5.2.	Cộng hai mã BCD có kết quả lớn hơn 9	41
2.5.3.	Phép cộng BCD có dấu	41
2.6.	CÁC HÀM ĐẠI SỐ LOGIC	42
2.6.1.	Định nghĩa về đại số Logic :	42
2.6.2.	Các toán tử Logic	43
2.6.3.	Giản đồ Venn	44
2.6.4.	Phương pháp biểu diễn hàm logic	44

<b>CHƯƠNG 3</b>	<b>CÁC CÔNG LOGIC</b>	50
3.1.	KHÁI NIỆM CHUNG	50
3.2	CÁC CÔNG LOGIC CƠ SỞ:	50
3.2.1.	Cổng HOẶC (OR)	50
3.2.2.	Cổng VÀ (AND)	51
3.2.3.	Cổng KHÔNG (NO)	51
3.3.	CÁC CÔNG LOGIC GHÉP	52
3.3.1.	Cổng KHÔNG VÀ (NAND)	52
3.3.2.	Cổng KHÔNG HOẶC (NOR)	52
3.4.	CÔNG KHÁC ĐẦU	52
3.4.1	Cổng HOẶC loại trừ (Exclusive OR)	52
3.4.2	Cổng KHÔNG HOẶC loại trừ (Exclusive NOR)	53
3.5.	CÔNG LOGIC 3 TRẠNG THÁI TS (THREE STATE)	55
3.6	CÁC MẠCH LOGIC CƠ BẢN	57
3.6.1.	Họ DDL (Diode Diode Logic)	57
3.6.2	Họ Logic DTL (Diode Transistor Logic)	57
3.6.3.	Họ logic RTL (Resistor Transistor Logic)	58
3.6.4.	Cổng logic họTTL (Transistor Transistor Logic)	59
3.6.5	Cổng logic họ ECL (Emitter Coupled Logic)	60
3.7.	CÔNG LOGIC CÓ ĐẦU RA 3 TRẠNG THÁI TS (THREE STATE):	61
3.8.	CÁC KHỐI CÔNG LOGIC THÔNG DỤNG	62
3.8.1:	Vì mạch logic loại TTL/LS:	62
3.8.2:	Vì mạch logic loại CMOS:	63
<b>CHƯƠNG IV</b>	<b>TRIGGER</b>	64
<b>CHƯƠNG IV</b>	<b>TRIGGER</b>	64
4.1	KHÁI NIỆM CHUNG:	64
4.1.1.	Mô tả Trigger và hoạt động	64
4.1.2.	Hoạt động của Trigger	64
4.1.3.	Phân loại trigger	66
4.3.	TRIGGER RS (RSFF SET - RESET FLIP FLOP)	67
4.3.1.	RSFF điều khiển trực tiếp:	67
4.3.2.	RSFF điều khiển đồng bộ:	68
4.3.3:	Phương trình đặc trưng của RSFF:	69
4.4.	TRIGGER JK (JUMP KEEP FLIP FLOP)	70
4.4.1	Ký hiệu:	70
4.4.2	Sơ đồ logic:	70
4.4.3.	Tác dụng của các đầu vào đặc biệt:	71
4.5.	TRIGGER D (DELAY FLIP FLOP)	72
4.5.1.	Ký hiệu:	72
4.5.2.	Sơ đồ logic:	72
4.5.3.	Tạo DFF từ JKFF:	73
4.6.	TRIGGER T (TOGGLE)	73
4.6.1.	Ký hiệu:	73
4.6.2.	Tạo TFF từ JKFF:	74
4.7:	TRIGGER CHỦ - TỚ (MASTER – SLAVE)	74
4.7.1:	Khái niệm:	74

4.7.2. RSFF chủ tớ:	75
<b>CHƯƠNG 5</b>	<b>CÁC HỆ LOGIC TỔ HỢP</b>
5.1. KHÁI NIỆM	77
5.2. MÃ HOÁ	78
5.2.1 Khái niệm:	78
5.2.2. Bộ mã hóa thập phân- BCD77	78
5.2.3. Bộ mã hóa bát phân – nhị phân	81
5.3. GIẢI MÃ	81
5.3.1 Khái niệm	81
5.3.2. Bộ giải mã nhị-bát phân	81
5.3.3. Bộ giải mã BCD ra thập phân	82
5.4. CÁC HỆ CHUYỂN ĐỔI MÃ	83
5.4.1. Bộ chuyển đổi mã BCD ra mã 7 thanh (seven segment)	83
5.4.2. Chuyển đổi mã hệ 2 – Gray	86
5.4.3. Chuyển đổi mã Gray sang nhị phân	87
5.5. BỘ SO SÁNH:	88
5.5.1: Khái niệm:	88
5.5.2. Bộ so sánh hai số nhị phân 1 bit	88
5.6 MẠCH SỐ HỌC:	90
5.6.1. Mạch bán cộng	90
5.6.2. Mạch cộng toàn phần (full Adder)	91
<b>CHƯƠNG 6</b>	<b>HỆ LOGIC TUẦN TỰ</b>
6.1. KHÁI NIỆM	94
6.1. KHÁI NIỆM	94
6.2. BỘ ĐẾM	94
6.2.1: Bộ đếm không đồng bộ	95
6.2.2. Bộ đếm đồng bộ	99
6.3. BỘ GHI DỊCH (Shift Register)	102
6.3.1. Bộ ghi dịch một hướng:	102
6.3.2. Bộ ghi dịch hai hướng:	103
6.3.3. Bộ ghi dịch dùng vi mạch:	104
6.3.4. Bộ ghi dịch làm bộ đếm vòng	104
<b>CHƯƠNG 7</b>	<b>CHUYỂN ĐỔI TÍN HIỆU</b>
7.1. TÍN HIỆU TƯƠNG TỰ VÀ TÍN HIỆU SỐ	106
7.2. BỘ BIẾN ĐỔI SỐ - TƯƠNG TỰ	106
7.2.1. Khái niệm	106
7.2.2. Bộ biến đổi số - tương tự loại điện trở trọng lượng (weighted resistor)	109
7.2.3. Bộ biến đổi số tương tự loại điện trở R - 2R	112
7.2.4. Độ phân giải	114
7.2.5: Vi mạch chuyển đổi số - tương tự.	114
7.3. BỘ BIẾN ĐỔI TƯƠNG TỰ - SỐ	115
7.3.1 Khái niệm	115
7.3.2 Bộ biến đổi Tương tự - Số ADC:	116