



# **Chương 1 : Các linh kiện điện tử cơ bản**

## **Mục lục**

- 1 Các linh kiện điện tử cơ bản
  - 1.1 Phân loại điện trở và cách đọc điện trở
  - 1.2 Phân loại tụ điện và cách đọc tụ điện
    - 1.2.1 Tụ hoá
    - 1.2.2 Tụ Tantal
    - 1.2.3 Tụ không phân cực
    - 1.2.4 Tụ điện biến đổi
    - 1.2.5 Tụ chặn
  - 1.3 Cuộn cảm
- 2 Một số các phương pháp kiểm tra thông thường
- 3 Tóm tắt chương
- 4 Tài liệu tham khảo

- 4.1 Bản quyền

### **Các linh kiện điện tử cơ bản**

Như đã đề cập trong phần trước, các linh kiện điện tử cơ bản trong một mạch điện tử bao gồm: điện trở, tụ điện, cuộn cảm. Do đây là các linh kiện cơ bản nên việc đầu tiên khi làm quen với các linh kiện này đó là cách nhận biết các loại linh kiện khác nhau, đồng thời đọc được giá trị các loại linh kiện khác nhau.

### **Phân loại điện trở và cách đọc điện trở**

Như đã đề cập, nói một cách nôm na, điện trở đặc trưng cho tính chất cản trở dòng điện. Chính vì thế, khi sử dụng điện trở cho một mạch điện thì một phần năng lượng điện sẽ bị tiêu hao để duy trì mức độ chuyển dời của dòng điện. Nói một cách khác thì khi điện trở càng lớn thì dòng điện đi qua càng nhỏ và ngược lại khi điện trở nhỏ thì dòng điện dễ dàng được truyền qua. Khi dòng điện cường độ  $I$  chạy qua một vật có điện trở  $R$ , điện năng được chuyển thành nhiệt năng với công suất theo phương trình sau:

$$P = I^2 \cdot R$$

trong đó:

$P$  là công suất, đo theo  $W$

$I$  là cường độ dòng điện, đo bằng  $A$

$R$  là điện trở, đo theo  $\Omega$

Chính vì lý do này, khi phân loại điện trở, người ta thường dựa vào công suất mà phân loại điện trở. Và theo cách phân loại dựa trên công suất, thì điện trở thường được chia làm 3 loại:

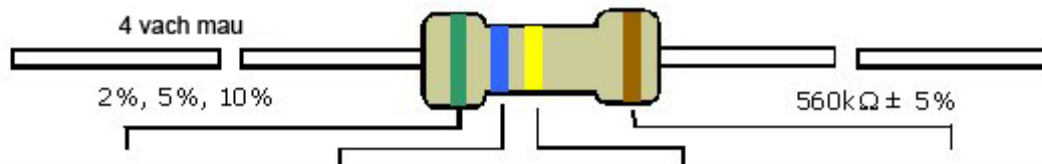
- Điện trở công suất nhỏ
- Điện trở công suất trung bình
- Điện trở công suất lớn.

Tuy nhiên, do ứng dụng thực tế và do cấu tạo riêng của các vật chất tạo nên điện trở nên thông thường, điện trở được chia thành 2 loại:

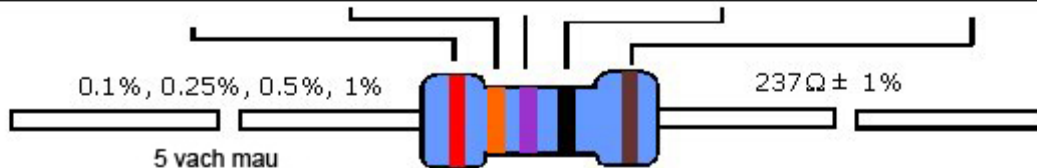
- Điện trở: là các loại điện trở có công suất trung bình và nhỏ hay là các điện trở chỉ cho phép các dòng điện nhỏ đi qua.
- Điện trở công suất: là các điện trở dùng trong các mạch điện tử có dòng điện lớn đi qua hay nói cách khác, các điện trở này khi mạch hoạt động sẽ tạo ra một lượng nhiệt năng khá lớn. Chính vì thế, chúng được cấu tạo nên từ các vật liệu chịu nhiệt.

Để tiện cho quá trình theo dõi trong tài liệu này, các khái niệm điện trở và điện trở công suất được sử dụng theo cách phân loại trên.

Cách đọc giá trị các điện trở này thông thường cũng được phân làm 2 cách đọc, tùy theo các ký hiệu có trên điện trở. Dưới đây là hình vẽ cách đọc điện trở theo vạch màu trên điện trở.



MAU	Vạch 1	Vạch 2	Vạch 3	He so	Dung sai
Đen	0	0	0	1Ω	
Nâu	1	1	1	10Ω	± 1% (F)
Đỏ	2	2	2	100Ω	± 2% (G)
Cam	3	3	3	1KΩ	
Vàng	4	4	4	10KΩ	
Xanh lá cây	5	5	5	100KΩ	±0.5% (D)
Xanh da trời	6	6	6	1MΩ	±0.25% (C)
Tím	7	7	7	10MΩ	±0.10% (B)
Xám	8	8	8		±0.05%
Trắng	9	9	9		
Vàng				0.1	± 5% (J)
Bạc				0.01	± 10% (K)



Nguồn **Electronix Express / RSR**  
<http://www.elexp.com>

Đối với các điện trở có giá trị được định nghĩa theo vạch màu thì chúng ta có 3 loại điện trở: Điện trở 4 vạch màu và điện trở 5 vạch màu và 6 vạch màu. Loại điện trở 4 vạch màu và 5 vạch màu được chỉ ra trên hình vẽ. Khi đọc các giá trị điện trở 5 vạch màu và 6 vạch màu thì chúng ta cần phải để ý một chút vì có sự khác nhau một chút về các giá trị. Tuy nhiên, cách đọc điện trở màu đều dựa trên các giá trị màu sắc được ghi trên điện trở 1 cách tuần tự:

### Đối với điện trở 4 vạch màu

- Vạch màu thứ nhất: Chỉ giá trị hàng chục trong giá trị điện trở

- Vạch màu thứ hai: Chỉ giá trị hàng đơn vị trong giá trị điện trở
- Vạch màu thứ ba: Chỉ hệ số nhân với giá trị số mũ của 10 dùng nhân với giá trị điện trở
- Vạch màu thứ 4: Chỉ giá trị sai số của điện trở

### **Đối với điện trở 5 vạch màu**

- Vạch màu thứ nhất: Chỉ giá trị hàng trăm trong giá trị điện trở
- Vạch màu thứ hai: Chỉ giá trị hàng chục trong giá trị điện trở
- Vạch màu thứ ba: Chỉ giá trị hàng đơn vị trong giá trị điện trở
- Vạch màu thứ 4: Chỉ hệ số nhân với giá trị số mũ của 10 dùng nhân với giá trị điện trở
- Vạch màu thứ 5: Chỉ giá trị sai số của điện trở

Ví dụ như trên hình vẽ, điện trở 4 vạch màu ở phía trên có giá trị màu lần lượt là: xanh lá cây/xanh da trời/vàng/nâu sẽ cho ta một giá trị tương ứng như bảng màu lần lượt là 5/6/4/1%. Ghép các giá trị lần lượt ta có  $56 \times 10^4 \Omega = 560 \text{k}\Omega$  và sai số điện trở là 1%.

Tương tự điện trở 5 vạch màu có các màu lần lượt là: Đỏ/cam/tím/đen/nâu sẽ tương ứng với các giá trị lần lượt là 2/3/7/0/1%. Như vậy giá trị điện trở chính là  $237 \times 10^0 = 237 \Omega$ , sai số 1%.

### **Phân loại tụ điện và cách đọc tụ điện**

Tụ điện theo đúng tên gọi chính là linh kiện có chức năng tích tụ năng lượng điện, nói một cách nôm na. Chúng thường được dùng kết hợp với các điện trở trong các

mạch định thời bởi khả năng tích tụ năng lượng điện trong một khoảng thời gian nhất định. Đồng thời tụ điện cũng được sử dụng trong các nguồn điện với chức năng làm giảm độ gợn sóng của nguồn trong các nguồn xoay chiều, hay trong các mạch lọc bởi chức năng của tụ nói một cách đơn giản đó là tụ ngăn mạch (cho dòng điện đi qua) đối với dòng điện xoay chiều và hở mạch đối với dòng điện 1 chiều.

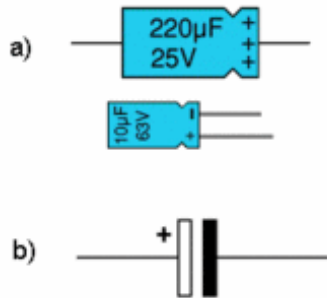
Trong một số các mạch điện đơn giản, để đơn giản hóa trong quá trình tính toán hay thay thế tương đương thì chúng ta thường thay thế một tụ điện bằng một dây dẫn khi có dòng xoay chiều đi qua hay tháo tụ ra khỏi mạch khi có dòng một chiều trong mạch. Điều này khá là cần thiết khi thực hiện tính toán hay xác định các sơ đồ mạch tương đương cho các mạch điện tử thông thường.

Hiện nay, trên thế giới có rất nhiều loại tụ điện khác nhau nhưng về cơ bản, chúng ta có thể chia tụ điện thành hai loại: Tụ có phân cực (có cực xác định) và tụ điện không phân cực (không xác định cực dương âm cụ thể).

Để đặc trưng cho khả năng tích trữ năng lượng điện của tụ điện, người ta đưa ra khái niệm là điện dung của tụ điện. Điện dung càng cao thì khả năng tích trữ năng lượng của tụ điện càng lớn và ngược lại. Giá trị điện dung được đo bằng đơn vị Farad (kí hiệu là F). Giá trị F là rất lớn nên thông thường trong các mạch điện tử, các giá trị tụ chỉ đo bằng các giá trị nhỏ hơn như micro fara ( $\mu\text{F}$ ), nano Fara (nF) hay pico Fara (pF).

$$1\text{F}=10^6\mu\text{F}=10^9\text{nF}=10^{12}\text{pF}$$

## **Tụ hoá**

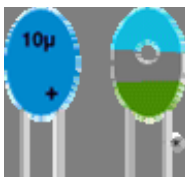


### Kí hiệu tụ hoá và hình dạng tụ hoá

Tụ hóa là một loại tụ có phân cực. Chính vì thế khi sử dụng tụ hóa yêu cầu người sử dụng phải cắm đúng chân của tụ điện với điện áp cung cấp. Thông thường, các loại tụ hóa thường có kí hiệu chân cụ thể cho người sử dụng bằng các ký hiệu + hoặc = tương ứng với chân tụ.

Có hai dạng tụ hóa thông thường đó là tụ hóa có chân tại hai đầu trụ tròn của tụ (tụ có ghi 220µF trên hình a) và loại tụ hóa có 2 chân nối ra cùng 1 đầu trụ tròn (tụ có ghi giá trị 10µF trên hình a). Đồng thời trên các tụ hóa, người ta thường ghi kèm giá trị điện áp cực đại mà tụ có thể chịu được. Nếu trường hợp điện áp lớn hơn so với giá trị điện áp trên tụ thì tụ sẽ bị phồng hoặc nổ tụ tùy thuộc vào giá trị điện áp cung cấp. Thông thường, khi chọn các loại tụ hóa này người ta thường chọn các loại tụ có giá trị điện áp lớn hơn các giá trị điện áp đi qua tụ để đảm bảo tụ hoạt động tốt và đảm bảo tuổi thọ của tụ hóa.

### Tụ Tantal

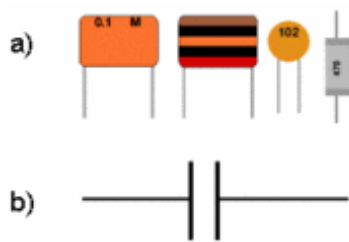


Tụ Tantal



Tụ Tantalum cũng là loại tụ hóa nhưng có điện áp thấp hơn so với tụ hóa. Chúng khá đắt nhưng nhỏ và chúng được dùng khi yêu cầu về tụ dung lớn nhưng kích thước nhỏ.

Các loại tụ Tantalum hiện nay thường ghi rõ trên nó giá trị tụ, điện áp cũng như cực của tụ. Các loại tụ Tantalum ngày xưa sử dụng mã màu để phân biệt. Chúng thường có 3 cột màu (biểu diễn giá trị tụ, một cột biểu diễn giá trị điện áp) và một chấm màu đặc trưng cho số các số không sau dấu phẩy tính theo giá trị  $\mu\text{F}$ . Chúng cũng dùng mã màu chuẩn cho việc định nghĩa các giá trị nhưng đối với các điểm màu thì điểm màu xám có nghĩa là giá trị tụ nhân với 0,01; trắng nhân 0,1 và đen là nhân 1. Cột màu định nghĩa giá trị điện áp thường nằm ở gần chân của tụ và có các giá trị như sau:



Tụ thường và kí hiệu

vàng=6,3V

Đen= 10V

Xanh lá cây= 16V

Xanh da trời= 20V

Xám= 25V

Trắng= 30V

Hồng= 35V

### Tụ không phân cực



Tụ thường

Các loại tụ nhỏ thường không phân cực. Các loại tụ này thường chịu được các điện áp cao mà thông thường là khoảng 50V hay 250V. Các loại tụ không phân cực này có rất nhiều loại và có rất nhiều các hệ thống chuẩn đọc giá trị khác nhau.

Rất nhiều các loại tụ có giá trị nhỏ được ghi thẳng ra ngoài mà không cần có hệ số nhân nào, nhưng cũng có các loại tụ có thêm các giá trị cho hệ số nhân. Ví dụ có các tụ ghi **0.1** có nghĩa giá trị của nó là  $0,1\mu\text{F}=100\text{nF}$  hay có các tụ ghi là **4n7** thì có nghĩa giá trị của tụ đó chính là  $4,7\text{nF}$

### Các loại tụ có dùng mã



Tụ thường

Mã số thường được dùng cho các loại tụ có giá trị nhỏ trong đó các giá trị được định nghĩa lần lượt như sau:

- Giá trị thứ 1 là số hàng chục
- Giá trị thứ 2 là số hàng đơn vị
- Giá trị thứ 3 là số số không nối tiếp theo giá trị của số đã tạo từ giá trị 1 và 2. Giá trị của tụ được đọc theo chuẩn là giá trị pico Fara (pF)
- Chữ số đi kèm sau cùng đó là chỉ giá trị sai số của tụ.

Ví dụ: tụ ghi giá trị 102 thì có nghĩa là 10 và thêm 2 số 0 đằng sau = 1000pF = 1nF chứ không phải 102pF

Hoặc ví dụ tụ 272J thì có nghĩa là 2700pF=2,7nF và sai số là 5%

### Tụ có dùng mã màu



### Tụ dùng mã màu

Sử dụng chủ yếu trên các tụ loại polyester trong rất nhiều năm. Hiện nay các loại tụ này đã không còn bán trên thị trường nữa nhưng chúng vẫn tồn tại trong khá nhiều các mạch điện tử cũ. Màu được định nghĩa cũng tương tự như đối với màu trên điện trở. 3 màu trên cùng lần lượt chỉ giá trị tụ tính theo pF, màu thứ 4 là chỉ dung sai và màu thứ 5 chỉ ra giá trị điện áp.

Ví dụ tụ có màu nâu/đen/cam có nghĩa là  $10000\text{pF} = 10\text{nF} = 0.01\mu\text{F}$ .

Chú ý rằng ko có khoảng trống nào giữa các màu nên thực tế khi có 2 màu cạnh nhau giống nhau thì nó tạo ra một mảng màu rộng. Ví dụ Dải đỏ rộng/vàng= $220\text{nF}=0.22\mu\text{F}$

### Tụ Polyester

Ngày nay, loại tụ này cũng hiếm khi được sử dụng. Giá trị của các loại tụ này thường được in ngay trên tụ theo giá trị pF. Tụ này có một nhược điểm là dễ bị hỏng do nhiệt hàn nóng. Chính vì thế khi hàn các loại tụ này người ta thường có các kỹ thuật riêng để thực hiện hàn, tránh làm hỏng tụ.



Tụ polyester

### Tụ điện biến đổi

Tụ điện biến đổi thường được sử dụng trong các mạch điều chỉnh radio và chúng thường được gọi là tụ xoay. Chúng thường có các giá trị rất nhỏ, thông thường nằm trong khoảng từ  $100\text{pF}$  đến  $500\text{pF}$ .



Tụ xoay

Rất nhiều các tụ xoay có vòng xoay ngấn nên chúng không phù hợp cho các dải biến đổi rộng như là điện trở hoặc các chuyên mạch xoay. Chính vì thế trong nhiều ứng dụng, đặc biệt là trong các mạch định thời hay các mạch điều chỉnh thời gian thì người ta thường thay các tụ xoay bằng các điện trở xoay và kết hợp với 1 giá trị tụ điện xác định.

### **Tụ chặn**

Tụ chặn là các tụ xoay có giá trị rất nhỏ. Chúng thường được gắn trực tiếp lên bản mạch điện tử và điều chỉnh sau khi mạch đã được chế tạo xong. Tương tự các biến trở hiện này thì khi điều chỉnh các tụ chặn này người ta cũng dùng các tuốc nơ vít loại nhỏ để điều chỉnh. Tuy nhiên do giá trị các tụ này khá nhỏ nên khi điều chỉnh, người ta thường phải rất cẩn thận và kiên trì vì trong quá trình điều chỉnh có sự ảnh hưởng của tay và tuốc nơ vít tới giá trị tụ.



### **Tụ chặn**

Các tụ chặn này thường có giá trị rất nhỏ, thông thường nhỏ hơn khoảng 100pF. Có điều đặc biệt là không thể giảm nhỏ được các giá trị tụ chặn về 0 nên chúng thường được chỉ định với các giá trị tụ điện tối thiểu, khoảng từ 2 tới 10 pF.

### **Cuộn cảm**

## INDUCTOR COLOR GUIDE

Result Is In  $\mu\text{H}$

4-BAND-CODE  $\rightarrow$    $\rightarrow$   $270\mu\text{H} \pm 5\%$

COLOR	1st BAND	2nd BAND	MULTIPLIER	TOLERANCE
BLACK	0	0	1	$\pm 20\%$
BROWN	1	1	10	Military $\pm 1\%$
RED	2	2	100	Military $\pm 2\%$
ORANGE	3	3	1,000	Military $\pm 3\%$
YELLOW	4	4	10,000	Military $\pm 4\%$
GREEN	5	5		
BLUE	6	6		
VIOLET	7	7		
GREY	8	8		
WHITE	9	9		
NONE				Military $\pm 20\%$
GOLD			0.1 / Mil. Dec. Pt.	Both $\pm 5\%$
SILVER			0.01	Both $\pm 10\%$

Military Identifier



$6.8\mu\text{H} \pm 10\%$   
MILITARY CODE

Electronix Express / RSR  
<http://www.elexp.com>

1-800-972-2225  
In NJ 732-381-8020

Tương tự như đối với điện trở, trên thế giới có một số loại cuộn cảm có cấu trúc tương tự như điện trở. Quy định màu và cách đọc màu đều tương tự như đối với các điện trở.

Tuy nhiên, do các giá trị của các cuộn cảm thường khá linh động đối với yêu cầu thiết kế mạch cho nên các cuộn cảm thường được tính toán và quấn theo số vòng dây xác định. Với mỗi loại dây, với mỗi loại lõi khác nhau thì giá trị cuộn cảm sẽ khác nhau. Trong phần giáo trình này không đề cập cụ thể tới cách tính toán và quấn các cuộn cảm khác nhau. Phần này sẽ được đề cập cụ thể trong phần sách sau này.

**Một số các phương pháp kiểm tra thông thường**

Để kiểm tra các giá trị tụ điện, cuộn cảm hoặc điện trở thì thông thường mọi người sử dụng các đồng hồ đo đa năng. Hiện nay, có các loại đồng hồ đo đa năng có chức năng đo chính xác các giá trị cuộn cảm, tụ điện và điện trở, điện áp, dòng điện, thậm chí xác định transistor và điốt. Chính vì thế, trong phần này, tôi không đề cập tới các phương pháp kiểm tra cũ (khi dùng đồng hồ cơ/kim) như trước đây.

### **Tóm tắt chương**

Trong chương này, các linh kiện điện tử cơ bản đã được trình bày một cách tương đối cụ thể. Yêu cầu duy nhất đối với người đọc đó là sau khi đọc chương này có thể nắm bắt được và nhận biết được các linh kiện điện tử cơ bản trước khi tìm hiểu và đi sâu hơn vào lĩnh vực điện tử. Yêu cầu nắm vững của phần chương này đó là phân biệt được các linh kiện cơ bản như điện trở, tụ điện, các phương pháp đọc điện trở và cao hơn nữa đó chính là khả năng đọc được giá trị của điện trở, tụ điện,...mà không cần phải tra cứu. Để đạt được điều này, yêu cầu đối với người đọc là phải thực hành so sánh và đọc giá trị các linh kiện thường xuyên.

# **Cơ bản về bán dẫn**

## **Mục lục**

- 1 Các khái niệm cơ bản về bán dẫn
  - 1.1 Bản chất dòng điện trong chất bán dẫn
  - 1.2 Bán dẫn tạp chất và bản chất dòng điện
  - 1.3 Điốt bán dẫn- Phân tử một mặt ghép p-n
    - 1.3.1 Phân cực thuận
    - 1.3.2 Phân cực ngược
    - 1.3.3 Đánh thủng
  - 1.4 Lý thuyết về điốt
    - 1.4.1 Phân loại điốt
    - 1.4.2 Cách kiểm tra Điốt
    - 1.4.3 Một số loại Điốt thông dụng
- 2 Bán dẫn nhiều lớp



- 2.1 Transistor
  - 2.1.1 Hai loại transistor cơ bản
  - 2.1.2 I. Transistor lưỡng cực (BJT)
    - 2.1.2.1 Đọc xong phần này bạn nên có thể:
    - 2.1.2.2 I.1 Transistor chưa phân cực
    - 2.1.2.3 I.1 Transistor đã phân cực
  - 2.1.3 II. Transistor hiệu ứng trường ( FET )
  - 2.1.4 Cách kiểm tra transistor
  - 2.1.5 Một số ứng dụng của Transistor

- 2.2 Thyristor

- 3 Tóm tắt chương
- 4 Câu hỏi tự đánh giá
- 5 Tài liệu tham khảo
  - 5.1 Bản quyền

## **Các khái niệm cơ bản về bán dẫn**

Trong quá trình phân loại vật chất đối với quá trình dẫn điện, người ta chia các vật liệu ra thành ba loại. Đó chính là các vật liệu dẫn điện (như kim loại) và các vật liệu không dẫn điện/cách điện và loại thứ ba là các vật liệu bán dẫn. Các vật liệu dẫn điện là các vật liệu cho phép các dòng điện truyền qua còn các vật liệu cách điện hay không dẫn điện là các vật liệu không cho dòng điện truyền qua.

Chất bán dẫn chủ yếu được cấu tạo từ các nguyên tử có 4 electron lớp ngoài trong cấu trúc nguyên tử của chúng. Như vậy, về bản chất, các chất bán dẫn có 4 electron lớp ngoài cùng mà đặc trưng là 2 chất bán dẫn Ge và Si.

Ở dạng rắn, các nguyên tử cấu tạo nên chất bán dẫn được sắp xếp theo một cấu trúc có thứ tự mà chúng ta gọi là dạng tinh thể. Mỗi nguyên tử chia sẻ các electron của chúng với các nguyên tử ngay cạnh để tạo nên một cấu trúc bền vững có 8 electron lớp ngoài cho nguyên tử nằm tại vị trí trung tâm. Như vậy, mỗi nguyên tử xung quanh nguyên tử trung tâm sẽ chia sẻ 1 electron với nguyên tử trung tâm để tạo thành một cấu trúc bền vững có 8 electron lớp ngoài (đối với nguyên tử trung tâm). Như vậy có thể nói, liên kết giữa nguyên tử trung tâm với 4 nguyên tử xung quanh sẽ dựa trên chủ yếu 4 liên kết hóa trị. Dưới tác dụng của nhiệt, các nguyên tử sẽ tạo ra các dao động xung quanh vị trí cân bằng và tại một giá trị xác định nào đó, nhiệt độ có thể phá vỡ các liên kết hóa trị và tạo ra các electron tự do. Tại vị trí của các electron tự do vừa bứt ra sẽ thiếu 1 electron và trở thành các lỗ trống. Lỗ trống này có xu hướng nhận thêm 1 electron nhằm tạo lại sự cân bằng.

## **Bản chất dòng điện trong chất bán dẫn**

Như đã nói ở trên, trong cấu trúc vật liệu của bản thân chất bán dẫn, dưới tác dụng của nhiệt độ môi trường cũng luôn tồn tại hai dạng điện tích. Một là điện tích âm

do electron và hai là điện tích dương do lỗ trống tạo ra. Dưới tác dụng của điện trường, các electron có xu hướng di chuyển về phía phía có năng lượng điện tích cao hơn. Do đó, lúc này, trong bản chất chất bán dẫn sẽ có 2 thành phần cân bằng. Một là electron tự do bứt ra khỏi liên kết hóa trị và hai là lỗ trống sinh ra do electron bứt ra. Electron bứt ra khỏi cấu trúc tinh thể sẽ di chuyển về phía điện trường có điện thế lớn. Đồng thời, lỗ trống cũng có xu hướng hút các electron ở xung quanh để điền đầy và đi về phía điện trường có điện thế nhỏ hơn. Như vậy, bản chất dòng điện trong chất bán dẫn được sinh ra bởi 2 dòng chuyển dời: dòng chuyển dời của các electron tự do và dòng chuyển dời của các lỗ trống. Các electron và các lỗ trống thường được gọi chung với một cái tên là hạt mang điện bởi chúng mang năng lượng điện tích dịch chuyển từ điểm này đến điểm khác.

### **Bán dẫn tạp chất và bản chất dòng điện**

Như đã biết, bán dẫn tạp chất được tạo ra bởi việc cung cấp các chất tạp chất thuộc nhóm 3 và nhóm 5 bảng tuần hoàn Mendeleev đưa vào trong cấu trúc tinh thể chất bán dẫn thuần.

Để tăng số lượng các electron tự do, thông thường, người ta thêm các tạp chất thuộc nhóm 5 trong bảng tuần hoàn Mendeleev vào. Khi đó, các thành phần tạp chất này sẽ tham gia xây dựng cấu trúc tinh thể của vật chất. Tương tự như giải thích về phần cấu tạo nguyên tử, khi 1 nguyên tử tạp chất đứng cạnh các nguyên tử bán dẫn thuần thì chúng cũng sẽ chia sẻ 1 electron với nguyên tử bán dẫn thuần, do đó sẽ còn 4 electron tại lớp ngoài cùng phân tử. Trong số 4 electron này chỉ có 3 electron tiếp tục tham gia tạo mạng tinh thể và 1 electron sẽ có xu hướng tách ra và trở thành các electron tự do. Do đó, khi so sánh với cấu trúc mạng tinh thể bán dẫn thuần, cấu trúc bán dẫn tạp chất loại này có nhiều các electron tự do hơn. Loại bán dẫn tạp chất này được gọi là bán dẫn loại n (n bản chất tiếng Anh là negative chỉ đặc trưng bản chất của việc thừa electron). Như vậy trong bán dẫn loại n sẽ tồn

tại 2 loại hạt mang điện. Hạt đa số chính là các electron tự do tích điện âm và hạt thiếu số là các lỗ trống (mang điện tích dương).

Tương tự nhưng với hướng ngược lại, người ta thêm tạp chất thuộc nhóm 3 trong bảng tuần hoàn Mendeleev vào trong cấu trúc tinh thể chất bán dẫn thuần. Các thành phần tạp chất này cũng tham gia xây dựng cấu trúc tinh thể của chất bán dẫn, nhưng do chỉ có 3 electron lớp ngoài nên trong cấu trúc nguyên tử sẽ có một vị trí không có electron tham gia xây dựng các liên kết. Các vị trí thiếu này vô hình chung đã tạo nên các lỗ trống. Do đó, trong cấu trúc tinh thể của loại bán dẫn tạp chất này sẽ có nhiều vị trí khuyết electron hơn hay còn gọi là các lỗ trống hơn. Loại bán dẫn này được gọi là bán dẫn loại p (p đặc trưng cho từ positive). Hạt đa số chính là các lỗ trống và hạt thiếu số sẽ là các electron. Tóm lại, bán dẫn loại n có nhiều electron tự do hơn và bán dẫn loại p có nhiều lỗ trống hơn. Do đó, n có khả năng cho electron và p có khả năng nhận electron.

### **Điốt bán dẫn- Phần tử một mặt ghép p-n**

Trong công nghệ chế tạo phần tử 1 mặt ghép p-n, người ta thực hiện pha trộn hai loại bán dẫn tạp chất lên trên một phiến đế tinh thể bán dẫn thuần với một bên là bán dẫn loại p và 1 bên là bán dẫn loại n. Do lực hút lẫn nhau, các electron tự do bên phía bán dẫn loại n có xu hướng khuếch tán theo mọi hướng. Một vài electron tự do khuếch tán vượt qua bề mặt ghép p-n. Khi một electron tự do của bán dẫn loại n đi vào vùng của bán dẫn loại p, nó trở thành hạt thiếu số. Do có một lượng lớn các lỗ trống nên các electron này sẽ nhanh chóng liên kết với lỗ trống để tinh thể trở về trạng thái cân bằng và đồng thời làm lỗ trống biến mất.

Mỗi lần một electron khuếch tán vượt qua vùng tiếp giáp thì nó tạo ra một cặp các ion. Khi một electron rời khỏi miền n thì nó để lại cho cấu trúc nguyên tử tạp chất một (thuộc nhóm 5 bảng tuần hoàn Mendeleev) sang trạng thái mới, trạng thái thiếu một electron. Nguyên tử tạp chất lúc này lại trở thành 1 ion dương. Nhưng

đồng thời, khi đi sang miền p và kết hợp với một lỗ trống thì nó vô hình đã làm nguyên tử tạp chất (thuộc nhóm 3 bảng tuần hoàn Medeleep) trở thành ion âm.

Quá trình này diễn ra liên tục và làm cho vùng tiếp xúc của chất bán dẫn lần lượt có ngày càng nhiều cặp ion dương và âm tương ứng ở miền n và miền p. Các cặp ion này sau khi hình thành sẽ tạo nên một vùng tại miền tiếp xúc bán dẫn mà ta gọi là miền tiếp xúc, có điện trường ngược lại với chiều khuếch tán tự nhiên của các electron tự do và các lỗ trống. Quá trình khuếch tán sẽ dừng khi số lượng các cặp ion sinh ra đủ lớn để cản trở sự khuếch tán tự do của các electron từ n sang p.

Như vậy, ký hiệu âm và dương tại miền tiếp xúc p-n chính là ký hiệu của các cặp ion sinh ra trong quá trình khuếch tán.

## **Phân cực thuận**

## **Phân cực ngược**

## **Đánh thủng**

## **Lý thuyết về điốt**

## **Phân loại điốt**

## **Cách kiểm tra Điốt**

Để kiểm tra một điốt còn khả năng hoạt động hay không, chúng ta có thể sử dụng các đồng hồ đo, đặt chế độ đo điện trở để đo khả năng dẫn dòng điện hay hạn chế dòng điện của điốt. Thông qua đó, chúng ta sẽ biết được điốt còn khả năng sử dụng hay không.

## **Chú ý:**

- Đối với một số loại Ohm kế cũ, dòng hoặc áp của Ohm kế có thể phá hủy 1 số loại diode sử dụng trong các mạch tần số cao.
- Giá trị của thang đo Ohm để xác định khả năng hoạt động của diode thường để khoảng vài trăm KiloOhm.
- Với các đồng hồ Digital Multimeter có chức năng kiểm tra diode, ta có thể sử dụng chức năng này để kiểm tra.

## **Một số loại Điốt thông dụng**

### **Bán dẫn nhiều lớp**

#### **Transistor**

Tín hiệu radio hay vô tuyến thu được từ ăng-ten yếu đến mức nó không đủ để chạy một cái loa hay một đèn điện tử ở tivi. Đây là lý do chúng ta phải khuếch đại tín hiệu yếu để nó có đủ năng lượng để trở nên hữu dụng. Trước năm 1951, ống chân không là thiết bị chính dùng trong việc khuếch đại các tín hiệu yếu. Mặc dù khuếch đại khá tốt, nhưng ống chân không lại có một số nhược điểm. Thứ nhất, nó có một sợi nung bên trong, nó đòi hỏi năng lượng 1 W hoặc hơn. Thứ hai, nó chỉ sống được vài nghìn giờ, trước khi sợi nung hỏng. Thứ ba, nó tốn nhiều không gian. Thứ tư, nó tỏa nhiệt, làm tăng nhiệt độ của các thiết bị điện tử.

Năm 1951, Shockley đã phát minh ra tranzitor có mặt tiếp giáp đầu tiên, một dụng cụ bán dẫn có khả năng khuếch đại các tín hiệu radio và vô tuyến. Các ưu điểm của tranzito khắc phục được các khuyết điểm của ống chân không. Thứ nhất, nó không có sợi nung hay vật làm nóng nào, do đó nó cần ít năng lượng hơn. Thứ hai, do nó là dụng cụ bán dẫn nên có thể sống vô hạn định. Thứ ba, do nó rất nhỏ nên cần ít không gian. Thứ tư, do nó sinh ra ít nhiệt hơn, vì vậy nhiệt độ của các thiết bị điện tử sẽ thấp hơn.

Tranzito đã dẫn tới nhiều phát minh khác, bao gồm: mạch tích hợp (IC), một thiết bị nhỏ chứa hàng ngàn tranzito. Nhờ IC mà máy vi tính và các thiết bị điện tử kỳ diệu khác có thể thực hiện được.

## **Hai loại transistor cơ bản**

Transistor được chia làm 2 loại là transistor lưỡng cực (BJT -Bipolar Junction Trasistor) và transistor hiệu ứng trường (FET- Field Effect Transistor).

### **I. Transistor lưỡng cực (BJT)**

#### **Đọc xong phần này bạn nên có thể:**

- Trình bày những hiểu biết về mối quan hệ giữa các dòng điện bazơ, emitor và collector của một transistor lưỡng cực.
- Vẽ sơ đồ của mạch CE và đánh dấu các cực, điện áp và điện trở.
- Vẽ một đường cong bazơ giả thuyết và tập hợp các đường cong emitor, ghi tên các trục.
- Thảo luận về các đặc tính của transistor lý tưởng và transistor xấp xỉ lần hai.
- Kể ra vài thông số đặc trưng của transistor hữu dụng đối với các nhà kỹ thuật.

#### **I.1 Transistor chưa phân cực**

Một transistor có ba miền pha tạp như trong hình 6.1. Miền dưới cùng được gọi là *emitor*, miền giữa được gọi là *bazơ*, miền trên cùng là *collector*. Loại transistor cụ thể ở đây là một thiết bị npn. Transistor còn có thể được sản xuất như các thiết bị pnp.

#### **Diode emitor và collector**

Transistor ở hình 6.1 có 2 tiếp giáp: một giữa emitter và base và cái kia là giữa base và collector. Do đó transistor tương tự hai diode. emitter và base tạo một diode, base và collector tạo thành một diode khác. Từ giờ, chúng ta sẽ gọi mấy diode này là *diode emitter* (cái dưới) và *diode collector* (cái trên).

### **Trước và sau sự khuếch tán**

Hình 6.1 chỉ ra các miền của transistor trước khi sự khuếch tán xảy ra. Như đã nói đến ở phần trước, electron tự do ở miền n khuếch tán qua vùng tiếp giáp và kết hợp với lỗ trống ở miền p. Hình dung các electron ở mỗi miền n ngang qua phần tiếp giáp và kết hợp với các lỗ trống. Kết quả là hai vùng nghèo như hình 6.2, Mỗi vùng nghèo này hàng rào thế xấp xỉ 0.7 V ở 25°C. Như đã nói, chúng ta nhân mạnh đến các thiết bị silic vì chúng được sử dụng rộng rãi hơn các thiết bị bằng germani.

### **I.1 Transistor đã phân cực**

## **II. Transistor hiệu ứng trường ( FET )**

### **1. Giới thiệu chung về FET**

a.FET hoạt động dựa trên hiệu ứng trường có nghĩa là điện trở của bán dẫn được điều khiển bởi điện trường bên ngoài, dòng điện trong FET chỉ do 1 loại hạt dẫn là electron hoặc lỗ trống tạo nên.

b.Phân loại: FET có 2 loại chính:

- JFET: Transistor trường điều khiển bằng tiếp xúc N-P.
- IGFET: Transistor có cực cửa cách điện, thông thường lớp cách điện này được làm bằng 1 lớp oxit nên có tên gọi khác là MOSFET ( Metal Oxide Semiconductor FET ).



Mỗi loại FET đều có 2 loại kênh N và kênh P. FET có 3 cực là cực Nguồn ( source - S ), cực Máng ( drain - D ), cực Cổng ( gate - G ).

## 2. JFET

### a. Cấu tạo:

JFET được cấu tạo bởi 1 miếng bán dẫn mỏng ( loại N hoặc loại P ) 2 đầu tương ứng là D và S, miếng bán dẫn này được gọi là kênh dẫn điện. 2 miếng bán dẫn ở 2 bên kênh dẫn được nối với cực G, lưu ý, cực G được tách ra khỏi kênh nhờ tiếp xúc N-P.

Đa phần các JFET có cấu tạo đối xứng nên có thể đổi chỗ cực D và S mà tính chất không thay đổi.

### b. Nguyên lý hoạt động

Muốn cho JFET hoạt động ta phải cung cấp  $U_{GS}$  sao cho cả 2 tiếp xúc N-P đều phân cực ngược, nguồn  $U_{DS}$  sao cho dòng hạt dẫn dịch chuyển từ cực S qua kênh tới cực D tạo thành dòng  $I_D$ .

#### - Khả năng điều khiển điện áp $I_D$ của $U_{GS}$ :

Giả sử với JFET kênh N,  $U_{DS} = \text{const}$ . Khi đặt  $U_{GS} = 0$ , tiếp giáp PN bắt đầu phân cực ngược mạnh dần, kênh hẹp dần từ S về D, nhưng lúc này độ rộng kênh là lớn nhất do vậy dòng qua kênh là lớn nhất kí hiệu là  $I_{D0}$ .

Khi  $U_{GS} < 0$ , PN phân cực ngược mạnh hơn do vậy bề rộng của kênh dẫn hẹp dần, tại thời điểm  $U_{GS} = U_{ngắt}$  thì 2 tiếp giáp PN phủ lên nhau, che lấp hết kênh, dòng  $I_D = 0$ . Dòng  $I_D$  được tính theo công thức:  $I_D = I_{D0} (1 - U_{GS}/U_{ngắt})^2$

Chú ý : giá trị của  $U_{ngắt}$  và  $I_{D0}$  phụ thuộc vào  $U_{DS}$ .

## **Cách kiểm tra transistor**

Đối với transistor nói chung, do cấu tạo của transistor gồm 2 tiếp xúc P-N nên có thể coi là 2 diode nối tiếp nhau từ đó có thể kiểm tra sự hoạt động của transistor tương tự như kiểm tra diode.

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**



# **ĐIỆN TỬ SỐ**

*(Dùng cho sinh viên hệ đào tạo đại học từ xa)*

**Lưu hành nội bộ**

**HÀ NỘI - 2006**

# ĐIỆN TỬ SỐ

Biên soạn : ThS. TRẦN THỊ THÚY HÀ

# LỜI GIỚI THIỆU

Cùng với sự tiến bộ của khoa học và công nghệ, các thiết bị điện tử đang và sẽ tiếp tục được ứng dụng ngày càng rộng rãi và mang lại hiệu quả cao trong hầu hết các lĩnh vực kinh tế kỹ thuật cũng như đời sống xã hội.

Việc xử lý tín hiệu trong các thiết bị điện tử hiện đại đều dựa trên cơ sở nguyên lý số. Bởi vậy việc hiểu sâu sắc về điện tử số là điều không thể thiếu được đối với kỹ sư điện tử hiện nay. Nhu cầu hiểu biết về kỹ thuật số không phải chỉ riêng đối với các kỹ sư điện tử mà còn đối với nhiều cán bộ kỹ thuật chuyên ngành khác có sử dụng các thiết bị điện tử.

Tài liệu này giới thiệu một cách hệ thống các phần tử cơ bản trong các mạch điện tử số kết hợp với các mạch điển hình, giải thích các khái niệm cơ bản về công nghệ điện tử số, các phương pháp phân tích và thiết kế mạch logic cơ bản.

Tài liệu bao gồm các kiến thức cơ bản về mạch công logic, cơ sở đại số logic, mạch logic tổ hợp, các trigơ, mạch logic tuần tự, các mạch phát xung và tạo dạng xung, các bộ nhớ thông dụng. Đặc biệt là trong tài liệu này có bổ xung thêm phần logic lập trình và ngôn ngữ mô tả phần cứng VHDL. Đây là ngôn ngữ phổ biến hiện nay dùng để tạo mô hình cho các hệ thống kỹ thuật số. Tất cả gồm 9 chương. Trước và sau mỗi chương đều có phần giới thiệu và phần tóm tắt để giúp người học dễ nắm bắt kiến thức hơn. Các câu hỏi ôn tập để người học kiểm tra mức độ nắm kiến thức sau khi học mỗi chương. Trên cơ sở các kiến thức căn bản, tài liệu đã cố gắng tiếp cận các vấn đề hiện đại, đồng thời liên hệ với thực tế kỹ thuật.

Tài liệu gồm có 9 chương được bố cục như sau:

Chương 1: Hệ đếm

Chương 2: Đại số Boole và các phương pháp biểu diễn hàm

Chương 3: Công nghệ logic TTL và CMOS

Chương 4: Mạch logic tổ hợp.

Chương 5: Mạch logic tuần tự.

Chương 6: Mạch phát xung và tạo dạng xung.

Chương 7: Bộ nhớ bán dẫn.

Chương 8: Logic lập trình.

Chương 9 : Ngôn ngữ mô tả phần cứng VHDL.

Do thời gian có hạn nên tài liệu này không tránh khỏi thiếu sót, rất mong người đọc góp ý. Các ý kiến xin gửi về Khoa Kỹ thuật Điện tử 1- Học viện Công nghệ Bưu chính viễn thông.

Xin trân trọng cảm ơn.

## CHƯƠNG 1: HỆ ĐẾM

### GIỚI THIỆU

Khi nói đến số đếm, người ta thường nghĩ ngay đến hệ thập phân với 10 chữ số được ký hiệu từ 0 đến 9. Máy tính hiện đại không sử dụng số thập phân, thay vào đó là số nhị phân với hai ký hiệu là 0 và 1. Khi biểu diễn các số nhị phân rất lớn, người ta thay nó bằng các số bát phân (Octal) và thập lục phân (Hexadecimal).

Đếm số lượng của các đại lượng là một nhu cầu của lao động, sản xuất. Ngừng một quá trình đếm, ta được một biểu diễn số. Các phương pháp đếm và biểu diễn số được gọi là **hệ đếm**. Hệ đếm không chỉ được dùng để biểu diễn số mà còn là công cụ xử lý.

Có rất nhiều hệ đếm, chẳng hạn như hệ La Mã, La Tinh ... Hệ đếm vừa có tính đa dạng vừa có tính đồng nhất và phổ biến. Mỗi hệ đếm có ưu điểm riêng của nó nên trong kỹ thuật số sẽ sử dụng một số hệ để bổ khuyết cho nhau.

Trong chương này không chỉ trình bày các hệ thập phân, hệ nhị phân, hệ bát phân, hệ thập lục phân và còn nghiên cứu cách chuyển đổi giữa các hệ đếm. Chương này cũng đề cập đến số nhị phân có dấu và khái niệm về dấu phẩy động.

### NỘI DUNG

#### 1.1. BIỂU DIỄN SỐ

Nguyên tắc chung của biểu diễn là dùng một số hữu hạn các ký hiệu ghép với nhau theo qui ước về vị trí. Các ký hiệu này thường được gọi là chữ số. Do đó, người ta còn gọi hệ đếm là hệ thống số. Số ký hiệu được dùng là cơ số của hệ ký hiệu là  $r$ . Giá trị biểu diễn của các chữ khác nhau được phân biệt thông qua trọng số của hệ. Trọng số của một hệ đếm bất kỳ sẽ bằng  $r^i$ , với  $i$  là một số nguyên dương hoặc âm.

Bảng 1.1 là liệt kê tên gọi, số ký hiệu và cơ số của một vài hệ đếm thông dụng.

Tên hệ đếm	Số ký hiệu	Cơ số ( $r$ )
Hệ nhị phân (Binary)	0, 1	2
Hệ bát phân (Octal)	0, 1, 2, 3, 4, 5, 6, 7	8
Hệ thập phân (Decimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	10
Hệ thập lục phân (Hexadecimal)	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F	16

Bảng 1.1

Người ta cũng có thể gọi hệ đếm theo cơ số của chúng. Ví dụ: Hệ nhị phân = Hệ cơ số 2, Hệ thập phân = Hệ cơ số 10...

Dưới đây, ta sẽ trình bày tóm tắt một số hệ đếm thông dụng.

### 1.1.1 Hệ thập phân

Các ký hiệu của hệ như đã nêu ở bảng 1.1. Khi ghép các ký hiệu với nhau ta sẽ được một biểu diễn. Ví dụ: 1265,34 là biểu diễn số trong hệ thập phân:

$$1265.34 = 1 \times 10^3 + 2 \times 10^2 + 6 \times 10^1 + 5 \times 10^0 + 3 \times 10^{-1} + 4 \times 10^{-2}$$

Trong phân tích trên,  $10^n$  là **trọng số của hệ**; các hệ số nhân chính là **ký hiệu** của hệ. Như vậy, giá trị biểu diễn của một số trong hệ thập phân sẽ bằng tổng các tích của ký hiệu (có trong biểu diễn) với trọng số tương ứng. Một cách tổng quát:

$$\begin{aligned} N_{10} &= d_{n-1} \times 10^{n-1} + \dots + d_1 \times 10^1 + d_0 \times 10^0 + d_{-1} \times 10^{-1} + \dots + d_{-m} \times 10^{-m} \\ &= \sum_{i=n-1}^{-m} d_i \times 10^i \end{aligned}$$

trong đó,  $N_{10}$ : biểu diễn bất kì theo hệ 10,

$d$ : các hệ số nhân (ký hiệu bất kì của hệ),

$n$ : số chữ số ở phần nguyên,

$m$ : số chữ số ở phần phân số.

**Ưu điểm** của hệ thập phân là tính truyền thống đối với con người. Đây là hệ mà con người dễ nhận biết nhất. Ngoài ra, nhờ có nhiều ký hiệu nên khả năng biểu diễn của hệ rất lớn, cách biểu diễn gọn, tốn ít thời gian viết và đọc.

**Nhược điểm chính** của hệ là do có nhiều ký hiệu nên việc thể hiện bằng thiết bị kỹ thuật sẽ khó khăn và phức tạp.

#### Biểu diễn số tổng quát:

Với cơ số bất kì  $r$  và  $d$  bằng hệ số  $a$  tùy ý ta sẽ có công thức biểu diễn số chung cho tất cả các hệ đếm:

$$\begin{aligned} N &= a_{n-1} \times r^{n-1} + \dots + a_1 \times r^1 + a_0 \times r^0 + a_{-1} \times r^{-1} + \dots + a_{-m} \times r^{-m} \\ &= \sum_{i=n-1}^{-m} a_i \times r^i \end{aligned}$$

Trong một số trường hợp, ta phải thêm chỉ số để tránh nhầm lẫn giữa biểu diễn của các hệ. Ví dụ:  $36_{10}$ ,  $36_8$ ,  $36_{16}$ .

### 1.1.2 Hệ nhị phân

#### 1.1.2.1. Tổ chức hệ nhị phân

Hệ nhị phân (Binary number system) còn gọi là hệ cơ số hai, gồm chỉ hai ký hiệu 0 và 1, cơ số của hệ là 2, trọng số của hệ là  $2^n$ . Cách đếm trong hệ nhị phân cũng tương tự như hệ thập phân. Khởi đầu từ giá trị 0, sau đó ta cộng liên tiếp thêm 1 vào kết quả đếm lần trước. Nguyên tắc cộng nhị phân là:  $0 + 0 = 0$ ,  $1 + 0 = 1$ ,  $1 + 1 = 10$  ( $10_2 = 2_{10}$ ).

Trong hệ nhị phân, mỗi chữ số chỉ lấy 2 giá trị hoặc 0 hoặc 1 và được gọi tắt là "bit". Như vậy, bit là số nhị phân 1 chữ số. Số bit tạo thành độ dài biểu diễn của một số nhị phân. Một số nhị phân có độ dài 8 bit được gọi là 1 byte. Số nhị phân hai byte gọi là một từ (word). Bit tận cùng bên phải gọi là bit bé nhất (LSB – Least Significant Bit) và bit tận cùng bên trái gọi là bit lớn nhất (MSB - Most Significant Bit).

Biểu diễn nhị phân dạng tổng quát :

$$N_2 = b_{n-1}b_{n-2} \dots b_1b_0.b_{-1}b_{-2} \dots b_{-m}$$

Trong đó, b là hệ số nhân của hệ. Các chỉ số của hệ số đồng thời cũng bằng lũy thừa của trọng số tương ứng. Ví dụ :

$$\begin{array}{cccccc} 1 & 1 & 0 & 0 & 0 & \rightarrow \text{số nhị phân phân số} \\ 2^2 & 2^1 & 2^0 & 2^{-1} & 2^{-2} & \rightarrow \text{trọng số tương ứng.} \end{array}$$

Các giá trị  $2^{10} = 1024$  được gọi là 1Kbit,  $2^{20} = 1048576$  - Mêga Bit ...

Ta có dạng tổng quát của biểu diễn nhị phân như sau:

$$\begin{aligned} N_2 &= b_{n-1} \times 2^{n-1} + \dots + b_1 \times 2^1 + b_0 \times 2^0 + b_{-1} \times 2^{-1} + \dots + b_{-m} \times 2^{-m} \\ &= \sum_{i=n-1}^{-m} b_i \times 2^i \end{aligned}$$

Trong đó, b là hệ số nhân lấy các giá trị 0 hoặc 1.

### 1.1.2.2. Các phép tính trong hệ nhị phân

#### a. Phép cộng

Qui tắc cộng hai số nhị phân 1 bit đã nêu ở trên.

#### b. Phép trừ

Qui tắc trừ hai bit nhị phân cho nhau như sau :

$$0 - 0 = 0 ; 1 - 1 = 0 ; 1 - 0 = 1 ; 10 - 1 = 1 \text{ (mượn 1)}$$

Khi trừ nhiều bit nhị phân, nếu cần thiết ta mượn bit kế tiếp có trọng số cao hơn. Lần trừ kế tiếp lại phải trừ thêm 1.

#### c. Phép nhân

Qui tắc nhân hai bit nhị phân như sau:

$$0 \times 0 = 0 , 0 \times 1 = 0 , 1 \times 0 = 0 , 1 \times 1 = 1$$

Phép nhân hai số nhị phân cũng được thực hiện giống như trong hệ thập phân.

Chú ý : Phép nhân có thể thay bằng phép dịch và cộng liên tiếp.

#### d. Phép chia

Phép chia nhị phân cũng tương tự như phép chia hai số thập phân.

**Ưu điểm** chính của hệ nhị phân là chỉ có hai ký hiệu nên rất dễ thể hiện bằng các thiết bị cơ, điện. Các máy vi tính và các hệ thống số đều dựa trên cơ sở hoạt động nhị phân (2 trạng thái). Do



đó, hệ nhị phân được xem là ngôn ngữ của các mạch logic, các thiết bị tính toán hiện đại - ngôn ngữ máy.

*Nhược điểm* của hệ là biểu diễn dài, mất nhiều thời gian viết, đọc.

### 1.1.3 Hệ bát phân và thập lục phân

#### 1.1.3.1 Hệ bát phân

**1. Tổ chức của hệ :** Nhằm khắc phục nhược điểm của hệ nhị phân, người ta thiết lập các hệ đếm có nhiều ký hiệu hơn, nhưng lại có quan hệ chuyển đổi được với hệ nhị phân. Một trong số đó là hệ bát phân (hay hệ Octal, hệ cơ số 8).

Hệ này gồm 8 ký hiệu : 0, 1, 2, 3, 4, 5, 6 và 7. Cơ số của hệ là 8. Việc lựa chọn cơ số 8 là xuất phát từ chỗ  $8 = 2^3$ . Do đó, mỗi chữ số bát phân có thể thay thế cho 3 bit nhị phân.

Dạng biểu diễn tổng quát của hệ bát phân như sau:

$$\begin{aligned} N_8 &= O_{n-1} \times 8^{n-1} + \dots + O_0 \times 8^0 + O_{-1} \times 8^{-1} + \dots + O_{-m} \times 8^{-m} \\ &= \sum_{i=-m}^{n-1} O_i \times 8^i \end{aligned}$$

Lưu ý rằng, hệ thập phân cũng đếm tương tự và có giải rộng hơn hệ bát phân, nhưng không thể tìm được quan hệ  $10 = 2^n$  (với n nguyên).

#### 2. Các phép tính trong hệ bát phân

##### a. Phép cộng

Phép cộng trong hệ bát phân được thực hiện tương tự như trong hệ thập phân. Tuy nhiên, khi kết quả của việc cộng hai hoặc nhiều chữ số cùng trọng số lớn hơn hoặc bằng 8 phải nhớ lên chữ số có trọng số lớn hơn kế tiếp.

##### b. Phép trừ

Phép trừ cũng được tiến hành như trong hệ thập phân. Chú ý rằng khi mượn 1 ở chữ số có trọng số lớn hơn thì chỉ cần cộng thêm 8 chứ không phải cộng thêm 10.

Các phép tính trong hệ bát phân ít được sử dụng. Do đó, phép nhân và phép chia dành lại như một bài tập cho người học.

#### 1.1.3.2 Hệ thập lục phân

##### 1. Tổ chức của hệ

Hệ thập lục phân (hay hệ Hexadecimal, hệ cơ số 16). Hệ gồm 16 ký hiệu là 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Trong đó,  $A = 10_{10}$ ,  $B = 11_{10}$ ,  $C = 12_{10}$ ,  $D = 13_{10}$ ,  $E = 14_{10}$ ,  $F = 15_{10}$ .

Cơ số của hệ là 16, xuất phát từ yếu tố  $16 = 2^4$ . Vậy, ta có thể dùng một từ nhị phân 4 bit (từ 0000 đến 1111) để biểu thị các ký hiệu thập lục phân. Dạng biểu diễn tổng quát:

$$N_{16} = H_{n-1} \times 16^{n-1} + \dots + H_0 \times 16^0 + H_{-1} \times 16^{-1} + \dots + H_{-m} \times 16^{-m}$$

$$= \sum_{i=n-1}^{-m} H_i \times 16^i$$

## 2. Các phép tính trong hệ cơ số 16

### a. Phép cộng

Khi tổng hai chữ số lớn hơn 15, ta lấy tổng chia cho 16. Số dư được viết xuống chữ số tổng và số thương được nhớ lên chữ số kế tiếp. Nếu các chữ số là A, B, C, D, E, F thì trước hết, ta phải đổi chúng về giá trị thập phân tương ứng rồi mới cộng.

### b. Phép trừ

Khi trừ một số bé hơn cho một số lớn hơn ta cũng mượn 1 ở cột kế tiếp bên trái, nghĩa là cộng thêm 16 rồi mới trừ.

### c. Phép nhân

Muốn thực hiện phép nhân trong hệ 16 ta phải đổi các số trong mỗi thừa số về thập phân, nhân hai số với nhau. Sau đó, đổi kết quả về hệ 16.

## 1.2. CHUYỂN ĐỔI CƠ SỐ GIỮA CÁC HỆ ĐẾM

### 1.2.1. Chuyển đổi từ hệ cơ số 10 sang các hệ khác

Để thực hiện việc đổi một số thập phân đầy đủ sang các hệ khác ta phải chia ra hai phần: phần nguyên và phần số.

**Đối với phần nguyên:** ta chia liên tiếp phần nguyên của số thập phân cho cơ số của hệ cần chuyển đến, số dư sau mỗi lần chia viết đảo ngược trật tự là kết quả cần tìm. Phép chia dừng lại khi kết quả lần chia cuối cùng bằng 0.

Ví dụ: Đổi số  $57_{10}$  sang số nhị phân.

Bước	chia	được	dư	
1	57/2	28	1	→ LSB
2	28/2	14	0	
3	14/2	7	0	
4	7/2	3	1	
5	3/2	1	1	
6	1/2	0	1	→ MSB

Viết đảo ngược trật tự, ta có :  $57_{10} = 111001_2$

**Đối với phần phân số :** ta nhân liên tiếp phần phân số của số thập phân với cơ số của hệ cần chuyển đến, phần nguyên thu được sau mỗi lần nhân, viết tuần tự là kết quả cần tìm. Phép nhân dừng lại khi phần phân số triệt tiêu.

**Ví dụ:** Đổi số  $57,34375_{10}$  sang số nhị phân.

Phần nguyên ta vừa thực hiện ở ví dụ a), do đó chỉ cần đổi phần phân số 0,375.

Bước	Nhân	Kết quả	Phần nguyên
1	0,375 x 2	0.75	0
2	0,75 x 2	1.5	1
3	0,5 x 2	1.0	1
4	0,0 x 2	0	0

Kết quả :  $0,375_{10} = 0,0110_2$

Sử dụng phần nguyên đã có ở ví dụ 1) ta có :  $57,375_{10} = 111001,0110_2$

### 1.2.2. Đổi một biểu diễn trong hệ bất kì sang hệ thập phân

Muốn thực hiện phép biến đổi, ta dùng công thức :

$$N_{10} = a_{n-1} \times r^{n-1} + \dots + a_0 \times r^0 + a_{-1} \times r^{-1} + \dots + a_{-m} \times r^{-m}$$

Thực hiện lấy tổng về phải sẽ có kết quả cần tìm. Trong biểu thức trên,  $a_i$  và  $r$  là hệ số và cơ số hệ có biểu diễn.

### 1.2.3. Đổi các số từ hệ nhị phân sang hệ cơ số 8 và 16

Vì  $8 = 2^3$  và  $16 = 2^4$  nên ta chỉ cần dùng một số nhị phân 3 bit là đủ ghi 8 ký hiệu của hệ cơ số 8 và từ nhị phân 4 bit cho hệ cơ số 16.

Do đó, muốn đổi một số nhị phân sang hệ cơ số 8 và 16 ta chia số nhị phân cần đổi, kể từ dấu phân số sang trái và phải thành từng nhóm 3 bit hoặc 4 bit. Sau đó thay các nhóm bit đã phân bằng ký hiệu tương ứng của hệ cần đổi tới.

**Ví dụ:**

#### a. Đổi số 110111,0111<sub>2</sub> sang số hệ cơ số 8

Tính từ dấu phân số, ta chia số này thành các nhóm 3 bit như sau :

$$\begin{array}{cccc} 110 & 111 & , & 011 & 100 \\ \downarrow & \downarrow & & \downarrow & \downarrow \\ 6 & 7 & & 3 & 4 \end{array}$$

Kết quả:  $110111,0111_2 = 67,34_8$ . ( Ta đã thêm 2 số 0 để tiện biến đổi).

#### b. Đổi số nhị phân 11110110,0110<sub>2</sub> sang số hệ cơ số 16

Ta phân nhóm và thay thế như sau :

$$\begin{array}{cccccc} 0001 & 1111 & 0110 & 0110 & 1000 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & F & 6 & 6 & 8 \end{array}$$

Kết quả:  $11110110,0110_2 = 1F6,68_{16}$

## 1.3 SỐ NHỊ PHÂN CÓ DẤU

### 1.3.1 Biểu diễn số nhị phân có dấu

Có ba phương pháp thể hiện số nhị phân có dấu sau đây.

**1. Sử dụng một bit dấu.** Trong phương pháp này ta dùng một bit phụ, đứng trước các bit trị số để biểu diễn dấu, '0' chỉ dấu dương (+), '1' chỉ dấu âm (-).

**2. Sử dụng phép bù 1.** Giữ nguyên bit dấu và lấy bù 1 các bit trị số (bù 1 bằng đảo của các bit cần được lấy bù).

#### 3. Sử dụng phép bù 2

Là phương pháp phổ biến nhất. Số dương thể hiện bằng số nhị phân không bù (bit dấu bằng 0), còn số âm được biểu diễn qua bù 2 (bit dấu bằng 1). Bù 2 bằng bù 1 cộng 1.

Có thể biểu diễn số âm theo phương pháp bù 2 xen kẽ: bắt đầu từ bit LSB, dịch về bên trái, giữ nguyên các bit cho đến gặp bit 1 đầu tiên và lấy bù các bit còn lại. Bit dấu giữ nguyên.

### 1.3.2 Các phép cộng và trừ số nhị phân có dấu

Như đã nói ở trên, phép bù 1 và bù 2 thường được áp dụng để thực hiện các phép tính nhị phân với số có dấu.

#### 1. Biểu diễn theo bit dấu

##### a. Phép cộng

Hai số cùng dấu: cộng hai phần trị số với nhau, còn dấu là dấu chung.

Hai số khác dấu và **số âm** có trị số **nhỏ hơn**: cộng trị số của số dương với bù 1 của số âm. Bit tràn được cộng thêm vào kết quả trung gian. Dấu là dấu dương.

Hai số khác dấu và **số âm** có trị số **lớn hơn**: cộng trị số của số dương với bù 1 của số âm. Lấy bù 1 của tổng trung gian. Dấu là dấu âm.

**b. Phép trừ.** Nếu lưu ý rằng,  $- (-) = +$  thì trình tự thực hiện phép trừ trong trường hợp này cũng giống phép cộng.

#### 2. Cộng và trừ các số theo biểu diễn bù 1

##### a. Cộng

Hai số dương: cộng như cộng nhị phân thông thường, kể cả bit dấu.

Hai số âm: biểu diễn chúng ở dạng bù 1 và cộng như cộng nhị phân, kể cả bit dấu. Bit tràn cộng vào kết quả. Chú ý, kết quả được viết dưới dạng bù 1.

Hai số khác dấu và số dương lớn hơn: cộng số dương với bù 1 của số âm. Bit tràn được cộng vào kết quả.

Hai số khác dấu và số âm lớn hơn: cộng số dương với bù 1 của số âm. Kết quả không có bit tràn và ở dạng bù 1.

##### b. Trừ

Để thực hiện phép trừ, ta lấy bù 1 của số trừ, sau đó thực hiện các bước như phép cộng.

### 3. Cộng và trừ nhị phân theo biểu diễn bù 2

#### a. Cộng

Hai số dương: cộng như cộng nhị phân thông thường. Kết quả là dương.

Hai số âm: lấy bù 2 cả hai số hạng và cộng, kết quả ở dạng bù 2.

Hai số khác dấu và số dương lớn hơn: lấy số dương cộng với bù 2 của số âm. Kết quả bao gồm cả bit dấu, bit tràn bỏ đi.

Hai số khác dấu và số âm lớn hơn: số dương được cộng với bù 2 của số âm, kết quả ở dạng bù 2 của số dương tương ứng. Bit dấu là 1.

#### b. Phép trừ

Phép trừ hai số có dấu là các trường hợp riêng của phép cộng. Ví dụ, khi lấy +9 trừ đi +6 là tương ứng với +9 cộng với -6.

## 1.4. DẤU PHẪY ĐỘNG

### 1.4.1 Biểu diễn theo dấu phẩy động

Gồm hai phần: số mũ E (phần đặc tính) và phần định trị M (trường phân số). E có thể có độ dài từ 5 đến 20 bit, M từ 8 đến 200 bit phụ thuộc vào từng ứng dụng và độ dài từ máy tính. Thông thường dùng 1 số bit để biểu diễn E và các bit còn lại cho M với điều kiện:

$$1/2 \leq |M| \leq 1$$

E và M có thể được biểu diễn ở dạng bù 2. Giá trị của chúng được hiệu chỉnh để đảm bảo mối quan hệ trên đây được gọi là chuẩn hóa.

### 1.4.2 Các phép tính với biểu diễn dấu phẩy động

Giống như các phép tính của hàm mũ. Giả sử có hai số theo dấu phẩy động đã chuẩn hóa:  $X = 2^{E_x} (M_x)$  và  $Y = 2^{E_y} (M_y)$  thì:

$$\text{Tích: } Z = X \cdot Y = 2^{E_x + E_y} (M_x \cdot M_y) = 2^{E_z} M_z$$

$$\text{Thương: } W = X / Y = 2^{E_x - E_y} (M_x / M_y) = 2^{E_w} M_w$$

Muốn lấy tổng và hiệu, cần đưa các số hạng về cùng số mũ, sau đó số mũ của tổng và hiệu sẽ lấy số mũ chung, còn định trị của tổng và hiệu sẽ bằng tổng và hiệu các định trị.

## TÓM TẮT

Trong chương này chúng ta giới thiệu về một số hệ đếm thường được sử dụng trong hệ thống số: hệ nhị phân, hệ bát phân, hệ thập lục phân. Và phương pháp chuyển đổi giữa các hệ đếm đó.

Ngoài ra còn giới thiệu các phép tính số học trong các hệ đó.

## CÂU HỎI ÔN TẬP

- Định nghĩa thế nào là bit, byte?
- Đổi số nhị phân sau sang dạng bát phân: 0101 1111 0100 1110
  - 57514
  - 57515
  - 57516
  - 57517
- Thực hiện phép tính hai số thập lục phân sau:  $132,44_{16} + 215,02_{16}$ .
  - 347,46
  - 357,46
  - 347,56
  - 357,67
- Thực hiện phép cộng hai số có dấu sau theo phương pháp bù 1:  
 $0000\ 1101_2 + 1000\ 1011_2$ 
  - 0000 0101
  - 0000 0100
  - 0000 0011
  - 0000 0010
- Thực hiện phép cộng hai số có dấu sau theo phương pháp bù 2:  
 $0000\ 1101_2 - 1001\ 1000_2$ 
  - 1000 1110
  - 1000 1011
  - 1000 1100
  - 1000 1110
- Hai byte có bao nhiêu bit?
  - 16
  - 8
  - 32
  - 64

## CHƯƠNG 2: ĐẠI SỐ BOOLE VÀ CÁC PHƯƠNG PHÁP BIỂU DIỄN HÀM

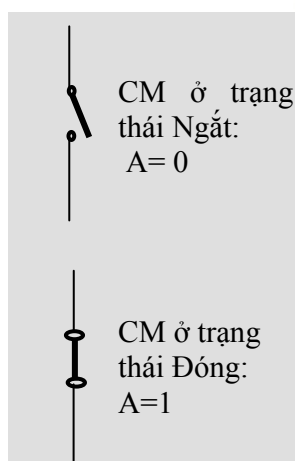
### GIỚI THIỆU CHUNG

Trong mạch số, các tín hiệu thường cho ở hai mức điện áp, ví dụ 0 V và 5 V. Những linh kiện điện tử dùng trong mạch số làm việc ở một trong hai trạng thái, ví dụ transistor lưỡng cực làm việc ở chế độ khóa (tắt), hoặc thông.

Do vậy, để mô tả hoạt động của các mạch số, người ta dùng hệ nhị phân (Binary), hai trạng thái của các linh kiện trong mạch được mã hóa tương ứng thành 1 và 0.

Một bộ môn đại số được phát triển từ cuối thế kỷ 19 mang tên chính người sáng lập ra nó, đại số Boole, còn được gọi là đại số logic rất thích hợp cho việc mô tả mạch số. Đại số Boole là công cụ toán học quan trọng để thiết kế và phân tích mạch số. Các kỹ sư, các nhà chuyên môn trong lĩnh vực điện tử, tin học, thông tin, điều khiển... đều cần phải nắm vững công cụ này để có thể đi sâu vào mọi lĩnh vực liên quan đến kỹ thuật số.

84 năm sau, đại số Boole đã được Shannon phát triển thành lý thuyết *chuyển mạch*. Nhờ các công trình của Shannon, về sau này, các nhà kỹ thuật đã dùng đại số Boole để phân tích và thiết kế các mạch vi tính. Trạng thái "đúng", "sai" trong bài toán logic được thay thế bằng trạng thái "đóng", "ngắt" của một *chuyển mạch* (CM). Mỗi quan hệ nhân quả trong bài toán logic được thay bởi mối quan hệ giữa dòng điện trong mạch với trạng thái các CM gắn trên đoạn mạch ấy. Mối quan hệ này sẽ được thể hiện bằng một hàm toán học, có tên là *hàm chuyển mạch*. Khi đó, các trạng thái của CM : "đóng" = 1 và "ngắt" = 0. Hình 2-1 mô tả điều vừa nói. Ở đây, trạng thái của CM được kí hiệu bằng chữ cái A.



Về thực chất, hàm chuyển mạch là một trường hợp cụ thể của hàm logic. Do đó, đại số Boole ứng với trường hợp này cũng được gọi là đại số chuyển mạch. Mặc dù vậy, trong một số tài liệu người ta vẫn thường gọi nó là đại số logic hay đại số Boole.

Ngày nay, đại số Boole không chỉ giới hạn trong lĩnh vực kỹ thuật chuyển mạch mà còn là công cụ phân tích và thiết kế các mạch số, đặc biệt là lĩnh vực máy tính. Cấu kiện làm chuyển mạch được thay bằng Diode, Transistor, các mạch tích hợp, băng từ... Hoạt động của các cấu kiện này cũng được đặc trưng bằng hai trạng thái: thông hay tắt, dẫn điện hay không dẫn điện... Do đó, hai giá trị hệ nhị phân vẫn được dùng để mô tả trạng thái của chúng.

Đại số logic chỉ có 3 hàm cơ bản nhất, đó là hàm "Và", hàm "Hoặc" và hàm "Đảo". Đặc điểm nổi bật của đại số logic là cả hàm lẫn biến chỉ lấy hai giá trị hoặc 1 hoặc 0.

Trong chương này, ta sẽ đề cập đến các tiên đề, định lý, các cách biểu diễn hàm Boole và một số phương pháp rút gọn hàm. Ngoài ra, chương này cũng xét các loại cổng logic và các tham số chính của chúng.

## NỘI DUNG

### 2.1 ĐẠI SỐ BOOLE

#### 2.1.1. Các định lý cơ bản:

STT	Tên gọi	Dạng tích	Dạng tổng
1	Đồng nhất	$X.1 = X$	$X + 0 = X$
2	Phần tử 0, 1	$X.0 = 0$	$X + 1 = 1$
3	Bù	$X.\bar{X} = 0$	$X + \bar{X} = 1$
4	Bất biến	$X.X = X$	$X + X = X$
5	Hấp thụ	$X + X.Y = X$	$X.(X + Y) = X$
6	Phủ định đúp	$\overline{\bar{X}} = X$	
7	Định lý DeMorgan	$\overline{(X.Y.Z\dots)} = \bar{X} + \bar{Y} + \bar{Z} + \dots$	$\overline{(X + Y + Z + \dots)} = \bar{X}.\bar{Y}.\bar{Z}.\dots$

**Bảng 2.1.** Một số định lý thông dụng trong đại số chuyển mạch

#### 2.1.2 Các định luật cơ bản:

+ Hoán vị:  $X.Y = Y.X$ ,  $X + Y = Y + X$

+ Kết hợp:  $X.(Y.Z) = (X.Y).Z$ ,  $X + (Y + Z) = (X + Y) + Z$

+ Phân phối:  $X.(Y + Z) = X.Y + X.Z$ ,  $(X + Y).(X + Z) = X + Y.Z$

### 2.2 CÁC PHƯƠNG PHÁP BIỂU DIỄN HÀM BOOLE

Như đã nói ở trên, hàm logic được thể hiện bằng những biểu thức đại số như các môn toán học khác. Đây là phương pháp tổng quát nhất để biểu diễn hàm logic. Ngoài ra, một số phương pháp khác cũng được dùng để biểu diễn loại hàm này. Mỗi phương pháp đều có ưu điểm và ứng dụng riêng của nó. Dưới đây là nội dung của một số phương pháp thông dụng.

#### 2.2.1 Bảng trạng thái

Liệt kê giá trị (trạng thái) mỗi biến theo từng cột và giá trị hàm theo một cột riêng (thường là bên phải bảng). Bảng trạng thái còn được gọi là **bảng sự thật** hay **bảng chân lý**.



m	A	B	C	f
m <sub>0</sub>	0	0	0	0
m <sub>1</sub>	0	0	1	0
m <sub>2</sub>	0	1	0	0
m <sub>3</sub>	0	1	1	0
m <sub>4</sub>	1	0	0	0
m <sub>5</sub>	1	0	1	0
m <sub>6</sub>	1	1	0	0
m <sub>7</sub>	1	1	1	1

**Bảng 2.2.** Bảng trạng thái hàm 3 biến

Đối với hàm n biến sẽ có  $2^n$  tổ hợp độc lập. Các tổ hợp này được kí hiệu bằng chữ  $m_i$ , với  $i = 0$  đến  $2^n - 1$  (xem bảng 2-2) và có tên gọi là các **hạng tích** hay còn gọi là **mintex**.

Vì mỗi hạng tích có thể lấy 2 giá trị là 0 hoặc 1, nên nếu có n biến thì số hàm mà bảng trạng thái có thể thiết lập được sẽ là:  $N = 2^{2^n}$

### 2.2.2 Phương pháp bảng Các nô (Karnaugh)

Tổ chức của bảng Các nô: Các tổ hợp biến được viết theo một dòng (thường là phía trên) và một cột (thường là bên trái). Như vậy, một hàm logic có n biến sẽ có  $2^n$  ô. Mỗi ô thể hiện một hạng tích hay một hạng tổng, các hạng tích trong hai ô *kế cận* chỉ khác nhau một biến.

Tính tuần hoàn của bảng Các nô: Không những các ô *kế cận khác nhau một biến* mà các ô *đầu dòng và cuối dòng, đầu cột và cuối cột* cũng chỉ khác nhau một biến (kể cả 4 góc vuông của bảng). Bởi vậy các ô này cũng gọi là *kế cận*.

Muốn thiết lập bảng Các nô của một hàm đã cho dưới dạng chuẩn tổng các tích, ta chỉ việc ghi giá trị 1 vào các ô ứng với hạng tích có mặt trong biểu diễn, các ô còn lại sẽ lấy giá trị 0 (theo định lý DeMorgan). Nếu hàm cho dưới dạng tích các tổng, cách làm cũng tương tự, nhưng các ô ứng với hạng tổng có trong biểu diễn lại lấy giá trị 0 và các ô khác lấy giá trị 1.

### 2.2.3 Phương pháp đại số

Có 2 dạng biểu diễn là dạng *tuyển (tổng các tích)* và dạng *hội (tích các tổng)*.

+ Dạng tuyển: Mỗi số hạng là một *hạng tích* hay *mintex*, thường kí hiệu bằng chữ "**m<sub>i</sub>**".

+ Dạng hội: Mỗi thừa số là *hạng tổng* hay *maxtex*, thường được kí hiệu bằng chữ "**M<sub>i</sub>**". Nếu trong tất cả mỗi hạng tích hay hạng tổng có đủ mặt các biến, thì dạng tổng các tích hay tích các tổng tương ứng được gọi là dạng *chuẩn*. Dạng chuẩn là duy nhất.

Tổng quát, hàm logic n biến có thể biểu diễn chỉ bằng một dạng tổng các tích:

$$f(X_{n-1}, \dots, X_0) = \sum_{i=0}^{2^n-1} a_i m_i$$

hoặc bằng chỉ một dạng tích các tổng:

$$f(X_{n-1}, \dots, X_0) = \prod_{i=0}^{2^n-1} (a_i + m_i)$$

Ở đây,  $a_i$  chỉ lấy hai giá trị 0 hoặc 1. Đối với một hàm thì mintex và maxtex là bù của nhau.

## 2.3 CÁC PHƯƠNG PHÁP RÚT GỌN HÀM

### 2.3.1. Phương pháp đại số

Dựa vào các định lý đã học để đưa biểu thức về dạng tối giản.

Ví dụ: Hãy đưa hàm logic về dạng tối giản:

$$f = AB + \bar{A}C + BC$$

Áp dụng định lý,  $A + \bar{A} = 1$ ,  $X + XY = X$  ta có:

$$\begin{aligned} f &= AB + \bar{A}C + BC(A + \bar{A}) \\ &= AB + ABC + \bar{A}C + \bar{A}BC \\ &= AB + \bar{A}C \end{aligned}$$

Vậy nếu trong tổng các tích, xuất hiện một biến và đảo của biến đó trong hai số hạng khác nhau, các thừa số còn lại trong hai số hạng đó tạo thành thừa số của một số hạng thứ ba thì số hạng thứ ba đó là thừa và có thể bỏ đi.

### 2.3.2 Phương pháp bảng Các nô

Phương pháp này thường được dùng để rút gọn các hàm có số biến không vượt quá 5.

Các bước tối thiểu hóa:

1. Gộp các ô kề cận có giá trị '1' (hoặc '0') lại thành từng nhóm 2, 4, ...,  $2^i$  ô. Số ô trong mỗi nhóm càng lớn kết quả thu được càng tối giản. Một ô có thể được gộp nhiều lần trong các nhóm khác nhau. Nếu gộp theo các ô có giá trị '0' ta sẽ thu được biểu thức bù của hàm.

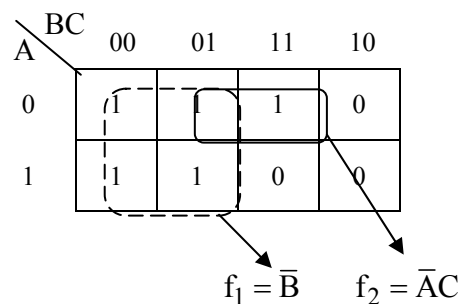
2. Thay mỗi nhóm bằng một hạng tích mới, trong đó giữ lại các biến giống nhau theo dòng và cột.

3. Cộng các hạng tích mới lại, ta có hàm đã tối giản.

Ví dụ: Hãy dùng bảng Các nô để giản ước hàm :

$$f(A, B, C) = \sum(1, 2, 3, 4, 5)$$

Lời giải:



Hình 2-2

+ Xây dựng bảng KN tương ứng với hàm đã cho.

+ Gộp các ô có giá trị 1 kề cận lại với nhau thành hai nhóm (hình 2-2)

Lời giải phải tìm :

$$f = f_1 + f_2 = \bar{B} + \bar{A}C$$

Nếu gộp các ô có giá trị 0 lại theo hai nhóm, ta thu được biểu thức hàm bù  $\bar{f}$  :

$$\bar{f} = AB + BC$$

### 2.3.3. Phương pháp Quine Mc. Cluskey

Phương pháp này có thể tối thiểu hóa được hàm nhiều biến và có thể tiến hành công việc nhờ máy tính.

Các bước tối thiểu hóa:

1. Lập bảng liệt kê các hạng tích dưới dạng nhị phân theo từng nhóm với số bit 1 giống nhau và xếp chúng theo số bit 1 tăng dần.

2. Gộp 2 hạng tích của mỗi cặp nhóm chỉ khác nhau 1 bit để tạo các nhóm mới. Trong mỗi nhóm mới, giữ lại các biến giống nhau, biến bỏ đi thay bằng một dấu ngang (-).

Lặp lại cho đến khi trong các nhóm tạo thành không còn khả năng gộp nữa. Mỗi lần rút gọn, ta đánh dấu # vào các hạng ghép cặp được. Các hạng không đánh dấu trong mỗi lần rút gọn sẽ được tập hợp lại để lựa chọn biểu thức tối giản.

Ví dụ. Hãy tìm biểu thức tối giản cho hàm:

$$f(A, B, C, D) = \sum(10, 11, 12, 13, 14, 15)$$

Giải: Bước 1: Lập bảng (bảng 2.3a):

Bảng a		Bảng b	
Hạng tích đã sắp xếp	Nhị phân A B C D	Rút gọn lần đầu. A B C D	Rút gọn lần thứ 2. A B C D
10	1 0 1 0	1 0 1 - # (10,11)	1 1 - - (12,13,14,15)
12	1 1 0 0	1 - 1 0 # (10,14)	1 - 1 - (10,11,14,15)
11	1 0 1 1	1 1 0 - # (12,13)	
13	1 1 0 1	1 1 - 0 # (12,14)	
14	1 1 1 0	1 - 1 1 # (11,15)	
15	1 1 1 1	1 1 - 1 # (13,15)	
		1 1 1 - # (14,15)	

**Bảng 2.3**

Bước 2: Thực hiện nhóm các hạng tích (bảng 2.3b).

Tiếp tục lập bảng lựa chọn để tìm hàm tối giản (Bảng 2.4):

<i>A BCD</i>	10	11	12	13	14	15
1 1 - -			x	x	x	x
1 - 1 -	x	x			x	x

**Bảng 2.4**

Từ bảng 2-4, ta nhận thấy rằng 4 cột có duy nhất một dấu "x" ứng với hai hàng 11-- và 1-1-. Do đó, biểu thức tối giản là :

$$f(A, B, C, D) = AB + AC$$

## 2.4 CÔNG LOGIC VÀ CÁC THAM SỐ CHÍNH

Công logic cơ sở là mạch điện thực hiện ba phép tính cơ bản trong đại số logic, vậy ta sẽ có ba loại công logic cơ sở là AND, OR và NOT.

### 2.4.1 Công logic cơ bản

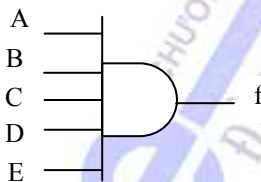
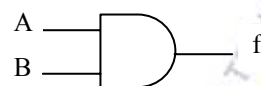
#### 2.4.1.1 Công AND

Công AND thực hiện hàm logic

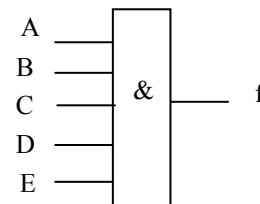
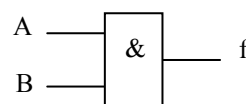
$$f = f(A, B) = A.B$$

hoặc nhiều biến:

$$f(A, B, C, D, \dots) = A.B.C.D\dots$$



a) Theo tiêu chuẩn ANSI



b) Theo tiêu chuẩn IEEE

Hình 2-4a,b. Ký hiệu của công AND.

Nguyên lý hoạt động của công AND:

Bảng trạng thái 2.5a,b là nguyên lý hoạt động của công AND (2 lối vào).

A	B	f
0	0	0
0	1	0
1	0	0
1	1	1

A	B	f
L	L	L
L	H	L
H	L	L
H	H	H

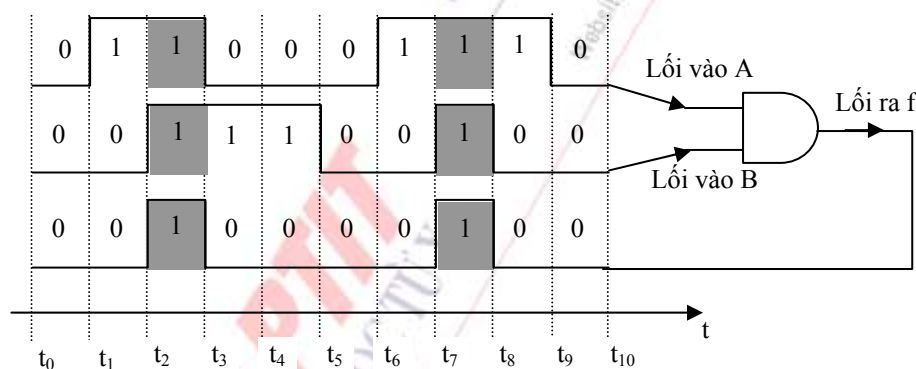
a) Ghi theo giá trị logic

b) Ghi theo mức logic

**Bảng 2.5a,b.** Bảng trạng thái mô tả hoạt động của cổng AND 2 lối vào.

Theo qui ước, logic 1 được thay bằng mức điện thế cao, viết tắt là H (High) còn logic 0 được thay bằng mức điện thế thấp, viết tắt là L (Low) (bảng 2-5b). Cổng AND có n lối vào sẽ có  $2^n$  hạng tích (dòng) trong bảng trạng thái.

Khi tác động tới lối vào các chuỗi xung số xác định, đầu ra cũng sẽ xuất hiện một chuỗi xung như chỉ hình 2-4. Đồ thị này thường được gọi là đồ thị dạng xung, đồ thị dạng sóng hay đồ thị thời gian.

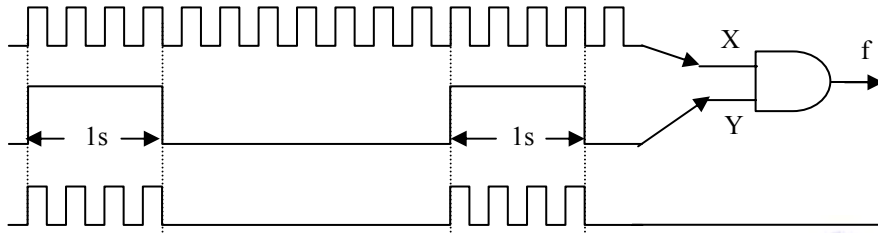


**Hình 2-4.** Đồ thị dạng xung vào, ra của cổng AND

Từ đồ thị, ta nhận thấy rằng, chỉ tại các thời điểm  $t_2$  đến  $t_3$  và  $t_7$  đến  $t_8$  trên cả hai lối vào đều có logic 1 nên lối ra cũng lấy logic 1. Ứng với các khoảng thời gian còn lại vì hoặc cả hai lối vào bằng 0, hoặc một trong hai lối vào bằng 0 nên lối ra lấy logic 0. Hoạt động của cổng AND nhiều lối vào cũng xảy ra tương tự.

Có thể giải thích dễ dàng một vài ứng dụng của cổng AND qua đồ thị dạng xung.

**Ví dụ :** Dùng cổng AND để tạo "cửa" thời gian. Trong ứng dụng này, trên hai lối vào của cổng AND được đưa tới 2 chuỗi tín hiệu số X, Y có tần số khác nhau. Giả sử tần số của X lớn hơn tần số của Y. Trên đầu ra cổng AND chỉ tồn tại tín hiệu X, gián đoạn theo từng chu kỳ của Y. Như vậy, chuỗi số Y chỉ giữ vai trò đóng, ngắt cổng AND và thường được gọi là tín hiệu "cửa". Hoạt động của mạch được mô tả bằng hình 2-5.



Hình 2-5. Mô hình dùng cổng AND để tạo “cửa” thời gian

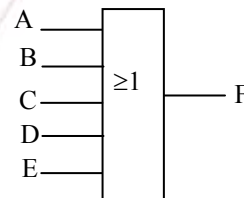
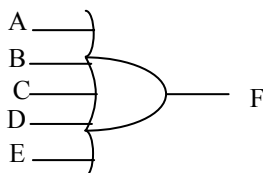
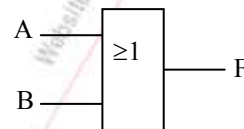
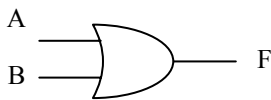
Tùy theo điều kiện cho trước, có thể ứng dụng mạch theo các mục đích khác nhau. Nếu đã biết độ rộng xung “cửa” Y ( thường lấy bằng 1s ) thì số xung xuất hiện đầu ra chính bằng tần số của X. Ngược lại, nếu tần số của X đã cho, chẳng hạn bằng 1 Hz (  $T_x = 1s$  ) thì chỉ cần đếm số xung trên đầu ra ta có thể tính được độ rộng xung “cửa” Y. Đây chính là phương pháp đo tần số và thời gian được ứng dụng trong kỹ thuật hiện nay.

### 2.4.1.2 Cổng OR

Cổng OR thực hiện hàm logic:  $f(A, B) = A + B$

hoặc với hàm nhiều biến:  $f(A, B, C, D, \dots) = A + B + C + D + \dots$

Ký hiệu của cổng OR được biểu diễn ở hình 2-6a, b.



a) Theo tiêu chuẩn ANSI

b) Theo tiêu chuẩn IEEE

Hình 2-6 a, b. Ký hiệu của cổng OR.

Tương tự như cổng AND, nguyên lý hoạt động của cổng OR có thể được giải thích thông qua bảng trạng thái (Bảng 2.6a,b) và đồ thị dạng xung - hình 2-7.

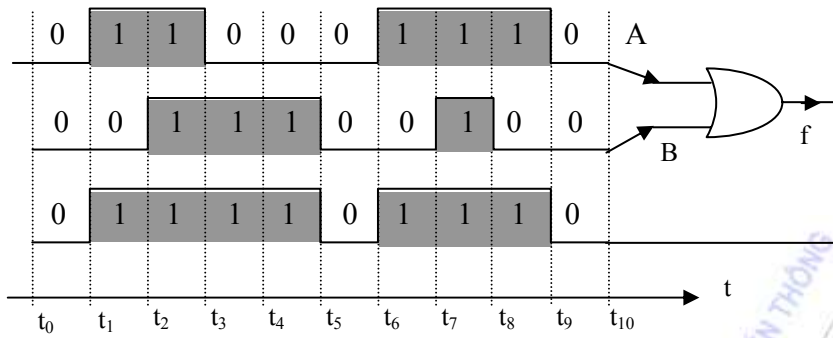
A	B	f
0	0	0
0	1	1
1	0	1
1	1	1

a) Theo giá trị logic

A	B	f
L	L	L
L	H	H
H	L	H
H	H	H

b) Theo mức điện thế

Bảng 2.6 a, b. Bảng trạng thái của cổng OR.



Hình 2-7. Đồ thị dạng xung của cổng OR.

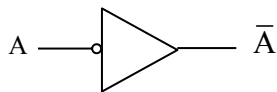
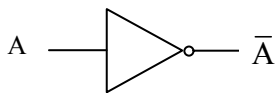
Một cổng OR có n lối vào sẽ có  $2^n$  hạng tích trong bảng trạng thái của nó.

### 2.4.1.3. Cổng NOT

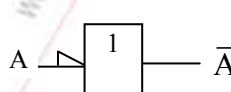
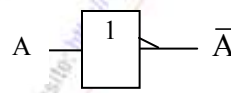
Cổng NOT thực hiện hàm logic:

$$f = \bar{A}$$

Ký hiệu của cổng NOT được chỉ ra trên hình 2-8 a, b.



a) Theo tiêu chuẩn ANSI.



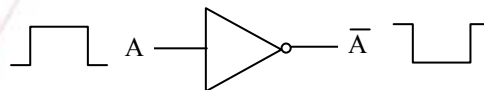
b) Theo tiêu chuẩn IEEE.

Hình 2-8a,b. Ký hiệu của cổng NOT

Hoạt động của cổng NOT khá đơn giản, nếu lối vào:

$A = 0$  thì  $\bar{A} = 1$ ,

nếu  $A = 1$  thì  $\bar{A} = 0$



Hình 2-9

Nguyên lý này được minh họa bằng đồ thị dạng xung ở hình 2-9.

Hoạt động của cổng NOT được tóm tắt ở bảng 2.7a,b.

A	f
0	1
1	0

a) Theo giá trị logic

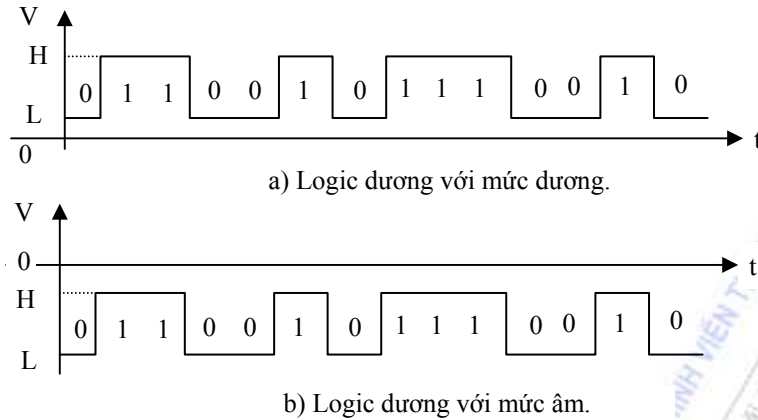
A	f
L	H
H	L

b) Theo mức logic

Bảng 2.7a, b. Bảng trạng thái của cổng NOT.

### 2.4.2 Logic dương và logic âm

Logic dương là logic có điện thế mức H luôn lớn hơn điện thế mức L (Hình 2-10).



**Hình 2-10a,b.** Đồ thị dạng xung của logic dương

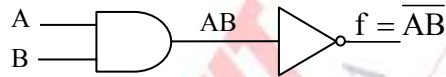
Logic âm thì ngược lại, logic 1 có điện thế thấp hơn mức 0. Khái niệm logic âm thường được dùng để biểu diễn trị các biến. Logic âm và mức âm của logic là hoàn toàn khác nhau.

### 2.4.3 Một số cổng ghép thông dụng

Khi ghép ba loại cổng logic cơ bản nhất sẽ thu được các mạch logic từ đơn giản đến phức tạp. Ở đây ta chỉ xét một vài mạch ghép đơn giản nhưng rất thông dụng.

#### 2.4.3.1 Cổng NAND

Ghép nối tiếp một cổng AND với một cổng NOT ta được cổng NAND (Hình 2-11).



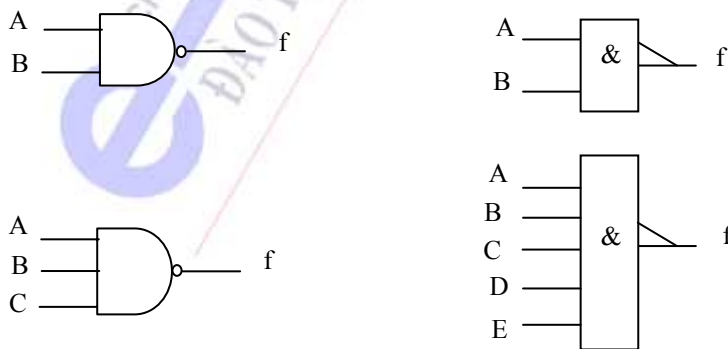
**Hình 2-11.** Sơ đồ cấu tạo cổng NAND

Hàm ra của cổng NAND 2 và nhiều biến vào như sau:

$$f = \overline{AB}$$

$$f = \overline{ABCD\dots}$$

Ký hiệu cổng NAND (hình 2-12a,b) và bảng trạng thái (bảng 2-8).



a) Theo tiêu chuẩn ANSI

b) Theo tiêu chuẩn IEEE

**Hình 2-12a,b.** Ký hiệu của cổng NAND



A	B	f
0	0	1
0	1	1
1	0	1
1	1	0

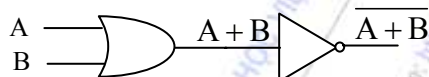
A	B	f
L	L	H
L	H	H
H	L	H
H	H	L

**Bảng 2.8a,b.** Bảng trạng thái của cổng NAND

### 2.4.3.2 Cổng NOR

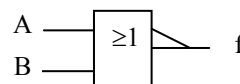
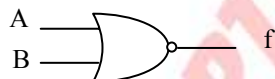
Cổng NOR được thiết lập bằng cách nối tiếp một cổng OR với một cổng NOT. Từ hình 2-13 ta có thể viết được hàm ra của cổng NOR 2 và nhiều lối vào như sau:

$$f = \overline{A + B} \quad \text{hay} \quad f = \overline{A+B+C+\dots}$$



**Hình 2-13.** Sơ đồ cấu tạo cổng NOR

Ký hiệu của cổng NOR 2 lối vào như chỉ ở hình 2-14a,b.



a) Theo tiêu chuẩn ANSI.

b) Theo tiêu chuẩn IEEE.

**Hình 2-14a, b.** Ký hiệu cổng NOR 2 lối vào

Hoạt động của cổng NOR được giải thích bằng bảng trạng thái như chỉ ở bảng 2.9a,b.

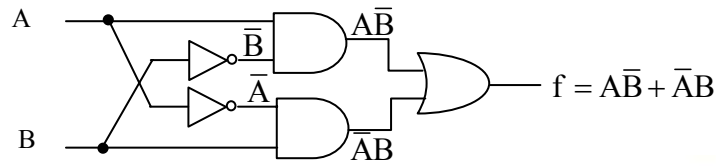
A	B	f
0	0	1
0	1	0
1	0	0
1	1	0

A	B	f
L	L	H
L	H	L
H	L	L
H	H	L

**Bảng 2.9a, b.** Bảng trạng thái của cổng NOR 2 lối vào.

### 2.4.3.3 Cổng khác dấu

Cổng khác dấu còn có một số tên gọi khác: cổng Cộng Modul-2, cổng XOR.

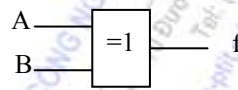
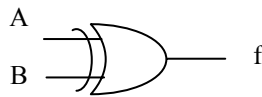


Hình 2-15. Sơ đồ của cổng XOR 2 lối vào

Từ hình 2-15, ta có biểu thức của hàm khác dấu 2 lối vào là:

$$f = A\bar{B} + \bar{A}B \quad \text{hay theo qui ước} \quad f = A \oplus B$$

Ký hiệu của cổng XOR 2 lối vào như hình 2-16a, b.



a) Theo tiêu chuẩn ANSI

b) Theo tiêu chuẩn IEEE

Hình 2-16a, b. Ký hiệu của cổng XOR 2 lối vào

Bảng trạng thái của cổng XOR hai lối vào được trình bày ở bảng 2.10a,b.

A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

A	B	F
L	L	L
L	H	H
H	L	H
H	H	L

Bảng 2-10a,b. Bảng trạng thái của cổng XOR 2 lối vào

Hoạt động cổng XOR nhiều lối vào cũng tương tự như cổng 2 lối vào, nghĩa là nếu số bit 1 trên tất cả các lối vào là một số lẻ, thì hàm ra lấy logic 1; ngược lại nếu tổng số bit 1 trên các lối vào là một số chẵn, thì hàm ra lấy logic 0. Có thể dùng cổng XOR 2 lối vào để thực hiện hàm XOR nhiều biến.

#### 2.4.3.4 Cổng đồng dấu (XNOR)

Cổng XNOR thực hiện biểu thức logic sau:

$$f = AB + \bar{A}\bar{B} \quad \text{hay} \quad f = \overline{A \oplus B} = A \sim B$$

Ký hiệu của cổng XNOR hai lối vào được trình bày ở hình 2-17.



a) Theo tiêu chuẩn ANSI

b) Theo tiêu chuẩn IEEE

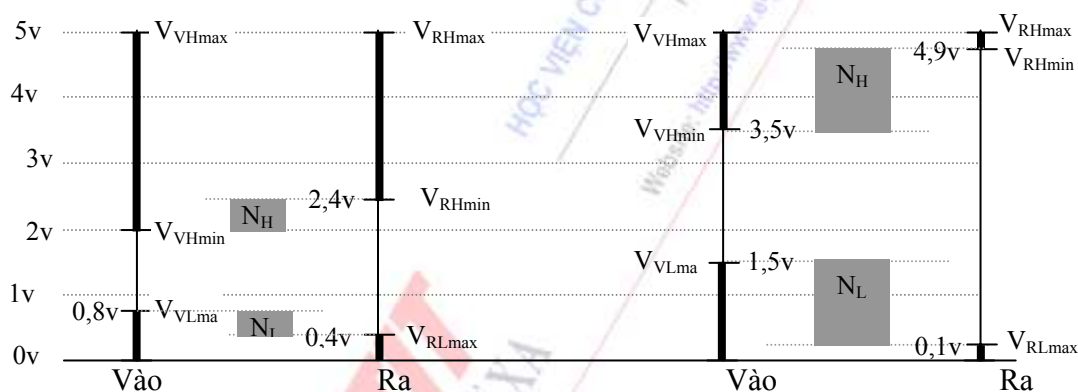
**Hình 2-17.** Ký hiệu của cổng XNOR 2 lối vào

Nếu tổng số bit 0 trên tất cả các lối vào là một số lẻ, thì hàm ra của XNOR sẽ lấy logic 1. Nếu tổng số bit 0 trên tất cả các lối vào là một số chẵn, thì hàm ra lại lấy logic 0.

XOR và XNOR là hai loại cổng có rất nhiều ứng dụng trong kỹ thuật số. Chúng là phần tử chính hợp thành bộ cộng, trừ, so sánh hai số nhị phân v.v...

## 2.4.4 Các tham số chính

### 2.4.4.1 Mức logic



a) Đối với họ TTL

b) Đối với họ CMOS

**Hình 2-19a, b.** Mức logic của các họ cổng TTL và CMOS

Mức logic là mức điện thế trên đầu vào và đầu ra của cổng tương ứng với logic "1" và logic "0", nó phụ thuộc điện thế nguồn nuôi của cổng ( $V_{CC}$  đối với họ TTL (Transistor Transistor Logic) và  $V_{DD}$  đối với họ MOS (Metal Oxide Semiconductor)). Lưu ý rằng, nếu mức logic vào vượt quá điện thế nguồn nuôi có thể gây hư hỏng cho cổng.

### Mức TTL

Mức TTL là một chuẩn quốc tế, trong đó qui định:

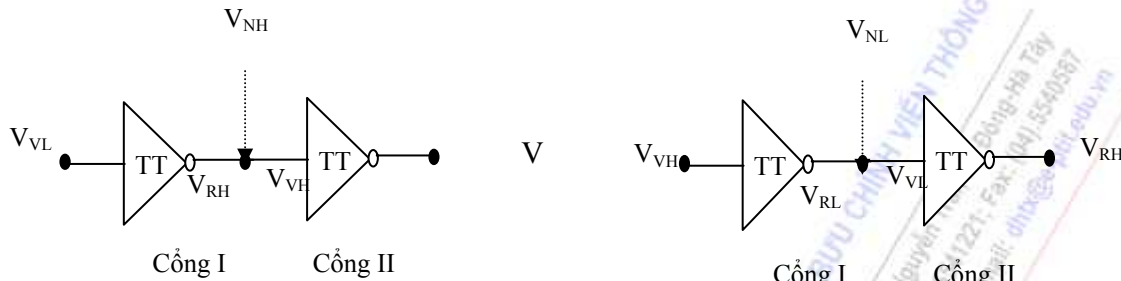
- Điện thế nguồn nuôi  $V_{CC}$ ,  $V_{DD}$  bằng + 5 vôn hoặc bằng - 5,2 vôn;
- Mức điện thế tương ứng với logic H và L trên đầu vào, đầu ra của cổng như chỉ ở hình 2-18a,b.

Nhận xét: + Mức vào ra đối với cổng TTL và CMOS (Complementary Metal Oxide Semiconductor) khác nhau rất nhiều;

+ Mức vào ra sẽ ảnh hưởng đến độ phòng vệ nhiễu của cổng.

### 2.4.4.2 Độ chống nhiễu

Độ chống nhiễu (hay độ phòng vệ nhiễu) là mức nhiễu lớn nhất tác động tới lối vào hoặc lối ra của cổng mà chưa làm thay đổi trạng thái vốn có của nó.



a) Tác động nhiễu khi mức ra cao

b) Tác động nhiễu khi mức ra thấp

**Hình 2-20a, b**, Mô tả tác động nhiễu đến các cổng logic

Ảnh hưởng của nhiễu có thể phân ra hai trường hợp :

+ Nhiễu mức cao: đầu ra cổng I lấy logic H (hình 2-20a), tất nhiên, đầu ra cổng II là logic L, nếu các cổng vẫn hoạt động bình thường. Khi tính tới tác động của nhiễu, ta có:

$$V_{RH\min} + V_{NH} \geq V_{VH\min} \Leftrightarrow V_{NH} \geq V_{VH\min} - V_{RH\min}$$

Với cổng TTL:  $V_{NL} \geq 2V - 2,4V = -0,4V$

Với cổng CMOS:  $V_{NL} \geq 3,5V - 4,9V = -1,4V$

+ Nhiễu mức thấp: đầu ra cổng I lấy logic L (hình 2-20b), tương tự ta có:

$$V_{RL\max} + V_{NL} \leq V_{VL\max} \Leftrightarrow V_{NL} \leq V_{VL\max} - V_{RL\max}$$

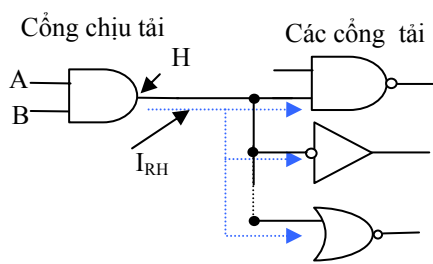
Với cổng TTL:  $V_{NL} \leq 0,8V - 0,4V = 0,4V$

Với cổng CMOS:  $V_{NL} \leq 1,5V - 0,1V = 1,4V$

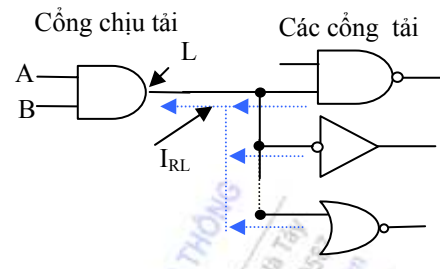
### 2.4.4.3 Hệ số ghép tải K

Cho biết khả năng nối được bao nhiêu lối vào tới đầu ra của một cổng đã cho.

Hệ số ghép tải phụ thuộc dòng ra (hay dòng phun) của cổng chịu tải và dòng vào (hay dòng hút) của các cổng tải ở cả hai trạng thái H, L.



a) Mức ra của cổng chịu tải là H



b) Mức ra của cổng chịu tải là L

Hình 2-21a,b. Mô tả về hệ số ghép tải.

#### 2.4.4.4. Công suất tiêu thụ



Hình 2-22. Hai trạng thái tiêu thụ dòng của cổng logic

$I_{CCH}$  - Là dòng tiêu thụ khi đầu ra lấy mức H,

$I_{CCL}$  - Là dòng tiêu thụ khi đầu ra lấy mức L.

Theo thống kê, tín hiệu số có tỷ lệ bit H / bit L khoảng 50%. Do đó, dòng tiêu thụ trung bình  $I_{CC}$  được tính theo công thức :

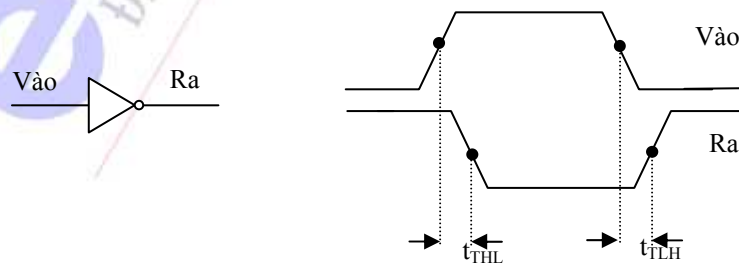
$$I_{CC} = (I_{CCH} + I_{CCL}) / 2$$

Công suất tiêu thụ trung bình của mỗi cổng sẽ là :

$$P_0 = I_{CC} \cdot V_{CC}$$

#### 2.4.4.5. Trễ truyền lan

Tín hiệu đi qua một cổng phải mất một khoảng thời gian, được gọi là trễ truyền lan.



Hình 2-23. Minh họa trễ truyền lan của tín hiệu

Trễ truyền lan xảy ra tại cả hai sườn của xung ra. Nếu kí hiệu trễ truyền lan ứng với sườn trước là  $t_{THL}$  và sườn sau là  $t_{TLH}$  thì trễ truyền lan trung bình là:

$$t_{Tb} = (t_{THL} + t_{TLH})/2$$

Thời gian trễ truyền lan hạn chế tần số công tác của cổng. Trễ càng lớn thì tần số công tác cực đại càng thấp.

## TÓM TẮT

Trong chương 2 chúng ta giới thiệu về các phương pháp biểu diễn và rút gọn hàm Boole. Ngoài ra còn giới thiệu một số cổng logic thông dụng và các tham số chính của chúng.

## CÂU HỎI ÔN TẬP

Bài 2.1 Rút gọn hàm sau theo phương pháp dùng bảng Karnaugh:

1.  $F(A, B, C) = \Sigma(0, 2, 4, 6, 7)$ .

- $AB + \bar{C}$
- $\bar{A}\bar{B} + C$
- $AB + C$
- $\bar{A}\bar{B} + C$

2.  $F(A, B, C, D) = \Sigma(0, 1, 8, 9, 10)$

- $BC + D$
- $\bar{B}\bar{C} + A\bar{B}\bar{D}$
- $\bar{B}\bar{C} + ABD$
- $\bar{B}\bar{C} + \bar{A}\bar{B}D$

2.2 Rút gọn hàm sau theo phương pháp đại số

1.  $\overline{CD + \bar{C}\bar{D}} \cdot \overline{\bar{A}C + D}$

- $CD$
- $C\bar{D}$
- $\bar{C}D$
- $\bar{C}\bar{D}$

2.  $\overline{\bar{A}\bar{B}\bar{C}} \cdot \overline{\bar{A}\bar{B} + \bar{B}\bar{C} + \bar{C}\bar{A}}$

- $AB + AC$
- $AB + AC + BC$
- $AC + BC$
- $AB + BC$

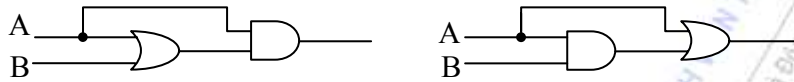
2.3 Rút gọn hàm sau theo phương pháp Quine-Mc.CLUSKEY:

$$F(A, B, C, D) = \Sigma(2, 3, 6, 7, 12, 13, 14, 15).$$

- $AC + AB$

- b.  $\bar{A}C + AD$
- c.  $AC + \bar{A}B$
- d.  $\bar{A}C + AB$

2.4 Hai mạch điện ở hình dưới đây là tương đương



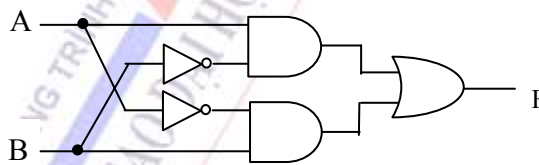
- a. Do đều bằng  $A+B$
- b. Do đều bằng  $B$
- c. Do đều bằng  $AB$
- d. Do đều bằng  $A+AB$

Bài 2.5 Phân tích ý nghĩa các tham số chính của các họ cổng logic.

Bài 2.6 Trình bày về độ phòng vệ nhiễu của các họ cổng logic? Tính độ phòng vệ nhiễu của một cổng logic họ TTL, biết  $V_{VL} = 0 \text{ V} \div 0,8 \text{ V}$ ,  $V_{VH} = 2,0 \text{ V} \div 5,0 \text{ V}$ ,  $V_{RL} = 0 \text{ V} \div 0,4 \text{ V}$ ,  $V_{RH} = 2,4 \text{ V} \div 5,0 \text{ V}$ ?

- a.  $V_{NH} = 0.4\text{V}$ ,  $V_{NL} = -0.4$
- b.  $V_{NH} = -0.4\text{V}$ ,  $V_{NL} = -0.4$
- c.  $V_{NH} = 0.4\text{V}$ ,  $V_{NL} = 0.4$
- d.  $V_{NH} = -0.4\text{V}$ ,  $V_{NL} = 0.4$

Bài 2.7 Cho mạch điện như hình 1. Biểu thức hàm ra là:



Hình 1

- a.  $AB + \bar{A}\bar{B}$
- b.  $\bar{A}B + \bar{A}\bar{B}$
- c.  $\bar{A}B + A\bar{B}$
- d.  $A\bar{B} + \bar{A}\bar{B}$

Bài 2.8 Phân tích ý nghĩa của việc tối ưu hoá mạch điện của các họ cổng logic? Cho ví dụ minh họa?

Bài 2.9 Chứng minh các đẳng thức:

- a.  $\overline{A \oplus B} = \bar{A} \bar{B} + AB$

- b.  $AB(A \oplus B \oplus C) = ABC$
- c.  $A \oplus B \oplus C = \bar{A} \oplus \bar{B} \oplus \bar{C}$

Bài 2.10 Liệt kê 3 phần tử logic cơ bản trong kỹ thuật số?

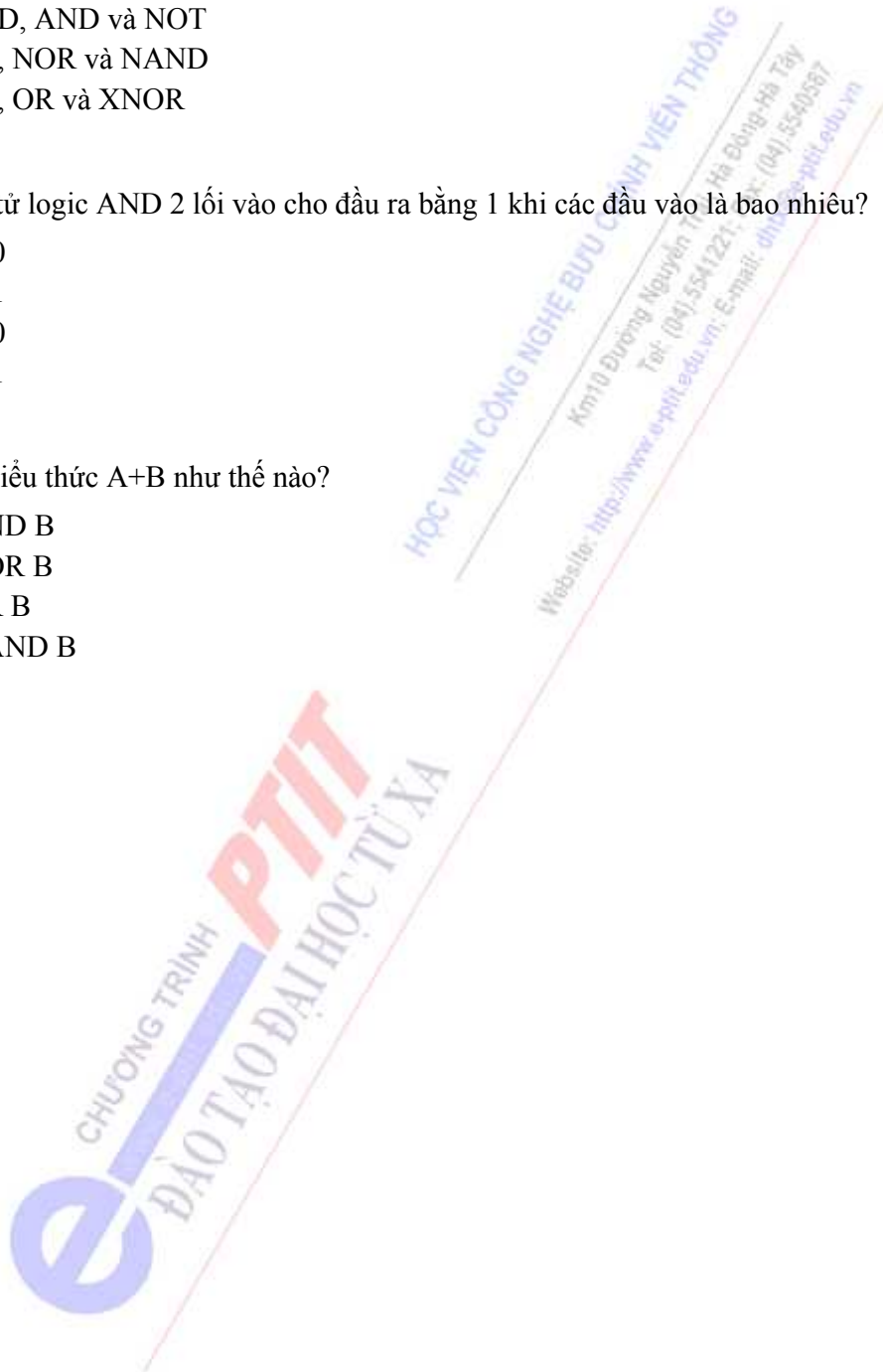
- a. AND, OR và NOT
- b. NAND, AND và NOT
- c. AND, NOR và NAND
- d. AND, OR và XNOR

Bài 2.11 Phần tử logic AND 2 lối vào cho đầu ra bằng 1 khi các đầu vào là bao nhiêu?

- a. 0 và 0
- b. 0 và 1
- c. 1 và 0
- d. 1 và 1

Bài 2.12 Đọc biểu thức  $A+B$  như thế nào?

- a. A AND B
- b. A XOR B
- c. A OR B
- d. A NAND B





## CHƯƠNG 3: CỔNG LOGIC TTL VÀ CMOS

### GIỚI THIỆU

Xét về mặt cơ bản thì có hai loại linh kiện bán dẫn đó là lưỡng cực và đơn cực. Dựa trên các linh kiện này, các mạch tích hợp được hình thành và có sẵn trên thị trường. Các chức năng kỹ thuật số khác nhau cũng được chế tạo trong nhiều dạng khác nhau bằng cách sử dụng công nghệ lưỡng cực và đơn cực. Một nhóm các IC tương thích với các mức logic giống nhau và các điện áp nguồn để thực hiện các chức năng logic đa dạng phải được chế tạo bằng cách sử dụng cấu hình mạch chuyên biệt được gọi là họ mạch logic.

Các yếu tố chính của một IC lưỡng cực là điện trở, điốt và các transistor. Có hai loại hoạt động cơ bản trong các mạch IC lưỡng cực:

- Bảo hoà.
- Không bảo hoà.

Trong mạch logic bảo hoà, các transistor được vận hành trong vùng bảo hoà, còn trong các mạch logic không bảo hoà thì các transistor không làm việc tại vùng bảo hoà.

Các họ mạch logic lưỡng cực được bảo hoà là:

- Mạch logic Điện trở - Transistor (RTL).
- Mạch logic Điốt – Transistor (DTL).
- Mạch logic Transistor – Transistor (TTL).

Các họ mạch logic lưỡng cực không bảo hoà là:

- Schottky TTL.
- Mạch logic ghép cực phát (ECL).

Các linh kiện MOS là các linh kiện đơn cực và chỉ có các MOSFET được vận hành trong các mạch logic MOS. Các họ mạch logic MOS là:

- PMOS.
- NMOS.
- CMOS

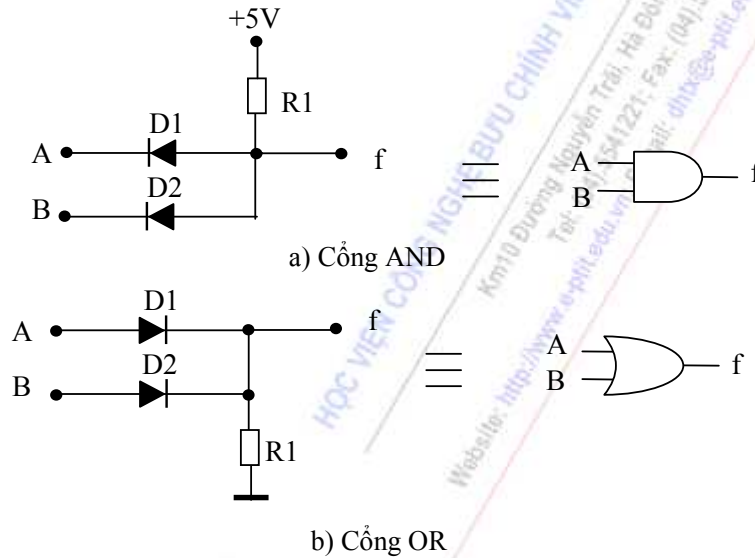
Trong chương 3 sẽ trình bày các họ cổng logic chủ yếu và được dùng phổ biến hiện nay. Phần cuối của chương trình bày một số mạch cho phép giao tiếp giữa các họ logic TTL và CMOS.

## NỘI DUNG

### 3.1. CÁC HỘ CÔNG LOGIC

#### 3.1.1. Hộ DDL

DDL (Diode Diode Logic) là hộ công logic do các diode bán dẫn tạo thành. Hình 3-1a,b là sơ đồ công AND, OR 2 lối vào hộ DDL.



**Hình 3-1.** Mạch điện công AND và OR hộ DDL.

Bảng trạng thái sau thể hiện nguyên lý hoạt động của mạch thông qua mức điện áp vào/ra của các công AND và OR hộ DDL

AND			OR		
A (V)	B (V)	F (V)	A (V)	B (V)	F (V)
0	0	0,7	0	0	0
0	3	0,7	0	5	4,3
3	0	0,7	5	0	4,3
3	3	4,7	5	5	4,3

Bảng 3-1. Bảng trạng thái của công AND và OR hộ DDL

*Ưu điểm* của hộ DDL:

- Mạch điện đơn giản, dễ tạo ra các công AND, OR nhiều lối vào. Ưu điểm này cho phép xây dựng các ma trận diode với nhiều ứng dụng khác nhau;
- Tần số công tác có thể đạt cao bằng cách chọn các diode chuyển mạch nhanh;
- Công suất tiêu thụ nhỏ.

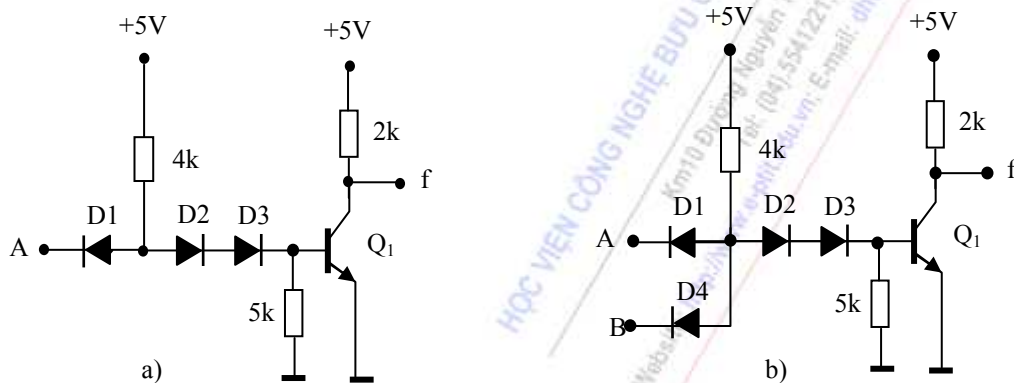
*Nhược điểm* :

- Độ phòng vệ nhiều thấp ( $V_{RL}$  lớn) ;
- Hệ số ghép tải nhỏ.

Để cải thiện độ phòng vệ nhiều ta có thể ghép nối tiếp ở mạch ra một diode. Tuy nhiên, khi đó  $V_{RH}$  cũng bị sụt đi 0,6V.

### 3.1.2. Họ DTL

Để thực hiện chức năng đảo, ta có thể đấu nối tiếp với các cổng DDL một transistor công tác ở chế độ khoá. Mạch cổng như thế được gọi là họ DTL (Diode Transistor Logic). Ví dụ, hình 3-2a, b là các cổng NOT, NAND thuộc họ này.



Hình 3-2. Sơ đồ mạch điện của họ cổng DTL.

Trong hai trường hợp trên, nhờ các diode D2, D3 độ chống nhiễu trên lối vào của  $Q_1$  được cải thiện. Mức logic thấp tại lối ra f giảm xuống khoảng 0,2 V (bằng thế bão hoà  $U_{CE}$  của  $Q_1$ ). Do  $I_{RHmax}$  và  $I_{RLmax}$  của bán dẫn có thể lớn hơn nhiều so với diode nên hệ số ghép tải của cổng cũng tăng lên.

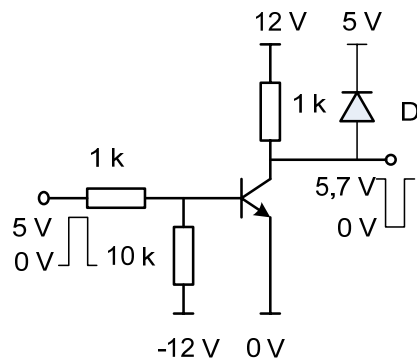
Bằng cách tương tự, ta có thể thiết lập cổng NOR hoặc các cổng liên hợp phức tạp hơn. Vì tải của các cổng là điện trở nên hệ số ghép tải (đặc biệt đối với  $N_H$ ) còn bị hạn chế, mặt khác trễ truyền lan của họ cổng này còn lớn. Những tồn tại trên sẽ được khắc phục từng phần ở các họ cổng sau.

### 3.1.3. Họ RTL

Họ RTL (Resistor Transistor Logic) là các cổng logic được cấu tạo bởi các điện trở và transistor. Hình 3-3 là sơ đồ của một mạch NOT họ RTL.

Khi điện áp lối vào là 0 V, điện áp trên base của transistor sẽ âm nên transistor cấm như vậy lối ra trên collector của transistor sẽ ở mức cao. Do lối ra này được nối lên nguồn +5 V thông qua diode D nên giá trị điện áp lối ra lúc này khoảng 5,7 V, nhận mức logic cao. Khi điện áp lối vào là 5 V do hai điện trở lối vào có giá trị lần lượt là 1 k và 10 k, nên điện áp tại base sẽ đủ lớn để làm transistor thông làm cho điện áp lối ra là 0 V. Như vậy logic lối ra sẽ là đảo của logic của tín hiệu lối vào.

Tương tự như mạch hình 3-3, nếu một điện trở được nối thêm ở lối vào như hình 3-4 sau mạch sẽ trở thành mạch NOR họ RTL.

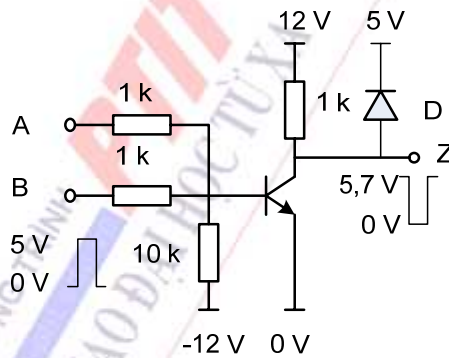


**Hình 3-3.** Cổng NOT họ RTL

Bảng 3-2 thể hiện quan hệ điện áp của cổng NOR họ RTL, chỉ khi cả hai lối vào A và B cùng ở giá trị 0 V thì transistor mới cấm và lối ra nhận logic cao. Các trường hợp khác đều dẫn đến transistor thông và làm giá trị logic lối ra ở mức thấp.

A (V)	B (V)	F (V)
0	0	5,7
0	5	0
5	0	0
5	5	0

Bảng 3-2. Bảng trạng thái của cổng NOR họ RTL

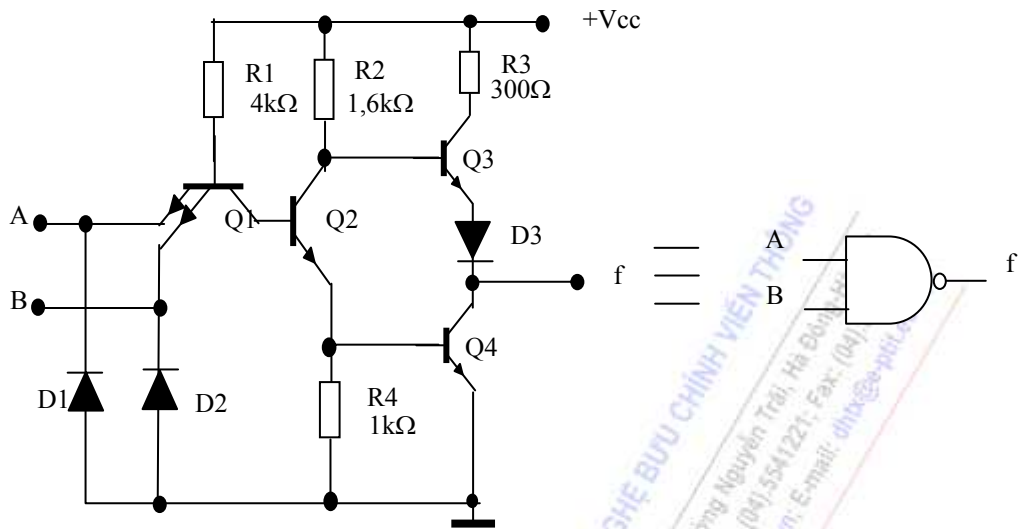


**Hình 3.4.** Cổng NOR họ RTL

### 3.1.4. Họ TTL

Do hạn chế về tốc độ, họ DTL đã trở nên lạc hậu và bị thay thế hoàn toàn bởi họ mạch TTL. Hạn chế tốc độ của DTL được giải quyết bằng cách thay các điốt đầu vào thành transistor đa lớp tiếp giáp BE.

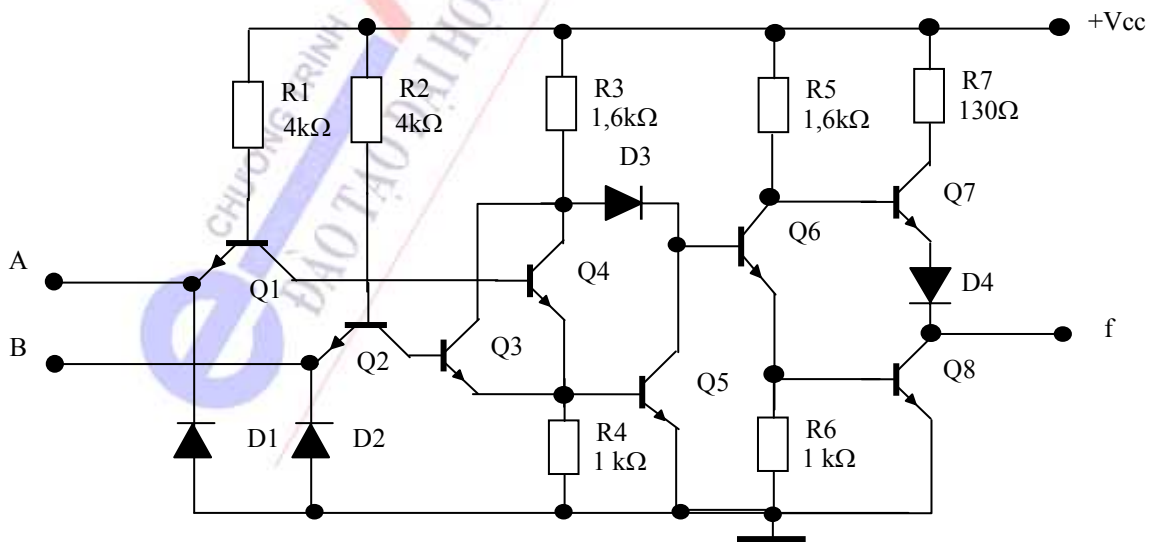
#### a. Cổng NAND TTL



**Hình 3-5.** Sơ đồ mạch điện một cổng NAND 2 lối vào.

Hình 3-5 là sơ đồ nguyên lý của mạch NAND TTL. Nó có thể được chia ra thành 3 phần. Transistor  $Q_1$ , trở  $R_1$  và các diode  $D_1, D_2$  tạo thành mạch đầu vào, mạch này thực hiện chức năng NAND. Transistor  $Q_2$ , các trở  $R_2, R_4$  tạo thành mạch giữa  $Q_3, Q_4, R_3$  và diode  $D_3$  tạo thành mạch lối ra như phân tích ở trên.

Khi bất kỳ một lối vào ở mức thấp thì  $Q_1$  đều trở thành thông bão hoà, do đó,  $Q_2$  và  $Q_4$  đóng, còn  $Q_3$  thông nên đầu ra của mạch sẽ ở mức cao. Lối ra sẽ chỉ xuống mức thấp khi tất cả các lối vào đều ở mức logic cao và làm transistor  $Q_1$  cấm. Diode  $D_3$  được sử dụng như mạch dịch mức điện áp, nó có tác dụng làm cho  $Q_3$  cấm hoàn toàn khi  $Q_2$  và  $Q_4$  thông. Diode này nhiều khi còn được mắc vào mạch giữa collector  $Q_2$  và base của  $Q_3$ .



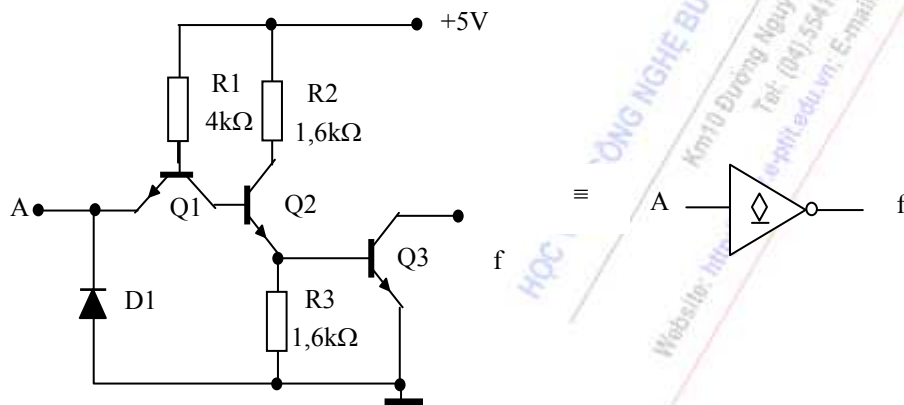
**Hình 3-6.** Sơ đồ mạch điện của một cổng OR 2 lối vào.

**b. Cổng OR TTL**

Hình 3-6 là sơ đồ của một cổng OR họ TTL tiêu chuẩn hai lối vào. Trong trường hợp này, mạch vào sử dụng các bán dẫn đơn. Tuy nhiên, nguyên lý hoạt động của mạch vào này cũng giống với cổng NAND hình 3-5.

**c. Cổng collector để hở**

Nhược điểm của họ cổng TTL có mạch ra khép kín là hệ số tải đầu ra không thể thay đổi, nên nhiều khi gây khó khăn trong việc kết nối với đầu vào của các mạch điện tử tầng sau. Cổng logic collector để hở khắc phục được nhược điểm này. Hình 3-7 là sơ đồ của một cổng TTL đảo collector hở tiêu chuẩn. Muốn đưa cổng vào hoạt động, cần đấu thêm trở gánh ngoài, từ cực collector đến +Vcc.



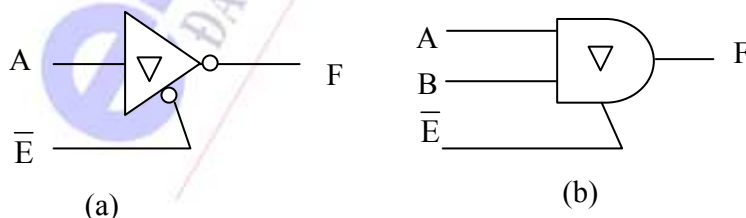
**Hình 3-7.** Mạch điện của một cổng NOT collector hở.

Một nhược điểm của cổng logic collector hở là tần số hoạt động của mạch sẽ giảm xuống do phải sử dụng điện trở gánh ngoài.

**d. Cổng TTL 3 trạng thái**

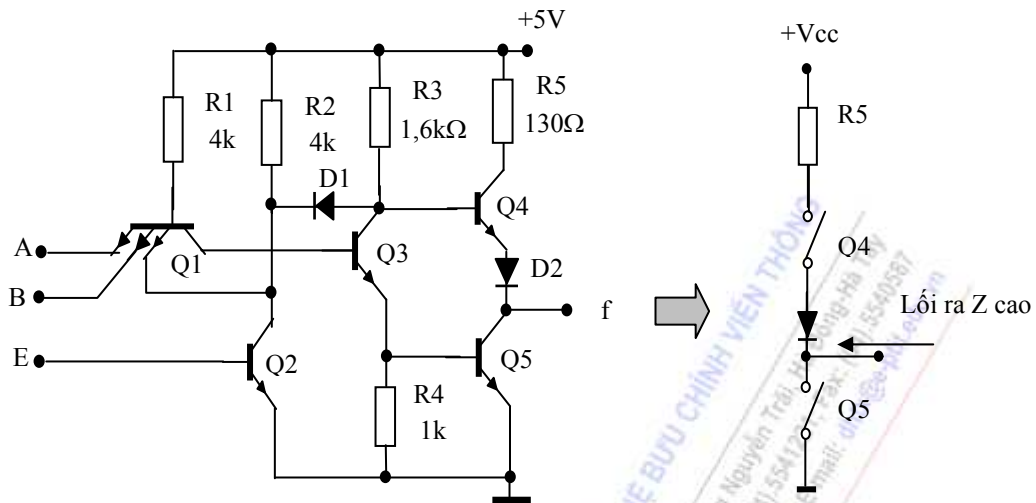
Một cổng logic, ngoài hai trạng thái cao và thấp tại đầu ra của nó còn có một trạng thái trung gian được gọi là cổng ba trạng thái. Trạng thái trung gian này còn có tên là trạng thái đầu ra có trở kháng Z cao hay trạng thái treo. Cổng có ký hiệu như chỉ ở hình 3-8.

Tương tự như cổng collector hở, các họ cổng logic đều có cổng 3 trạng thái. Hình 3-8 là một ví dụ về mạch điện của cổng NAND ba trạng thái họ TTL tiêu chuẩn .



**Hình 3-8.** Ký hiệu của cổng ba trạng thái : (a) cổng NOT; (b) cổng AND.

Hoạt động của cổng NAND 3 trạng thái được giải thích bằng bảng trạng thái 3-3. Khi trên lối vào E có mức logic thấp, cổng hoạt động như một cổng NAND. Trên lối ra f sẽ tồn tại hai trạng thái cao và thấp như thường lệ.



Hình 3-9. Mạch điện cổng NAND 3 trạng thái và sơ đồ tương đương của nó.

E	A	B	f
L	L	L	H
L	L	H	H
L	H	L	H
L	H	H	L
H	x	x	-
H	x	x	-
H	x	x	-
H	x	x	-

Bảng 3-3. Bảng trạng thái của cổng 3 trạng thái.

Ngược lại, khi trên lối vào E ở mức cao thì bất luận trên hai lối vào A, B có giá trị logic nào (dấu x trong bảng trạng thái mang ý nghĩa tùy chọn) lối ra f luôn ở trạng thái treo, hay thả nổi. Trạng thái này tương đương với trạng thái đầu ra không được nối tới một điểm nào trong mạch. Ứng với trạng thái này, trở kháng Z trên đầu ra của cổng, nhìn từ phía tải vào sẽ rất lớn. Theo sơ đồ tương đương, lúc này cả Q<sub>4</sub>, Q<sub>5</sub> đều khoá. Lối ra f dường như bị treo trong mạch. Do đó, trạng thái này còn được gọi là trạng thái treo.

Trong kỹ thuật số, cổng ba trạng thái thường được dùng làm các bộ đệm đầu ra, khoá điều khiển hướng dữ liệu ...

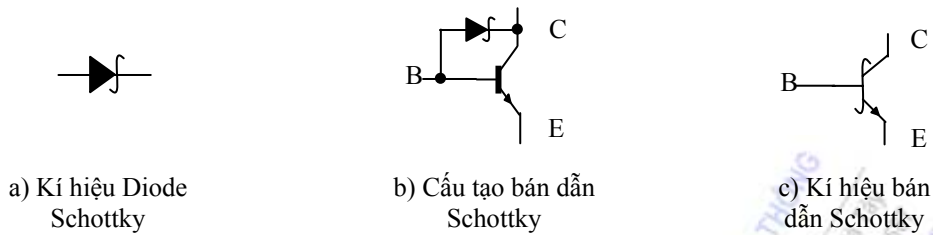
### e. Họ TTL có diode Schottky ( TTL + S )

Cổng TTL tiêu chuẩn có nhược điểm chung là thời gian trễ truyền lan lớn. Nguyên nhân của nhược điểm này là do tất cả bán dẫn trong mạch đều công tác ở chế độ bão hoà. Một trong những biện pháp giảm nhỏ trễ truyền lan là sử dụng diode Schottky để chống hiện tượng bão hoà này.

#### – Diode và bán dẫn Schottky

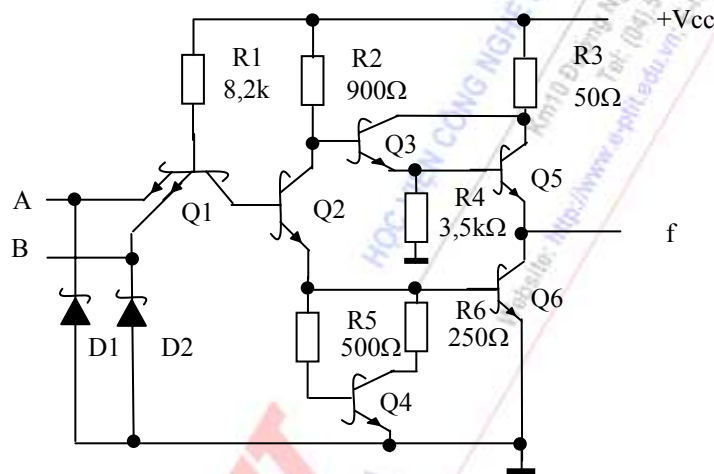
Cấu tạo của diode Schottky cũng giống như diode Silic. Nhờ việc chèn thêm một lớp oxit kim loại vào giữa tiếp giáp p-n mà điện thế phân cực của nó là 0,4 Vdc (thấp hơn 0,6 vôn đối với diode Silic và cao hơn 0,2 với diode Ge).

Ký hiệu của diode và bán dẫn Schottky cho ở hình 3-10.



Hình 3-10. Cấu tạo của diode Schottky

Mạch điện dùng diode Schottky chống bão hoà cho các bán dẫn như hình 3-10b. Để đơn giản, người ta gọi mạch này là bán dẫn Schottky và ký hiệu như hình 3-10c.



Hình 3-11. Mạch điện của cổng NAND 2 lối vào họ TTL+S

### – Mạch điện họ cổng TTL + S

Nếu thay tất cả diode và bán dẫn trong mạch điện của họ TTL tiêu chuẩn bằng các diode và bán dẫn Schottky, ta sẽ có mạch điện họ cổng TTL+S. Hình 3-11 là một ví dụ về cổng NAND dùng diode Schottky.

Nhờ sử dụng diode và bán dẫn Schottky mà tần số công tác của họ cổng này tăng đáng kể. Thời gian trễ truyền lan của cổng TTL+S khoảng 3 ns, công suất tiêu thụ khoảng 19 mW.

Khi chỉ tiêu thời gian trễ không cần cao thì giá trị các điện trở phân cực được tăng lên để giảm dòng tiêu thụ của mỗi bán dẫn xuống. Họ cổng như thế có tên gọi là TTL+LS (Transistor Transistor Logic + Lowpower Schottky Diode). Công suất tiêu thụ của họ cổng này chỉ khoảng 2 mW và thời gian trễ truyền lan vẫn đạt khoảng 9,5 ns.

Nếu cần nâng cao tần số công tác, ngoài việc giảm trị số các điện trở phân cực, người ta còn dùng các cách nối mạch cải tiến. Họ cổng thu được có tên là TTL+AS.



### 3.1.5. Họ MOS FET

Bán dẫn trường (MOS FET) cũng được dùng rất phổ biến để xây dựng mạch điện các loại cổng logic. Đặc điểm chung và nổi bật của họ này là:

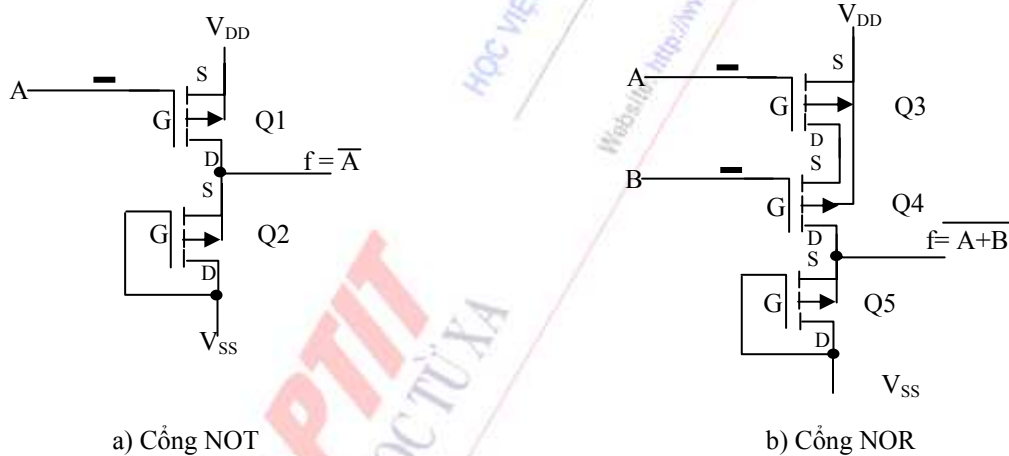
- Mạch điện chỉ bao gồm các MOS FET mà không có điện trở
- Dải điện thế công tác rộng, có thể từ +3 đến +15 V
- Độ trễ thời gian lớn, nhưng công suất tiêu thụ rất bé

Tùy theo loại MOS FET được sử dụng, họ này được chia ra các tiểu họ sau.

#### 1. Loại PMOS

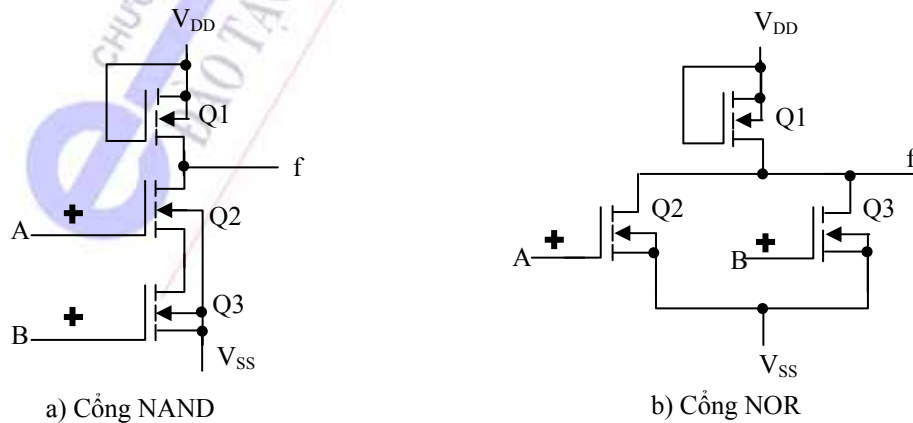
Mạch điện của họ cổng này chỉ dùng MOSFET có kênh dẫn loại P. Công nghệ PMOS cho phép sản xuất các mạch tích hợp với mật độ cao nhất.

Hình 3-12 là sơ đồ cổng NOT và cổng NOR loại PMOS. Ở đây MOSFET Q2, Q5 đóng chức năng các điện trở.



**Hình 3-12.** Mạch điện của cổng NOT và NOR theo công nghệ PMOS.

#### 2. Loại NMOS



**Hình 3-13.** Mạch điện cổng NAND và NOR theo công nghệ NMOS.

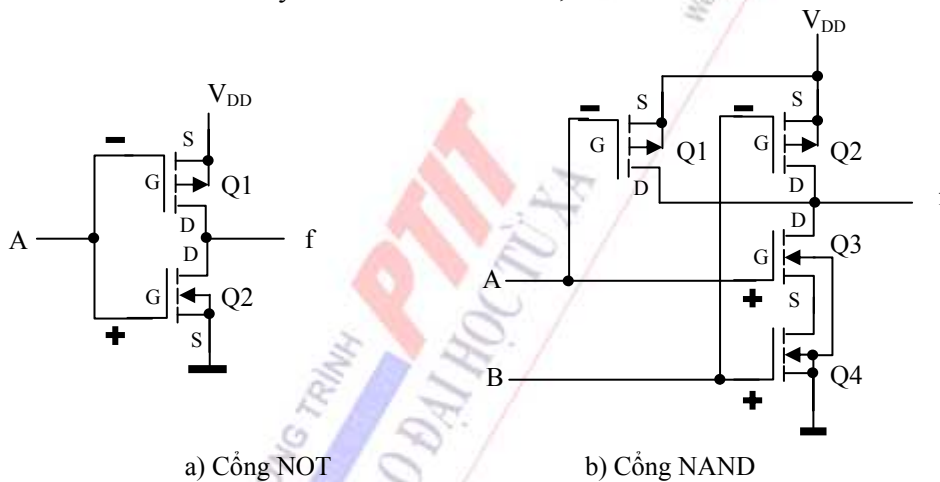
Hình 3-13 là sơ đồ cổng NAND và NOR dùng NMOS. Dấu + trên các lối vào muốn chỉ cực tính của tín hiệu kích thích. Trong trường hợp này,  $Q_1$  cũng đóng chức năng là một điện trở. Đối với cổng NAND, ta nhận thấy rằng chỉ khi trên cả hai lối vào A và B đều lấy mức cao thì đầu ra mới có mức thấp. Ứng với 3 tổ hợp biến vào còn lại, lối ra f đều có logic thấp. Hoạt động của cổng NOR cũng được giải thích tương tự.

### 3. Cổng CMOS

CMOS là viết tắt các từ tiếng Anh “Complementary MOS”. Mạch điện của họ cổng logic này sử dụng cả hai loại MOS FET kênh dẫn P và kênh dẫn N. Bởi vậy có hiện tượng bù dòng điện trong mạch. Chính vì thế mà công suất tiêu thụ của họ cổng, đặc biệt trong trạng thái tĩnh là rất bé.

Hình 3-14 là mạch điện của cổng NOT và NAND thuộc họ CMOS. Điểm nổi bật trong mạch điện của họ cổng này là không tồn tại vai trò các điện trở. Chức năng logic được thực hiện bằng cách thay đổi trạng thái các chuyển mạch có cực tính ngược nhau. Dấu trừ và dấu cộng trên cực của các MOSFET chỉ ra cực tính điều khiển chuyển mạch. Nhờ đặc điểm cấu trúc mạch, mức  $V_{RL}$ ,  $V_{RH}$  đạt được gần như lý tưởng.

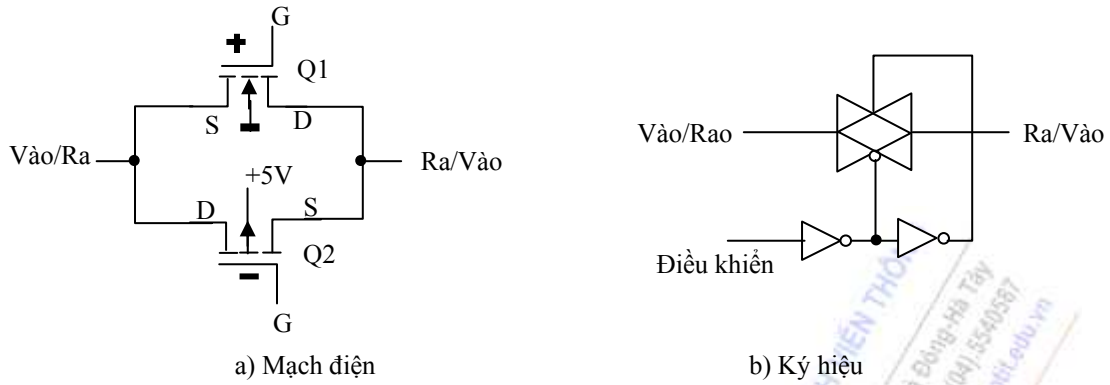
Để minh họa, ta có thể tìm hiểu hoạt động của cổng NOT. Từ hình 3-14a, dễ thấy rằng, nếu tác động tới lối vào A logic thấp thì  $Q_1$  sẽ thông,  $Q_2$  khoá. Lối ra f gần như được nối tắt tới  $V_{DD}$  và cách ly hẳn với đất, nghĩa là  $V_{RH} \approx V_{DD}$ . Ngược lại, khi A lấy mức cao,  $Q_1$  mở và  $Q_2$  đóng. Do đó, lối ra f gần như nối đất và cách ly với  $V_{DD}$ . Nói khác đi,  $V_{RL} \approx 0$ .



Hình 3-14. Mạch điện của họ cổng CMOS.

### 4. Cổng truyền dẫn

Dựa trên công nghệ CMOS, người ta sản xuất loại cổng có thể cho qua cả tín hiệu số lẫn tín hiệu tương tự. Bởi vậy cổng được gọi là cổng truyền dẫn. Sơ đồ nguyên lý và ký hiệu cổng truyền dẫn như hình 3-15.



**Hình 3-15.** Cổng truyền dẫn.

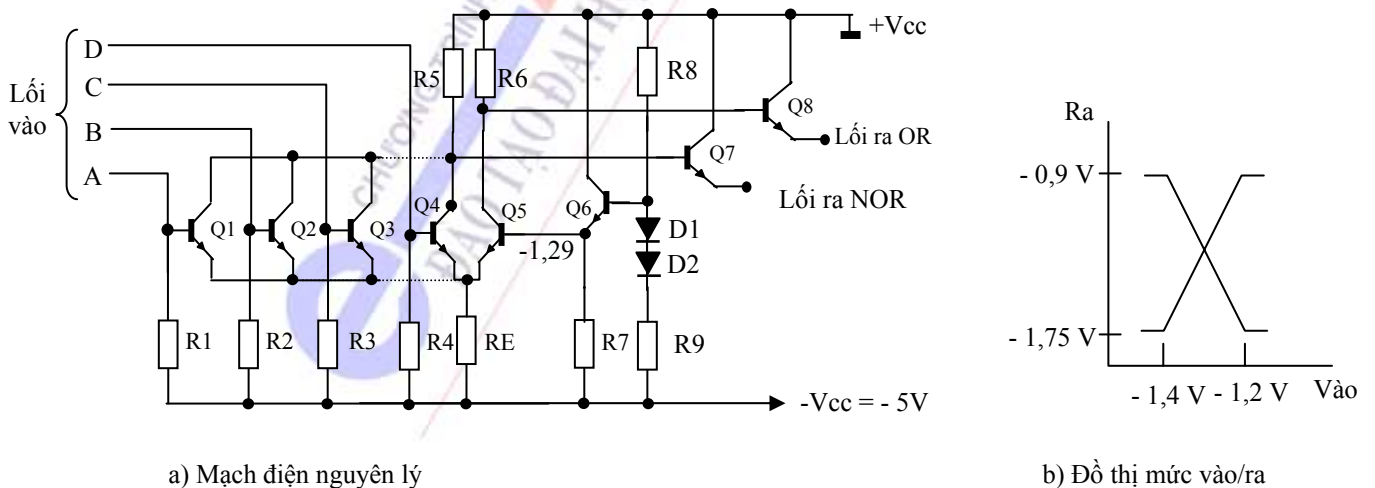
Mạch nguyên lý của cổng truyền dẫn cũng sử dụng hai MOSFET có kênh dẫn ngược nhau. Tuy nhiên cách điều khiển trạng thái các chuyển mạch lại khác với cổng logic thông thường. Trong trường hợp này, người ta phân cực sao cho khi có tín hiệu điều khiển thì cả hai chuyển mạch  $Q_1$  và  $Q_2$  cùng dẫn điện. Khi đó, mạch tương đương như một dây dẫn. Các cổng đảo (trong sơ đồ ký hiệu) đảm bảo cực tính điều khiển phù hợp cho cả hai cực G của mỗi MOSFET.

Tính dẫn điện của cổng truyền dẫn phụ thuộc mạnh vào tần số công tác và giá trị tải. Vì sử dụng công nghệ CMOS nên tần số công tác của cổng chỉ giới hạn ở 6 MHz.

Họ CMOS cũng có cổng D để hở và cổng ba trạng thái như họ TTL.

### 3.1.6- Họ ECL

ECL (Emitter Coupled Logic) là họ cổng logic có cực E của một số bán dẫn nối chung với nhau. Họ mạch này cũng sử dụng công nghệ TTL, nhưng cấu trúc mạch có những điểm khác hẳn với họ TTL. Ngoài việc sử dụng hồi tiếp âm trên điện trở  $R_E$  để chống bão hoà, mạch điện của họ ECL còn tận dụng được ưu điểm của mạch khuếch đại vi sai, nên tần số công tác họ này là cao nhất trong các họ. Ngoại trừ thời gian trễ, tất cả các tham số còn lại đều kém hơn các họ khác.



**Hình 3-16.** Cổng OR/NOR thuộc họ ECL.

Hình 3-16 là mạch điện và đồ thị mức vào ra của một cổng OR/NOR thuộc họ ECL. Vì điện thế ở trên hai cực collector của  $Q_4, Q_5$  là bù nhau nên có thể lấy ra ở cực E của  $Q_7$  chức năng OR và ở cực E của  $Q_8$  chức năng NOR. Để mạch hoạt động theo logic mức âm,  $+V_{cc}$  được nối đất, -

Vcc được nối tới âm nguồn. Mức logic trong mạch được biến đổi từ giá trị thấp là -1,75 V đến giá trị cao là -0,9 V so với điện thế đất. Khi muốn có mức logic ra dương các cực E nối tới đất.

### 3.2. GIAO TIẾP GIỮA CÁC CỔNG LOGIC CƠ BẢN TTL-CMOS VÀ CMOS-TTL

Trong nhiều ứng dụng, yêu cầu chuyển đổi các tín hiệu giữa các mức logic khác nhau như từ TTL sang CMOS hoặc ngược lại. Các cổng logic collector hở hoặc các mạch khuếch đại transistor đơn giản thường được sử dụng trong các mạch chuyển đổi này.

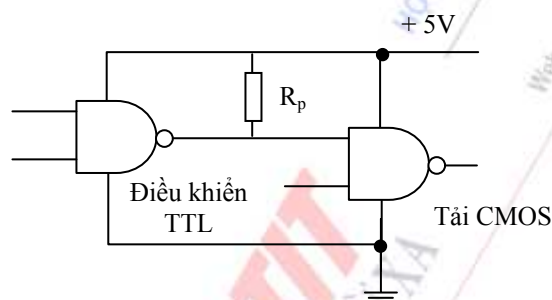
#### 3.2.1. Giao tiếp giữa TTL và CMOS.

Để tạo được giao tiếp giữa TTL và CMOS thì ta phải để ý đến nguồn cung cấp của 2 họ. Họ TTL cần điện áp cung cấp là +5V, họ CMOS có thể dùng điện áp cung cấp từ +3V đến +15V.

##### a. Cùng điện áp cung cấp +5V.

Trong trường hợp này điện áp ra của TTL nhỏ hơn so với điện áp vào của CMOS. Do vậy ta phải dùng mạch bổ sung để tương hợp hai loại IC khác nhau.

Giải pháp tiêu chuẩn là dùng điện trở kéo lên giữa điều khiển TTL và tải CMOS như hình 3-17.



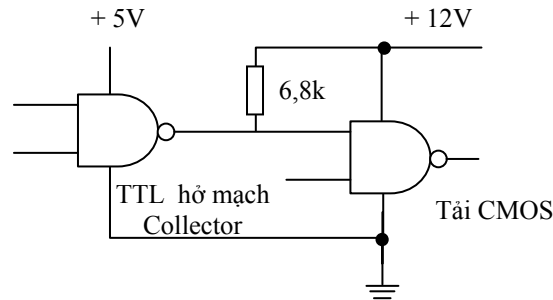
Hình 3-17. Điều khiển TTL và tải CMOS

##### b. Khác điện áp cung cấp.

Điện áp cung cấp dùng cho IC CMOS thích hợp nhất là từ +9V đến +12V. Một cách dùng để điện áp cung cấp lớn là sử dụng IC TTL hở mạch Collector như ở hình 3-18, vì tầng ra của TTL hở cực C chỉ gồm transistor nhận dòng với cực C thả nổi. Ở hình này cực C để hở được nối với nguồn cung cấp +12V qua điện trở kéo lên 6,8kΩ. Khi lối ra của họ TTL ở mức L thì dòng của nó là:

$$I_{\text{nhận dòng}} = \frac{12V}{6,8k\Omega} = 1,76mA$$

Khi lối ra của TTL ở mức H thì lối ra của cực C để hở tăng lên một cách thụ động đến +12V. Trong trường hợp nào thì các lối ra của TTL cũng đều tương hợp với các trạng thái ở lối vào của CMOS.

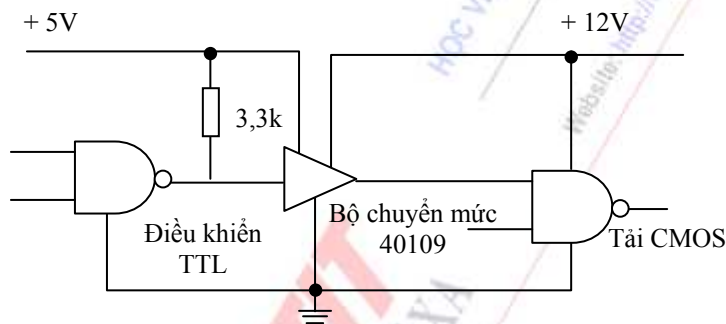


**Hình 3-18.** Điều khiển TTL hở mạch Collector và tải CMOS

**c. Bộ chuyển mức nguồn dùng CMOS.**

Hình 3-19 là bộ chuyển mức CMOS 40109. Tầng lối vào của IC dùng điện áp cung cấp +5V trong khi tầng lối ra dùng +12V.

Trong hình 3-19, IC TTL tiêu chuẩn điều khiển bộ chuyển mức nguồn, nó kéo IC TTL lên ít nhất là +2,4V. Điện trở kéo lên tiếp tục đưa điện áp lên cao đến mức +5V, mức này đảm bảo chắc chắn lối vào ở mức H. Lối ra của bộ chuyển mức nối với nguồn +12V.



**Hình 3-19.** Bộ chuyển mức CMOS cho phép sử dụng hai loại nguồn +5V và +12V.

**3.2.2. Giao tiếp giữa CMOS và TTL**

Để tạo ra được giao tiếp giữa họ CMOS và TTL thì ta phải quan tâm đến vấn đề chuyển mức điện áp cho tới khi trạng thái lối ra của CMOS phù hợp với lối vào của TTL. Ta phải đảm bảo chắc chắn lối ra ở trạng thái L của CMOS luôn luôn nhỏ hơn 0,8 V (đây là điện áp lối vào lớn nhất ở trạng thái L của họ TTL). Điện áp lối ra ở trạng thái H của CMOS luôn luôn lớn hơn 2 V (đây là điện áp lối vào nhỏ nhất ở trạng thái H của họ TTL).

**a. Cùng điện áp cung cấp +5V.**

Theo số liệu kỹ thuật của IC 74Cxx thì trường hợp xấu nhất dòng lối ra của CMOS điều khiển TTL là:

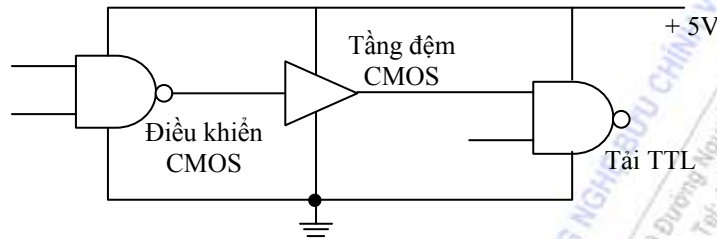
$$I_{OL\ MAX} = 360\mu A \quad ; \quad I_{OH\ MAX} = - 360\mu A$$

Điều này có nghĩa là điều khiển CMOS có thể cho nhận dòng là 360  $\mu$ A khi ở trạng thái L, đó là dòng vào đối với IC TTL loại Schottky công suất thấp. Mặt khác, điều khiển CMOS có thể cho dòng nguồn 360  $\mu$ A, nó lớn hơn mức cần thiết để điều khiển dòng vào ở trạng thái H. Như vậy hệ số ghép tải giữa CMOS và 74LS là bằng 1.

Đối với loại IC TTL công suất thấp thì có dòng lỗi vào là  $180 \mu\text{A}$  thì hệ số ghép tải giữa CMOS và 74L là bằng 2.

IC CMOS không thể điều khiển trực tiếp IC TTL tiêu chuẩn, vì dòng lỗi vào ở trạng thái L yêu cầu là  $1,6 \text{ mA}$ , mà transistor nhận dòng của IC CMOS có điện trở xấp xỉ  $1,1 \text{ k}\Omega$  (trường hợp xấu nhất). Nên điện áp lỗi ra của IC CMOS bằng  $1,6 \text{ mA} \times 1,1 \text{ k}\Omega = 1,78 \text{ V}$ . Điện áp này quá lớn đối với lỗi vào ở trạng thái L của IC TTL.

- Dùng tầng đệm bằng CMOS.



**Hình 3-20.** Tầng đệm CMOS có thể điều khiển tải TTL tiêu chuẩn

Hình 3-20 là mạch điều khiển IC CMOS với hệ số tải qua tầng đệm. Tầng đệm có dòng ra lớn. Ví dụ IC 74C902 có 6 tầng đệm CMOS, mỗi tầng đệm có dòng ở lỗi ra trong trường hợp xấu nhất là:

$$I_{OL \text{ MAX}} = 3.60 \text{ mA}$$

$$I_{OH \text{ MAX}} = 800 \mu\text{A}$$

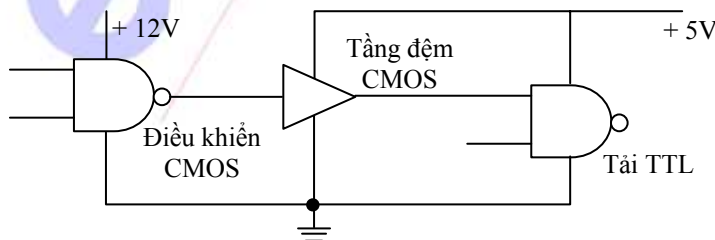
Vì tải TTL tiêu chuẩn có dòng lỗi vào ở trạng thái L bằng  $1,6 \text{ mA}$  và dòng lỗi vào ở trạng thái H là  $48 \mu\text{A}$ , IC 74C902 có thể điều khiển hai tải TTL tiêu chuẩn.

Các IC khác được dùng làm tầng đệm như hình 5-19 là IC CD4049A, 4050: đảo; CD405CA: không đảo, 74C901: đảo...

### b. Khác điện áp cung cấp.

Các tầng đệm CMOS như 74C902 có thể dùng điện áp cung cấp từ  $+3\text{V}$  đến  $+15\text{V}$  và điện áp lỗi vào từ  $-0,3 \text{ V}$  đến  $+15\text{V}$ . Điện áp lỗi vào có thể lớn hơn điện áp cung cấp mà không làm hỏng loại IC dùng làm tầng đệm này. Ví dụ ta có thể dùng điện áp lỗi vào ở trạng thái H là  $+12\text{V}$  ngay khi điện áp cung cấp chỉ bằng  $5\text{V}$ .

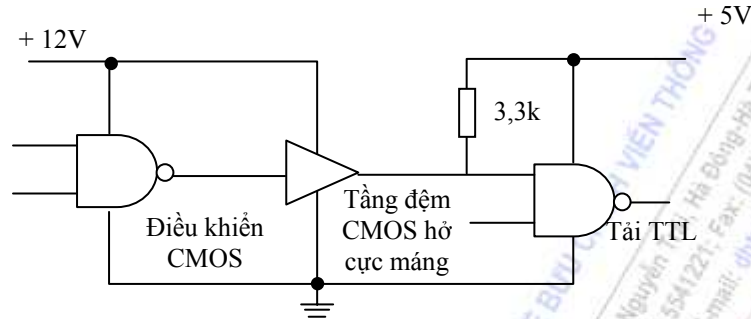
Hình 5-23 là mạch điều khiển CMOS dùng điện áp cung cấp  $+12\text{V}$ , trong khi tầng đệm CMOS có điện áp cung cấp là  $+5\text{V}$ .



**Hình 3-21.** Điều khiển CMOS hoạt động thích hợp nhất với nguồn cung cấp  $+12\text{V}$ .

**c. Giao diện của hở cực máng.**

Ta đã biết ở IC TTL hở mạch Collector, tầng lối ra của transistor nhận dòng với cực C thả nổi. Tương tự như vậy đối với IC CMOS cũng có hở cực máng. Ví dụ: IC 74C906 có 6 tầng đệm hở cực máng.



**Hình 3-22.** Tầng đệm CMOS hở cực máng làm tăng dòng nhận.

Hình 3-22 là mạch dùng tầng đệm CMOS hở cực máng làm giao diện điều khiển CMOS và tải TTL. Điện áp cung cấp cho hầu hết các tầng đệm là +12V. Tuy vậy có thể nối tầng đệm hở cực máng với nguồn cung cấp +5V qua một điện trở kéo lên (pull up) có giá trị 3,3kΩ. Cách nối này có ưu điểm là cả điều khiển CMOS và tầng đệm CMOS đều được cung cấp nguồn +12V, không kể lối ra hở cực máng giao diện với TTL

**TÓM TẮT**

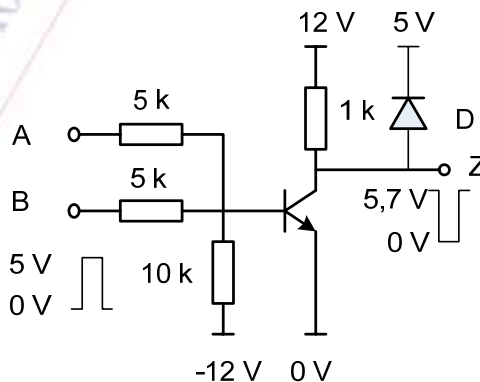
Chương 3 đã trình bày cấu trúc, nguyên lý và đặc điểm của cổng thường dùng. Xuất phát từ thực tế mạch điện đã vi mạch hoá, nên trọng tâm chú ý nghiên cứu của chúng ta là các cổng được vi mạch hoá.

Có 2 loại vi mạch số phổ biến nhất : TTL và MOS. TTL là công nghệ điển hình trong nhóm công nghệ transistor bao gồm TTL, HTL, ECL..., MOS là công nghệ vi mạch sử dụng MOSFET, trong đó điển hình là MOS...

Đồng thời trong chương 3 cũng đưa ra vấn đề giao tiếp giữa các họ cổng đó với nhau.

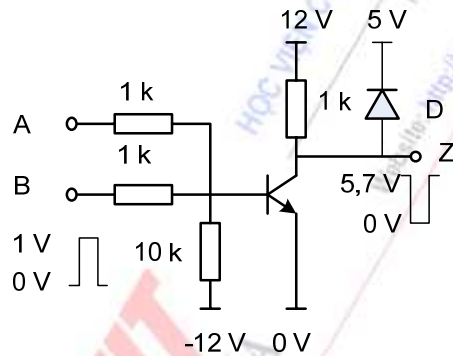
**CÂU HỎI ÔN TẬP**

1. Chức năng của mạch logic RTL có sơ đồ như hình vẽ sau:

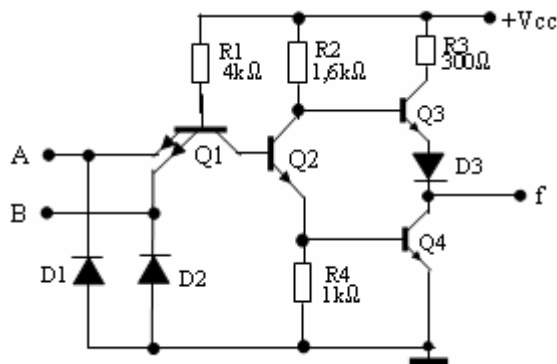


- a. NOR

- b. OR  
 c. AND  
 d. NAND
2. Với mạch có sơ đồ như trong câu hỏi 1, nhưng điện áp logic lồi vào tương ứng với các mức logic cao và thấp lần lượt là 10 V và 0 V thì chức năng của mạch là gì?
- a. NOR  
 b. OR  
 c. AND  
 d. NAND
3. Cho mạch có sơ đồ như sơ đồ sau, điện áp logic lồi vào tương ứng với các mức logic cao và thấp lần lượt là 1 V và 0 V, nêu chức năng của mạch?



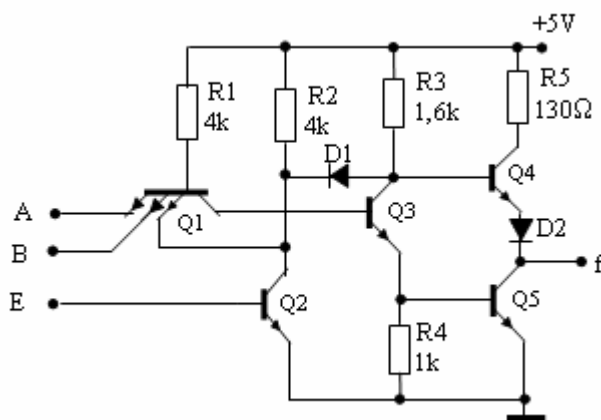
- a. NOR  
 b. OR  
 c. AND  
 d. NAND
4. Chức năng của diode  $D_3$  trong sơ đồ sau là gì?



- a. Cách ly transistor  $Q_3$  và  $Q_4$   
 b. Dịch mức điện áp làm cho  $Q_3$  và  $Q_4$  không bao giờ cùng đóng hoặc cùng mở



- c. Chống nhiễu lối ra
  - d. Cách ly  $Q_4$  khỏi mạch ngoài nối vào đầu ra f
5. Chức năng của mạch biểu diễn trong sơ đồ như câu hỏi 4 sẽ thay đổi thế nào nếu diode  $D_3$  chuyển tới chân base của transistor  $Q_3$  (cathode  $D_3$  nối với base  $Q_3$  còn anode nối với collector  $Q_2$ )?
- a.  $Q_3$  luôn cấm
  - b.  $Q_3$  luôn mở
  - c. Chức năng của mạch không thay đổi
  - d. Lối ra luôn ở trạng thái treo
6. Cổng collector hở sẽ hoạt động bình thường như các cổng logic bình thường nếu :
- a. Lối ra được nối lên nguồn thông qua một trở gánh
  - b. Lối ra được nối lên nguồn thông qua một tụ gánh
  - c. Lối ra nối xuống đất thông qua một trở
  - d. Lối ra nối xuống đất thông qua một tụ
7. Tác dụng của trạng thái trở kháng lối ra cao trong cổng ba trạng thái là :
- a. Đưa ra mức logic thứ 3 là trung bình của hai mức cao và thấp
  - b. Cách ly giữa các lối ra của các cổng logic khi chúng cùng được nối vào một lối vào
  - c. Có mức logic thấp nhưng trở kháng cao
  - d. Có mức logic cao nhưng trở kháng cao
8. Mạch điện được biểu diễn trong sơ đồ sau có còn hoạt động như bình thường không nếu như diode  $D_1$  bị nối tắt ?



- a. Mạch trở thành cổng NAND với hai trạng thái lối ra như các cổng NAND thường
- b. Mạch trở thành cổng NOR

- c. Trạng thái lỗi ra không theo logic cơ bản nào
- d. Vẫn hoạt động bình thường là cổng NAND 3 trạng thái
9. Mạch điện như trong câu hỏi 8 có còn hoạt động như bình thường không nếu như điện trở  $R_4$  có giá trị bằng 10 k?
- a. Nó sẽ hoạt động như mạch NOR
- b. Nó sẽ hoạt động như mạch XOR
- c. Vẫn hoạt động bình thường
- d. Cả ba cách trả lời trên đều sai
10. Với mạch điện TTL như sơ đồ trong câu hỏi 4, hiện tượng gì sẽ xảy ra khi một trong hai lỗi vào để lửng?
- a. Lỗi vào này được tính logic 0
- b. Lỗi vào này được tính logic 1
- c. Mạch không hoạt động
- d. Cả ba cách trả lời trên đều sai
11. So sánh cổng NOT họ MOS và CMOS ta thấy :
- a. Công suất tiêu thụ của MOS cao hơn CMOS
- b. Công suất tiêu thụ của CMOS cao hơn MOS
- c. Công suất tiêu thụ của hai họ như nhau
- d. Cả ba cách trả lời trên đều sai
12. Có cho phép đầu vào của mạch CMOS để lơ lửng không? Có thể nói đầu vào để lửng tương đương với mức cao không?
- a. Được- Có thể coi là mức 1
- b. Được- Phải coi là mức 0
- c. Không được- Để mạch hoạt động bình thường thì đầu vào không dùng phải nối với mức logic 0
- d. Không được- Để mạch hoạt động bình thường thì đầu vào không dùng phải nối với mức logic 1
13. Cổng truyền dẫn là cổng
- a. Chỉ cho phép tín hiệu số đi qua theo một chiều nhất định
- b. Chỉ cho phép tín hiệu số đi qua theo hai chiều
- c. Chỉ cho phép tín hiệu tương tự đi qua theo một chiều nhất định
- d. Cho phép tín hiệu tương tự đi qua theo hai chiều

14. Ưu điểm của các cổng logic họ ECL là

- a. Tần số công tác nhanh
- b. Điện áp nguồn nuôi thấp
- c. Công suất tiêu thụ thấp
- d. Độ chống nhiễu cao



## CHƯƠNG 4: MẠCH LOGIC TỔ HỢP

### GIỚI THIỆU CHUNG

Các hàm logic được thực hiện nhờ các hệ vật lý gọi là các *hệ logic* hay là các *mạch logic*. Trong chương 4 chúng ta đề cập đến các mạch logic tổ hợp, tức là các mạch mà tín hiệu ở đầu ra chỉ phụ thuộc vào tín hiệu ở đầu vào của mạch tại thời điểm đang xét. Nói cách khác, các tín hiệu ra không phụ thuộc vào "lịch sử" của tín hiệu vào trước đó, nghĩa là các hệ này làm việc theo nguyên tắc không có nhớ. Hoạt động của các mạch tổ hợp được mô tả bằng các bảng trạng thái hoặc bằng các hàm chuyển mạch logic đặc trưng cho quan hệ giữa các đại lượng vào và ra của hệ thống. Về mặt cấu trúc, các mạch tổ hợp không chứa một thiết bị hoặc một phần tử nhớ thông tin nào cả.

Trong chương này đề cập đến các mạch điện cụ thể thực hiện các chức năng khác nhau của hệ thống số. Các mạch điện này được thiết kế dựa trên các cổng logic tổ hợp. Các cổng logic này được tích hợp trong một IC cỡ vừa (MSI) có chứa khoảng vài chục tới vài trăm các cổng logic cơ sở đó được xét đến ở chương 4. Những linh kiện này được chế tạo nhằm thực hiện một số các hoạt động thu nhận, truyền tải, biến đổi các dữ liệu thông qua tín hiệu nhị phân, xử lý chúng theo một phương thức nào đó.

Phần đầu của chương giới thiệu cách phân tích và thiết kế các mạch logic tổ hợp đơn giản.

Phần tiếp theo giới thiệu về Hazard trong mạch logic tổ hợp. Đây là phần rất quan trọng khi thiết kế mạch. Nếu không để ý đến hiện tượng này có thể dẫn đến sự làm việc sai lệch của cả hệ thống. Phân tích và nhận dạng Hazard có ý nghĩa rất quan trọng không những trong tổng hợp các hệ logic mà cả trong tự động chẩn đoán trạng thái làm việc của chúng.

Phần tiếp theo giới thiệu một số mạch tổ hợp thông dụng trong các hệ thống số:

- Mã hoá và giải mã các luồng dữ liệu nhị phân.
- Hợp kênh và phân kênh để chọn hoặc chia tách các luồng số nhị phân theo những yêu cầu nhất định để định tuyến cho chúng trong việc truyền dẫn thông tin,
- Các mạch cộng, trừ.
- Các phép so sánh số để đánh giá định tính và định lượng trọng số của các số nhị phân.
- Mạch tạo và kiểm tra tính chẵn lẻ.
- Đơn vị số học và logic (ALU).

## NỘI DUNG

### 4.1 KHÁI NIỆM CHUNG

Căn cứ vào đặc điểm và chức năng logic, các mạch số được chia thành 2 loại chính: mạch tổ hợp và mạch tuần tự (mạch tuần tự được trình bày ở chương sau).

#### 1) Đặc điểm cơ bản của mạch tổ hợp

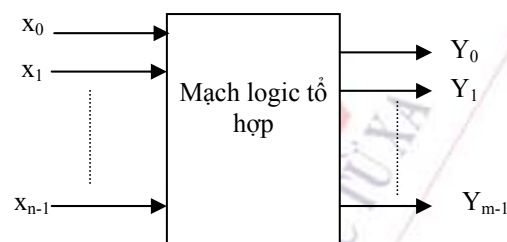
Trong mạch số, mạch tổ hợp là mạch mà trị số ổn định của tín hiệu đầu ra ở thời điểm đang xét chỉ phụ thuộc vào tổ hợp các giá trị tín hiệu đầu vào. Đặc điểm cấu trúc mạch tổ hợp là được cấu trúc nên từ các cổng logic. Vậy các mạch điện cổng ở chương 2 và các mạch logic ở chương 3 đều là các mạch tổ hợp.

#### 2) Phương pháp biểu diễn chức năng logic

Các phương pháp thường dùng để biểu diễn chức năng logic của mạch tổ hợp là hàm số logic, bảng trạng thái, sử dụng logic, bảng Cac nô (Karnaugh), cũng có khi biểu thị bằng đồ thị thời gian dạng xung.

Đối với vi mạch cỡ nhỏ (SSI) thường biểu diễn bằng hàm logic. Đối với vi mạch cỡ vừa (MSI) thường biểu diễn bằng bảng trạng thái.

Sơ đồ khối tổng quát của mạch logic tổ hợp được trình bày ở hình 4-1.



**Hình 4-1** Sơ đồ khối tổng quát của mạch logic tổ hợp.

Như vậy, mạch logic tổ hợp có thể có  $n$  lối vào và  $m$  lối ra. Mỗi lối ra là một hàm của các biến vào. Quan hệ vào, ra này được thể hiện bằng hệ phương trình tổng quát sau:

$$Y_0 = f_1(x_0, x_1, \dots, x_{n-1});$$

$$Y_1 = f_2(x_0, x_1, \dots, x_{n-1});$$

...

$$Y_{m-1} = f_{m-1}(x_0, x_1, \dots, x_{n-1}).$$

Từ đó, ta thấy rằng đặc điểm nổi bật của mạch logic tổ hợp là hàm ra chỉ phụ thuộc các biến vào mà không phụ thuộc vào trạng thái của mạch. Cũng chính vì thế, trạng thái ra chỉ tồn tại trong thời gian có tác động vào.

Thể loại của mạch logic tổ hợp rất phong phú. Phạm vi ứng dụng của chúng cũng rất rộng.

## 4.2 PHÂN TÍCH MẠCH LOGIC TỔ HỢP

Phân tích mạch logic tổ hợp là đánh giá, phê phán một mạch đó. Trên cơ sở đó, có thể rút gọn, chuyển đổi dạng thực hiện của mạch điện để có được lời giải tối ưu theo một nghĩa nào đấy.

Mạch tổ hợp có thể bao gồm hai hay nhiều tầng, mức độ phức tạp của của mạch cũng rất khác nhau.

Nếu mạch đơn giản thì ta tiến hành lập bảng trạng thái, viết biểu thức, rút gọn, tối ưu (nếu cần) và cuối cùng vẽ lại mạch điện.

Nếu mạch phức tạp thì ta tiến hành phân đoạn mạch để viết biểu thức, sau đó rút gọn, tối ưu (nếu cần) và cuối cùng vẽ lại mạch điện.

## 4.3 THIẾT KẾ MẠCH LOGIC TỔ HỢP

Thiết kế là bài toán ngược với bài toán phân tích. Nội dung thiết kế được thể hiện theo tuần tự sau:

1- Phân tích bài toán đã cho để gắn hàm và biến, xác lập mối quan hệ logic giữa hàm và các biến đó;

2- Lập bảng trạng thái tương ứng;

4- Từ bảng trạng thái có thể viết trực tiếp biểu thức đầu ra hoặc thiết lập bảng Các nô tương ứng;

4- Dùng phương pháp thích hợp để rút gọn, đưa hàm về dạng tối giản hoặc tối ưu theo mong muốn;

5- Vẽ mạch điện thể hiện.

Ví dụ : Một ngôi nhà hai tầng. Người ta lắp hai chuyển mạch hai chiều tại hai tầng, sao cho ở tầng nào cũng có thể bật hoặc tắt đèn. Hãy thiết kế một mạch logic mô phỏng hệ thống đó?

Lời giải:

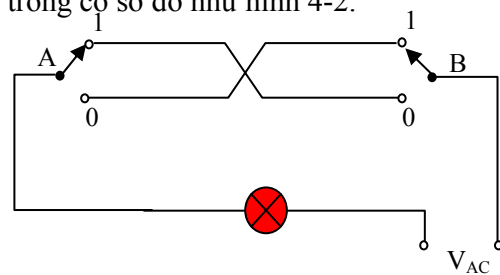
+ Nếu ký hiệu hai công tắc là hai biến A, B. Khi ở tầng 1 ta bật đèn và lên tầng 2 thì tắt đèn đi và ngược lại. Như vậy đèn chỉ có thể sáng ứng với hai tổ hợp chuyển mạch ở vị trí ngược nhau. Còn đèn tắt khi ở vị trí giống nhau. Hệ thống chiếu sáng trong sơ đồ như hình 4-2.

Bảng trạng thái mô tả hoạt động của hệ như chỉ ở bảng 4-1.

Biểu thức của hàm là:  $f = \bar{A}B + A\bar{B} = A \oplus B$

hoặc

$$f = \overline{AB} \overline{A \bar{B}}$$

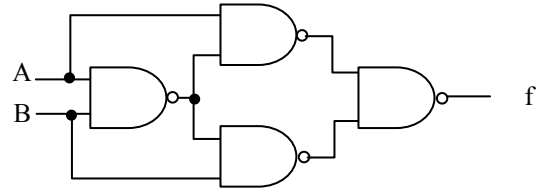


Hình 4-2 Mạch điện của hệ thống chiếu sáng

Đây là hàm cộng XOR đã quen thuộc ở các chương trước. Hàm này có thể được thể hiện bằng nhiều kiểu mạch khác nhau. Hình 4-3 là một dạng sơ đồ thể hiện hàm f.

A	B	f
0	0	0
0	1	1
1	0	1
1	1	0

Bảng 4-1. Bảng trạng thái mô tả hoạt động của hệ chiếu sáng



Hình 4-3. Sơ đồ logic thể hiện hàm f

## 4.4 HAZARD TRONG MẠCH TỔ HỢP

### 4.4.1. Khái niệm.

Việc thiết kế các mạch logic nhìn chung không phức tạp, vì cần có biểu thức toán là ta có thể vẽ ra được mạch điện và lắp ráp thành hệ thống điều khiển. Trên thực tế, không phải mạch nào cũng có thể hoạt động tốt được, nguyên nhân là do cấu trúc của mạch tổ hợp gây ra, hiện tượng hoạt động không ổn định xảy ra trong mạch tổ hợp được gọi là *hazard*.

Hazard còn được gọi là sự "sai nhầm", hoạt động lúc được lúc không của mạch logic. Sự "sai nhầm" này có thể xảy ra trong một mạch điện hoàn toàn không có hỏng hóc linh kiện. Tức là trong mạch, các linh kiện hoàn toàn tốt nhưng điều khiển chức năng lúc được lúc không. Nói chung là mạch hoạt động không có sự tin cậy. Hiện tượng của Hazard trong mạch tổ hợp có thể gặp là:

- Hazard chỉ xuất hiện một lần và không bao giờ gặp lại nữa.
- Hazard có thể xuất hiện nhiều lần (theo một chu kỳ nào đó hoặc không theo một chu kỳ nào).
- Hazard có thể do chính chức năng của mạch điện gây ra. Đây là trường hợp khó giải quyết nhất khi thiết kế.

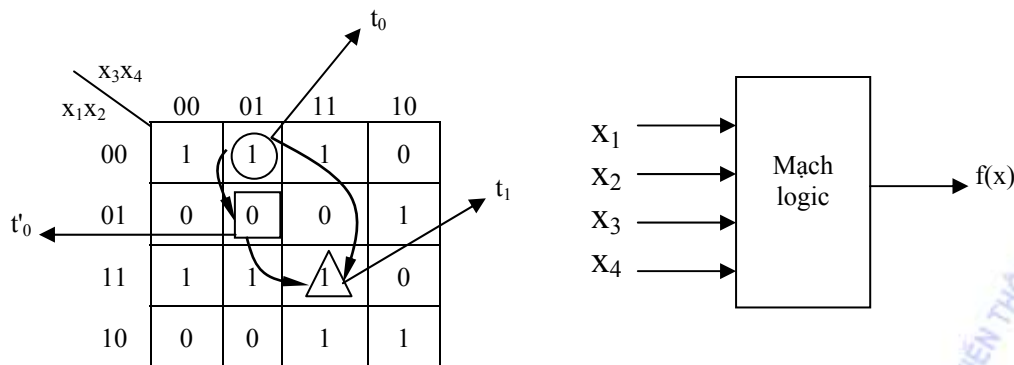
Như ta đã biết, một trong các đặc tính quan trọng nhất của mạch điện khi hoạt động là quán tính, độ linh động hay sự chậm trễ của mạch. Chính sự chậm trễ này làm cho tín hiệu từ đầu vào không thể truyền ngay tức khắc tới đầu ra của mạch điện, điều này làm cho các thiết bị điều khiển phía sau không thể có phản ứng tức khắc đối với tín hiệu đưa vào. Do tất cả các mạch điện đều có thời gian trễ nhất định, ngay cả ở các mạch vi điện tử cũng có thời gian trễ. Sự thay đổi nhiệt độ môi trường cũng làm cho thời gian trễ thay đổi, dẫn đến sự sai lệch khi điều khiển của mạch logic, đó chính là hazard.

### 4.4.2. Bản chất của Hazard

Để hiểu được nguyên nhân xuất hiện hazard trong mạch logic tổ hợp, hazard chỉ xuất hiện trong mạch tổ hợp mà không xuất hiện ở bất kỳ hệ thống điện tử nào khác. Ta xét ví dụ sau:

Giả sử tín hiệu vào là  $X = (x_1, x_2, x_3, x_4)$  thay đổi giá trị từ (0 0 0 1) đến (1 1 1 1), tức là (X) thay đổi từ Q→P. Nhìn vào bảng Các nô (hình 4-4) ta thấy đáp ứng ra của mạch logic tổ hợp khi tín hiệu vào bị thay đổi có giá trị:

$$f(Q) = f(0001) = \textcircled{1} \rightarrow f(P) = f(1111) = \textcircled{1}$$



Hình 4-4. Mạch chức năng logic

Như vậy tín hiệu vào (X) thay đổi giá trị từ Q = (0001) đến P = (1111) làm cho đáp ứng ra của mạch bị thay đổi giá trị từ  $\textcircled{1}$  sang  $\triangle 1$ , sự thay đổi điều khiển ở đầu ra của mạch theo sự thay đổi tín hiệu vào (X)  $\rightarrow$  điều này hoàn toàn chính xác, khi đó hazard không xuất hiện và không xảy ra điều khiển bị sai nhầm.

Nhưng thực tế có thể không được như vậy vì khi tín hiệu vào thay đổi từ Q = (0001) đến P = (1111), ta thấy tín hiệu  $x_1, x_2, x_3$  bị thay đổi còn giá trị  $x_4$  không bị thay đổi. Mạch điện nào cũng xuất hiện thời gian trễ là ( $\tau$ ) và sự thay đổi giá trị (0 $\rightarrow$ 1 hay 1 $\rightarrow$ 0) của tín hiệu đều có thời gian trễ nhất định.

Trong trường hợp này, các tín hiệu vào ( $x_1, x_2, x_3$ ) có giá trị logic bị thay đổi khi ta thay đổi bộ tín hiệu vào, và chúng sẽ có một thời gian trễ nhất định (có thể rất nhỏ, cỡ  $\mu s$  hay ns). Mặt khác, thời gian trễ của mỗi đường tín hiệu vào ( $x_i$ ) lại khác nhau, dù cùng một chủng loại IC. Như vậy nếu ( $x_1, x_2, x_3$ ) được thay đổi đồng thời và chúng có thời gian trễ khác nhau thì vẫn xảy ra hiện tượng "chạy đua" của tín hiệu vào tới đầu ra của mạch điện.

Vì có sự "chạy đua" giữa ba tín hiệu vào ( $x_1, x_2, x_3$ ) ( $x_4$  không thay đổi nên không đua), giả sử  $x_2$  chạy nhanh hơn (có thời gian trễ nhỏ hơn)  $x_1, x_3$  (giả sử thời gian trễ của hai tín hiệu này bằng nhau). Mối quan hệ này ta có thể biểu diễn như sau:

(X)	—	( $x_1$ )	( $x_2$ )	( $x_3$ )	( $x_4$ )	Đáp ứng ra
$t_0$	—	0	0	0	1	$f(Q) = \textcircled{1}$
			↓			↓
$t'_0$	—	0	1	0	1	$f(0101) = \square 0$
			↓	↓		↓
$t_1$	—	1	1	1	1	$f(P) = \triangle 1$

Do  $x_2$  "chạy" nhanh hơn  $x_1$  và  $x_3$  nên giá trị của  $x_2$  chuyển từ 0 sang 1 trước giá trị của  $x_1$  và  $x_3$ . Sau một thời gian thì ( $x_1, x_3$ ) mới chuyển từ 0 sang 1.

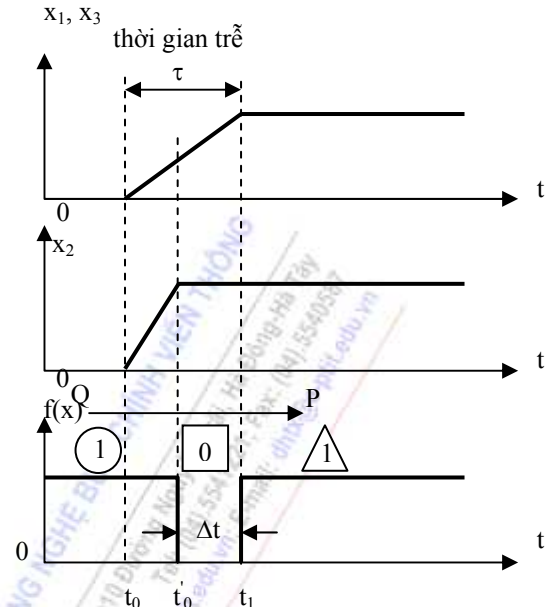
Quan hệ "chạy đua" giữa ba tín hiệu vào được minh họa bằng biểu đồ sau:

Do  $x_2$  "chạy nhanh" hơn ( $x_1, x_3$ ) nên trong khoảng thời gian  $\Delta t$  đã xuất hiện một xung zêrô nhất thời. Như vậy trong thời gian trễ  $\tau$  của mạch tín hiệu ra đã thay đổi từ 1 $\rightarrow$ 0 $\rightarrow$ 1 (đúng ra là không được thay đổi), tạo ra một xung kim nhất thời. Hiện tượng xuất hiện một xung zêrô ở đầu ra của mạch được gọi là hiện tượng hazard và đây là hazard nhất thời, nó chỉ xuất hiện trong thời gian trễ  $\tau$  sau đó lại mất ngay. Như vậy ta có thể nói rằng sự "chạy đua" của tín hiệu vào gây ra



hazard, hay thời gian trễ của mạch sẽ làm xuất hiện hazard, đó là tín hiệu điều khiển không mong muốn ở đầu ra.

Xung Hazard là một xung kim xuất hiện ở đầu ra của mạch logic tổ hợp, vì thời gian xuất hiện ( $\Delta t$ ) nhỏ hơn thời gian trễ của mạch ( $\tau$ ) nên xung hazard có thể xuất hiện nhưng không gây nguy hiểm, không gây ra sự điều khiển sai lầm. Vì xung hazard quá hẹp nên năng lượng của nó không đủ lớn để có thể kích nhảm hay kích được các mạch điện tiếp theo, do đó dù có xung hazard nhưng mạch điện vẫn hoạt động tốt. Xung hazard chỉ thật sự nguy hiểm khi độ rộng  $\Delta t$  đủ lớn thì nó có đủ năng lượng để lật chuyển mạch điện tiếp theo gây ra hiện tượng điều khiển nhảm.



Hình 4-5. Hiện tượng hazard

Như vậy có thể thấy với bộ tín hiệu vào thay đổi kiểu khác với tổ hợp trên thì có thể không xuất hiện xung hazard. Hay với một chức năng khác dù có hiện tượng "chạy đua" tín hiệu vào giữa ( $x_1, x_3$  và  $x_2$ ) như ví dụ trên nhưng có  $f(0101) = 1$  thì hazard cũng không thể xuất hiện do xung zêrô nhất thời không có. Do vậy ta thấy hiện tượng hazard xuất hiện rất ngẫu nhiên cho dù mạch điện chứa toàn các linh kiện tốt.

#### 4.4.3. Phân loại.

Đầu tiên ta đề cập đến một số định nghĩa tên gọi khi nói về hazard như sau:

$$Q = (q_1, q_2, \dots, q_k, q_{k+1}, \dots, q_n)$$

$$P = (\overline{q_1}, \overline{q_2}, \dots, \overline{q_k}, q_{k+1}, \dots, q_n)$$

Ở đây P và Q là tập tín hiệu vào của mạch, nhưng yêu cầu giữa P và Q cần có số lượng vị trí thay đổi giá trị logic  $\geq 2$ , vì chỉ khi tập tín hiệu vào thay đổi giá trị logic đồng thời với ít nhất 2 vị trí (2 biến số) thì mới xuất hiện hiện tượng "chạy đua" tín hiệu vào, và khi đó hazard mới có khả năng xuất hiện. Còn nếu tín hiệu vào chỉ thay đổi giá trị lần lượt trên từng đầu vào một thì sẽ không có hiện tượng chạy đua tín hiệu và hazard không thể xuất hiện được.

**Định nghĩa 1:** Nếu tập tín hiệu vào (X) thay đổi từ Q sang P thì được gọi là có sự chuyển đổi từ Q sang P ( $Q \rightarrow P$ ).

**Định nghĩa 2:** Hazard nhất thời xuất hiện trong mạch logic tổ hợp là hiện tượng tín hiệu ra ở một hoặc nhiều đầu ra của mạch xuất hiện khác với các giá trị quy định cho chúng theo hàm Boole trong thời gian chuyển đổi từ  $Q \rightarrow P$ .

**Định nghĩa 3:** Hazard nhất thời xuất hiện trong mạch logic tổ hợp trong thời gian chuyển đổi từ  $Q \rightarrow P$  gọi là *hazard tĩnh* nếu và chỉ nếu  $f(Q) = f(P)$ . Ở đây  $f(X)$  là hàm logic được thực hiện bởi các mạch đã cho.

**Định nghĩa 4:** Hazard nhất thời xuất hiện trong mạch logic tổ hợp trong thời gian chuyển đổi từ  $Q \rightarrow P$  gọi là *hazard động* nếu và chỉ nếu  $f(Q) = \overline{f(P)}$ . Như vậy khi có hazard nhất thời

động thì tín hiệu ở đầu ra thay đổi ít nhất ba lần, ví dụ  $1 \rightarrow 0 \rightarrow 1 \rightarrow 0$ , nghĩa là có ít nhất hai xung nhiễu xuất hiện. Loại hazard này thường xảy ra trong các mạch tổ hợp.

**Định nghĩa 5:** Hazard nhất thời gọi là *hazard hàm số* trong thời gian chuyển đổi từ  $Q \rightarrow P$  nếu:

-  $f(Q) = f(P)$

- Hàm  $f(X)$  lấy cả hai giá trị 1 và 0 trong thời gian chuyển đổi từ  $Q \rightarrow P$

**Định nghĩa 6:** Hazard nhất thời gọi là *hazard logic* trong thời gian chuyển đổi từ  $Q \rightarrow P$  nếu:

-  $f(Q) \neq f(P)$

- Hàm  $f(X)$  chỉ nhận một giá trị như nhau (hoặc 0 hoặc 1)

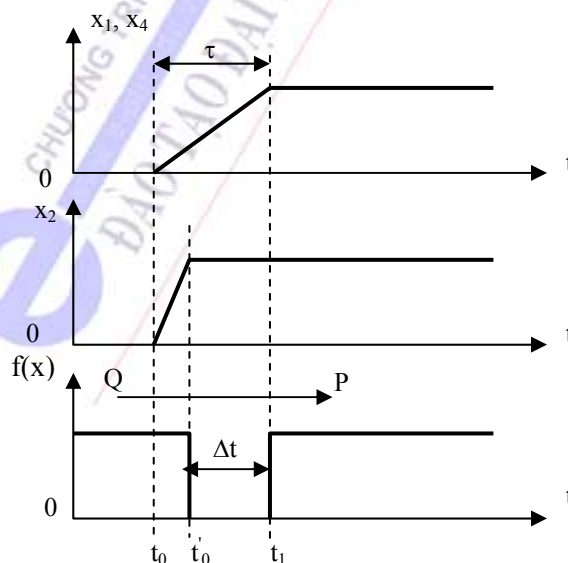
- Trong thời gian chuyển đổi từ  $Q \rightarrow P$  xuất hiện một xung hazard ở đầu ra.

#### 4.4.3.1. Hazard tĩnh trong mạch logic.

Do có hiện tượng "chạy đua" giữa các tín hiệu vào với nhau trong thời gian chuyển từ  $Q \rightarrow P$  mà xuất hiện hazard. Nếu  $f(Q) = f(P)$  tức là có sự thay đổi của tín hiệu vào nhưng sự điều khiển ở đầu ra của mạch logic vẫn không đổi dù là 0 hay 1, nhưng xuất hiện hazard, khi số lượng tín hiệu chạy đua không nhiều, đó chính là hazard tĩnh.

Hazard nhất thời cũng chính là hazard tĩnh, tức là loại hazard chỉ xuất hiện như một xung không theo quy định của hàm logic. Hiện tượng này không nguy hiểm, vì độ rộng của xung hazard tĩnh  $\Delta t$  luôn nhỏ hơn thời gian trễ  $\tau$  của mạch, nên mạch logic vẫn hoạt động bình thường dù có xuất hiện hazard.

Nhưng hazard tĩnh nguy hiểm ở chỗ: nó có thể gây ra "sai nhầm" cho điều khiển của hệ thống logic khi giá trị độ rộng hazard ( $\Delta t$ ) đủ lớn, điều này sẽ xảy ra khi sự "chạy đua" của tín hiệu vào quá chênh lệch, nghĩa là có tín hiệu vào "chạy" quá nhanh còn tín hiệu khác lại "chạy" quá chậm, hiện tượng này được minh họa ở hình 4-6.



Hình 4-6. Chạy đua ở hazard tĩnh

Ta thấy  $x_2$  trong quá trình "chạy đua" (thay đổi giá trị logic) đã "chạy" nhanh hơn so với tín hiệu  $x_1, x_4$ , thể hiện ở hình vẽ độ dốc xung  $x_2$  lớn hơn, điều đó làm cho  $\Delta t$  của xung hazard tăng theo, khi đó xung hazard trở nên "nguy hiểm" hơn vì nó có thể kích lật chuyển một mạch điện tiếp sau hệ thống mạch logic, gây hiện tượng điều khiển "sai nhầm" trong mạch logic.

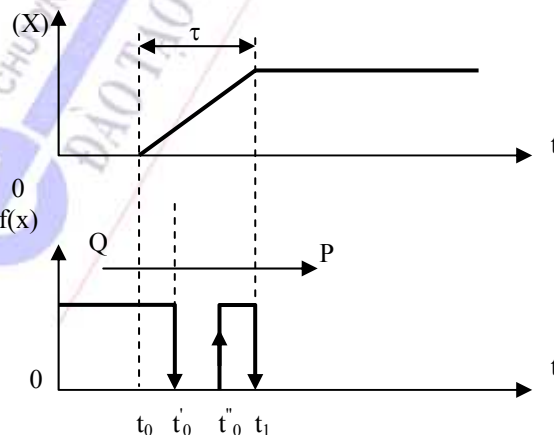
**4.4.3.2. Hazard động trong mạch logic.**

Trong thực tế khi thay đổi tín hiệu vào của mạch logic ứng với quá trình chuyển đổi ( $Q \rightarrow P$ ) có thể có rất nhiều tín hiệu vào cùng thay đổi khi đó có sự chạy đua của các tín hiệu vào tới đầu ra của mạch. Ví dụ trường hợp  $Q = (0000); P = (1101)$ , dễ dàng nhận thấy có sự chạy đua (X)

(X)	$x_1$	$x_2$	$x_3$	$x_4$	
$t_0$	0	0	0	0	$f(Q) = 1$
$t'_0$	0	1	0	0	$f(X') = 0$
$t''_0$	1	1	0	0	$f(X'') = 1$
$t_1$	1	1	0	1	$f(P) = 0$

Do có nhiều tín hiệu vào đồng thời thay đổi giá trị logic từ 0 sang 1 và từ 1 về 0 mà mỗi tín hiệu vào có tốc độ "chạy" khác nhau nên vô tình làm cho giá trị hàm  $f(X)$  ở đầu ra thay đổi như ở hình bên. Hiện tượng tín hiệu ra  $f(X)$  thay đổi giá trị từ  $1 \rightarrow 0 \rightarrow 1 \rightarrow 0$  được gọi là hazard động, tức là xuất hiện nhiều xung không cần thiết trong khoảng thời gian trễ của mạch ( $\tau$ ). Như vậy trong thời gian rất nhỏ  $\tau$  xuất hiện rất nhiều xung hazard nhỏ hơn  $\tau$  thì ta có thể hiểu là xung hazard động không có gì nguy hiểm cả, vì một xung bị chia ra nhiều xung con thì năng lượng còn rất nhỏ và độ rộng xung quá bé nên không đủ kích mạch khác được. Hiện tượng này ta có thể hiểu là khi đèn đang sáng ta cho tín hiệu thay đổi để đèn tắt nhưng do có hiện tượng chạy đua nên sau khi đèn tắt thì lại hơi sáng lên rồi mới tắt hẳn.

Hazard động ít có khả năng gây ra điều khiển "sai nhầm" trong mạch logic tổ hợp.



Hình 4-7. Hazard động

**4.4.3.3. Hazard hàm số trong mạch logic.**

Hazard có thể xuất hiện do chức năng của mạch trong cả hai trường hợp là hàm  $f(X)$  lấy giá trị logic là 0 hoặc 1.

Hazard nhất thời gọi là *hazard hàm số* trong thời gian chuyển đổi từ  $Q \rightarrow P$  nếu:

-  $f(Q)=f(P)$

- Hàm  $f(X)$  lấy cả hai giá trị 1 và 0 trong thời gian chuyển đổi từ  $Q \rightarrow P$

Điều này có nghĩa là trong thời gian chuyển đổi  $Q \rightarrow P$  thì hàm logic không thay đổi giá trị ( $f(Q)=f(P)$ ), nhưng nếu lấy  $f(Q)=f(P) = 0$  thì hazard vẫn xuất hiện hoặc lấy  $f(Q)=f(P)=1$  thì hazard vẫn xảy ra. Hiện tượng này được gọi là hazard hàm số. Trên thực tế có những hàm số hazard nhất thời chỉ xuất hiện khi điều khiển logic là 1 ( $f(X) = 1$ ) còn điều khiển logic ở đầu ra là 0 thì không có hazard nhất thời xuất hiện và ngược lại có thể điều khiển ra không bị hazard.

Độ nguy hiểm của hazard hàm số cũng giống như hazard tĩnh, nhưng nó nguy hiểm hơn một mức nữa vì bất kỳ quá trình điều khiển nào (0 hay 1) đều có khả năng xuất hiện hazard, tức là điều có khả năng gây ra "sai nhầm" khi điều khiển mạch.

**4.4.3.4. Hazard logic trong mạch logic.**

Đây là loại hazard nguy hiểm nhất, hay gây ra điều khiển "sai nhầm" nhiều nhất trong các hệ thống mạch tổ hợp điều khiển.

Bản chất của loại hazard này như sau:

Khi tập tín hiệu vào của hàm logic thay đổi đồng thời nhiều biến trong thời gian chuyển đổi  $Q \rightarrow P$ , mà mỗi một lần tín hiệu vào có thời trễ khác nhau, trong quá trình "chạy đua" này gặp phải trường hợp  $Q = (00000)$ ,  $P = (11101)$

(X)	—	( $x_1$	$x_2$	$x_3$	$x_4$	$x_5$ )	
$t_0$	—	0	0	0	0	0	$f(Q) = 1$
				↓			↓
$t'_0$	—	0	0	1	0	0	$f(X') = 0$
				↓			↓
$t''_0$	—	0	1	1	0	0	$f(X'') = 0$
					↓		↓
$t'''_0$	—	0	1	1	0	1	$f(X''') = 0$
				↓			↓
$t_1$	—	1	1	1	0	1	$f(P) = 1$

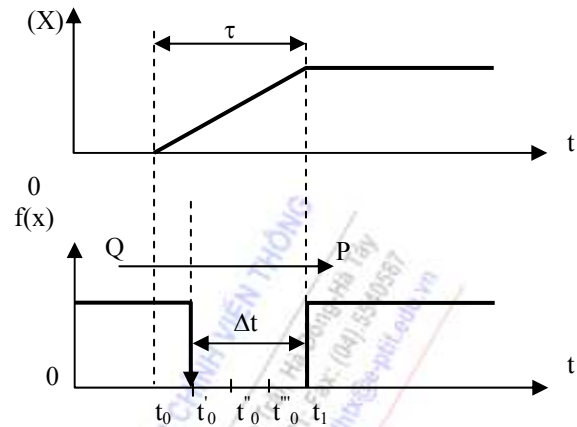
Hiện tượng hazard logic được mô tả trên hình 4-8:

Hazard nhất thời gọi là *hazard logic* trong thời gian chuyển đổi từ  $Q \rightarrow P$  nếu:

$$- f(Q) = f(P)$$

- Hàm  $f(X)$  chỉ nhận một giá trị như nhau (hoặc 0 hoặc 1)

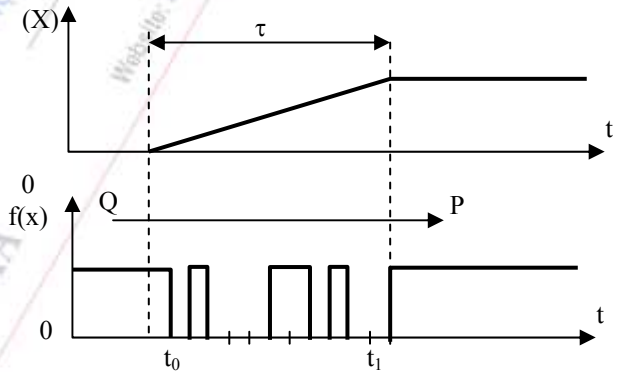
- Trong thời gian chuyển đổi từ  $Q \rightarrow P$  xuất hiện một xung hazard có độ rộng  $\Delta t$  lớn ở đầu ra, khi quá trình chạy đua ngẫu nhiên của các tín hiệu vào tạo ra hàm  $f(X)$  có cùng một giá trị logic.



Hình 4-8. Hazard logic

Như vậy trong quá trình chuyển đổi từ  $Q \rightarrow P$  của tập tín hiệu vào, có nhiều tín hiệu cùng thay đổi giá trị và hàm logic vô tình hay ngẫu nhiên xảy ra trường hợp có cùng một giá trị logic hazard ở đầu ra  $f(X)$  của mạch. Điều đó tạo nên một xung hazard ở đầu ra của của mạch độ rộng  $\Delta t$  lớn lên rất nhiều, khi  $\Delta t$  lớn làm cho xung hazard có năng lượng lớn đủ khả năng kích chuyển một mạch tiếp theo sau mạch điều khiển, điều đó gây ra hiện tượng điều khiển "sai nhầm" trong hệ thống logic tổ hợp. Đây là điều vô cùng nguy hiểm đối với các hệ thống tổ hợp cỡ lớn có nhiều đầu vào.

Trên thực tế quá trình chuyển đổi từ  $Q \rightarrow P$  trong mạch logic tổ hợp rất phức tạp, rất ít khi gặp từng loại hazard riêng biệt mà gặp sự tổ hợp hỗn loạn các loại hazard trên. Hiện tượng này được minh họa bằng hình 4-9.

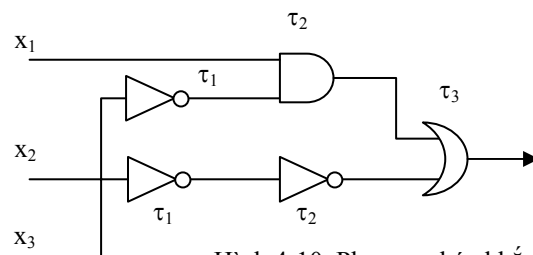


Hình 4-9. Hiện tượng tổng quát xuất hiện Hazard

Tóm lại, mỗi một mạch điều khiển có thể xuất hiện nhiều loại hazard, có mạch logic có số lượng biến số "chạy đua" rất lớn nhưng hazard lại không xuất hiện, nhưng có mạch rất đơn giản thì hazard lại xuất hiện và gây ra điều khiển "sai nhầm". Vì vậy muốn khắc phục được hazard thì phải căn cứ vào mạch điện cụ thể của nó, rồi dùng kỹ thuật phân tích phát hiện khả năng xuất hiện hazard, sau đó tìm cách khắc phục hazard. Sau đây là một vài biện pháp khắc phục và hạn chế sự xuất hiện hazard trong hệ thống logic điều khiển.

#### 4.4.4. Các biện pháp khắc phục Hazard.

Như đã phân tích ở trên, hazard xuất hiện do có sự chạy đua tín hiệu vào trong hệ logic tổ hợp, nói cách khác hazard xuất hiện là do sự khác nhau về thời gian trễ truyền lan từ đầu vào đến đầu ra của mạch, từ đó ta có những biện pháp khắc phục hazard như sau:



Hình 4-10. Phương pháp khắc phục Hazard

- Biện pháp đơn giản nhất làm biến mất hazard là không để xuất hiện quá trình chạy đua của các tín hiệu vào trong mạch logic, nghĩa là chỉ thay đổi giá trị logic trên một đầu vào tín hiệu. Khi chỉ có một tín hiệu vào "chạy" trong mạch logic thì sẽ không còn "đua" tín hiệu nữa và chắc chắn hazard không thể xuất hiện. Nhưng như vậy cũng có nghĩa là từng tín hiệu vào thay đổi giá trị logic sẽ làm cho mạch hoạt động chậm chạp, và không phải quá trình điều khiển nào cũng cho phép làm như vậy, thông thường có sự thay đổi nhiều tín hiệu vào cùng một lúc.

- Tiếp theo khi phải chấp nhận quá trình chuyển đổi từ  $Q \rightarrow P$  có nhiều tín hiệu thay đổi hay có nhiều biến (X) chạy đua. Cách khắc phục là chọn giá trị linh kiện hay IC có thời gian trễ  $\tau$  nhỏ. Vì ta biết hazard chỉ xuất hiện trong thời gian trễ của mạch,  $\tau$  càng nhỏ nghĩa là xung hazard có độ rộng  $\Delta t$  nhỏ, và như vậy nó không có đủ năng lượng để kích chuyển mạch tiếp theo. Nhưng khi chọn linh kiện lắp ráp hệ thống hay chọn IC có  $\tau$  nhỏ tức là phải chọn linh kiện, IC có chất lượng cao, nghĩa là giá thành của hệ điều hành tăng, đây cũng là vấn đề cần quan tâm khi thiết mạch.

- Khi ta chấp nhận có sự chạy đua tín hiệu vào (X) trong quá trình chuyển đổi từ  $Q \rightarrow P$ , đồng thời không dùng linh kiện có chất lượng cao để giảm giá thành và mạch vẫn hoạt động tốt đồng thời không có hazard xuất hiện, thì ta có thể dùng phương pháp khắc phục hazard bằng cách thêm các mạch trễ trên đường truyền tín hiệu, để đảm bảo cho thời gian chạy đua của các tín hiệu là tương đương nhau. Phương pháp này được minh họa ở hình 4-10:

Ta biết tín hiệu  $x_2$  chạy nhanh tới đầu ra, nên trên đường truyền của  $x_2$  ta cho thêm hai công đảo có thời gian trễ là  $\tau_1$  và  $\tau_2$  để cho tín hiệu trên  $x_2$  xuất hiện đồng thời với  $x_1$  và  $x_3$ , khi đó hazard sẽ không xuất hiện hoặc sẽ làm giảm bớt hazard. Phương pháp này có gây ra hazard nếu đường trễ thêm vào lại làm cho  $x_2$  chạy quá chậm và lại phát sinh hiện tượng chạy đua tín hiệu vào.

Để tránh xảy ra hiện tượng chạy đua tín hiệu vào, cần biết chính xác thời gian trễ  $\tau_1$  và  $\tau_2$ , sau đó phải tạo ra được công đảo có thời gian trễ bằng đúng giá trị  $\tau_1$  và  $\tau_2$ .

- Ở mức cao hơn khi ta phải chấp nhận có sự chạy đua tín hiệu vào trong quá trình chuyển đổi  $Q \rightarrow P$ , không muốn dùng linh kiện có chất lượng cao, đồng thời đã thêm các mạch trễ (không ảnh hưởng tới chức năng của mạch logic) nhưng vẫn không thể khắc phục hết hazard thì khi đó ta dùng xung đồng bộ, tức là ta bắt chấp có sự chạy đua của tín hiệu vào, và giữa các đường truyền tín hiệu từ đầu vào tới đầu ra có thời gian trễ khác nhau. Nhưng tín hiệu truyền lan trong hệ logic dù nhanh, dù chậm, đến trước hay đến sau thì chúng chỉ được lan truyền khi có sự cho phép của xung đồng bộ. Xung đồng bộ thông thường "chờ" theo đường tín hiệu chạy chậm nhất, khi đó các xung đến sớm phải "chờ" cho đầy đủ các tín hiệu khác khi đó xung đồng bộ mới cho phép truyền tiếp. Nếu cho thêm vào mạch điều khiển xung đồng bộ thì cũng có thể giảm đáng kể ảnh hưởng của hazard.

- Trong trường hợp các phương pháp nêu trên đều được áp dụng nhưng hiện tượng hazard vẫn xuất hiện thì ta buộc phải thay đổi chức năng điều khiển, tức là thay đổi chức năng của hàm logic của hệ thống điều khiển tức là phải xây dựng mạch điện khác.

Như vậy để có được một mạch điều khiển tốt, chất lượng cao thì phần cứng xây dựng nên mạch điện mang tính quyết định. Người thiết kế phải hiểu rất kỹ và sâu sắc hệ thống kỹ thuật mà mình thiết kế thì mới có thể khắc phục được hazard trong mạch điện, cũng như phải biết thêm hay bớt các mạch điện phụ như thế nào mà không làm thay đổi chức năng của hệ thống. Từ đó làm

cho mạch có chất lượng cao hơn, giá trị kinh tế cũng cao hơn. Điều này cũng dễ hiểu là các mạch điện có cùng chức năng điều khiển nhưng mỗi hãng sản xuất lại đưa ra một mạch khác nhau và giá trị kinh tế của chúng cũng khác nhau, tùy thuộc vào trình độ và sự quan tâm đến việc tăng độ tin cậy, tăng chất lượng điều khiển mạch của hãng. Nhưng bản chất vẫn chỉ là làm giảm tối đa khả năng xuất hiện hazard trong mạch.

## 4.5. MẠCH MÃ HOÁ VÀ GIẢI MÃ

### 4.5.1. Một số loại mã thông dụng.

#### 4.5.1.1. Mã BCD và mã dư 3.

MÃ BCD (Binary Coded Decimal) là mã được cấu tạo bằng cách dùng từ nhị phân 4 bit để mã hóa 10 kí hiệu thập phân, nhưng cách biểu diễn vẫn theo thập phân. Ví dụ đối với mã NBCD, các chữ số thập phân được nhị phân hoá theo trọng số như nhau  $2^3, 2^2, 2^1, 2^0$  nên có 6 tổ hợp dư, ứng với các số thập phân 10,11,12,13,14 và 15. Sự xuất hiện các tổ hợp này trong bản tin được gọi là lỗi dư.

Do trọng số nhị phân của mỗi vị trí biểu diễn thập phân là tự nhiên nên máy có thể thực hiện trực tiếp các phép tính cộng, trừ, nhân, chia theo mã NBCD. Tuy nhiên nhược điểm chính của mã là tồn tại tổ hợp toàn Zero, gây khó khăn trong việc đồng bộ khi truyền dẫn tín hiệu.

Vì vậy, người ta sử dụng mã Dư-3 được hình thành từ mã NBCD bằng cách cộng thêm 3 vào mỗi tổ hợp mã. Như vậy, mã không bao gồm tổ hợp toàn Zero. Mã Dư-3 chủ yếu được dùng để truyền dẫn tín hiệu mà không dùng cho việc tính toán trực tiếp.

Thập phân	BCD 8421	Mã dư 3
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

Bảng 4-2. Mã BCD 8421 và mã dư 3

#### 4.5.1.2. Mã Gray.

Mã Gray còn được gọi là mã cách 1, là loại mã mà các tổ hợp mã kế nhau chỉ khác nhau duy nhất 1 bit. Loại mã này không có tính trọng số. Do đó, giá trị thập phân đã được mã hóa chỉ được giải mã thông qua bảng mã mà không thể tính theo tổng trọng số như đối với mã BCD.

Mã Gray có thể được tổ chức theo nhiều bit. Bởi vậy, có thể đếm theo mã Gray.

Cũng tương tự như mã BCD, ngoài mã Gray chính còn có mã Gray dư-3.

Thập phân	Gray	Gray dư 3
0	0000	0010
1	0001	0110
2	0011	0111
3	0010	0101
4	0110	0100
5	0111	1100
6	0101	1101
7	0100	1111
8	1100	1110
9	1101	1010
10	1111	1011
11	1110	1001
12	1010	1000
13	1011	0000
14	1001	0001
15	1000	0011

Bảng 4-3. Mã Gray và Gray dư 3

#### 4.5.1.3. Mã chẵn, lẻ.

Mã chẵn và mã lẻ là hai loại mã có khả năng phát hiện lỗi hay dùng nhất. Để thiết lập loại mã này ta chỉ cần thêm một bit chẵn/ lẻ (bit parity) vào tổ hợp mã đã cho, nếu tổng số bit 1 trong từ mã (bit tin tức + bit chẵn/lẻ) là chẵn thì ta được mã chẵn và ngược lại ta được mã lẻ.

BCD 8421	BCD 8421 chẵn		BCD 8421 lẻ
		P <sub>C</sub>	P <sub>L</sub>
0000	0000	0	0000 1
0001	0001	1	0001 0
0010	0010	1	0010 0
0011	0011	0	0011 1
0100	0100	1	0100 0
0101	0101	0	0101 1
0110	0110	0	0110 1
0111	0111	1	0111 0
1000	1000	1	1000 0
1001	1001	0	1001 1

Bảng 4-4. Mã BCD 8421 chẵn / lẻ

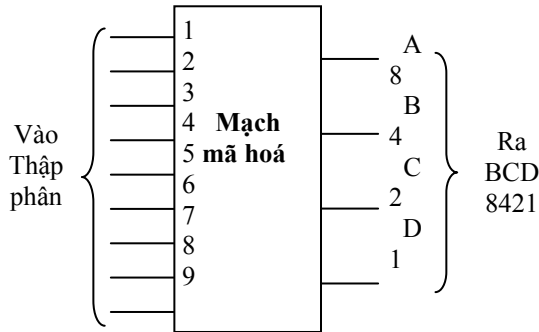
#### 4.5.2. Mạch mã hoá.

Mạch điện thực hiện việc chuyển tin tức sang mã, được gọi là mạch mã hoá hay mạch ghi mã.

##### 4.5.1.1. Mạch mã hoá từ thập phân sang BCD 8421

Sơ đồ khối tổng quát của mạch Mã hoá như hình 4-7. Mạch gồm 9 lối vào (biến) ứng Với các chữ số thập phân từ 1 đến 9. Lối vào zero là không cần thiết, vì khi tất cả các lối vào khác bằng 0 thì lối ra cũng bằng 0. Bốn lối ra A, B, C, D (hàm) thể hiện tổ hợp mã tương ứng với mỗi chữ số thập phân trên lối vào theo trọng số 8421. Bảng trạng thái của mạch như bảng 4-5.





Hình 4-11 Sơ đồ khối của mạch mã hoá

Vào thập phân	Ra BCD			
	8	4	2	1
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

Bảng 4-5. Bảng trạng thái của mạch mã hoá.

Từ bảng trạng thái ta viết được các hàm ra như sau:

$$A = 8 + 9 = \Sigma (8,9)$$

$$B = 4 + 5 + 6 + 7 = \Sigma (4,5,6,7)$$

$$C = 2 + 3 + 6 + 7 = \Sigma (2,3,6,7)$$

$$D = 1 + 3 + 5 + 7 + 9 = \Sigma (1,3,5,7,9)$$

Căn cứ hệ phương trình, ta xây dựng được mạch điện của bộ mã hoá. Hoặc dùng ma trận diode (cổng OR) để xây dựng

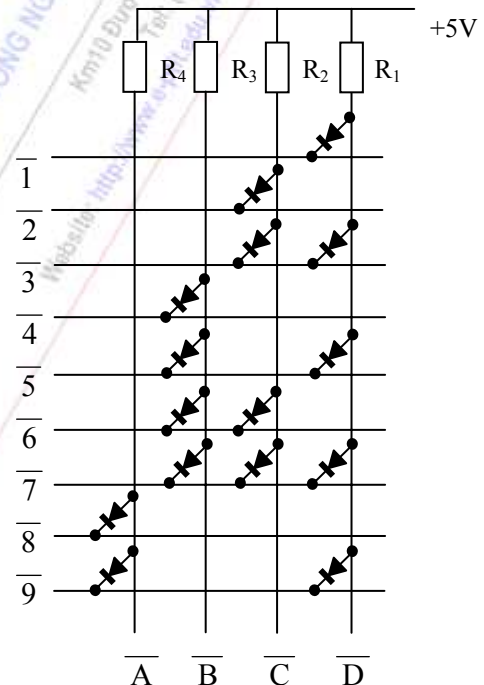
Hoặc có thể được viết lại như sau (dùng định lý DeMorgan) và dùng ma trận diode (cổng AND) để xây dựng mạch:

$$\overline{A} = \overline{8+9} = \overline{8} \cdot \overline{9}$$

$$\overline{B} = \overline{4+5+6+7} = \overline{4} \cdot \overline{5} \cdot \overline{6} \cdot \overline{7}$$

$$\overline{C} = \overline{2+3+6+7} = \overline{2} \cdot \overline{3} \cdot \overline{6} \cdot \overline{7}$$

$$\overline{D} = \overline{1+3+5+7+9} = \overline{1} \cdot \overline{3} \cdot \overline{5} \cdot \overline{7} \cdot \overline{9}$$



Hình 4-12 Mạch điện của bộ mã hoá dùng diode.

#### 4.5.1.2. Mạch mã hoá ưu tiên

Trong bộ mã hoá vừa xét trên, tín hiệu vào tồn tại độc lập, (không có trường hợp có 2 tổ hợp trở lên đồng thời tác động). Bộ mã hoá ưu tiên ra đời để giải quyết trường hợp có nhiều đầu vào tác động đồng thời. Đối với các trường hợp này thì bộ mã hoá ưu tiên chỉ tiến hành mã hoá tín hiệu vào nào có cấp ưu tiên cao nhất ở thời điểm xét. Việc xác định cấp ưu tiên cho mỗi tín hiệu vào là do người thiết kế mạch.

Bây giờ ta xét nguyên tắc hoạt động và quá trình thiết kế của bộ mã hoá ưu tiên 9 lỗi vào, 4 lỗi ra.

Vào Thập phân L <sub>1</sub> L <sub>2</sub> L <sub>3</sub> L <sub>4</sub> L <sub>5</sub> L <sub>6</sub> L <sub>7</sub> L <sub>8</sub> L <sub>9</sub>	Ra			
	A	B	C	D
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
x 1 0 0 0 0 0 0 0	0	0	1	0
x x 1 0 0 0 0 0 0	0	0	1	1
x x x 1 0 0 0 0 0	0	1	0	0
x x x x 1 0 0 0 0	0	1	0	1
x x x x x 1 0 0 0	0	1	1	0
x x x x x x 1 0 0	0	1	1	1
x x x x x x x 1 0	1	0	0	0
x x x x x x x x 1	1	0	0	1

Bảng 4-6. Bảng trạng thái của bộ mã hoá ưu tiên

Theo đề bài, sự mã hoá thực hiện theo mức độ ưu tiên từ L<sub>1</sub> đến L<sub>9</sub>, khi các tín hiệu cùng tác động thì các tín hiệu có mức ưu tiên thấp không tác dụng, nghĩa là bất kể mức logic của nó là 0 hay 1 đều không ảnh hưởng đến lối ra nên gọi nó là điều kiện tùy chọn, ký hiệu là "x".

Bảng trạng thái phản ánh yêu cầu thiết kế, mã hoá theo cấp ưu tiên.

Từ bảng trạng thái ta có thể viết được biểu thức lối ra như sau:

D = 1 tại các lối: + L<sub>1</sub> và bằng 0 tại các lối L<sub>2</sub>, L<sub>4</sub>, L<sub>6</sub>, L<sub>8</sub>  
 + L<sub>3</sub> và bằng 0 tại các lối L<sub>4</sub>, L<sub>6</sub>, L<sub>8</sub>  
 + L<sub>5</sub> và bằng 0 tại các lối L<sub>6</sub>, L<sub>8</sub>  
 + L<sub>7</sub> và bằng 0 tại các lối L<sub>8</sub>  
 + L<sub>9</sub>

Nên ta viết được hàm D:

$$D = L_1 \cdot \overline{L_2} \cdot \overline{L_4} \cdot \overline{L_6} \cdot \overline{L_8} + L_3 \cdot \overline{L_4} \cdot \overline{L_6} \cdot \overline{L_8} + L_5 \cdot \overline{L_4} \cdot \overline{L_6} \cdot \overline{L_8} + L_7 \cdot \overline{L_8} + L_9$$

Tương tự như vậy ta viết được hàm của B, C và A như sau:

$$C = L_2 \cdot \overline{L_4} \cdot \overline{L_5} \cdot \overline{L_8} \cdot \overline{L_9} + L_3 \cdot \overline{L_4} \cdot \overline{L_5} \cdot \overline{L_8} \cdot \overline{L_9} + L_6 \cdot \overline{L_8} \cdot \overline{L_9} + L_7 \cdot \overline{L_8} \cdot \overline{L_9}$$

$$B = L_4 \cdot \overline{L_8} \cdot \overline{L_9} + L_5 \cdot \overline{L_8} \cdot \overline{L_9} + L_6 \cdot \overline{L_8} \cdot \overline{L_9} + L_7 \cdot \overline{L_8} \cdot \overline{L_9}$$

$$A = L_8 \cdot L_9$$

Một vài IC thường dùng: 74147 là bộ mã hoá ưu tiên NBCD 4 bit, 74148 là bộ mã hoá ưu tiên NBCD 3 bit.

#### 4.5.2. Bộ giải mã.

Mạch điện thực hiện việc chuyển từ mã sang tín tức được gọi là mạch giải mã hoá.

**4.5.2.1. Bộ giải mã nhị phân**

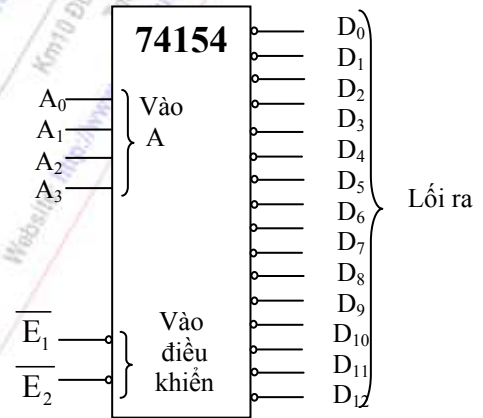
Bộ giải mã nhị phân còn có tên là bộ giải mã "1 từ n", bộ giải mã địa chỉ hoặc bộ chọn địa chỉ nhị phân. Chức năng của nó là lựa chọn duy nhất một lối ra (lấy giá trị 1 hoặc 0), khi tác động tới đầu vào một số nhị phân.

Như vậy, nếu số nhị phân là  $n$  bit ( $n$  lối vào) sẽ nhận diện được  $2^n$  địa chỉ khác nhau (trên  $2^n$  lối ra). Nói khác đi, mạch chọn địa chỉ nhị phân là một mạch logic tổ hợp có  $n$  lối vào và  $2^n$  lối ra, nếu tác động tới đầu vào một số nhị phân thì chỉ duy nhất một lối ra được lựa chọn, lấy giá trị 1 (tích cực cao) hoặc 0 (tích cực thấp), các lối ra còn lại đều không được lựa chọn, lấy giá trị 0 hoặc 1. Sơ đồ khối tổng quát của bộ chọn địa chỉ nhị phân như chỉ ở hình 4-13.



**Hình 4-13.** Sơ đồ khối của bộ giải mã nhị phân

IC 74154 là một bộ chọn địa chỉ nhị phân 4 vào 16 ra. Ký hiệu logic của nó được chỉ ra ở hình 4-14. Các lối vào  $E_1, E_2$ , hoạt động theo tích cực thấp thường được sử dụng để mở rộng dung lượng hoặc thay đổi chức năng logic của bộ chọn địa chỉ.



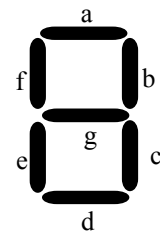
**Hình 4-14.** Ký hiệu logic của IC 74154

Ta có thể mở rộng dung lượng bộ chọn địa chỉ nhị phân bằng cách ghép các IC có dung lượng nhỏ lại với nhau.

**4.5.2.2. Mạch giải mã 7 đoạn**

**a) Dụng cụ 7 đoạn**

Để hiển thị chữ số của một hệ đếm phân bất kỳ, ta có thể dùng dụng cụ 7 đoạn. Cấu tạo của nó như chỉ ở hình 4-15. Các đoạn được hình thành bằng nhiều loại vật liệu khác nhau, nhưng phải có khả năng hiển thị được trong các điều kiện ánh sáng khác nhau và tốc độ chuyển mạch phải đủ lớn. Trong kỹ thuật số, các đoạn thường được dùng là LED hoặc tinh thể lỏng (LCD).



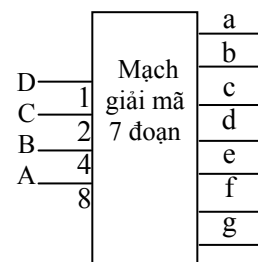
**Hình 4-15** Cấu tạo dụng cụ 7 đoạn sáng

Đối với LED, mỗi đoạn là một Diode phát quang và khi có dòng điện đi qua đủ lớn (5 đến 30 mA) thì đoạn tương ứng sẽ sáng.

Ngoài 7 đoạn sáng chính, mỗi LED cũng có thêm Diode để hiển thị dấu phân số khi cần thiết. LED có hai loại chính: LED Anốt chung và Ktốt chung. Do đó, logic của tín hiệu điều khiển hai loại này là ngược nhau.

**b) Mạch giải mã 7 đoạn**

Nhiệm vụ của ta là phải thiết kế một mạch logic liên hợp với 4 lối vào và 7 lối ra để chuyển mã NBCD thành mã 7 đoạn. Sơ đồ khối tổng quát của bộ giải mã như hình 4-16. Từ hình 4-15 để



**Hình 4-16** Sơ đồ khối của mạch giải mã 7 đoạn sáng

nhận thấy rằng, đoạn  $a$  sẽ sáng khi hiển thị chữ số : 0 hoặc 2, hoặc 3, hoặc 5, hoặc 7, hoặc 8, hoặc 9. Do đó, ta có thể viết:

$$a = \sum (0,2,3,5,6,7,8,9). \text{ Tương tự, ta có:}$$

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

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

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

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

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

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

IC 7447, 74247 (Anốt chung), 7448 (K chung ), 4511 (CMOS) là các IC giải mã từ NBCD sang thập phân theo phương pháp hiển thị 7 đoạn.

## 4.6 BỘ HỢP KÊNH VÀ PHÂN KÊNH

### 4.6.1 Bộ hợp kênh (MUX-Multiplexer)

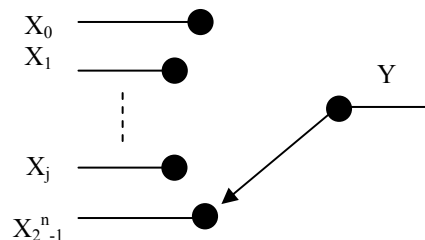
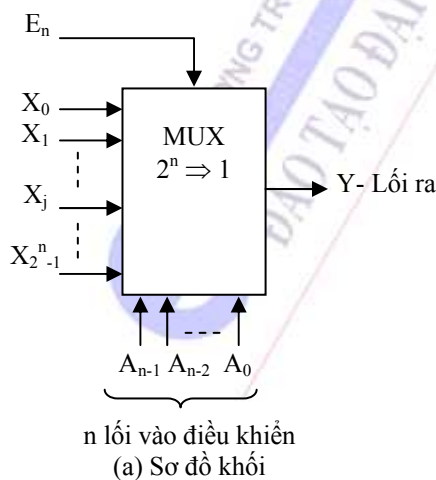
**Định nghĩa:** Bộ hợp kênh là mạch có  $2^n$  lối vào dữ liệu,  $n$  lối vào điều khiển, 1 lối vào chọn mạch và 1 lối ra.

Tùy theo giá trị của  $n$  lối vào điều khiển mà lối ra sẽ bằng một trong những giá trị ở lối vào ( $X_j$ ). Nếu giá trị thập phân của  $n$  lối vào điều khiển bằng  $j$  thì  $Y = X_j$ .

Sơ đồ khối của MUX  $2^n \Rightarrow 1$  ( $2^n$  lối vào, 1 lối ra) được biểu diễn ở hình 4-17a.

Phương trình tín hiệu ra là:

$$Y = X_0 (\overline{A_{n-1}} \overline{A_{n-2}} \dots \overline{A_1} \overline{A_0}) + X_1 (\overline{A_{n-1}} \overline{A_{n-2}} \dots \overline{A_1} A_0) + \dots + X_{2^n-1} (A_{n-1} A_{n-2} \dots A_1 A_0)$$



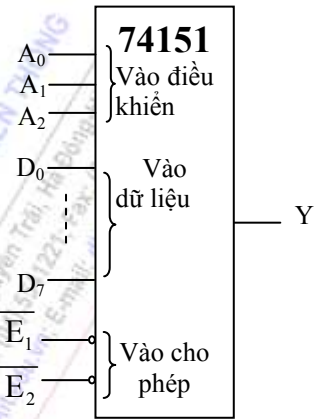
(b). MUX là một chuyển mạch điện tử

**Hình 4-17.** Bộ hợp kênh MUX  $2^n \Rightarrow 1$

Thực chất, MUX là chuyển mạch điện tử dùng các tín hiệu điều khiển ( $A_{n-1}A_{n-2}\dots A_0$ ) để điều khiển sự nối mạch của lối ra với 1 trong số  $2^n$  lối vào (hình 4-17b).

Hiện nay, bộ MUX được dùng như một phần tử vạn năng để xây dựng những mạch tổ hợp khác.

IC 74151 là bộ MUX 8 lối vào dữ liệu - 1 lối ra. Hình 4-18 là ký hiệu logic của IC 74151.



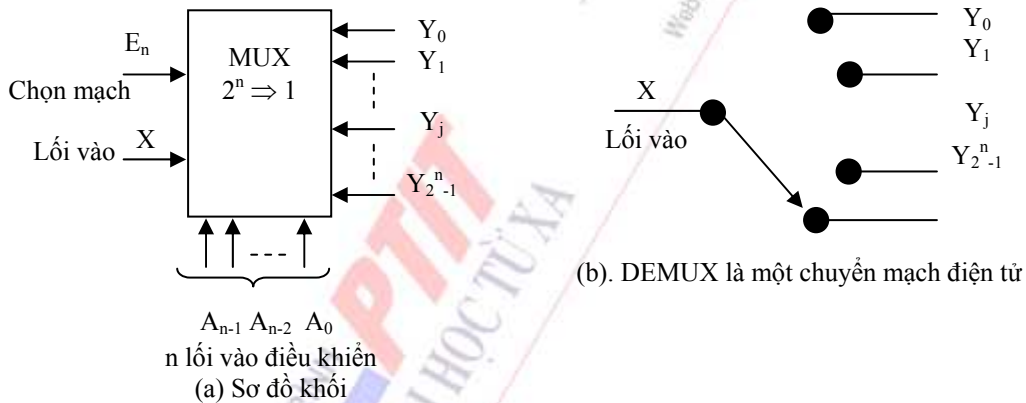
Hình 4-18. Ký hiệu logic của IC 74151

#### 4.6.2 Bộ phân kênh (DEMUX-DeMultiplexer)

**Định nghĩa:** Bộ phân kênh là mạch có 1 lối vào dữ liệu,  $n$  lối vào điều khiển, 1 lối vào chọn mạch và  $2^n$  lối ra.

Tùy theo giá trị của  $n$  lối vào điều khiển mà lối ra thứ  $i$  ( $Y_i$ ) sẽ bằng giá trị của lối vào. Cụ thể nếu gọi  $n$  lối vào điều khiển là  $A_{n-1}A_{n-2}\dots A_0$  thì  $Y_i = X$  khi  $(A_{n-1}A_{n-2}\dots A_1A_0)_2 = (i)_{10}$ .

Sơ đồ khối của bộ DEMUX 1 lối vào  $2^n$  lối ra được biểu diễn ở hình 4-19.



Hình 4-19. Bộ phân kênh DEMUX  $1 \Rightarrow 2^n$

Phương trình tín hiệu ra của DEMUX  $1 \Rightarrow 2^n$ :

$$Y_0 = X \cdot \bar{A}_{n-1} \bar{A}_{n-2} \dots \bar{A}_1 \bar{A}_0$$

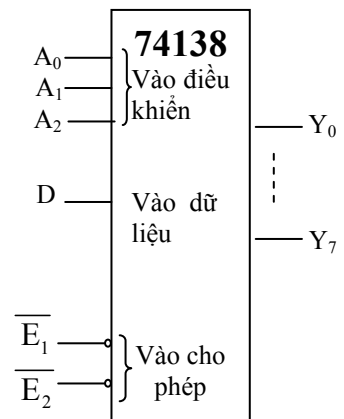
$$Y_1 = X \cdot \bar{A}_{n-1} \bar{A}_{n-2} \dots \bar{A}_i \dots \bar{A}_1 A_0$$

.....

$$Y_{2^n-1} = X \cdot A_{n-1} A_{n-2} \dots A_i \dots A_0$$

Bộ phân kênh còn được gọi là bộ giải mã 1 trong số  $2^n$ . Tại một thời điểm chỉ có 1 trong số  $2^n$  lối ra ở mức tích cực.

IC 74138 là bộ DEMUX 1 lối vào dữ liệu - 8 lối ra. Hình 4-20 là ký hiệu logic của IC 74138.



Hình 4-20. Ký hiệu logic của IC 74138

### 4.7. MẠCH CỘNG.

#### 4.7.1. Mạch toàn tổng.

Mạch cộng hay (bộ cộng) là mạch số học nhị phân quan trọng, vì trong xử lý nhị phân phần lớn các phép tính được thực hiện thông qua phép cộng.

Mạch logic thực hiện phép cộng hai số nhị phân 1 bit có lối nhớ đầu vào được gọi là mạch toàn tổng. Sơ đồ khối tổng quát của một mạch toàn tổng được biểu diễn ở hình 4-21.

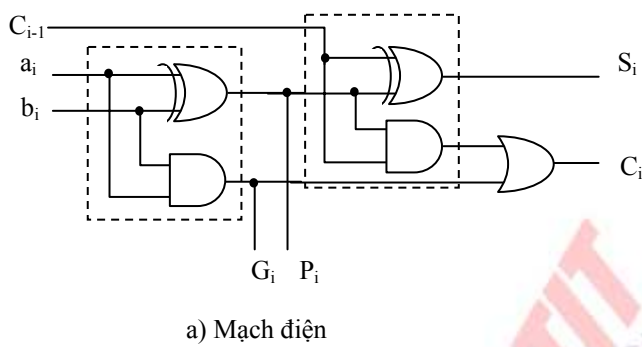
Theo hình 4-21 và nguyên lý cộng hai số nhị phân một bit có trọng số bất kỳ, ta có thể lập bảng trạng thái cho mạch toàn tổng.

Các hàm ra  $S_i, C_i$  sẽ có dạng:

$$S_i = a_i \oplus b_i \oplus C_{i-1}$$

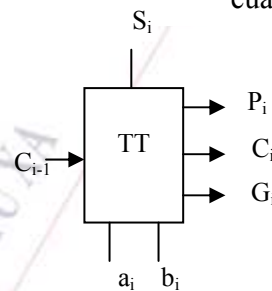
$$C_i = a_i b_i C_{i-1} + \overline{a_i} b_i C_{i-1} + a_i \overline{b_i} C_{i-1} \quad \text{hay}$$

$$C_i = a_i b_i + (a_i \oplus b_i) C_{i-1}$$



$C_{i-1}$	$a_i$	$b_i$	$S_i$	$C_i$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Bảng 4-7. Bảng trạng thái của mạch toàn tổng.

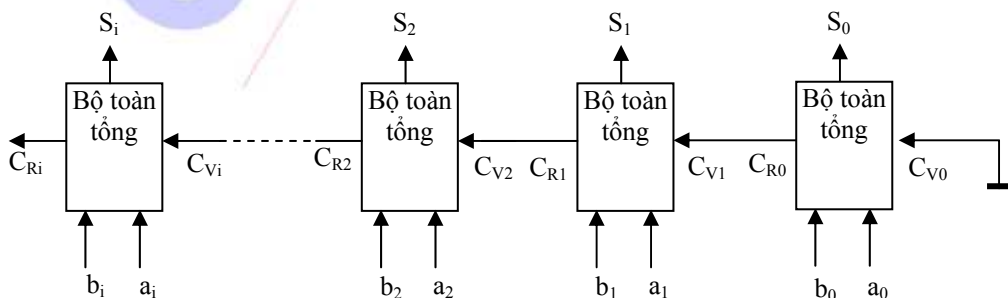


Hình 4-21 a, b Mạch toàn tổng và ký hiệu

Mạch logic thực hiện biểu thức lối ra tổng và lối ra nhớ được trình bày ở hình 4-21a và ký hiệu của nó là hình 4-21b.

#### 4.7.2 Mạch cộng nhị phân song song

Ta có thể ghép nhiều bộ cộng hai số nhị phân một bit lại với nhau để thực hiện phép cộng hai số nhị phân nhiều bit. Sơ đồ khối của bộ cộng được trình bày ở hình 4-22 và được gọi là bộ cộng song song.



Hình 4-22 Sơ đồ khối của bộ cộng nhị phân song song

Để giảm bớt mức độ phức tạp của mạch, trong thực tế người ta thường sản xuất bộ tổng 4 bit. Muốn cộng nhiều bit, có thể hợp nối tiếp một vài bộ tổng một bit theo phương pháp nêu trên.

Một trong những bộ cộng thông dụng hiện nay là 7483. IC này được sản xuất theo hai loại: 7483 và 7483A với logic vào, ra khác nhau.

#### 4.8. MẠCH SO SÁNH.

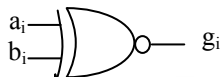
Trong các hệ thống số, đặc biệt là trong máy tính, thường thực hiện việc so sánh hai số. Hai số cần so sánh có thể là các số nhị phân, có thể là các ký tự đã mã hoá nhị phân. Mạch so sánh có thể hoạt động theo kiểu nối tiếp hoặc theo kiểu song song. Trong phần này ta sẽ nghiên cứu bộ so sánh theo kiểu song song.

##### 4.8.1. Bộ so sánh bằng nhau.

###### 4.8.1.1. Bộ so sánh bằng nhau 1 bit.

Xét 2 bit  $a_i$  và  $b_i$ , gọi  $g_i$  là kết quả so sánh. Từ đó là có bảng trạng thái 4-8.

$$g_i = \overline{a_i} \cdot \overline{b_i} + a_i \cdot b_i = a_i \oplus b_i$$



Hình 4-23. Sơ đồ logic hàm ra của bộ so sánh bằng 1 bit

$a_i$	$b_i$	$g_i$
0	0	1
0	1	0
1	0	0
1	1	1

Bảng 4-8. Bảng trạng thái của mạch so sánh bằng.

###### 4.8.1.2. Bộ so sánh bằng nhau 4 bit.

So sánh hai số nhị phân 4 bit  $A = a_3a_2a_1a_0$  với  $B = b_3b_2b_1b_0$ . Vậy hai số A và B bằng nhau khi  $a_3 = b_3, a_2 = b_2, a_1 = b_1, a_0 = b_0$ .

Biểu thức đầu ra tương ứng là:

$$G = g_3 \cdot g_2 \cdot g_1 \cdot g_0$$

$$g_3 = a_3 \oplus b_3$$

$$\text{với } g_2 = a_2 \oplus b_2$$

$$g_1 = a_1 \oplus b_1$$

$$g_0 = a_0 \oplus b_0$$

##### 4.8.2. Bộ so sánh.

###### 4.8.2.1. Bộ so sánh 1 bit.

Từ bảng trạng thái 4-9 ta có biểu thức ra:

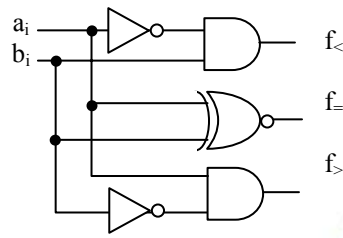
$$f_{<} = \overline{a_i} \cdot b_i$$

$$f_{=} = a_i \oplus b_i$$

$$f_{>} = a_i \cdot \overline{b_i}$$

$a_i$	$b_i$	$f_<$	$f_=<$	$f_>$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

Bảng 4-9. Bảng trạng thái của mạch so sánh.



Hình 4-24. Mạch điện của bộ so sánh 1 bit

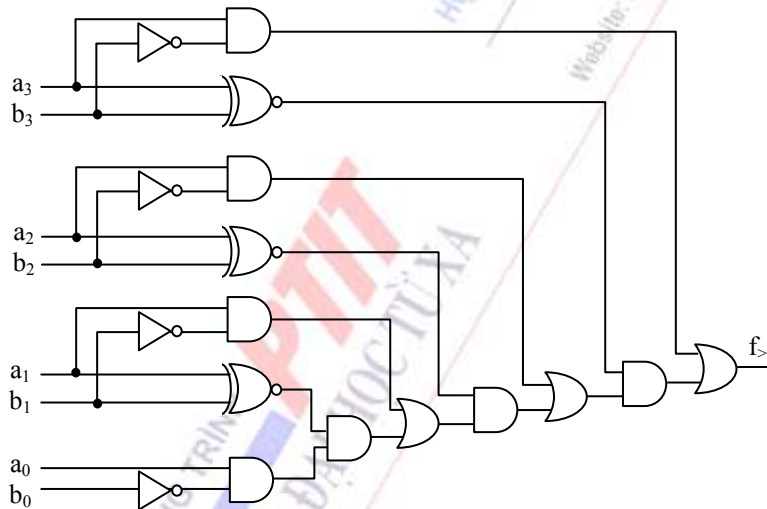
#### 4.8.2.2. Bộ so sánh 4 bit (So sánh lớn hơn).

So sánh hai số nhị phân 4 bit  $A = a_3a_2a_1a_0$  với  $B = b_3b_2b_1b_0$ . Số A lớn hơn số B khi:

$a_3 > b_3$  hoặc  $a_3 = b_3$  và  $a_2 > b_2$  hoặc  $a_3 = b_3$  và  $a_2 = b_2$  và  $a_1 > b_1$  hoặc  $a_3 = b_3$  và  $a_2 = b_2$  và  $a_1 = b_1$  và  $a_0 > b_0$ .

Từ đó ta có biểu thức hàm ra là:

$$f_> = a_3 \cdot \overline{b_3} + a_3 \oplus b_3 \cdot a_2 \cdot \overline{b_2} + a_3 \oplus b_3 \cdot a_2 \oplus b_2 \cdot a_1 \cdot \overline{b_1} + a_3 \oplus b_3 \cdot a_2 \oplus b_2 \cdot a_1 \oplus b_1 \cdot a_0 \cdot \overline{b_0}$$



Hình 4-26. Mạch điện của bộ so sánh lớn hơn 4 bit

Một trong những bộ so sánh thông dụng hiện nay là 7485. IC này so sánh 2 số nhị phân 4 bit.

#### 4.9. MẠCH TẠO VÀ KIỂM TRA CHẴN LẺ.

Có nhiều phương pháp mã hoá dữ liệu để phát hiện lỗi và sửa lỗi khi truyền dữ liệu từ nơi này sang nơi khác. Phương pháp đơn giản nhất là thêm một bit vào dữ liệu được truyền đi sao cho số chữ số 1 trong dữ liệu luôn là chẵn hoặc lẻ. Bit thêm vào đó được gọi là bit chẵn/lẻ.

Để thực hiện được việc truyền dữ liệu theo kiểu đưa thêm bit chẵn, lẻ vào dữ liệu chúng ta phải:

- Xây dựng sơ đồ tạo được bit chẵn, lẻ để thêm vào  $n$  bit dữ liệu.



- Xây dựng sơ đồ kiểm tra hệ xem đó là hệ chẵn hay lẻ với  $(n + 1)$  bit ở đầu vào ( $n$  bit dữ liệu, 1 bit chẵn/lẻ).

**4.9.1. Mạch tạo bit chẵn/lẻ.**

Gọi 3 bit của dữ liệu là  $d_1, d_2, d_3$  và  $X_e, X_o$  là 2 bit chẵn, lẻ thêm vào dữ liệu. Từ đó lập được bảng trạng thái 4-10:

Vào			Ra	
$d_1$	$d_2$	$d_3$	$X_e$	$X_o$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Bảng 4-10. Bảng trạng thái của mạch tạo bit chẵn lẻ

Từ bảng trạng thái ta thấy  $X_o = \overline{X_e}$  hay  $X_e = \overline{X_o}$ .

Và biểu thức của  $X_o$  và  $X_e$  là

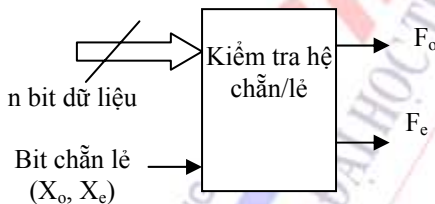
$$X_o = d_1 \oplus d_2 \oplus d_3$$

$$X_e = \overline{X_o} = \overline{d_1 \oplus d_2 \oplus d_3}$$



Hình 4-27. Sơ đồ khối của mạch tạo bit chẵn/lẻ

**4.9.2. Mạch kiểm tra chẵn/lẻ.**



Hình 4-28. Sơ đồ khối của mạch kiểm tra chẵn/lẻ

Vào				Ra	
$d_1$	$d_2$	$d_3$	$X$	$F_e$	$F_o$
0	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	1	0

Bảng 4-11. Bảng trạng thái của mạch kiểm tra chẵn/lẻ

Bảng trạng thái của mạch kiểm tra tính chẵn/lẻ của hệ được cho ở bảng 4-11.

Từ bảng trạng thái ta thấy:

- $F_e = 1$  nếu hệ là chẵn ( $F_e$  chỉ ra tính chẵn của hệ).
- $F_o = 1$  nếu hệ là lẻ ( $F_o$  chỉ ra tính lẻ của hệ).

Hai hàm này luôn là phủ định của nhau. Mặt khác do tính chất của hàm cộng XOR, ta có:

- $F_o = d_1 \oplus d_2 \oplus d_3 \oplus X$
- $F_e = \overline{F_o}$

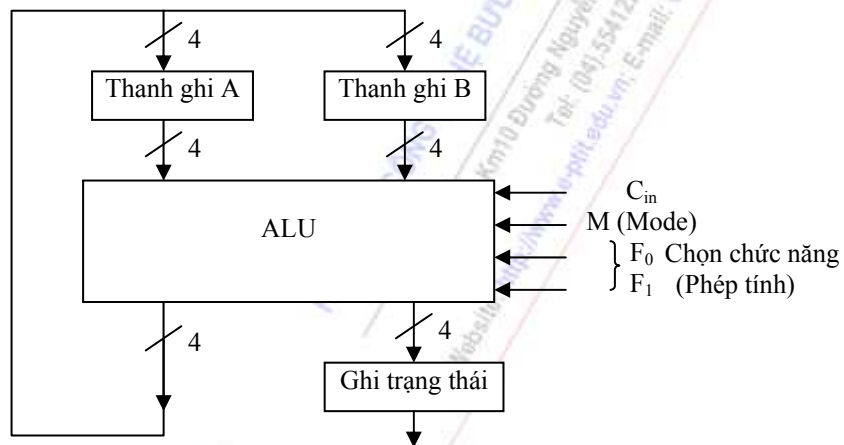
#### 4.10. ĐƠN VỊ SỐ HỌC VÀ LOGIC (ALU).

Đơn vị số học và logic (Arithmetic – Logic Unit) là một thành phần cơ bản không thể thiếu được trong các máy tính. Nó bao gồm 2 khối chính là khối logic và khối số học và một khối ghép kênh.

- Khối logic: Thực hiện các phép tính logic như là AND, OR, NOT, XOR.
- Khối số học: Thực hiện các phép tính số học như là: cộng, trừ, tăng 1, giảm 1.

Sơ đồ khối của 1 đơn vị số học – logic ALU 4 bit được mô tả ở hình 4-29:

M là lối vào chọn phép tính số học hay logic.  $F_0, F_1$  là hai lối vào chọn chức năng. Sau khi một phép tính số học hay logic được thực hiện thì kết quả sẽ được ghi lên 1 thanh ghi, ví dụ thanh ghi A. Kết quả này có thể được sử dụng để thực hiện phép tính sau. Bộ ALU còn tạo ra các bit trạng thái chuyển đổi thanh ghi. Ví dụ: Carry out: nếu có nhớ; Zero: nếu kết quả phép tính bằng 0.



Hình 4-29. Sơ đồ khối của ALU 4 bit

#### TÓM TẮT

Trong chương này, chúng ta đã giới thiệu mạch logic tổ hợp. Mạch tổ hợp do các phần tử logic cơ bản cấu trúc nên. Đặc điểm của mạch tổ hợp là tín hiệu đầu ra ở thời điểm bất kỳ nào cũng chỉ phụ thuộc vào tín hiệu ở đầu vào ở thời điểm đó mà không liên quan đến trạng thái vốn có của mạch.

Mạch tổ hợp rất phong phú, ta không thể xem xét hết trong chương 4. Trọng tâm của chúng ta là nắm vững đặc điểm mạch tổ hợp và phương pháp chung khi thiết kế, phân tích mạch tổ hợp. Vì vậy, chúng ta đã giới thiệu một cách chọn lọc bộ mã hoá, bộ giải mã, bộ hợp kênh, phân kênh, mạch cộng, trừ, mạch so sánh... trong quá trình đó, ta đã xem xét phương pháp phân tích và thiết kế mạch tổ hợp.

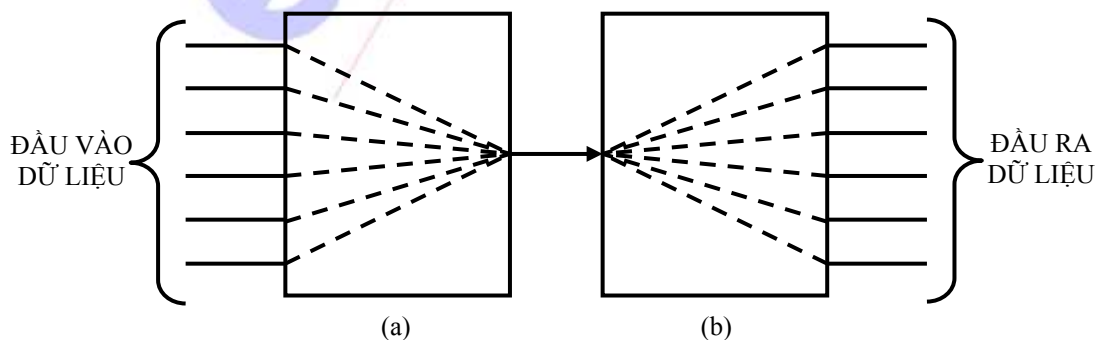
Khi phân tích mạch tổ hợp đã cho, ta có thể viết ra hàm logic đầu ra cho từng cấp của sơ đồ, rồi tiến hành tối thiểu hoá hàm logic đó để biểu thị rõ mối quan hệ giữa đầu ra với đầu vào. Cần lưu ý thêm rằng phải xem xét đến hiện tượng Hazard- là hiện tượng chạy đua trong mạch logic và cách khắc phục hiện tượng này.

Việc tối thiểu hoá hàm logic rất quan trọng. Vì việc này làm cho mạch logic đơn giản, kinh tế. Chúng ta mong muốn mạch điện càng ít linh kiện càng tốt, số đầu vào của mạch cổng cũng không thể quá nhiều

## CÂU HỎI ÔN TẬP

1. Mạch logic tổ hợp là mạch:
  - a. Có tín hiệu ở đầu ra chỉ phụ thuộc vào tín hiệu ở đầu vào của mạch tại thời điểm đang xét.
  - b. Không những tín hiệu ở đầu ra phụ thuộc vào tín hiệu ở đầu vào mà còn phụ thuộc vào trạng thái trong của mạch tại thời điểm đang xét.
  - c. Cả hai phương án trên đều đúng.
  - d. Không có phương án nào đúng.
2. Loại Hazard có trong mạch logic tổ hợp có thể là loại:
  - a. Hazard chỉ xuất hiện 1 lần và không bao giờ gặp nữa.
  - b. Hazard có thể xuất hiện nhiều lần.
  - c. Hazard có thể do chức năng của mạch điện gây ra.
  - d. Cả 3 phương án trên đều đúng.
3. Loại Hazard nào trong mạch logic tổ hợp là loại nguy hiểm nhất?
  - a. Hazard tĩnh.
  - b. Hazard động.
  - c. Hazard logic.
  - d. Không có phương án nào đúng.
4. Bộ mã hoá ưu tiên là bộ mã hoá cho phép mã hoá khi:
  - a. Chỉ có một tín hiệu tác động vào.
  - b. Chỉ hai tín hiệu tác động vào.
  - c. Có hai tín hiệu trở lên đồng thời tác động vào.
  - d. Cả 3 phương án trên đều đúng.
5. Bộ giải mã BCD sang thập phân làm nhiệm vụ biến đổi:
  - a. đầu vào nhị phân thành đầu ra thập lục phân (hệ hexa).
  - b. đầu vào thập phân thành mã BCD 8-4-2-1.
  - c. đầu vào BCD 8-4-2-1 thành đầu ra thập phân tương ứng.
  - d. Không có phương án nào đúng.
6. Dụng cụ hiển thị 7-đoạn:
  - a. chỉ có thể chỉ thị các ký tự từ 0 đến 9.

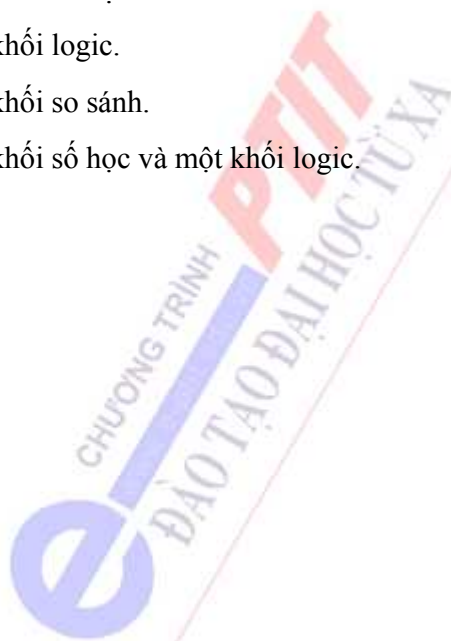
- b. chỉ có thể chỉ thị các ký tự từ A đến F.
  - c. chỉ có thể chỉ thị các ký tự từ 0 đến 9 và từ A đến F.
  - d. có thể được cấu tạo để chỉ thị các ký hiệu số, chữ cái hoặc các ký hiệu đặc biệt khác.
7. Dụng cụ hiển thị 7-đoạn Anốt chung:
- a. được biểu diễn bằng một Anốt đơn bên trong.
  - b. được biểu diễn bằng bảy đèn LED riêng lẻ.
  - c. được biểu diễn bằng một catốt đơn bên trong.
  - d. không có phương án nào đúng.
8. Bộ hợp kênh có khả năng:
- a. nối một lối vào mạch với một lối ra trong một nhóm các lối ra.
  - b. nối đồng thời một lối vào mạch với một hoặc nhiều lối ra của một nhóm các lối ra.
  - c. nối một lối vào trong một nhóm các lối vào với một lối ra.
  - d. nối đồng thời một hoặc nhiều lối vào với một lối ra.
9. Bộ phân kênh có khả năng:
- a. nối một lối vào mạch với một lối ra trong một nhóm các lối ra.
  - b. nối đồng thời một lối vào mạch với một hoặc nhiều lối ra trong một nhóm các lối ra.
  - c. nối một lối vào trong một nhóm các lối vào với một lối ra.
  - d. nối đồng thời một hoặc nhiều lối vào với một lối ra.
10. Mạch minh họa trong hình 4-29 là:
- a. cặp giải mã (a)/ mã hóa (b).
  - b. cặp mã hoá (a)/ giải mã (b).
  - c. cặp hợp kênh (a)/phân kênh (b).
  - d. cặp phân kênh (a)/hợp kênh (b).



Hình 4-29.

11. IC trong hình 4-29(a) được gọi là:
  - a. bộ hợp kênh 8 vào - 1 ra.
  - b. bộ phân kênh 8 vào - 1 ra.
  - c. bộ hợp kênh 1 vào - 8 ra.
  - d. bộ phân kênh 1 vào - 8 ra.
12. IC trong hình 4-29(b) được gọi là:
  - a. bộ hợp kênh 8 vào - 1 ra.
  - b. bộ phân kênh 8 vào - 1 ra.
  - c. bộ hợp kênh 1 vào - 8 ra.
  - d. bộ phân kênh 1 vào - 8 ra.
13. Thuật ngữ *parity* (tính chẵn lẻ):
  - a. dùng để chỉ kích thước đường dữ liệu của hệ thống.
  - b. chỉ có thể dùng cho các hệ thống 8-bit.
  - c. liên quan đến quá trình kiểm tra lỗi.
  - d. dùng cho thanh ghi dịch.
14. Nếu bộ tạo bit chẵn lẻ nhận một bit kiểm tra parity chẵn, nó yêu cầu nhận:
  - a. dữ liệu parity chẵn.
  - b. dữ liệu parity lẻ.
  - c. một trong hai trường hợp trên.
  - d. Không phải hai trường hợp trên.
15. Khi ghép bộ cộng 2 số nhị phân 4 bit có thể :
  - a. Cộng thành các số 8 bit.
  - b. Cộng thành các số 4 bit.
  - c. Tạo ra một tổng 8 bit.
  - d. Tạo ra một số 8 bit khác.
16. Lỗi ra của từng tổng của bộ cộng có được là do thực hiện cộng :
  - a. Tất cả 4 bit của từng số nhị phân.
  - b. từng cặp bit một.
  - c. Bit nhớ.
  - d. 1 với bit trước đó.
17. Nếu lỗi ra  $A > B$  của bộ so sánh được kích hoạt, thì:

- a. Giá trị của số A lớn hơn giá trị của số B.
  - b. Cả hai số ở lối vào đều có giá trị giống nhau.
  - c. Giá trị của số A nhỏ hơn giá trị của số B.
  - d. Giá trị của số B lớn hơn giá trị của số A.
18. Nếu lối ra  $A=B$  của bộ so sánh được kích hoạt, thì:
- a. Giá trị của số A lớn hơn giá trị của số B.
  - b. Cả hai số ở lối vào đều có giá trị giống nhau.
  - c. Giá trị của số A nhỏ hơn giá trị của số B.
  - d. Giá trị của số B lớn hơn giá trị của số A.
19. Nếu lối ra  $A<B$  của bộ so sánh được kích hoạt, thì:
- a. Giá trị của số A lớn hơn giá trị của số B.
  - b. Cả hai số ở lối vào đều có giá trị giống nhau.
  - c. Giá trị của số A nhỏ hơn giá trị của số B.
  - d. Giá trị của số B nhỏ hơn giá trị của số A.
20. Một ALU có chứa:
- a. Một khối số học.
  - b. Một khối logic.
  - c. Một khối so sánh.
  - d. Một khối số học và một khối logic.



PHÂN VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
Km10 Đường Nguyễn Trãi, Hà Đông-Hà Tây  
Tel: (04) 5541221; Fax: (04) 5540587  
Website: <http://www.e-pht.com.vn>; E-mail: [dhk@pht.com.vn](mailto:dhk@pht.com.vn)

## CHƯƠNG 5: MẠCH LOGIC TUẦN TỰ

### GIỚI THIỆU.

Chúng ta đã nghiên cứu về phép phân tích và thiết kế các mạch logic tổ hợp. Mặc dù rất quan trọng nhưng nó chỉ là một phần của các hệ thống kỹ thuật số. Một phần quan trọng của các hệ thống kỹ thuật số khác là phân tích và thiết kế mạch tuần tự. Tuy nhiên việc thiết kế các mạch tuần tự lại phụ thuộc vào việc thiết kế mạch tổ hợp đã được đề cập ở chương 4.

Có nhiều ứng dụng mà đầu ra số phải được tạo để phù hợp với tuần tự nhận được các tín hiệu vào. Yêu cầu này không thể được thỏa mãn bằng việc sử dụng hệ thống logic tổ hợp.

Những ứng dụng này yêu cầu đầu ra không chỉ phụ thuộc vào các điều kiện đầu vào hiện có mà còn phụ thuộc vào lịch sử của các đầu vào. Lịch sử được cung cấp bằng cách phản hồi từ đầu ra về lại đầu vào.

Mạch logic tuần tự không những phụ thuộc vào trạng thái các lối vào và còn phụ thuộc vào trạng thái trong của nó. Mạch tuần tự được chia làm hai loại chính là mạch tuần tự không đồng bộ và mạch tuần tự đồng bộ.

Trong phần này chúng ta sẽ giới thiệu về các phần tử nhớ của mạch tuần tự. Cách phân tích và thiết kế mạch tuần tự đơn giản và phức tạp.

### NỘI DUNG

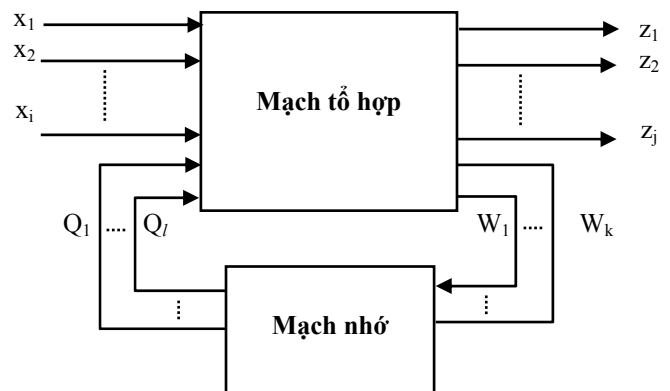
#### 5.1. KHÁI NIỆM CHUNG VÀ MÔ HÌNH TOÁN HỌC

##### 5.1.1. Khái niệm chung

Trong chương này, chúng ta sẽ nói đến hệ thống số được gọi là mạch logic tuần tự (hay còn gọi là mạch dây - Sequential Circuit). Hoạt động của hệ này có tính chất kế tiếp nhau, tức là trạng thái hoạt động của mạch điện không những phụ thuộc trực tiếp lối vào mà còn phụ thuộc vào trạng thái bên trong trước đó của chính nó. Nói cách khác các hệ thống này làm việc theo nguyên tắc có nhớ.

##### 5.1.2. Mô hình toán học

Mạch tuần tự là mạch bao gồm mạch logic tổ hợp và mạch nhớ. Mạch nhớ là các trigơ. Đối với mạch tuần tự, đáp ứng ra của hệ thống mạch điện không chỉ phụ thuộc trực tiếp vào tín hiệu vào ( $X$ ) mà còn phụ thuộc vào trạng thái nội ( $Q$ ) của nó. Có thể mô tả sơ đồ khối tổng quát của mạch tuần tự.



Hình 5-1. Sơ đồ khối của mạch tuần tự.

- Ở đây: X - tập tín hiệu vào.  
 Q - tập trạng thái trong trước đó của mạch.  
 W - hàm kích.  
 Z - các hàm ra

Hoạt động của mạch tuần tự được mô tả bằng mối quan hệ toán học sau:

$$Z = f(Q, X)$$

Trong phương trình toán học của mạch tuần tự ta thấy có hai thông tin. Đó là thông tin về trạng thái tiếp theo của mạch tuần tự và thông tin về tín hiệu ra của mạch. Hai thông tin này cùng phụ thuộc đồng thời vào trạng thái bên trong trước đó của mạch (Q) và tín hiệu tác động vào (X) của nó. Ta có thể viết lại biểu thức trên như sau:

$$Z = f(Q(n), X).$$

$$Q(n+1) = f(Q(n), X)$$

Trong đó: Q(n+1): là trạng thái tiếp theo của mạch.

Q(n): là trạng thái bên trong trước đó. Để tiện cho việc nghiên cứu ta sẽ ký hiệu Q(n+1) là  $Q^k$ , Q(n) là Q.

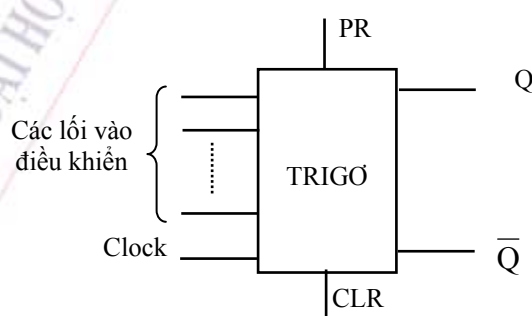
Để hiểu rõ hơn về mạch tuần tự ta đi xét các phần tử có trong mạch. Như ta đã biết mạch logic tổ hợp đã được xét ở chương 4. Bây giờ ta sẽ tìm hiểu về mạch nhớ, mà phần tử nhớ chính là các trigơ.

## 5.2. PHẦN TỬ NHỚ CỦA MẠCH TUẦN TỰ

### 5.2.1. Các loại Trigo

**Định nghĩa:** Trigo là phần tử có khả năng lưu trữ (nhớ) một trong hai trạng thái 0 và 1.

Trigo có từ 1 đến một vài lối điều khiển, có hai lối ra luôn luôn ngược nhau là Q và  $\bar{Q}$ . Tùy từng loại trigơ có thể có thêm các lối vào lập (PRESET) và lối vào xoá (CLEAR). Ngoài ra, trigơ còn có lối vào đồng bộ (CLOCK). Hình 5-2 là sơ đồ khối tổng quát của trigơ.



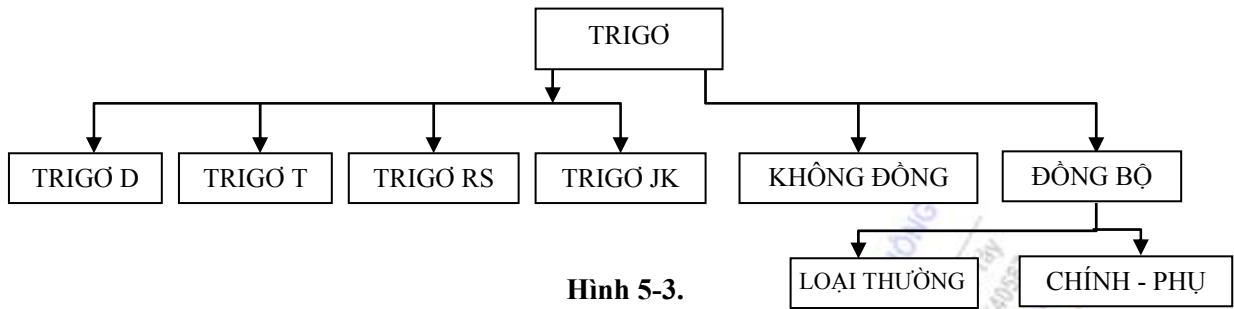
Hình 5-2. Sơ đồ tổng quát của một Trigo

**Phân loại:**

- ◆ Theo chức năng làm việc của của các lối vào điều khiển: hiện nay thường sử dụng loại trigơ 1 lối vào như trigơ D, T; loại hai lối vào như trigơ RS, trigơ JK.
- ◆ Theo phương thức hoạt động thì ta có hai loại: trigơ đồng bộ và trigơ không đồng bộ. Trong loại trigơ đồng bộ lại được chia làm hai loại: trigơ thường và trigơ chính - phụ (Master- Slave).

Sơ đồ khối của sự phân loại trigơ được cho ở hình 5-3.

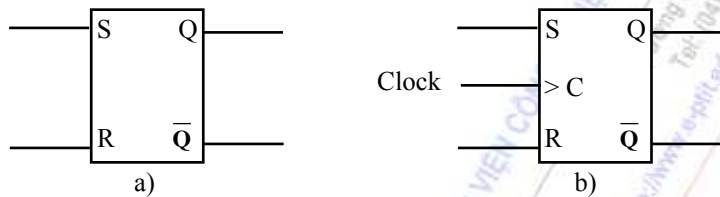




Hình 5-3.

5.2.1.1. Trơ RS

Trơ RS là loại có hai lối vào điều khiển S, R. Chân S gọi là lối vào "lập" (SET) và R được gọi là lối vào "xoá" (RESET).



Hình 5-4. Sơ đồ ký hiệu của trơ RS

Hình 5-4 là ký hiệu của trơ RS trong các sơ đồ logic (hình a là sơ đồ của trơ RS không đồng bộ, hình b là sơ đồ của trơ RS đồng bộ). Hình 5-5 là sơ đồ nguyên lý của trơ RS và RS đồng bộ. Trạng thái ở đầu ra của Q phụ thuộc vào các tín hiệu logic ở hai lối vào điều khiển S, R theo bảng trạng thái 5-1 và 5-2 :

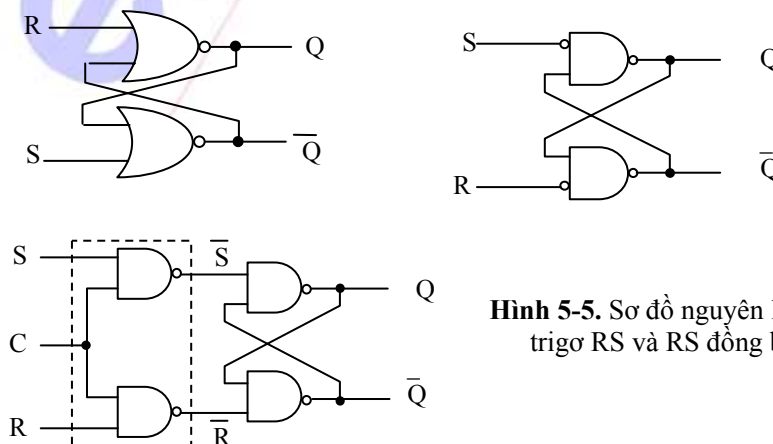
S	R	Q <sup>k</sup>	Mod hoạt động
0	0	Q	Nhớ
0	1	0	Xoá
1	0	1	Lập
1	1	x	Cấm

Bảng 5-1. Bảng trạng thái của trơ RS

C	S	R	Q <sup>k</sup>	Mod hoạt động
0	x	x	Q	Nhớ
1	0	0	Q	Nhớ
1	0	1	0	Xoá
1	1	0	1	Lập
1	1	1	x	Cấm

Bảng 5-2. Bảng trạng thái của trơ RS đồng bộ cổng NAND

Trong bảng, ký hiệu Q<sup>k</sup> là giá trị ở lối ra Q ở thời điểm kế tiếp, Q là giá trị tại thời điểm hiện tại.



Hình 5-5. Sơ đồ nguyên lý của trơ RS và RS đồng bộ

Ta thấy khi  $S = 1, R = 0$  thì  $Q^k = 1$ ; khi  $S = 0, R = 1$  thì  $Q^k = 0$ . Đây chính là hai điều kiện điều khiển ở lối vào khiến cho lối ra của trigơ có thể lật trạng thái.  $S$  và  $R$  là các lối vào điều khiển. Trường hợp  $S = 0, R = 0$  thì  $Q^k = Q$ , điều này có nghĩa là khi không có tín hiệu điều khiển thì trigơ vẫn giữ nguyên trạng thái vốn có của nó. Cuối cùng khi  $S = R = 1$  thì lối ra  $Q^k$  và  $\bar{Q}^k$  có giá trị bằng nhau (có thể là 1, có thể là 0) nên ta nói trạng thái của trigơ là không xác định hay gọi là trạng thái cấm. Vậy, **không bao giờ được sử dụng trường hợp này.**

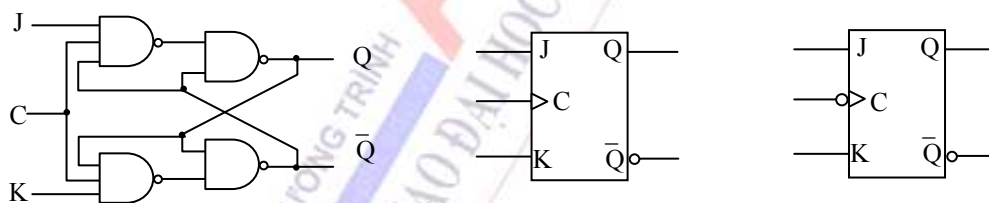
### 5.2.1.2. Trigơ JK

Trigơ JK là loại trigơ có hai lối vào điều khiển  $J, K$ . Trigơ này có ưu điểm hơn trigơ RS là không còn tồn tại tổ hợp cấm bằng các đường hồi tiếp từ  $Q$  về chân  $R$  và từ  $\bar{Q}$  về  $S$ . Tuy nhiên, điểm đặc biệt là trigơ JK còn có thêm đầu vào đồng bộ  $C$ . Trigơ có thể lập hay xoá trong khoảng thời gian ứng với sườn âm hoặc sườn dương của xung đồng bộ  $C$ . Ta nói, trigơ RS thuộc loại **đồng bộ**.

Sự hoạt động của trigơ JK được trình bày bằng bảng trạng thái 5-2

C	J	K	$Q^k$	Mod hoạt động
0	x	x	Q	Nhớ (đối với loại trigơ JK dùng cổng NAND)
1	x	x	Q	Nhớ (đối với loại trigơ JK dùng cổng NOR)
Ck	0	0	Q	Nhớ
Ck	0	1	0	Xoá
Ck	1	0	1	Lập
Ck	1	1	$\bar{Q}$	Thay đổi trạng thái theo mỗi xung nhịp

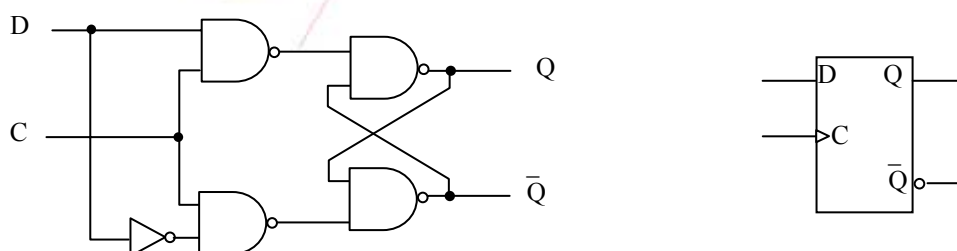
**Bảng 5-3.** Bảng trạng thái của trigơ JK đồng bộ



**Hình 5-6.** Sơ đồ nguyên lý và ký hiệu của trigơ JK đồng bộ

Sơ đồ nguyên lý và sơ đồ ký hiệu của trigơ JK được trình bày ở hình 5-6.

### 5.2.1.3. Trigơ D



**Hình 5-7.** Sơ đồ nguyên lý và ký hiệu của trigơ D đồng bộ

Trigơ D là loại trigơ có một lối vào điều khiển D. Tín hiệu ở lối vào điều khiển sẽ truyền tới lối ra Q ( $Q^k = D$ ) mỗi khi xuất hiện xung nhịp C. Trigơ D thường được dùng làm bộ ghi dịch dữ liệu hay bộ chốt dữ liệu. Sơ đồ nguyên lý và sơ đồ ký hiệu của trigơ D được biểu diễn ở hình 5-7.

#### 5.2.1.4. Trigơ T

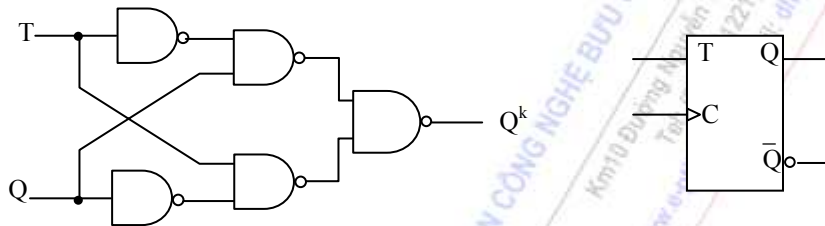
Trigơ T là loại trigơ có một lối vào điều khiển T. Mỗi khi có xung tới lối vào T thì lối ra Q sẽ thay đổi trạng thái.

Bảng 5-3 là bảng trạng thái của trigơ T

Sơ đồ nguyên lý và ký hiệu của trigơ T được biểu diễn ở hình 5-8.

T	$Q^k$
0	Q
1	$\bar{Q}$

**Bảng 5-4.** Bảng trạng thái của trigơ T



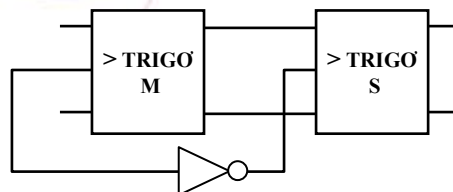
**Hình 5-8.** Sơ đồ nguyên lý và ký hiệu của trigơ T

- ◆ Nhận xét: Từ các bảng trạng thái của các trigơ trên ta thấy rằng: Các trigơ D và RS có thể làm việc được ở chế độ không đồng bộ vì mỗi tập tín hiệu vào điều khiển D, RS luôn luôn tồn tại ít nhất 1 trong 2 trạng thái ổn định. Trạng thái ổn định là trạng thái thỏa mãn điều kiện  $Q^k = Q$ . Còn trigơ T và trigơ JK không thể làm việc được ở chế độ không đồng bộ vì mạch sẽ rơi vào trạng thái dao động nếu như tập tín hiệu vào là '11' đối với trigơ JK hoặc là '1' đối với trigơ T. Như vậy, trigơ D, trigơ RS có thể làm việc ở cả hai chế độ: đồng bộ và không đồng bộ còn trigơ T và trigơ JK chỉ có thể làm việc ở chế độ đồng bộ.

#### 5.2.1.5. Các loại trigơ Chính- Phụ (MS-Master- Slave).

Do các loại trigơ đồng bộ trên đều hoạt động tại sườn dương hay sườn âm của xung nhịp nên khi làm việc ở tần số cao thì lối ra Q không đáp ứng kịp với sự thay đổi của xung nhịp, dẫn đến mạch hoạt động ở tình trạng không được tin cậy. Loại trigơ MS khắc phục được nhược điểm này. Lối ra của trigơ MS thay đổi tại sườn dương và sườn âm của xung nhịp, nên cấu trúc của nó gồm 2 trigơ giống nhau nhưng cực tính điều khiển của xung Clock thì ngược nhau để đảm bảo sao cho tại mỗi sườn của xung sẽ có một trigơ hoạt động. Về nguyên tắc hoạt động của loại trigơ MS (RS-MS, JK-MS, D-MS, T-MS) hoàn toàn giống như các loại trigơ thông thường (RS, JK, D, T).

Cấu trúc chung của một trigơ MS được minh họa ở hình 5-9.

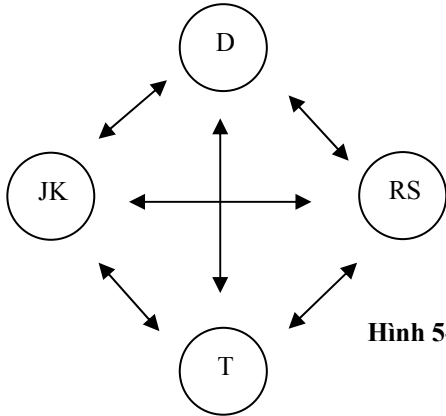


**Hình 5-9.** Cấu trúc của trigơ MS

**5.2.2. Chuyển đổi giữa các loại trigơ.**

Có 4 loại trigơ đã được giới thiệu là trigơ RS, JK, D và T. Trên thực tế có khi trigơ loại này lại được sử dụng như trigơ loại khác. Nội dung phần này là xây dựng các trigơ yêu cầu từ các trigơ cho trước.

Với 4 loại trigơ trên thì có 12 khả năng chuyển đổi sang nhau.



**Hình 5-10.** Các khả năng chuyển đổi giữa các loại trigơ.

**5.2.2.1. Phương pháp chuyển đổi giữa các loại trigơ.**

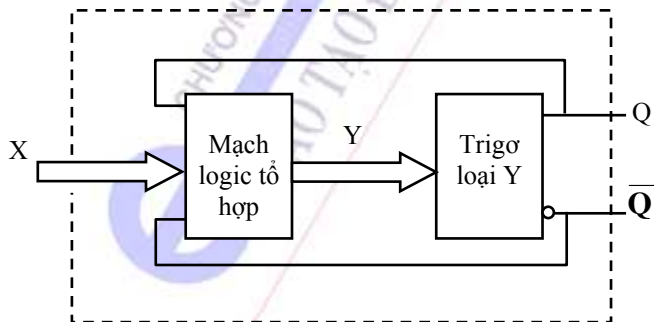
Một trong các phương pháp để xây dựng trigơ loại X từ loại Y cho trước được cho ở sơ đồ khối ở hình 5-11.

Các lối vào X là các lối vào của trigơ loại X cần thiết kế. Lối ra của mạch logic là các lối vào của trigơ Y cho trước. Như vậy, bài toán chuyển đổi từ trigơ loại Y sang trigơ loại X là xây dựng mạch tổ hợp có các đầu vào là X và Q; các lối ra là Y biểu diễn bởi hệ hàm:

$$Y = f(X, Q)$$

Để thực hiện chuyển đổi trigơ loại Y sang loại X cần thực hiện các bước sau:

- ◆ Xác định hệ hàm  $Y = f(X, Q)$  theo bảng hàm kích.
- ◆ Tối thiểu hoá các hàm này và xây dựng các sơ đồ.



**Hình 5-11.** Sơ đồ khối của trigơ loại X

Bảng hàm kích của các loại trigơ được cho ở bảng 5-5.

Q	Q <sup>k</sup>	RS	JK	D	T
0	0	X0	0X	0	0
0	1	01	1X	1	1
1	0	10	X1	0	1
1	1	0X	X0	1	0

**Bảng 5-5.** Bảng hàm kích của các loại trigơ

Sau đây ta xét một số ví dụ xây dựng các trigơ từ các trigơ cho trước thường hay được sử dụng trong thực tế.

Ví dụ: Chuyển đổi từ trigơ RS sang trigơ JK.

Ta cần phải thiết kế mạch logic tổ hợp của các hàm logic:

$$R = f_1(Q, J, K)$$

$$S = f_2(Q, J, K)$$

Từ bảng hàm kích thích trên ta thu được bảng Karnaugh (bảng 5-6) cho S và R với các biến vào là Q, J, K.

JK \ Q	00	01	11	10
0	0	0	1	1
1	X	0	0	X

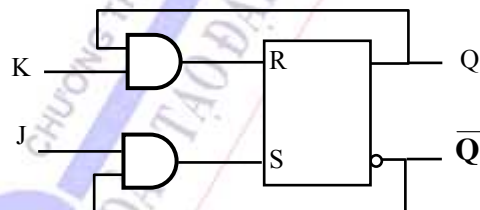
$$S = J\bar{Q}$$

JK \ Q	00	01	11	10
0	X	X	0	0
1	0	1	1	0

$$R = KQ$$

**Bảng 5-6.** Bảng tính S và R

Mạch thực hiện chuyển đổi được cho ở hình 5-12.



**Hình 5-12.** Trigơ JK xây dựng từ trigơ RS

Tương tự như vậy ta cũng có thể làm như vậy đối với các chuyển đổi khác.

### 5.3. PHƯƠNG PHÁP MÔ TẢ MẠCH TUẦN TỰ.

Thiết bị được thiết kế phải được mô tả bằng lời hay một số hình thức khác. Công việc đầu tiên của người thiết kế là phải phiên dịch các dữ kiện đó thành 1 hình thức mô tả hoạt động của thiết bị cần phải thiết kế một cách trung thực và duy nhất. Nói cách khác là phải hình thức hoá dữ liệu ban đầu.

Có hai cách hình thức hoá thường dùng đó là dùng bảng và đồ hình trạng thái.

5.3.1. Bảng

5.3.1.1. Bảng chuyển đổi trạng thái.

Bảng chuyển đổi trạng thái bao gồm các hàng và các cột, các hàng ghi các trạng thái trong, các cột ghi các giá trị của tín hiệu vào. Các ô ghi giá trị các trạng thái trong kế tiếp mà mạch sẽ chuyển đến ứng với các giá trị ở hàng và cột. Bảng chuyển đổi trạng thái được mô tả ở bảng 5-6.

		Tín hiệu vào				
		V	V <sub>1</sub>	V <sub>2</sub>	.....	V <sub>n</sub>
Trạng thái trong	S					
	S <sub>1</sub>					
	S <sub>2</sub>					
	:					
	S <sub>n</sub>					

Trạng thái kế tiếp Q<sup>k</sup>

Bảng 5-6. Bảng chuyển đổi trạng thái

5.3.1.2. Bảng tín hiệu ra.

Các hàng của bảng ghi các trạng thái trong, các cột ghi các tín hiệu vào. Các ô ghi giá trị của tín hiệu ra tương ứng. Bảng tín hiệu ra được mô tả ở bảng 5-7.

		Tín hiệu vào				
		V	V <sub>1</sub>	V <sub>2</sub>	.....	V <sub>n</sub>
Trạng thái trong	S					
	S <sub>1</sub>					
	S <sub>2</sub>					
	:					
	S <sub>n</sub>					

Tín hiệu ra - R

Bảng 5-7. Bảng tín hiệu ra

Có thể gộp hai bảng chuyển đổi trạng thái và bảng tín hiệu ra thành một bảng chung gọi là bảng chuyển đổi trạng thái / ra. Lúc đó trên các ô ghi các giá trị của trạng thái kế tiếp và tín hiệu ra (S<sup>k</sup> / R) tương ứng với trạng thái hiện tại và tín hiệu vào.

Bảng chuyển đổi trạng thái và tín hiệu ra được mô tả ở bảng 5-8.

		Tín hiệu vào				
		V	V <sub>1</sub>	V <sub>2</sub>	.....	V <sub>n</sub>
Trạng thái trong	S					
	S <sub>1</sub>					
	S <sub>2</sub>					
	:					
	S <sub>n</sub>					

Trạng thái kế tiếp S<sup>k</sup> và Tín hiệu ra - R

5.3.2. Đồ hình trạng thái.

Bảng 5-8. Bảng chuyển đổi trạng thái và tín hiệu ra

Đồ hình trạng thái là hình vẽ phản ánh quy luật chuyển đổi trạng thái và tình trạng các giá trị ở lối vào và lối ra tương ứng của mạch tuần tự.

Đồ hình trạng thái là một đồ hình có hướng gồm hai tập:

M - Tập các đỉnh và K - Tập các cung có hướng.

**a). Đối với mô hình Mealy thực hiện ánh xạ.**

Tập các trạng thái trong là tập các đỉnh M; Tập các tín hiệu vào / ra là tập các cung K.

Trên cung có hướng đi từ trạng thái trong  $S_i$  đến trạng thái trong  $S_j$  ghi tín hiệu vào/ra tương ứng.

**b). Đối với mô hình Moore.**

Vì tín hiệu ra chỉ phụ thuộc vào trạng thái trong của mạch mà không phụ thuộc vào tín hiệu vào cho nên thực hiện ánh xạ:

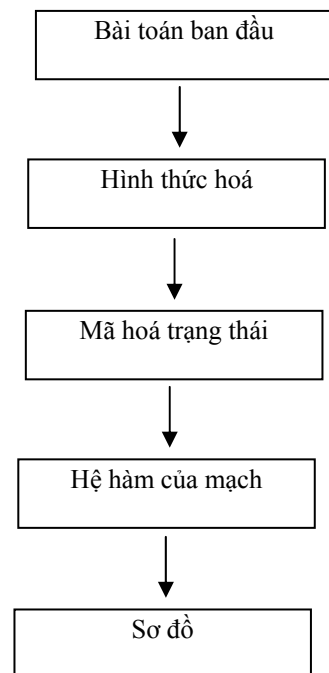
Tập các trạng thái trong, tín hiệu ra là tập các đỉnh M.

Tập các tín hiệu vào là tập các cung K.

**5.4. CÁC BƯỚC THIẾT KẾ MẠCH TUẦN TỰ.**

Quá trình thiết kế mạch tuần tự được mô tả theo lưu đồ sau

- ◆ **Bài toán ban đầu:** Nhiệm vụ thiết kế được mô tả bằng ngôn ngữ hoặc bằng lưu đồ thuật toán.
- ◆ **Hình thức hoá:** Từ các dữ kiện đề bài cho mà ta mô tả hoạt động của mạch bằng cách hình thức hoá dữ kiện ban đầu ở dạng bảng trạng thái, bảng ra hay đồ hình trạng thái. Sau đó rút gọn các trạng thái của mạch để có được số trạng thái trong ít nhất.
- ◆ **Mã hoá trạng thái:** Mã hoá tín hiệu vào ra, trạng thái trong để nhận được mã nhị phân (hoặc có thể là các loại mã khác) có tập tín hiệu vào là X, tập tín hiệu ra là Y, tập các trạng thái trong là Q.
- ◆ **Hệ hàm của mạch:** Xác định hệ phương trình logic của mạch và tối thiểu hoá các phương trình này. Nếu mạch tuần tự khi thiết kế cần dùng các trigơ và mạch tổ hợp thì tùy theo yêu cầu mà ta viết hệ phương trình cho các lối vào kích cho từng loại trigơ đó.
- ◆ **Xây dựng sơ đồ:** Từ hệ phương trình của mạch đã viết được ta xây dựng mạch điện thực hiện.



**Hình 5-13.** Các bước thiết kế mạch tuần tự

**5.4.1. Thiết kế mạch tuần tự từ đồ hình trạng thái.**

**Giả thiết:** Cho đồ hình trạng thái của mạch có tập tín hiệu vào V, tập tín hiệu ra R, tập trạng thái trong S (chưa mã hoá nhị phân).

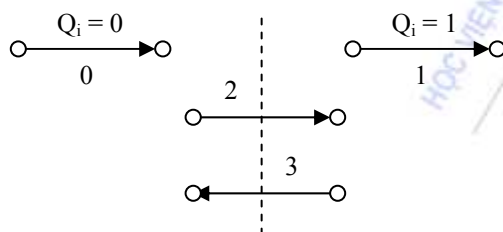
**Xác định:** Hệ phương trình nhị phân của mạch (đã tối thiểu hoá). Trên cơ sở đó vẽ mạch điện.

#### 5.4.1.1. Các bước thiết kế

- ◆ Mã hoá tín hiệu vào  $V$ , tín hiệu ra  $R$ , trạng thái trong  $S$  để chuyển thành mạch dạng nhị phân có các tập tín hiệu vào  $X$ , tín hiệu ra  $Y$ , trạng thái trong  $Q$ .
- ◆ Xác định hệ phương trình tín hiệu ra:  $Y_i = f_i(X, Q)$ . Phương trình này được xác định trên các cung với mô hình kiểu Mealy, trên các đỉnh với mô hình kiểu Moore. Tối thiểu các hàm này.
- ◆ Xác định hệ phương trình hàm kích cho các trigơ và tối thiểu hoá nó.

Sau đây giới thiệu thuật toán xác định phương trình lối vào kích cho các trigơ từ đồ hình trạng thái.

Đối với trigơ  $Q_i$  bất kỳ sự thay đổi trạng thái từ  $Q_i$  đến  $Q_i^k$  chỉ có thể có 4 khả năng như hình 5-16.



**Hình 5-14.** Các cung biểu diễn sự thay đổi trạng thái từ  $Q_i$  đến  $Q_i^k$  của trigơ  $Q_i$

Trong đó các cung biểu diễn sự thay đổi từ  $Q_i$  đến  $Q_i^k$  được ký hiệu như sau:  $0 \rightarrow 0$  là (0),  $1 \rightarrow 1$  (là 1),  $0 \rightarrow 1$  là (2),  $1 \rightarrow 0$  là (3).

Từ quy ước có thuật toán sau:

##### a. Thuật toán xác định phương trình lối vào kích cho trigơ $Q_i$ loại D.

Phương trình đặc trưng của trigơ D :  $Q_i^k = D_i$ . Từ đó ta rút ra

$$\begin{aligned} D_i = Q_i^k &= \text{tuyển tất cả các cung đi tới đỉnh có } Q_i = 1. \\ &= \Sigma \text{ các cung loại (2), kể cả khuyên tại đỉnh đó tức là cung loại 1} \\ &= \Sigma (1) \text{ và (2)} \end{aligned}$$

Tối thiểu hoá hàm  $D_i$  vừa tìm được rút ra phương trình lối vào kích cho trigơ loại D.

##### b. Thuật toán xác định phương trình lối vào kích cho trigơ T

Phương trình đặc trưng của trigơ T:  $Q_i^k = T_i \oplus Q_i \Rightarrow T_i = Q_i \oplus Q_i^k = Q_i'$

Trong đó  $Q_i'$  bằng 1 khi  $Q_i$  thay đổi trạng thái từ  $0 \Rightarrow 1$  hoặc từ  $1 \Rightarrow 0$ , ta làm như sau:

- Điền sự thay đổi giá trị của  $Q_i$  vào các cung.

-  $T_i = Q_i' = \Sigma$  các cung có  $Q_i$  thay đổi (cung loại 2, loại 3) =  $\Sigma$  (2) và (3).

Tối thiểu hoá hàm  $T_i$  vừa tìm được rút ra phương trình kích cho trigơ T.



**c. Thuật toán xác định phương trình lối vào kích cho trigơ JK**

Phương trình đặc trưng của trigơ JK:  $Q_i^k = J \overline{Q_i} + \overline{K} Q_i$

Xác định:

$T_{on} = \Sigma$  các cung mà  $Q_i$  được bật ( $Q_i$  thay đổi từ 0  $\Rightarrow$  1 - cung loại 2) =  $\Sigma$  (2). Đưa phương trình của  $T_{on}$  về dạng:

$$T_{on} = (T^*) \overline{Q_i} \Rightarrow \text{rút ra } J = T^*$$

$T_{off} = \Sigma$  các cung mà  $Q_i$  được tắt ( $Q_i$  thay đổi từ 1  $\Rightarrow$  0 - cung loại 3) =  $\Sigma$  (3). Đưa phương trình của  $T_{off}$  về dạng:

$$T_{off} = (T^{**}) \overline{Q_i} \Rightarrow \text{rút ra } K = T^{**}$$

**d. Thuật toán xác định phương trình lối vào kích cho trigơ RS**

Phương trình lối vào S của trigơ RS được xác định như sau:

$$S = T_{on} + [\text{Các cung loại (1)}]$$

$$R = T_{off} + [\text{Các cung loại (0)}]$$

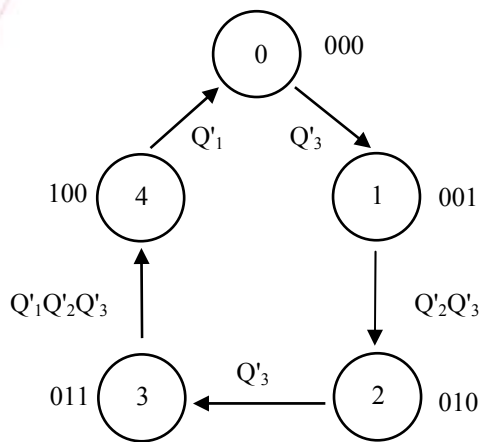
Các cung loại (1), các cung loại (0) để trong dấu [ ] ở biểu thức của S, R được lấy giá trị không xác định. Những giá trị này và những trạng thái không được sử dụng sẽ được dùng để tối thiểu hoá sao cho biểu thức nhận được là tối giản nhất.

**5.4.1.2. Ví dụ**

Để minh hoạ, xét ví dụ sau: Thiết kế bộ đếm đồng bộ có  $M_d = 5$  với đồ hình trạng thái và mã hoá trạng thái như ở hình 5-17, dùng

$Q_2 Q_3$ \ $Q_1$	00	01	11	10
0	0	1	3	2
1	4	x	x	x

b). Bảng mã hoá trạng thái  
Hình 5-15.



a). Đồ hình trạng thái

- a) Trigơ D và các mạch AND.
- b) Trigơ T và các mạch AND.

- c) Trigrơ JK và các mạch AND.
- d) Trigrơ RS và các mạch AND.

Bộ đếm M=5 nên có 5 trạng thái 0, 1, 2, 3, 4. Để đơn giản, trên đồ hình ta không ghi các tín hiệu vào đếm và tín hiệu ra. Tín hiệu ra của bộ đếm chỉ xuất hiện khi bộ đếm đang ở trạng thái 4 và có tín hiệu vào đếm, lúc đó bộ đếm quay trở về trạng thái ban đầu 0 và cho ra tín hiệu ra.

Mạch có 5 trạng thái và do vậy được mã hoá ít nhất bằng 3 biến nhị phân tương ứng với 3 trigrơ:  $Q_1, Q_2, Q_3$  như trên bảng mã hoá trạng thái hình 5-17b. Điền mã tương ứng vào các trạng thái trên đồ hình 5-17a.

Từ đó ta viết được phương trình cho tín hiệu ra Y:  $Y = Q_1 \overline{Q_2} \overline{Q_3} \cdot X_d$ .

Sử dụng các trạng thái tùy chọn để tối thiểu hoá, từ đó ta nhận được kết quả

$$Y = Q_1 X_d$$

Bây giờ ta xác định các phương trình kích cho các trigrơ :

**a) Trigrơ D.**

Nhìn vào đồ hình trạng thái ta thấy:  $Q_1 = 1$  tại đỉnh (4),  $Q_2 = 1$  tại đỉnh (2), (3),  $Q_3 = 1$  tại đỉnh (1), (3).

$$D_1 = \sum \text{Các cung đi đến đỉnh (4) = (3)} = \overline{Q_1} Q_2 Q_3.$$

$$D_2 = \sum \text{Các cung đi đến đỉnh (2), (3)} = (1) + (2) = \overline{Q_1} \overline{Q_2} Q_3 + \overline{Q_1} Q_2 \overline{Q_3}.$$

$$D_3 = \sum \text{Các cung đi đến đỉnh (1), (3)} = (0) + (2) = \overline{Q_1} \overline{Q_2} \overline{Q_3} + \overline{Q_1} Q_2 \overline{Q_3}.$$

$Q_2 Q_3 \backslash Q_1$	00	01	11	10
0			1	
1		x	x	x

$Q_2 Q_3 \backslash Q_1$	00	01	11	10
0		1		1
1		x	x	x

$$D_1 = Q_2 Q_3$$

$$D_2 = \overline{Q_2} Q_3 + Q_2 \overline{Q_3} = Q_2 \oplus Q_3$$

$Q_2 Q_3 \backslash Q_1$	00	01	11	10
0	1			1
1		x	x	x

$$D_3 = \overline{Q_1} \overline{Q_3}$$

**Bảng 5-9.** Bảng tìm hàm kích

Dùng bảng Karnaugh 5-9 ta thu được kết quả

$$D_1 = Q_2 Q_3$$

$$D_2 = \overline{Q_2} Q_3 + Q_2 \overline{Q_3} = Q_2 \oplus Q_3$$

$$D_3 = \overline{Q_1} \overline{Q_3}$$

**b) Xác định phương trình kích cho Trigo T.**

Điền sự thay đổi giá trị của  $Q_i$  ( $Q_i$ ) vào các cung. Khi mạch đếm từ trạng thái (0)  $\Rightarrow$  (1) (nghĩa là từ 000  $\Rightarrow$  001) thì  $Q_3$  thay đổi từ 0  $\Rightarrow$  1 nên ta ghi  $Q_3$  lên cung đó. Khi mạch chuyển từ trạng thái (1)  $\Rightarrow$  (2) (tương ứng từ 001  $\Rightarrow$  010):  $Q_1$  không thay đổi trạng thái (= 0),  $Q_2$  thay đổi từ 0  $\Rightarrow$  1 và  $Q_3$  thay đổi từ 1  $\Rightarrow$  0, nên ta ghi  $Q_2 Q_3$  lên cung từ (1)  $\Rightarrow$  (2). Tương tự như vậy ta có:

$$T_1 = Q_1 = \sum \text{các cung có } Q_1 \text{ thay đổi} = (3) + (4) = \overline{Q_1} Q_2 Q_3 + Q_1 \overline{Q_2} \overline{Q_3}$$

$$T_2 = Q_2 = \sum \text{các cung có } Q_2 \text{ thay đổi} = (1) + (3) = \overline{Q_1} \overline{Q_2} Q_3 + \overline{Q_1} Q_2 Q_3$$

$$T_3 = Q_3 = \sum \text{các cung có } Q_3 \text{ thay đổi} = (0) + (1) + (2) + (3) = \overline{Q_1} \overline{Q_2} \overline{Q_3} + \overline{Q_1} \overline{Q_2} Q_3 + \overline{Q_1} Q_2 \overline{Q_3} + \overline{Q_1} Q_2 Q_3$$

$Q_2 Q_3 \backslash Q_1$	00	01	11	10
0			1	
1	1	x	x	x

$$T_1 = Q_1 + Q_2 Q_3$$

$Q_2 Q_3 \backslash Q_1$	00	01	11	10
0		1	1	
1		x	x	x

$$T_2 = Q_3$$

$Q_2 Q_3 \backslash Q_1$	00	01	11	10
0	1	1	1	1
1	x	x	x	x

$$T_3 = \overline{Q_1}$$

Bảng 5-10.

Lập bảng Karnaugh 5-10 cho các hàm trên ta thu được kết quả:

$$T_1 = Q_1 + Q_2 Q_3$$

$$T_2 = Q_3$$

$$T_3 = \overline{Q_1}$$

**c) Xác định phương trình kích cho Trigo JK.**

Chú ý khi viết các biểu thức  $T_{on}$ ,  $T_{off}$  của trigo thứ I ta cần phải đơn giản các biểu thức đó và đưa về dạng:

$$T_{on} = (T^*) \overline{Q_i} \Rightarrow \text{rút ra } J_i = T^*$$

$$T_{off} = (T^{**}) \overline{Q_i} \Rightarrow \text{rút ra } K_i = T^{**}$$

Viết các biểu thức  $T_{on}$ ,  $T_{off}$  cho các trigo và từ đó xác định phương trình kích cho các trigo như sau:

$$T_{on1} = \sum \text{Các cung mà } Q_1 \text{ được bật (Chuyển từ } 0 \Rightarrow 1) = (3) = \overline{Q_1} Q_2 Q_3$$

$$T_{off1} = \sum \text{Các cung mà } Q_1 \text{ tắt (Chuyển từ } 1 \Rightarrow 0) = (4) = Q_1 \overline{Q_2} \overline{Q_3}$$

$$T_{on2} = \sum \text{Các cung mà } Q_2 \text{ được bật (Chuyển từ } 0 \Rightarrow 1) = (1) = \overline{Q_1} \overline{Q_2} Q_3$$

$$T_{off2} = \sum \text{Các cung mà } Q_2 \text{ tắt (Chuyển từ } 1 \Rightarrow 0) = (3) = \overline{Q_1} Q_2 Q_3$$

$$T_{on3} = \sum \text{Các cung mà } Q_3 \text{ được bật (Chuyển từ } 0 \Rightarrow 1) = (0) + (2) = \overline{Q_1} \overline{Q_3}$$

$$T_{off3} = \sum \text{Các cung mà } Q_3 \text{ tắt (Chuyển từ } 1 \Rightarrow 0) = (1) + (3) = \overline{Q_1} Q_3$$

Biểu diễn các hàm này trên bảng Karnaugh, sử dụng các trạng thái tùy chọn để tối thiểu hoá. Các trạng thái tùy chọn bao gồm 3 số không nằm trong phạm vi đếm 5, 6, 7. Ngoài ra còn một số trạng thái khác tùy vào từng bảng. Ví dụ, đối với bảng tính  $J_1$  giá trị tùy chọn ngoài 3 số trên còn thêm ô có giá trị  $Q_1 = 1$ , bảng tính  $K_1$  có thêm các ô có giá trị  $Q_1 = 0$ , tương tự như vậy với các bảng còn lại.

$Q_2 Q_3$	00	01	11	10
$Q_1$			1	
0			1	
1	x	x	x	x

$$J_1 = Q_2 Q_3$$

$Q_2 Q_3$	00	01	11	10
$Q_1$				
0	x	x	x	x
1	1	x	x	x

$$K_1 = 1$$

$Q_2 Q_3$	00	01	11	10
$Q_1$				
0		1	x	
1		x	x	x

$$J_2 = Q_3$$

$Q_2 Q_3$	00	01	11	10
$Q_1$				
0	x	x	1	
1	x	x	x	x

$$K_2 = Q_3$$

$Q_2 Q_3$	00	01	11	10
$Q_1$				
0	1	x	x	1
1		x	x	x

$$J_3 = \overline{Q_1}$$

$Q_2 Q_3$	00	01	11	10
$Q_1$				
0	x	1	1	x
1	x	x	x	x

$$K_3 = 1$$

**Bảng 5-11.** Bảng tìm hàm kích

Ta thu được kết quả từ bảng 5-11 như sau:

$$J_1 = Q_2 Q_3; \quad K_1 = 1$$

$$J_2 = Q_3; \quad K_2 = Q_3$$

$$J_3 = \overline{Q_1}; \quad K_3 = 1$$

**d) Xác định phương trình kích cho Trigrơ RS.**

$$S_1 = T_{on1} + [\text{Các cung loại (1)}] = (3) + [\phi]$$

$$R_1 = T_{off1} + [\text{Các cung loại (0)}] = (4) + [(0), (1), (2)]$$

$$S_2 = T_{on2} + [\text{Các cung loại (1)}] = (1) + [(2)]$$

$$R_2 = T_{off2} + [\text{Các cung loại (0)}] = (3) + [(0), (4)]$$

$$S_3 = T_{on3} + [\text{Các cung loại (1)}] = (0) + (2) + [\phi]$$

$$R_3 = T_{off3} + [\text{Các cung loại (0)}] = (1) + (3) + [(4)]$$

Biểu diễn các hàm này trên bảng Karnaugh và tối thiểu hoá chúng.

$Q_2 Q_3 \backslash Q_1$	00	01	11	10
0			1	
1		x	x	x

$$S_1 = Q_2 Q_3$$

$Q_2 Q_3 \backslash Q_1$	00	01	11	10
0	x	x		x
1	1	x	x	x

$$R_1 = Q_1 \text{ hoặc } R_1 = \overline{Q_2} \text{ hoặc } R_1 = \overline{Q_3}$$

$Q_2 Q_3 \backslash Q_1$	00	01	11	10
0		1		x
1		x	x	x

$$S_2 = \overline{Q_2} Q_3$$

$Q_2 Q_3 \backslash Q_1$	00	01	11	10
0	x		1	
1	x	x	x	x

$$R_2 = Q_2 Q_3$$

$Q_2 Q_3 \backslash Q_1$	00	01	11	10
0	1			1
1		x	x	x

$$S_3 = \overline{Q_1} \overline{Q_3}$$

$Q_2 Q_3 \backslash Q_1$	00	01	11	10
0		1	1	
1	x	x	x	x

$$R_3 = \overline{Q_2} Q_3$$

**Bảng 5-12.** Bảng tìm hàm kích

Sau khi rút gọn từ bảng 5-12 ta thu được kết quả sau:

$$S_1 = Q_2 Q_3 \quad ; \quad R_1 = Q_1 \text{ hoặc } R_1 = \overline{Q_2} \text{ hoặc } R_1 = \overline{Q_3}$$

$$S_2 = \overline{Q_2} Q_3 \quad ; \quad R_2 = Q_2 Q_3$$

$$S_3 = \overline{Q_1} \overline{Q_3} \quad ; \quad R_3 = \overline{Q_2} Q_3$$

#### 5.4.2. Thiết kế mạch tuần tự từ bảng.

**Giả thiết:** Cho bảng chuyển đổi trạng thái, bảng ra của mạch (chưa mã hoá nhị phân).

**Xác định:** Hệ phương trình nhị phân của mạch vào gồm hệ hàm ra, hệ hàm kích cho các trigơ. Trên cơ sở đó vẽ sơ đồ mạch.

Các bước thực hiện:

- ◆ Mã hoá tín hiệu vào V, tín hiệu ra R, trạng thái trong của mạch S để chuyển mạch ban đầu thành mạch nhị phân có tập tín hiệu vào X, tập tín hiệu ra Y, tập trạng thái trong Q.
- ◆ Lập bảng chuyển đổi trạng thái, bảng ra của mạch nhị phân ứng với sự mã hoá trên.
- ◆ Dựa vào bảng các lỗi vào kích của các trigơ xác định các lỗi vào kích cho các trigơ ứng với sự chuyển đổi trong bảng trạng thái.
- ◆ Viết phương trình lỗi vào kích cho từng  $Q_i$  của trigơ và các hàm ra rồi tối thiểu các hàm này. Trên cơ sở đó xây dựng mạch điện.

Ví dụ: Thiết kế bộ đếm có  $K_d = 5$ , đồ hình trạng thái cho ở hình 5- 15a. Từ đó lập bảng chuyển đổi trạng thái như hình 5- 16a, mã hoá trạng thái như hình 5-16b. Dựa vào hai bảng này và căn cứ vào bảng hàm kích thích cho trigơ ở hình 5- 16c ta lập được bảng như hình 5- 16d. Từ đó xác định được các phương trình các lỗi vào kích cho các loại trigơ. Bảng Karnaugh và kết quả tối giản giống như ở mục 5.4.1.2.

S	S <sup>k</sup>
0	1
1	2
2	3
3	4
4	0

a) Bảng chuyển đổi trạng thái

Q <sub>2</sub> Q <sub>3</sub>	00	01	11	10
Q <sub>1</sub>	0	1	3	2
	0	1	x	x

b) Bảng mã hoá trạng thái

Q	Q <sup>k</sup>	D	T	RS	JK
0	0	0	0	X 0	0 X
0	1	1	1	0 1	1 X
1	0	0	0	1 0	X 1
1	1	1	1	0 X	X 0

c) Bảng hàm kích cho các trigơ

Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sup>k</sup> <sub>1</sub>	Q <sup>k</sup> <sub>2</sub>	Q <sup>k</sup> <sub>3</sub>	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	T <sub>1</sub>	T <sub>2</sub>	T <sub>3</sub>	R <sub>1</sub> S <sub>1</sub>	R <sub>2</sub> S <sub>2</sub>	R <sub>2</sub> S <sub>3</sub>	J <sub>1</sub> K <sub>1</sub>	J <sub>2</sub> K <sub>2</sub>	J <sub>3</sub> K <sub>3</sub>
0	0	0	0	0	1	0	0	1	0	0	1	X 0	X 0	0 1	0 X	0 X	1 X
0	0	1	0	1	0	0	1	0	0	1	1	X 0	0 1	1 0	0 X	1 X	X 1
0	1	0	0	1	1	0	1	1	0	0	1	X 0	0 X	0 1	0 X	X 0	1 X
0	1	1	1	0	0	1	0	0	1	1	1	0 1	1 0	1 0	1 X	X 1	X 1
1	0	0	0	0	0	0	0	0	1	0	0	1 0	X 0	X 0	X 1	0 X	0 X
1	0	1	X	X	X	X	X	X	X	X	X	X X	X X	X X	X X	X X	X X
1	1	0	X	X	X	X	X	X	X	X	X	X X	X X	X X	X X	X X	X X
1	1	1	X	X	X	X	X	X	X	X	X	X X	X X	X X	X X	X X	X X

d) Bảng trạng thái nhị phân và đầu vào kích cho các loại trigơ

Hình 5-16. (a), (b), (c), (d) : Các bước thiết kế mạch tuần tự

### 5.5 MẠCH TUẦN TỰ ĐỒNG BỘ

Phần này trình bày phương pháp cơ bản để phân tích và thiết kế mạch tuần tự đồng bộ. Mạch tuần tự đồng bộ là một mạch số bao gồm các mạch tổ hợp và các phần tử nhớ (trigơ), hoạt động của mạch được đồng bộ bởi xung nhịp C. Trên thực tế để giảm nhỏ công suất tiêu thụ, thời gian trễ và để cho các mạch thực hiện đơn giản, người ta thường thiết kế sơ đồ sử dụng các trigơ JK và các mạch NAND.

Để nắm vững các vấn đề thiết kế mạch tuần tự đồng bộ, trước hết ta sẽ đi phân tích mạch tuần tự.

### 5.5.1. Phân tích mạch tuần tự đồng bộ.

#### 5.5.1.1. Các bước phân tích mạch tuần tự đồng bộ.

Bài toán phân tích là bài toán xác định chức năng của một mạch cho trước. Khi tiến hành phân tích cần tuân theo các bước sau:

- **Sơ đồ mạch:** Từ sơ đồ cho trước cần xác định chức năng từng phần tử cơ bản của sơ đồ, mối quan hệ giữa các phần tử đó.

- **Xác định các đầu vào và ra, số trạng thái trong của mạch:** Coi mạch như một hộp đen cần phải xác định các đầu vào và ra của mạch, đặc điểm của các đầu vào, đầu ra. Để xác định được số trạng thái trong của mạch cần phải xác định xem mạch được xây dựng từ bao nhiêu phần tử nhớ (trigơ JK) từ đó xác định được số trạng thái trong có thể có của mạch.

Gọi số trigơ là  $n$  thì số trạng thái có thể có của mạch là  $2^n$ .

- **Xác định phương trình hàm ra, phương trình hàm kích của các trigơ.**

- **Lập bảng trạng thái, bảng ra nhị phân** là bảng biểu diễn mối quan hệ trạng thái kế tiếp, tín hiệu ra nhị phân với trạng thái hiện tại và các tín hiệu vào tương ứng.

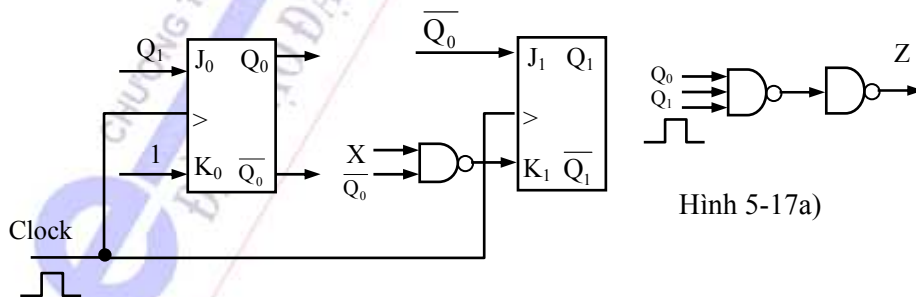
Dựa vào phương trình đặc trưng của trigơ xác định được trạng thái kế tiếp và tín hiệu ra tương ứng với tín hiệu vào và trạng thái hiện tại của mạch.

- **Đồ hình trạng thái:** Từ bảng trạng thái xây dựng đồ hình trạng thái và tín hiệu ra của mạch.

- **Chức năng của mạch:** Dựa vào đồ hình trạng thái xác định được chức năng của mạch

#### 5.5.1.2. Ví dụ.

Phân tích mạch tuần tự đồng bộ có sơ đồ được biểu diễn như hình 5-17a.



**Bước 1.** Sơ đồ trên có hai đầu vào là tín hiệu X và xung nhịp Clock. Có một tín hiệu Z ra, mạch sử dụng hai phần tử nhớ là hai trigơ JK ( $Q_0$  và  $Q_1$ ).

**Bước 2:** Xác định đầu vào, đầu ra và số trạng thái trong của mạch.

Mạch này có thể được biểu diễn bằng một “hộp đen” có hai đầu vào và một đầu ra. Do mạch được cấu tạo bằng hai trigơ nên số trạng thái có thể có của mạch là 4. Cụ thể là:

$$Q_1Q_0 = 00, 01, 10 \text{ và } 11.$$

Bước 3: Xác định phương trình hàm ra và hàm kích cho trigơ.

Từ sơ đồ trên ta tìm được:

+ Phương trình hàm ra:

$$Z = C Q_1 Q_0$$

+ Phương trình hàm kích

$$J_0 = Q_1; K_0 = 1$$

$$J_1 = \overline{Q_0}; K_1 = \overline{X Q_0} = \overline{X} + Q_0$$

Bước 4. Bảng chuyển đổi trạng thái.

Phương trình đặc trưng của trigơ JK là  $Q^k = J\overline{Q} + \overline{K}Q$

Phương trình chuyển đổi trạng thái:

$$Q_0^k = J_0 \overline{Q_0} + \overline{K_0} Q_0 = Q_1 \overline{Q_0}$$

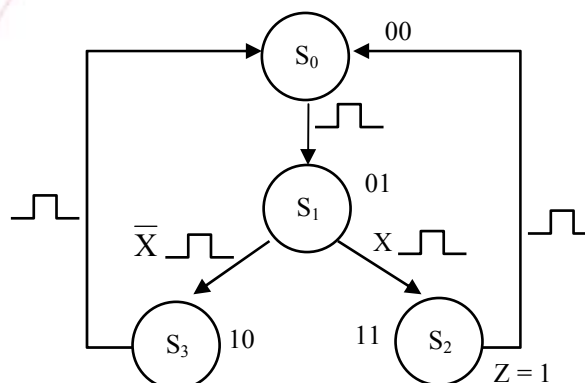
$$Q_1^k = J_1 \overline{Q_1} + \overline{K_1} Q_1 = \overline{Q_0} \overline{Q_1} + \overline{\overline{X} + Q_0} Q_1 = \overline{Q_0} \overline{Q_1} + X \overline{Q_0} Q_1$$

Từ các phương trình trên ta lập được bảng chuyển đổi trạng thái

	Trạng thái hiện tại	Trạng thái kế tiếp		Tín hiệu ra	
		X = 0	X = 1	X = 0	X = 1
	$Q_0 Q_1$	$Q_0 Q_1$	$Q_0 Q_1$	Z	Z
$S_0$	00	01	01	0	0
$S_1$	01	10	11	0	0
$S_2$	11	00	00	1	1
$S_3$	10	00	00	0	0

Hình 5-17 b). Bảng chuyển đổi trạng thái

Bước 5: Đồ hình trạng thái. Từ bảng chuyển đổi trạng thái trên ta xây dựng được đồ hình trạng thái như hình 5-17 c) (mô hình Mealy). Đồ hình gồm 4 trạng thái trong  $S_0, S_1, S_2, S_3$ . Các trigơ JK hoạt động tại sườn âm của xung nhịp. Nhìn vào đồ hình trạng thái ta thấy ở trạng thái trong  $S_2$  ( $Q_0 Q_1 = 11$ ) khi có xung nhịp C thì mạch sẽ đưa ra tín hiệu  $Z = 1$ .



Hình 5-17 c). Đồ hình trạng thái

Bước 6: Chức năng của mạch: Trên đồ hình trạng thái ta thấy có hai đường chuyển đổi trạng thái là  $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_0$  và  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0$ . Theo đường  $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_0$  thì tín hiệu ra  $Z = 1$  sẽ được đưa ra cùng thời điểm có xung nhịp thứ 3. Theo đường  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0$  thì không có tín hiệu ra ( $Z = 0$ ). Do vậy ta sẽ phân tích theo con đường thứ nhất  $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_0$ : Sự chuyển đổi trạng thái đầu tiên từ  $S_0 \rightarrow S_1$  chỉ nhờ tác động của xung nhịp mà không phụ

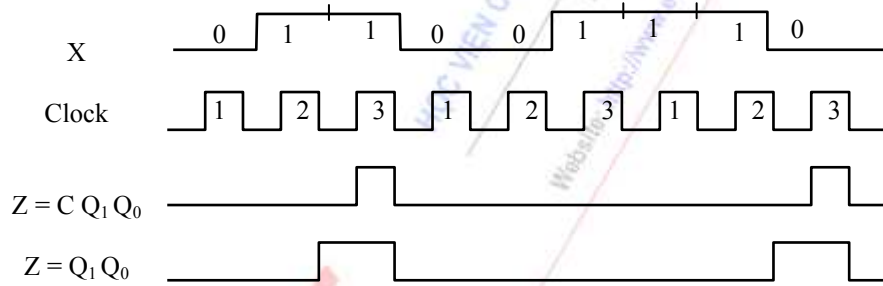


thuộc vào trạng thái của X. Chuyển đổi trạng thái thứ hai từ  $S_1 \rightarrow S_2$  nhờ tác động của xung nhịp và sự tác động của tín hiệu vào  $X = 1$ . Còn sự chuyển đổi trạng thái thứ ba từ  $S_2 \rightarrow S_0$  chỉ nhờ tác động của xung nhịp mà không phụ thuộc vào tín hiệu vào.

Như vậy, mạch chỉ đưa ra tín hiệu ra  $Z = 1$  khi đường chuyển đổi đi qua  $S_2$  tức là mạch chỉ đưa ra tín hiệu ra  $Z = 1$  khi dãy tín hiệu vào X có dạng 010, 011, 110 và 111. Có thể biểu diễn dãy tín hiệu vào để mạch có tín hiệu ra  $Z = 1$  như sau:

$$\begin{array}{cc} 0 & 1 \\ \downarrow & \downarrow \\ 0 \leftarrow 1 \rightarrow 1 & 0 \leftarrow 1 \rightarrow 1 \end{array}$$

Tóm lại, mạch cho ở sơ đồ trên có chức năng kiểm tra dãy tín hiệu vào X ở dạng chuỗi có độ dài bằng 3. Nếu chuỗi tín hiệu vào có dạng là 1 trong 4 dãy: 010, 011, 110 và 111 mạch sẽ cho tín hiệu ra  $Z = 1$  tại thời điểm có xung nhịp thứ 3. Độ rộng của tín hiệu ra Z bằng độ rộng xung nhịp ( $Z = C Q_1 Q_0$ ).



Hình 5-17d) Dạng xung ra của mạch

Hình 5-17 a, b, c, d. Phân tích mạch tuần tự đồng bộ

### 5.5.2. Thiết kế mạch tuần tự đồng bộ.

#### 5.5.2.1. Các bước thiết kế mạch tuần tự đồng bộ.

*Bước 1:* Xác định bài toán, gán hàm và biến, tìm hiểu mối quan hệ giữa chúng.

*Bước 2:* Xây dựng đồ hình trạng thái, bảng chuyển đổi trạng thái và hàm ra.

*Bước 3:* Rút gọn trạng thái (tối thiểu hoá trạng thái).

Việc tối thiểu hoá trạng thái chủ yếu dựa vào khái niệm trạng thái tương đương. Các trạng thái tương đương với nhau có thể được thay bằng một trạng thái chung đại diện cho chúng.

*Bước 4:* Mã hoá trạng thái.

Số biến nhị phân dùng để mã hoá các trạng thái trong của mạch phụ thuộc vào số lượng trạng thái trong của mạch. Nếu số lượng trạng thái trong là N, số biến nhị phân cần dùng là n thì n phải thoả mãn điều kiện:  $n \geq \log_2 N$ .

Có rất nhiều cách mã hoá khác nhau, mỗi cách cho một sơ đồ thực hiện mạch khác nhau. Vấn đề là phải mã hoá sao cho sơ đồ mạch thực hiện là đơn giản nhất.

*Bước 5:* Xác định hệ phương trình của mạch. Có hai cách xác định:

+ Lập bảng chuyển đổi trạng thái và tín hiệu ra, từ đó xác định các phương trình kích cho các trigơ.

+ Dựa trực tiếp vào đồ hình trạng thái, viết hệ phương trình  $T_{on}$ ,  $T_{off}$  của các trigơ và phương trình hàm ra.

Bước 6: Vẽ sơ đồ thực hiện.

### 5.5.2.2. Ví dụ.

Thiết kế mạch tuần tự thực hiện nhiệm vụ kiểm tra dãy tín hiệu vào ở dạng nhị phân có độ dài bằng 3 được đưa vào liên tiếp trên đầu vào X. Nếu dãy tín hiệu vào có dạng là 010 hoặc 011 hoặc 110 hoặc 111 thì  $Z = 1$ . Các trường hợp khác  $Z = 0$ .

Bước 1: Xác định bài toán. Mạch được thiết kế có nhiệm vụ phát hiện tín hiệu vào. Khi nhận được 1 trong các dãy tín hiệu trên thì mạch sẽ báo rằng đã nhận được.

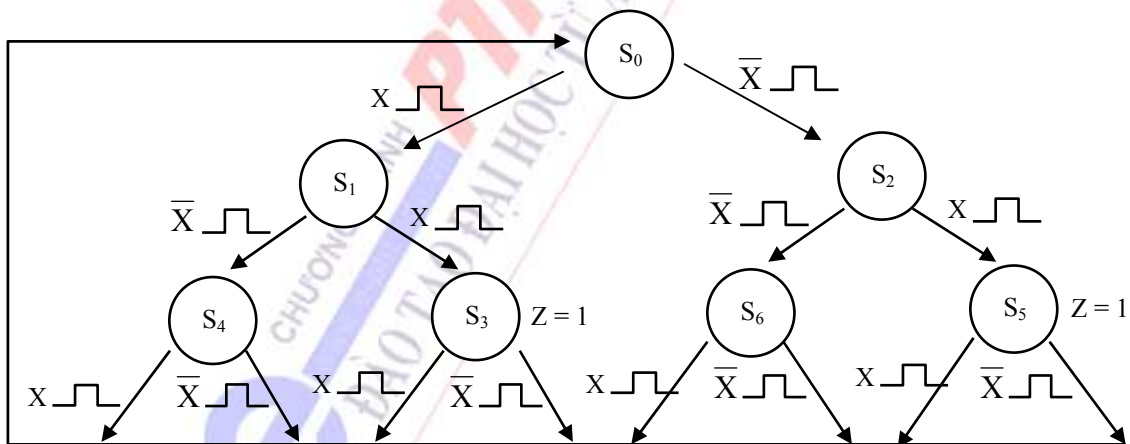
Mạch phải thiết kế là mạch đồng bộ, nên sẽ có các lối vào là X- tín hiệu vào, Ck- xung nhịp điều khiển, Z – tín hiệu ra.

Bước 2: Xây dựng đồ hình trạng thái, bảng chuyển đổi trạng thái

Giả sử trạng thái ban đầu là  $S_0$ :

Khi tín hiệu vào là X. Ck thì mạch sẽ chuyển tới trạng thái  $S_1$ . Khi tín hiệu vào là  $\bar{X}$ . Ck mạch sẽ chuyển đến trạng thái  $S_2$ .

Tương tự như vậy. Khi mạch ở trạng thái  $S_1$  thì khi có tín hiệu X. Ck mạch chuyển đến trạng thái  $S_3$  và chuyển đến trạng thái  $S_4$  khi có tín hiệu  $\bar{X}$ . Ck. Tương tự ta xây dựng được đồ hình sau 5-18 a.



Hình 5-18 a). Đồ hình trạng thái

Nếu mạch ở 1 trong 4 trạng thái  $S_3, S_4, S_5, S_6$ : khi có tín hiệu vào X. Ck hoặc  $\bar{X}$ . Ck thì mạch sẽ chuyển về trạng thái ban đầu  $S_0$ . Khi dãy tín hiệu vào là 110 hoặc 111 (ứng với đường chuyển đổi trạng thái là  $S_0 \rightarrow S_1 \rightarrow S_3 \rightarrow S_0$ ) hay khi dãy tín hiệu vào là 010 hoặc 011 (ứng với đường chuyển đổi trạng thái là  $S_0 \rightarrow S_2 \rightarrow S_5 \rightarrow S_0$ ) thì mạch sẽ cho tín hiệu ra  $Z = 1$  tại thời điểm xung thứ 3. Với các đường chuyển đổi khác  $Z = 0$ .

Từ đồ hình trạng thái ta xây dựng được bảng chuyển đổi trạng thái như sau:

S	S <sup>k</sup>		Z	
	X = 0	X = 1	X = 0	X = 1
S <sub>0</sub>	S <sub>2</sub>	S <sub>1</sub>	0	0
S <sub>1</sub>	S <sub>4</sub>	S <sub>3</sub>	0	0
S <sub>2</sub>	S <sub>6</sub>	S <sub>5</sub>	0	0
S <sub>3</sub>	S <sub>0</sub>	S <sub>0</sub>	1	1
S <sub>4</sub>	S <sub>0</sub>	S <sub>0</sub>	0	0
S <sub>5</sub>	S <sub>0</sub>	S <sub>0</sub>	1	1
S <sub>6</sub>	S <sub>0</sub>	S <sub>0</sub>	0	0

Hình 5-18b). Bảng chuyển đổi trạng thái

**Bước 3:** Tối thiểu hoá trạng thái. Để có được sơ đồ mạch đơn giản ta phải tối thiểu hoá các trạng thái. Trong phần này sẽ giới thiệu phương pháp tối thiểu hoá Caldwell. Cơ sở lý thuyết của việc tối thiểu hoá là dựa vào khái niệm các trạng thái tương đương.

**Định nghĩa các trạng thái tương đương:**

Trạng thái S<sub>i</sub> được gọi là trạng thái tương đương với trạng thái S<sub>j</sub> (S<sub>i</sub> ≈ S<sub>j</sub>) khi và chỉ khi: nếu lấy S<sub>i</sub> và S<sub>j</sub> là hai trạng thái ban đầu thì với mọi dãy tín hiệu vào có thể chúng luôn cho dãy tín hiệu ra giống nhau.

Nếu có nhiều trạng thái tương đương với nhau từng đôi một thì chúng tương đương với nhau (tính chất bắc cầu). Để kiểm tra một nhóm các trạng thái xem chúng có tương đương với nhau không, có thể sử dụng bảng trạng thái và tín hiệu ra như sau:

- Nhóm các trạng thái tương đương phải có những hàng trong bảng tín hiệu ra giống nhau.
- Nhóm các trạng thái tương đương phải có những hàng trong bảng trạng thái ở cùng một cột (ứng với cùng một tổ hợp tín hiệu vào) là tương đương. Nghĩa là ứng với cùng một tổ hợp tín hiệu vào các trạng thái kế tiếp của chúng là tương đương.

**Quy tắc Caldwell:**

Những hàng (tương ứng với trạng thái trong) của bảng chuyển đổi trạng thái và tín hiệu ra sẽ được kết hợp với nhau và được biểu diễn bằng một hàng chung - đặc trưng (trạng thái đặc trưng) cho chúng nếu như chúng thoả mãn hai điều kiện sau:

1. Các hàng tương ứng trong ma trận ra giống nhau.
2. Trong ma trận ra, các hàng tương ứng phải thoả mãn 1 trong 3 điều sau:
  - Các hàng trong ma trận trạng thái giống nhau.
  - Các trạng thái ở trong cùng một cột nằm trong nhóm trạng thái được xét.
  - Các trạng thái ở trong cùng một cột là các trạng thái tương đương.

Sau khi đã thay thế các trạng thái tương đương bằng một trạng thái chung đặc trưng cho chúng, lặp lại các công việc tìm các trạng thái tương đương khác cho đến khi không thể tìm được

các trạng thái tương đương nào nữa thì dừng lại. Số trạng thái trong bảng chuyển đổi trạng thái là tối thiểu.

Nhược điểm của phương pháp này là khi số trạng thái quá lớn thì công việc tối thiểu hoá mất nhiều thời gian.

Áp dụng quy tắc Caldwell cho bài toán trên ta thấy trạng thái  $S_4$  tương đương với trạng thái  $S_6$  ( $S_4 \approx S_6$ ),  $S_3$  tương đương với  $S_5$  ( $S_3 \approx S_5$ ). Thay thế các trạng thái tương đương bằng một trạng thái chung đặc trưng cho chúng. Ví dụ thay thế  $S_4, S_6$  bằng  $S_{46}$ , thay thế  $S_3, S_5$  bằng  $S_{35}$ . Từ đó lập được bảng chuyển đổi trạng thái 5-18c) và 5-18 d):

X \ S	0	1
$S_0$	$S_2$ $Z = 0$	$S_1$ $Z = 0$
$S_1$	$S_{46}$ $Z = 0$	$S_{35}$ $Z = 0$
$S_2$	$S_{46}$ $Z = 0$	$S_{35}$ $Z = 0$
$S_{35}$	$S_0$ $Z = 1$	$S_0$ $Z = 1$
$S_{46}$	$S_0$ $Z = 0$	$S_0$ $Z = 0$

Hình 5-18c) Bảng chuyển đổi trạng thái sau khi gộp  $S_3$  và  $S_5$ ,  $S_4$  và  $S_6$

X \ S	0	1
$S_0$	$S_{12}$ $Z = 0$	$S_{12}$ $Z = 0$
$S_{12}$	$S_{46}$ $Z = 0$	$S_{35}$ $Z = 0$
$S_{35}$	$S_0$ $Z = 1$	$S_0$ $Z = 1$
$S_{46}$	$S_0$ $Z = 0$	$S_0$ $Z = 0$

Hình 5-18d) Bảng chuyển đổi trạng thái sau khi gộp  $S_1$  và  $S_2$

Bước 4: Sau khi gộp hai trạng thái  $S_1$  và  $S_2$  thành trạng thái chung  $S_{12}$  thì mạch chỉ còn 4 trạng thái  $S_0, S_{12}, S_{35}, S_{46}$ . Mã hoá 4 trạng thái này bằng hai biến nhị phân  $Q_1$  và  $Q_0$ .

$Q_0$	$Q_1$	Mã hoá S
0	0	$S_0$
0	1	$S_{12}$
1	1	$S_{35}$
1	0	$S_{46}$

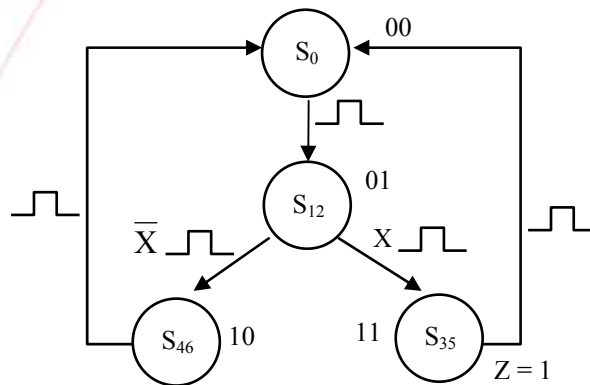
Hình 5-18 e) Bảng mã hoá trạng thái

Bước 5: Xác định hệ phương trình của mạch.

Có hai cách xác định hệ phương trình này.

Cách 1:

Dựa vào bảng chuyển đổi trạng thái ta lập bảng hàm kích 5-13 cho hai trigơ  $Q_0$  và  $Q_1$ .



Hình 5-18f). Đồ hình trạng thái tối giản

Dùng bảng Karnaugh 5-14 để rút gọn, ta thu được kết quả sau:

$$J_0 = Q_1 ; K_0 = 1$$

$$J_1 = \overline{Q_0} ; K_1 = \overline{X} + Q_0$$

$$Z = X Q_0 Q_1$$

Trạng thái hiện tại	Trạng thái kế tiếp		Các đầu vào của trigơ							
	X = 0	X = 1	X = 0		X = 1		X = 0		X = 1	
$Q_0 Q_1$	$Q_0 Q_1$	$Q_0 Q_1$	$J_0$	$K_0$	$J_0$	$K_0$	$J_1$	$K_1$	$J_1$	$K_1$
00	01 $Z = 0$	01 $Z = 0$	0	X	0	X	1	X	1	X
01	10 $Z = 0$	11 $Z = 0$	1	X	1	X	X	1	X	0
11	00 $Z = 1$	00 $Z = 1$	X	1	X	1	X	1	X	1
10	00 $Z = 0$	00 $Z = 0$	X	1	X	1	0	X	0	X

Bảng 5-13. Bảng hàm kích thích

$Q_0 Q_1$	00	01	11	10
X				
0		1	x	x
1			x	x

$$J_0 = Q_1$$

$Q_0 Q_1$	00	01	11	10
X				
0	x	x	1	x
1	x	x	1	x

$$K_0 = 1$$

$Q_0 Q_1$	00	01	11	10
X				
0	1	x	x	
1	1	x	x	

$$J_1 = \overline{Q_0}$$

$Q_0 Q_1$	00	01	11	10
X				
0	x	1	1	x
1	x		1	x

$$K_1 = \overline{X} + Q_0$$

$Q_0 Q_1$	00	01	11	10
X				
0			1	
1			1	

$$Z = X Q_0 Q_1$$

Bảng 5-14. Bảng tính hàm kích

Cách 2: Dựa trực tiếp vào đồ hình trạng thái viết phương trình  $T_{on}$ ,  $T_{off}$  của từng trigơ và phương trình tín hiệu ra.

Đối với trigơ JK nếu:

$$T_{onQ} = T * \bar{Q} \Rightarrow J_Q = T *$$

$$T_{offQ} = T ** Q \Rightarrow K_Q = T **$$

Đối với trường hợp này ta có:

$$T_{onQ_0} = S_{12}X + S_{12}\bar{X} = S_{12} = \bar{Q}_0Q_1 \Rightarrow J_0 = Q_1$$

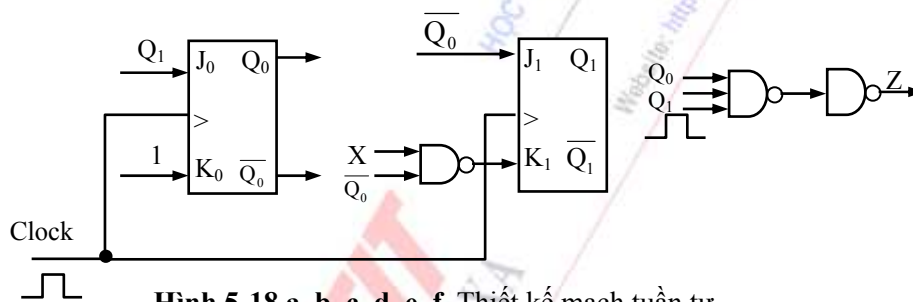
$$T_{offQ_0} = S_{35} + S_{46} = Q_0Q_1 + Q_0\bar{Q}_1 = Q_0 \Rightarrow K_0 = 1$$

$$T_{onQ_1} = S_0X = \bar{Q}_0\bar{Q}_1 \Rightarrow J_1 = \bar{Q}_0$$

$$T_{offQ_1} = S_{12}\bar{X} + S_{35} = \bar{Q}_0Q_1\bar{X} + Q_0Q_1 = Q_1(\bar{Q}_0\bar{X} + Q_0) \Rightarrow K_1 = \bar{Q}_0\bar{X} + Q_0 = \bar{X} + Q_0$$

Phương trình hàm ra  $Z = Q_0Q_1Ck$

Bước 6: Sơ đồ mạch điện:



Hình 5-18 a, b, c, d, e, f. Thiết kế mạch tuần tự

## 5.6. MẠCH TUẦN TỰ KHÔNG ĐỒNG BỘ

Phần 5.6 đã nghiên cứu các mạch tuần tự đồng bộ, hoạt động của chúng được điều khiển bởi các xung nhịp. Nhưng trên thực tế có nhiều mạch lại được điều khiển bởi các sự kiện mà không tuân theo một quy luật nào cả. Ví dụ một hệ thống chống trộm sẽ chỉ hoạt động khi có trộm. Những mạch tuần tự hoạt động theo kiểu như vậy gọi là mạch tuần tự không đồng bộ.

Mạch tuần tự không đồng bộ có thể thiết kế:

- Chỉ dùng những mạch NAND.
- Dùng trigơ RS không đồng bộ và các mạch NAND.

Việc thiết kế mạch tuần tự không đồng bộ dùng các trigơ loại không đồng bộ khác hoàn toàn tương tự.

### 5.6.1. Các bước thiết kế mạch tuần tự không đồng bộ

Bước 1: Xác định bài toán, gán hàm và biến, tìm hiểu mối quan hệ giữa chúng.

Bước 2: Xây dựng đồ hình trạng thái, bảng chuyển đổi trạng thái và hàm ra.

Bước 3: Rút gọn trạng thái (tối thiểu hoá trạng thái).

Việc tối thiểu hoá trạng thái chủ yếu dựa vào khái niệm trạng thái tương đương. Các trạng thái tương đương với nhau có thể được thay bằng một trạng thái chung đại diện cho chúng.

*Bước 4:* Mã hoá trạng thái.

Số biến nhị phân dùng để mã hoá các trạng thái trong của mạch phụ thuộc vào số lượng trạng thái trong của mạch. Nếu số lượng trạng thái trong là  $N$ , số biến nhị phân cần dùng là  $n$  thì  $n$  phải thoả mãn điều kiện:  $n \geq \log_2 N$ .

Có rất nhiều cách mã hoá khác nhau, mỗi cách cho một sơ đồ thực hiện mạch khác nhau. Vấn đề là phải mã hoá sao cho sơ đồ mạch thực hiện là đơn giản nhất.

Do mạch không đồng bộ hoạt động không có sự tác động của xung nhịp cho nên trong mạch thường có các hiện tượng chạy đua làm cho hoạt động của mạch bị sai, vì vậy khi mã hoá trạng thái phải tránh hiện tượng này.

*Bước 5:* Xác định hệ phương trình của mạch. Có hai cách xác định:

+ Lập bảng chuyển đổi trạng thái và tín hiệu ra, từ đó xác định các phương trình kích cho các trigơ.

+ Dựa trực tiếp vào đồ hình trạng thái, viết hệ phương trình  $T_{on}$ ,  $T_{off}$  của các trigơ và phương trình hàm ra.

Cả hai cách này đều có hai loại phương trình:

- Phương trình của mạch chỉ dùng NAND.
- Phương trình của mạch dùng trigơ RS không đồng bộ và các mạch NAND

*Bước 6:* Vẽ sơ đồ thực hiện.

Sau đây là nội dung của từng phương pháp.

**Cách 1: Dựa vào bảng chuyển đổi trạng thái.**

a) Chỉ dùng các mạch NAND

Ký hiệu :  $A, B, \dots, N$  là các biến nhị phân dùng để mã hoá các trạng thái trong của mạch.

$X_1, X_2, \dots, X_m$  là các tín hiệu vào đã được mã hoá nhị phân.

$Z_1, Z_2, \dots, Z_m$  là các tín hiệu ra đã được mã hoá nhị phân.

Dựa vào bảng chuyển đổi trạng thái xác định hệ phương trình:

$$A^k = f_A(A, B, \dots, N, X_1, X_2, \dots, X_m)$$

$$B^k = f_B(A, B, \dots, N, X_1, X_2, \dots, X_m)$$

.....

$$N^k = f_N(A, B, \dots, N, X_1, X_2, \dots, X_m)$$

$$Z_1 = g_1(A, B, \dots, N, X_1, X_2, \dots, X_m)$$

$$Z_2 = g_2(A, B, \dots, N, X_1, X_2, \dots, X_m)$$

.....

$$Z_n = g_n(A, B, \dots, N, X_1, X_2, \dots, X_m)$$

Tối thiểu hoá hệ hàm và viết phương trình ở dạng chỉ dùng NAND.

b) Mạch dùng trigơ RS và các mạch NAND

Trong bảng trạng thái căn cứ vào sự thay đổi trạng thái của từng trigơ:

$A \Rightarrow A^k, B \Rightarrow B^k, \dots, N \Rightarrow N^k$ , xác định được giá trị tương ứng của đầu vào kích R, S cho từng trigơ, từ đó viết được hệ phương trình:

$$R_A = \Phi_1(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$S_A = \Phi_2(A, \dots, N, X_1, X_2, \dots, X_m)$$

Tối thiểu hoá các hàm và viết phương trình ở dạng chỉ dùng NAND.

Tương tự với B, C, ..., N cũng như vậy.

Ta xác định tín hiệu ra :

$$Z = \Phi(A, \dots, N, X_1, X_2, \dots, X_m)$$

Tối thiểu hoá và viết phương trình ở dạng chỉ dùng NAND.

**Cách 2: Dựa trực tiếp vào đồ hình trạng thái**

Ta có phương trình đầu vào kích (R, S) của trigơ A là:

$$S_A = \text{tập hợp bật của } A + [(1)]$$

$$R_A = \text{tập hợp tắt của } A + [(0)]$$

Làm tương tự với các trigơ khác.

a) Chỉ dùng mạch NAND

Ta có phương trình đặc trưng của trigơ RS

$$Q^k = S + \bar{R}Q \Rightarrow A^k = S_A + \bar{R}_A A$$

Sau đó ta phải tối thiểu hoá phương trình và viết dưới dạng chỉ dùng NAND. Đối với các trigơ khác cũng làm như vậy.

b) Dùng các trigơ RS không đồng bộ và các mạch NAND

$$R_A = \Phi_{1A}(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$S_A = \Phi_{2A}(A, \dots, N, X_1, X_2, \dots, X_m)$$

.....

$$R_N = \Phi_{1N}(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$S_N = \Phi_{2N}(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$Z_1 = \Psi_1(A, \dots, N, X_1, X_2, \dots, X_m)$$

$$Z_2 = \Psi_2(A, \dots, N, X_1, X_2, \dots, X_m)$$

.....

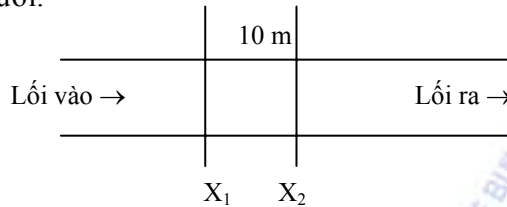


$$Z_n = \Psi_n(A, \dots, N, X_1, X_2, \dots, X_m)$$

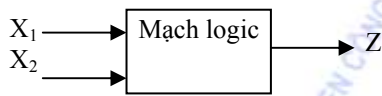
Tối thiểu hoá hệ phương trình.

### 5.6.2. Ví dụ

Một mạch tuần tự không đồng bộ được thiết kế để đếm số người vào thăm một viện bảo tàng. Mạch gồm hai đèn  $X_1, X_2$  được bố trí cách nhau 10 mét. Mạch được thiết kế sao cho mỗi lần chỉ đếm được một người.



Hình 5-19 a) Bố trí các đèn ở cửa vào viện bảo tàng



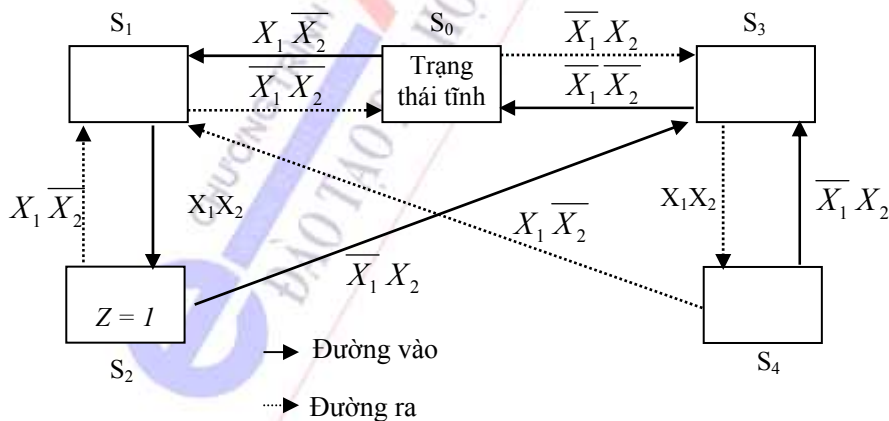
Hình 5-19 b) Sơ đồ khối của mạch

Khi có một người đi vào thì hai đèn sẽ bị chẵn liên tiếp. Đầu tiên  $X_1$  bị chẵn, tiếp đến cả  $X_1$  và  $X_2$  cùng bị chẵn, sau đó đến  $X_2$  bị chẵn. Khi đó mạch cho ra tín hiệu  $Z = 1$ . Khi một người ra thì sẽ ngược lại. Đầu tiên đèn  $X_2$  sẽ bị chẵn, sau đó cả  $X_1$  và  $X_2$  cùng bị chẵn và cuối cùng chỉ có  $X_1$  bị chẵn. Sơ đồ khối của mạch tạo tín hiệu đếm  $Z$  được mô tả bởi hình 5-19b.

Hai lối vào của mạch là  $X_1, X_2$ . Đầu ra  $Z$  được đưa tới lối vào của bộ giải mã.

Ta quy ước: đèn bị chẵn =  $X$ ; ngược lại thì =  $\bar{X}$ .

Đồ hình trạng thái được mô tả ở hình 5-19c.



Hình 5-19 c) Đồ hình trạng thái

$S_0$  là trạng thái ban đầu của mạch. Nếu một người đi vào thì sự chuyển đổi của mạch sẽ là  $S_0 \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_0$ . Nếu một người đi ra thì quá trình chuyển đổi trạng thái của mạch là  $S_0 \rightarrow S_3 \rightarrow S_4 \rightarrow S_1 \rightarrow S_0$ . Khi có một người ngập ngừng sau đó lại quay ra ban đầu chẵn đèn  $X_1$  sau đó quay ra thì mạch sẽ chuyển đổi trạng thái  $S_0 \rightarrow S_1 \rightarrow S_0$ , lúc đó mạch sẽ không thực hiện đếm.

Tương ứng với đồ hình trạng thái trên ta lập được bảng chuyển đổi trạng thái ở hình 5-19d:

Trạng thái hiện tại	Trạng thái kế tiếp và tín hiệu ra							
	X <sub>1</sub>	X <sub>2</sub>	X <sub>1</sub>	X <sub>2</sub>	X <sub>1</sub>	X <sub>2</sub>	X <sub>1</sub>	X <sub>2</sub>
	0	0	0	1	1	1	1	0
S <sub>0</sub>	(S <sub>0</sub> ) / Z = 0		S <sub>3</sub> / Z = 0				S <sub>1</sub> / Z = 0	
S <sub>1</sub>	S <sub>0</sub> Z = 0				S <sub>2</sub> Z = 0		(S <sub>1</sub> ) / Z = 0	
S <sub>2</sub>			S <sub>3</sub> / Z = 0		(S <sub>2</sub> ) / Z = 1		S <sub>1</sub> / Z = 0	
S <sub>3</sub>	S <sub>0</sub> / Z = 0		(S <sub>3</sub> ) / Z = 0		S <sub>4</sub> / Z = 0			
S <sub>4</sub>			S <sub>3</sub> / Z = 0		(S <sub>4</sub> ) / Z = 0		S <sub>1</sub> / Z = 0	

**Hình 5-19 d) Bảng chuyển đổi trạng thái và hàm ra**

Bảng có 5 hàng ứng với 5 trạng thái hiện tại có thể xuất hiện và 4 cột, mỗi cột ứng với một tổ hợp giá trị có thể của  $X_1, X_2$ . Mỗi ô của bảng biểu diễn trạng thái kế tiếp và tín hiệu ra tương ứng với trạng thái hiện tại và giá trị của tín hiệu vào  $X_1, X_2$ .

Trong bảng chuyển đổi trạng thái, những ô được khoanh tròn là những ô có trạng thái kế tiếp bằng trạng thái hiện tại. Những trạng thái đó là những trạng thái ổn định. Điều kiện cho trạng thái ổn định là  $S^k = S$ .

Trên bảng có những ô trống. Những ô này tương ứng với các tổ hợp tín hiệu không xuất hiện ở đầu vào. Những ô này có thể điền giá trị tùy chọn để tối thiểu hoá hệ phương trình của mạch.

Tiến hành tối thiểu hoá:

Có thể gán trạng thái kế tiếp và tín hiệu ra vào các ô trống sao cho hàng có ô trống có thể kết hợp với các hàng khác.

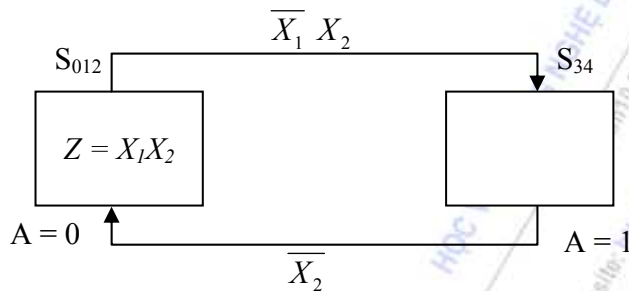
Ở bảng chuyển đổi trạng thái các hàng S<sub>0</sub>, S<sub>1</sub>, S<sub>2</sub>, và S<sub>3</sub>, S<sub>4</sub> có các trạng thái kế tiếp và tín hiệu ra tương ứng là giống nhau nếu như ta gán:

- ô trống của hàng đầu tiên (ứng với S<sub>0</sub>) là S<sub>2</sub> / Z = 1,
- ô trống của hàng thứ hai là S<sub>3</sub> / Z = 0,
- ô trống của hàng thứ tư là S<sub>1</sub> / Z = 0,
- ô trống của hàng thứ ba và thứ năm là S<sub>0</sub> / Z = 0,

Khi đó bảng chuyển đổi trạng thái được rút gọn lại như sau:

Trạng thái hiện tại	Trạng thái kế tiếp và tín hiệu ra							
	$X_1$	$X_2$	$X_1$	$X_2$	$X_1$	$X_2$	$X_1$	$X_2$
	0	0	0	1	1	1	0	0
$S_{012}$	$S_{012}$	$Z=0$	$S_{34}$	$Z=0$	$S_{012}$	$Z=0$	$S_{012}$	$Z=0$
$S_{34}$	$S_{012}$	$Z=0$	$S_{34}$	$Z=0$	$S_{34}$	$Z=0$	$S_{012}$	$Z=0$

**Hình 5-19 e) Bảng chuyển đổi trạng thái và hàm ra rút gọn**



Hình 5-19 f) Đồ hình trạng thái sau khi rút gọn

Mạch chỉ có hai trạng thái nên để mã hoá ta chỉ cần sử dụng một biến nhị phân A. Để mã hoá trạng thái  $S_{012}$  thì  $A = 0$ ,  $S_{34}$  thì  $A = 1$ . Tín hiệu ra  $Z = 1$  ở trạng thái  $S_{012}$  khi  $X_1 X_2 = 11$ .

Ta dùng trigơ RS để thiết kế (dựa vào bảng hàm kích của trigơ RS-bảng 5-15 ).

Ta có phương trình đầu vào kích (R, S) của trigơ là:

$S =$  tập hợp bật của  $Q + [(1)]$  ; Tập hợp bật của Q ( $T_{on}$ ) là các cung mà Q chuyển từ  $0 \rightarrow 1$ .

$$S_A = \overline{A} \overline{X_1} X_2 + A \overline{X_1} X_2 = \overline{X_1} X_2$$

$R =$  tập hợp tắt của  $Q + [(0)]$  ; Tập hợp tắt của Q ( $T_{off}$ ) là các cung mà Q chuyển từ  $1 \rightarrow 0$ .

$$R_A = A \overline{X_2} + \overline{A} \overline{X_1} \overline{X_2} + \overline{A} X_1 \overline{X_2} = \overline{X_2}$$

Các cung [(0)], [(1)] được lấy giá trị không xác định (x) và được dùng để tối thiểu hoá.

Phương trình đặc trưng của trigơ RS

$$Q^k = S_A + \overline{R_A} Q_A$$

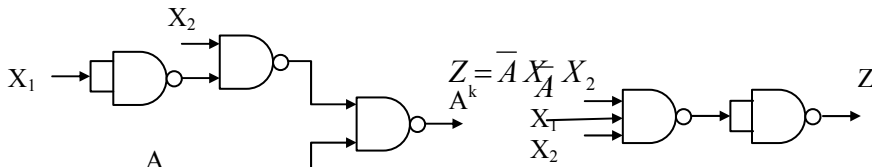
Thay giá trị của  $R_A, S_A$  vào biểu thức thu được kết quả:

$$A^k = \overline{X_1} X_2 + X_2 A = \overline{\overline{\overline{\overline{\overline{X_1} X_2}}}} + \overline{\overline{\overline{\overline{X_2 A}}}} = \overline{\overline{\overline{\overline{X_1} X_2}}}. \overline{\overline{\overline{\overline{X_2 A}}}}$$

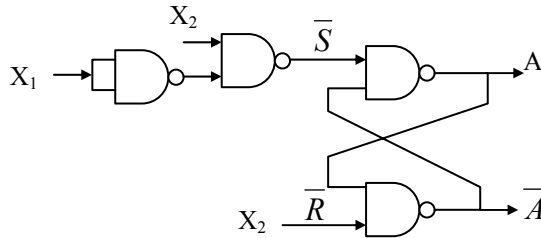
Phương trình ra:

Cung	Q	$Q^k$	R	S
0	0	0	X	0
$T_{on}$	0	1	0	1
$T_{off}$	1	0	1	0
1	1	1	0	X

Bảng 5-15. Bảng hàm kích



Hình 5-19 g) Sơ đồ mạch chỉ dùng NAND



Hình 5-19 h) Sơ đồ mạch chỉ dùng trigơ RS  
 Hình 5-19 a, b, c, d, e, f, g, h. Thiết kế mạch tuần tự.

Nếu thiết kế mạch dùng trigơ RS và các mạch NAND ta có:

$$S_A = \overline{X_1} X_2 \quad R_A = \overline{X_2}$$

Và mạch được biểu diễn ở hình 5-19 g, h.

### 5.7. HIỆN TƯỢNG CHU KỲ VÀ CHẠY ĐUA TRONG MẠCH KHÔNG ĐỒNG BỘ

Đối với mạch tuần tự đồng bộ, việc mã hoá trạng thái là làm sao cho sơ đồ thực hiện mạch là đơn giản nhất.

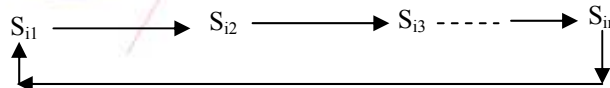
Đối với mạch tuần tự không đồng bộ, trong mạch thường xảy ra các hiện tượng hoặc là chu kỳ hoặc là chạy đua. Những hiện tượng này làm cho mạch hoạt động sai lệch đi so với chức năng của nó. Vì vậy, khi mã hoá trạng thái của mạch tuần tự không đồng bộ ta phải tránh các trường hợp đó.

#### 5.7.1. Hiện tượng chu kỳ trong mạch tuần tự không đồng bộ.

**Định nghĩa:**

Hiện tượng chu kỳ là hiện tượng tại một tổ hợp tín hiệu vào nào đó, mạch liên tục chuyển từ trạng thái này sang trạng thái khác theo một chu kỳ kín. Nghĩa là trong quá trình đó không có trạng thái nào ổn định. Do vậy, khi thay đổi tín hiệu vào không xác định được mạch đang ở trạng thái nào trong dãy trạng thái nói trên.

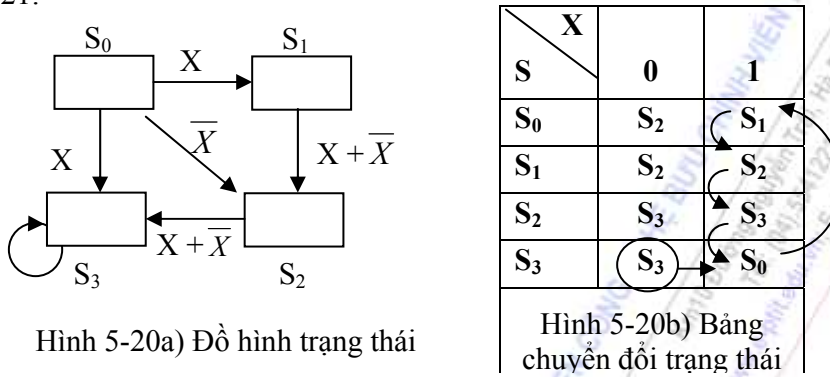
Ví dụ: ứng với một tổ hợp tín hiệu vào quá trình chuyển đổi trạng thái theo chu trình sau:



Trên bảng trạng thái hiện tượng chu kỳ được thể hiện ở chỗ: cột ứng với tổ hợp tín hiệu vào đó không có trạng thái nào được khoanh tròn (không có trạng thái nào ổn định).

Ví dụ: Đồ hình trạng thái của một mạch tuần tự không đồng bộ được biểu diễn trên hình 5-20a. Việc mã hoá trạng thái sử dụng biến nhị phân A và B là tùy chọn. Từ đồ hình trạng thái ta lập bảng chuyển đổi trạng thái 5-20b.

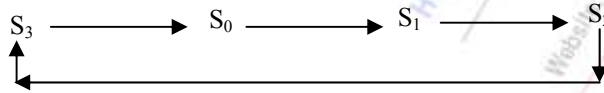
Giả thiết ban đầu mạch ở trạng thái  $S_3$  ( $AB = 10$ ) và  $X = 0$ . Sau đó tín hiệu vào  $X$  thay đổi từ 0 đến 1 thì mạch sẽ chuyển trạng thái từ  $S_3$  sang  $S_0$ . Nếu  $X$  vẫn bằng 1 thì mạch sẽ lần lượt chuyển đến các trạng thái tiếp theo là  $S_1, S_2, \dots S_0$ . Khi  $X = 1$  chu trình chuyển đổi trạng thái như hình 5-21:



Hình 5-20a) Đồ hình trạng thái

Hình 5-20b) Bảng chuyển đổi trạng thái

Hình 5-20 a,b. Bảng trạng thái hiện tượng chu kỳ



Hình 5-21. Chu trình chuyển đổi trạng thái

Khi đó mạch không có trạng thái ổn định.

### 5.7.2. Hiện tượng chạy đua trong mạch tuần tự không đồng bộ.

#### Định nghĩa:

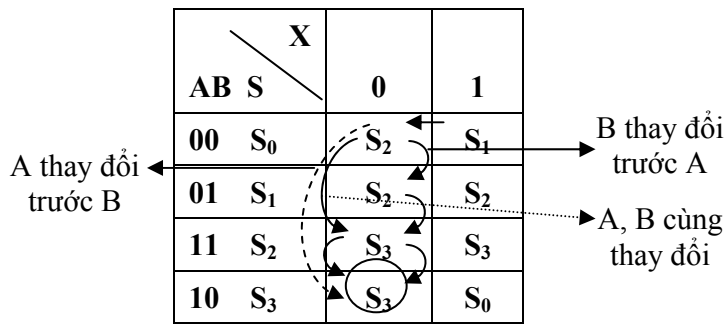
Hiện tượng chạy đua trong mạch không đồng bộ là hiện tượng: do tính không đồng nhất của các phần tử nhị phân dùng để mã hoá trạng thái, vì mạch hoạt động không đồng bộ, khi mạch chuyển trạng thái từ  $S_i \rightarrow S_j$  mạch có thể chuyển biến trạng thái theo những con đường khác nhau.

Nếu trạng thái cuối cùng của những con đường đó là ổn định và duy nhất thì chạy đua không nguy hiểm. Ngược lại, chạy đua nguy hiểm là những cách chuyển biến trạng thái khác nhau đó cuối cùng dẫn đến các trạng thái ổn định khác nhau, có thể tới trạng thái khoá và không thoát ra được.

Ví dụ: Chạy đua không nguy hiểm: Một mạch tuần tự không đồng bộ có bảng trạng thái mô tả ở hình 5-22 .

Nhìn vào bảng ta thấy nếu mạch đang ở trạng thái  $S_0$  ( $AB = 00$ ) tín hiệu vào  $X$  thay đổi từ 0  $\rightarrow$  1 mạch sẽ chuyển trực tiếp tới trạng thái  $S_2$  ( $AB = 01$ ) và nếu  $X$  vẫn bằng 0 trạng thái tiếp theo của mạch sẽ là  $S_3$ , nó sẽ là trạng thái ổn định cuối cùng của mạch nếu như  $X$  vẫn bằng 0.

Mạch có thể thay đổi trạng thái theo những con đường khác nhau tùy thuộc vào thứ tự thay đổi (hay thời gian quá độ) của A và B



**Hình 5-22.** Hiện tượng chạy đua không nguy hiểm trong mạch tuần tự không đồng bộ

Nếu A và B thay đổi đồng thời mạch sẽ chuyển trạng thái sang S<sub>2</sub> rồi mới sang S<sub>3</sub>.

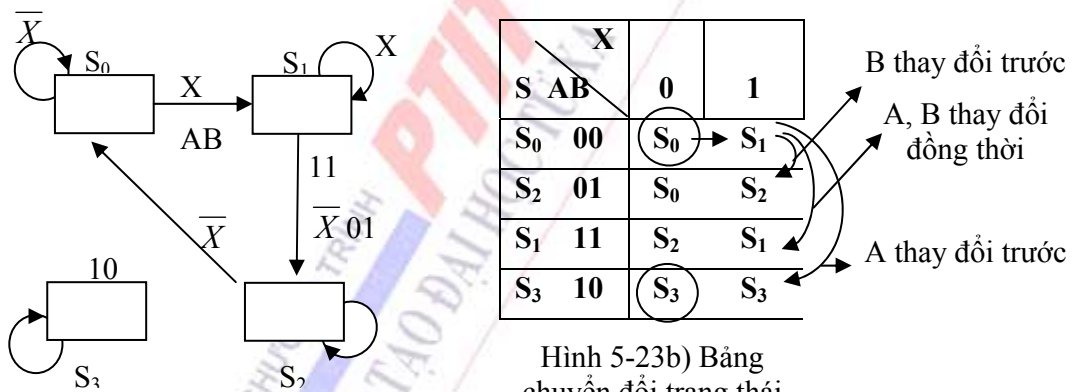
Nếu B thay đổi trước A thì mạch sẽ lần lượt chuyển qua S<sub>1</sub>, S<sub>2</sub> rồi mới sang S<sub>3</sub>.

Nếu A thay đổi trước B mạch sẽ chuyển đổi từ S<sub>0</sub> → S<sub>3</sub>.

Ta thấy rằng cả ba con đường đều dẫn đến cùng một trạng thái ổn định S<sub>3</sub>. Vậy hiện tượng chạy đua này không nguy hiểm.

Khi mạch đang ở trạng thái ổn định (trạng thái được khoanh tròn), nó chỉ thay đổi trạng thái khi tín hiệu vào thay đổi.

**Chạy đua nguy hiểm:** Đồ hình trạng thái của mạch không đồng bộ mô tả ở hình 5-23a.



Hình 5-23b) Bảng chuyển đổi trạng thái

Hình 5-23a) Đồ hình trạng thái

**Hình 5-23.** Hiện tượng chạy đua nguy hiểm trong mạch không đồng bộ

Giả thiết trạng thái ban đầu của mạch là S<sub>0</sub> (AB = 00) và tín hiệu vào X = 0. Nếu X thay đổi từ 0 → 1 thì mạch sẽ chuyển đổi trạng thái như sau:

- Nếu A, B thay đổi đồng thời thì mạch sẽ chuyển đến trạng thái S<sub>1</sub>.
- Nếu B thay đổi trước A thì mạch sẽ chuyển đến trạng thái S<sub>2</sub>.
- Nếu A thay đổi trước B thì mạch sẽ chuyển đến trạng thái S<sub>3</sub>.

Ở đây trạng thái S<sub>3</sub> là trạng thái “khóa”. Như vậy khi A thay đổi trước B thì mạch sẽ rơi vào trạng thái khóa và không thoát ra được.

Chạy đua này là chạy đua nguy hiểm.

### 5.7.3. Tối thiểu hoá và mã hoá trạng thái trong mạch tuần tự không đồng bộ.

#### 5.7.3.1. Tối thiểu hoá trạng thái

Tối thiểu hoá trạng thái là giảm bớt số trạng thái (nếu có thể) để mạch thiết kế là đơn giản và do vậy tin cậy hơn.

Đối với các ô trống trong bảng chuyển đổi trạng thái (những ô này ứng với tổ hợp tín hiệu vào không xuất hiện) có thể lấy giá trị tùy chọn để kết quả tối thiểu hoá là tối giản.

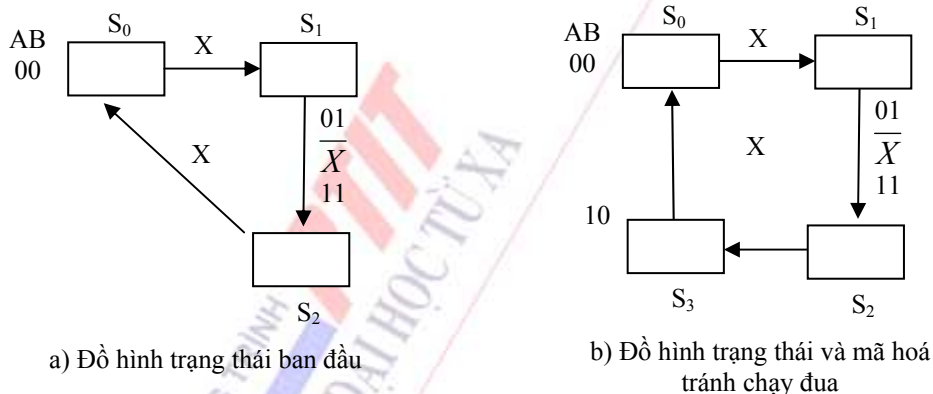
#### 5.7.3.2. Mã hoá trạng thái

Sử dụng các biến nhị phân để mã hoá các trạng thái trong của mạch. Đối với mạch tuần tự không đồng bộ phải mã hoá trạng thái để tránh được hiện tượng chu kỳ và chạy đua.

Để tránh được hiện tượng chu kỳ thì khi có mọi tín hiệu vào nhưng mạch phải luôn có một trạng thái ổn định.

Để tránh hiện tượng chạy đua, phải mã hoá trạng thái sao cho với tất cả các chuyển đổi trạng thái có thể có của mạch chỉ có duy nhất một biến thay đổi.

Ví dụ. Đồ hình trạng thái của mạch tuần tự không đồng bộ được mô tả như hình 5-24:



**Hình 5-24.** Tránh chạy đua trong mạch không đồng bộ

Cần hai biến nhị phân A và B để mã hoá 3 trạng thái này. Giả sử chọn cách mã hoá như hình 5-24a.

Với cách mã hoá này khi thay đổi từ  $S_2 \rightarrow S_0$  cả hai biến A và B đều thay đổi. Điều này dẫn đến hiện tượng chạy đua trong mạch.

Do vậy, để tránh hiện tượng chạy đua đưa thêm một trạng thái giả  $S_3$  để cho thay đổi từ  $S_2 \rightarrow S_0$  thông qua trạng thái giả này bảo đảm quá trình thay đổi trạng thái luôn chỉ có một biến thay đổi. Đồ hình này tránh được hiện tượng chạy đua.

Khi sử dụng các trạng thái giả để mã hoá cho mạch cần lưu ý tìm cách cho mạch thoát khỏi các trạng thái giả đó. Phần lớn các trường hợp ta cho mạch thoát khỏi các trạng thái giả đó vô điều kiện.

## 5.8. MỘT SỐ MẠCH TUẦN TỰ THÔNG DỤNG

### 5.8.1. Bộ đếm.

Bộ đếm là mạch tuần tự đơn giản, nó được xây dựng từ các phần tử nhớ là các trigơ và các mạch logic tổ hợp.

Các bộ đếm là thành phần cơ bản của các hệ thống số, chúng được sử dụng để đếm thời gian, chia tần số, điều khiển các mạch khác... Bộ đếm được sử dụng rất nhiều trong máy tính, trong thông tin. Để xây dựng bộ đếm, người ta có thể dùng mã nhị phân hoặc các loại mã khác như mã Gray, mã NBCD, mã vòng...

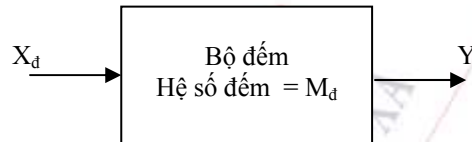
Phần này sẽ đưa ra những đặc điểm cơ bản nhất của bộ đếm và các phương pháp thiết kế bộ đếm.

#### 5.8.1.1. Định nghĩa và phân loại bộ đếm

##### 1. Định nghĩa.

Bộ đếm là một mạch tuần tự tuần hoàn có một lối vào đếm và một lối ra, mạch có số trạng thái trong bằng chính hệ số đếm (ký hiệu là  $M_d$ ). Dưới tác dụng của tín hiệu vào đếm, mạch sẽ chuyển từ trạng thái trong này đến một trạng thái trong khác theo một thứ tự nhất định. Cứ sau  $M_d$  tín hiệu vào đếm mạch lại trở về trạng thái xuất phát ban đầu.

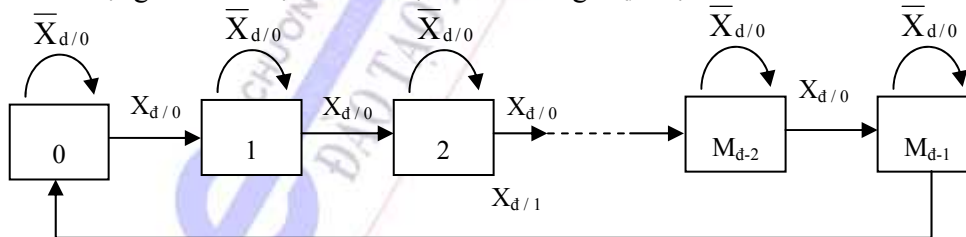
Sơ đồ khối được mô tả như ở hình 5- 25.



Hình 5- 25 Sơ đồ khối của bộ đếm

##### 2. Đồ hình trạng thái tổng quát của bộ đếm.

Đồ hình trạng thái của bộ đếm có hệ số đếm bằng  $M_d$  được mô tả ở hình 5-26.



Hình 5-26. Đồ hình trạng thái của bộ đếm  $M_d$

Khi không có tín hiệu vào đếm ( $X_d$ ) mạch giữ nguyên trạng thái cũ, khi có tín hiệu đếm thì mạch sẽ chuyển đến trạng thái kế tiếp.

Tính chất tuần hoàn của bộ đếm thể hiện ở chỗ: sau  $M_d$  tín hiệu vào  $X_d$  thì mạch lại quay trở về trạng thái xuất phát ban đầu.

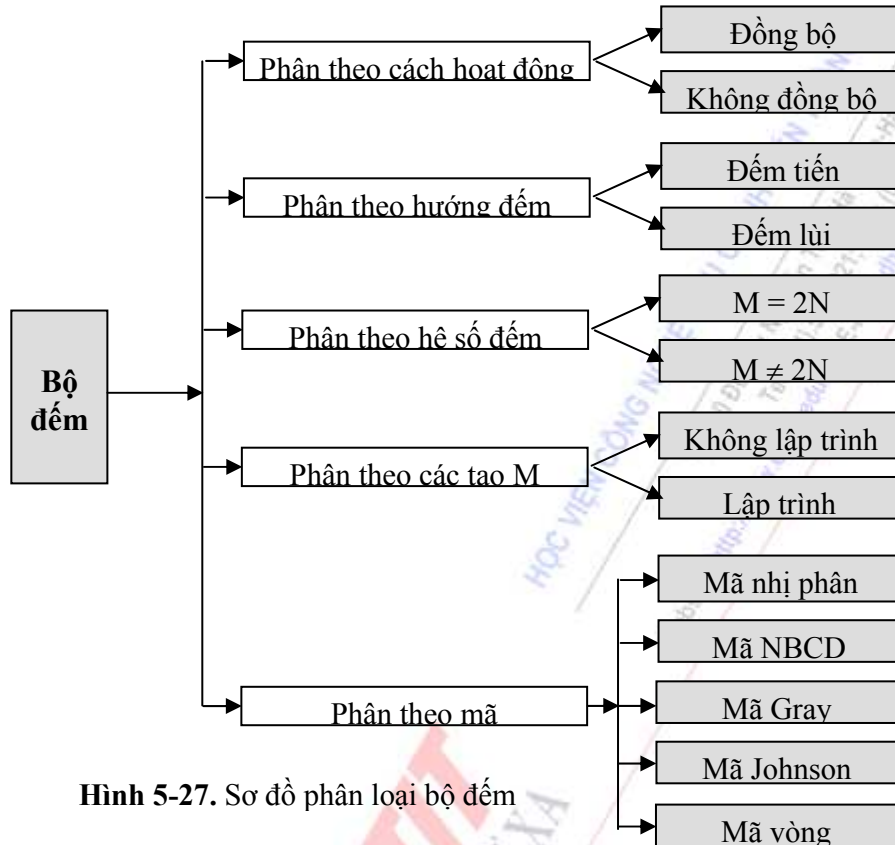
Tín hiệu ra của bộ đếm chỉ xuất hiện ( $Y = 1$ ) duy nhất trong trường hợp: bộ đếm đang ở trạng thái  $M_d - 1$  và có tín hiệu vào  $X_d$ . Khi đó bộ đếm sẽ chuyển về trạng thái 0.



Trong trường hợp cần hiển thị trạng thái của bộ đếm thì phải dùng thêm mạch giải mã.

### 2. Phân loại bộ đếm.

Có nhiều cách phân loại bộ đếm. Hình 5-27 là cách phân loại điển hình của bộ đếm.



Hình 5-27. Sơ đồ phân loại bộ đếm

#### 5.8.1.2. Các bước thiết kế bộ đếm

Hình 5-28 là lưu đồ thiết kế bộ đếm.



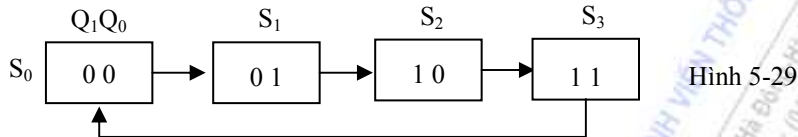
Hình 5-28. Các bước thiết kế bộ đếm

**A. Bộ đếm đồng bộ.**

**A.1. Bộ đếm nhị phân**

Thiết kế bộ đếm nhị phân đồng bộ có  $M_d = 4$ .

Do  $M_d = 4$  nên lập được đồ hình trạng thái ở hình 5-29.

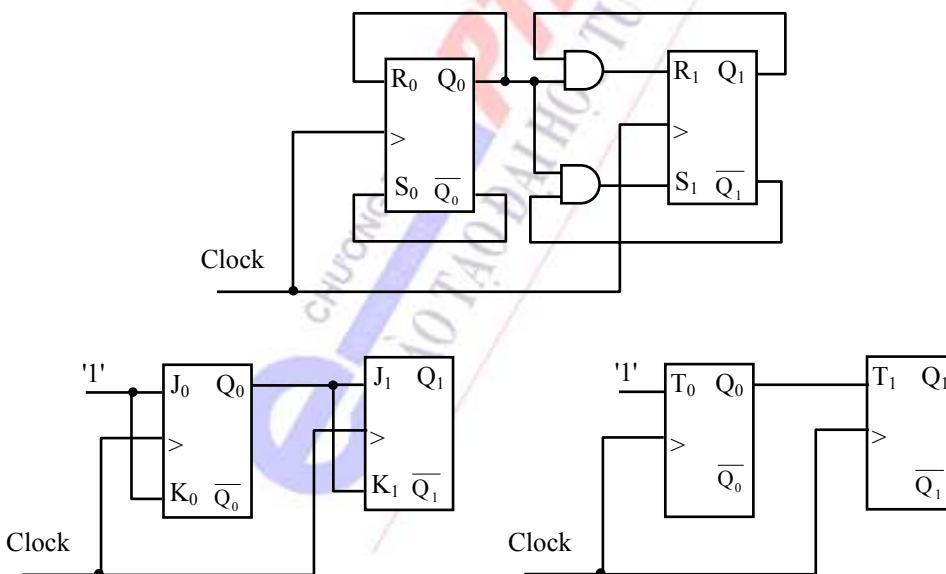


Từ đó xác định được số trigơ cần dùng để thiết kế bộ đếm ( $n = 2$ ) và mã hoá các trạng thái đó. Hai trigơ cần để mã hoá các thái là  $Q_1$  và  $Q_0$ . Dùng bảng hàm kích 5-16 để xác định các lối vào kích cho các trigơ.

n		n + 1		Trigơ $Q_1$					Trigơ $Q_0$				
$Q_1$	$Q_0$	$Q_1^k$	$Q_0^k$	$R_1$	$S_1$	$J_1$	$K_1$	$T_1$	$R_0$	$S_0$	$J_0$	$K_0$	$T_0$
0	0	0	1	X	0	0	X	0	0	1	1	X	1
0	1	1	0	0	1	1	X	1	1	0	X	1	1
1	0	1	1	0	X	X	0	0	0	1	1	X	1
1	1	0	0	1	0	X	1	1	1	0	X	1	1

Bảng 5-16

Tối thiểu hoá hàm kích của các trigơ, nhận được kết quả:



**Hình 5-30.** Bộ đếm Mod 4 dùng trigơ RS, JK, T.

Đối với trigơ  $Q_0$ :

$$R_0 = Q_0; \quad S_0 = \overline{Q_0}$$

$$J_0 = K_0 = 1;$$

$$T_0 = 1;$$

Đối với trigơ  $Q_1$ :

$$R_1 = Q_1 Q_0; \quad S_1 = \overline{Q_1} Q_0$$

$$J_1 = K_1 = Q_0;$$

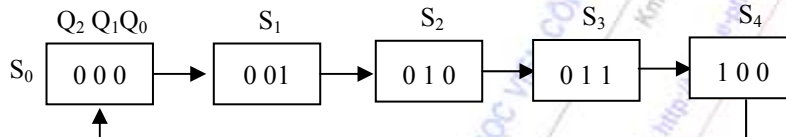
$$T_1 = Q_0;$$

Sơ đồ mạch điện như ở hình 5-30.

### A. 2. Bộ đếm có mod đếm bất kỳ

Thiết kế bộ đếm đồng bộ có  $M_d = 5$ .

Do  $M_d = 5$  nên lập được đồ hình trạng thái như hình 5-31.



Hình 5-31. Đồ hình trạng thái của bộ đếm Mod 5

Từ đó xác định được số trigơ cần dùng để thiết kế bộ đếm ( $n = 3$ ) và mã hoá các trạng thái đó. Ba trigơ cần để mã hoá các thái là  $Q_2$ ,  $Q_1$  và  $Q_0$ . Dùng bảng hàm kích 5-17 để xác định các lối vào kích cho các trigơ.

n			n + 1			Trigơ $Q_2$		Trigơ $Q_1$		Trigơ $Q_0$	
$Q_2$	$Q_1$	$Q_0$	$Q_2^k$	$Q_1^k$	$Q_0^k$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	0	0	0	X	1	0	X	0	X

Bảng 5-17. Bảng hàm kích cho trigơ

Tối thiểu hoá hàm kích của các trigơ, nhận được kết quả:

$$J_0 = \overline{Q_2}; \quad K_0 = 1;$$

$$J_1 = K_1 = Q_0;$$

$$J_2 = Q_1 Q_0; \quad K_2 = 1;$$

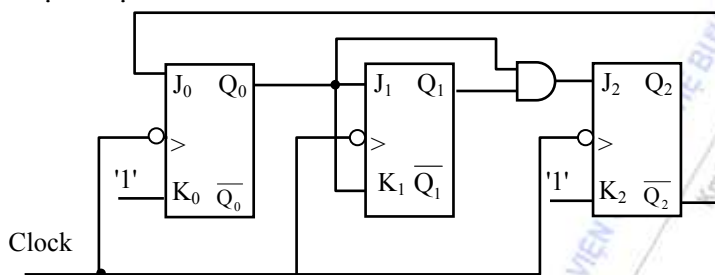
Kiểm tra khả năng tự khởi động bằng bảng 5-18.

Nhìn vào bảng trạng thái 5-18, ta thấy các trạng thái dư sau 1 số xung nhịp đều quay trở lại vòng đếm nên ta nói bộ đếm này tự khởi động.

n			n + 1		
$Q_2$	$Q_1$	$Q_0$	$Q_2^k$	$Q_1^k$	$Q_0^k$
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	0	0	0

Bảng 5-18. Kiểm tra khả năng tự khởi động

Sơ đồ mạch điện ở hình 5-32:



Hình 5-32. Bộ đếm Mod 5 đồng bộ

## B. Bộ đếm không đồng bộ.

### B. 1. Bộ đếm nhị phân

Các bộ đếm này có sơ đồ rất đơn giản với đặc điểm:

- Chỉ dùng một loại trigơ T hoặc JK. Nếu dùng trigơ T thì lối vào T luôn được nối với mức logic '1', nếu dùng trigơ JK thì J và K được nối với nhau và nối với mức '1'.

- Đầu ra của trigơ trước được nối với lối vào xung nhịp của trigơ sau kế tiếp. Khi đếm tiến thì lấy ở đầu ra Q, khi đếm lùi thì lấy ở đầu ra  $\bar{Q}$  (với giả thiết xung Clock tích cực tại sườn âm ↓).

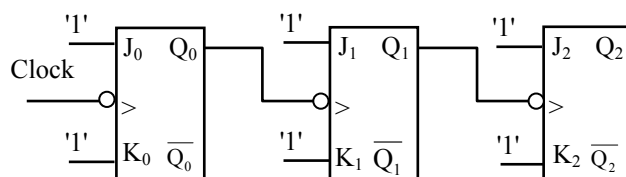
- Tín hiệu vào  $X_d$  luôn được đưa tới lối vào xung nhịp của trigơ có trọng số nhỏ nhất.

Ví dụ đối với bộ đếm nhị phân không đồng bộ  $M_d = 2^n$  dùng các trigơ  $Q_0, Q_1 \dots Q_{n-1}$  với  $Q_0$  là bit có trọng số nhỏ nhất,  $Q_{n-1}$  là bit có trọng số lớn nhất, ta có:

- Khi đếm tiến:  $C_{Q_0} = X; C_{Q_1} = Q_0 \dots C_{Q_{n-1}} = C_{Q_{n-2}}$ .

- Khi đếm lùi:  $C_{Q_0} = X; C_{Q_1} = \bar{Q}_0 \dots C_{Q_{n-1}} = C_{\bar{Q}_{n-2}}$

Sơ đồ của bộ đếm nhị phân không đồng bộ 3 bit ( $M_d = 8$  - đếm tiến) dùng trigơ JK được cho ở hình 5-33

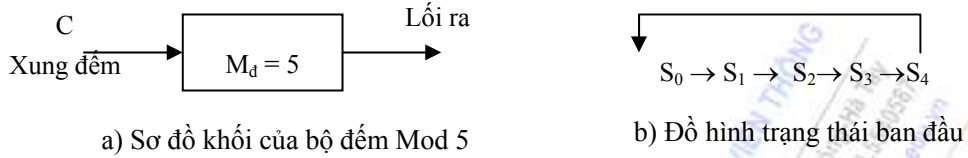


Hình 5-33. Bộ đếm nhị phân không đồng bộ 3 bit

**B. 2. Bộ đếm có mod đếm bất kỳ.**

Ví dụ: Thiết kế bộ đếm  $M_d = 5$  không đồng bộ.

Từ yêu cầu bài toán ta xây dựng sơ đồ khối và đồ hình trạng thái như ở hình 5-34.



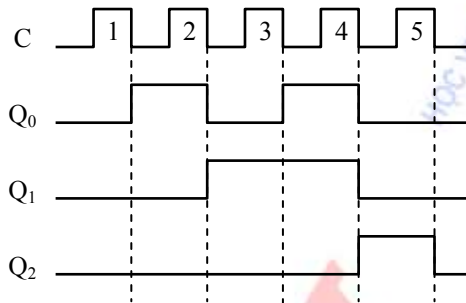
Hình 5-34. Mô hình thiết kế bộ đếm

- Chọn lựa mã hoá trạng thái

Có 5 trạng thái nên số trigơ bằng 3, chọn trigơ JK. Chọn mã BCD8421.

$S_0 = 000$ ;  $S_1 = 001$ ;  $S_2 = 010$ ;  $S_3 = 011$ ;  $S_4 = 100$ .

- Chọn xung đồng hồ từ giản đồ xung 5-35.



Hình 5-35. Giản đồ xung của bộ đếm Mod 5

$C_1 = \downarrow C$ ;  $C_2 = \downarrow Q_0$ ;  $C_3 = \downarrow C$ ;

- Tìm hệ phương trình:

$Q_1 Q_0$	00	01	11	10
$Q_2$	00	01	11	10
0	001	010	100	011
1	000	x	x	x

$Q_1 Q_0$	00	01	11	10
$Q_2$	00	01	11	10
0	1	0	0	1
1	0	x	x	x

$$Q_0^k = \overline{Q_2} \overline{Q_0}$$

$Q_1 Q_0$	00	01	11	10
$Q_2$	00	01	11	10
0	x	1	0	x
1	x	x	x	x

$$Q_1^k = \overline{Q_1}$$

$Q_1 Q_0$	00	01	11	10
$Q_2$	00	01	11	10
0	0	0	1	0
1	0	x	x	x

$$Q_2^k = \overline{Q_2} \overline{Q_1} Q_0$$

Bảng 5-19. Bảng tính hàm kích của bộ đếm

Sau khi tối thiểu hoá bảng bảng 5-19 ta nhận được hệ phương trình:

$$Q_0^k = \overline{Q_2} \overline{Q_0}$$

$$Q_1^k = \overline{Q_1}$$

$$Q_2^k = \overline{Q_2} Q_1 Q_0$$

Kiểm tra khả năng tự khởi động bằng bảng 5-20:

n			n + 1		
Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q <sup>k</sup> <sub>2</sub>	Q <sup>k</sup> <sub>1</sub>	Q <sup>k</sup> <sub>0</sub>
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	0	0	0

Bảng 5-20. Kiểm tra khả năng tự khởi động

Nhìn vào bảng 5-20, ta thấy các trạng thái dư sau 1 số xung nhịp đều quay trở lại vòng đếm nên ta nói bộ đếm này tự khởi động.

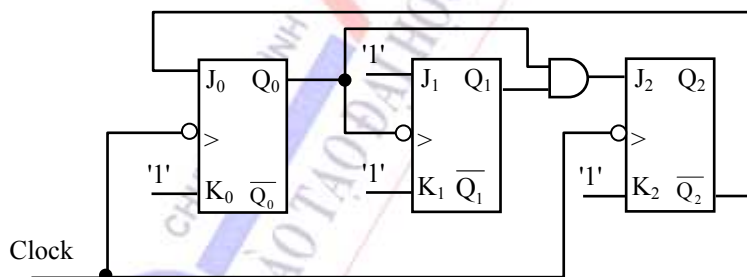
Từ đó ta tìm được phương trình hàm kích:

$$J_0 = \overline{Q_2}; \quad K_0 = 1;$$

$$J_1 = K_1 = 1;$$

$$J_2 = Q_1 Q_0; \quad K_2 = 1;$$

Từ đó ta vẽ được mạch điện của bộ đếm Mod 5 không đồng bộ như hình 5-36.



Hình 5-36. Sơ đồ mạch điện của bộ đếm Mod 5 đồng bộ

## 5.8.2. Bộ ghi dịch.

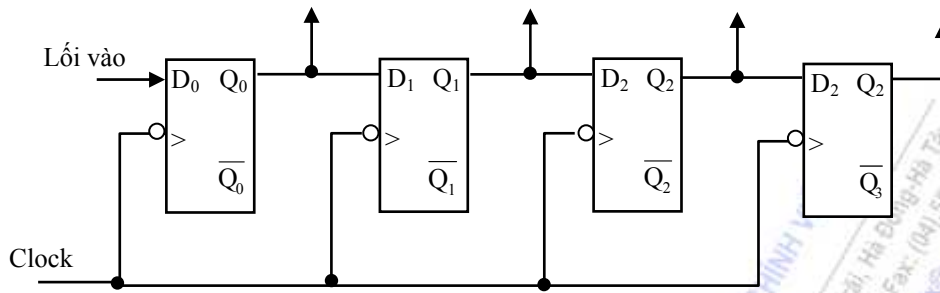
Bộ ghi dịch có khả năng ghi giữ và dịch thông tin.

### 5.8.2.1. Cấu tạo và phân loại

#### a) Cấu tạo:

Bộ ghi dịch gồm một dãy các phần tử đơn bit mắc liên tiếp và đóng trên cùng một chip. Các trigơ sử dụng trong bộ ghi dịch thường là trigơ D hoặc các loại trigơ khác mắc theo kiểu D. Để

ghi  $n$  bit thông tin, người ta sử dụng  $n$  trigơ, đầu ra của trigơ này mắc tới đầu vào của trigơ kế tiếp. Bộ ghi dịch ghi được  $n$  bit thông tin được gọi là bộ ghi dịch  $n$  bit. Hình 5- 37 là sơ đồ của một bộ ghi dịch 4 bit dùng trigơ D



Hình 5-37. Bộ ghi dịch 4 bit dịch phải

Thông tin được nạp vào bộ ghi dịch từng bit một và được đồng bộ với xung nhịp C.

**b) Phân loại:**

- Phân theo cách đưa tín hiệu vào và lấy tín hiệu ra:

- ◆ Vào nối tiếp, ra song song: thông tin được đưa vào thành ghi dịch tuần tự từng bit một, số liệu được đưa ra đồng thời tức là tất cả  $n$  trigơ của thành ghi được đọc cùng một lúc.
- ◆ Vào song song, ra song song: thông tin được đưa vào và lấy ra đồng thời ở  $n$  trigơ.
- ◆ Vào nối tiếp, ra nối tiếp: thông tin được đưa vào và lấy ra tuần tự từng bit một.
- ◆ Vào song song, ra nối tiếp: thông tin được đưa vào đồng thời cả  $n$  trigơ, lấy ra tuần tự từng bit một dưới sự điều khiển của xung nhịp.

- Phân theo hướng dịch:

- ◆ Dịch phải, dịch trái, dịch hai hướng, dịch vòng

- Phân theo đầu vào:

- ◆ Đầu vào đơn: mỗi trigơ trong bộ ghi dịch chỉ sử dụng một đầu vào điều khiển, ví dụ như trigơ D hay các trigơ khác mắc theo kiểu D.
- ◆ Đầu vào đôi: các trigơ trong bộ ghi dịch sử dụng cả hai đầu vào điều khiển, ví dụ hai lỗi vào điều khiển của trigơ JK hay trigơ RS.

- Phân theo đầu ra:

- ◆ Đầu ra đơn: mỗi trigơ trong bộ ghi dịch chỉ có một đầu ra  $Q_i$  (hay  $\overline{Q}_i$ ) được đưa ra chân của vi mạch.
- ◆ Đầu ra đôi: cả hai đầu ra của trigơ  $Q_i$  và  $\overline{Q}_i$  đều được đưa ra chân của vi mạch.

**c) Ứng dụng của bộ ghi dịch**

Bộ ghi dịch được sử dụng rộng rãi để nhớ dữ liệu, chuyển dữ liệu từ song song thành nối tiếp và ngược lại. Bộ ghi dịch là thành phần không thể thiếu được trong CPU của các hệ vi xử lý, trong các cổng vào/ra có khả năng lập trình.

Bộ ghi dịch còn được dùng để thiết kế bộ đếm, tạo dãy tín hiệu nhị phân tuần hoàn...

### 5.8.2.2. Hoạt động cơ bản của bộ ghi dịch

Trong phần này ta giới thiệu bộ ghi dịch 4 bit nạp vào nối tiếp hoặc song song, ra nối tiếp và song song, dịch phải.

Sơ đồ bộ ghi dịch này được trình bày trên hình 5- 37.

Bộ ghi dịch này có thể nạp thông tin vào nối tiếp hoặc song song. Đầu ra nối tiếp được lấy ra ở trigơ cuối cùng, đầu ra song song được lấy ra đồng thời trên cả 4 trigơ. Việc nạp thông tin vào song song được thực hiện bởi một trong hai đầu vào Preset 1 và Preset 2 (đây là 2 lối vào phụ). Trước khi làm việc cần phải xoá tất cả các trigơ về trạng thái '0' nhờ lối vào Clear. Thông tin trong bộ ghi dịch này được dịch phải.

## TÓM TẮT

Khác với mạch logic tổ hợp, mạch logic tuần tự có tín hiệu đầu ra phụ thuộc không những tín hiệu đầu vào ở thời điểm xét mà cả vào trạng thái mạch điện sẵn có ở thời điểm đó. Đây là đặc điểm chức năng logic của mạch tuần tự. Để nhớ trạng thái mạch điện, mạch tuần tự phải có phần tử nhớ - đó là các trigơ.

### 1- Tính chất cơ bản của Trigơ

Trigơ là linh kiện logic cơ bản của mạch số. Trigơ có hai trạng thái ổn định, dưới tác dụng của tín hiệu bên ngoài có thể chuyển đổi từ trạng thái ổn định này sang trạng thái ổn định kia, nếu không có tác dụng tín hiệu bên ngoài thì nó duy trì mãi trạng thái ổn định vốn có. Vì thế, trigơ có thể được dùng làm phần tử nhớ của số nhị phân.

### 2- Quan hệ giữa chức năng logic và hình thức cấu trúc của trigơ

Chức năng logic và hình thức cấu trúc của trigơ là hai khái niệm khác nhau. Chức năng logic là quan hệ giữa trạng thái tiếp theo của đầu ra với trạng thái hiện tại của đầu ra và các tín hiệu đầu vào. Do chức năng logic khác nhau mà trigơ được phân thành các loại RS, D, T, JK. Còn do hình thức cấu trúc khác nhau mà trigơ lại được phân thành loại trigơ thường và loại trigơ chính phụ.

Một trigơ có chức năng logic xác định có thể thực hiện bằng các hình thức cấu trúc khác nhau. Ví dụ, các trigơ cấu trúc loại chính phụ và loại thường đều có thể thực hiện chức năng của một trigơ khác. Nghĩa là cùng một cấu trúc có thể đảm trách những chức năng khác nhau.

3- Mạch tuần tự cụ thể có rất nhiều chủng loại. Chương này chỉ giới thiệu một số loại mạch tuần tự điển hình: bộ đếm, bộ ghi dịch... Đồng thời với việc nắm vững cấu trúc, nguyên lý công tác và đặc điểm của các mạch tuần tự đó, chúng ta cũng phải nắm vững được đặc điểm chung của mạch tuần tự và phương pháp chung khi phân tích và thiết kế mạch tuần tự.

## CÂU HỎI ÔN TẬP CHƯƠNG 5

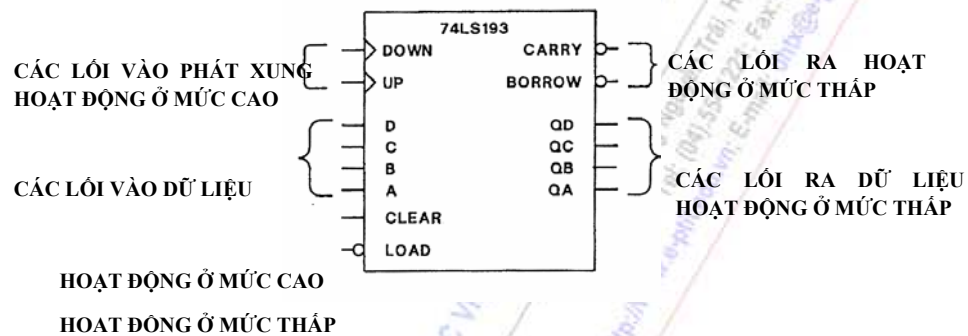
1. Cho các trigơ cơ bản loại RS, JK, D và T. Loại trigơ nào trong số các loại này có thể thực hiện được mà không cần tín hiệu đồng bộ.
  - a. Trigơ RS và trigơ D.



- b. Trigr JK và trigr T.
  - c. Trigr RS và trigr T.
  - d. Trigr JK và trigr D
2. Trong các loại trigr sau, trigr nào còn tồn tại tổ hợp cấm:
    - a. Trigr D.
    - b. Trigr T
    - c. Trigr RS.
    - d. Trigr JK.
  3. Cần bao nhiêu cổng NAND để thực hiện tạo ra một trigr RS đồng bộ:
    - a. 2.
    - b. 3.
    - c. 4.
    - d. 5.
  4. Nếu đầu vào D của trigr thay đổi từ cao đến thấp thì đầu ra
    - a. thay đổi trạng thái của nó một cách tức thời
    - b. sẽ thay đổi sau khi có xung nhịp clock ở đầu vào .
    - c. sẽ thay đổi sau khi có 2 xung nhịp clock ở đầu vào .
    - d. sẽ không thay khi có xung nhịp tiếp theo.
  5. Một trigr JK được ở chế độ lật. Nếu tần số Clock của nó là 1000 hz thì tần số tại lối ra là:
    - a. 2000 hz.
    - b. 1000 hz.
    - c. 100 hz.
    - d. 500 hz.
  6. Mô hình Mealy là mô hình:
    - a. có hàm ra phụ thuộc vào tín hiệu vào và trạng thái trong của mạch.
    - b. có hàm ra phụ thuộc vào tín hiệu vào.
    - c. có hàm ra phụ thuộc vào trạng thái trong của mạch.
    - d. không có phương án nào đúng.
  7. Mô hình Moore là mô hình:
    - a. có hàm ra phụ thuộc vào tín hiệu vào và trạng thái trong của mạch.
    - b. có hàm ra phụ thuộc vào tín hiệu vào.

- c. có hàm ra phụ thuộc vào trạng thái trong của mạch.
  - d. không có phương án nào đúng.
8. Các phương pháp mô tả mạch tuần tự:
- a. Bảng chuyển đổi trạng thái.
  - b. Bảng tín hiệu ra.
  - c. Đồ hình trạng thái.
  - d. Cả ba phương án trên đều đúng.
9. Các phần tử nhớ của bộ ghi dịch là:
- a. Trùng D.
  - b. Trùng RS.
  - c. Trùng JK.
  - d. Bất kỳ loại trùng nào nhưng phải đưa về dạng trùng D.
10. Cần bao nhiêu trùng để thực hiện tạo ra một bộ ghi dịch 4 bit:
- a. 2.
  - b. 3.
  - c. 4.
  - d. 5.
11. Bằng cách nào tạo ra được một Trùng Chính - phụ (MS):
- a. Từ hai trùng cùng loại đồng bộ.
  - b. Từ hai trùng cùng loại.
  - c. Từ ba trùng cùng loại.
  - d. Từ 4 trùng cùng loại.
12. Bộ đếm mã Johnson là:
- a. Bộ đếm vòng.
  - b. Bộ đếm vòng xoắn.
  - c. Bộ đếm nhị phân.
  - d. Cả ba phương án trên đều đúng.
13. Một bộ đếm nhị phân 4 bit thì tần số tại lối ra của bit có trọng số lớn nhất so với tần số xung nhịp:
- a. nhỏ hơn 2 lần.
  - b. nhỏ hơn 4 lần.
  - c. nhỏ hơn 8 lần.

- d. nhỏ hơn 16 lần.
14. Trên bộ đếm đồng bộ, các lối vào Clock
- phải được nối với tầng LSB của bộ đếm.
  - phải được nối với tầng MSB của bộ đếm.
  - là chung cho mỗi tầng của bộ đếm.
  - phải là dạng xung được phát theo kiểu đơn bước.



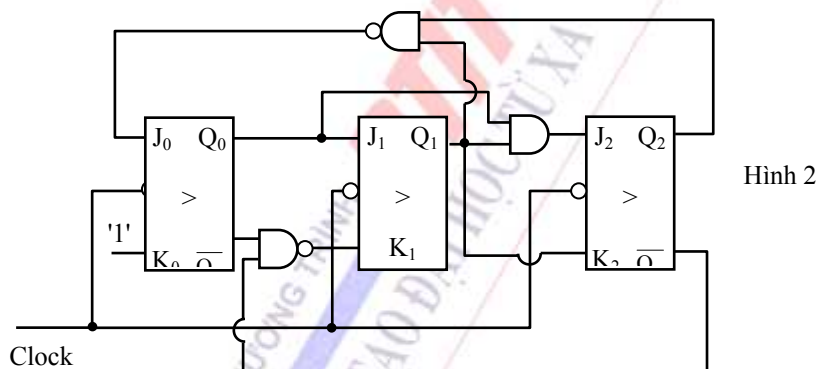
Hình 1

15. Với IC xuất hiện trên hình 1, chân CLEAR
- xoá tất cả 6 lối ra của IC.
  - lập tất cả 6 lối ra của IC.
  - Chỉ xoá các lối ra từ QD đến QA.
  - Chỉ xoá các lối ra CARRY và BORROW.
16. Nếu các lối vào của LS 193 có giá trị là 1010, thì các lối ra của bộ đếm sẽ là:
- hiển thị giá trị 1010 sau khi chức năng LOAD được kích hoạt.
  - hiển thị giá trị 0101 là giá trị đảo của 1010 sau khi chức năng LOAD được kích hoạt.
  - hiển thị giá trị 1010 sau một xung clock.
  - sẽ tăng lên nhưng không thể giảm xuống.
17. Các lối ra CARRY và BORROW của bộ đếm LS 193:
- binh thường ở mức thấp và sẽ phát ra một xung hoạt động ở mức cao.
  - có thể được đưa lên mức cao bằng cách kích hoạt chức năng LOAD.
  - có thể được đưa xuống mức thấp bằng cách kích hoạt chức năng CLEAR.
  - binh thường ở mức cao và sẽ phát ra một xung hoạt động ở mức thấp.
18. Trên bộ đếm LS 193, bộ đếm thực hiện đếm tiến:
- nếu chân DOWN được cấp xung và chân UP nối lên  $V_{CC}$ .

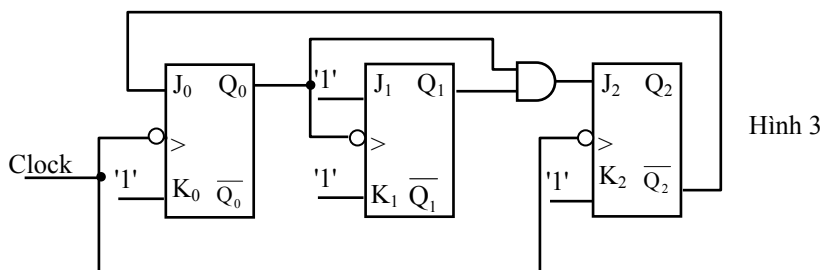
- b. nếu chân UP được cấp xung và chân DOWN nối lên  $V_{CC}$ .
  - c. chân UP và DOWN được cấp xung đồng thời.
  - d. chân UP và DOWN đều được nối lên  $V_{CC}$ .
19. Trên bộ đếm LS 193, bộ đếm thực hiện đếm lùi:
- a. nếu chân DOWN được cấp xung và chân UP nối lên  $V_{CC}$ .
  - b. nếu chân UP được cấp xung và chân DOWN nối lên  $V_{CC}$ .
  - c. chân UP và DOWN được cấp xung đồng thời.
  - d. chân UP và DOWN đều được nối lên  $V_{CC}$ .
20. Một bộ đếm không đồng bộ 5 bit thì cung cấp hệ số chia tần hay hệ số chia số đếm là bao nhiêu :
- a. 32.
  - b. 16.
  - c. 8.
  - d. Không có trường hợp nào ở trên.
21. Với bộ đếm không đồng bộ, qua mỗi trigơ thì lối ra của nó chia tần số đầu vào ra làm :
- a. 4.
  - b. 2.
  - c. 10.
  - d. 16.
22. Tần số đầu vào của một bộ đếm không đồng bộ 4 bit là 100KHz. Vậy tần số tại đầu ra tại lối ra có trọng số lớn nhất (MSB) là bao nhiêu ?
- a. 100 KHz.
  - b. 50 KHz.
  - c. 12,5 KHz.
  - d. 6,25 KHz.
23. Khi tần số xung nhịp của bộ đếm không đồng bộ tăng thì :
- a. Các đầu vào xoá (CLEAR) và lập (SET) không điều khiển tất cả các trigơ của bộ đếm.
  - b. Chức năng của các đầu vào xoá (CLEAR) và lập (SET) không bị ảnh hưởng gì.
  - c. Tăng khả năng đếm lớn nhất của nó.
  - d. Giảm khả năng đếm lớn nhất của nó.
24. Một xung clock vào :
- a. Cho phép một bộ đếm không đồng bộ chạy trong chế độ không đồng bộ.

- b. Xác định số đếm lớn nhất của bộ đếm không đồng bộ.
  - c. Thay đổi lần lượt các chế độ hoạt động của bộ đếm không đồng bộ.
  - d. Chuyển một bộ đếm không đồng bộ thành một bộ đếm nối tiếp.
25. Khi phát xung vào bộ đếm không đồng bộ thì xung clock là :
- a. Tín hiệu điều khiển tất cả các đầu vào.
  - b. Tín hiệu điều khiển tầng LSB của bộ đếm.
  - c. Tín hiệu điều khiển tầng MSB của bộ đếm.
  - d. Trạng thái tĩnh.
26. Khi chân CLEAR (xoá) của bộ đếm không đồng bộ được đưa xuống mức thấp thì bộ đếm :
- a. Không tiếp nhận xung xoá bởi vì xung CLOCK chạy tự do.
  - b. Tiếp nhận xung xoá, lúc này tất cả các đầu ra không đảo được đặt cố định ở mức thấp.
  - c. Tiếp nhận xung xoá, lúc này tất cả các đầu ra không đảo được đặt tạm thời ở mức thấp.
  - d. Dao động giữa giá trị đếm lớn nhất và giá trị nhỏ nhất.
27. Khi chân SET (lập) của bộ đếm không đồng bộ được đưa xuống mức thấp thì bộ đếm:
- a. Không tiếp nhận xung lập bởi vì xung CLOCK chạy tự do.
  - b. Tiếp nhận xung lập, lúc này tất cả các đầu ra không đảo được đặt cố định ở mức cao.
  - c. Tiếp nhận xung lập, lúc này tất cả các đầu ra không đảo được đặt tạm thời ở mức cao.
  - d. Dao động giữa giá trị đếm lớn nhất và giá trị nhỏ nhất.
28. Một bộ đếm không đồng bộ được coi như là một bộ đếm nối tiếp là bởi vì :
- a. Tất cả các đầu ra thay đổi đồng thời.
  - b. Một tín hiệu xung nhịp điều khiển tất cả các trigơ.
  - c. Tất cả các đầu ra là đảo.
  - d. Các trigơ trong bộ đếm hoạt động theo phương pháp chuỗi cánh hoa (daisy-chain). (Điều này có nghĩa là lối ra của trigơ trước sẽ điều khiển lối vào của trigơ sau).
29. Hệ số chia tần số cho một bộ đếm không đồng bộ 4 bit là :
- a. 1, 2, 4 và 8.
  - b. 1, 2, 4 và 16.
  - c. 2, 4, 8 và 16.
  - d. Tất cả các trường hợp trên, phụ thuộc vào tần số xung clock.

30. Nếu một bộ đếm không đồng bộ 4 bit có các lối ra đảo thì chúng
- Đếm từ 15  $\rightarrow$  0.
  - Đếm từ 0  $\rightarrow$  15.
  - Luôn là 0.
  - Luôn là 15.
31. Cần bao nhiêu chu kỳ xung clock đầu vào để phát ra một chu kỳ hoàn chỉnh tại lối ra có trọng số lớn nhất (MSB) của bộ đếm không đồng bộ 4 bit.
- 32.
  - 16.
  - 8.
  - Không có trường hợp nào ở trên.
32. Các Trơ JK sử dụng trong bộ đếm không đồng bộ được xây dựng bằng cách:
- Nối lối vào J và K với  $V_{CC}$  và vô hiệu hoá các lối vào CLR (xóa) và PR (lập).
  - Cấu trúc mạch Trơ JK giống như một mạch Trơ T.
  - Nối tất cả các lối vào J, K, CLR và PR với  $V_{CC}$ .
  - Sử dụng bất kỳ cấu trúc nào ở trên.
33. Cho bộ đếm hình 2. Cho biết đây là bộ đếm Mod mấy?

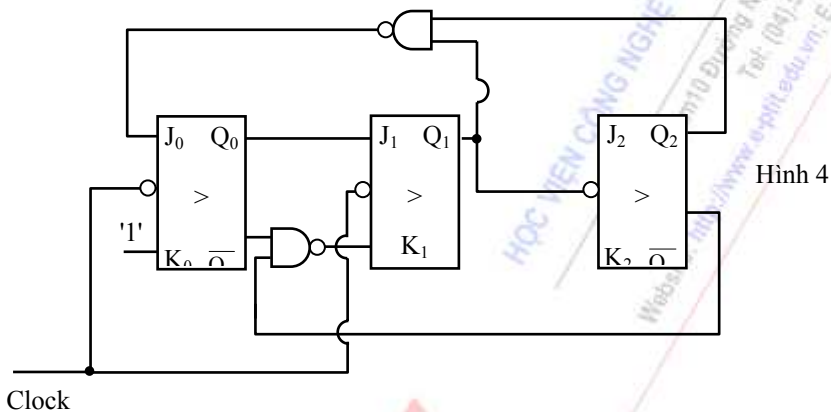


- Mod 5.
  - Mod 6.
  - Mod 7.
  - Mod 8.
34. Cho bộ đếm hình 3. Cho biết đây là bộ đếm Mod mấy?



- a. Mod 5.
- b. Mod 6.
- c. Mod 7.
- d. Mod 8.

35. Cho bộ đếm hình 4. Cho biết đây là bộ đếm Mod mấy?



- a. Mod 5.
- b. Mod 6.
- c. Mod 7.
- d. Mod 8.

36. Thiết kế bộ đếm Mod 9 đồng bộ.

37. Thiết kế bộ đếm Mod 9 không đồng bộ.

38. Bộ ghi dịch của bạn được reset. Sau 4 sườn dương của xung clock tất cả 4 lối ra đều ở mức cao. Kết luận của bạn về các lối vào dữ liệu là:

- a. được đặt ở mức thấp.
- b. Luân lượt thay đổi giữa hai trạng thái cao và thấp.
- c. Luân lượt thay đổi giữa hai trạng thái thấp và cao.
- d. được đặt ở mức cao.

39. Nếu mạch của bạn được thiết kế để dịch trái dữ liệu vào nối tiếp, sau đó luồng bit dữ liệu chuyển động từ:

- a. Trái qua phải.
- b. Từ phải qua trái.

- c. Một trong hai trường hợp trên.
  - d. Không có trường hợp nào ở trên.
40. Nếu mạch của bạn được định hình để dịch phải dữ liệu vào nối tiếp, sau đó luồng bit dữ liệu chuyển động từ:
- a. Trái qua phải.
  - b. Từ phải qua trái.
  - c. Một trong hai trường hợp trên.
  - d. Không có trường hợp nào ở trên.





## CHƯƠNG 6: MẠCH PHÁT XUNG VÀ TẠO DẠNG XUNG

### GIỚI THIỆU

Hầu hết các hệ thống kỹ thuật số đều yêu cầu một vài loại dạng sóng định thời, ví dụ một nguồn xung của bộ dao động cần thiết cho tất cả các hệ thống tuần tự định thời. Trong các hệ thống kỹ thuật số, một dạng sóng xung vuông thường được sử dụng nhất. Sự tạo ra các dạng sóng xung vuông được gọi là bộ đa hài.

Có ba loại bộ đa hài:

- Bộ dao động đa hài (chạy tự do).
- Bộ đa hài đơn ổn (một nhịp).
- Bộ đa hài hai trạng thái ổn định (trigơ).

Một bộ dao động đa hài chỉ là một bộ dao động để tạo ra dạng xung. Nó có hai trạng thái chuẩn mà không yêu cầu sự kích hoạt từ bên ngoài. Bộ này thường được dùng làm xung điều khiển cho các mạch tuần tự.

Một bộ đa hài đơn ổn chỉ có một trạng thái ổn định, tức là trong điều kiện trạng thái ổn định thì đầu ra của nó cố định. Đầu ra này ở trạng thái LOW hoặc ở trạng thái HIGH. Mạch này cần một xung kích khởi từ bên ngoài để cho mạch chuyển sang trạng thái khác. Mạch này vẫn giữ nguyên trạng thái cũ trong một khoảng thời gian, khoảng thời gian này phụ thuộc vào các thành phần được dùng trong mạch. Trạng thái của mạch này được xem là trạng thái ổn định bởi vì nó phục hồi trở về trạng thái ổn định mà không cần bất kỳ xung kích hoạt nào từ bên ngoài. Độ rộng của xung kích khởi rất nhỏ, độ rộng của xung đầu ra chỉ phụ thuộc vào khoảng thời gian mà mạch giữ lại ở trạng thái ổn định. Mạch này được gọi là mạch một nhịp (one-shot) bởi vì một xung kích khởi chỉ tạo được một xung nhưng độ rộng xung lại khác. Mạch này rất hữu dụng bởi vì nó có thể tạo ra một xung tương đối dài (hàng chục mili giây) từ một xung hẹp, do đó nó còn được gọi là bộ giãn xung (pulse stretcher).

Ví dụ, một bộ vi xử lý có thể phát tín hiệu cho một thiết bị bên ngoài để in một nội dung nào đó bằng cách truyền qua một xung. Thiết bị đầu ra nói chung có tốc độ chậm hơn bộ vi xử lý, do đó nó yêu cầu một xung tín hiệu trong một khoảng thời gian lâu hơn. Điều này đạt được bằng một mạch giao tiếp có chứa bộ đa hài đơn ổn.

Một mạch đa hài trong đó cả hai trạng thái đều ổn định thì được gọi là mạch đa hài hai trạng thái ổn định hay trigơ. Mạch này thực hiện việc chuyển tiếp từ một trạng thái ổn định này sang một trạng thái ổn định khác chỉ lúc xung kích khởi được áp vào. Mạch này thường được dùng làm các thành phần trong bộ nhớ trong các hệ thống kỹ thuật số và đã được thảo luận ở chương 5.

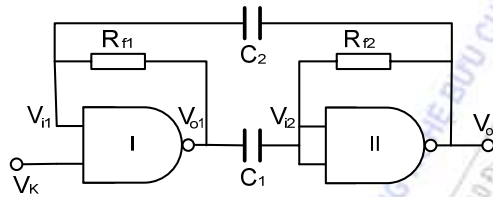
Chương này tập trung vào sơ đồ, nguyên tắc hoạt động, ứng dụng của các mạch dao động đa hài, mạch dao động đa hài đợi, trigơ Schmitt dựa trên các cổng TTL, CMOS và IC định thời 555. Sau chương này độc giả có thể tự thiết kế các mạch dao động theo các yêu cầu cơ bản cho các ứng dụng khác nhau.

## NỘI DUNG

### 6.1. MẠCH PHÁT XUNG

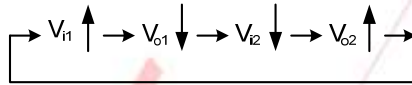
#### 6.1.1. Mạch dao động đa hài cơ bản cổng NAND TTL

Cổng NAND khi làm việc trong vùng chuyển tiếp có thể khuếch đại mạnh tín hiệu đầu vào. Nếu 2 cổng NAND được ghép điện dung thành mạch vòng như hình 6-1 ta được bộ dao động đa hài.  $V_K$  là đầu vào điều khiển, khi ở mức cao mạch phát xung, và khi ở mức thấp mạch ngừng phát.

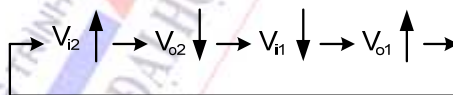


**Hình 6-1.** Bộ dao động đa hài cấu trúc bằng cổng NAND

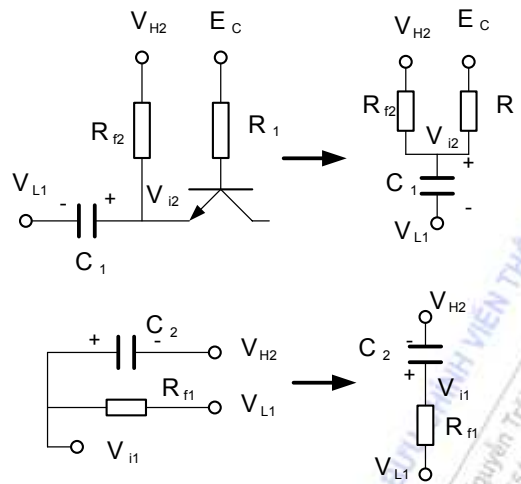
Nếu các cổng I và II thiết lập điểm công tác tĩnh trong vùng chuyển tiếp và  $V_K = 1$ , thì mạch sẽ phát xung khi được nối nguồn. Nguyên tắc làm việc của mạch như sau: Giả sử do tác động của nhiễu làm cho  $V_{i1}$  tăng một chút, lập tức xuất hiện quá trình phản hồi dương sau:



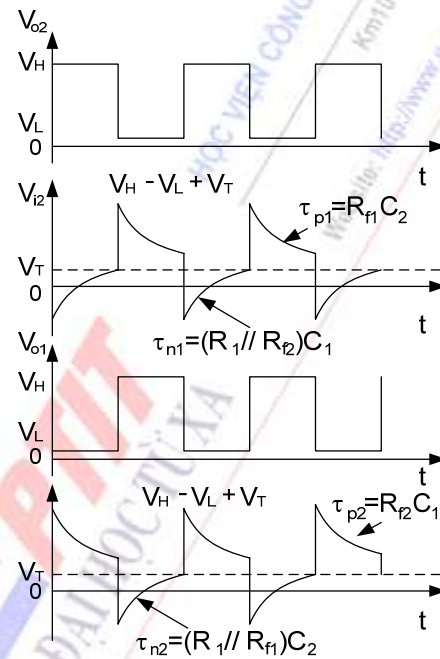
Khi đó, cổng I nhanh chóng trở thành thông bão hoà, cổng II nhanh chóng ngắt, mạch bước vào trạng thái tạm ổn định. Lúc này,  $C_1$  nạp điện và  $C_2$  phóng điện theo mạch đơn giản hoá được thể hiện trong hình 6-2.  $C_1$  nạp đến khi  $V_{i2}$  tăng đến ngưỡng thông  $V_T$ , trong mạch xuất hiện quá trình phản hồi dương như sau:



Kết quả quá trình này là: cổng I nhanh chóng ngắt còn cổng II thông bão hoà, mạch điện bước vào trạng thái tạm ổn định mới. Lúc này  $C_2$  nạp điện còn  $C_1$  phóng cho đến khi  $V_{i1}$  bằng ngưỡng thông  $V_T$  làm xuất hiện quá trình phản hồi dương đưa mạch về trạng thái ổn định ban đầu. Mạch không ngừng dao động, khi bỏ qua điện trở đầu ra của các cổng NAND, dựa vào hình 6-2 giản đồ xung của mạch được thể hiện trên hình 6-3.



Hình 6-2. Mạch vòng nạp phóng điện của tụ C1, C2



Hình 6-3. Dạng sóng gần đúng của điện áp tại các điểm trên mạch bộ dao động đa hài.

Vì thời gian nạp điện nhanh hơn thời gian phóng, nên thời gian duy trì trạng thái ổn định tạm thời phụ thuộc vào thời gian nạp điện của hai tụ điện C1 và C2. Từ hình 6-2 ta có thời gian nạp điện của tụ C1 là  $\tau_1 = (R_{f2} // R_1) C_1$ , thời gian để  $V_{i2}$  nạp điện đến  $V_T$  là:

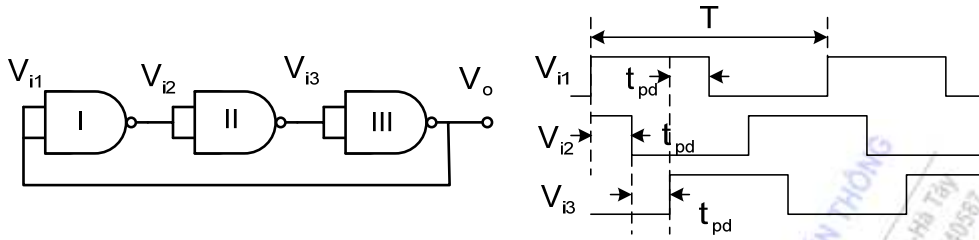
$$t_{M2} = (R_{f2} // R_1) C_1 \ln \frac{2V_{OH} - (V_T + V_{OL})}{V_{OH} - V_T}$$

Nếu  $R_{f1} = R_{f2} = R_f$ ,  $C_1 = C_2 = C$ ,  $V_{OH} = 3V$ ,  $V_{OL} = 0,35V$ ,  $V_T = 1,4V$  thì ta có:

$$T \approx 2(R_f // R_1)C$$

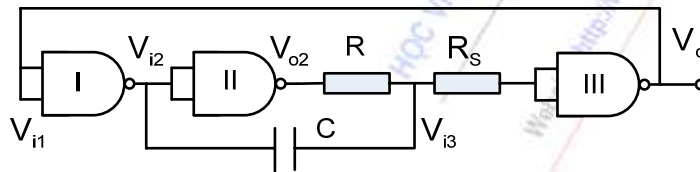
T là chu kỳ của tín hiệu đa hài lối ra.

6.1.2. Mạch dao động đa hài vòng RC



Hình 6-4. Bộ dao động vòng và dạng sóng

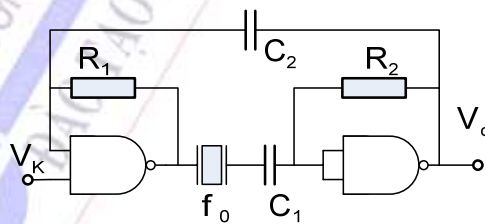
Bộ dao động vòng có cấu trúc gồm 3 cổng NAND mắc nối tiếp như hình 6-4. Phản hồi dương từ  $V_o$  đến  $V_{i1}$  làm cho mạch này không có trạng thái ổn định. Tần số của tín hiệu lối ra phụ thuộc vào thời gian trễ của cổng NAND, và không thể điều chỉnh được tần số này. Tần số của mạch phát sẽ điều chỉnh được khi một mạch trễ RC được mắc thêm vào mạch như hình 6-5. Tần số dao động của mạch điều chỉnh được thông qua giá trị của tụ điện C và điện trở R.



Hình 6-5. Bộ dao động đa hài có mạch RC

6.1.3. Mạch dao động đa hài thạch anh

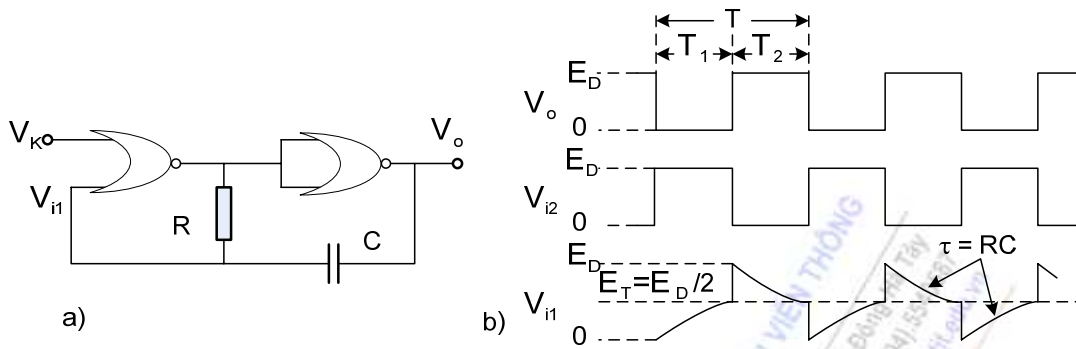
Để có các tín hiệu đồng hồ có tần số chính xác và có độ ổn định cao, các mạch đa hài trình bày trên đây không đáp ứng được. Tinh thể thạch anh thường được sử dụng trong các trường hợp này. Thạch anh có tính ổn định tần số tốt, hệ số phẩm chất rất cao dẫn đến tính chọn lọc tần số rất cao. Hình 6-6 là một mạch dao động đa hài điển hình sử dụng tinh thể thạch anh. Tần số của mạch dao động chỉ phụ thuộc vào tinh thể thạch anh mà không phụ thuộc vào giá trị các tụ điện và điện trở trong mạch.



Hình 6-6. Mạch dao động đa hài thạch anh

6.1.4. Mạch dao động đa hài CMOS

Hình 6-7a là mạch dao động đa hài cơ bản sử dụng hai cổng NOR CMOS và các linh kiện định thời trở và tụ. Giản đồ xung của mạch được thể hiện trên hình 6-7b. Chu kỳ dao động của mạch được tính gần đúng như sau:

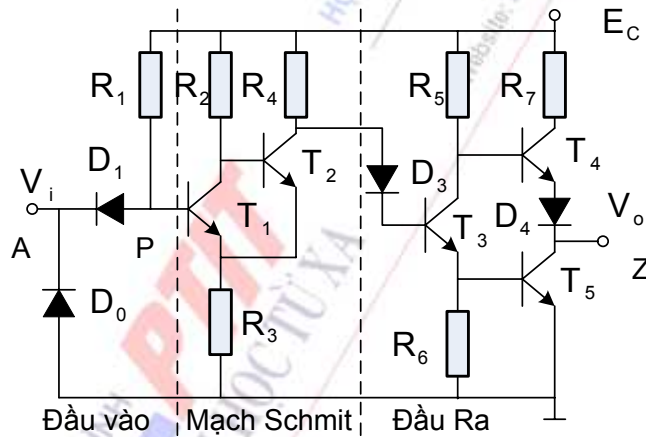


Hình 6-7. Bộ dao động đa hài dùng cổng NOR CMOS và giản đồ xung

$$T = T_1 + T_2 = RC \ln \left( \frac{E_D}{E_D - V_T} + \frac{E_D}{V_T} \right)$$

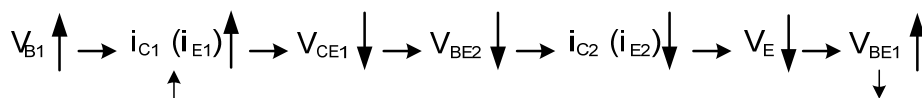
Nếu giả thiết  $V_T = E_D/2$  thì  $T_1 = T_2$ , khi đó  $T = RC \ln 4 \approx 1,4RC$ .

## 6.2. TRIGƠ SCHMIT

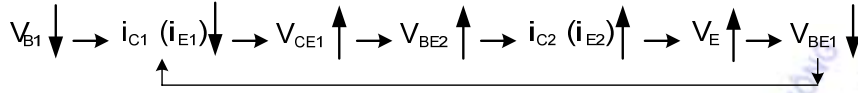


Hình 6-8. Sơ đồ nguyên lý của trigơ Schmit

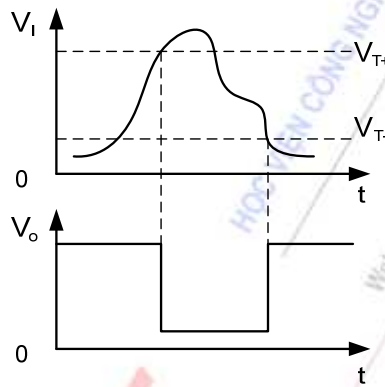
Hình 6-8 là sơ đồ nguyên lý của trigơ schmitt, hay còn được gọi là bộ đảo pha trigơ schmit. Nó gồm 3 phần: mạch đầu vào, mạch schmit và tầng công suất lối ra. Nguyên tắc làm việc của mạch như sau: Nếu  $V_{B1}$  ở mức thấp thì  $T_1$  ngắt,  $T_2$  thông bão hoà và ngược lại nếu  $V_{B1}$  ở mức cao thì  $T_1$  thông bão hoà,  $T_2$  ngắt. Khi  $V_{B1}$  tăng từ mức thấp lên mức cao đến trị số  $V_{BE1} = V_{B1} - I_1 R_3 = 0,5 \text{ V}$  thì  $T_1$  bắt đầu chuyển từ trạng thái ngắt vào trạng thái khuếch đại. Do  $V_{B1}$  tiếp tục tăng nên  $V_{CE1} = V_{BE2}$  giảm xuống. Sau khi  $T_2$  rời khỏi trạng thái bão hoà mà  $V_{B1}$  tiếp tục tăng thì xảy ra quá trình phản hồi dương sau:



Nhờ phản hồi dương mạch điện nhanh chóng chuyển sang trạng thái  $T_1$  thông bão hoà,  $T_2$  ngắt. Nếu  $V_{B1}$  sau khi tăng đến cực đại thì bắt đầu giảm; khi  $V_{B1}$  giảm đến mức làm  $T_1$  ra khỏi vùng bão hoà,  $T_2$  ra khỏi vùng ngắt thì mạch điện lại xảy ra quá trình phản hồi dương sau:



Kết quả mạch điện nhanh chóng lật sang trạng thái  $T_1$  ngắt,  $T_2$  thông bão hoà. Chúng ta gọi giá trị điện áp đầu vào  $V_I$  trong quá trình tăng lên của nó đạt đến ngưỡng làm lật mạch schmit để đầu ra từ mức cao xuống mức thấp là ngưỡng trên  $V_{T+}$  và giá trị ngược lại là ngưỡng dưới của trigơ schmit  $V_{T-}$  (hình 6- 9). Hiệu điện áp tương ứng với ngưỡng trên và ngưỡng dưới được gọi là độ chênh lệch điện áp chuyển mạch  $\Delta V = V_{T+} - V_{T-}$ .



**Hình 6-9.** Dạng sóng đầu vào  $V_I$  và đầu ra  $V_O$  của trigơ schmit

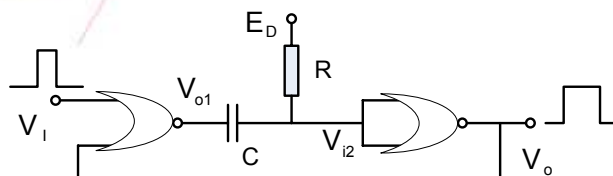
Trigơ schmit thực chất là một bộ so sánh hai ngưỡng nên nó được dùng ứng dụng khác nhau như: Các mạch dao động, các mạch so sánh, lọc nhiễu v.v..

### 6.3. MẠCH ĐA HÀI ĐỢI

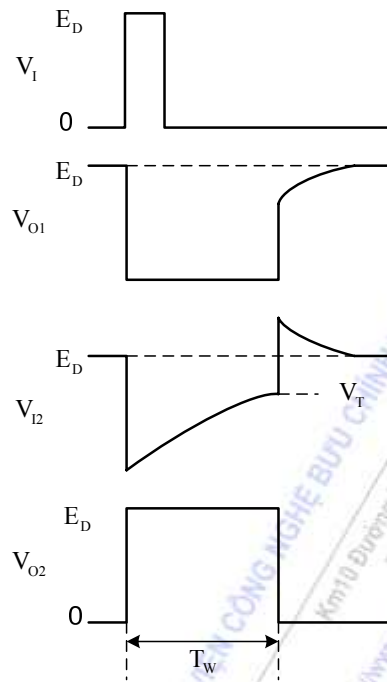
Mạch đa hài đợi có một trạng thái ổn định và một trạng thái tạm ổn định. Khi có tác dụng của xung ngoài, mạch có thể chuyển đổi từ trạng thái ổn định sang trạng thái tạm ổn định. Sau khi duy trì một thời gian, mạch sẽ tự động quay lại trạng thái ổn định. Thời gian tạm ổn định phụ thuộc vào các thông số của mạch mà không phụ thuộc vào xung kích. Mạch đa hài đợi được ứng dụng trong các mạch định thời, tạo dạng xung, trễ v.v..

#### 6.3.1. Mạch đa hài đợi CMOS

##### 1. Mạch đa hài đợi kiểu vi phân



**Hình 6-10.** Đa hài đợi kiểu vi phân dùng cổng NOR CMOS



**Hình 6-11.** Dạng sóng của mạch đa hài đợi kiểu vi phân

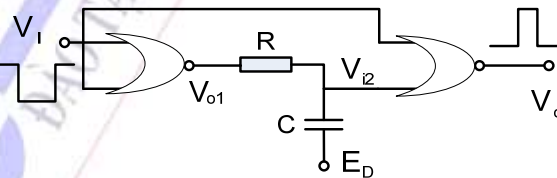
Hình 6-10 là sơ đồ nguyên lý của mạch đa hài đợi kiểu vi phân. Tại trạng thái ổn định,  $V_1=0$  thì  $V_{01}=E_D$ ,  $V_{12}=E_D$ ,  $V_{02}=0$ . Khi có một xung kích thích lối vào làm cho cổng 1 nhanh chóng cấm và lối ra bằng 0, xem giản đồ 6-11. Mạch điện RC sẽ nạp điện cho tụ điện C. Trong quá trình nạp, điện áp  $V_{12}$  tăng dần đến ngưỡng  $V_T$  và làm cổng 2 đóng, điện áp  $V_{02}=0$ . Khi đó, cổng 1 nhanh chóng chuyển về trạng thái cấm và làm cho mạch đa hài đợi trở về trạng thái ổn định.

Độ rộng xung tại đầu ra của mạch được xác định bằng công thức sau:

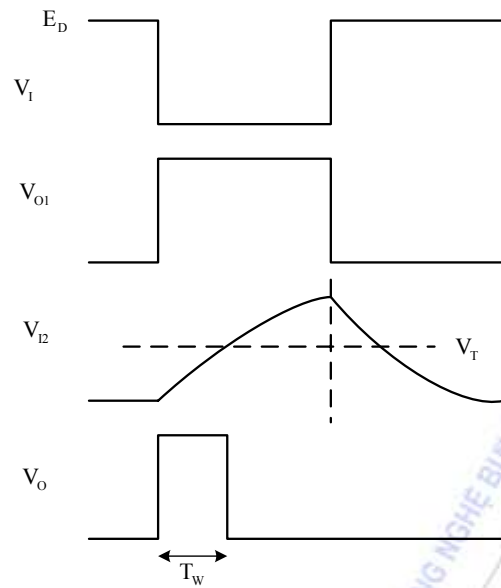
$$T_w = (R + R_0) C * \ln \frac{E_D}{E_D - V_T}$$

trong đó  $R_0$  là điện trở đầu ra của cổng 1, nếu  $V_T=E_D/2$  thì  $T_w = 0,7(R + R_0)C$

## 2. Mạch đa hài đợi kiểu tích phân dùng cổng NOR CMOS



**Hình 6-12.** Đa hài đợi kiểu tích phân dùng cổng NOR CMOS



**Hình 6-13.** Dạng sóng của mạch đa hài đơn kiểu tích phân

Hình 6-12 biểu diễn sơ đồ nguyên lý của mạch đa hài đơn kiểu tích phân. Tại trạng thái ổn định,  $V_I=1$  thì  $V_{O1}=0$ ,  $V_{I2}=0$ ,  $V_{O2}=0$ . Khi lối vào  $V_I$  chuyển từ 1 xuống 0 lối ra  $V_{O2}$  nhảy từ trạng thái 0 lên 1 và đồng thời mạch RC bắt đầu tích điện cho tụ điện C, khi điện áp  $V_{I2} = V_T$  điện áp lối ra  $V_{O2}$  chuyển xuống trạng thái 0. Sau khi hết xung lối vào tụ điện phóng điện thông qua trở R và mạch trở về trạng thái ổn định.

Độ rộng xung lối ra của mạch đa hài đơn được tính theo công thức:

$$T_w = (R + R_0)C * \ln \frac{E_D}{E_D - V_T}$$

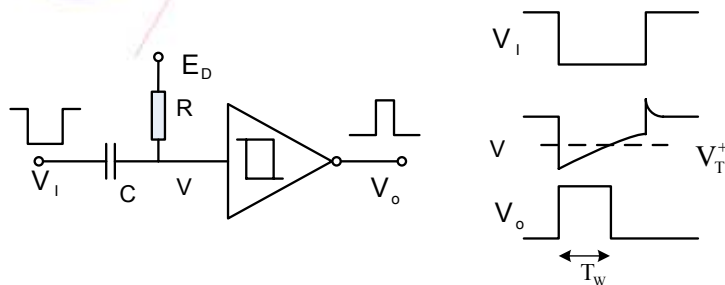
trong đó  $R_0$  là điện trở đầu ra của cổng 1, nếu  $V_T=E_D/2$  thì  $T_w = 0,7(R + R_0)C$

### 3. Mạch đa hài đơn dùng trigơ Schmitt

Dựa vào đặc tính so sánh của trigơ Schmitt, mạch nguyên lý chỉ ra trên hình 6-14 là bộ đa hài đơn. Độ rộng xung lối ra phụ thuộc vào ngưỡng trên của trigơ Schmitt và giá trị của tụ điện C và điện trở R theo công thức sau:

$$T_w = RC * \ln \frac{E_D}{E_D - V_T^+}$$

nếu  $V_T=E_D/2$  thì  $T_w = 0,7RC$





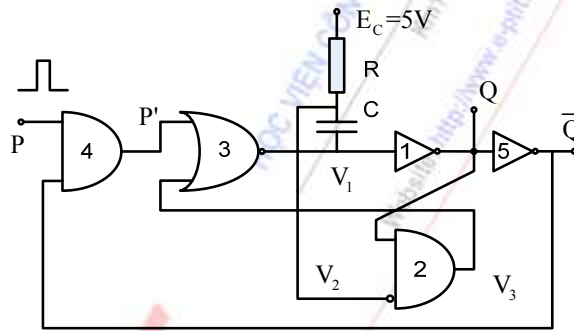
**Hình 6-14.** Sơ đồ nguyên lý và giản đồ thời gian của mạch đa hài dùng trigơ Schmitt

**6.3.2. Mạch đa hài đợi TTL**

Hình 6-15 là sơ đồ nguyên lý mạch đa hài đợi họ TTL, trong đó các cổng 1, 2, 3 cấu trúc lên mạch flip-flop, cổng 4,5 là mạch tạo dạng xung. Các cổng này thuộc họ TTL nên có mức logic 1 là 3,6 V và logic 0 là 0,3 V. Đầu vào  $V_2$  biểu thị sử dụng mạch đảo. Mạch đảo này thông bão hoà thì  $V_2 \sim 0,7$  V, còn ngưỡng thông của nó cỡ 0,6 V.

Tại trạng thái ổn định  $P = P' = 0$ . Mạch đảo đầu vào  $V_2$  là bộ khuếch đại transistor emitter chung ở trạng thái bão hoà và khi đó  $V_2 = 0,7$  V,  $V_3 = 0$ ,  $V_1 = 1$ ,  $Q = 0$ ,  $\bar{Q} = 1$ .

Khi có xung dương tác động ở đầu vào thì  $P = 1$ ,  $P' = 1$ ,  $V_1 = 0$ ,  $Q = 1$ ,  $\bar{Q} = 0$ , mạch ở trạng thái tạm ổn định. Do  $\bar{Q} = 0$  khoá cổng 4, nên sau khi bị kích thích bởi sườn dương xung P thì mạch bị cách ly khỏi xung P.



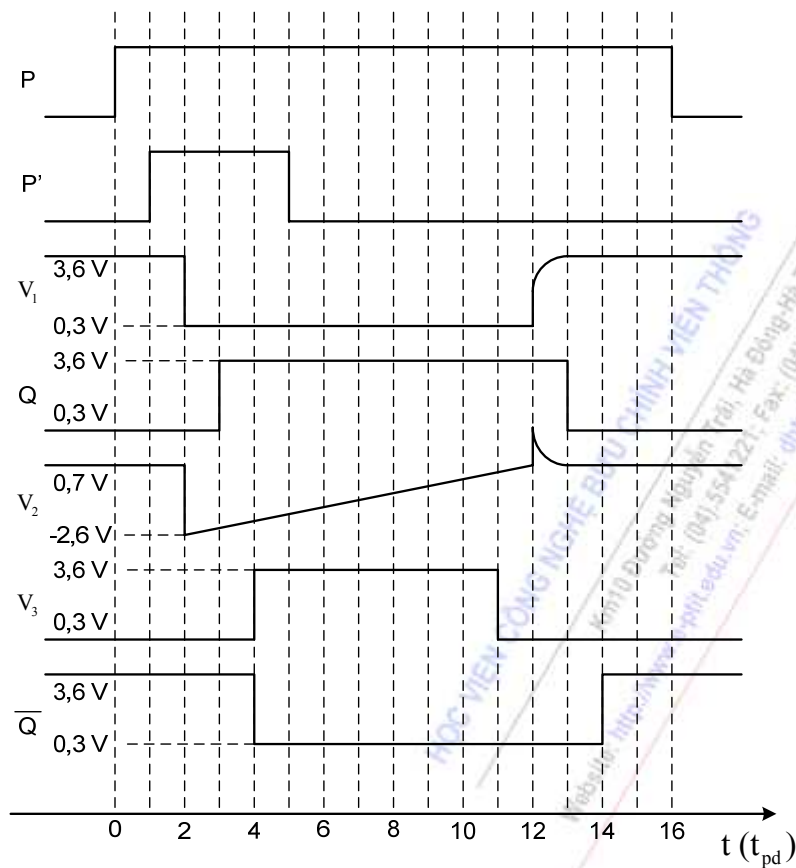
**Hình 6-15.** Sơ đồ nguyên lý mạch đa hài đợi họ TTL

Vì điện áp trên tụ C không tăng đột biến nên khi  $V_1$  từ mức cao 3,6 V đột biến xuống 0,3 V thì  $V_2$  từ mức 0,7 V đột biến xuống -2,6 V. Bắt đầu quá trình nạp điện của tụ điện C.  $V_2$  tăng dần lên. Khi  $V_2$  Tăng lên đến ngưỡng thông 0,6 V thì sinh ra quá trình phản hồi dương sau:

$$V_2 \uparrow \rightarrow V_3 \downarrow \rightarrow V_1 \uparrow \rightarrow Q \downarrow$$

Quá trình này làm mạch nhanh chóng trở về trạng thái ổn định ban đầu  $V_3 = 0$ ,  $V_1 = 1$ ,  $Q = 0$ ,  $\bar{Q} = 1$ . Tiếp đó tụ điện C phóng điện,  $V_2$  dần dần hồi phục về 0,7 V. Hình 6-16 chỉ ra giản đồ xung của mạch đa hài đợi họ TTL với giả thiết thời gian trễ truyền đạt của các cổng và bộ đảo pha đều bằng  $t_{pd}$ .

Độ rộng xung ra được tính theo công thức  $T_w = 0,7RC$ . Mạch dao động đa hài đợi được thiết kế sẵn trong một số họ IC TTL như 74LS121, 74LS123 ... bằng cách thay đổi các giá trị tụ và trở mắc ngoài sẽ cho các xung lối ra mong muốn.



Hình 6-16. Giản đồ xung của mạch dao động đa hài đợi TTL với giả thiết độ trễ của các cổng là  $t_{pd}$ .

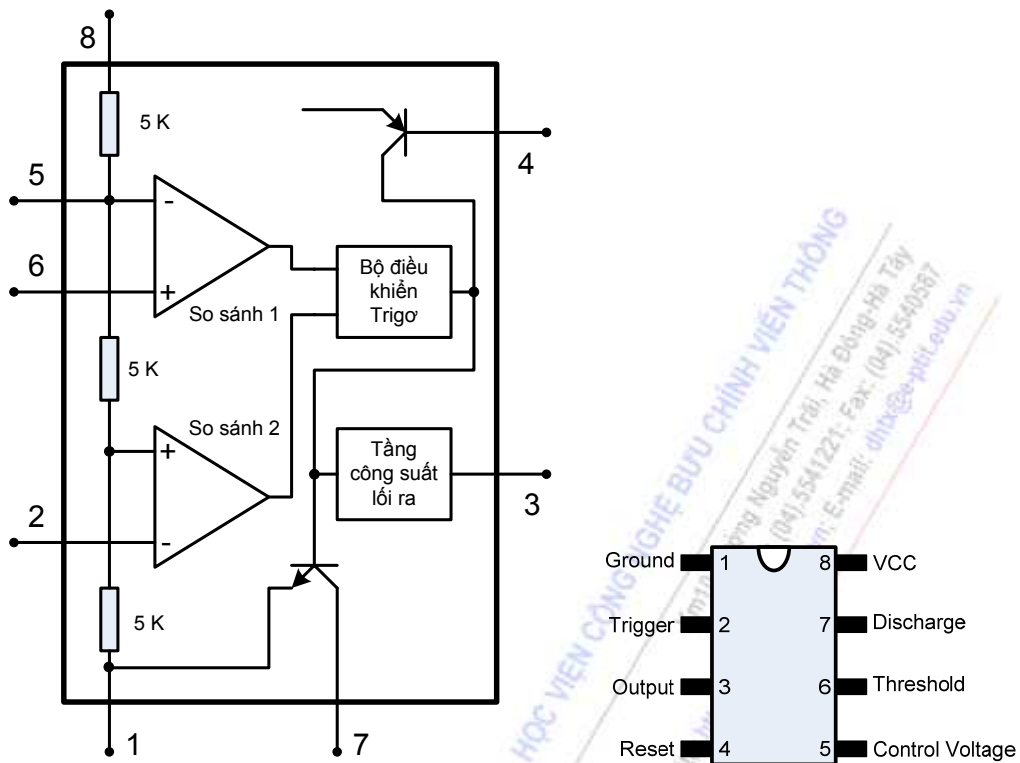
#### 6.4. IC ĐỊNH THỜI

Bộ định thời 555 được sử dụng rất rộng rãi trong các bộ dao động đa hài, đa hài đợi, và các bộ so sánh v.v... Hình 6-17 là sơ đồ khối nguyên lý của IC định thời này, trong đó chức năng của các chân được chỉ ra trong bảng sau:

Chân	Chức năng	Chân	Chức năng
1	Đất - GND	5	Điện áp điều khiển
2	Chân kích thích	6	Chân ngưỡng
3	Đầu ra	7	Đầu phóng điện
4	Xoá - Reset	8	Nguồn - Vcc

Bảng chức năng của IC 555

TH	TRIG	$\bar{R}$	OUT	DIS
X	X	L	L	Thông
$> \frac{2}{3} E_c$	$> \frac{1}{3} E_c$	H	L	Thông
$< \frac{2}{3} E_c$	$> \frac{1}{3} E_c$	H	Không đổi	Không đổi
X	$> \frac{1}{3} E_c$	H	H	Ngắt

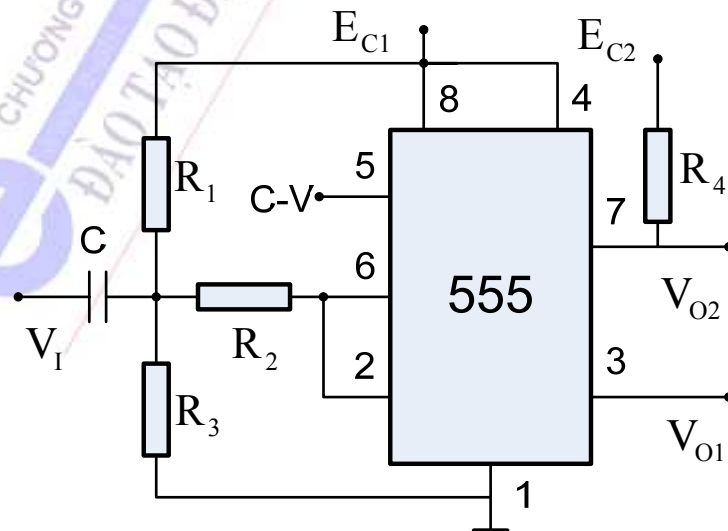


Hình 6-17. Sơ đồ khối nguyên lý của IC định thời 555

### Một vài ứng dụng của IC định thời 555

#### 1) Trigo Schmitt

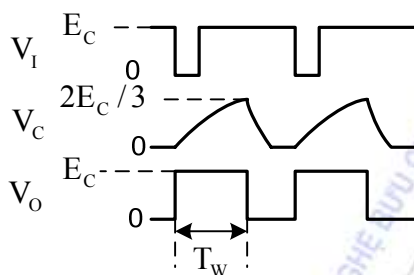
Hình 6-18 là sơ đồ nguyên lý của trigo schmitt dùng IC 555. Với sơ đồ này ngưỡng trên  $V_{T+} = \frac{2}{3} * E_{C1}$  và ngưỡng dưới  $V_{T-} = \frac{1}{3} * E_{C1}$ . Độ chênh lệch điện áp  $\Delta V = V_{T+} - V_{T-} = \frac{1}{3} * E_{C1}$ . Nếu đưa điện áp vào đầu vào C-V thì có thể điều chỉnh được  $V_{T+}$ ,  $V_{T-}$  và  $\Delta V$ .



Hình 6-18. Mạch trigo Schmitt dùng IC 555

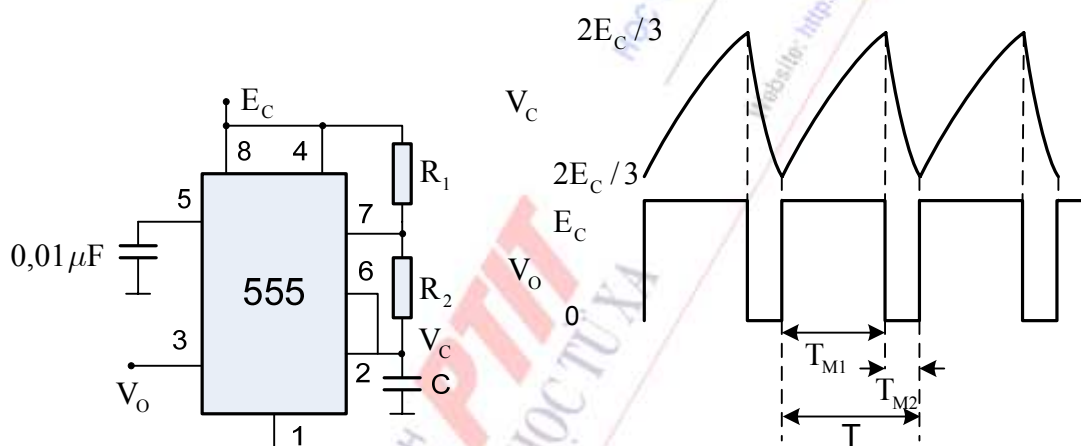
## 2) Mạch đa hài đợi

Hình 6-19 là sơ đồ nguyên lý và giản đồ thời gian của mạch đa hài đợi dùng IC 555, trong đó RC là mạch định thời. Độ kéo dài xung lồi ra được xác định bằng công thức  $T_w \approx RC \ln 3 \approx 1,1RC$ . Mạch dao động đa hài đợi này yêu cầu độ rộng xung lồi vào nhỏ hơn độ rộng xung lồi ra, nếu nó lớn hơn thì yêu cầu dùng thêm mạch vi phân ở lồi vào.



Hình 6-19. Mạch đa hài đợi dùng IC 555 và dạng sóng

## 3) Mạch đa hài



Hình 6-20. Mạch đa hài dùng IC 555 và dạng sóng

Hình 6-20 là sơ đồ mạch đa hài và dạng sóng, điện trở  $R_1$ ,  $R_2$  và tụ điện  $C$  đóng vai trò là mạch định thời. Chu kỳ dao động của tín hiệu lồi ra được xác định thông qua thời gian phóng và nạp của tụ điện  $C$  như sau:

$$T_{M1} = (R_1 + R_2)C * \ln 2 = 0,7(R_1 + R_2)C$$

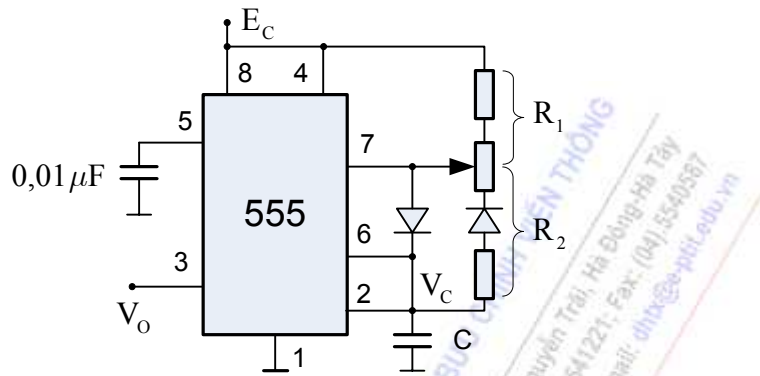
$$T_{M2} = R_2C * \ln 2 = 0,7R_2C$$

$$T = T_{M1} + T_{M2} = 0,7(R_1 + 2R_2)C$$

$$f = 1/T = \frac{1,43}{(R_1 + 2R_2)C}$$

Như ta thấy xung lồi ra có độ lấp đầy phụ thuộc vào cả điện trở  $R_1$  và  $R_2$  và không thể tạo ra xung vuông với độ lấp đầy bằng 50% thông qua việc thay đổi giá trị  $R_1$  và  $R_2$ . Để có được xung vuông với độ lấp đầy bằng 50%, người ta sử dụng mạch có thêm 2 diode khi đó trở phóng và

nạp điện cho Tụ có thể thay đổi độc lập và tạo ra xung mong muốn. Hình 6-21 là sơ đồ nguyên lý của mạch đa hài dùng IC 555 mà độ lặp đầy có thể thay đổi được.



Hình 6-21. Mạch đa hài điều chỉnh được độ lặp đầy xung dùng IC 555

## TÓM TẮT

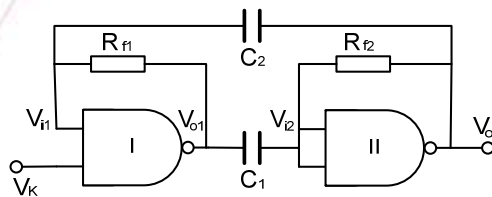
Trong chương này chúng ta đã tìm hiểu các mạch tạo xung. Mạch dao động xung tự kích không cần tín hiệu ngoài đưa vào; sau khi được cấp nguồn một chiều mạch tự động sinh ra xung vuôn. Thuộc loại dao động tự kích này có các mạch: bộ dao động đa hài cơ bản cổng NAND họ TTL, bộ dao động vòng, bộ dao động thạch anh, bộ dao động đa hài cơ bản CMOS.

Mạch tạo dạng xung không tự động phát xung nhưng có thể biến tín hiệu đầu vào hình dạng khác thành xung vuôn theo yêu cầu của mạch số. Trong số mạch tạo dạng xung, chúng ta đã tìm hiểu: trigơ Schmit và đơn ổn.

Cách mạch phát xung và tạo dạng xung trên đây, ngoài dùng làm xung đồng hồ ra còn có ứng dụng vô cùng rộng rãi trong các hệ thống xung - số. Bộ dao động đa hài thường dùng làm bộ tạo xung chuẩn thời gian và chuẩn tần số. Mạch đơn ổn thường dùng để định thời và làm trễ xung. Trigơ Schmit ngoài ứng dụng tạo dạng xung còn ứng dụng so sánh mức và giám sát mức...

## CÂU HỎI ÔN TẬP

1. Trong mạch dao động đa hài cơ bản dùng cổng NAND họ TTL, hình 6-1, nếu giá trị trị điện trở  $R_{f1} = 5 \cdot R_{f2} = 10 \text{ k}\Omega$ , giá trị  $C_1 = C_2 = 1 \mu\text{F}$  thì mạch có hoạt động không? dạng tín hiệu tương đối lối ra sẽ như thế nào?

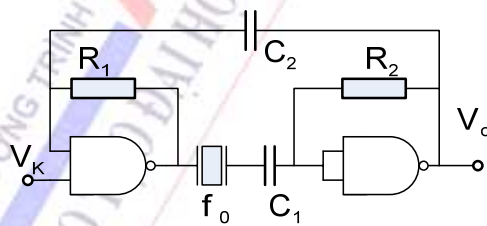


Hình 6-1. Bộ dao động đa hài cấu trúc bằng cổng NAND

- Lỗi ra luôn ở mức logic thấp
  - Lỗi ra luôn ở mức logic cao
  - Tín hiệu lỗi ra là xung vuông với độ lấp đầy nhỏ hơn 50%
  - Tín hiệu lỗi ra là xung vuông có độ lấp đầy lớn hơn 50%
2. Với câu hỏi như câu 1 và giả thiết  $R_1 = 3 \text{ k}\Omega$ , tính tần số dao động của mạch và vẽ dạng sóng lỗi ra.

- $f = 28 \text{ Hz}$  và dạng sóng lỗi ra có dạng : 
- $f = 28 \text{ Hz}$  và dạng sóng lỗi ra có dạng : 
- $f = 28 \text{ Hz}$  và dạng sóng lỗi ra có dạng : 
- $f = 0 \text{ Hz}$  và dạng sóng lỗi ra có dạng : 

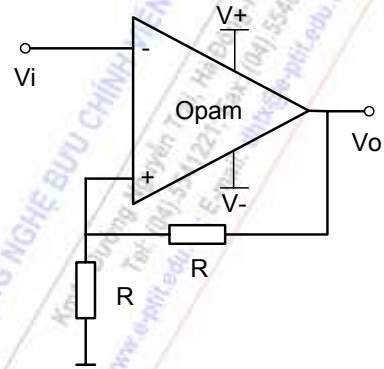
3. Đặc điểm nổi bật nhất của mạch dao động đa hài dùng thạch anh là gì?
- Biên độ tín hiệu lỗi ra ổn định
  - Tần số tín hiệu lỗi ra ổn định
  - Biên độ lỗi ra có thể điều chỉnh được
  - Tần số lỗi ra có thể điều chỉnh được
4. Trong mạch dao động đa hài dùng thạch anh như hình 6-6, nếu không có tụ  $C_1$ , lỗi ra của thạch anh được nối trực tiếp với đầu vào của cổng NAND thứ hai thì mạch:



Hình 6-6. Mạch dao động đa hài thạch anh

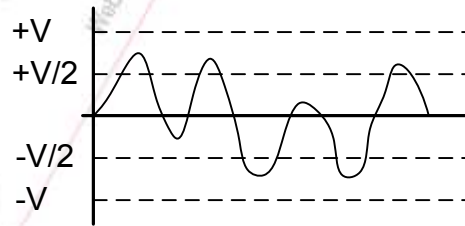
- Không dao động lỗi ra luôn thấp
  - Không dao động lỗi ra luôn cao
  - Có xung lỗi ra nhưng tần số thay đổi
  - Tần số xung lỗi ra không thay đổi
5. Đặc điểm quan trọng nhất của trigơ Schmitt là gì?

- a. Tần số hoạt động cao
  - b. Tính chống nhiễu cao vì nó hoạt động như bộ so sánh hai ngưỡng
  - c. Công suất tiêu thụ thấp
  - d. Là bộ so sánh một ngưỡng
6. Mạch có sơ đồ nguyên lý như hình sau có chức năng như thế nào?

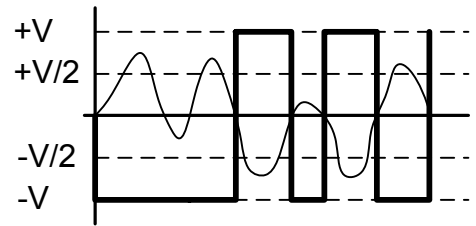
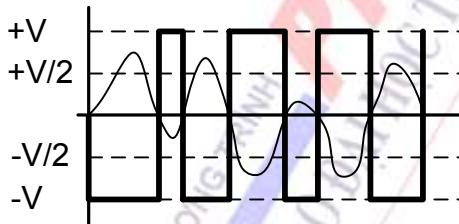


- a. Bộ so sánh một ngưỡng
- b. Trigon Schmitt
- c. Mạch dao động đa hài
- d. Mạch dao động đa hài đợi

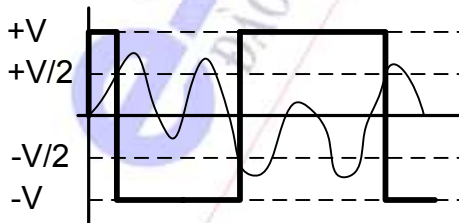
7. Với mạch điện như câu hỏi 6, nếu tín hiệu lối vào có dạng tín hiệu như hình sau, tín hiệu lối ra nằm ở hình nào.



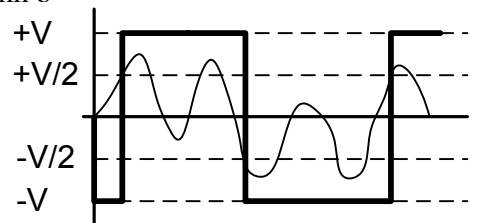
Hình a.



Hình b



Hình c



Hình d

- a. Hình a.
  - b. Hình b.
  - c. Hình c.
  - d. Hình d.
8. Chức năng của mạch đa hài đợi là gì?
- a. Là mạch phát xung vuông
  - b. Là mạch dao động đa hài có chân điều khiển
  - c. Là mạch dao động đa hài có một trạng thái ổn định và một trạng thái tạm ổn định
  - d. Là mạch phát xung điều hoà
9. Trong mạch đa hài đợi kiểu vi phân như hình 6-10, nếu xung điều khiển có độ rộng lớn hơn xung đa hài đợi lối ra thì :
- a. Mạch vẫn hoạt động bình thường
  - b. Tín hiệu lối ra luôn bằng 0
  - c. Tín hiệu lối ra luôn bằng 1
  - d. Xung lối ra bằng xung lối vào
10. Trong mạch đa hài hình 6-20, nếu điện trở  $R_2$  bị nối tắt thì:
- a. Mạch vẫn phát xung và tần số lối ra chỉ phụ thuộc vào giá trị của  $R_1$  và  $C$
  - b. Xung lối ra là xung vuông có độ lấp đầy là 50%
  - c. Mạch vẫn phát xung nhưng tần số rất cao
  - d. Không có tín hiệu lối ra



## CHƯƠNG 7: BỘ NHỚ BÁN DẪN

### GIỚI THIỆU

Bộ nhớ bán dẫn thay thế các loại bộ nhớ bằng vật liệu từ. Các tiến bộ mới của công nghệ bán dẫn trong thời gian gần đây đã cung cấp nhiều mạch nhớ loại MSI và LSI có độ tin cậy cao và giá thành hạ. Vào đầu thập kỷ 60 của thế kỷ 20, giá thành thương phẩm của một bit nhớ vào khoảng 2 USD. Đến nay (những năm đầu thế kỷ 21), giá thương phẩm của 128 Mbyte vào khoảng 20 USD. Như vậy giá thành thương phẩm của một bit nhớ sau khoảng 40 năm đã giảm đi khoảng  $105.10^6$  lần. Bộ nhớ bán dẫn điển hình có các tế bào nhớ sắp xếp theo hình chữ nhật, gắn trong khối hộp nhỏ bằng nhựa dạng DIP (Dual in line package). Tế bào nhớ cơ bản là một mạch trigơ, transistor hay mạch có khả năng tích trữ điện tích, tế bào nhớ này dùng để lưu trữ một bit tin.

Trong phần này giới thiệu một số bộ nhớ bán dẫn cơ bản.

### NỘI DUNG

#### 7.1. KHÁI NIỆM CHUNG

##### 7.1.1. Khái niệm

Bộ nhớ là một thiết bị có khả năng lưu trữ thông tin (nhị phân). Muốn sử dụng bộ nhớ, trước tiên ta phải ghi dữ liệu và các thông tin cần thiết vào nó, sau đó lúc cần thiết phải lấy dữ liệu đã ghi trước đó để sử dụng. Thủ tục ghi vào và đọc ra phải được kiểm soát chặt chẽ, tránh nhầm lẫn nhờ định vị chính xác từng vị trí ô nhớ và nội dung của nó theo một mã địa chỉ duy nhất.

##### 7.1.2. Những đặc trưng chính của bộ nhớ

###### 7.1.2.1. Dung lượng của bộ nhớ.

Dung lượng bộ nhớ là số bit thông tin tối đa có thể lưu giữ trong nó. Dung lượng cũng có thể biểu thị bằng số từ nhớ  $n$  bit. **Từ nhớ  $n$  bit** là số bit ( $n$ ) thông tin mà ta có thể đọc hoặc ghi đồng thời vào bộ nhớ. Ví dụ: Một bộ nhớ có dung lượng là 256 bit; nếu nó có cấu trúc để có thể truy cập cùng một lúc 8 bit thông tin, thì ta cũng có thể biểu thị dung lượng bộ nhớ là 32 từ nhớ  $\times$  8 bit = 32 byte.

###### 7.1.2.2. Cách truy cập thông tin.

Các bộ nhớ có thể có một trong hai cách truy cập thông tin.

**Truy cập trực tiếp**, hay còn gọi là truy cập ngẫu nhiên (random access). Ở cách này, không gian bộ nhớ được chia thành nhiều ô nhớ. Mỗi ô nhớ chứa được 1 từ nhớ  $n$  bit và có một địa chỉ xác định, mã hoá bằng số nhị phân  $k$  bit. Như vậy, người sử dụng có thể truy cập trực tiếp thông tin ở ô nhớ có địa chỉ nào đó trong bộ nhớ. Mỗi bộ nhớ có  $k$  bit địa chỉ sẽ có  $2^k$  ô nhớ và có thể ghi được  $2^k$  từ nhớ  $n$  bit.

**Truy cập liên tiếp** (serial access) hay còn gọi là kiểu truy cập tuần tự. Các đĩa từ, băng từ, trống từ, thanh ghi dịch... có kiểu truy cập này. Các bit thông tin được đưa vào và lấy ra một cách tuần tự.

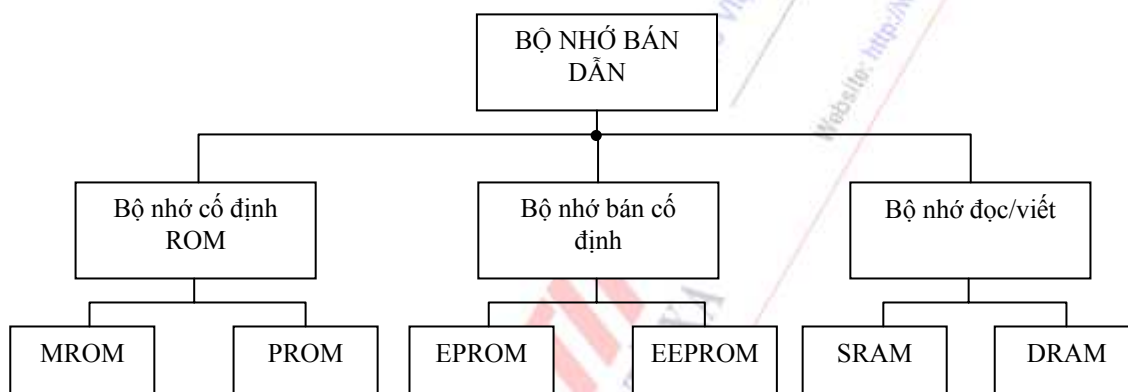
### 7.1.2.3. Tốc độ truy cập thông tin.

Đây là thông số rất quan trọng của bộ nhớ. Nó được đặc trưng bởi thời gian cần thiết để truy cập thông tin.

Thời gian truy cập thông tin ở các bộ nhớ truy cập kiểu trực tiếp gồm thời gian tìm địa chỉ của ô nhớ và thời gian đọc/viết thông tin trên đó. Thời gian truy cập thông tin phụ thuộc chủ yếu vào công nghệ chế tạo. Với công nghệ MOS thì thời gian truy cập khoảng 30 đến vài trăm ns.

Ở các bộ nhớ truy cập kiểu tuần tự, thời gian truy cập phụ thuộc vào vị trí của thông tin cần truy cập trong tập tin (file). Đối với các băng từ, đĩa từ thời gian truy cập của nó được định nghĩa là thời gian trung bình hoặc cực đại để truy cập một thông tin và nằm trong khoảng vài msec đến nhiều sec.

### 7.1.3. Phân loại



Dựa trên thời gian viết và cách viết, có thể chia thành bộ nhớ cố định, bộ nhớ bán cố định và bộ nhớ đọc/viết được. Bộ nhớ có nội dung được viết sẵn một lần khi chế tạo được gọi là bộ nhớ cố định và được ký hiệu là ROM (Read Only Memory). Sau khi đã được viết (bằng mặt nạ-mask) từ nhà máy thì ROM loại này không viết lại được nữa đó chính là MROM. PROM là một dạng khác, các bit có thể được viết bằng thiết bị ghi của người sử dụng trong một lần (Programmable ROM).

Bộ nhớ có thể đọc/ viết nhiều lần được gọi là RAM (Random Access Memory) gồm hai loại: bộ nhớ RAM tĩnh-SRAM (Static RAM) thường được xây dựng trên các mạch điện tử trigơ và RAM động-DRAM (Dynamic RAM) được xây dựng trên cơ sở nhớ các điện tích ở tụ điện; bộ nhớ này phải được hồi phục nội dung đều đặn, nếu không nội dung sẽ mất đi theo sự rò điện tích trên tụ. Giữa ROM và RAM có một lớp các bộ nhớ được gọi là EPROM (Erasable PROM), dữ liệu trong đó có thể xoá được bằng tia cực tím và ghi lại được, EEPROM (Electric EPROM) có thể xoá được bằng dòng điện. Các loại này còn được gọi là bộ nhớ bán cố định. Các bộ nhớ DRAM thường thoả mãn những yêu cầu khi cần bộ nhớ có dung lượng lớn; trong khi đó khi cần có tốc độ truy xuất lớn thì phải dùng các bộ nhớ SRAM có giá thành đắt hơn. Nhưng cả hai loại này đều có nhược điểm là thuộc loại “bay hơi” (volatile), thông tin sẽ bị mất đi khi nguồn nuôi bị

ngắt. Do vậy các chương trình dùng cho việc khởi động PC như BIOS thường phải nạp trên các bộ nhớ ROM.

#### 7.1.4. Tổ chức của bộ nhớ

Bộ nhớ thường được tổ chức gồm nhiều vi mạch nhớ được ghép lại để có độ dài từ và tổng số từ cần thiết. Những chip nhớ được thiết kế sao cho có đầy đủ một số chức năng của bộ nhớ như:

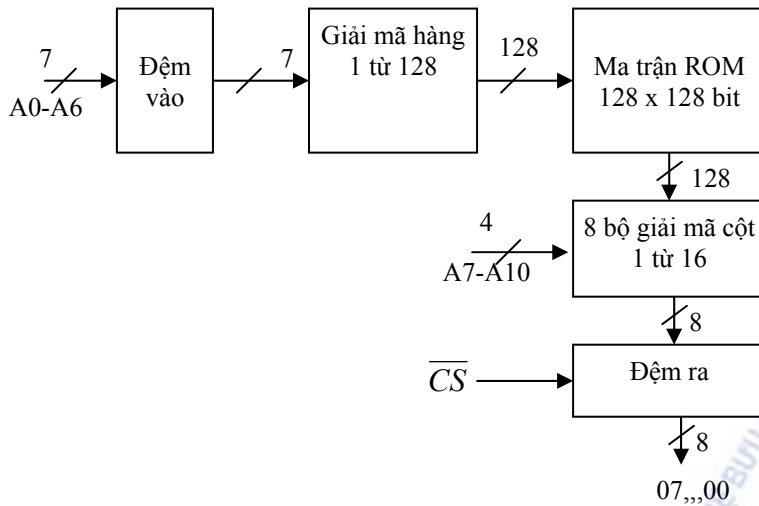
- Một ma trận nhớ gồm các ô nhớ, mỗi ô nhớ ứng với một bit nhớ.
- Mạch logic giải mã địa chỉ ô nhớ.
- Mạch logic cho phép đọc nội dung ô nhớ.
- Mạch logic cho phép viết nội dung ô nhớ.
- Các bộ đệm vào, bộ đệm ra và bộ mở rộng địa chỉ.

Cách tổ chức đơn giản nhất là tổ chức theo từ (word organized) với sự chọn tuyến tính. Một ma trận nhớ như vậy có độ dài của cột bằng số lượng từ  $W$  và độ dài của hàng bằng số lượng bit  $B$  trong một từ. Bộ chọn từ phải giải mã 1 từ  $W$ , nghĩa là giải mã để có một đầu ra duy nhất cho một từ trong bộ nhớ. Phương pháp này có thời gian truy nhập ngắn nhưng cần một bộ giải mã lớn khi tổng số từ lớn, do đó làm tăng giá thành sản phẩm.

Kích thước của phân giải mã địa chỉ sẽ giảm đi khi tổ chức ma trận nhớ và phân logic chọn từ cho phép giải mã hai bước. Ma trận nhớ sử dụng giải mã hai bước ứng với từ vật lý và từ logic. Từ vật lý bao gồm số lượng bit trong một hàng của ma trận. Từ logic bao gồm số lượng bit tương ứng với một từ logic được nhận biết và gửi ra cùng một lúc. Cần hai bộ giải mã: một bộ giải mã hàng để chọn một từ vật lý và một bộ giải mã cột gồm có một vài mạch hợp kênh chọn một từ logic từ một từ vật lý đã chọn. Một từ vật lý được chia thành  $S$  từ logic. Bộ giải mã hàng là bộ giải mã chọn 1 từ  $W$  mà  $B = W/S$  và bộ chọn cột chứa  $B$  bộ hợp kênh một đường từ  $S$ .

Ví dụ sơ đồ ROM dung lượng  $2048 \times 8$  (2048 từ, mỗi từ chứa 8 bit) tổ chức giải mã hai bước như hình 7- 1.

Ma trận nhớ là  $128 \times 128$ , như vậy có  $128 = 2^7$  từ vật lý. Một từ vật lý được chọn bởi 7 đường địa chỉ từ  $A_0$  đến  $A_6$ . Bộ giải mã hàng chọn 1 hàng từ 128 hàng. Một từ vật lý được chia thành  $128/8 = 16$  nhóm 8 bit. Nhóm thứ nhất chứa những bit có trọng số cao nhất của 16 từ logic. Nhóm thứ hai chứa các bit cao tiếp theo của 16 từ logic...Nhóm cuối cùng chứa những bit thấp nhất của 16 từ logic, do đó  $S = 16$ . Như vậy, những bộ giải mã cột gồm 8 bộ hợp kênh một đường từ 16 đường để cung cấp một từ logic ra 8 bit. Những địa chỉ từ  $A_7$  đến  $A_{10}$  điều khiển các bộ giải mã cột. Trường hợp đặc biệt khi số phần tử trong một từ vật lý bằng số bit trong một từ vật lý thì đó là bộ nhớ tổ chức theo bit có nghĩa là mỗi từ logic có độ dài 1 bit.



Hình 7-1. Một ví dụ về giải mã hai bước cho ma trận ROM 128 x 128

Các bộ đệm ra đảm bảo các mức logic mong muốn và cung cấp đủ dòng điện, ngoài ra nó còn có đầu ra collector hở hoặc 3 trạng thái cho phép nối chung đầu ra của một vài chip với nhau. Bộ đệm ra được điều khiển bởi một hay nhiều đầu vào như chọn mạch CS (Chip Select), cho phép mở CE (Chip Enable) hay cho phép mở đầu ba trạng thái OE (Output Enable).

## 7.2. DRAM

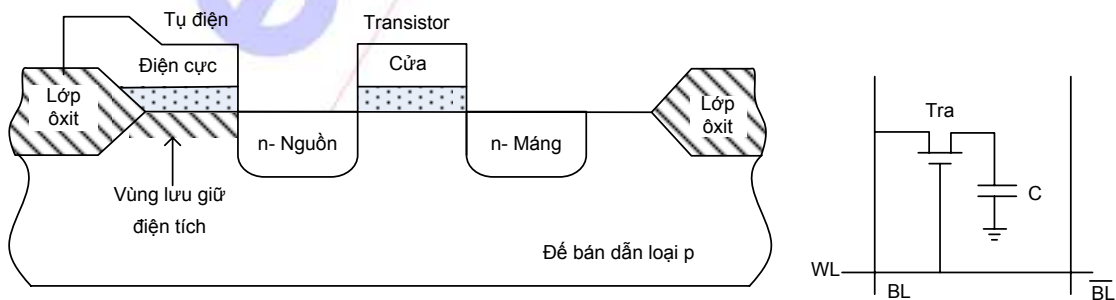
### 7.2.1. Cấu tạo của DRAM

Các ô nhớ được sắp xếp theo hàng và cột trong một ma trận nhớ. Địa chỉ ô nhớ được chia thành hai phần: địa chỉ hàng và cột. Hai địa chỉ này được đọc vào bộ đệm một cách lần lượt. Xử lý kiểu này được gọi là hộp kênh, lý do là để giảm kích thước bộ giải mã, tức là giảm kích thước và giá thành vi mạch. Quá trình dò tìm địa chỉ này được điều khiển bởi các tín hiệu RAS (Row Access Strobe) và CAS (Column Access Strobe).

Nếu  $\overline{\text{RAS}}$  ở mức tích cực thấp thì DRAM nhận được địa chỉ đặt vào nó và sử dụng như địa chỉ hàng.

Nếu  $\overline{\text{CAS}}$  ở mức tích cực thấp thì DRAM nhận được địa chỉ đặt vào nó và sử dụng như địa chỉ cột.

Một ô nhớ của DRAM gồm có một transistor trường MOS có trở lối vào rất lớn và một tụ điện C là linh kiện lưu trữ một bit thông tin tương ứng với hai trạng thái có hoặc không có điện tích trên tụ.

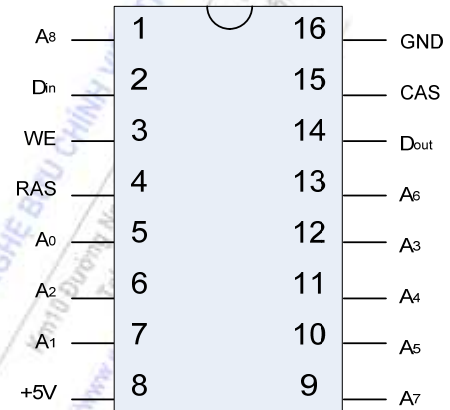


Hình 7-2. Cấu tạo một ô nhớ của DRAM

Transistor hoạt động như một công tắc, cho phép nạp hay phóng điện tích của tụ khi thực hiện phép đọc hay viết. Cực cửa (Gate) của transistor được nối với dây hàng (còn gọi là dây từ - WL-Word Line) và cực máng (Drain) được nối với dây cột (còn được gọi là dây bit BL hoặc  $\overline{BL}$  - Bit Line), cực nguồn (Source) được nối với tụ điện. Điện áp nạp trên tụ tương đối nhỏ, vì thế cần sử dụng khuếch đại nhạy trong mạch nhớ. Do dòng rò của transistor nên ô nhớ cần được nạp lại trước khi điện áp trên tụ thấp hơn một ngưỡng nào đó. Quá trình này được thực hiện nhờ một chu kỳ “làm tươi” (refresh), khi đó điện áp trên tụ được xác định (ở trạng thái 0 hay 1) và mức điện áp logic này được viết lại vào ô nhớ.

Một số loại chip DRAM thường gặp là: TMS 4116: có dung lượng 16k x 1 bit; 41256 có dung lượng 256k x 1 bit. Thời gian truy cập thông tin khoảng 150 nsec, công suất tiêu thụ khoảng 280 mW khi làm việc (khi chờ = 28 mW)

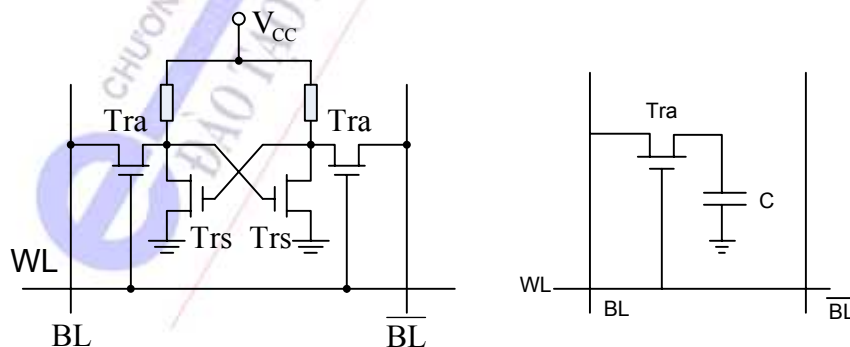
Hình 7-3 là vỏ của IC 41256 dung lượng 256k x 1 bit. Mạch cần 18 bit địa chỉ để mã hoá cho các địa chỉ hàng và cột; nhưng trên vỏ chỉ có 9 đương địa chỉ từ A<sub>0</sub> đến A<sub>8</sub>. Hai chân RAS, CAS hoạt động ở mức cao, dùng để điều khiển 9 bit địa chỉ trên chip tới bộ giải mã địa chỉ hàng hay cột.



Hình 7-3. IC 41256

### 7.3. SRAM

Một ô nhớ của SRAM giữ thông tin bởi trạng thái của mạch trigơ. Thuật ngữ “tĩnh” chỉ ra rằng khi nguồn nuôi chưa bị cắt thì thông tin của ô nhớ vẫn được giữ nguyên. Khác với ô nhớ DRAM, ở đây ô nhớ trigơ cung cấp một tín hiệu số mạch hơn nhiều vì đã có các transistor trong các ô nhớ, chúng có khả năng khuếch đại tín hiệu và do đó có thể cấp trực tiếp cho các đường bit. Trong DRAM, sự khuếch đại tín hiệu trong các bộ khuếch đại cần nhiều thời gian và do đó thời gian truy nhập dài hơn. Khi định địa chỉ trong các trigơ ở SRAM, các transistor bổ sung cho các trigơ, các bộ giải mã địa chỉ...cũng được đòi hỏi như ở DRAM.



Hình 7-4. Cấu tạo một ô nhớ của SRAM và DRAM

Như trong DRAM, cực cửa của transistor được nối với đường từ và cực máng nối với cặp đường bit. Nếu số liệu được đọc từ ô nhớ, khi đó bộ giải mã hàng kích hoạt đường dây từ WL tương ứng. Hai transistor T dẫn và nối trigơ nhớ với cặp dây bit. Như vậy hai lối ra Q và  $\overline{Q}$  được

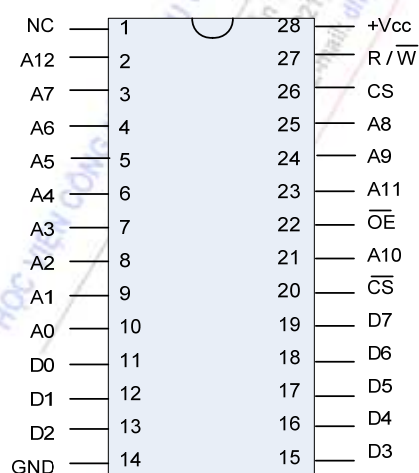
nối với các đường bit và các tín hiệu được truyền tới bộ khuếch đại ở cuối đường dây này. Vì điện thế chênh lệch lớn nên xử lý khuếch đại như vậy sẽ nhanh hơn trong DRAM (cỡ 10 ns hoặc ngắn hơn), do đó chip SRAM cần địa chỉ cột sớm hơn nếu thời gian truy nhập không được giảm. Như vậy SRAM không cần thực hiện phân kênh các địa chỉ hàng và cột. Sau khi số liệu ổn định, bộ giải mã cột chọn cột phù hợp và cho ra tín hiệu số liệu tới bộ đệm số liệu ra và tới mạch ra.

Viết số liệu được thực hiện theo cách ngược lại. Qua bộ đệm vào và bộ giải mã cột, số liệu viết được đặt vào bộ khuếch đại phù hợp. Cùng lúc đó bộ giải mã hàng kích hoạt đường dây từ và làm transistor T dẫn. Triggers đưa số liệu được lưu trữ vào cặp dây bit. Tuy vậy, bộ khuếch đại nhạy hơn các transistor nên nó sẽ cấp cho các đường bit một tín hiệu phù hợp với số liệu viết. Do đó, trigger sẽ chuyển trạng thái phù hợp với số liệu mới hoặc giữ giá trị đã được lưu trữ phụ thuộc vào việc số liệu viết trùng với số liệu đã lưu trữ hay không.

Một số IC DRAM thường gặp là 2148, 2114-2 của hãng Intel. Dung lượng 1k x 4 bit. Thời gian truy cập thông tin khoảng 200 ns, công suất tiêu thụ 525 mW.

IC TMS 4016 dung lượng 2k x 8 bit.

IC HM 6116, họ CMOS, dung lượng 2kbyte, thời gian truy cập là 120 nsec, công suất tiêu thụ khi làm việc là  $P = 180 \text{ mW}$  (khi chờ  $\approx \mu\text{W}$ ). Hình 7-5 giới thiệu IC 6264, dung lượng 8 kbyte, và bảng điều kiện thao tác của nó.



Hình 7-5. Sơ đồ chân của SRAM 6264

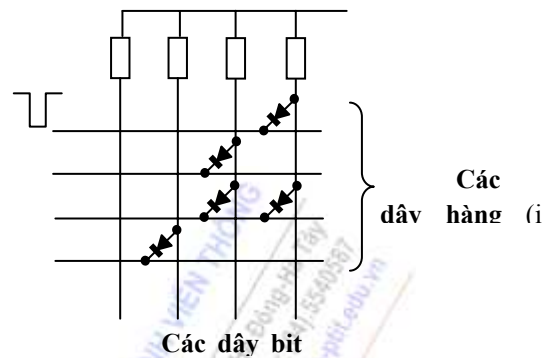
Phương thức hoạt động	$\overline{\text{CS}}$	CS	$\overline{\text{WE}}$	$\overline{\text{OE}}$
Không được chọn	H	X	X	X
Đọc	L	H	H	L
Đọc nhưng không xuất dữ liệu	L	H	H	H
Ghi	L	H	L	L

### 7.3. BỘ NHỚ CỐ ĐỊNH - ROM

Các chip RAM không thích hợp cho các chương trình khởi động do các thông tin trên đó bị mất khi tắt nguồn. Do vậy phải dùng đến ROM, trong đó các số liệu cần lưu trữ được viết một lần theo cách không bay hơi để nhằm giữ được mãi.

#### 7.3.1. MROM

ROM lập trình theo kiểu mặt nạ được gọi là MROM. Nó được chế tạo trên một phiến silic theo một số bước xử lý như quang khắc và khêch tán để tạo ra những tiếp giáp bán dẫn có tính dẫn điện theo một chiều (như diode, transistor trường). Người thiết kế định rõ chương trình muốn ghi vào ROM, thông tin này được sử dụng để điều khiển quá trình làm mặt nạ. Hình 7-6 là một ví dụ đơn giản về sơ đồ MROM dùng diode.



Hình 7-6. MROM diode

Chỗ giao nhau giữa các dây từ (hàng) và các dây bit (cột) tạo nên một phần tử nhớ (ô nhớ). Một diode được đặt tại đó (hình vẽ) sẽ cho phép lưu trữ số liệu “0”. Ngược lại những vị trí không có diode thì sẽ cho phép lưu trữ số liệu “1”. Khi đọc một từ số liệu thứ  $i$  của ROM, bộ giải mã sẽ đặt dây từ đó xuống mức logic thấp, các dây còn lại ở mức cao. Do vậy chỉ những diode nối với dây này được phân cực thuận, do đó nó sẽ dẫn làm cho điện thế lồi ra trên các dây bit tương ứng ở mức logic thấp, các dây bit còn lại sẽ giữ ở mức cao.

Cả hai công nghệ MOS và lưỡng cực được dùng để chế tạo MROM. Thời gian truy nhập của bộ nhớ lưỡng cực khoảng từ 50 – 90 ns, bộ nhớ MOS lâu hơn khoảng 10 lần. Do đó ROM lưỡng cực nhanh hơn và có khả năng kích hoạt tốt hơn trong khi mạch nhớ MOS cùng dung lượng có kích thước nhỏ hơn và tiêu thụ năng lượng ít hơn.

### 7.3.2. PROM

PROM cũng gồm có các diode như ở MROM nhưng chúng có mặt đầy đủ tạo các vị trí giao nhau giữa dây từ và dây bit. Mỗi diode được nối với một cầu chì. Bình thường khi chưa lập trình, các cầu chì còn nguyên vẹn, nội dung của PROM sẽ toàn là 0. Khi định vị đến một bit bằng cách đặt một xung điện ở lồi ra tương ứng, cầu chì sẽ bị đứt và bit này sẽ bằng 1. Bằng cách đó ta có thể lập trình toàn bộ các bit trong PROM. Như vậy, việc lập trình đó có thể được thực hiện bởi người sử dụng chỉ một lần duy nhất, không thể sửa đổi được.

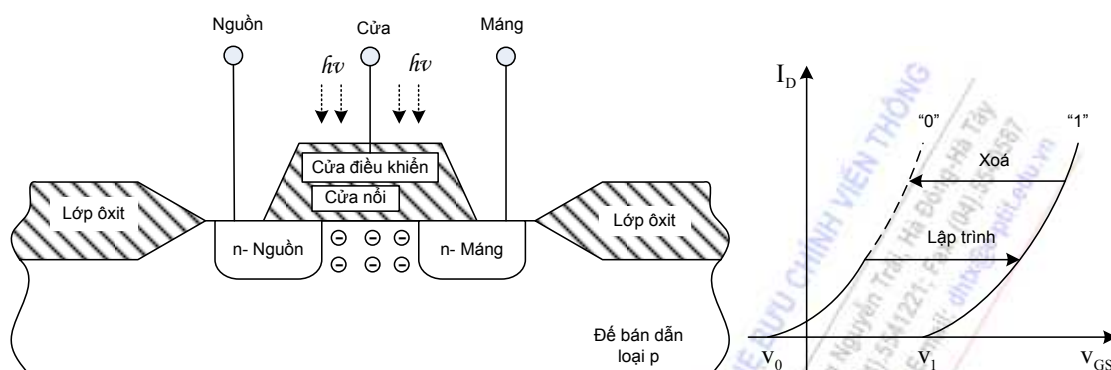
## 7.4. BỘ NHỚ BÁN CỐ ĐỊNH

### 7.4.1. EPROM (Erasable PROM)

Số liệu vào có thể được viết vào bằng xung điện nhưng được lưu giữ theo kiểu không bay hơi. Đó là loại ROM có thể lập trình được và xóa được. Hình 7- 7 chỉ ra cấu trúc của một transistor dùng để làm một ô nhớ gọi là FAMOST (Floating gate avalanche injection MOS transistor).

Trong ô nhớ dùng transistor này, cực cửa được nối với đường từ, cực máng được nối với đường bit và cực nguồn được nối với nguồn chuẩn được coi là nguồn cho mức logic 1. Khác với transistor MOS bình thường, transistor loại này còn có thêm một cửa gọi là cửa nổi (floating gate); đó là một vùng vật liệu được thêm vào vào giữa lớp cách điện cao như ở hình 7-7. Nếu cửa nổi không có điện tích thì nó không ảnh hưởng gì đến cực cửa điều khiển và transistor hoạt động như bình thường. Tức là khi dây từ được kích hoạt (cực cửa có điện thế dương) thì transistor dẫn, cực máng và nguồn được nối với nhau qua kênh dẫn và dây bit có mức logic 1. Nếu cửa nổi có các điện tử trong đó với điện tích âm thì chúng sẽ ngăn trường điều khiển của cửa cửa và dù dây

từ được kích hoạt thì cũng không thể phát ra trường đều mạnh với cực cửa điều khiển để làm thông transistor. Lúc này đường bit không được nối với nguồn chuẩn và ô nhớ coi như được giữ giá trị 0.



Hình 7-7. Cấu trúc của một EPROM

Việc nạp các điện tử vào vùng cửa nổi, tức là tạo ra các ô nhớ mang giá trị 0 được thực hiện bởi xung điện có độ dài cỡ 50 ms và độ lớn + 20 V đặt giữa cực cửa và cực máng. Lúc đó những điện tích mang năng lượng lớn sẽ đi qua lớp cách điện giữa đế và cửa nổi. Chúng tích tụ trong vùng cửa nổi và được giữ ở đây sau khi xung lập trình tắt. Đó là do cửa nổi được cách điện cao với xung quanh và các điện tử không còn đủ năng lượng sau khi lạnh đi, để có thể vượt ra ngoài lớp cách điện đó nữa. Chúng sẽ được giữ ở đây trong một thời gian rất dài (ít nhất là 10 năm).

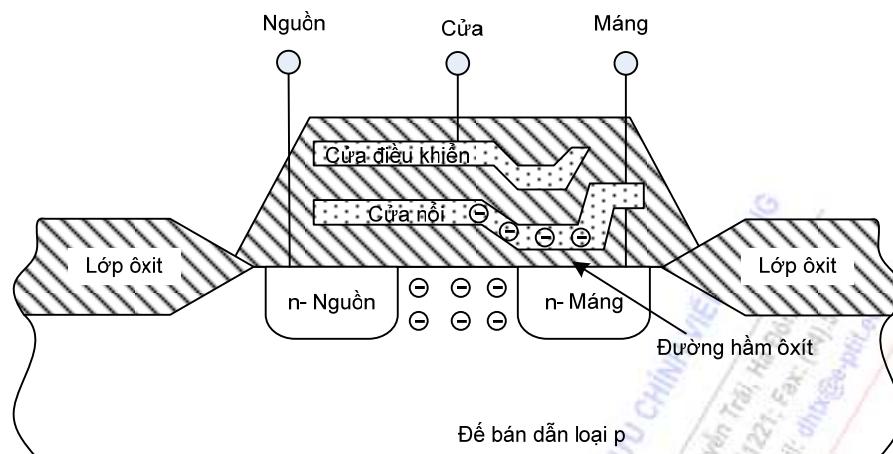
Để xoá các thông tin, tức là làm mất các điện tích điện tử trong vùng cửa nổi, phải chiếu ánh sáng tử ngoại UV vào chip nhớ. Lúc này, những điện tử hấp thụ đượ năng lượng và sẽ nhảy lên các mức năng lượng cao và rời khỏi cửa nổi giống như cách mà chúng đã thâm nhập vào. Trong chip EPROM có một cửa sổ làm bằng thủy tinh thạch anh chỉ để cho ánh sáng tử ngoại đi qua khi cần xoá số liệu trong bộ nhớ.

#### 7.4.2. EEPROM (Electrically Erasable PROM)

Cửa sổ thạch anh có giá thành khá đắt và không tiện lợi nên những năm gần đây xuất hiện các chip PROM có thể xoá số liệu bằng phương pháp điện. Cấu trúc của ô nhớ giống như hình 7-8.

Việc nạp các điện tử cho cửa nổi được thực hiện như cách ở EPROM. Bằng một xung điện tương đối dài, các điện tích mang năng lượng cao được phát ra trong đế sẽ thâm qua lớp cửa ôxit và tích tụ trong cửa nổi. Để xoá EEPROM, một lớp kênh màng mỏng ôxit giữa vùng cửa nổi trải xuống dưới đế và cực máng giữ vai trò quan trọng. Các lớp cách điện không thể là lý tưởng được, các điện tích có thể thâm qua lớp phân cách với một xác suất thấp. Xác suất này tăng lên khi bề dày của lớp giảm đi và điện thế giữa hai điện cực ở hai mặt lớp cách điện tăng lên. Muốn phóng các điện tích trong vùng cửa nổi một điện thế (-20 V) được đặt vào cực cửa điều khiển và cực máng. Lúc này các điện tử âm trong cửa nổi được chảy về cực máng qua kênh màng mỏng ôxit và số liệu lưu giữ được xoá đi. Điều lưu ý là phải làm sao cho dòng điện tích này chảy không quá lâu vì nếu không vùng cửa nổi này lại trở nên tích điện dương làm cho hoạt động của transistor không được trạng thái bình thường (mức nhớ 1).





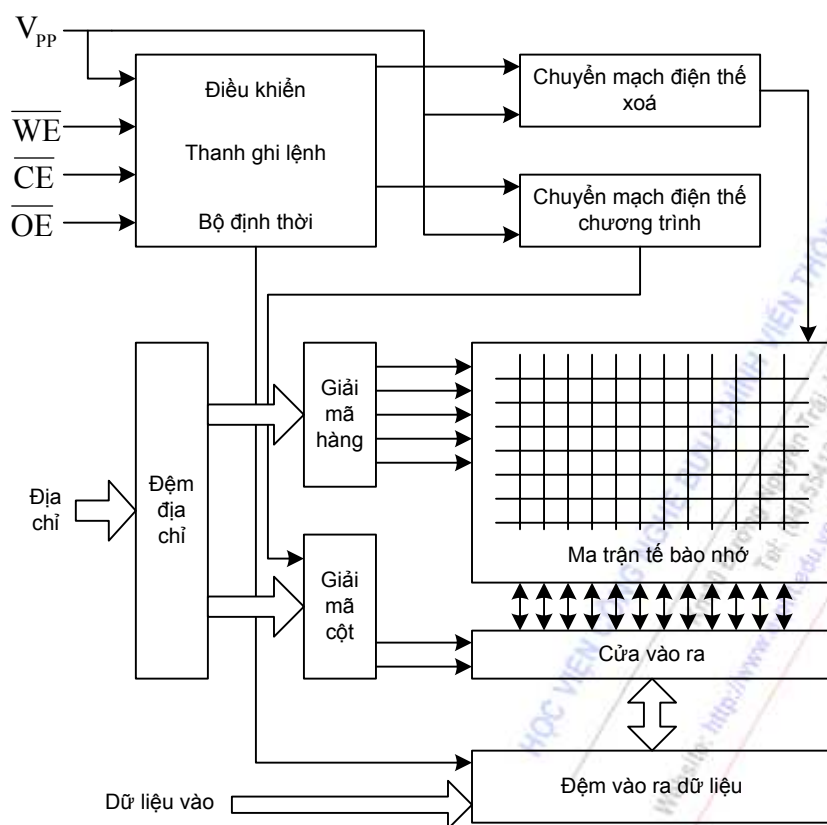
**Hình 7-8. Cấu trúc của EEPROM**

Các chip ROM hiện nay có thời gian truy nhập từ 120 ns đến 150 ns dài hơn nhiều thời gian đó trong các chip nhớ RAM.

### 7.4.3. Đĩa cứng silicon- Bộ nhớ FLASH

Trong những năm gần đây, một loại bộ nhớ không bay hơi mới đã xuất hiện trên thị trường, thường được sử dụng thay thế cho các ổ đĩa mềm và cứng trong những máy tính. Đó là bộ nhớ flash. Cấu trúc của chúng cơ bản như EEPROM, chỉ có lớp kênh ôxít ở các ô nhớ mỏng hơn. Do vậy chỉ cần điện thế cỡ 12 V là có thể cho phép thực hiện 10 000 chu trình xoá và lập trình. Bộ nhớ flash có thể hoạt động gần mềm dẻo như DRAM và SRAM nhưng lại không bị mất số liệu khi bị cắt điện. Hình 7- 9 chỉ ra sơ đồ khối của nó.

Phần chính là mạng nhớ bao gồm các ô nhớ FAMOST như được mô tả ở mục trên. Giống như SRAM, bộ nhớ flash không dồn phân kênh địa chỉ. Các bộ giải mã hàng và cột chọn một đường từ và một hoặc nhiều cặp đường bit. Số liệu đọc được đưa ra ngoài bộ đệm số liệu I/O hoặc được viết vào ô nhớ đã được định địa chỉ bởi bộ đệm này qua cổng I/O. Xử lý đọc được thực hiện với điện thế MOS thông thường là 5V. Để lập trình một ô nhớ, đơn vị điều khiển flash đặt một xung điện thế ngắn cỡ 10  $\mu$ s và 12 V gây nên một sự chọc thủng thác lũ vào transistor nhớ để nạp vào cửa nổi. Một chip nhớ flash 1 Mb có thể được lập trình trong khoảng 2 sec, nhưng khác với EEPROM việc xoá được thực hiện từng chip một. Thời gian xoá cho toàn bộ bộ nhớ flash khoảng 1 sec. Xử lý đọc, lập trình và xoá được điều khiển bởi các lệnh có độ dài 2 byte được bộ xử lý viết vào các thanh ghi lệnh của mạch điều khiển flash.



**Hình 7-9. Sơ đồ bộ nhớ FLASH**

Mục đích sử dụng chính của bộ nhớ flash là để thay thế cho các ổ đĩa mềm và ổ đĩa cứng dung lượng nhỏ. Do nó là mạch tích hợp nên có ưu điểm là kích thước nhỏ và tiêu thụ năng lượng thấp, không bị ảnh hưởng của va đập. Các đĩa cứng chất rắn dựa trên cơ sở các bộ nhớ flash có lợi thế về công suất tiêu thụ cũng như giá thành có dung lượng tới vài Mbyte. Các card nhớ loại này có ưu điểm là không gặp phải vấn đề mất thông tin như trường hợp RAM CMOS khi pin Ni-Cd bị hỏng. Thời gian lưu trữ thông tin trong bộ nhớ flash ít nhất là 10 năm, thông thường là 100 năm, với khoảng thời gian này thì các đĩa mềm và cứng đã bị hỏng rồi.

Nhược điểm của bộ nhớ flash là chỉ có thể xóa theo kiểu lần lượt từng chip hoặc lần lượt từng trang.

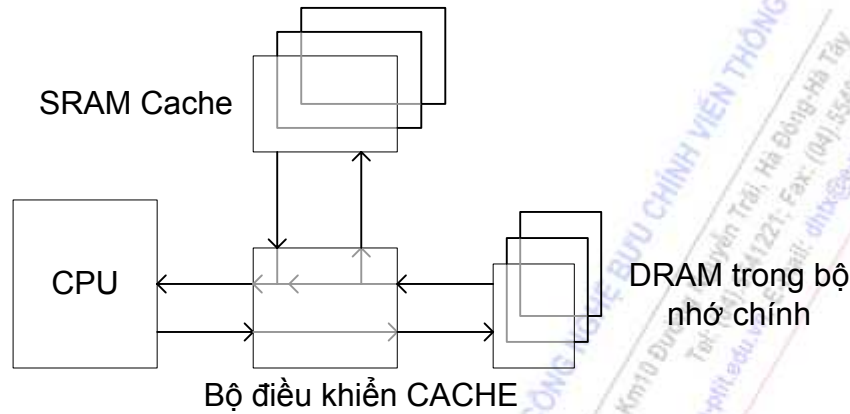
### 7.4.3. Bộ nhớ CACHE

Với các máy tính có tốc độ nhanh (trên 33MHz), cần phải xen các trạng thái đợi khi truy xuất dữ liệu tới các DRAM rẻ tiền nhưng có thời gian thâm nhập chậm (60-120ns). Điều này làm giảm hiệu suất của máy. Có thể giải quyết bằng cách dùng các SRAM có thời gian thâm nhập ngắn hơn (20-25 ns, thậm chí 12 ns) nhưng giá thành lại rất đắt. Bộ nhớ Cache kết hợp được các lợi điểm nhanh của SRAM và rẻ của DRAM. Giữa CPU và bộ nhớ chính bằng DRAM, người ta xen vào một bộ nhớ SRAM nhanh có dung lượng nhỏ bằng 1/10 hoặc 1/100 lần bộ nhớ chính gọi là cache; dưới sự điều khiển của mạch điều khiển cache, bộ nhớ này sẽ lưu trữ tạm thời các số liệu thường được gọi và cung cấp nó cho CPU trong thời gian ngắn.

Cache chứa các thông tin mới vừa được CPU sử dụng gần đây nhất. Khi CPU đọc số liệu nó sẽ đưa ra một địa chỉ tới bộ điều khiển cache. Sau đó một trong hai quá trình sau sẽ xảy ra:

- Cache hit: nếu địa chỉ đó đã có sẵn trong RAM cache.
- Cache miss: ngược lại, nếu địa chỉ đó không có sẵn trong RAM cache.

Như vậy, cache hit tỷ lệ với truy xuất thông tin có sẵn trong bộ nhớ cache SRAM, còn cache miss lại tỷ lệ với truy xuất thông tin có trong bộ nhớ chính là các DRAM.



Hình 7-10. Nguyên lý của Cache

## 7.5. MỞ RỘNG DUNG LƯỢNG BỘ NHỚ

Các vi mạch nhớ bán dẫn chỉ có dung lượng xác định. Muốn có bộ nhớ có dung lượng lớn hơn, ta tìm cách ghép nhiều vi mạch nhớ nhằm một trong ba mục đích sau:

- Tăng độ dài nhớ, nhưng không làm tăng số lượng từ nhớ.
- Tăng số lượng từ nhớ nhưng không làm tăng độ dài từ nhớ.
- Tăng cả số lượng và độ dài từ nhớ.

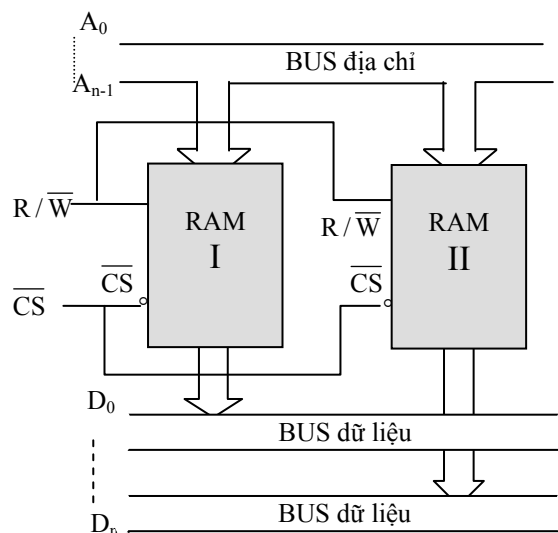
### 7.5.1 Mở rộng độ dài từ

Trên một chip nhớ, có thể có được 1 đến một số hữu hạn lối ra, thường là 4 hoặc 8 bit. Muốn có độ dài từ lớn hơn, chẳng hạn từ 4 lên 8 hoặc 16 bit, ta tiến hành ghép nhiều chip nhớ như chỉ ở hình 7-10 đối với RAM. Đối với ROM cách làm cũng tương tự, chỉ khác trong trường hợp này, có thể không có lối vào R/  $\bar{W}$ .

### 7.5.2 Mở rộng dung lượng

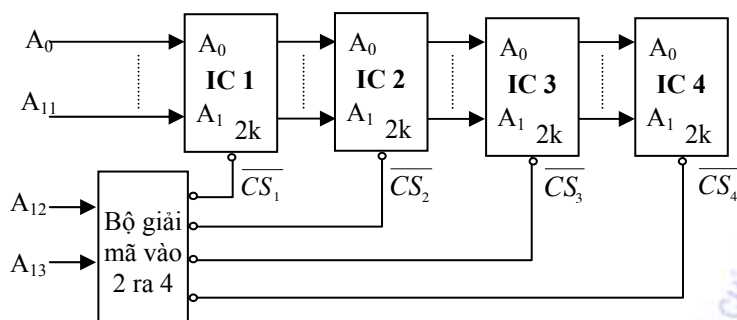
Muốn mở rộng dung lượng, ta cũng ghép nhiều chip lại với nhau. Như đã biết, dung lượng có liên quan đến số lối vào địa chỉ ( $C = 2^N \times$  độ dài từ, với N là số lối vào địa chỉ). Cứ tăng 1 chip thì cần có thêm một lối vào địa chỉ.

Khác với trường hợp mở rộng độ dài từ, khi mở rộng dung lượng các lối vào/ra dữ liệu D và



Hình 7-10. Sơ đồ mở rộng độ dài từ.

$R/\overline{W}$  được nối song song. Một phần dung lượng được trữ vào mỗi chip. Sự phân chia này dựa trên cơ sở tổ hợp địa chỉ vào và lối vào điều khiển. Hình 7-11 là một sơ đồ ví dụ.



Hình 7-11. Phương pháp mở rộng dung lượng.

Để thực hiện phép mở rộng ta phải sử dụng một số lối vào địa chỉ dành riêng cho bộ giải mã (thường là các địa chỉ có trọng số cao). Ở sơ đồ trên ta chọn 2 địa chỉ  $A_{12}$  và  $A_{13}$  để giải mã. Do đó ta có thể nhận được 4 giá trị ra tương ứng. Các giá trị này tác động lên các lối vào  $\overline{CS}$  để mở tuần tự các IC nhớ. Các IC nhớ này có thể làm ROM hoặc RAM hoặc cả hai là tùy chọn. Tuần tự mở các IC theo  $A_{12}, A_{13}$  như chỉ ra ở bảng hoạt động sau.

$A_{13}$	$A_{12}$	$\overline{CS}$	IC mở	Khoảng địa chỉ
0	0	$\overline{CS}_1$	IC I	$0000_{16} - 0FFF_{16}$
0	1	$\overline{CS}_2$	IC II	$1000_{16} - 1FFF_{16}$
1	0	$\overline{CS}_3$	IC III	$2000_{16} - 2FFF_{16}$
1	1	$\overline{CS}_4$	IC IV	$3000_{16} - 3FFF_{16}$

Kỹ thuật này thường được ứng dụng trong các hệ thống vi xử lý, phổ biến nhất là các máy vi tính. Phương pháp này không chỉ cho phép mở rộng dung lượng, mà còn tạo ra sự phân vùng nhớ. Chỉ cần ba địa chỉ giải mã đã có thể tạo ra được 8 vùng nhớ với dung lượng tùy thuộc các chip thành phần.

## TÓM TẮT

Trong chương này chúng ta trình bày nguyên lý cấu tạo, các tính năng cơ bản của các loại bộ nhớ bán dẫn: ROM, PROM, EPROM, EEPROM, SRAM, DRAM, FLASH, CACHE.

Các chip RAM không thích hợp cho các chương trình khởi động do các thông tin trên đó bị mất khi tắt nguồn. Do vậy phải dùng đến ROM, trong đó các số liệu cần lưu trữ được viết một lần theo cách không bay hơi để nhằm giữ được mãi.

Trong những năm gần đây, một loại bộ nhớ không bay hơi mới đã xuất hiện trên thị trường, thường được sử dụng thay thế cho các ổ đĩa mềm và cứng trong những máy tính. Đó là bộ nhớ flash. Cấu trúc của chúng cơ bản như EEPROM, chỉ có lớp kênh ôxít ở các ô nhớ mỏng hơn.

Với các máy tính có tốc độ nhanh (trên 33MHz), cần phải xen các trạng thái đợi khi truy xuất dữ liệu tới các DRAM rẻ tiền nhưng có thời gian thâm nhập chậm (60-120ns). Điều này làm giảm hiệu suất của máy. Có thể giải quyết bằng cách dùng các SRAM có thời gian thâm nhập

ngắn hơn (20-25 ns, thậm chí 12 ns) nhưng giá thành lại rất đắt. Bộ nhớ Cache kết hợp được các lợi điểm nhanh của SRAM và rẻ của DRAM.

Trong chương này còn giới thiệu cách mở rộng dung lượng và độ dài từ của bộ nhớ bán dẫn.

## CÂU HỎI ÔN TẬP

1. Bộ nhớ ROM là bộ nhớ:
  - a. Chỉ có thể đọc.
  - b. Chỉ có thể viết.
  - c. Có thể vừa đọc vừa viết.
  - d. Không có phương án nào đúng.
2. Bộ nhớ RAM là bộ nhớ:
  - a. Chỉ có thể đọc.
  - b. Chỉ có thể viết.
  - c. Có thể vừa đọc vừa viết.
  - d. Không có phương án nào đúng.
3. Linh kiện lưu giữ bit thông tin của DRAM là:
  - a. Transistor.
  - b. Trơ.
  - c. Tụ điện.
  - d. Diode.
4. Linh kiện lưu giữ bit thông tin của SRAM là:
  - a. Transistor.
  - b. Trơ.
  - c. Tụ điện.
  - d. Diode.
5. MROM được chế tạo bởi công nghệ :
  - a. Lưỡng cực.
  - b. MOS.
  - c. Lưỡng cực và MOS.
  - d. Không có phương án nào đúng.
6. PROM là loại ROM có thể:
  - a. Chỉ lập trình được một lần.
  - b. Lập trình được nhiều lần.

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG  
Km10 Đường Nguyễn Trãi, Hà Đông-Hà Tây  
Tel: (04) 5541221; Fax: (04) 5540587  
Website: <http://www.c-ptit.edu.vn>; E-mail: [dhkc@ptit.edu.vn](mailto:dhkc@ptit.edu.vn)

CHƯƠNG TRÌNH PTIT  
ĐÀO TẠO ĐẠI HỌC TỪ XA

- c. Lập trình được và xoá được.
  - d. Không có phương án nào đúng.
7. Linh kiện lưu giữ bit thông tin của EPROM là:
- a. Transistor lưỡng cực.
  - b. Transistor trường.
  - c. Tụ điện.
  - d. Diode.
8. Trong EPROM, việc nạp các điện tích vào vùng cửa nổi có nghĩa là:
- a. Tạo các ô nhớ mang giá trị 0.
  - b. Tạo các ô nhớ mang giá trị 1.
  - c. Tạo các ô nhớ mang giá trị 0 và 1.
  - d. Không có phương án nào đúng.
9. EEPROM là loại ROM có thể:
- a. Chỉ lập trình được một lần.
  - b. Lập trình được và xoá được một lần.
  - c. Lập trình được và xoá được nhiều lần.
  - d. Không có phương án nào đúng.
10. Muốn xoá dữ liệu trong EEPROM thì cần:
- a. Chiếu tia tử ngoại vào.
  - b. Cần đặt vào cực cửa điều khiển và cực máng một điện thế có giá trị 20V.
  - c. Cần đặt vào cực cửa điều khiển và cực máng một điện thế có giá trị - 20V.
  - d. Cả phương án trên đều đúng.
11. Bộ nhớ FLASH là loại bộ nhớ:
- a. Mất dữ liệu khi mất nguồn nuôi.
  - b. Không mất dữ liệu khi mất nguồn nuôi.
  - c. Bị mất dần dữ liệu ngay cả khi có nguồn nuôi.
  - d. Không có phương án nào đúng.
12. Bộ nhớ FLASH là loại bộ nhớ có thể thay thế cho:
- a. Ổ đĩa mềm.
  - b. Ổ đĩa cứng.
  - c. Ổ mềm và ổ cứng có dung lượng nhỏ.
  - d. Không có phương án nào đúng.

## CHƯƠNG 8: LOGIC LẬP TRÌNH (PLD)

### GIỚI THIỆU

Các mạch kỹ thuật số tổ hợp và tuần tự đã được đề cập ở các chương trước. Các IC số rất đa dạng từ thực hiện các phép tính kỹ thuật số căn bản đến các chức năng phức tạp khác như: bộ hợp kênh, phân kênh, bộ cộng, so sánh, bộ mã hoá, giải mã, bộ đếm... Chúng là các IC số có chức năng cố định, tức là mỗi IC thực hiện một chức năng chuyên biệt. Những linh kiện này được sản xuất một số lượng lớn để đáp ứng nhu cầu ứng dụng phong phú.

Để thiết kế một mạch, nhà thiết kế có thể chọn từ các IC có sẵn phù hợp nhất cho mạch điện. Phần thiết kế này có thể được chỉnh sửa để đáp ứng các yêu cầu chuyên biệt của những linh kiện này.

Ưu điểm của phương pháp này là:

1. Chi phí phát triển thấp.
2. Vận hành nhanh xung quanh bản thiết kế.
3. Tương đối dễ thử nghiệm các mạch

Nhược điểm:

1. Các yêu cầu về kích thước trong bảng mạch lớn.
2. Yêu cầu về điện lớn.
3. Thiếu tính bảo mật. (Các bảng mạch có thể bị sao chép).
4. Các yêu cầu về chi phí bổ sung, khoảng trống, điện... cần thiết để chỉnh sửa bản thiết kế hoặc trình bày các tính năng khác.

Để khắc phục những nhược điểm của thiết kế bằng cách sử dụng các IC chức năng cố định, các mạch tích hợp chuyên biệt ứng dụng (ASIC-Application Specific IC) đã được phát triển. Các ASIC đã được thiết kế để đáp ứng các yêu cầu chuyên biệt của một mạch và được giới thiệu bởi một nhà sản xuất IC. Các thiết kế này quá phức tạp không thể thực hiện bằng cách sử dụng các IC chức năng cố định được.

Ưu điểm của phương pháp này là:

1. Giảm thiểu được kích thước thông qua việc sử dụng mức tích hợp cao.
2. Giảm thiểu được yêu cầu về điện.
3. Nếu được sản xuất theo một quy mô lớn thì chi phí giảm đáng kể.
4. Việc thiết kế được thực thi dưới dạng này thì hoàn toàn không thể sao chép được.

Nhược điểm:

1. Chi phí phát triển ban đầu có thể cực kỳ lớn.

2. Các phương pháp thử nghiệm phải được phát triển và điều này làm gia tăng chi phí và công sức.

Có một phương pháp khác có các ưu điểm của hai phương pháp trên là sử dụng các thiết bị logic có thể lập trình được (PLD). Một thiết bị logic có thể lập trình là một IC mà người dùng có thể cấu hình để chúng có khả năng thực thi các chức năng logic như mong muốn. Đây là một chip LSI có chứa một cấu trúc “binh thường” và cho phép nhà thiết kế tạo tùy biến cho nó để dùng cho bất kỳ ứng dụng đặc biệt nào, tức là nó có thể được người dùng lập trình để thực hiện một chức năng cần thiết cho ứng dụng của họ.

Các PLD có các ưu điểm sau:

1. Chu kỳ thiết kế ngắn.
2. Chi phí phát triển thấp.
3. Giảm thiểu được yêu cầu khoảng trống trên bảng mạch.
4. Giảm thiểu được yêu cầu về điện.
5. Bảo đảm tính bảo mật của thiết kế.
6. Mạch được kết chặt lại.
7. Tốc độ đảo mạch nhanh hơn.
8. Mật độ tích hợp cao.
9. Chi phí sản xuất số lượng lớn thấp.

PLD cũng cho phép nhà thiết kế có nhiều phương tiện linh động hơn để thí nghiệm với các bản thiết kế bởi vì chúng có thể được lập trình lại trong vài giây.

Với nhiều ưu điểm như vậy nên hiện nay có một số lượng lớn các PLD được các nhà sản xuất IC tạo ra với nhiều tính năng đa dạng và nhiều tùy chọn có sẵn để nhà thiết kế mạch có thể sử dụng một cách phổ biến. Cấu trúc và các tính năng đa dạng khác của các PLD như ROM, các mảng logic lập trình (PLA). Logic mảng có thể lập trình (PAL), thiết bị logic có thể lập trình đơn giản (SPLD), và các mảng cổng có thể lập trình trường (FPGA) sẽ được đề cập ở đây. Công dụng của những thiết bị này yêu cầu phải có thay đổi thiết kế truyền thống, mặc dầu các khái niệm cơ bản vẫn được giữ lại không đổi.

## NỘI DUNG

### 8.1. GIỚI THIỆU CHUNG VỀ LOGIC KHẢ TRÌNH (PLD)

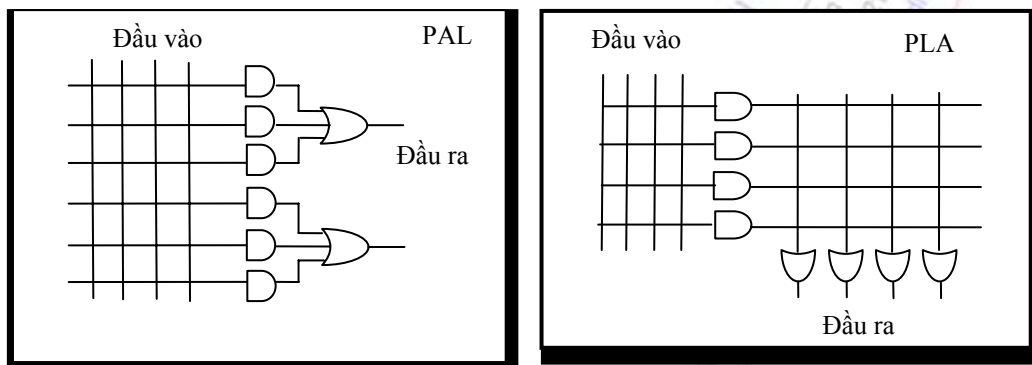
Vì mạch lập trình, viết tắt là PLD (Programmable Logic Device), là loại cấu kiện điện tử có nhiều ưu điểm và hiện nay đang được phát triển rất mạnh. Về nguyên lý, chúng có cấu tạo rất giống với PROM. Việc lập trình cho PLD có thể được thực hiện bằng các công nghệ khác nhau, dựa trên cơ sở bề cầu chì hoặc chuyển mạch. Tuy nhiên, ứng dụng của PLD lại rất khác với PROM. Một PLD, được tạo thành bằng một số cổng AND, OR, XOR hoặc cả các trigơ, có thể thực hiện nhiều hàm Boole khác nhau.



## 8.2 SPLD

SPLD - cấu kiện logic khả trình đơn giản. Đây là loại cấu kiện số có nhiều ưu điểm và cũng đã được phát triển rất mạnh. Về nguyên lý, chúng có cấu tạo rất giống với PROM. Việc lập trình cho SPLD có thể được thực hiện bằng các công nghệ khác nhau, dựa trên cơ sở thực hiện các kết nối bằng cách sử dụng cầu chì hoặc chuyên mạch. Một SPLD, được tạo thành bằng một số mảng cổng AND, OR, XOR hoặc cả các trigger, có thể thực hiện nhiều hàm Boole khác nhau.

Các SPLD đều có cấu tạo dựa trên một trong hai dạng cấu trúc chính: mảng logic khả trình PLA (Programmable Logic Array) và logic mảng khả trình PAL (Programmable Array Logic).



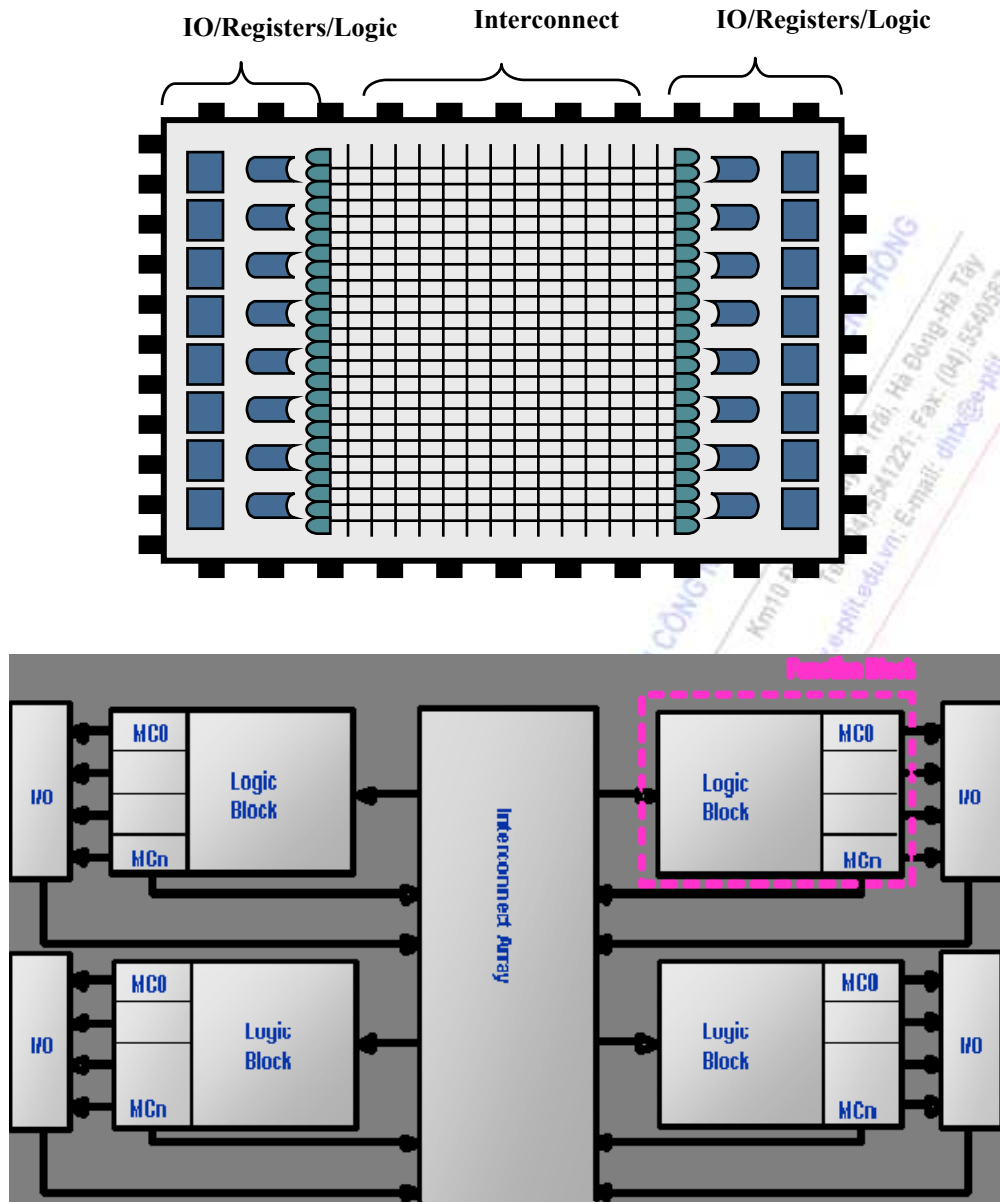
Hình 8.1 - So sánh giữa PAL và PLA

Thành phần cơ bản của PLA là một mảng AND và một mảng OR lập trình được. Mỗi mảng AND, OR gồm các hàng và các cột liên kết với nhau. Tại mỗi điểm giao giữa hàng và cột, có một cầu chì. Khi cầu chì đóng, tại điểm đó có kết nối giữa hàng và cột, khi cầu chì ngắt, tại đó không có kết nối. Việc đóng ngắt cầu chì được thực hiện bằng phần mềm (do lập trình viên hoặc sử dụng công cụ In- System Programming (ISP) – lập trình trên hệ thống).

Cấu trúc PLA tạo ra sự tổ hợp tùy ý giữa các cổng AND và OR, cho mật độ logic cao nhưng tốc độ chậm, số lượng cầu chì lớn. Vì vậy, sau này người ta đã đưa ra một kiểu kiến trúc khác là logic mảng khả trình PAL (Programmable Array Logic).

Công nghệ PLD xuất hiện từ rất sớm với các công ty như Xilinx – sản xuất vi mạch CMOS công suất cực thấp dựa trên công nghệ Flash. PLD dựa trên công nghệ Flash cho phép lập trình và xoá vi mạch nhiều lần bằng điện, nhờ đó tiết kiệm được thời gian so với xoá vi mạch bằng tia cực tím.

## 8.3. CPLD (Complex PLD)



Hình vẽ 8.2 - Kiến trúc của CPLD

Thiết bị logic khả trình phức hợp (CPLD) có mật độ logic cao hơn so với các PLD đơn giản như đã xét ở trên (PLA và PAL). CPLD bao gồm nhiều mạch logic, mỗi mạch có thể coi là một SPLD. Trong một mạch đơn chỉ thực hiện các chức năng logic đơn giản. Các chức năng logic phức tạp hơn cần số lượng khối nhiều hơn, sử dụng ma trận liên kết chung giữa các khối để tạo kết nối. CPLD thường dùng để điều khiển ghép công phức hợp ở tốc độ rất cao (5ns, tương đương với 200 MHz). Kiến trúc cơ bản của CPLD được minh họa trong hình vẽ 8.2.

CPLD có cấu trúc đồng nhất gồm nhiều khối chức năng "**Function Block**" được kết nối với nhau thông qua một ma trận kết nối trung tâm "**Interconnect Array**". Mỗi khối **function block** gồm có một khối logic - gồm các hạng tích AND và OR sắp xếp giống PLA hoặc PAL, cho phép thực hiện các hàm logic tổ hợp- và nhiều khối MC (Macrocell) có chứa tài nguyên là các Trigrơ cho phép xây dựng các thanh ghi và mạch tuần tự. Phần lõi bên trong của CPLD được nối ra bên ngoài thông qua các khối vào ra I/O cho phép thiết lập chức năng cho các chân của IC có chức

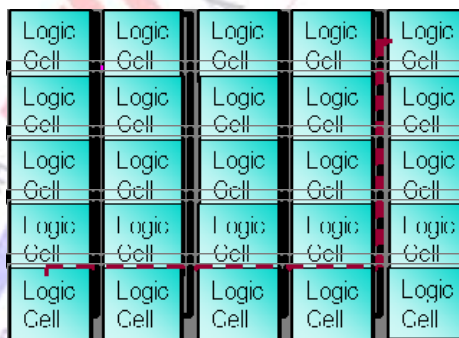
năng vào hoặc ra hoặc vừa là chân vào vừa là chân ra, ngoài ra còn có thể thiết lập các chân I/O này làm việc ở các mức logic khác nhau, có điện trở pull-up hoặc pull-down ...

Với cấu trúc đồng nhất, giá thành rẻ, tính năng khá mạnh, để sử dụng CPLD đã và đang được sử dụng rất rộng rãi trong thực tế, giúp cho nhà sản xuất phát triển nhanh sản phẩm của mình với giá thành rẻ. Đặc biệt hiện nay các hãng đã phát triển các họ CPLD với tính năng rất mạnh, công suất tiêu thụ thấp, chúng đang được sử dụng rất nhiều để phát triển các sản phẩm điện tử, viễn thông, công nghệ thông tin, nhất là trong các thiết bị cầm tay, di động...

Trong thực tế rất có nhiều loại CPLD khác nhau, của các hãng khác nhau, và đã được phát triển với nhiều chủng loại, thế hệ CPLD khác nhau. Cấu tạo, dung lượng, tính năng, đặc điểm, ứng dụng... của mỗi loại CPLD cũng rất khác nhau. Trong giáo trình này không đi sâu trình bày cấu tạo cụ thể của các họ CPLD, mà chỉ trình bày kiến trúc chung đơn giản nhất của CPLD. Khi sử dụng cụ thể loại CPLD nào, người học nên tham khảo các tài liệu khác, nhất là tham khảo các tài liệu kỹ thuật được cung cấp kèm theo cấu kiện do các hãng đưa ra. Các hãng điện tử nổi tiếng trên thế giới đang sở hữu, phát triển, cung cấp các loại cấu kiện CPLD là Xilinx, Altera...

#### 8.4. FPGA

FPGA (Field Programmable Gate Array - Ma trận cổng lập trình được theo trường): có cấu trúc và hoạt động phức tạp hơn CPLD. Nó có thể thực hiện những chức năng phức tạp ưu việt hơn CPLD. Năm 1985, công ty Xilinx đưa ra ý tưởng hoàn toàn mới, đó là kết hợp thời gian hoàn thành sản phẩm và khả năng điều khiển được của PLD với mật độ và ưu thế về chi phí của GateArray. Từ đó, FPGA ra đời. Hiện nay, Xilinx vẫn là nhà sản xuất chip FPGA số một trên thế giới.

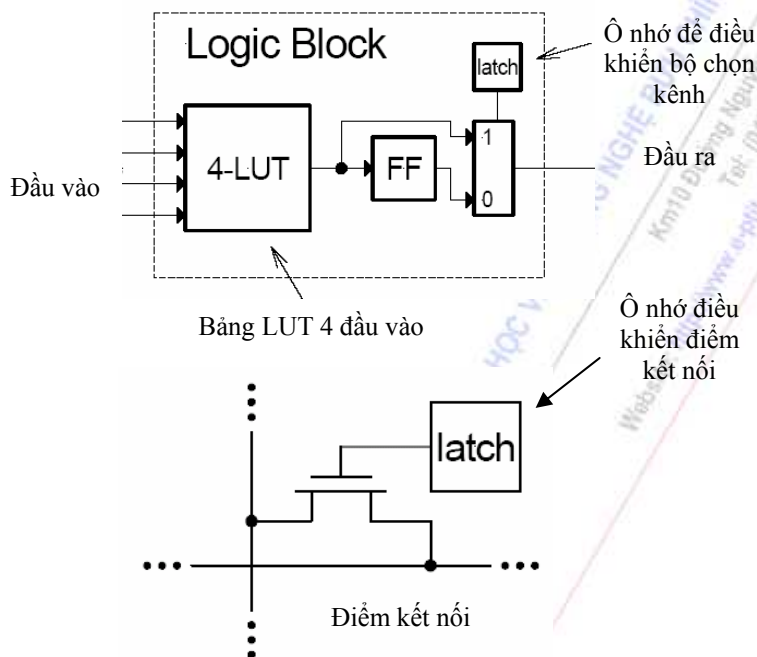


**Hình 8-3. Kiến trúc chung của FPGA**

Cấu trúc FPGA đơn giản gồm các tế bào logic (Logic Cell), các khối cách đều nhau, liên kết nhờ các đường kết nối có thể thay đổi được theo yêu cầu của người thiết kế. Nghĩa là người thiết kế có quyền thiết kế, lập trình và thay đổi mạch điện. Hiện nay, FPGA có mật độ khá cao, lên tới hàng trăm tỷ cổng và cấu trúc cũng đa dạng, phức tạp hơn. Nhiều chức năng phức tạp đã được tích hợp sẵn để tăng hiệu quả sử dụng FPGA. Ví dụ như ngoài những khối tế bào logic, nhiều họ FPGA đã được tích hợp thêm các khối chức năng như các bộ nhân cứng, khối nhớ, PLL, thậm chí cả một bộ vi xử lý mạnh...

Có hai loại FPGA cơ bản: loại lập trình lại được, dựa trên công nghệ SRAM và loại lập trình một lần.

- Loại lập trình lại được (dựa trên SRAM):
  - SRAM xác định các kết nối
  - SRAM định nghĩa các hàm logic trong bảng ánh xạ (LUT- Look Up Table)
- Loại lập trình một lần:
  - Kết nối dạng bẻ cầu chì
  - Sử dụng các cổng logic truyền thống



Hình 8.4 - Cấu trúc của logic cell đơn giản

Hai dạng này khác nhau về quy trình thực hiện tế bào logic và cơ chế được sử dụng để tạo kết nối trong thiết bị.

Chip FPGA lập trình một lần sử dụng phương pháp bẻ cầu chì (kết nối được tạo ra bằng cách đóng cầu chì) để tạo kết nối tạm thời trong chip, do đó không cần SPROM hoặc các phương tiện khác để nạp chương trình vào FPGA. Tuy nhiên, mỗi lần thay đổi thiết kế, phải bỏ hoàn toàn chip cũ đi. Tế bào logic OTP tương tự như PLD với các cổng và các trigơ định trước.

Dạng FPGA quan trọng hơn và được dùng phổ biến hơn cả là dạng lập trình lại được, dựa trên SRAM. Trên thực tế, FPGA SRAM được lập trình lại mỗi khi bật nguồn, vì FPGA là dạng chip nhớ tạm thời. Do đó, mỗi chip FPGA đều cần có một bộ nhớ PROM nối tiếp hoặc một bộ nhớ hệ thống.

Trong tế bào logic SRAM, thay vì các cổng thông thường, người ta sử dụng bảng ánh xạ (LUT). Bảng này xác định các giá trị đầu ra dựa trên các giá trị đầu vào, sử dụng để xây dựng các hàm logic tổ hợp. Trong sơ đồ “Tế bào logic SRAM” minh họa ở hình vẽ 8-3, 16 tổ hợp khác nhau của 4 đầu vào sẽ xác định giá trị của đầu ra). Các các ô nhớ SRAM cũng được sử dụng để điều khiển kết nối .

### 8.5. SO SÁNH GIỮA CPLD VÀ FPGA

CPLD	FPGA
- Cấu trúc theo mảng các hạng tích	- Cấu trúc dựa vào LUT
- Mảng kết nối trung tâm	- Ma trận kết nối 2 chiều X-Y
- Mật độ tích hợp trung bình	- Mật độ tích hợp cao
- Tỷ lệ số chân I/O trên microcell lớn	- Tỷ lệ số chân I/O trên microcell nhỏ
- Cấu hình được lưu lại khi mất điện, và không đổi trong quá trình hoạt động	- Cấu hình nạp vào SRAM, khi mất điện sẽ không còn, cần có bộ nhớ cấu hình PROM, cấu hình có thể được nạp động trong quá trình hoạt động.
- Cấu trúc đồng nhất	- Cấu trúc không đồng nhất
	- Nhiều tài nguyên: DLL (Delay_Locked Loop: Vòng khoá pha trễ), bộ nhớ, các bộ nhân
- Ứng dụng: mã hoá và giải mã logic, các máy trạng thái hay các giao diện bus chuẩn (SPI, I2C, SMBus...), ưu điểm nổi bật khi thiết kế các mạch logic nhiều đầu vào.	- Ứng dụng: PCI (Peripheral Component Interface), giao tiếp nối tiếp tốc độ cao và các bộ vi xử lý nhúng ...., ưu thế nổi bật khi thiết kế phức tạp, cần nhiều tài nguyên.

### 8.6. QUY TRÌNH THIẾT KẾ CHO CPLD/FPGA

Trong thực tế có rất nhiều hãng điện tử trên thế giới cung cấp các sản phẩm PLD và bộ công cụ phần mềm thiết kế đi kèm. Mỗi họ CPLD, FPGA của các hãng có những quy trình thiết kế khác nhau dành cho chúng, tuy nhiên về cơ bản chúng vẫn có quy trình thiết kế chung nhất định. Không mất tính tổng quát, để người học tiếp cận dễ dàng hơn, trong tài liệu này việc trình bày quy trình thiết kế cho CPLD/FPGA được lấy ví dụ, cụ thể hoá cho CPLD/FPGA của hãng Xilinx – Một hãng cung cấp các sản phẩm PLD số 1 thế giới hiện nay – sử dụng bộ công cụ phần mềm thiết kế ISE.

#### 8.6.1. Yêu cầu chung khi thiết kế với CPLD/FPGA

##### 8.6.1.1 Chọn vi mạch CPLD hoặc FPGA phù hợp

Khi phát triển các hệ thống số sử dụng CPLD/FPGA bước đầu tiên cần được thực hiện là phân tích bài toán, lựa chọn vi mạch CPLD hoặc FPGA phù hợp. Việc chọn được vi mạch, công nghệ phù hợp nhất cho các tiêu chuẩn thiết kế, được tiến hành theo các yêu cầu sau:

**Mật độ:** là mật độ logic dự tính của linh kiện, đặc trưng bởi khái niệm "số lượng cổng".

**Số lượng thanh ghi:** Phải tính được số thanh ghi cần cho bộ đếm, máy trạng thái, thanh ghi và bộ chốt. Số lượng macrocell trong vi mạch tối thiểu phải bằng số thanh ghi cần có.

Số lượng chân vào/ra: phải xác định vi mạch thiết kế cần bao nhiêu đầu vào, bao nhiêu đầu ra.

**Yêu cầu về tốc độ:** Tuyến tổ hợp nhanh nhất sẽ xác định tpd (trễ truyền trong vi mạch, tính theo ns). Mạch tuần tự nhanh nhất sẽ xác định tần số tối đa của vi mạch ( $f_{Max}$ ).

**Đóng vỏ:** Phải xác định vi mạch cần gọn nhất hay chỉ sử dụng dạng QFP thông thường. Hoặc vi mạch thiết kế thuộc dạng có lắp chân cắm, trong trường hợp này là vi mạch PLCC.

**Công suất thấp:** Phải xác định sản phẩm sẽ sử dụng nguồn pin hay năng lượng mặt trời, thiết kế có yêu cầu công suất tiêu thụ thấp hay không, vấn đề tổn hao nhiệt có quan trọng hay không?

**Chức năng cấp hệ thống:** Phải xác định bo mạch có bao gồm nhiều vi mạch đa mức điện áp hay không, giữa các vi mạch có phải chuyển mức hay không, có yêu cầu sửa dạng xung đồng bộ hay không, có yêu cầu giao tiếp giữa bộ nhớ và bộ vi xử lý hay không?

### 8.6.1.2 Chọn giải pháp cấu hình cho CPLD/FPGA

#### Lập trình ngay trên hệ thống

Các CPLD và FPGA của các hãng nói chung, của Xilinx nói riêng có thể được lập trình ngay trên hệ thống (vi mạch đã được hàn vào mạch ứng dụng) thông qua giao thức JTAG (Joint Test Advisory Group: Chuẩn giao tiếp) đã được tích hợp sẵn trong IC. Người thiết kế sử dụng cấp nạp để nạp cấu hình cho CPLD hoặc FPGA. Xilinx đưa ra một chuẩn cấp nạp như sau:

+ **MultiLINX** : Cấp nạp dựa trên giao chuẩn giao tiếp nối tiếp USB hoặc RS232, cấp nạp này có tốc độ truyền trong dải rộng và giao diện có điện áp điều chỉnh được để phù hợp với việc giao tiếp với các hệ thống và các chân I/O hoạt động ở các mức điện áp khác nhau 5V; 3,3V; 2,5V. Và được thiết kế để hỗ trợ để cho các phần mềm gỡ rối phần cứng trước kia, nay chúng đã trở lên lỗi thời khi có sự ra đời của công cụ gỡ rối phần cứng ChipScope ILA.

+ **Parallel Cable IV**: Cấp nạp sử dụng cổng giao tiếp song song của máy tính, được phát triển để thay thế cho chuẩn cấp nạp Parallel Cable III và cho phép tăng tốc độ lên hơn 10 lần và hỗ trợ cho tất cả các vi mạch sử dụng mức điện áp I/O từ 5V xuống 1,5V. Hiện nay chuẩn cấp nạp này được dùng phổ biến hơn cả.

#### Lập trình bên ngoài

Các CPLD và FPGA của Xilinx cũng có thể được lập trình bên ngoài bởi bộ lập trình chip HW130 của Xilinx cũng như các bộ lập trình của các nhà phát triển khác. Điều này cũng thuận tiện cho việc sử dụng các chip được lập trình trước trong thời gian sản xuất.

Cấu hình của CPLD được nạp vào FLASH nên khi mất điện cấu hình không bị mất đi, trong khi đó cấu hình khi hoạt động của FPGA được ghi vào SRAM nên sẽ mất đi khi mất điện, vì vậy cần sử dụng FPGA và kết hợp với PROM lưu cấu hình phù hợp, mỗi khi bật nguồn, cấu hình sẽ nạp tự động từ PROM vào FPGA. Có thể sử dụng PROM nối tiếp hoặc song song, tuy nhiên thì loại PROM nối tiếp hay được sử dụng hơn cả. Khi thiết kế cần chọn loại PROM có dung lượng phù hợp với mật độ của các loại FPGA khác nhau.

Ngoài ra Xilinx còn cung cấp các giải pháp được thiết kế trước, dễ sử dụng để cấu hình cho tất cả CPLD và FPGA của Xilinx, nhất là khi thiết kế các hệ thống phức tạp. Tất cả các nội dung liên quan đến cấu hình, PROM cho FPGA hay ISP cho CPLD, đều được đưa ra. Các giải pháp sử

dụng công cụ 3rd part boundary scan, các giải pháp phần mềm kèm theo, cấp ISP, thiết bị kiểm tra tự động ATE và hỗ trợ lập trình cũng như các thiết bị lưu trữ cấu hình.

Giải pháp cấu hình hiện đại nhất là nhóm cấu hình System ACE. Với giải pháp System ACE, người thiết kế có thể dễ dàng sử dụng giao diện vi xử lý trong System ACE để trực tiếp phối hợp cấu hình FPGA theo các yêu cầu của hệ thống. Giải pháp đầu tiên trong nhóm này là System ACE CF, cung cấp công nghệ điều khiển ổ đĩa Microdrive kích thước một inch và CompactFlash cũng như bộ lưu trữ cấu hình có dung lượng 8 gigabits. Ngoài ra, System ACE CF cũng được thiết kế trước, cung cấp các đặc tính hiện đại để tận dụng khả năng cấu hình lại linh hoạt của FPGA, bao gồm:

- Cấu hình multi-board từ một nguồn duy nhất
- Quản lý bitstream đa cấu hình
- Nâng cấp cấu hình qua mạng (IRL)
- Hot-swapping
- Khởi tạo trung tâm xử lý và lưu trữ phần mềm
- Mã hóa

Với System ACE CF, người thiết kế có thể thực hiện được gần như toàn bộ các yêu cầu cấu hình cho FPGA. Các khả năng hỗ trợ hệ thống này cho phép người thiết kế sử dụng FPGA thỏa mãn các yêu cầu định trước về mặt thiết kế và thời gian xử lý lỗi. Ngoài ra, các cổng vi xử lý và cổng kiểm tra JTAG còn cho phép tích hợp System ACE trong mọi hệ thống.

Một số đặc điểm của giải pháp cấu hình System ACE:

**- Độ linh hoạt:** Với System ACE CF, có thể sử dụng một thiết kế cho nhiều ứng dụng khác nhau, nhờ đó giảm đáng kể thời gian hoàn thành sản phẩm. Thay vì thiết kế vài bo mạch tương tự nhau phù hợp với các chuẩn khác nhau, giờ đây người thiết kế chỉ phải thiết kế một bo mạch duy nhất với nhiều cấu hình được lưu trữ trong bộ nhớ System ACE CF. Mỗi bo có thể chọn các cấu hình phù hợp với các chuẩn khác nhau bằng cách khởi tạo giá trị mặc định tương ứng được lưu trong bộ nhớ ACE. Hệ thống còn cho phép lưu nhiều cấu hình cho một thiết kế trong một System ACE CF đơn. Ví dụ như trong quá trình thiết kế mẫu, người thiết kế có thể lưu các cấu hình hoạt động, cấu hình kiểm tra và cấu hình gỡ rối trong bộ nhớ ACE, đồng thời có thể chọn các cấu hình khác để chạy thử bản thiết kế của mình.

Để hỗ trợ quản lý nhiều bitstream và tích hợp điều khiển cấu hình FPGA với hoạt động của hệ thống, System ACE có một cổng vi xử lý trong hệ thống. Cổng này cho phép bộ xử lý của hệ thống thay đổi cấu hình mặc định, cấu hình lại trigger, cấu hình lại từng FPGA hoặc một nhóm FPGA, truy nhập vào các file không cấu hình được lưu trong khối CompactFlash, hoặc dùng khối CompactFlash làm bộ nhớ chung cho hệ thống.

Với các FPGA có trung tâm xử lý kèm theo, System ACE CF cung cấp giải pháp 3 trong 1 để quản lý phần cứng và phần mềm. System ACE CF có thể cấu hình khung FPGA, khởi tạo trung tâm vi xử lý, và cung cấp các ứng dụng phần mềm cho trung tâm này nếu cần mà không phải thêm bất cứ thiết bị phần cứng nào.

+ **Mật độ:** Với mật độ logic cao chưa từng thấy (trên 8 Gb), một System ACE CF có thể cấu hình cho hàng trăm FPGA và có thể thay thế cho các mảng PROM cấu hình. Người thiết kế có thể lưu một số lượng lớn các thiết kế khác nhau cho một mảng FPGA trong cùng một khối nhớ. System ACE CF sử dụng hệ thống file FAT (File Allocation Table: bảng sắp xếp file tiêu chuẩn), do đó người thiết kế có thể lưu cả những file không ở dạng bitstream hoặc sử dụng bộ nhớ thừa làm bộ nhớ chuẩn cho hệ thống.

+ **Khả năng quản lý tập trung:** System ACE CF được thiết kế để quản lý cấu hình theo yêu cầu. Một System ACE CF có thể cấu hình cho một hoặc nhiều bo FPGA kết nối qua một back-plane. Khả năng tập trung cho phép đơn giản hóa quá trình quản lý và nâng cấp cấu hình. Để thay đổi hay nâng cấp cấu hình của một hệ thống, người thiết kế có thể vào khối nhớ, thực hiện các thay đổi cần thiết trên màn hình máy tính, chỉnh lại nội dung trong hệ thống qua công vi xử lý; hoặc tải cấu hình mới về qua mạng, sử dụng IRL.

### 8.6.1.3 Chọn công cụ phần mềm phù hợp

Xilinx đã cung cấp các công cụ thiết kế điện tử hoàn chỉnh, cho phép thực hiện thiết kế trên các thiết bị logic khả trình của Xilinx. Các công cụ này kết hợp công nghệ tiên tiến với giao diện đồ họa linh hoạt, dễ sử dụng để người thiết kế có được thiết kế tối ưu. Bộ công cụ phần mềm hiện đang được sử dụng rộng rãi là ISE với phiên bản mới nhất là 7.0 (năm 2005).

Xilinx cũng cung cấp ISE dưới dạng các gói phần mềm có cấu hình khác nhau với giá thành khác nhau:

- + ISE WebPACK - bản miễn phí có thể dùng để thiết kế cho tất cả các họ CPLD của Xilinx
- + Gói phần mềm cơ bản BASEX: có thể thiết kế cho các loại chip sau:

Virtex-4, FPGA LX15, LX25, SX25, FX12, Spartan-3 FPGA lên đến 1500 ngàn cổng và tất cả các họ CPLD.

- + Gói phần mềm Foundation: có thể thiết kế cho tất cả các loại FPGA và CPLD của Xilinx

Ngoài ra Xilinx còn phát triển các bộ công cụ phần mềm tiện ích khác như System Generator hỗ trợ cho các thiết kế DSP (Digital Signal Processor: Bộ xử lý tín hiệu số), hay EDK (Embladed Development Kit: Bộ phần mềm phát triển hệ thống) hỗ trợ cho các thiết kế nhúng.

ISE được dùng kết hợp với phần mềm mô phỏng ModelSim của Mentor Graphics phiên bản XE được phát triển riêng hỗ trợ cho các họ CPLD/FPGA của Xilinx.

### 8.6.2 Lưu đồ thiết kế cho CPLD của Xilinx

Quá trình thiết kế cho CPLD chủ yếu là thực hiện trên các công cụ phần mềm, lưu đồ thiết kế chung cho CPLD (Ví dụ sử dụng phần mềm ISE) như hình vẽ sau, bao gồm các bước như sau:

- + Nhập thiết kế (Design Entry):

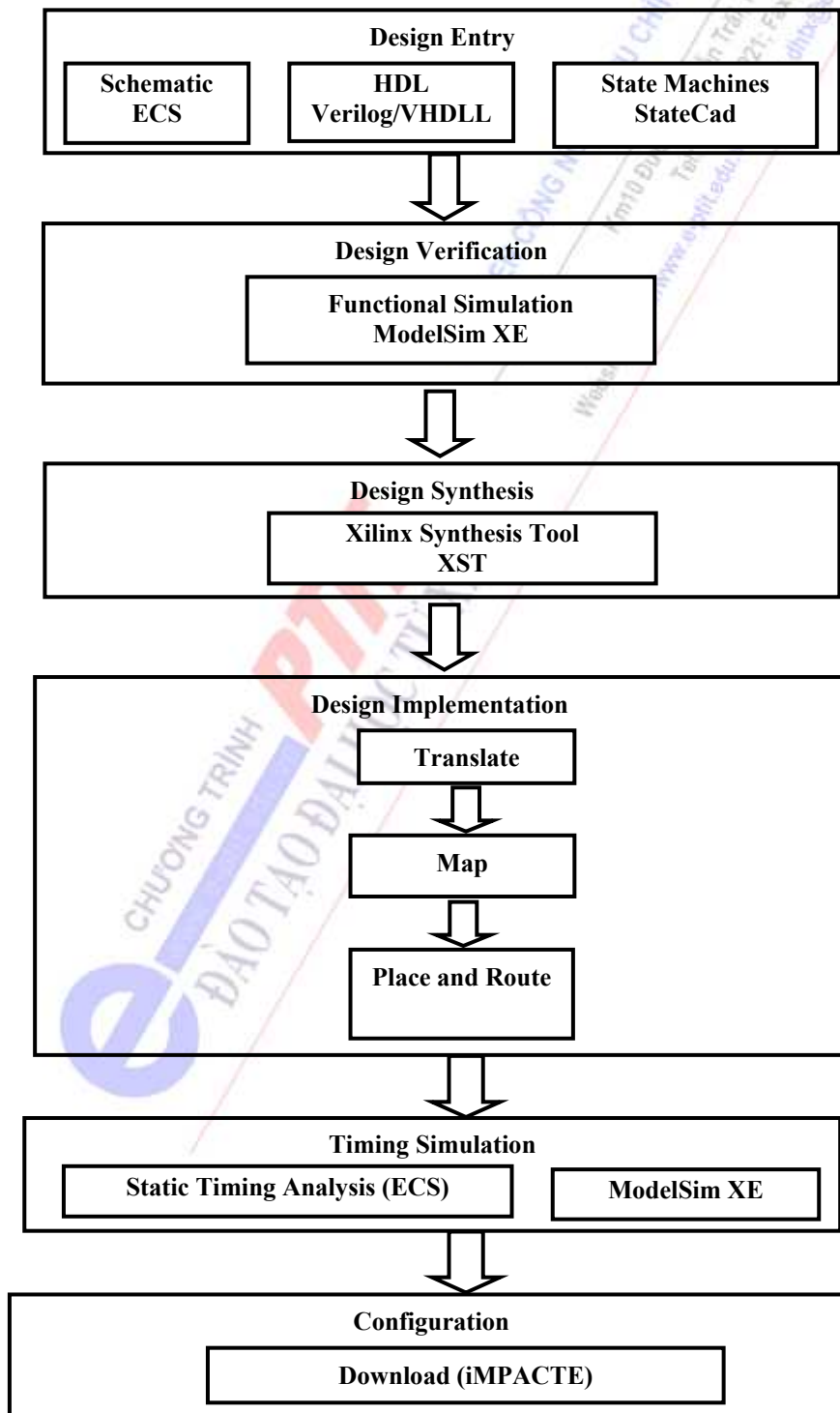
Đây là bước đầu tiên và quan trọng nhất của quá trình thiết kế cho CPLD. Các công cụ thiết kế cho phép nhập thiết kế cho phép nhập thiết kế theo các cách sau:

- Nhập thiết kế theo sơ đồ nguyên lý Schematic, người thiết kế sử dụng các modul đã có sẵn trong thư viện Schematic để ghép nối chúng với nhau tạo thành bản thiết kế theo yêu cầu, cách này có thể thực hiện thiết kế nhanh nhưng sẽ rất khó khăn và không tối ưu tài nguyên của CPLD



khi thiết kế phức tạp, và thiết kế không thể sử dụng sang công cụ thiết kế CPLD của các hãng khác. Từ sơ đồ nguyên lý thiết kế được công cụ phần mềm sẽ chuyển đổi sang file ngôn ngữ mô tả phần cứng HDL, mà phổ biến là VHDL hoặc Verilog.

- Nhập thiết kế sử dụng ngôn ngữ mô tả phần cứng HDL (VHDL, Verilog, ABEL, AHDL...), Người thiết kế có thể sử dụng chương trình soạn thảo để thực hiện việc mô tả toàn bộ bản thiết kế của mình dưới dạng ngôn ngữ HDL nào đó mà công cụ thiết kế có thể tổng hợp được. Có rất nhiều phương pháp mô tả, mức độ trừu tượng khác nhau khi thiết kế, mỗi cách mô tả khác nhau có thể tạo ra một cấu trúc mạch khác nhau trong CPLD mặc dù chúng có cùng chức năng.



**Hình 8.5- Lưu đồ thiết kế CPLD**

Do đó người thiết kế cần thực hiện phân tích bài toán, tìm hiểu tài nguyên, cấu trúc của CPLD, yêu cầu về thời gian thiết kế để sử dụng kiểu mô tả. Mức độ trừu tượng trong khi mô tả phù hợp vừa đảm bảo yêu cầu về thời gian thiết kế vừa tối ưu được việc sử dụng tài nguyên của CPLD.

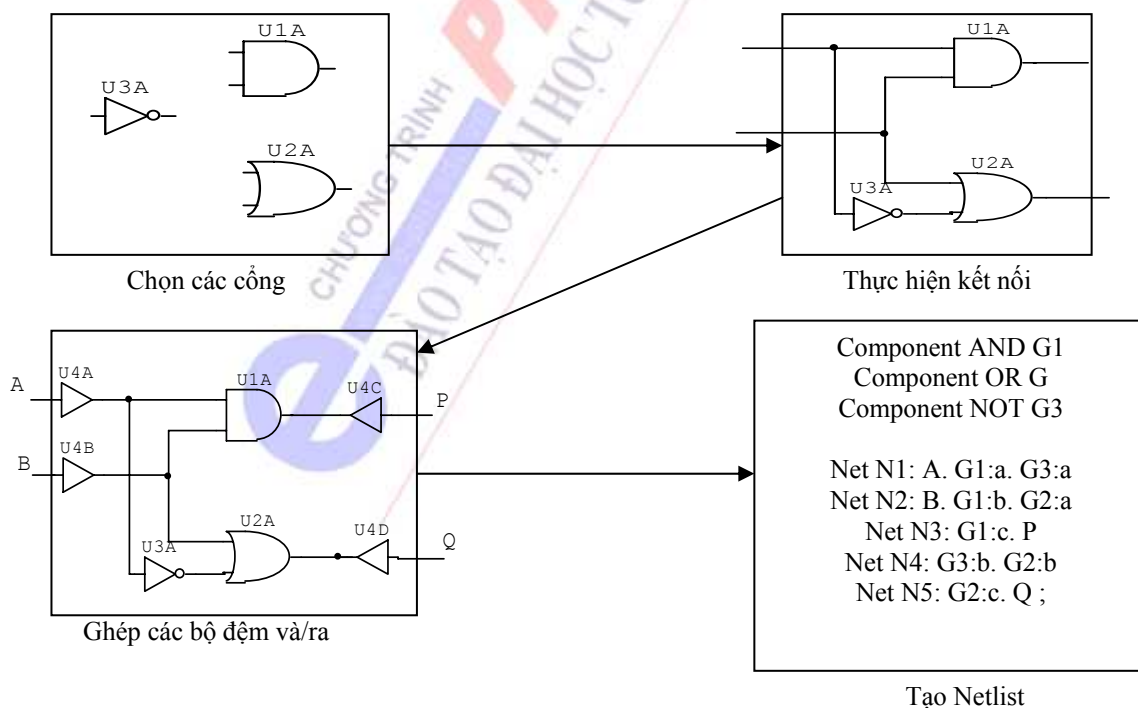
- Nhập thiết kế dưới dạng sơ đồ: Công cụ thiết kế còn cho phép nhập thiết kế vào dưới dạng sơ đồ mà điển hình là đồ hình trạng thái, sau đó chúng cũng được chuyển đổi sang HDL.

Việc nhập thiết kế rất linh hoạt, có thể sử dụng cả 3 cách trên để thực hiện các phần khác nhau của thiết kế.

+ **Kiểm tra, mô phỏng thiết kế (Design Verification)**: Thực hiện kiểm tra, mô phỏng chức năng hoạt động của thiết kế HDL đã tạo ra ở trên. Các công cụ thiết kế đều hỗ trợ việc mô phỏng chức năng hoạt động của bản thiết kế HDL theo mô hình hoạt động (Behavioral Model), mức độ mô phỏng này độc lập với loại CPLD đã được lựa chọn. Bước này có thể không cần phải thực hiện trong khi thiết kế.

+ **Tổng hợp thiết kế (Design Synthesis)**: Sau khi hoàn thành mô phỏng thiết kế, bước tổng hợp tiếp theo có nhiệm vụ chuyển thiết kế dưới dạng file văn bản HDL thành dạng file netlist, thực hiện mô tả mạch thực ở mức thấp dưới dạng cổng logic và kết nối giữa chúng với nhau. Có thể sử dụng các công cụ tổng hợp của các hãng khác nhau.

Mỗi công cụ có thể tạo ra file netlist theo định dạng riêng (ví dụ của XST của Xilinx XNF- Xilinx Netlist Format) nhưng có thể đặt lựa chọn để tạo ra file netlist dưới dạng định dạng chuẩn EDIF (Electronic Digital Interchange Format) mà tất cả các công cụ có thể hiểu được.



Hình 8.6- Ví dụ tổng hợp ra file netlist

+ **Thực hiện thiết kế (Design Implementation)**: Sau khi có file netlist, bước tiếp theo là thực hiện thiết kế, nghĩa là xây dựng cấu hình cho CPLD. Bước này sử dụng file netlist và file ràng buộc "constraints File" (mô tả các nguyên tắc thiết kế, các ràng buộc về vật lý như gán vị trí cho các đầu vào/ra trên chip, các ràng buộc về tốc độ, thời gian, tần số...) để tạo thiết kế sử dụng tài nguyên có sẵn của CPLD. Bước này bao gồm các bước: Translate (biên dịch), Map (Phân bố bản thiết kế vào chip), Place and Route (Định vị và định tuyến kết nối).

+ **Translate (biên dịch)**: Bước này nhằm thực hiện kiểm tra thiết kế và đảm bảo netlist phù hợp với kiến trúc đã chọn, kiểm tra file ràng buộc "constraints File" của người sử dụng để phát hiện các lỗi mâu thuẫn với tham số của chip đã chọn. Biên dịch thường bao gồm các quá trình: tối ưu hoá, biên dịch thành các thành phần vật lý của thiết bị; kiểm tra ràng buộc thiết kế. Khi kết thúc bước biên dịch, sẽ có một bản báo cáo về các chương trình được sử dụng, danh sách các cổng I/O và các thiết bị được sử dụng trong thiết kế, nhờ đó người thiết kế sẽ lựa chọn được phương án thiết kế tối ưu.

+ **Map**: tạo bản phân bố thiết kế tới các tài nguyên cụ thể trong CPLD. Nếu thiết kế quá lớn so với thiết bị được chọn, quy trình này không thể hoàn thành nhiệm vụ của mình. Quá trình Map có các tham số ràng buộc của thiết kế, ví dụ như tham số tốc độ, thời gian của thiết kế, và đôi khi quyết định gắn thêm các thành phần logic để đáp ứng các yêu cầu về thời gian. Map có khả năng thay đổi thiết kế xung quanh các bảng ánh xạ để tạo khả năng thực hiện tốt nhất cho thiết kế. Quy trình này được thực hiện hoàn toàn tự động và cần rất ít tác động đầu vào từ người sử dụng. Bước này nhằm đưa mạch thiết kế vào một thiết bị cụ thể. Bước này cũng tạo ra báo cáo xác nhận các tài nguyên được sử dụng trong chip, mô tả chính xác các phần trong thiết kế được đặt ở vị trí nào trong chip thực tế.

+ **Place and Route (PAR - Định vị trí và định tuyến kết nối)** Place là quá trình lựa chọn vị trí phù hợp của mỗi khối chức năng trong thiết kế và đưa các cổng logic của phần đó vào các khối logic hay các modul cụ thể trong CPLD trên cơ sở tối ưu việc kết nối và đảm bảo về các ràng buộc về thời gian. Những phần logic hoạt động tốc độ cao sẽ được xếp cạnh nhau để giảm độ dài đường kết nối. Route là quá trình tạo liên kết vật lý giữa các khối logic. Hầu hết các nhà sản xuất cung cấp công cụ Place and Route tự động cho người sử dụng. Ngoài công cụ tự động, người thiết kế có thể tự Place and Route trong khi thiết kế. Nhà sản xuất cũng cung cấp các công cụ, như "Floorplanner", để nâng cao hiệu suất quá trình Place and Route do người thiết kế thực hiện so với quá trình tự động.

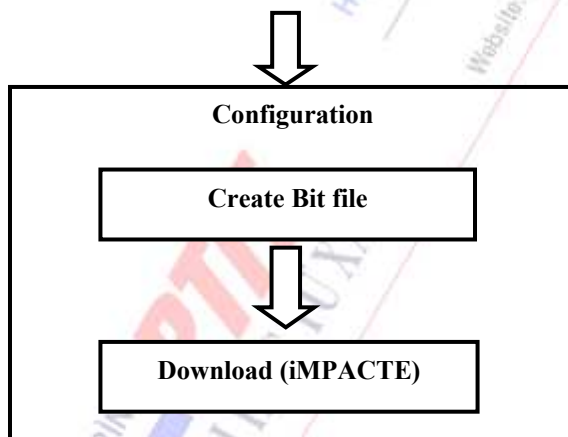
Place and Route là quá trình phức tạp, do đó nó chiếm thời gian nhiều nhất. Tuy nhiên, bước này chỉ có thể hoạt động tốt nếu chip đã chọn đáp ứng đủ các tuyến liên kết cho thiết kế. Nếu không, người thiết kế sẽ phải chọn chip có dung lượng lớn hơn. Sau bước này tạo ra được file cấu hình \*.jed có thể được nạp vào cho CPLD.

+ **Timing Simulation (Mô phỏng có tham số thời gian)**: Sau bước Place and Route người thiết kế có thể thực hiện mô phỏng thiết kế ở mức cổng logic đã được định vị trí và định tuyến trên CPLD, phần mềm sử dụng file cấu hình đã được tạo ra và kết hợp với thư viện về mô hình thời gian của các họ CPLD (Ví dụ ISE của Xilinx thì dùng thư viện VITAL), để thực hiện mô phỏng hoạt động của thiết kế mà có tính đến các tham số thời gian trễ, thời gian thiết lập... của các cổng logic trong CPLD. Bước này rất quan trọng với những thiết kế phức tạp, tốc độ lớn.

+**Configuration (Cấu hình)**: Gọi chương trình điều khiển việc nạp cấu hình, thực kết nối thiết bị nạp (cáp nạp) đến CPLD và nạp file cấu hình cho CPLD. Với CPLD của hãng Xilinx, quá trình lập trình có thể thực hiện ngay trong hệ thống nhờ công cụ JTAG, hoặc sử dụng bộ lập trình thiết bị chuyên dùng, ví dụ như công cụ JTAG Data I/O, theo chuẩn IEEE/ ANSI 1149.1\_1190. Công cụ JTAG là một bộ các nguyên tắc thiết kế, hỗ trợ quá trình kiểm tra, lập trình cho thiết bị và gỡ rối trên chip, trên bo mạch và trên hệ thống. Khả năng lập trình trên hệ thống là ưu điểm của CPLD, cho phép hàn trực tiếp thiết bị lên PCB. Nếu có thay đổi trong thiết kế, sẽ không phải tháo thiết bị ra khỏi bo mạch, mà đơn giản chỉ phải lập trình lại trên hệ thống.

### 8.6.3 Lưu đồ thiết kế cho FPGA

Lưu đồ thiết kế cho FPGA cũng tương tự như lưu đồ thiết kế cho CPLD, chỉ khác ở bước cuối cùng - bước Cấu hình cho FPGA. Ở bước này, đối với FPGA có thêm bước "Create Bit file" để tạo ra file "bitstream" để nạp vào bộ nhớ cấu hình trong FPGA thường là bộ nhớ tạm thời như SRAM. Dòng bit được nạp mang tất cả thông tin để định nghĩa các hàm logic và các liên kết trong thiết kế. Mỗi thiết kế khác nhau có một dòng bit khác nhau. Các thiết bị SRAM mất toàn bộ thông tin mỗi khi ngắt nguồn, do đó khi cần thiết phải nạp dòng bit cấu hình này vào trong PROM (thường sử dụng PROM nối tiếp). Mỗi khi thiết bị được bật nguồn file cấu hình từ PROM sẽ được nạp tự động vào bộ nhớ SRAM của FPGA, và FPGA hoạt động theo cấu hình đã được nạp đó.



Hình 8.6- Lưu đồ thiết kế FPGA

## TÓM TẮT

Trong chương này trình bày các khái niệm cơ bản của logic lập trình. Với sự phát triển của các thiết bị logic lập trình ta có thể thiết kế các hệ thống kỹ thuật số phức tạp. Các kỹ thuật thiết kế ở cấp cao và các công cụ trợ giúp máy tính cần thiết để tạo nên chức năng thực thi PLD và FPGA hiệu quả. Việc thử nghiệm tính thực thi của PLD và FPGA cũng yêu cầu phải có các công cụ thử nghiệm và sự trợ giúp của máy tính.

## CHƯƠNG 9: NGÔN NGỮ MÔ TẢ PHẦN CỨNG VHDL

### GIỚI THIỆU

Trong toàn bộ lưu đồ thiết kế cho CPLD hoặc FPGA, bước nhập thiết kế là bước quan trọng và tốn nhiều công sức nhất, nó quyết định phần lớn đến kết quả của công việc thiết kế. Các công cụ thiết kế hỗ trợ nhiều phương pháp nhập thiết kế khác nhau, tuy nhiên phương pháp nhập thiết kế dùng ngôn ngữ mô tả phần cứng HDL là ưu việt hơn cả và được sử dụng chủ yếu trong quá trình thiết kế số nói chung và thiết kế cho CPLD/FPGA nói riêng. Hiện nay có nhiều ngôn ngữ HDL được sử dụng, tuy nhiên trong phần này chỉ giới thiệu phương pháp thiết kế dùng ngôn ngữ VHDL và giới thiệu những đặc điểm của VHDL khiến nó được trở thành một ngôn ngữ HDL đang được giảng dạy và sử dụng ở nhiều trường đại học trên thế giới.

Ngày nay, các mạch tích hợp ngày càng thực hiện được nhiều chức năng, do đó, vấn đề thiết kế mạch càng trở nên phức tạp. Những phương pháp truyền thống như dùng phương pháp tối thiểu hoá hàm Boolean hay dùng sơ đồ các phần tử không còn đáp ứng được các yêu cầu đặt ra khi thiết kế. Nhược điểm lớn nhất của các phương pháp này là chúng chỉ mô tả được hệ thống dưới dạng mạng nối các phần tử với nhau. Người thiết kế cần phải đi qua hai bước thực hiện hoàn toàn thủ công: đó là chuyển từ các yêu cầu về chức năng của hệ thống sang biểu diễn theo dạng hàm Boolean, sau các bước tối thiểu hoá hàm này ta lại phải chuyển từ hàm Boolean sang sơ đồ mạch của hệ thống. Cũng tương tự khi phân tích một hệ thống người phân tích cần phải phân tích sơ đồ mạch của hệ thống, rồi chuyển nó thành các hàm Boolean, sau đó mới lập lại các chức năng, hoạt động của hệ thống. Tất cả các bước nói trên hoàn toàn phải thực hiện thủ công không có bất kỳ sự trợ giúp nào của máy tính. Người thiết kế chỉ có thể sử dụng máy tính làm công cụ hỗ trợ trong việc vẽ sơ đồ mạch của hệ thống và chuyển từ sơ đồ mạch sang công cụ tổng hợp mạch vật lý dùng công cụ Synthesis. Một nhược điểm khác nữa của phương pháp thiết kế truyền thống là sự giới hạn về độ phức tạp của hệ thống được thiết kế. Phương pháp dùng hàm Boolean chỉ có thể dùng để thiết kế hệ thống lớn nhất biểu diễn bởi vài trăm hàm. Còn phương pháp dựa trên sơ đồ chỉ có thể dùng để thiết kế hệ thống lớn nhất chứa khoảng vài nghìn phần tử.

Phương pháp thiết kế, thử nghiệm, phân tích các hệ thống số sử dụng các ngôn ngữ mô tả phần cứng nổi bật lên với các ưu điểm hơn hẳn và sẽ dần thay thế các phương pháp truyền thống. Sự ra đời của ngôn ngữ mô phỏng phần cứng đã giải quyết được rất nhiều nhược điểm lớn của các phương pháp thiết kế trước đây: Nếu các phương pháp cũ đòi hỏi phải chuyển đổi từ mô tả hệ thống (các chỉ tiêu về chức năng) sang tập hợp các hàm logic bằng tay thì bước chuyển đó hoàn toàn không cần thiết khi dùng HDL. Hầu hết các công cụ thiết kế dùng ngôn ngữ mô phỏng phần cứng đều cho phép sử dụng biểu đồ trạng thái (finite-state-machine) cho các hệ thống tuần tự cũng như cho phép sử dụng bảng chân lý cho hệ thống tổng hợp. Việc chuyển đổi từ các biểu đồ trạng thái và bảng chân lý sang mã ngôn ngữ mô phỏng phần cứng được thực hiện hoàn toàn tự động.

Nhờ tính dễ kiểm tra thử nghiệm hệ thống trong suốt quá trình thiết kế mà người thiết kế có thể dễ dàng phát hiện các lỗi thiết kế ngay từ những giai đoạn đầu, giai đoạn chưa đưa vào sản xuất thử, do đó tiết kiệm được lượng chi phí đáng kể bởi từ ý thiết kế đến tạo ra sản phẩm đúng như mong muốn là một việc rất khó tránh khỏi những khó khăn, thất bại.

Khi mọi lĩnh vực của khoa học đều phát triển không ngừng thì sự phức tạp của hệ thống điện tử cũng ngày một tăng theo và gần như không thể tiến hành thiết kế thủ công mà không có sự trợ giúp của các loại máy tính hiện đại. Ngày nay, ngôn ngữ mô tả phần cứng HDL được dùng nhiều để thiết kế cho các thiết bị logic lập trình được PLD từ loại đơn giản đến các loại phức tạp như FPGA.

## NỘI DUNG

### 9.1. GIỚI THIỆU NGÔN NGỮ MÔ TẢ PHẦN CỨNG VHDL

VHDL là ngôn ngữ mô tả phần cứng cho các mạch tích hợp tốc độ rất cao, là một loại ngôn ngữ mô tả phần cứng được phát triển dùng cho trương trình VHSIC( Very High Speed Itergrated Circuit) của bộ quốc phòng Mỹ. Mục tiêu của việc phát triển VHDL là có được một ngôn ngữ mô phỏng phần cứng tiêu chuẩn và thống nhất cho phép thử nghiệm các hệ thống số nhanh hơn cũng như cho phép dễ dàng đưa các hệ thống đó vào ứng dụng trong thực tế. Ngôn ngữ VHDL được ba công ty Intermetics, IBM và Texas Instruments bắt đầu nghiên cứu phát triển vào tháng 7 năm 1983. Phiên bản đầu tiên được công bố vào tháng 8-1985. Sau đó VHDL được đề xuất để tổ chức IEEE xem xét thành một tiêu chuẩn chung. Năm 1987 đã đưa ra tiêu chuẩn về VHDL( tiêu chuẩn IEEE-1076-1987).

VHDL được phát triển để giải quyết các khó khăn trong việc phát triển, thay đổi và lập tài liệu cho các hệ thống số. Như ta đã biết, một hệ thống số có rất nhiều tài liệu mô tả. Để có thể vận hành bảo trì sửa chữa một hệ thống ta cần tìm hiểu kỹ lưỡng tài liệu đó. Với một ngôn ngữ mô phỏng phần cứng tốt việc xem xét các tài liệu mô tả trở nên dễ dàng hơn vì bộ tài liệu đó có thể được thực thi để mô phỏng hoạt động của hệ thống. Như thế ta có thể xem xét toàn bộ các phần tử của hệ thống hoạt động trong một mô hình thống nhất.

VHDL được phát triển như một ngôn ngữ độc lập không gắn với bất kỳ một phương pháp thiết kế, một bộ mô tả hay công nghệ phần cứng nào. Người thiết kế có thể tự do lựa chọn công nghệ, phương pháp thiết kế trong khi chỉ sử dụng một ngôn ngữ duy nhất. Và khi đem so sánh với các ngôn ngữ mô phỏng phần cứng khác đã kể ra ở trên ta thấy VHDL có một số ưu điểm hơn hẳn các ngôn ngữ khác:

+ Thứ nhất là tính công cộng: VHDL được phát triển dưới sự bảo trợ của chính phủ Mỹ và hiện nay là một tiêu chuẩn của IEEE. VHDL được sự hỗ trợ của nhiều nhà sản xuất thiết bị cũng như nhiều nhà cung cấp công cụ thiết kế mô phỏng hệ thống.

+ Thứ hai là khả năng hỗ trợ nhiều công nghệ và phương pháp thiết kế. VHDL cho phép thiết kế bằng nhiều phương pháp ví dụ phương pháp thiết kế từ trên xuống, hay từ dưới lên dựa vào các thư viện sẵn có. VHDL cũng hỗ trợ cho nhiều loại công cụ xây dựng mạch như sử dụng công nghệ đồng bộ hay không đồng bộ, sử dụng ma trận lập trình được hay sử dụng mảng ngẫu nhiên.

+ Thứ ba là tính độc lập với công nghệ: VHDL hoàn toàn độc lập với công nghệ chế tạo phần cứng. Một mô tả hệ thống dùng VHDL thiết kế ở mức cổng có thể được chuyển thành các bản tổng hợp mạch khác nhau tùy thuộc công nghệ chế tạo phần cứng mới ra đời nó có thể được áp dụng ngay cho các hệ thống đã thiết kế .

+ Thứ tư là khả năng mô tả mở rộng: VHDL cho phép mô tả hoạt động của phần cứng từ mức hệ thống số cho đến mức cổng. VHDL có khả năng mô tả hoạt động của hệ thống trên nhiều mức nhưng chỉ sử dụng một cú pháp chặt chẽ thống nhất cho mọi mức. Như thế ta có thể mô phỏng một bản thiết kế bao gồm cả các hệ con được mô tả chi tiết.

+ Thứ năm là khả năng trao đổi kết quả: Vì VHDL là một tiêu chuẩn được chấp nhận, nên một mô hình VHDL có thể chạy trên mọi bộ mô tả đáp ứng được tiêu chuẩn VHDL. Các kết quả mô tả hệ thống có thể được trao đổi giữa các nhà thiết kế sử dụng công cụ thiết kế khác nhau nhưng cùng tuân theo tiêu chuẩn VHDL. Cũng như một nhóm thiết kế có thể trao đổi mô tả mức cao của các hệ thống con trong một hệ thống lớn (trong đó các hệ con đó được thiết kế độc lập).

+ Thứ sáu là khả năng hỗ trợ thiết kế mức lớn và khả năng sử dụng lại các thiết kế: VHDL được phát triển như một ngôn ngữ lập trình bậc cao, vì vậy nó có thể được sử dụng để thiết kế một hệ thống lớn với sự tham gia của một nhóm nhiều người. Bên trong ngôn ngữ VHDL có nhiều tính năng hỗ trợ việc quản lý, thử nghiệm và chia sẻ thiết kế. Và nó cũng cho phép dùng lại các phần đã có sẵn.

## 9.2. CẤU TRÚC NGÔN NGỮ CỦA VHDL

VHDL là ngôn ngữ cho phép mô tả các thiết bị phần cứng số trừu tượng, nó không dựa vào công nghệ thiết bị phần cứng số, phương pháp được sử dụng để thiết kế thiết bị số, mà những khái niệm, mô hình trừu tượng của thiết bị phần cứng số được đưa ra như là nền tảng của ngôn ngữ. Do đó dùng VHDL cho phép mô tả được hầu hết các hệ thống phần cứng số. Các mô hình trừu tượng gồm:

- Mô hình hoạt động (a Model of Behavior).
- Mô hình thời gian (a Model of Time).
- Mô hình cấu trúc (a Model of Structure).

Để thực hiện mô tả cho một hệ thống số nào đó cần thực hiện theo các bước như sau:

- + Phân tích yêu cầu của hệ thống số cần phải thiết kế hoặc cần phải mô tả.
- + Phân tách hệ thống thành những khối con.
- + Xác định mô hình mô tả phù hợp cho mỗi khối con hoặc cho cả hệ thống.
- + Sử dụng ngôn ngữ VHDL để mô tả hệ thống số theo các mô hình đã xác định.

Như vậy việc nắm chắc cấu trúc, cú pháp, các mô hình mô tả của ngôn ngữ là rất quan trọng, quyết định chủ yếu đến thành công trong việc mô tả hệ thống số cần thiết kế.

VHDL cũng có nhiều điểm giống như một ngôn ngữ lập trình bậc cao, có cấu trúc, có cú pháp riêng, có cách tổ chức chương trình, có từ khóa, có phương pháp biểu diễn số liệu riêng...

**Chú ý:** - Trong các đoạn mã mô tả VHDL trong chương các từ khóa đều được in đậm.

- Trong VHDL không phân biệt chữ hoa, chữ thường.

### 9.2.1 Đối tượng trong VHDL

Trong ngôn ngữ VHDL gồm có 3 đối tượng là: tín hiệu - **signal**, biến - **variable**, hằng - **constant**, mỗi đối tượng được khai báo dựa vào từ khóa tương ứng và chúng có mục đích sử dụng như sau:

+ **Tín hiệu – Signal**: là đối tượng để biểu diễn đường kết nối các giữa các cổng vào/ra của thực thể, giữa các cổng vào/ra của các khối thành phần phần cứng xuất hiện trong thực thể... Chúng là phương tiện truyền dữ liệu động giữa các thành phần của thực thể.

Tín hiệu có tính toàn cục rất cao, chúng có thể được khai báo trong package (tín hiệu toàn cục, được sử dụng bởi một số thực thể), khai báo trong thực thể - Entity (tín hiệu nội bộ dùng trong thực thể, có thể được tham chiếu bởi bất kỳ kiến trúc nào của thực thể đó), khai báo trong kiến trúc – Architecture (tín hiệu nội bộ dùng trong kiến trúc, có thể được sử dụng trong bất cứ cấu trúc lệnh nào trong kiến trúc). Các tín hiệu có thể được sử dụng nhưng không được khai báo trong tiến trình – process, trong chương trình con. Vì tiến trình và chương trình con là thành phần cơ sở của mô hình và chúng được coi như các hộp đen. Cú pháp khai báo tín hiệu như sau:

```
Signal tên_tín_hiệu {,tên_tín_hiệu}:kiểu_dữ_liệu [:=giá_trị_khởi_tạo];
```

Ví dụ: **Signal** a,b,c: Bit:= '1';

```
Signal y, reg: std_logic_vector(3 downto 0):="0000";
```

+ **Biến – Variable**: là đối tượng cục bộ được sử dụng để chứa các kết quả trung gian. Biến chỉ được khai báo và sử dụng trong process và trong chương trình con. Cú pháp khai báo của biến cũng tương tự như khai báo tín hiệu:

```
variable tên_biến {,tên_biến}: kiểu_dữ_liệu [:=giá_trị_khởi_tạo];
```

Ví dụ: **variable** x : Bit:= '1';

```
variable Q: std_logic_vector(3 downto 0);
```

Nếu không được khởi tạo giá trị ban đầu biến sẽ nhận giá trị khởi tạo ban đầu là giá trị thấp nhất trong các giá trị thuộc miền xác định của kiểu dữ liệu. Tín hiệu cũng có thể chứa dữ liệu nhưng chúng lại không được sử dụng vì những lý do sau:

- Việc sử dụng biến hiệu quả hơn vì giá trị của biến được gán ngay lập tức trong process khi tín hiệu chỉ được lập kế hoạch để thực hiện và chỉ được cập nhật toàn bộ sau khi kết thúc process.
- Biến chiếm ít bộ nhớ hơn trong khi tín hiệu cần nhiều thông tin để có thể lập kế hoạch thực hiện cũng như để chứa các thuộc tính của tín hiệu.
- Sử dụng tín hiệu yêu cầu có lệnh **wait** để thực hiện đồng bộ phép gán tín hiệu với phép lập thực hiện theo cách sử dụng quen thuộc.

+ **Hằng –constant**: là đối tượng hằng được gán cho các giá trị cụ thể của một kiểu khi được tạo ra và không đổi trong toàn bộ quá trình thực hiện. Hằng cũng có tính toàn cục giống như tín hiệu và có thể được khai báo trong package, entity, architecture, proceduce, process... Cú pháp khai báo hằng:

```
constant tên_hằng {,tên_hằng}: kiểu_dữ_liệu :=giá_trị_khởi_tạo;
```

Ví dụ: **constant** GND : std\_logic:= '0';

```
constant PI: real:=3.1414;
```



Tóm lại: Các đối tượng trong VHDL có mục đích sử dụng, phạm vi sử dụng khác nhau, nhưng chúng có cú pháp khai báo chung như sau:

```
Đối_tượng tên_đối_tượng : kiểu_dữ_liệu {:=giá_trị_khởi_tạo}
```

Các đối tượng khi khai báo phải được xác định kiểu dữ liệu tương ứng. VHDL định nghĩa nhiều kiểu dữ liệu khác nhau để phù hợp với việc mô tả, thiết kế, mô phỏng các hệ thống số khác nhau trong thực tế.

## 9.2.2 Kiểu dữ liệu trong VHDL

Trong VHDL có 4 dạng dữ liệu:

- Vô hướng : gồm các dữ liệu có giá trị đơn như bit, boolean, integer, real, physical, character, std\_logic và std\_ulogic, enumerated (kiểu liệt kê)...
- Kiểu ghép: các dữ liệu dưới dạng một nhóm các thành phần như mảng, bảng ghi (record). Bit\_logic\_vector, std\_logic\_vector và String đều là những dạng dữ liệu ghép đã được định nghĩa sẵn.
- 2-D Arrays: các dữ liệu có dạng mảng 2 chiều, được tạo nên từ 1 mảng của một mảng 1 chiều ( hay một bản ghi).
- VHDL Subtypes: dạng dữ liệu con do người dùng tự định nghĩa dựa trên những dạng có sẵn.

Các kiểu dữ liệu đã được định nghĩa trong gói Standard chứa trong thư viện chuẩn Standard Library của VHDL là: bit, boolean, integer, real, physical, character, std\_logic and std\_ulogic, Bit\_logic\_vector, std\_logic\_vector và String và một số kiểu dữ liệu con. Cú pháp chung định nghĩa kiểu dữ liệu như sau:

```
Type Tên_kiểu is giới_hạn_giá_trị_của_kiểu
```

### a. Kiểu vô hướng

- **Kiểu Bit** : Kiểu liệt kê với 2 giá trị '0' và '1'. Kiểu Bit đã được định nghĩa như sau: **Type**  
Bit **is** ('0', '1');

- **Kiểu Boolean**: Kiểu liệt kê với 2 giá trị false và true. Kiểu Boolean đã được định nghĩa như sau: **Type** Boolean **is** (false, true);

- **Kiểu Integer**: Kiểu số nguyên với những giá trị dương hoặc âm, độ lớn mặc định là 32 bit với giới hạn giá trị: từ -2147483647 đến +2147483647. Khi sử dụng có thể giới hạn miền xác định theo giới hạn giảm dần dùng từ khóa **downto** hoặc tăng dần dùng từ khóa **to**:

```
signal A : integer range 0 to 7; -- A số nguyên 3 bit
```

```
variable B : integer range 15 downto 0; -- B số nguyên 4 bit
```

```
signal B : integer range 15 downto -15; -- B số nguyên 5 bit
```

Các cách biểu diễn số nguyên dạng thập phân:

+ digit[digit]digit, ví dụ : 0, 1, 123\_456\_789 , -123\_5678...

+ digit(**E**)digit, ví dụ: 987E6 (=987.10<sup>6</sup>)...

Các cách biểu diễn dưới dạng cơ số xác định:

+ base#based\_integer#[exponent], ví dụ: 2#1100\_0100#, 16#C4#, 4#301#E1, (=196)

- **Kiểu Real:** Kiểu số thực có giới hạn từ -1.0E+38 đến 1.0E+38, khác với kiểu integer kiểu số thực khi sử dụng thường được định nghĩa thành kiểu dữ liệu riêng và có giới hạn miền xác định:

```
signal a: Real := -123E-4;
type CAPACITY is range -25.0 to 25.0 ;
signal Sig_1 : CAPACITY := 3.0 ;
type PROBABILITY is range 1.0 downto 0.0;
constant P : PROBABILITY := 0.5 ;
```

Các cách biểu diễn số thực:

+ Biểu diễn dưới dạng thập phân: integer[.integer][exponent], ví dụ: 0.0, 0.5, 1.1234\_5678, 12.4E-9...

+ Biểu diễn dưới dạng cơ số xác định:

base#based\_integer[.based\_integer ]#[exponent]

Ví dụ: 2#1.111\_1111\_111#E+11, 16#F.FF#E2 (=4095.0)

- **Kiểu Character:** Kiểu kiểu ký tự, liệt kê với miền xác định là tập hợp các ký tự ASCII. Biểu diễn của giá trị Character: 'A', 'a', '\*', ' ', NUL, ESC...

- **Kiểu Vật lý – Physical:** được sử dụng để biểu diễn các đại lượng vật lý như khoảng cách, điện trở, dòng điện, thời gian... Kiểu vật lý cung cấp đơn vị cơ bản và các đơn vị kế tiếp được định nghĩa theo đơn vị cơ bản, đơn vị nhỏ nhất có thể biểu diễn được là đơn vị cơ bản. Trong thực việc chuẩn Time (kiểu dữ liệu thời gian) là kiểu vật lý duy nhất đã được định nghĩa.

```
type Time is range <xác_định_giới_hạn>
```

**units**

```
fs; -- Đơn vị cơ bản
```

```
ps = 1000 fs;
```

```
ns = 1000 ps;
```

```
us = 1000 ns;
```

```
ms = 1000 us;
```

```
sec = 1000 ms;
```

```
min = 60 sec;
```

```
hr = 60 min;
```

**End Units;**

Ví dụ sử dụng:

```
constant Tpd : time := 3ns ;
```

...

```
Z <= A after Tpd ;
```

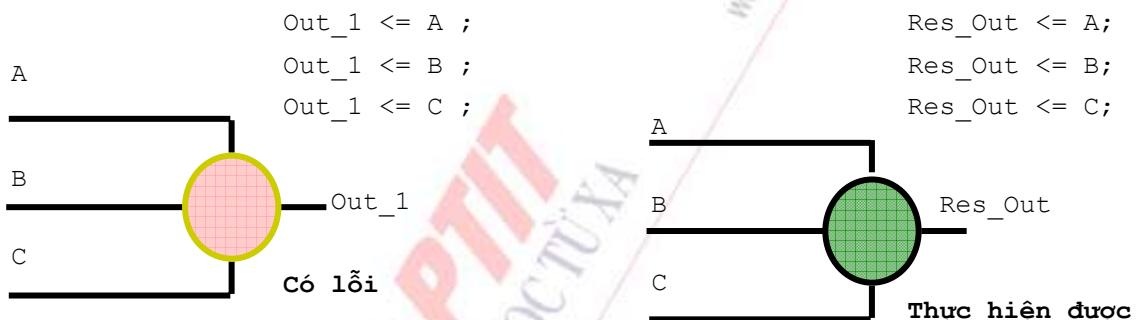
- **Kiểu std\_logic và std\_ulogic:** kiểu dữ liệu logic nhiều mức đã được định nghĩa trong gói std\_logic\_1164, so với kiểu Bit thì chúng có thể mô tả chính xác và chi tiết hơn cho các phần cứng số, chúng còn xác định được cường độ khác nhau của các tín hiệu.

<pre> <b>type</b> std_ulogic <b>is</b> ( 'U', -- Uninitialize   'X', -- Forcing Unknown   '0', -- Forcing Zero   '1', -- Forcing One   'Z', -- High Impedance   'W', -- Weak Unknown   'L', -- Weak Zero   'H', -- Weak One   '- ' -- Don't Care ) ;         </pre>	<pre> <b>type</b> std_logic <b>is</b> ( 'U', -- Uninitialize   'X', -- Forcing Unknown   '0', -- Forcing Zero   '1', -- Forcing One   'Z', -- High Impedance   'W', -- Weak Unknown   'L', -- Weak Zero   'H', -- Weak One   '- ' -- Don't Care ) ;         </pre>
---	--

Hai kiểu dữ liệu `std_logic` và `std_ulogic` tương tự nhau, chúng chỉ khác nhau ở chỗ là kiểu `std_ulogic` không có hàm phân dải (unresolved) – hàm quyết định giá trị tín hiệu, do đó sẽ có lỗi khi các tín hiệu kiểu `std_ulogic` được nối chung vào 1 điểm. Thư viện cũng cung cấp hàm phát hiện lỗi này của các tín hiệu kiểu `std_ulogic`.

```

signal A,B,C,Res_Out : std_logic ;
signal Out_1 : std_ulogic ;
    
```



(Ký hiệu “<=” dùng ở trên là lệnh gán tín hiệu, lệnh gán tín hiệu thực hiện được với 2 dữ liệu cùng kiểu, cùng độ lớn, giá trị của tín hiệu bên phải sẽ được gán cho tín hiệu bên trái).

- **Kiểu dữ liệu liệt kê tự định nghĩa:** Kiểu dữ liệu liệt kê, do người sử dụng tự định nghĩa, cho phép mô tả rất sáng sủa, và linh hoạt cho các mô hình phần cứng số với mức độ trừu tượng cao. Kiểu dữ liệu này dùng nhiều mô tả đồ hình trạng thái, các hệ thống phức tạp...

Ví dụ: `type My_State is (RST, LOAD, FETCH, STOR, SHIFT) ;`

```

. . .
signal STATE, NEXT_STATE : My_State ;
    
```

**b. Kiểu dữ liệu ghép**

Tương tự các ngôn ngữ lập trình, VHDL cũng có các kiểu dữ liệu ghép là nhóm các phần tử dữ liệu theo dạng mảng (array) hoặc bảng ghi (record).

**+ Mảng – Array:**

Mảng là nhóm nhiều phần tử có cùng kiểu dữ liệu với nhau thành đối tượng duy nhất. Mỗi phần tử của mảng có thể được truy cập bằng một hoặc nhiều chỉ số của mảng. Cú pháp định nghĩa kiểu dữ liệu mảng như sau:

**Type** tên\_mảng **is array** (khoảng\_của\_chi\_số) **of** kiểu\_của\_phần\_tử;

Ví dụ một số cách khai báo và sử dụng dữ liệu mảng:

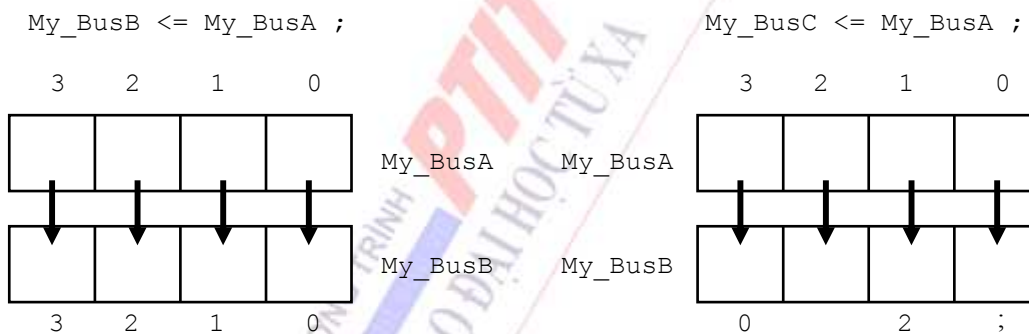
```
type WORD is array (3 downto 0) of std_logic ;
signal B_bus : WORD ;
type DATA is array (3 downto 0) of integer range 0 to 9 ;
signal C_bus : DATA ;
```

Các kiểu dữ liệu mảng đã được định nghĩa trong thư viện chuẩn của VHDL là: Bit\_logic\_vector (mảng dữ liệu kiểu Bit), std\_logic\_vector (mảng dữ liệu kiểu std\_logic) và String (mảng dữ liệu kiểu Character). Một số ví dụ sử dụng các kiểu dữ liệu này như sau:

```
signal My_BusA, My_BusB: bit_vector (3 downto 0);
signal My_BusC : bit_vector (0 to 3) ;
signal Data_Word : std_logic_vector (11 downto 0);
variable Warning2: string(1 to 30):= " Unstable, Aborting Now" ;
constant Warning3: string(1 to 20):= " Entering FSM State2";
```

**Một số phép toán thao tác với phần tử mảng:**

- **Phép gán cho mảng:** 2 mảng phải cùng kiểu, cùng độ lớn, phép gán sẽ thực hiện gán theo từng phần tử theo thứ tự từ trái sang phải:



```
Data_Word <= "101001101111" ;
Data_Word <= X"A6F";
Data_Word <= O"5157";
Data_Word <= B"1010_0110_1111" ;
```

Cách biểu diễn số liệu bit\_vector và std\_logic\_vector: B|O|X "giá trị" (dùng dấu nháy kép). Trong đó B : Binary -Kiểu nhị phân, O: Octal - kiểu bát phân, X: hexadecimal.

$$X"1AF"=B"0001_1010_1111"= B"000_110_101_111"=O"0657"$$

- **Phép gộp ( ):** cho phép nhóm cả dữ liệu vô hướng và dữ liệu mảng để thuận tiện cho các phép gán cho mảng:

```

signal H_BYTE, L_BYTE: std_logic_vector ( 0 to 7);
signal Q_Out : std_logic_vector (31 downto 0);
signal A, B, C, D : std_logic;
signal WORD : std_logic_vector (3 downto 0);
...
(A,B,C,D) <= WORD;

```

**Chú ý:** Phép gộp ở vế bên trái chỉ dùng với kiểu dữ liệu vô hướng.

```

WORD <= ( 2 => '1', 3 => D, others => '0' ) ;
Q_Out <= (others => '0' ) ;
WORD <= ( A, B, C, D ) ;
H_Byte <= (7|6|0=>'1', 2 to 5 => '0' );
L_Byte <= (3=>'1', 1 to 2 => '0', 4 to 7 => '1');

```

**Chú ý:** “others” có thể được sử dụng khi gán mặc định, nó có ý nghĩa là các tất cả các phần tử còn lại được gán bằng một giá trị nào đó).

**+ Bảng ghi – Record:**

Bảng ghi là nhóm nhiều phần tử có kiểu dữ liệu khác nhau thành đối tượng duy nhất.

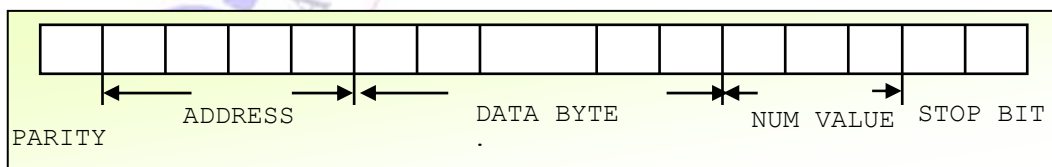
- Mỗi phần tử của bản ghi được truy nhập tới theo tên trường.
- Các phần tử của bản ghi có thể nhận mọi kiểu của ngôn ngữ VHDL kể cả mảng và bảng ghi.

Ví dụ định nghĩa kiểu dữ liệu bảng ghi như sau:

```

type OPCODE is record
    PARITY : bit;
    ADDRESS : std_logic_vector ( 0 to 3 );
    DATA_BYTE : std_logic_vector ( 7 downto 0 );
    NUM_VALUE : integer range 0 to 6;
    STOP_BITS : bit_vector (1 downto 0);
end record ;
...
signal TX_PACKET, RX_PACKET : OPCODE;

```



Cách truy nhập và gán dữ liệu cho các trường của bản ghi: Các phần tử của bản ghi được truy nhập theo tên bản ghi và tên trường, 2 thành phần này được ngăn cách bởi dấu ‘.’

```

TX_PACKET <= ( '1', "0011", "11101010", 5, "10" ) ;
TX_PACKET.ADDRESS <= ("0011");
TX_PACKET <= RX_PACKET;

```

```
TX_PACKET.ADDRESS <= RX_PACKET.ADDRESS;
```

### c. Kiểu dữ liệu mảng 2 chiều (2-D Array)

Mảng 2 chiều là kiểu dữ liệu mảng của các phần tử mạng một chiều hay bảng ghi. Một số ví dụ định nghĩa và khai báo kiểu dữ liệu mảng 2 chiều như sau:

```
type Mem_Array is array (0 to 3) of std_logic_vector (7 downto 0);
type Data_Array is array ( 0 to 2 ) of OPCODE ;
...
signal My_Mem:Mem_Array ;
signal My_Data:Data_Array ;
```

Ví dụ ứng dụng dùng mảng 2 chiều khởi tạo một vùng nhớ ROM

```
constant My_ROM : REM_Array := (0 => (others=>'1'),
                                1 => "10100010",
                                2 => "00001111",
                                3 => "11110000");
```

### d. Kiểu dữ liệu con:

Là một tập hợp con của các kiểu dữ liệu đã được định nghĩa khác. Phép khai báo kiểu dữ liệu con có thể nằm ở mọi vị trí cho phép khai báo kiểu dữ liệu. Cú pháp khai báo chung:

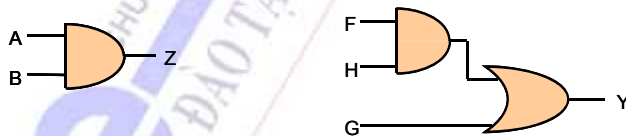
**Subtype** Tên\_kiểu\_dữ\_liệu\_con **is** xác\_định\_kiểu\_dữ\_liệu\_con;

Ví dụ:

```
subtype My_Int is integer range 0 to 255 ;
subtype My_Small_Int is My_Int range 5 to 30 ;
subtype word is bit_vector(31 downto 0);
```

## 9.2.3 Các phép toán trong VHDL

**Toán tử logic:** được sử dụng cho các dạng dữ liệu là bit, boolean, bit\_vector và std\_logic\_vector. Toán tử logic gồm có: **and**, **or**, **nand**, **nor**, **xor**, **not**, **xnor**.



Ví dụ: Z <= A **and** B;

Y <= G **or** (F **and** H);

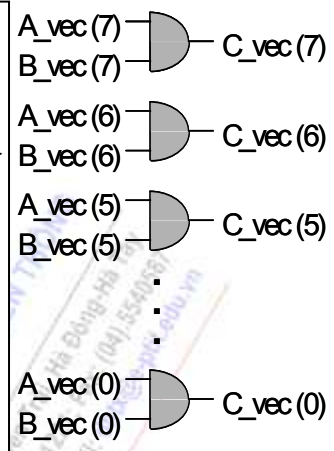
- Toán tử logic dùng cho kiểu dữ liệu mảng:

```

signal   A_vec, B_vec, C_vec :
            bit_vector(7 downto 0 ) ;
...
C_vec <= A_vec and B_vec ;
    
```

Nguyên tắc thực hiện phép logic với dữ liệu mảng:

- + Chỉ thực hiện với các mảng phải cùng kiểu, cùng độ lớn.
- + Phép toán logic thực hiện với từng phần tử của mảng và theo thứ tự từ trái sang phải.



**Toán tử quan hệ:** được sử dụng cho hầu hết các dạng dữ liệu, tất cả các toán tử quan hệ đều cho giá trị trả về dưới dạng boolean. Toán tử quan hệ gồm có: =, /=, <, <=, >, >=.

Ví dụ: 

```

signal   FLAG_BIT : boolean ;
signal   A, B : integer ;
FLAG_BIT <= ( A > B ) ;
    
```

- Nguyên tắc thực hiện phép quan hệ với dữ liệu mảng:

+ Các mảng phải cùng kiểu, độ dài có thể khác nhau.

+ Mảng có độ dài khác nhau thì phép quan hệ thực hiện ưu tiên phần tử từ trái sang phải và so sánh theo giá trị ASCII.

```

signal   A_vec : bit_vector ( 7 downto 0 ) := "11000110" ;
signal   B_vec : bit_vector ( 5 downto 0 ) := "111001" ;
...
if ( A_vec > B_vec ) then
    State <= Normal;
else
    State <= Code_Red;
end if;
    
```

**Toán tử số học:** được sử dụng cho số nguyên, số thực, và các dạng dữ liệu vật lý, std\_logic. Std\_logic\_vector, Bit, Bit\_vector. Cần chú ý rằng không phải tất cả toán tử số học đều có thể sử dụng cho mảng. Các toán tử số học là: +, -, \*, /, abs (trị tuyệt đối), \*\* (hàm mũ).

**Toán tử dịch:** mỗi toán tử tác động vào thành phần bên trái của một toán hạng hoặc toán hạng bên phải của số nguyên để tạo ra rất nhiều toán tử dịch và quay. Số âm chỉ ra cách hướng khác được sử dụng. Mỗi toán tử cho kết quả cùng dạng và kích thước với toán hạng ban đầu. Các toán tử dịch trong VHDL là: sll (dịch trái logic), srl (dịch phải logic), sla (dịch trái số học), sra (dịch phải số học), rol (quay trái), ror (quay phải).

Ví dụ: 

```

signal A_vec : bit_vector ( 7 downto 0 ) := "11000110";
    
```

```
signal D_vec : bit_vector (7 downto 0);
```

```
D_vec <= A_vec sll 2;
D_vec <= A_vec sra 2;
D_vec <= A_vec ror 3;
D_vec <= A_vec srl 2;
D_vec <= A_vec sra -2;
```



```
D_vec = "00011000"
D_vec = "11110001"
D_vec = "11011000"
D_vec = "00110001"
D_vec = "00011000"
```

**Toán tử ghép nối:** toán tử “&” cho phép ghép nối một cách linh hoạt các dữ liệu đơn và dữ liệu dạng mảng thành các mảng lớn hơn.

Ví dụ: signal A\_vector, B\_vector: std\_logic\_vector (7 downto 0);

```
signal Z_vector: std_logic_vector (15 downto 0);
```

```
Z_vector <= A_vector & B_vector;
```

**Toán tử tách:** cho phép ta lấy ra một số thành phần của mảng, chiều chỉ số của phép tách phải cùng chiều đánh chỉ số đã định nghĩa cho mảng.

Ví dụ: signal Z\_vec: std\_logic\_vector (15 downto 0);

```
signal B_vec: std_logic_vector (7 downto 0);
```

```
B_vec <= Z_vec (12 downto 5);
```

**Toán tử thuộc tính:** Xác định thuộc tính dữ liệu của đối tượng biến và tín hiệu. Cú pháp chung:

Đối\_tượng'thuộc\_tính

- Các thuộc tính được định nghĩa trước cho kiểu dữ liệu mảng trong VHDL là:

+ **left, right:** trả lại chỉ số của phần tử bên trái nhất hoặc bên phải nhất của dữ liệu mảng.

+ **high, low :** trả lại chỉ số của phần tử cao nhất hoặc thấp nhất của kiểu dữ liệu mảng.

+ **range, reverse\_range :** xác định khoảng của chỉ số của mảng.

+ **length :** trả về số lượng các phần tử của mảng.

+ **event, stable :** thuộc tính chỉ dùng cho đối tượng là tín hiệu, trả về giá trị boolean, chỉ ra rằng trên đường tín hiệu đang xét có xuất hiện sự kiện thay đổi hay giá trị trên đường tín hiệu ổn định tại thời điểm hiện tại. Các thuộc tính này dùng nhiều với lệnh **wait** và **if**. Ví dụ sử dụng toán tử thuộc tính như sau:

```
signal a : std_logic:= '0';
```

```
...
```

```
PROCESS(a)
```

```
TYPE bit4 IS ARRAY(0 TO 3) OF BIT;
```

```
TYPE bit_strange IS ARRAY(10 TO 20) OF BIT;
```

```
VARIABLE len1, len2 : INTEGER;
```

```
BEGIN
```

```
If (a'event and a='1') then -- sự kiện có xườn dương của a.
```

```
len1 := bit4'LENGTH; -- returns 4
```



```

len2 := bit_strange'LENGTH; -- returns 11
End if;
END PROCESS;

```

### 9.2.4 Các đơn vị thiết kế trong VHDL:

VHDL sử dụng 6 đơn vị thiết kế gồm 2 loại: đơn vị cơ bản và đơn vị thiết kế thứ cấp:

- Đơn vị thiết kế cơ bản:

- **Library:** Cho phép tạo thư viện trong VHDL
- **Package:** Tạo các gói giữ liệu trong Library, như các khai báo các đối tượng, khai báo chương trình con, hàm...
- **Entity:** (Thực thể) - cho phép khai báo các giao diện của một khối thiết kế số nào đó: như khai báo các chân vào/ra, các tham số của khối mạch...

- Đơn vị thiết kế thứ cấp (Phụ thuộc vào một đơn vị thiết kế cơ bản):

- **Architecture:** Mô tả hoạt động bên trong của một Entity hay đây chính là phần mô tả hoạt động của khối mạch số.
- **Package Body:** Mô tả chi tiết cho các khai báo trong Package như viết các hàm, các thủ tục ...
- **Configuration:** Đơn vị thiết kế cấu hình cho phép gắn các phiên bản của thực thể vào những kiến trúc khác nhau. Cấu hình cũng có thể được sử dụng để thay thế một cách nhanh chóng các phần tử của thực thể trong các biểu diễn cấu trúc của thiết kế.

+ **Entity - (Thực thể) :**

Khai báo thực thể trong VHDL phần định nghĩa các chỉ tiêu phía ngoài của một phần tử hay một hệ thống. Thực chất của việc khai báo thực thể chính là khai báo giao diện của hệ thống với bên ngoài. Ta có thể có tất cả các thông tin để kết nối mạch vào mạch khác hoạt động thiết kế tác nhân đầu vào phục vụ cho mục đích thử nghiệm. Tuy nhiên hoạt động thật sự của mạch không nằm ở phần khai báo này. Cú pháp khai báo chung của một Entity như sau:

```

entity Tên_thực_thể is
  generic(--Khai báo danh sách các tham số generic
    Tên_tham_số : [Kiểu_dạng_liệu] [:=giá_trị_khởi_tạo];
    ...
  );
  port(-- Khai báo danh sách cổng các port vào ra
    Tên_cổng : [mode] [Kiểu_dạng_liệu] [:=giá_trị_khởi_tạo];
    ...
  );
end Tên_thực_thể;

```

Trong khai báo trên:

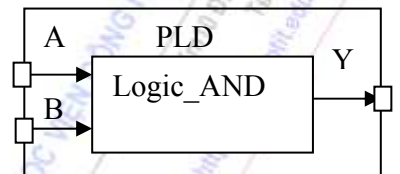
+ Tham số khai báo trong phần **generic** dùng để kiểm soát cấu trúc, hoạt động của thực thể, chúng sẽ được truyền giá trị hoặc lấy giá trị mặc định ban đầu khi thực thể được khởi tạo.

+ "--" Dấu đánh dấu dòng chú thích (comment) trong mã mô tả VHDL

+ [mode]: chỉ hướng tín hiệu của cổng có thể là: (**in**, **out**, **inout** hoặc **buffer**). Trong đó cổng dạng **in** chỉ dùng để đọc dữ liệu. Cổng dạng **out** chỉ dùng để gán giá trị dữ liệu. Cổng **inout** cho phép đồng thời vừa đọc vừa gán giá trị dữ liệu ở trong và ngoài chương trình. Cổng dạng **buffer** cho phép cả 2 thao tác đọc và gán dữ liệu từ bên trong chương trình, nhưng chỉ cho phép đọc dữ liệu từ ngoài chương trình.

Ví dụ khai báo thực thể cho một cổng logic AND:

```
entity Logic_AND is
    port(A, B : in std_logic ;
         Y   : out std_logic) ;
end Logic_AND;
```



+ **Architecture – (Kiến trúc) :**

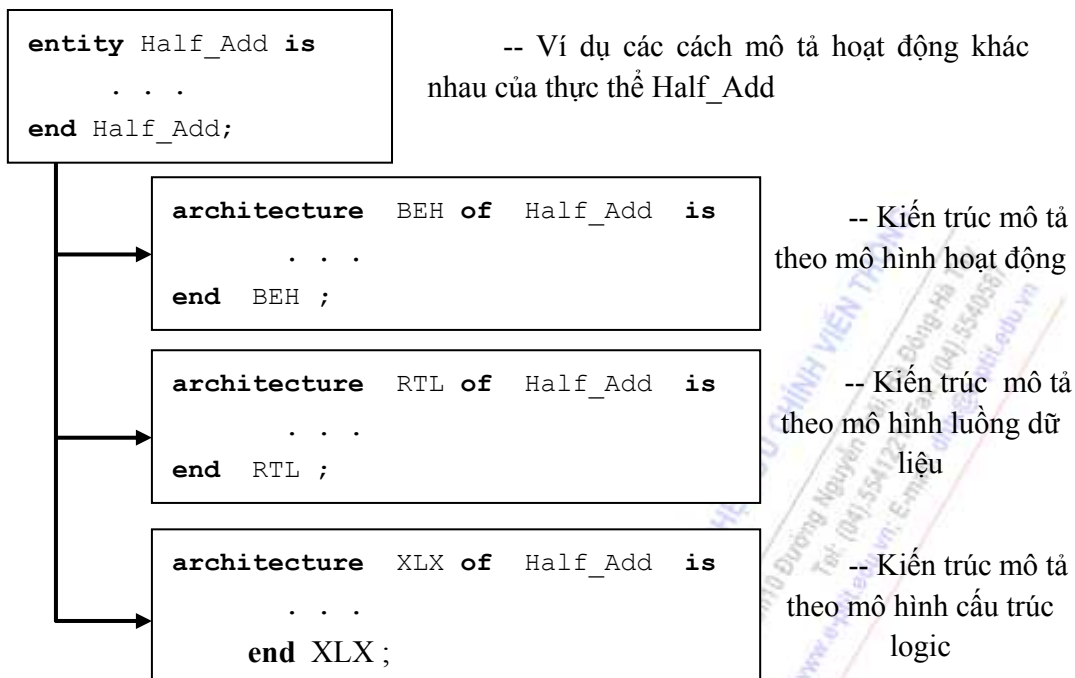
Cấu trúc này cho phép mô tả hoạt động bên trong của thực thể. Cú pháp chung của một **Architecture:**

```
Architecture Tên_kiến_trúc of Tên_thực_thể is
    -- Thực hiện các khai báo cho kiến trúc
    ...
Begin
    -- Viết các mô tả hoạt động bên trong cho thực thể
    ...
End Tên_kiến_trúc;
```

Phân khai báo kiến trúc có thể bao gồm các khai báo về các đối tượng signal, constant, kiểu dữ liệu, khai báo các phần tử bên trong hệ thống (component), hay các hàm (function) và thủ tục (procedure) sử dụng để mô tả hoạt động của hệ thống. Tên của kiến trúc là nhân được đặt tùy theo người sử dụng

VHDL cho phép tạo ra nhiều mô tả **Architecture** cho một thực thể, cho phép thực hiện nhiều cách mô tả hoạt khác nhau cho một thực thể. Mỗi cách mô tả hoạt động sẽ tối ưu về mặt thời gian thiết kế hay độ tin cậy hay tối ưu về tài nguyên sử dụng khi tổng hợp...

Có 3 cách chính mô tả kiến trúc của một phần tử (hoặc hệ thống số) đó là mô hình hoạt động (Behaviour), mô tả theo mô hình cấu trúc logic (Structure), và mô hình luồng dữ liệu. Tuy nhiên để mô tả cho một hệ thống, trong một kiến trúc có thể kết hợp sử dụng 2 hoặc cả 3 mô hình mô tả trên để thực hiện cho từng thành phần con tương ứng của hệ thống số. Trong phần sau của tài liệu này sẽ trình bày chi tiết hơn các phương pháp mô tả này.



+ **Package** và **Package Body**

**Package** ( gói dữ liệu) là đơn vị thiết kế cơ bản dùng để chứa những khai báo cho các đối tượng, khai báo chương trình con, hàm, kiểu dữ liệu, component có thể dùng chung cho những thiết kế, project, cấu trúc.

**Package Body** là đơn vị thiết kế phụ thuộc được dùng để chứa những mô tả chi tiết cho các khai báo trong đơn vị thiết kế **Package** nào đó, mô tả chi tiết nội dung của các hàm, các thủ tục ... **Package Body** thường được viết ngay sau **Package**. Cú pháp chung các đơn vị thiết kế **Package** và **Package Body** :



<pre> <b>package</b> My_Pack <b>is</b>  constant. . . . . . <b>function</b> bv_to_integer (BV: bit_v..                         <b>return</b> integer . . . component . . . . . . subtype. . .  <b>end package</b> My_pack;  <b>package body</b> My_Pack <b>is</b>      <b>function</b> bv_to_integer (BV: bit_v..                         <b>return</b> integer <b>is</b>     <b>variable</b> ...     <b>begin</b>         for index in BV'range loop             . . . .         <b>end function</b>; . . . <b>end</b> My_Pack ; </pre>	<pre> -- Cách sử dụng package trong file mô tả VHDL. <b>library</b> IEEE;-- Thu việc chuẩn <b>use</b> IEEE.std_logic_1164.all ; . . . -- Trong phần mềm thiết kế ISE gói dữ liệu do người sử dụng tạo ra thường được tổ chức mặc định trong thư viện "work" <b>use</b> work.My_Pack.all;  <b>entity</b> . . . </pre>
--	--

**+ Library (thư viện)**

Trong VHDL có các thư viện thiết kế chuẩn, ngoài ra người thiết kế có thể tạo các thư viện thiết kế riêng. Trong một thiết kế VHDL nhiều đoạn chương trình có thể được gọi từ các thư viện khác nhau.

Phân tích VHDL là một quá trình kiểm tra các đơn vị thiết kế VHDL để cho đúng cú pháp và ngữ nghĩa, các đơn vị thiết kế VHDL được lưu vào thư viện để sử dụng sau này. Thư viện thiết kế chứa các những phần tử thư viện sau:

- **Package:** chứa những mô tả khai báo được dùng chung.
- **Entity:** là những mô tả giao diện thiết kế được dùng chung.
- **Architecture:** những mô tả hoạt động thiết kế được dùng chung.
- **Configuration:** là những phiên bản của thực thể được dùng chung.

Các đơn vị thư viện là các cấu trúc VHDL có thể được phân tích riêng rẽ theo trình tự nhất định.

Trong VHDL có thư viện thiết kế đặc biệt có tên là "WORK". Khi người thiết kế biên dịch một chương trình viết trên VHDL nhưng không chỉ rõ thư viện đích, chương trình này sẽ được biên dịch và chứa vào thư viện "WORK".

Ví dụ cách gọi và sử dụng thư viện như sau:

```

library My_Lib ;
use My_Lib.Fast_Counters.all ;

entity Mod1 is
  port ( . . .

```

#### + Configuration (Cấu hình)

Một thực thể có thể có một vài kiến trúc mô tả hoạt động cho nó. Trong quá trình thiết kế có thể phải thử nghiệm một vài biến thể của thiết kế bằng cách sử dụng các kiến trúc khác nhau. Cấu hình là thành phần cơ bản của đơn vị thiết kế. Cấu hình cho phép gắn các phiên bản của thực thể vào những kiến trúc khác nhau. Cấu hình cũng có thể được sử dụng để thay thế một cách nhanh chóng các phần tử của thực thể trong các biểu diễn cấu trúc của thiết kế.

Cú pháp của mô tả cấu hình như sau:

```

Configuration tên_cấu_hình of tên_thực_thể is
-- Phần khai báo của cấu hình (cho phép sử dụng
-- các phần tử trong package và library.
for đặc_tả_của_khối
  {mệnh_đề_use}
  {các_phần_tử_của_cấu_hình}
end for;

```

Ví dụ:

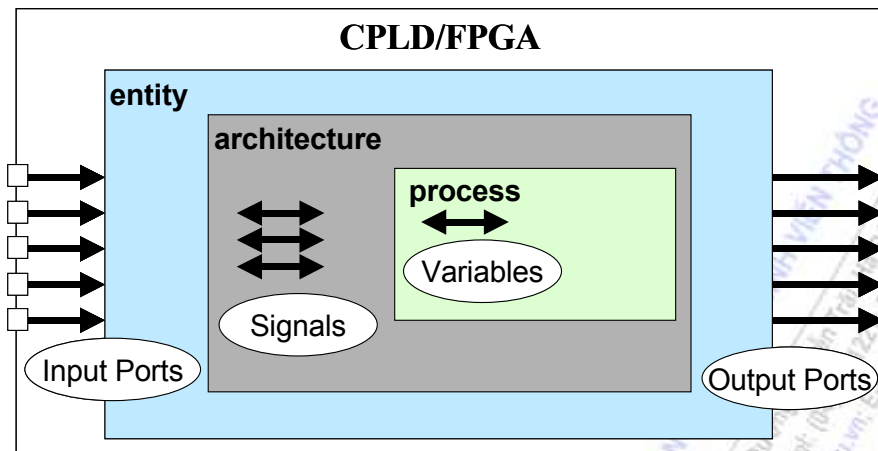
```

library ttl, work;
configuration v4_27_87 of processor is
  use work.all;
for structure_view
for al:alu
use configuration ttl.sn74ls181;
end for;
for m1,m2,m3: mux
use entity multiplex4 (behavior);
end for;
for all: latch -- use defaults
end for;
end for;
end configuration v4_27_87;

```

#### 9.2.5 Cấu trúc chung của một chương trình mô tả VHDL

Mô hình cấu trúc mô tả phần cứng số và phạm vi sử dụng của các đối tượng trong VHDL có thể được tổng kết đơn giản như trong hình 9-1 dưới đây:



Hình 9-1. Cấu trúc mô tả phần cứng và các đối tượng trong VHDL.

Sau đây là cấu trúc chung đơn giản của một chương trình mô tả VHDL:

```
-- Ví dụ cấu trúc 1 file mô tả cho một hệ thống phần cứng số dùng VHDL
-- Khai báo thư viện, (mặc định cần khai báo thư viện IEEE (thư viện
-- chuẩn đã được xây dựng) .
library IEEE;...
-- Khai báo gói dữ liệu (package) trong thư viện cần sử dụng:
use IEEE.STD_LOGIC_1164.ALL;...
-- Khai báo thực thể
Entity Tên_thực_thể is
    -- Khai báo các tham số generic nếu cần:
    Generic( -- khai báo danh sách các tham số);
    Port(-- Khai báo danh sách các cổng vào/ra
        );
End Tên_thực_thể;
-- Bắt đầu viết
Architecture Tên_kiến_trúc of Tên_thực_thể is
{Khai báo:kiểu dữ liệu, các component,các đối tượng constant, signal}
Begin
    { Viết các mô tả dùng cấu trúc lệnh song song }
    ...
    Process(-- danh sách tín hiệu kích thích nếu cần)
        {Khai báo:kiểu dữ liệu, các đối tượng biến constant, variable }
    Begin
        { Viết các mô tả dùng cấu trúc lệnh tuần tự }
    End process;
```

```

...
{ Viết các mô tả dùng cấu trúc lệnh song song hay process khác }
...
End Tên_kiến_trúc;

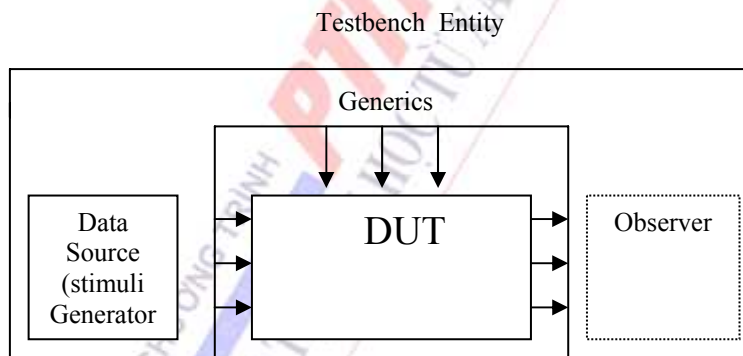
```

### 9.2.6 Môi trường kiểm tra testbench

Một trong các nhiệm vụ rất quan trọng là kiểm tra bản mô tả thiết kế. Kiểm tra một mô hình VHDL được thực hiện bằng cách quan sát hoạt động của nó trong khi mô phỏng và các giá trị thu được có thể đem so sánh với yêu cầu thiết kế.

Môi trường kiểm tra có thể hiểu như một mạch kiểm tra ảo. Môi trường kiểm tra sinh ra các tác động lên bản thiết kế và cho phép quan sát hoặc so sánh kết quả hoạt động của bản mô tả thiết kế. Thông thường thì các bản mô tả đều cung cấp chương trình thử. Nhưng ta cũng có thể tự xây dựng chương trình thử (testbench). Mạch thử thực chất là sự kết hợp của tổng hợp nhiều thành phần. Nó gồm ba thành phần. Mô hình VHDL đã qua kiểm tra, nguồn dữ liệu và bộ quan sát. Hoạt động của mô hình VHDL được kích thích bởi các nguồn dữ liệu và kiểm tra tính đúng đắn thông qua bộ quan sát. Hình 9-2 là sơ đồ tổng quát của một chương trình thử (Testbench).

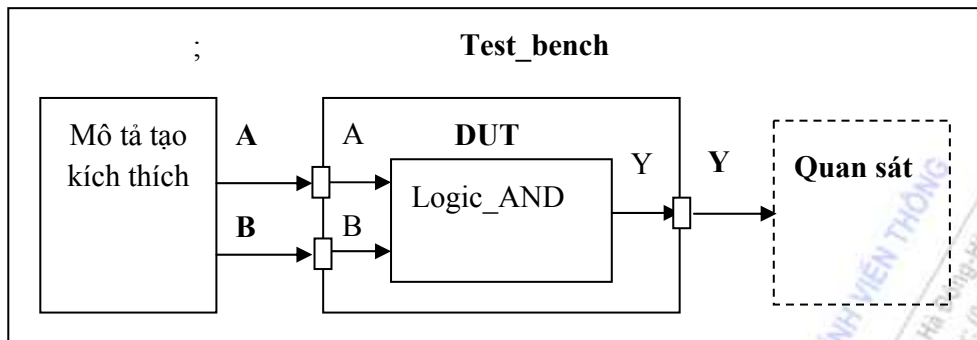
Testbench được mô tả như một Entity không có đầu vào đầu ra, chỉ có tín hiệu bên trong được ghép tới khối DUT cần được kiểm tra **theo kiểu cấu trúc**. Người thiết kế sẽ mô tả các tín hiệu bên trong này tạo ra tín hiệu kích thích cho các đầu vào của DUT và đọc kết quả ra để quan sát...



Trong đó: DUT: (device under test) mô hình VHDL cần kiểm tra  
 Observer: khối quan sát kết quả  
 Data source: nguồn dữ liệu (khối tạo ra các tín hiệu kích thích)

**Hình 9.2. Sơ đồ tổng quát chương trình thử Testbench**

Viết Testbench cho thực thể Logic\_AND đã mô tả ở phần trước:



Thực thể **Test\_bench** không có các cổng vào ra mà chỉ khai báo các tín hiệu nội bộ A, B, Y để nối tới khối DUT cần kiểm tra (Logic\_AND). Trong phần kiến trúc mô tả hoạt động của Test\_bench, coi khối Logic\_AND như một component để tạo thành khối Test\_bench. Toàn bộ mã mô tả cho Test\_bench như sau:

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.numeric_std.ALL;
-- Khai báo thực thể Test_bench;
ENTITY Test_bench IS
END Test_bench;
-- Mô tả kiến trúc của Test_bench
ARCHITECTURE behavior OF Test_bench IS
    COMPONENT Logic_AND
    PORT (
        A : IN std_logic;
        B : IN std_logic;
        Y : OUT std_logic
    );
    END COMPONENT;
    SIGNAL A : std_logic := '0';
    SIGNAL B : std_logic := '0';
    SIGNAL Y : std_logic;
BEGIN
-- Nối chân cổng vào ra của DUT với các tín hiệu của Test_bench
    uut: Logic_AND PORT MAP (
        a => a,
        b => b,
        y => y
    );
    tb : PROCESS
    BEGIN
        -- Viết mô tả tạo kích thích
```



```

A<= '1' after 10ns;
B<= '1' after 20ns;
...
END PROCESS;
-- *** End Test Bench - User Defined Section ***
END;
```

Trong các phần mềm thiết kế sau khi hoàn thành các mô tả cho Test\_bench, người thiết kế sẽ chạy công cụ mô phỏng, các tín hiệu đầu ra của DUT sẽ được mặc tính đọc ra và cho phép người thiết kế quan sát dễ dàng dưới dạng giản đồ thời gian, hay các file số liệu...

Người thiết kế có thể dễ dàng viết các mô tả kích thích để tạo ra các yêu cầu kiểm tra tùy ý cho bản thiết kế của mình. Nhiều chức năng mô phỏng, kiểm tra được hỗ trợ rất mạnh bởi các phần mềm thiết kế.

### 9.2.7 Các cấu trúc lệnh song song

Như đã trình bày trong phần cấu trúc chung của chương trình mô tả VHDL, trong mô tả một kiến trúc (Architecture) có chứa nhiều cấu trúc lệnh song song. Mỗi cấu trúc lệnh song song sẽ tương ứng với một thành phần phần cứng nào đó khi thực hiện tổng hợp mạch, mỗi cấu trúc song song có thể viết ở bất kỳ vị trí nào trong đoạn mô tả Architecture mà chức năng hoạt động của thực thể không thay đổi. Các cấu trúc lệnh song song có trong VHDL gồm:

- + Cấu trúc process.
- + Lệnh gán tín hiệu song song.
- + Lệnh gán có điều kiện.
- + Lệnh gán tín hiệu có lựa chọn.
- + Khối.
- + Phép gọi chương trình con song song.

#### a. Cấu trúc Process:

Cấu trúc Process được tạo thành từ một tập hợp cấu trúc lệnh tuần tự (được trình bày chi tiết ở phần sau). Nó là khối cơ bản của việc mô tả hoạt động của thực thể. Tất cả các

```

[Nhãn] Process [ (Danh sách tín hiệu kích thích) ]
[ Khai báo:kiểu dữ liệu, các đối tượng biến constant, variable ]
Begin
    { Viết các mô tả dùng cấu trúc lệnh tuần tự }
End process;
```

Process trong một thiết kế được thực hiện song song. Tuy nhiên, tại một thời điểm xác định chỉ có một câu lệnh tuần tự được thực hiện trong mỗi cấu trúc Process. Cấu trúc tổng quát:

Trong đó các phần đặt trong dấu [ ] có thể có hoặc không.

- Nhãn\_lệnh: Tùy thuộc người thiết kế đặt tên.
- Danh sách tín hiệu kích thích: Danh sách các yếu tố kích thích hoạt động

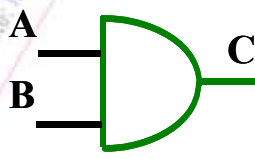
Nếu Process chứa (danh sách tín hiệu kích thích) thì lúc đó Process sẽ được thực hiện khi có bất kỳ sự thay đổi nào của bất kỳ tín hiệu nào trong danh sách tín hiệu kích thích. Điều này tương đương với Process không chứa danh sách tín hiệu kích thích nhưng lại chứa lệnh **wait** ở vị trí câu lệnh cuối cùng trong quá trình:

**Wait on** <danh sách tín hiệu kích thích>

Khi tổng hợp mạch thì mỗi Process sẽ tương ứng với một khối mạch chức năng nào đó. Còn khi thực hiện mô phỏng, việc thực hiện một Process bao gồm việc thực hiện lặp lại các cấu trúc lệnh tuần tự chứa bên trong thân của Process. Giống như một vòng lặp vô hạn và mỗi bước lặp được thực hiện mỗi khi có sự thay đổi của bất kỳ tín hiệu nào trong danh sách tín hiệu kích thích.

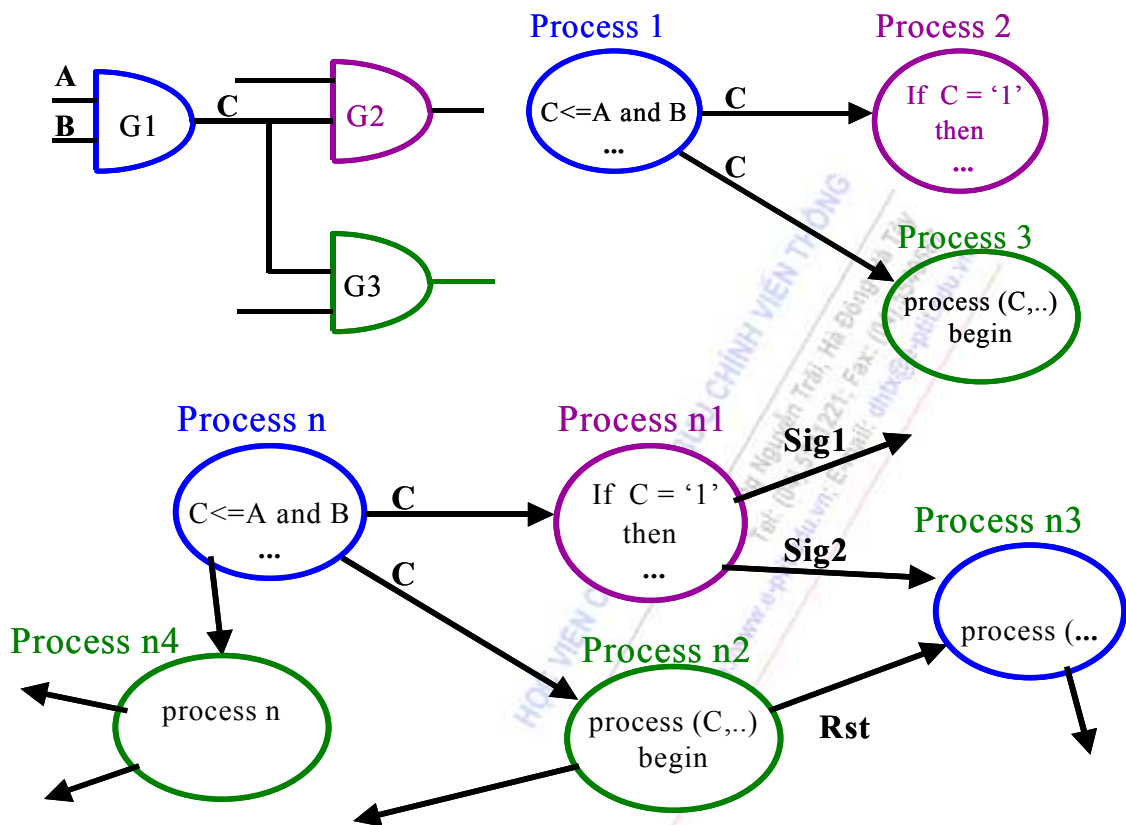
Ví dụ Process mô tả mạch logic AND như sau:

```
entity Logic_AND is
  Port ( A,B : in std_logic;
         C   : out std_logic);
end Logic_AND;
architecture Behavioral of Logic_AND is
begin
  Process (A,B)
  begin
    C<= A and B;
  end Process;
end Behavioral;
```



Một Process liên kết với phần còn lại của thiết kế thông qua các thao tác đọc các giá trị từ các tín hiệu vào, các cổng được khai báo ngoài Process và ghi giá trị vào các tín hiệu ra, cổng ra. Chú ý khi thiết kế là một tín hiệu có thể vào (được đọc ra) bởi nhiều Process nhưng chỉ nên được ghi ra bởi một Process.

Sự hoạt động đồng thời của mỗi Process và mô hình kết nối của chúng được mô tả như hình vẽ 9-3, trong đó tín hiệu sẽ truyền giá trị giữa những Process hoạt động đồng thời.:



Hình 9-3. Mô hình kết nối của các Process.

**b. Các phép gán tín hiệu song song**

Phép gán tín hiệu song song sử dụng bên trong các Architecture nhưng bên ngoài Process. Dạng đơn giản nhất của phép gán tín hiệu song song có cú pháp như sau:

<tín\_hiệu\_đích> <=> <biểu\_thức> [after <biểu\_thức\_thời\_gian>];

Trong đó <tín\_hiệu\_đích> nhận giá trị của <biểu\_thức>, chú ý là lệnh **after** chỉ dùng cho mô phỏng còn khi tổng hợp mạch nó sẽ được bỏ qua. Phép gán song song tương đương một Process chứa 1 phép gán tín hiệu.

Ví dụ mô tả mạch AND và OR có cùng 4 đầu vào như sau:

```

...
architecture Behavioral of logic1 is
  signal I1, I2, I3, I4, AND_out, OR_out: std_logic;
begin
  ...
  AND_out<= I1 and I2 and I3 and I4;
  OR_out<= I1 or I2 or I3 or I4;
  ...
end Behavioral;

```

Đoạn chương trình trên tương đương với đoạn chương trình VHDL với các Process chứa các phép gán tín hiệu tuần tự sau:

```
...
architecture Behavioral of logic1 is
    signal I1, I2, I3, I4, AND_out, OR_out: std_logic;
begin
    ...
    process(I1, I2, I3, I4)
    begin
        AND_out<= I1 and I2 and I3 and I4;
    end process;
    process(I1, I2, I3, I4)
    begin
        OR_out<= I1 or I2 or I3 or I4;
    end process;
    ...
end Behavioral;
```

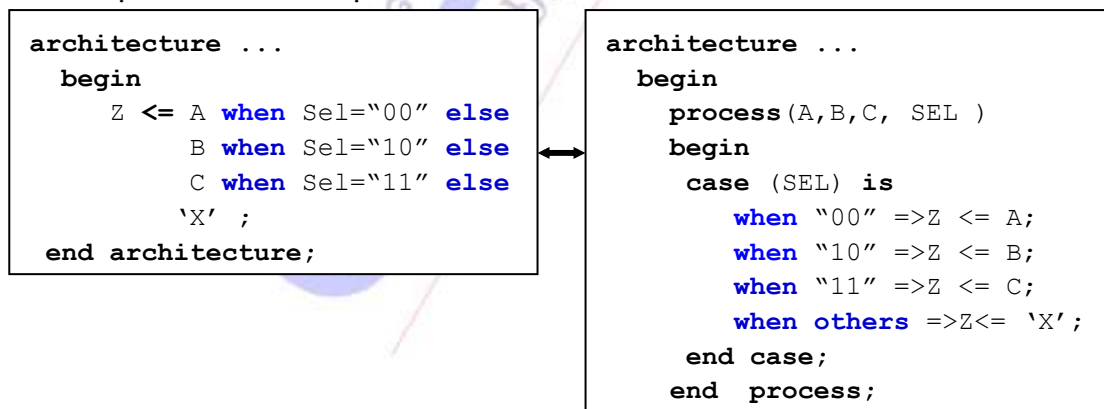
### c. Phép gán tín hiệu có điều kiện

Phép gán tín hiệu có điều kiện là cấu trúc lệnh song song thực hiện phép gán giá trị của các biểu thức cho một tín hiệu đích tùy theo các điều kiện đặt ra. Cú pháp chung:

```
<tín_hiệu_đích> <=> <biểu_thức>[after <biểu_thức_thời_gian>] when <điều_kiện> else
    <biểu_thức>[after <biểu_thức_thời_gian>] when <điều_kiện> else
    ...
    <biểu_thức>[after <biểu_thức_thời_gian>];
```

Cấu trúc phép gán tín hiệu có điều kiện có thể coi là cấu trúc song song của lệnh tuần tự **If** được thay thế tương đương với **Process** chứa lệnh tuần tự **if**.

Ví dụ mô tả cấu trúc chọn kênh như sau:

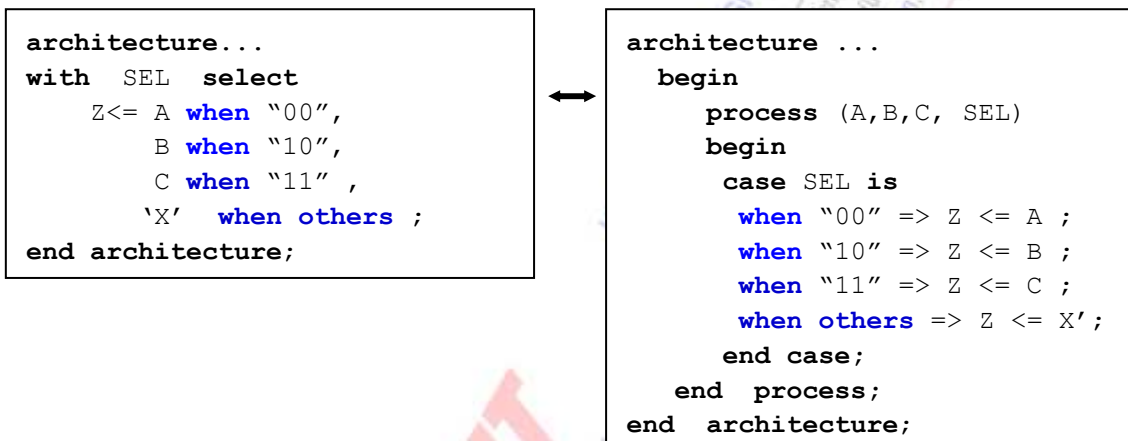


### d. Phép gán tín hiệu theo lựa chọn

Phép gán tín hiệu theo lựa chọn thực hiện gán cho một tín hiệu đích với biểu thức **with**. Cấu trúc này có thể coi như là cấu trúc song song của lệnh tuần tự **case**, nó có thể thay thế tương đương với Process chứa lệnh tuần tự **case**. Cú pháp chung của lệnh **with** như sau:

```
With <biểu_thức_lựa_chọn> select
    <tín_hiệu_đích> <=> <biểu_thức> [after <biểu_thức_thời_gian>]
                                when <giá_trị_lựa_chọn>,
    <biểu_thức> [after <biểu_thức_thời_gian>]
                                when <giá_trị_lựa_chọn>,
    ...
    <biểu_thức> [after <biểu_thức_thời_gian>]
                                when others;
```

Ví dụ mô tả cấu trúc chọn kênh như sau:



### e. Khối (Block)

Block bao gồm tập hợp các cấu trúc lệnh song song. Một kiến trúc có thể phân tách thành một số cá cấu trúc logic. Mỗi khối biểu diễn một thành phần của mô hình và thường được sử dụng để tổ chức một tập hợp các cấu trúc song song phân cấp. Cú pháp chung:

```
<nhãn>: Block
    {<phần_khai_báo>}
begin
    {<câu_lệnh_song_song>} - có trình tự bất kỳ
end block;
```

<phần\_khai\_báo> : xác định các đối tượng tồn tại cục bộ trong khối và có thể là các khai báo sau:

- Khai báo hằng, kiểu dữ liệu, tín hiệu.
- Thân chương trình con.
- Khai báo bí danh.
- Khai báo component.
- Luật use.

**f. Gọi chương trình con song song**

Phép gọi chương trình con song song tương đương với các process bao gồm các phép gọi chương trình con tuần tự tương ứng. Mỗi phép gọi chương trình con tương đương với một process không chứa dãy danh sách các tín hiệu kích thích, phần khai báo cổng và phần thân chứa một phép gọi chương trình con, tiếp theo là một câu lệnh **wait**.

**9.2.8 Cấu trúc lệnh tuần tự**

Trong ngôn ngữ VHDL một cấu trúc đồng thời quan trọng là **Process**. Cấu trúc này được sử dụng để mô tả hành vi hoạt động của mạch số. Trong kiến trúc, tất cả các Process sẽ được tổng hợp thành một khối mạch chức năng và thực hiện đồng thời khi mô phỏng. Một Process n được xây dựng từ các cấu trúc lệnh tuần tự. Khi mô phỏng các lệnh tuần tự được thực hiện lần lượt trong một chu trình vô hạn bắt đầu từ lệnh thứ nhất đến lệnh cuối và được kích hoạt trở lại thực hiện lệnh đầu mỗi khi có bất kỳ sự thay đổi nào trong danh sách tín hiệu kích thích hay trong danh sách tín hiệu trong câu lệnh **wait**.

Các cấu trúc lệnh tuần tự cơ bản trong VHDL gồm:

- Câu lệnh gán cho biến.
- Câu lệnh gán cho tín hiệu.
- Câu lệnh **if**.
- Câu lệnh **case**.
- Câu lệnh cổng **Null**.
- Các lệnh lặp.

**a. Phép gán biến**

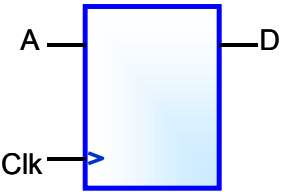
Cú pháp của phép gán biến như sau:

biến := biểu\_thức

Phép gán biến được thực hiện với thời gian mô phỏng bằng 0, và giá trị biến sẽ được cập nhật ngay giá trị của biểu thức. Đối tượng <biến> chỉ được khai báo và sử dụng trong Process và chương trình con, nó được sử dụng để lưu trữ các kết quả trung gian. Ví dụ:

```

...
process ( Clk )
  variable B, C, D : bit := '1' ;
  begin
    if (Clk'event and Clk = '1') then
      B := A ;
      C := B ;
      D := C ;
    end if ;
  end process ;...
    
```



```

...
process( Clk )
  variable B, C, D : bit := '1';
  begin
    if ( Clk'event and Clk = '1' ) then
      D := C;
      C := B;
      B := A;
    end if;
  end process ;

```

Chú ý trong 2 ví dụ trên, giá trị các biến được cập nhập tức thì, ví dụ thứ nhất khi tổng hợp mạch chỉ tạo ra một trigger D. Còn với ví dụ thứ 2 thứ tự gán biến thay đổi, kết quả tạo ra 3 trigger D. Trong ví dụ 1, nếu B,C,D là tín hiệu thì kết quả hoàn toàn khác. Xem ví dụ ở phần phép gán tín hiệu.

### b. Phép gán tín hiệu

Cú pháp của phép gán tín hiệu như sau:

Tín\_hiệu\_đích <= biểu\_thức [after giá\_trị\_thời\_gian];

Khác với phép gán biến, phép gán tín hiệu trong **Process** không được cập nhập ngay tức thì mà phép gán đó chỉ được đặt kế hoạch thực hiện và kết quả chỉ được cập nhập sau khi kết thúc **Process**.

Ví dụ:

```

Architecture Behavior of Triger is
  signal Clk, A, B, C, D : bit := '1';
Begin
  process( Clk )
  begin
    if ( Clk'event and Clk = '1' ) then
      B <= A ;
      C <= B ;
      D <= C ;
    end if ;
  end process ;
End Behavior

```

### c. Lệnh if

Lệnh này cho phép các phép toán được thực hiện trên một điều kiện nào đó. Có ba dạng cơ bản là:

+ Dạng 1:

```

if (Điều_kiện) then
  <Các_câu_lệnh_tuần_tự>;
end if;

```

+ Dạng 2:

```

if (Điều_kiện) then
    <Các_câu_lệnh_tuần_tự>;
else
    <Các_câu_lệnh_tuần_tự>;
end if;
+ Dạng 3:
if (Điều_kiện_1) then
    <Các_câu_lệnh_tuần_tự>;
elsif (Điều_kiện_2) then
    <Các_câu_lệnh_tuần_tự>;
elsif (Điều_kiện_3) then
    <Các_câu_lệnh_tuần_tự>;
else
    <Các_câu_lệnh_tuần_tự>;
end if;
    
```

Trong lệnh **if/else**, ta phải chú ý một số điều sau:

- +) Điều kiện đúng đầu tiên được tìm thấy sẽ được thực hiện.
- +) Các điều kiện có thể chồng lấp lên nhau.
- +) Điều kiện đầu tiên trong lệnh **if/else** được ưu tiên.

Ví dụ:

```

process (A, B, C, D, Sel)
begin
if (Sel = "00") then
    Z <= A ;
elsif (Sel = "01") then
    Z <= B ;
elsif (Sel = "10") then
    Z <= C ;
elsif (Sel = "11") then
    Z <= D ;
end if;
end process ;
    
```

#### d. Lệnh case:

Lệnh **case** được sử dụng trong trường hợp có một biểu thức để kiểm soát nhiều rẽ nhánh trong chương trình VHDL. Các lệnh tương ứng với một trong các lựa chọn sẽ được thực hiện nếu biểu thức kiểm soát có giá trị bằng giá trị tương ứng của lựa chọn đó. Có hai dạng cơ bản:

Dạng 1:

```

Case (biểu_thức_kiểm_soát) is
    When <giá_trị_lựa_chọn> =>
        <Các_câu_lệnh_tuần_tự>;
    When <giá_trị_lựa_chọn> =>
    
```



<Các\_câu\_lệnh\_tuần\_tự>;

...

**end case;**

Dạng 2:

**Case** (selector expression) **is**

**When** <giá\_trị\_lựa\_chọn> =>

<Các\_câu\_lệnh\_tuần\_tự>;

**When** <giá\_trị\_lựa\_chọn> =>

<Các\_câu\_lệnh\_tuần\_tự>;

...

**When others** =>

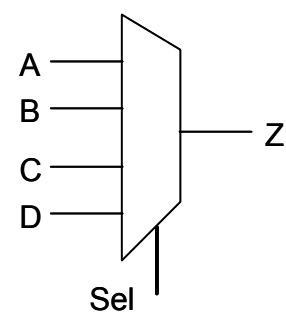
<Các\_câu\_lệnh\_tuần\_tự>;

**end case;**

Các chú ý khi dùng lệnh case:

- + ) Tất cả các giá trị của biểu thức lựa chọn phải được chỉ rõ.
- + ) Không có các giá trị lựa chọn bị chồng lấp lên nhau.

Ví dụ:

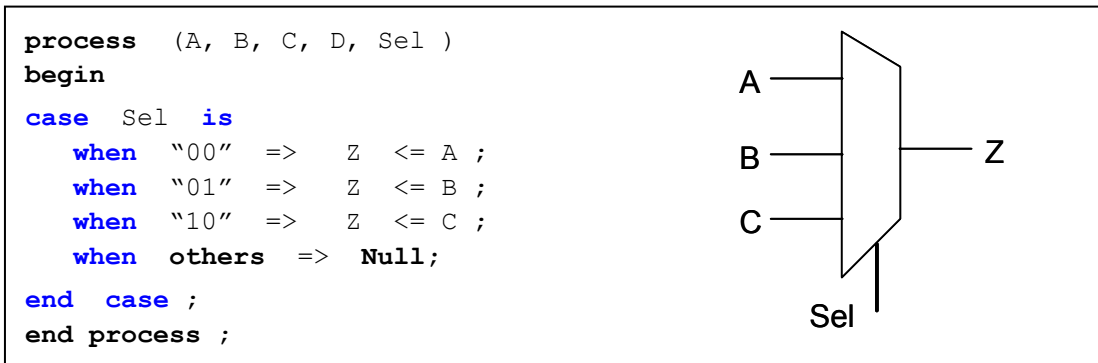
<pre> <b>process</b> (A, B, C, D, Sel ) <b>begin</b>   <b>case</b> Sel <b>is</b>     <b>when</b> "00" =&gt; Z &lt;= A ;     <b>when</b> "01" =&gt; Z &lt;= B ;     <b>when</b> "10" =&gt; Z &lt;= C ;     <b>when</b> "11" =&gt; Z &lt;= D ;   <b>end case</b> ; <b>end process</b> ;         </pre>	
--	---

### e. Câu lệnh rỗng Null

Câu lệnh rỗng có cú pháp như sau:

**Null;**

Trong VHDL khi chương trình mô phỏng gặp câu lệnh **Null** nó sẽ bỏ qua lệnh này và thực hiện lệnh tiếp theo sau. Thông thường lệnh **Null** dùng để chỉ trường hợp không thực hiện của lệnh một cách tường minh khi có các điều kiện trả lại giá trị true. Do đó lệnh Null thường được dùng trong các câu lệnh **case** đối với những giá trị lựa chọn không cần thao tác. Ví dụ:



### f. Các lệnh lặp

Lệnh lặp **loop** chứa thân vòng lặp bao gồm dãy các câu lệnh sẽ được thực hiện không hoặc nhiều lần. Cú pháp của lệnh lặp như sau:

```

[<nhãn>:] [<sơ_đồ_lặp>] loop
  {<lệnh_tuần_tự>}|
  {next [<nhãn>] [when <điều_kiện>];}|
  {exit [<nhãn>] [when <điều_kiện>];}
end loop [<nhãn>];
    
```

- <nhãn>: nhãn của vòng lặp và thường được dùng để xây dựng những vòng lặp lồng nhau, trong đó mỗi vòng lặp được kết thúc bởi từ khóa **end loop**.

- <sơ\_đồ\_lặp>: vòng lặp với sơ đồ lặp **for** hoặc vòng lặp **while**, và vòng lặp không chứa các sơ đồ lặp.

Với những vòng lặp không chứa [<sơ\_đồ\_lặp>], các lệnh trong dãy lệnh tuần tự sẽ được thực hiện cho tới khi được ngắt bởi câu lệnh **exit**. Trong đó câu lệnh **next** cũng được dùng để thay đổi trình tự thực hiện thân của vòng lặp.

Ví dụ vòng lặp không chứa sơ đồ lặp:

```

Count_down: Process
  Variable Min,Sec: integer range 0 to 60;
Begin
  L1: loop
    L2: loop
      exit L2 when (Sec=0);
      wait until CLK'event and CLK='1';
      Sec:=Sec-1;
    End loop L2;
    Exit L1 when (Min=0);
    Min:=Min-1;
    Sec:=60;
  End loop L1;
End process Count_down;
    
```

Ví dụ vòng lặp chứa <so\_đồ\_lặp> dạng **for**:

```

process ( A, B_bus )
begin
  for i in 7 downto 0 loop
    C_bus (i) <= A and B_bus (i);
  end loop ;
end process;

hoặc:

process ( A, B_bus )
begin
  for i in 0 to 7 loop
    C_bus (i) <= A and B_bus (i);
  end loop ;
end process;
        
```

Ví dụ vòng lặp chứa <so\_đồ\_lặp> dạng **while** như sau:

```

process ( A, B_bus )
  variable i:integer:=0;
begin
  while (i<8) loop
    C_bus (i) <= A and B_bus (i);
    i:=i+1;
  end loop ;
end process;
        
```

### 9.3. CÁC MỨC ĐỘ TRỪU TƯỢNG VÀ PHƯƠNG PHÁP MÔ TẢ HỆ THỐNG PHẦN CỨNG SỐ

Sử dụng VHDL cho phép mô tả hệ thống phần cứng số theo các mức độ trừu tượng khác nhau. Hình vẽ 9-4 mô tả các mức độ mô tả trừu tượng giảm dần khi sử dụng VHDL.

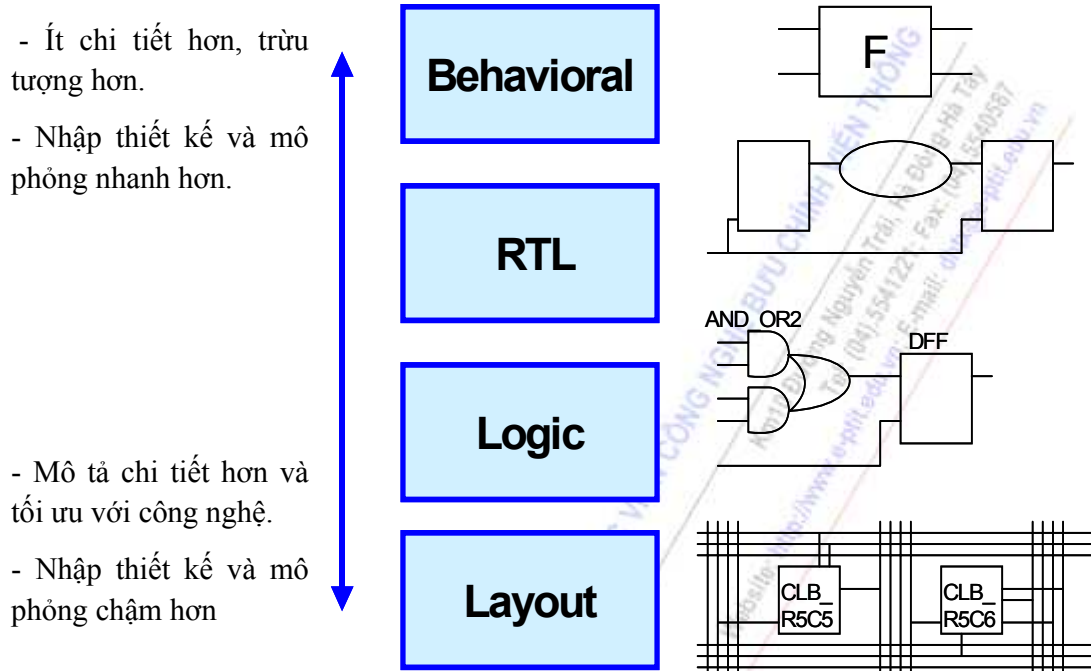
+ Mức mô tả theo mô hình hành vi (**Behavioral**): mức độ mô tả trừu tượng cao nhất, kiểu mô tả này thường dùng cho mô hình phần cứng và mô phỏng.

+ Mức mô tả theo mô hình luồng dữ liệu **RTL** (Register Transfer Level): Kiểu mô tả này khá tối ưu và có cho khả năng tổng hợp cao, độc lập với công nghệ.

+ Mức mô tả theo mô hình cấu trúc **logic**: Kiểu mô tả này thường sử dụng các cấu trúc logic đã được xây dựng sẵn, hoặc chọn trong thư viện của nhà cung cấp phù hợp với loại công nghệ sử dụng.

+ Mức mô tả theo cấu trúc **layout**. Mức độ mô tả chi tiết nhất, mô tả chi tiết tới cấu trúc bên trong những tài nguyên đã sẵn có trong cấu kiện, cách này tối ưu cho việc tổng hợp trên loại cấu kiện, công nghệ đã sử dụng.

Với hệ thống số thông thường được mô tả theo 3 mức: hành vi, RTL và cấu trúc logic. Trong một thiết kế có thể chỉ sử dụng theo một cách mô tả, hoặc cũng có thể phải dùng kết hợp cả 3 cách tùy theo độ phức tạp của thiết kế, yêu cầu về thời gian thiết kế, yêu cầu về sự tối ưu phần cứng...



Hình 9-4. Các mức độ mô tả hệ thống phần cứng số.

### 9.3.1. Phương pháp mô tả theo mô hình cấu trúc logic

Mô hình cấu trúc của một phần tử (hoặc hệ thống) có thể bao gồm nhiều cấp cấu trúc bắt đầu từ một cổng logic đơn giản đến xây dựng mô tả cho một hệ thống hoàn thiện. Thực chất của việc mô tả theo mô hình cấu trúc là mô tả các phần tử con bên trong hệ thống và sự kết nối của các phần tử con đó. Cách thức mô tả cấu trúc của thành phần con cũng tương tự như cách thức mô tả thực thể. Trước hết để mô tả cấu trúc của thành phần con, chúng ta phải xác định rõ các giao diện của thành phần con. Các giao diện này chính là các đường tín hiệu vào và ra từ thành phần con.

Trước khi được sử dụng trong kiến trúc của cả hệ thống, các thành phần phải được khai báo một cách tường minh theo cú pháp sau:

```

Component <tên_thành_phần>
    Port (<khai_báo_danh_sách_các_cổng_cục_bộ;>)
    -- Tương tự như khai báo trong thực thể
End component;
    
```

Chú ý: Các cổng vào ra của mỗi thành phần con không được kết nối trực tiếp với nhau mà phải kết nối thông qua tín hiệu nội bộ có cùng kiểu, cùng độ lớn với các cổng vào ra đó.

Cú pháp mô tả mối nối giữa các thành phần con như sau:

```

<nhãn_khởi_tạo>:<tên_thành_phần>
    
```

```
port map ([<tên_cổng_cục_bộ> =>] <biểu_thức>
{[<tên_cổng_cục_bộ>=>]<biểu_thức>});
```

Cấu trúc **port map** ánh xạ các cổng của phần tử vào các tín hiệu. Ánh xạ này có thể hiểu như việc kết nối cổng tương ứng của phần tử vào đường tín hiệu. Cấu trúc port map đặt tương ứng mỗi cổng thực của phiên bản với một cổng cục bộ thành phần. Thực hiện nối các cổng vào ra của các thành phần con với các chân vào ra của hệ thống hoặc nối với tín hiệu nội bộ trong hệ thống để kết nối tới các cổng vào ra của các thành phần con khác. Ánh xạ được thực hiện theo vị trí theo tên:

+ Khi sử dụng ánh xạ theo vị trí, chúng ta đưa ra danh sách các tín hiệu tuân theo đúng trật tự mà cổng được khai báo.

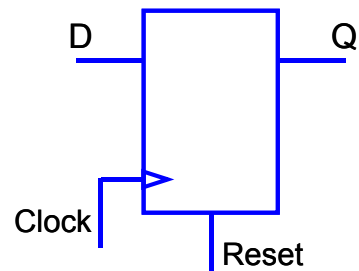
+ Đối với trường hợp ánh xạ theo tên, chúng ta sử dụng cấu trúc ánh xạ tương minh đặt tương ứng với mỗi cổng với các tín hiệu thực:

<tên\_cổng\_cục\_bộ> => <tên\_tín\_hiệu\_thực>

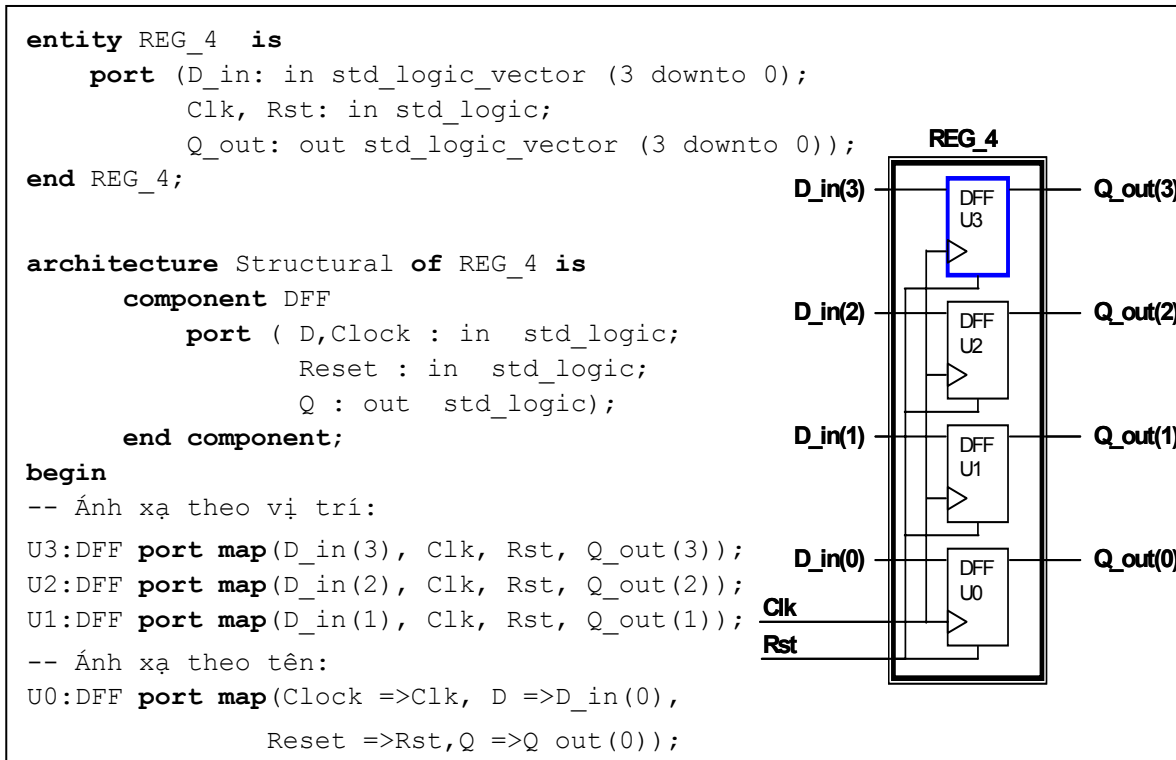
Ví dụ mô tả mô hình cấu trúc một thanh ghi 4 bit được xây dựng từ 4 triger D. Có thể mô tả triger D sau đó sau đó mô tả sơ đồ mớ nối các phần tử triger D tạo thành thanh ghi.

- Mô tả triger D như sau:

```
entity DFF is
  port ( D, Clock : in std_logic ;
        Reset : in std_logic ;
        Q : out std_logic) ;
end entity DFF ;
architecture RTL of DFF is
begin
  process (Clock, Reset)
  begin
    If (Reset = '1' ) then
      Q <= '0' ;
    elsif (Clock'event and Clock = '1') then
      Q <= D ;
    end if;
  end process ;
end architecture RTL;
```



- Ví dụ Mô tả cấu trúc của thanh ghi:

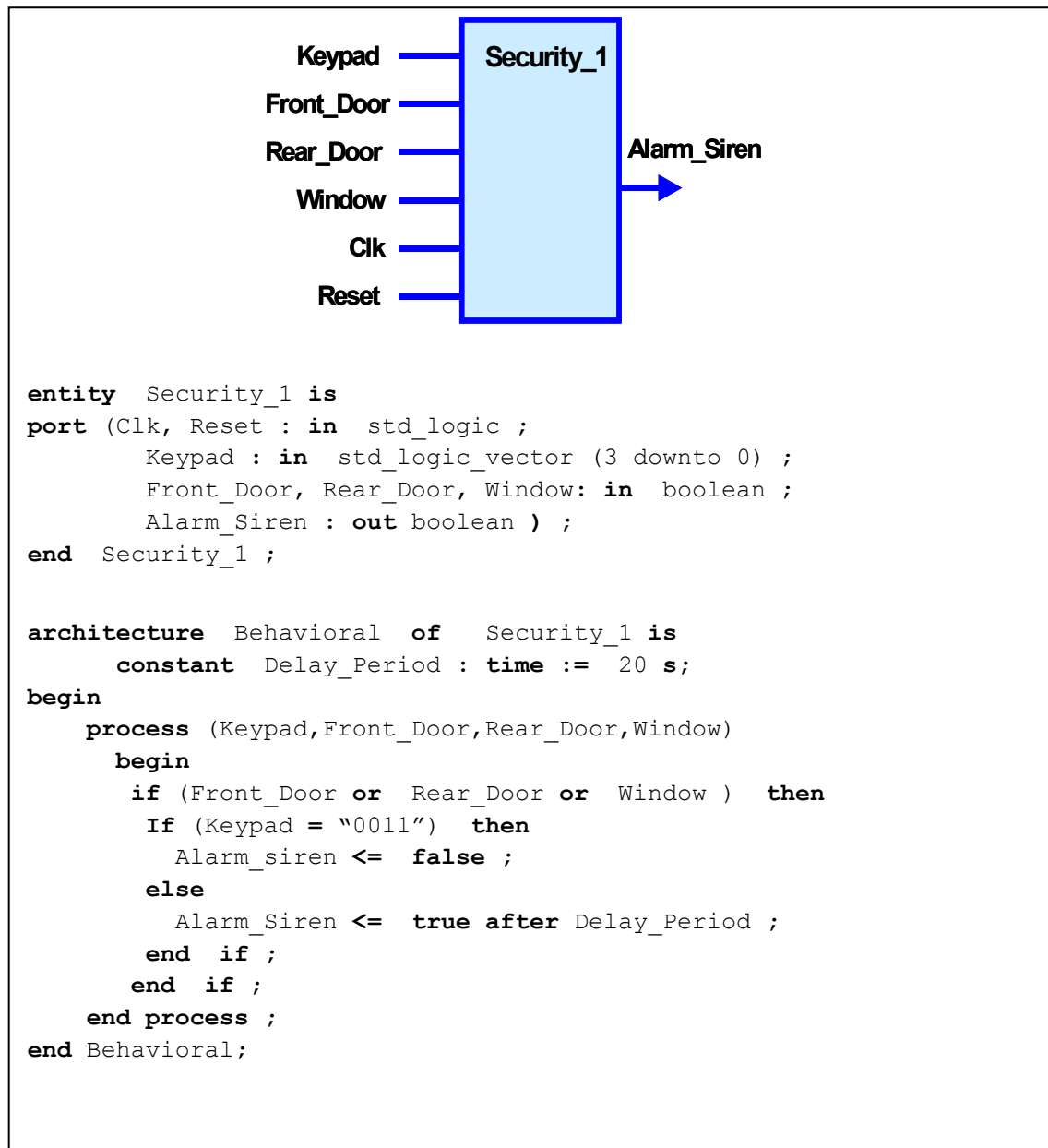


### 9.3.2. Phương pháp mô tả theo mô hình hành vi (Behavioral):

Đây là mức độ mô tả trừu tượng nhất, chủ yếu là mô tả theo chức năng của hệ thống số theo yêu cầu đầu vào và đáp ứng ra sử dụng các cấu trúc lệnh như của ngôn ngữ lập trình bậc cao như PROCESS, WAIT, IF, CASE, FOR-LOOP... Mô tả theo cách này tính ngữ nghĩa tự nhiên và giải thuật rất cao, nhập thiết kế rất nhanh, nhưng cấu trúc của phần cứng thường không rõ. Tuy nhiên với những hệ thống phức tạp, yêu cầu cần thiết kế nhanh, mà không cần yêu cầu về mức độ tối ưu phần cứng cao thường dùng cách mô tả này. Người thiết kế chỉ mô tả chức năng, hành vi được mong đợi của thiết kế bằng cách sử dụng mô tả dạng văn bản và các phần tử đồ thị. Phương pháp mô tả này thường dùng cho mô phỏng.

Ví dụ mô tả hệ thống cảnh báo theo mô hình hành vi. Hệ thống gồm có đầu vào từ các sensor (**Front\_Door**, **Rear\_Door**, **Window**), đầu vào từ bàn phím bấm Keypad, tín hiệu **Clk**, **Reset** và đầu ra điều khiển còi báo động **Alarm\_Siren**.

Chức năng hoạt động của hệ thống như sau: Nếu mỗi khi có một sensor nào đó được kích hoạt, thì hệ thống kiểm tra mã bàn phím. Nếu sau 20 giây mà không có mã bàn phím nhập đúng nhập vào thì còi báo động sẽ được bật lên.



### 9.3.3 Phương pháp mô tả theo mô hình luồng dữ liệu RTL

Hệ thống được biểu diễn theo mô hình RTL bao gồm tập các thanh ghi và các phép toán được thực hiện trên dữ liệu số nhị phân được lưu trong các thanh ghi. Luồng dữ liệu và việc xử lý dữ liệu thực hiện trên số liệu được chứa trong các thanh ghi được coi như là hoạt động chuyển đổi giữa các thanh ghi. Ví dụ mô hình RTL này được sử dụng để biểu diễn cấu trúc bộ vi xử lý. Hệ thống số được biểu diễn theo mô hình RTL khi chúng được xác định bởi 3 thành phần như sau:

- Tập các thanh ghi trong hệ thống.
- Các phép toán được thực hiện trên dữ liệu được lưu trong các thanh ghi.
- Những điều khiển để giám sát chuỗi tuần tự các phép toán trong hệ thống.

Thanh ghi gồm nhóm các Trigrơ chứa dữ liệu nhị phân và có khả năng thực hiện một hoặc nhiều phép toán cơ bản. Một thanh ghi có thể nạp thông tin mới, dịch thông tin... Một bộ đếm được coi như là một thanh ghi có khả năng tăng, giảm giá trị tuần tự. Một Trigrơ có thể coi như là thanh ghi 1 bit. Phần mạch gồm có các Trigrơ và các cổng liên quan trong bất cứ mạch tuần tự nào có thể được gọi là những thanh ghi.

Các phép toán được thực hiện trên dữ liệu chứa trong các thanh ghi là những phép toán cơ bản có thể được thực hiện song song trên chuỗi bit trong một chu kỳ clock. Kết quả của phép toán có thể thay thế dữ liệu trước đó của thanh ghi, hoặc kết quả có thể được chuyển đến thanh ghi khác. Có 4 kiểu phép toán như sau:

- + Phép chuyển đổi: truyền dữ liệu từ thanh ghi này sang thanh ghi khác.
- + Phép toán số học.
- + Phép toán logic.
- + Phép dịch.

Điều khiển khởi tạo chuỗi các phép toán bao gồm tín hiệu định thời cho phép thực hiện tuần tự các phép toán theo cách đã được mô tả trước. Có thể coi mô hình RTL là mô hình mô tả hành vi theo từng xung clock của hệ thống số.

Hệ thống số được mô tả bằng VHDL theo mô hình RTL có khả năng tổng hợp rất cao và rất dễ dàng trong việc trao đổi giữa các công cụ tổng hợp, thiết kế, và có thể tổng hợp trên các công nghệ PLD khác nhau.

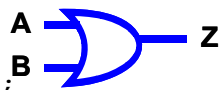
Theo mô hình RTL, hệ thống số được mô tả bằng các tiến trình tổ hợp (combinatorial process) và các tiến hoạt động theo clock (clocked process)

### 9.3.3.1. Mô tả mạch tổ hợp

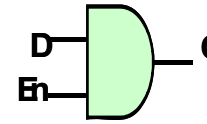
Mạch logic tổ hợp có thể mô tả bằng các cấu trúc lệnh song, tuy nhiên thường dùng các **process** tổ hợp. Trong các process tổ hợp *tất cả các tín hiệu vào* của mạch tổ hợp phải được đưa vào danh sách tín hiệu kích thích.

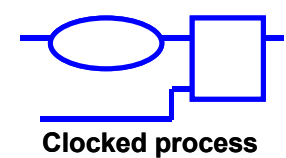
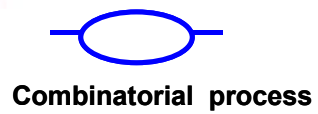
Ví dụ tiến trình tổ hợp như sau:

```
process (A, B)
begin
    Z <= A or B ;
end process;
```



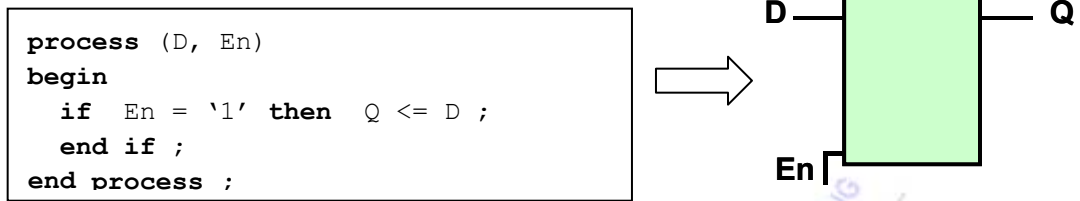
```
process (D, En)
begin
    -- gán mặc định đầu ra
    Q <= 0;
    if En = '1' then Q <= D ;
    end if ;
end process ;
```





**Chú ý** trong các process tổ hợp nên có phép gán giá trị mặc định cho đầu ra để tránh trường hợp mạch bị biến thành mạch chốt theo mức.





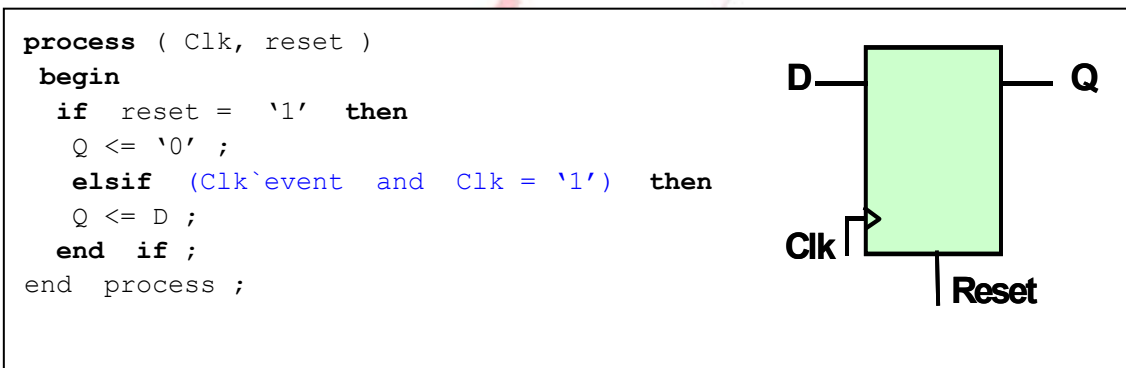
Khi mô tả mạch logic tổ hợp các biến và tín hiệu trong một process không được nhận giá trị khởi tạo trước bởi vì mạch tổ hợp không chứa các phần tử nhớ. Khi trong mô hình mạch các biến hoặc tín hiệu được khởi tạo giá trị trước, chương trình tổng hợp sẽ tạo ra các phần tử nhớ để lưu trữ các giá trị khởi tạo, mạch trở thành mạch có nhớ.

Mọi câu lệnh tuần tự trừ các lệnh wait, loop, if với những tín hiệu điều khiển theo sườn đều có thể dùng để mô tả các mạch tổ hợp. Các phép toán số học, logic, quan hệ đều có thể được sử dụng trong biểu thức.

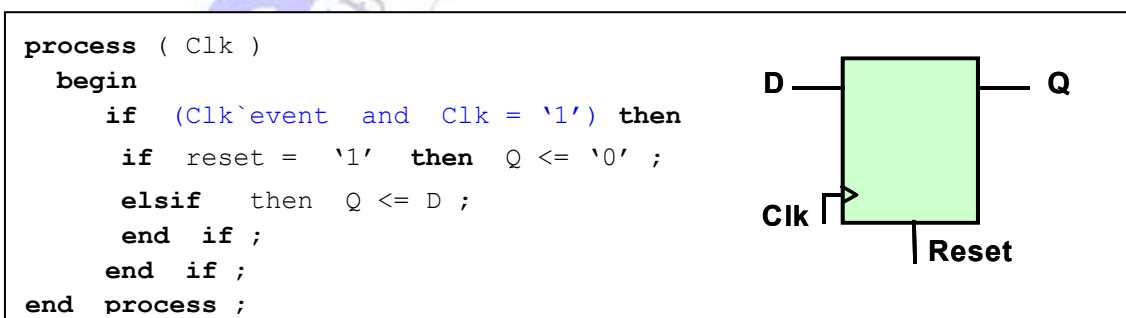
### 9.3.3.2. Mô tả mạch tuần tự:

Tiến trình hoạt động theo clock có thể được mô tả thành tiến trình đồng bộ (danh sách tín hiệu kích thích chỉ có duy nhất tín hiệu clock, mọi biến đổi của mạch được đồng bộ theo sườn clock) hoặc thành tiến trình không đồng bộ.

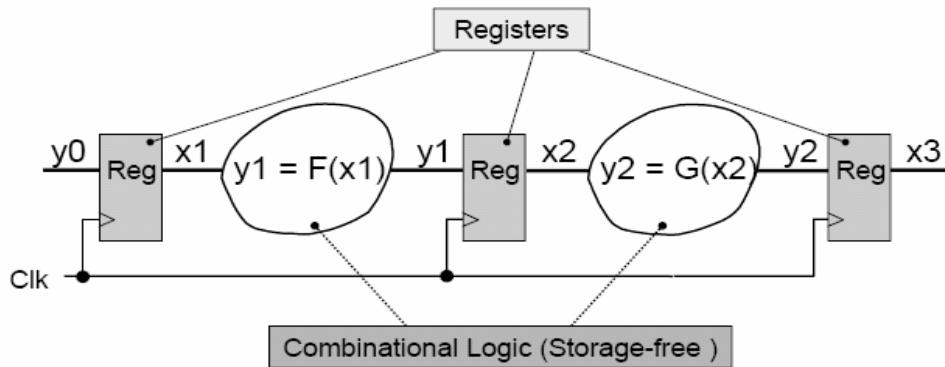
Ví dụ mô tả hoạt động của Triger D làm việc theo sườn dương với các tín hiệu Reset không đồng bộ như sau:



Ví dụ mô tả hoạt động của Triger D làm việc theo sườn dương với các tín hiệu Reset đồng bộ như sau:



Tóm lại biểu diễn hệ thống số theo mô hình RTL cần sử dụng các cấu trúc thanh ghi (Registers) và mạch tổ hợp (combinational logic), ví dụ tả datapath theo mô hình RTL như hình vẽ 9-5 sau:



Hình 9-5. Một mô hình RTL

Mô tả VHDL cho mô hình trên có thể thực hiện theo 2 cách như sau:

```

architecture SPLIT of DATAPATH is
    signal X1, Y1, X2, Y2 : ...
begin
    REG : process (CLK)
    begin
    if (CLK'event and CLK = '1') then
        X1 <= Y0;
        X2 <= Y1;
        X3 <= Y2;
    end if;
    end process;
    LOGIC : process (X1, X2)
    begin
        Y1 <= F(X1);
        Y2 <= G(X2);
    end process;
end SPLIT;

```

} **Registe**  
rs

```

architecture COMBINED of DATAPATH is
    signal X1, X2 : ...
begin
    process (CLK)                                -- Registers
    begin
    if (CLK'event and CLK = '1') then          Combinational Logic
        X2 <= F(X1);
        X3 <= G(X2);
        X1 <= Y0;
    end if;
    end process;

```

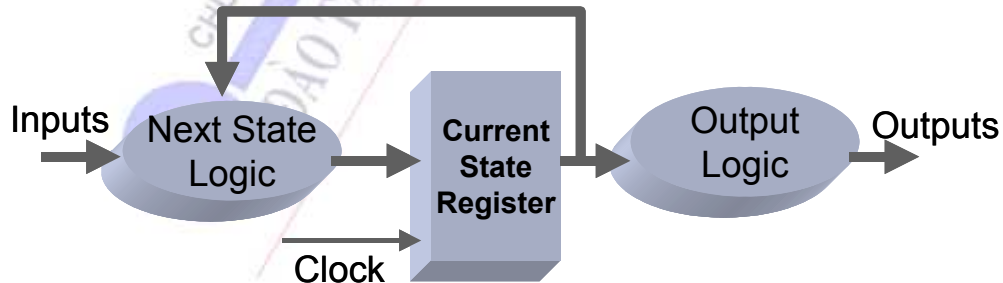
### 9.3.4 Phương pháp mô tả theo mô hình đồ hình trạng thái (máy trạng thái State Machine)

Hoạt động của một hệ thống số tuần tự có thể được mô tả dưới dạng đồ hình trạng thái Moore hoặc Mealy. Dùng VHDL có thể mô tả được đồ hình chuyển đổi trạng thái đó. Bảng sau cho biết khả năng mô tả đồ hình trạng thái dùng VHDL:

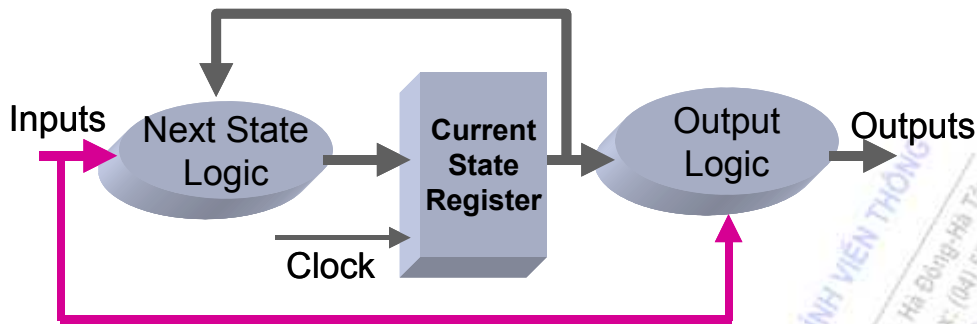
STT	Yêu cầu mô tả	Sử dụng cấu trúc trong VHDL
1	- Trạng thái logic hiện tại	- <b>Process</b> hoạt động theo clock
2	- Xác định trạng thái logic tiếp theo	- <b>Process</b> tổ hợp
3	- Xác định đầu ra	- <b>Process</b> tổ hợp
4	- Đặt tên cho các trạng thái	- Kiểu dữ liệu liệt kê
5	- Đánh giá mỗi trạng thái	- Lệnh <b>Case</b>
6	- Đánh giá các điều kiện đầu vào	- Lệnh <b>if/else</b>

Tổng kết lại các kiểu đồ hình trạng thái như sau:

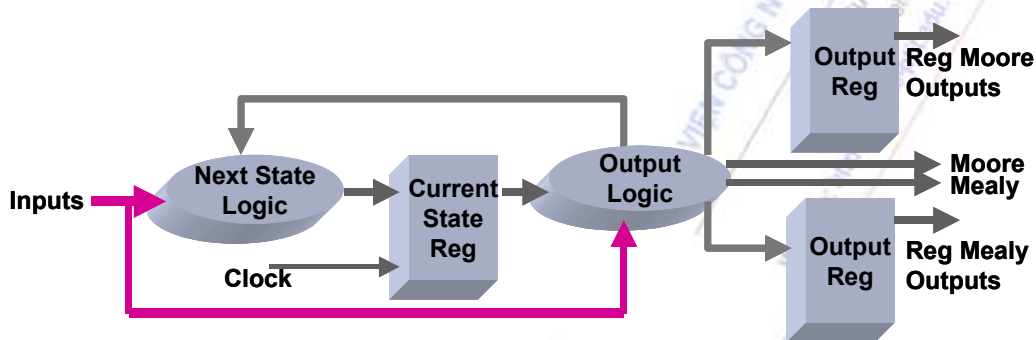
- **Mô hình Moore:** Kết quả đầu ra chỉ phụ thuộc vào trạng thái hiện tại.



- **Mô hình Mealy:** Đầu ra phụ thuộc vào cả trạng thái hiện tại và tín hiệu vào.



Trong thực tế hệ thống số thường được mô tả bằng việc kết hợp cả mô hình Moore và Mealy:



- Cách sử dụng kiểu dữ liệu liệt kê để đặt tên cho các trạng thái như sau:

```
architecture RTL of FSM is
    . . .
    type My_State is ( Init, Load, Fetch, Stor_A, Stor_B ) ;
    signal Current_State, Next_State : My_State;
    . . .
begin
```

- Cách sử dụng hằng để mã hóa các trạng thái theo như mong muốn:

```
subtype My_State is std_logic_vector( 0 to 5 ) ;

constant Init      : My_State := "111000" ;
constant Load     : My_State := "101010" ;
constant Init     : My_State := "000011" ;
signal Curr_State, Next_State : My_State ;
. . .
begin --architecture
```

- Để mô tả quá trình chuyển đổi trạng thái và cập nhật kết quả đầu ra ứng với mỗi trạng thái thông thường sử dụng cách mô tả bằng nhiều tiến trình

+ Tiến trình cập nhập trạng thái mới của hệ thống (tiến trình Sync).

```
Sync: process ( CLK , RST)
begin
    . . .
end process Sync ;
```

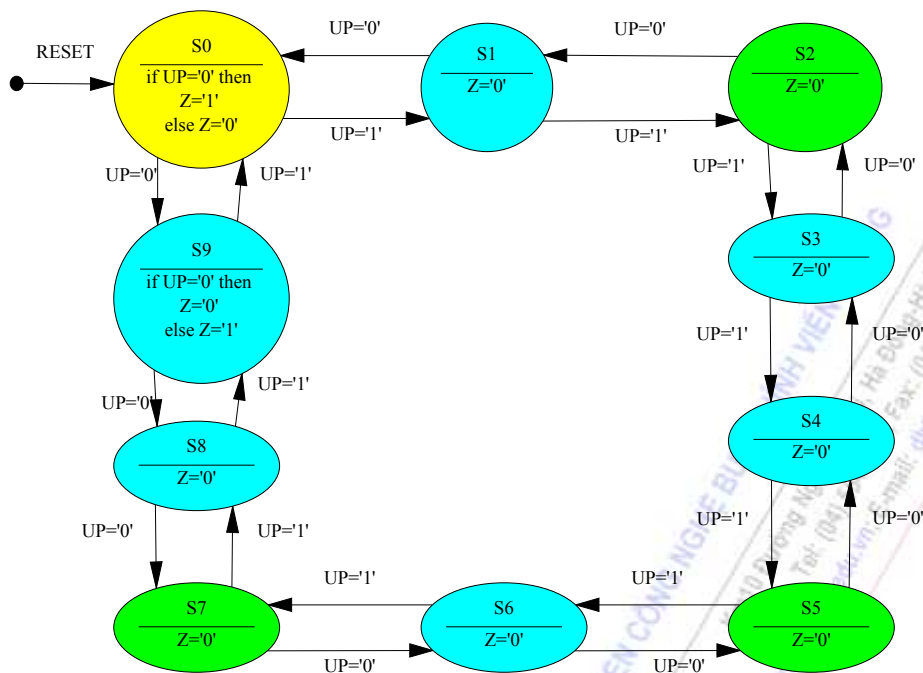
+ Tiến trình kiểm tra điều kiện chuyển đổi trạng thái (tiến trình Comb).

```
Comb: process ( Curr_State, In1, In2...)
begin
    . . .
end process Comb ;
```

+ Tiến trình cập kết quả đầu ra ứng với mỗi trạng thái (tiến trình Outputs).

```
Outputs: process ( Curr_State, In1, In2...)
begin
    . . .
end process Outputs ;
```

- Ví dụ bộ đếm thập phân thuận nghịch đồng bộ có đồ đồ hình trạng thái như sau:



- Mô tả VHDL cho đồ hình trạng thái trên như sau:

```

LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY FSM IS
    PORT (CLK,RESET,UP: IN std_logic;
          Z : OUT std_logic);
END;
ARCHITECTURE BEHAVIOR OF FSM IS
    SIGNAL sreg : std_logic_vector (3 DOWNTO 0);
    SIGNAL next_sreg : std_logic_vector (3 DOWNTO 0);
    CONSTANT S0 : std_logic_vector (3 DOWNTO 0) := "0000";
    CONSTANT S1 : std_logic_vector (3 DOWNTO 0) := "0001";
    CONSTANT S2 : std_logic_vector (3 DOWNTO 0) := "0010";
    CONSTANT S3 : std_logic_vector (3 DOWNTO 0) := "0011";
    CONSTANT S4 : std_logic_vector (3 DOWNTO 0) := "0100";
    CONSTANT S5 : std_logic_vector (3 DOWNTO 0) := "0101";
    CONSTANT S6 : std_logic_vector (3 DOWNTO 0) := "0110";
    CONSTANT S7 : std_logic_vector (3 DOWNTO 0) := "0111";
    CONSTANT S8 : std_logic_vector (3 DOWNTO 0) := "1000";
    CONSTANT S9 : std_logic_vector (3 DOWNTO 0) := "1001";

    SIGNAL next_Z : std_logic;
BEGIN
    Sync: PROCESS (CLK)
    BEGIN
        IF CLK='1' AND CLK'event THEN
    
```

```

        if RESET='1' then
            sreg<= S0;
        else
            sreg <= next_sreg;
        end if;
    END IF;
END PROCESS;

Comb: PROCESS (sreg,UP)
BEGIN
    CASE sreg IS
        WHEN S0 =>
            IF ( UP='0' ) THEN      next_sreg<=S9;
            ELSE                      next_sreg<=S1;
            END IF;
        WHEN S1 =>
            IF ( UP='0' ) THEN      next_sreg<=S0;
            ELSE                      next_sreg<=S2;
            END IF;
        WHEN S2 =>
            IF ( UP='0' ) THEN      next_sreg<=S1;
            ELSE                      next_sreg<=S3;
            END IF;
        WHEN S3 =>
            IF ( UP='0' ) THEN      next_sreg<=S2;
            ELSE                      next_sreg<=S4;
            END IF;
        WHEN S4 =>
            IF ( UP='0' ) THEN      next_sreg<=S3;
            ELSE                      next_sreg<=S5;
            END IF;
        WHEN S5 =>
            IF ( UP='0' ) THEN      next_sreg<=S4;
            ELSE                      next_sreg<=S6;
            END IF;
        WHEN S6 =>
            IF ( UP='0' ) THEN      next_sreg<=S5;
            ELSE                      next_sreg<=S7;
            END IF;
        WHEN S7 =>
            IF ( UP='0' ) THEN      next_sreg<=S6;
            ELSE                      next_sreg<=S8;
    
```

```
        END IF;
    WHEN S8 =>
        IF ( UP='0' ) THEN      next_sreg<=S7;
        ELSE                    next_sreg<=S9;
        END IF;
    WHEN S9 =>
        IF ( UP='0' ) THEN      next_sreg<=S8;
        ELSE                    next_sreg<=S0;
        END IF;
    WHEN OTHERS => next_sreg<=S0;
END CASE;
END PROCESS;
Outputs: PROCESS (sreg,UP)
BEGIN
    IF UP='1' THEN
        if sreg=S9 then      Z<= '1';
        else                 Z<= '0';
        end if;
    ELSE
        if sreg=S0 then      Z<= '1';
        else                 Z<= '0';
        end if;
    END IF;
END PROCESS;
END BEHAVIOR;
```

## TÓM TẮT

Thiết kế với sự trợ giúp của máy tính của các hệ thống kỹ thuật số được dùng rộng rãi trong công nghiệp. Do đó, ta cần phải hiểu các khái niệm khác nhau trong quá trình thiết kế. Ngôn ngữ mô tả phần cứng phổ biến VHDL là loại ngôn ngữ được trình bày trong chương này. Đây là một chủ đề rất rộng nên chúng tôi không thể trình bày chi tiết của VHDL. Tuy nhiên các khái niệm cơ bản được trình bày ở đây sẽ giúp cho chúng ta học những chi tiết về ngôn ngữ từ những quyển sách viết về VHDL



## CÂU HỎI ÔN TẬP CHƯƠNG 8 VÀ CHƯƠNG 9

- Đặc điểm nào dưới đây là nhược điểm của phương pháp thiết kế mạch dùng IC có chức năng cố định?
  - Chi phí thiết kế thấp
  - Vận hành nhanh xung quanh bản thiết kế
  - Khó khăn khi triển khai các thiết kế phức tạp
  - Tương đối dễ dàng khi thử nghiệm các mạch thiết kế
- Đặc điểm nào dưới đây là ưu điểm của phương pháp thiết kế mạch dùng IC có chức năng cố định?
  - Yêu cầu công suất điện tiêu thụ lớn
  - Khó khăn khi sửa chữa, nâng cấp thiết kế
  - Thiếu tính bảo mật
  - Tương đối dễ dàng khi thử nghiệm mạch thiết kế
- Trong số các loại cấu kiện logic sau, loại nào không thuộc họ PLD
  - CPLD
  - FPGA
  - Vi xử lý
  - SPLD
- Đặc điểm nào dưới đây không phải là ưu điểm của PLD
  - Mật độ tích hợp cao.
  - Bảo đảm tính bảo mật của thiết kế
  - Thời gian thiết kế ngắn
  - Chi phí sản xuất số lượng lớn cao
- Trong cấu trúc của SPLD không có phần tử nào
  - Mảng các cổng logic AND,OR.
  - Ma trận kết nối
  - Bộ nhớ RAM
  - Triger
- Khối nào sau đây không có trong cấu trúc của CPLD
  - Khối logic gồm ma trận hạng tích AND, OR
  - Khối Microcell chứa tài nguyên về các Triger, thanh ghi

- C. Ma trận kết nối trung tâm
  - D. Vi xử lý
7. Để thực hiện hàm logic tổ hợp trong FPGA sử dụng
- A. Ma trận hạng tích AND, OR.
  - B. Cấu trúc bảng tra LUT dựa vào SDRAM .
  - C. Các cấu trúc thanh ghi
  - D. Cấu trúc vào/ra.
8. Xác định phát biểu sai trong số các phát biểu sau
- A. FPGA có cấu trúc không đồng nhất
  - B. CPLD có cấu trúc đồng nhất
  - C. Cấu hình của CPLD được lưu lại khi mất điện
  - D. Cấu hình trong FPGA dựa vào công nghệ SRAM được lưu lại khi mất điện
9. Trình tự thực hiện trong lưu đồ thiết kế cho CPLD/FPGA là:
- A. Nhập thiết kế, kiểm tra thiết kế, tổng hợp thiết kế, mô phỏng định thời, thực hiện thiết kế, cấu hình.
  - B. Nhập thiết kế, kiểm tra thiết kế, thực hiện thiết kế, tổng hợp thiết kế, mô phỏng định thời, cấu hình.
  - C. Nhập thiết kế, tổng hợp thiết kế, kiểm tra thiết kế, thực hiện thiết kế, mô phỏng định thời, cấu hình.
  - D. Nhập thiết kế, kiểm tra thiết kế, tổng hợp thiết kế, thực hiện thiết kế, mô phỏng định thời, cấu hình.
10. Kết quả của bước tổng hợp thiết kế trong lưu đồ thiết kế cho CPLD/FPGA là:
- A. File mô tả VHDL
  - B. File cấu hình
  - C. File netlist
  - D. File sơ đồ mạch
11. Kết quả của bước thực hiện thiết kế trong lưu đồ thiết kế cho CPLD/FPGA là:
- A. File mô tả VHDL
  - B. File cấu hình
  - C. File netlist
  - D. File sơ đồ mạch
12. Trong bước thực hiện thiết kế của lưu đồ thiết kế cho CPLD/FPGA gồm các chức năng:

- A. Mô phỏng chức năng, tổng hợp thiết kế.
  - B. Biên dịch, map, Định vị trí và định tuyến kết nối.
  - C. Mô phỏng định thời, tạo cấu hình, biên dịch.
  - D. Tạo file mô tả HDL, tổng hợp thiết kế, Định vị trí và định tuyến kết nối.
13. VHDL là ngôn ngữ:
- A. Lập trình hợp ngữ
  - B. Lập trình bậc cao
  - C. Lập trình mạng
  - D. Mô tả phần cứng
14. Trình tự sắp xếp theo mức độ mô tả trừu tượng tăng dần dùng VHDL là:
- A. Mức hành vi, mức luồng dữ liệu RTL, mức logic, mức layout.
  - B. Mức hành vi, mức logic, mức luồng dữ liệu RTL, mức layout.
  - C. Mức layout, mức logic, mức hành vi, mức luồng dữ liệu RTL.
  - D. Mức layout, mức logic, mức luồng dữ liệu RTL, mức hành vi.
15. Đối tượng tín hiệu (signal) trong ngôn ngữ VHDL để :
- A. Lưu các kết quả trung gian
  - B. Biểu diễn đường kết nối trong hệ thống phần cứng số
  - C. Lưu những giá trị cố định
  - D. Biểu diễn cổng vào hoặc ra của thực thể
16. Đối tượng biến (variable) trong ngôn ngữ VHDL để :
- A. Lưu các kết quả trung gian
  - A. Biểu diễn đường kết nối trong hệ thống phần cứng số
  - C. Lưu những giá trị cố định
  - D. Biểu diễn cổng vào hoặc ra của thực thể
17. Cho khai báo của các đối tượng như sau:
- ```
signal A : in std_logic;
```
- Phép gán nào đúng:
- A. A:= '1';
  - B. A<=1;
  - C. A<='1';
  - D. A<=true;

18. Cho khai báo của các đối tượng như sau:

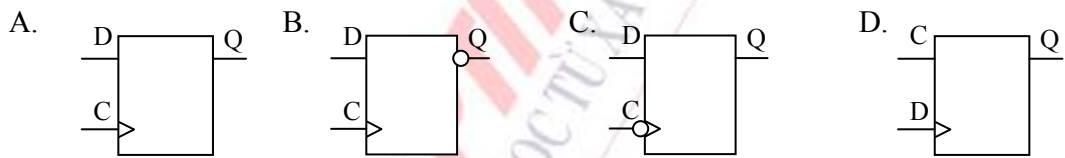
Variable A : in std\_logic;

Phép gán nào đúng:

- A. A<=true;
- B. A:=1;
- C. A<='1';
- D. A:='1';

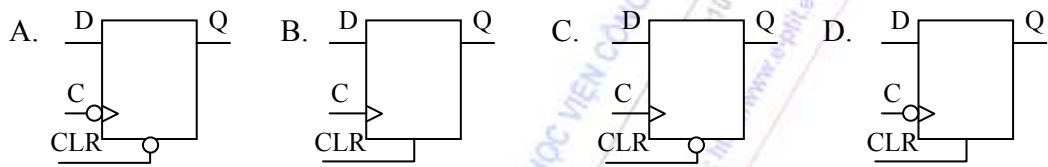
19. Mô hình phần cứng nào tổng hợp được ứng với đoạn mô tả như sau:

```
library ieee;
use ieee.std_logic_1164.all;
entity flop is
  port(C, D : in std_logic;
        Q : out std_logic);
end flop;
architecture archi of flop is
begin
  process (C)
  begin
    if (C'event and C='1') then
      Q <= D;
    end if;
  end process;
end archi;
```



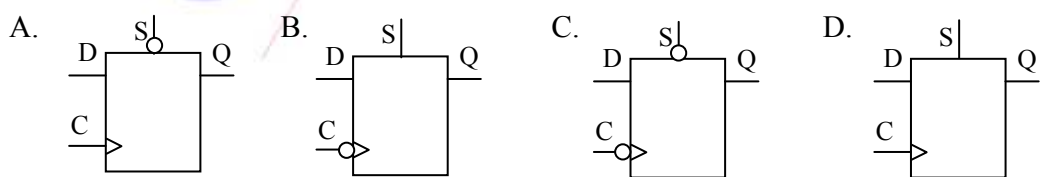
20. Mô hình phần cứng nào tổng hợp được ứng với đoạn mô tả như sau:

```
entity flop is
  port(C, D, CLR : in std_logic;
        Q          : out std_logic);
end flop;
architecture archi of flop is
begin
  process (C, CLR)
  begin
    if (CLR = '1')then
      Q <= '0';
    elsif (C'event and C='0')then
      Q <= D;
    end if;
  end process;
end archi;
```



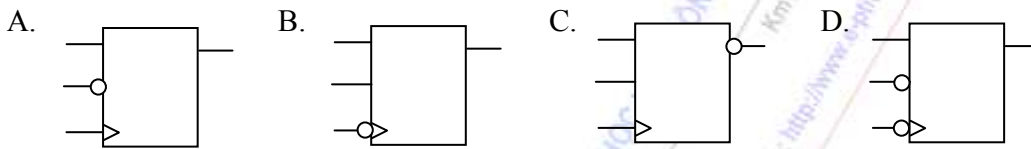
21. Mô hình phần cứng nào tổng hợp được ứng với đoạn mô tả như sau:

```
entity flop is
  port(C, D, S : in std_logic;
        Q      : out std_logic);
end flop;
architecture archi of flop is
begin
  process (C)
  begin
    if (C'event and C='1') then
      if (S='1') then
        Q <= '1';
      else
        Q <= D;
      end if;
    end if;
  end process;
end archi;
```

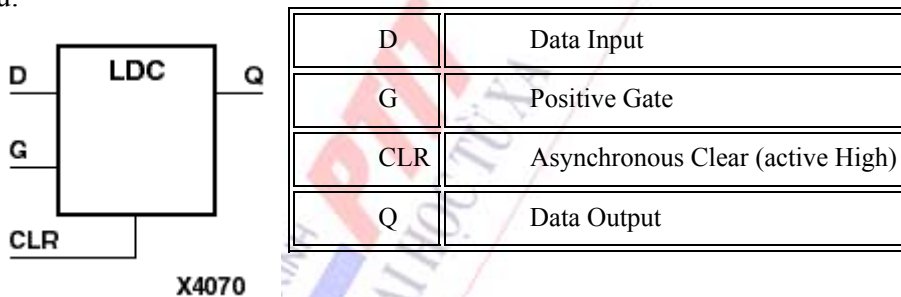


22. Mô hình phần cứng nào tổng hợp được ứng với đoạn mô tả như sau:

```
entity flop is
  port(C, D, CE : in std_logic;
        Q : out std_logic);
end flop;
architecture archi of flop is
  begin
    process (C)
      begin
        if (C'event and C='1') then
          if (CE='0') then
            Q <= D;
          end if;
        end if;
      end process;
    end archi;
```



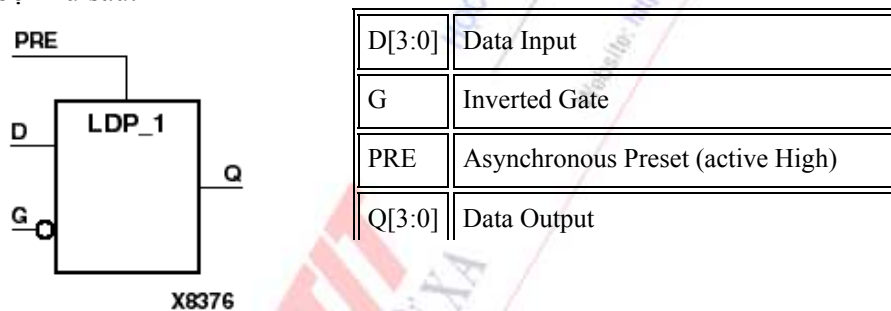
23. Đoạn mô tả VHDL nào mô tả cho mô hình mạch chốt cổng dương và xóa không đồng bộ như sau:



|                                                                                                                                                                                                                                                                                                                                            |                                                                                                                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>A.</p> <pre>entity latch is   port(G, D, CLR : in std_logic;         Q : out std_logic); end latch; architecture archi of latch is   begin     process (CLR, D, G)       begin         if (CLR='1') then           Q &lt;= '1';         elsif (G='1') then           Q &lt;= D;         end if;       end process;     end archi;</pre> | <p>B.</p> <pre>entity latch is   port(G, D, CLR : in std_logic;         Q : out std_logic); end latch; architecture archi of latch is   begin     process (CLR, D, G)       begin         if (CLR='0') then           Q &lt;= '0';         elsif (G='1') then           Q &lt;= D;         end if;       end process;     end archi;</pre> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                                                                                                                                                                                                                                                                                        |                                                                                                                                                                                                                                                                                                        |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>C.</p> <pre>entity latch is   port(G, D, CLR : in std_logic;         Q : out std_logic); end latch; architecture archi of latch is begin   process (CLR, D, G)   begin     if (CLR='1') then       Q &lt;= '0';     elsif (G='1') then       Q &lt;= D;     end if;   end process; end archi;</pre> | <p>D.</p> <pre>entity latch is   port(G, D, CLR : in std_logic;         Q : out std_logic); end latch; architecture archi of latch is begin   process (CLR, D, G)   begin     if (CLR='1') then       Q &lt;= '0';     elsif (G='0') then       Q &lt;= D;     end if;   end process; end archi;</pre> |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

24. Đoạn mô tả kiến trúc nào mô tả cho mô hình mạch chốt cổng đảo và Preset không đồng bộ như sau:



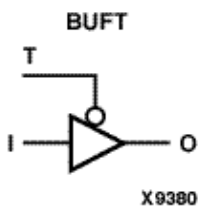
Trong đó mô tả thực thể như sau:

```
entity latch is
  port(D : in std_logic_vector(3 downto 0);
        G, PRE : in std_logic;
        Q : out std_logic_vector(3 downto 0));
end latch;
```

|                                                                                                                                                                                                                |                                                                                                                                                                                                                |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>A.</p> <pre>architecture archi of latch is begin   process (PRE, G)   begin     if (Q='1') then       Q &lt;= "1111";     elsif (PRE='0') then       Q &lt;= D;     end if;   end process; end archi;</pre> | <p>B.</p> <pre>architecture archi of latch is begin   process (PRE, G)   begin     if (PRE='1') then       Q &lt;= "1111";     elsif (G='0') then       Q &lt;= D;     end if;   end process; end archi;</pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                                                                                                                                                                                             |                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>C.</p> <pre>architecture archi of latch is begin   process (PRE)   begin     if (PRE='1') then       Q &lt;= "1111";     elsif (G='0') then       Q &lt;= D;     end if;   end process; end archi;</pre> | <p>D.</p> <pre>architecture archi of latch is begin   process (PRE, G)   begin     if (PRE='1') then       Q &lt;= "1111";     elsif (G='1') then       Q &lt;= D;     end if;   end process; end archi;</pre> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

25. Đoạn mô tả kiến trúc nào mô tả cho cổng 3 trạng thái sau:



Trong đó mô tả thực thể như sau:

```
entity three_st is
  port( T, I : in std_logic;
        O : out std_logic);
end three_st;
```

|                                                                                                                                                                                              |                                                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>A.</p> <pre>architecture archi of three_st is begin   process (I, T)   begin     if (T='0') then       O &lt;= I;     else       O &lt;= 'X';     end if;   end process; end archi;</pre> | <p>B.</p> <pre>architecture archi of three_st is begin   process (I, T)   begin     if (T='1') then       O &lt;= I;     else       O &lt;= 'Z';     end if;   end process; end archi;</pre> |
| <p>C.</p> <pre>architecture archi of three_st is begin   O &lt;= I when T='1' else     'Z'; end archi;</pre>                                                                                 | <p>D.</p> <pre>architecture archi of three_st is begin   O &lt;= I when T='0' else     'Z'; end archi;</pre>                                                                                 |



26. Đoạn mô tả kiến trúc nào mô tả hoạt động của bộ đếm tiến 4 bit có xóa không đồng bộ có mô tả thực thể như sau:

```
entity counter is
  port( Clk, CLR : in  std_logic;
        Q : out std_logic_vector(3 downto 0));
end counter;
```

|                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                        |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>A.</p> <pre>architecture archi of counter is   signal tmp: std_logic_vector(3                                 downto 0); begin   process (Clk, CLR)   begin     if (CLR='1') then       tmp &lt;= "0000";     elsif (Clk'event and Clk='1')     then       tmp &lt;= tmp + 1;     end if;   end process;   Q &lt;= tmp; end archi;</pre> | <p>B.</p> <pre>architecture archi of counter is   signal tmp: std_logic_vector(3                                 downto 0); begin   process (Clk)   begin     if (Clk'event and Clk='1') then       if (CLR='1') then         tmp &lt;= "0000";       else tmp &lt;= tmp + 1;       end if;     end if;   end process;   Q &lt;= tmp; end archi;</pre> |
| <p>C.</p> <pre>architecture archi of counter is begin   process (Clk, CLR)   begin     if (CLR='1') then       Q &lt;= "0000";     elsif (Clk'event and Clk='0')     then       Q &lt;= Q + 1;     end if;   end process; end archi;</pre>                                                                                                  | <p>D.</p> <pre>architecture archi of counter is   signal tmp: std_logic_vector(3                                 downto 0); begin   process (Clk)   begin     if (Clk'event and Clk='0') then       if (CLR='1') then         tmp &lt;= "0000";       else tmp &lt;= tmp - 1;       end if;     end if;   end process;   Q &lt;= tmp; end archi;</pre> |

27. Mô hình mạch số nào có mô tả VHDL như sau:

```
entity counter is
  port( Clk, S : in  std_logic;
        Q : out std_logic_vector(3 downto 0));
end counter;
architecture archi of counter is
  signal tmp: std_logic_vector(3 downto 0);
begin
  process (Clk)
  begin
    if (Clk'event and Clk='1') then
      if (S='1') then
        tmp <= "1111";
      else
        tmp <= tmp - 1;
      end if;
    end if;
  end process;
  Q <= tmp;
end archi;
```

- A. Bộ đếm lùi 4 bit ra Q[3:0], hoạt động ở sườn âm của clock CLK, tín hiệu thiết lập S tích cực dương và đồng bộ
- B. Bộ đếm lùi 4 bit ra Q[3:0], hoạt động ở sườn dương của clock CLK, tín hiệu thiết lập S tích cực dương và đồng bộ.
- C. Bộ đếm tiến 4 bit ra Q[3:0], hoạt động ở sườn dương của clock CLK, tín hiệu thiết lập S tích cực dương và đồng bộ
- D. Bộ đếm lùi 4 bit ra Q[3:0], hoạt động ở sườn dương của clock CLK, tín hiệu thiết lập S tích cực dương và không đồng bộ.

28. Đoạn mô tả kiến trúc nào mô tả hoạt động của bộ đếm tiến 4 bit nạp không đồng bộ từ tín hiệu đầu vào, hoạt động ở sườn clock âm và có mô tả thực thể như sau:

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity counter is
  port( Clk, ALOAD : in  std_logic;  -- Clock và tín hiệu nạp
        D : in std_logic_vector(3 downto 0); -- Đầu vào bộ đếm
        Q : out std_logic_vector(3 downto 0)); -- Đầu ra bộ đếm
end counter;
    
```

|                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                           |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>A.</p> <pre> architecture archi of counter is   signal tmp: std_logic_vector(3                                 downto 0); begin   process (Clk,ALOAD, D)   begin     if (ALOAD='1') then       tmp &lt;= D;     elsif (Clk'event and Clk='1')     then tmp &lt;= tmp + 1;     end if;   end process;   Q &lt;= tmp; end archi;         </pre> | <p>B.</p> <pre> architecture archi of counter is   signal tmp: std_logic_vector(3                                 downto 0); begin   process (Clk,D)   begin     if (ALOAD='1') then       tmp &lt;= D;     elsif (Clk'event and Clk='0')     then tmp &lt;= tmp + 1;     end if;   end process;   Q &lt;= tmp; end archi;         </pre> |
| <p>C.</p> <pre> architecture archi of counter is   signal tmp: std_logic_vector(3                                 downto 0); begin   process (Clk,ALOAD, D)   begin     if (ALOAD='1') then       tmp &lt;= D;     elsif (Clk'event and Clk='0')     then tmp &lt;= tmp + 1;     end if;   end process;   Q &lt;= tmp; end archi;         </pre> | <p>D.</p> <pre> architecture archi of counter is   signal tmp: std_logic_vector(3                                 downto 0); begin   process (Clk)   begin     if (ALOAD='1') then       tmp &lt;= D;     elsif (Clk'event and Clk='0')     then tmp &lt;= tmp + 1;     end if;   end process;   Q &lt;= tmp; end archi;         </pre>   |

29. Mô hình mạch số nào có mô tả VHDL như sau:

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity counter is
  port( Clk, SLOAD : in  std_logic;
        Q : out std_logic_vector(3 downto 0));
end counter;
architecture archi of counter is
  signal tmp: std_logic_vector(3 downto 0);
  begin
    process (Clk)
      begin
        if (Clk'event and Clk='1') then
          if (SLOAD='1') then
            tmp <= "1010";
          else
            tmp <= tmp + 1;
          end if;
        end if;
      end process;
      Q <= tmp;
    end archi;
  
```

- A. Bộ đếm tiến 4 bit ra Q[3:0], hoạt động ở sườn âm của clock CLK, nạp đồng bộ hằng số "1010" (theo mức tích cực dương).
- B. Bộ đếm tiến 4 bit ra Q[3:0], hoạt động ở sườn dương của clock CLK, nạp không đồng bộ hằng số "1010" (theo mức tích cực dương).
- C. Bộ đếm tiến 4 bit ra Q[3:0], hoạt động ở sườn dương của clock CLK, nạp đồng bộ hằng số "1010" (theo mức tích cực dương).
- D. Bộ đếm lùi 4 bit ra Q[3:0], hoạt động ở sườn dương của clock CLK, nạp đồng bộ hằng số "1010" (theo mức tích cực dương).

30. Đoạn mô tả kiến trúc nào mô tả cho mô hình thanh ghi 4 bit hoạt động sườn dương của clock, có tín hiệu chốt clock và thiết lập không đồng bộ,



Mô tả thực thể của thanh ghi như sau:

```
library ieee;
use ieee.std_logic_1164.all;
entity flop is
  port( C, CE, PRE : in std_logic;
        D : in std_logic_vector (3 downto 0);
        Q : out std_logic_vector (3 downto 0));
end flop;
```

|                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>A.</p> <pre>architecture archi of flop is begin   process (C)   begin     if (PRE='1') then       Q &lt;= "1111";     elsif (C'event and C='1') then       if (CE='1') then         Q &lt;= D;       end if;     end if;   end process; end archi;</pre>      | <p>B.</p> <pre>architecture archi of flop is begin   process (C, PRE)   begin     if (PRE='1') then       Q &lt;= "1111";     elsif (C'event and C='1') then       if (CE='0') then         Q &lt;= D;       end if;     end if;   end process; end archi;</pre> |
| <p>C.</p> <pre>architecture archi of flop is begin   process (C, PRE)   begin     if (PRE='1') then       Q &lt;= "1111";     elsif (C'event and C='1') then       if (CE='1') then         Q &lt;= D;       end if;     end if;   end process; end archi;</pre> | <p>D.</p> <pre>architecture archi of flop is begin   process (C, PRE)   begin     if (PRE='1') then       Q &lt;= "0000";     elsif (C'event and C='1') then       if (CE='1') then         Q &lt;= D;       end if;     end if;   end process; end archi;</pre> |

31. Mô hình mạch số nào có đoạn mô tả VHDL như sau:

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity counter is
  port( Clk, SLOAD : in  std_logic;
        Q : out std_logic_vector(3 downto 0));
end counter;
architecture archi of counter is
  signal tmp: std_logic_vector(3 downto 0);
  begin
    process (Clk)
    begin
      if (Clk'event and Clk='1') then
        if (SLOAD='1') then tmp <= "1010";
        else tmp <= tmp + 1;
        end if;
      end if;
    end process;
    Q <= tmp;
  end archi;

```

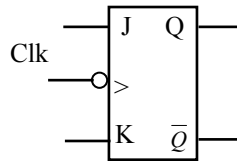
- A. Bộ đếm tiến 4 bit đầu ra Q [3:0] hoạt động với sườn âm clock, nạp đồng bộ giá trị cố định “1010” mức tích cực cao
- B. Bộ đếm tiến 4 bit đầu ra Q [3:0] hoạt động với sườn dương clock, nạp đồng bộ giá trị cố định “1010” mức tích cực cao
- C. Bộ đếm tiến 4 bit đầu ra Q [3:0] hoạt động với sườn dương clock, nạp đồng bộ giá trị cố định “1010” mức tích cực thấp
- D. Bộ đếm tiến 4 bit đầu ra Q [3:0] hoạt động với sườn dương clock, nạp không đồng bộ giá trị cố định “1010” mức tích cực cao

32. Đoạn mô tả kiến trúc nào mô tả cho mô hình bộ đếm thuận/ngược 4 bit có xóa không đồng bộ, có mô tả thực thể như sau:

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity counter is
    port( C, CLR, up_down : in std_logic; -- C - clock
          Q : out std_logic_vector(3 downto 0));
end counter;
```

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>A.</p> <pre>architecture archi of counter is     signal tmp: std_logic_vector(3                                 downto 0); begin     process (C, CLR)     begin         if (CLR='1') then             tmp &lt;= "0000";         elsif (C'event and C='1') then             if (up_down='1') then                 tmp &lt;= tmp + 1;             else tmp &lt;= tmp - 1;             end if;         end if;     end process;     Q &lt;= tmp; end archi;</pre> | <p>B.</p> <pre>architecture archi of counter is     signal tmp: std_logic_vector(3                                 downto 0); begin     process (C)     begin         if (CLR='1') then             tmp &lt;= "0000";         elsif (C'event and C='1') then             if (up_down='1') then                 tmp &lt;= tmp + 1;             else tmp &lt;= tmp - 1;             end if;         end if;     end process;     Q &lt;= tmp; end archi;</pre>      |
| <p>C.</p> <pre>architecture archi of counter is begin     process (C, CLR)     begin         if (CLR='1') then             Q &lt;= "0000";         elsif (C'event and C='1') then             if (up_down='1') then                 Q &lt;= Q + 1;             else tmp &lt;= tmp - 1;             end if;         end if;     end process; end archi;</pre>                                                                                                      | <p>D.</p> <pre>architecture archi of counter is     signal tmp: std_logic_vector(3                                 downto 0); begin     process (C, CLR)     begin         if (CLR='1') then             tmp &lt;= "1111";         elsif (C'event and C='1') then             if (up_down='1') then                 tmp &lt;= tmp + 1;             else tmp &lt;= tmp - 1;             end if;         end if;     end process;     Q &lt;= tmp; end archi;</pre> |

33. Đoạn mô tả nào mô tả cho trigger JK sau:

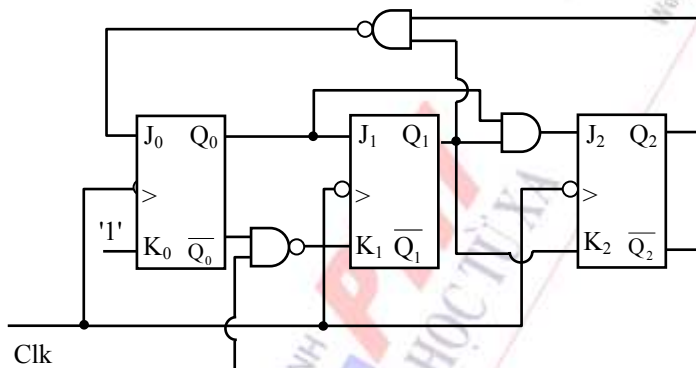


|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>A.</p> <pre>entity JKFF is     Port (J,K,Clk:in std_logic;           Q, notQ:out std_logic); end JKFF; architecture Behavioral of JKFF is signal Qtemp: std_logic; signal JK:std_logic_vector(0 to 1); begin     JK&lt;=(J,K);     process (Clk)     begin         if (Clk'event and Clk='1') then             case JK is                 when "00" =&gt; Null;                 when "01" =&gt; Qtemp&lt;='0';                 when "10" =&gt; Qtemp&lt;='1';                 when others=&gt;Qtemp&lt;=not Qtemp;             end case;         end if;     end process;     Q&lt;=Qtemp;     notQ&lt;=not Qtemp; end Behavioral;</pre> | <p>C.</p> <pre>entity JKFF is     Port (J,K,Clk:in std_logic;           Q, notQ:out std_logic); end JKFF; architecture Behavioral of JKFF is signal Qtemp: std_logic; signal JK:std_logic_vector(0 to 1); begin     JK&lt;=(J,K);     process (Clk)     begin         if (Clk'event and Clk='0') then             case JK is                 when "00" =&gt; Null;                 when "01" =&gt; Qtemp&lt;='0';                 when "10" =&gt; Qtemp&lt;='1';                 when others=&gt;Qtemp&lt;=not Qtemp;             end case;         end if;     end process;     Q&lt;=Qtemp;     notQ&lt;=not Qtemp; end Behavioral;</pre> |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



|                                                                                                                                                                                                                                                                                                     |                                                                                                                                                                                                                                                                                                     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>B.</b></p> <pre> entity JKFF is   Port (J,K,Clk:in std_logic;         Q, notQ:out std_logic); end JKFF; architecture Behavioral of JKFF is begin   process (Clk)   begin     if (Clk'event and Clk='1')     then       Q&lt;=J; notQ&lt;=K;     end if;   end process; end Behavioral; </pre> | <p><b>D.</b></p> <pre> entity JKFF is   Port (J,K,Clk:in std_logic;         Q, notQ:out std_logic); end JKFF; architecture Behavioral of JKFF is begin   process (Clk)   begin     if (Clk'event and Clk='0')     then       Q&lt;=J; notQ&lt;=K;     end if;   end process; end Behavioral; </pre> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

34. Đoạn mô tả nào mô tả đúng cho mạch sau theo mô hình RTL:



|                                                                                                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>A.</b></p> <pre> architecture Behavioral of cau33 is begin   notQ&lt;=not Q;   J(0)&lt;=Q(1) nand Q(2); K(0)&lt;='1';   J(1)&lt;=Q(0); K(1)&lt;= notQ(0) nand notQ(2);   J(2)&lt;=Q(1) and Q(0); K(2)&lt;=Q(1); end Behavioral; </pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

B.

```
architecture Behavioral of cau33 is
  signal Clk: std_logic;
  signal J,K,Q,notQ: std_logic_vector(0 to 2);
  signal JK0,JK1,JK2: std_logic_vector(0 to 1);
begin
  JK0 <=(J(0),K(0));JK1 <=(J(1),K(1));
  JK2 <=(J(2),K(2));
  notQ<=not Q;
  J(0)<=Q(1) nand Q(2); K(0)<='1';
  J(1)<=Q(0);      K(1)<= notQ(0) nand notQ(2);
  J(2)<=Q(1) and Q(0); K(2)<=Q(1);
end Behavioral;
```

C.

```
architecture Behavioral of cau33 is
begin
  process(Clk)
  begin
    if(Clk'event and Clk='1') then
      case JK0 is
        when "00" => Null;
        when "01" => Q(0)<='0';
        when "10" => Q(0)<='1';
        when others => Q(0)<= not Q(0);
      end case;
    end if;
  end process;
end Behavioral;
```

```

D.
architecture Behavioral of cau33 is
    signal Clk: std_logic;
    signal J,K,Q,notQ: std_logic_vector(0 to 2);
    signal JK0,JK1,JK2: std_logic_vector(0 to 1);
begin
    JK0 <=(J(0),K(0));JK1 <=(J(1),K(1));
    JK2 <=(J(2),K(2));
    process(Clk)
        begin
            if(Clk'event and Clk='0') then
                case JK0 is
                    when "00" => Null;
                    when "01" => Q(0)<='0';
                    when "10" => Q(0)<='1';
                    when others => Q(0)<= not Q(0);
                end case;
                case JK1 is
                    when "00" => Null;
                    when "01" => Q(1)<='0';
                    when "10" => Q(1)<='1';
                    when others => Q(1)<= not Q(1);
                end case;
                case JK2 is
                    when "00" => Null;
                    when "01" => Q(2)<='0';
                    when "10" => Q(2)<='1';
                    when others => Q(2)<= not Q(2);
                end case;
            end if;
        end process;
        notQ<=not Q;
        J(0)<=Q(1) nand Q(2); K(0)<='1';
        J(1)<=Q(0); K(1)<= notQ(0) nand notQ(2);
        J(2)<=Q(1) and Q(0); K(2)<=Q(1);
    end Behavioral;

```

35. Đoạn mô tả nào mô tả đúng cho mạch giải mã BCD sang mã 7 segment.

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>A.</b></p> <pre>entity BCDto7seg is Port( BCD:in       std_logic_vector(3 downto 0));       Seg : out       std_logic_vector(6 downto 0)); end BCDto7seg; architecture Beh of BCDto7seg is begin with BCD select     --abcdefg"     Seg&lt;= "1111110" when x"0",           "0110000" when x"1",           "1101101" when x"2",           "1111001" when x"3",           "0110011" when x"4",           "1011011" when x"5",           "1011111" when x"6",           "1110000" when x"7",           "1111111" when x"8",           "1111011" when x"9",           "0000000" when others; end Beh;</pre> | <p><b>C.</b></p> <pre>entity BCDto7seg is Port ( BCD:in       std_logic_vector(3 downto 0));       Seg : out       std_logic_vector(6 downto 0)); end BCDto7seg; architecture Beh of BCDto7seg is begin with BCD select     --abcdefg"     Seg&lt;= "1111110" when x"0",           "0110000" when x"1",           "1101101" when x"2",           "1111001" when x"3",           "0110011" when x"4",           "1011011" when x"5",           "1011111" when x"6",           "1111111" when x"7",           "1111111" when x"8",           "1111111" when x"9",           "0000000" when others; end Beh;</pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>B.</b></p> <pre> entity BCDto7seg is Port ( BCD:in       std_logic_vector(3 downto 0));       Seg : out       std_logic_vector(6 downto 0)); end BCDto7seg; architecture Beh of BCDto7seg is begin with BCD select   --abcdefg"   Seg&lt;= "1111110" when x"0",         "0110000" when x"1",         "1101101" when x"2",         "1111001" when x"3",         "0000000" when others; end Beh; </pre> | <p><b>D.</b></p> <pre> entity BCDto7seg is Port ( BCD:in       std_logic_vector(3 downto 0));       Seg : out       std_logic_vector(6 downto 0)); end BCDto7seg; architecture Beh of BCDto7seg is begin with BCD select   --abcdefg"   Seg&lt;= "1011111" when x"6",         "1110000" when x"7",         "1111111" when x"8",         "1111011" when x"9",         "0000000" when others; end Beh; </pre> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|



36. Đoạn mô tả nào mô tả đúng cho mạch hợp kênh 8 vào – 1 ra:

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>A.</b></p> <pre>entity Mux is end Mux; architecture Behavioral of Mux is     signal I :         std_logic_vector(8 downto 0);     signal SEL:         std_logic_vector(4 downto 0);     signal Y :std_logic; begin     with SEL select         --abcdefg"     Y &lt;=  I(0) when "0000",         I(1) when "0001",         I(2) when "0010",         I(3) when "0011",         I(4) when "0100",         I(5) when "0101",         I(6) when others; end Behavioral;</pre> | <p><b>C.</b></p> <pre>architecture Behavioral of Mux is     signal I :         std_logic_vector(7 downto 0);     signal SEL:         std_logic_vector(2 downto 0);     signal Y :std_logic; begin     process     begin         case SEL is             when "000" =&gt;                 Y&lt;=I(0);             when "001" =&gt;                 Y&lt;=I(1);             when "010" =&gt;                 Y&lt;=I(2);             when "011" =&gt;                 Y&lt;=I(3);             when "100" =&gt;                 Y&lt;=I(4);             when "101" =&gt;                 Y&lt;=I(5);             when "110" =&gt;                 Y&lt;=I(6);             when others =&gt;                 Y&lt;=I(7);         end case;     end process; end Behavioral;</pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>B.</b></p> <pre> entity Mux is end Mux; architecture Behavioral of Mux is     signal I :         std_logic_vector(7 downto 0);     signal SEL:         std_logic_vector(2 downto 0);     signal Y :std_logic; begin     with SEL select         --abcdefg"     Y &lt;=  I(0) when "000",         I(1) when "001",         I(2) when "010",         I(3) when "011",         I(4) when "100",         I(5) when "101",         I(6) when "110",         I(7) when others; end Behavioral; </pre> | <p><b>D.</b></p> <pre> architecture Behavioral of Mux is     signal I :         std_logic_vector(7 downto 0);     signal SEL:         std_logic_vector(2 downto 0);     signal Y : std_logic; begin     process(I)     begin         case SEL is             when "000" =&gt;                 Y&lt;=I(0);             when "001" =&gt;                 Y&lt;=I(1);             when "010" =&gt;                 Y&lt;=I(2);             when "011" =&gt;                 Y&lt;=I(3);             when "100" =&gt;                 Y&lt;=I(4);             when "101" =&gt;                 Y&lt;=I(5);             when "110" =&gt;                 Y&lt;=I(6);             when others =&gt;                 Y&lt;=I(7);         end case;     end process; end Behavioral; </pre> |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## ĐÁP ÁN VÀ HƯỚNG DẪN TRẢ LỜI

### CHƯƠNG 1

1. Bit là số nhị phân có một chữ số. 1byte = 8 bit.
2. b
3. c
4. a
5. d
6. a

### CHƯƠNG 2

Bài 1.

1. a
2. b

Bài 2.2

1. c
2. b

Bài 2.3

d

Bài 2.4

d. Do đều bằng  $A+AB$

Bài 2.5

- Mức logic và phân tích
- Trễ truyền lan và phân tích
- Công suất tiêu thụ và phân tích
- Hệ số ghép tải và phân tích
- Độ phòng vệ nhiễu và phân tích
- Một số tham số khác

Bài 2.6

c

Bài 2.7

c

Bài 2.8

- Nêu được khái niệm về tối ưu hoá mạch điện các họ cổng

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Km10 Đường Nguyễn Trãi, Hà Đông-Hà Tây  
Tel: (04) 5541221; Fax: (04) 5540587

Website: <http://www.ct-ptit.edu.vn>; E-mail: [dlhkc@ptit.edu.vn](mailto:dlhkc@ptit.edu.vn)

CHƯƠNG TRÌNH PTIT  
ĐẠI HỌC TẠO ĐẠI HỌC TỪ XA



- Công cụ tối ưu hoá
- Đưa ra ví dụ và phân tích hiệu quả kỹ thuật, kinh tế của việc tối ưu hoá

Bài 2.10

a

Bài 2.11

d

Bài 2.12

c

### CHƯƠNG 3

- |      |      |
|------|------|
| 1.d  | 2.a  |
| 3.d  | 4.b  |
| 5.c  | 6.a  |
| 7.b  | 8.c  |
| 9.d  | 10.b |
| 11.a | 12.d |
| 13.d | 14.a |

### CHƯƠNG 4

- |      |      |
|------|------|
| 1.a  | 2.d  |
| 3.c  | 4.c  |
| 5.c  | 6.d  |
| 7.b  | 8.c  |
| 9.a  | 10.c |
| 11.a | 12.d |
| 13.c | 14.a |
| 15.b | 16.b |
| 17.a | 18.b |
| 19.c | 20.d |

### CHƯƠNG 5

- |     |     |
|-----|-----|
| 1.a | 2.c |
|-----|-----|

- |      |                    |
|------|--------------------|
| 3.c  | 4.b                |
| 5.d  | 6.a                |
| 7.c  | 8.d                |
| 9.d  | 10.c               |
| 11.a | 12.b               |
| 13.d | 14.c               |
| 15.c | 16.a               |
| 17.d | 18.b               |
| 19.a | 20.a               |
| 21.b | 22.d               |
| 23.b | 24.a               |
| 25.b | 26.c               |
| 27.c | 28.d               |
| 29.c | 30.a               |
| 31.b | 32.d               |
| 33.c | 34.a               |
| 35.c | 36. Xem ví dụ phần |

5.4.1.2

37. Xem ví dụ phần

5.4.1.2

39.b 40.a

## CHƯƠNG 6

- |     |      |
|-----|------|
| 1.c | 2.a  |
| 3.b | 4.d  |
| 5.b | 6.b  |
| 7.c | 8.c  |
| 9.a | 10.d |

## CHƯƠNG 7

- |     |     |
|-----|-----|
| 1.a | 2.c |
| 3.c | 4.b |
| 5.c | 6.a |

7.b

8.a

9.c

10.c

## CHƯƠNG 8 VÀ CHƯƠNG 9

1.C

2.D

3.C

4.D

5.C

6.D

7.B

8.D

9.D

10.C

11.B

12.B

13.D

14.D

15.B

16.A

17.C

18.D

19.A

20.D

21.D

22.A

23.C

24.B

25.D

26.A

27.B

28.C

29.C

30.C

31.B

32.A

33.C

34.D

35.A

36.B

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

Kinh Doanh Nguyễn Trãi, Hà Đông-Hà Tây  
Tel: (04) 5541221; Fax: (04) 5540587  
Website: <http://hcm.vtc.edu.vn>; E-mail: [dhk@vtc.edu.vn](mailto:dhk@vtc.edu.vn)

CHƯƠNG TRÌNH PTIT  
ĐẠI HỌC TẠO ĐẠI HỌC TỪ XA

## TÀI LIỆU THAM KHẢO

1. *Giáo trình Kỹ thuật số* - Trần Văn Minh, NXB Bưu điện 2002.
2. *Cơ sở kỹ thuật điện tử số*, Đại học Thanh Hoa, Bắc Kinh, NXB Giáo dục 1996 .
3. *Kỹ thuật số*, Nguyễn Thúy Vân, NXB Khoa học và kỹ thuật 1994.
4. *Toán logic và kỹ thuật số*, Nguyễn Nam Quân - Khoa ĐHTC xuất bản - 1974
5. *Lý thuyết mạch logic và Kỹ thuật số*, Nguyễn Xuân Quỳnh - NXB Bưu điện - 1984
6. *Fundamentals of logic design*, fourth edition, Charles H. Roth, Prentice Hall 1991.
7. *Digital engineering design*, Richard F.Tinder, Prentice Hall 1991 .
8. *Digital design principles and practices*, John F.Wakerly, Prentice Hall 1990.
9. *VHDL for Programmable Logic* by Kevin Skahill, Addison Wesley, 1996
10. *The Designer's Guide to VHDL* by Peter Ashenden, Morgan Kaufmann, 1996.
11. *Analysis and Design of Digital Systems with VHDL* by Dewey A., PWS Publishing, 1993.

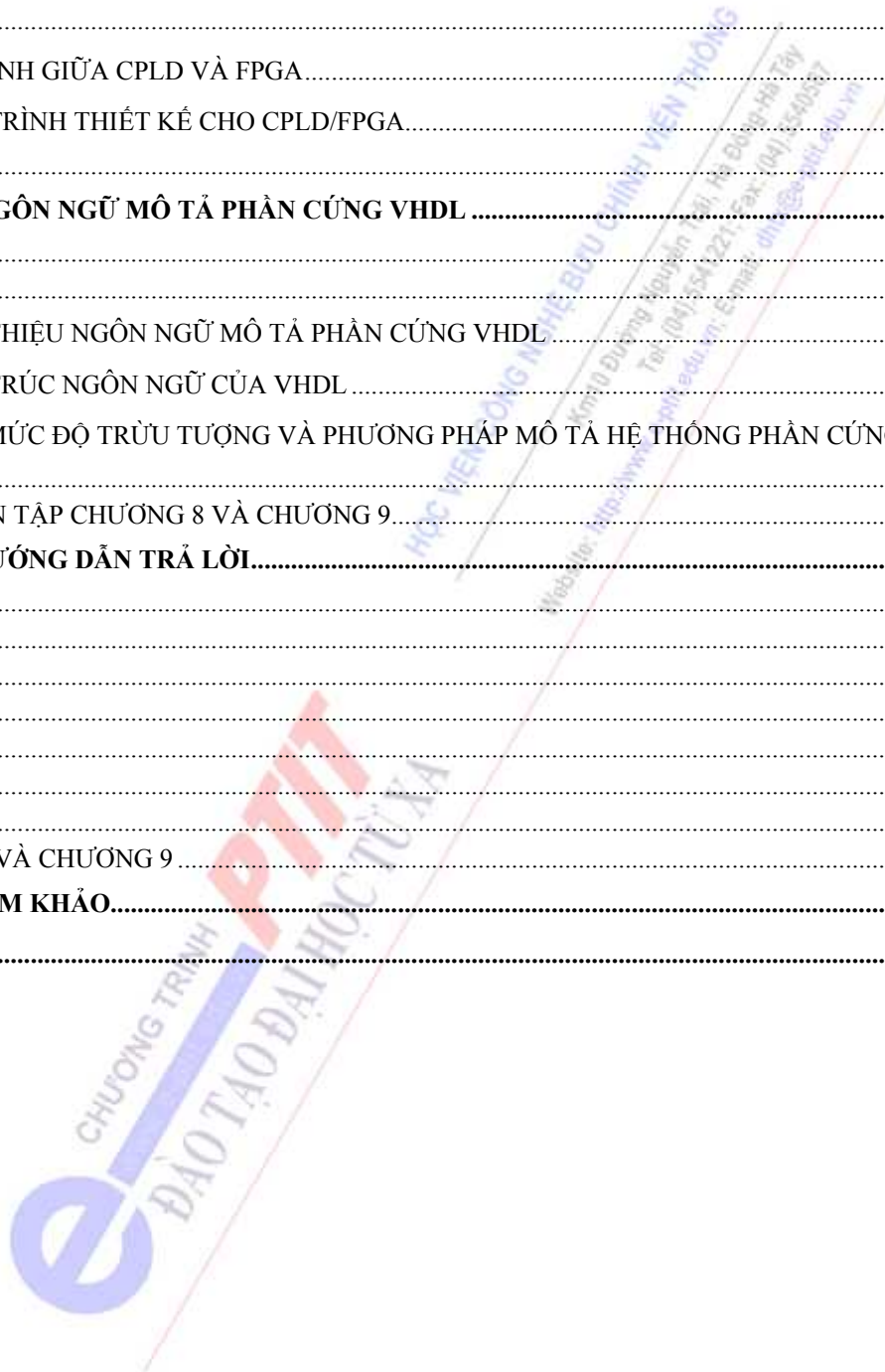


# MỤC LỤC

|                                                                     |           |
|---------------------------------------------------------------------|-----------|
| <b>LỜI GIỚI THIỆU .....</b>                                         | <b>1</b>  |
| <b>CHƯƠNG 1: HỆ ĐẾM .....</b>                                       | <b>2</b>  |
| GIỚI THIỆU .....                                                    | 2         |
| NỘI DUNG.....                                                       | 2         |
| 1.1. BIỂU DIỄN SỐ.....                                              | 2         |
| 1.2. CHUYỂN ĐỔI CƠ SỐ GIỮA CÁC HỆ ĐẾM .....                         | 6         |
| 1.3. SỐ NHỊ PHÂN CÓ DẤU.....                                        | 8         |
| 1.4. DẤU PHẪY ĐỘNG.....                                             | 9         |
| TÓM TẮT.....                                                        | 9         |
| CÂU HỎI ÔN TẬP.....                                                 | 10        |
| <b>CHƯƠNG 2: ĐẠI SỐ BOOLE VÀ CÁC PHƯƠNG PHÁP BIỂU DIỄN HÀM.....</b> | <b>11</b> |
| GIỚI THIỆU CHUNG.....                                               | 11        |
| NỘI DUNG.....                                                       | 12        |
| 2.1 ĐẠI SỐ BOOLE.....                                               | 12        |
| 2.2 CÁC PHƯƠNG PHÁP BIỂU DIỄN HÀM BOOLE .....                       | 12        |
| 2.3 CÁC PHƯƠNG PHÁP RÚT GỌN HÀM.....                                | 14        |
| 2.4 CÔNG LOGIC VÀ CÁC THAM SỐ CHÍNH .....                           | 16        |
| TÓM TẮT.....                                                        | 26        |
| CÂU HỎI ÔN TẬP.....                                                 | 26        |
| <b>CHƯƠNG 3: CÔNG LOGIC TTL VÀ CMOS.....</b>                        | <b>29</b> |
| GIỚI THIỆU .....                                                    | 29        |
| NỘI DUNG.....                                                       | 30        |
| 3.1. CÁC HỘ CÔNG LOGIC .....                                        | 30        |
| 3.2. GIAO TIẾP GIỮA CÁC CÔNG LOGIC CƠ BẢN TTL-CMOS VÀ CMOS-TTL..... | 40        |
| TÓM TẮT.....                                                        | 43        |
| CÂU HỎI ÔN TẬP.....                                                 | 43        |
| <b>CHƯƠNG 4: MẠCH LOGIC TỔ HỢP .....</b>                            | <b>48</b> |
| GIỚI THIỆU CHUNG.....                                               | 48        |
| NỘI DUNG.....                                                       | 49        |
| 4.1 KHÁI NIỆM CHUNG.....                                            | 49        |
| 4.2 PHÂN TÍCH MẠCH LOGIC TỔ HỢP .....                               | 50        |
| 4.3 THIẾT KẾ MẠCH LOGIC TỔ HỢP.....                                 | 50        |
| 4.4 HAZARD TRONG MẠCH TỔ HỢP .....                                  | 51        |
| 4.5. MẠCH MÃ HOÁ VÀ GIẢI MÃ .....                                   | 59        |
| 4.6 BỘ HỢP KÊNH VÀ PHÂN KÊNH.....                                   | 64        |
| 4.7. MẠCH CỘNG.....                                                 | 66        |

|                                                                   |            |
|-------------------------------------------------------------------|------------|
| 4.8. MẠCH SO SÁNH .....                                           | 67         |
| 4.9. MẠCH TẠO VÀ KIỂM TRA CHẶN LỀ .....                           | 68         |
| 4.10. ĐƠN VỊ SỐ HỌC VÀ LOGIC (ALU) .....                          | 70         |
| TÓM TẮT .....                                                     | 70         |
| CÂU HỎI ÔN TẬP .....                                              | 71         |
| <b>CHƯƠNG 5: MẠCH LOGIC TUẦN TỰ .....</b>                         | <b>75</b>  |
| GIỚI THIỆU .....                                                  | 75         |
| NỘI DUNG .....                                                    | 75         |
| 5.1. KHÁI NIỆM CHUNG VÀ MÔ HÌNH TOÁN HỌC .....                    | 75         |
| 5.2. PHẦN TỬ NHỚ CỦA MẠCH TUẦN TỰ .....                           | 76         |
| 5.3. PHƯƠNG PHÁP MÔ TẢ MẠCH TUẦN TỰ .....                         | 81         |
| 5.4. CÁC BƯỚC THIẾT KẾ MẠCH TUẦN TỰ .....                         | 83         |
| 5.5. MẠCH TUẦN TỰ ĐỒNG BỘ .....                                   | 90         |
| 5.6. MẠCH TUẦN TỰ KHÔNG ĐỒNG BỘ .....                             | 98         |
| 5.7. HIỆN TƯỢNG CHU KỶ VÀ CHẠY ĐUA TRONG MẠCH KHÔNG ĐỒNG BỘ ..... | 104        |
| 5.8. MỘT SỐ MẠCH TUẦN TỰ THÔNG DỤNG .....                         | 108        |
| TÓM TẮT .....                                                     | 116        |
| CÂU HỎI ÔN TẬP CHƯƠNG 5 .....                                     | 116        |
| <b>CHƯƠNG 6: MẠCH PHÁT XUNG VÀ TẠO DẠNG XUNG .....</b>            | <b>125</b> |
| GIỚI THIỆU .....                                                  | 125        |
| NỘI DUNG .....                                                    | 126        |
| 6.1. MẠCH PHÁT XUNG .....                                         | 126        |
| 6.2. TRIGƠ SCHMIT .....                                           | 129        |
| 6.3. MẠCH ĐA HÀI ĐỢI .....                                        | 130        |
| 6.4. IC ĐỊNH THỜI .....                                           | 134        |
| TÓM TẮT .....                                                     | 137        |
| CÂU HỎI ÔN TẬP .....                                              | 137        |
| <b>CHƯƠNG 7: BỘ NHỚ BÁN DẪN .....</b>                             | <b>141</b> |
| GIỚI THIỆU .....                                                  | 141        |
| NỘI DUNG .....                                                    | 141        |
| 7.1. KHÁI NIỆM CHUNG .....                                        | 141        |
| 7.2. DRAM .....                                                   | 144        |
| 7.3. SRAM .....                                                   | 145        |
| 7.3. BỘ NHỚ CỐ ĐỊNH - ROM .....                                   | 146        |
| 7.4. BỘ NHỚ BÁN CỐ ĐỊNH .....                                     | 147        |
| 7.5. MỞ RỘNG DUNG LƯỢNG BỘ NHỚ .....                              | 151        |
| TÓM TẮT .....                                                     | 152        |
| CÂU HỎI ÔN TẬP .....                                              | 153        |
| <b>CHƯƠNG 8: LOGIC LẬP TRÌNH (PLD) .....</b>                      | <b>155</b> |
| GIỚI THIỆU .....                                                  | 155        |

|                                                                           |            |
|---------------------------------------------------------------------------|------------|
| NỘI DUNG.....                                                             | 156        |
| 8.1. GIỚI THIỆU CHUNG VỀ LOGIC KHẢ TRÌNH (PLD).....                       | 156        |
| 8.2. SPLD.....                                                            | 157        |
| 8.3. CPLD (Complex PLD).....                                              | 157        |
| 8.4. FPGA.....                                                            | 159        |
| 8.5. SO SÁNH GIỮA CPLD VÀ FPGA.....                                       | 161        |
| 8.6. QUY TRÌNH THIẾT KẾ CHO CPLD/FPGA.....                                | 161        |
| TÓM TẮT.....                                                              | 168        |
| <b>CHƯƠNG 9: NGÔN NGỮ MÔ TẢ PHẦN CỨNG VHDL.....</b>                       | <b>169</b> |
| GIỚI THIỆU.....                                                           | 169        |
| NỘI DUNG.....                                                             | 170        |
| 9.1. GIỚI THIỆU NGÔN NGỮ MÔ TẢ PHẦN CỨNG VHDL.....                        | 170        |
| 9.2. CẤU TRÚC NGÔN NGỮ CỦA VHDL.....                                      | 171        |
| 9.3. CÁC MỨC ĐỘ TRỪU TƯỢNG VÀ PHƯƠNG PHÁP MÔ TẢ HỆ THỐNG PHẦN CỨNG SỐ.... | 199        |
| TÓM TẮT.....                                                              | 212        |
| CÂU HỎI ÔN TẬP CHƯƠNG 8 VÀ CHƯƠNG 9.....                                  | 213        |
| <b>ĐÁP ÁN VÀ HƯỚNG DẪN TRẢ LỜI.....</b>                                   | <b>236</b> |
| CHƯƠNG 1.....                                                             | 236        |
| CHƯƠNG 2.....                                                             | 236        |
| CHƯƠNG 3.....                                                             | 237        |
| CHƯƠNG 4.....                                                             | 237        |
| CHƯƠNG 5.....                                                             | 237        |
| CHƯƠNG 6.....                                                             | 238        |
| CHƯƠNG 7.....                                                             | 238        |
| CHƯƠNG 8 VÀ CHƯƠNG 9.....                                                 | 239        |
| <b>TÀI LIỆU THAM KHẢO.....</b>                                            | <b>240</b> |
| <b>MỤC LỤC.....</b>                                                       | <b>241</b> |



# ĐIỆN TỬ SỐ

Mã số : 497DTS210

Chịu trách nhiệm bản thảo

TRUNG TÂM ĐÀO TẠO BƯU CHÍNH VIỄN THÔNG 1



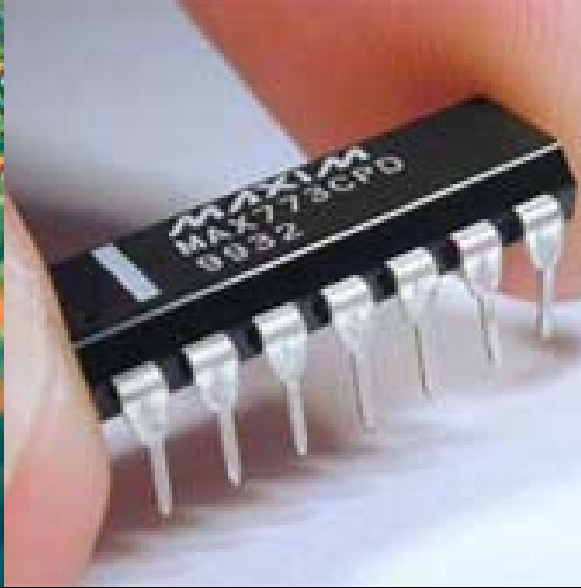
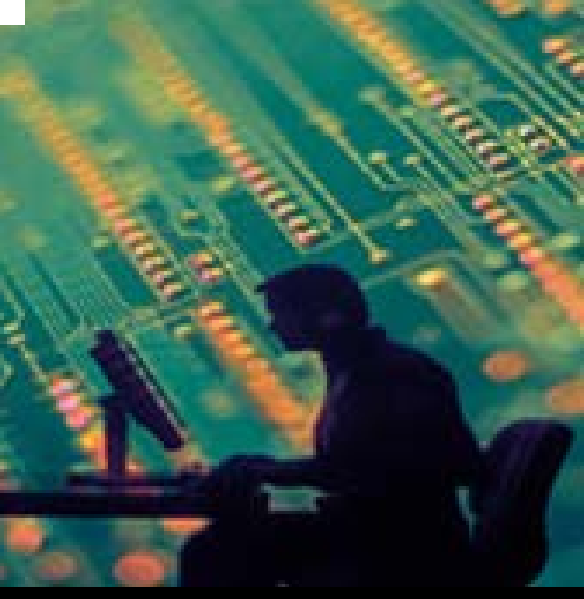
TRUNG TÂM ĐÀO TẠO BƯU CHÍNH VIỄN THÔNG  
Km10 Đường Nguyễn Trãi, Hà Đông-Hà Tây  
Tel: (04).5541221; Fax: (04).5540587  
Website: <http://www.ptth.edu.vn>; E-mail: [dhkx@ptth.edu.vn](mailto:dhkx@ptth.edu.vn)



A decorative border of palm trees surrounds the text. The border consists of a top row of 18 palm trees, a bottom row of 18 palm trees, and two vertical columns of 18 palm trees each on the left and right sides.

# Giáo trình

# ĐIỆN TỬ SỐ



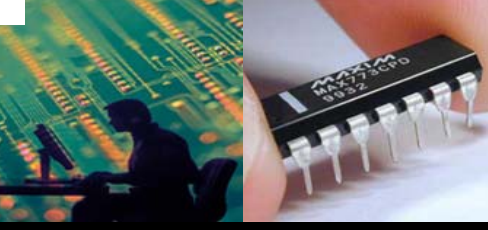
# ĐIỆN TỬ SỐ

**Digital Electronics**

Bộ môn Kỹ thuật máy tính

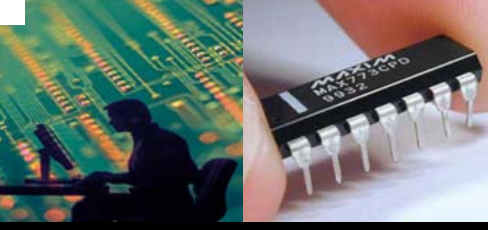
Khoa Điện Điện Tử

Trường ĐH Giao Thông Vận Tải

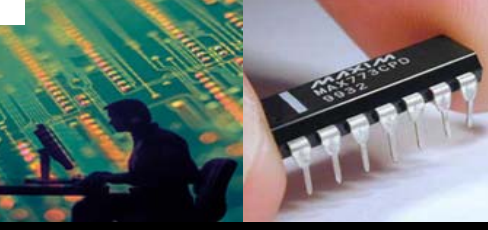


[nguyenvanbientbd47@gmail.com](mailto:nguyenvanbientbd47@gmail.com)



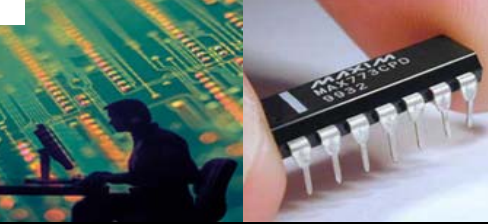


- Cung cấp các kiến thức cơ bản về:
  - Cấu tạo
  - Nguyên lý hoạt động
  - Ứng dụngcủa các mạch số (mạch logic, IC, chip...)
- Trang bị nguyên lý
  - Phân tích
  - Thiết kếcác mạch số cơ bản
- Tạo cơ sở cho tiếp thu các kiến thức chuyên ngành



- **Introductory Digital Electronics** - *Nigel P. Cook* - Prentice Hall, 1998
- **Digital Systems - Principles and Applications** - *Tocci & Widmer* - Prentice Hall, 1998
- <http://ktmt.shorturl.com>





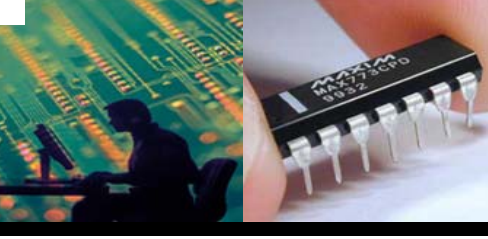
- Tổng thời lượng: 60 tiết
  - Lý thuyết: 45 tiết, tại giảng đường
  - Thực hành: 15 tiết.

Mô phỏng một số mạch điện tử số trong giáo trình sử dụng phần mềm Multisim v8.0

- Hướng dẫn thực hành tại phòng máy
  - C1-325, Cô Nguyệt Bộ môn KTMT liên hệ
- Nộp báo cáo thực hành kèm bài thi
- Không có báo cáo thực hành => 0 điểm.



- Chương 1. Giới thiệu về Điện tử số
- Chương 2. Các hàm logic
- Chương 3. Các phần tử logic cơ bản
- Chương 4. Hệ tổ hợp
- Chương 5. Hệ dãy

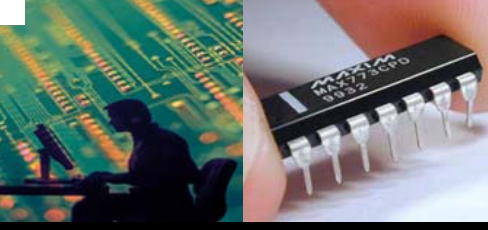


# Chương 1

## GIỚI THIỆU VỀ ĐIỆN TỬ SỐ

Bộ môn Kỹ thuật Máy tính, Khoa Công nghệ Thông tin  
Trường Đại học Bách Khoa Hà Nội





# Giới thiệu về Điện tử số

## Điện tử số



# Giới thiệu về Điện tử số (tiếp)

- Hệ thống điện tử, thiết bị điện tử

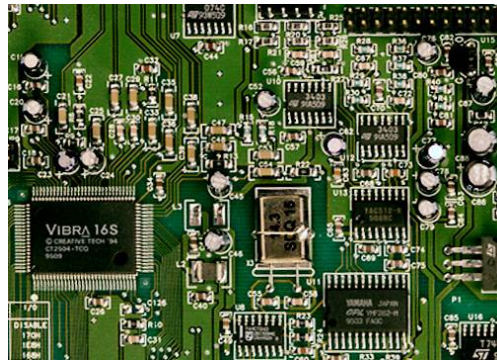
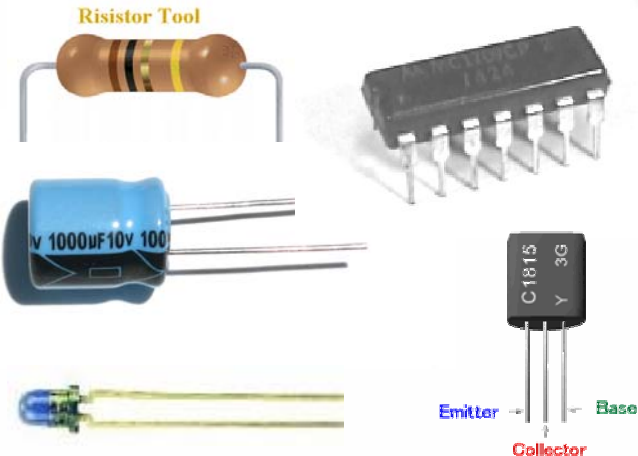
Các  
linh kiện  
điện, điện tử  
(component)

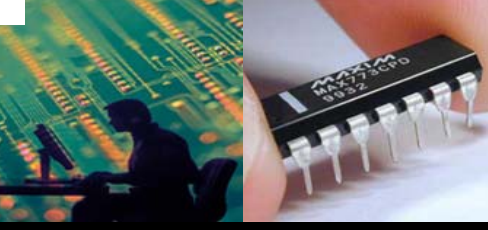


Các  
mạch  
điện tử  
(circuit)



Các  
thiết bị,  
hệ thống  
điện tử  
(equipment,  
system)

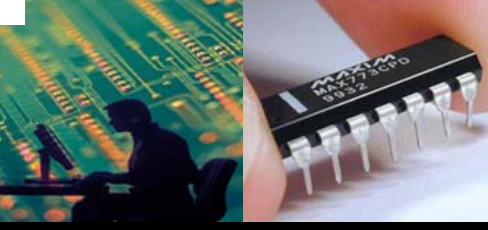




# Giới thiệu về Điện tử số (tiếp)

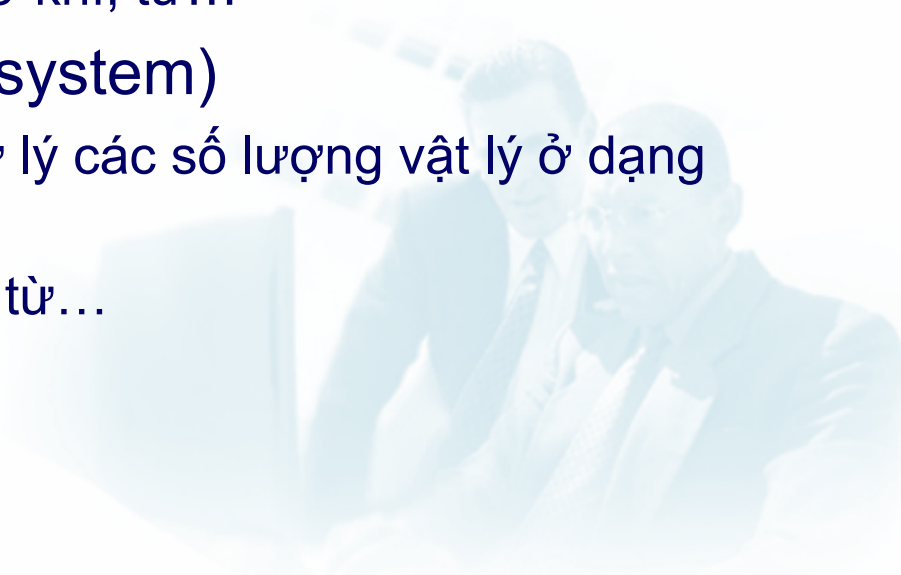
## ■ Số và tương tự:

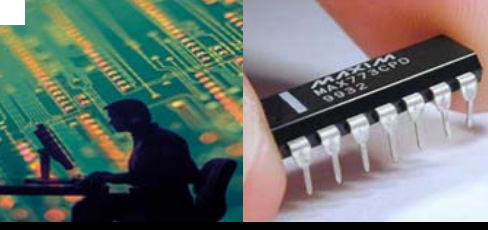
- Trong khoa học, công nghệ hay cuộc sống đời thường, ta thường xuyên phải tiếp xúc với số lượng
- Số lượng có thể đo, quản lý, ghi chép, tính toán nhằm giúp cho các xử lý, ước đoán phức tạp hơn
- Có 2 cách biểu diễn số lượng:
  - Dạng tương tự (Analog)
  - Dạng số (Digital)
- Dạng tương tự:
  - VD: Nhiệt độ, tốc độ, điện thế của đầu ra micro...
  - Là dạng biểu diễn với sự biến đổi liên tục của các giá trị (continuous)
- Dạng số:
  - VD: Thời gian hiện trên đồng hồ điện tử
  - Là dạng biểu diễn trong đó các giá trị thay đổi từng nấc rời rạc (discrete)



# Giới thiệu về Điện tử số (tiếp)

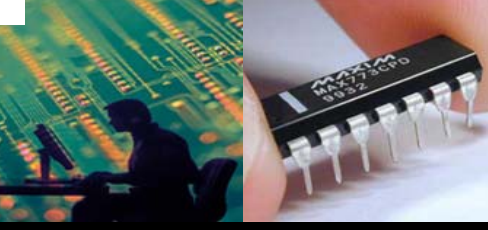
- Hệ thống số và tương tự:
  - Hệ thống số (Digital system)
    - Là tổ hợp các thiết bị được thiết kế để xử lý các thông tin logic hoặc các số lượng vật lý dưới dạng số
    - VD: Máy vi tính, máy tính, các thiết bị hình ảnh âm thanh số, hệ thống điện thoại...
    - Ứng dụng: lĩnh vực điện tử, cơ khí, từ...
  - Hệ thống tương tự (Analog system)
    - Chứa các thiết bị cho phép xử lý các số lượng vật lý ở dạng tương tự
    - VD: Hệ thống âm-ly, ghi băng từ...





# Giới thiệu về Điện tử số (tiếp)

- Công nghệ số - ưu, nhược điểm so với tương tự
- Dùng công nghệ số để thực hiện các thao tác của giải pháp tương tự**
  - Ưu điểm của công nghệ số:
    - Các hệ thống số dễ thiết kế hơn:
      - Không cần giá trị chính xác U, I, chỉ cần khoảng cách mức cao thấp
    - Lưu trữ thông tin dễ
      - Có các mạch chốt có thể giữ thông tin lâu tùy ý
    - Độ chính xác cao hơn
      - Việc nâng từ độ chính xác 3 chữ số lên 4 chữ số đơn giản chỉ cần lắp thêm mạch
      - Ở hệ tương tự, lắp thêm mạch sẽ ảnh hưởng U, I và thêm nhiễu
    - Các xử lý có thể lập trình được
    - Ít bị ảnh hưởng bởi nhiễu
    - Có thể chế tạo nhiều mạch số trong các chip

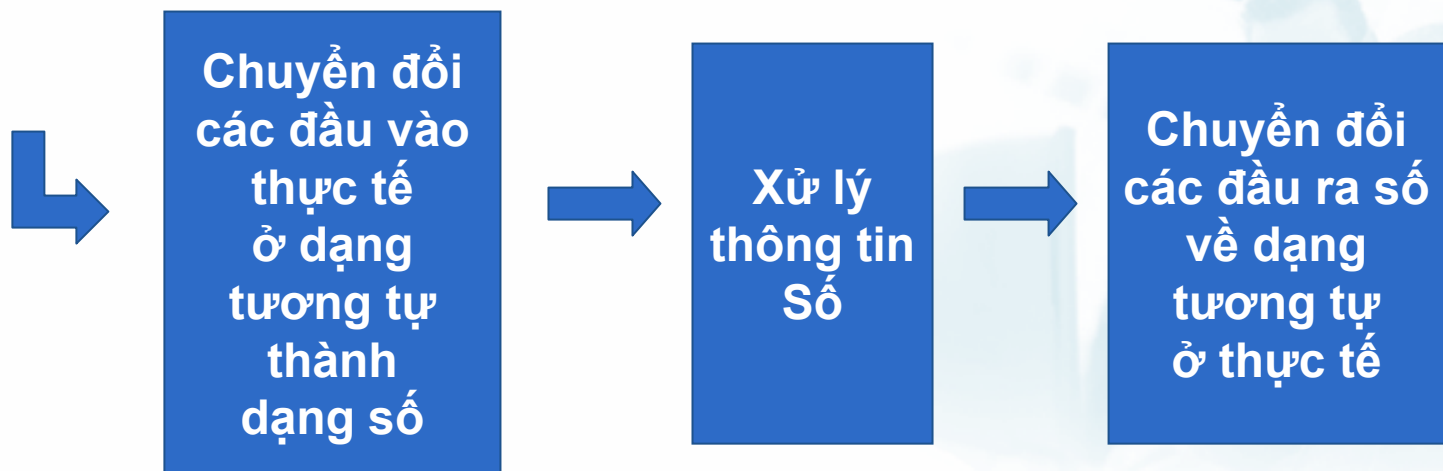


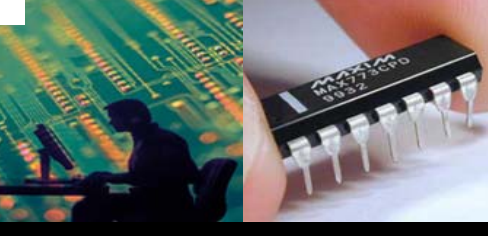
# Giới thiệu về Điện tử số (tiếp)

- Công nghệ số - ưu, nhược điểm so với tương tự
  - Hạn chế:

## **Thế giới thực chủ yếu là tương tự**

- Các số lượng vật lý trong thực tế, tự nhiên chủ yếu là ở dạng tương tự.
- VD: nhiệt độ, áp suất, vị trí, vận tốc, độ rắn, tốc độ dòng chảy...

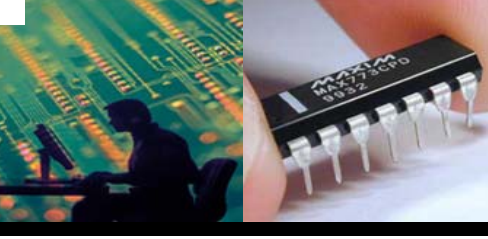




# Giới thiệu về Điện tử số (tiếp)

**Sự kết hợp của  
công nghệ số và tương tự!**





# Chương 2

## CÁC HÀM LOGIC

Bộ môn Kỹ thuật Máy tính, Khoa Công nghệ Thông tin  
Trường Đại học Bách Khoa Hà Nội



## 2.1. Giới thiệu

## 2.2. Đại số Boole

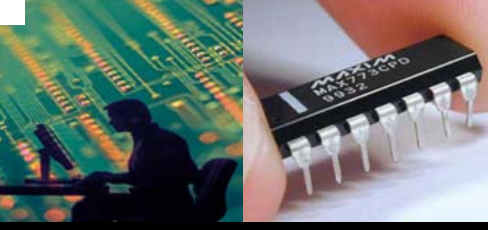
## 2.2. Biểu diễn các hàm logic dưới dạng chính quy

## 2.3. Tối thiểu hóa các hàm logic

- Mạch logic (mạch số) hoạt động dựa trên chế độ nhị phân:
  - Điện thế ở đầu vào, đầu vào hoặc bằng 0, hoặc bằng 1
  - Với 0 hay 1 tượng trưng cho các khoảng điện thế được định nghĩa sẵn
  - VD:  $0 \rightarrow 0.8V$  : 0  
 $2.5 \rightarrow 5V$  : 1
- ↳ Cho phép ta sử dụng Đại số Boole như là một công cụ để phân tích và thiết kế các hệ thống số

## ▪ Đại số Boole:

- Do George Boole sáng lập vào thế kỷ 19
- Các hằng, biến và hàm chỉ nhận 1 trong 2 giá trị: 0 và 1
- Là công cụ toán học khá đơn giản cho phép mô tả mối liên hệ giữa các đầu ra của mạch logic với các đầu vào của nó dưới dạng biểu thức logic
- Là cơ sở lý thuyết, là công cụ cho phép nghiên cứu, mô tả, phân tích, thiết kế và xây dựng các hệ thống số, hệ thống logic, mạch số ngày nay.



- Các phần tử logic cơ bản:
  - Còn gọi là các cổng logic, mạch logic cơ bản
  - Là các khối cơ bản cấu thành nên các mạch logic và hệ thống số khác



- Mục tiêu của chương: sinh viên có thể
  - Tìm hiểu về Đại số Boole
  - Các phần tử logic cơ bản và hoạt động của chúng
  - Dùng Đại số Boole để mô tả và phân tích cách cấu thành các mạch logic phức tạp từ các phần tử logic cơ bản

2.1. Giới thiệu

**2.2. Đại số Boole**

2.2. Biểu diễn các hàm logic dưới dạng chính quy

2.3. Tối thiểu hóa các hàm logic

# 1. Các định nghĩa

- **Biến logic:** là 1 đại lượng có thể biểu diễn bằng 1 ký hiệu nào đó, về mặt giá trị chỉ lấy giá trị 0 hoặc 1.
- **Hàm logic:** là biểu diễn của nhóm các biến logic, liên hệ với nhau thông qua các phép toán logic, về mặt giá trị cũng lấy giá trị 0 hoặc 1.
- **Phép toán logic:** có 3 phép toán logic cơ bản:
  - Phép Và - "AND"
  - Phép Hoặc - "OR"
  - Phép Đảo - "NOT"

# Các định nghĩa (tiếp)

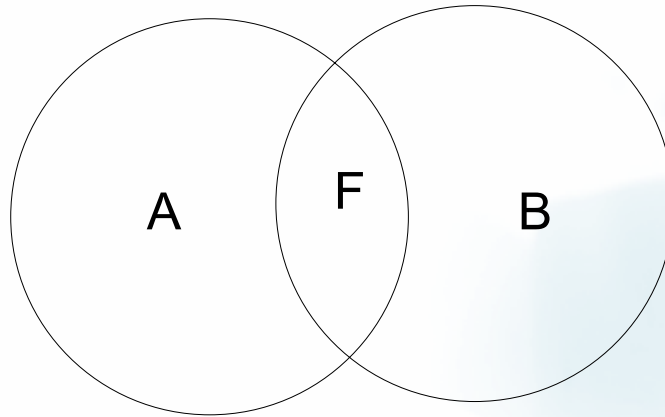
- Các giá trị 0, 1 không tượng trưng cho các con số thực mà tượng trưng cho trạng thái giá trị điện thế hay còn gọi là mức logic (logic level)
- Một số cách gọi khác của 2 mức logic:

| Mức logic 0        | Mức logic 1          |
|--------------------|----------------------|
| Sai (False)        | Đúng (True)          |
| Tắt (Off)          | Bật (On)             |
| Thấp (Low)         | Cao (High)           |
| Không (No)         | Có (Yes)             |
| (Ngắt) Open switch | (Đóng) Closed switch |



## 2. Biểu diễn biến và hàm logic

- Dùng biểu đồ Venn (O'le):
  - Mỗi biến logic chia không gian thành 2 không gian con.
  - Không gian con thứ nhất, biến nhận giá trị đúng (=1), không gian con thứ còn lại, biến nhận giá trị sai (=0).
  - VD:  $F = A \text{ AND } B$



# Biểu diễn biến và hàm logic (tiếp)

- Dùng biểu thức đại số:
  - Ký hiệu phép Và – AND:  $\cdot$
  - Ký hiệu phép Hoặc – OR:  $+$
  - Ký hiệu phép Đảo – NOT:  $\bar{\quad}$
  - VD:  $F = A \text{ AND } B$  hay  $F = A \cdot B$

# Biểu diễn biến và hàm logic (tiếp)

## ■ Dùng bảng thật:

- Dùng để mô tả sự phụ thuộc đầu ra vào các mức điện thế đầu vào của các mạch logic
- Bảng thật biểu diễn 1 hàm logic n biến có:
  - (n+1) cột:
    - n cột đầu tương ứng với n biến
    - cột còn lại tương ứng với giá trị của hàm
  - $2^n$  hàng:
    - tương ứng với  $2^n$  giá trị của tổ hợp biến

$$F = A.B$$

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$$F = A+B$$

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

# Biểu diễn biến và hàm logic (tiếp)

## ■ Dùng bìa Các-nô:

- Đây là cách biểu diễn tương đương của bảng thật.
- Trong đó, mỗi ô trên bìa tương ứng với 1 dòng của bảng thật.
- Tọa độ của ô xác định giá trị của tổ hợp biến.
- Giá trị của hàm được ghi vào ô tương ứng.

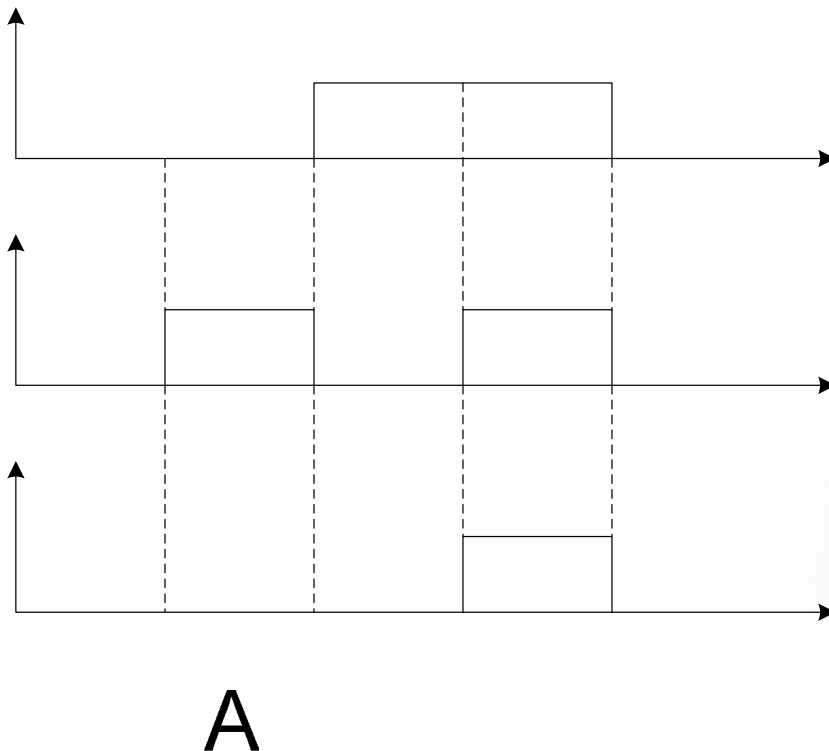
$$\mathbf{F = A.B}$$

| a\b | 0 | 1 |
|-----|---|---|
| 0   | 0 | 0 |
| 1   | 0 | 1 |

# Biểu diễn biến và hàm logic (tiếp)

- Dùng biểu đồ thời gian:

- Là đồ thị biểu diễn sự biến đổi theo thời gian của biến và hàm logic
- VD: với  $F = A \cdot B$



# 3. Các phép toán logic cơ bản

## Hàm HOẶC - OR

VD:  $F(A, B) = A + B$

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

## Hàm VÀ - AND

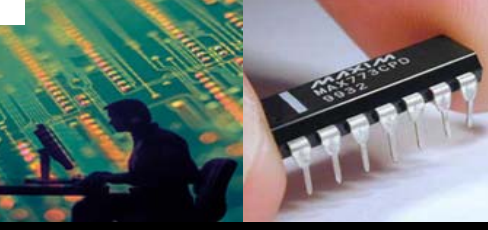
VD:  $F(A, B) = A.B$

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

## Hàm ĐẢO (Phủ định) - NOT

VD:  $F(A) = \bar{A}$

| A | F |
|---|---|
| 0 | 1 |
| 1 | 0 |



## 4. Tính chất của phép toán logic cơ bản

- **Tồn tại phần tử trung tính duy nhất trong phép toán AND và OR**
  - Của phép AND là **1**:  $A \cdot 1 = A$
  - Của phép OR là **0**:  $A + 0 = A$
- **Tính chất giao hoán**  
 $A \cdot B = B \cdot A$   
 $A + B = B + A$
- **Tính chất kết hợp**  
 $(A \cdot B) \cdot C = A \cdot (B \cdot C) = A \cdot B \cdot C$   
 $(A + B) + C = A + (B + C) = A + B + C$

- **Tính chất phân phối**

$$(A + B).C = A.C + B.C$$

$$(A.B) + C = (A + C).(B + C)$$

- **Tính chất không số mũ, không hệ số**

$$A.A.A. \dots .A = A$$

$$A+A+A+ \dots +A = A$$

- **Phép bù**

$$\overline{\overline{A}} = A$$

$$A + \overline{A} = 1$$

$$A . \overline{A} = 0$$



## 5. Định lý DeMorgan

- Đảo của một “tổng” bằng “tích” các đảo thành phần

$$\overline{(a + b)} = \bar{a} \cdot \bar{b}$$

- Đảo của một “tích” bằng “tổng” các đảo thành phần

$$\overline{(a \cdot b)} = \bar{a} + \bar{b}$$

- Tổng quát:

$$\overline{f(., +, a_1, a_2, \dots, a_n)} = f(+, .., \bar{a}_1, \bar{a}_2, \dots, \bar{a}_n)$$

## 6. Nguyên lý đối ngẫu

- Đối ngẫu:

- + đối ngẫu với .

- 0 đối ngẫu với 1

- Ví dụ:

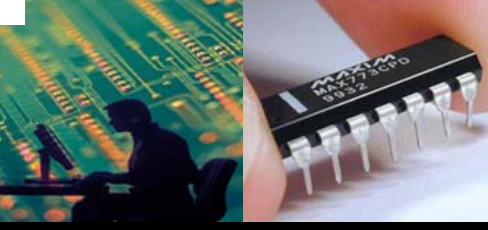
$$(A + B).C = A.C + B.C \quad \Leftrightarrow \quad (A.B) + C = (A + C).(B + C)$$

2.1. Giới thiệu

2.2. Đại số Boole

**2.2. Biểu diễn các hàm logic dưới dạng chính quy**

2.3. Tối thiểu hóa các hàm logic



## 2.2. Biểu diễn các hàm logic dưới dạng chính quy

- Một hàm logic thông thường được biểu diễn dưới 2 dạng:

- Tuyến: dạng tổng các tích

$$\underline{VD}: f(a,b,c) = ab + acb + cb$$

- Hội: dạng tích các tổng

$$\underline{VD}: f(a,b,c) = (a+b)(a+c+b)$$

- Một hàm logic được gọi là biểu diễn dưới dạng chính quy nếu mỗi số hạng của nó đều có đầy đủ các biến.

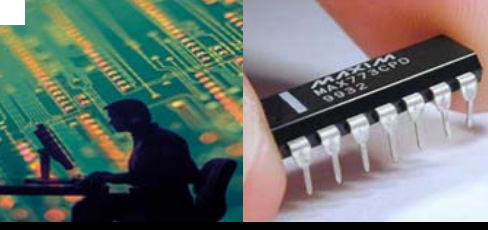
- Tuyến chính quy:

$$\underline{VD}: f(a,b,c) = abc + \bar{a}\bar{b}c$$

- Hội chính quy:

$$\underline{VD}: f(a,b,c) = (a+b+c)(\bar{a}+\bar{b}+c)$$

- Một hàm logic được gọi là biểu diễn dưới dạng không chính quy nếu như có ít nhất một biến vắng mặt trong ít nhất một số hạng.



# 1. Tuyển chính quy

- **Định lý Shannon:** một hàm logic bất kỳ có thể được triển khai theo 1 trong các biến dưới dạng tổng của 2 tích logic như sau:

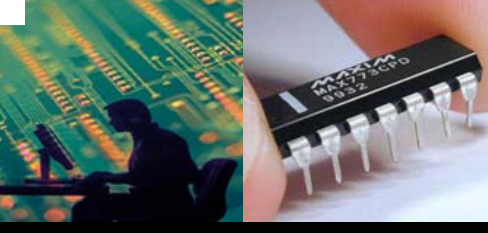
$$F(A_1, A_2, \dots, A_n) = A_1.F(1, A_2, \dots, A_n) + \overline{A_1}.F(0, A_2, \dots, A_n)$$

- Ví dụ:

$$\begin{aligned} F(A, B) &= A.F(1, B) + \overline{A}.F(0, B) \\ &= A.[B.F(1,1) + \overline{B}.F(1,0)] + \overline{A}.[B.F(0,1) + \overline{B}.F(0,0)] \\ &= AB.F(1,1) + A\overline{B}.F(1,0) + \overline{A}B.F(0,1) + \overline{A}\overline{B}.F(0,0) \end{aligned}$$

- Một hàm logic bất kỳ đều có thể chuyển về dạng tuyển chính quy nhờ áp dụng định lý Shannon cho dạng tuyển

# Áp dụng nhanh định lý Shannon



| <b>A</b> | <b>B</b> | <b>C</b> | <b>F<sub>1</sub></b> | <b>F<sub>2</sub></b> |
|----------|----------|----------|----------------------|----------------------|
| 0        | 0        | 0        | 0                    | 0                    |
| 0        | 0        | 1        | 1                    | 0                    |
| 0        | 1        | 0        | 1                    | 0                    |
| 0        | 1        | 1        | 1                    | 1                    |
| 1        | 0        | 0        | 0                    | 0                    |
| 1        | 0        | 1        | 1                    | 1                    |
| 1        | 1        | 0        | 1                    | 0                    |
| 1        | 1        | 1        | 0                    | 1                    |

$$F_1 = \bar{A} \bar{B} C + \bar{A} B \bar{C} + \bar{A} B C + A \bar{B} C + A B \bar{C}$$

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



## 2. Hội chính quy

- **Định lý Shannon:** một hàm logic bất kỳ có thể được triển khai theo 1 trong các biến dưới dạng tích của 2 tổng logic như sau:

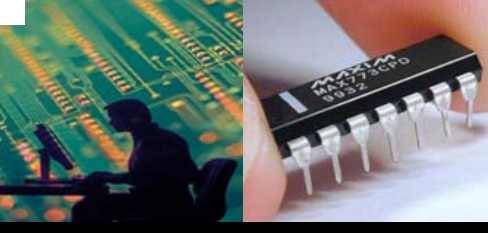
$$F(A_1, A_2, \dots, A_n) = [A_1 + F(0, A_2, \dots, A_n)].[\bar{A}_1 + F(1, A_2, \dots, A_n)]$$

- Ví dụ:

$$\begin{aligned} F(A, B) &= [A + F(0, B)].[\bar{A} + F(1, B)] \\ &= (A + [B + F(0,0)].[\bar{B} + F(0,1)]).(A + [B + F(1,0)].[\bar{B} + F(1,1)]) \\ &= [A + B + F(0,0)].[A + \bar{B} + F(0,1)].[\bar{A} + B + F(1,0)].[\bar{A} + \bar{B} + F(1,1)] \end{aligned}$$

- Một hàm logic bất kỳ đều có thể chuyển về dạng hội chính quy nhờ áp dụng định lý Shannon cho dạng hội

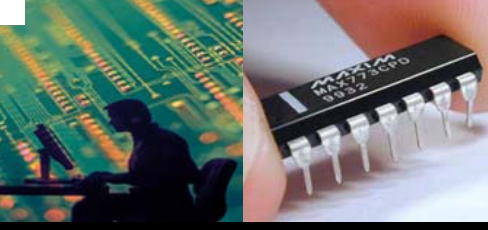
# Áp dụng nhanh định lý Shannon



| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

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





# 3. Biểu diễn hàm logic dưới dạng số

**Tuyển chính quy:**

$$F1(A,B,C) = R(1,4,5,7)$$

$$F1(A,B,C) = \overline{A}BC + A\overline{B}C + A\overline{B}\overline{C} + ABC$$

**Hội chính quy:**

$$F2(A,B,C) = J(0,3,5)$$

$$F2(A,B,C) = (A + B + C)(A + \overline{B} + \overline{C})(\overline{A} + B + \overline{C})$$

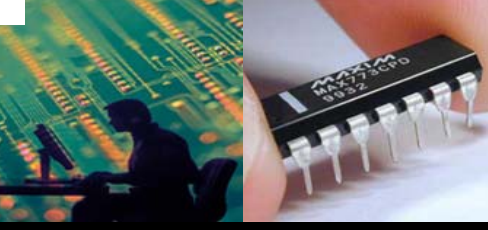
|   | A | B | C | F1 | F2 |
|---|---|---|---|----|----|
| 0 | 0 | 0 | 0 | 0  | 0  |
| 1 | 0 | 0 | 1 | 1  | 1  |
| 2 | 0 | 1 | 0 | 0  | 1  |
| 3 | 0 | 1 | 1 | 0  | 0  |
| 4 | 1 | 0 | 0 | 1  | 1  |
| 5 | 1 | 0 | 1 | 1  | 0  |
| 6 | 1 | 1 | 0 | 0  | 1  |
| 7 | 1 | 1 | 1 | 1  | 1  |

2.1. Giới thiệu

2.2. Đại số Boole

2.2. Biểu diễn các hàm logic dưới dạng chính quy

2.3. Tối thiểu hóa các hàm logic



## 2.3. Tối thiểu hóa các hàm logic

- Một hàm logic được gọi là tối thiểu hoá nếu như nó có số lượng số hạng ít nhất và số lượng biến ít nhất.
- Mục đích của việc tối thiểu hoá: Mỗi hàm logic có thể được biểu diễn bằng các biểu thức logic khác nhau. Mỗi 1 biểu thức logic có một mạch thực hiện tương ứng với nó. Biểu thức logic càng đơn giản thì mạch thực hiện càng đơn giản.
- Có hai phương pháp để tối thiểu hoá hàm logic:
  - Phương pháp đại số
  - Phương pháp bìa Các-nô

# 1. Phương pháp đại số

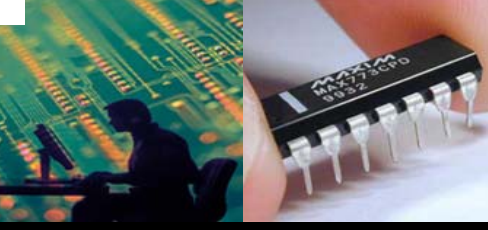
$$\begin{aligned} AB + A\bar{B} &= A & \Leftrightarrow & (A+B)(A+\bar{B}) &= A \\ A + AB &= A & \Leftrightarrow & A(A+B) &= A \\ A + \bar{A}B &= A + B & \Leftrightarrow & A(\bar{A} + B) &= AB \end{aligned}$$

CM:

$$\begin{aligned} AB + A\bar{B} &= A(B + \bar{B}) = A.1 = A \\ A + AB &= A(1 + B) = A.1 = A \\ A + \bar{A}B &= (A + \bar{A})(A + B) \\ &= 1(A + B) \\ &= A + B \end{aligned}$$

# Phương pháp nhóm số hạng

$$\begin{aligned}F(A,B,C,D) &= ABC + AB\bar{C} + A\bar{B}CD \\&= AB(C + \bar{C}) + A\bar{B}CD \\&= AB + A\bar{B}CD \\&= A(B + \bar{B}CD) \\&= A(B + CD) \\&= AB + ACD\end{aligned}$$



# Thêm số hạng đã có vào biểu thức

$$\begin{aligned}F(x,y,z) &= xyz + \bar{x}yz + xy\bar{z} + x\bar{y}z \\ &= xyz + \bar{x}yz + xyz + xy\bar{z} + xyz + x\bar{y}z \\ &= yz + xy + xz\end{aligned}$$



# Loại bỏ số hạng thừa

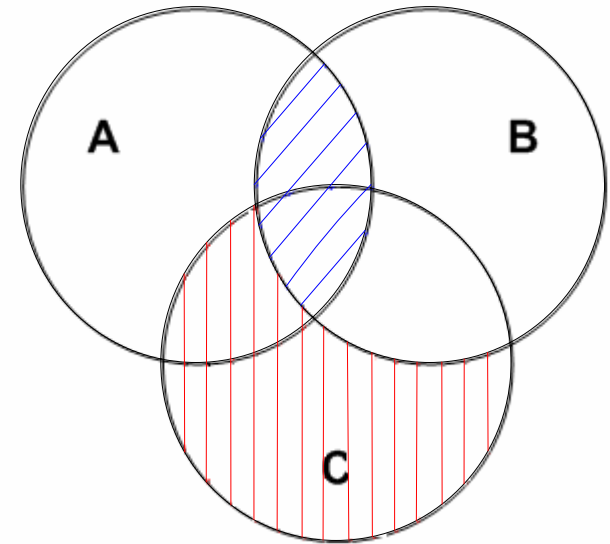
- Trong ví dụ sau, AC là số hạng thừa:

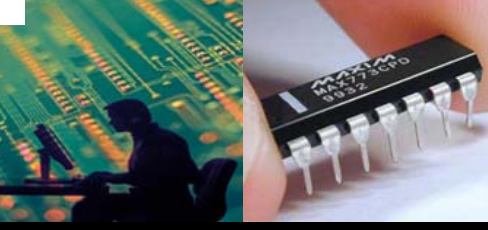
$$F(A,B,C) = AB + \bar{B}C + AC$$

Tối thiểu hóa?

Ta có:

$$\begin{aligned} F &= AB + \bar{B}C + AC(B + \bar{B}) \\ &= AB + \bar{B}C + ACB + AC\bar{B} \\ &= AB + ABC + \bar{B}C + A\bar{B}C \\ &= AB(1 + C) + \bar{B}C(1 + A) \\ &= AB + \bar{B}C \end{aligned}$$





# Bài tập áp dụng

- VD1: Tối thiểu hóa các hàm sau bằng phương pháp đại số:

a.  $F(A, B, C, D) = (A + BC) + \bar{A} \cdot (\bar{B} + \bar{C})(AD + C)$

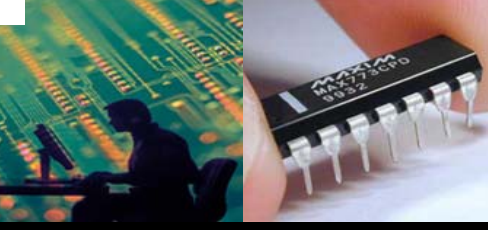
b.  $F(A, B, C, D) = (A + B + C)(A + B + \bar{C})(\bar{A} + B + C)(\bar{A} + B + \bar{C})$





## 2. Phương pháp bìa Các-nô

- Quy tắc lập bìa Các-nô:
  - 2 ô liền kề nhau chỉ sai khác nhau 1 giá trị của 1 biến (tương ứng với tổ hợp biến khác nhau 1 giá trị)
  - Bìa Các-nô có tính không gian



# Bìa Các-nô cho hàm 2, 3, 4 biến

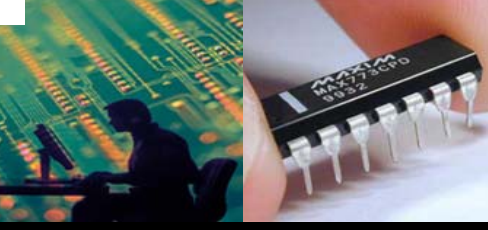
|   |  |  |
|---|--|--|
| \ |  |  |
|   |  |  |
|   |  |  |

|   |  |   |   |   |
|---|--|---|---|---|
| \ |  | A | B | 0 |
|   |  | A |   |   |
|   |  | 0 |   |   |

1

|      |    |    |    |    |
|------|----|----|----|----|
| \ CD | 00 | 01 | 11 | 10 |
| AB   |    |    |    |    |
| 00   |    |    |    |    |
| 01   |    |    |    |    |
| 11   |    |    |    |    |
| 10   |    |    |    |    |

1



# Quy tắc nhóm (dạng tuyến chính quy)

- Nhóm các ô liền kề mà giá trị của hàm cùng bằng 1 lại với nhau sao cho:
  - Số lượng các ô trong nhóm là lớn nhất có thể được,
  - Đồng thời số lượng ô trong nhóm phải là lũy thừa của 2,
  - Và hình dạng của nhóm phải là hình chữ nhật hoặc hình vuông
- Nhóm có  $2^n$  ô  $\Rightarrow$  loại bỏ được  $n$  biến
- Biến nào nhận được giá trị ngược nhau trong nhóm thì sẽ bị loại
- Các nhóm có thể trùng nhau một vài phần tử nhưng không được trùng hoàn toàn và phải nhóm hết các ô bằng 1
- Số lượng nhóm chính bằng số lượng số hạng sau khi đã tối thiểu hóa (mỗi nhóm tương ứng với 1 số hạng)

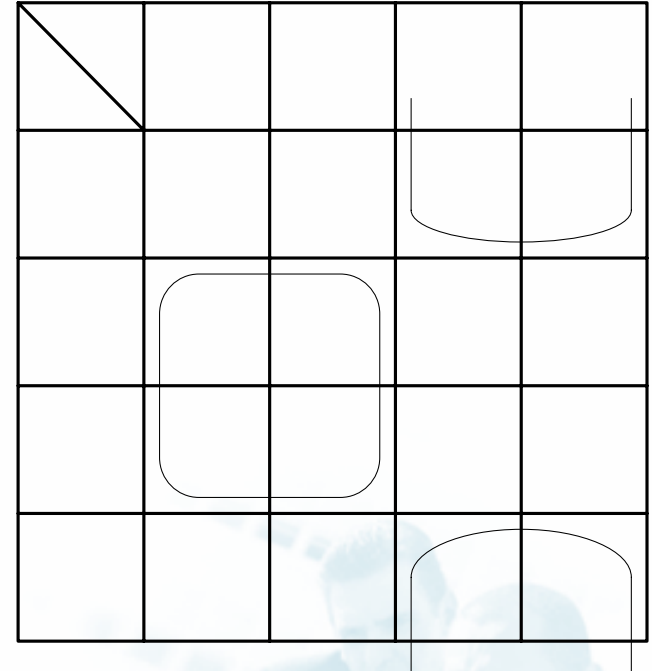
$$F(A, B, C) = \overline{\overline{A}}\overline{B}C + \overline{\overline{A}}B\overline{C} + A\overline{\overline{B}}\overline{C} + A\overline{B}C + ABC + A\overline{B}\overline{C}$$

$$F(A, B, C) = A + \overline{B}C + B\overline{C}$$

| A \ BC | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0      | 0  | 1  | 0  | 1  |
| 1      | 1  | 1  | 1  | 1  |

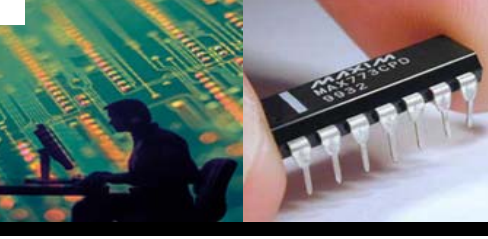
# Trường hợp đặc biệt

- Nếu giá trị hàm không xác định tại một vài tổ hợp biến nào đó:
  - Kí hiệu các ô không xác định bằng dấu –
  - Nhóm các ô – với các ô 1
  - Không nhất thiết phải nhóm hết các ô –



$$F(A, B, C, D) = \overline{BC} + \overline{BC}$$

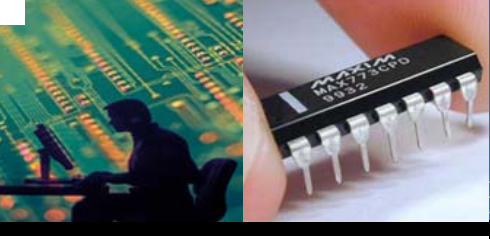
- Tối thiểu hóa các hàm sau bằng phương pháp bìa Các nô:
  - a.  $F(A,B,C,D) = R(0,2,5,6,9,11,13,14)$
  - b.  $F(A,B,C,D) = R(1,3,5,8,9,13,14,15)$
  - c.  $F(A,B,C,D) = R(2,4,5,6,7,9,12,13)$
  - d.  $F(A,B,C,D) = R(1,5,6,7,11,13)$  và  $F$  không xác định với tổ hợp biến 12,15.



# Chương 3

## CÁC PHẦN TỬ LOGIC CƠ BẢN

Bộ môn Kỹ thuật Máy tính, Khoa Công nghệ Thông tin  
Trường Đại học Bách Khoa Hà Nội



## 3.1. Khái niệm

3.2. Thực hiện phần tử AND, OR dùng Diode

3.3. Thực hiện phần tử NOT dùng Transistor

3.4. Các mạch tích hợp số



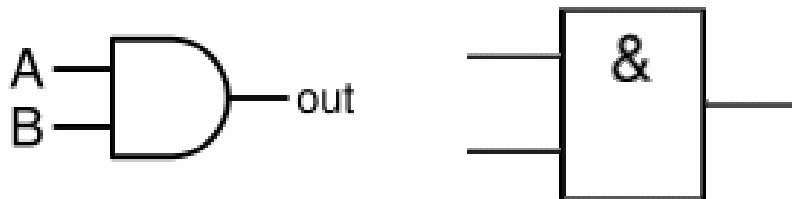


## 3.1. Khái niệm

- Có 3 phép toán logic cơ bản:
  - VÀ (AND)
  - HOẶC (OR)
  - ĐẢO (NOT)
- Phần tử logic cơ bản (mạch logic cơ bản, cổng logic) thực hiện phép toán logic cơ bản:
  - Cổng VÀ (AND gate)
  - Cổng HOẶC (OR gate)
  - Cổng ĐẢO (NOT inverter)
- Các mạch số đặc biệt khác: các cổng NAND, NOR, XOR, XNOR

# 1. Cổng VÀ (AND gate)

- Chức năng:
  - Thực hiện phép toán logic VÀ (AND)
  - Đầu ra chỉ bằng 1 khi tất cả các đầu vào bằng 1
- Cổng VÀ 2 đầu vào:
  - Ký hiệu:



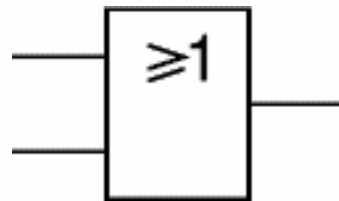
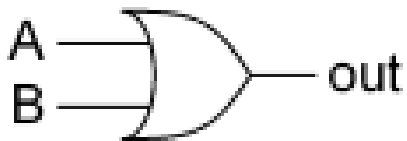
- Bảng thật:
- Biểu thức:  $out = A \cdot B$

| A | B | out |
|---|---|-----|
| 0 | 0 | 0   |
| 0 | 1 | 0   |
| 1 | 0 | 0   |
| 1 | 1 | 1   |

## 2. Cổng HOẶC (OR gate)

- Chức năng:
  - Thực hiện phép toán logic HOẶC (OR)
  - Đầu ra chỉ bằng 0 khi tất cả các đầu vào bằng 0
- Cổng HOẶC 2 đầu vào:

- Ký hiệu:

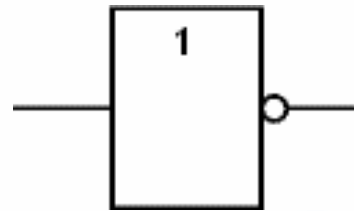
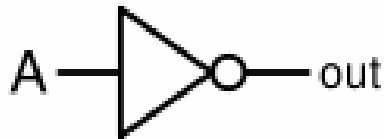


| A | B | out |
|---|---|-----|
| 0 | 0 | 0   |
| 0 | 1 | 1   |
| 1 | 0 | 1   |
| 1 | 1 | 1   |

- Bảng thật:
- Biểu thức:  $out = A + B$

# 3. Cổng ĐẢO (NOT inverter)

- Chức năng:
  - Thực hiện phép toán logic ĐẢO (NOT)
- Cổng ĐẢO chỉ có 1 đầu vào:
  - Ký hiệu:

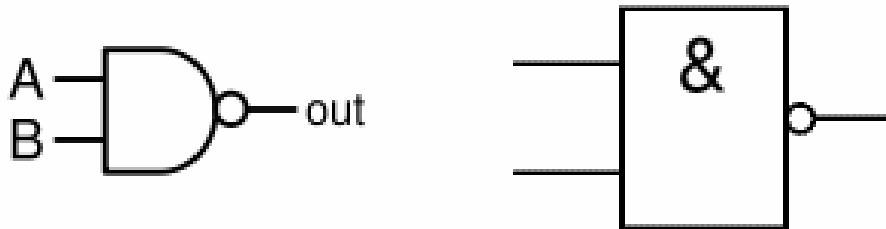


- Bảng thật:
- Biểu thức:  $out = \bar{A}$

| A | out |
|---|-----|
| 0 | 1   |
| 1 | 0   |

# 4. Cổng VÀ ĐẢO (NAND gate)

- Chức năng:
  - Thực hiện phép ĐẢO của phép toán logic VÀ
  - Đầu ra chỉ bằng 0 khi tất cả các đầu vào bằng 1
- Cổng VÀ ĐẢO 2 đầu vào:
  - Ký hiệu:

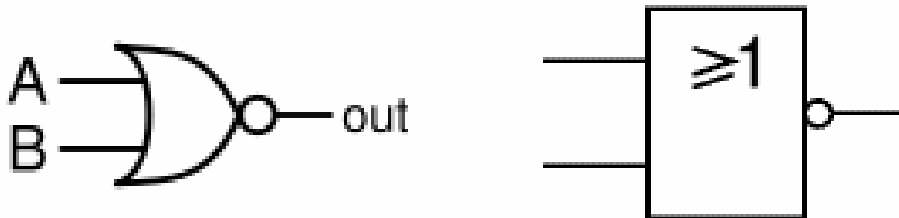


| A | B | out |
|---|---|-----|
| 0 | 0 | 1   |
| 0 | 1 | 1   |
| 1 | 0 | 1   |
| 1 | 1 | 0   |

- Bảng thật:
- Biểu thức:  $out = \overline{A \cdot B}$

# 5. Cổng HOẶC ĐẢO (NOR gate)

- Chức năng:
  - Thực hiện phép ĐẢO của phép toán logic HOẶC
  - Đầu ra chỉ bằng 1 khi tất cả các đầu vào bằng 0
- Cổng HOẶC ĐẢO 2 đầu vào:
  - Ký hiệu:

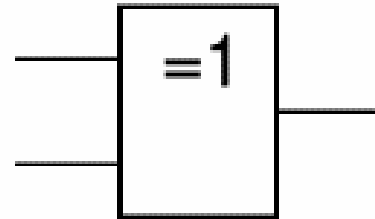
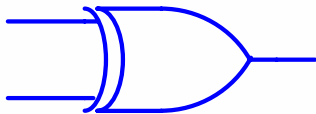


| A | B | out |
|---|---|-----|
| 0 | 0 | 1   |
| 0 | 1 | 0   |
| 1 | 0 | 0   |
| 1 | 1 | 0   |

- Bảng thật:
- Biểu thức:  $out = \overline{A + B}$

## 6. Cổng XOR (XOR gate)

- Chức năng:
  - Exclusive-OR
  - Thực hiện biểu thức logic HOẶC CÓ LOẠI TRỪ (phép toán XOR - hay còn là phép cộng module 2)
  - Đầu ra chỉ bằng 0 khi tất cả các đầu vào giống nhau
- Cổng XOR 2 đầu vào:
  - Ký hiệu:



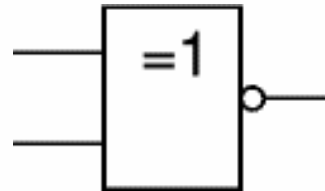
| A | B | out |
|---|---|-----|
| 0 | 0 | 0   |
| 0 | 1 | 1   |
| 1 | 0 | 1   |
| 1 | 1 | 0   |

- Bảng thật:

- Biểu thức:  $out = A \oplus B = \bar{A}.B + A.\bar{B}$

# 7. Cổng XNOR (XNOR gate)

- Chức năng:
  - Exclusive-NOR
  - Thực hiện phép ĐẢO của phép toán XOR
  - Đầu ra chỉ bằng 1 khi tất cả các đầu vào giống nhau
- Cổng XNOR 2 đầu vào:
  - Ký hiệu:



| A | B | out |
|---|---|-----|
| 0 | 0 | 1   |
| 0 | 1 | 0   |
| 1 | 0 | 0   |
| 1 | 1 | 1   |

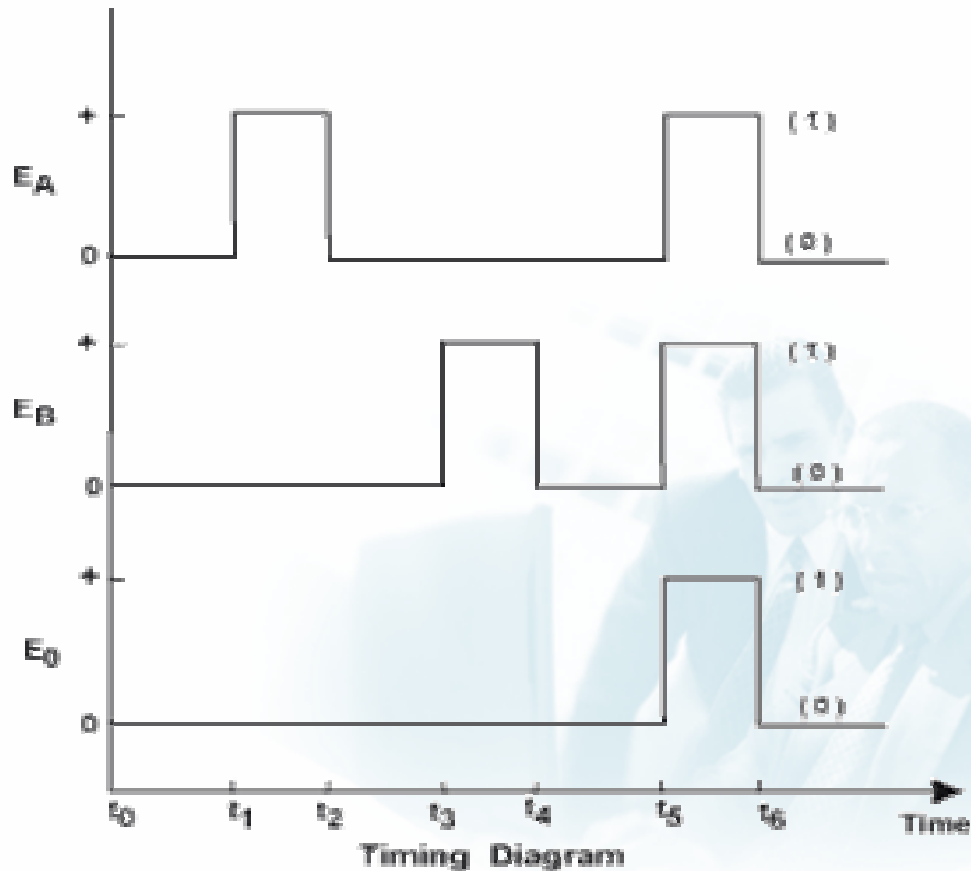
- Bảng thật:

- Biểu thức:  $out = \overline{A \oplus B} = A.B + \overline{A}.\overline{B}$

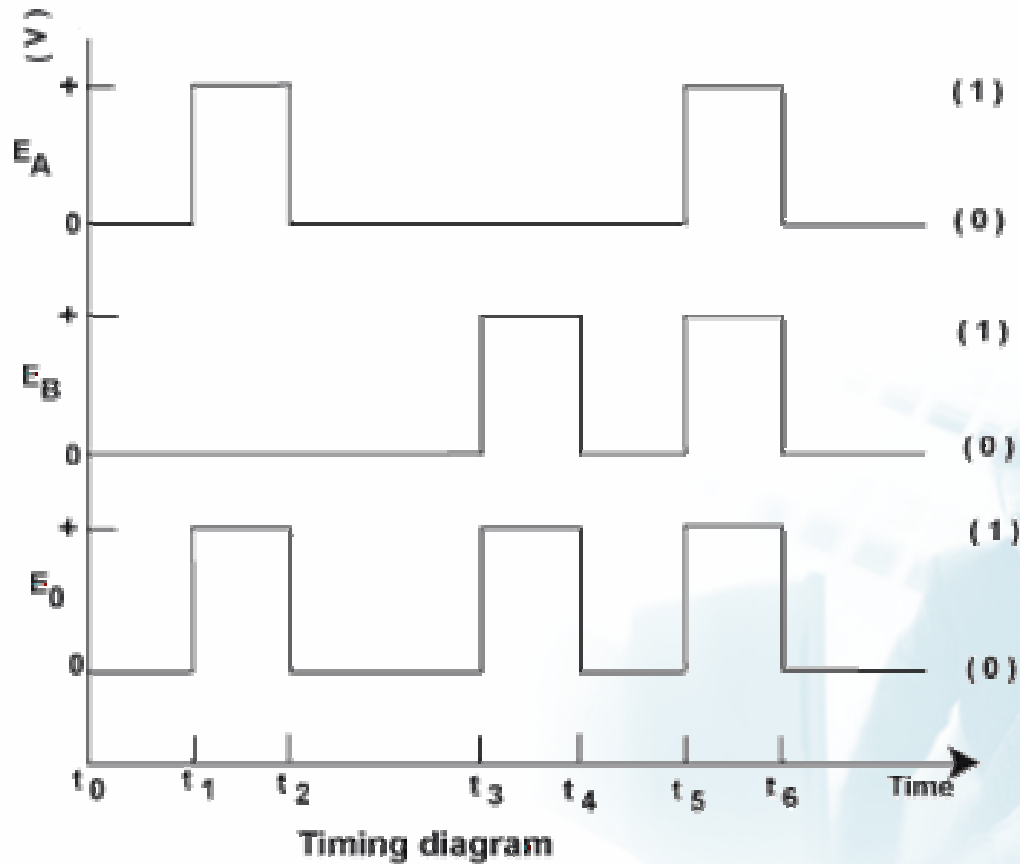


# 8. Bài tập

- Cho các biểu đồ thời gian sau, hãy cho biết từng biểu đồ thời gian biểu diễn hoạt động của cổng nào?
- $E_0 (E_A, E_B) = ?$



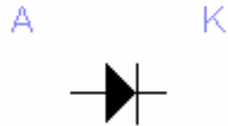
- $E_0(E_A, E_B) = ?$



## 3.2. Thực hiện phần tử AND, OR

### ■ Diode:

#### ○ Kí hiệu:



#### ○ Chức năng: cho dòng điện đi qua theo 1 chiều từ A đến K

#### ○ Hoạt động:

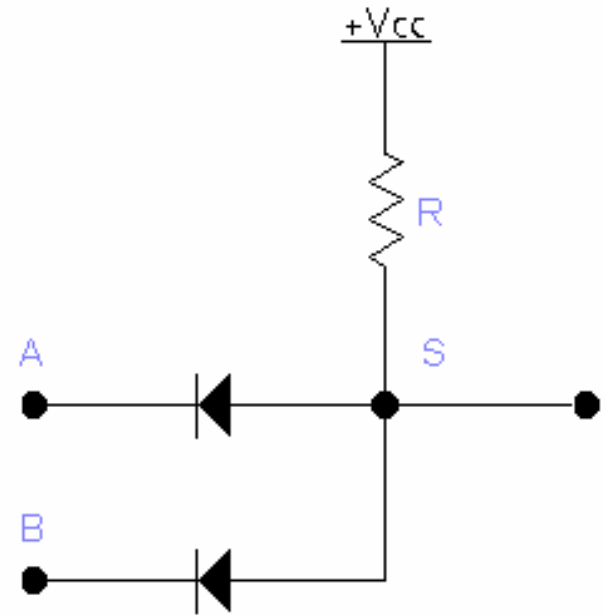
- Nếu  $U_A > U_K$  thì  $I_{AK} > 0$ , Diode làm việc ở chế độ Thông

- Nếu  $U_A \leq U_K$  thì  $I_{AK} = 0$ , Diode làm việc ở chế độ Tắt



# Phần tử AND 2 đầu vào dùng Diode

- Xét mạch ở hình bên.
- Giả sử lấy TTL làm chuẩn cho hoạt động của mạch.
- Lần lượt đặt điện áp 0V và 5V vào 2 đầu vào A và B, sau đó đo điện áp tại đầu ra S, ta có:



| $U_A$ | $U_B$ | $U_S$ |                        |
|-------|-------|-------|------------------------|
| 0     | 0     | 0     | $D_A, D_B$ thông       |
| 0     | 5     | 0     | $D_A$ thông, $D_B$ tắt |
| 5     | 0     | 0     | $D_A$ tắt, $D_B$ thông |
| 5     | 5     | 5     | $D_A, D_B$ tắt         |

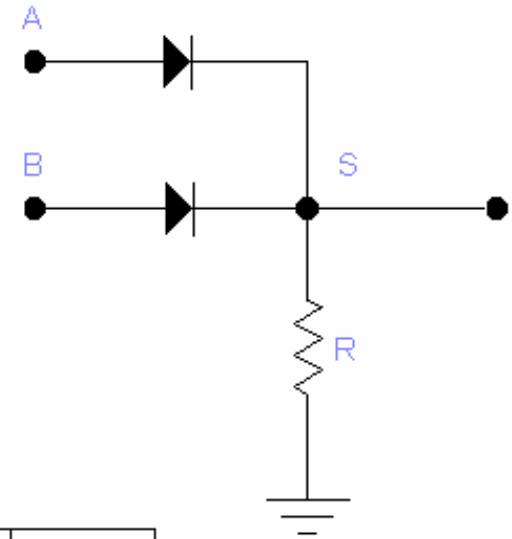
→

| A | B | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$$S = A.B$$

# Phần tử OR 2 đầu vào dùng Diode

- Xét mạch ở hình bên.
- Giả sử lấy TTL làm chuẩn cho hoạt động của mạch.
- Lần lượt đặt điện áp 0V và 5V vào 2 đầu vào A và B, sau đó đo điện áp tại đầu ra S, ta có:



| $U_A$ | $U_B$ | $U_S$ |                        |
|-------|-------|-------|------------------------|
| 0     | 0     | 0     | $D_A, D_B$ tắt         |
| 0     | 5     | 5     | $D_A$ tắt, $D_B$ thông |
| 5     | 0     | 5     | $D_A$ thông, $D_B$ tắt |
| 5     | 5     | 5     | $D_A, D_B$ thông       |

→

| A | B | S |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

$$S = A + B$$

## 3.3. Thực hiện phần tử NOT

### ■ Transistor lưỡng cực:

- Có 2 loại: NPN và PNP

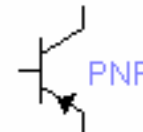
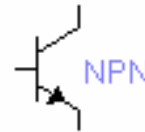
- Transistor có 3 cực:

- B: Base – cực gốc
- C: Collector – cực góp
- E: Emitter – cực phát

- Chức năng: Dùng để khuếch đại (thông) dòng  $I_C$  bằng việc điều khiển dòng  $I_B$

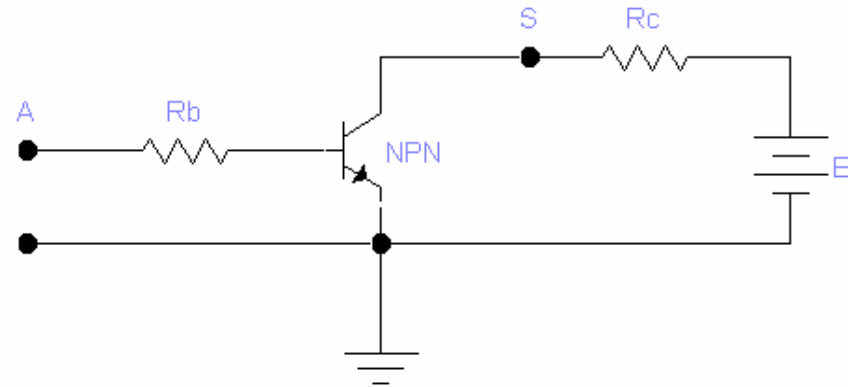
- Hoạt động:

- $I_B = 0$ , Transistor làm việc ở chế độ không khuếch đại (tắt),  $I_C = 0$
- $I_B > 0$ , Transistor làm việc ở chế độ khuếch đại (thông),  $I_C = \beta \cdot I_B$ , trong đó  $\beta$  là hệ số khuếch đại.



# Phần tử NOT dùng Transistor

- Xét mạch ở hình sau.



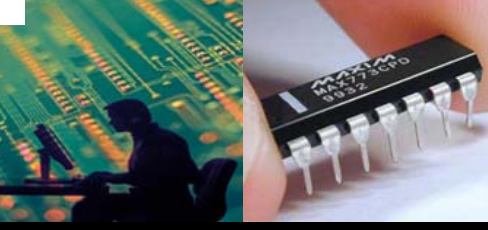
- Giả sử lấy TTL làm chuẩn cho hoạt động của mạch.
- Lần lượt đặt điện áp 0V và 5V vào đầu vào A và chọn  $R_b$  đủ nhỏ sao cho Transistor thông bão hòa, sau đó đo điện áp tại đầu ra S, ta có:

| $U_A$ | $U_S$     |
|-------|-----------|
| 0     | 5 T tắt   |
| 5     | 0 T thông |

→

| A | S |
|---|---|
| 0 | 1 |
| 1 | 0 |

$$S = \bar{A}$$



## 3.4. Các mạch tích hợp số

- Các phần tử logic được cấu thành từ các linh kiện điện tử
- Các linh kiện điện tử này khi kết hợp với nhau thường ở dạng các mạch tích hợp hay còn gọi là IC (Integrated Circuit).
- Mạch tích hợp hay còn gọi là IC, chip, vi mạch, bo... có đặc điểm:
  - Ưu điểm: mật độ linh kiện, làm giảm thể tích, giảm trọng lượng và kích thước mạch.
  - Nhược điểm: hỏng một linh kiện thì hỏng cả mạch.
- Có 2 loại mạch tích hợp:
  - Mạch tích hợp tương tự: làm việc với các tín hiệu tương tự
  - Mạch tích hợp số: làm việc với các tín hiệu số



# Phân loại mạch tích hợp số

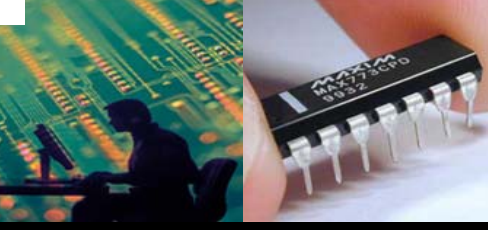
## ■ Theo mật độ linh kiện:

### ○ Tính theo số lượng cổng (gate).

- Một cổng có khoảng 2÷10 transistor
- VD: cổng NAND 2 đầu vào có cấu tạo từ 4 transistor

### ○ Có các loại sau:

- SSI - Small Scale Integration: các vi mạch có mật độ tích hợp cỡ nhỏ: < 10 cổng/chip
- MSI - Medium Scale Integration: các vi mạch có mật độ tích hợp cỡ trung bình: 10 ÷ 100 cổng/chip
- LSI - Large Scale Integration: các vi mạch có mật độ tích hợp cỡ lớn: 100 ÷ 1000 cổng/chip
- VLSI - Very Large Scale Integration: các vi mạch có mật độ tích hợp cỡ rất lớn:  $10^3 \div 10^6$  cổng/chip
- ULSI - Ultra Large Scale Integration: các vi mạch có mật độ tích hợp cỡ cực kỳ lớn:  $> 10^6$  cổng/chip

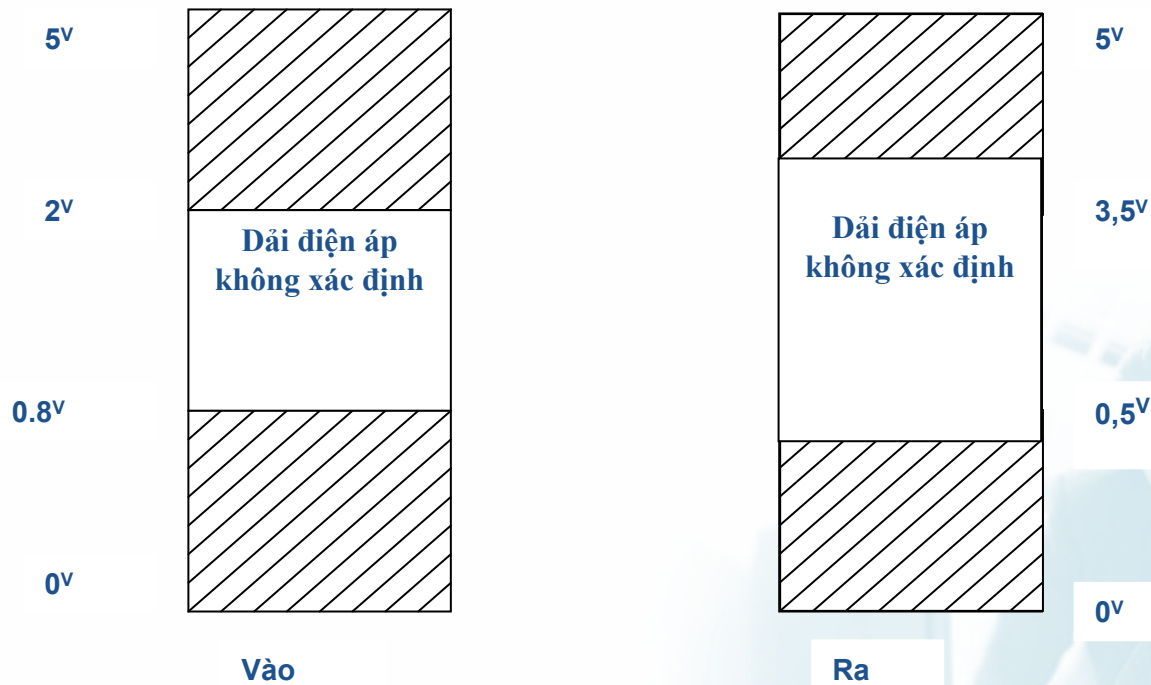


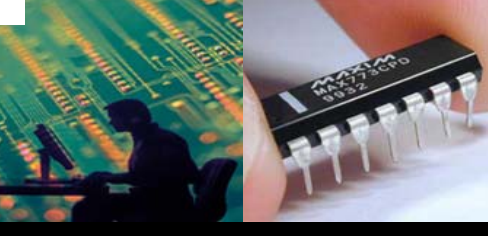
# Phân loại mạch tích hợp số (tiếp)

- Theo bản chất linh kiện được sử dụng:
  - IC sử dụng Transistor lưỡng cực:
    - RTL Resistor Transistor Logic (đầu vào mắc điện trở, đầu ra là Transistor)
    - DTL Diode Transistor Logic (đầu vào mắc Diode, đầu ra là Transistor)
    - TTL Transistor Transistor Logic (đầu vào mắc Transistor, đầu ra là Transistor)
    - ECL Emitter Coupled Logic (Transistor ghép nhiều cực emitter)
  - IC sử dụng Transistor trường - FET (Field Effect Transistor)
    - MOS Metal Oxide Semiconductor
    - CMOS Complementary MOS

# Đặc tính điện của IC

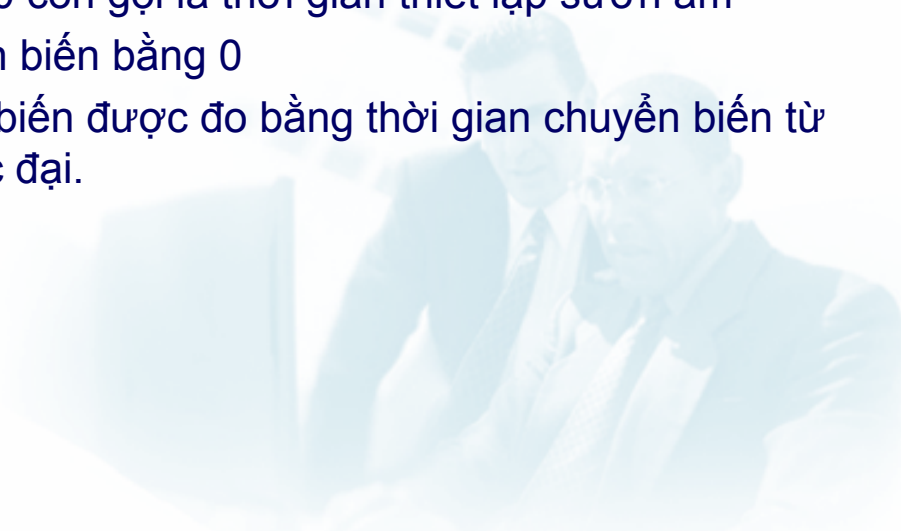
- Dải điện áp quy định mức logic
- VD: với chuẩn TTL ta có:





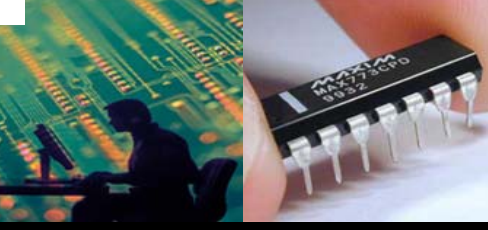
# Đặc tính điện của IC (tiếp)

- Thời gian truyền: tín hiệu truyền từ đầu vào tới đầu ra của mạch tích hợp phải mất một khoảng thời gian nào đó. Thời gian đó được đánh giá qua 2 thông số:
  - Thời gian trễ: là thời gian trễ thông tin của đầu ra so với đầu vào
  - Thời gian chuyển biến: là thời gian cần thiết để chuyển biến từ mức 0 lên mức 1 và ngược lại.
    - Thời gian chuyển biến từ 0 đến 1 còn gọi là thời gian thiết lập sườn dương
    - Thời gian chuyển biến từ 1 đến 0 còn gọi là thời gian thiết lập sườn âm
    - Trong lý thuyết: thời gian chuyển biến bằng 0
    - Trong thực tế, thời gian chuyển biến được đo bằng thời gian chuyển biến từ 10% đến 90% giá trị biên độ cực đại.



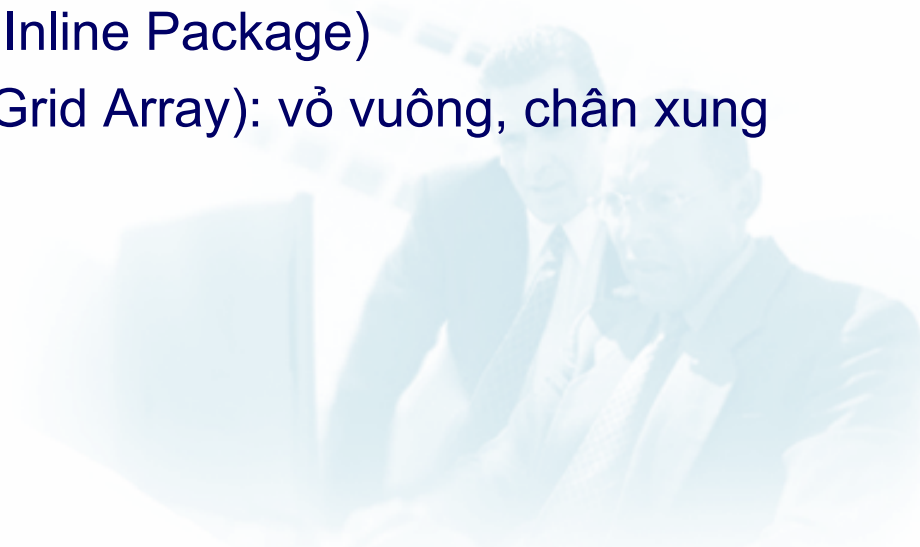
# Đặc tính điện của IC (tiếp)

- Công suất tiêu thụ ở chế độ động:
  - Chế độ động là chế độ làm việc có tín hiệu
  - Là công suất tổn hao trên các phần tử trong vi mạch, nên cần càng nhỏ càng tốt.
  - Công suất tiêu thụ ở chế độ động phụ thuộc
    - Tần số làm việc.
    - Công nghệ chế tạo: công nghệ CMOS có công suất tiêu thụ thấp nhất.



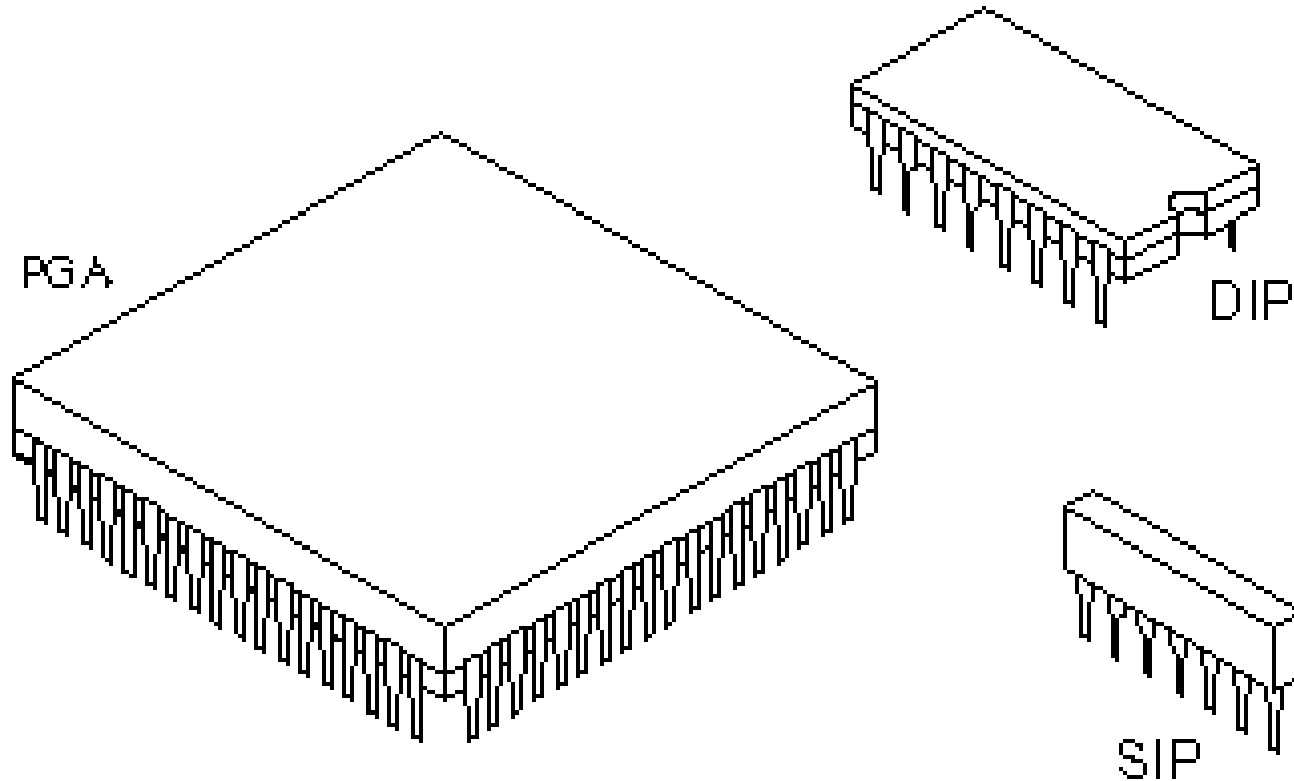
# Đặc tính cơ của IC

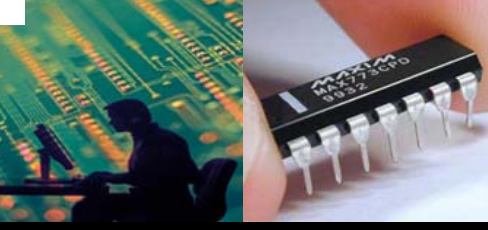
- Là đặc tính của kết cấu vỏ bọc bên ngoài.
- Có 2 loại thông dụng:
  - Vỏ tròn bằng kim loại, số chân < 10
  - Vỏ dẹt bằng gốm, chất dẻo, có 3 loại
    - IC một hàng chân SIP (Single Inline Package) hay SIPP (Single In-line Pin Package)
    - IC có 2 hàng chân DIP (Dual Inline Package)
    - IC chân dạng lưới PGA (Pin Grid Array): vỏ vuông, chân xung quanh



# Đặc tính cơ của IC (tiếp)

- Một số dạng IC:



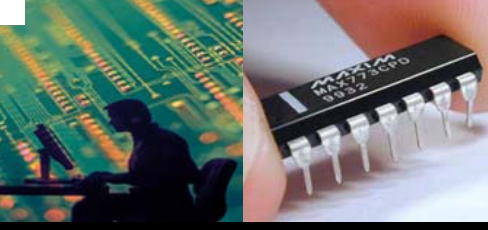


# Đặc tính nhiệt của IC

- Mỗi một loại IC được chế tạo để sử dụng ở một điều kiện môi trường khác nhau tùy theo mục đích sử dụng nó.
  - IC dùng trong công nghiệp:  $0^{\circ}\text{C} \div 70^{\circ}\text{C}$
  - IC dùng trong quân sự:  $-55^{\circ}\text{C} \div 125^{\circ}\text{C}$

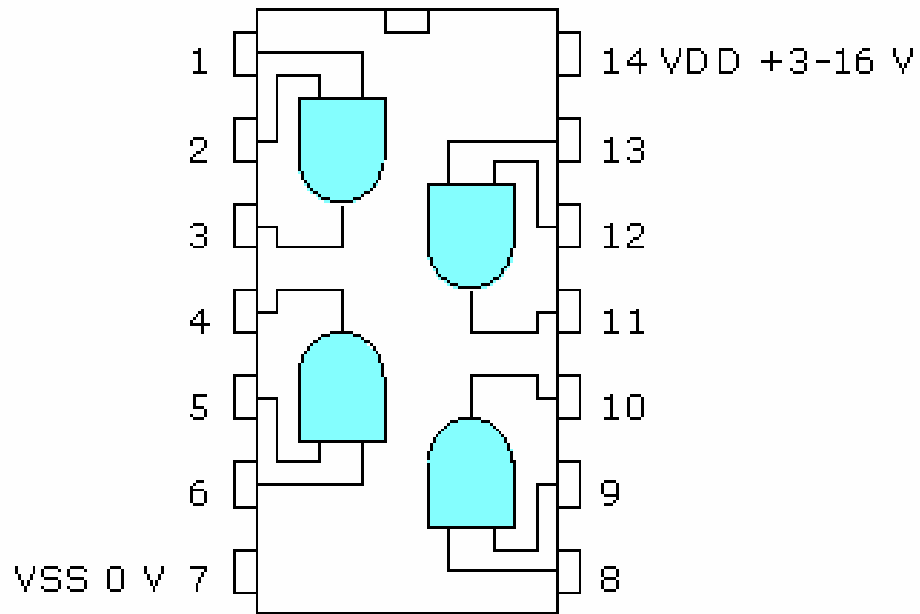




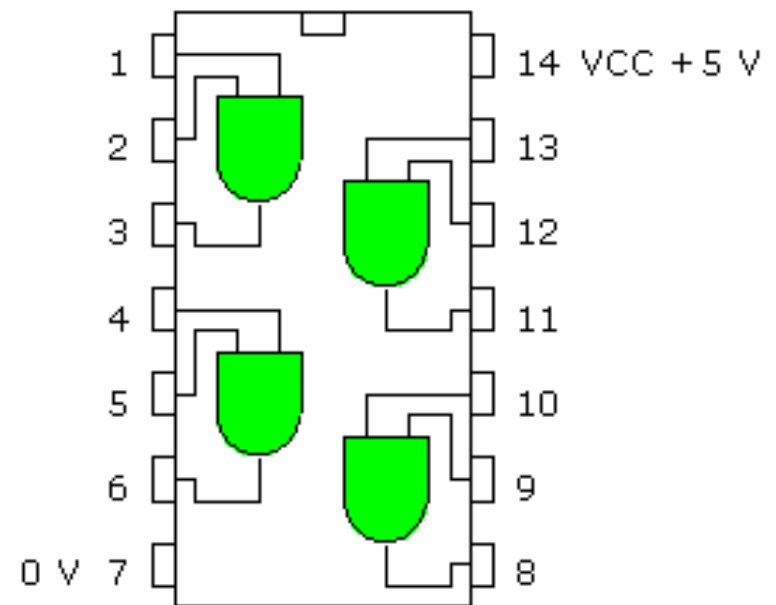


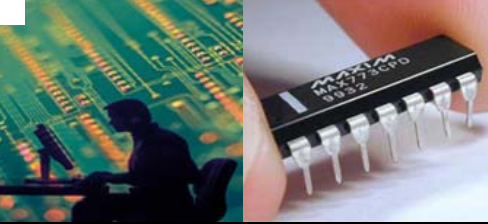
# VD: Phần tử AND dùng IC

4081



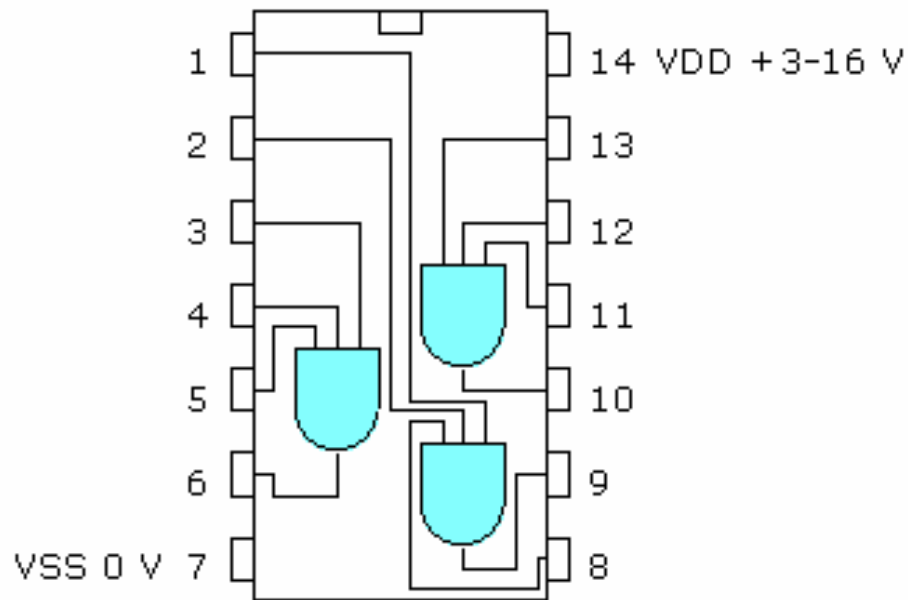
74LS08



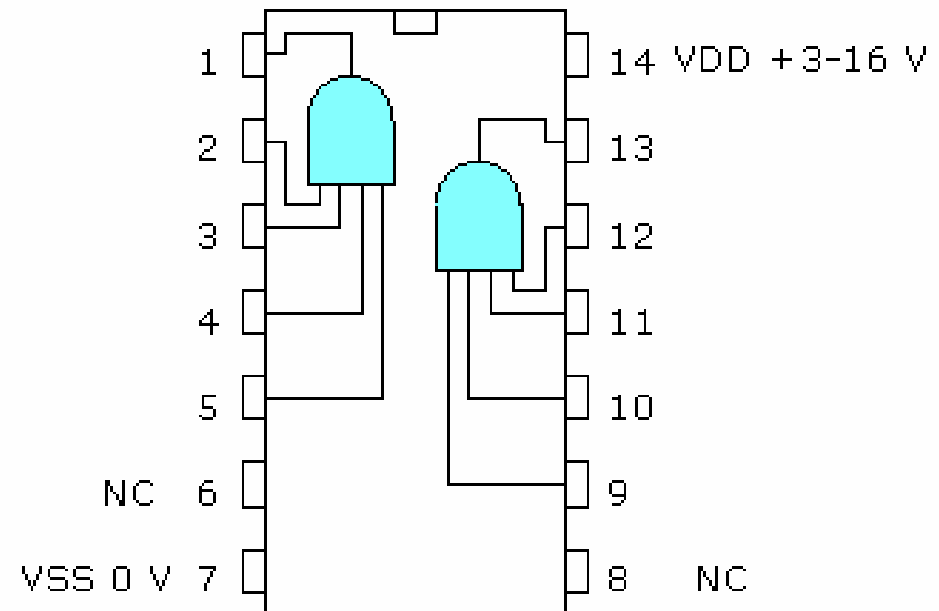


# VD: Phần tử AND dùng IC (tiếp)

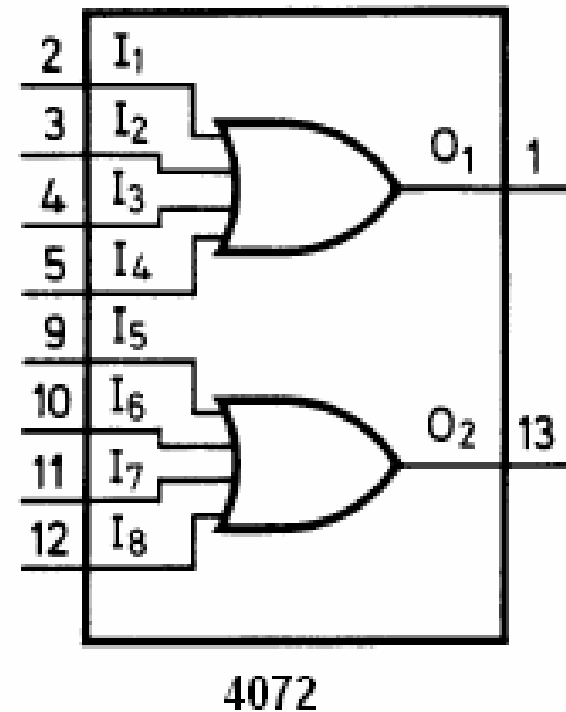
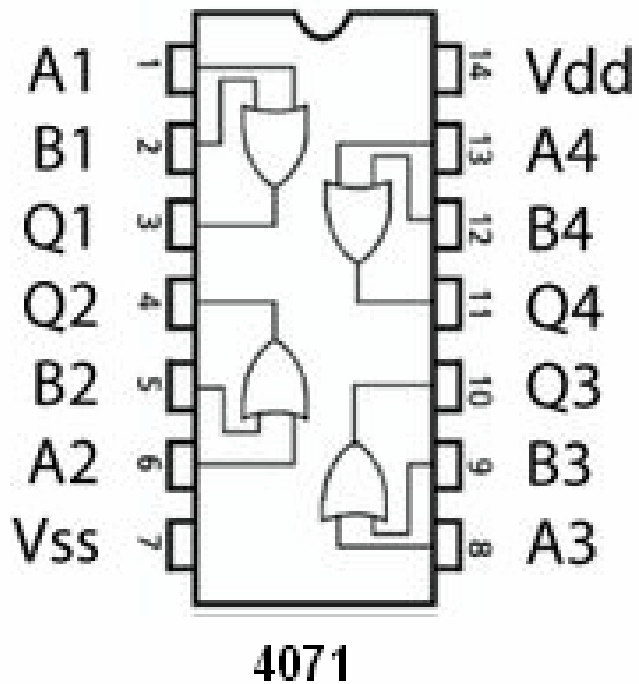
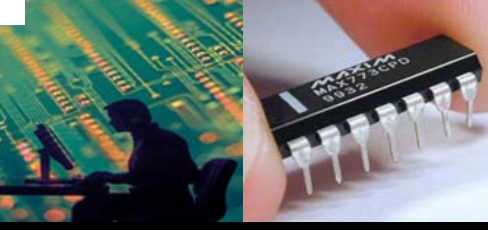
4073



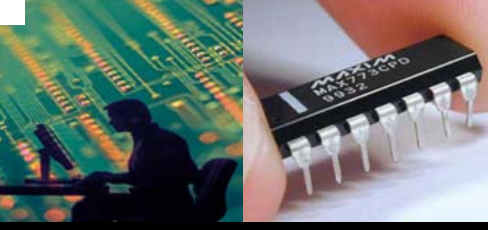
4082



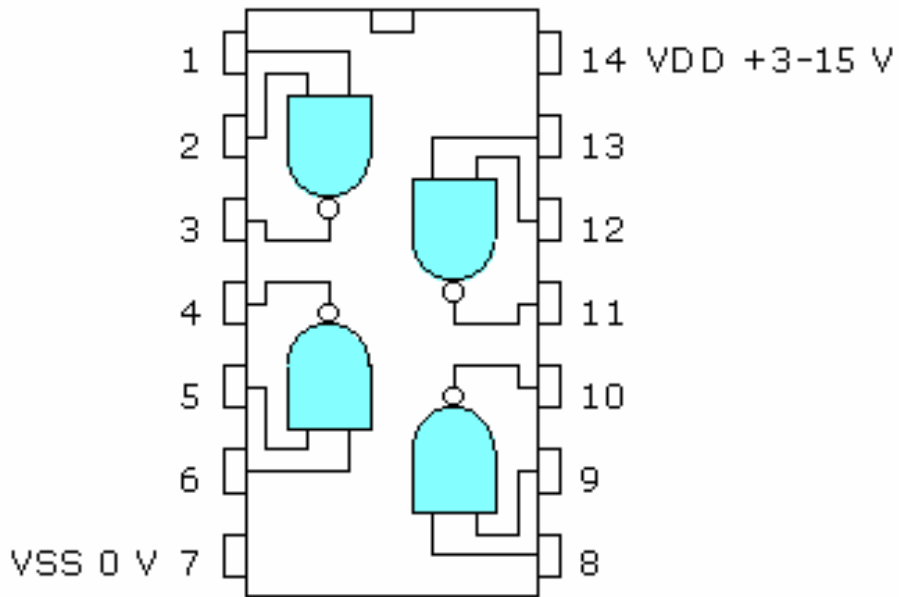
# VD: Phần tử OR dùng IC



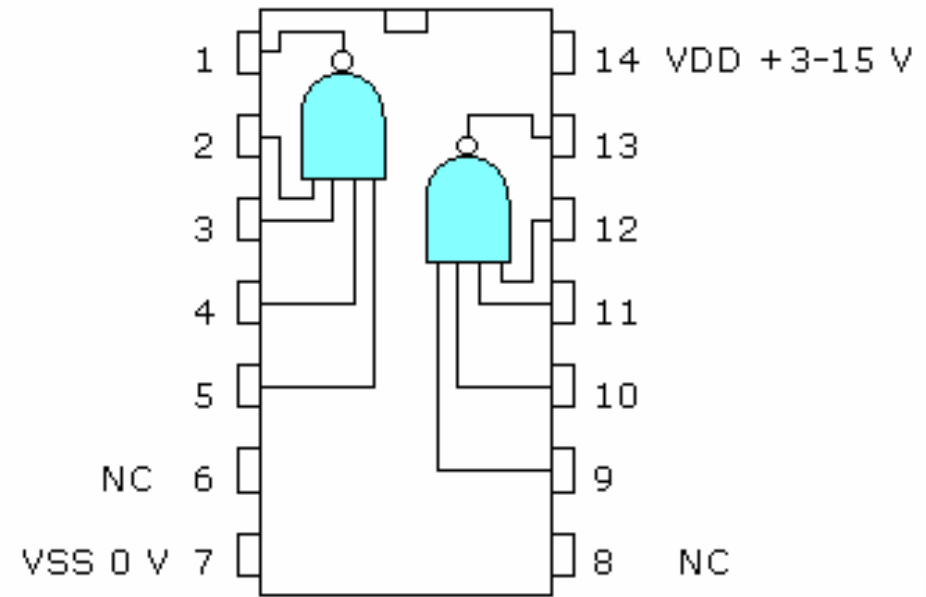
# VD: Phần tử NAND dùng IC



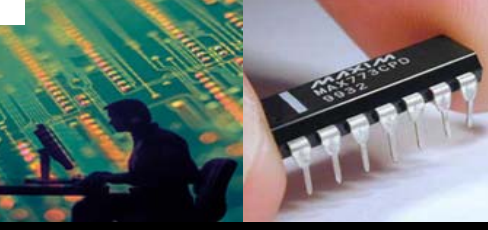
4011



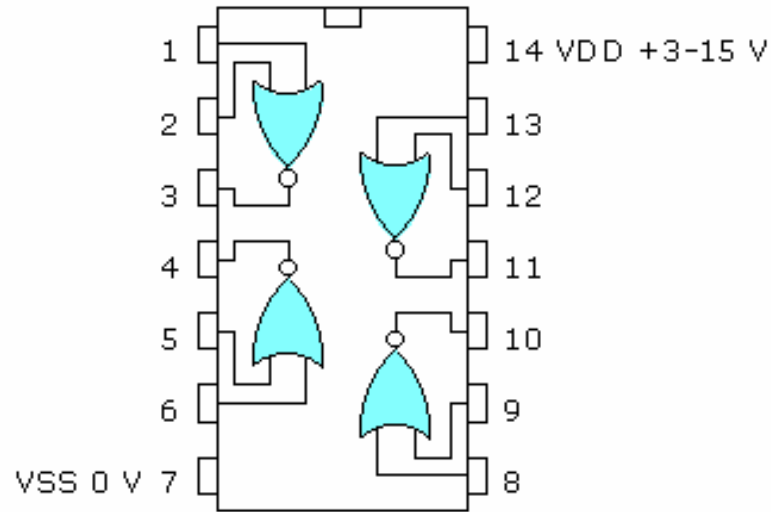
4012



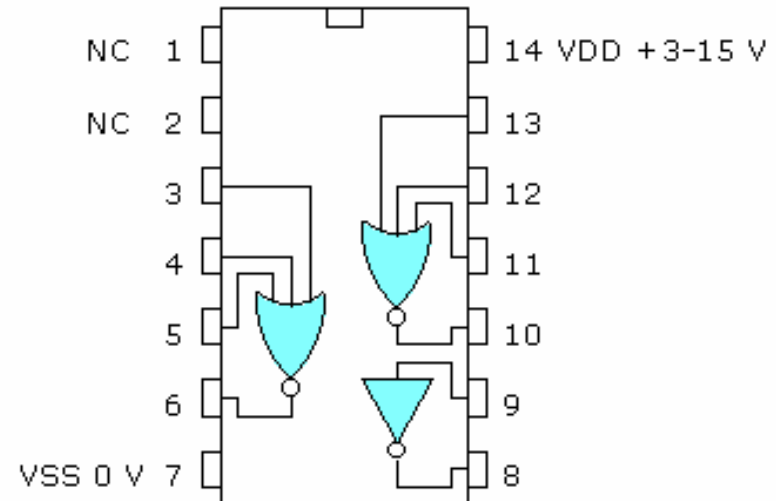
# VD: Phần tử NOR dùng IC



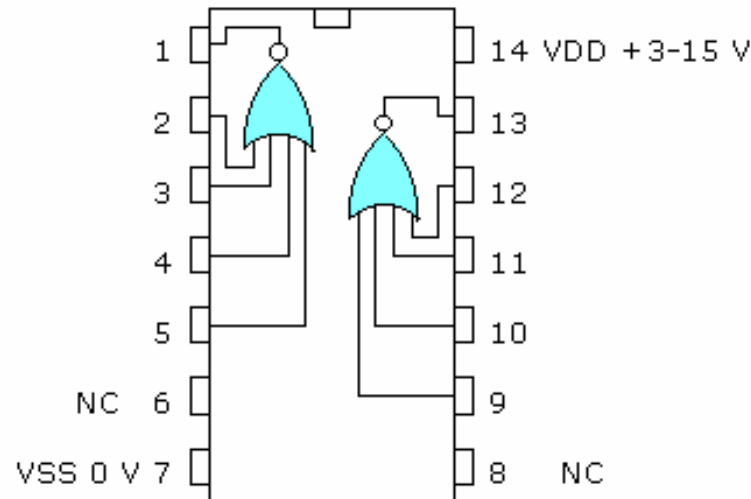
4001

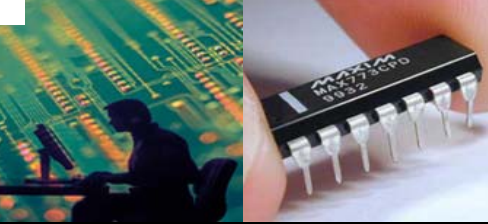


4000

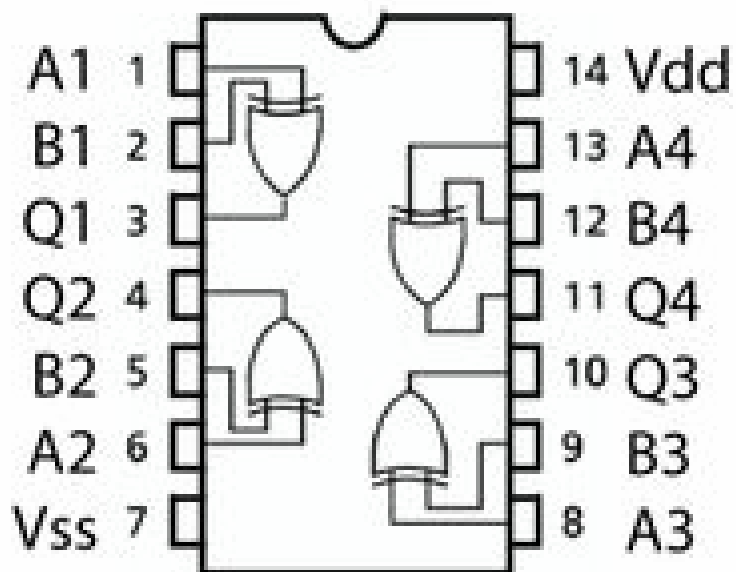


4002

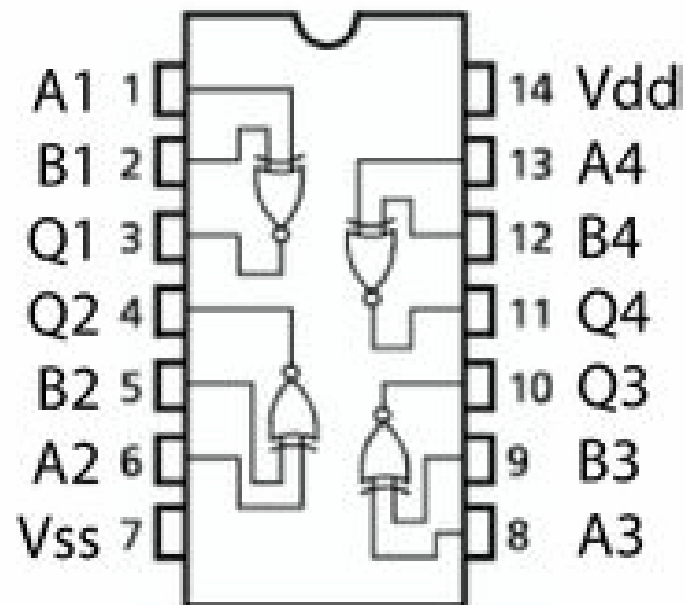




# VD: Phần tử XOR và XNOR dùng IC



**4070/4030**



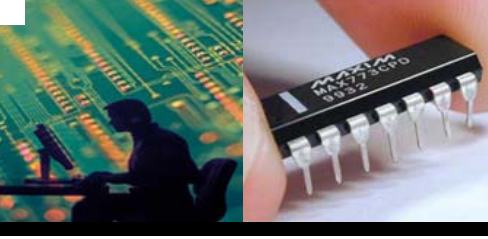
**4077**

# Các phần tử logic cơ bản

- AND: 74LS08
- OR: 74LS32
- NOT: 74LS04/05
- NAND: 74LS00
- NOR: 74LS02
- XOR: 74LS136
- NXOR: 74LS266

- Biểu diễn các phần tử logic hai đầu vào AND, OR và phần tử logic một đầu vào NOT chỉ dùng phần tử NAND.

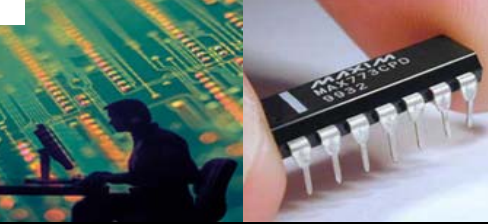




# Chương 4

## HỆ TỔ HỢP

Bộ môn Kỹ thuật Máy tính, Khoa Công nghệ Thông tin  
Trường Đại học Bách Khoa Hà Nội



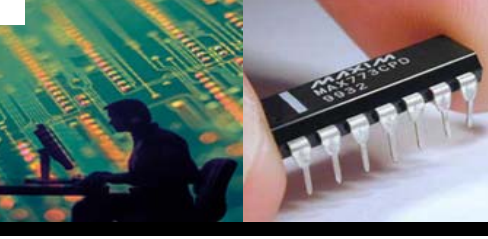
## 4.1. Khái niệm

## 4.2. Một số hệ tổ hợp cơ bản



## 4.1. Khái niệm

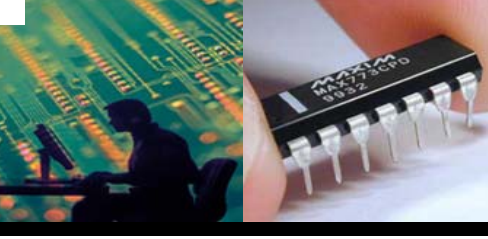
- Hệ tổ hợp là hệ mà tín hiệu ra chỉ phụ thuộc vào tín hiệu vào tại thời điểm hiện tại
- Hệ tổ hợp còn được gọi là hệ không có nhớ
- Hệ tổ hợp chỉ cần thực hiện bằng những phần tử logic cơ bản



4.1. Khái niệm

4.2. Một số hệ tổ hợp cơ bản





## 4.2. Một số hệ tổ hợp cơ bản

1. Bộ mã hóa
2. Bộ giải mã
3. Bộ chọn kênh
4. Bộ phân kênh
5. Các mạch số học

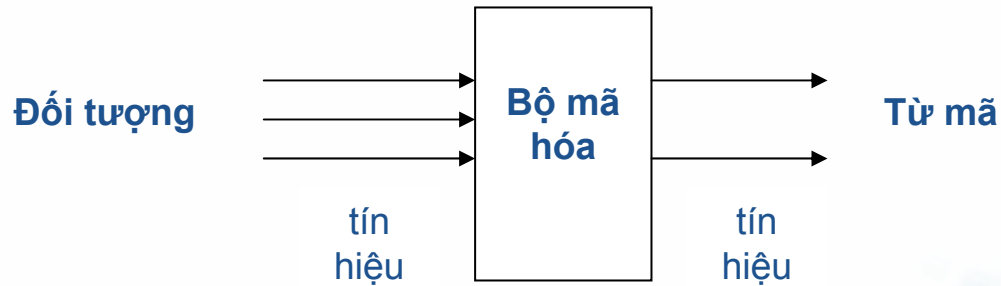


# 1. Bộ mã hóa

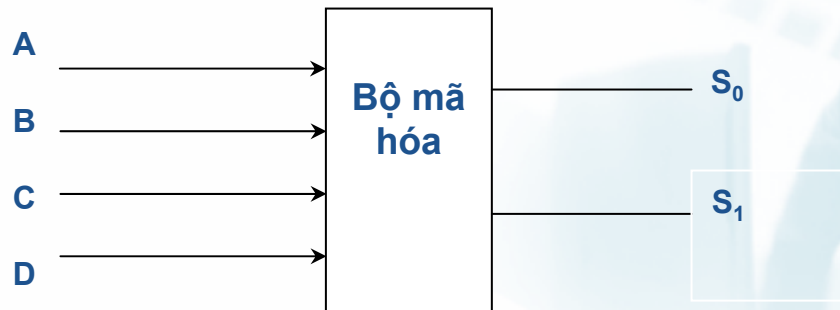
- Mã hóa là việc sử dụng ký hiệu để biểu diễn đặc trưng cho một đối tượng nào đó.
- Ký hiệu tương ứng với một đối tượng được gọi là từ mã.
- Ví dụ:

| Đối tượng | Từ mã thập phân | Từ mã nhị phân |
|-----------|-----------------|----------------|
| A         | 0               | 00             |
| B         | 1               | 01             |
| C         | 2               | 10             |
| D         | 3               | 11             |

- Chức năng: thực hiện việc mã hóa các tín hiệu tương ứng với các đối tượng thành các từ mã nhị phân.



- Thí dụ:



# Ví dụ - Bộ mã hóa bàn phím

- Mã hóa bàn phím:
  - Mỗi phím được gán một từ mã khác nhau.
  - Khi một phím được nhấn, bộ mã hóa sẽ cho ra đầu ra là từ mã tương ứng đã gán cho phím đó.
- Hãy thiết kế bộ mã hóa cho một bàn phím gồm có 9 phím với giả thiết trong một thời điểm chỉ có duy nhất 1 phím được nhấn.



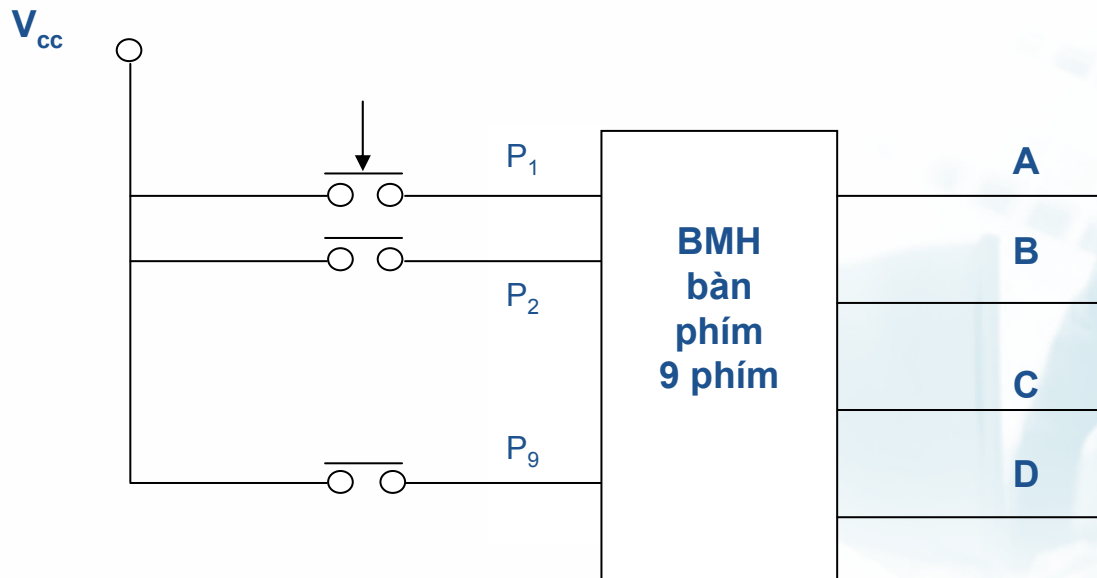
# Bộ mã hóa bàn phím (tiếp)

## ■ Sơ đồ khối:

- Một bộ 9 phím, phải sử dụng 4 bit để mã hóa.
- Vậy có 9 đầu vào, 4 đầu ra.

## ■ Mã hóa ưu tiên:

- Nếu 2 hoặc nhiều phím đồng thời được nhấn, thì bộ mã hóa chỉ coi như 1 phím được nhấn, và phím đó có mã cao nhất.



# Bộ mã hóa bàn phím (tiếp)

- Bảng mã hóa:

| <b>P</b> | <b>A</b> | <b>B</b> | <b>C</b> | <b>D</b> |
|----------|----------|----------|----------|----------|
| 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        |

# Bộ mã hóa bàn phím (tiếp)

- Lập biểu thức đầu ra phụ thuộc đầu vào:

- $A = 1$  khi  $P_8$  hoặc  $P_9$  được nhấn, tức là khi  $P_8 = 1$  hoặc  $P_9 = 1$

$$\text{Vậy } A = P_8 + P_9$$

- $B = 1$  khi  $P_4$  hoặc  $P_5$  hoặc  $P_6$  hoặc  $P_7$  được nhấn, tức là khi  $P_4 = 1$  hoặc  $P_5 = 1$  hoặc  $P_6 = 1$  hoặc  $P_7 = 1$

$$\text{Vậy } B = P_4 + P_5 + P_6 + P_7$$

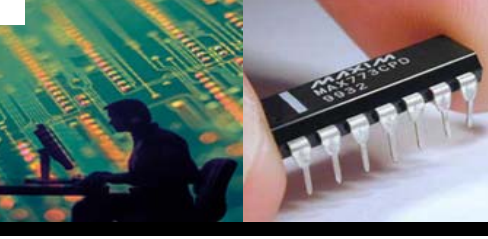
- $C = 1$  khi  $P_2$  hoặc  $P_3$  hoặc  $P_6$  hoặc  $P_7$  được nhấn, tức là khi  $P_2 = 1$  hoặc  $P_3 = 1$  hoặc  $P_6 = 1$  hoặc  $P_7 = 1$

$$\text{Vậy } C = P_2 + P_3 + P_6 + P_7$$

- $D = 1$  khi  $P_1$  hoặc  $P_3$  hoặc  $P_5$  hoặc  $P_7$  hoặc  $P_9$  được nhấn, tức là khi  $P_1 = 1$  hoặc  $P_3 = 1$  hoặc  $P_5 = 1$  hoặc  $P_7 = 1$  hoặc  $P_9 = 1$

$$\text{Vậy } D = P_1 + P_3 + P_5 + P_7 + P_9$$

- Vẽ mạch: ...



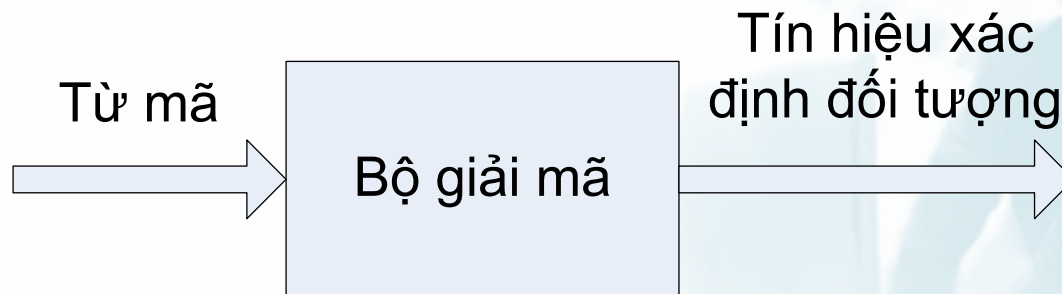
- Tìm hiểu hoạt động của bàn phím máy tính đơn giản
  - TLTK: [www.wikipedia.org](http://www.wikipedia.org)



## 2. Bộ giải mã

### ■ Chức năng:

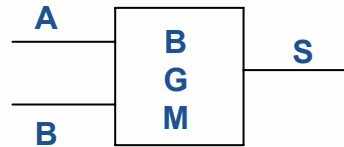
- Bộ giải mã thực hiện chức năng ngược với bộ mã hóa.
- Cung cấp thông tin ở đầu ra khi đầu vào xuất hiện tổ hợp các biến nhị phân ứng với 1 hay nhiều từ mã đã được chọn.
- Từ từ mã xác định được tín hiệu tương ứng với đối tượng đã mã hóa.



# Hai trường hợp giải mã

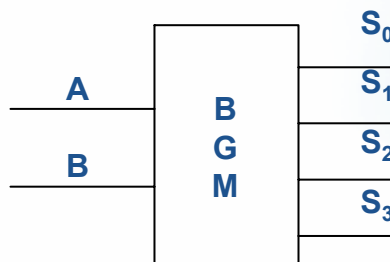
## ■ Giải mã cho 1 từ mã:

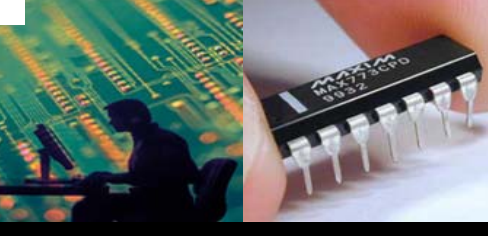
- Nguyên lý: ứng với một tổ hợp cần giải mã ở đầu vào thì đầu ra bằng 1, các tổ hợp đầu vào còn lại, đầu ra bằng 0.
- VD:  $S = 1$  nếu  $(AB) = (10)$ ,  $S = 0$  nếu  $(AB) \neq (10)$



## ■ Giải mã cho toàn bộ mã:

- Nguyên lý: ứng với một tổ hợp nào đó ở đầu vào thì 1 trong các đầu ra bằng 1, các đầu ra còn lại bằng 0.





# Ví dụ - Bộ giải mã BCD

- BCD: mã hóa số nguyên thập phân bằng nhị phân

| Chữ số thập phân | Từ mã nhị phân |
|------------------|----------------|
| 0                | 0000           |
| 1                | 0001           |
| 2                | 0010           |
| 3                | 0011           |
| 4                | 0100           |
| 5                | 0101           |
| 6                | 0110           |
| 7                | 0111           |
| 8                | 1000           |
| 9                | 1001           |



# Bộ giải mã BCD (tiếp)

- Xác định đầu vào và đầu ra:
  - Vào: từ mã nhị phân 4 bit ( $\Rightarrow$  có 16 tổ hợp)
  - Ra: các tín hiệu tương ứng với các số nhị phân mà từ mã mã hóa
- Ta chỉ sử dụng 10 tổ hợp, còn 6 tổ hợp không sử dụng đến được coi là không xác định.



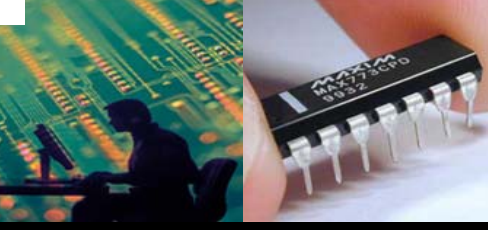
BCD – Binary Coding Decimal



# Bộ giải mã BCD – Bảng thật

| ABCD | S <sub>0</sub> | S <sub>1</sub> | S <sub>2</sub> | S <sub>3</sub> | S <sub>4</sub> | S <sub>5</sub> | S <sub>6</sub> | S <sub>7</sub> | S <sub>8</sub> | S <sub>9</sub> |
|------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 0000 | 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 0001 | 0              | 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 0010 | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 0011 | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 0              | 0              |
| 0100 | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 0              |
| 0101 | 0              | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 0              |
| 0110 | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0              |
| 0111 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0              | 0              |
| 1000 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0              |
| 1001 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              |
| 1010 | -              | -              | -              | -              | -              | -              | -              | -              | -              | -              |
| 1011 | -              | -              | -              | -              | -              | -              | -              | -              | -              | -              |
| 1100 | -              | -              | -              | -              | -              | -              | -              | -              | -              | -              |
| 1101 | -              | -              | -              | -              | -              | -              | -              | -              | -              | -              |
| 1110 | -              | -              | -              | -              | -              | -              | -              | -              | -              | -              |
| 1111 | -              | -              | -              | -              | -              | -              | -              | -              | -              | -              |

□



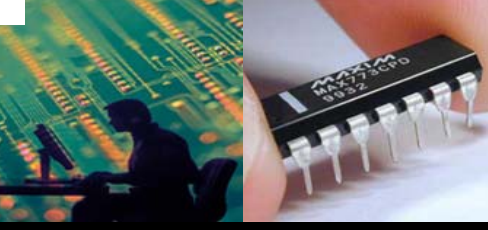
# Tìm biểu thức của từng đầu ra

$$S_0(A,B,C,D) = \bar{A} \bar{B} \bar{C} \bar{D}$$

| CD \ AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 1  | 0  | 0  | 0  |
| 01      | 0  | 0  | 0  | 0  |
| 11      | -  | -  | -  | -  |
| 10      | 0  | 0  | -  | -  |

$$S_1(A,B,C,D) = \bar{A} \bar{B} \bar{C} D$$

| CD \ AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 1  | 0  | 0  |
| 01      | 0  | 0  | 0  | 0  |
| 11      | -  | -  | -  | -  |
| 10      | 0  | 0  | -  | -  |



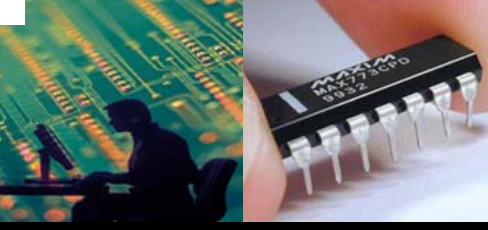
# Tìm biểu thức của từng đầu ra (tiếp)

$$S_2(A,B,C,D) = \bar{B} C \bar{D}$$

| CD \ AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 0  | 0  | 1  |
| 01      | 0  | 0  | 0  | 0  |
| 11      | -  | -  | -  | -  |
| 10      | 0  | 0  | -  | -  |

$$S_3(A,B,C,D) = \bar{B} C D$$

| CD \ AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 0  | 1  | 0  |
| 01      | 0  | 0  | 0  | 0  |
| 11      | -  | -  | -  | -  |
| 10      | 0  | 0  | -  | -  |



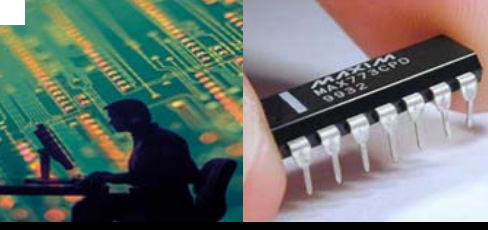
# Tìm biểu thức của từng đầu ra (tiếp)

$$S_4(A,B,C,D) = B \bar{C} \bar{D}$$

| CD \ AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 0  | 0  | 0  |
| 01      | 1  | 0  | 0  | 0  |
| 11      | -  | -  | -  | -  |
| 10      | 0  | 0  | -  | -  |

$$S_5(A,B,C,D) = B \bar{C} D$$

| CD \ AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 0  | 0  | 0  |
| 01      | 0  | 1  | 0  | 0  |
| 11      | -  | -  | -  | -  |
| 10      | 0  | 0  | -  | -  |



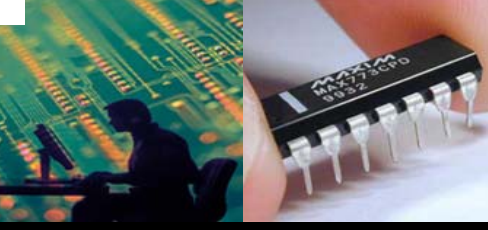
# Tìm biểu thức của từng đầu ra (tiếp)

$$S_6(A,B,C,D) = BC \bar{D}$$

| CD \ AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 0  | 0  | 0  |
| 01      | 0  | 0  | 0  | 1  |
| 11      | -  | -  | -  | -  |
| 10      | 0  | 0  | -  | -  |

$$S_7(A,B,C,D) = BCD$$

| CD \ AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 0  | 0  | 0  |
| 01      | 0  | 0  | 1  | 0  |
| 11      | -  | -  | -  | -  |
| 10      | 0  | 0  | -  | -  |



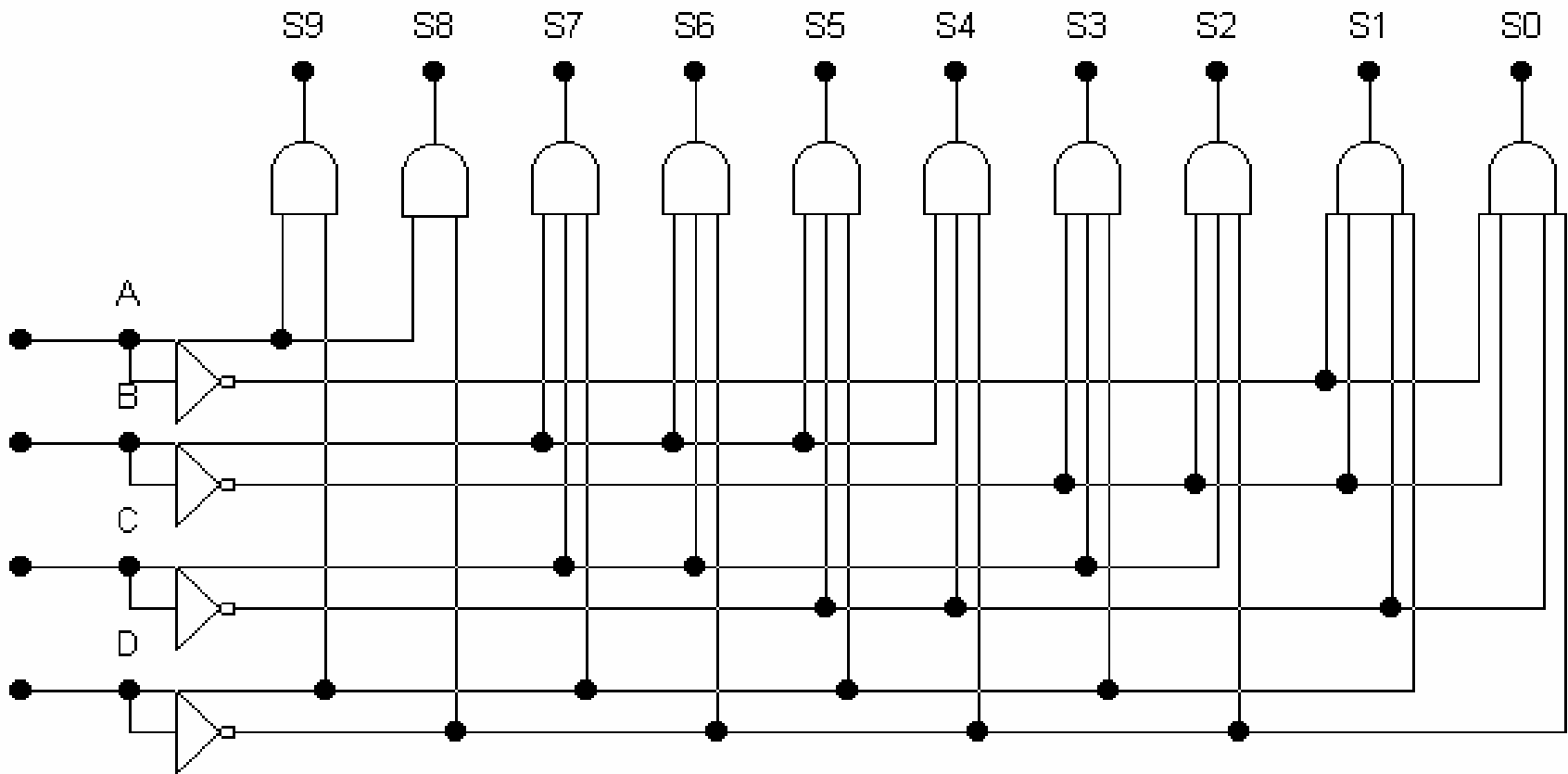
# Tìm biểu thức của từng đầu ra (tiếp)

$$S_8(A,B,C,D) = A\bar{D}$$

| CD \ AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 0  | 0  | 0  |
| 01      | 0  | 0  | 0  | 0  |
| 11      | -  | -  | -  | -  |
| 10      | 1  | 0  | -  | -  |

$$S_9(A,B,C,D) = AD$$

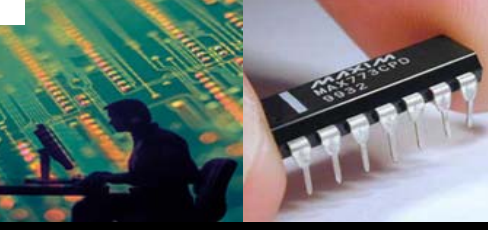
| CD \ AB | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 0  | 0  | 0  |
| 01      | 0  | 0  | 0  | 0  |
| 11      | -  | -  | -  | -  |
| 10      | 0  | 1  | -  | -  |



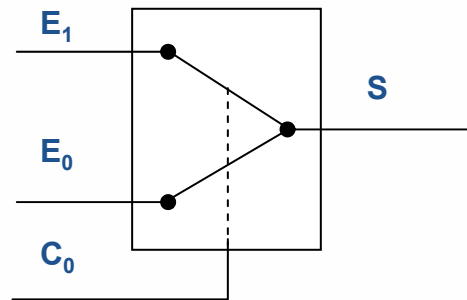
### 3. Bộ chọn kênh

- MultiPlexor – MUX
- Có nhiều đầu vào tín hiệu và 1 đầu ra
- Chức năng: chọn 1 tín hiệu trong nhiều tín hiệu đầu vào để đưa ra đầu ra





- Sơ đồ khối:

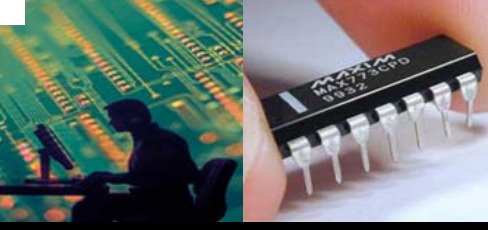


- Tín hiệu chọn:

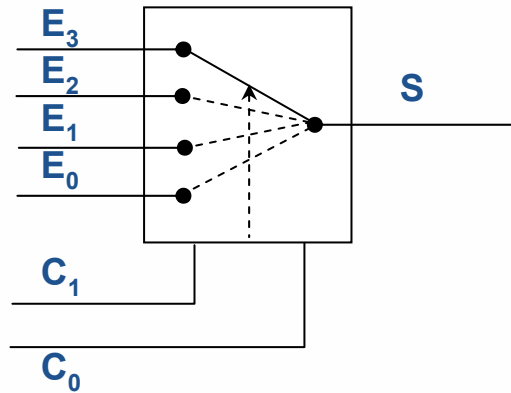
| $C_0$ | $S$   |
|-------|-------|
| 0     | $E_0$ |
| 1     | $E_1$ |

- Tín hiệu ra:

$$S = \overline{C_0}E_0 + C_0E_1$$



- Sơ đồ khối:



- Tín hiệu chọn:

| $C_1$ | $C_0$ | $S$   |
|-------|-------|-------|
| 0     | 0     | $E_0$ |
| 0     | 1     | $E_1$ |
| 1     | 0     | $E_2$ |
| 1     | 1     | $E_3$ |

- Tín hiệu ra:

$$S = \overline{C_1}\overline{C_0}E_0 + \overline{C_1}C_0E_1 + C_1\overline{C_0}E_2 + C_1C_0E_3$$

# Ví dụ - Thiết kế MUX 2-1

- Bảng thật:

| $C_0$ | $E_1$ | $E_0$ | $S$ |
|-------|-------|-------|-----|
| 0     | 0     | 0     | 0   |
| 0     | 0     | 1     | 1   |
| 0     | 1     | 0     | 0   |
| 0     | 1     | 1     | 1   |
| 1     | 0     | 0     | 0   |
| 1     | 0     | 1     | 0   |
| 1     | 1     | 0     | 1   |
| 1     | 1     | 1     | 1   |

# Ví dụ - Thiết kế MUX 2-1 (tiếp)

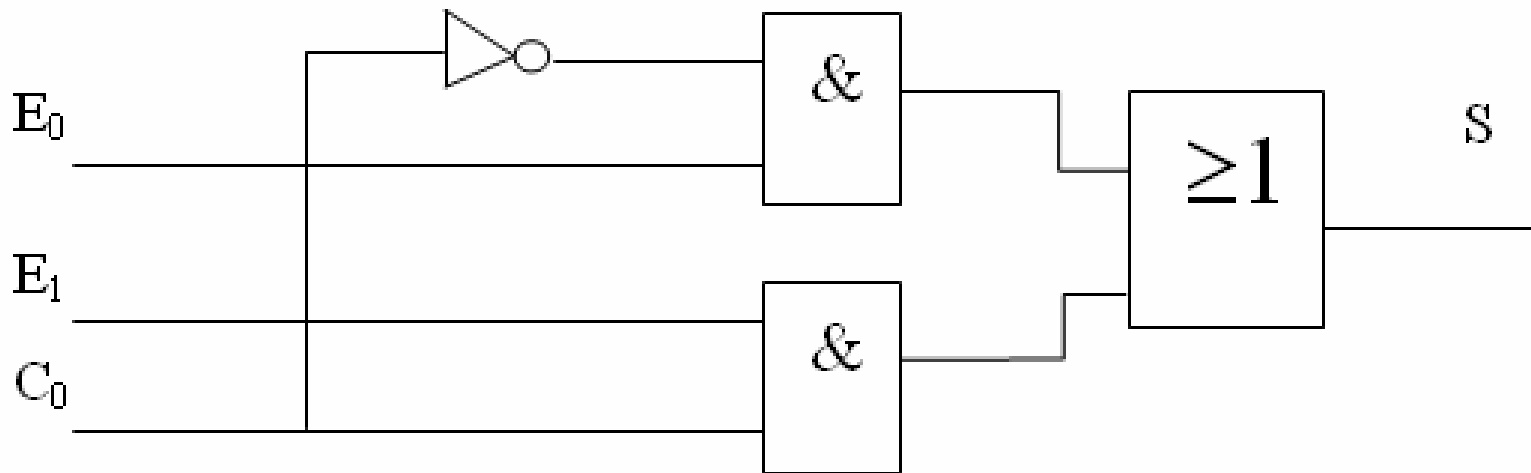
- Biểu thức đầu ra S:

$$S = \overline{C_0} E_0 + C_0 E_1$$

| $C_0 \backslash E_1 E_0$ | 00 | 01 | 11 | 10 |
|--------------------------|----|----|----|----|
| 0                        | 0  | 1  | 1  | 0  |
| 1                        | 0  | 0  | 1  | 1  |

# Ví dụ - Thiết kế MUX 2-1 (tiếp)

- Sơ đồ mạch:



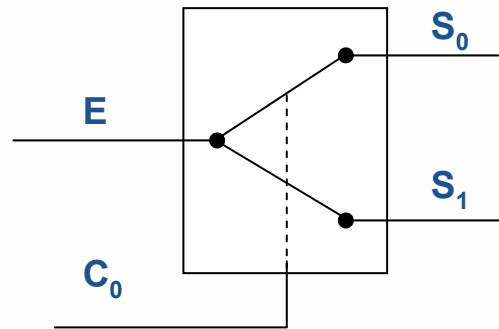
Minh họa



## 4. Bộ phân kênh

- DeMultiPlexor – DeMUX
- Có 1 đầu vào tín hiệu và nhiều đầu ra
- Chức năng: đưa tín hiệu từ đầu vào tới 1 trong những đầu ra

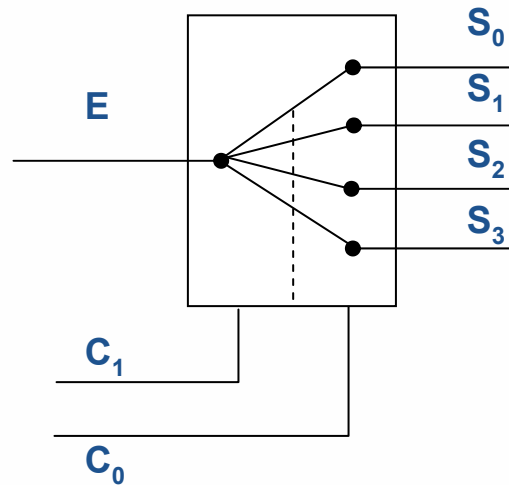
- Sơ đồ khối:



- Tín hiệu chọn:

| $C_0$ | $S_0$ | $S_1$ |
|-------|-------|-------|
| 0     | E     | 0     |
| 1     | 0     | E     |

- Sơ đồ khối:



- Tín hiệu chọn:

| $C_1$ | $C_0$ | $S_0$ | $S_1$ | $S_2$ | $S_3$ |
|-------|-------|-------|-------|-------|-------|
| 0     | 0     | E     | 0     | 0     | 0     |
| 0     | 1     | 0     | E     | 0     | 0     |
| 1     | 0     | 0     | 0     | E     | 0     |
| 1     | 1     | 0     | 0     | 0     | E     |



# Ví dụ - Thiết kế DeMUX 1-2

- Bảng thật:

| $C_0$ | $S_0$ | $S_1$ |
|-------|-------|-------|
| 0     | E     | 0     |
| 1     | 0     | E     |

| $C_0$ | E | $S_0$ | $S_1$ |
|-------|---|-------|-------|
| 0     | 0 | 0     | 0     |
| 0     | 1 | 1     | 0     |
| 1     | 0 | 0     | 0     |
| 1     | 1 | 0     | 1     |

- Biểu thức đầu ra:

$$S_0 = \overline{C_0}E$$

$$S_1 = C_0E$$

## 5. Các mạch số học

- a. Bộ cộng
- b. Bộ trừ
- c. Bộ so sánh

- Chức năng: thực hiện phép cộng giữa 2 số nhị phân.
- Bán tổng (Half-Adder):
  - Thực hiện phép cộng giữa 2 bit thấp nhất của phép cộng 2 số nhị phân.
  - Sơ đồ khối:



- Bảng thật:

| $a_i$ | $b_i$ | $s_i$ | $r_{i+1}$ |
|-------|-------|-------|-----------|
| 0     | 0     | 0     | 0         |
| 0     | 1     | 1     | 0         |
| 1     | 0     | 1     | 0         |
| 1     | 1     | 0     | 1         |

- Biểu thức đầu ra phụ thuộc đầu vào:

$$s_i = a_i \oplus b_i$$

$$r_{i+1} = a_i \cdot b_i$$

- Sơ đồ mạch: ...

Minh họa



Mạch test



# Bộ cộng đầy đủ (Full-Adder)

- Chức năng: thực hiện phép cộng giữa 2 bit bất kỳ của phép cộng 2 số nhị phân.
- Sơ đồ khối:
  - $r_i$ : bit nhớ đầu vào
  - $r_{i+1}$ : bit nhớ đầu ra



- Bảng thật:

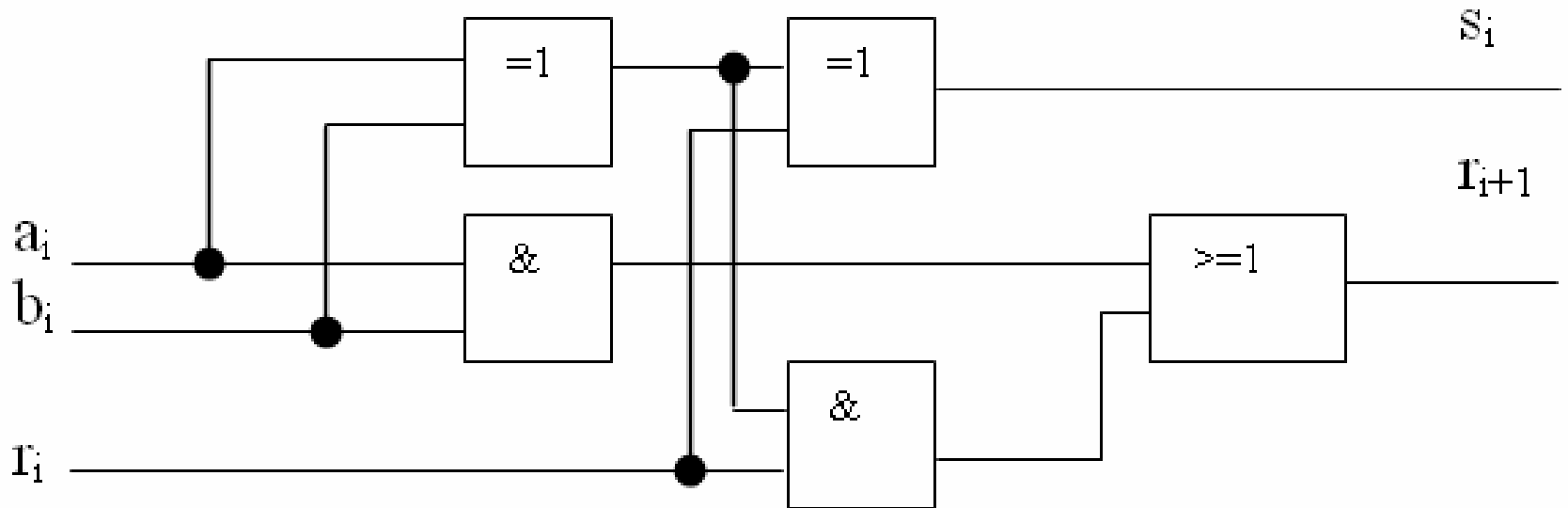
| $a_i$ | $b_i$ | $r_i$ | $s_i$ | $r_{i+1}$ |
|-------|-------|-------|-------|-----------|
| 0     | 0     | 0     | 0     | 0         |
| 0     | 0     | 1     | 1     | 0         |
| 0     | 1     | 0     | 1     | 0         |
| 0     | 1     | 1     | 0     | 1         |
| 1     | 0     | 0     | 1     | 0         |
| 1     | 0     | 1     | 0     | 1         |
| 1     | 1     | 0     | 0     | 1         |
| 1     | 1     | 1     | 1     | 1         |

- Biểu thức đầu ra phụ thuộc đầu vào:

$$s_i = a_i \oplus b_i \oplus r_i$$

$$r_{i+1} = a_i \cdot b_i + r_i(a_i + b_i)$$

- Sơ đồ mạch:



Minh họa

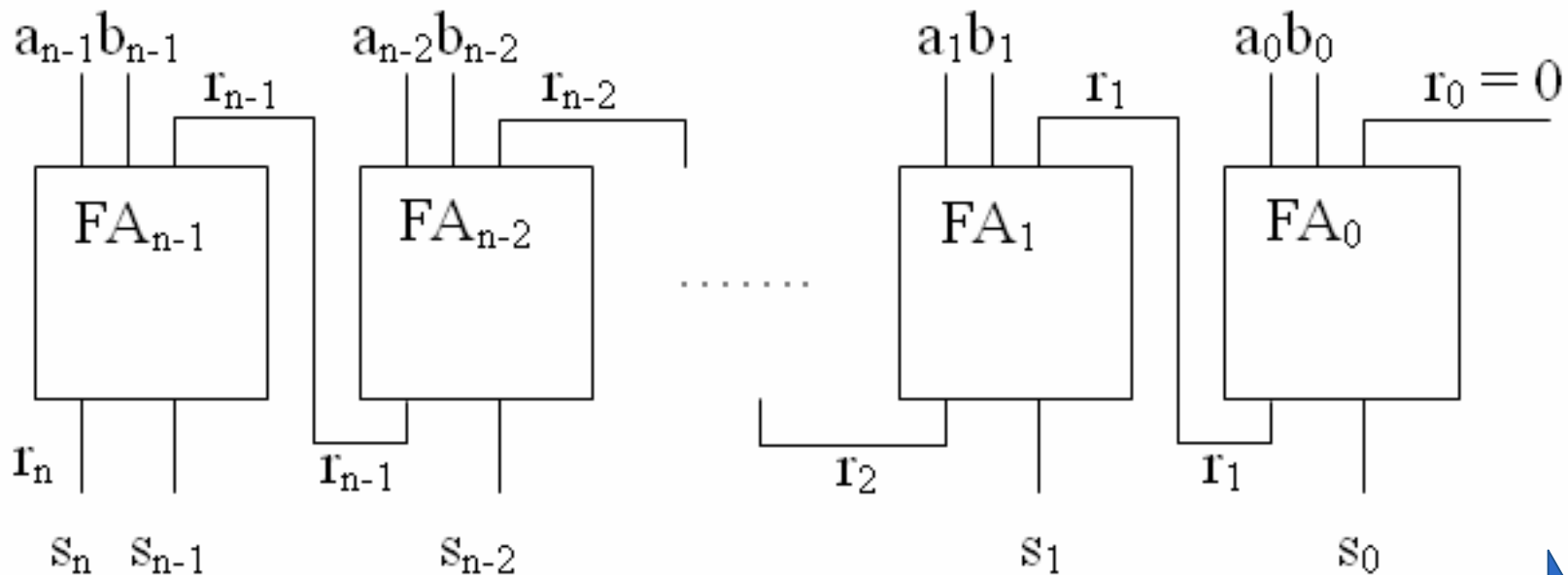


Mạch test



# Bộ cộng nhiều bit

- Đây là bộ cộng 2 số nhị phân  $n$  bit, kết quả nhận được là 1 số nguyên  $n+1$  bit.
- Sơ đồ:

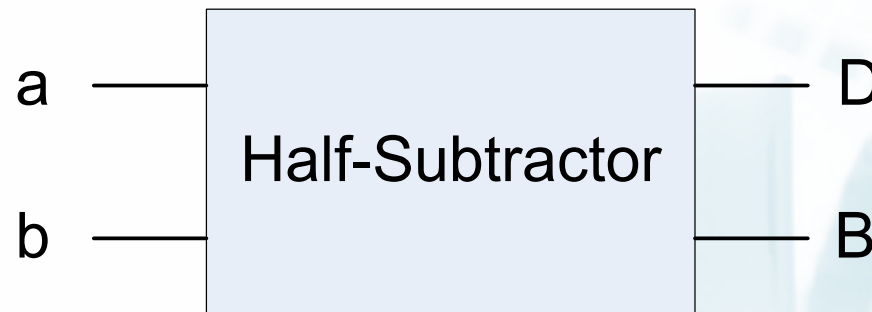


Minh họa

Mạch test



- Chức năng: thực hiện phép trừ giữa 2 số nhị phân.
- Bán hiệu (Half-Subtractor):
  - Dùng để thực hiện phép trừ giữa 2 bit thấp nhất trong phép trừ giữa 2 số nhị phân
  - Sơ đồ khối:
    - $D_i$ : hiệu
    - $B_{i+1}$ : bit mượn



- Bảng thật:

| $a_i$ | $b_i$ | $D_i$ | $B_{i+1}$ |
|-------|-------|-------|-----------|
| 0     | 0     | 0     | 0         |
| 0     | 1     | 1     | 1         |
| 1     | 0     | 1     | 0         |
| 1     | 1     | 0     | 0         |

- Biểu thức đầu ra phụ thuộc đầu vào:

$$D_i = a_i \oplus b_i$$

$$B_{i+1} = \overline{a_i} \cdot b_i$$

- Sơ đồ mạch: ...

Minh họa

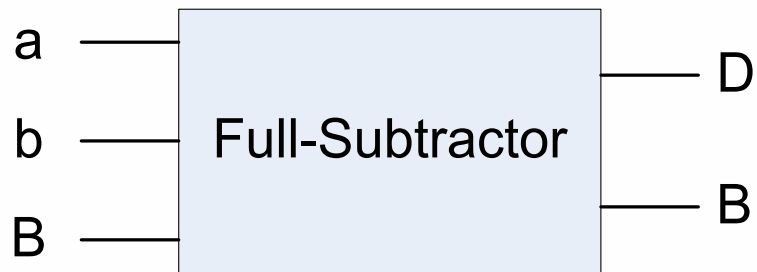


Mạch test



# Bộ trừ đầy đủ (Full-Subtractor)

- Chức năng: dùng để thực hiện phép trừ giữa 2 bit bất kỳ trong phép trừ 2 số nhị phân.
- Sơ đồ khối:



- Bảng thật:

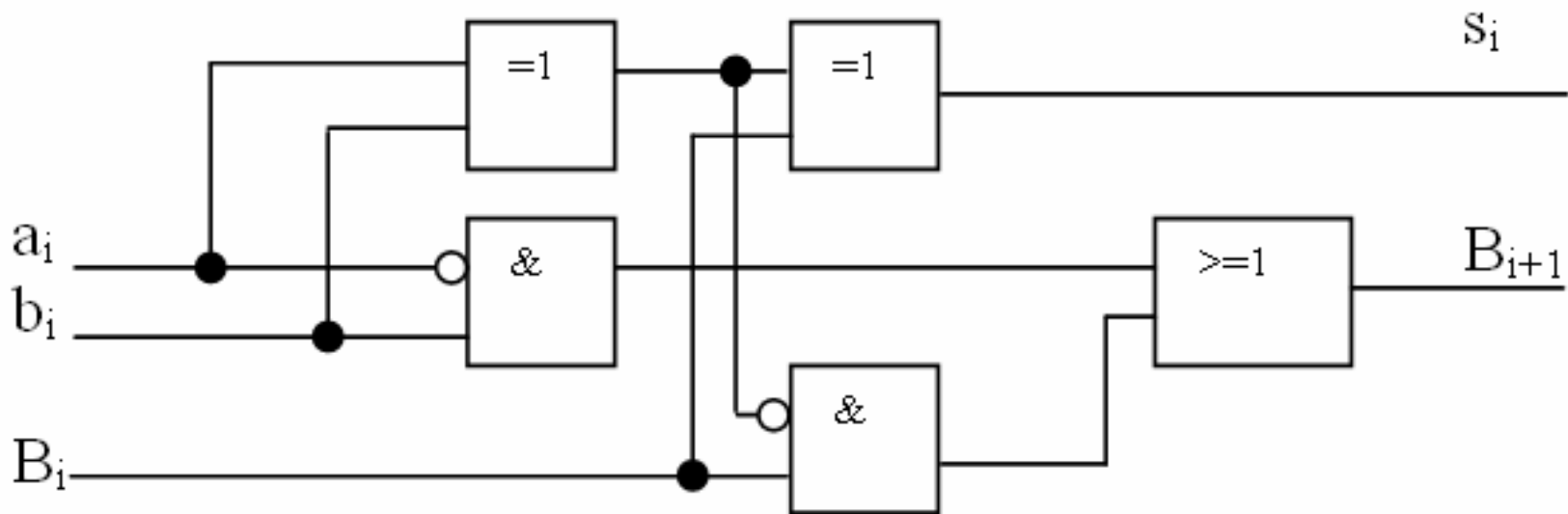
| $B_i$ | $a_i$ | $b_i$ | $D_i$ | $B_{i+1}$ |
|-------|-------|-------|-------|-----------|
| 0     | 0     | 0     | 0     | 0         |
| 0     | 0     | 1     | 1     | 1         |
| 0     | 1     | 0     | 1     | 0         |
| 0     | 1     | 1     | 0     | 0         |
| 1     | 0     | 0     | 1     | 1         |
| 1     | 0     | 1     | 0     | 1         |
| 1     | 1     | 0     | 0     | 0         |
| 1     | 1     | 1     | 1     | 1         |

- Biểu thức đầu ra phụ thuộc đầu vào:

$$D_i = a_i \oplus b_i \oplus B_i$$

$$B_{i+1} = \overline{a_i} \cdot b_i + B_i \cdot \overline{(a_i \oplus b_i)}$$

- Sơ đồ mạch:



Minh họa



Mạch test



- Dùng để so sánh 2 số nhị phân
- Có 2 kiểu so sánh:
  - So sánh đơn giản:
    - Kết quả so sánh: bằng nhau, khác nhau
  - So sánh đầy đủ:
    - Kết quả so sánh: lớn hơn, nhỏ hơn, bằng nhau
- Có 2 loại bộ so sánh:
  - Bộ so sánh đơn giản
  - Bộ so sánh đầy đủ

# Bộ so sánh đơn giản

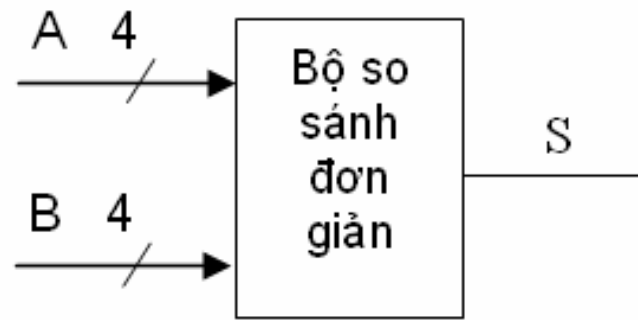
- Giả sử cần xây dựng bộ so sánh đơn giản 2 số A và B:

A       $a_3 a_2 a_1 a_0$

B       $b_3 b_2 b_1 b_0$

Đầu ra S

- $S = 1 \iff A = B$
- $S = 0 \iff A \neq B$



# Bộ so sánh đơn giản (tiếp)

- Ta có:

$$A = B \Leftrightarrow \begin{cases} a_3 = b_3 \\ a_2 = b_2 \\ a_1 = b_1 \\ a_0 = b_0 \end{cases} \Leftrightarrow \begin{cases} a_3 \oplus b_3 = 0 \\ a_2 \oplus b_2 = 0 \\ a_1 \oplus b_1 = 0 \\ a_0 \oplus b_0 = 0 \end{cases} \Leftrightarrow \begin{cases} \overline{a_3 \oplus b_3} = 1 \\ \overline{a_2 \oplus b_2} = 1 \\ \overline{a_1 \oplus b_1} = 1 \\ \overline{a_0 \oplus b_0} = 1 \end{cases}$$

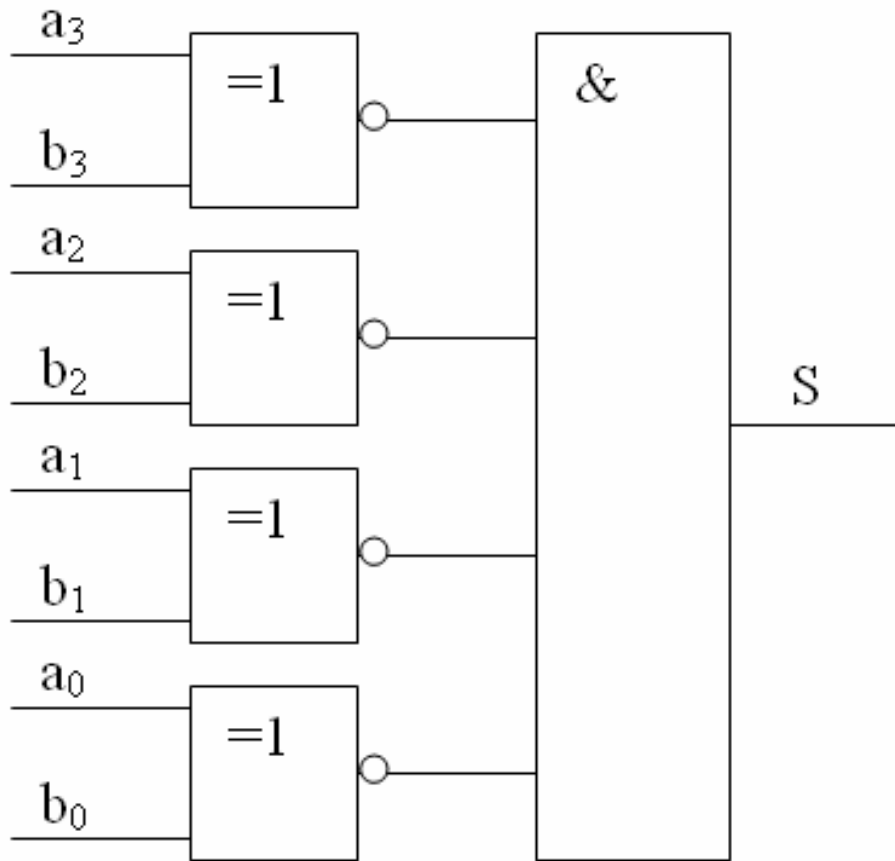
- Suy ra:

$$S = \overline{a_3 \oplus b_3} \cdot \overline{a_2 \oplus b_2} \cdot \overline{a_1 \oplus b_1} \cdot \overline{a_0 \oplus b_0}$$



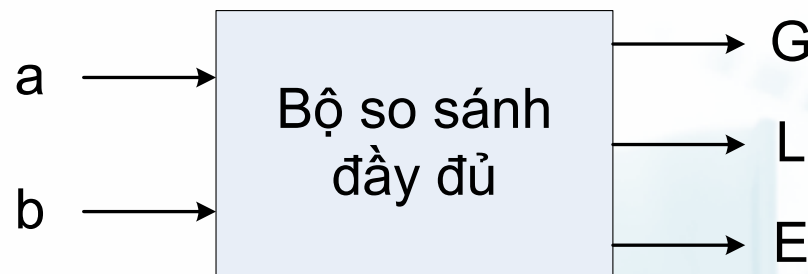
# Bộ so sánh đơn giản (tiếp)

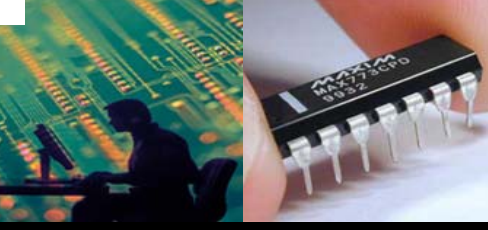
- Sơ đồ mạch:



## ▪ Bộ so sánh 2 bit đầy đủ:

- Đầu vào: 2 bit cần so sánh  $a_i$  và  $b_i$
- Đầu ra: 3 tín hiệu để báo kết quả lớn hơn, nhỏ hơn, bằng nhau của 2 bit
  - $a_i > b_i \Leftrightarrow G_i = 1$  còn  $E_i, L_i = 0$
  - $a_i < b_i \Leftrightarrow L_i = 1$  còn  $E_i, G_i = 0$
  - $a_i = b_i \Leftrightarrow E_i = 1$  còn  $G_i, L_i = 0$
- Sơ đồ khối:





# Bộ so sánh 2 bit đầy đủ (tiếp)

- Bảng thật:

| $a_i$ | $b_i$ | $G_i$ | $L_i$ | $E_i$ |
|-------|-------|-------|-------|-------|
| 0     | 0     | 0     | 0     | 1     |
| 0     | 1     | 0     | 1     | 0     |
| 1     | 0     | 1     | 0     | 0     |
| 1     | 1     | 0     | 0     | 1     |

- Biểu diễn đầu ra theo đầu vào:

$$G_i = a_i \cdot \overline{b_i}$$

$$L_i = \overline{a_i} \cdot b_i$$

$$E_i = \overline{a_i \oplus b_i}$$

- Sơ đồ mạch: ...

Minh họa



# Bộ so sánh đầy đủ 2 số nhị phân

- Cấu tạo: gồm các bộ so sánh 2 bit
- Có tín hiệu CS (Chip Select)
  - CS = 0, tất cả các đầu ra = 0 (không so sánh)
  - CS = 1, hoạt động bình thường
- Biểu diễn các đầu ra của bộ so sánh 2 bit theo đầu vào:

$$G_i = CS \cdot a_i \cdot \overline{b_i}$$

$$L_i = CS \cdot \overline{a_i} \cdot b_i$$

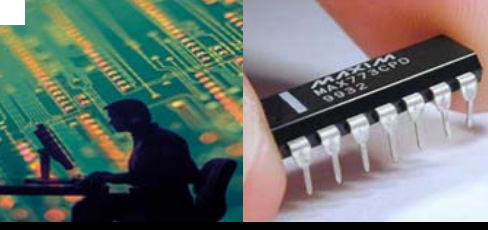
$$E_i = CS \cdot \overline{(a_i \oplus b_i)}$$

Minh họa



Mạch test

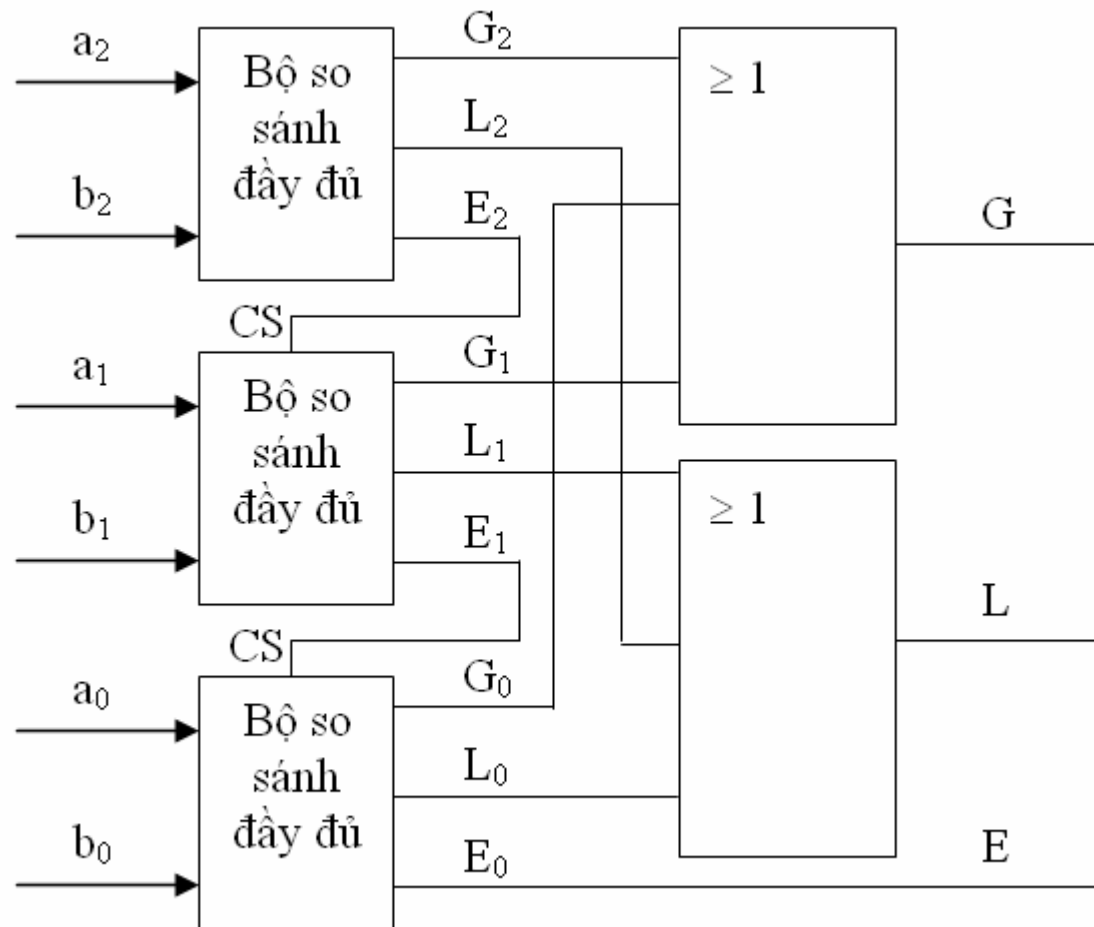




# VD: Bộ so sánh 2 số nhị phân 3 bit

## ▪ Sơ đồ mạch bộ so sánh 2 số nhị phân 3 bit:

- $A = a_2a_1a_0$
- $B = b_2b_1b_0$

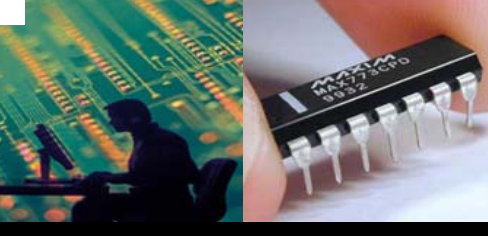


Minh họa



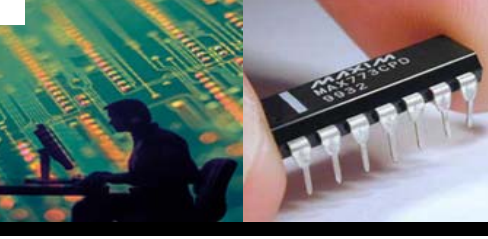
Mạch test





- **Bài 1:** Tổng hợp bộ chọn kênh 4-1.
- **Bài 2:** Thiết kế bộ trừ/nhân 2 số 2 bit.
- **Bài 3:** Tổng hợp bộ chọn kênh 2-1 chỉ dùng NAND.
- **Bài 4:** Tổng hợp mạch tổ hợp thực hiện phép toán sau :  $M = N + 3$ , biết rằng N là số 4 bit mã BCD còn M là số 4 bit.





# Chương 5

## HỆ DÂY

Bộ môn Kỹ thuật Máy tính, Khoa Công nghệ Thông tin  
Trường Đại học Bách Khoa Hà Nội

## 5.1. Khái niệm

5.2. Mô hình của hệ dây

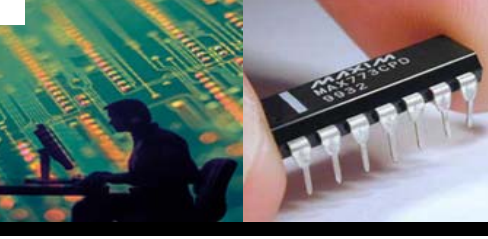
5.3. Các Trigger

5.4. Một số ứng dụng của hệ dây



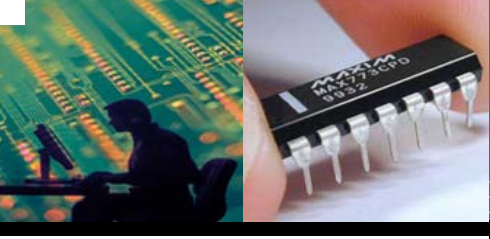
## 5.1. Khái niệm

- Hệ dãn là hệ mà tín hiệu ra không chỉ phụ thuộc vào tín hiệu vào tại thời điểm hiện tại mà còn phụ thuộc vào quá khứ của tín hiệu vào.
- Hệ dãn còn được gọi là hệ có nhớ.
- Để thực hiện được hệ dãn, nhất thiết phải có phần tử nhớ. Ngoài ra còn có thể có các phần tử logic cơ bản.



- Hệ dây đồng bộ: khi làm việc cần có 1 tín hiệu đồng bộ để giữ nhịp cho toàn bộ hệ hoạt động.
- Hệ dây không đồng bộ: không cần tín hiệu này để giữ nhịp chung cho toàn bộ hệ hoạt động.
- Hệ dây đồng bộ nhanh hơn hệ dây không đồng bộ tuy nhiên lại có thiết kế phức tạp hơn.





5.1. Khái niệm

5.2. Mô hình của hệ dây

5.3. Các Trigger

5.4. Một số ứng dụng của hệ dây



# Mô hình của hệ dĩa

- Mô hình của hệ dĩa được dùng để mô tả hệ dĩa thông qua tín hiệu vào, tín hiệu ra và trạng thái của hệ mà không quan tâm đến cấu trúc bên trong của hệ.



# Mô hình của hệ dây (tiếp)

- Có 2 loại mô hình:
  - Mealy
  - Moore
- Hai loại mô hình trên có thể chuyển đổi qua lại cho nhau.

## a. Mô hình Mealy

- Mô hình Mealy mô tả hệ dãy thông qua 5 tham số:
  - $X = \{x_1, x_2, \dots, x_n\}$
  - $Y = \{y_1, y_2, \dots, y_l\}$
  - $S = \{s_1, s_2, \dots, s_m\}$
  - $F_S(S, X)$
  - $F_Y(S, X)$

## ■ Giải thích các kí hiệu:

- $X$  là tập hợp hữu hạn  $n$  tín hiệu đầu vào
- $Y$  là tập hợp hữu hạn  $l$  tín hiệu đầu ra
- $S$  tập hợp hữu hạn  $m$  trạng thái trong của hệ
- $F_S$  là hàm biến đổi trạng thái. Đối với mô hình kiểu Mealy thì  $F_S$  phụ thuộc vào  $S$  và  $X \rightarrow F_S = F_S(S, X)$
- $F_Y$  là hàm tính trạng thái đầu ra:  $F_Y = F_Y(S, X)$

## b. Mô hình Moore

- Mô hình Moore giống như mô hình Mealy, nhưng khác ở chỗ là  $F_Y$  chỉ phụ thuộc vào  $S$ :

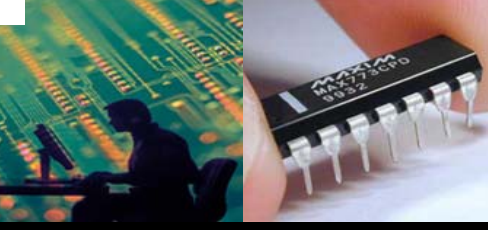
$$F_Y = F_Y(S)$$



# Bảng chuyển trạng thái

- Mô hình Mealy:

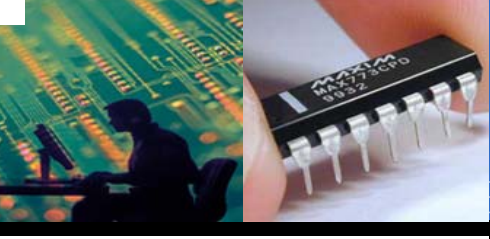
| S     | X                              |     |                                |
|-------|--------------------------------|-----|--------------------------------|
|       | $x_1$                          | ... | $x_n$                          |
| $s_1$ | $F_S(s_1, x_1), F_Y(s_1, x_1)$ | ... | $F_S(s_1, x_n), F_Y(s_1, x_n)$ |
| ...   | ...                            | ... | ...                            |
| $s_m$ | $F_S(s_m, x_1), F_Y(s_m, x_1)$ | ... | $F_S(s_m, x_n), F_Y(s_m, x_n)$ |



# Bảng chuyển trạng thái (tiếp)

- Mô hình Moore:

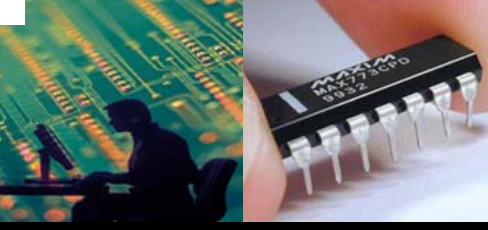
| S     | X               |     |                 | Y          |
|-------|-----------------|-----|-----------------|------------|
|       | $x_1$           | ... | $x_n$           |            |
| $s_1$ | $F_S(s_1, x_1)$ | ... | $F_S(s_1, x_n)$ | $F_Y(s_1)$ |
| ...   | ...             | ... | ...             | ...        |
| $s_m$ | $F_S(s_m, x_1)$ |     | $F_S(s_m, x_n)$ | $F_Y(s_m)$ |



# Ví dụ về mô hình hệ dãy

- Sử dụng mô hình Mealy và Moore để mô tả hệ dãy thực hiện phép cộng.
- Ví dụ:

|   | $t_4$ | $t_3$ | $t_2$ | $t_1$ | $t_0$ |
|---|-------|-------|-------|-------|-------|
| A | 0     | 1     | 1     | 0     | 0     |
| B | 0     | 1     | 1     | 1     | 0     |
| S | 1     | 1     | 0     | 1     | 0     |



# Ví dụ: Mô hình Mealy

- $X = \{00, 01, 10, 11\}$  - do có 2 đầu vào
- $Y = \{0, 1\}$  - do có 1 đầu ra
- $S = \{s_0, s_1\}$ 
  - $s_0$ : trạng thái không nhớ
  - $s_1$ : trạng thái có nhớ

- Hàm trạng thái  $F_S(S, X)$ :

$$F_S(s_0, 00) = s_0$$

$$F_S(s_0, 11) = s_1$$

$$F_S(s_1, 00) = s_0$$

$$F_S(s_1, 01) = s_1$$

$$F_S(s_0, 01) = s_0$$

$$F_S(s_0, 10) = s_0$$

$$F_S(s_1, 10) = s_1$$

$$F_S(s_1, 11) = s_1$$

# Ví dụ: Mô hình Mealy (tiếp)

- Hàm ra  $F_Y(S, X)$ :

$$F_Y(s_0, 00) = 0$$

$$F_Y(s_0, 01) = 1$$

$$F_Y(s_1, 00) = 1$$

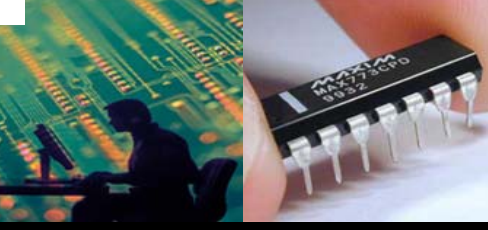
$$F_Y(s_1, 11) = 1$$

$$F_Y(s_0, 11) = 0$$

$$F_Y(s_0, 10) = 1$$

$$F_Y(s_1, 10) = 0$$

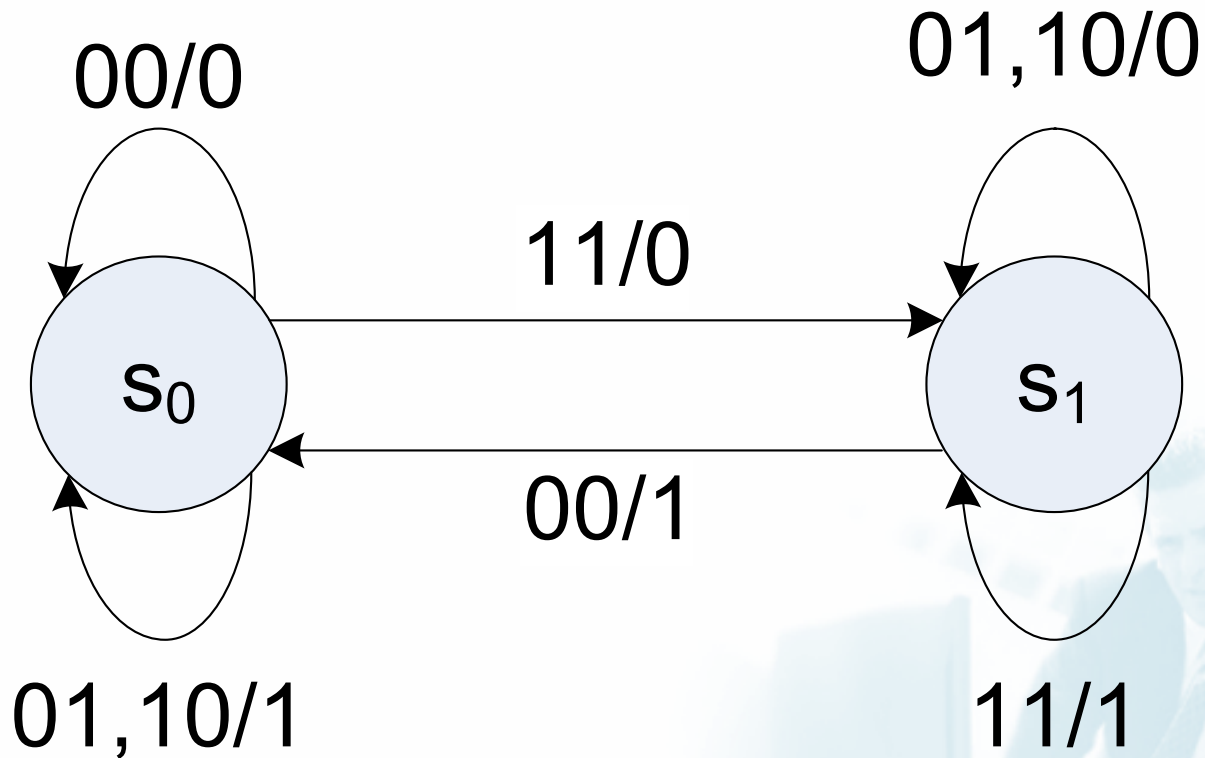
$$F_Y(s_1, 01) = 0$$



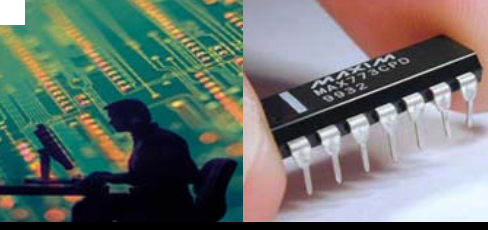
# Bảng chuyển trạng thái

| $S$   | $X$      |          |          |          |
|-------|----------|----------|----------|----------|
|       | 00       | 01       | 10       | 11       |
| $s_0$ | $s_0, 0$ | $s_0, 1$ | $s_0, 1$ | $s_1, 0$ |
| $s_1$ | $s_0, 1$ | $s_1, 0$ | $s_1, 0$ | $s_1, 1$ |

# Đồ hình chuyển trạng thái



# Ví dụ: Mô hình Moore



- $X = \{00, 01, 10, 11\}$  - do có 2 đầu vào
- $Y = \{0, 1\}$  - do có 1 đầu ra
- $S = \{s_{00}, s_{01}, s_{10}, s_{11}\}$  -  $s_{ij}$ :
  - $i = 0$  là không nhớ
  - $i = 1$  là có nhớ
  - $j$  = tín hiệu ra





# Ví dụ: Mô hình Moore (tiếp)

- Hàm trạng thái  $F_S(S, X)$ :

$$F_S(s_{00}, \underline{00}) = s_{00}$$

$$F_S(s_{00}, 01) = s_{01}$$

$$F_S(s_{01}, 00) = s_{00}$$

$$F_S(s_{01}, 01) = s_{01}$$

$$F_S(s_{10}, 00) = s_{01}$$

$$F_S(s_{10}, 01) = s_{10}$$

$$F_S(s_{11}, 00) = s_{01}$$

$$F_S(s_{11}, 11) = s_{11}$$

$$F_S(s_{00}, 10) = s_{01}$$

$$F_S(s_{00}, 11) = s_{10}$$

$$F_S(s_{01}, 10) = s_{01}$$

$$F_S(s_{01}, 11) = s_{10}$$

$$F_S(s_{10}, 10) = s_{10}$$

$$F_S(s_{10}, 11) = s_{11}$$

$$F_S(s_{11}, 01) = s_{10}$$

$$F_S(s_{11}, 10) = s_{10}$$

- Hàm ra  $F_Y(S)$ :

$$F_Y(s_{00}) = \underline{0}$$

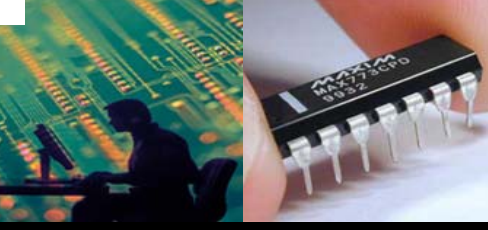
$$F_Y(s_{10}) = 0$$

$$F_Y(s_{01}) = 1$$

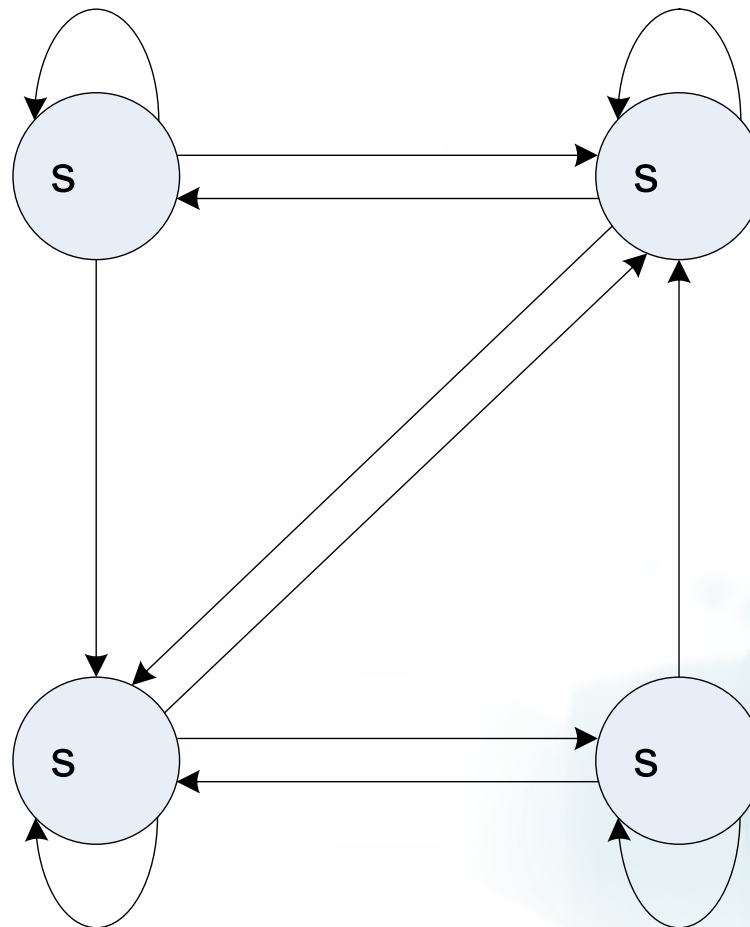
$$F_Y(s_{11}) = 1$$

# Bảng chuyển trạng thái

| S        | X        |          |          |          | Y |
|----------|----------|----------|----------|----------|---|
|          | 00       | 01       | 10       | 11       |   |
| $s_{00}$ | $s_{00}$ | $s_{01}$ | $s_{01}$ | $s_{10}$ | 0 |
| $s_{01}$ | $s_{00}$ | $s_{01}$ | $s_{01}$ | $s_{10}$ | 1 |
| $s_{10}$ | $s_{01}$ | $s_{10}$ | $s_{10}$ | $s_{11}$ | 0 |
| $s_{11}$ | $s_{01}$ | $s_{10}$ | $s_{10}$ | $s_{11}$ | 1 |



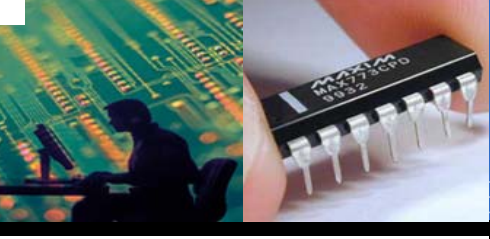
# Đồ hình chuyển trạng thái



00

01

00/0



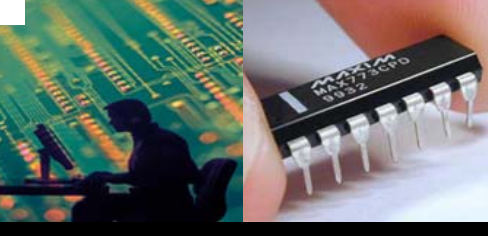
5.1. Khái niệm

5.2. Mô hình của hệ dây

**5.3. Các Trigger**

5.4. Một số ứng dụng của hệ dây





- Phần tử cơ bản của hệ dây chính là các phần tử nhớ hay còn gọi là các trigger
- Đầu ra của trigger chính là trạng thái của nó
- Một trigger có thể làm việc theo 2 kiểu:
  - Trigger không đồng bộ: đầu ra của trigger thay đổi chỉ phụ thuộc vào tín hiệu đầu vào
  - Trigger đồng bộ: đầu ra của trigger thay đổi phụ thuộc vào tín hiệu vào và tín hiệu đồng bộ



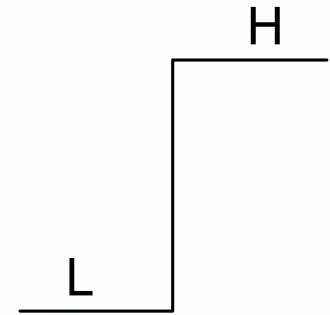
## ■ Đồng bộ theo mức:

### ○ Mức cao:

- Khi tín hiệu đồng bộ có giá trị logic bằng 0 thì hệ nghỉ (giữ nguyên trạng thái)
- Khi tín hiệu đồng bộ có giá trị logic bằng 1 thì hệ làm việc bình thường.

### ○ Mức thấp:

- Khi tín hiệu đồng bộ có giá trị logic bằng 1 thì hệ nghỉ (giữ nguyên trạng thái)
- Khi tín hiệu đồng bộ có giá trị logic bằng 0 thì hệ làm việc bình thường.



Đồng bộ theo mức

# Các kiểu đồng bộ (tiếp)

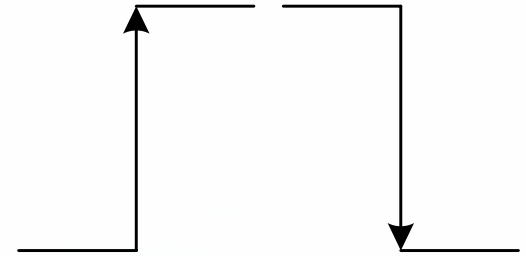
## ▪ Đồng bộ theo sườn:

### ○ Sườn dương:

- Khi tín hiệu đồng bộ xuất hiện sườn dương (sườn đi lên, từ 0  $\rightarrow$  1) thì hệ làm việc bình thường
- Trong các trường hợp còn lại, hệ nghỉ (giữ nguyên trạng thái).

### ○ Sườn âm:

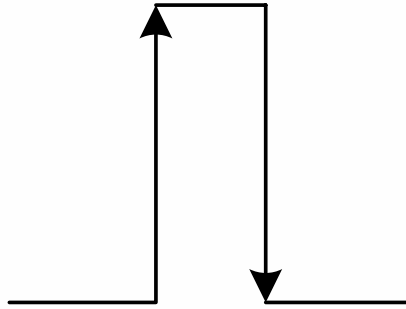
- Khi tín hiệu đồng bộ xuất hiện sườn âm (sườn đi xuống, từ 1  $\rightarrow$  0), hệ làm việc bình thường
- Trong các trường hợp còn lại, hệ nghỉ (giữ nguyên trạng thái).



Đồng bộ theo sườn

# Các kiểu đồng bộ (tiếp)

- Đồng bộ kiểu xung:
  - Khi có xung thì hệ làm việc bình thường
  - Khi không có xung thì hệ nghỉ (giữ nguyên trạng thái).



Đồng bộ kiểu xung

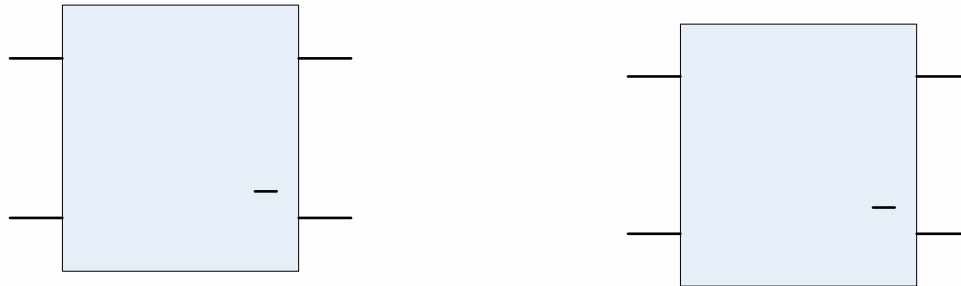


## ■ Có 4 loại Trigger:

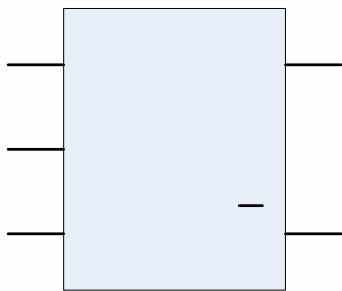
- RS      Reset - Set      Xóa - Thiết lập
- D      Delay      Trễ
- JK      Jordan và Kelly      Tên 2 nhà phát minh
- T      Toggle      Bật bênh, bật tắt

# a. Trigger RS

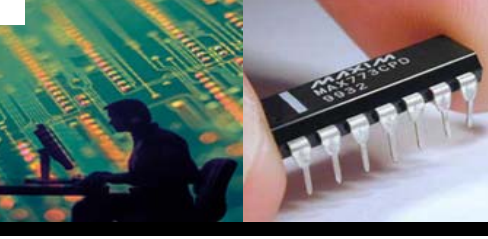
- Sơ đồ khối:



- Trigger RS hoạt động được ở cả 2 chế độ đồng bộ và không đồng bộ



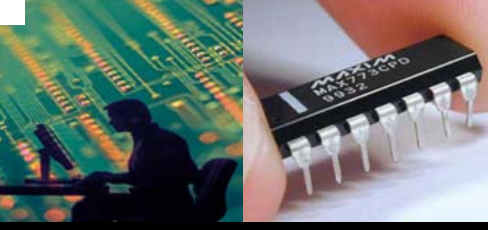
Đồng bộ sườn âm



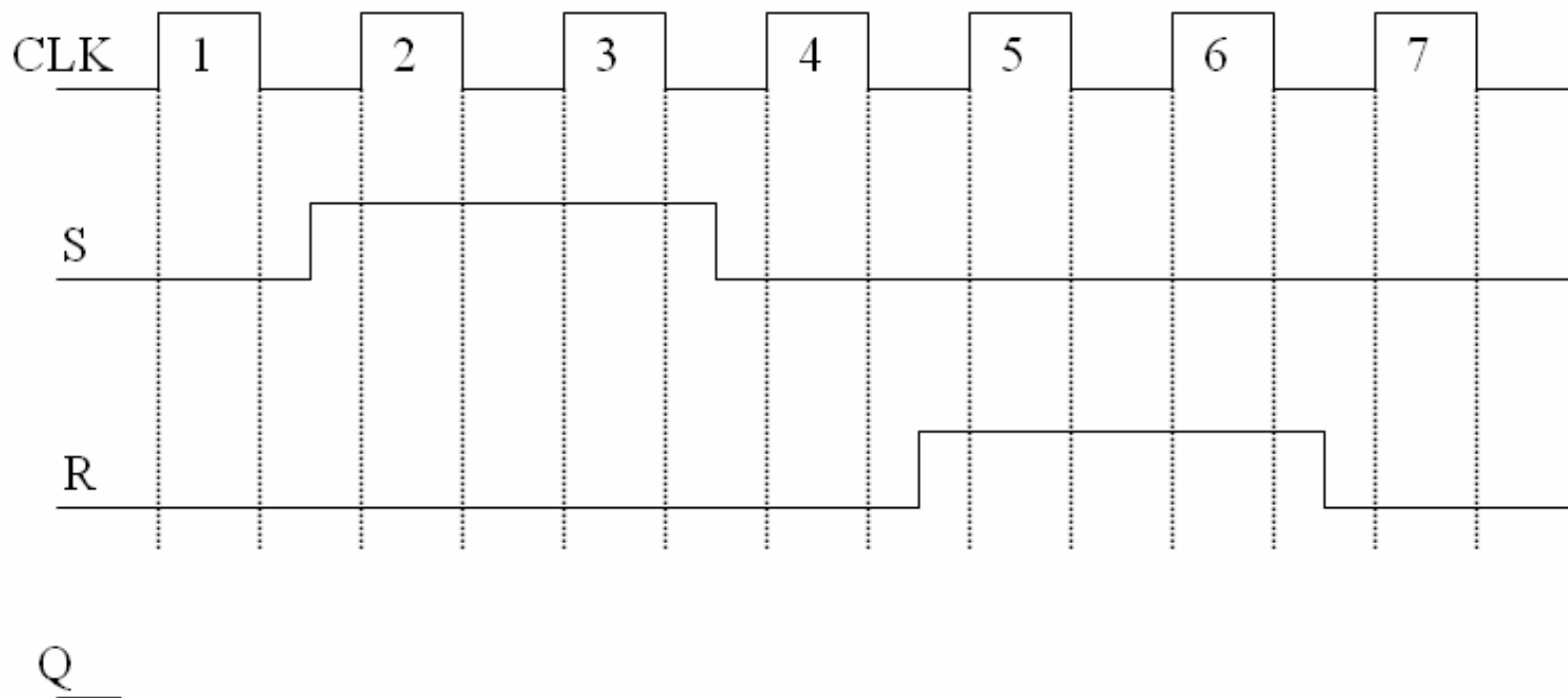
# Bảng chuyển trạng thái của RS

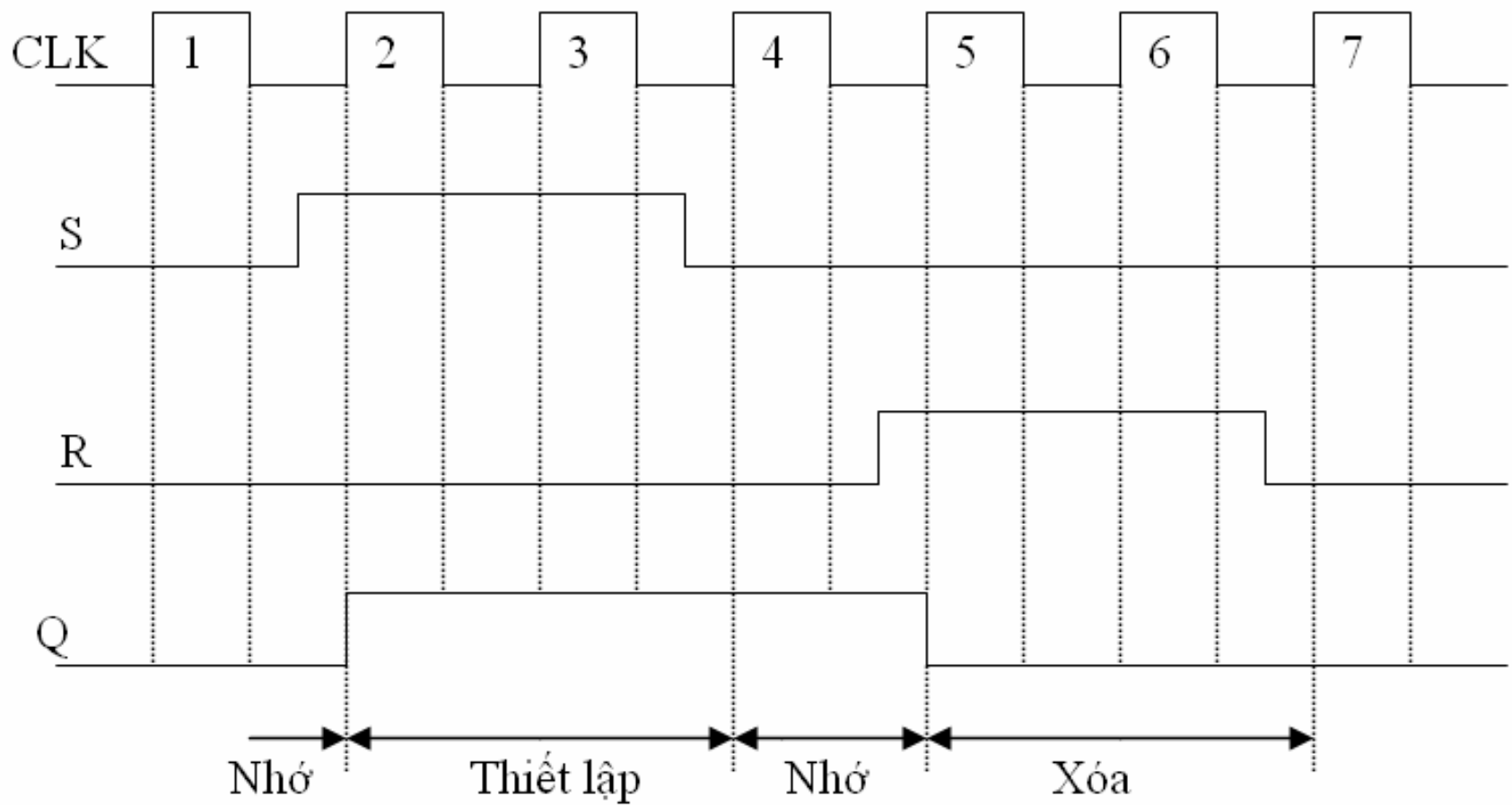
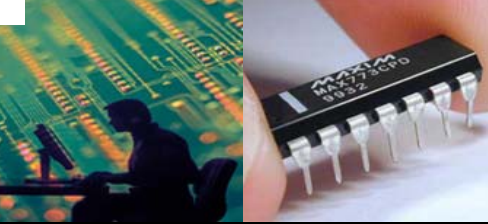
| q \ RS | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0      | 0  | 1  | -  | 0  |
| 1      | 1  | 1  | -  | 0  |

$$Q = S + q\bar{R}$$



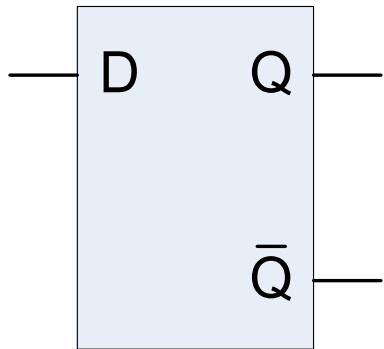
- Cho Trigger RS đồng bộ mức cao và đồ thị các tín hiệu R, S như hình vẽ. Hãy vẽ đồ thị tín hiệu ra Q.



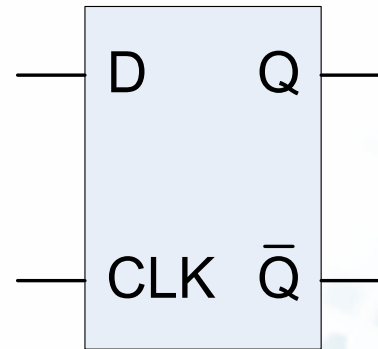


## b. Trigger D

- Trigger D có 1 đầu vào là D và hoạt động ở 2 chế độ đồng bộ và không đồng bộ.
- Ta chỉ xét trigger D hoạt động ở chế độ đồng bộ.



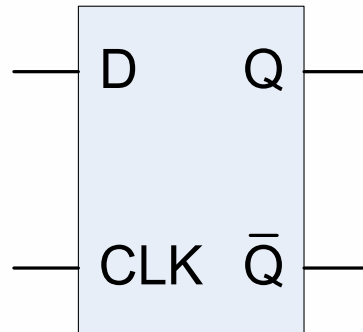
Không đồng bộ



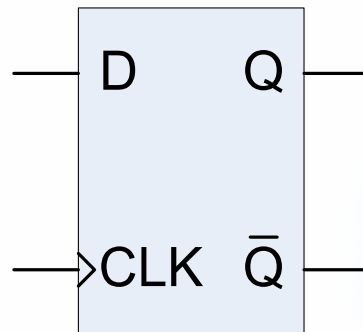
Đồng bộ

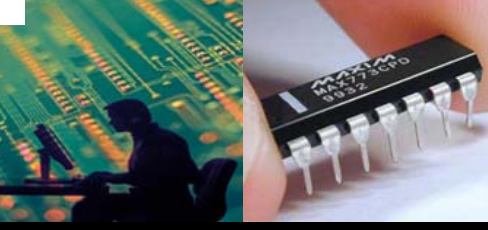
# Trigger D đồng bộ

- Trigger D đồng bộ theo mức gọi là chốt D (Latch)



- Trigger D đồng bộ theo sườn được gọi là xúc phát sườn (Edge triggered)





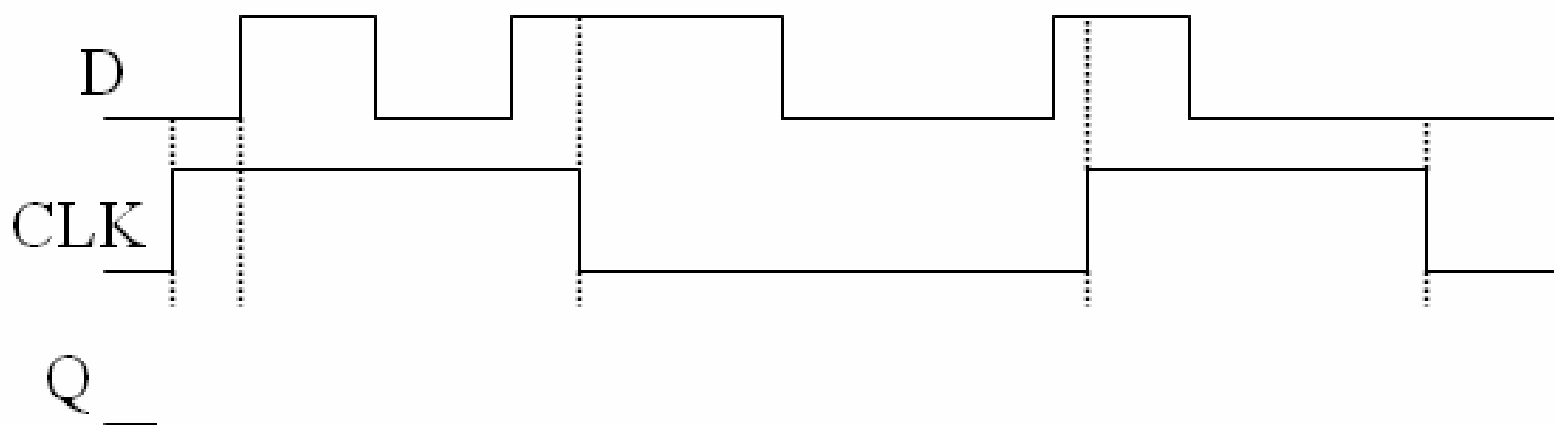
# Bảng chuyển trạng thái của D

| q \ D | 0 | 1 |
|-------|---|---|
| 0     | 0 | 1 |
| 1     | 0 | 1 |

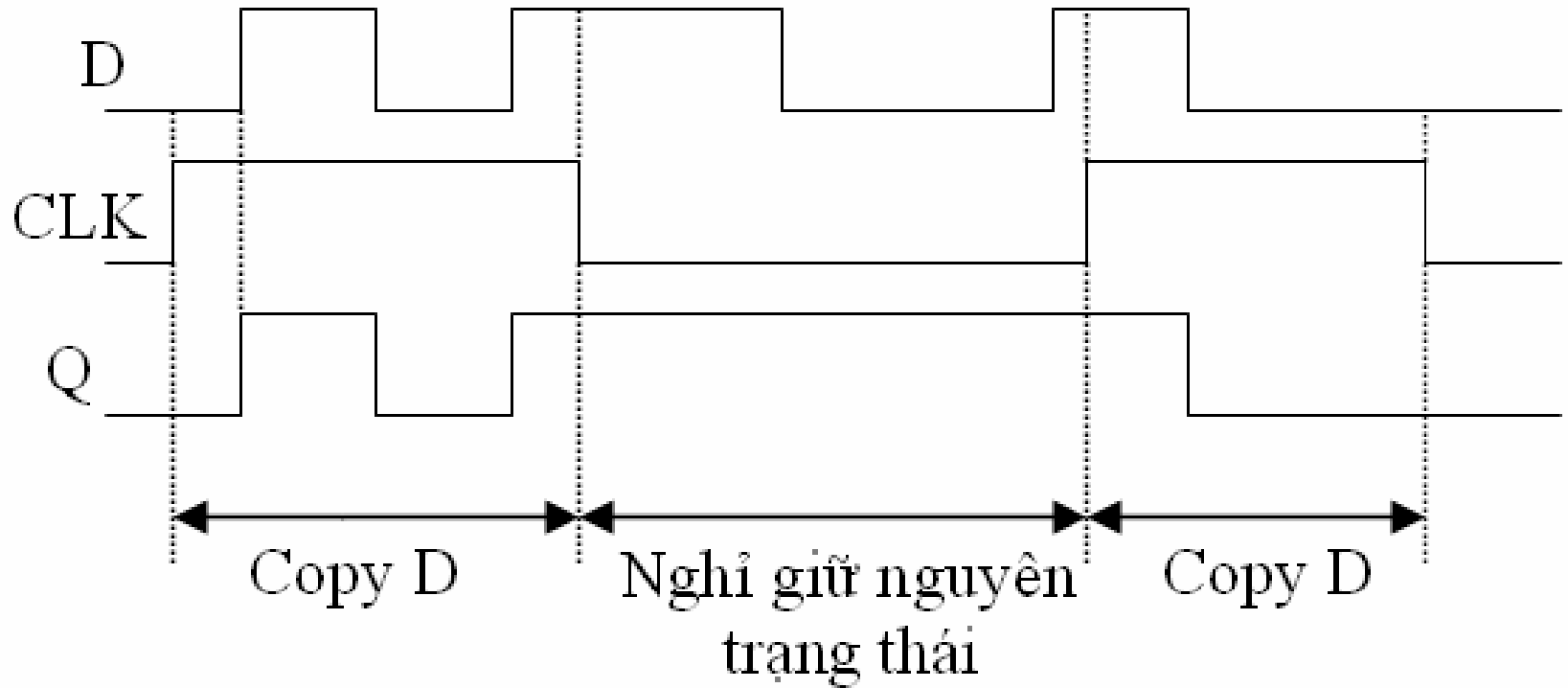
$$Q = D$$

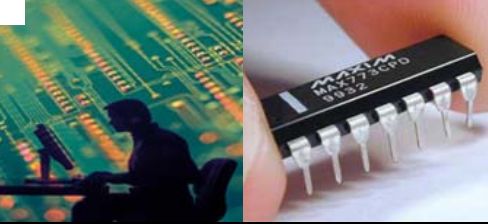


- Cho chốt D kích hoạt mức cao. Hãy vẽ tín hiệu ra Q đúng trên cùng trục thời gian với tín hiệu vào D.

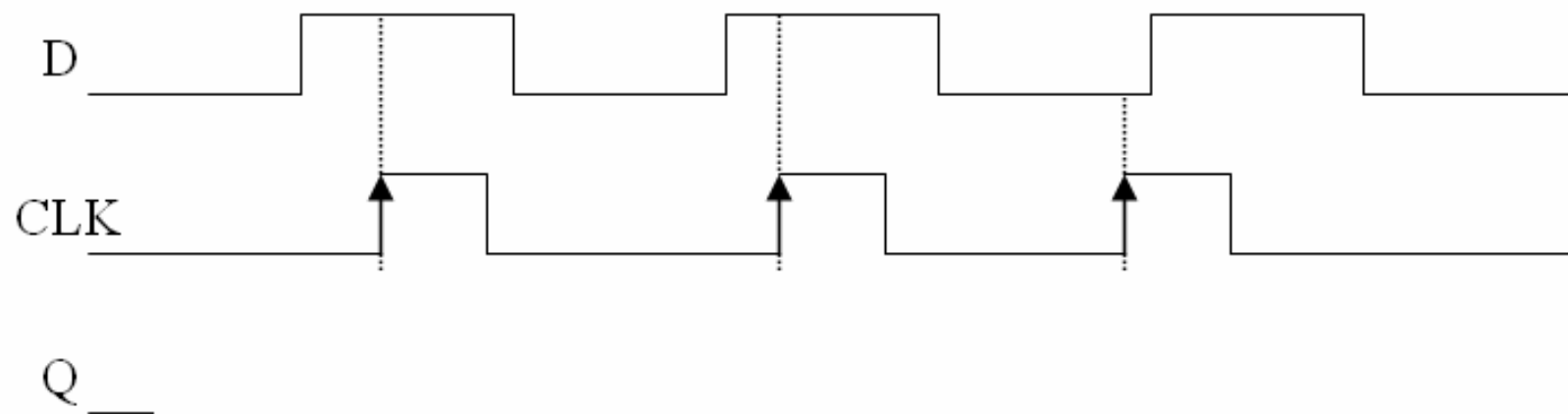


# Ví dụ 1 (tiếp)

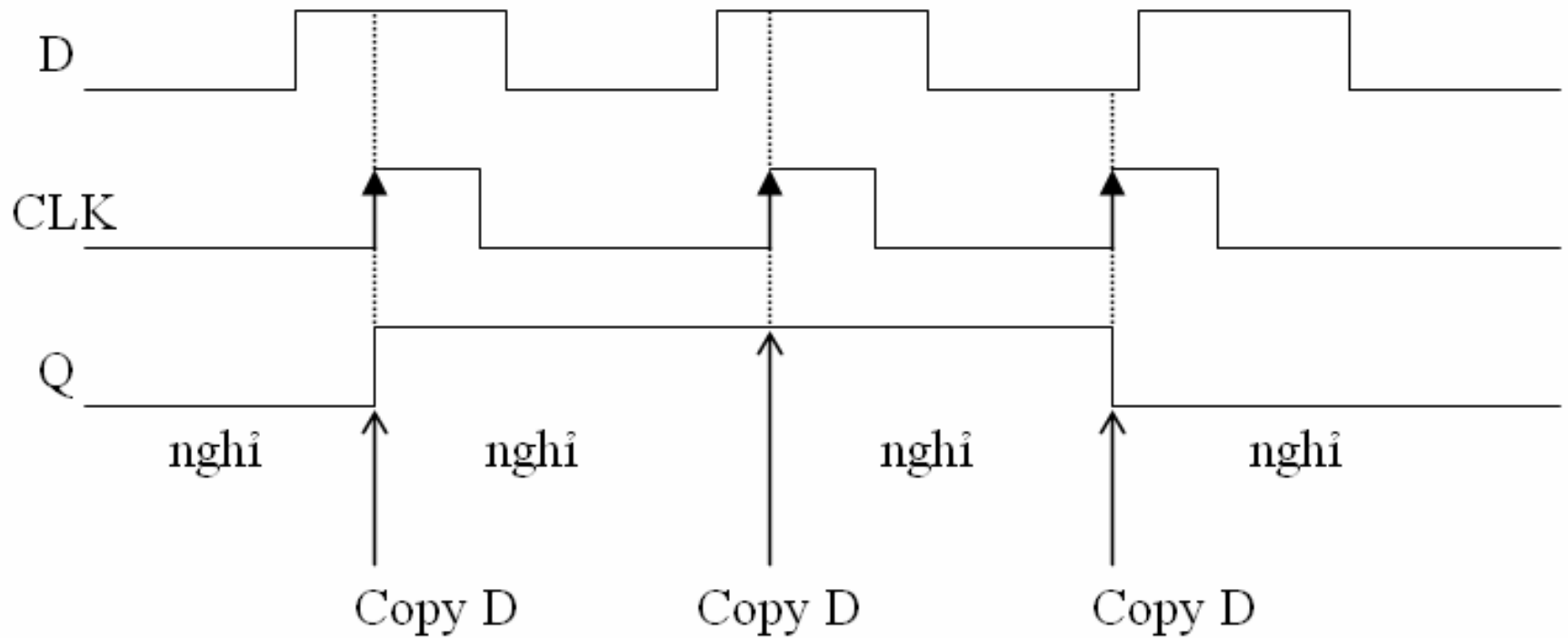




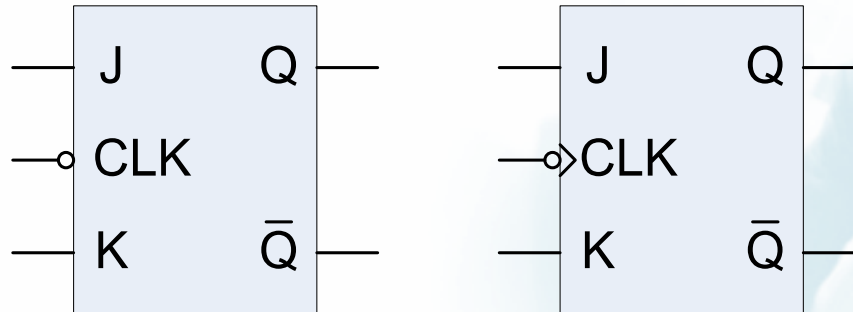
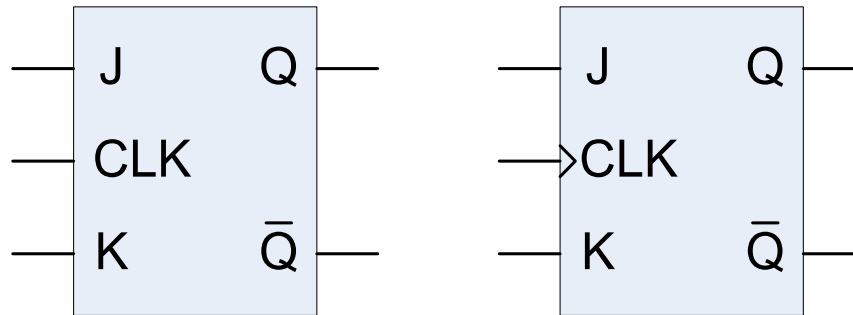
- Cho trigger D xúc phát sườn dương. Hãy vẽ tín hiệu ra Q đúng trên cùng trục thời gian với tín hiệu vào D.

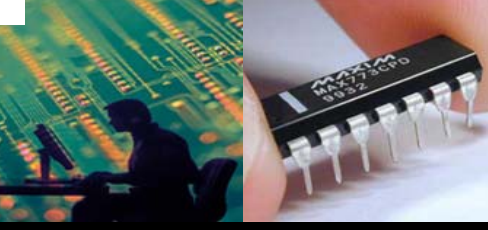


# Ví dụ 2 (tiếp)



- Trigger JK chỉ hoạt động ở chế độ đồng bộ
- Sơ đồ khối:





# Bảng chuyển trạng thái của JK

J ~ S  
K ~ R

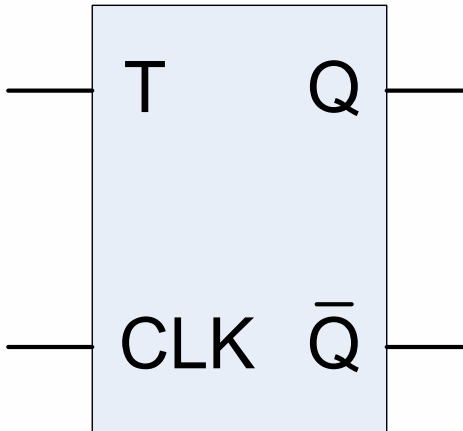
| JK \ q | 00 | 01 | 11 | 10 |
|--------|----|----|----|----|
| 0      | 0  | 0  | 1  | 1  |
| 1      | 1  | 0  | 0  | 1  |

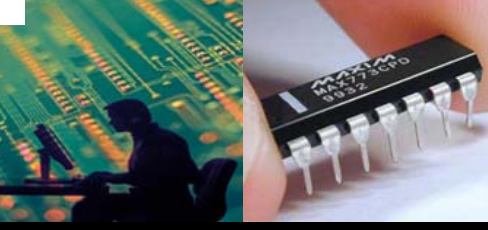
nhớ      xóa      lật      thiết lập

$$Q = \bar{q}J + q\bar{K}$$

## d. Trigger T

- Trigger T chỉ hoạt động ở chế độ đồng bộ
- Sơ đồ khối:





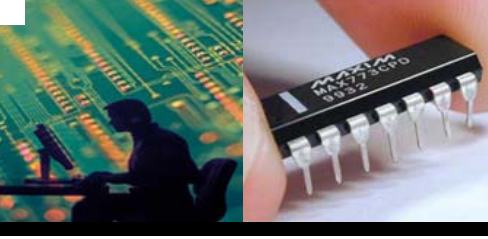
# Bảng chuyển trạng thái của T

| q \ T | 0 | 1 |
|-------|---|---|
| 0     | 0 | 1 |
| 1     | 1 | 0 |

nhớ      lật

$$Q = \bar{q}T + q\bar{T} = q \oplus T$$





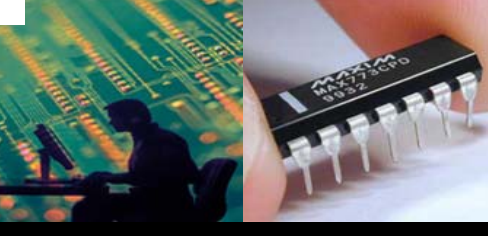
5.1. Khái niệm

5.2. Mô hình của hệ dây

5.3. Các Trigger

5.4. Một số ứng dụng của hệ dây



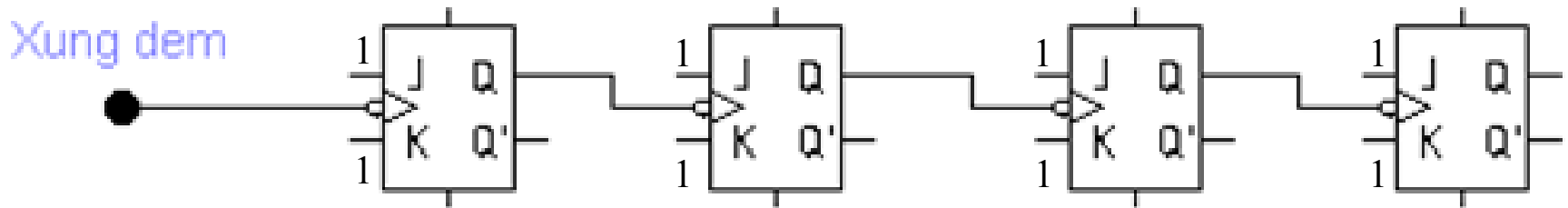


# 1. Bộ đếm và chia tần số

- Bộ đếm được dùng để đếm xung
- Bộ đếm được gọi là module  $n$  nếu nó có thể đếm được  $n$  xung: từ 0 đến  $n-1$
- Có 2 loại bộ đếm:
  - Bộ đếm không đồng bộ: không đồng thời đưa tín hiệu đếm vào các đầu vào của các trigger
  - Bộ đếm đồng bộ: có xung đếm đồng thời là xung đồng hồ clock đưa vào tất cả các trigger của bộ đếm

# Bộ đếm không đồng bộ module 16

- Đếm từ 0 đến 15 và có 16 trạng thái
- Mã hóa thành 4 bit A,B,C,D tương ứng với  $Q_4, Q_3, Q_2, Q_1$
- Cần dùng 4 trigger (giả sử dùng trigger JK)



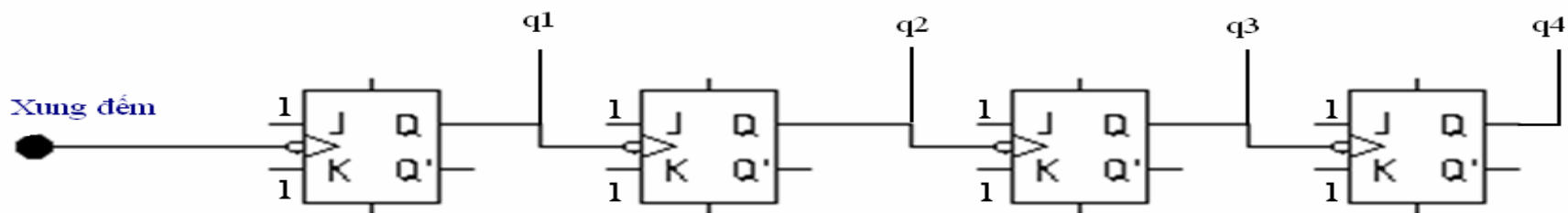
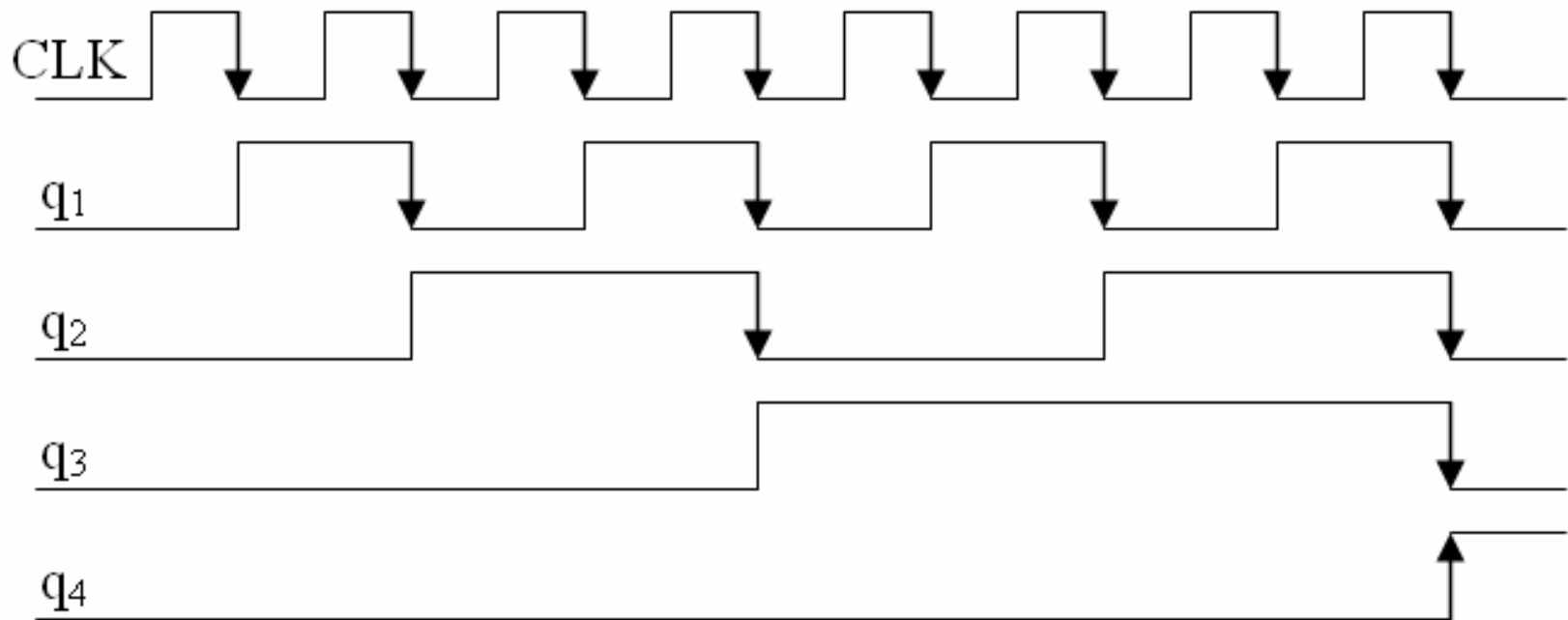
# Bộ đếm không đồng bộ module 16

- Bảng đếm xung:

| Xung | q <sub>4</sub> | q <sub>3</sub> | q <sub>2</sub> | q <sub>1</sub> |
|------|----------------|----------------|----------------|----------------|
| 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              |
| 16   | 0              | 0              | 0              | 0              |

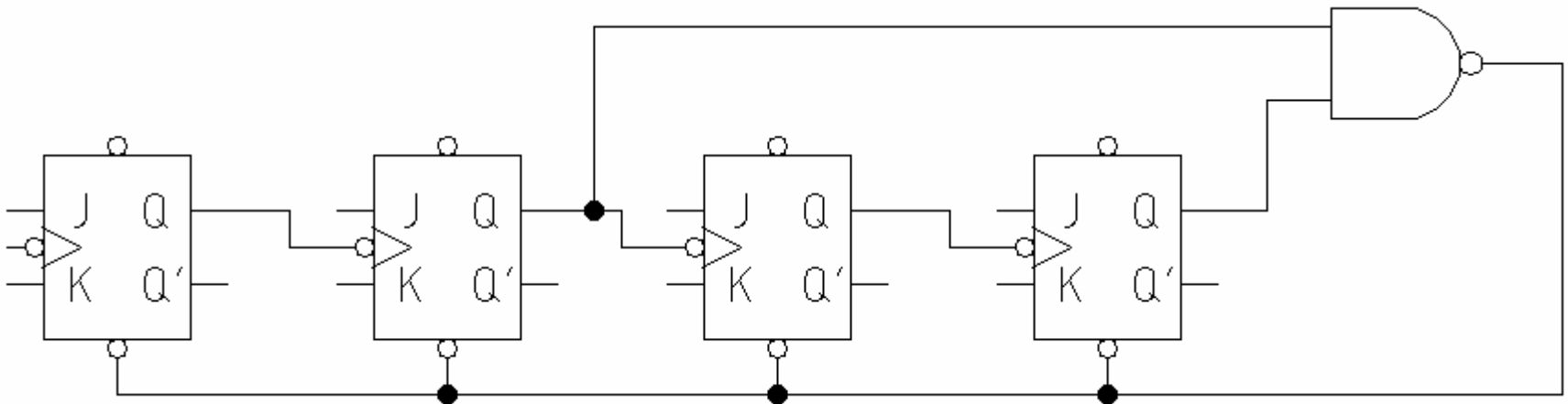
# Bộ đếm không đồng bộ module 16

- Biểu đồ thời gian:



# Bộ đếm không đồng bộ module 10

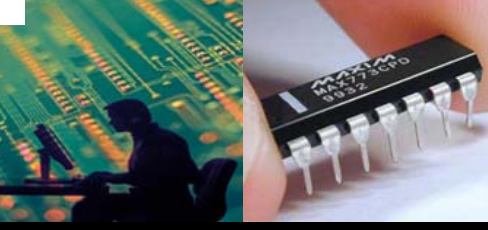
- Có 10 trạng thái  $\Rightarrow$  cần dùng 4 Trigger
- Giả sử dùng Trigger JK có đầu vào CLR (CLEAR: xóa) tích cực ở mức thấp
  - Nếu CLR = 0 thì q = 0
- Cứ mỗi khi đếm đến xung thứ 10 thì tất cả các q bị xóa về 0
- Sơ đồ: (các J=K=1)



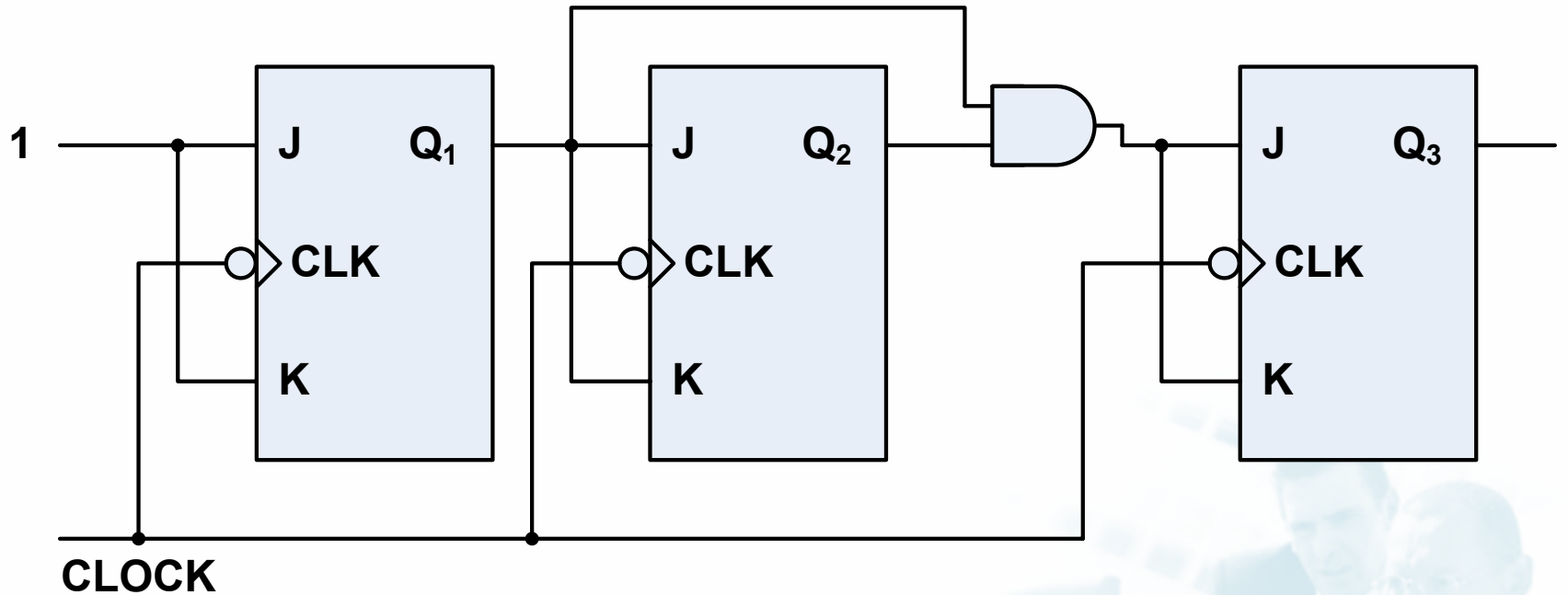
# Bộ đếm đồng bộ module 8

- Có 8 trạng thái  $\Rightarrow$  cần dùng 3 Trigger
- Giả sử dùng các Trigger JK
- Bảng đếm xung:

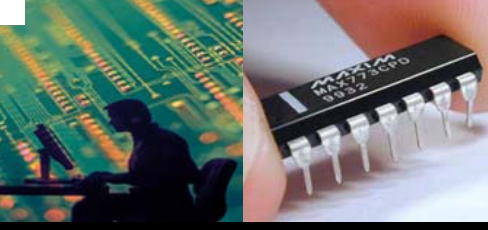
| xung | q <sub>3</sub> | q <sub>2</sub> | q <sub>1</sub> |
|------|----------------|----------------|----------------|
| 0    | 0              | 0              | 0              |
| 1    | 0              | 0              | 1              |
| 2    | 0              | 1              | 0              |
| 3    | 0              | 1              | 1              |
| 4    | 1              | 0              | 0              |
| 5    | 1              | 0              | 1              |
| 6    | 1              | 1              | 0              |
| 7    | 1              | 1              | 1              |
| 8    | 0              | 0              | 0              |



# Bộ đếm đồng bộ module 8 (tiếp)



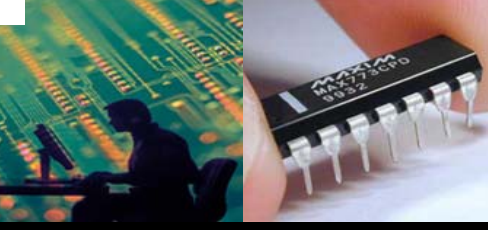




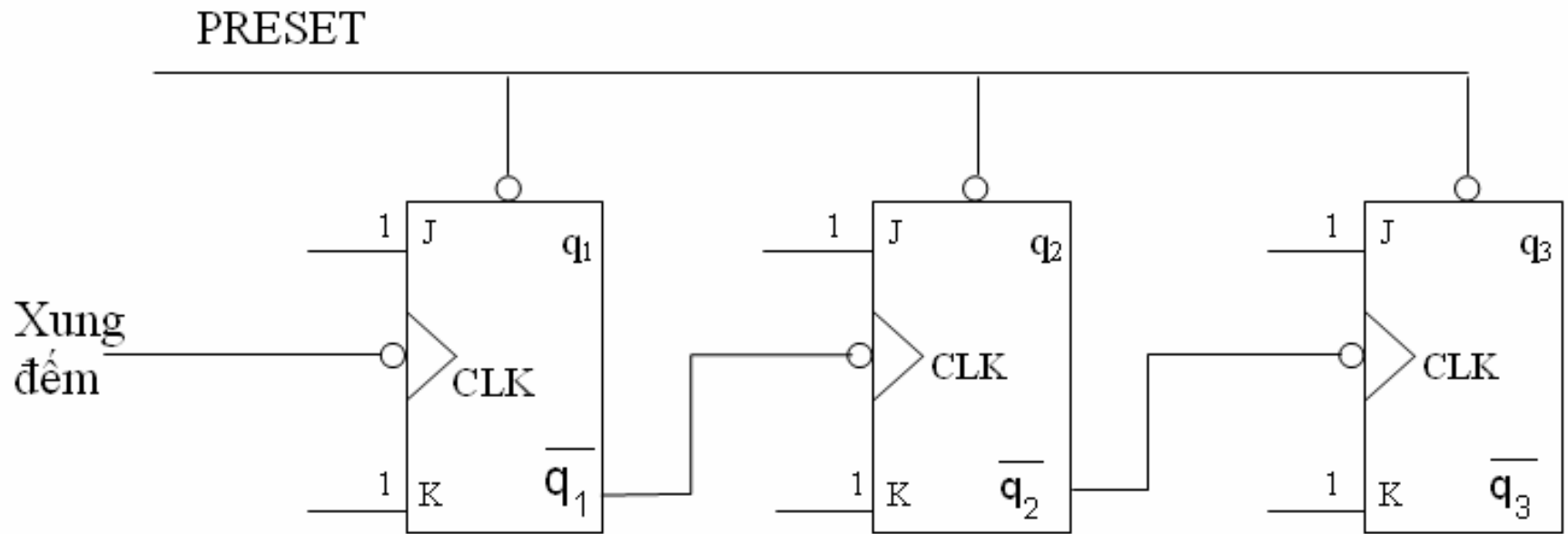
# Bộ đếm lùi không đồng bộ module 8

- Giả sử dùng Trigger JK có đầu vào PR (PRESET: thiết lập trước) tích cực ở mức thấp
  - Nếu  $PR = 0$  thì  $q = 1$
- Đầu tiên cho  $PR = 0$  thì  $q_1q_2q_3 = 111$
- Sau đó cho  $PR = 1$ , hệ hoạt động bình thường

| xung | $q_3$ | $q_2$ | $q_1$ | Số đếm |
|------|-------|-------|-------|--------|
| 0    | 1     | 1     | 1     | 7      |
| 1    | 1     | 1     | 0     | 6      |
| 2    | 1     | 0     | 1     | 5      |
| 3    | 1     | 0     | 0     | 4      |
| 4    | 0     | 1     | 1     | 3      |
| 5    | 0     | 1     | 0     | 2      |
| 6    | 0     | 0     | 1     | 1      |
| 7    | 0     | 0     | 0     | 0      |
| 8    | 1     | 1     | 1     | 7      |



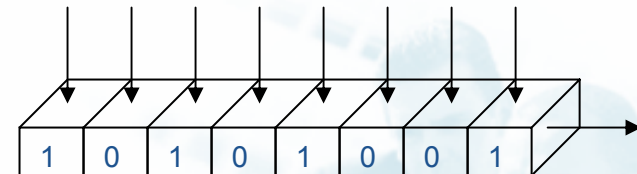
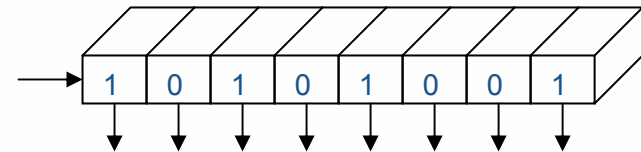
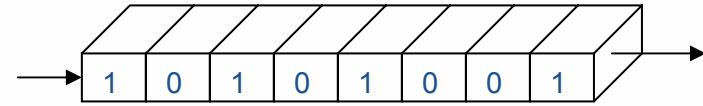
# Bộ đếm lùi không đồng bộ module 8



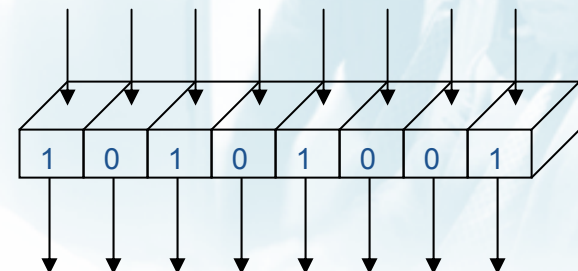
## 2. Thanh ghi

- Thanh ghi có cấu tạo gồm các trigger nối với nhau
- Chức năng:
  - Để lưu trữ tạm thời thông tin
  - Dịch chuyển thông tin
- Lưu ý: cả thanh ghi và bộ nhớ đều dùng để lưu trữ thông tin, nhưng thanh ghi có chức năng dịch chuyển thông tin. Do đó, thanh ghi có thể sử dụng làm bộ nhớ, nhưng bộ nhớ không thể làm được thanh ghi.

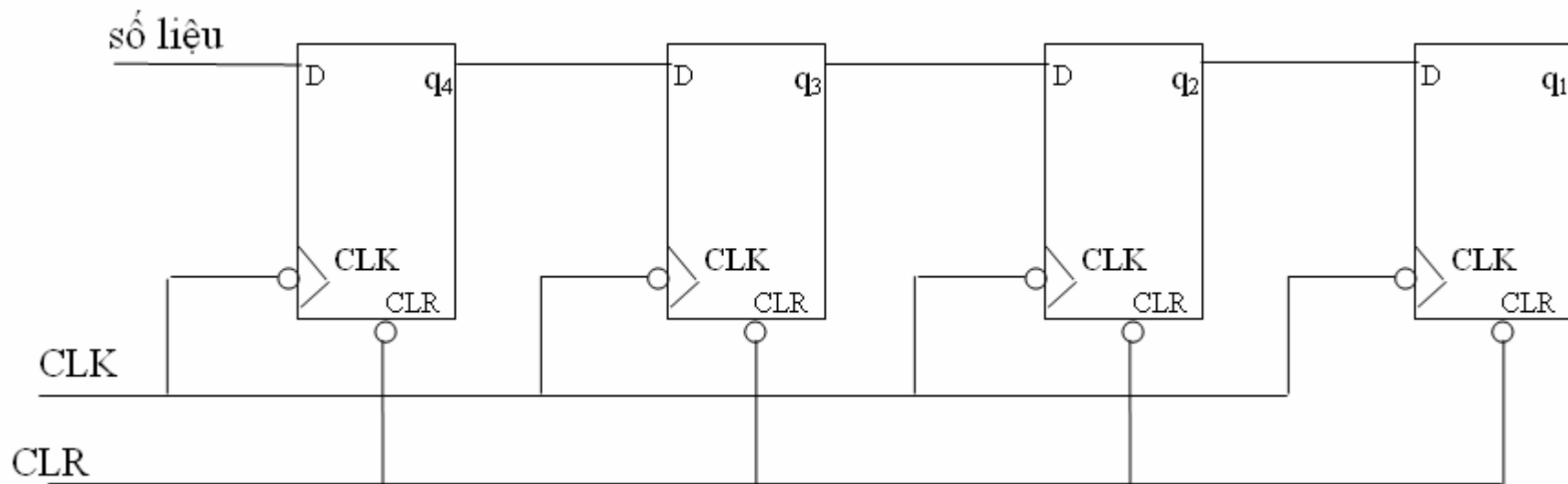
- Vào nối tiếp ra nối tiếp
- Vào nối tiếp ra song song
- Vào song song ra nối tiếp



- Vào song song ra song song

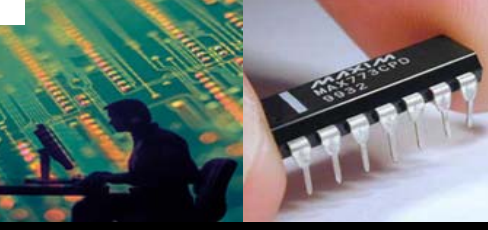


- Thanh ghi 4 bit vào nối tiếp ra song song dùng Trigger D



- Bảng số liệu khảo sát:

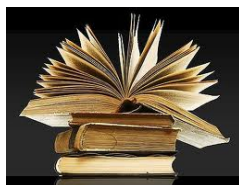
| Dòng | Vào |         |     | Ra |   |   |   |
|------|-----|---------|-----|----|---|---|---|
|      | CLR | số liệu | CLK | A  | B | C | D |
| 1    | 0   | 0       | 0   | 0  | 0 | 0 | 0 |
| 2    | 1   | 1       | 0   | 0  | 0 | 0 | 0 |
| 3    | 1   | 1       | 1   | 1  | 0 | 0 | 0 |
| 4    | 1   | 1       | 2   | 1  | 1 | 0 | 0 |
| 5    | 1   | 1       | 3   | 1  | 1 | 1 | 0 |
| 6    | 1   | 0       | 4   | 0  | 1 | 1 | 1 |
| 7    | 1   | 0       | 5   | 0  | 0 | 1 | 1 |
| 8    | 1   | 0       | 6   | 0  | 0 | 0 | 1 |
| 9    | 1   | 0       | 7   | 0  | 0 | 0 | 0 |
| 10   | 1   | 0       | 8   | 0  | 0 | 0 | 0 |
| 11   | 1   | 1       | 9   | 1  | 0 | 0 | 0 |
| 12   | 1   | 0       | 10  | 0  | 1 | 0 | 0 |
| 13   | 1   | 0       | 11  | 0  | 0 | 1 | 0 |
| 14   | 1   | 0       | 12  | 0  | 0 | 0 | 1 |
| 15   | 1   | 0       | 13  | 0  | 0 | 0 | 0 |



- **KS28:** <http://k28cntt.wordpress.com>
  - User: k28cntt
  - Pass: “tap the”
  
- **SPKT Tin K50**
  - Lớp phó: Trần Thị Dung 0976324219.



TRƯỜNG ĐẠI HỌC ĐÀ LẠT  
KHOA CÔNG NGHỆ THÔNG TIN



GIÁO TRÌNH

# ĐIỆN TỬ CĂN BẢN



THÁNG 1/2005



**TRƯỜNG ĐẠI HỌC ĐÀ LẠT**  
**KHOA CÔNG NGHỆ THÔNG TIN**

**Giáo trình**

**ĐIỆN TỬ**  
**CĂN BẢN**

**Tháng 1 - 2005**

# LỜI NÓI ĐẦU

---

Giáo trình **ĐIỆN TỬ CĂN BẢN** là tài liệu học tập dành cho sinh viên Khoa Công nghệ Thông tin.

**Điện tử căn bản** trình bày cấu tạo và hoạt động của các linh kiện điện tử và mạch của chúng. Đây là những kiến thức cơ sở để hiểu biết cấu trúc máy tính và các thiết bị phần cứng của kỹ thuật công nghệ thông tin. Nội dung chủ yếu của giáo trình là mô tả cấu tạo, đặc trưng của các linh kiện điện tử bán dẫn như diode, transistor, IC và các mạch ứng dụng căn bản của chúng.

Giáo trình gồm 11 chương

Chương 1: Một số khái niệm

Chương 2: Diode bán dẫn và mạch diode

Chương 3: Transistor

Chương 4: Phân cực transistor

Chương 5: Khuếch đại transistor

Chương 6: Khuếch đại công suất

Chương 7: Các hiệu ứng tần số của mạch khuếch đại

Chương 8: Các linh kiện bán dẫn đặc biệt

Chương 9: Khuếch đại thuật toán

Chương 10: Các mạch dao động

Chương 11: Nguồn nuôi

Nội dung của giáo trình rất rộng mà thời gian lại hạn chế trong 60 tiết do đó một số vấn đề bị bỏ qua. Sinh viên có thể tham khảo thêm textbook bằng tiếng Anh sau đây tại thư viện Khoa Công nghệ Thông tin.

**Electronic Principles** Malvino, Mc Graw-Hill, 1999

Sinh viên cũng có thể vào Website: [www.alldatasheet.com](http://www.alldatasheet.com) để có thêm các thông tin chi tiết về số liệu kỹ thuật của các linh kiện.

Do trình độ người viết có hạn, chắc chắn giáo trình còn có nhiều thiếu sót. Rất mong được sự góp ý của bạn đọc.

Đà Lạt, tháng 1 năm 2005

Phan Văn Nghĩa

# Chương I **MỘT SỐ KHÁI NIỆM**

## **I.1 SỰ GẦN ĐÚNG**

Trong cuộc sống chúng ta thường xuyên dùng sự gần đúng hay xấp xỉ. Trong kỹ thuật cũng vậy. Chúng ta thường dùng các mức gần đúng sau:

- ◆ Gần đúng lý tưởng (đôi khi gọi là gần đúng bậc 1)
- ◆ Gần đúng bậc 2
- ◆ Gần đúng bậc 3
- ◆ Mô tả chính xác

1) **Gần đúng lý tưởng.** Một đoạn dây AWG22 dài 1 inch (2.54cm) có điện trở thuần  $R=0.016\Omega$ , cuộn cảm  $L=0.24\mu\text{H}$  và tụ  $C=3.3\text{pF}$ . Nếu chúng ta tính tới tất cả các ảnh hưởng của RLC thì tính toán liên quan đến dòng và thế sẽ mất nhiều thời gian và có thể phức tạp. Vì vậy trong nhiều trường hợp, để đơn giản, có thể bỏ qua RLC của đoạn dây dẫn.

Sự gần đúng lý tưởng, là mạch tương đương đơn giản nhất của thiết bị. Ví dụ, gần đúng lý tưởng của một đoạn dây nối là một vật dẫn có trở kháng  $Z=0$ . Sự gần đúng này là đủ cho các thiết bị điện tử thông thường. Trường hợp ngoại lệ sẽ xảy ra tại tần số cao. Khi đó phải xét đến cảm kháng và dung kháng. Giả sử rằng 1 inch dây nối có  $L=0.24\mu\text{H}$  và  $C=3.3\text{pF}$  thì tại tần số  $f=10\text{MHz}$  cảm kháng và dung kháng tương đương của chúng là  $15.1\Omega$  và  $4.82\text{K}\Omega$ . Chúng ta thường dùng gần đúng lý tưởng đối với dây nối khi tần số  $f < 1\text{MHz}$ . Tuy nhiên không có nghĩa là chúng ta không cần để ý đến chiều dài của dây nối. Trên thực tế, cần làm cho dây nối ngắn đến mức có thể.

Trong khi tìm hỏng cho mạch hay thiết bị, một gần đúng lý tưởng là đủ dùng. Trong giáo trình này chúng ta dùng gần đúng lý tưởng cho các thiết bị bán dẫn bằng cách giản lược chúng như các mạch tương đương đơn giản. Bằng cách dùng gần đúng lý tưởng, chúng ta dễ dàng phân tích và hiểu hoạt động của các mạch bán dẫn.

2) **Gần đúng bậc 2.** Gần đúng bậc 2 thêm một hoặc nhiều thành phần vào gần đúng lý tưởng. Nếu gần đúng lý tưởng của 1 viên pin là 1.5V thì gần đúng bậc 2 của 1 viên pin là một nguồn thế 1.5V nối tiếp với 1 điện trở  $10\Omega$ . Điện trở này gọi là điện trở trong hay điện trở nguồn của viên pin. Nếu điện trở tải bé hơn  $100\Omega$ , thế trên tải có thể bé hơn 1.5V do sụt thế qua điện trở nguồn. Lúc này các tính toán cần phải kèm theo cả điện trở nguồn của pin.

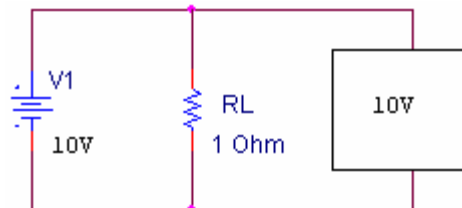
3) **Gần đúng bậc 3 và các gần đúng cao hơn.** Gần đúng bậc 3 kèm theo một số phần tử nữa vào mạch tương đương của thiết bị. Thậm chí các gần đúng cao hơn nữa cần phải làm khi phân tích mạch. Tính toán bằng tay đối với các mạch tương đương gần đúng cao hơn bậc 2 trở nên rất khó khăn. Trong trường hợp này chúng ta sẽ dùng chương trình máy tính. Ví dụ EWB (Electronics Work Bench) hoặc Pspice là các phần mềm máy tính trong đó dùng các gần đúng bậc cao để phân tích mạch.

Tóm lại, việc sử dụng gần đúng loại nào là phụ thuộc vào yêu cầu công việc mà chúng ta phải làm. Nếu chúng ta đang tìm lỗi hay sửa chữa thiết bị, gần đúng bậc 1 là đủ. Trong nhiều trường hợp gần đúng bậc 2 là lựa chọn tốt vì dễ dùng và không yêu cầu máy tính. Đối với các gần đúng cao hơn cần phải dùng máy tính và một chương trình.

## I.2 NGUỒN THẾ

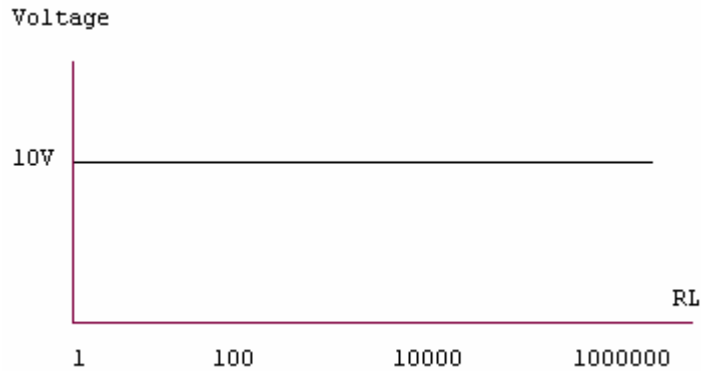
Một nguồn thế lý tưởng tạo ra một hiệu điện thế là hằng số trên tải. Ví dụ đơn giản nhất của một nguồn thế lý tưởng là một acqui hoàn hảo, một acqui mà điện trở trong của nó bằng 0.

Hình 1-1a là hình vẽ một mạch, trong đó nguồn thế  $V_1=10V$  nối với điện trở tải  $R_L=1\Omega$ . Vôn kế chỉ 10V, đúng bằng giá trị của nguồn thế.



Hình 1-1a: Nguồn thế và tải

Hình 1-1b cho thấy giản đồ của hiệu điện thế trên tải và điện trở tải. Theo giản đồ, hiệu điện thế trên tải vẫn 10V khi điện trở tải thay đổi từ  $1\Omega$  đến  $1M\Omega$ . Nói một cách khác, một nguồn thế lý tưởng tạo ra một thế trên tải là hằng số bất chấp điện trở tải là lớn hay bé. Với một nguồn thế lý tưởng, chỉ có dòng tải thay đổi khi điện trở tải thay đổi.

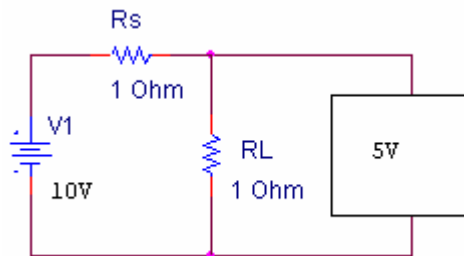


Hình 1-1b: Quan hệ giữa thế tải và trở tải

### **Gần đúng bậc 2 của nguồn thế.**

Nguồn thế lý tưởng là thiết bị chỉ có về mặt lý thuyết, nó không tồn tại trong thực tế. Vì khi điện trở tải gần bằng 0, dòng tải sẽ gần bằng vô cùng. Không có một nguồn thế thực nào có thể tạo ra một dòng tải vô hạn vì nguồn thế thực luôn luôn có điện trở trong (điện trở nguồn). Gần đúng bậc 2 của một nguồn thế phải kèm theo điện trở trong này.

Hình 1-2a mô tả ý tưởng này. Điện trở trong  $1\Omega$  nối tiếp với bộ acqui lý tưởng. Khi đó giá trị chỉ trên Vôn kế là 5V thay vì 10V.

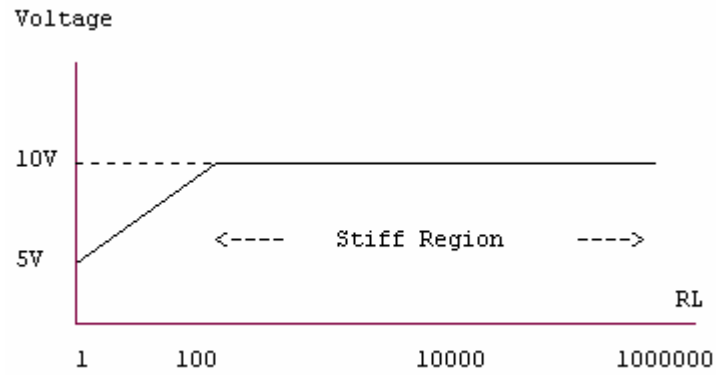


Hình 1-2a: Nguồn thế với điện trở trong

Hình 1-2b là giản đồ của thế trên tải và điện trở tải của một nguồn thế thực. Thế trên tải chỉ đạt được giá trị 10V khi điện trở tải lớn hơn điện trở nguồn nhiều lần, lớn hơn đến mức có thể bỏ qua điện trở nguồn.

### **Nguồn thế mạnh (Stiff Voltage Source)**

Chúng ta có thể bỏ qua điện trở nguồn khi nó nhỏ hơn điện trở tải ít nhất là 100 lần. Tất cả các nguồn thế thỏa mãn điều kiện này gọi là nguồn thế mạnh.



**Hình 1-2b:** Thế trên tải và trở tải đối với nguồn thế thực

Một nguồn thế mạnh nếu thỏa điều kiện:

$$R_S < 0.01R_L \quad (1-1)$$

Điện trở tải bé nhất mà nguồn thế vẫn mạnh là:

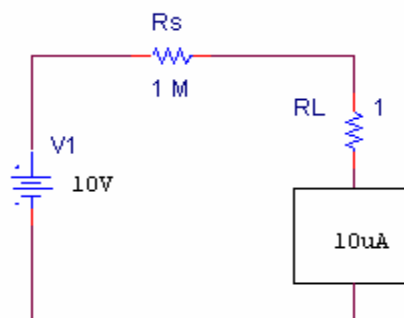
$$R_{L(\min)} = 100R_S \quad (1-2)$$

Theo (1-2) điện trở tải bé nhất phải bằng 100 lần điện trở nguồn. Trong trường hợp này, sai số tính toán do bỏ qua điện trở nguồn là 1%. Giá trị sai số này là đủ nhỏ để bỏ qua trong gần đúng bậc 2.

Lưu ý:

- Định nghĩa về nguồn thế mạnh áp dụng cho cả nguồn DC lẫn nguồn AC.
- Gần đúng bậc 2 chỉ có ý nghĩa tại tần số thấp. Tại tần số cao, các hệ số cần phải xem xét thêm là cảm kháng và dung kháng.

### I.3 NGUỒN DÒNG



**Hình 1-3:** Nguồn dòng

Một nguồn thế DC cung cấp một thế trên tải không đổi đối với các điện trở tải khác nhau. Nguồn dòng DC tạo ra một dòng tải là hằng số đối với các điện trở khác nhau. Ví dụ một nguồn dòng lý tưởng là một acqui có điện trở trong rất lớn như hình 1-3.

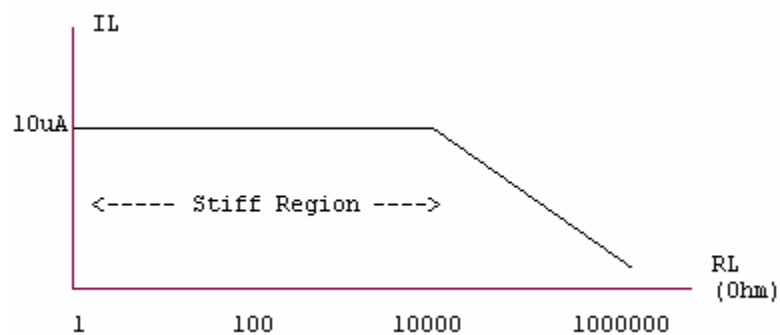
Trong mạch hình 1-3, dòng tải tính bởi:

$$I_L = V / (R_s + R_L)$$

với  $R_L = 1\Omega$ ,  $R_s = 1M\Omega$ , dòng tải bằng:

$$I_L = 10V / (1M + 1) = 10\mu A$$

Trong tính toán trên đây, điện trở tải ảnh hưởng không đáng kể lên dòng tải.



**Hình 1-4:** ảnh hưởng của điện trở tải đối với dòng tải

Hình 1-4 chỉ ra ảnh hưởng của điện trở tải đối với dòng tải. Dòng tải vẫn là  $10\mu A$  trong một vùng rộng của điện trở tải. Khi điện trở tải lớn hơn  $10K\Omega$  ( $R_L > 1\% R_s$ ) thì dòng tải bắt đầu thay đổi.

#### **Nguồn dòng mạnh.**

Chúng ta có thể bỏ qua ảnh hưởng của điện trở nguồn của một nguồn dòng nếu nó lớn hơn điện trở tải ít nhất là 100 lần. Mọi nguồn dòng thỏa điều kiện này gọi là nguồn dòng mạnh.

Nguồn dòng mạnh nếu thỏa điều kiện:

$$R_s > 100R_L \quad (1-3)$$

Trong trường hợp giới hạn, điện trở tải lớn nhất mà nguồn vẫn được xem là nguồn dòng mạnh khi

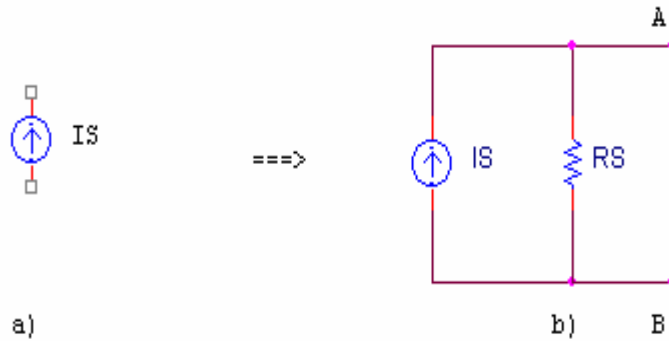
$$R_L(\max) = 0.01R_s \quad (1-4)$$

Theo (1-4) điện trở tải lớn nhất bằng 1/100 điện trở nguồn.

Hình 1-5a ký hiệu một nguồn dòng lý tưởng, trong đó thiết bị tạo ra một dòng hằng  $I_s$  với điện trở nội của nguồn  $R_s$  là vô cùng.

Hình 1-5b chỉ ra gần đúng bậc 2 của nguồn dòng. Ở đó điện trở trong  $R_s$  mắc song song với nguồn dòng lý tưởng  $I_s$ . Phần cuối của chương này sẽ

xem xét định lý Norton, khi đó chúng ta sẽ biết tại sao  $R_s$  lại mắc song song với nguồn dòng  $I_s$ .

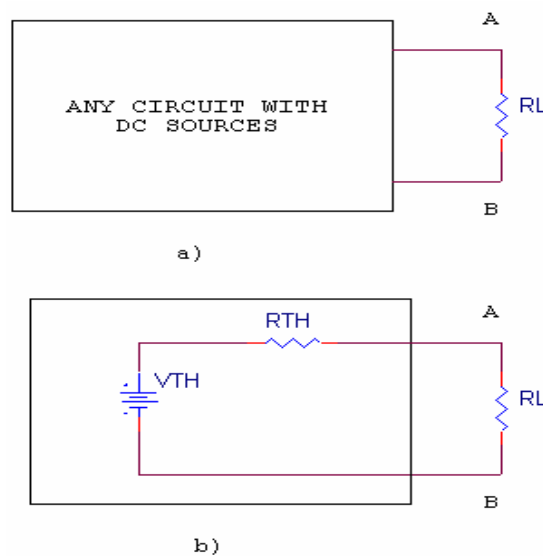


Hình 1-5: Nguồn dòng

Bảng sau cho thấy sự khác nhau giữa nguồn dòng và nguồn thế.

| Đại lượng | Nguồn thế       | Nguồn dòng      |
|-----------|-----------------|-----------------|
| $R_s$     | Rất bé          | Rất lớn         |
| $R_L$     | $> 100 R_s$     | $< 0.01 R_s$    |
| $V_L$     | Hằng            | Phụ thuộc $R_L$ |
| $I_L$     | Phụ thuộc $R_L$ | Hằng            |

## I.4 ĐỊNH LÝ THEVENIN



Hình 1-6: Thế Thevenin



Định lý là một mệnh đề có thể chứng minh bằng toán học. Sau đây chúng ta xem xét một số khái niệm liên quan đến định lý Thevenin, tên một kỹ sư người Pháp.

**Thế Thevenin ( $V_{TH}$ ):** Trên hình 1-6, thế Thevenin là thế đo được giữa 2 đầu điện trở tải (hai đầu AB) khi không có điện trở tải (điện trở tải hở mạch). Vì vậy đôi khi thế Thevenin còn gọi là thế hở mạch.

Thế Thevenin:

$$V_{TH} = V_{OC} \quad (1-5)$$

**Trở Thevenin ( $R_{TH}$ ):** là điện trở đo được giữa 2 đầu điện trở tải khi điện trở tải hở mạch và khi tất cả các nguồn giảm tới 0.

Giảm nguồn tới 0 có ý nghĩa khác nhau đối với nguồn dòng và nguồn thế. Cụ thể như sau:

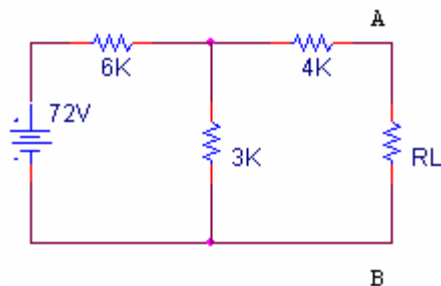
- ◆ Đối với nguồn thế: ngắn mạch
- ◆ Đối với nguồn dòng: hở mạch

Vậy định lý Thevenin đề cập đến cái gì? Theo định lý Thevenin, mọi hộp đen chứa mạch gồm nguồn DC và các điện trở tuyến tính (là điện trở không thay đổi giá trị khi thay đổi thế trên nó) như hình 1-6a có thể thay thế bằng một nguồn thế Thevenin và một điện trở Thevenin tương đương như hình 1-6b. Khi đó dòng qua tải bằng

$$I_L = V_{TH} / (R_{TH} + R_L) \quad (1-6)$$

Định lý Thevenin là một công cụ mạnh. Nó không chỉ giúp đơn giản các tính toán mà còn giúp giải thích hoạt động của các mạch mà nếu chỉ dùng các phương trình Kirchoff thì không thể làm được.

Ví dụ: Tính thế và trở Thevenin cho mạch hình 1-7.



Hình 1-7

Để tính thế Thevenin chúng ta hở mạch điện trở tải  $R_L$ . Dễ dàng thấy rằng  $V_{TH} = 24V$ .

Để tính trở Thevenin cần hở mạch tải và ngắn mạch nguồn 72V. Khi đó:

$$R_{TH} = 4 + (3//6) = 6K\Omega$$

Có thể dùng Vôn kế và Ohm kế để đo thế Thevenin và trở Thevenin. Độ chính xác của các phép đo phụ thuộc vào loại máy đo được sử dụng. Ví dụ nếu sử dụng máy đo thế loại chỉ thị kim có độ nhạy  $20K\Omega/V$  tại thang đo 30V thì trở kháng vào của máy đo là  $600K\Omega$ . Khi đó thế đo được sẽ bé hơn thế Thevenin một chút. Thường người ta dùng vôn kế có trở kháng vào vào lớn hơn trở Thevenin ít nhất là 100 lần. Khi đó sai số sẽ bé hơn 1%. Để có trở kháng vào cao, ngày nay người ta dùng vôn kế số (Digital Multimeter) với trở kháng vào cỡ  $10M\Omega$ .

## 1.5 ĐỊNH LÝ NORTON

Trên hình 1-8a, dòng Norton  $I_N$  được định nghĩa là dòng tải khi điện trở tải ngắn mạch. Vì vậy dòng Norton còn gọi là dòng ngắn mạch.

$$I_N = I_{SC} \quad (1-7)$$

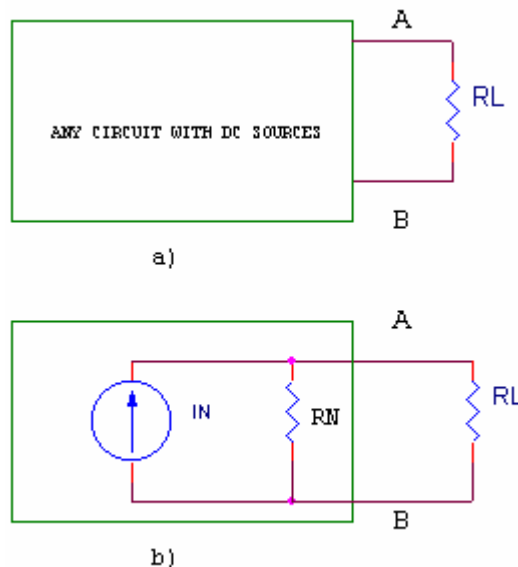
Điện trở Norton là điện trở đo giữa hai đầu điện trở tải khi hở mạch điện trở tải và tất cả các nguồn giảm tới 0.

$$R_N = R_{OC} \quad (1-8)$$

Do điện trở Thevenin cũng bằng  $R_{OC}$ , nên thể viết:

$$R_{TH} = R_N \quad (1-9)$$

nghĩa là điện trở Thevenin và điện trở Norton là bằng nhau.



Hình 1-8: Mạch Norton

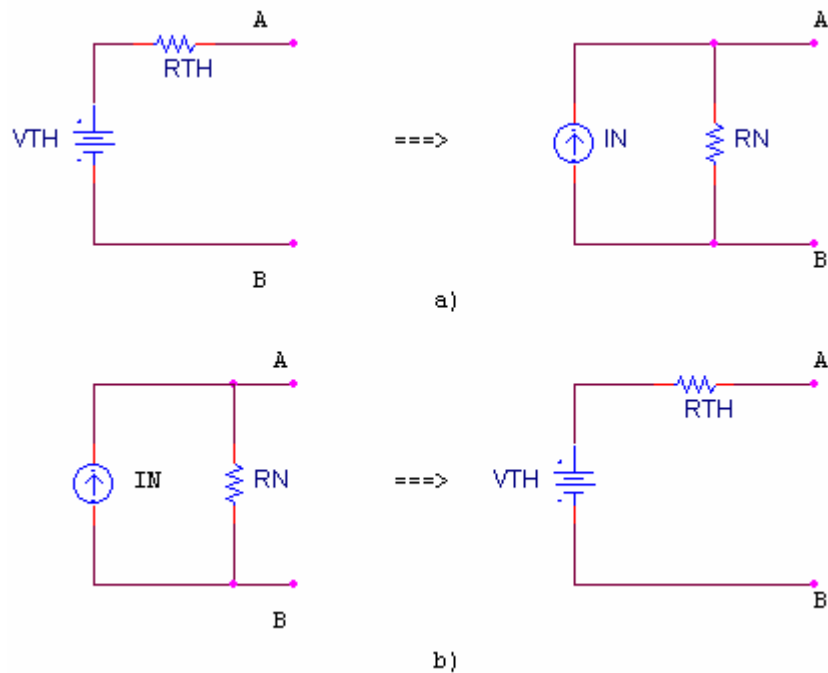
Trong hình 1-8a, hộp đen chứa mạch bất kỳ gồm nguồn DC và các điện trở tuyến tính. Định lý Norton phát biểu rằng, có thể thay thế mạch hình 1-8a bằng mạch hình 1-8b.

Dưới dạng biểu thức:

$$V_L = I_N(R_N // R_L) \quad (1-10)$$

Theo (1-10) thế trên tải bằng dòng Norton nhân với điện trở tải mắc song song với điện trở Norton.

Định lý Norton và Thevenin là tương đương. Trên thực tế, có thể biến đổi nguồn thế Thevenin thành nguồn dòng Norton và ngược lại. Hình 1-9 cho thấy các cách biến đổi.



Hình 1-9: Biến đổi Thevenin - Norton

Có thể thấy rằng trở Norton và trở Thevenin là giống nhau. Quan hệ giữa dòng Norton và thế Thevenin là

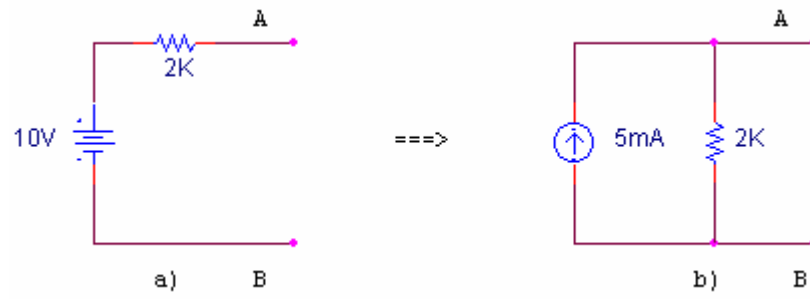
$$I_N = V_{TH} / R_{TH} \quad (1-11)$$

Ví dụ: Giả sử rằng chúng ta đã rút gọn một mạch thành mạch Thevenin như hình 1-10. Hãy biến đổi mạch này thành mạch Norton.

Lời giải: Dùng phương trình (1-11) ta có:

$$I_N = 10V/2K = 5mA$$

Hình 1-10b vẽ mạch Norton tương đương của mạch Thevenin trên hình 1-10a.



Hình 1-10

## Chương II **DIODE BÁN DẪN VÀ MẠCH DIODE**

### **II.1 CÁC LOẠI CHẤT BÁN DẪN**

Theo tính chất dẫn điện, có 3 loại vật chất:

- ◆ Chất dẫn điện
- ◆ Chất không dẫn điện (điện môi)
- ◆ Chất bán dẫn

Trong chất dẫn điện thường chỉ có 1 electron ở vùng hoá trị, trong khi đó các chất điện môi có 8 electron ở vùng hoá trị. Bán dẫn có tính chất trung gian giữa điện môi và chất dẫn điện, chúng có 4 electron ở vùng hoá trị.

Germanium (Ge) và silicon (Si) là các chất bán dẫn điển hình. Ở trạng thái tinh thể tinh khiết (không bị pha tạp), mỗi nguyên tử Ge và Si dùng 4 electron hoá trị của chúng để liên kết với 4 electron hoá trị của 4 nguyên tử khác tạo ra cấu trúc tinh thể bền vững về mặt hoá học.

**Khái niệm lỗ trống trong chất bán dẫn.** Ở nhiệt độ trên 0 độ tuyệt đối ( $> -273^{\circ}\text{C}$ ) các electron trong mạng tinh thể sẽ chuyển động nhiệt. Nhiệt độ càng cao thì chuyển động nhiệt của các electron càng lớn. Chuyển động nhiệt này có thể làm cho 1 electron trong vùng hoá trị chuyển lên các quỹ đạo có năng lượng cao hơn. Lúc này electron là tự do. Nó di chuyển trong vùng dẫn. Cùng với sự tạo thành một electron tự do, sẽ xuất hiện một lỗ trống (mang điện tích dương) trong vùng hoá trị. Số electron tự do đúng bằng số lỗ trống. Lỗ trống là điểm khác biệt quan trọng nhất giữa bán dẫn và vật dẫn.

Nếu tồn tại 1 điện trường ngoài, thì trong chất bán dẫn sẽ có dòng chạy qua. Dòng này là dòng của các electron tự do và lỗ trống ngược chiều nhau. Độ dẫn điện của bán dẫn tinh khiết tăng theo nhiệt độ và có giá trị bé.

Để tăng độ dẫn điện của bán dẫn tinh khiết cần phải pha tạp (doping). Có 2 cách thường dùng:

**Pha tạp loại N (negative).** Để tăng số electron tự do trong bán dẫn, người ta pha tạp nguyên tử hoá trị 5 (còn gọi là chất cho, Photpho chẳng hạn) với bán dẫn tinh khiết, tạo thành bán dẫn loại N. Trong bán dẫn loại N, dễ dàng thấy rằng nguyên tử chất cho sẽ thừa 1 electron và làm cho số electron trong bán dẫn loại N chiếm đa số. Lỗ trống là phần tử thiếu số trong bán dẫn loại N.

**Pha tạp loại P (positive).** Người ta pha tạp nguyên tử hoá trị 3 (còn gọi là chất nhận, Nhôm chẳng hạn) vào bán dẫn tinh khiết để tạo ra chất bán dẫn

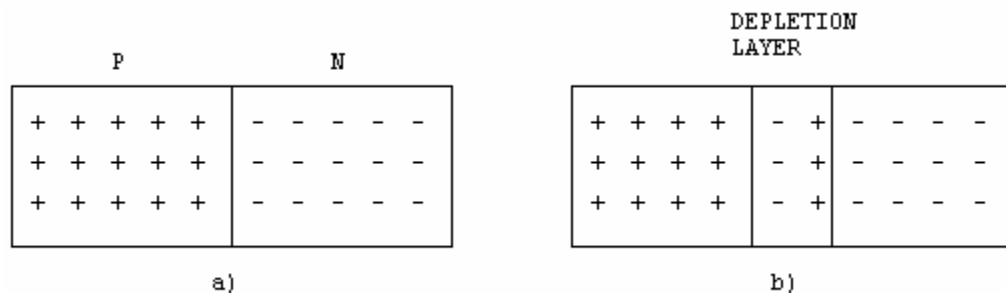
loại P. Trong bán dẫn loại P, phần tử tải điện đa số là lỗ trống, phần tử tải điện thiểu số là electron tự do.

Bán dẫn loại N và loại P có thể chế tạo từ tinh thể Ge hoặc Si. Công nghệ Ge là công nghệ của những năm 60 (thế kỷ 20). Ngày nay, hầu hết các chất bán dẫn là Si.

## II.2 TIẾP XÚC PN

Giả sử có một mẫu bán dẫn Si tinh khiết. Người ta pha tạp mẫu bán dẫn sao cho phía bên trái là bán dẫn loại P, còn phía bên phải là bán dẫn loại N. Biên giới giữa bán dẫn loại P và bán dẫn loại N gọi là **tiếp xúc PN**. Tiếp xúc PN đã dẫn đến các phát minh về diode, transistor, IC (Integrated Circuits)... Việc hiểu biết tính chất của tiếp xúc PN là cơ sở để hiểu biết hoạt động của các linh kiện và thiết bị bán dẫn.

**Tiếp xúc PN** còn gọi là một diode bán dẫn (từ nay trở đi gọi là diode). Chúng ta hãy xem xét các tính chất của một diode khi không phân cực.



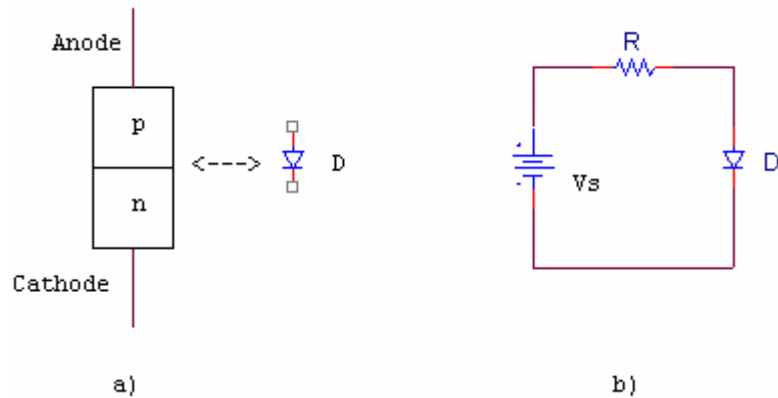
Hình 2-1: Tiếp xúc PN không phân cực

Tại lớp tiếp xúc, sẽ hình thành một vùng nghèo điện tích (depletion layer) do sự khuếch tán của electron từ N vào P sau đó các electron này tái hợp với lỗ trống làm cho số phần tử tải điện tại vùng này giảm. Sự khuếch tán cũng tạo ra một hàng rào thế năng hướng từ N sang P. Ở nhiệt độ 25<sup>0</sup>C, hàng rào thế năng có giá trị cỡ 0.3V đối với Ge và 0.7V đối với Si. Sự hiện diện của rào thế ngăn cản quá trình khuếch tán tiếp tục và hệ ở trạng thái dừng.

## II.3 DIODE BÁN DẪN CÓ PHÂN CỰC

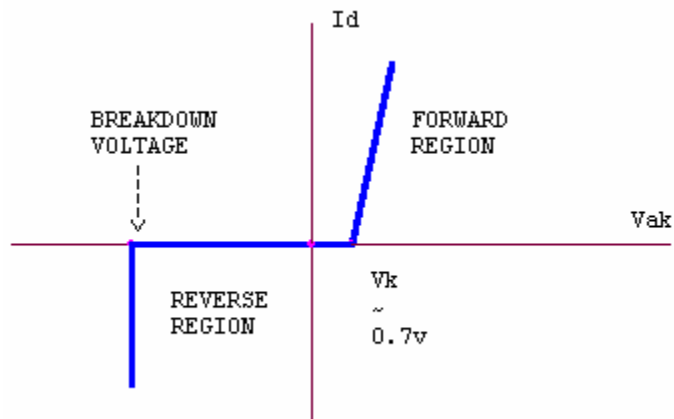
Hình 2-2a cho thấy ký hiệu của một diode. Bên bán dẫn P gọi là Anode (ký hiệu là A), bên bán dẫn N gọi là Cathode (ký hiệu là K). Trên sơ đồ người ta ký hiệu diode như một mũi tên chỉ từ P sang N hay từ Anode sang Cathode.

Hình 2-2b trình bày một mạch diode. Trong mạch này diode được phân cực thuận ( $V_a > V_k$ ). Sự phân cực thuận làm cho các electron tự do bên bán dẫn N và lỗ trống bên bán dẫn P vượt qua mối nối tạo thành dòng điện trong diode (dòng  $I_{ak}$ ).



Hình 2-2: Diode và phân cực thuận diode

Trong phòng thí nghiệm có thể setup một mạch như hình 2-2b. Bằng cách đo dòng và thế trên diode ứng với phân cực thuận và phân cực nghịch ( $V_a < V_k$ ) có thể vẽ giản đồ quan hệ giữa dòng và thế trên diode như hình 2-3.



Hình 2-3: Giản đồ IV của diode

Theo hình 2-3, khi phân cực thuận, dòng qua diode sẽ không đáng kể cho đến khi  $V_k >$  hàng rào thế năng (barrier potential). Ngược lại, khi phân cực ngược, có 1 dòng điện rất bé qua diode cho đến điện áp đặt lên diode vượt qua điện thế đánh thủng (Breakdown Voltage =BV).

Trong vùng phân cực thuận, điện thế tại đó dòng  $I_k$  bắt đầu tăng nhanh gọi là điện thế mỗi nối (knee voltage) của diode. Điện thế mỗi nối có giá trị bằng hàng rào thế năng. Khi phân tích mạch diode phân cực thuận chúng ta thường xét xem điện thế trên diode là bé hơn hay lớn hơn điện thế mỗi nối. Nếu lớn hơn, diode dễ dàng dẫn điện. Nếu bé hơn, diode không dẫn điện (dẫn điện kém). Chúng ta định nghĩa điện thế mỗi nối của diode silicon là:

$$V_k \approx 0.7V \quad (2-1)$$

Điện thế mỗi nối của diode germanium là 0.3V. Hiện nay diode germanium ít được dùng, nhưng điện thế mỗi nối của nó thấp là một ưu điểm và vì vậy một số ứng dụng vẫn dùng diode germanium.

Khi điện thế trên diode vượt qua điện thế mỗi nối thì dòng qua diode tăng nhanh và theo quy luật tuyến tính. Lúc này diode đóng vai trò như điện trở. Chúng ta gọi điện trở này là điện trở Bulk ( $R_B$ ) của diode.

$$R_B = R_P + R_N \quad (2-2)$$

Trong đó  $R_P$  và  $R_N$  là điện trở tương ứng của vùng P và vùng N. Chúng phụ thuộc vào mật độ pha tạp và kích thước của các vùng này. Thông thường  $R_B < 1\Omega$ . Chúng ta chỉ quan tâm đến  $R_B$  của diode trong gần đúng bậc 3. Trong giáo trình này chúng ta không xem xét đến gần đúng bậc 3.

Nếu dòng điện qua diode quá lớn, sự quá nhiệt sẽ phá hủy diode. Vì vậy trong bảng số liệu kỹ thuật (data sheet) của nhà máy sản xuất có ghi dòng cực đại của một diode. Đó là dòng điện tối đa mà diode có thể hoạt động bình thường và không làm giảm tuổi thọ cũng như các đặc trưng của nó. Dòng thuận tối đa của 1 diode thường được ghi bằng  $I_{max}$ ,  $I_{F(max)}$ ,  $I_o$ ... Ví dụ diode 1N456 có  $I_{max} = 135mA$ .

Có thể tính công suất tiêu tán (power dissipation) của một diode giống như tính công suất tiêu tán của một điện trở. Nó bằng tích giữa dòng và thế trên diode.

$$P_D = V_D \cdot I_D \quad (2-3)$$

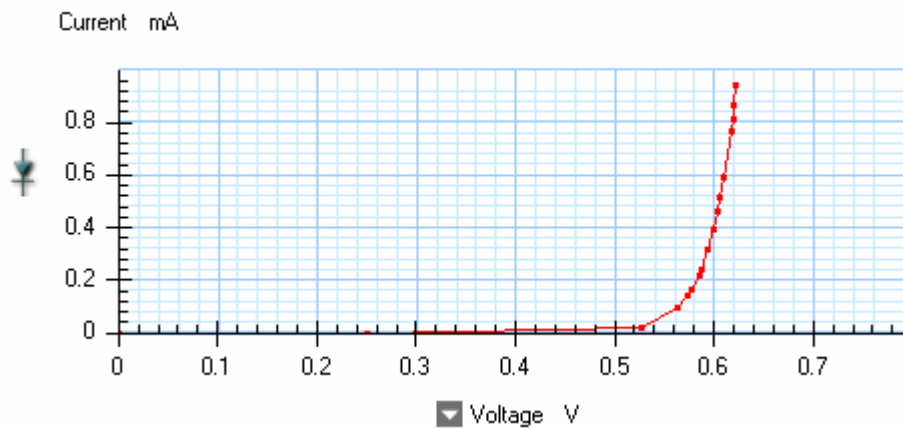
Giới hạn công suất (power rating) của một diode là công suất tối đa mà diode có thể tiêu tán và không làm giảm tuổi thọ cũng như các đặc tính khác. Nếu ký hiệu giới hạn công suất là  $P_{max}$  thì

$$P_{max} = V_{max} \cdot I_{max} \quad (2-4)$$



## II.4 DIODE LÝ TƯỞNG

Hình 2-4 cho thấy giản đồ dòng thế của một diode trong vùng phân cực thuận. Lưu ý rằng dòng qua diode xấp xỉ bằng 0 cho đến khi thế trên diode đạt tới giá trị hàng rào thế. Trong vùng lân cận 0.6V đến 0.7V dòng qua diode tăng. Khi thế trên diode lớn hơn 0.8V dòng qua diode tăng rất mạnh và đồ thị là đường thẳng.



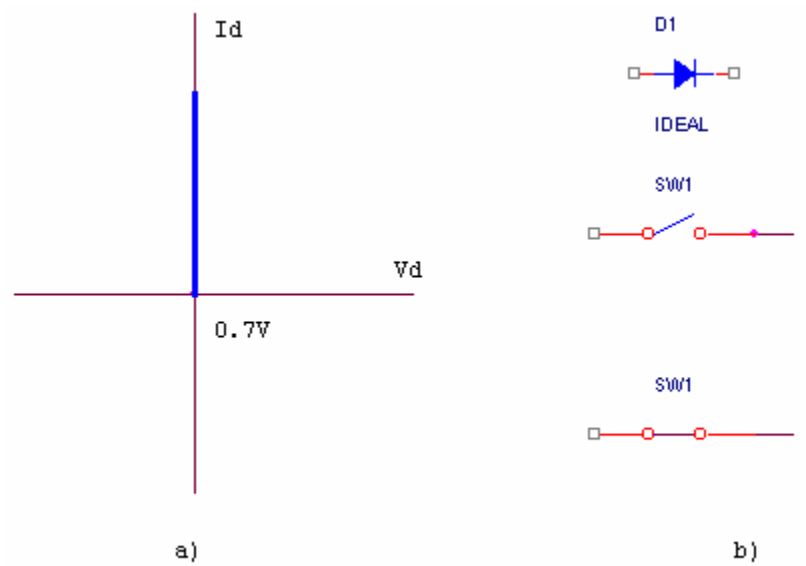
Hình 2-4: Giản đồ dòng thế của diode phân cực thuận

Tùy thuộc vào kích thước vật lý và mật độ pha tạp, các đặc trưng của diode như dòng thuận tối đa, giới hạn công suất... có thể có giá trị rất khác nhau. Mặc dù giá trị dòng và thế của các diode thì khác nhau nhưng dạng của giản đồ quan hệ giữa dòng và thế trên mọi diode tương tự nhau như hình 2-4. Tất cả các diode silicon đều có điện thế mối nối xấp xỉ 0.7V.

Trong khi phân tích mạch, hầu như chúng ta không cần sự chính xác tuyệt đối. Do đó có thể dùng gần đúng cho diode. Chúng ta hãy bắt đầu bằng gần đúng lý tưởng. Theo đó, diode như một thiết bị có tính chất sau: nó dẫn điện tốt (điện trở bằng 0) khi phân cực thuận, và hoàn toàn không dẫn điện (điện trở vô cùng) khi phân cực ngược.

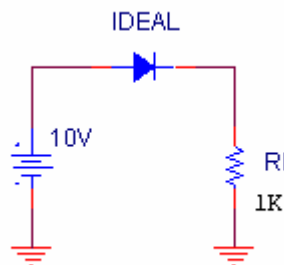
Hình 2-5a chỉ ra giản đồ dòng thế của 1 diode lý tưởng. Theo đó diode lý tưởng có điện trở bằng 0 khi phân cực thuận và có điện trở bằng vô cùng khi phân cực ngược. Nói cách khác, diode lý tưởng giống như một công tắc

(switch) như hình 2-5b. Nó đóng (close) khi phân cực thuận và hở (open) khi phân cực ngược.



Hình 2-5: Đường cong dòng thế của diode lý tưởng và mô hình

**Ví dụ:** Dùng mô hình diode lý tưởng tính thế trên tải và dòng tải trên sơ đồ hình 2-6.



Hình 2-6: Mạch diode lý tưởng

Do diode phân cực thuận, nó như công tắc đang đóng. Do đó toàn bộ nguồn thế 10V đặt lên trở tải. Vậy

$$V_L = 10V$$

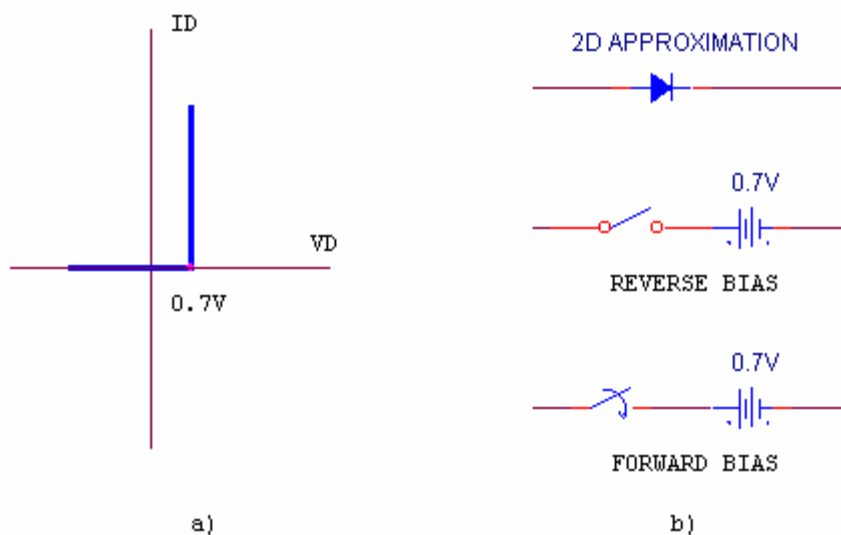
Theo định luật Ohm, dòng tải bằng:

$$I_L = 10V / 1K = 10mA$$

## II.5 GẦN ĐÚNG BẬC 2 CỦA DIODE

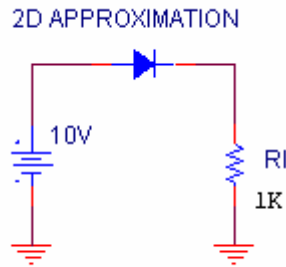
Chúng ta sẽ dùng gần đúng bậc 2 khi muốn tính chính xác hơn các giá trị dòng và thế trên diode.

Hình 2-7a chỉ ra giản đồ dòng thế của 1 diode trong gần đúng bậc 2. Theo đó, sẽ không có dòng qua diode chừng nào thế trên diode chưa vượt qua giá trị 0.7V. Hình 2-7b cho thấy mạch tương đương của diode silicon trong gần đúng bậc 2. Nó gồm một công tắc nối tiếp với một hàng rào thế 0.7V. Nếu thế Thevenin áp lên diode lớn hơn 0.7V, diode sẽ đóng (dẫn điện thuận). Khi diode đang dẫn, thế rơi trên diode là 0.7V đối với mọi giá trị của dòng thuận. Nói cách khác, nếu thế Thevenin bé hơn 0.7V, công tắc là hở và không có dòng qua diode.



**Hình 2-7:** Gần đúng bậc 2 của diode

**Ví dụ.** Dùng gần đúng bậc 2 của diode để tính dòng, thế và công suất tiêu tán trên diode cho ở mạch hình 2-8.



**Hình 2-8:** Mạch diode gần đúng bậc 2

Do diode phân cực thuận, nó tương đương một pin 0.7V. Điều này có nghĩa là thế trên tải bằng

$$V_L = 10V - 0.7V = 9.3V$$

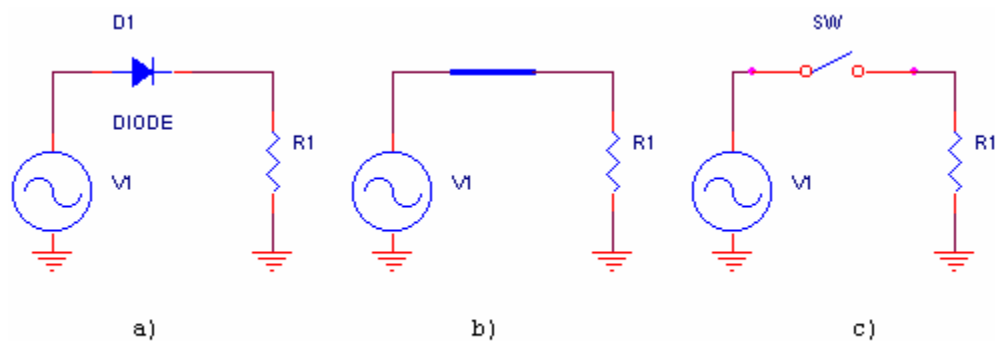
Theo định luật Ohm, dòng tải bằng

$$I_L = 9.3V / 1K = 9.3mA$$

Công suất tiêu tán trên diode bằng

$$P_D = (0.7V) \cdot (9.3mA) = 6.51mW$$

## II.6 NẮN ĐIỆN NỬA CHU KỲ



**Hình 2-9:** Mạch nắn điện dùng diode

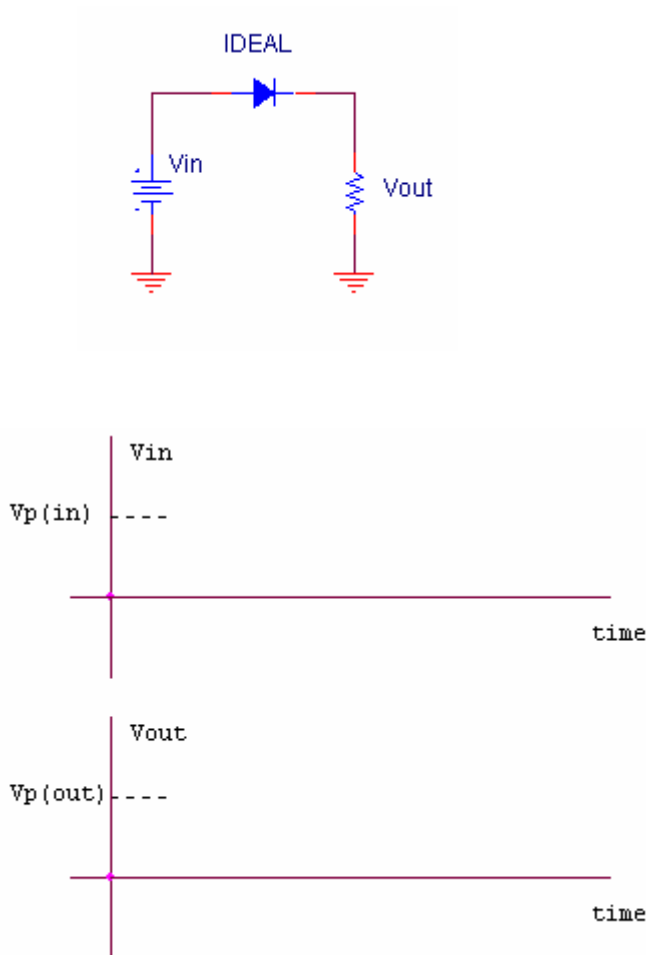
Hình 2-9a chỉ ra mạch nắn điện nửa chu kỳ. Nguồn ac tạo ra một điện áp xoay chiều. Giả sử rằng diode là lý tưởng. Ở nửa chu kỳ dương của nguồn thế, diode phân cực thuận. Diode sẽ như một công tắc đang đóng như hình 2-9b. Tín hiệu nửa chu kỳ dương của nguồn thế sẽ xuất hiện trên điện trở tải.

Vào nửa chu kỳ âm của nguồn thế, diode như công tắc hở mạch, trên tải sẽ không có 1 điện thế nào (hình 2-9c).

### **Dạng sóng lý tưởng.**

Mạch nắn điện nửa sóng như hình 2-10a sẽ chỉ làm cho diode dẫn trong nửa chu kỳ dương và không dẫn trong nửa chu kỳ âm của nguồn ac.

Hình 2-10b là giản đồ dạng sóng lối vào. Nó là một sóng sin có giá trị tức thời là  $v_{in}$  và giá trị đỉnh là  $V_p(in)$ . Do đó trên tải sẽ thu được tín hiệu dạng nửa sóng. Điều này cũng có nghĩa là dòng qua điện trở tải là dòng một chiều.



Hình 2-10: Dạng sóng của mạch nắn  $\frac{1}{2}$  chu kỳ

Tín hiệu lối ra nửa sóng như hình 2-10c là điện thế dc kiểu xung. Nó tăng từ 0 đến cực đại, rồi lại giảm về 0, sau đó tiếp tục bằng 0 trong nửa chu kỳ âm. Để có được nguồn dc dùng cho các thiết bị điện tử, cần phải lọc dạng tín hiệu nửa sóng này.

Giá trị của thế lối ra lý tưởng là:

$$V_p(\text{out})=V_p(\text{in}) \quad (2-5)$$

### **Giá trị dc của tín hiệu nửa sóng.**

Giá trị dc của 1 tín hiệu là giá trị trung bình của tín hiệu đó. Nếu đo tín hiệu bằng vôn kế dc thì số chỉ chính là giá trị trung bình của tín hiệu.

Giá trị dc của một tín hiệu nửa sóng bằng:

$$V_{dc}=V_p/\pi=0.318V_p \quad (2-6)$$

Theo (2-6) nếu giá trị đỉnh của 1 tín hiệu nửa sóng là 100V thì giá trị dc hay trung bình của nó là 31.8V.

Tần số tín hiệu lối ra của mạch nắn nửa chu kỳ bằng tần số của nguồn ac lối vào.

$$f_{\text{out}} = f_{\text{in}} \quad (2-7)$$

Có thể thấy rằng dòng qua diode bằng dòng qua tải.

$$I_{\text{diode}} = I_{dc}$$

### **Gần đúng bậc 2.**

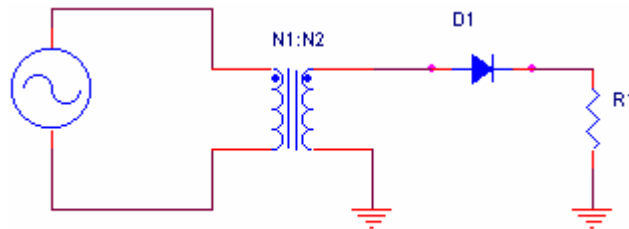
Trong gần đúng bậc 2, thế trên tải nhỏ hơn thế của nguồn vào một lượng 0.7V như công thức sau:

$$V_p(\text{out})=V_p(\text{in}) - 0.7V \quad (2-84)$$

## **II.7 BIẾN THẾ**

Tại Việt nam, nhà điện cung cấp điện áp lưới (Line Voltage) danh định 220V, tần số 50Hz. Điện áp thực mà chúng ta nhận được có thể thay đổi từ 200V đến 240V phụ thuộc vào thời điểm trong ngày, vị trí và nhiều yếu tố khác. Điện áp 220V là quá cao đối với các mạch điện trong các thiết bị điện tử. Đó là lý do tại sao phải dùng một biến thế hạ thế trong hầu hết các thiết bị điện tử. Biến thế giảm điện áp lưới từ 220V xuống các giá trị bé hơn và an toàn hơn để dùng với diode, transistor và các thiết bị bán dẫn khác.

Hình 2-11 cho thấy một biến thế. Điện áp lưới đặt trực tiếp vào cuộn sơ cấp của biến thế. Gọi  $N_1/N_2$  là tỷ số giữa cuộn sơ cấp và cuộn thứ cấp. Để biến thế là hạ thế thì  $N_1 > N_2$ .



Hình 2-11: Biến thế

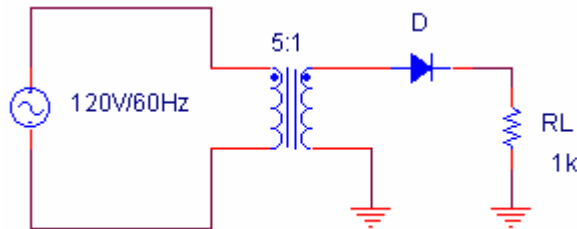
**Dấu chấm pha.** Để biểu thị quan hệ về pha trên các cuộn dây của biến thế người ta dùng dấu chấm pha. Các đầu dây có dấu chấm sẽ có cùng pha.

Quan hệ giữa thế và số vòng trên 2 cuộn của biến thế là:

$$V_2/V_1=N_2/N_1 \quad (2-9)$$

Có thể dùng công thức (2-9) cho giá trị đỉnh, giá trị hiệu dụng và giá trị tức thời.

Ví dụ: Tính thế tải cực đại và thế tải dc cho mạch hình 2-12.



Hình 2-12: Biến thế và mạch nắn 1/2 chu kỳ

Giải: Tỷ số biến thế là 5:1. Do đó thế trên cuộn thứ cấp là:

$$V_2=120/5=24V$$

Điện thế đỉnh trên cuộn thứ cấp bằng:

$$V_p= 24/0.707 = 34V$$

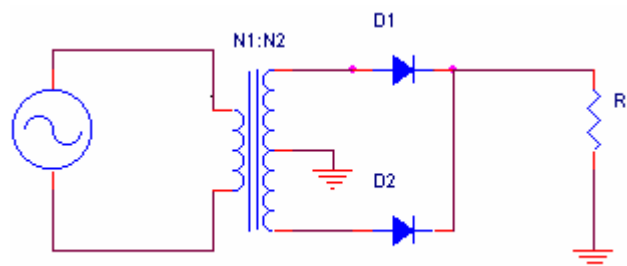
Với diode lý tưởng, thế đỉnh trên tải bằng 34V.

Thế dc trên tải bằng:

$$V_{dc}=V_p/\pi = 34/ \pi= 10.8V$$

Nếu dùng xấp xỉ bậc 2 cho diode, thế đỉnh trên tải và thế dc trên tải tương ứng bằng 33.3V và 10.6V

## II.8 NẮN CẢ CHU KỲ



Hình 2-13: Mạch nắn cả chu kỳ

Hình 2-13 là một mạch nắn điện toàn sóng (cả chu kỳ). Cuộn thứ cấp của biến thế có điểm giữa được nối đất. Mạch nắn toàn sóng tương đương 2 mạch nắn nửa sóng ghép lại. Vì biến thế có điểm giữa, mỗi mạch nắn có điện thế vào chỉ bằng  $\frac{1}{2}$  điện thế cuộn thứ cấp. Diode D1 dẫn trong nửa chu kỳ dương trong khi đó D2 dẫn trong nửa chu kỳ âm. Trong cả 2 nửa chu kỳ, điện thế trên tải có cùng cực tính, dòng tải vì vậy là dòng có hướng. Chúng ta sẽ phân tích một số đặc điểm của tín hiệu lối ra toàn sóng này sau đây.

### Giá trị dc hay trung bình.

Do tín hiệu lối ra toàn sóng gồm 2 lần tín hiệu nửa sóng nên giá trị dc cho bởi:

$$V_{dc} = 2V_p / \pi = 0.63V_p \quad (2-10)$$

Theo (2-6) giá trị trung bình bằng 63% giá trị đỉnh. Ví dụ, nếu điện thế đỉnh là 10V thì giá trị trung bình lối ra là 6.3V.

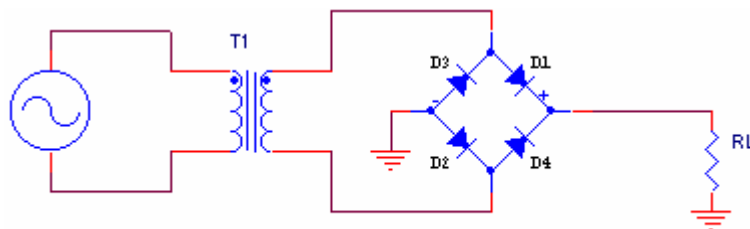
Tần số của tín hiệu lối ra toàn sóng là gấp đôi tần số nguồn ac lối vào.

$$f(\text{out}) = 2f(\text{in}) \quad (2-11)$$

Dòng qua mỗi diode bằng một nửa dòng tải:

$$I_{\text{diode}} = I_{dc} / 2$$

## II.9 NẮN CẦU



Hình 2-14: Mạch nắn cầu

Hình 2-14 cho thấy một mạch nắn cầu. Mạch nắn cầu tương tự mạch nắn toàn sóng vì nó tạo ra điện thế lối ra toàn sóng. Mạch dùng 4 diode. D1 và D2 dẫn trong nửa chu kỳ dương. D3 và D4 dẫn trong nửa chu kỳ âm.

Mạch nắn cầu tương đương với hai mạch nửa sóng nối lại với nhau. Trong cả hai nửa chu kỳ, thế trên tải có cùng cực tính và dòng tải là dòng một



hướng. Mạch này biến đổi thế ac ở đầu vào thành thế dc dạng xung (nhấp nhô) ở lối ra.

**Giá trị trung bình và tần số tín hiệu lối ra.**

Bởi vì mạch nắn cầu tạo ra tín hiệu toàn sóng nên phương trình tính giá trị dc hay trung bình cho thế ra là:

$$V_{dc} = 2V_p / \pi = 0.63V_p$$

và tần số tín hiệu lối ra bằng

$$f(\text{out}) = 2 f(\text{in})$$

Dòng qua diode trong mạch nắn cầu bằng một nửa dòng tải

$$I_{\text{diode}} = 1/2 I_{dc}$$

Gần đúng bậc 2 cho thấy rằng thế ra đỉnh bằng thế vào đỉnh trừ đi 2 lần thế rơi trên diode:

$$V_p(\text{out}) = V_p(\text{in}) - 1.4V \quad (2-12)$$

## II.10 CÁC BỘ LỌC

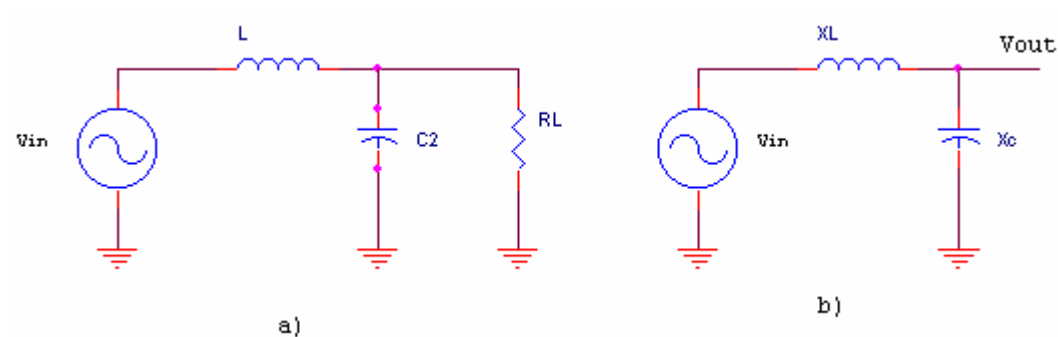
Lối ra của các mạch nắn là thế dc dạng xung (nhấp nhô). Nói cách khác, thế trên tải gồm thành phần dc và ac. Có thể lọc bỏ thành phần ac bằng các mạch lọc (filter).

Có 2 loại mạch lọc được dùng phổ biến là:

- Mạch lọc LC
- Mạch lọc RC

**a) Mạch lọc LC**

Hình 2-15a cho thấy mạch lọc dùng cuộn L và tụ C.



Hình 2-15

Nguồn ac tạo ra dòng trên cuộn dây, tụ điện và điện trở. Dòng ac trên các linh kiện này phụ thuộc vào cảm kháng của L, dung kháng của C và R.

Cuộn L có trở kháng

$$X_L = 2\pi fL$$

Tụ C có dung kháng bằng

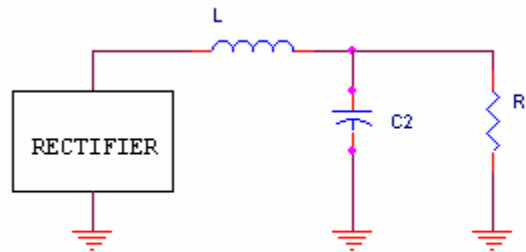
$$X_C = 1/2\pi fC$$

Mạch lọc LC được gọi là thiết kế tốt nếu tại tần số tín hiệu vào, giá trị của  $X_C$  nhỏ hơn  $R_L$  nhiều. Khi điều kiện này được thỏa mãn có thể bỏ qua  $R_L$  như mạch hình 2-15b. Mặt khác người ta cũng thiết kế sao cho tại tần số tín hiệu  $X_L$  lớn hơn  $X_C$  nhiều. Rõ ràng rằng khi đó thế xoay chiều trên tải sẽ rất bé và có thể xem bằng 0.

$$V_{out} = (X_C/X_L)V_{in} \quad (2-13)$$

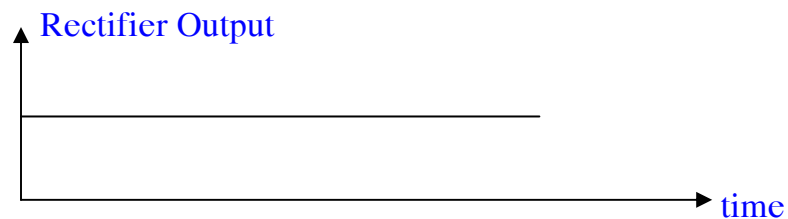
Ví dụ: tại tần số tín hiệu  $X_C = 100\Omega$  và  $X_L = 10K\Omega$  thì thế ra chỉ bằng 1/100 thế vào. Hay mạch lọc đã giảm thế xoay chiều 100 lần.

### Lọc lõi ra của các mạch nắn



Hình 2-16

Hình 2-16 chỉ ra mạch lọc nằm giữa mạch nắn và tải. Mạch nắn có thể là nửa sóng, toàn sóng hay nắn cầu. Để phân tích tác dụng của mạch lọc, chúng ta dùng nguyên lý chồng chất. Theo đó có thể xem rằng lõi ra của mạch nắn có 2 thành phần: thành phần dc và thành phần ac như hình 2-17.



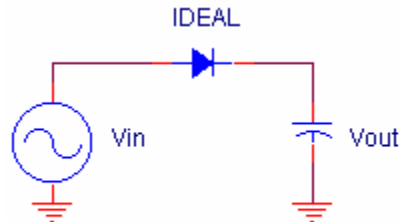
Hình 2-17

Như vậy, do tác dụng của bộ lọc thành phần ac trên tải rất bé có thể bỏ qua. Chỉ còn lại thành phần dc. Ở tần số dc, trở kháng của L rất bé, chỉ còn lại điện trở  $R_s$  của L mắc nối tiếp với tải  $R_L$ . Nếu như  $R_s$  rất bé so với  $R_L$  thì toàn bộ thế DC được đặt lên tải.

Nhược điểm của mạch lọc LC là do L có điện trở nội và làm giảm đáng kể thành phần dc nếu mạch yêu cầu dòng tải lớn. Tuy nhiên trong các bộ nguồn ổn áp kiểu xung dùng cho máy tính và các thiết bị điện tử khác, bộ lọc LC vẫn được dùng vì ở tần số cao ( $f=20\text{KHz}$ ) có thể thiết kế các cuộn cảm nhỏ mà vẫn đảm bảo hệ số lọc thành phần ac rất cao. Trong các mạch có dòng tải bé, người ta có thể thay thế mạch lọc LC bằng mạch lọc RC.

### b) Mạch lọc RC

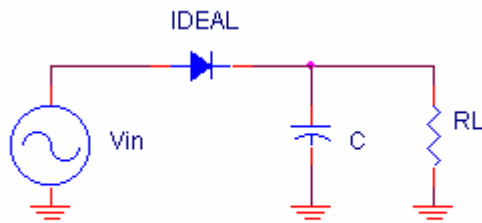
Hình 2-18 cho thấy mạch gồm tụ C, diode và nguồn ac.



Hình 2-18

Ban đầu tụ C không tích điện. Tại  $1/4$  chu kỳ đầu tiên, diode được phân cực thuận và dẫn, tụ C được nạp qua D. Thế cực đại trên tụ C bằng với thế nguồn vào thời điểm cuối  $1/4$  chu kỳ đầu tiên. Sau khi đạt đến giá trị đỉnh  $V_p$ , thế vào bắt đầu giảm, nhưng thế trên C vẫn giữ nguyên hằng số và bằng  $V_p$ . Lưu ý rằng mạch này không có tải.

### **Ảnh hưởng của điện trở tải.**



Hình 2-19: Mạch lọc có tải

Chúng ta nối song song với mạch lọc một trở tải  $R_L$  như hình 2-19. Khi đó diode D vẫn nạp điện cho tụ ở  $\frac{1}{4}$  chu kỳ đầu tiên. Trong thời gian còn lại của chu kỳ, tụ C phóng qua R với hằng số thời gian  $R_L C$ . Nếu thỏa mãn điều kiện  $R_L C > T$  (chu kỳ tín hiệu ac) thì thế trên tải dường như vẫn là hằng số, mặc dù có một chút xoay chiều trên tải mà chúng ta gọi là thế gợn sóng (ripple voltage).

Công thức tính giá trị đỉnh-đỉnh của thế gợn sóng là:

$$V_R = I / fC \quad (2-14)$$

Trong đó I là dòng tải, f là tần số gợn sóng. Nếu dòng tải bằng 0 thì thế gợn sóng  $V_R=0$ .

Ví dụ, nếu dòng tải  $I=10\text{mA}$ , tụ  $C=20\mu\text{F}$  và mạch nắn cầu ( $f=100\text{Hz}$ ) thì  $V_R = 10\text{mA}/(100.20\mu\text{F}) = 0.5\text{V}_{pp}$

## II.11 DIODE ZENER

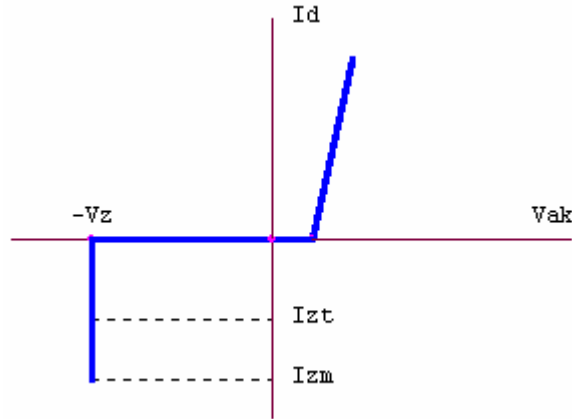
Các diode tín hiệu bé và nắn điện chỉ hoạt động ở vùng phân cực thuận mà không hoạt động ở vùng đánh thủng vì điều này sẽ làm hỏng chúng. Trong phần này chúng ta sẽ xem xét diode Zener, một diode silicon được chế tạo để hoạt động ở vùng đánh thủng. Diode zener là thành phần chính của mạch ổn áp. Đó là mạch giữ điện áp ra gần như là hằng số bất chấp sự thay đổi của thế vào và trở tải.



Hình 2-20: Diode Zener

### **Giải đồ IV của diode Zener.**

Hình 2-20 cho thấy hình vẽ của một diode Zener, ký hiệu là  $V_z$ . Bằng cách thay đổi mật độ pha tạp, nhà máy có thể tạo ra các Zener có điện áp đánh thủng từ 2V đến 1000V. Các diode này có thể hoạt động ở 3 vùng: phân cực thuận, đánh thủng và dòng rò.



Hình 2-21: Giải đồ IV của diode Zener

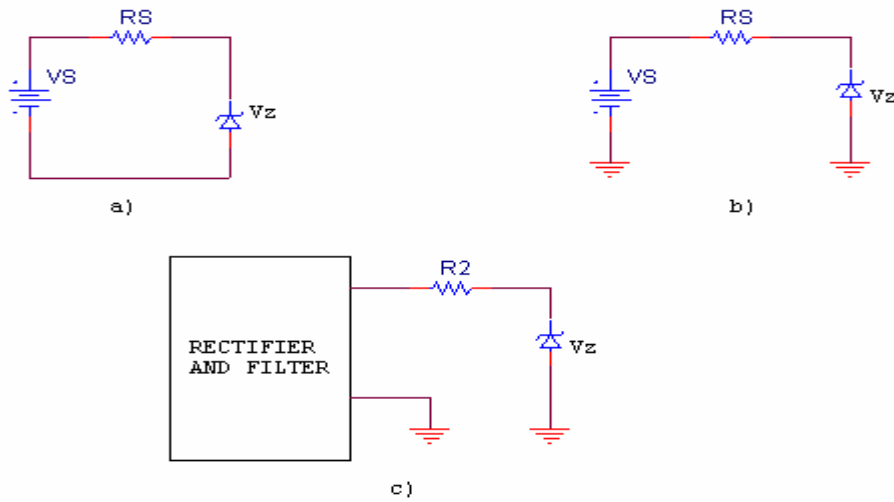
Hình 2-21 là giải đồ IV của 1 diode Zener. Trong vùng phân cực thuận, khi thế trên diode lớn hơn 0.7V nó dẫn điện như một diode silicon loại thường. Trong vùng rò, chỉ có một dòng rò rất nhỏ qua diode. Đặc tuyến của vùng đánh thủng gần như thẳng đứng. Khi diode Zener bị đánh thủng,  $V_z$  gần như hằng số. Trong bảng số liệu của nhà sản xuất người ta thường ghi  $V_z$  tại dòng test  $I_{ZT}$  nào đó. Hình 2-21 cũng cho biết dòng Zener tối đa  $I_{ZM}$ . Trong thiết kế mạch diode Zener phải đảm bảo diode Zener hoạt động ở dòng đánh thủng nhỏ hơn  $I_{ZM}$ .

#### **Điện trở Zener.**

Điện trở nội của diode gọi là điện trở Zener. Điện trở Zener có giá trị rất bé. Điện trở này phản ánh sự kiện dòng Zener tăng thì thế Zener tăng. Tuy nhiên sự tăng này rất ít, cỡ vài phần của Vôn. Đặc tuyến vùng đánh thủng càng dốc thì điện trở Zener càng bé. Trong các phân tích của chúng ta, điện trở Zener bị bỏ qua.

#### **Ổn áp Zener.**

Diode Zener đôi khi được gọi là diode ổn áp vì nó có đặc tính giữ điện áp giữa anode và cathode ( $V_z$ ) là hằng số bất chấp sự thay đổi của dòng qua diode. Để sử dụng tính chất này của Zener cần phải phân cực ngược nó như hình 2-22a. Mặt khác  $V_s$  phải lớn hơn điện áp đánh thủng  $V_z$ . Một điện trở nối tiếp  $R_s$  được sử dụng để hạn chế dòng qua Zener, không cho phép nó vượt quá giá trị giới hạn tối đa.



**Hình 2-22:** ổn áp dùng Zener

Hình 2-22b cho thấy một cách vẽ khác của hình 2-22a, trong đó mạch có điểm nối đất (Ground). Khi đó để đo hiệu điện thế giữa các điểm, có thể đo thế của chúng so với đất rồi tính ra hiệu điện thế giữa các điểm cần đo.

Hình 2-22c cho thấy hình vẽ lối ra của một bộ nguồn nối tới điện trở  $R_s$  và Zener. Mạch này gọi là mạch ổn áp Zener hay mạch Zener.

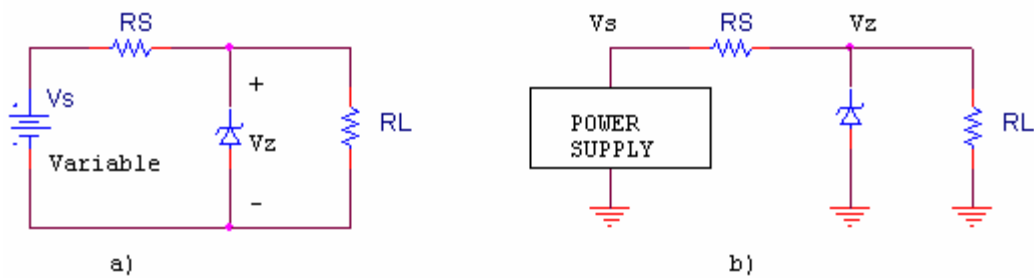
Dòng chảy qua  $R_s$  bằng:

$$I_s = (V_s - V_z) / R_s \quad (2-15)$$

$I_s$  cũng chính là dòng chảy qua Zener. Phải chọn  $R_s$  sao cho  $I_s < I_{ZM}$

Gần đúng lý tưởng của Zener là một pin, hay một nguồn thế có giá trị bằng  $V_z$ .

## II.12 ỔN ÁP ZENER CÓ TẢI



**Hình 2-23:** ổn áp Zener có tải

Hình 2-23a chỉ ra một mạch ổn áp Zener có tải. Hình 2-23b chỉ ra một mạch như thế và có điểm nối đất. Vì Zener hoạt động ở vùng đánh thủng, do đó nó giữ điện áp trên tải là hằng số bất chấp sự thay đổi của thế vào và dòng tải.

Giả sử diode Zener không nối vào mạch hình 2-23b. Khi đó thế Thevenin trên tải  $R_L$  là:

$$V_{TH} = V_s \cdot (R_L / (R_s + R_L)) \quad (2-16)$$

Thế Thevenin phải lớn hơn  $V_Z$ , nếu không sự đánh thủng sẽ không xảy ra.

Trên mạch hình 2-23, dòng qua  $R_s$  bằng:

$$I_s = (V_s - V_Z) / R_s \quad (2-17)$$

Thế trên tải bằng thế Zener

$$V_L = V_Z \quad (2-18)$$

Dòng tải bằng:

$$I_L = V_L / R_L \quad (2-19)$$

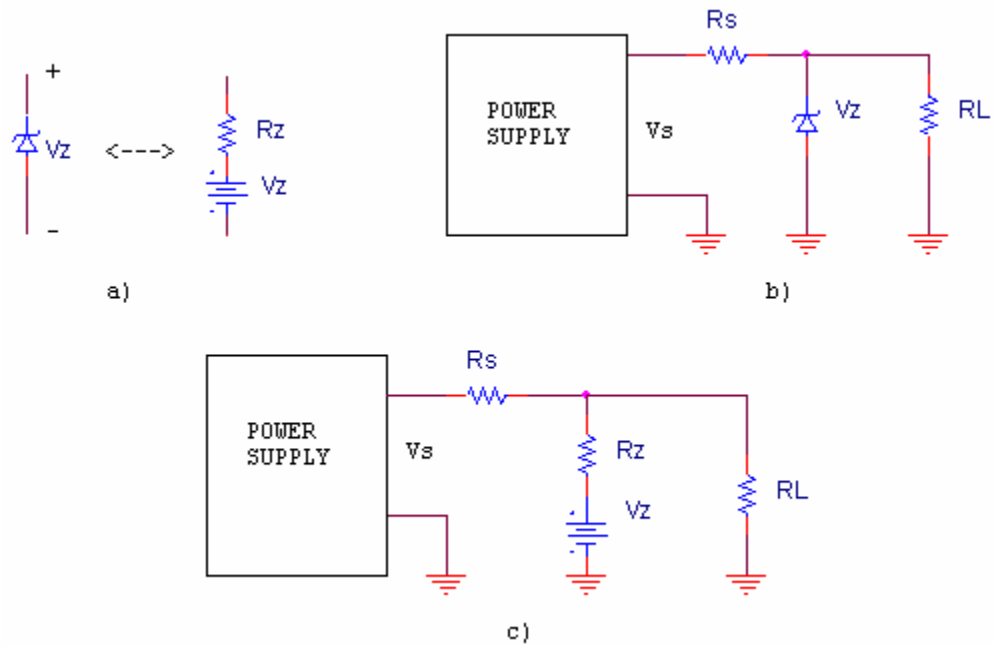
**Dòng Zener.** Theo định luật Kirchhoff, ta có:

$$I_s = I_L + I_Z \quad (2-20)$$

hay dòng qua Zener bằng:

$$I_Z = I_s - I_L \quad (2-21)$$

## II.13 GẮN ĐÚNG BẬC 2 CỦA DIODE ZENER



Hình 2-24

Hình 2-24a cho thấy gần đúng bậc 2 của diode Zener. Khi đó Zener như một nguồn thế lý tưởng  $V_Z$  nối tiếp với trở Zener  $R_Z$ . Tổng điện thế trên Zener bằng thế đánh thủng  $V_Z$  cộng với sụt thế qua điện trở Zener. Vậy ảnh hưởng của trở Zener lên thế lối ra như thế nào? Chúng ta hãy xét sơ đồ hình 2-24b và 2-24c.

Theo hình 2-24c, thế tải bằng:

$$V_L = V_Z + I_Z R_Z$$

Sự thay đổi của thế trên tải so với trường hợp lý tưởng là:

$$\Delta V = I_Z R_Z \quad (2-22)$$

Thường  $\Delta V$  nhỏ, cỡ vài phần của 1 Vôn.

Điện trở Zener cũng ảnh hưởng đến thế nhấp nhô trên tải. Mạch tương đương khi xét đến  $R_Z$  cho trên hình 2-24c.

Thành phần xoay chiều của thế ra bằng:

$$V_{R(out)} = V_{R(in)} R_Z / (R_Z + R_S)$$

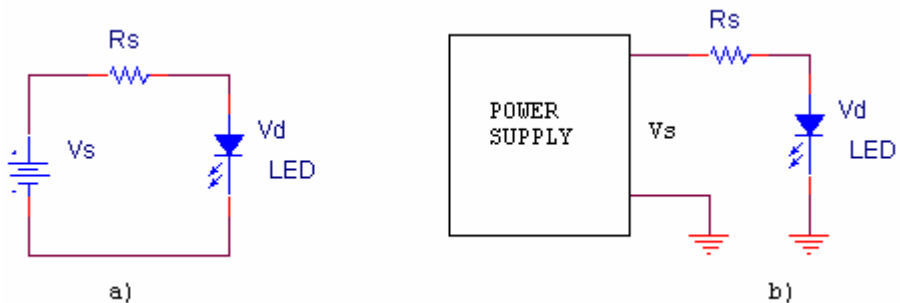


$$= R_Z V_{R(in)}/R_S \quad (2-23)$$

## II.14 CÁC THIẾT BỊ OPTOELECTRONICS

Optoelectronics là công nghệ kết hợp điện tử và quang học. Lĩnh vực này bao gồm các thiết bị hoạt động dựa trên tính chất của một tiếp xúc PN. Ví dụ các LED, photodiodes và optocouplers.

### a) LED (Light Emitting Diode)



Hình 2-25

Hình 2-25a vẽ một LED nối với nguồn thế qua điện trở  $R_S$ . Các mũi tên hướng ra ngoài ký hiệu cho ánh sáng phát ra từ LED. Khi phân cực thuận, các electron tự do khi qua mối nối PN bị bắt bởi lỗ trống. Những electron này chuyển từ mức có năng lượng cao về mức năng lượng thấp hơn và vì vậy giải phóng năng lượng. Đối với diode thường năng lượng này phát ra dưới dạng nhiệt. Nhưng đối với LED, năng lượng giải phóng dưới dạng các bức xạ ánh sáng. Trong nhiều ứng dụng, LED được dùng thay cho các đèn neon vì hoạt động ở điện thế thấp, tuổi thọ cao và tắt mở nhanh. Bằng cách dùng các vật liệu khác nhau, nhà máy có thể tạo ra các LED màu đỏ, lục, lơ, vàng, cam hoặc hồng ngoại (không nhìn thấy). Các LED phát ra màu thấy được thường dùng trong các bộ chỉ thị như màn hình của máy tính tay, đồng hồ. Các LED hồng ngoại dùng trong các thiết bị báo động, các ổ đĩa CD.

### **Dòng và thế của LED.**

Điện trở  $R_S$  trên hình 2-25b nhằm hạn chế dòng qua LED không vượt qua giới hạn cho phép. Theo định luật Ohm:

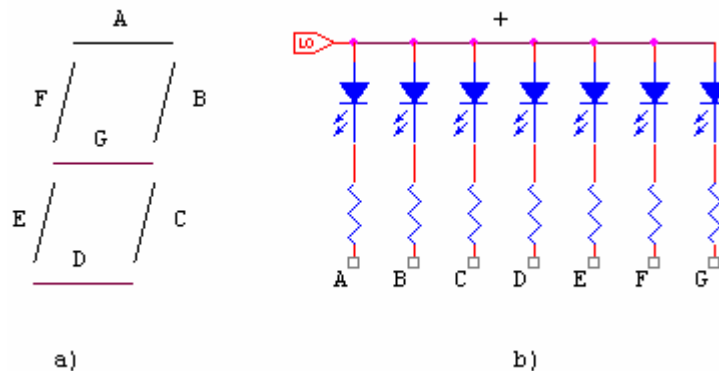
$$I_S = (V_S - V_D)/R_S \quad (2-24)$$

Hầu hết các diode thương mại có  $V_D$  từ 1.5V đến 2.5V với dòng từ 10mA đến 50mA. Giá trị chính xác của thế rơi trên LED phụ thuộc vào dòng qua LED, màu LED và một số yếu tố khác. Trừ khi được chú thích, chúng ta sẽ dùng  $V_D=2V$  để phân tích các mạch có liên quan đến LED trong giáo trình này.

Thế đánh thủng của LED rất bé, chỉ cỡ 3-5V. LED thường dùng để chỉ thị tình trạng có hay không có nguồn của thiết bị và gọi là power LED. Khi đó một diode chỉnh lưu được mắc song song ngược với LED để bảo vệ LED khỏi bị đánh thủng khi phân cực ngược.

### b) Seven - Segment Display.

Hình 2-26a trình bày một bộ chỉ thị dùng đèn 7 đoạn. Nó chứa 7 đoạn LED, đánh số từ A đến G. Mỗi LED là 1 đoạn vì nó là 1 thành phần của ký tự mà nó chỉ thị. Hình 2-26b là sơ đồ của một bộ chỉ thị 7 đoạn. Các điện trở dùng để hạn chế dòng qua LED. Các LED được nối theo kiểu anode chung (cũng có đèn 7 đoạn nối cathode chung). Bằng cách nối đất một hoặc nhiều điện trở, chúng ta sẽ nhận được các số từ 0 đến 9 và các ký tự như A,b,C,d, E và F.



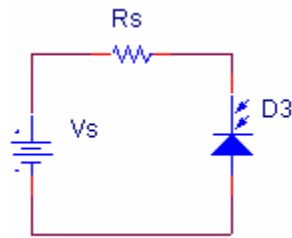
Hình 2-26: Chỉ thị 7 đoạn

### c) Photodiode

Trong các phần trước chúng ta biết rằng, một thành phần trong dòng ngược của diode là dòng các hạt mang thiểu số. Các hạt mang này tồn tại vì năng lượng nhiệt có thể làm bật (dislodging) một electron hoá trị khỏi quỹ đạo của nó tạo ra một electron tự do và một lỗ trống. Thời

gian sống của các hạt mạng thiếu số này ngắn, nhưng trong khi tồn tại, chúng tạo ra dòng ngược của diode.

Khi chiếu sáng mối nối PN, năng lượng của ánh sáng (năng lượng nhiệt) có thể làm bật electron hoá trị ra khỏi quỹ đạo và tạo ra một cặp electron - lỗ trống. Photodiode là một diode được chế tạo sao cho nó nhạy với ánh sáng. Trong photodiode, một cửa sổ dẫn ánh sáng đến mối nối PN. Ánh sáng càng mạnh, số cặp electron -lỗ trống càng nhiều và do đó, dòng ngược trong photodiode càng lớn.

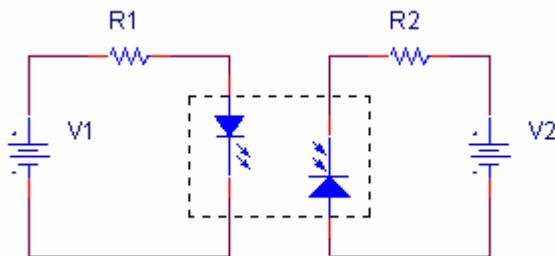


Hình 2-27: Photodiode

Hình 2-27 trình bày ký hiệu của một photodiode. Các mũi tên dùng để biểu thị cho ánh sáng tới. Photodiode thường được ứng dụng khi ghép nó với LED như phần sau đây.

#### **d) Optocoupler**

Optocoupler còn gọi là optoisolator gồm một LED và một photodiode đặt trong một vỏ nhựa với 4 chân ra, như hình 2-28.



Hình 2-28

Nguồn thế bên trái tạo ra một dòng qua LED. Ánh sáng của LED tạo ra dòng ngược trong photodiode. Nếu như thế lối vào thay đổi thì dòng qua photodiode thay đổi. Nghĩa là chúng ta đã dùng một thế (ở bên trái hình vẽ - đầu vào) để điều khiển một dòng (ở bên phải hình vẽ - đầu ra).

Ưu điểm chủ yếu của optocoupler là sự cách điện giữa đầu vào và đầu ra của hệ. Điện trở giữa đầu vào và ra của optocoupler có thể đến hàng ngàn  $M\Omega$ . Optocoupler có thể dùng trong các ứng dụng cao thế, ở đó thế giữa đầu vào và ra có thể khác nhau đến hàng ngàn vôn.

e) **Laser diode (Light Amplification by Stimulated Emission of Radiation)**

Sự chuyển ngẫu nhiên từ mức năng lượng cao về mức năng lượng thấp của các electron tự do trong LED làm cho ánh sáng phát ra từ một LED có pha thay đổi trong một vùng rộng từ  $0$  đến  $360^{\circ}$ . Nguồn sáng có pha thay đổi gọi là nguồn sáng không kết hợp (Incoherent).

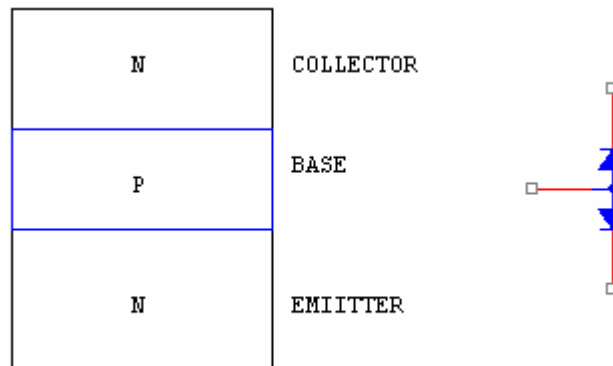
Trong diode laser thì khác. Nó có thể tạo ra một nguồn sáng kết hợp (Coherent), nghĩa là nguồn sáng mà tất cả các sóng ánh sáng đều cùng pha với nhau. Trong diode laser người ta dùng buồng cộng hưởng để kích thích các ánh sáng bức xạ tại một tần số với pha duy nhất. Đặc điểm của nguồn sáng do diode laser phát ra là đơn sắc, cùng pha và vì vậy có thể hội tụ tốt và có cường độ lớn. Các diode laser cũng được gọi là các laser bán dẫn. Chúng có thể tạo ra ánh sáng nhìn thấy cũng như không nhìn thấy (hồng ngoại). Diode laser được dùng nhiều trong các ứng dụng công nghệ thông tin và truyền thông.

### Chương III

## **TRANSISTOR**

Năm 1951 William Shockley đã phát minh ra transistor, còn gọi là transistor lưỡng cực. Transistor là một thiết bị bán dẫn mà nó có thể khuếch đại các tín hiệu điện tử (dòng hoặc thế). Sự khám phá ra transistor đã dẫn đến việc tạo ra hàng loạt các thiết bị điện tử và các mạch tích hợp mà chúng ta thường gọi là các IC (Integrated Circuits). Nhờ có IC mà ngày nay chúng ta có máy tính, truyền hình, điện thoại di động, người máy và v.v...

### **III.1 CẤU TẠO VÀ CÁC DÒNG ĐIỆN TRONG TRANSISTOR**



Hình 3-1: Cấu tạo của transistor

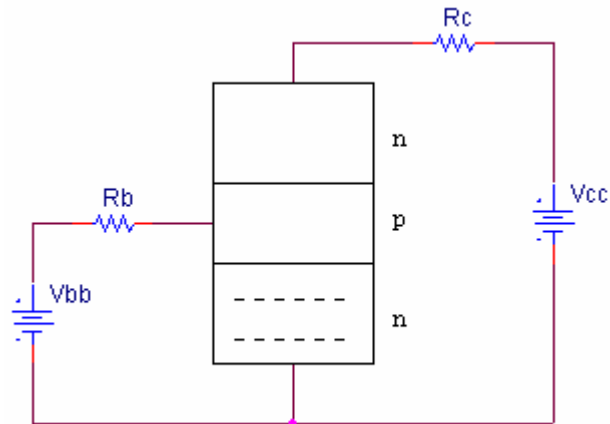
Một transistor có 3 vùng bán dẫn NPN như hình 3-1. Vùng dưới cùng là vùng Emitter (phát), vùng giữa gọi là vùng Base (gốc), vùng trên cùng gọi là vùng Collector (góp). Transistor như hình 3-1 gọi là transistor NPN.

Transistor cũng được chế tạo dưới dạng PNP. Hoạt động của transistor NPN và transistor PNP về cơ bản giống nhau. Trong chương này chúng ta sẽ chỉ phân tích transistor NPN.

Trong transistor vùng emitter được pha tạp nhiều. Vùng base pha tạp rất ít. Vùng collector pha tạp trung bình.

Transistor trong hình 3-1 gồm 2 mối nối PN: mối nối base-emitter và mối nối base-collector. Vì vậy có thể xem như transistor gồm 2 diode nối ngược nhau. Cấu hình này chỉ đúng khi transistor không phân cực. Người ta thường dùng cấu hình này khi đo thử transistor. Nếu 2 diode trong transistor còn tốt thì nhiều khả năng transistor cũng còn tốt.

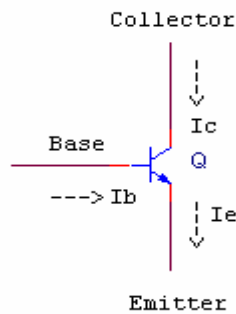
Transistor không phân cực giống như 2 diode nối ngược. Mỗi diode có một hàng rào thế cỡ 0.7V (đối với transistor Si). Khi nối transistor với nguồn ngoài, sẽ có các dòng điện qua các vùng khác nhau trong transistor.



Hình 3-2: phân cực transistor

Trong hình 3-2 dấu trừ biểu thị các electron tự do trong vùng emitter. Nguồn  $V_{BB}$  phân cực thuận diode base - emitter. Trong khi nguồn  $V_{CC}$  phân cực ngược diode base - collector. Do emitter được pha tạp mạnh, nó phát các electron vào base. Sự pha tạp ít của base có ý nghĩa là làm cho hầu hết các electron từ emitter không bị tái hợp mà sẽ đến collector. Collector có nghĩa là thu góp. Cực collector có nhiệm vụ thu các electron từ emitter.

Khi được phân cực, nếu thế  $V_{BB}$  lớn hơn hàng rào thế, các electron từ emitter sẽ vào vùng base. Về mặt lý thuyết các electron này sẽ đi theo 2 hướng. Thứ nhất là vào vùng base qua  $R_B$  để đến cực dương của nguồn  $V_{BB}$ . Thứ hai là vào collector. Do sự pha tạp của vùng base rất ít, nên hầu hết các electron di chuyển lên vùng collector. Tại đây nó bị hút về cực dương của nguồn  $V_{CC}$  qua điện trở  $R_C$ .



Hình 3-3: Ký hiệu và các dòng điện trong transistor

Hình 3-3 chỉ ra ký hiệu trên sơ đồ của một transistor.

Theo định luật Kirchoff, ta có:

$$I_E = I_B + I_C \quad (3-1)$$

Điều này chứng tỏ rằng: Dòng cực e bằng tổng của dòng cực c và dòng cực b.

Vì dòng cực b rất bé nên có thể xem rằng

$$I_E \approx I_C \quad (3-2)$$

Người ta định nghĩa Hệ số  $\alpha_{dc}$  của transistor như sau:

$$\alpha_{dc} = I_C / I_E \quad (3-3)$$

$\alpha$  lớn hơn 0.99 đối với các transistor công suất bé, còn đối với các transistor công suất  $\alpha$  lớn hơn 0.95.

Người ta cũng định nghĩa Hệ số khuếch đại dòng một chiều

$$\beta_{dc} = I_C / I_B \quad (3-4)$$

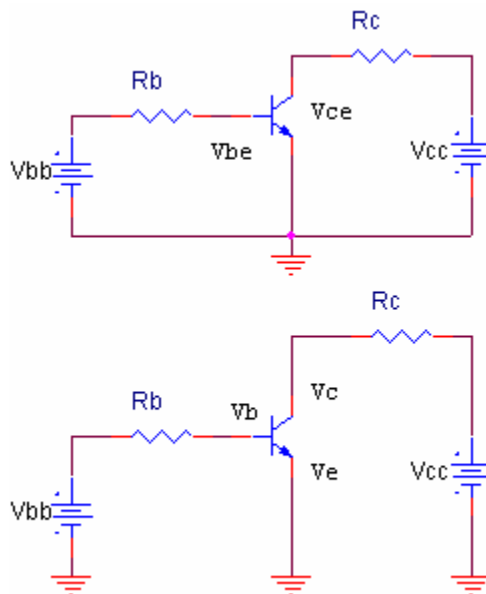
Ví dụ: một transistor có dòng  $I_C = 10\text{mA}$ , dòng  $I_B = 40\mu\text{A}$  thì

$$\beta_{dc} = 10\text{mA} / 40\mu\text{A} = 250$$

## III.2 NỐI EMITTER CHUNG

Có 3 cách nối transistor thường dùng là:

- Nối E chung (Common Emitter - CE)
- Nối C chung (Common Collector - CC)
- Nối B chung (Common Base - CB)



Hình 3-4: Nối CE

Trong hình 3-4, phần chung là emitter vì nó được nối với phần đất (Ground - GND) của 2 nguồn  $V_{BB}$  và  $V_{CC}$ . Vì vậy mạch này gọi là mạch chung emitter. Mạch có 2 vòng kín: vòng base và vòng collector.

Trong mạch vòng base, diode base - emitter được phân cực bởi nguồn  $V_{BB}$ . Điện trở  $R_B$  để hạn chế dòng base. Bằng cách thay đổi  $V_{BB}$  hoặc  $R_B$  chúng ta có thể thay đổi dòng base và vì vậy có thể thay đổi dòng collector. Nói cách khác chúng ta có thể điều khiển dòng collector bằng cách điều khiển dòng base. Điều này rất quan trọng vì chúng ta có thể dùng một dòng bé (dòng base) để điều khiển một dòng lớn (dòng collector).

Lưu ý: Các chỉ số dưới dùm trong mạch transistor có ý nghĩa như sau:

- Chúng ta dùng cùng chỉ số dưới để biểu thị nguồn nuôi. Ví dụ  $V_{BB}$  hoặc  $V_{CC}$ .
- Ngược lại chúng ta dùng chỉ số dưới khác nhau để biểu thị hiệu điện thế giữa các điểm. Ví dụ  $V_{CE}$  là hiệu điện thế giữa cực C và cực E.

Ta có:

$$V_{CE} = V_C - V_E$$

$$V_{CB} = V_C - V_B$$

$$V_{BE} = V_B - V_E$$

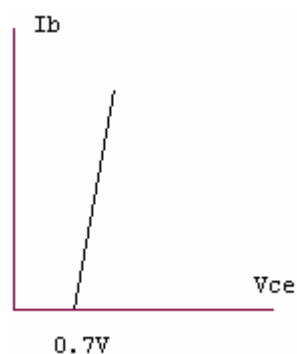
Trong cách nối CE, do  $V_E = 0$  nên

$$V_{CE} = V_C$$

$$V_{BE} = V_B$$

### III.3 ĐẶC TUYẾN BASE

Đặc tuyến base của một transistor giống như của một diode thường như hình vẽ 3-5. Vì vậy có thể dùng các gán đúng của diode khi phân tích mạch base của transistor.



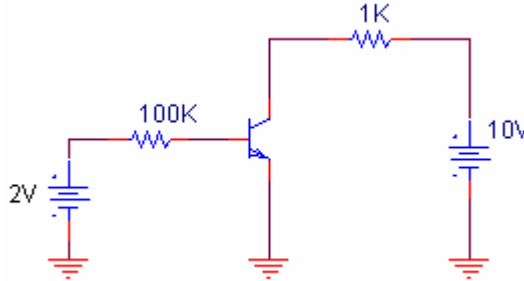
Hình 3-5: Đặc tuyến base của transistor



Áp dụng định luật Ohm cho mạch vòng base, chúng ta thu được:

$$I_B = (V_{BB} - V_{BE}) / R_B \quad (3-5)$$

Xem diode lý tưởng thì  $V_{BE}=0$ . Trong gần đúng bậc 2,  $V_{BE}=0.7V$ .



Hình 3-6

Ví dụ: Dùng gần đúng bậc 2 để tính dòng base trong hình 3-6.

Sụt thế qua  $R_B$  bằng bao nhiêu? Tính dòng collector biết  $\beta_{dc}=200$ .

Giải: Thế sụt qua  $R_B$  là

$$V_B = V_{BB} - V_{BE} = 2V - 0.7V = 1.3V$$

Dòng base bằng

$$I_B = (V_{BB} - V_{BE}) / R_B = (2V - 0.7V) / 100K\Omega = 13\mu A$$

Với hệ số khuếch đại  $\beta_{dc}=200$ , dòng collector bằng

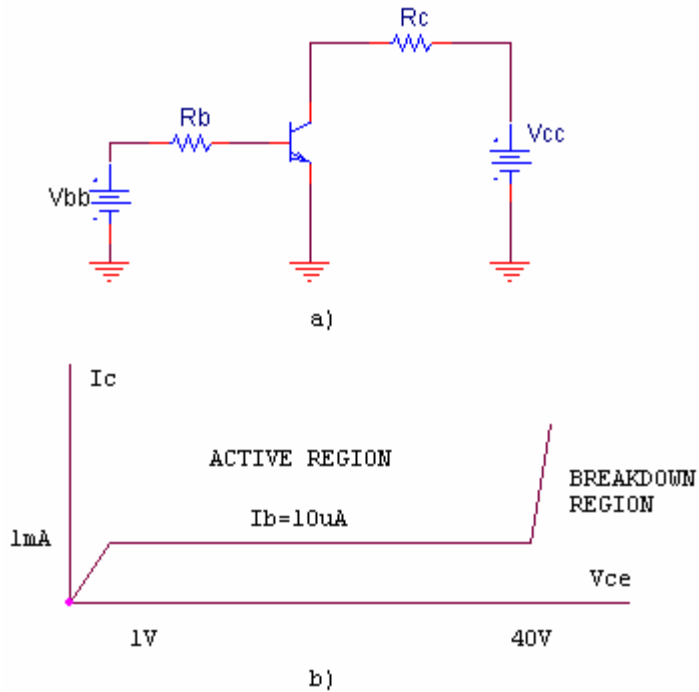
$$I_C = \beta_{dc} I_B = 200 \times 13\mu A = 2.6mA$$

### III.4 ĐẶC TUYẾN COLLECTOR

Đặc tuyến collector là đường cong mô tả quan hệ giữa dòng  $I_C$  và điện áp rơi trên transistor  $V_{CE}$ .

Trên hình 3-7a, bằng cách thay đổi  $V_{BB}$  hoặc  $V_{CC}$  có thể tạo ra các thế và dòng khác nhau trên transistor. Chẳng hạn, cố định  $V_{BB}$  để  $I_B=10\mu A$ , thay đổi  $V_{CC}$ , đo  $I_C$  và  $V_{CE}$  tương ứng, chúng ta có thể vẽ đặc tuyến collector như hình 3-7b. Số liệu được lấy ứng với transistor 2N3904. Các transistor khác có thể có số liệu khác nhưng dạng của đường cong là tương tự.

Khi  $V_{CE}$  bằng 0 diode collector không phân cực nên dòng collector bằng 0. Khi tăng  $V_{CE}$  tăng thì dòng  $I_C$  cũng tăng. Dòng  $I_C$  đạt giá trị bão hoà 1mA khi  $V_{CE}$  lớn hơn cỡ vài phần của một vôn.



Hình 3-7

Vùng dòng hằng của transistor trên hình 3-7b liên quan đến cấu tạo của transistor. Sau khi diode collector phân cực ngược, các electron tự do phát xạ từ emitter đều bị collector thu góp. Giá trị của dòng này chỉ phụ thuộc số electron phát xạ từ emitter hay chỉ phụ thuộc mạch base.

Khi tăng  $V_{CE}$  lên quá 40V, diode CB bị đánh thủng. Khi đó dòng collector tăng vọt. Hoạt động bình thường của transistor đã bị phá hủy. Các transistor không được phép hoạt động ở chế độ này vì nó sẽ bị hỏng. Các bảng số liệu của nhà sản xuất ghi giá trị thể đánh thủng collector-emitter là  $V_{CE(max)}$  hoặc  $BV_{CE}$ .

**Điện áp collector và công suất.**

Áp dụng định luật Kirchoff cho mạch vòng collector, ta có

$$V_{CE} = V_{CC} - I_C R_C \quad (3-6)$$

Công suất tiêu tán trên transistor bằng

$$P_D = V_{CE} \cdot I_C \quad (3-7)$$

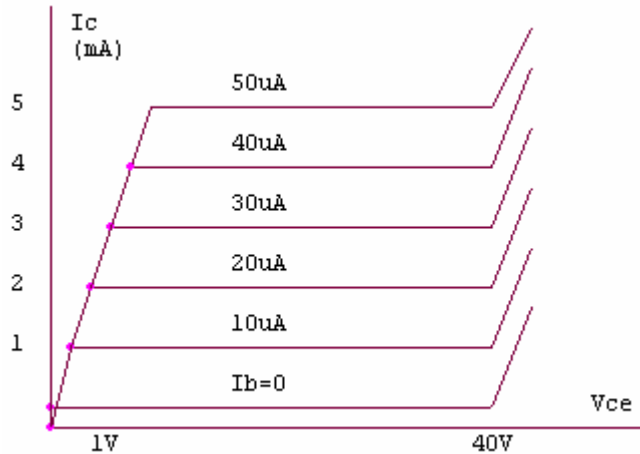
Công suất tiêu tán  $P_D$  làm cho nhiệt độ mối nối CB tăng nhanh. Công suất tiêu tán càng cao, nhiệt độ mối nối CB càng cao. Transistor sẽ hỏng nếu nhiệt độ mối nối vượt quá  $150^{\circ}C$ . Trong bảng số liệu của nhà sản xuất có ghi  $P_{D(max)}$  là công suất tiêu tán tối đa. Để transistor hoạt động bình thường thì  $P_D$  tính theo (3-7) phải bé hơn  $P_{D(max)}$ .

Theo đường đặc tuyến collector, transistor có 3 vùng hoạt động:

- ◆ Vùng tác động (active region) còn gọi là vùng hoạt.
- ◆ Vùng đánh thủng (breakdown region).
- ◆ Vùng bão hoà (saturation region).

Vùng tác động của transistor là vùng có  $I_C$  là hằng số. Vùng đánh thủng ứng với  $V_{CE} > 40V$ . Vùng bão hoà của transistor là vùng có thể  $V_{CE}$  bé hơn  $1V$ . Người ta sử dụng transistor ở vùng tác động để khuếch đại tín hiệu.

Nếu vẽ nhiều đường đặc tuyến collector trên cùng một đồ thị chúng ta có hình vẽ 3-8.



Hình 3-8

Trên hình 3-8 có một đường cong đặc biệt. Đó là đường ở dưới cùng, ở đó  $I_B=0$  nhưng vẫn có dòng  $I_C$  bé chảy qua transistor. Đó là dòng rỉ của transistor do các hạt mang thiểu số tạo ra. Miền giới hạn bởi đường cong có  $I_B=0$  và trục hoành gọi là vùng ngưng dẫn của transistor.

Tóm lại, transistor có 4 vùng hoạt động: tác động, ngưng dẫn, bão hoà và đánh thủng. Vùng tác động thường ứng dụng để khuếch đại tín hiệu bé. Vùng tác động còn gọi là vùng tuyến tính vì sự thay đổi của dòng  $I_C$  (tín hiệu ra) tỷ lệ tuyến tính với sự thay đổi của dòng  $I_B$  (tín hiệu vào). Trong các thiết bị kỹ thuật số và máy tính, transistor thường hoạt động ở vùng ngưng dẫn hoặc bão hoà hay còn gọi là chế độ khoá (switching mode).

Ví dụ: Transistor trên hình 3-9 có  $\beta_{dc} = 300$ . Tính  $I_B$ ,  $I_C$ ,  $V_{CE}$  và  $P_D$ .

Giải:

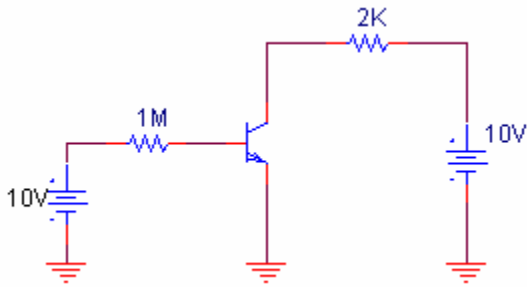
$$I_B = (V_{BB} - V_{BE}) / R_B = (10V - 0.7V) / 1M\Omega = 9.3\mu A$$

Dòng collector bằng

$$I_C = \beta_{dc} I_B = 300 \cdot (9.3\mu A) = 2.79mA$$

Thế collector bằng

$$V_{CE} = V_{CC} - I_C R_C = 10V - (2.79mA)(2K\Omega) = 4.42V$$

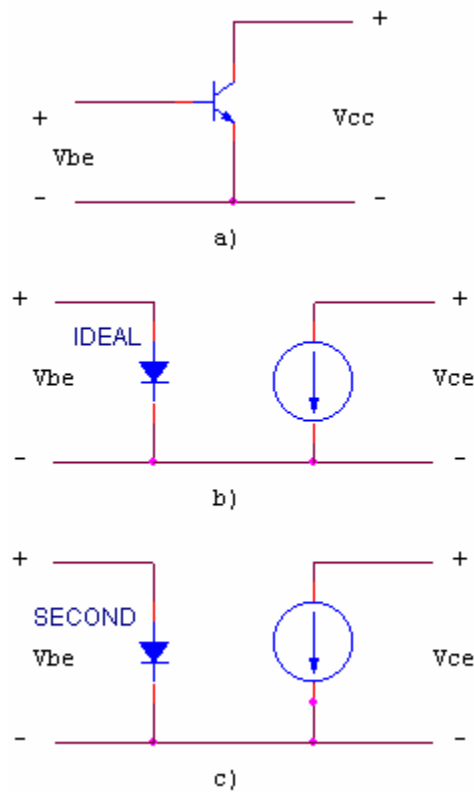


Hình 3-9

Công suất tiêu tán

$$P_D = V_{CE} \cdot I_C = (4.42V) \cdot (2.79mA) = 12.3mW$$

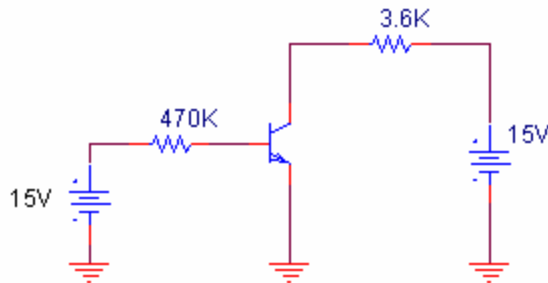
### III.5 CÁC GÃN ĐÚNG CHO TRANSISTOR



Hình 3-10

Hình 3-10a vẽ sơ đồ transistor.  $V_{BE}$  là thế trên diode emitter.  $V_{CE}$  là thế giữa các cực collector và emitter. Chúng ta hãy xem xét các mạch tương đương của transistor này.

- a) **Gần đúng lý tưởng.** Hình 3-10b là mạch tương đương của một transistor trong gần đúng lý tưởng. Mô hình này thường dùng khi phân tích mạch với nguồn  $V_{BB}$  lớn. Lúc này chúng ta dùng gần đúng lý tưởng cho diode emitter khi tính dòng base.
- b) **Gần đúng bậc 2.** Hình 3-10c là mạch tương đương của một transistor trong gần đúng bậc 2. Mô hình này thường dùng khi phân tích mạch với nguồn  $V_{BB}$  bé. Lúc này chúng ta dùng xấp xỉ bậc 2 cho diode emitter khi tính dòng base. Đối với transistor Si,  $V_{BE}=0.7V$ . Transistor Ge có  $V_{BE}=0.3V$ . Trong gần đúng bậc 2, dòng base và vì vậy dòng collector khác một chút so với gần đúng lý tưởng.
- c) **Xấp xỉ cao hơn.** Điện trở nội của diode emitter trở nên quan trọng chỉ trong các ứng dụng công suất cao mà ở đó dòng rất lớn. Ảnh hưởng của điện trở nội emitter là ở chỗ thế  $V_{BE}$  có thể đến cỡ 1V khi dòng base lớn. Cũng như diode base, điện trở nội của diode collector có ảnh hưởng đáng kể trong một số ứng dụng. Cùng với điện trở nội emitter và collector, transistor có một số các hiệu ứng bậc cao khác làm cho việc tính toán bằng tay trở nên khó khăn. Vì lý do này, các tính toán sau gần đúng bậc 2 cần phải sử dụng máy tính.



Hình 3-11

Ví dụ 1: Dùng gần đúng lý tưởng tính thế  $V_{CE}$  của transistor trong mạch hình 3-11.

Transistor lý tưởng có nghĩa là  $V_{BE}=0$ . Do đó dòng base bằng

$$I_B = 15V / 470K\Omega = 31.9\mu A$$

Dòng collector bằng

$$I_C = 100(31.9\mu A) = 3.19mA$$

Thế collector-emitter của transistor bằng

$$V_{CE} = 15V - (3.19mA)(3.6K\Omega) = 3.52V$$

Trong các tính toán, ít khi phải tính dòng emitter. Ở đây, chúng ta thử tính dòng emitter. Nó bằng

$$I_E = I_C + I_B = 3.19mA + 31.9\mu A = 3.2mA$$

Có thể thấy rằng  $I_E \approx I_C$

Ví dụ 2: Dùng gần đúng bậc 2 tính thế  $V_{CE}$  của transistor trong mạch hình 3-11.

Transistor gần đúng bậc 2 có  $V_{BE} = 0.7V$ , do đó dòng base bằng

$$I_B = 14.3V / 470K\Omega = 30.4\mu A$$

Dòng collector bằng

$$I_C = 100(30.4\mu A) = 3.04mA$$

Thế collector - emitter của transistor bằng

$$V_{CE} = 15V - (3.04mA)(3.6K\Omega) = 4.06V$$

**Nhận xét:**

Sự khác nhau của  $V_{CE}$  trong gần đúng lý tưởng và gần đúng bậc 2 là cỡ 0.5V. Tùy thuộc vào ứng dụng mà chúng ta sử dụng gần đúng hợp lý.

### III.6 SỰ THAY ĐỔI CỦA HỆ SỐ KHUYẾCH ĐẠI DÒNG ĐIỆN

Hệ số khuếch đại dòng điện của transistor phụ thuộc vào 3 yếu tố:

- ◆ loại transistor
- ◆ dòng collector
- ◆ nhiệt độ

Vì vậy Bảng số liệu của transistor 2N3904 cho thấy  $h_{FE_{min}} = 100$  và  $h_{FE_{max}} = 300$  tại nhiệt độ  $25^{\circ}C$  và dòng collector 10mA. Nếu chúng ta lắp ráp hàng ngàn mạch transistor 2N3904 chúng ta sẽ nhận được hệ số khuếch đại dòng bé nhất là 100 và lớn nhất là 300.

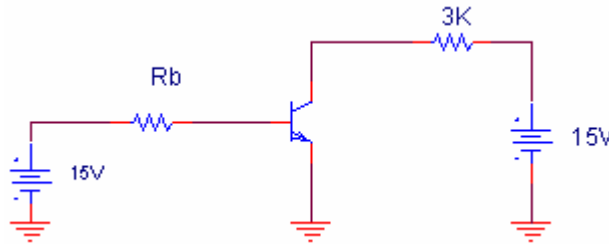
Ảnh hưởng của dòng collector và nhiệt độ đối với hệ số khuếch đại có thể thấy rõ qua bảng số liệu sau:

| $I_C$ (mA) | $h_{FE}$ (min) | $h_{FE}$ (max) |
|------------|----------------|----------------|
| 0.1        | 40             | -              |
| 1          | 70             | -              |
| 10         | 100            | 300            |
| 50         | 60             | -              |
| 100        | 30             | -              |

Rõ ràng rằng 2N3904 sẽ hoạt động tốt nhất tại dòng  $I_C=10\text{mA}$ .

### III.7 ĐƯỜNG TẢI (LOAD LINE)

Hình 3-12 là mạch transistor nối CE. Cho các giá trị của  $R_B$ ,  $\beta_{dc}$  chúng ta có thể tính dòng  $I_C$  và thế  $V_{CE}$ .



Hình 3-12

**Sự phân cực base.** Hình 3-12 là một ví dụ về mạch phân cực base mà nó có nghĩa là tạo ra một dòng base cố định. Ví dụ, nếu  $R_B=1\text{M}\Omega$  thì dòng base bằng  $14.3\mu\text{A}$  (gần đúng bậc 2). Dòng này luôn xấp xỉ giá trị  $14.3\mu\text{A}$  khi chúng ta thay thế transistor khác và ngay cả khi nhiệt độ thay đổi.

Nếu  $\beta_{dc}=100$  thì dòng collector xấp xỉ  $1.43\text{mA}$  và thế collector - base bằng:

$$V_{CE} = V_{CC} - I_C R_C = 15\text{V} - (1.43\text{mA}) \cdot (3\text{K}\Omega) = 10.7\text{V}$$

Vậy điểm tĩnh Q (quiescent) xác định dòng và thế trên transistor là

$$I_C = 1.43\text{mA} \text{ và } V_{CE} = 10.7\text{V}$$

**Phương pháp đồ thị.** Chúng ta có thể tìm điểm Q bằng phương pháp đồ thị dựa trên đường tải (load line) của transistor. Đường tải là đường cong cho quan hệ  $I_C$  và  $V_{CE}$  của một transistor theo phương trình sau

$$V_{CE} = V_{CC} - I_C R_C$$

Hay

$$I_C = (V_{CC} - V_{CE}) / R_C \quad (3-8)$$

(3-8) là một phương trình tuyến tính và đồ thị của nó là một đường thẳng. Đường này gọi là đường tải vì nó phản ánh sự ảnh hưởng của  $R_C$  lên  $I_C$  và  $V_{CE}$ .

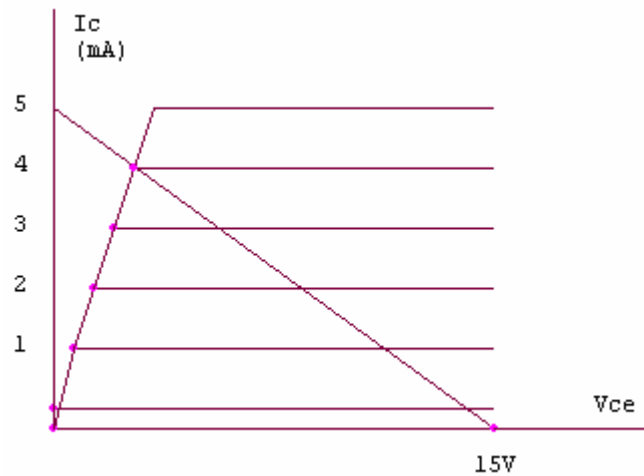
Theo (3-8), với  $V_{CC}=15\text{V}$ ,  $R_C = 3\text{K}\Omega$  đường tải cắt trục tung tại

$$I_C = V_{CC} / R_C = 15\text{V} / 3\text{K}\Omega = 5\text{mA}$$

và cắt trục hoành tại

$$V_{CE} = V_{CC} = 15\text{V}$$

Hình 3-13 cho thấy đường thẳng này.



Hình 3-13

Đường tải cho thấy điểm tĩnh Q có thể nhận những giá trị khả dĩ nào. Nói cách khác, đường tải cho biết tất cả các khả năng khả dĩ của điểm Q.

**Điểm bão hoà.** Khi  $R_B$  bé dòng  $I_C$  rất lớn làm cho thế  $V_{CE}$  của transistor đạt giá trị xấp xỉ 0V. Trong trường hợp này, transistor đã ở trạng thái bão hoà, nghĩa là dòng  $I_C$  đã tăng đến giá trị cực đại khả dĩ.

Điểm bão hoà là điểm mà ở đó đường tải tiệm cận với vùng bão hoà của transistor. Vì  $V_{CE}$  rất bé nên điểm bão hoà nằm gần đầu trên của đường tải. Điểm bão hoà cho thấy dòng collector đã đạt được giá trị cực đại khả dĩ đối với mạch đang xét. Ví dụ, trên mạch hình 3-14a transistor ở vào trạng thái bão hoà khi  $I_C = 5\text{mA}$ , lúc này  $V_{CE}$  xấp xỉ 0V.

Để tìm ra dòng  $I_C$  bão hoà chúng ta hình dung ngắn mạch cực C và cực E của transistor như hình 3-14b. Vì  $V_{CE} = 0\text{V}$  nên toàn bộ nguồn  $V_{CC} = 15\text{V}$  được đặt lên  $R_C$ . Vậy dòng  $I_C$  bão hoà bằng 5mA.

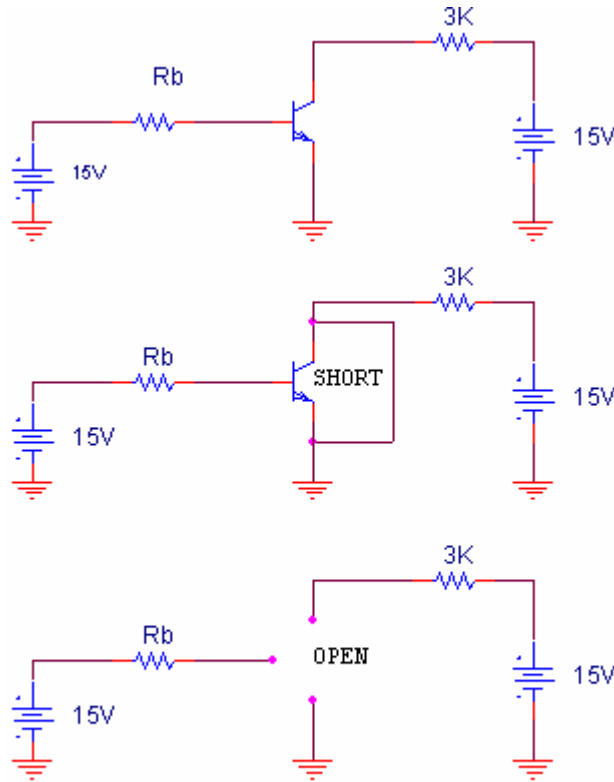
Biểu thức tính dòng bão hoà của transistor là

$$I_{C(\text{sat})} = V_{CC} / R_E \quad (3-9)$$

**Điểm ngưng dẫn (CUTOFF POINT).** Điểm ngưng dẫn là điểm mà tại đó đường tải tiệm cận với vùng ngưng dẫn của transistor. Vì dòng collector tại điểm cutoff rất bé nên điểm cutoff gần với đầu cuối phía dưới của đường tải như hình 3-13. Điểm cutoff cho thấy thế  $V_{CE}$  cực đại đối với một mạch. Để tìm điểm cutoff chúng ta hình dung transistor hở mạch giữa cực C và cực E như hình 3-14c, khi đó thế  $V_{CE}$  cực đại khả dĩ là 15V, bằng giá trị của nguồn cung cấp  $V_{CC}$ . Không có dòng  $I_C$  nên nguồn  $V_{CC}$  đặt hết lên cực C và E của transistor.

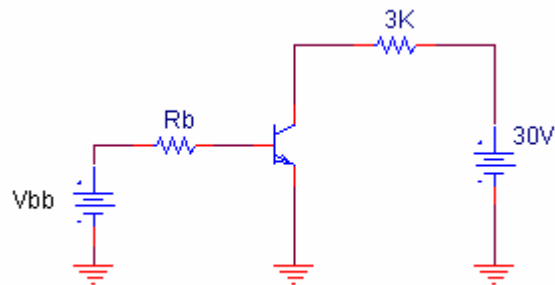


$$V_{CE(\text{cutoff})} = V_{CC} \quad (3-10)$$



Hình 3-14: Tìm 2 điểm đầu cuối của đường tải

Ví dụ: Tính dòng bão hoà và thế cutoff cho mạch hình 3-15



Hình 3-15

$$I_{C(\text{sat})} = V_{CC} / R_E = 30\text{V} / 3\text{K}\Omega = 10\text{mA}$$

$$V_{CE(\text{cutoff})} = V_{CC} = 30\text{V}$$

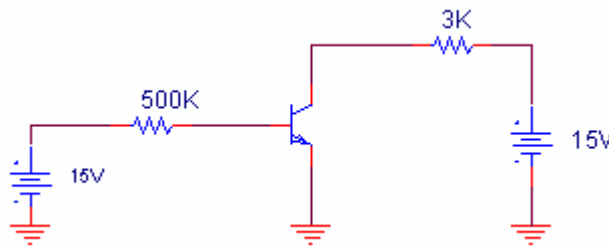
### III.8 ĐIỂM LÀM VIỆC VÀ CÁC CHẾ ĐỘ CỦA TRANSISTOR

Mỗi mạch transistor có một đường tải. Cho một mạch transistor bất kỳ có thể tính dòng bão hòa và thế ngưng dẫn. Chúng chính là các điểm tương ứng trên trục tung ( $I_{C(sat)}$ ) và trên trục hoành ( $V_{CC}$ ) của đường gánh.

Hình 3-16 chỉ ra một mạch phân cực base với  $R_B=500K\Omega$ . Đường tải đi qua  $I_{C(sat)} = 5mA$  và  $V_{CC} = 15V$  (thế giữa C và E khi transistor ngưng dẫn).

Giả sử transistor là lý tưởng, dòng base bằng

$$I_B = 15V / 500K\Omega = 30\mu A.$$



Hình 3-16

Nếu hệ số khuếch đại dòng là 100 thì dòng collector bằng

$$I_C = 100 \cdot 30\mu A = 3mA$$

Thế  $V_{CE}$  của transistor bằng

$$V_{CE} = 15V - (3mA)(3K\Omega) = 6V$$

Vậy transistor trên mạch hình 3-16 có điểm làm việc Q xác định bởi hai tham số

$$I_C = 3mA \text{ và } V_{CE} = 6V$$

Gọi là điểm Q (quiescent) vì nó xác định bởi thế và dòng DC (tĩnh) của transistor. Điểm Q của transistor có thể thay đổi vì nhiều lý do. Chẳng hạn hệ số khuếch đại dòng thay đổi từ 50 đến 150 nhưng  $I_B$  không đổi và bằng  $30\mu A$  thì dòng  $I_C$  thay đổi từ 1.5mA đến 4.5mA và vì vậy thế  $V_{CE}$  sẽ thay đổi từ 10.5V đến 1.5V. Nếu  $\beta_{dc}$  thay đổi nhiều hơn nữa, điểm Q có thể rơi vào vùng bão hòa hoặc ngưng dẫn và transistor trong mạch sẽ không còn khuếch đại tín hiệu nữa.

Rõ ràng là điểm Q của mạch phân cực base rất nhạy với sự thay đổi của  $\beta_{dc}$ . Người ta có thể khắc phục nhược điểm này bằng các mạch phân cực loại khác.

Có 2 chế độ làm việc của các transistor là:

- ◆ Chế độ khuếch đại

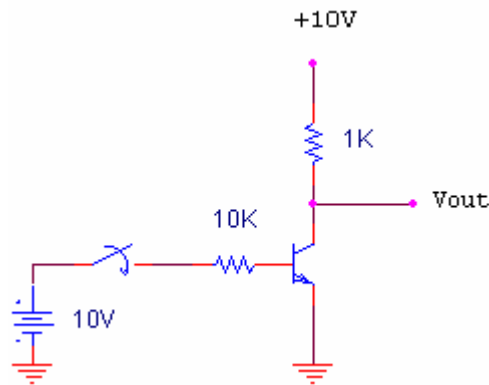
◆ Chế độ công tắc (switch)

Trong chế độ khuếch đại, điểm Q phải ở trong vùng tác động dưới mọi điều kiện. Trong chế độ switch, transistor chuyển giữa 2 trạng thái bão hoà và ngưng dẫn. Việc phân tích hoạt động của transistor ở chế độ switch sẽ được trình bày trong phần sau đây.

### III.9 TRANSISTOR SWITCH

Mạch phân cực base thường được dùng trong các mạch số (digital circuits) vì chúng được thiết kế để transistor làm việc ở vùng bão hoà và ngưng dẫn. Điểm Q của các transistor trong mạch số chuyển giữa vùng bão hoà và ngưng dẫn mà không ở trong vùng khuếch đại. Trong trường hợp này sự thay đổi của Q không quan trọng.

Hình 3-17 là sơ đồ một mạch transistor làm việc ở chế độ khoá. Trong đó transistor được phân cực base ở chế độ bão hoà sâu (dòng phân cực base lớn vì vậy làm cho dòng collector lớn hơn dòng collector bão hoà).



Hình 3-17

Khi SW đóng, dòng base xấp xỉ 1mA (lớn hơn dòng base cần thiết để transistor bão hoà) và làm cho transistor bão hoà sâu. Vì vậy thế  $V_{CE}=0$ . Khi SW hở, không có dòng base và vì vậy  $V_{CE}=V_{CC}$

Thế lối ra (Vout) của mạch transistor trong chế độ SW có 2 trạng thái: điện thế cao (bằng  $V_{CC}$ ) và điện thế thấp (0V). Lưu ý rằng giá trị chính xác của thế lối ra trong mạch switch không quan trọng mà quan trọng là ở chỗ 2 trạng thái này phải khác nhau (phân biệt được).

Mạch khoá còn gọi là mạch 2 trạng thái tương ứng với 2 trạng thái của lối ra.

Ví dụ: nếu nguồn  $V_{CC}$  trong mạch 3-17 giảm còn 5V và dòng rò collector là 50nA. Tính thế lối ra ở 2 trạng thái biết rằng thế bão hoà của transistor là  $V_{C(sat)}=0.15V$ .

Khi transistor ngưng dẫn thế trên transistor bằng

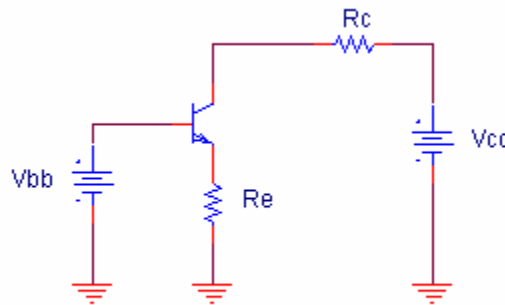
$$V_{CE}=5V - (50nA)(1K\Omega)=4.99995V$$

Khi transistor bão hoà, thế  $V_{CE}=V_{CE(sat)}=0.15V$

Trừ khi là nhà thiết kế, chúng ta sẽ không quan tâm đến giá trị chính xác của thế ra trong các mạch số. Vấn đề là ở chỗ thế ra phải có 2 giá trị phân biệt (gần 5 V khi ngưng dẫn và xấp xỉ 0V khi bão hoà).

### III.10 PHÂN CỰC EMITTER

Trong các mạch khuếch đại chúng ta cần mạch transistor mà điểm Q có khả năng ổn định hay không bị ảnh hưởng bởi sự thay đổi của  $\beta_{dc}$ .



Hình 3-18

Hình 3-18 cho thấy một mạch phân cực emitter trong đó điện trở đã được chuyển từ mạch base sang mạch emitter. Điểm Q của mạch này rất ổn định. Nó hầu như không thay đổi khi  $\beta_{dc}$  thay đổi từ 50 đến 150.

Nguồn  $V_{BB}$  bây giờ áp trực tiếp vào base. Thế cực E cho bởi phương trình

$$V_E = V_{BB} - V_{BE} \quad (3-11)$$

Nếu  $V_{BB} > 20V_{BE}$ , gần đúng lý tưởng sẽ cho kết quả chính xác. Nếu  $V_{BB} < 20V_{BE}$ , sẽ phải dùng gần đúng bậc 2.

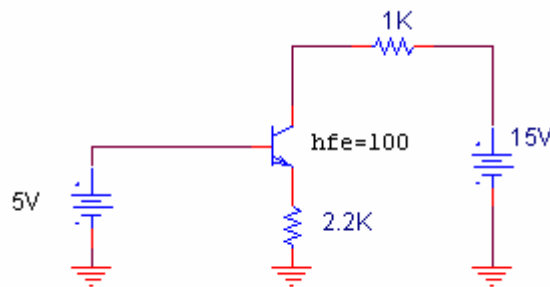
Chúng ta hãy tính các thông số của mạch phân cực emitter trên hình 3-19. Vì  $V_{BB} = 5V$ , chúng ta sẽ dùng xấp xỉ bậc 2. Khi đó

$$V_E = 5V - 0.7V = 4.3V$$

Dòng emitter bằng

$$I_E = 4.3V / 2.2K\Omega = 1.95mA$$

Điều này cũng có nghĩa là  $I_C$  xấp xỉ 1.95mA.



Hình 3-19

Dòng  $I_C$  này sẽ tạo một sụt thế 1.95V trên  $R_C$ . Do đó thế tại C so với đất bằng:

$$V_C = 15V - 1.95V = 13.1V$$

Từ nay trở đi, chúng ta gọi thế của collector so với đất là thế collector.

Hiệu điện thế rơi trên transistor bằng

$$V_{CE} = 13.1V - 4.3V = 8.8V$$

Vậy mạch phân cực emitter trên hình 3-19 có điểm Q xác định như sau:  $I_C = 1.95mA$  và  $V_{CE} = 8.8V$

Chúng ta hãy xem mạch chống lại sự thay đổi của hệ số khuếch đại dòng ra sao? Quá trình tính toán để tìm ra điểm Q đã cho thấy điều đó. Thực vậy, để tìm ra điểm Q của mạch phân cực emitter chúng ta đã:

- ◆ tính thế emitter
- ◆ tính dòng emitter
- ◆ tìm thế collector
- ◆ trừ thế collector cho thế emitter để có được thế  $V_{CE}$ .

Trong quá trình tính Q chúng ta không sử dụng  $\beta_{dc}$ . Điều đó có nghĩa là giá trị  $\beta_{dc}$  không ảnh hưởng đến giá trị của Q. Trên thực tế chúng ta đã cố định thế emitter và do đó cố định dòng emitter. Vì vậy kết quả là điểm Q không thay đổi khi  $\beta_{dc}$  thay đổi. Một cách chính xác thì  $\beta_{dc}$  có ảnh hưởng tới điểm Q như phân tích sau đây.

### Ảnh hưởng của hệ số khuếch đại dòng

$\beta_{dc}$  có ảnh hưởng nhỏ lên dòng collector. Dưới mọi điều kiện hoạt động, các dòng trong transistor quan hệ với nhau bởi phương trình

$$I_E = I_C + I_B$$

hay

$$I_E = I_C + I_C / \beta_{dc}$$

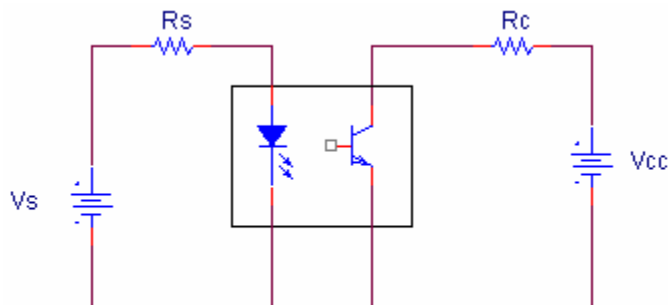
Vậy

$$I_C = (\beta_{dc} / (\beta_{dc} + 1)) I_E$$

Đại lượng  $\beta_{dc} / (\beta_{dc} + 1)$  gọi là hệ số sai số. Nếu  $\beta_{dc} = 100$  thì dòng  $I_C$  bằng 99% dòng  $I_E$ . Vậy khi ta nói dòng  $I_C$  bằng dòng  $I_E$  thì sai số là 1%.

## III.11 PHOTOTRANSISTOR

Phototransistor giống hệt như photodiode nhưng rất nhạy. Dòng ra của phototransistor lớn hơn dòng vào  $\beta_{dc}$  lần.



Hình 3-20

Nhưng phototransistor có tốc độ chậm hơn photodiode. Một photodiode có dòng ra cỡ nA nhưng có thể on/off cỡ ns. Trong khi đó dòng ra của phototransistor cỡ mA nhưng tốc độ on/off cỡ  $\mu$ s.

Hình 3-20 là sơ đồ một phototransistor nối với một LED. Thiết bị này gọi là Optocoupler hay còn gọi là isolator. Nó cách ly giữa đầu vào và đầu ra nhưng mọi sự thay đổi của thế vào  $V_s$  đều phản ánh ở thế ra trên collector của transistor. Thật vậy, nguồn  $V_s$  thay đổi làm cho dòng qua LED thay đổi và vì vậy lượng ánh sáng đến cực B phototransistor thay đổi. Kết quả là dòng  $I_C$  của transistor thay đổi và thế  $V_C$  thay đổi.

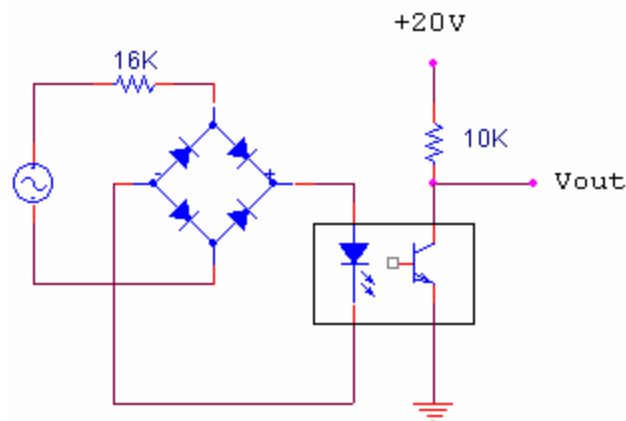
Ví dụ: Hình 3-21 là sơ đồ nhận biết thời điểm thế xoay chiều qua giá trị 0V (zero crossing) nhằm đồng bộ hoạt động của 1 số thiết bị với tần số điện lưới. Ưu điểm của sơ đồ là không cần biến thế cách ly giữa đầu vào và đầu ra. Phototransistor chính là mạch cách ly.

Bỏ qua sụt thế trên các diode nắn, dòng qua LED bằng

$$I_{LED} = 1.411(115V)/16K\Omega = 10.2mA$$

Giá trị bão hòa của dòng phototransistor là

$$I_{C(sat)} = 20V/ 10K\Omega = 2mA$$



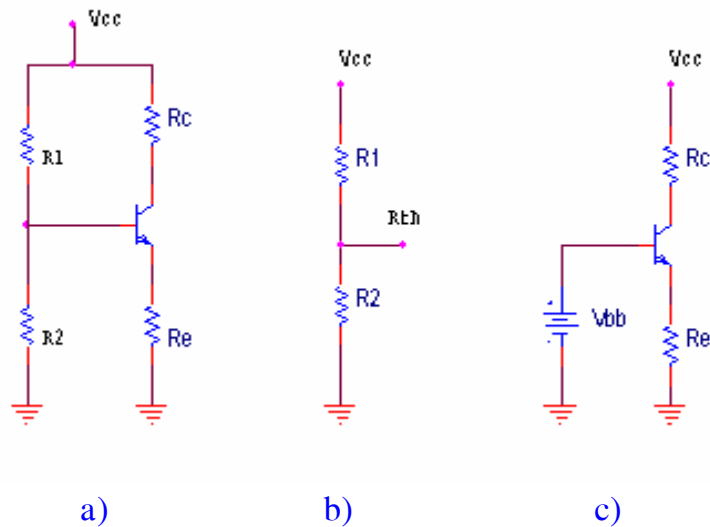
Hình 3-21

Khi xảy ra zero crossing, dòng qua LED bằng 0 và vì vậy phototransistor ngưng dẫn. Tại collector chúng ta thu được thế cao bằng nguồn cung cấp.

Chương IV  
**PHÂN CỰC TRANSISTOR**

**IV.1 PHÂN CỰC BẰNG CẦU CHIA THỂ**

Hình 4-1a là mạch phân cực transistor được dùng rất phổ biến trong các ứng dụng: phân cực bằng cầu chia thể. Mạch phân cực là cầu chia thể gồm 2 điện trở  $R_1$  và  $R_2$ . Vì vậy mạch gọi là mạch phân cực bằng cầu chia thể. Tiếng Anh là Voltage Divider Bias (VDB).



Hình 4-1: Phân cực VDB

**Phân tích đơn giản (simplified analysis).** Trong mạch VDB được thiết kế tốt dòng base nhỏ hơn dòng chảy qua cầu chia thể. Do đó dòng base ảnh hưởng không đáng kể lên cầu chia thể. Vì vậy có thể ngắt mạch base khỏi cầu chia thể để có mạch tương đương như hình 4-1b. Trong mạch này, thế ra của cầu chia thể bằng

$$V_{TH} = (R_2 V_{CC}) / (R_1 + R_2)$$

Lý tưởng thì  $V_{TH}$  chính là nguồn nuôi mạch base  $V_{BB}$  như hình 4-1c.

Phân cực bằng cầu chia thể thực chất là phân cực emitter. Nói cách khác hình 4-1c là mạch tương đương của hình 4-1a. Vậy mạch phân cực VDB tạo ra dòng emitter cố định. Do đó mạch phân cực VDB có điểm Q không phụ thuộc  $\beta_{dc}$ .



Có một sai số nhỏ trong cách tiếp cận đơn giản này. Chúng ta sẽ phân tích nó trong các phần sau. Điểm mấu chốt (crucial) ở đây là: trong một mạch VDB được thiết kế tốt, sai số khi sử dụng mạch tương đương hình 4-1c là rất bé. Nói cách khác, các nhà thiết kế chọn lựa có cân nhắc các giá trị linh kiện sao cho mạch hình 4-1a giống như mạch hình 4-1c.

**Kết luận:** Trong mạch phân cực VDB, sau khi tính  $V_{BB}$ , phần tính toán còn lại giống hệt như các bước tính cho mạch phân cực emitter ở chương 3. Ở đây chúng ta liệt kê ra các phương trình có thể dùng khi phân tích mạch VDB.

$$V_{BB} = (R_2 / (R_1 + R_2)) V_{CC} \quad (4-1)$$

$$V_E = V_{BB} - V_{BE} \quad (4-2)$$

$$I_E = V_E / R_E \quad (4-3)$$

$$I_C \approx I_E \quad (4-4)$$

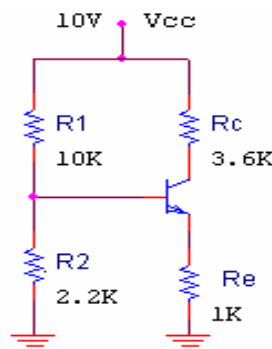
$$V_C = V_{CC} - I_C R_C \quad (4-5)$$

$$V_{CE} = V_C - V_E \quad (4-6)$$

Các bước phải làm là:

- Tính thế  $V_{BB}$  của cầu chia thế
- Trừ 0.7V để có thế emitter
- Chia cho điện trở emitter để có dòng emitter
- Giả sử dòng collector xấp xỉ dòng emitter
- Tính thế collector so với đất bằng cách trừ  $V_{CC}$  cho sụt thế trên  $R_C$
- Tính thế  $V_{CE}$  bằng cách trừ  $V_C$  cho  $V_E$

Ví dụ: Tính  $V_{CE}$  cho mạch hình 4-2.



Hình 4-2

Cầu chia thế tạo ra thế không tải bằng

$$V_{BB} = (2.2K\Omega / (10K\Omega + 2.2K\Omega)) 10V = 1.8V$$

$$V_E = 1.8V - 0.7V = 1.1V$$

Dòng emitter bằng

$$I_E = 1.1V / 1K\Omega = 1.1mA$$

Vì dòng collector xấp xỉ dòng emitter nên thế collector so với đất bằng

$$V_C = 10V - (1.1mA)(3.6K\Omega) = 6.04V$$

Hiệu thế giữa collector và emitter bằng

$$V_{CE} = 6.04V - 1.1V = 4.94V$$

## IV.2 PHÂN TÍCH CHÍNH XÁC MẠCH VDB

Chúng ta bắt đầu phần này bằng định nghĩa mạch VDB được thiết kế tốt. Đó là mạch mà trong đó cầu chia thế tỏ ra mạnh (stiff) đối với trở kháng vào của mạch base.

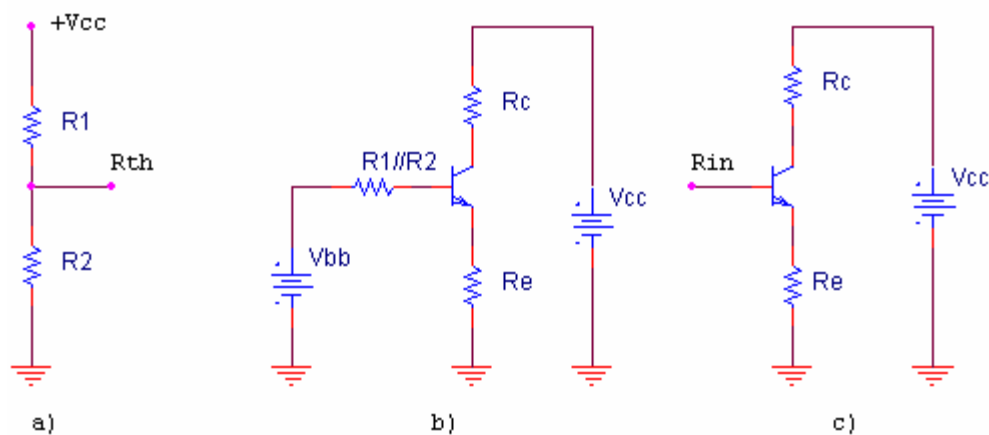
**Điện trở nguồn.** Trong chương 1 chúng ta đã nói đến nguồn thế mạnh như sau:

Nguồn thế mạnh có  $R_S < 0.01R_L$ . Nếu thỏa điều kiện này thì thế trên tải chỉ khác giá trị lý tưởng 1%.

Bây giờ chúng ta mở rộng ý tưởng này cho cầu chia thế.

Trước hết tính trở Thevenin của cầu chia thế trên hình 4-3a. Nó bằng

$$R_{TH} = R_1 // R_2$$



Hình 4-3

Do có  $R_{TH}$  nên các phân tích chính xác hơn cho mạch VDB phải tính đến  $R_{TH}$  như hình 4-3b. Dòng chảy qua điện trở  $R_{TH}$  làm giảm thế  $V_{BB}$ .

Vậy thế base giảm bao nhiêu so với giá trị lý tưởng? Theo hình 4-3c, nhìn từ cực base chúng ta thấy một điện trở vào  $R_{IN}$ . Để cầu chia thế mạnh đối với base thì theo quy tắc 100:1 chúng ta có

$$R_1 // R_2 < 0.01R_{IN} \quad (4-7)$$

Một mạch VDB được gọi là thiết kế tốt nếu thỏa mãn (4-7)

### Cầu chia thế mạnh

Nếu transistor trên mạch hình 4-3c có  $\beta_{dc} = 100$  thì dòng collector lớn hơn dòng base 100 lần hay dòng emitter cũng lớn hơn dòng base 100 lần. Nhìn từ phía base, trở emitter 100 lần lớn hơn. Tổng quát, khi nhìn từ base điện trở emitter lớn hơn  $\beta_{dc}$  lần.

$$R_{IN} = \beta_{dc} R_E \quad (4-8)$$

Phương trình (4-7) có thể viết lại như sau

$$R1//R2 < 0.01 \beta_{dc} R_E \quad (4-9)$$

Khi có thể, các nhà thiết kế chọn các giá trị của mạch theo qui tắc 100:1 bởi vì nó tạo ra mạch có điểm Q siêu ổn định.

### Cầu chia thế yếu (firm)

Đôi khi thiết kế cầu chia thế mạnh tạo ra kết quả là các điện trở R1 và R2 quá bé do đó gây ra một số hệ quả khác. Trong trường hợp này các nhà thiết kế phải thỏa hiệp bằng quy tắc cầu chia thế yếu như sau:

Cầu chia thế yếu nếu thỏa điều kiện

$$R1//R2 < 0.1 \beta_{dc} R_E \quad (4-10)$$

Mọi cầu chia thế thỏa mãn qui tắc 10:1 là cầu chia thế yếu. Trong trường hợp xấu nhất, dùng cầu chia thế yếu có thể làm dòng collector giảm 10% so với giá trị cầu chia thế mạnh. Điều này là chấp nhận được trong nhiều ứng dụng bởi vì mạch VDB vẫn có điểm Q đủ ổn định.

### **Xấp xỉ gần hơn**

Để tính chính xác giá trị của dòng  $I_E$  có thể dùng phương trình sau:

$$I_E = (V_{BB} - V_{BE}) / (R_E + (R1//R2)/\beta_{dc}) \quad (4-11)$$

Phương trình (4-11) cho giá trị  $I_E$  chính xác hơn. Nhưng (4-11) khá phức tạp. Vì vậy khi có yêu cầu tính toán chính xác mạch VDB người ta dùng EWB.

Ví dụ: Cầu chia thế trên hình 4-4 có là cầu chia thế mạnh? Tính giá trị chính xác của  $I_E$  dùng (4-11).

Cầu chia thế mạnh nếu thỏa điều kiện

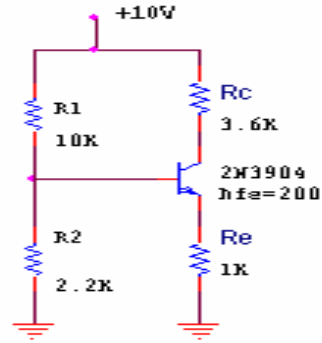
$$R1//R2 < 0.01 \beta_{dc} R_E$$

Trên hình 4-4, điện trở Thevenin của cầu chia thế bằng

$$R1//R2 = 10K\Omega // 2.2K\Omega = 1.8K\Omega$$

Điện trở vào của base bằng

$$R_{IN} = \beta_{dc} R_E = 200(1K\Omega) = 200K\Omega$$



Hình 4-4

Điện trở Thevenin của cầu là  $1.8K\Omega$  nhỏ hơn  $2K\Omega$  (1% của  $R_{IN}$ ) do đó cầu chia thế là mạnh.

Theo (4-11), dòng  $I_E$  bằng

$$I_E = (1.8V - 0.7V) / (1K\Omega + 1.8K\Omega / 200) = 1.09mA$$

Giá trị này gần với  $1.1mA$  là giá trị tính bằng phân tích đơn giản.

Vậy khi cầu chia thế là mạnh thì chúng ta không dùng công thức (4-11) để tính  $I_E$ . Từ nay về sau các tính toán của chúng ta đối với mạch VDB là theo phân tích đơn giản.

### IV.3 ĐƯỜNG TẢI VDB VÀ ĐIỂM Q

Trên hình 4-5, vì cầu chia thế là mạnh nên thế emitter giữ giá trị  $1.1V$  trong các phân tích sau đây.

#### Điểm Q

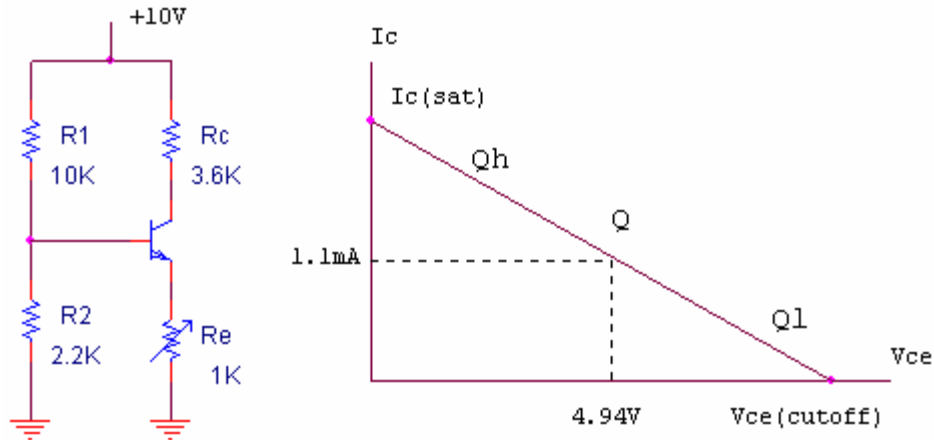
Điểm Q đã được tính trong bài IV-1. Nó có dòng  $I_C = 1.1mA$  và  $V_{CE} = 4.94V$ . Điểm Q như thế được vẽ trên hình 4-5. Do phân cực VDB là biến thế của phân cực emitter nên điểm Q không phụ thuộc  $\beta_{dc}$ . Để thay đổi điểm Q trên hình 4-5 có thể thay đổi  $R_E$ .

Ví dụ, nếu  $R_E$  thay đổi đến  $2.2K\Omega$  thì dòng  $I_E$  và thế  $V_C$  thay đổi như sau

$$I_E = 1.1V / 2.2K\Omega = 0.5mA$$

$$V_C = 10V - (0.5mA)(3.6K\Omega) = 8.2V$$

$$V_{CE} = 8.2V - 1.1V = 7.1V$$



Hình 4-5

Điểm Q mới là  $Q_L$  (0.5mA, 7.1V)

Ngược lại nếu giảm  $R_E$  còn 510Ω thì dòng emitter và thế collector mới là:

$$I_E = 1.1V / 510\Omega = 2.15mA$$

$$V_C = 10V - (2.15mA)(3.6K\Omega) = 2.26V$$

$$V_{CE} = 2.26V - 1.1V = 1.16V$$

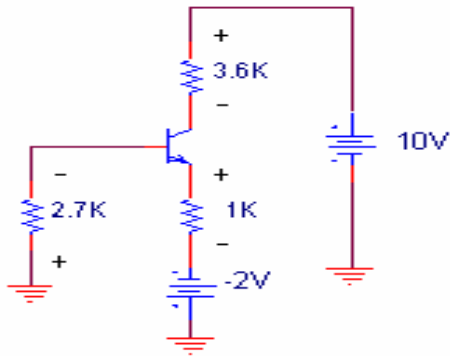
Trong trường hợp này điểm Q mới là  $Q_H$  (2.15mA, 1.16V)

#### **Điểm Q ở giữa đường tải.**

Các giá trị  $V_{CC}$ ,  $R_1$ ,  $R_2$ ,  $R_C$  đều có ảnh hưởng đến điểm Q. Trong thiết kế thường người ta chọn các giá trị này theo yêu cầu cho trước. Khi đó nếu  $R_E$  thay đổi sẽ làm cho Q dịch chuyển trên đường tải. Nếu  $R_E$  quá lớn điểm Q sẽ dịch đến vùng ngưng dẫn. Nếu  $R_E$  quá bé điểm Q sẽ dịch đến vùng bão hòa. Để đạt được sự ổn định, điểm Q thường được chọn ở giữa đường tải cho mạch khuếch đại.

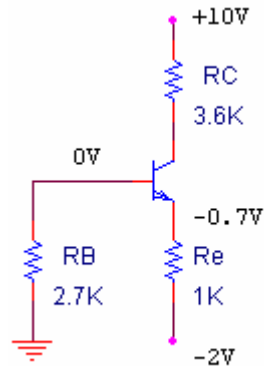
### **IV.4 PHÂN CỰC EMITTER DÙNG 2 NGUỒN NUÔI**

Một số thiết bị điện tử có bộ nguồn 2 cực tính: nguồn dương và nguồn âm. Ví dụ, mạch hình 4-6 chỉ ra một transistor có 2 nguồn nuôi: +10V và -2V. Nguồn âm -2V phân cực thuận cho diode emitter. Nguồn dương +10V phân cực ngược cho diode collector. Mạch phân cực này là mạch phân cực emitter. Do đó nó có tên là mạch phân cực emitter 2 nguồn nuôi (Two Supply Emitter Bias = TSEB).



**Hình 4-6**

Trong các mạch TSEB thiết kế đúng, dòng base rất bé vì vậy thế base xấp xỉ 0V như hình 4-7.



**Hình 4-7**

Hiệu thế giữa 2 đầu  $R_E$  là

$$R_E = -0.7V - (-2V) = +1.3V$$

Dòng  $I_E$  bằng

$$I_E = 1.3V / 1K\Omega = 1.3mA$$

Dòng  $I_C$  xấp xỉ  $I_E$  chảy qua  $R_C$  do đó thế tại cực C của transistor bằng

$$V_C = 10V - (1.3mA)(3.6K\Omega) = 5.32V$$

$$\text{Thế } V_{CE} = V_C - V_E = 5.32V - (-0.7V) = 6.02V$$

Tóm lại, khi một mạch TSEB thiết kế tốt, nó tương đương với mạch phân cực VDB và nếu thỏa mãn qui tắc 100:1 tức là

$$R_B < 0.01\beta_{dc}R_E \quad (4-12)$$

Khi đó có thể dùng các phương trình sau để phân tích mạch TSEB

$$V_B = 0V \quad (4-13)$$

$$I_E = (V_{EE} - 0.7V)/R_E \quad (4-14)$$

$$V_C = V_{CC} - I_C R_C \quad (4-15)$$

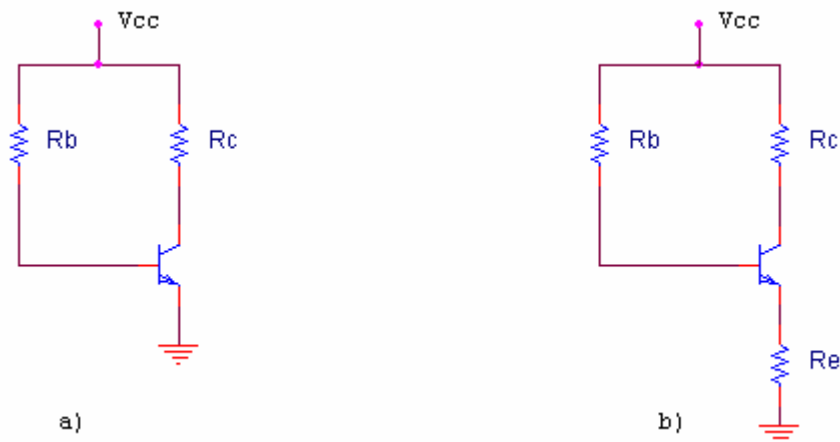
$$V_{CE} = V_C + 0.7V \quad (4-16)$$

## IV.5 CÁC PHÂN CỰC KHÁC

Trong phần này chúng ta xem xét một số kiểu phân cực khác. Việc phân tích chi tiết các kiểu phân cực này là không cần thiết vì chúng ít được dùng trong các thiết kế mới.

### a) Phân cực phản hồi emitter

Nhược điểm chính của phân cực base trên hình 4-8a là  $Q$  thay đổi theo  $\beta_{dc}$ . Để khắc phục người ta dùng mạch phân cực phản hồi emitter như hình 4-8b. Ý tưởng chính là đưa vào điện trở emitter để chống lại sự trôi của  $Q$  khi  $\beta_{dc}$  thay đổi. Cụ thể là khi  $I_C$  tăng làm cho  $V_E$  tăng  $\rightarrow V_B$  tăng  $\rightarrow I_B$  giảm  $\rightarrow I_C$  giảm. Sự phản hồi này là phản hồi âm (negative feedback). Gọi là phản hồi vì sự thay đổi thế emitter được phản hồi trở lại thế base. Mặc khác, phản hồi là âm vì sau quá trình phản hồi, dòng  $I_C$  thay đổi ngược lại với sự thay đổi ban đầu. Vậy phản hồi âm tạo ra sự ổn định.



Hình 4-8: Phân cực phản hồi emitter

Sau đây là các phương trình dùng cho mạch phân cực phản hồi emitter.

$$I_E = (V_{CC} - V_{BE}) / (R_E + R_B / \beta_{dc}) \quad (4-17)$$

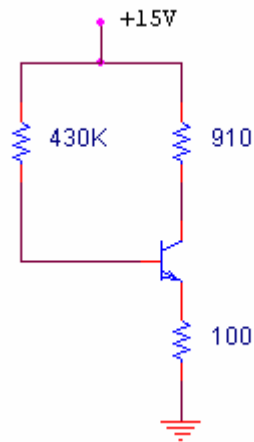
$$V_E = R_E I_E \quad (4-18)$$

$$V_B = V_E + 0.7V \quad (4-19)$$

$$V_C = V_{CC} - R_C I_C \quad (4-20)$$

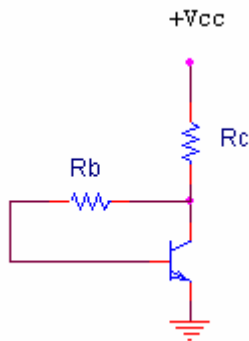
Ý định của phân cực phản hồi emitter là loại bỏ  $\beta_{dc}$  khỏi phương trình (4-17) bằng cách chọn  $R_E > R_B / \beta_{dc}$ . Nếu điều kiện này thỏa thì (4-17) không phụ thuộc  $\beta_{dc}$ .

Hình 4-9 là một mạch phân cực phản hồi emitter cụ thể. Trên thực tế phân cực này không tốt hơn phân cực base.



Hình 4-9

### b) Phân cực phản hồi collector



Hình 4-10: Phân cực phản hồi collector

Hình 4-10 chỉ ra mạch phân cực phản hồi collector, còn gọi là mạch tự phân cực. Ý tưởng cơ bản là cố ổn định điểm Q bằng cách phản hồi một điện thế về base để bù lại sự thay đổi của dòng collector. Ví dụ  $I_C$  tăng  $\rightarrow V_C$  giảm  $\rightarrow V_B$  giảm  $\rightarrow I_B$  giảm  $\rightarrow I_C$  giảm.



Sau đây là các phương trình dùng cho mạch phân cực phản hồi collector

$$I_E = (V_{CC} - V_{BE}) / (R_C + R_B / \beta_{dc}) \quad (4-21)$$

$$V_B = 0.7V \quad (4-22)$$

$$V_C = V_{CC} - I_C R_C \quad (4-23)$$

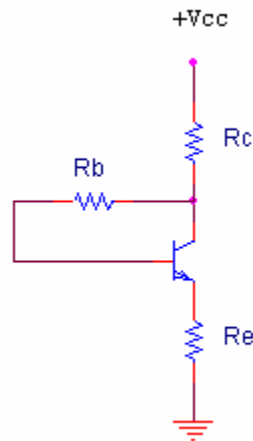
Điểm Q nằm giữa đường tải nếu chọn

$$R_B = \beta_{dc} R_C \quad (4-24)$$

Mạch phân cực phản hồi collector có tác dụng ổn định điểm Q tốt hơn phân cực phản hồi emitter. Mặc dù vẫn có sự trôi của Q theo  $\beta_{dc}$  nhưng mạch phân cực này thường được dùng vì sự đơn giản.

### c) Phân cực phản hồi collector và emitter

Phân cực phản hồi emitter và phản hồi collector là nhằm ổn định điểm Q. Mặc dù ý tưởng là tốt nhưng sự phản hồi lại không đủ để ổn định Q theo  $\beta_{dc}$ . Hình 4-11 là một mạch phân cực trong đó sử dụng cả 2 loại phản hồi: phản hồi emitter và phản hồi collector nhằm cải thiện độ ổn định của Q.



Hình 4-11

Sau đây là các phương trình dùng cho mạch phân cực phản hồi emitter - collector.

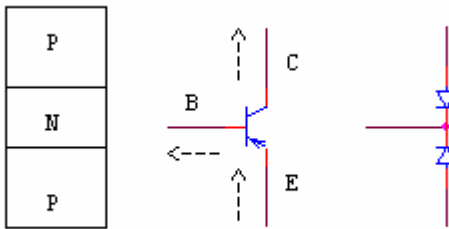
$$I_E = (V_{CC} - V_{BE}) / (R_C + R_B / \beta_{dc}) \quad (4-25)$$

$$V_E = R_E I_E \quad (4-26)$$

$$V_B = V_E + 0.7V \quad (4-27)$$

$$V_C = V_{CC} - R_C I_C \quad (4-28)$$

## IV.6 TRANSISTOR PNP

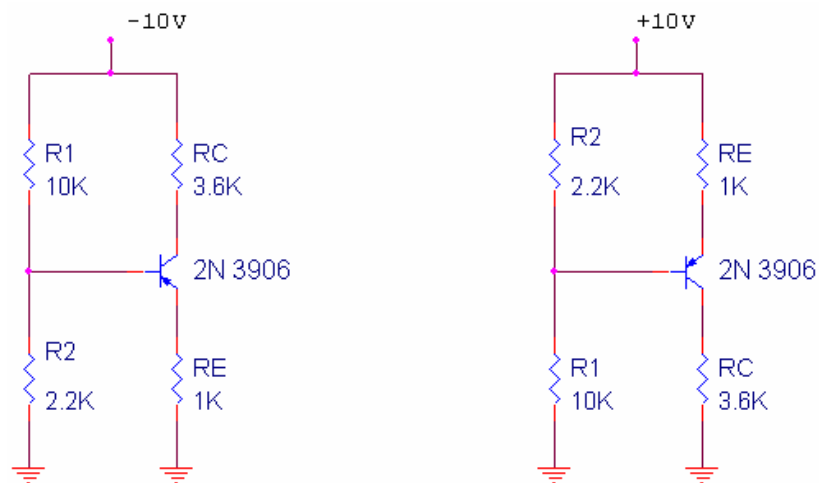


**Hình 4-12: Transistor PNP**

Hình 4-12 chỉ ra cấu trúc , ký hiệu và mô hình tương đương của transistor PNP. Cũng giống như transistor NPN, tổng dòng trong transistor tuân theo định luật Kirchhoff.

Mạch cho transistor PNP (xem hình 4-13) hoàn toàn giống mạch cho transistor NPN. Chỉ có 2 điều khác biệt:

- Transistor là loại PNP
- Nguồn âm



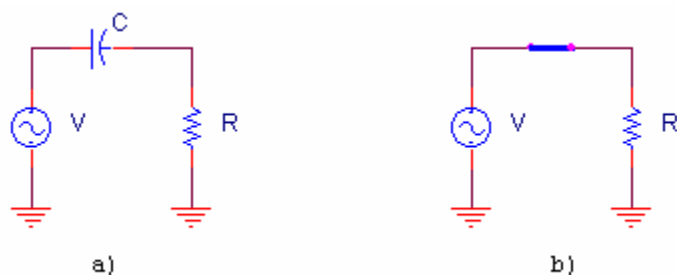
**Hình 4-13: Mạch transistor PNP**

## Chương V **CÁC CHẾ ĐỘ AC CỦA TRANSISTOR**

### **V.1 KHUYẾCH ĐẠI PHÂN CỰC BASE**

Trong phần này chúng ta sẽ phân tích mạch khuếch đại phân cực base. Mặc dù mạch khuếch đại base không phổ biến trong các sản phẩm điện tử nhưng các ý tưởng cơ bản của nó được dùng nhiều trong các mạch khuếch đại phức hợp khác.

#### TU NỐI



Hình 5-1

Hình 5-1a là một nguồn thế ac được nối tới đầu vào của một mạch RC. Cấu hình này là mạch tương đương ở đầu vào của nhiều mạch khuếch đại. Vì trở kháng của tụ tỷ lệ nghịch với tần số nên tụ C ngăn cản hoàn toàn thành phần dc của nguồn và chỉ truyền thành phần ac. Ở tần số đủ cao thì trở kháng của tụ bé do đó hầu hết thành phần ac của nguồn xuất hiện trên R. Trong trường hợp này tụ C gọi là tụ nối vì nó nối hoặc chuyển tín hiệu ac đến R. Vai trò của tụ C là cho phép truyền thành phần ac từ nguồn đến mạch khuếch đại mà không làm thay đổi điểm Q của mạch.

Để cho một tụ C có thể xem là tụ nối thì tại tần số tín hiệu thấp nhất, trở kháng của tụ phải bé hơn nhiều so với R. Người ta định nghĩa

Tụ nối tốt nếu thỏa điều kiện:

$$X_C < 0.1R \quad (5-1)$$

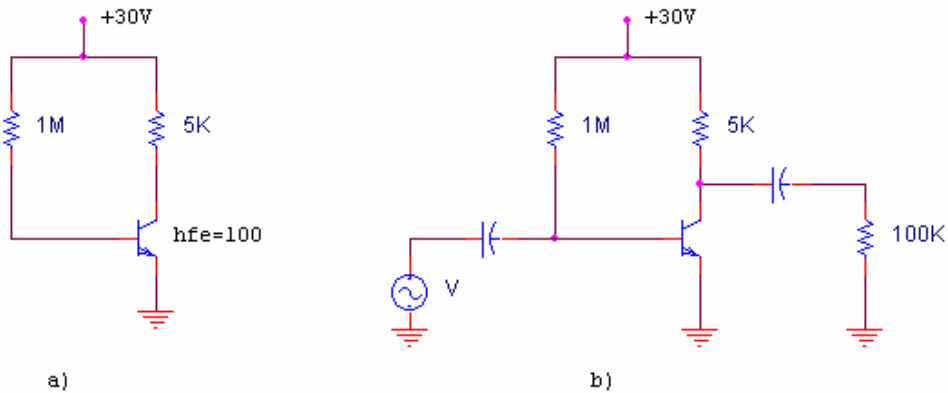
Nghĩa là trở kháng của tụ C phải bé hơn 10 lần R tại tần số hoạt động thấp nhất.

Khi thỏa mãn qui tắc 10:1, hình 5-1a có thể thay bằng hình 5-1b. Nói cách khác đối với các mạch thỏa qui tắc 10:1 có thể thay thế tất cả các tụ nối C bằng một ngắn mạch đối với thành phần ac.

Mặt khác nguồn dc có tần số bằng 0 nên trở kháng của C đối với thành phần dc là vô cùng. Bởi vậy, chúng ta sẽ dùng gần đúng sau cho tụ C:

- ◆ Đối với thành phần dc tụ C xem như hở mạch
- ◆ Đối với thành phần ac tụ C xem như nối tắt

### CHẾ ĐỘ DC



Hình 5-2

Chúng ta hãy xét mạch phân cực base trên hình 5-2a. Thế dc tại base là 0.7V, trong gần đúng bậc 1, dòng  $I_B$  bằng:

$$I_B = 30\mu\text{A}$$

Với hệ số khuếch đại dòng 100 thì dòng  $I_C$  bằng

$$I_C = 3\text{mA}$$

Thế collector bằng

$$V_C = 30\text{V} - (3\text{mA})(5\text{K}\Omega) = 15\text{V}$$

Điểm Q có tọa độ 3mA và 15V.

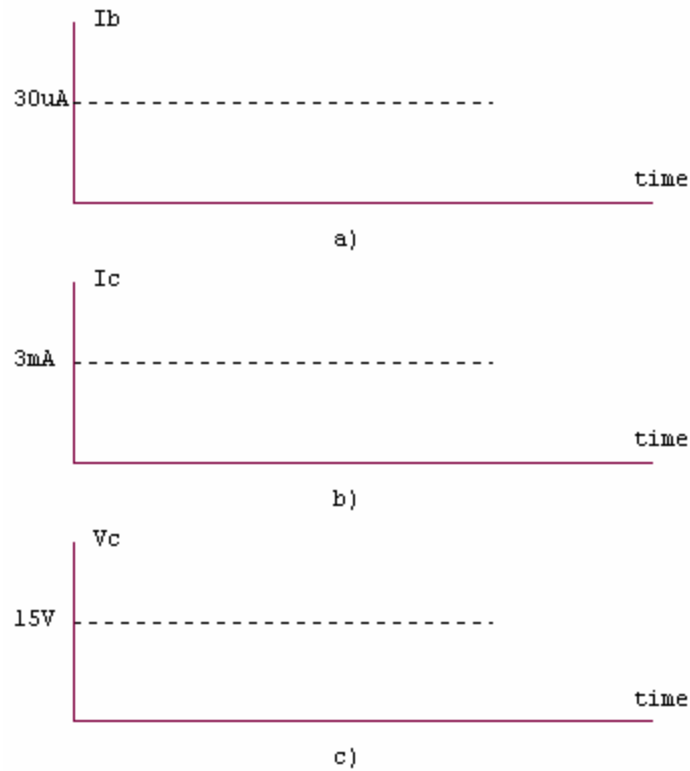
### MẠCH KHUYẾT ĐẠI

Hình 5-2b cho thấy một mạch khuếch đại transistor phân cực base. Các tụ nối được dùng ở đầu vào và đầu ra nhằm cách ly thành phần dc của mạch với nguồn ac vào và với tải  $R_L$ . Mục đích chính là không cho nguồn ac và trở tải  $R_L$  thay đổi điểm Q.

Trên hình 5-2b, nguồn thế ac có giá trị  $100\mu\text{V}$ . Do tụ nối ngắn mạch đối với thành phần ac nên toàn bộ thành phần ac của nguồn thế vào xuất hiện tại base. Thế ac này sẽ tạo ra dòng base xoay chiều mà nó cộng thêm vào với dòng base một chiều do phân cực. Nói cách khác, dòng base tổng cộng bao gồm dòng dc và ac.

Hình 5-3a mô tả tình hình này. Thành phần ac được cộng với thành phần dc. Trong nửa chu kỳ dương, dòng base ac cộng với dòng  $30\mu\text{A}$  của dòng base dc. Trong nửa chu kỳ âm dòng base bị trừ đi. Sự thay đổi của dòng base làm cho dòng  $I_C$  cũng thay đổi theo cùng qui luật nhưng lớn hơn  $\beta_{dc}$  lần. Hình 5-3b cho thấy thành phần dc của dòng collector là  $3\text{mA}$ . Sự chông chắt của dòng collector dc và ac tạo ra tín hiệu như hình 5-3c.

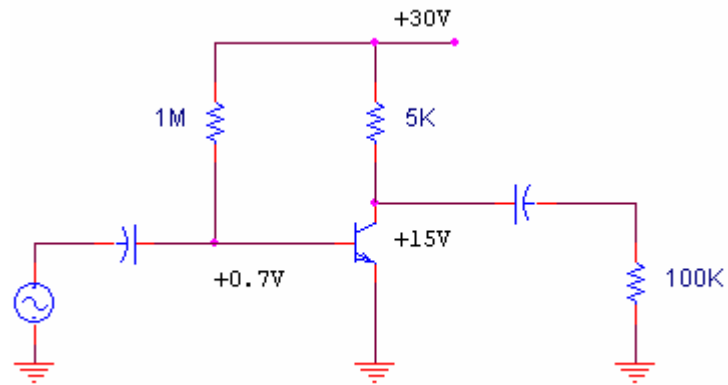
Do điện trở tải  $R_C$ , thế tại collector của transistor có dạng như nguồn thế đầu vào nhưng ngược pha như hình 5-3c.



Hình 5-3

### DẠNG SÓNG

Hình 5-4 là mạch khuếch đại phân cực base và dạng sóng (wave form) của nó. Nguồn thế ac là một hiệu thế điều hoà có biên độ bé. Nó được nối tới base, tại đây nó chông chắt với thành phần dc  $0.7\text{V}$ . Sự biến đổi của thế base tạo ra sự biến đổi dòng base, dòng collector và vì vậy cả thế collector. Kết quả thế collector là một thế dạng sin có thành phần dc là  $15\text{V}$ . Do tác dụng của tụ nối, trên tải  $R_L$  chỉ có thành phần ac.



Hình 5-4

### HỆ SỐ KHUYẾT ĐẠI THỂ

Hệ số khuyết đại thể của một mạch khuyết đại là tỷ số thế ac lối ra chia cho thế ac lối vào.

$$A = V_{\text{out}} / V_{\text{in}} \quad (5-2)$$

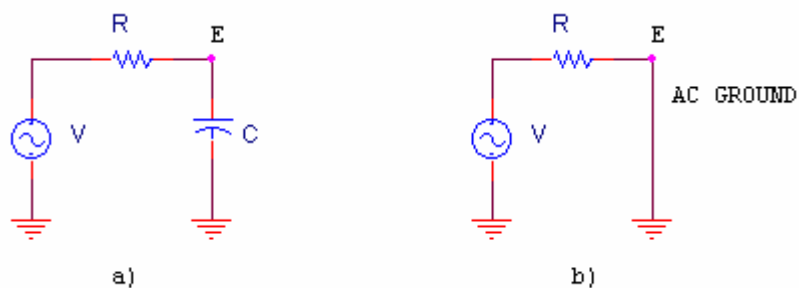
Ví dụ, nếu thế trên tải là 50mV trong khi thế vào là 100μV thì

$$A = 50\text{mV} / 100\mu\text{V} = 500$$

$A=500$  có nghĩa là thế ra lớn hơn thế vào 500 lần.

## **V.2 KHUYẾT ĐẠI PHÂN CỰC EMITTER**

### TỤ THÔNG DẪN



Hình 5-5

Tụ thông dẫn tương tự tụ nối bởi vì nó hành xử như là hở mạch đối với thành phần dc và ngắn mạch đối với thành phần ac. Nhưng tụ thông dẫn không dùng để nối tín hiệu giữa 2 điểm mà nó dùng để tạo ra một đất ac.

Hình 5-5a cho thấy một nguồn thế ac được nối tới một điện trở và một tụ. Ở tần số cao, trở kháng của tụ rất bé vì vậy toàn bộ nguồn thế ac coi như

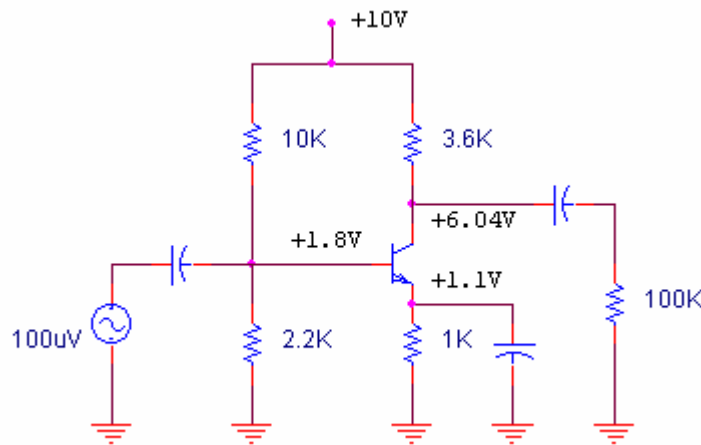
được đặt trên R. Nói cách khác, điểm E là được nối đất về mặt ac. Trong trường hợp này tụ C gọi là tụ thông dẫn vì nó cho phép nối điểm E với đất về mặt ac. Tụ thông dẫn cho phép chúng ta nối một điểm nào đó với GND mà không làm thay đổi Q.

Điều kiện để một tụ được xem là thông dẫn tốt là

$$X_C < 0.1R \quad (5-3)$$

Khi (5-3) thỏa thì mạch hình 5-5a có thể thay bằng mạch hình 5-5b.

### KHUYẾCH ĐẠI VDB



Hình 5-6

Hình 5-6 cho thấy một mạch khuếch đại VDB. Để tính thế và dòng DC, chúng ta tưởng tượng rằng tất cả các tụ đều hở mạch. Khi đó mạch tương đương như mạch phân cực VDB. Giá trị dc hay giá trị tĩnh cho mạch này như sau:

$$V_B = 1.8V$$

$$V_E = 1.1V$$

$$V_C = 6.04V$$

$$I_C = 1.1mA$$

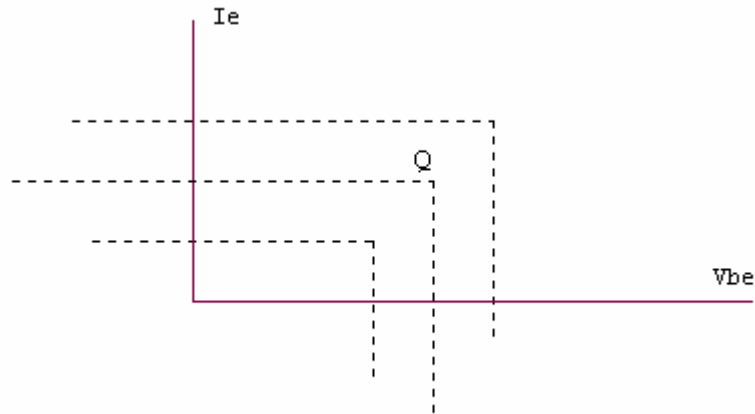
Như đã phân tích trên đây, chúng ta dùng các tụ nối để nối thế nguồn với base, nối thế collector với tải. Chúng ta cũng dùng tụ thông dẫn giữa emitter và đất. Ý nghĩa của tụ này là ở chỗ: Nó làm tăng hệ số khuếch đại thế của transistor so với khi không có tụ (các phần sau sẽ phân tích rõ hơn).

Trong hình 5-6, nguồn thế ac là  $100\mu V$ . Nó được đưa vào base. Do tụ thông dẫn  $C_E$  toàn bộ thế này được đặt lên diode base emitter. Tại collector chúng ta thu được thế base nhưng đã được khuếch đại.

### DẠNG SỐNG VDB

Trên hình 5-6, nguồn thế ac đầu vào là điều hoà và thành phần dc bằng 0. Thế base là chồng chất của thế ac lối vào và thành phần dc 1.8V. Thế ac ở collector là thế đầu vào đã được khuếch đại đảo pha chồng chất với thành phần dc bằng 6.04V. Thế trên tải giống như thế collector với thành phần dc bằng 0.

### **V.3 CHẾ ĐỘ TÍN HIỆU BÉ**



Hình 5-7

Hình 5-7 chỉ ra giản đồ dòng thế của diode base emitter. Khi nguồn thế ac được nối tới cực base của transistor, một thế ac sẽ xuất hiện trên diode base. Điều này tạo ra sự biến thiên tuần hoàn trên  $V_{BE}$  như hình 5-7.

### **ĐIỂM LÀM VIỆC TỨC THỜI.**

Khi thế vào thay đổi, điểm làm việc tức thời di chuyển từ vị trí Q ban đầu lên phía trên hoặc xuống phía dưới. Thế base emitter tổng cộng gồm thế ac và thành phần dc. Biên độ của thế ac quy định biên độ thay đổi của Q. Thế base ac lớn tạo ra sự biến động lớn của Q, thế base bé tạo ra sự biến động bé của Q.

### **SỰ SÁI DẠNG**

Thế base tạo ra dòng emitter cùng tần số như hình 5-7. Nếu nguồn thế base ac là điều hoà, dòng emitter cũng điều hoà. Nhưng dòng emitter không là



điều hoà lý tưởng vì quan hệ dòng emitter thế base không hoàn toàn là đường thẳng. Điều này tạo ra sự sai dạng (méo) tín hiệu mà chúng ta không mong muốn trong các hệ thống Hi-Fi (High Fidelity).

Để giảm thiểu sự sai dạng người ta dùng các cách sau:

- ◆ Giữ thế base bé. Khi đó quan hệ base- emitter của transistor là tuyến tính.
- ◆ Qui tắc 10:1.

Tổng dòng emitter bằng

$$I_E = I_{EQ} + i_c$$

Trong đó  $I_{EQ}$  là dòng emitter phân cực còn  $i_c$  là dòng emitter ac. Để giảm thiểu sự sai dạng, giá trị đỉnh của dòng  $i_c$  phải bé hơn dòng emitter tĩnh 10 lần theo qui tắc 10:1 sau đây.

$$\text{Tín hiệu bé: } i_{c\text{pp}} < 0.1 I_{EQ} \quad (5-4)$$

Các bộ khuếch đại thỏa điều kiện (5-4) là bộ khuếch đại tín hiệu bé. Chúng thường được dùng trong các mạch tiền khuếch đại của các mạch xử lý tín hiệu.

#### V.4 HỆ SỐ BETA AC VÀ ĐIỆN TRỞ AC CỦA DIODE EMITTER

Hệ số khuếch đại mà chúng ta sử dụng từ trước đến nay là hệ số khuếch đại dòng một chiều.

$$\beta_{dc} = I_C / I_B \quad (5-5)$$

$\beta_{dc}$  phụ thuộc vị trí điểm Q do sự cong của đặc tuyến  $I_C, I_B$ .

Người ta định nghĩa hệ số beta xoay chiều là

$$\beta = i_c / i_b \quad (5-6)$$

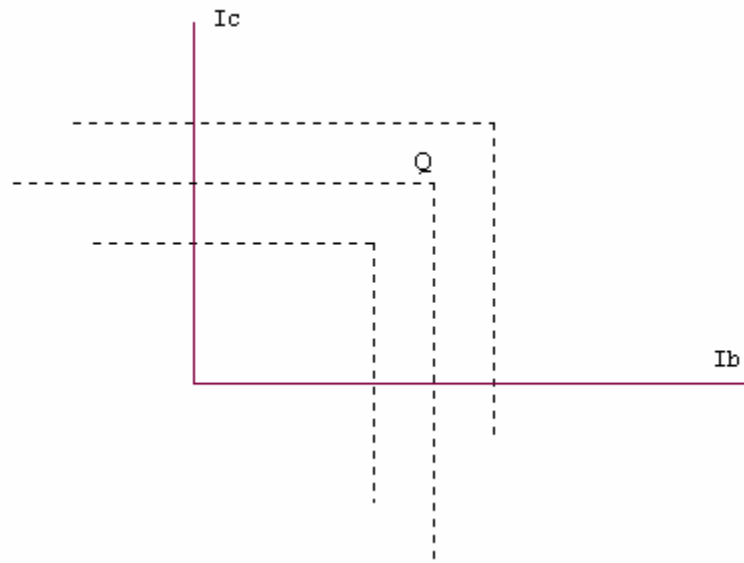
Theo (5-6), hệ số khuếch đại dòng ac bằng tỷ số giữa dòng collector ac chia cho dòng base ac.

Trên hình 5-8, tín hiệu ac thay đổi quanh điểm Q, do đó giá trị của  $\beta_{dc}$  và  $\beta$  có khác nhau.

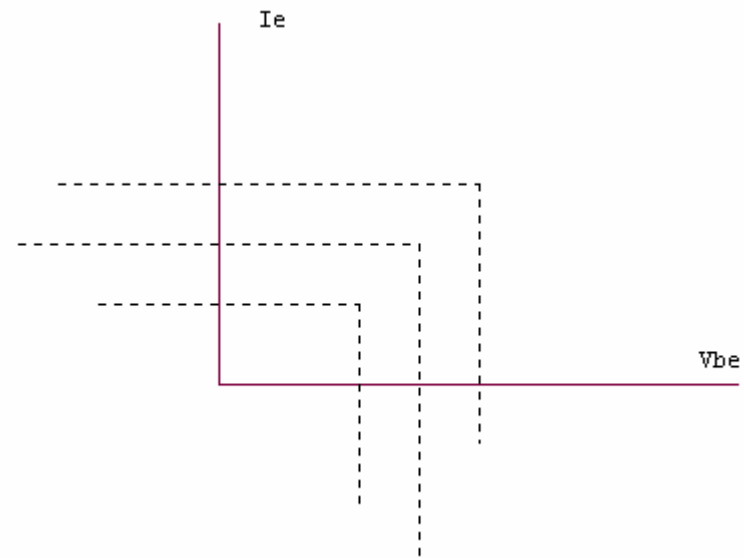
Về mặt đồ thị,  $\beta$  chính là độ dốc của đường cong  $I_C, I_B$  tại điểm Q.

Trong bảng số liệu của nhà sản xuất,  $\beta_{dc}$  được ký hiệu là  $h_{FE}$  còn  $\beta$  được ký hiệu là  $h_{fe}$ .

Chúng ta lưu ý rằng các ký hiệu về dòng và thế, nếu viết hoa là dc, còn viết thường là ac.



Hình 5-8



Hình 5-9

Hình 5-9 cho thấy quan hệ dòng thế của diode emitter.

Ta có:

$$I_E = I_{EQ} + i_c$$

$$V_{BE} = V_{BEQ} + v_{be}$$

## ĐIỆN TRỞ AC CỦA DIODE EMITTER

Trên hình 5-9 sự thay đổi điều hòa của  $V_{BE}$  tạo ra sự thay đổi điều hòa của  $I_E$ . Giá trị đỉnh – đỉnh của  $i_c$  phụ thuộc điểm Q. Do quan hệ dòng thế emitter là đường cong nên với  $v_{be}$  cố định, dòng  $i_c$  lớn hơn nếu điểm phân cực Q nằm ở phía cao hơn. Nói cách khác, điện trở ac của emitter giảm khi dòng emitter dc tăng.

Điện trở emitter ac được định nghĩa như sau:

$$r'_e = v_{be} / i_c \quad (5-7)$$

Theo vật lý chất rắn, điện trở ac của emitter có thể tính qua công thức đơn giản sau:

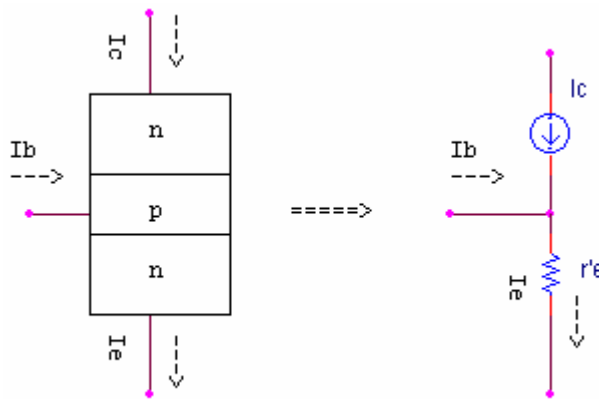
$$r'_e = 25\text{mV} / I_E \quad (5-8)$$

Sau này sẽ thấy, giá trị của  $r'_e$  có ảnh hưởng đến hệ số khuếch đại thế của mạch transistor.

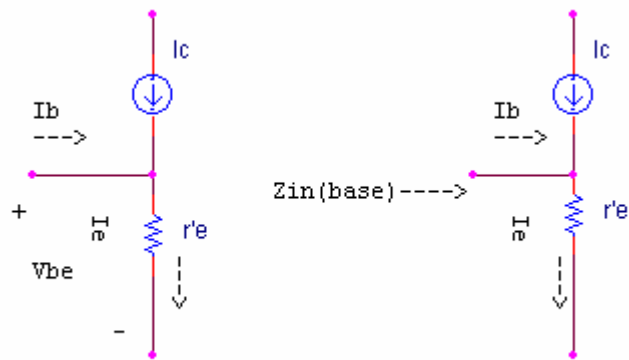
## V.5 HAI MÔ HÌNH CỦA TRANSISTOR

Để mô tả hoạt động của mạch khuếch đại transistor ở chế độ ac, cần một mạch tương đương cho transistor. Nói cách khác chúng ta cần một mô hình cho transistor mà nó mô phỏng được hoạt động của transistor khi có tín hiệu ac.

Một trong những mô hình transistor sớm nhất do Ebers Moll đề xuất là mô hình trên hình vẽ 5-10. Mô hình này còn gọi là mô hình chữ T.



Hình 5-10: Mô hình chữ T của transistor



Hình 5-11

Theo mô hình chữ T, khi một thế ac xuất hiện ở lối vào bộ khuếch đại thì một thế ac xuất hiện trên diode base emitter như hình 5-11. Thế này tạo ra dòng base ac. Nói cách khác nguồn thế ac vào đã được tải trên trở kháng vào của base. Nhìn vào base của transistor, ta thấy một trở kháng vào  $Z_{in(base)}$ . Tại tần số thấp, trở kháng này bằng:

$$Z_{in(base)} = v_{be} / i_b \quad (5-9)$$

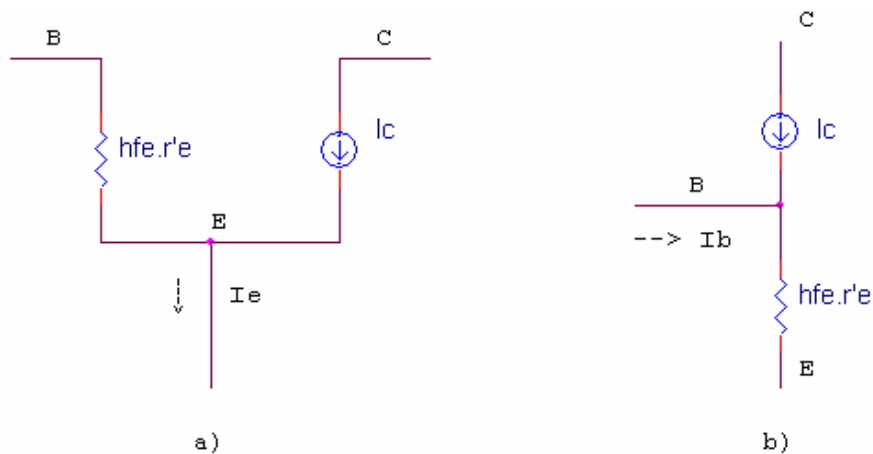
Áp dụng định luật Ohm cho diode emitter, có thể viết

$$v_{be} = i_e \cdot r'_e$$

$$Z_{in(base)} = i_e \cdot r'_e / i_b$$

Vì  $i_e \approx i_c$  nên

$$Z_{in(base)} = \beta r'_e \quad (5-10)$$



Hình 5-12: Mô hình chữ  $\pi$  của transistor

Một mô hình khác của transistor cũng được sử dụng, đó là mô hình chữ  $\pi$ . Hình 5-12a cho thấy sơ đồ mô hình  $\pi$  của transistor. So với mô hình chữ T (hình 5-12b) mô hình chữ  $\pi$  dễ dùng hơn. Chúng ta có thể dùng một trong hai mô hình, mô hình chữ T hoặc mô hình chữ  $\pi$ , khi phân tích mạch transistor ở chế độ ac.

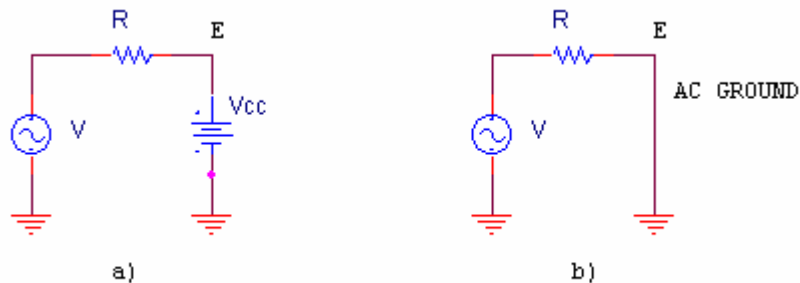
## V.6 PHÂN TÍCH MỘT KHUYẾCH ĐẠI DÙNG TRANSISTOR

Phân tích một mạch khuếch đại là phức tạp vì cả thành phần ac lẫn dc đều tồn tại trong mạch. Chúng ta có thể phân tích chế độ dc và phân tích chế độ ac một cách riêng rẽ. Hoạt động thực của mạch là chồng chất của 2 chế độ dc và ac.

### MẠCH TƯƠNG ĐƯƠNG DC

Trong phân tích dc chúng ta tính dòng và thế dc của mạch. Để làm việc này chúng ta tưởng tượng hở mạch đối với tất cả các tụ. Mạch còn lại là mạch tương đương dc. Trong phân tích dc, quan trọng nhất là tính  $I_E$  vì nó liên quan đến  $r'_e$  trong phân tích ac.

### ẢNH HƯỞNG CỦA NGUỒN DC



Hình 5-13

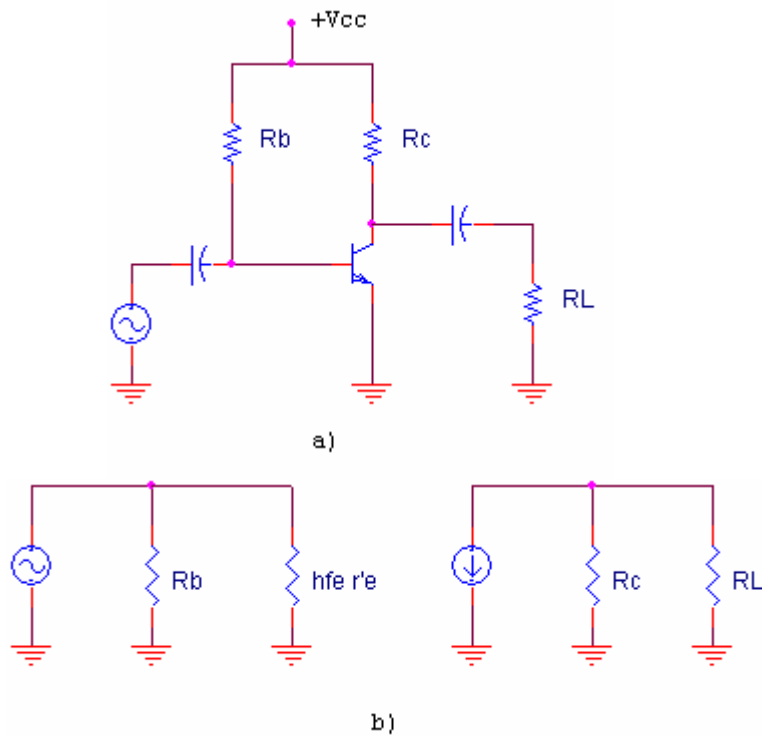
Hình 5-13a là sơ đồ mạch gồm có cả nguồn ac và dc. Đối với tín hiệu ac, nguồn dc là ngắn mạch. Điều này là đúng vì trở nội của nguồn dc bé do đó nó không gây sụt thế ac nào. Vậy khi phân tích ac chúng ta ngắn mạch tất cả các nguồn dc.

### MẠCH TƯƠNG ĐƯƠNG AC

Sau khi phân tích dc, chúng ta sẽ bắt đầu phân tích ac bằng cách ngắn mạch tất cả các tụ và nguồn dc. Transistor có thể thay thế bằng mô hình T hay

mô hình  $\pi$ . Sau đây chúng ta sẽ dẫn ra các mạch tương đương ac của các mạch khuếch đại phân cực base và phân cực VDB.

### a) MẠCH KHUYẾCH ĐẠI PHÂN CỰC BASE

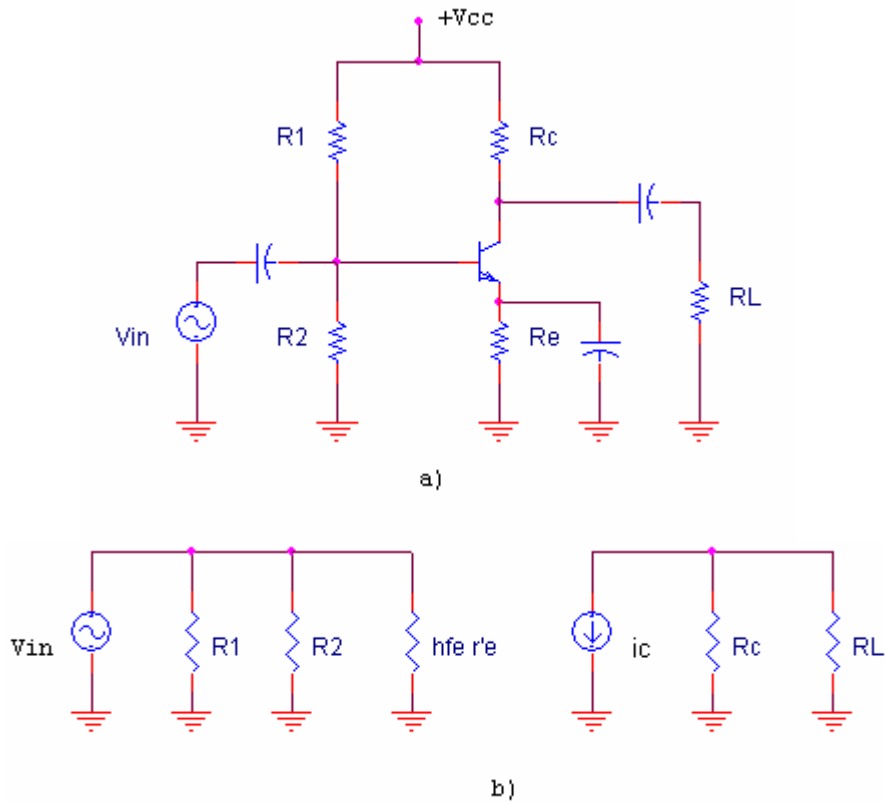


Hình 5-14

Hình 5-14a là một mạch khuếch đại phân cực base. Để có mạch tương đương ac, chúng ta ngắt mạch tất cả các tụ. Điểm có nhãn  $V_{cc}$  cũng được nối đất về mặt ac. Hình 5-14b cho thấy mạch tương đương ac. Transistor đã được thay thế bởi mô hình  $\pi$ . Trong mạch base, thế ac lối vào đặt trên  $R_B$  song song với  $\beta r'_e$ . Trong mạch collector, nguồn dòng  $i_c$  chảy qua  $R_C$  và  $R_L$  mắc song song.

### b) MẠCH KHUYẾCH ĐẠI VDB

Hình 5-15a là một mạch khuếch đại VDB. Hình 5-15b cho thấy mạch tương đương ac. Transistor đã được thay thế bởi mô hình  $\pi$ . Trong mạch base, thế ac lối vào đặt trên  $R_1$  song song với  $\beta r'_e$  và với  $R_2$ . Trong mạch collector, nguồn dòng  $i_c$  chảy qua  $R_C$  và  $R_L$  mắc song song.



**Hình 5-15**

### c) MẠCH KHUYẾCH ĐẠI CE

Các mạch khuếch đại trên hình 5-14, hình 5-15 là các ví dụ về mạch khuếch đại chung emitter (common emitter = CE). Mạch khuếch đại này gọi là mạch khuếch đại chung emitter vì emitter được nối GND về mặt ac.

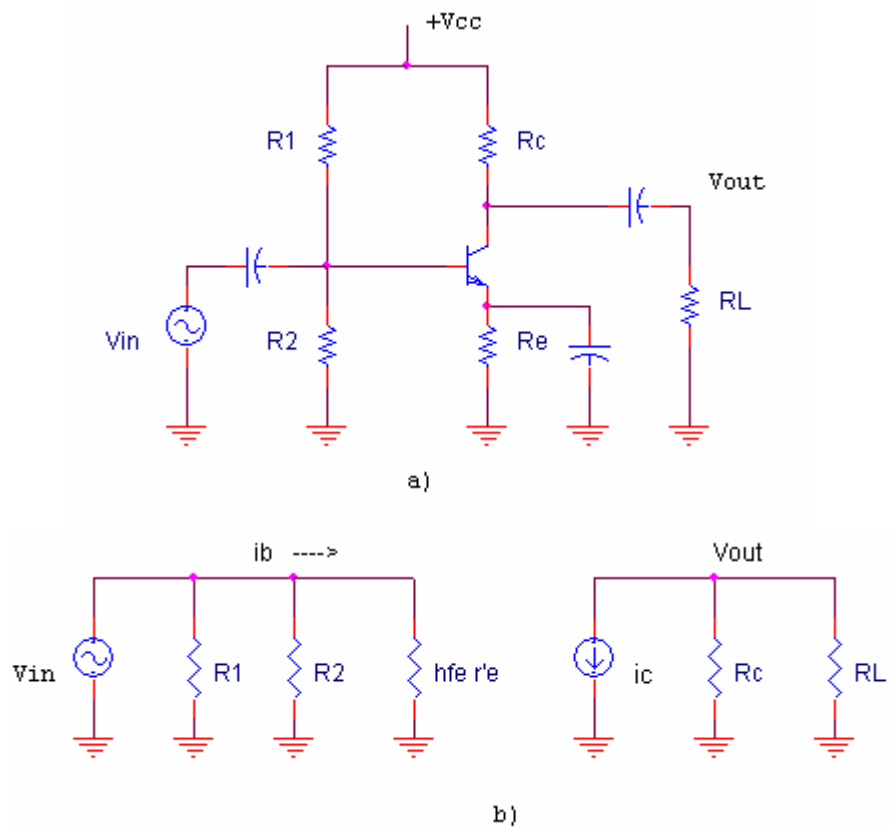
Trong mạch khuếch đại chung emitter tín hiệu cần khuếch đại được đưa đến cực base, tín hiệu đã được khuếch đại xuất hiện ở collector.

Có 2 kiểu mạch khuếch đại transistor khác là mạch khuếch đại chung base (CB) và mạch khuếch đại chung collector (CC). Chúng được dùng trong một số ứng dụng nhưng không phổ biến bằng mạch khuếch đại CE.

Tóm lại, để có được mạch tương đương về mặt ac, chúng ta phải làm mấy việc chủ yếu sau:

- ◆ Ngắt mạch tất cả các tụ
- ◆ Tưởng tượng tất cả các nguồn dc là GND xoay chiều
- ◆ Thay thế transistor bằng mô hình T hoặc mô hình  $\pi$
- ◆ Vẽ mạch tương đương ac

## V.7 HỆ SỐ KHYUẾCH ĐẠI THỂ



**Hình 5-16**

Hình 5-16a cho thấy một mạch khuếch đại VDB. Hệ số khuếch đại điện thế của mạch được định nghĩa là tỷ số giữa thế ac lối ra và thế ac lối vào. Phần này sẽ dẫn ra biểu thức cho hệ số khuếch đại thể.

Hình 5-16b là sơ đồ tương đương ac của mạch hình 5-16a theo mô hình  $\pi$ . Dòng base chảy qua trở kháng vào của mạch base, do đó có thể viết:

$$v_{in} = i_b \beta r'_e$$

Trên collector, dòng  $i_c$  chảy qua  $R_C // R_L$  do đó

$$v_{out} = i_c (R_C // R_L)$$

Hệ số khuếch đại thể bằng:

$$A = v_{out} / v_{in} = (R_C // R_L) / r'_e \quad (5-11)$$

Điện trở collector ac. Trở tải ac nhìn từ collector là  $R_C // R_L$ . Nếu chúng ta đặt

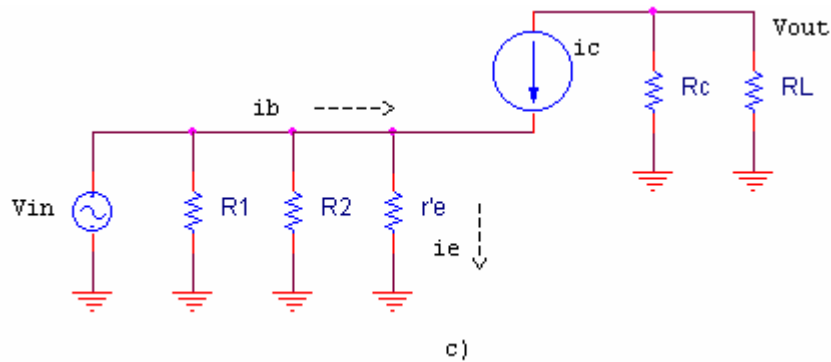
$$r_c = R_C // R_L \quad (5-12)$$

thì (5-11) trở thành

$$A = r_c / r'_e \quad (5-13)$$

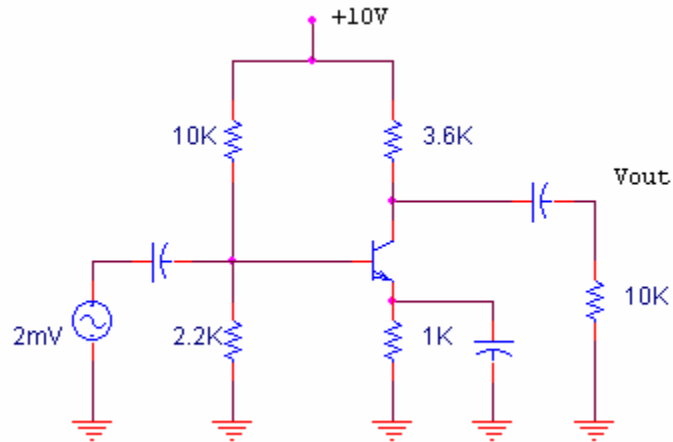


Theo (5-13) hệ số khuếch đại thế bằng điện trở ac collector chia cho điện trở ac của diode emitter.



Hình 5-16

Có thể thấy rằng từ mạch tương đương 5-16c chúng ta cũng sẽ thu được kết quả tương tự.



Hình 5-17

Ví dụ: Cho mạch hình 5-17. Tính A và thế trên tải  $R_L$ .

Giải: Trở collector ac bằng

$$r_c = R_C / R_L = (3.6K / 10K) = 2.65K$$

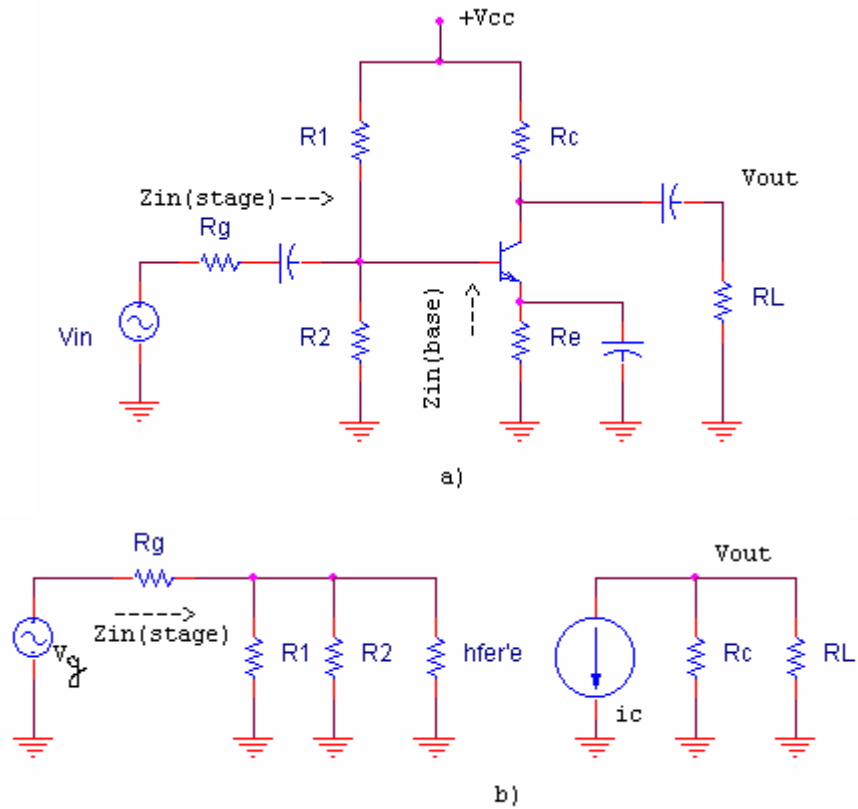
Giả sử transistor có  $r'_e = 22.7\Omega$ , ta có

$$A = r_c / r'_e = 2.65K / 22.7 = 117$$

Thế trên tải bằng

$$V_{out} = A V_{in} = 117 \cdot (2mV) = 234mV$$

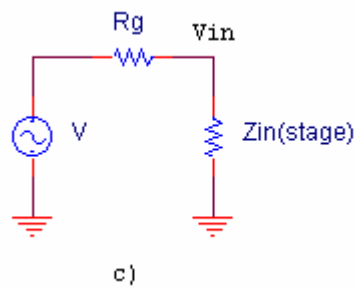
## V.8 ẢNH HƯỞNG CỦA TRỞ KHÁNG VÀO



Hình 5-18

Cho đến lúc này chúng ta vẫn giả sử nguồn thế ac vào là lý tưởng với trở kháng nội bằng 0. Trong phần này chúng ta sẽ xem xét ảnh hưởng của trở kháng nội của các nguồn thế thực (là các nguồn thế có trở kháng nội khác không).

Trên hình 5-18a, một nguồn thế ac ký hiệu là  $v_{in}$  có trở kháng nội  $R_g$ . Khi đó có một sụt thế ac ngang qua  $R_g$ . Điều này làm cho thế base bé hơn giá trị lý tưởng.



Hình 5-18

Trở kháng vào của mạch ký hiệu là  $Z_{in(stage)}$  bằng

$$Z_{in(stage)} = R_1 // R_2 // \beta r'_e$$

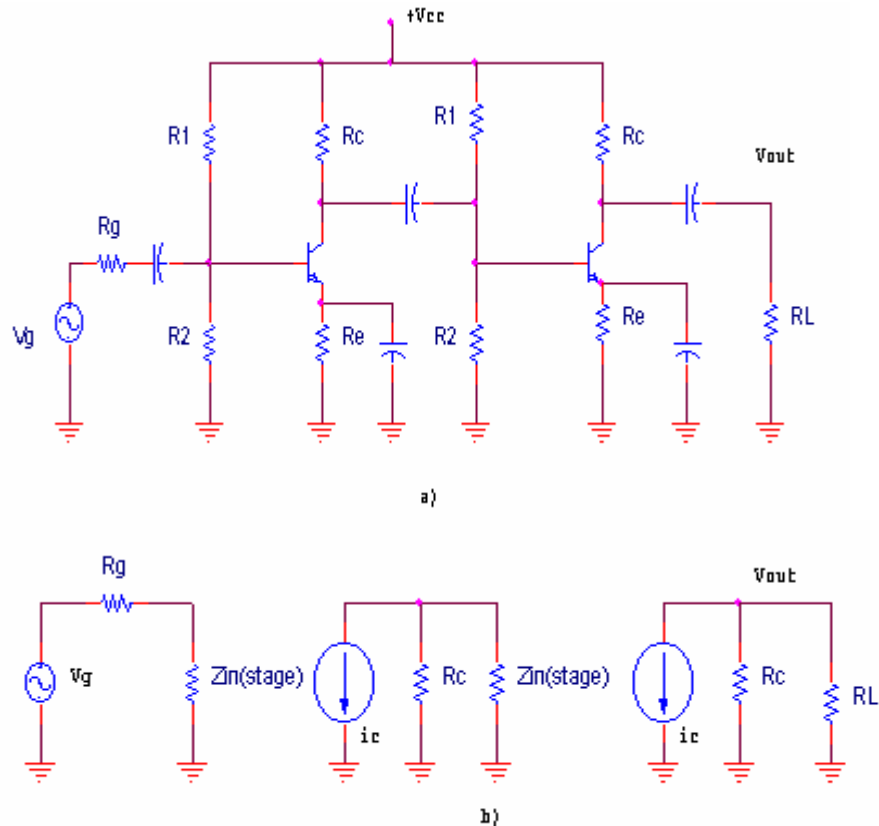
$$V_{in} = Z_{in(stage)} V_g / (R_g + Z_{in(stage)}) \quad (5-14)$$

Phương trình (5-14) có giá trị đối với mọi mạch khuếch đại. Lưu ý rằng nguồn  $v_g$  là mạnh nếu  $R_g < 0.01 Z_{in(stage)}$

Phương trình (5-14) cũng cho thấy rằng do  $R_g$  mà thế vào mạch khuếch đại bé hơn thế của nguồn cần khuếch đại.

## V.9 KHUYẾCH ĐẠI NHIỀU TẦNG

Để thu được hệ số khuếch đại lớn, chúng ta nối hai hoặc nhiều tầng khuếch đại với nhau. Điều này có nghĩa là lối ra của tầng thứ nhất là lối vào của tầng thứ hai và cứ thế tiếp tục...



Hình 5-19

Hình 5-19a cho thấy một mạch khuếch đại gồm 2 tầng. Tín hiệu ra của tầng thứ nhất (đảo pha so với tín hiệu vào) được đưa vào tầng thứ hai. Trên tải  $R_L$  chúng ta thu được tín hiệu cùng pha với tín hiệu lối vào nhưng đã được khuếch đại nhiều lần.

Hình 5-19b cho thấy mạch tương đương ac. Trở kháng collector ac của tầng thứ nhất là

$$r_c = R_C // z_{in(stage)}$$

Hệ số khuếch đại của tầng thứ nhất bằng

$$A_1 = (R_C // z_{in(stage)}) / r'_e$$

Hệ số khuếch đại của tầng thứ hai bằng

$$A_2 = (R_C // R_L) / r'_e$$

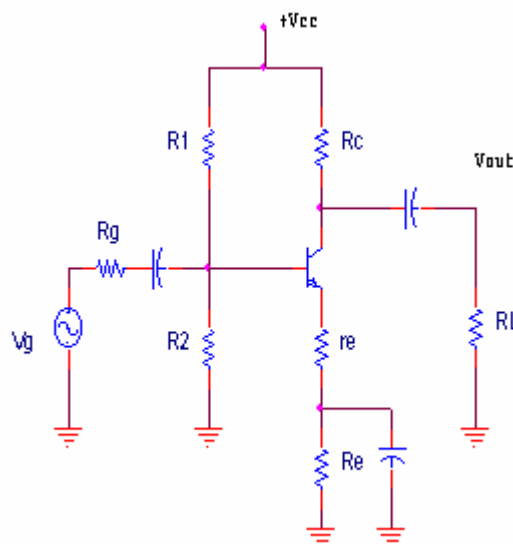
Hệ số khuếch đại của toàn mạch bằng

$$A = A_1 A_2 \quad (5-15)$$

## V.10 KHUYẾCH ĐẠI CE CẢI TIẾN

Hệ số khuếch đại thế A của mạch khuếch đại CE phụ thuộc vào dòng tĩnh, sự thay đổi của nhiệt độ và bản thân transistor.

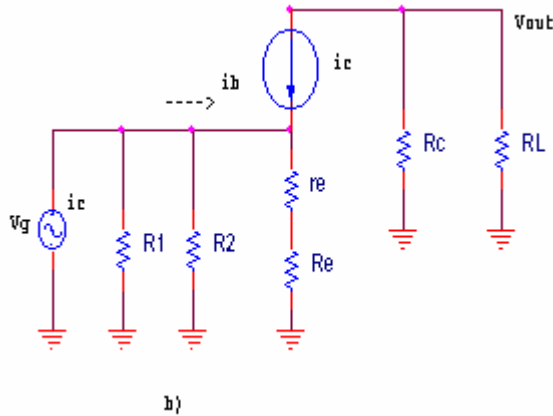
Một phương pháp nhằm ổn định hệ số A là chèn một điện trở  $r_e$  trong mạch emitter như hình vẽ 5-20a.



Hình 5-20a

Lưu ý rằng  $r_c$  không có tụ mắc song song. Khi dòng  $i_c$  chảy qua mạch emitter sẽ tạo ra một thế ac trên  $r_c$ . Điều này tạo ra sự phản hồi âm để ổn định hệ số A. Chẳng hạn, nếu dòng collector ac tăng do nhiệt độ tăng thì thế trên  $r_c$  cũng sẽ tăng. Điều này làm giảm  $v_{be}$  và do đó giảm  $i_b \rightarrow$  giảm  $i_c$ . Quá trình này diễn ra theo chiều hướng ngược lại với sự tăng của dòng collector mà chúng ta gọi là sự phản hồi âm.

Hình 5-20b là sơ đồ tương đương của mạch hình 5-20a theo mô hình T.



Hình 5-20

Thế vào bằng

$$v_{in} = i_c(r_c + r'_e)$$

Hệ số khuếch đại thế A trong trường hợp này bằng

$$A = i_c r_c / i_c (r_c + r'_e)$$

Xem rằng  $i_c \approx i_c$  thì

$$A = r_c / (r_c + r'_e) \quad (5-16)$$

Nếu  $r_c \gg r'_e$  thì (5-16) trở thành

$$A = r_c / r_c \quad (5-17)$$

Đây là ví dụ về mạch khuếch đại CE cải tiến. Giá trị  $r_c$  lớn đã loại bỏ ảnh hưởng của  $r'_e$  và làm cho A không phụ thuộc  $r'_e$ .

Trở kháng vào của mạch khuếch đại CE cải tiến là

$$Z_{in(base)} = v_{in} / i_b \quad (5-18)$$

trong đó  $v_{in} = i_c(r_c + r'_e)$

do  $i_c \approx i_c$  chúng ta nhận được

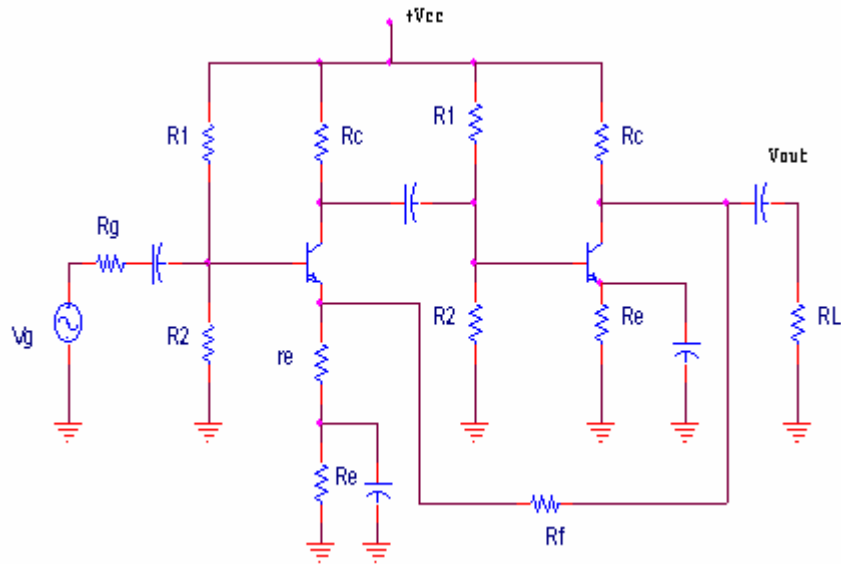
$$Z_{in(base)} = v_{in} / i_b = \beta (r_c + r'_e) \quad (5-19)$$

Nhận xét: mạch khuếch đại CE cải tiến có 3 ưu điểm:

- ◆ Hệ số A ổn định
- ◆ Tăng trở kháng vào
- ◆ Giảm sai dạng khi tín hiệu lớn (do A không phụ thuộc  $r'_e$ )

## V.11 PHẢN HỒI HAI TẦNG

Mạch khuếch đại CE cải tiến là ví dụ về khuếch đại phản hồi 1 tầng. Mục đích của mạch khuếch đại CE cải tiến là ổn định hệ số A, tăng trở kháng vào và giảm sai dạng khi tín hiệu lớn. Mạch khuếch đại phản hồi qua 2 tầng có đặc trưng tốt hơn.



Hình 5-21

Hình 5-21 là mạch khuếch đại phản hồi 2 tầng. Tầng thứ nhất là mạch khuếch đại CE cải tiến. Tầng thứ hai là mạch khuếch đại CE thông thường. Tín hiệu lối ra của tầng khuếch đại thứ hai được phản hồi qua điện trở  $R_f$  (feedback) tới emitter của tầng thứ nhất. Do cầu chia thế, thế tại emitter của tầng thứ nhất so với đất bằng:

$$v_e = (r_e v_{out}) / (R_f + r_e)$$

Ý tưởng chính ở đây là: Giả sử do sự tăng nhiệt độ làm thế ra tăng. Vì một phần của thế ra được phản hồi lại emitter transistor thứ nhất nên làm giảm  $v_{be}$  của transistor thứ nhất. Kết quả là thế lối ra của transistor thứ nhất và vì vậy thế lối ra của transistor thứ hai cũng giảm theo. Ngược lại, nếu thế ra giảm thì sau quá trình phản hồi thế lối ra sẽ tăng. Mọi thay đổi trên lối ra đều bị phản hồi về lối vào và làm cho thế lối ra chỉ có thể thay đổi trong phạm vi nhất định.

Dễ dàng thấy rằng hệ số khuếch đại của mạch phản hồi qua 2 tầng này bằng

$$A = R_f / r_c + 1 \quad (5-20)$$

Thường  $R_f / r_c \gg 1$  nên (5-20) trở thành

$$A = R_f / r_c \quad (5-21)$$

Ý nghĩa của (5-21) là ở chỗ: hệ số A chỉ phụ thuộc các điện trở trong mạng phản hồi  $R_f$  và  $r_c$  mà không phụ thuộc các yếu tố khác.

---

## Chương VI **KHUYẾCH ĐẠI CÔNG SUẤT**

### **VI.1 THUẬT NGỮ BỘ KHUYẾCH ĐẠI**

Có nhiều cách khác nhau để phân loại một mạch khuếch đại. Chẳng hạn có thể phân loại theo chế độ hoạt động, phân loại theo kiểu nối giữa các tầng, phân loại theo dải tần số, mức tín hiệu...

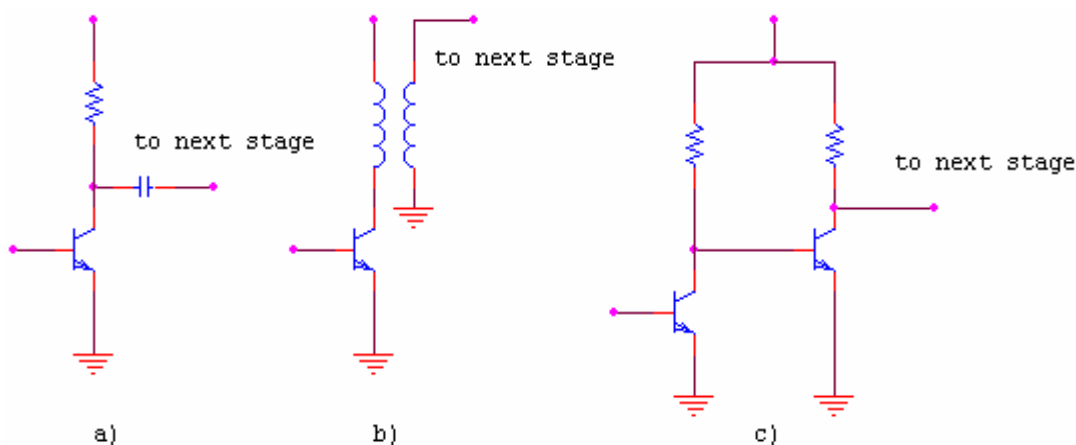
#### PHÂN LOẠI THEO CHẾ ĐỘ HOẠT ĐỘNG

**Chế độ A (hạng A).** Trong chế độ này, transistor hoạt động trong vùng tác động. Trong chế độ A, các nhà thiết kế chọn điểm Q ở giữa đường tải. Theo cách này tín hiệu có thể có biên độ tối đa mà không bị méo do transistor bão hoà hoặc ngưng dẫn.

**Chế độ B (hạng B).** Trong chế độ này, transistor được phân cực sao cho điểm Q nằm ở biên giới vùng tác động và vùng ngưng dẫn. Transistor chỉ dẫn ở nửa chu kỳ dương của nguồn ac. Điều này làm giảm nhiệt hao phí trên transistor công suất.

**Chế độ C (hạng C).** Trong chế độ này, transistor được phân cực sao cho điểm Q nằm sâu trong vùng ngưng dẫn. Transistor chỉ dẫn ở một phần của nửa chu kỳ dương của nguồn ac. Kết quả là tại collector chúng ta chỉ nhận được một xung ngắn.

#### PHÂN LOẠI THEO KIỂU NỐI TẦNG



Hình 6-1



Hình 6-1a là sơ đồ nối tầng bằng tụ. Tụ nối truyền tín hiệu đã được khuếch đại đến tầng tiếp theo. Hình 6-1b là sơ đồ nối tầng bằng biến thế. Thế ac được ghép qua biến thế và truyền đến tầng tiếp theo.

Trong hai cách nối trên đây, thành phần dc được cách ly hoàn toàn giữa các tầng khuếch đại. Hình 6-1c là sơ đồ nối tầng trực tiếp. Trong đó collector của transistor thứ nhất được nối trực tiếp đến base của transistor thứ hai. Trong cách nối trực tiếp, cả thành phần ac lẫn thành phần dc đều được nối. Tín hiệu dc được ghép từ tầng thứ nhất sang tầng thứ hai mà không bị mất mát. Khuếch đại ghép trực tiếp còn gọi là khuếch đại dc.

### PHÂN LOẠI THEO MIỀN TẦN SỐ

Một cách khác để phân loại mạch khuếch đại là dùng tần số. Ví dụ, khuếch đại âm tần (audio amplifier) chỉ các mạch khuếch đại trong vùng tần số từ 20Hz đến 20KHz. Khuếch đại cao tần (Radio Frequency) để chỉ các bộ khuếch đại có tần số trên 20KHz. Chẳng hạn khuếch đại RF trong các radio AM khuếch đại các tín hiệu từ 535KHz đến 1605KHz. Khuếch đại RF trong các radio FM khuếch đại các tín hiệu từ 88MHz đến 108MHz.

Theo tần số người ta cũng phân loại khuếch đại band hẹp và khuếch đại band rộng. Khuếch đại band hẹp chỉ khuếch đại tín hiệu trong một vùng tần số, chẳng hạn từ 450 KHz đến 460KHz. Khuếch đại band rộng khuếch đại tín hiệu trong một vùng tần số rộng chẳng hạn từ 0 đến 1MHz. Khuếch đại band hẹp còn gọi là khuếch đại RF cộng hưởng. Tải của mạch khuếch đại là mạch cộng hưởng LC. Nó thường dùng trong các mạch chọn kênh trong radio hoặc TV.

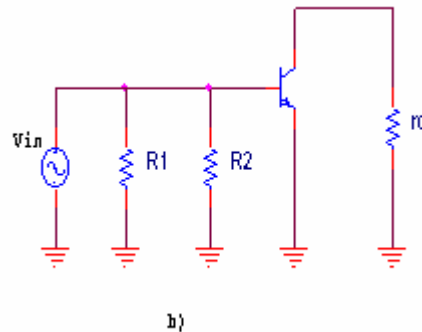
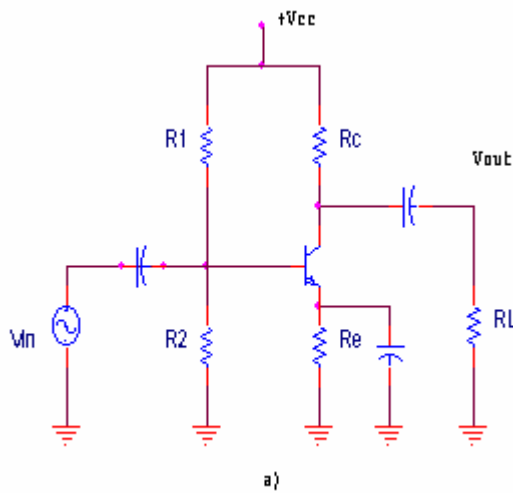
### PHÂN LOẠI THEO MỨC TÍN HIỆU

Chúng ta đã xem xét hoạt động của transistor ở chế độ tín hiệu bé. Trong đó dòng collector khi có tín hiệu ac chỉ bằng 1/10 biên độ dòng collector tĩnh. Trong chế độ tín hiệu lớn, biên độ đỉnh – đỉnh của tín hiệu có thể chiếm hết biên độ đường tải. Trong các hệ thống xử lý âm thanh, tín hiệu bé từ đầu CD được dùng như tín hiệu vào của bộ tiền khuếch đại. Tiền khuếch đại là mạch khuếch đại tín hiệu bé. Ở đây, tín hiệu được khuếch đại lên biên độ thích hợp cho tầng điều chỉnh âm sắc hoặc điều chỉnh âm lượng. Sau đó tín hiệu được đưa vào tầng khuếch đại công suất. Khuếch đại công suất là mạch khuếch đại tín hiệu lớn. Nó tạo ra một công suất từ vài mW đến hàng trăm W để đưa ra loa.

## VI.2 HAI ĐƯỜNG TẢI

Mỗi mạch khuếch đại có mạch tương đương dc và mạch tương đương ac. Do đó mỗi mạch khuếch đại có 2 đường tải: đường tải ac và đường tải dc. Trong chế độ tín hiệu bé, vị trí của Q không quan trọng nhưng trong chế độ tín hiệu lớn điểm Q cần phải ở giữa đường tải ac để có được tín hiệu cực đại mà không bị méo.

### Đường tải DC



Hình 6-2

Hình 6-2a là mạch khuếch đại VDB. Một cách để di chuyển điểm Q là thay đổi  $R_2$ . Khi  $R_2$  rất lớn transistor rơi vào vùng bão hoà với dòng  $I_C$  bão hoà cho bởi phương trình

$$I_{C(sat)} = V_{CC} / (R_C + R_E) \quad (6-1)$$

Nếu  $R_2$  rất bé transistor sẽ rơi vào vùng ngưng dẫn với thế trên transistor cực đại và bằng

$$V_{CE(sat)} = V_{CC} \quad (6-2)$$

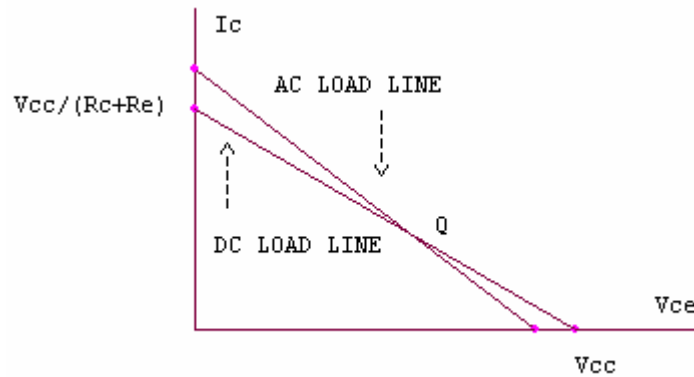
### Đường tải AC

Hình 6-2b là mạch tương đương ac của mạch khuếch đại VDB. Do  $R_E$  nối GND về mặt xoay chiều nên không ảnh hưởng đến hoạt động của mạch. Hơn nữa trở collector ac bé hơn trở collector dc. Khi có tín hiệu ac vào, điểm Q tức thời di chuyển trên đường tải ac như hình 6-3. Nói cách khác, dòng và thế xoay chiều cực đại qui định bởi đường tải ac.

Do độ dốc của đường tải ac lớn hơn độ dốc của đường tải dc nên giá trị lồi ra đỉnh - đỉnh cực đại (maximum peak to peak) MPP bé hơn nguồn cung cấp  $V_{CC}$ .

$$MPP < V_{CC} \quad (6-3)$$

### Sự cắt khi tín hiệu lớn



Hình 6-3

Khi điểm Q nằm giữa đường tải dc như hình 6-3, tín hiệu ac không thể dùng hết đường tải ac mà không bị cắt. Một mạch khuếch đại tín hiệu lớn được thiết kế tốt có Q nằm giữa đường tải ac. Trong trường hợp này chúng ta thu được tín hiệu lớn nhất mà không bị cắt.

### BIÊN ĐỘ TÍN HIỆU RA

Ứng với một mạch khuếch đại, giá trị đỉnh của tín hiệu ra là

$$MP = I_{CQ} \cdot r_c \quad (6-4)$$

Tín hiệu ra đỉnh - đỉnh cực đại bằng

$$MPP = 2MP \quad (6-5)$$

### VI.3 KHUYẾCH ĐẠI HẠNG A

Hình 6-4a là mạch khuếch đại VDB hạng A với tín hiệu ra không bị cắt. Chúng ta hãy dẫn ra một số phương trình cho mạch khuếch đại này.

#### HỆ SỐ KHUYẾCH ĐẠI CÔNG SUẤT

Chúng ta định nghĩa hệ số khuếch đại công suất là

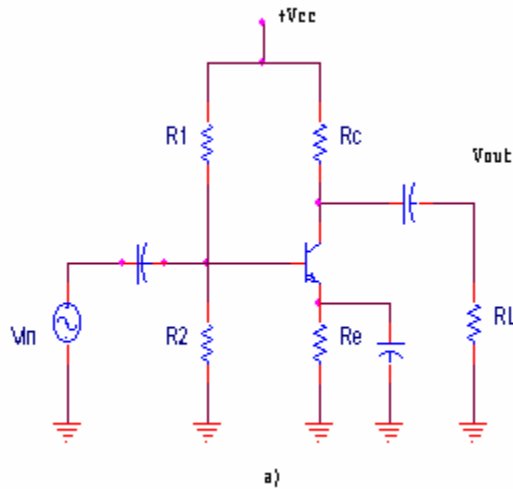
$$G = P_{\text{out}} / P_{\text{in}} \quad (6-6)$$

Hệ số khuếch đại công suất bằng tỷ số công suất lối ra chia cho công suất lối vào.

Ví dụ mạch khuếch đại hình 6-4a có công suất ra là 10mW và công suất vào 10μW thì hệ số khuếch đại công suất bằng

$$G = 10\text{mW} / 10\mu\text{W} = 1000$$

#### CÔNG SUẤT RA



Hình 6-4

Nếu đo thế ra trên hình 6-4a bằng giá trị hiệu dụng thì công suất ra của mạch khuếch đại trên là

$$P_{\text{out}} = V_{\text{rms}}^2 / R_L \quad (6-7)$$

Nếu ký hiệu  $V_{\text{out}}$  là thế ra đỉnh – đỉnh thì (6-7) trở thành

$$P_{\text{out}} = V_{\text{out}}^2 / 8R_L \quad (6-8)$$

Công suất ra cực đại khi  $V_{\text{out}} = \text{MPP}$  hay

$$P_{\text{out(max)}} = \text{MPP}^2 / 8R_L \quad (6-9)$$

### Công suất tiêu tán của transistor

Khi không có tín hiệu đưa vào mạch khuếch đại trên hình 6-4a, công suất tiêu tán tĩnh bằng

$$P_{DQ} = V_{CEQ} \cdot I_{CQ} \quad (6-10)$$

Khi có tín hiệu ac, công suất tiêu tán của transistor giảm vì transistor biến một phần công suất tĩnh thành công suất tín hiệu. Do đó giới hạn công suất của transistor trong mạch khuếch đại hạng A phải lớn hơn  $P_{DQ}$  nếu không transistor sẽ bị phá hủy.

### Hiệu suất

Nếu gọi  $I_{dc}$  là dòng mà nguồn nuôi cung cấp cho bộ khuếch đại thì công suất mà nguồn dc cung cấp cho bộ khuếch đại bằng

$$P_{dc} = I_{dc} V_{CC} \quad (6-11)$$

Để so sánh khả năng biến đổi năng lượng của nguồn thành tín hiệu của các mạch khuếch đại người ta định nghĩa hiệu suất

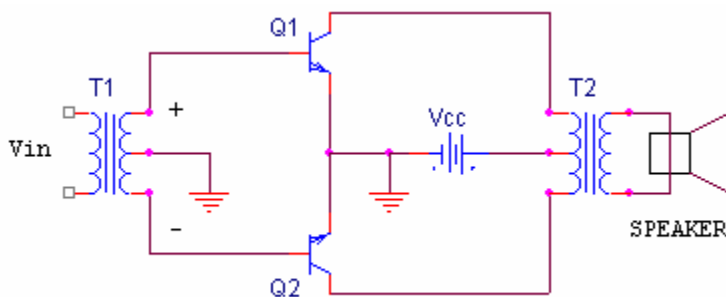
$$\eta = (P_{out} / P_{dc}) \times 100\% \quad (6-12)$$

Thông thường hiệu suất của một khuếch đại hạng A là 25%.

## VI.4 KHUYẾCH ĐẠI HẠNG B

Mạch khuếch đại hạng A là cách thường dùng để sử dụng transistor trong các mạch tuyến tính vì nó làm cho mạch phân cực ổn định nhất và đơn giản nhất. Nhưng khuếch đại hạng A không phải là cách hiệu quả nhất để vận hành transistor. Trong một số hệ thống dùng pin vấn đề hiệu suất rất quan trọng. Phần này sẽ giới thiệu những ý tưởng chính của khuếch đại hạng B.

Hình 6-5



Hình 6-5 là mạch khuếch đại đẩy kéo hạng B. Khi một transistor hoạt động ở chế độ B, nó cắt một nửa chu kỳ tín hiệu. Để tránh méo dạng chúng ta có thể dùng 2 transistor kiểu đẩy kéo như hình 6-5. Push pull nghĩa là trong một nửa chu kỳ của tín hiệu chỉ có một transistor dẫn còn transistor kia ngưng dẫn.

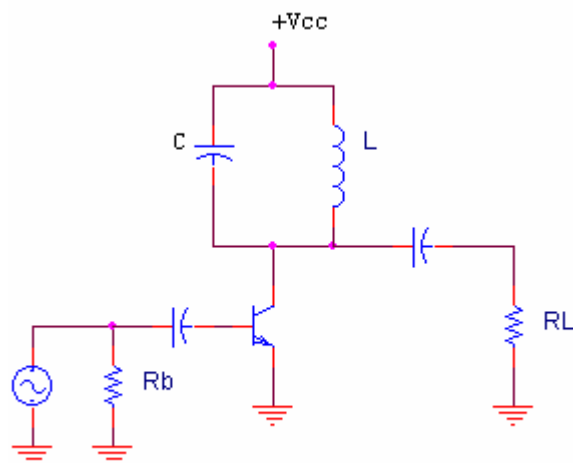
Hoạt động của mạch như sau: vào nửa chu kỳ dương của tín hiệu vào cuộn thứ cấp của  $T_1$  có thế  $v_1$  và  $v_2$  như hình vẽ 6-5. Do đó transistor Q1 dẫn còn transistor Q2 ngưng dẫn. Dòng collector chảy qua Q<sub>1</sub> và qua cuộn sơ cấp của biến thế ra  $T_2$ . Điều này tạo ra thế đảo pha và được khuếch đại. Biến thế  $T_2$  ghép tín hiệu đã được khuếch đại với loa. Vào nửa chu kỳ âm của tín hiệu vào, cực tính bị đảo ngược. Bây giờ transistor Q2 dẫn, transistor Q1 ngưng dẫn. Tín hiệu của nửa chu kỳ âm lại xuất hiện trên loa thông qua  $T_2$ . Trong khuếch đại đẩy kéo mỗi transistor khuếch đại một nửa chu kỳ, loa nhận được cả chu kỳ của tín hiệu nhưng đã được khuếch đại.

#### Ưu điểm và nhược điểm

Do không phân cực nên khi không có tín hiệu vào, nguồn nuôi không cung cấp một dòng dc nào. Ưu điểm thứ hai là hiệu suất được cải thiện đến 78.5%. Do đó khuếch đại đẩy kéo hạng B thường dùng trong các tầng ra (tầng khuếch đại cuối cùng của thiết bị).

Nhược điểm của sơ đồ là dùng biến thế nặng nề và đắt tiền. Ngày nay các thiết kế mới đã thay thế cho khuếch đại đẩy kéo kiểu này.

## VI.5 KHUYẾCH ĐẠI HẠNG C



Hình 6-6

Khuyếch đại hạng C cần dùng một mạch cộng hưởng cho tải. Do đó khuyếch đại hạng C còn gọi là khuyếch đại cộng hưởng.

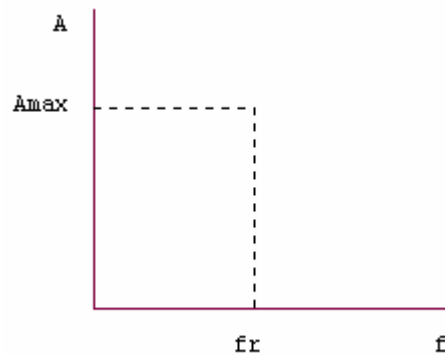
### Tần số cộng hưởng

Trong khuyếch đại hạng C, dòng collector chảy qua transistor chỉ trong một phần thời gian của một nửa chu kỳ. Một mạch cộng hưởng song song có thể lọc các xung của dòng collector và tạo ra một thế hình sin. Ứng dụng chủ yếu của khuyếch đại hạng C là mạch khuyếch đại RF cộng hưởng. Hiệu suất tối đa của mạch khuyếch đại hạng C là 100%.

Hình 6-6 là sơ đồ một mạch khuyếch đại RF cộng hưởng. Tín hiệu xoay chiều đưa vào cực base. Thế lối ra đã được khuyếch đại xuất hiện tại collector. Do có mạch cộng hưởng thế ra có biên độ lớn nhất tại tần số cộng hưởng của mạch

$$f_r = 1/2\pi(LC)^{1/2} \quad (6-13)$$

Hai bên tần số cộng hưởng hệ số khuyếch đại thế giảm nhanh như hình 6-7.



Hình 6-7

Mạch khuyếch đại hạng C thường dùng cho các ứng dụng khuyếch đại RF cộng hưởng hay khuyếch đại band hẹp. Chúng được dùng nhiều trong các mạch khuyếch đại RF cho tín hiệu radio và TV.

Các công thức sau đây là áp dụng cho khuyếch đại hạng C

$$G = P_{out} / P_{in} \quad (\text{hệ số khuyếch đại công suất})$$

$$p_{out} = v_{out}^2 / 8R_L \quad (\text{công suất ra ac})$$

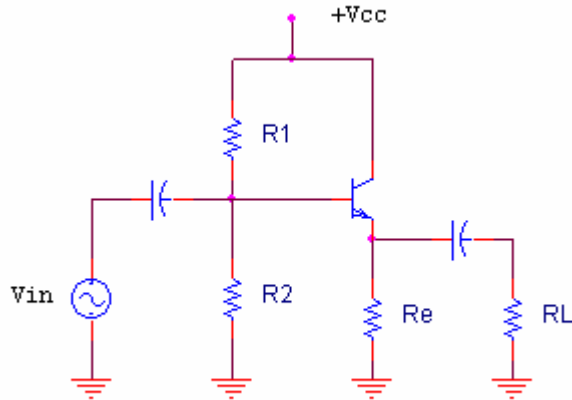
$$p_{out(max)} = MPP^2 / 8R_L \quad (\text{công suất ra ac tối đa})$$

$$P_{dc} = V_{CC}I_{dc} \quad (\text{công suất vào dc})$$

$$\eta = (p_{out} / p_{in}) \cdot 100\% \quad (\text{hiệu suất})$$

## VI.6 KHUYẾT ĐẠI CHUNG COLLECTOR

Mạch khuếch đại chung collector còn gọi là mạch lặp lại emitter. Tín hiệu ac cần khuếch đại được đưa vào base. Thế ra lấy trên emitter.



Hình 6-8

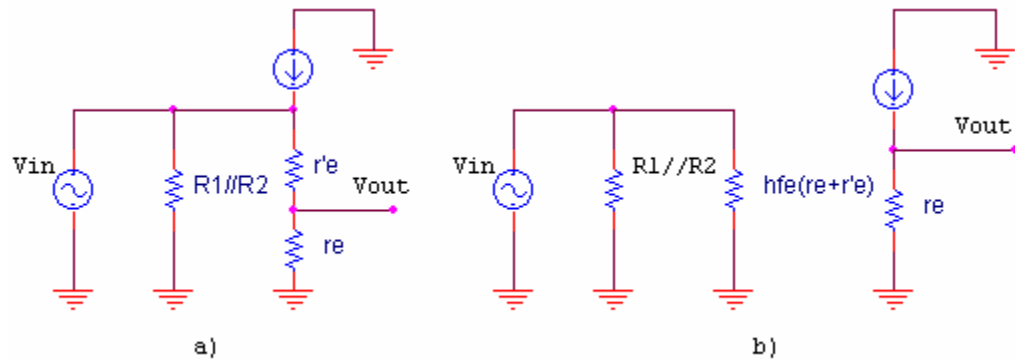
Hình 6-8 là sơ đồ một mạch khuếch đại CC. Thế vào được đưa vào base. Nó setup một dòng emitter và tạo ra thế emitter. Thế này nối qua tụ ra tải. Gọi là mạch khuếch đại lặp lại vì thế ra giống thế vào về biên độ và pha. Thực vậy, có thể rút ra các giá trị của mạch như sau.

Trở emitter ac

$$r_c = R_E // R_L \quad (6-14)$$

**Hệ số khuếch đại thế**

Hình 6-9 là sơ đồ tương đương mô hình chữ T của mạch CC.



Hình 6-9



Dùng định luật Ohm có thể viết các phương trình sau

$$V_{out} = i_c r_c$$

$$V_{in} = i_c (r_c + r'_c)$$

$$A = r_c / (r_c + r'_c) \quad (6-15)$$

Thông thường chọn  $r_c > r'_c$  nên  $A \approx 1$ .

Trở kháng vào của mạch base

$$Z_{in(base)} = \beta(r_c + r'_c) \quad (6-16)$$

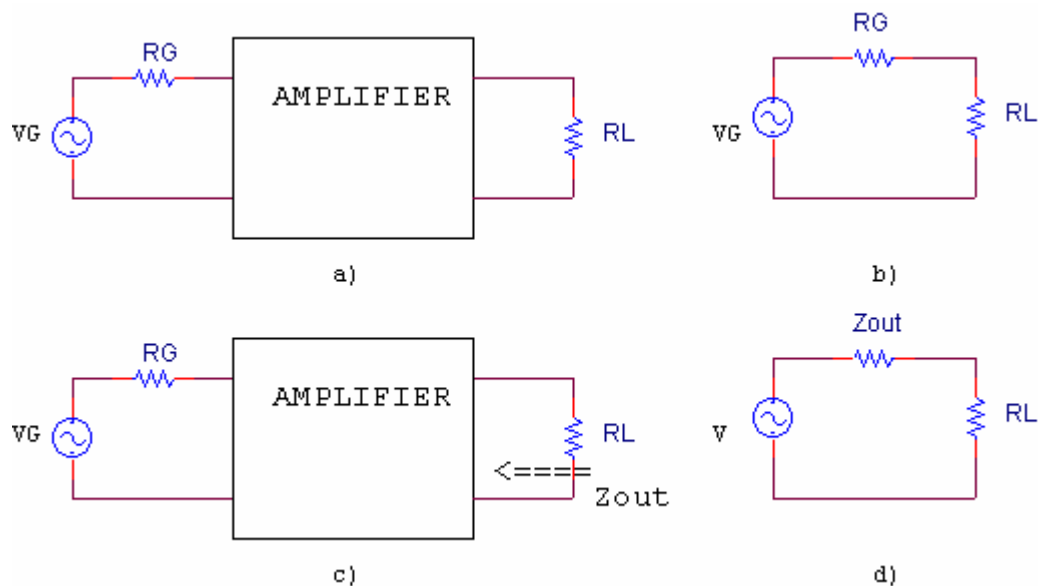
Trở kháng vào của mạch khuếch đại CC

$$Z_{in(stage)} = R_1 // R_2 // \beta(r_c + r'_c) \quad (6-17)$$

## VI.7 TRỞ KHÁNG RA CỦA MẠCH KHUYẾCH ĐẠI

Trở kháng ra của một bộ khuếch đại giống như trở kháng Thevenin. Ưu điểm của mạch khuếch đại CC là có trở kháng ra thấp.

Công suất trên tải tối đa khi có sự phù hợp giữa trở kháng nguồn và trở kháng tải. Chẳng hạn để có công suất tối đa trên tải là loa thì cần sử dụng mạch khuếch đại CC có trở kháng ra thấp.

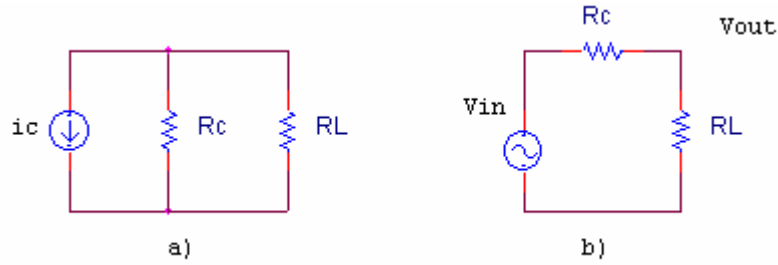


Hình 6-10

Hình 6-10a chỉ ra một nguồn ac thúc một bộ khuếch đại. Bên lối vào do có  $R_G$  nên sẽ có sụt thế qua nó. Chúng ta có thể dùng mạch hình 6-10b để tính thế vào.

Bên lối ra cũng có thể áp dụng định lý Thevenin. Từ phía tải nhìn vào bộ khuếch đại chúng ta thấy một trở kháng ra  $Z_{out}$ .

## Trở kháng ra của mạch khuếch đại CE



Hình 6-11

Hình 6-11a cho thấy mạch tương đương ac ở phía ra của bộ khuếch đại CE. Khi áp dụng định lý Thevenin chúng ta thu được hình 6-11b. Trở kháng ra của mạch chính là  $R_C$ . Vì hệ số khuếch đại của mạch CE phụ thuộc  $R_C$ , do đó mạch CE khó có thể có trở kháng ra thấp mà không mất mát hệ số khuếch đại thế.

## Trở kháng ra của mạch khuếch đại CC

Hình 6-12a là sơ đồ mạch tương đương ac của một bộ khuếch đại CC. Trở kháng ra của mạch bằng

$$z_{out} = R_E // (r'_e + (R_G // R_1 // R_2) / \beta) \quad (6-18)$$

Trong thiết kế  $R_1$  và  $R_2$  và điện trở ac  $r'_e$  của emitter có thể bỏ qua bên cạnh giá trị bé của  $R_G / \beta$ . Vì vậy cuối cùng

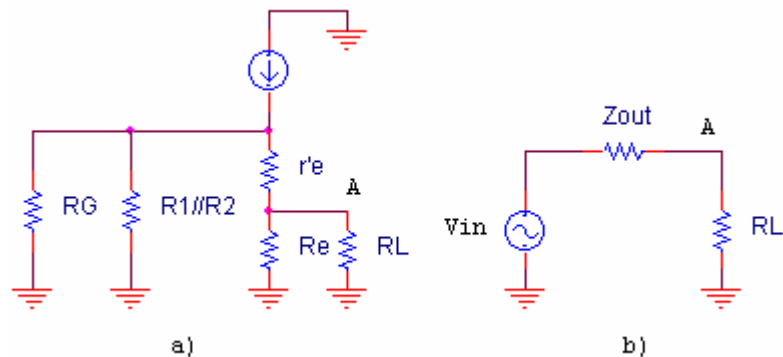
$$z_{out} = R_G / \beta \quad (6-19)$$

Theo (6-19), mạch CC làm giảm trở kháng nguồn  $\beta$  lần. Do đó nó có thể cung cấp một công suất đủ lớn trên tải là loa có trở kháng thấp.

Để đạt được công suất tối đa trên tải  $R_L$  các nhà thiết kế chọn giá trị của các linh kiện sao cho

$$Z_{out} = R_L \quad (6-20)$$

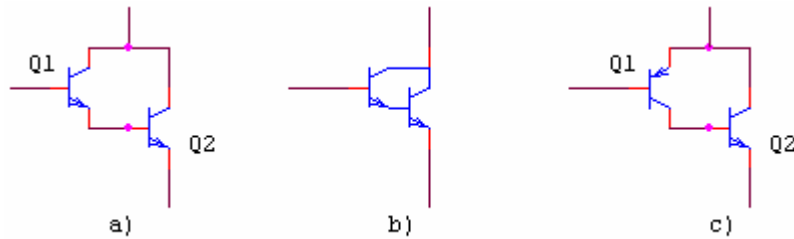
Phương trình (6-20) gọi là điều kiện phối hợp trở kháng.



Hình 6-12

## VI.8 NỐI DARLINGTON

Nối Darlington là nối giữa 2 transistor để tạo thành một transistor có hệ số khuếch đại dòng bằng tích hệ số khuếch đại dòng của các transistor riêng rẽ. Mạch transistor nối Darlington cho trở kháng vào rất cao và dòng ra rất lớn. Nối Darlington thường dùng trong các mạch ổn áp và khuếch đại công suất.



Hình 6-13

### CẶP DARLINGTON.

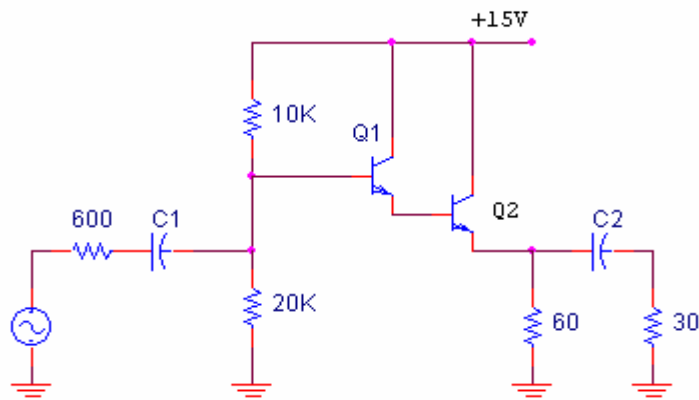
Hình 6-13a cho thấy một cặp Darlington. Do dòng emitter của  $Q_1$  là dòng base của  $Q_2$  nên cặp Darlington có hệ số khuếch đại toàn thể bằng:

$$\beta = \beta_1 \beta_2 \quad (6-21)$$

Nhà máy sản xuất linh kiện bán dẫn có thể đặt cặp transistor Darlington trong một vỏ và gọi là transistor Darlington như hình 6-13b. Ví dụ TP102 là transistor Darlington công suất có hệ số  $\beta=1000$  tại dòng bằng 3A.

### DARLINGTON BÙ

Hình 6-13c cho thấy một cách nối Darlington khác gọi là Darlington bù. Darlington bù gồm một transistor pnp và một transistor npn. Dòng collector của  $Q_1$  là dòng base của  $Q_2$ . Hệ số  $\beta$  của Darlington bù cũng xác định bởi phương trình (6-21).



Hình 6-14

Ví dụ: Cho mạch trên hình 6-14. Transistor Darlington có  $\beta=10000$ . Tính trở kháng vào và tín hiệu ra đỉnh - đỉnh cực đại của mạch.

$$r_c = 60\Omega // 30\Omega = 20\Omega$$

Trở kháng vào của  $Q_1$  bằng

$$Z_{in(base)} = 10000(20\Omega) = 200K\Omega$$

Dòng emitter dc của  $Q_2$  là:

$$I_{EQ} = (10V - 1.4V) / 60\Omega = 143mA$$

Thế collector-emitter tĩnh bằng

$$V_{CEQ} = 15V - (60\Omega \cdot 143mA) = 15V - 8.6V = 6.4V$$

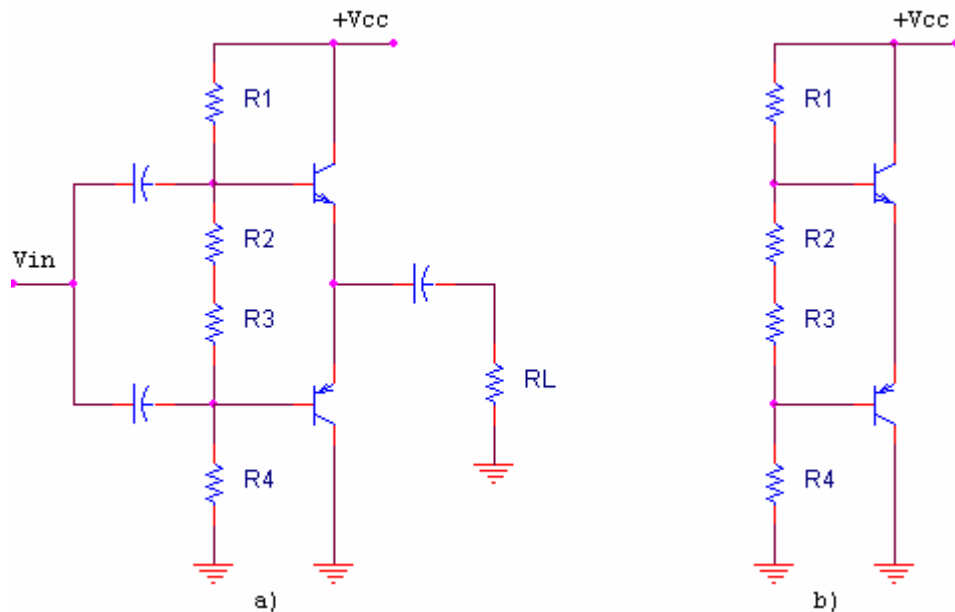
Thế xoay chiều đỉnh tại emitter bằng

$$I_{CQ} \cdot r_c = (143mA)(20\Omega) = 2.86V$$

$$MPP = 2 V_{CEQ} = 2(2.86) = 5.72V$$

## VI.9 LẬP LẠI EMITTER ĐẨY KÉO HẠNG B

Khuyếch đại hạng B có nghĩa là dòng collector chỉ tồn tại trong  $\frac{1}{2}$  chu kỳ của tín hiệu. Để điều này xảy ra, điểm Q phải ở trong vùng ngưng dẫn. Ưu điểm của khuyếch đại hạng B là dòng cung cấp từ nguồn bé và hiệu suất cao.



Hình 6-15

Hình 6-15a là sơ đồ mạch lặp lại emitter đẩy kéo hạng B.

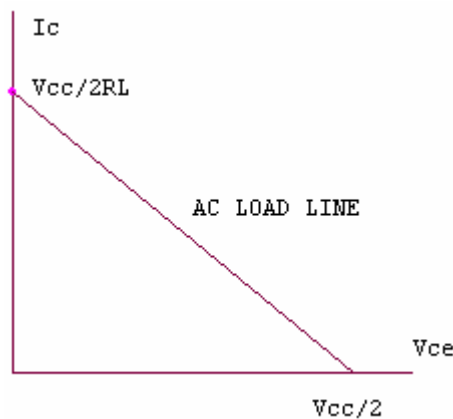
Các điện trở phân cực nhằm setup điểm Q tại vùng ngưng dẫn với thế trên diode emitter cỡ 0.6V đến 0.7V. Một cách lý tưởng thì phân cực hạng B làm cho

$$I_{CQ}=0$$

Vì các điện trở phân cực có giá trị bằng nhau, các diode emitter có điện áp cũng bằng nhau nên thế rơi trên mỗi transistor bằng một nửa giá trị nguồn nuôi, hay

$$V_{CEQ} = V_{CC} / 2 \quad (6-22)$$

### ĐƯỜNG TẢI AC



Hình 6-16

Hình 6-16 cho thấy đường tải ac. Khi transistor đang dẫn điểm hoạt động của transistor dịch chuyển dọc theo đường tải từ vị trí ngưng dẫn đến bão hoà. Thế ra cực đại đỉnh đỉnh bằng

$$MPP=V_{CC} \quad (6-23)$$

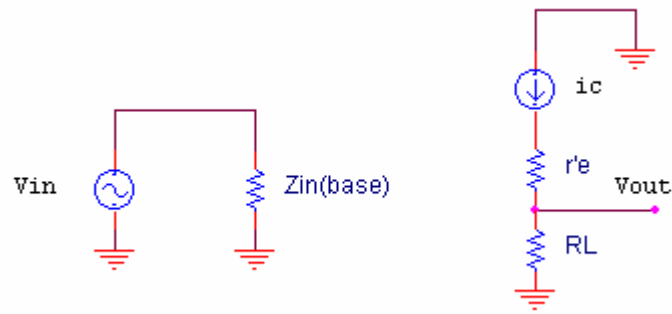
### PHÂN TÍCH AC

Hình 6-17 là mạch tương đương của transistor đang dẫn. Mạch này giống mạch tương đương ac của khuếch đại transistor hạng A. Bỏ qua  $r'_e$  thì

$$A \approx 1 \quad (6-24)$$

Trở kháng vào của base bằng

$$Z_{in(base)} = \beta R_L \quad (6-25)$$



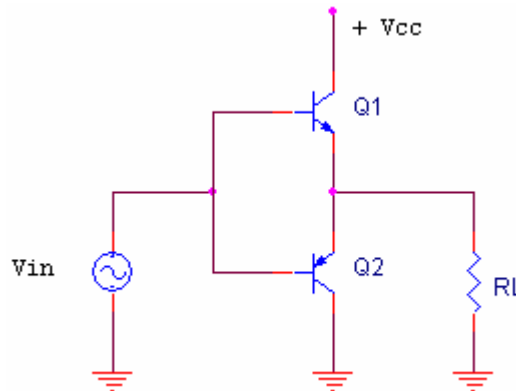
Hình 6-17

Hoạt động tổng quát của mạch như sau:

Nửa chu kỳ dương transistor phía trên dẫn, transistor phía dưới ngưng dẫn. Transistor phía trên giống như một mạch lặp lại emitter thông thường do đó thế trên tải bằng thế vào. Ngược lại, nửa chu kỳ âm transistor phía dưới dẫn, transistor phía trên ngưng dẫn. Transistor phía dưới cũng giống như một mạch lặp lại emitter thông thường do đó thế trên tải bằng thế vào.

Đặc điểm của mạch là có trở kháng vào cao nếu như dùng các transistor Darlington.

### MÉO XUYÊN TÂM



Hình 6-18

Nếu không phân cực cho 2 transistor như mạch hình 6-18, thế ra của bộ khuếch đại sẽ bị méo khi tín hiệu vào chưa vượt mức  $\pm 0.7V$ . Để tránh sai dạng người ta phân cực nhẹ cho transistor để điểm Q nằm phía trên vùng ngưng dẫn một chút. Thông thường chọn  $I_{CQ}$  bằng 1 đến 5% của  $I_{C(sat)}$ .

Các công thức sau đây áp dụng cho khuếch đại đẩy kéo hạng B.

$$G = P_{\text{out}} / P_{\text{in}} \quad (\text{hệ số khuếch đại công suất})$$

$$p_{\text{out}} = v_{\text{out}}^2 / 8R_L \quad (\text{công suất ra ac})$$

$$p_{\text{out(max)}} = MPP^2 / 8R_L \quad (\text{công suất ra ac tối đa})$$

$$P_{\text{dc}} = V_{\text{CC}} I_{\text{dc}} \quad (\text{công suất vào dc})$$

$$\eta = (p_{\text{out}} / p_{\text{in}}) \cdot 100\% \quad (\text{hiệu suất})$$

### CÔNG SUẤT TIÊU TÁN CỦA TRANSISTOR

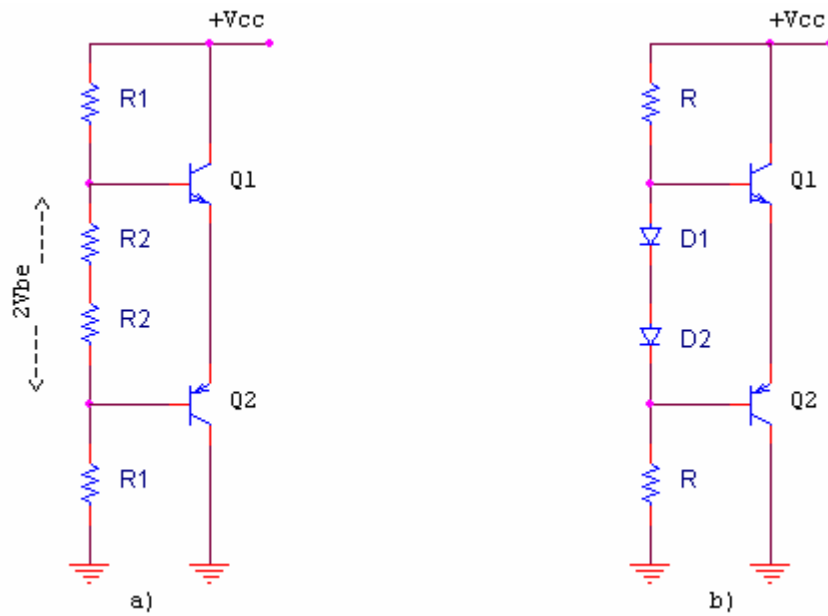
Lý tưởng thì công suất tiêu tán của transistor hạng B bằng 0 vì không có dòng dc. Nếu có phân cực chống méo xuyên tâm thì công suất tiêu tán của transistor cũng rất nhỏ.

Khi có tín hiệu vào, công suất tiêu tán của transistor trở nên đáng kể. Công suất tiêu tán của mỗi transistor trong khuếch đại đẩy kéo hạng B có thể tính theo công thức sau:

$$P_{D(\text{max})} = MPP^2 / 40R_L \quad (6-26)$$

## VI.10 PHÂN CỰC KHUYẾCH ĐẠI ĐẨY KÉO HẠNG B

Khó khăn nhất khi thiết kế mạch khuếch đại đẩy kéo hạng B là đặt điểm Q gần vùng ngưng dẫn. Hình 6-19a cho thấy mạch phân cực bằng cầu chia thế. Để tránh sai dạng xuyên tâm người ta phân cực nhẹ cho cặp transistor với  $V_{\text{BE}} \approx 0.6\text{V}$  đến  $0.7\text{V}$ . Nhưng có một khó khăn là dòng collector rất nhạy với thay đổi của thế  $V_{\text{BE}}$ . Số liệu của transistor cho thấy nếu  $V_{\text{BE}}$  tăng 60mV thì dòng collector có thể tăng 10 lần. Thường người ta thay điện trở bằng biến trở để chọn điểm Q phù hợp. Nhưng nó vẫn không giải quyết được vấn đề nhiệt độ. Điểm Q có thể rất tốt tại nhiệt độ phòng nhưng nó sẽ thay đổi khi nhiệt độ thay đổi (điều này là không tránh khỏi tại tầng công suất). Với cách phân cực như hình 6-19a thì giả sử nhiệt độ tăng mà thế phân cực base không đổi (trong khi  $V_{\text{BE}}$  giảm) sẽ làm tăng nhanh chóng dòng collector, kết quả là transistor sẽ bị chết do quá nhiệt.



Hình 6-19

Cần phải bù lại thế phân cực base của transistor khi nhiệt độ thay đổi. Giải pháp là dùng mạch diode bù như hình 6-19b. Tính chất của diode bù phải giống như diode emitter của transistor. Nếu nhiệt độ tăng  $1^{\circ}\text{C}$  thì thế trên diode bù giảm  $2\text{mV}$ . Do đó mạch sẽ có tác dụng bù nhiệt độ. Thực tế là khi nhiệt độ tăng thế phân cực sẽ giảm, do đó dòng collector cũng sẽ giảm.

Ví dụ: Cho mạch khuếch đại đẩy kéo hạng B như hình 6-20. Tính dòng collector tĩnh và hiệu suất cực đại của mạch.

Giải: Dòng phân cực qua diode bù bằng

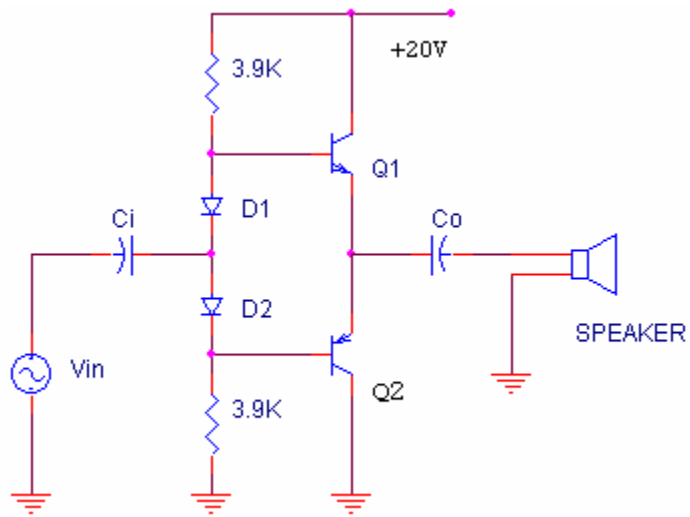
$$I_{\text{bias}} = (20\text{V} - 1.4\text{V}) / 2(3.9\text{K}) = 2.38\text{mA}$$

Đây chính là giá trị của dòng collector tĩnh do hiệu ứng gương dòng điện.

Dòng collector bão hoà bằng

$$I_{\text{c(sat)}} = V_{\text{CEQ}} / R_{\text{L}} = 10\text{V} / 10 = 1\text{A}$$





Hình 6-20

Giá trị trung bình của dòng collector bằng

$$I_{av} = I_{c(sat)}/\pi = 1A / \pi = 0.318A$$

Dòng collector tổng bằng

$$I_{dc} = 2.38mA + 0.318A = 0.32A$$

Công suất vào dc bằng

$$P_{dc} = (20V)(0.32A) = 6.4W$$

Công suất ra ac cực đại bằng

$$P_{out(max)} = MPP^2/8R_L = (20V)^2 / 8.10 = 5W$$

Hiệu suất của mạch bằng

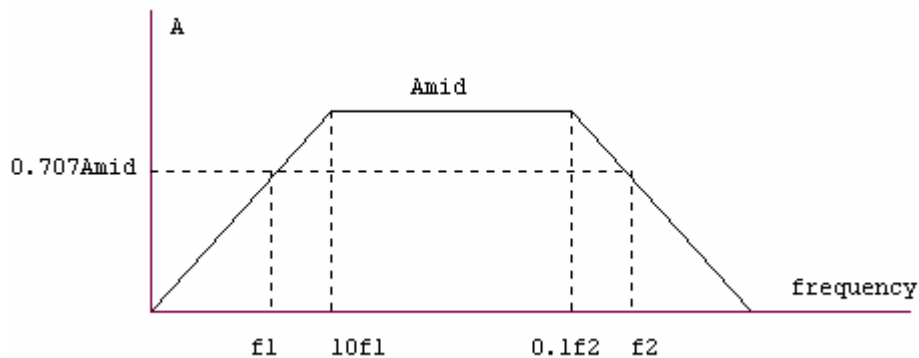
$$\eta = (P_{out}/P_{dc}) \times 100\% = (5/6.4) \times 100\% = 78.1\%$$

## Chương VII **CÁC HIỆU ỨNG TẦN SỐ CỦA MẠCH KHUYẾCH ĐẠI**

### **VII.1 ĐÁP TUYẾN TẦN SỐ CỦA MỘT BỘ KHUYẾCH ĐẠI**

Đáp tuyến tần số của một bộ khuếch đại là giản đồ quan hệ giữa hệ số khuếch đại và tần số. Trong phần này chúng ta sẽ phân tích đáp tuyến tần số của một mạch khuếch đại ac và dc.

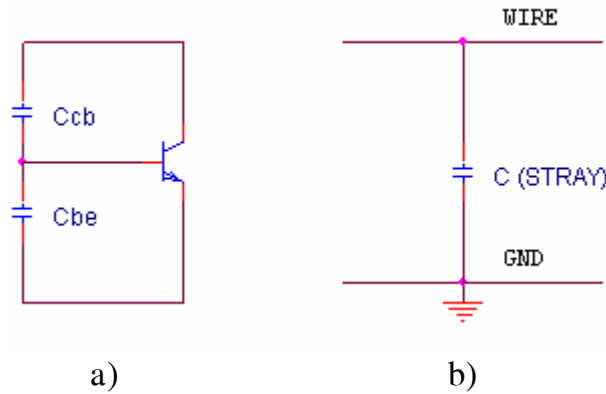
#### **ĐÁP TUYẾN CỦA MẠCH KHUYẾCH ĐẠI AC**



Hình 7-1

Hình 7-1 cho thấy đáp tuyến tần số của mạch khuếch đại ac. Ở giữa đáp tuyến hệ số khuếch đại có giá trị lớn nhất. Nói cách khác mạch hoạt động bình thường ở vùng giữa các tần số. Tại tần thấp do các tụ nối và tụ thông dẫn không hoàn toàn nối tắt nên hệ số khuếch đại thể giảm. Càng gần 0Hz hệ số A càng giảm.

Tại tần số cao hệ số A cũng bị suy giảm vì những lý do khác. Chẳng hạn, giữa các mối nối của transistor có các tụ điện ký sinh như hình 7-2a. Những tụ này tạo ra đường thông dẫn cho tín hiệu ac. Ở tần số cao, trở kháng của các tụ này đủ bé và chúng ngăn cản hoạt động bình thường của transistor. Kết quả là giá trị của A bị suy giảm ở tần số cao. Một lý do khác làm cho A suy giảm ở tần số cao là các tụ ký sinh của các dây nối. Về nguyên tắc một dây dẫn là 1 bản tụ. Giữa một dây dẫn và GND hình thành một tụ điện ký sinh như hình 7-2b.



Hình 7-2

### TẦN SỐ CẮT

Tần số mà tại đó hệ số khuếch đại thế bằng 0.707 giá trị cực đại gọi là tần số cắt. Trên hình 7-1,  $f_1$  là tần số cắt thấp còn  $f_2$  là tần số cắt cao. Tần số cắt còn được gọi là tần số nửa công suất vì tại tần số cắt công suất ra chỉ bằng  $\frac{1}{2}$  công suất cực đại.

### MIDBAND

Chúng ta định nghĩa midband của một bộ khuếch đại là vùng của các tần số nằm giữa  $10f_1$  và  $0.1f_2$ . Trong midband hệ số khuếch đại thế xấp xỉ giá trị cực đại và ký hiệu là  $A_{mid}$ . Ba đặc trưng quan trọng của một mạch khuếch đại là  $A_{mid}$ ,  $f_1$  và  $f_2$ .

### OUTSIDE THE MIDBAND

Một mạch khuếch đại bình thường chỉ hoạt động trong vùng midband. Tuy nhiên chúng ta sẽ xem xét hệ số khuếch đại thế ở ngoài vùng midband. Đây là công thức tính A cho một mạch khuếch đại ac.

$$A = A_{mid} / ((1 + (f_1/f)^2) + (1 + f/f_2)^2)^{1/2} \quad (7-1)$$

Phương trình này giả sử rằng có 1 tụ nào đó quy định tần số cắt thấp  $f_1$  và một tụ khác quy định tần số cắt cao  $f_2$ .

Có thể thấy rằng tại midband  $A = A_{mid}$ .

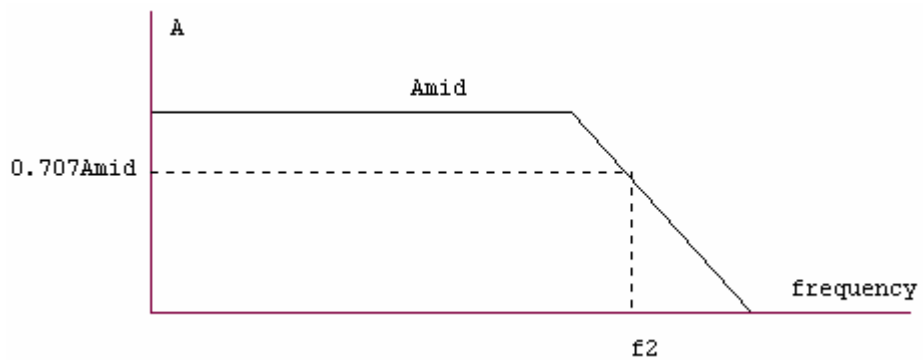
Trên midband

$$A = A_{mid} / (1 + f/f_2)^2 \quad (7-2)$$

Dưới midband

$$A = A_{mid} / (1 + f_1/f)^2 \quad (7-3)$$

## ĐÁP TUYẾN CỦA MỘT KHUYẾT ĐẠI DC



Hình 7-3

Hình 7-3 là đáp tuyến của một bộ khuếch đại dc. Do không có tần số cắt thấp  $f_1$  nên chỉ có 2 thông số của đáp tuyến tần số của bộ khuếch đại dc là  $A_{mid}$  và  $f_2$ .

Hầu hết các bộ khuếch đại thường dùng là bộ khuếch đại dc vì chúng được tích hợp trong IC. Đó là các bộ khuếch đại OPAMP. Đặc điểm của khuếch đại OP AMP là hệ số khuếch đại thế rất lớn, trở kháng vào cao, trở kháng ra thấp.

## VII.2 SỐ ĐO DECIBEL

Trong phần này chúng ta sẽ xem xét khái niệm decibel như một số đo thông số của mạch khuếch đại. Nhưng trước hết hãy ôn lại một chút lý thuyết toán học về logarit.

Cho phương trình

$$x = 10^y \quad (7-4)$$

Khi đó

$$y = \log_{10} x$$

Thông thường 10 được hiểu ngầm do đó có thể viết

$$y = \log x \quad (7-5)$$

Chẳng hạn

$$y = \log 10 = 1$$

$$y = \log 100 = 2$$

$$y = \log 1000 = 3$$

nếu  $x$  tăng 10 lần thì  $y$  tăng 1 lần.

$$y = \log 0.1 = -1$$

$$y = \log 0.01 = -2$$

$$y = \log 0.001 = -3$$

nếu x giảm 10 lần thì y giảm 1 lần.

### ĐỊNH NGHĨA HỆ SỐ $G_{dB}$

Chúng ta đã định nghĩa hệ số khuếch đại công suất

$$G = p_{out} / p_{in}$$

Hệ số khuếch đại công suất decibel được định nghĩa là

$$G_{dB} = 10 \log G \quad (7-6)$$

G là đại lượng không có thứ nguyên, do đó  $G_{dB}$  cũng không có thứ nguyên, nhưng để tránh nhầm lẫn giữa G và  $G_{dB}$  chúng ta thêm đơn vị decibel (dB) sau  $G_{dB}$ .

Bảng sau đây cho thấy quan hệ giữa A và  $A_{dB}$  cho một vài giá trị điển hình của G.

| G    | $G_{dB} = 10 \log G$ |
|------|----------------------|
| 1    | 0                    |
| 2    | +3                   |
| 0.5  | -3                   |
| 10   | +10                  |
| 0.01 | -10                  |
| 100  | +20                  |
| 1000 | +30                  |

### ĐỊNH NGHĨA HỆ SỐ $A_{dB}$

Chúng ta đã định nghĩa hệ số khuếch đại thế

$$A = v_{out} / v_{in}$$

Hệ số khuếch đại thế decibel được định nghĩa là

$$A_{dB} = 20 \log A \quad (7-7)$$

Lý do của việc sử dụng hệ số 20 thay cho hệ số 10 trong phương trình (7-7) là vì công suất tỷ lệ với bình phương hiệu điện thế. Theo (7-7) nếu một mạch khuếch đại có hệ số  $A=10^5$  thì hệ số khuếch đại thế decibel bằng

$$A_{dB} = 20 \log 10^5 = 100 \text{dB}$$

Bảng sau đây cho thấy quan hệ giữa A và  $A_{dB}$  cho một vài giá trị điển hình của A.

|      |                     |
|------|---------------------|
| A    | $A_{dB} = 20\log A$ |
| 1    | 0                   |
| 2    | +6                  |
| 0.5  | -6                  |
| 10   | +20                 |
| 0.01 | -20                 |
| 100  | +40                 |
| 1000 | +60                 |

Khi biết  $G_{dB}$  hoặc  $A_{dB}$  có thể tính ra hệ số khuếch đại công suất  $G$  và hệ số khuếch đại thế  $A$  theo phương trình sau

$$G = \text{antilog}(G_{dB}/10) \quad (7-8)$$

$$A = \text{antilog}(A_{dB}/20) \quad (7-9)$$

Đại lượng Decibel đôi khi còn dùng như một số đo chuẩn của công suất hoặc thế.

#### CÔNG SUẤT MILLIWATT

Decibel đôi khi được dùng để đo công suất lớn hơn 1 mW. Khi đó người ta ký hiệu là dBm (chữ m là viết tắt của miliwatt).

$$P_{dBm} = 10\log(P/1mW) \quad (7-10)$$

Ví dụ nếu công suất là 2W thì

$$P_{dBm} = 10\log(2W/1mW) = 10\log 2000 = 33dBm$$

Có thể tính ra  $P$  nếu biết dBm theo phương trình sau:

$$P = \text{antilog}(P_{dBm}/10) \quad (7-11)$$

(mW)

Bảng sau cho thấy quan hệ giữa công suất  $P$  và  $P_{dBm}$

| Power       | $P_{dBm}$ |
|-------------|-----------|
| 1 $\mu$ W   | -30       |
| 10 $\mu$ W  | -20       |
| 100 $\mu$ W | -10       |
| 1mW         | 0         |
| 10mW        | 10        |
| 100mW       | 20        |
| 1W          | 30        |

## THẾ CHUẨN 1VOLT

Decibel đôi khi cũng được dùng để đo mức thế lớn hơn 1V. Khi đó người ta ký hiệu là dBV (chữ V là viết tắt của Volt).

$$V_{dBV} = 20 \log V \quad (7-12)$$

Ví dụ nếu thế là 25V thì

$$V_{dBV} = 20 \log 25 = 28 \text{ dBV}$$

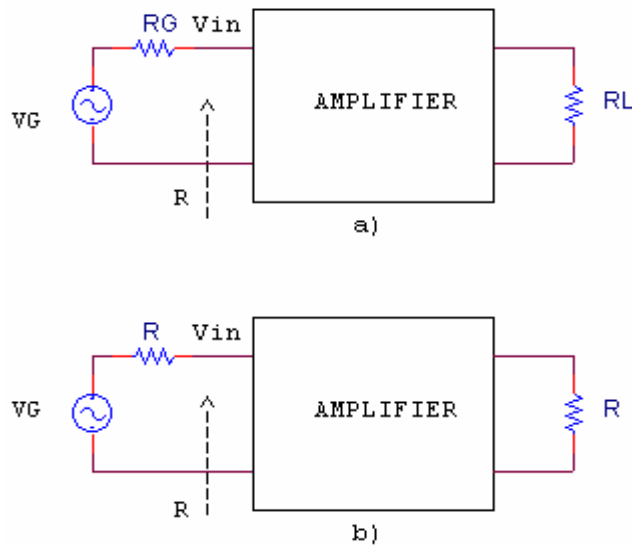
Có thể tính ra V nếu biết dBV theo phương trình sau:

$$V = \text{antilog} (V_{dBV} / 20) \quad (7-13)$$

Bảng sau cho thấy quan hệ giữa thế V và  $V_{dBV}$

| Voltage          | $V_{dBV}$ |
|------------------|-----------|
| $10\mu\text{V}$  | -50       |
| $100\mu\text{V}$ | -40       |
| 1mV              | -30       |
| 10mV             | -20       |
| 100mV            | -10       |
| 1V               | 0         |
| 10V              | +10       |
| 100V             | +20       |

## VII.3 PHỐI HỢP TRỞ KHÁNG



Hình 7-4

Hình 7-4a cho thấy một tầng khuếch đại có trở kháng nguồn là  $R_G$ , điện trở vào  $R_{in}$ , điện trở ra  $R_{out}$  và trở tải là  $R_L$ . Về nguyên tắc giá trị các trở kháng vào ra ấy là khác nhau.

Trong các hệ thống truyền tin như microware, television, telephone, network... thường có điều kiện phối hợp trở kháng, nghĩa là

$$R_G = R_{in} = R_{out} = R_L$$

Hình 7-4b mô tả ý tưởng này. Tất cả các điện trở bằng nhau và bằng  $R$ . Trở kháng  $R$  bằng  $50\Omega$  trong các hệ thống viba,  $75\Omega$  đối với cable đồng trục (cable mạng) hoặc  $300\Omega$  trong cable truyền hình hay  $600\Omega$  trong cable điện thoại. Sự phối hợp trở kháng được dùng trong các hệ thống này vì nó tạo ra công suất truyền tối đa.

Trên hình 7-4b, công suất vào bằng

$$p_{in} = v_{in}^2 / R$$

Công suất ra bằng

$$p_{out} = v_{out}^2 / R$$

Hệ số khuếch đại công suất

$$G = p_{out} / p_{in} = v_{out}^2 / v_{in}^2$$

$$\text{Vậy } G = A^2 \quad (7-14)$$

Theo (7-14) hệ số khuếch đại công suất bằng bình phương hệ số khuếch đại thế trong các hệ thống có phối hợp trở kháng.

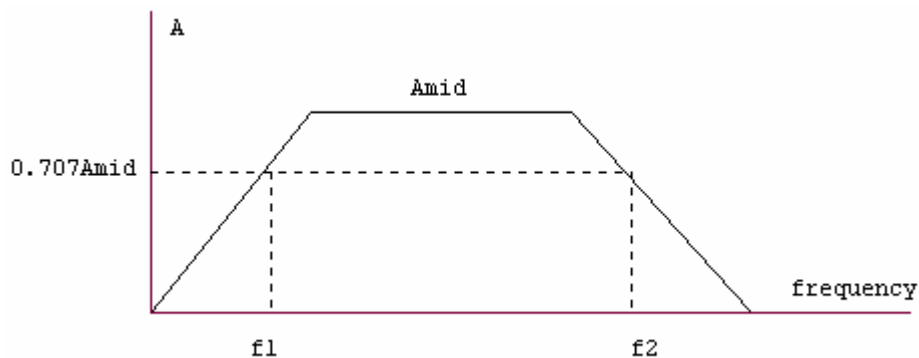
Biểu diễn bằng decibel thì

$$G_{dB} = 10 \log G = 10 \log A^2 = 20 \log A$$

$$\text{Suy ra } G_{dB} = A_{dB} \quad (7-15)$$

Theo (7-15), hệ số khuếch đại công suất decibel bằng hệ số khuếch đại thế decibel trong các hệ thống có phối hợp trở kháng.

## VII.4 GIẢN ĐỒ BODE



Hình 7-5



Hình 7-5 cho thấy đáp tuyến tần số của một bộ khuếch đại ac. Mặc dù nó chứa một số thông tin như hệ số khuếch đại tần số giữa và các tần số cắt nhưng nó không cho phép mô tả đầy đủ về hoạt động của một bộ khuếch đại. Biểu đồ Bode là công cụ cho nhiều thông tin hơn về hoạt động của một bộ khuếch đại ở ngoài vùng tần số giữa.

### OCTAVES

Trong hệ 2, một số đứng bên trái một số khác lớn hơn nó 2 lần. Trong âm nhạc, từ octave có nghĩa là gấp đôi về tần số. Khi di chuyển lên trên một octave chúng ta có tần số gấp đôi.

Trong điện tử, octave có nghĩa tương tự.

Khi  $f_1/f = 2$  chúng ta nói rằng  $f$  dưới  $f_1$  một octave.

Khi  $f/f_2 = 2$  chúng ta nói rằng  $f$  trên  $f_2$  một octave.

### DECADES

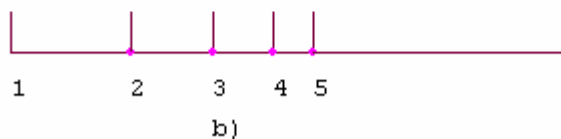
Decades có nghĩa tương tự octave nhưng dùng hệ số 10 thay vì hệ số 2.

Khi  $f_1/f = 10$  chúng ta nói rằng  $f$  dưới  $f_1$  một decade.

Khi  $f/f_2 = 10$  chúng ta nói rằng  $f$  trên  $f_2$  một decade.

### THANG ĐO TUYẾN TÍNH VÀ THANG ĐO LOGARIT

Các biểu đồ thông thường là biểu đồ dùng thang đo tuyến tính trên cả 2 trục. Điều này có nghĩa là khoảng cách giữa các số là giống nhau đối với mọi giá trị trên trục số như hình 7-6a. Trong thang đo tuyến tính các số bắt đầu từ số 0 và tăng tuyến tính đến các số lớn.



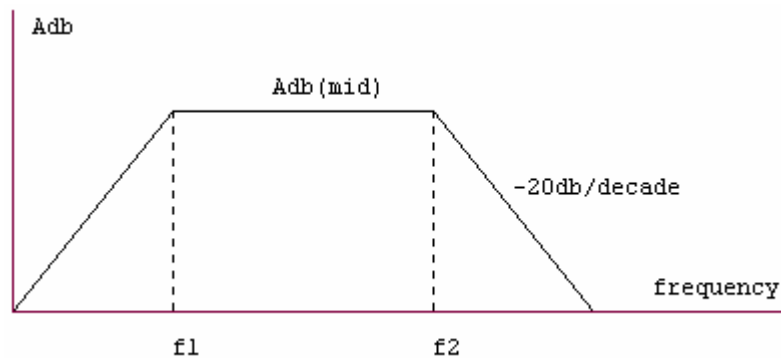
Hình 7-6

Đôi khi chúng ta dùng thang đo logarit vì nó nén các giá trị rất lớn của số liệu và cho phép chúng ta nhìn thấy nhiều decade. Hình 7-6b cho thấy một thang đo logarit. Lưu ý rằng số được đánh từ 1. Khoảng cách giữa 1 và 2 là lớn hơn khoảng cách giữa 9 và 10. Thang đo logarit thích hợp với các số đo

decibel. Ngoài thang đo tuyến tính và thang đo logarit người ta còn dùng thang đo bán logarit. Thang đo bán logarit dùng thang đo tuyến tính trên trục tung còn trục hoành dùng thang đo logarit. Chúng ta dùng thang đo bán logarit để biểu diễn các quan hệ như hệ số khuếch đại thế trên nhiều decade của tần số.

### GIẢN ĐỒ DECIBEL CỦA HỆ SỐ KHUYẾCH ĐẠI THỂ

Hình 7-7 chỉ ra đáp tuyến tần số của một bộ khuếch đại ac.



Hình 7-7

Giản đồ tương tự hình 7-5, nhưng chúng ta biểu thị hệ số khuếch đại bằng decibel theo tần số trên thang logarit. Giản đồ như vậy gọi là giản đồ Bode. Trục tung là thang đo tuyến tính còn trục hoành là thang đo logarit.

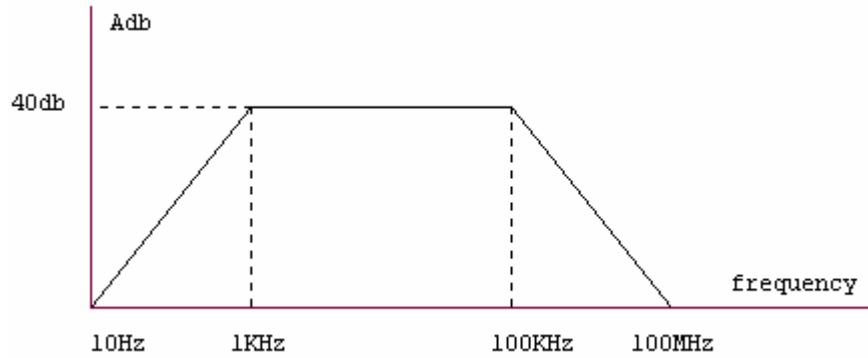
Theo giản đồ, hệ số khuếch đại decibel đạt giá trị cực đại ở giữa. Tại tần số cắt hệ số khuếch đại giảm và giảm với tốc độ 20dB/dec.

Tại tần số cắt, hệ số khuếch đại thế bằng 0.707. Do đó hệ số khuếch đại theo decibel tại tần số cắt bằng

$$A_{dB} = 20\log 0.707 = -3dB$$

Giản đồ Bode hình 7-7 là giản đồ Bode lý tưởng đã được lý tưởng hoá. Giản đồ Bode lý tưởng cho phép vẽ đáp tuyến tần số của một bộ khuếch đại nhanh và dễ dàng.

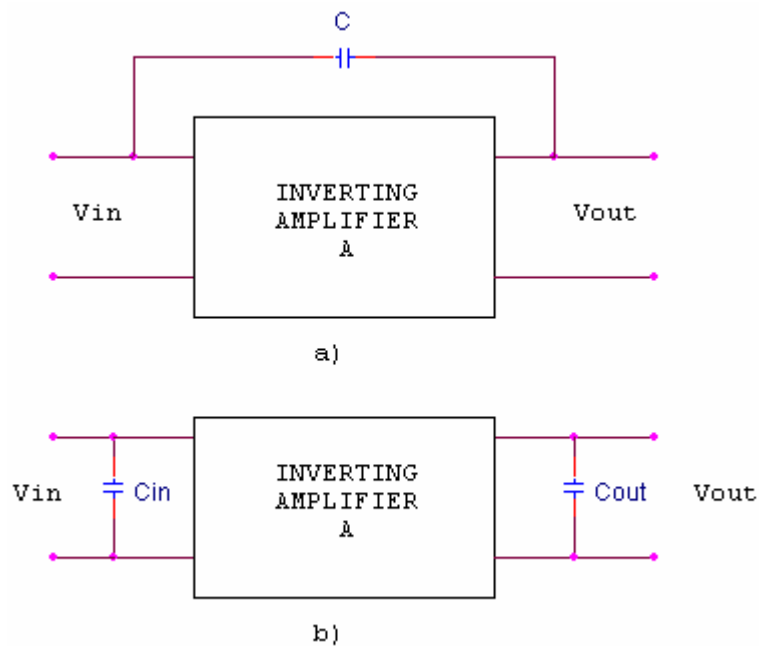
Ví dụ chúng ta có giản đồ Bode lý tưởng như hình 7-8. Chúng ta có thể thấy các thông tin sau đây. Hệ số khuếch đại giữa là 40dB. Tần số cắt là 1KHz và 100KHz. Độ dốc của đặc tuyến bên dưới và trên tần số cắt là 20dB/dec. Ngoài ra chúng ta cũng thấy rằng hệ số khuếch đại thế decibel bằng 0dB ( $A=1$ ) tại tần số 10Hz và 100MHz.



Hình 7-8

### VII.5 HIỆU ỨNG MILLER

Hình 7-9a chỉ ra một mạch khuếch đại đảo có hệ số khuếch đại thế bằng A. Trên hình 7-9a, tụ điện C nối giữa lối vào và ra của bộ khuếch đại gọi là tụ phản hồi bởi vì tín hiệu lối ra được đưa trở lại lối vào.



Hình 7-9

Mạch như thế rất khó phân tích vì tụ phản hồi ảnh hưởng đến cả mạch lối ra lẫn mạch lối vào.

Định lý Miller cho phép phân tích tụ phản hồi thành 2 tụ riêng rẽ như hình 7-9b. Trong đó

$$C_{in} = C(A+1) \quad (7-16)$$

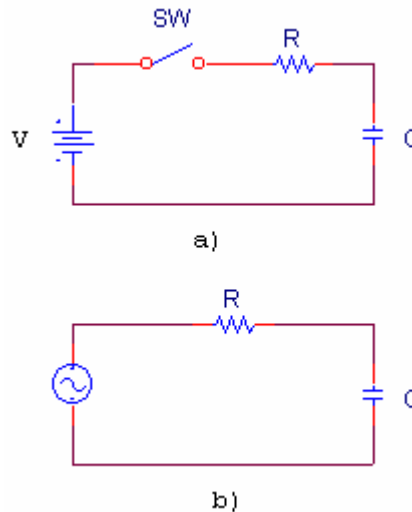
$$C_{out} = C(A+1)/A \quad (7-17)$$

Có thể thấy rằng  $C_{out} \approx C$  nhưng  $C_{in}$  lớn hơn tụ phản hồi  $A+1$  lần. Hiện tượng này gọi là hiệu ứng Miller. Nó được dùng để tạo ra một tụ ảo lớn hơn tụ phản hồi nhiều lần.

Mặt khác hiệu ứng Miller còn cho thấy rằng tụ phản hồi tạo ra một mạch trễ (lag circuit) ở lối vào của bộ khuếch đại. Chính tụ  $C_{in}$  là tụ chủ yếu quy định tần số cắt thấp của bộ khuếch đại.

### QUAN HỆ GIỮA BANDWIDTH VÀ THỜI GIAN TĂNG

Xét mạch RC ở đầu vào của bộ khuếch đại. Ban đầu tụ không tích điện như hình 7-10a.



Hình 7-10

Nếu đóng mạch tụ sẽ được nạp theo hàm mũ từ giá trị 0 đến nguồn cung cấp  $V$ . Thời gian tăng  $T_R$  của mạch RC được định nghĩa là thời gian mà thế trên tụ tăng từ 10% đến 90% giá trị cực đại.

Để dàng tính được

$$T_R = 2.2RC \quad (7-18)$$

Thời gian tăng lớn hơn thời hằng RC một chút.

Như đã nói trên, một bộ khuếch đại dc có một mạch trễ RC ở lối vào mà nó làm cho hệ số khuếch đại thế decibel giảm 20dB/dec. Tần số cắt của mạch RC này cho bởi

$$f_2 = 1/2\pi RC$$

Lưu ý đến phương trình (7-18) thì

$$f_2=0.35/T_R \quad (7-19)$$

Phương trình (7-19) là quan hệ giữa thời gian tăng và band thông của mạch. Trong mạch khuếch đại dc band thông có nghĩa là tất cả các tần số từ 0 đến tần số cắt. Band thông là cách nói khác của tần số cắt. Chẳng hạn nói một mạch khuếch đại dc có band thông 100KHz có nghĩa là tần số cắt của mạch cũng là 100KHz.

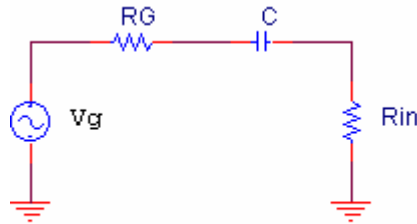
## VII.6 PHÂN TÍCH TẦN SỐ CỦA BỘ KHUYẾCH ĐẠI

Chúng ta sẽ phân tích ở đây các hiệu ứng tần số của một bộ khuếch đại CE phân cực bằng cầu chia thế.

Tần số cắt thấp  $f_l$  của bộ khuếch đại CE quy định bởi:

### a) Tụ nối vào

Khi một tín hiệu ac được đưa vào bộ khuếch đại, mạch tương đương như hình 7-11.



Hình 7-11

Mạch nối có tần số cắt bằng

$$f_l = 1/2\pi RC \quad (7-20)$$

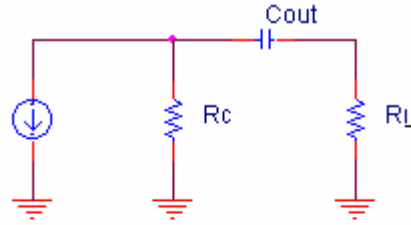
trong đó R gồm  $R_G$  và  $R_{in}$ .

### b) Tụ nối ra

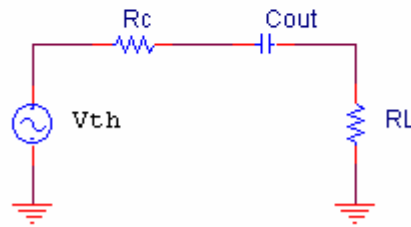
Hình 7-12a cho thấy mạch tương đương lối ra của một bộ khuếch đại.

Áp dụng định lý Thevenin chúng ta thu được mạch hình 7-12b.

Có thể dùng (7-20) để tính tần số cắt thấp, trong đó R bằng tổng của  $R_L$  và  $R_C$



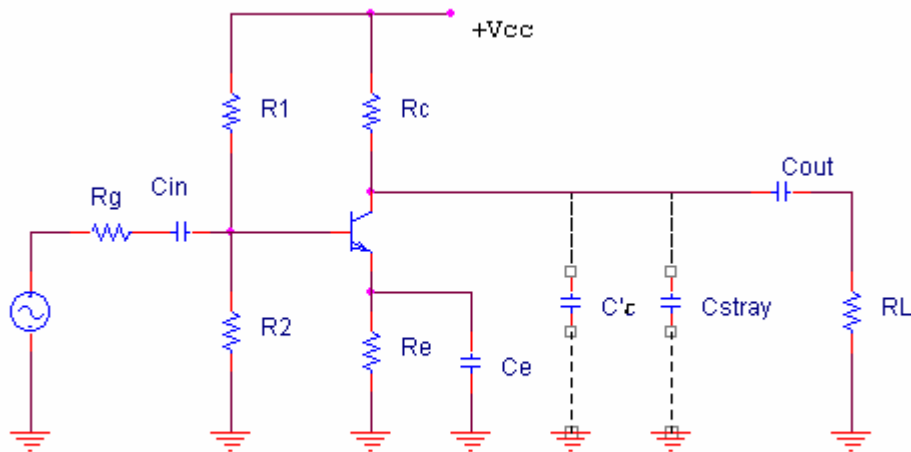
a)



b)

Hình 7-12

Tần số cắt cao  $f_2$  của khuếch đại CE quy định bởi các tụ ký sinh trên collector



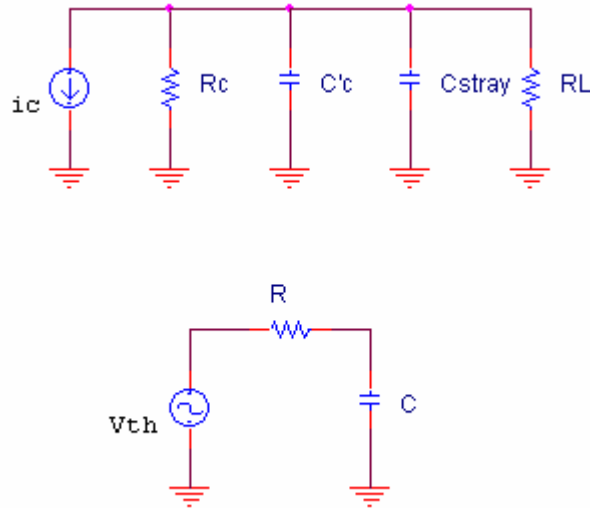
Hình 7-13

Hình 7-13 cho thấy một mạch CE với tụ ký sinh  $C_{stray}$ . Tụ  $C'_c$  là tụ giữa cực C và cực B bên trong transistor. Mặc dù  $C_{stray}$  và  $C'_c$  rất bé nhưng tại tần số cao chúng sẽ có ảnh hưởng.

Hình 7-14 là mạch tương đương Thevenin của mạch hình 7-13. Tần số cắt của mạch này là

$$f_2 = 1/2\pi RC \quad (7-22)$$

trong đó  $R = R_L // R_C$  và  $C = C_{\text{stray}} + C'_c$



Hình 7-14

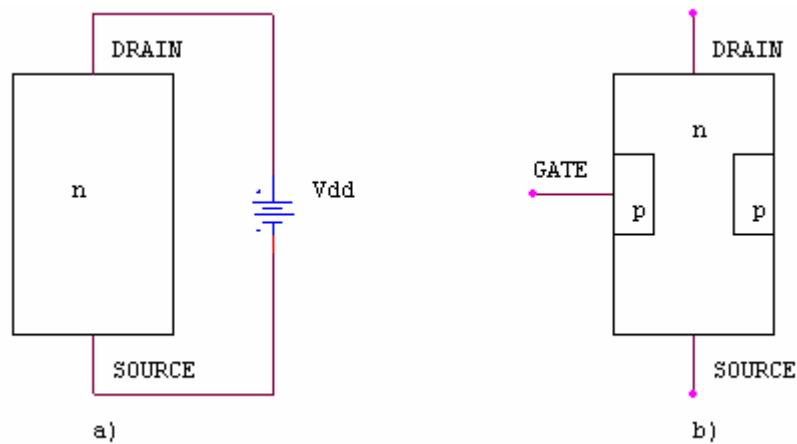
## Chương VIII **CÁC LINH KIỆN BÁN DẪN ĐẶC BIỆT**

Transistor lưỡng cực có 2 loại điện tích: electron tự do và lỗ trống. Điều này giải thích tại sao gọi là transistor lưỡng cực. Trong chương này chúng ta sẽ xem xét một loại transistor khác gọi là transistor hiệu ứng trường (field effect transistor- FET). Thiết bị này là đơn cực vì hoạt động của nó chỉ dựa trên một loại điện tích hoặc là electron tự do hoặc là lỗ trống. Nói cách khác FET chỉ có phần tử tải cơ bản mà không có phần tử tải không cơ bản.

Đối với hầu hết các ứng dụng tuyến tính, transistor thích hợp hơn. Nhưng có một số ứng dụng, FET lại có nhiều ưu điểm hơn do trở kháng vào cao và một số đặc trưng khác. Mặt khác trong các thiết bị số, FET lại tốt hơn transistor vì nó có tốc độ nhanh hơn transistor.

Có 2 loại transistor đơn cực là JFET và MOSFET. Trong chương này sẽ xem xét chúng.

### **VIII.1 JFET (Junction FET)**

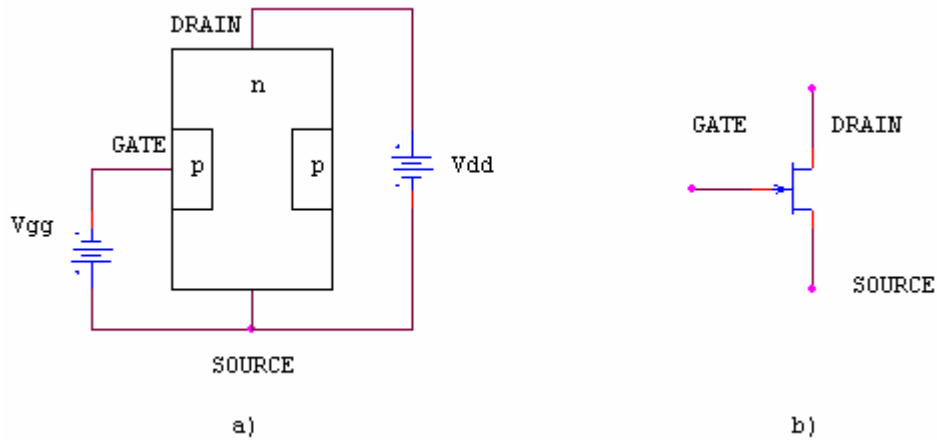


Hình 8-1

Hình 8-1a cho thấy một mẫu bán dẫn loại n. Đầu phía dưới gọi là nguồn (Source), đầu phía trên gọi là máng (Drain). Nguồn cung cấp  $V_{DD}$  buộc các electron chảy thành dòng từ nguồn đến máng. Để có JFET, nhà máy pha tạp 2 vùng p vào thanh bán dẫn loại n như hình 8-1b. Các vùng bán dẫn loại p này được nối ở bên trong và chỉ có một đầu ra gọi là cực cổng G.



Hình 8-2a cho thấy mạch phân cực cho JFET.



Hình 8-2

Đối với JFET, chúng ta luôn luôn phân cực ngược diode cổng nguồn. Do phân cực ngược, dòng cổng xấp xỉ bằng 0. Điều này có nghĩa là JFET có điện trở vào bằng vô cùng (thường là hàng trăm  $M\Omega$ ). Đây là ưu điểm của JFET so với transistor. Đó là lý do giải thích vì sao JFET là rất tốt trong các ứng dụng yêu cầu trở kháng vào cao, chẳng hạn bộ lặp lại nguồn (source follower).

#### THỂ GATE ĐIỀU KHIỂN DÒNG MÁNG

Trên hình 8-2a, các electron chảy từ nguồn đến máng phải qua kênh hẹp giữa các vùng nghèo. Khi thế phân cực âm hơn, lớp nghèo rộng ra và kênh dẫn của JFET hẹp hơn. Rõ ràng là bằng cách thay đổi thế  $V_{GS}$  có thể điều khiển dòng điện chảy giữa nguồn và máng.

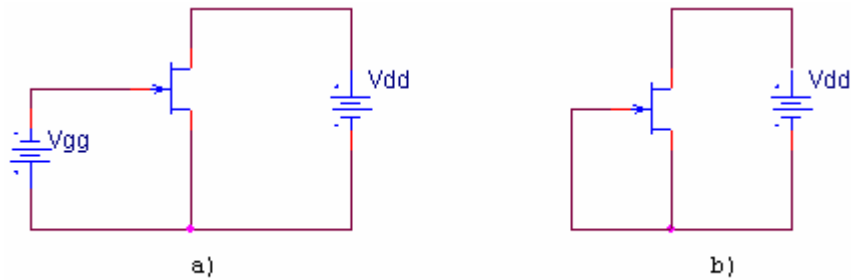
JFET là thiết bị điều khiển bởi điện thế vì thế vào điều khiển dòng ra. Trong JFET, thế  $V_{GS}$  quy định dòng máng nguồn. Khi  $V_{GS}$  bằng 0 dòng máng là cực đại. Khi  $V_{GS}$  đủ lớn hai vùng nghèo chạm nhau thì dòng máng bằng không.

Hình 8-2b là sơ đồ ký hiệu của JFET kênh n vì kênh dẫn giữa nguồn và máng là bán dẫn loại n. Trong các ứng dụng tần số thấp cực máng D và cực nguồn G là có thể đổi chỗ cho nhau. Nhưng trong các ứng dụng tần số cao thì không thể trao đổi D và G vì tụ tập tán giữa DG và DS là khác nhau đáng kể.

Ngoài JFET kênh n còn có JFET kênh p. Ký hiệu JFET kênh p như JFET kênh n nhưng mũi tên hướng ngược lại. Hoạt động của JFET kênh p là bổ túc với JFET kênh n.

## ĐƯỜNG CONG MÁNG

Hình 8-3a cho thấy một mạch phân cực JFET.



Hình 8-3

Trong mạch này thế  $V_{GS}$  bằng nguồn  $V_{GG}$  và thế nguồn máng  $V_{DS}$  bằng thế nguồn cực máng  $V_{DD}$ .

Cố định  $V_{GS}=0V$  (bằng cách ngắn mạch cực cổng như hình 8-3b) và bằng cách thay đổi thế  $V_{DS}$  chúng ta sẽ thu được đường cong  $I_{DS}(V_{DS})$ .

Khi tăng  $V_{DS}$  từ giá trị 0, dòng  $I_{DS}$  tăng tuyến tính. Khi  $V_{DS}=V_P$  thì dòng  $I_{DS}$  đạt đến giá trị bão hoà  $I_{DSS}$ . Khi  $V_{DS}$  vượt quá  $V_{DS(max)}$  dòng qua JFET tăng nhanh do nó bị đánh thủng. Giá trị  $V_P$  gọi là thế pinchoff.

Vùng tác động của JFET nằm giữa  $V_P$  và  $V_{DS(max)}$ . Trong vùng này JFET đóng vai trò một nguồn dòng có giá trị  $I_{DSS}$  khi  $V_{GS}=0$ .

Vùng giới hạn bởi thế  $V_P$  và  $V_{DS}=0$  gọi là vùng ohmic. Đó là vùng mà JFET hoạt động như một điện trở có giá trị bằng:

$$R_{DS} = V_P / I_{DSS}$$

Chẳng hạn nếu  $V_P=4V$  và  $I_{DSS}=10mA$  thì

$$R_{DS} = 4V / 10mA = 400\Omega$$

Khi JFET này hoạt động trong vùng Ohmic nó luôn luôn có điện trở bằng  $400\Omega$ .

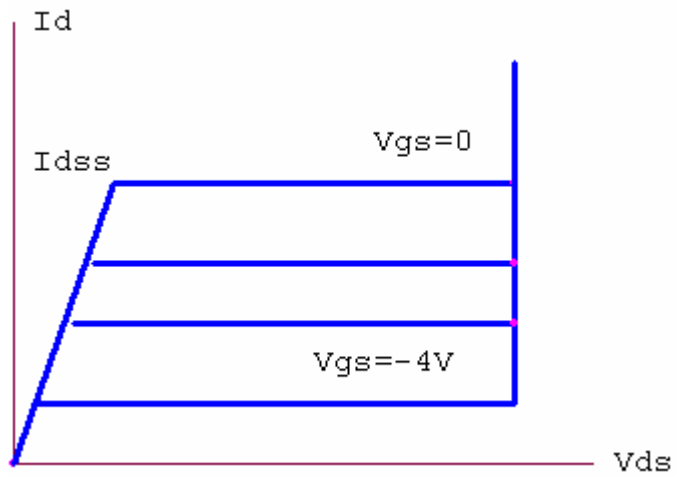
Hình 8-4 là họ đặc tuyến  $I_{DS}(V_{DS})$ .

Đường cong trên cùng ứng với  $V_{GS}=0$ . Đường cong dưới cùng ứng với trường hợp  $V_{GS} = -4V$ , tại đó dòng máng hầu như bằng 0. Thế này gọi là thế tắt cực cổng, ký hiệu là  $V_{GS(off)}$ .

Thế  $V_P=4V$  và thế  $V_{GS(off)} = -4V$

Điều này không phải ngẫu nhiên vì ứng với chúng, các vùng nghèo điện tích chạm nhau. Các bảng tra cứu cho một trong hai giá trị này và chúng ta suy ra giá trị còn lại nhờ phương trình

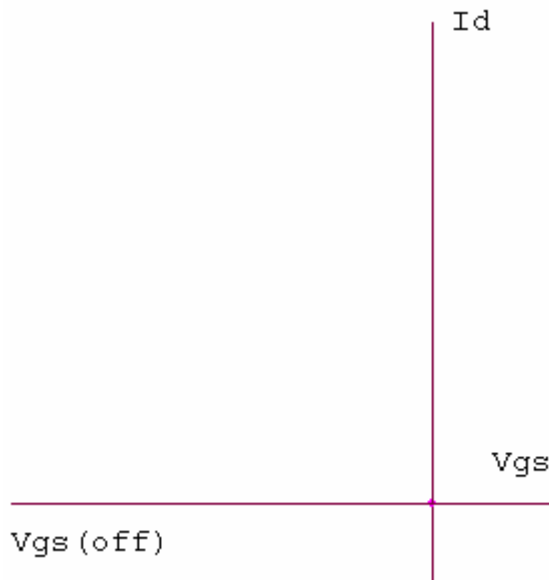
$$V_p = -V_{GS(off)} \quad (8-1)$$



Hình 8-4

### Đường cong truyền dẫn

Đường cong truyền dẫn của JFET là đường cong quan hệ  $I_D$  và  $V_{GS}$  như hình 8-5a.



Hình 8-5

Đường cong truyền dẫn của mọi JFET là giống nhau như hình 8-5. Các điểm đầu cuối của chúng là  $V_{GS(off)}$  và  $I_{DSS}$ .

Phương trình của đường cong này là:

$$I_D = I_{DSS}(1 - V_{GS} / V_{GS(off)})^2 \quad (8-2)$$

## VIII.2 PHÂN CỰC JFET

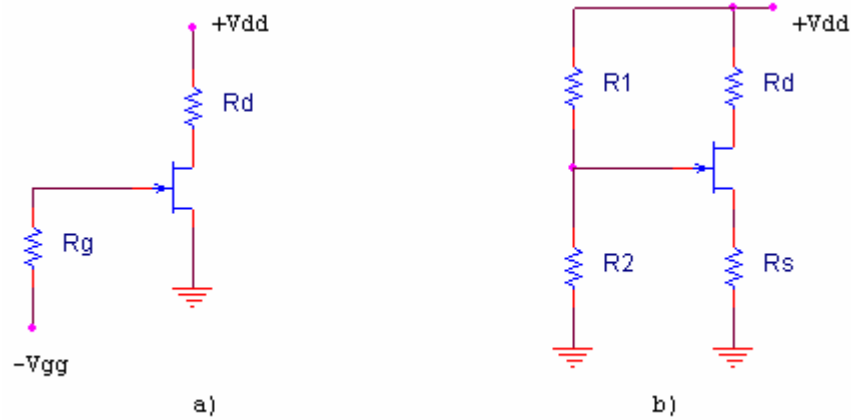
Hình 8-6a cho thấy mạch phân cực cực cổng. Một nguồn âm  $-V_{GG}$  áp vào cực cổng thông qua trở phân cực  $R_G$ . Điều này tạo ra dòng máng nhỏ hơn dòng máng bão hoà  $I_{DSS}$ . Thế máng bằng

$$V_D = V_{DD} - I_D R_D \quad (8-3)$$

Phân cực cực cổng là phân cực kém ổn định nhất. Tuy nhiên người ta hay dùng cách phân cực này khi sử dụng JFET trong vùng điện trở.

### PHÂN CỰC TRONG VÙNG TÁC ĐỘNG

Hình 8-5b cho thấy mạch phân cực bằng cầu chia thế.



Hình 8-6

Thế trên điện trở nguồn  $R_S$  bằng

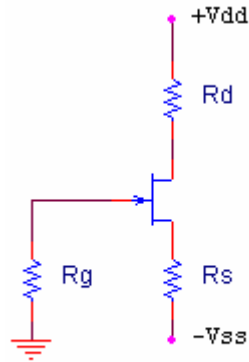
$$V_S = V_G - V_{GS} \quad (8-4)$$

Dòng máng

$$I_D = (V_G - V_{GS}) / R_S \approx V_G / R_S \quad (8-5)$$

## PHÂN CỰC BẰNG NGUỒN ĐÔI

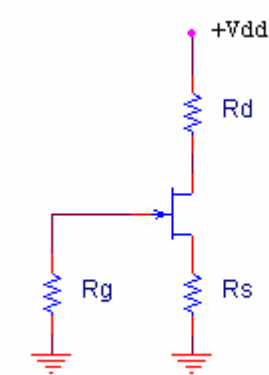
Hình 8-7 cho thấy mạch phân cực cho JFET bằng nguồn đôi.  
Dòng máng cho bởi



Hình 8-7

$$\begin{aligned} I_D &= (V_{SS} - V_{GS}) / R_S \\ &= V_{SS} / R_S \end{aligned} \quad (8-6)$$

Theo (8-6) dòng máng gần như không phụ thuộc nhiệt độ và bản thân JFET.



Hình 8-8

## TỰ PHÂN CỰC

Hình 8-8 cho thấy mạch tự phân cực của JFET.

Dòng máng tạo ra thế trên cực nguồn bằng

$$V_S = I_D R_S \quad (8-7)$$

Do đó thế

$$V_{GS} = - I_D R_S \quad (8-8)$$

## HỆ SỐ TRUYỀN DẪN CỦA JFET

Để phân tích một mạch khuếch đại dùng JFET chúng ta định nghĩa hệ số truyền dẫn (Transconductance) của JFET như sau:

$$g_m = i_d / v_{gs} \quad (8-8)$$

Trong đó  $i_d$  là dòng máng ac còn  $v_{gs}$  là thế cổng nguồn ac.

$g_m$  cho thấy ảnh hưởng của thế nguồn cổng trong việc điều khiển dòng máng. Rõ ràng rằng  $g_m$  càng lớn thì với  $v_{gs}$  cho trước có thể tạo dòng máng lớn hơn.

Ví dụ nếu  $i_d=0.2\text{mA}$  và  $v_{gs}=0.1\text{Vpp}$  thì

$$g_m = 0.2\text{mA} / 0.1\text{V} = 2 \cdot 10^{-3} \text{ mho} = 2000 \mu\text{S}$$

Đơn vị của hệ số truyền dẫn là mho hay Siemen (S)

Liên hệ giữa  $g_{m0}$  và  $V_{GS(\text{off})}$  của một JFET như sau

$$V_{GS(\text{off})} = -2I_{DSS} / g_{m0} \quad (8-10)$$

Trong đó  $g_{m0}$  là  $g_m$  tại  $V_{GS}=0$ .

Hệ số truyền dẫn tại  $V_{GS}$  bất kỳ tính theo công thức

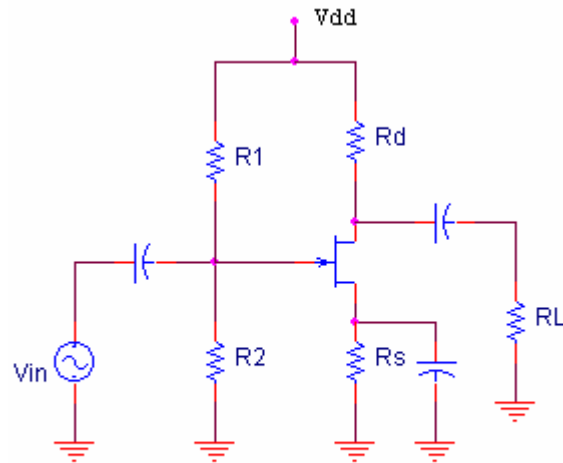
$$g_m = g_{m0} (1 - V_{GS} / V_{GS(\text{off})}) \quad (8-11)$$

### VIII.3 KHUYẾCH ĐẠI DÙNG JFET

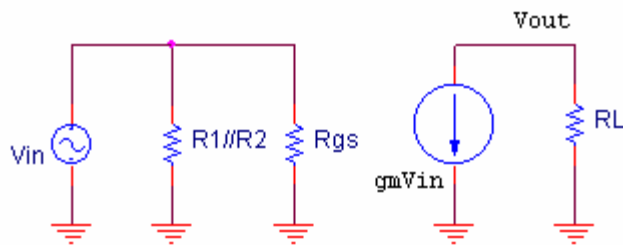
Hình 8-9a cho thấy một mạch khuếch đại dùng JFET kiểu nguồn chung CS. Ở chế độ ac các tụ là ngắn mạch. Nguồn ac vào được đặt lên các cực GS. Điều này tạo ra dòng máng ac. Dòng máng ac tạo sụt thế trên trở máng  $R_D$ . Thế trên cực máng của JFET là đã được khuếch đại đảo pha so với thế vào.

Hình 8-9b cho thấy mạch tương đương ac. Điện trở máng  $r_d$  bằng

$$r_d = R_D // R_L$$



a)



b)

Hình 8-9

Hệ số khuếch đại thế bằng

$$A = v_{out} / v_{in} = g_m v_{in} r_d / v_{in}$$

$$= g_m r_d \quad (8-12)$$

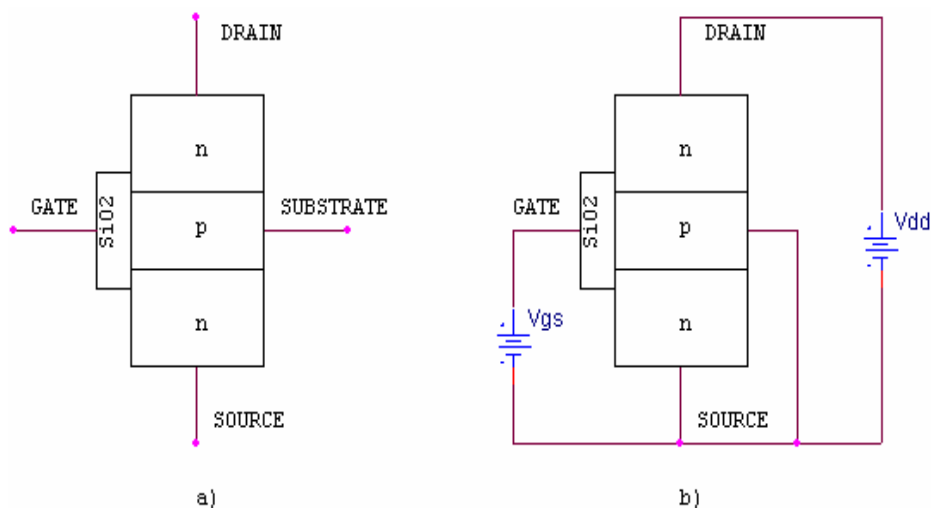
JFET thường dùng trong các mạch khuếch đại lỗi vào nhiễu thấp, các bộ đệm. JFET cũng thường dùng trong các chuyển mạch số, các bộ dồn và tách kênh. Một số ứng dụng dùng FET như một điện trở điều khiển bởi điện thế (mạch AGC).

### VIII.4 MOSFET

FET bán dẫn oxit kim loại (Metal Oxide Semiconductor FET) là thiết bị 3 cực: Cổng, Nguồn và Máng như FET. Nhưng cực cổng của MOSFET cách điện hoàn toàn với kênh dẫn. Do đó dòng cổng của MOSFET nhỏ hơn dòng cổng của JFET.

Có 2 loại MOSFET là MOSFET chế độ hiếm và MOSFET chế độ cải tiến. MOSFET chế độ cải tiến được dùng rộng rãi trong mạch rời cũng như mạch tích hợp. Trong mạch tích hợp MOSFET dùng chủ yếu như digital switch, một quá trình cơ bản trong máy tính.

Trong phần này chỉ trình bày MOSFET cải tiến, còn gọi là E-MOSFET. Hình 8-10a cho thấy một E-MOSFET.

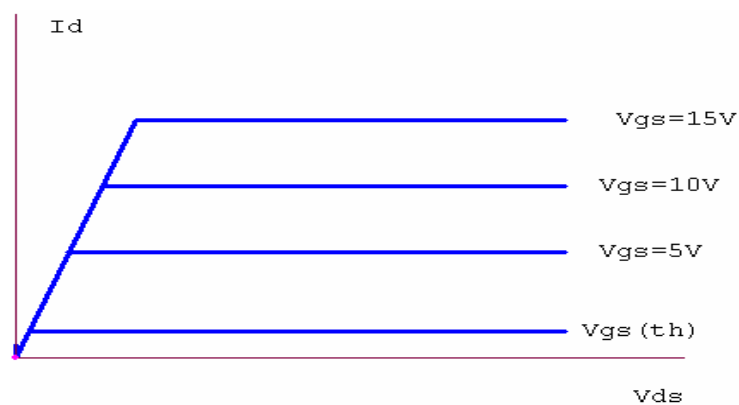


Hình 8-10



Cực cổng là cách điện hoàn toàn với kênh dẫn bằng  $\text{SiO}_2$ . Vùng P gọi là Substrate. Khi phân cực Substrate được nối với cực nguồn. E-MOSFET dùng thế cổng nguồn  $V_{GS}$  dương. Khi  $V_{GS}=0$  kênh dẫn giữa S và D bị nghẽn. Khi  $V_{GS}$  dương, nó hút các electron tự do vào vùng P. Tại đây chúng bị tái hợp với lỗ trống bên cạnh lớp silicon oxide. Khi  $V_{GS}$  đủ lớn, nó làm các lỗ trống gần lớp oxide silicon bị lấp đầy và do đó tạo ra một lớp bán dẫn loại n. Khi nó tồn tại các electron có thể chảy từ cực nguồn đến cực máng tạo ra dòng máng. Thế  $V_{GS}$  tối thiểu tạo ra lớp bán dẫn loại n gọi là thế ngưỡng, ký hiệu là  $V_{GS(th)}$ . Giá trị của  $V_{GS(th)}$  thường là từ 1 đến 3V.

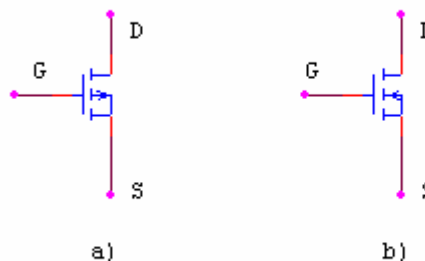
Hình 8-11 là đặc tuyến  $I_D(V_{DS})$  của E-MOSFET tín hiệu bé.



Hình 8-11

Đường cong dưới cùng ứng với  $V_{GS(th)}$ . Khi  $V_{GS} < V_{GS(th)}$  dòng máng bằng 0. Khi  $V_{GS} > V_{GS(th)}$  E-MOSFET dẫn và dòng máng phụ thuộc  $V_{GS}$ . Mặc dù có thể sử dụng E-MOSFET trong vùng tác động, người ta thường dùng E-MOSFET chủ yếu trong vùng điện trở.

**Ký hiệu sơ đồ:** E-MOSFET gồm 2 loại: kênh n và kênh p ký hiệu như hình vẽ 8-12.



Hình 8-12

**Thế cổng nguồn tối đa.** MOSFET có một lớp oxide silicon mỏng đóng vai trò cách điện giữa cổng và kênh dẫn. Khi  $V_{GS}$  vượt quá một giá trị nào đó có thể đánh thủng sự cách điện này và MOSFET bị hỏng. Chẳng hạn 2N7000 có  $V_{GS(max)}=20V$ . Nếu thế  $V_{GS}$  dương hơn hoặc âm hơn 20V thì có thể phá huỷ 2N7000. Các xung chuyển khi cắm hoặc tháo MOSFET vào thiết bị đang có nguồn có thể làm hỏng MOSFET. Thậm chí điện tích tĩnh (electrostatic) cũng có thể vượt quá  $V_{GS(max)}$  và làm hỏng MOSFET khi chúng ta cầm chúng trên tay. Vì vậy các thiết bị MOSFET cần phải được chống tĩnh điện (bằng vỏ chống tĩnh điện) và không được cắm nóng (hot plug).

**Điện trở máng nguồn khi dẫn.** Khi MOSFET được phân cực trong vùng điện trở, nó tương đương với một điện trở  $R_{DS(on)}$ . Các tài liệu kỹ thuật thường cho  $R_{DS(on)}$  ứng với một giá trị của  $I_D$  và thế  $V_{GS}$ . Khi đó

$$R_{DS(on)} = V_{DS(on)} / I_D(on) \quad (8-13)$$

Bảng 8-1 cho thấy một số các số liệu của vài MOSFET

| Device  | $V_{GS(th)}$<br>(V) | $V_{GS(on)}$<br>(V) | $I_D(on)$<br>(mA) | $R_{DS(on)}$<br>$\Omega$ | $I_D(max)$<br>(mA) | $P_D(max)$<br>(mW) |
|---------|---------------------|---------------------|-------------------|--------------------------|--------------------|--------------------|
| VN2406L | 1.5                 | 2.5                 | 100               | 10                       | 200                | 350                |
| BS107   | 1.75                | 2.6                 | 20                | 28                       | 250                | 350                |
| 2N7000  | 2                   | 4.5                 | 75                | 6                        | 200                | 350                |
| VN10LM  | 2.5                 | 5                   | 200               | 7.5                      | 300                | 1000               |
| MPF830  | 2.5                 | 10                  | 1000              | 0.8                      | 2000               | 1000               |
| IRFD120 | 3                   | 10                  | 600               | 0.3                      | 800                | 1000               |

## VIII.5 CHUYỂN MẠCH SỐ

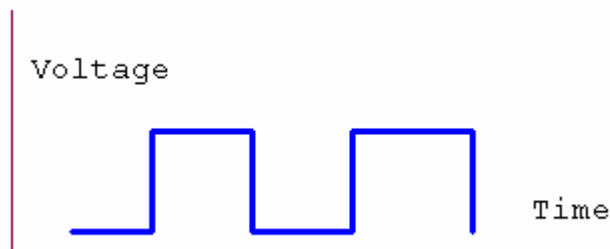
MOS đã làm nên cuộc cách mạng trong công nghiệp máy tính. Trước khi có MOS, máy tính dùng đèn hoặc transistor có kích thước lớn, tiêu tốn nhiều năng lượng. Với thế ngưỡng bé và dòng vào bằng 0, MOS là thiết bị chuyển mạch lý tưởng. Khi thế vào lớn hơn thế ngưỡng MOS chuyển từ Off sang bão hoà. Sự chuyển giữa 2 trạng thái on/off là ý tưởng chính để xây dựng máy tính. Máy tính dùng hàng chục triệu MOS như là các công tắc để xử lý số

liệu (số liệu là tất cả các số, chữ, đồ họa và những thông tin khác được mã hoá dưới dạng nhị phân).

### MẠCH TƯƠNG TỰ, MẠCH SỐ VÀ MẠCH CÔNG TẮC

Từ Analog có nghĩa là tương tự hay liên tục. Chẳng hạn sóng sin. Khi chúng ta nói tín hiệu analog có nghĩa là tín hiệu ấy biến thiên liên tục theo thời gian.

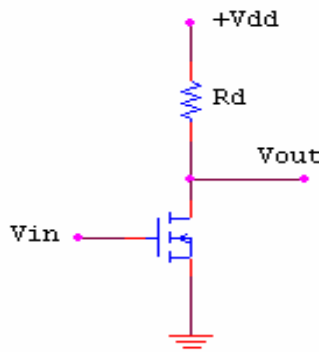
Từ Digital có nghĩa là số hay rời rạc. Tín hiệu digital là các tín hiệu không liên tục. Chúng biến thiên gián đoạn hay có sự chuyển từ giá trị cao sang thấp một cách đột ngột như hình 8-13. Máy tính dùng các tín hiệu này để mã hoá số, chữ và các thông tin khác.



Hình 8-13

Từ Switching có nghĩa rộng hơn digital. Nó có nghĩa là các mạch chuyển (công tắc). Các mạch chuyển (công tắc) bao gồm các mạch số như là tập con. Mạch công tắc cũng bao hàm mạch để điều khiển motor, heater...

### PASSIVE LOAD SWITCHING



Hình 8-14

Hình 8-14 cho thấy một mạch công tắc tải thụ động (tải là điện trở  $R_D$  thông thường). Trong mạch này  $V_{in}$  có 2 trạng thái: cao và thấp. Khi  $V_{in}$  thấp,

MOS ngưng dẫn và  $V_{out} = V_{DD}$ . Khi vin cao, MOS bão hòa và  $v_{out}$  xấp xỉ 0V.  
Để mạch hoạt động đúng cần có điều kiện

$$R_{DS(on)} \ll R_D$$

Mạch hình 8-14 là mạch đơn giản nhất trong máy tính. Đó là mạch đảo (inverter gate) vì thế lối ra đảo ngược với thế lối vào. Giá trị chính xác của mạch không quan trọng mà quan trọng là thế ra có 2 giá trị cao và thấp phân biệt được.

#### ACTIVE LOAD SWITCHING

Mạch tích hợp IC chứa hàng ngàn transistor nhỏ hoặc là transistor lưỡng cực hoặc là MOS. Các thế hệ IC đầu tiên có mạch công tắc với tải thụ động. Nhưng chúng có nhược điểm là kích thước lớn. Công tắc có tải động khắc phục nhược điểm này. Hình 8-15a cho thấy mạch công tắc tải động. MOS phía dưới là công tắc. MOS phía trên là tải động. Điện trở của MOS trên bằng

$$R_D = V_{DS(active)} / I_{D(active)} \quad (8-14)$$

Trong đó  $V_{DS(active)}$  và  $I_{D(active)}$  là thế và dòng của MOS trên vùng active.

Để mạch hoạt động đúng  $R_D$  của MOS trên phải lớn hơn  $R_{DS(on)}$  của MOS dưới.

Hình 8-15b cho thấy cách tính  $R_D$  của MOS trên.

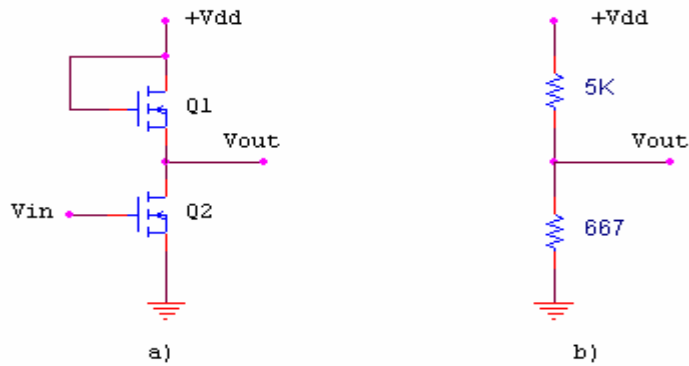
Vì  $V_{GS}=V_{DS}$  nên đặc tuyến của MOS trên là đường thẳng.

Có thể thấy rằng tại các điểm ứng với  $V_{DS}=15V$ , 10V và 5V các dòng  $I_D$  tương ứng là 3mA, 1.6mA và 0.7mA do đó trở  $R_D$  tương ứng bằng 5K $\Omega$ , 6.25K $\Omega$  và 7.2K $\Omega$ .

Nếu MOS dưới có tham số như MOS trên thì

$$R_{D(on)} = 2V/3mA = 667 \Omega.$$

Có thể thấy rằng tại mọi điểm hoạt động, thế ra khi MOS dưới on là rất thấp (xấp xỉ 0V).

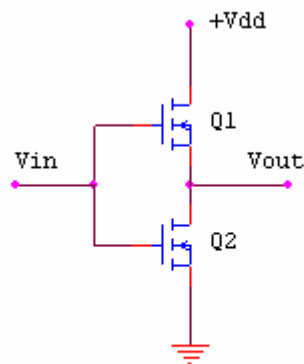


Hình 8-15

## VIII.6 CMOS

Trong công tắc tải động, dòng tải khi lối ra thấp là  $I_{D(sat)}$ . Dòng này khá lớn và vì vậy không thích hợp với các thiết bị hoạt động bằng pin. Một cách để giảm dòng là dùng CMOS (complementary MOS) như hình 8-16. CMOS gồm một MOS kênh n và một MOS kênh p. Chúng được gọi là bổ túc vì có các thông số dòng, thế bằng và đối nghịch như  $V_{GS(th)}$ ,  $V_{GS(on)}$ ,  $I_{D(on)}$ .

Mạch này tương tự mạch khuếch đại đẩy kéo hạng B vì một MOS dẫn thì MOS kia ngưng dẫn.

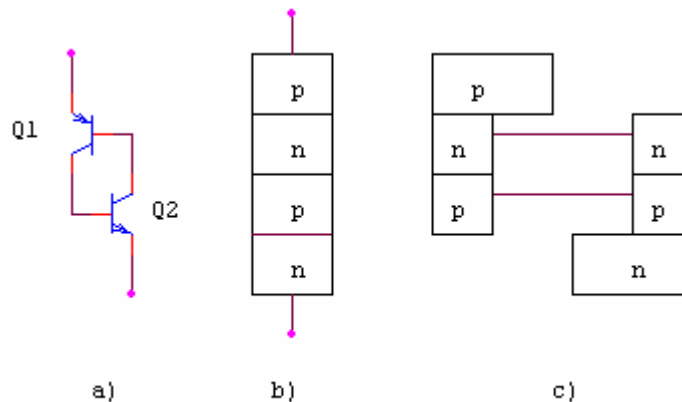


Hình 8-16

Khi CMOS là công tắc thì thế vào là từ 0 đến  $V_{DD}$ . Khi thế vào cao, Q1 ngưng còn Q2 dẫn, thế ra bằng 0V. Khi thế vào thấp, Q1 dẫn còn Q2 ngưng, thế ra cao.

Ưu điểm chủ yếu của CMOS là công suất tiêu tán rất bé. Khi thế vào bằng  $\frac{1}{2}$  thế  $V_{DD}$  cả hai MOS đều dẫn (thế ra bằng  $\frac{1}{2}V_{DD}$ ) và dòng máng từ nguồn là lớn nhất. Tuy nhiên thời gian này rất ngắn nên công suất tiêu tán động trung bình rất bé. CMOS thích hợp cho các hệ thống hoạt động bằng pin (chẳng hạn RTC của máy tính).

## VIII.7 THYRISTOR VÀ ỨNG DỤNG



Hình 8-17

Thyristor có nguồn gốc từ chữ Hylạp là cửa mở ra cho một cái gì đó đi qua. Thyristor là thiết bị bán dẫn sử dụng phản hồi bên trong để tạo ra một tác động như công tắc. Thyristor quan trọng nhất là SCR (Silicon Controlled Rectifier) và Triac. Chúng được sử dụng như các chuyển mạch có dòng lớn, tốc độ cao. Thyristor dùng nhiều trong các mạch điều khiển motor, heater, các hệ thống ánh sáng và các thiết bị có dòng tải lớn.

### DIODE 4 LỚP

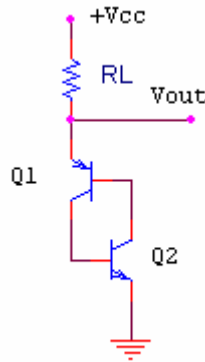
Hình 8-17a cho thấy mạch tương đương của diode 4 lớp. Diode 4 lớp có tên gọi nguyên thủy là diode Shockley, diode pnpn và Silicon Unilateral Switch (SUS).

Sơ đồ nối bên trong diode bốn lớp tạo ra phản hồi dương mà điều này có nghĩa là: nếu dòng của một trong hai transistor tăng sẽ làm cho cả hai bão

hòa, ngược lại nếu dòng của một trong hai transistor giảm sẽ làm cho cả hai transistor ngưng dẫn.

Nếu cả hai transistor bão hoà diode 4 lớp coi như công tắc đóng.

Nếu cả 2 transistor ngưng dẫn, diode 4 lớp coi như công tắc hở. Cả hai trạng thái này là bền. Vì mạch có thể ở một trong hai trạng thái bền nên mạch gọi là mạch LACTH (chốt)



Hình 8-18

### **Đóng mạch chốt**

Hình 8-18 cho thấy một mạch chốt nối với tải đến nguồn  $V_{CC}$ . Giả sử rằng mạch chốt đang hở. Chỉ có một cách để đóng mạch chốt là dùng Breakover. Điều này có nghĩa là dùng một nguồn  $V_{CC}$  đủ lớn để đánh thủng diode collector Q1. Khi đó do phản hồi dương cả hai transistor sẽ bão hoà. Khi mạch chốt đóng sụt thế qua diode 4 lớp gần bằng không.

### **Mở mạch chốt**

Khi mạch chốt đã đóng, để mở mạch chốt cần giảm nguồn cung cấp  $V_{CC}$ . Điều này buộc cả hai transistor ngưng dẫn và mạch chốt hở.

### **SCR (Silicon Controlled Rectifier)**

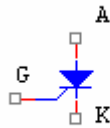
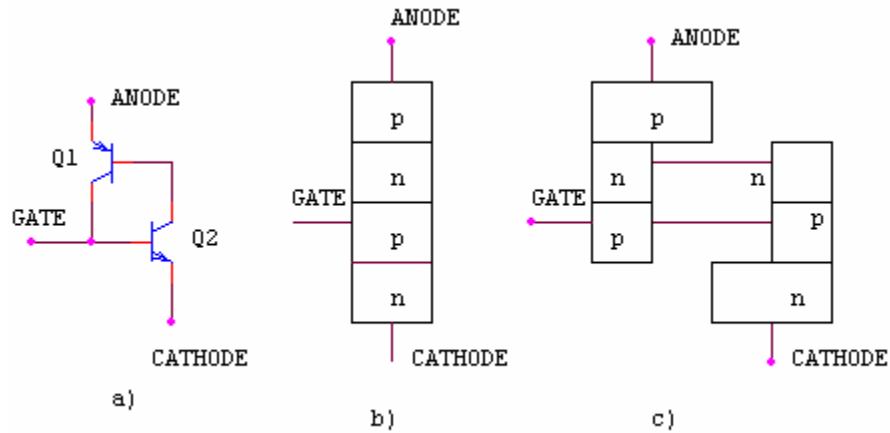
Trong diode 4 lớp, bằng cách nối base của Q2 ra ngoài (ký hiệu là G hay cực cổng) chúng ta có SCR.

Cực cổng có khả năng điều khiển hoạt động của SCR. Khi mạch chốt đang hở, nếu chúng ta đưa vào cực G một xung dương thì mạch chốt sẽ đóng ngay cả khi xung kích thích mất đi. Nhà máy thường chỉ ra dòng tối thiểu để kích SCR bằng tham số  $I_{GT}$ . Thế kích SCR là  $V_{GT} = 0.7V$ .

Thế vào tối thiểu để kích SCR là

$$V_{in} = V_{GT} + I_{GT}R_G \quad (8-15)$$

Trong đó  $R_G$  là điện trở Thevenin của mạch kích SCR.



Hình 8-19

### RESET SCR

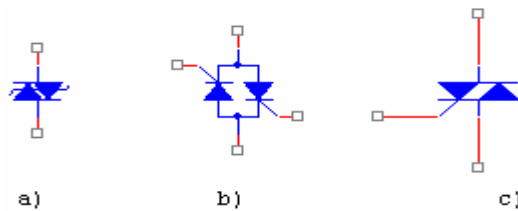
Sau khi SCR dẫn, nó vẫn tiếp tục dẫn ngay cả khi  $V_{in}=0$ . Chỉ có một cách để reset SCR là giảm dòng của SCR đến dưới giá trị dòng giữ (holding current). Nói cách khác thế cung cấp phải bé hơn

$$V_{CC} = 0.7V + I_H R_L \quad (8-16)$$

Trong đó  $I_H$  là dòng giữ.

### DIAC và TRIAC

Diac là 2 diode 4 lớp ghép đối song (hình 8-20a). Còn Triac là 2 SCR ghép đối song (hình 8-20b). Chúng được gọi là các Thyritor lưỡng hướng vì có thể cho dòng qua cả hai hướng.



Hình 8-20



## Chương IX **KHUYẾCH ĐẠI THUẬT TOÁN**

### **IX.1 KHUYẾCH ĐẠI VI SAI**

Thuật ngữ khuếch đại thuật toán (OP AMP) dùng để chỉ các bộ khuếch đại mà chúng thực hiện các phép toán. Về mặt lịch sử các bộ khuếch đại thuật toán được dùng trong các máy tính analog, ở đó chúng thực hiện các phép toán cộng, trừ... Vào thời kỳ đó, khuếch đại OP AMP được lắp bằng linh kiện rời. Ngày nay OP AMP được chế tạo dưới dạng IC.

Một mạch OP AMP là mạch khuếch đại DC với hệ số khuếch đại rất lớn, trở kháng vào rất cao và trở kháng ra rất thấp. Trong khuếch đại OP AMP hệ số khuếch đại là hằng số từ 1Hz đến hơn 20MHz. Một vi mạch OP AMP là một khối chức năng đầy đủ với một số các chân bên ngoài. Bằng cách nối nguồn nuôi và 1 số các linh kiện khác, chúng ta có thể xây dựng nhanh tất cả các loại mạch thông thường.

Trong vi mạch chỉ có transistor, diode, điện trở và cả tụ (nhỏ hơn 50pF). Vì lý do này các nhà thiết kế IC không dùng tụ nối và tụ thông dẫn như cách thiết kế mạch rời. Thay vào đó họ dùng các nối trực tiếp giữa các tầng.

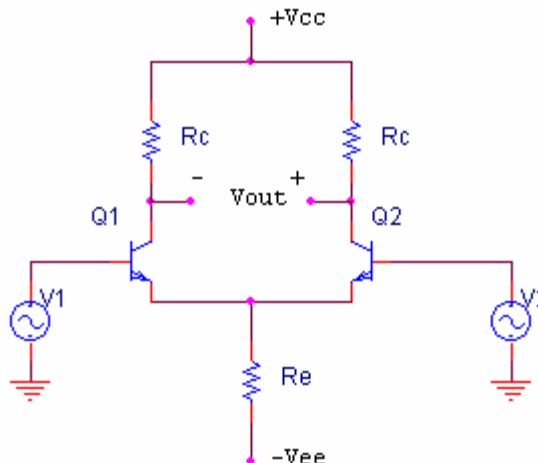
Khuếch đại vi sai có cùng ý tưởng như vậy. Mạch khuếch đại vi sai hạn chế được tụ thông dẫn emitter. Vì lý do này và một vài lý do khác mạch khuếch đại vi sai được dùng như là tầng vào của OP AMP.

#### **LỐI VÀO VÀ LỐI RA VI SAI**

Hình 9-1 cho thấy một mạch khuếch đại vi sai.

Hình 9-1

Nó gồm 2 tầng khuếch đại CE có điện trở emitter chung. Mặc dù có 2 lối vào và 2 lối ra nhưng về tổng thể mạch xem như một tầng khuếch đại. Vì không sử dụng tụ nối nên tần số cắt thấp  $f_1=0$ .



Thế ra  $v_{out}$  là hiệu thế giữa 2 collector với cực tính như trên hình 9-1.

$$V_{out} = V_{c2} - V_{c1} \quad (9-1)$$

Thế ra này gọi là thế ra vi sai vì nó là tổ hợp của 2 thế ra ac thành một thế bằng hiệu thế của 2 collector.

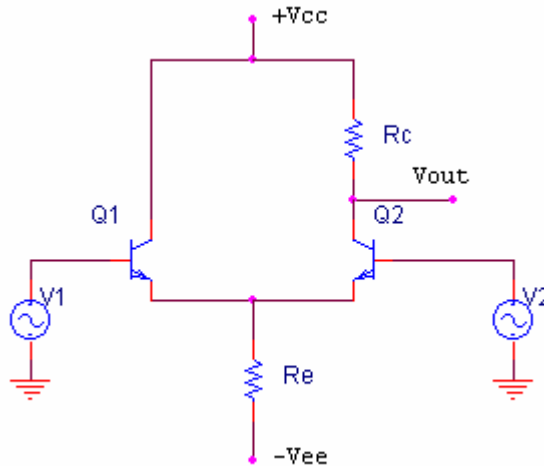
Một cách lý tưởng thì 2 transistor và các điện trở collector phải giống hệt nhau. Do đó  $v_{out} = 0$  khi  $v_1 = v_2$ . Khi  $v_1 > v_2$  thế ra có cực tính như hình 9-1. Khi  $v_1 < v_2$  thế ra có cực tính ngược lại.

Khuyếch đại vi sai trên hình 9-1 có 2 lối vào riêng rẽ. Lối vào  $v_1$  gọi là lối vào không đảo vì lối ra cùng pha với  $v_1$ . Ngược lại lối vào  $v_2$  gọi là lối vào đảo vì thế ra ngược pha với  $v_2$ . Trong một số ứng dụng, chỉ có lối vào  $v_1$  được sử dụng còn lối vào  $v_2$  nối GND. Ngược lại một số ứng dụng khác có  $v_1$  nối GND trong khi đó tín hiệu được đưa vào  $v_2$ .

Khi có tín hiệu vào ở cả 2 lối vào, cấu hình này gọi là lối vào vi sai. Khi đó thế ra bằng

$$V_{out} = A(v_1 - v_2) \quad (9-2)$$

trong đó A là hệ số khuyếch đại thế.



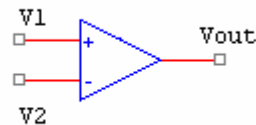
Hình 9-2a

LỐI RA MỘT BÊN

Lối ra vi sai như hình 9-1 yêu cầu một tải nổi (không có điểm nối đất). Hình 9-2a chỉ ra một cách dùng rộng rãi mạch khuếch đại vi sai vì sai vì nó có thể nối với tải thông thường như mạch CE, mạch CC.

Thế ra lấy từ collector bên phải. Do đó collector transistor trái nối với nguồn  $V_{CC}$ . Vì thế lối vào vẫn là vi sai nên thế lối ra là  $v_{out} = A(v_1 - v_2)$ . Nhưng lưu ý hệ số khuếch đại thế chỉ bằng  $\frac{1}{2}$  hệ số khuếch đại thế khi lối ra là vi sai.

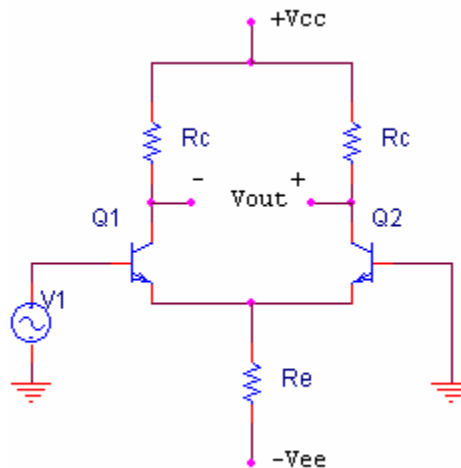
Hình 9-2b là ký hiệu của khuếch đại vi sai (giống như OP AMP). Đầu vào có dấu + là đầu vào không đảo. Đầu vào - là đầu vào đảo.



Hình 9-2b

### CẤU HÌNH LỐI VÀO KHÔNG ĐẢO

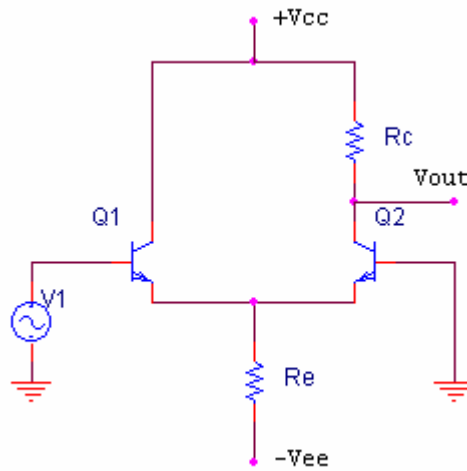
Thông thường chỉ có một lối vào active còn lối vào kia được nối GND như hình 9-3a.



Hình 9-3a

Cấu hình này có một đầu vào không đảo và lối ra vi sai. Do  $v_2=0$  nên  $v_{out} = Av_1$  (9-3)

Hình 9-3b là một cấu hình khác, trong đó có một đầu vào không đảo, lối ra 1 bên. Hệ số khuếch đại chỉ bằng  $\frac{1}{2}$  cấu hình 9-3a.



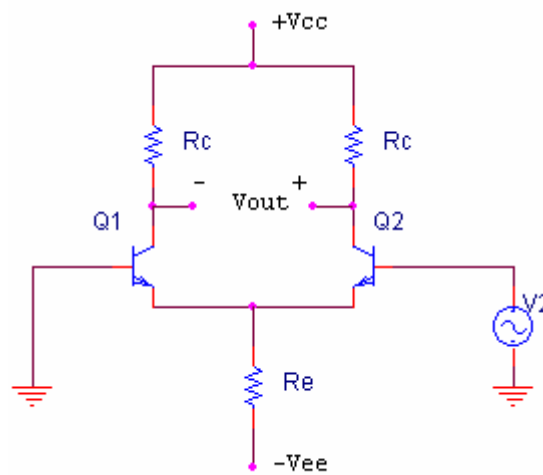
Hình 9-3b

### CẤU HÌNH LỐI VÀO ĐẢO

Trong một số ứng dụng,  $v_2$  là lối vào trong khi đó  $v_1$  được nối GDN như hình 9-4a.

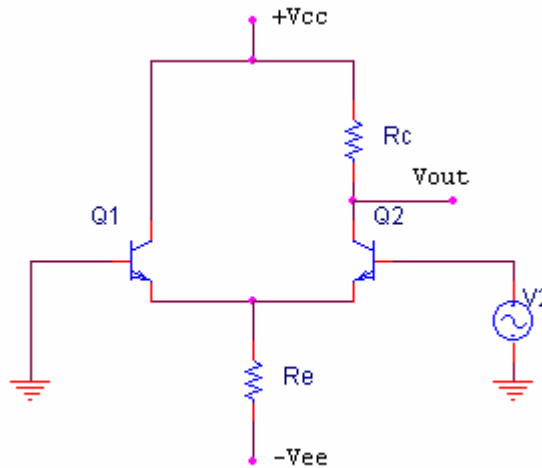
Cấu hình này có thể lối ra

$$V_{out} = -Av_2 \quad (9-4)$$



Hình 9-4a

Hình 9-4b là một cấu hình khác, trong đó có một lối vào đảo, lối ra 1 bên. Hệ số khuếch đại chỉ bằng 1/2 cấu hình 9-4a.



Hình 9-4b

Bảng 9-1 chỉ ra 4 cấu hình cơ bản của một khuếch đại vi sai

| INPUT  | OUTPUT | $V_{in}$         | $V_{out}$         |
|--------|--------|------------------|-------------------|
| Vi sai | Vi sai | $V_1 - V_2$      | $V_{c2} - V_{c1}$ |
| Vi sai | Đơn    | $V_1 - V_2$      | $V_{c2}$          |
| Đơn    | Vi sai | $v_1$ hoặc $v_2$ | $V_{c2} - V_{c1}$ |
| Đơn    | Đơn    | $v_1$ hoặc $v_2$ | $V_{c2}$          |

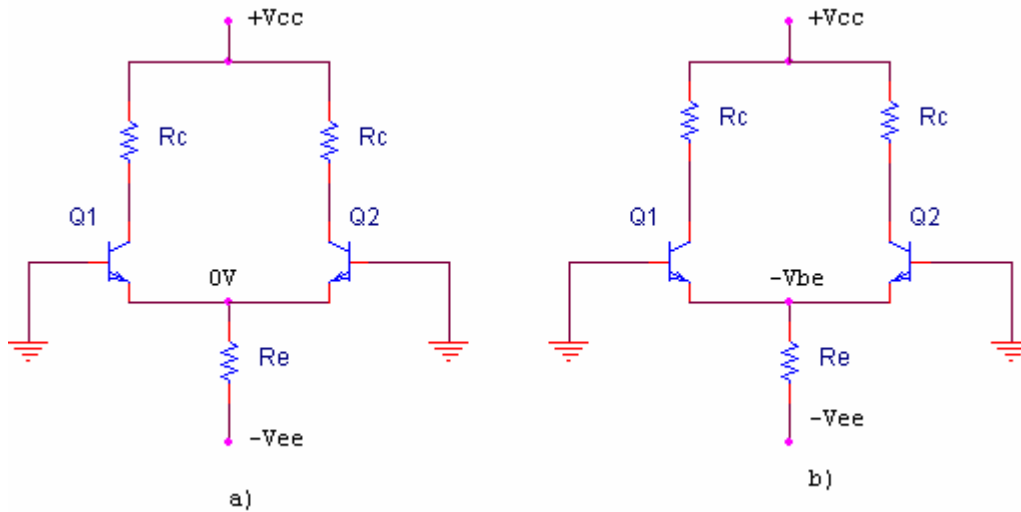
## IX.2 PHÂN TÍCH DC MỘT KHUYẾT ĐẠI VI SAI

Hình 9-5a cho thấy mạch tương đương dc của một khuếch đại vi sai. Mạch phân cực ở đây là mạch TSEB. Hầu như nguồn cung cấp âm trong mạch TSEB là đặt lên  $R_E$ . Điều này tạo ra dòng emitter cố định.

### PHÂN TÍCH LÝ TƯỞNG

Mạch khuếch đại vi sai đôi khi còn gọi là mạch khuếch đại đuôi dài (long tail) vì hai transistor cùng chia sẻ trở  $R_E$ . Dòng chảy qua điện trở dùng chung này gọi là dòng đuôi ( $I_T$ ). Nếu bỏ qua  $V_{BE}$  thì dòng đuôi có giá trị bằng

$$I_T = V_{EE} / R_E. \quad (9-5)$$



Hình 9-5

Dòng qua mỗi transistor bằng

$$I_E = I_T / 2 \quad (9-6)$$

Thế DC trên mỗi collector bằng

$$V_C = V_{CC} - I_C R_C \quad (9-7)$$

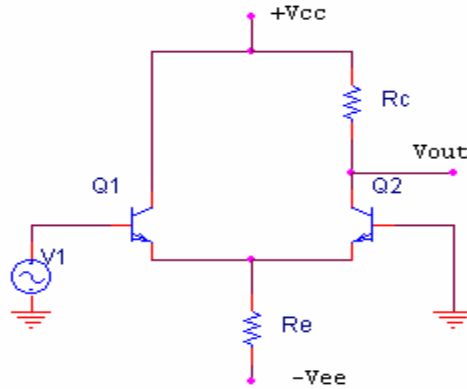
Trong gần đúng bậc 2 cần phải tính đến  $V_{BE} = 0.7V$ , khi đó dòng đuôi bằng

$$I_T = (V_{EE} - V_{BE}) / R_E \quad (9-8)$$

### IX.3 PHÂN TÍCH AC MỘT KHUYẾT ĐẠI VI SAI

Trong phần này chúng ta sẽ dẫn ra hệ số khuếch đại thế của một bộ khuếch đại vi sai. Chúng ta hãy bắt đầu bằng cấu hình đơn giản nhất: lối vào không đảo và lối ra đơn.

Hình 9-6 cho thấy một mạch khuếch đại vi sai có đầu vào không

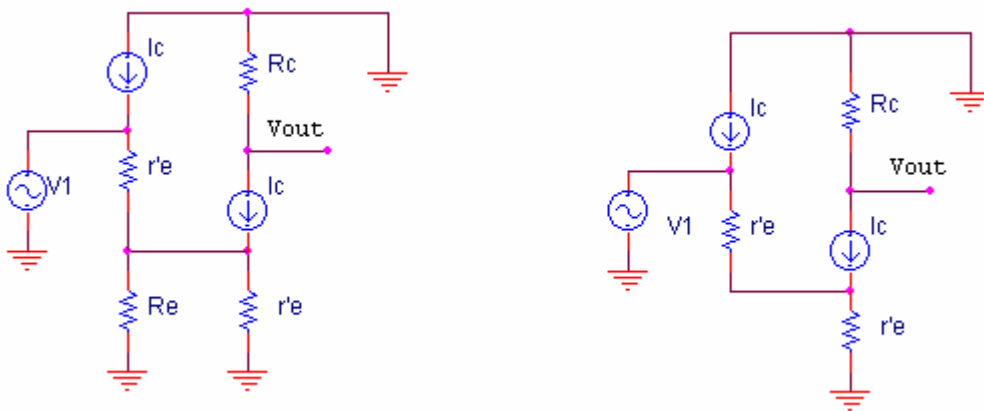


đầu

Hình 9-6

và lối ra đơn. Với  $R_E$  lớn, dòng đầu hầu như không đổi khi có tín hiệu ac nhỏ ở lối vào. Do đó một sự tăng dòng emitter  $Q_1$  tạo ra một sự giảm tương ứng dòng emitter  $Q_2$ . Ngược lại một sự giảm dòng emitter  $Q_1$  tạo ra sự tăng tương ứng dòng emitter  $Q_2$ .

Trên hình 9-6, transistor bên trái đóng vai trò như một bộ lặp lại emitter. Tuy nhiên cần lưu ý là thế trên emitter chỉ bằng  $\frac{1}{2}$  thế vào. Trong nửa chu kỳ dương, dòng emitter  $Q_1$  tăng và dòng emitter  $Q_2$  giảm làm cho thế collector  $Q_2$  tăng. Ngược lại vào nửa chu kỳ âm thế collector



$Q_2$  giảm.

Hình 9-7

Hình 9-7a cho thấy sơ đồ tương đương. Trong thiết kế  $R_E \gg r'_e$  do đó có thể bỏ qua  $R_E$  trong các phân tích sơ cấp và chúng ta thu được mạch tương đương đơn giản như hình 9-7b.

## HỆ SỐ KHUYẾCH ĐẠI LỐI RA ĐƠN

Trong hình 9-7a, thế ra bằng

$$v_{\text{out}} = i_c R_C$$

Thế vào ac là

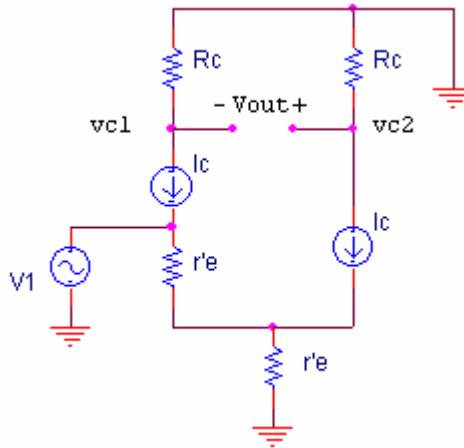
$$v_{\text{in}} = i_c r'_e + i_c r'_e = 2i_c r'_e$$

Hệ số khuếch đại lối ra đơn bằng

$$A = R_C / 2r'_e \quad (9-9)$$

## HỆ SỐ KHUYẾCH ĐẠI LỐI VI SAI

Hình 9-8 cho thấy sơ đồ tương đương lối ra cho mạch có lối vào đơn, lối ra vi sai. Phân tích giống như phần trước ngoại trừ thế lối ra tăng gấp



đôi.

Hình 9-8

$$v_{\text{out}} = v_{c2} - v_{c1} = i_c R_C - (-i_c R_C) = 2 i_c R_C$$

Thế ac lối vào là

$$v_{\text{in}} = 2i_c r'_e$$



Hệ số khuếch đại thế lối ra vi sai bằng

$$A = R_C / r'_e \quad (9-10)$$

### CẤU HÌNH LỐI VÀO ĐẢO

Có thể thấy rằng cấu hình lối vào đảo hoàn toàn giống cấu hình lối vào không đảo. Chỉ khác một điều là tín hiệu ra đảo pha so với tín hiệu vào.

### CẤU HÌNH LỐI VÀO VI SAI

Cấu hình lối vào vi sai có hai lối vào đều active. Phân tích ac của cấu hình lối vào vi sai có thể dùng nguyên lý chồng chất như sau: Chúng ta đã biết hoạt động của cấu hình lối vào đảo và không đảo một cách riêng rẽ. Bằng cách tổ hợp hai kết quả này chúng ta thu được phương trình cho cấu hình lối vào vi sai.

Lối ra cho cấu hình không đảo là

$$v_{out} = Av_1$$

Lối ra cho cấu hình đảo là

$$v_{out} = -Av_2$$

Lối ra cho lối vào vi sai bằng

$$v_{out} = A(v_1 - v_2)$$

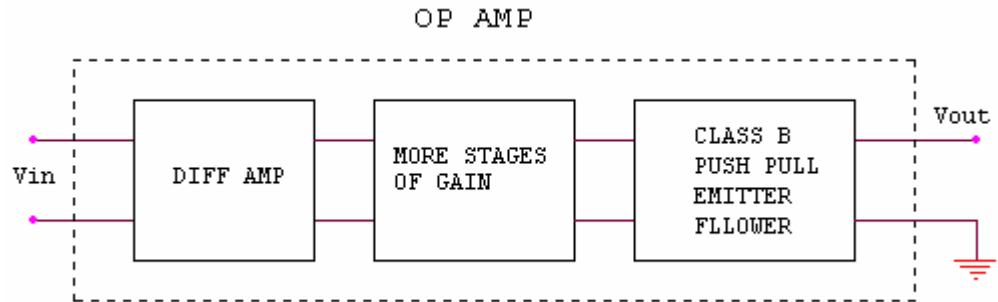
Bảng 9-2 cho thấy hệ số khuếch đại đối với các cấu hình khác nhau.

| INPUT  | OUTPUT | A             | $v_{out}$           |
|--------|--------|---------------|---------------------|
| Vi sai | Vi sai | $R_C / r'_e$  | $Av_1$              |
| Vi sai | Đơn    | $R_C / 2r'_e$ | $A(v_1 - v_2)$      |
| Đơn    | Vi sai | $R_C / r'_e$  | $Av_1$ hoặc $-Av_2$ |
| Đơn    | Đơn    | $R_C / 2r'_e$ | $Av_1$ hoặc $-Av_2$ |

Trở kháng vào của bộ khuếch đại vi sai

$$Z_{in} = 2\beta r'_c \quad (9-11)$$

## IX.4 KHUYẾCH ĐẠI OP AMP

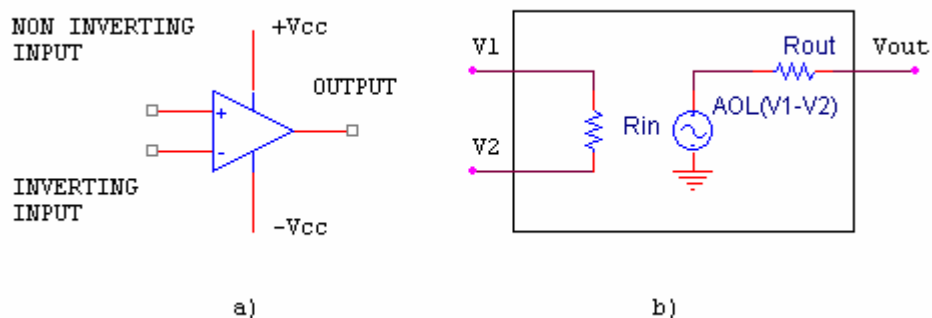


Hình 9-9

Hình 9-9 cho thấy sơ đồ khối một OP AMP.

Tầng vào là một khuếch đại vi sai. Sau nó là các tầng khuếch đại thế. Lối ra là bộ lặp lại emitter kiểu đẩy kéo hạng B. Khuếch đại vi sai ở lối vào quy định những đặc trưng vào của OP AMP như trở kháng vào rất cao, hệ số cách chung cao. Trong hầu hết các ứng dụng, lối ra của OP AMP là lối ra đơn như hình 9-9. Với việc sử dụng nguồn nuôi kép, lối ra đơn được thiết kế để có các giá trị tĩnh bằng 0. Theo cách này thế lối vào bằng 0 là lý tưởng để thế lối ra bằng 0.

Không phải tất cả các OP AMP đều có thiết kế như hình 9-9. Chẳng hạn có 1 số OP AMP không dùng lối ra đơn, một số khác không dùng tầng đẩy kéo hạng B. Cho dù thiết kế bằng cách nào đi nữa thì hình 9-9 là sơ đồ đơn giản nhất của một OP AMP. Trên thực tế vi mạch OP AMP có cấu trúc rất phức tạp. Trong OP AMP, người ta dùng các kỹ thuật pha tạp transistor dòng điện gương, tải động và nhiều kỹ thuật khác mà kỹ thuật



mạch rời không làm được.

Hình 9-10

Hình 9-10a là sơ đồ ký hiệu của một OP AMP. Nó có một đầu vào không đảo, một đầu vào đảo và một lối ra đơn. Một cách lý tưởng, sơ đồ này có hệ số khuếch đại thế bằng vô cùng, trở kháng vào rất lớn và trở kháng ra bằng 0. Một OP AMP lý tưởng biểu diễn cho một bộ khuếch đại thế hoàn hảo và thường được gọi là một nguồn thế được điều khiển bởi thế (Voltage Controlled Voltage Source - VCVS). Sơ đồ tương đương ac của VCVS như hình 9-10b. Trong 9-10b trở kháng vào  $R_{in}$  bằng vô cùng, trở kháng ra  $R_{out}$  bằng 0.

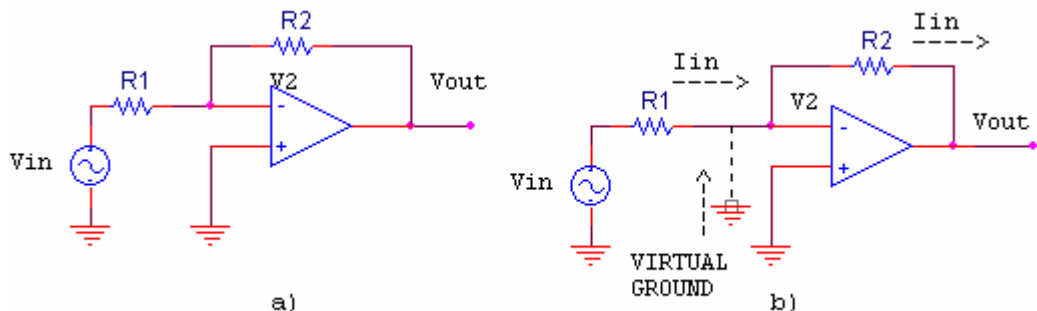
Bảng 9-3 cho thấy một số OP AMP điển hình và các đặc trưng của chúng

| Quantity                  | Symbol         | Ideal   | LM741C     | LF157A         |
|---------------------------|----------------|---------|------------|----------------|
| Hệ số khuếch đại vòng     | $A_{OL}$       | Vô cùng | 90000      | 200000         |
| Tần số tại đó $A=1$       | $f_{unity}$    | Vô cùng | 1MHz       | 20MHz          |
| Trở kháng vào             | $R_{in}$       | Vô cùng | $2M\Omega$ | $9^{12}\Omega$ |
| Trở kháng ra              | $R_{out}$      | Zero    | $75\Omega$ | $90\Omega$     |
| Dòng phân cực vào         | $i_{in(bias)}$ | Zero    | 80nA       | 30pA           |
| Dòng offset vào           | $i_{in(off)}$  | Zero    | 20nA       | 3pA            |
| Thế offset lối vào        | $V_{in(off)}$  | Zero    | 2mV        | 1mV            |
| Tỷ số loại trừ cách chung | CMRR           | Vô cùng | 90dB       | 90dB           |

Bảng 9-3

## IX.5 KHUYẾCH ĐẠI ĐẢO

Khuếch đại đảo là mạch OP AMP cơ bản nhất. Nó dùng phản hồi âm để ổn định hệ số khuếch đại tổng thể của mạch. Lý do cần phải ổn định hệ số khuếch đại là vì  $A_{OL}$  quá lớn và không ổn định. Ví dụ 741C có  $A_{OL}$  bé nhất là 20000 đến  $A_{OL}$  lớn nhất là 200000.



## Hình 9-11

Hình 9-11 chỉ ra mạch khuếch đại đảo. Để đơn giản chúng ta không vẽ ra sơ đồ nguồn nuôi. Thế vào  $v_{in}$  được đưa vào lối vào đảo thông qua điện trở  $R_{in}$ . Điều này tạo ra trên đầu vào đảo một thế  $v_2$ . Thế lối vào được khuếch đại và tạo ra thế lối ra đảo pha so với thế vào. Thế ra được phản hồi lại đầu vào thông qua điện trở phản hồi  $R_2$ . Đây là sự phản hồi âm vì thế lối ra đảo pha so với thế vào.

Cách mà sự phản hồi âm ổn định hệ số khuếch đại tổng thể của mạch là như sau: Nếu  $A_{OL}$  tăng vì bất cứ lý do gì thì thế lối ra sẽ tăng và do có sự phản hồi âm thế lối vào sẽ giảm.

### Đất ảo (virtual ground)

Khi chúng ta nối một dây dẫn từ một điểm trên mạch xuống đất, thế tại điểm đó sẽ bằng 0. Trong trường hợp đó dây dẫn tạo ra đường dẫn cho dòng điện chảy xuống đất. Đất cơ học là đất cả thế lẫn dòng.

Đất ảo thì khác. Đó là khái niệm về một điểm nào đó trên mạch có thế và dòng bằng không. Khái niệm đất ảo làm cho việc phân tích mạch trở nên rất đơn giản. Đất ảo dựa trên tính chất của mạch OP AMP. Khi một OP AMP là lý tưởng, nó có hệ số khuếch đại vòng hở bằng vô cùng và trở vào cũng bằng vô cùng. Do đó chúng ta có thể suy ra (deduce) các thuộc tính lý tưởng sau đây cho mạch khuếch đại OP AMP có đầu vào đảo:

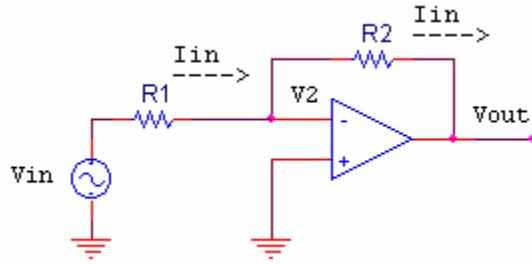
- ◆ Do  $R_{in}$  bằng vô cùng nên dòng vào  $i_2$  bằng 0
- ◆ Do  $A_{OL}$  bằng vô cùng nên thế vào  $v_2$  bằng 0

Do  $i_2$  bằng 0 nên dòng chảy qua  $R_1$  phải bằng dòng chảy qua  $R_2$  như trên hình vẽ. Mặt khác  $v_2$  bằng 0 nên đầu vào – là đất ảo. Đất ảo là khái niệm tương đương nửa đất, nghĩa là thế bằng không nhưng không có dòng chảy qua. Trên hình vẽ chúng ta biểu thị đất ảo bằng đường chấm chấm. Mặc dù đất ảo là gần đúng lý tưởng nhưng nó tỏ ra chính xác khi phản hồi âm sâu.

### HỆ SỐ KHUYẾCH ĐẠI THẾ

Trên hình 9-12, chúng ta hình dung đầu vào đảo là đất ảo. Khi đó có thể viết

$$\begin{aligned}v_{in} &= i_{in}R_1 \\v_{out} &= -i_{in}R_2\end{aligned}$$



Hình 9-12

Hệ số khuếch đại thế (vòng kín) bằng

$$A_{CL} = v_{out} / v_{in} = -R_2 / R_1 \quad (9-12)$$

Đây là hệ số khuếch đại thế vòng kín nó là hệ số khuếch đại khi có vòng phản hồi âm giữa lối ra và lối vào. Do đó phản hồi âm nên  $A_{CL} < A_{OL}$ . Chẳng hạn,  $R_2=50K\Omega$  còn  $R_1=1K\Omega$  thì  $A_{CL}=50$  kể cả khi thay đổi nhiệt độ, nguồn nuôi và thậm chí thay thế OP AMP.

#### **Trở kháng vào**

Do đầu vào đảo là đất ảo do đó trở kháng vào của vòng kín là

$$Z_{in(CL)} = R_1 \quad (9-13)$$

Ưu điểm của mạch khuếch đại OP AMP đảo là có thể chọn giá trị của trở kháng vào.

#### **BANDWIDTH**

Bandwidth vòng hở hay tần số cắt của của OP AMP rất thấp do có tụ bù bên trong. Chẳng hạn với 741C

$$f_{2(OL)} = 9\text{Hz.}$$

Tại tần số này hệ số khuếch đại bắt đầu giảm. Khi phản hồi âm được dùng, bandwidth tổng thể tăng. Đây là lý do: Khi tần số vào lớn hơn  $f_{2(OL)}$  thì  $A_{OL}$  giảm 20dB/dec. Vì  $v_{out}$  giảm nên phản hồi âm cũng giảm. Điều này làm cho  $v_2$  tăng và bù lại sự giảm của  $A_{OL}$ . Do đó  $A_{CL}$  bị giảm tại tần số cao hơn  $f_{2(OL)}$ . Phản hồi âm càng sâu thì tần số cắt vòng kín càng cao. Nói cách khác  $A_{CL}$  càng bé thì  $f_{2(CL)}$  càng lớn. Đây là phương trình cho tần số cắt vòng kín

$$f_{2(CL)} = f_{unity} / (A_{CL}+1)$$

Thông thường  $A_{CL} \gg 1$  do đó

$$f_{2(CL)} = f_{unity} / A_{CL} \quad (9-14)$$

Ví dụ khi  $A_{CL}=10$  thì

$$f_{2(CL)} = 1\text{MHz} / 10 = 100\text{KHz}$$

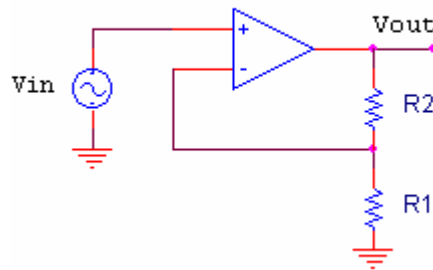
Lưu ý rằng

$$f_{unity} = f_{2(CL)} A_{CL} \quad (9-15)$$

Tần số khuếch đại đơn vị bằng tích giữa band thông và hệ số khuếch đại vòng kín. Nhiều tài liệu ra cứ gọi  $f_{unity}$  là GBW (gain bandwidth).

## IX.6 KHUYẾCH ĐẠI KHÔNG ĐẢO

Khuyếch đại không đảo là một mạch OP AMP cơ bản khác. Nó cũng dùng sự phản hồi âm để ổn định hệ số khuếch đại toàn thể. Sự phản hồi âm cũng làm tăng trở kháng vào và làm giảm trở kháng ra.



Hình 9-13

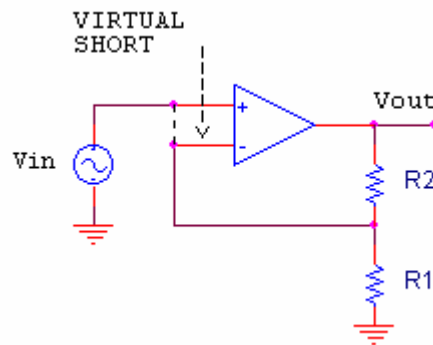
Hình 9-13 cho thấy mạch tương đương ac của một khuếch đại không đảo. Một thế vào ac được đưa tới lối vào không đảo. Thế này được khuếch đại ở lối ra. Thế lối ra có cùng pha với thế lối vào. Một phần thế lối ra được phản hồi lại đầu vào đảo thông qua cầu chia thế  $R_1$  và  $R_2$ . Thế

phản hồi hầu như bằng thế vào. Do hệ số khuếch đại vòng hở rất lớn nên sự khác nhau giữa  $v_1$  và  $v_2$  rất bé.

Sự ổn định do phản hồi trong mạch này như sau: Giả sử  $A_{OL}$  tăng thì thế phản hồi vào đầu vào đảo cũng tăng. Điều này làm giảm  $v_1 - v_2$  và kết quả là  $v_{out}$  giảm.

### NGẮN MẠCH ẢO (VIRTUAL SHORT)

Khi chúng ta dùng dây nối để nối 2 điểm trên mạch chúng ta có sự ngắn mạch cơ học. Sự ngắn mạch cơ học làm cho thế và dòng giữa các điểm bị ngắn mạch bằng 0.



Hình 9-14

Sự ngắn mạch ảo là một khái niệm để chỉ ra rằng 2 điểm trên mạch có thế bằng 0 và vì vậy không có dòng chảy qua nó. Hình 9-14 cho thấy ngắn mạch ảo giữa 2 đầu vào của OP AMP trong cấu hình không đảo. Sở dĩ có thể hình dung như vậy là vì đối với một OP AMP lý tưởng

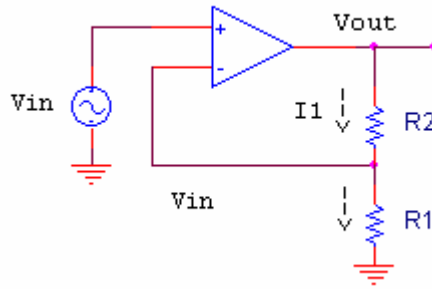
- ◆ Do  $R_{in}$  bằng vô cùng nên không có dòng chảy qua các đầu + và -
- ◆ Do  $A_{OL}$  bằng vô cùng nên  $v_1 - v_2 = 0$

Trong mạch OP AMP không đảo và một số mạch OP AMP khác chúng ta hình dung ngắn mạch giữa 2 lối vào.

### Hệ số khuếch đại vòng kín

Trên hình 9-15, hình dung ngắn mạch giữa 2 lối vào, ta có

$$v_{in} = i_1 R_{in}$$



Hình 9-15

Mặt khác do không có dòng chảy vào OP AMP nên dòng  $i_1$  cũng chảy qua  $R_2$  vì vậy

$$v_{out} = i_1(R_2 + R_1)$$

Hệ số khuếch đại thế vòng kín bằng

$$A_{CL} = v_{out} / v_{in} = (R_2 + R_1) / R_1 = R_2 / R_1 + 1$$

$$A_{CL} = R_2 / R_1 + 1 \quad (9-16)$$

Trở kháng vào của mạch khuếch đại không đảo xấp xỉ bằng vô cùng.

Do ảnh hưởng của mạch phản hồi âm, tần số cắt vòng kín bằng

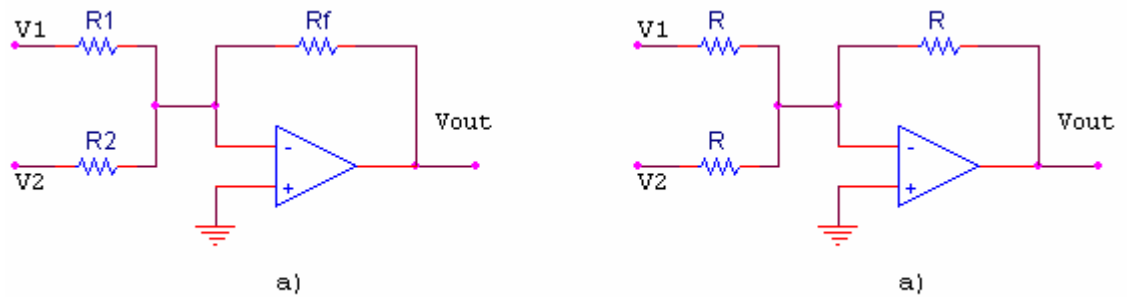
$$f_{2(CL)} = f_{unity} / A_{CL}$$

## IX.7 HAI ỨNG DỤNG CỦA KHUYẾCH ĐẠI OP AMP

### 1) MẠCH CỘNG



Khi chúng ta cần cộng hai hay nhiều tín hiệu lại với nhau thì mạch hình 9-16a được dùng.



Hình 9-16

Có thể thấy rằng mạch hình 9-16a là chồng chất của 2 mạch khuếch đại đảo. Hệ số khuếch đại vòng kín cho mỗi lối vào (mỗi kênh) riêng rẽ là

$$A_{CL1} = R_F / R_1$$

Và

$$A_{CL2} = R_F / R_2$$

Tín hiệu ra khi chồng chất hai mạch lại là

$$V_{out} = A_{CL1}V_1 + A_{CL2}V_2 \quad (9-17)$$

Dễ dàng chứng minh phương trình (9-17). Thật vậy do lối vào đảo là đất ảo nên:

$$i_{in} = i_1 + i_2 = v_1 / R_1 + v_2 / R_2$$

Cũng vì đất ảo nên

$$V_{out} = (i_1 + i_2)R_F = R_F v_1 / R_1 + R_F v_2 / R_2$$

Đây chính là phương trình (9-17)

Nếu  $A_{CL1} = A_{CL2} = 1$  thì

$$V_{out} = V_1 + V_2$$



Chúng ta cũng có thể thu được kết quả này từ phương trình (9-16) trong đó thay  $R_2=0$  và  $R_1 = \text{vô cùng}$ .

Mạch lặp lại thế dùng OP AMP là mạch lặp lại hoàn hảo vì thế ra đúng bằng với thế vào.

Hình 9-18b là mạch lặp lại thế trong đó thế vào có trở kháng cao  $R_{\text{high}}$ . Tải của mạch là trở kháng thấp  $R_{\text{low}}$ . Do phản hồi âm sâu, trở kháng vào của mạch là cao hơn trở kháng vào vòng hở ( $2M\Omega$ ) và trở kháng ra thấp hơn trở kháng ra vòng hở ( $75\Omega$ ).

Mạch lặp lại thế là mạch giao tiếp lý tưởng giữa nguồn có trở kháng cao với tải có trở kháng thấp.

Trong mạch lặp lại thế do  $A_{\text{CL}}=1$  nên

$$f_{2(\text{CL})} = f_{\text{unity}} \quad (9-18)$$

---

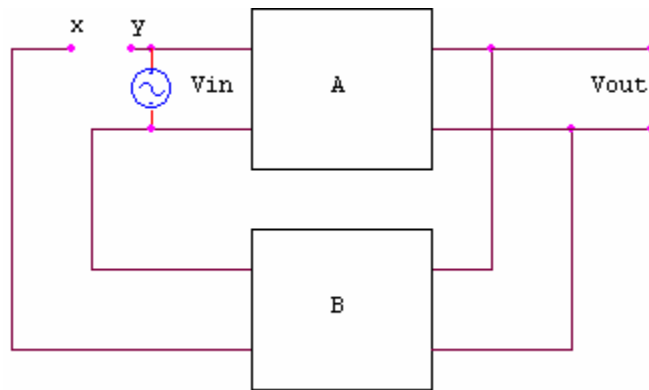
## Chương X **CÁC MẠCH DAO ĐỘNG**

### **X.1 LÝ THUYẾT DAO ĐỘNG ĐIỀU HOÀ**

Trong giáo trình này chúng ta chỉ xem xét dao động điều hoà (sóng sin). Các dao động tạo sóng vuông sẽ được đề cập trong giáo trình Mạch và Hệ thống số. Để có một mạch dao động điều hoà chúng ta cần một bộ khuếch đại phản hồi dương (positive feedback). Phản hồi dương là đưa một phần tín hiệu lối ra trở lại lối vào sao cho nó có tác dụng cùng chiều với tín hiệu ra. Nếu tín hiệu phản hồi là đủ lớn và có pha đúng nó sẽ tạo ra một tín hiệu ra mà không cần có tín hiệu vào.

#### **HỆ SỐ KHUYẾTCH ĐẠI VÒNG VÀ PHA**

Hình 10-1 cho thấy một nguồn ac được đưa vào một bộ khuếch đại.



Hình 10-1

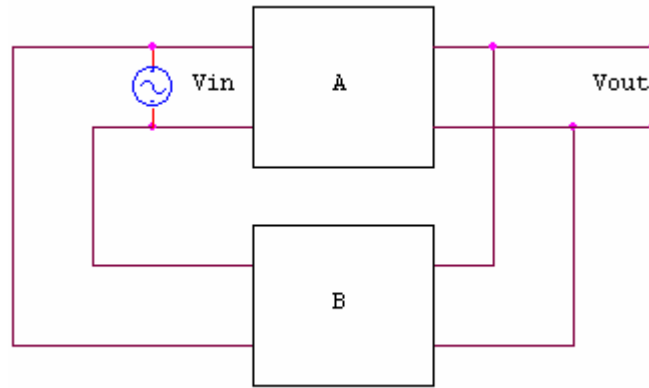
Thế ra của bộ khuếch đại bằng

$$V_{out} = AV_{in}$$

Thế ra này đưa vào mạch phản hồi mà thông thường là mạch cộng hưởng. Do đó chỉ có một tần số nào đó được phản hồi nhiều nhất. Thế phản hồi lại đầu vào của bộ khuếch đại bằng

$$V_f = ABV_{in}$$

Nếu sự dịch pha của bộ khuếch đại và mạch phản hồi bằng 0 thì thế phản hồi cùng pha với thế vào. Giả sử chúng ta nối điểm x với điểm y và tháo bỏ nguồn  $v_{in}$  thì thế phản hồi  $ABv_{in}$  sẽ thúc bộ khuếch đại như hình 10-2.



Hình 10-2

Sẽ có các trường hợp sau đây xảy ra:

Nếu  $AB < 1$  thì thế ra sẽ tắt dần

Nếu  $AB > 1$  thì thế ra sẽ tăng dần đến tình trạng bị cắt

Nếu  $AB = 1$  thì thế ra không đổi và tự duy trì

Trong các mạch dao động thực, ban đầu hệ số khuếch đại vòng  $AB$  lớn hơn 1 khi bật nguồn cho mạch. Thế ra sẽ tăng dần đến một giá trị cần thiết thì  $AB$  tự động tiến đến 1 và dao động được duy trì.

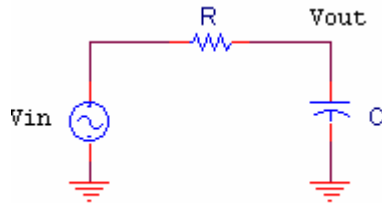
Vậy thì thế lối vào ban đầu trong mạch dao động là gì? Chúng ta biết rằng trong mỗi điện trở có các electron tự do. Do nhiệt độ môi trường các electron này chuyển động tự do theo mọi hướng và tạo ra thế nhiễu (noise voltage) trên điện trở. Sự chuyển động hỗn độn này chứa các tần số từ 0 đến 1000GHz. Có thể hình dung điện trở là một nguồn thế ac nhỏ chứa tất cả các tần số.

Khi bật điện cho mạch lần đầu tiên, trên mạch chỉ có thế nhiễu tạo ra bởi các điện trở. Các thế nhiễu này được khuếch đại và xuất hiện ở lối ra của mạch. Nhưng chỉ có tín hiệu có tần số đúng bằng tần số cộng hưởng của mạch phản hồi mới được đưa trở lại vào đầu vào. Sau quá trình quá độ chỉ còn tín hiệu có tần số thỏa điều kiện  $AB = 1$  và cùng pha với tín hiệu ra. Tất cả các tần số khác không xuất hiện ở lối ra.

## X.2 DAO ĐỘNG CẦU WIEN

Mạch dao động cầu Wien là mạch dao động trong vùng tần số thấp đến trung bình, tức là khoảng 5Hz đến 1MHz. Chúng được ứng dụng trong các máy phát âm tần thương mại và các ứng dụng tần số thấp.

### MẠCH LAG



Hình 10-3

Hệ số khuếch đại thế của mạch trên hình 10-3 là

$$V_{\text{out}} / V_{\text{in}} = X_C / (R^2 + X_C^2)^{1/2}$$

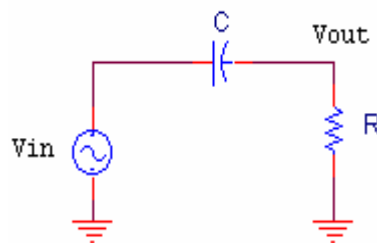
Góc pha giữa tín hiệu vào và ra bằng

$$\phi = - \arctan(R/X_C)$$

Dấu trừ trong phương trình góc pha chứng tỏ rằng thế lối ra chậm (lag) pha hơn thế lối vào.

### MẠCH LEAD

Hình 10-4 cho thấy một mạch lead.



Hình 10-4

Hệ số khuếch đại thế của mạch này là

$$V_{\text{out}} / V_{\text{in}} = R / (R^2 + X_C^2)^{1/2}$$

Góc pha giữa tín hiệu vào và ra bằng

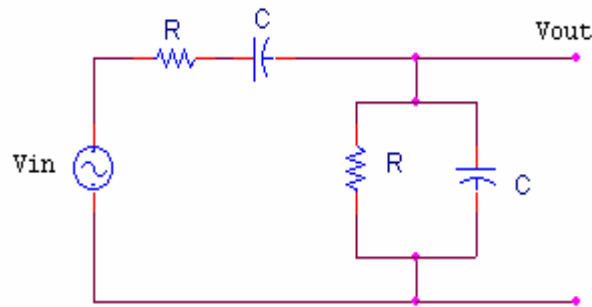
$$\phi = \arctan(X_C/R)$$

Góc pha dương chứng tỏ rằng thế lối ra nhanh (lead) pha hơn thế lối vào.

Mạch lead và mạch lag như trên đây là thí dụ về các mạch dịch pha. Chúng có thể làm cho pha tín hiệu ra dịch pha (nhanh hoặc chậm) so với tín hiệu vào. Các mạch dao động điều hoà thường sử dụng các loại mạch dịch pha để tạo ra dao động tại một tần số.

### MẠCH LEAD-LAG

Dao động cầu Wien dùng mạng phản hồi cộng hưởng gọi là mạch lead-lag như hình 10-5.



Hình 10-5

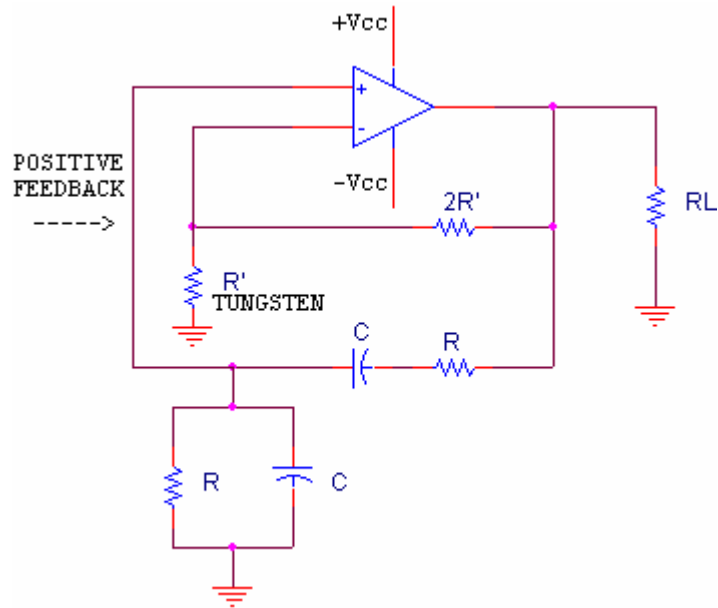
Tại tần số rất thấp, tụ nối tiếp xem như hở mạch đối với tín hiệu vào. Tại tần số rất cao tụ mắc shunt có tác dụng ngắn mạch đối với tín hiệu ra. Giữa hai giá trị này thế ra đạt giá trị cực đại. Tần số tại đó tín hiệu ra cực đại gọi là tần số cộng hưởng  $f_r$ . Cũng tại tần số này hệ số phản hồi  $\beta$  đạt đến giá trị cực đại và bằng 1/3. Tại tần số cộng hưởng góc pha bằng 0.

Phân tích bằng số phức chúng ta thu được hai phương trình sau cho mạch hình 10-5.

$$\beta = 1 / (10 - X_C / (R^2 + X_C^2))^{1/2} \quad (10-1)$$

và

$$\phi = \arctan((X_C/R - R/X_C)/3) \quad (10-2)$$



**Hình 10-6**

Hệ số phản hồi  $\beta$  đạt giá trị cực đại  $1/3$  khi  $X_C=R$  .Từ đó suy ra:

$$f_r = 1/2\pi RC \quad (10-3)$$

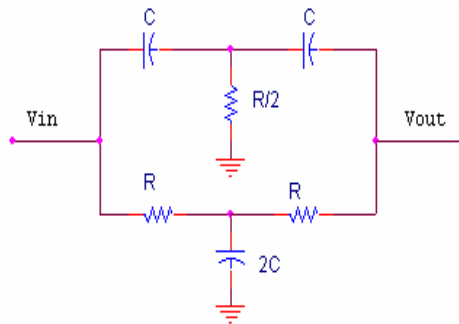
Hình 10-6 là mạch dao động cầu Wien. Nó dùng cả phản hồi âm và phản hồi dương. Phản hồi dương thực hiện qua mạch lead-lag về lối vào không đảo. Phản hồi âm qua cầu chia thế về lối vào đảo. Khi mới cấp nguồn cho mạch, lượng phản hồi dương nhiều hơn phản hồi âm (trở của dây tóc bóng đèn ban đầu có giá trị bé do chưa được nung nóng). Điều này cho phép mạch tự dao động như đã mô tả trước đây. Sau khi tín hiệu ra đạt đến giá trị mong muốn, sự phản hồi âm đủ lớn làm cho hệ số khuếch đại vòng kín  $A\beta=1$ .

### **X.3    DAO ĐỘNG RC**

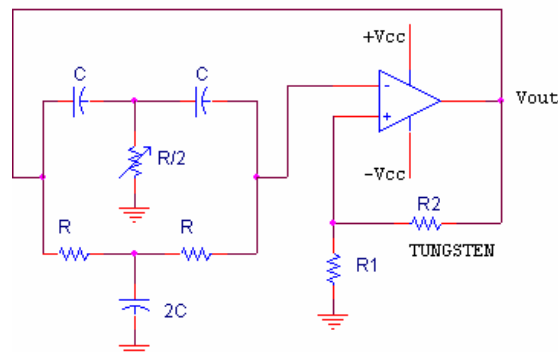
Mặc dù dao động cầu Wien là dao động chuẩn cho các ứng dụng có tần số dưới 1MHz, nhưng các dao động RC khác cũng được dùng. Phần này sẽ phân tích mạch dao động chữ T kép và mạch dao động dịch pha.

#### **BỘ LỘC CHỮ T**





Hình 10-7



Hình 10-8

Hình 10-7 cho thấy một mạch lọc chữ T kép. Phân tích toán học cho mạch này giống như mạch lead-lag. Mặc khác có một tần số cộng hưởng  $f_r$  mà tại đó độ dịch pha bằng 0 và hệ số khuếch đại bằng 0. Phương trình cho tần số cộng hưởng của bộ lọc chữ T kép giống như với mạch dao động cầu Wien.

$$f_r = 1/2\pi RC$$

### DAO ĐỘNG CHỮ T

Hình 10-8 cho thấy mạch dao động chữ T kép.

Phản hồi dương thực hiện bởi cầu chia thế đến đầu vào không đảo. Sự phản hồi âm thông qua bộ lọc chữ T kép.

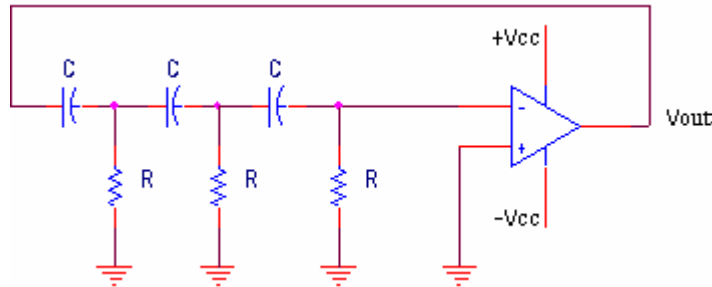
Khi mới bật điện, điện trở  $R_2$  của bóng đèn thấp và do đó phản hồi dương là tối đa. Khi dao động đã được xác lập,  $R_2$  tăng và phản hồi dương giảm dần. Khi đạt được tình trạng  $A\beta=1$  thì mạch trở nên ổn định.

### DAO ĐỘNG DỊCH PHA

Hình 10-9 là một mạch dao động dịch pha.

Nó có 3 mạch lead trên đường phản hồi âm. Mạch lead tạo ra góc dịch pha giữa 0 và  $100^{\circ}$ . Nếu như mỗi mạch lead có góc dịch pha là  $60^{\circ}$  thì góc dịch pha tổng là  $180^{\circ}$ . Pha lối ra lại bị xoay  $180^{\circ}$ . Do đó góc dịch pha của vòng là  $0^{\circ}$ . Nếu thỏa mãn điều kiện  $A\beta=1$  thì mạch sẽ dao động.

Nhược điểm chủ yếu của các mạch dao động dịch pha là khó điều chỉnh tần số trong một vùng rộng.



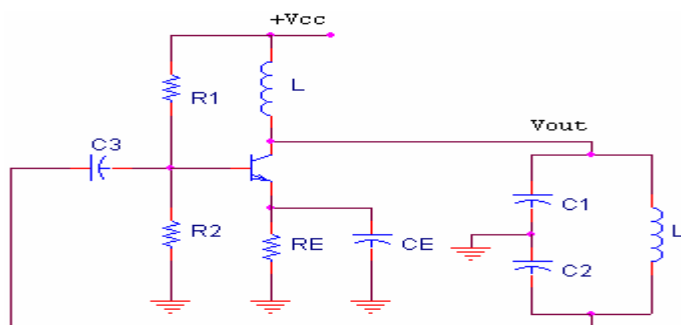
Hình 10-9

#### X.4 DAO ĐỘNG COLPITTS

Dao động cầu Wien chỉ thích hợp ở tần số thấp do giá trị hữu hạn của  $f_{\text{unity}}$  của OP AMP. Để tạo ra dao động tần số cao từ 1MHz đến 500MHz cần mạch dao động LC. Dải tần này lớn hơn  $f_{\text{unity}}$  của hầu hết các OP AMP. Do đó trong mạch dao động LC người ta dùng transistor hoặc FET. Trong mạch LC có thể phản hồi tín hiệu với biên độ và pha hợp lý cho sự dao động.

Tuy nhiên khó khăn chủ yếu của việc phân tích và thiết kế dao động LC là do tần số cao. Khi đó ảnh hưởng của các tụ ký sinh là đáng kể.

Hình 10-10 cho thấy một mạch dao động Colpitts.



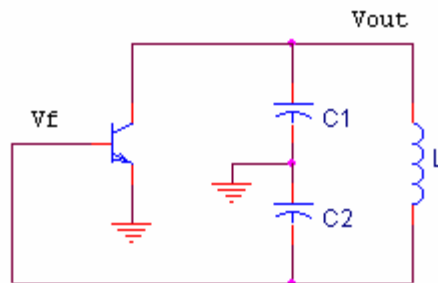
Hình 10-10

Cầu chia thế setup điểm phân cực tĩnh. Cuộn RF có cảm kháng rất cao do đó nó xem như hở mạch đối với tín hiệu ac. Hệ số khuếch đại tại tần số thấp bằng

$$r_c / r'_e \quad (10-4)$$

trong đó  $r_c$  là trở kháng xoay chiều của mạch collector. Do RF có trở kháng rất lớn, trở kháng ac của collector chủ yếu là trở kháng của mạch cộng hưởng. Trở kháng này có giá trị cực đại tại tần số cộng hưởng.

Có thể gặp các mạch dao động Colpitts kiểu khác. Dấu hiệu chung là cầu chia thế bằng tụ  $C_1$  và  $C_2$ . Chúng tạo ra sự phản hồi cần thiết cho dao động.



Hình 10-11

Hình 10-11 là mạch tương đương ac của dao động Colpitts. Lưu ý rằng thế ra trên  $C_1$  còn thế phản hồi lấy trên  $C_2$ .

#### **Tần số cộng hưởng**

Mạch cộng hưởng RC trong dao động Colpitts có tần số cộng hưởng bằng

$$f_r = 1/2\pi(LC)^{1/2} \quad (10-5)$$

trong đó  $C = C_1 C_2 / (C_1 + C_2)$  do  $C_1$  và  $C_2$  nối tiếp nhau.

Hệ số phản hồi của mạch bằng

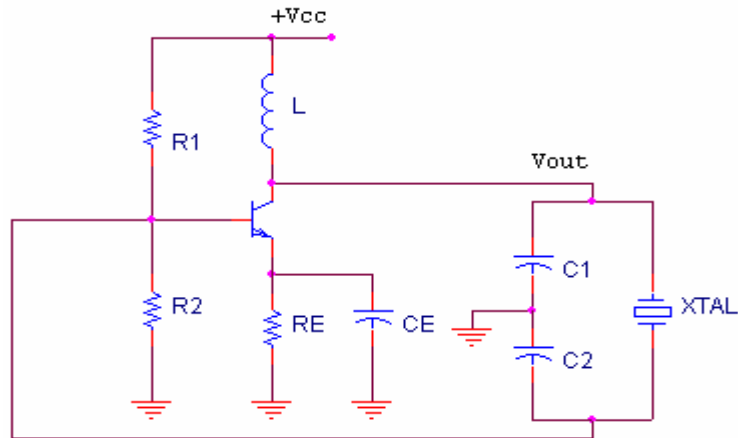
$$\beta = C_1 / C_2$$

do đó hệ số khuếch đại thế bé nhất bằng

$$A_{\min} = C_2 / C_1$$

## X.5 DAO ĐỘNG THẠCH ANH

Khi độ chính xác và độ ổn định của tần số là quan trọng thì dao động thạch anh được dùng. Hình 10-12 cho thấy một dao động thạch anh.



Hình 10-12

Thạch anh (XTAL) đóng vai trò như một cuộn cảm lớn nối tiếp với một tụ bé. Do đó tần số cộng hưởng hầu như không ảnh hưởng bởi transistor và các tụ ký sinh.

Đối với mạch dao động thạch anh, có thể dùng các biểu thức sau

$$f_r = 1/2\pi(LC)^{1/2}$$

$$C = 1 / (1/C_1 + 1/C_2 + 1/C_3)$$

$$\beta = C_1 / C_2$$

$$A_{\min} = C_2 / C_1$$

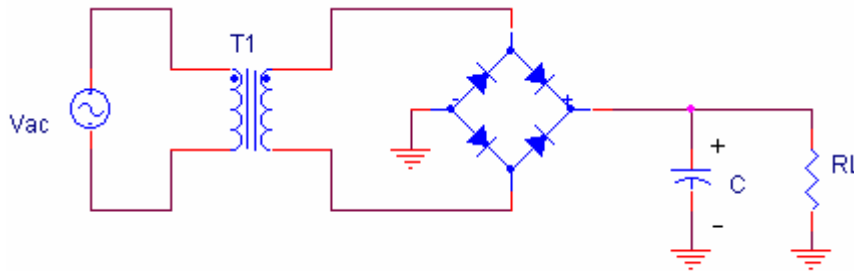
## Chương XI **NGUỒN NUÔI**

### **XI.1 CÁC ĐẶC TRƯNG CỦA NGUỒN NUÔI**

Chất lượng của một bộ nguồn phụ thuộc vào độ ổn định tải, độ ổn định nguồn và trở kháng ra của nó. Chúng ta hãy xem xét các đặc trưng này của bộ nguồn.

#### **Độ ổn định tải (Load Regulation)**

Hình 11-1 cho thấy bộ nguồn với mạch nắn cầu và bộ lọc dùng tụ C.



Hình 11-1

Thay đổi tải  $R_L$  sẽ làm thay đổi thế tải. Nếu trở tải nhỏ sẽ làm cho độ gợn sóng trên tải tăng, sụt thế qua các diode và cuộn thứ cấp cũng sẽ tăng. Do đó một sự giảm trở tải sẽ làm giảm thế tải. Độ ổn định tải cho biết thế tải thay đổi ra sao khi dòng tải thay đổi.

Định nghĩa độ ổn định tải

$$\text{LOAD REGULATION} = (V_{NL} - V_{FL}) \times 100\% / V_{FL} \quad (11-1)$$

Trong đó  $V_{NL}$  là thế trên tải khi dòng tải bằng 0

$V_{FL}$  là thế trên tải khi dòng tải tối đa

Ví dụ mạch hình 11-1 có

$$V_{NL} = 10.6\text{V} \text{ khi } I_L = 0$$

$$V_{FL} = 9.25\text{V} \text{ khi } I_L = 1\text{A}$$

$$\text{Thì Load Regulation} = (10.6 - 9.25) \times 100\% / 9.25 = 14.6\%$$

Độ ổn định tải càng bé thì bộ nguồn càng tốt. Các bộ nguồn ổn áp tốt có độ ổn định tải bé hơn 1%, nghĩa là thế tải thay đổi ít hơn 1% trong toàn miền của dòng tải.

### **Độ ổn định lưới (Line Regulation)**

Trên hình 11-1 nguồn vào có giá trị danh định là 120V (tại Mỹ). Giá trị thực tại ổ cắm có thể thay đổi từ 105V đến 125V phụ thuộc vào thời gian, địa điểm và một số yếu tố khác. Do thế thứ cấp tỷ lệ trực tiếp với thế vào do đó thế tải sẽ thay đổi khi thế lưới thay đổi.

Độ ổn định lưới định nghĩa như sau:

$$\text{LINE REGULATION} = (V_{\text{HL}} - V_{\text{LL}}) \times 100\% / V_{\text{LL}} \quad (11-2)$$

Trong đó  $V_{\text{HL}}$  là thế tải khi thế nguồn cao.

$V_{\text{LL}}$  là thế tải khi thế nguồn thấp.

Ví dụ  $V_{\text{HL}} = 11.2\text{V}$  khi thế lưới là 125V,  $V_{\text{LL}} = 9.2\text{V}$  khi thế lưới là 105V

Độ ổn định lưới =  $(11.2 - 9.2) \times 100\% / 9.2 = 21.7\%$

Cũng như độ ổn định tải, độ ổn định lưới càng bé càng tốt. Các bộ nguồn ổn áp tốt có độ ổn định lưới bé hơn 0.1%.

### **ĐIỆN TRỞ RA**

Điện trở Thevenin hay điện trở ra của bộ nguồn quy định bởi độ ổn định tải.

Một bộ nguồn có trở ra thấp thì độ ổn định tải của nó cũng sẽ thấp.

Điện trở ra của một bộ nguồn tính theo công thức sau:

$$R_{\text{TH}} = (V_{\text{NL}} - V_{\text{FL}}) / I_{\text{FL}} \quad (11-3)$$

Chẳng hạn đối với mạch hình 11-1, ta có

$V_{\text{NL}} = 10.6\text{V}$  khi  $I_{\text{L}} = 0$

$V_{\text{FL}} = 9.25\text{V}$  khi  $I_{\text{L}} = 1\text{A}$

$R_{\text{TH}} = (10.6 - 9.25) / 1 = 1.35\Omega$

Biểu thức tương đương cho độ ổn định tải là

$$\text{Độ ổn định tải} = R_{\text{TH}} \times 100\% / R_{\text{L}(\text{min})} \quad (11-4)$$

Ví dụ, một bộ nguồn có trở ra  $1.5\Omega$  và trở tải tối thiểu là  $10\Omega$  thì độ ổn định tải bằng

$$\text{Độ ổn định tải} = (1.5/10) \times 100\% = 15\%$$

## XI.2 CÁC ỔN ÁP MẮC SHUNT

Độ ổn định lưới và độ ổn định tải của một bộ nguồn không ổn áp là quá cao đối với hầu hết các ứng dụng. Bằng cách sử dụng bộ ổn áp (Voltage Regulator) giữa bộ nguồn và tải chúng ta có thể cải thiện một cách đáng kể độ ổn định lưới và độ ổn định tải.

Một bộ ổn áp thế kiểu tuyến tính dùng các linh kiện trong vùng tuyến tính để giữ thế trên tải là hằng số. Có 2 loại ổn áp tuyến tính:

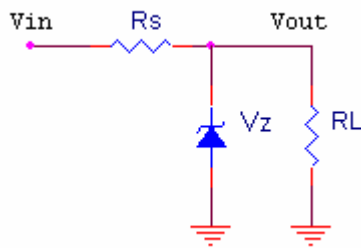
Ổn áp tuyến tính mắc shunt

Ổn áp tuyến tính mắc nối tiếp

Trong phần này chúng ta xét ổn áp tuyến tính mắc shunt, một loại ổn áp mà mạch ổn áp mắc song song với tải.

### ỔN ÁP ZENER

Ổn áp mắc shunt đơn giản nhất là mạch diode Zener như hình 11-2.



Hình 11-2

Trong mạch này diode Zener hoạt động trong vùng đánh thủng. Khi dòng tải thay đổi, dòng qua Zener tăng hoặc giảm một cách tương ứng để giữ cho thế tải không đổi. Trên hình 11-2, dòng qua  $R_S$  bằng

$$I_S = (V_{in} - V_{out}) / R_S$$

Thế trên tải bằng

$$V_{out} = V_Z$$

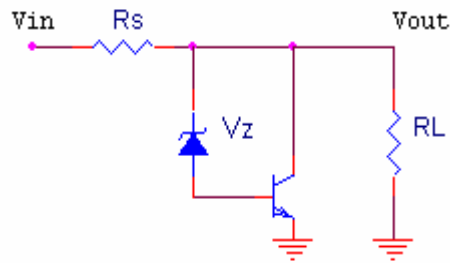
Khi thế vào là không đổi, dòng vào gần như không đổi khi dòng tải thay đổi.

Dòng trên tải

$$I_L = I_S - I_Z$$

Dòng tải có giá trị tối đa bằng dòng vào và khi dòng qua Zener bằng 0.

Khi dòng tải lớn, mạch hình 11-2 có độ ổn định tải kém vì sự thay đổi của dòng qua điện trở Zener có thể làm thay đổi thế ra đáng kể. Một cách để cải thiện độ ổn định tải khi dòng tải lớn là thêm một transistor như hình 11-3.



Hình 11-3

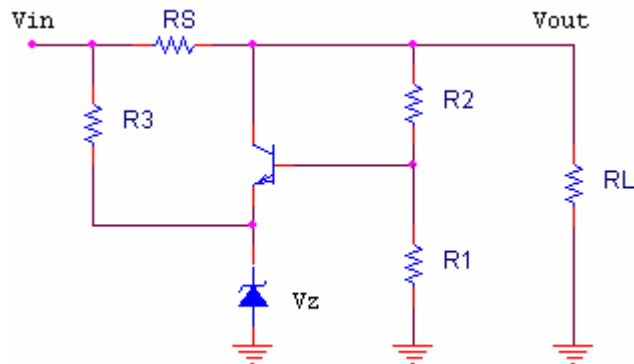
Với mạch này thế ra bằng

$$V_{out} = V_Z + V_{BE} \quad (11-5)$$

Nếu thế ra tăng, thế phản hồi về base của transistor tăng làm cho sụt thế qua  $R_S$  tăng. Kết quả là thế ra lại giảm.

### THẾ RA CAO HƠN

Hình 11-4 cho thấy một mạch ổn áp có thế ra cao mặc dù sử dụng Zener có thế đánh thủng thấp.



Hình 11-4



Mạch dùng phản hồi âm. Mọi thay đổi trên thế lối ra được phản hồi về base của transistor.

Thế base của transistor cho bởi công thức

$$V_B = R_1 V_{out} / (R_1 + R_2)$$

Do đó thế ra bằng

$$V_{out} = (R_1 + R_2) V_B / R_1$$

Trên hình 11-4, thế cực base bằng

$$V_B = V_Z + V_{BE}$$

Vậy

$$V_{out} = (R_1 + R_2) (V_Z + V_{BE}) / R_1 \quad (11-6)$$

Một cách để loại trừ ảnh hưởng của  $V_{BE}$  vào thế ra thay transistor bằng OP AMP. Với OP AMP có hệ số khuếch đại lớn, thế ra cho bởi

$$V_{out} = (R_1 + R_2) V_Z / R_1 \quad (11-7)$$

### BẢO VỆ NGẮN MẠCH

Ưu điểm của các mạch ổn áp mắc shunt là tự bảo vệ ngắn mạch. Khi ngắn mạch dòng qua  $R_S$  là tối đa và bằng

$$V_{in} / R_S$$

Hiệu suất của ổn áp

Hiệu suất

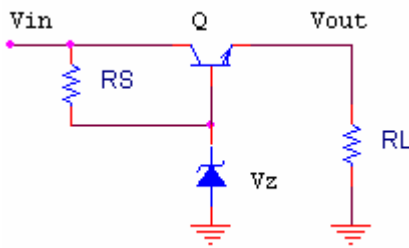
$$\eta = (P_{out} / P_{in}) \times 100\% \quad (11-8)$$

### XI.3 CÁC ỔN ÁP MẮC NỐI TIẾP

Nhược điểm của ổn áp mắc shunt là hiệu suất thấp. Để nâng cao hiệu suất người ta dùng ổn áp nối tiếp hoặc ổn áp xung. Ổn áp xung cho hiệu suất cao từ 75% đến 95%. Nhưng ổn áp xung lại tạo ra nhiều tần số cao RFI (radio frequency interference). Mặc khác ổn áp xung khá phức tạp.

Các ổn áp mắc nối tiếp là thích hợp cho các ứng dụng cần công suất bé hơn 10W vì thiết kế đơn giản, hiệu suất từ 50 đến 70%. Trong các ổn áp loại này transistor hoạt động trong miền tác động.

Ổn áp nối tiếp đơn giản nhất là mạch lặp lại zener như hình 11-5.



Hình 11-5

Thế ra của mạch ổn áp bằng

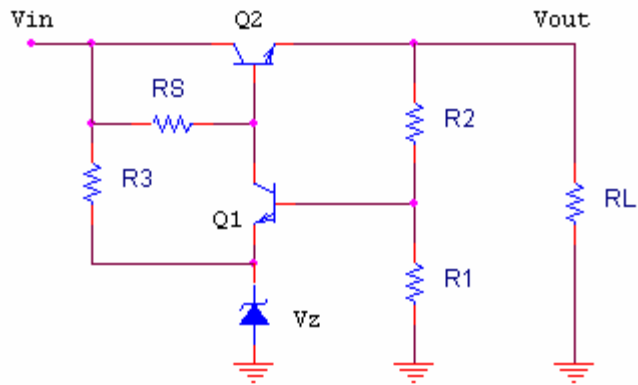
$$V_{\text{out}} = V_Z + V_{\text{BE}} \quad (11-9)$$

Nếu thế vào hoặc dòng tải thay đổi, thế zener và thế emitter base sẽ thay đổi chút ít. Trong ổn áp mắc nối tiếp dòng tải xấp xỉ dòng vào vì dòng qua Zener khá bé. Transistor trong mạch gọi là transistor thông dẫn vì dòng tải chảy qua nó.

Hiệu suất của mạch này lớn hơn mạch mắc shunt vì chúng ta đã thay điện trở bởi transistor. Dòng vào của ổn áp nối tiếp thay đổi theo dòng tải chứ không phải là hằng số như ổn áp mắc shunt.

#### ỔN ÁP HAI TRANSISTOR

Hình 11-6 cho thấy mạch ổn áp dùng 2 transistor.



Hình 11-6

Nếu thế ra tăng do thế vào tăng hoặc do dòng tải giảm thì thế phản hồi về  $Q_1$  tăng làm tăng dòng qua  $Q_1$  và vì vậy làm giảm thế base của  $Q_2$ . Kết quả là thế lối ra giảm.

Thế ra của mạch cho bởi

$$V_{out} = (R_1 + R_2)(V_Z + V_{BE}) / R_1 \quad (11-10)$$

Dòng qua transistor  $Q_2$  bằng

$$I_C = I_L + I_2$$

Trong đó  $I_2$  là dòng chảy qua cầu phân thế  $R_1$  và  $R_2$ .

Thường  $I_2$  rất bé so với  $I_L$  vì vậy  $I_C = I_L$ .

Công suất tiêu tán

$$P_D = (V_{in} - V_{out})I_L \quad (11-12)$$

Hiệu suất

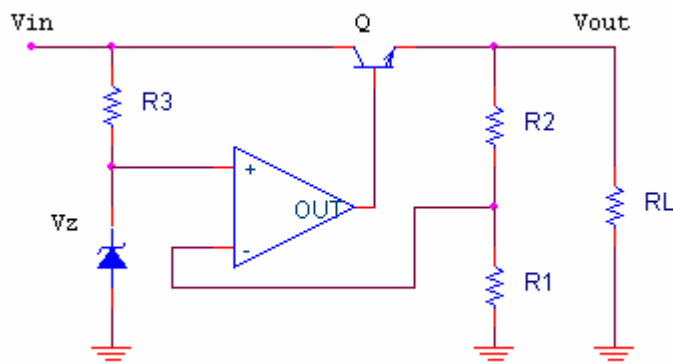
$$\eta = (V_{out} / V_{in}) \times 100\% \quad (11-13)$$

### ỔN ÁP CẢI TIẾN

Hình 11-7 cho thấy một ổn áp nối tiếp cải tiến nhờ sử dụng OP AMP.

Thế ra bằng

$$V_{out} = (R_1 + R_2)V_Z / R_1 \quad (11-14)$$



Hình 11-7

#### XI.4 CÁC IC ỔN ÁP TUYẾN TÍNH

Có nhiều ổn áp tuyến tính được tích hợp trong IC. Tất cả chúng là ổn áp nối tiếp. Thông thường các IC này có 3 chân: Chân điện áp vào chưa ổn áp, chân điện áp ra đã ổn áp và chân GND. Các ổn áp 3 chân có vỏ bằng nhựa hoặc vỏ kim loại này ngày càng phổ biến vì rẻ tiền và dễ dùng. Các ổn áp IC 3 chân này có thể cho điện áp ra dương hoặc âm cố định từ 5 đến 24V với dòng tải đến 1A. Một số ổn áp cho phép điều chỉnh điện áp lối ra từ 2 đến 40V.

Đối với các IC ổn áp, nhà máy định nghĩa lại hệ số ổn định nguồn và tải như sau:

Độ ổn định nguồn =  $\Delta V_{out}$  đối với toàn miền của dòng tải

Độ ổn định tải =  $\Delta V_{out}$  đối với toàn miền của thế vào

Bảng 11-1 cho thấy một số các IC ổn áp điển hình và các thông số của chúng.

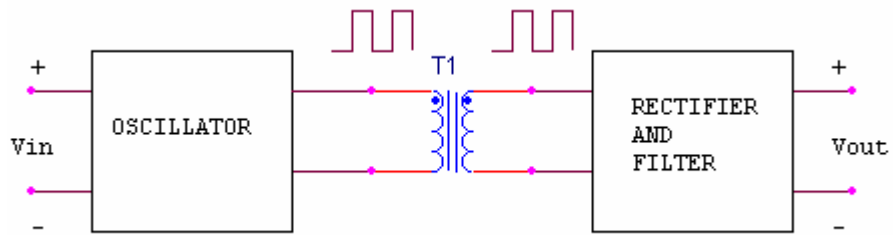
Lưu ý: thế drop out là thế sụt qua IC ổn áp bé nhất mà ổn áp còn hoạt động được. Nói cách khác thế vào tối thiểu của ổn áp IC phải bằng thế ra danh định cộng với thế drop out. Ví dụ với LM7805 thế vào tối thiểu là 8V.

| Number  | V <sub>out</sub> (V) | I <sub>max</sub> (A) | Load Reg (mV) | Line Reg | Drop out (V) |
|---------|----------------------|----------------------|---------------|----------|--------------|
| LM7805  | 5                    | 1.5                  | 10            | 3        | 2            |
| LM7806  | 6                    | 1.5                  | 12            | 5        | 2            |
| LM7808  | 8                    | 1.5                  | 12            | 6        | 2            |
| LM7812  | 12                   | 1.5                  | 12            | 4        | 2            |
| LM7815  | 15                   | 1.5                  | 12            | 4        | 2            |
| LM7818  | 18                   | 1.5                  | 12            | 15       | 2            |
| LM7811  | 11                   | 1.5                  | 12            | 18       | 2            |
| LM78L05 | 5                    | 0.1                  | 20            | 18       | 1.7          |
| LM78L12 | 12                   | 0.1                  | 30            | 30       | 1.7          |
| LM2931  | 3 to 11              | 0.1                  | 14            | 4        | 0.3          |
| LM7905  | -5                   | 1.5                  | 10            | 3        | 2            |
| LM7912  | -12                  | 1.5                  | 12            | 4        | 2            |
| LM7915  | -15                  | 1.5                  | 12            | 4        | 2            |
| LM317   | 1.2 to 37            | 1.5                  | 0.3%          | 0.02%/V  | 2            |
| LM337   | -1.2 to -37          | 1.5                  | 0.3%          | 0.01%/V  | 2            |
| LM338   | 1.2 to 32            | 5                    | 0.3%          | 0.02%/V  | 2.7          |

## XI.5 CÁC BỘ BIẾN ĐỔI DC - DC

Đôi khi chúng ta cần biến đổi một thế DC này thành một thế DC khác. Ví dụ biến đổi từ 5V thành 12V. Các bộ biến đổi DC-DC là rất hiệu quả vì chúng sử dụng transistor ở chế độ công tắc. Trong phần này chúng ta sẽ phân tích bộ biến đổi DC-DC không ổn áp. Phần sau sẽ phân tích bộ biến đổi DC-DC ổn áp sử dụng sự biến điệu độ rộng xung hay còn gọi là ổn áp xung (Switching Regulators)

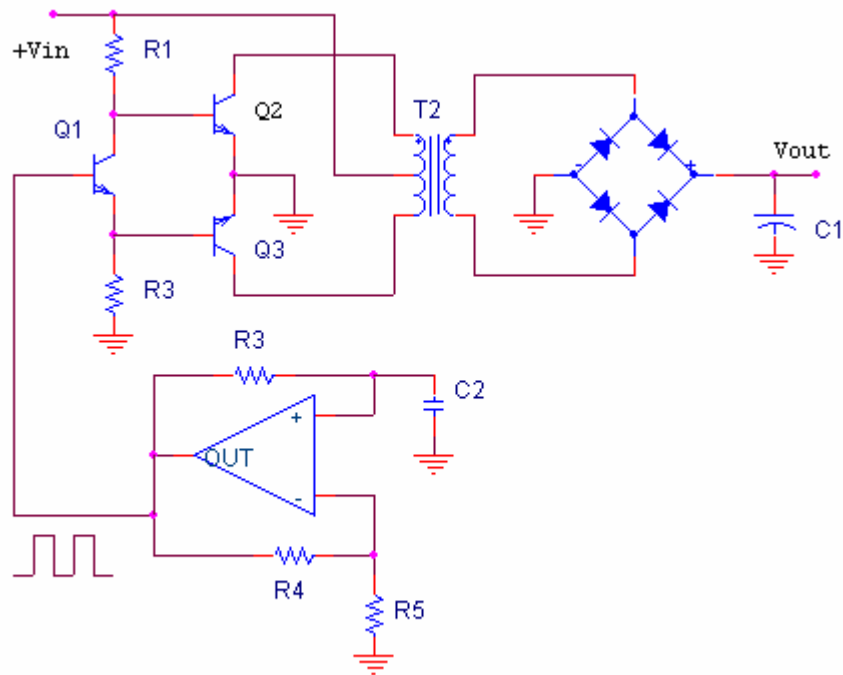
Một bộ biến đổi DC-DC căn bản (xem hình 11-8) gồm một bộ tạo xung vuông được nuôi bởi thế DC cần biến đổi. Sóng vuông này (có giá trị đỉnh bằng giá trị của nguồn DC vào) kích thích cuộn sơ cấp biến áp. Tại thứ cấp của biến áp chúng ta cũng nhận được xung vuông nhưng có biên độ lớn hơn hoặc bé hơn sóng vuông điều khiển. Qua mạch nắn và lọc chúng ta nhận được thế DC cần thiết.



Hình 11-8

Tần số hoạt động của bộ biến đổi (sóng vuông) là khoảng 10 KHz đến 100KHz.

Hình 11-9 là một mạch biến đổi DC-DC không ổn áp, sử dụng transistor.



Hình 11-9

Bộ tạo sóng vuông dùng OP AMP hoạt động tại tần số 20KHz. Q<sub>1</sub> là mạch tách pha để tạo 2 tín hiệu ngược pha kích thích cặp transistor đẩy kéo Q<sub>2</sub> và Q<sub>3</sub>.

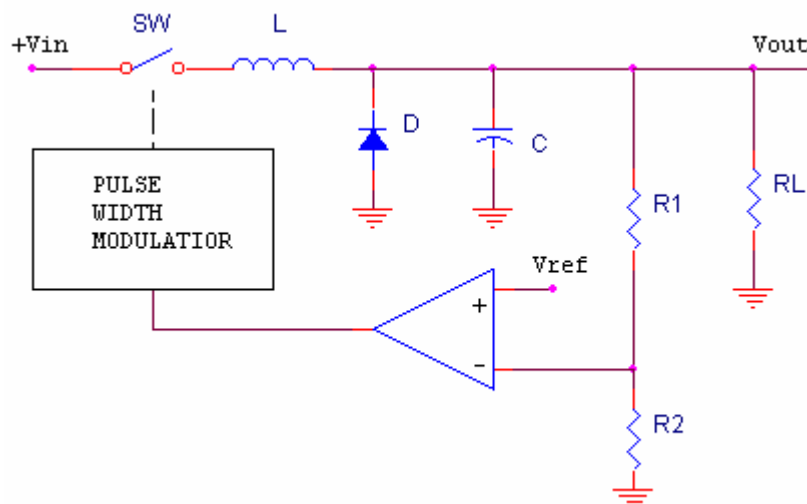
Bộ biến đổi DC-DC không ổn áp cần phải có thể lõi vào ổn định (lỗi ra của ổn áp chẳng hạn)

## XI.6 NGUỒN ỔN ÁP XUNG

Ổn áp xung thuộc loại biến đổi DC-DC nhưng sử dụng mạch ổn áp bên trong. Người ta thường dùng sự biến điệu độ rộng xung (Pulse Width Modulation PWM) để điều khiển thời gian on/off của transistor. Bằng cách thay đổi độ rộng của xung có thể ổn định điện áp lõi ra khi dòng tải hoặc thế nguồn thay đổi.

Trong ổn áp nối tiếp, transistor thông dẫn luôn luôn dẫn do đó công suất tiêu tán lớn và hiệu suất thấp. Để giảm công suất tiêu tán và nâng cao hiệu suất, một xung vuông điều khiển transistor thông dẫn chuyển giữa 2 chế độ bão hoà hoặc ngưng dẫn. Khi transistor ngưng dẫn công suất tiêu tán bằng 0. Khi transistor bão hoà công suất tiêu tán rất thấp vì sụt thế qua transistor bão hoà rất bé. Do vậy ổn áp xung có thể đạt hiệu suất đến 95%. Vì hiệu suất cao và kích thước bé nên ổn áp xung dùng nhiều trong các thiết bị điện tử thế hệ mới.

Có nhiều cấu hình cho ổn áp xung tùy theo dải công suất và yêu cầu về chất lượng. Hình 11-10 là một ổn áp xung thông thường.



Hình 11-10

Switch là transistor lưỡng cực hoặc FET công suất. Bộ biến điệu độ rộng xung là dao động sóng vuông có tần số cố định (từ 10 KHz đến 100KHz) nhưng độ xấp (tỷ số giữa thời gian xung ở mức cao và thời gian xung ở mức thấp) thay đổi được. Mạch so sánh dùng OP AMP.

Ban đầu khi power on, thế ra của mạch bằng 0. Khi đó thế phản hồi về lối vào – của OP AMP cũng bằng 0. Thế ra của OP AMP rất lớn làm cho độ xấp của xung là cực đại. Tại phần dương của xung điều khiển, SW đóng. Lúc này diode bị phân cực ngược và dòng vào chảy qua cuộn L. Dòng này tạo ra từ trường xung quanh cuộn L. Năng lượng từ trường cho bởi

$$W=0.5 LI^2$$

Dòng điện này nạp cho tụ C và chảy qua tải. Khi xung điều khiển có mức thấp, SW hở. Lúc này dòng qua cuộn giảm làm từ trường qua cuộn L cũng giảm làm xuất hiện thế tự cảm ngược chiều. Thế tự cảm này phân cực thuận diode và duy trì dòng qua cuộn như ban đầu. Dòng này chảy qua tải cũng với hướng như khi SW đóng. Nói cách khác khi SW hở cuộn cảm đóng vai trò như nguồn và tiếp tục cung cấp dòng qua tải cho đến khi cuộn trả hết năng lượng cho mạch hoặc khi SW đóng lại. Quá trình cứ thế tiếp tục và trên tải có dòng DC.

Giá trị trung bình của thế ra bằng

$$V_{out} = D V_{in} \quad (11-15)$$

Trong đó D là độ xấp của xung. Bằng cách kiểm soát D có thể làm cho  $V_{out}$  không thay đổi.

Vì OP AMP có hệ số khuếch đại rất lớn và ngắn mạch ảo giữa 2 lối vào nên

$$V_{FB} = V_{REF}$$

Do đó

$$V_{out} = (R_1 + R_2) V_{REF} / R_1 \quad (11-16)$$

Các bộ nguồn ổn áp kiểu xung thế hệ mới sử dụng khuếch đại đẩy kéo và biến áp xung cách ly giữa sơ cấp và thứ cấp của bộ nguồn. Trong trường hợp đó phải thực hiện nối quang học (dùng phototransistor) để phản hồi thế lối



ra về lối vào. Bộ nguồn của TV, máy tính, màn hình và máy in hiện nay đều là bộ ổn áp xung có công suất từ vài chục đến vài trăm W. Thế vào ac được nắn lọc trực tiếp để tạo ra nguồn dc chưa ổn áp. Nguồn dc chưa ổn áp này sau đó lại được biến đổi DC-DC kiểu xung nhằm tạo ra các điện thế thấp và ổn định cho các mạch điện tử. Tùy theo ứng dụng, các điện áp dc ở lối ra của các bộ nguồn kiểu xung có thể là 3.3V, 5V, 12V và một số các điện áp dc khác.

.....

# Giáo trình Điện tử cơ bản/Giới thiệu chung

Bài từ Tủ sách Khoa học VLOS.

Cập nhật 12:40, 17/8/2009, bởi [VLoSer](#)

Jump to: [navigation](#), [search](#)

## Mục lục

[[giấu](#)]

- [1 Giới thiệu chung](#)
  - [1.1 Khái niệm chung](#)
  - [1.2 Tổng quan về các hệ thống trang thiết bị điện tử](#)
- [2 Kết luận chương](#)
- [3 Tài liệu tham khảo](#)
  - [3.1 Bản quyền](#)

## Giới thiệu chung

### Khái niệm chung

Để bắt đầu với khái niệm liên quan tới lĩnh vực điện tử, chúng ta quay lại với những khái niệm cơ bản của điện học trước khi đề cập tới điện tử và các linh kiện ứng dụng. Để tiện cho việc tổng kết các khái niệm đó, phần dưới sẽ liệt kê một loạt các khái niệm cơ bản liên quan tới lĩnh vực điện-điện tử. Chúng bao gồm:

**Khái niệm dòng điện:** Một dòng điện là dòng chuyển dời có hướng của các hạt mang điện. Đó chính là dòng chuyển dời của các electron qua các dây dẫn và các linh kiện điện tử. Chúng ta có thể hình dung sự chuyển dời của các electron này tương tự quá trình chảy của một dòng nước thông qua các ống dẫn nước. Nước là quá trình chuyển dời của các phân tử nước trong các ống dẫn nước dưới tác động của máy bơm nước thì dòng điện chuyển dời trong các dây dẫn được thực hiện dưới tác động của nguồn pin.

Vậy nguồn pin tác động như thế nào để tạo ra dòng điện? Như đã biết các hạt đồng dấu thì đẩy nhau, các hạt trái dấu thì hút nhau. Cụ thể các hạt mang điện tích cùng dương hoặc cùng âm sẽ đẩy nhau còn các hạt mang điện tích trái dấu thì hấp dẫn lẫn nhau. Chính nhờ lý do đó, một nguồn pin sẽ có hai cực. Cực âm sẽ đẩy các electron đi vào trong dây dẫn và cực dương sẽ hút các electron về phía nó. Do đó, nếu tạo thành một vòng kín, thì các electron sẽ chuyển dời theo một hướng xác định và đó chính là dòng điện.

Dòng điện này có chiều không đổi nên được gọi là *dòng một chiều*.

Trong trường hợp, hai cực của nguồn pin tuân tự đổi cực tính từ dương sang âm và từ âm sang dương thì lúc này dòng điện trong dây dẫn cũng lần lượt đổi chiều tương ứng với sự đổi cực của các điện cực. Dòng điện này chính là *dòng xoay chiều*.

### **Khái niệm điện áp (hay hiệu điện thế):**

Điện áp hay hiệu điện thế là giá trị chênh lệch điện thế giữa hai điểm. Cũng tương tự như dòng điện, điện áp có 2 loại điện áp một chiều và điện áp xoay chiều. Điện áp một chiều là sự chênh lệch điện thế giữa hai điểm mà tại đó sự chênh lệch điện thế tạo ra các dòng điện một chiều. Điện áp xoay chiều tương ứng với trường hợp sự thay đổi liên tục về cực tính giữa hai điểm tương ứng và điều này chính là nguyên nhân tạo ra sự thay đổi chiều dòng điện và chúng ta có dòng điện xoay chiều.

"Các linh kiện cơ bản"

Trong các mạch điện tử, các linh kiện cơ bản được biết đến chính là [điện trở](#), [tụ điện](#) và [cuộn cảm](#). Đây là 3 linh kiện cơ bản chủ yếu và không thể thiếu trong bất kỳ một mạch điện tử nào. Mỗi linh kiện đều có các đặc trưng riêng của nó. Ví dụ như điện trở là khái niệm vật lý đặc trưng cho tính chất cản trở dòng điện của một vật dẫn điện nên linh kiện điện trở trong mạch thường dùng để hạn chế các dòng điện trong mạch. Việc sử dụng điện trở trong mạch điện tử nếu được dùng đúng cách và đúng các giá trị điện trở thì mạch điện sẽ hoạt động một cách tối ưu, còn ngược lại, nếu sử dụng một cách tùy tiện sẽ gây ra những sai số không đáng có trong một mạch điện tử. Chính vì thế, trong các mạch điện tử, việc lựa chọn giá trị điện trở cũng là một vấn đề cần lưu tâm trong quá trình thiết kế mạch. Cũng tương tự đối với các linh kiện tụ điện và cuộn cảm. Chi tiết về cách sử dụng các linh kiện này sẽ được đề cập chi tiết trong phần sau.

### **Các định luật cơ bản**

Một trong các định luật cơ bản nhất trong điện học đó chính là định luật Ôm và nó được biểu diễn bởi công thức  $U=I.R$ . trong đó  $U$  là hiệu điện thế giữa hai đầu điện trở đo bằng Vôn (ký hiệu  $V$ ),  $I$  là cường độ dòng điện đi qua điện trở (đo bằng Ampe (ký hiệu  $A$ ) và  $R$  là điện trở lắp trong mạch (đo bằng Ôm, ký hiệu  $\Omega$ ). Trong trường hợp dòng điện là xoay chiều thì khái niệm điện trở được gọi một cách tổng quát là tổng trở kháng, trong đó điện trở được gọi là trở kháng thuần, điện dung của tụ điện được gọi là dung kháng và điện cảm của cuộn cảm được gọi là cảm kháng.

### **Tổng quan về các hệ thống trang thiết bị điện tử**

Trong phần này, sẽ trình bày về các cấu trúc chung của các hệ thống trang thiết bị điện tử dưới dạng sơ đồ khối để rồi từ đó, chúng ta có thể dành thời gian đi sâu phân tích thêm về các khối chức năng khác nhau trong mạch điện tử.

# **Giáo trình Điện tử cơ bản/Các linh kiện điện tử cơ bản**

**Bài từ Tủ sách Khoa học VLOS.**

## Mục lục

[[giấu](#)]

- [1 Các linh kiện điện tử cơ bản](#)
  - [1.1 Phân loại điện trở và cách đọc điện trở](#)
  - [1.2 Phân loại tụ điện và cách đọc tụ điện](#)
    - [1.2.1 Tu hoá](#)
    - [1.2.2 Tu Tantal](#)
    - [1.2.3 Tụ không phân cực](#)
    - [1.2.4 Tụ điện biến đổi](#)
    - [1.2.5 Tụ chặn](#)
  - [1.3 Cuộn cảm](#)
- [2 Một số các phương pháp kiểm tra thông thường](#)
- [3 Tóm tắt chương](#)
- [4 Tài liệu tham khảo](#)
  - [4.1 Bản quyền](#)

## Các linh kiện điện tử cơ bản

Như đã đề cập trong phần trước, các linh kiện điện tử cơ bản trong một mạch điện tử bao gồm: [điện trở](#), [tụ điện](#), [cuộn cảm](#). Do đây là các linh kiện cơ bản nên việc đầu tiên khi làm quen với các linh kiện này đó là cách nhận biết các loại linh kiện khác nhau, đồng thời đọc được giá trị các loại linh kiện khác nhau.

### Phân loại điện trở và cách đọc điện trở

Như đã đề cập, nói một cách nôm na, điện trở đặc trưng cho tính chất cản trở dòng điện. Chính vì thế, khi sử dụng điện trở cho một mạch điện thì một phần năng lượng điện sẽ bị tiêu hao để duy trì mức độ chuyển dời của dòng điện. Nói một cách khác thì khi điện trở càng lớn thì dòng điện đi qua càng nhỏ và ngược lại khi điện trở nhỏ thì dòng điện dễ dàng được truyền qua. Khi dòng điện cường độ  $I$  chạy qua một vật có điện trở  $R$ , điện năng được chuyển thành nhiệt năng với công suất theo phương trình sau:

$$P = I^2 \cdot R$$

trong đó:

$P$  là công suất, đo theo  $W$

$I$  là cường độ dòng điện, đo bằng  $A$

$R$  là điện trở, đo theo  $\Omega$

Chính vì lý do này, khi phân loại điện trở, người ta thường dựa vào công suất mà phân loại điện trở. Và theo cách phân loại dựa trên công suất, thì điện trở thường được chia làm 3 loại:

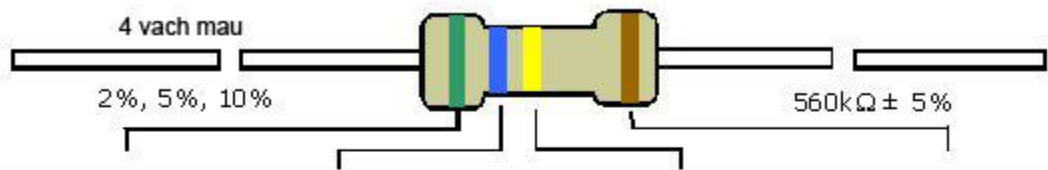
- Điện trở công suất nhỏ
- Điện trở công suất trung bình
- Điện trở công suất lớn.

Tuy nhiên, do ứng dụng thực tế và do cấu tạo riêng của các vật chất tạo nên điện trở nên thông thường, điện trở được chia thành 2 loại:

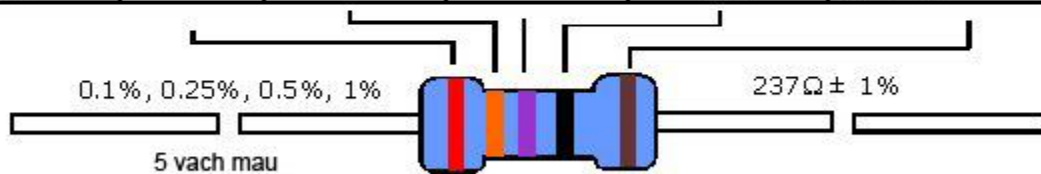
- Điện trở: là các loại điện trở có công suất trung bình và nhỏ hay là các điện trở chỉ cho phép các dòng điện nhỏ đi qua.
- Điện trở công suất: là các điện trở dùng trong các mạch điện tử có dòng điện lớn đi qua hay nói cách khác, các điện trở này khi mạch hoạt động sẽ tạo ra một lượng nhiệt năng khá lớn. Chính vì thế, chúng được cấu tạo nên từ các vật liệu chịu nhiệt.

Để tiện cho quá trình theo dõi trong tài liệu này, các khái niệm điện trở và điện trở công suất được sử dụng theo cách phân loại trên.

Cách đọc giá trị các điện trở này thông thường cũng được phân làm 2 cách đọc, tùy theo các ký hiệu có trên điện trở. Dưới đây là hình vẽ cách đọc điện trở theo vạch màu trên điện trở.



| MAU          | Vạch 1 | Vạch 2 | Vạch 3 | He so | Dung sai |     |
|--------------|--------|--------|--------|-------|----------|-----|
| Đen          | 0      | 0      | 0      | 1Ω    |          |     |
| Nâu          | 1      | 1      | 1      | 10Ω   | ± 1%     | (F) |
| Đỏ           | 2      | 2      | 2      | 100Ω  | ± 2%     | (G) |
| Cam          | 3      | 3      | 3      | 1KΩ   |          |     |
| Vàng         | 4      | 4      | 4      | 10KΩ  |          |     |
| Xanh lá cây  | 5      | 5      | 5      | 100KΩ | ±0.5%    | (D) |
| Xanh da trời | 6      | 6      | 6      | 1MΩ   | ±0.25%   | (C) |
| Tím          | 7      | 7      | 7      | 10MΩ  | ±0.10%   | (B) |
| Xám          | 8      | 8      | 8      |       | ±0.05%   |     |
| Trắng        | 9      | 9      | 9      |       |          |     |
| Vàng         |        |        |        | 0.1   | ± 5%     | (J) |
| Bạc          |        |        |        | 0.01  | ± 10%    | (K) |



Nguồn **Electronix Express / RSR**  
<http://www.elexp.com>

Đối với các điện trở có giá trị được định nghĩa theo vạch màu thì chúng ta có 3 loại điện trở: Điện trở 4 vạch màu và điện trở 5 vạch màu và 6 vạch màu. Loại điện trở 4 vạch màu và 5 vạch màu được chỉ ra trên hình vẽ. Khi đọc các giá trị điện trở 5 vạch màu và 6 vạch màu thì chúng ta cần phải để ý một chút vì có sự khác nhau một chút về các giá trị. Tuy nhiên, cách đọc điện trở màu đều dựa trên các giá trị màu sắc được ghi trên điện trở 1 cách tuần tự:

#### Đối với điện trở 4 vạch màu

- Vạch màu thứ nhất: Chỉ giá trị hàng chục trong giá trị điện trở
- Vạch màu thứ hai: Chỉ giá trị hàng đơn vị trong giá trị điện trở
- Vạch màu thứ ba: Chỉ hệ số nhân với giá trị số mũ của 10 dùng nhân với giá trị điện trở
- Vạch màu thứ 4: Chỉ giá trị sai số của điện trở

#### Đối với điện trở 5 vạch màu

- Vạch màu thứ nhất: Chỉ giá trị hàng trăm trong giá trị điện trở
- Vạch màu thứ hai: Chỉ giá trị hàng chục trong giá trị điện trở
- Vạch màu thứ ba: Chỉ giá trị hàng đơn vị trong giá trị điện trở
- Vạch màu thứ 4: Chỉ hệ số nhân với giá trị số mũ của 10 dùng nhân với giá trị điện trở
- Vạch màu thứ 5: Chỉ giá trị sai số của điện trở

Ví dụ như trên hình vẽ, điện trở 4 vạch màu ở phía trên có giá trị màu lần lượt là: xanh lá cây/xanh da trời/vàng/nâu sẽ cho ta một giá trị tương ứng như bảng màu lần lượt là 5/6/4/1%. Ghép các giá trị lần lượt ta có  $56 \times 10^4 \Omega = 560 \text{k}\Omega$  và sai số điện trở là 1%.

Tương tự điện trở 5 vạch màu có các màu lần lượt là: Đỏ/cam/tím/đen/nâu sẽ tương ứng với các giá trị lần lượt là 2/3/7/0/1%. Như vậy giá trị điện trở chính là  $237 \times 10^0 = 237 \Omega$ , sai số 1%.

## Phân loại tụ điện và cách đọc tụ điện

Tụ điện theo đúng tên gọi chính là linh kiện có chức năng tích tụ năng lượng điện, nói một cách nôm na. Chúng thường được dùng kết hợp với các điện trở trong các mạch định thời bởi khả năng tích tụ năng lượng điện trong một khoảng thời gian nhất định. Đồng thời tụ điện cũng được sử dụng trong các nguồn điện với chức năng làm giảm độ gợn sóng của nguồn trong các nguồn xoay chiều, hay trong các mạch lọc bởi chức năng của tụ nói một cách đơn giản đó là tụ ngắn mạch (cho dòng điện đi qua) đối với dòng điện xoay chiều và hở mạch đối với dòng điện 1 chiều.

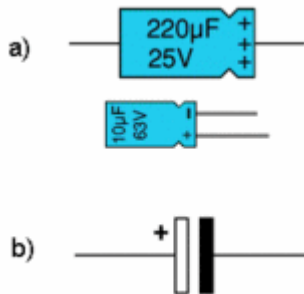
Trong một số các mạch điện đơn giản, để đơn giản hóa trong quá trình tính toán hay thay thế tương đương thì chúng ta thường thay thế một tụ điện bằng một dây dẫn khi có dòng xoay chiều đi qua hay tháo tụ ra khỏi mạch khi có dòng một chiều trong mạch. Điều này khá là cần thiết khi thực hiện tính toán hay xác định các sơ đồ mạch tương đương cho các mạch điện tử thông thường.

Hiện nay, trên thế giới có rất nhiều loại tụ điện khác nhau nhưng về cơ bản, chúng ta có thể chia tụ điện thành hai loại: Tụ có phân cực (có cực xác định) và tụ điện không phân cực (không xác định cực dương âm cụ thể).

Để đặc trưng cho khả năng tích trữ năng lượng điện của tụ điện, người ta đưa ra khái niệm là điện dung của tụ điện. Điện dung càng cao thì khả năng tích trữ năng lượng của tụ điện càng lớn và ngược lại. Giá trị điện dung được đo bằng đơn vị Farad (kí hiệu là F). Giá trị F là rất lớn nên thông thường trong các mạch điện tử, các giá trị tụ chỉ đo bằng các giá trị nhỏ hơn như micro fara ( $\mu\text{F}$ ), nano Fara (nF) hay pico Fara (pF).

$$1\text{F} = 10^6 \mu\text{F} = 10^9 \text{nF} = 10^{12} \text{pF}$$

## Tụ hoá

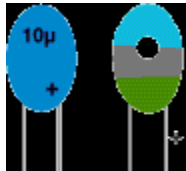


Kí hiệu tụ hoá và hình dạng tụ hoá

Tụ hóa là một loại tụ có phân cực. Chính vì thế khi sử dụng tụ hóa yêu cầu người sử dụng phải cắm đúng chân của tụ điện với điện áp cung cấp. Thông thường, các loại tụ hóa thường có kí hiệu chân cụ thể cho người sử dụng bằng các ký hiệu + hoặc = tương ứng với chân tụ.

Có hai dạng tụ hóa thông thường đó là tụ hóa có chân tại hai đầu trụ tròn của tụ (tụ có ghi 220 $\mu$ F trên hình a) và loại tụ hóa có 2 chân nổi ra cùng 1 đầu trụ tròn (tụ có ghi giá trị 10 $\mu$ F trên hình a). Đồng thời trên các tụ hóa, người ta thường ghi kèm giá trị điện áp cực đại mà tụ có thể chịu được. Nếu trường hợp điện áp lớn hơn so với giá trị điện áp trên tụ thì tụ sẽ bị phồng hoặc nổ tụ tùy thuộc vào giá trị điện áp cung cấp. Thông thường, khi chọn các loại tụ hóa này người ta thường chọn các loại tụ có giá trị điện áp lớn hơn các giá trị điện áp đi qua tụ để đảm bảo tụ hoạt động tốt và đảm bảo tuổi thọ của tụ hóa.

## Tụ Tantal

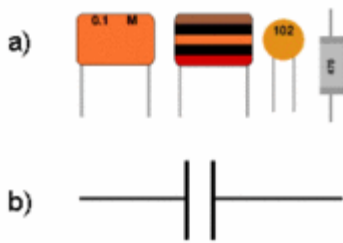


Tụ Tantal

Tụ Tantal cũng là loại tụ hóa nhưng có điện áp thấp hơn so với tụ hóa. Chúng khá đắt nhưng nhỏ và chúng được dùng khi yêu cầu về tụ dung lớn nhưng kích thước nhỏ.

Các loại tụ Tantal hiện nay thường ghi rõ trên nó giá trị tụ, điện áp cũng như cực của tụ. Các loại tụ Tantal ngày xưa sử dụng mã màu để phân biệt. Chúng thường có 3 cột màu (biểu diễn giá trị tụ, một cột biểu diễn giá trị điện áp) và một chấm màu đặc trưng cho số các số không sau dấu phẩy tính theo giá trị  $\mu$ F. Chúng cũng dùng mã màu chuẩn cho việc định nghĩa các giá trị nhưng đối với các điểm màu thì điểm màu xám có nghĩa là giá trị tụ nhân với 0,01; trắng nhân 0,1 và đen là nhân 1. Cột màu định nghĩa giá trị điện áp thường nằm ở gần chân của tụ và có các giá trị như sau:





Tụ thường và kí hiệu

Vàng=6,3V

Đen= 10V

Xanh lá cây= 16V

Xanh da trời= 20V

Xám= 25V

Trắng= 30V

Hồng= 35V

### Tụ không phân cực



Tụ thường

Các loại tụ nhỏ thường không phân cực. Các loại tụ này thường chịu được các điện áp cao mà thông thường là khoảng 50V hay 250V. Các loại tụ không phân cực này có rất nhiều loại và có rất nhiều các hệ thống chuẩn đọc giá trị khác nhau.

Rất nhiều các loại tụ có giá trị nhỏ được ghi thẳng ra ngoài mà không cần có hệ số nhân nào, nhưng cũng có các loại tụ có thêm các giá trị cho hệ số nhân. Ví dụ có các tụ ghi **0.1** có nghĩa giá trị của nó là  $0,1\mu\text{F}=100\text{nF}$  hay có các tụ ghi là **4n7** thì có nghĩa giá trị của tụ đó chính là  $4,7\text{nF}$

### Các loại tụ có dùng mã



Tụ thường

Mã số thường được dùng cho các loại tụ có giá trị nhỏ trong đó các giá trị được định nghĩa lần lượt như sau:

- Giá trị thứ 1 là số hàng chục
- Giá trị thứ 2 là số hàng đơn vị
- Giá trị thứ 3 là số số không nối tiếp theo giá trị của số đã tạo từ giá trị 1 và 2. Giá trị của tụ được đọc theo chuẩn là giá trị pico Fara (pF)
- Chữ số đi kèm sau cùng đó là chỉ giá trị sai số của tụ.

Ví dụ: tụ ghi giá trị 102 thì có nghĩa là 10 và thêm 2 số 0 đằng sau =  $1000\text{pF} = 1\text{nF}$  chứ không phải 102pF

Hoặc ví dụ tụ 272J thì có nghĩa là  $2700\text{pF} = 2,7\text{nF}$  và sai số là 5%

### Tụ có dùng mã màu



Tụ dùng mã màu

Sử dụng chủ yếu trên các tụ loại polyester trong rất nhiều năm. Hiện nay các loại tụ này đã không còn bán trên thị trường nữa nhưng chúng vẫn tồn tại trong khá nhiều các mạch điện tử cũ. Màu được định nghĩa cũng tương tự như đối với màu trên điện trở. 3 màu trên cùng lần lượt chỉ giá trị tụ tính theo pF, màu thứ 4 là chỉ dung sai và màu thứ 5 chỉ ra giá trị điện áp.

Ví dụ tụ có màu nâu/đen/cam có nghĩa là  $10000\text{pF} = 10\text{nF} = 0.01\mu\text{F}$ .

Chú ý rằng ko có khoảng trống nào giữa các màu nên thực tế khi có 2 màu cạnh nhau giống nhau thì nó tạo ra một mảng màu rộng. Ví dụ Dải đỏ rộng/vàng=  $220nF=0.22\mu F$

## Tụ Polyester

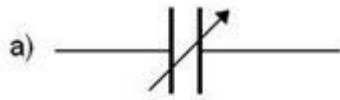
Ngày nay, loại tụ này cũng hiếm khi được sử dụng. Giá trị của các loại tụ này thường được in ngay trên tụ theo giá trị pF. Tụ này có một nhược điểm là dễ bị hỏng do nhiệt hàn nóng. Chính vì thế khi hàn các loại tụ này người ta thường có các kỹ thuật riêng để thực hiện hàn, tránh làm hỏng tụ.



Tụ polyester

## Tụ điện biến đổi

Tụ điện biến đổi thường được sử dụng trong các mạch điều chỉnh radio và chúng thường được gọi là tụ xoay. Chúng thường có các giá trị rất nhỏ, thông thường nằm trong khoảng từ 100pF đến 500pF.



b)



Tụ xoay

Rất nhiều các tụ xoay có vòng xoay ngắn nên chúng không phù hợp cho các dải biến đổi rộng như là điện trở hoặc các chuyển mạch xoay. Chính vì thế trong nhiều ứng dụng, đặc biệt là trong các mạch định thời hay các mạch điều chỉnh thời gian thì người ta thường thay các tụ xoay bằng các điện trở xoay và kết hợp với 1 giá trị tụ điện xác định.

## Tụ chặn

Tụ chặn là các tụ xoay có giá trị rất nhỏ. Chúng thường được gắn trực tiếp lên bản mạch điện tử và điều chỉnh sau khi mạch đã được chế tạo xong. Tương tự các biến trở hiện nay thì khi điều chỉnh các tụ chặn này người ta cũng dùng các tuốc nơ vít loại nhỏ để điều chỉnh. Tuy nhiên do giá trị các tụ này khá nhỏ nên khi điều chỉnh, người ta thường phải rất cẩn thận và kiên trì vì trong quá trình điều chỉnh có sự ảnh hưởng của tay và tuốc nơ vít tới giá trị tụ.



Tụ chặn

Các tụ chặn này thường có giá trị rất nhỏ, thông thường nhỏ hơn khoảng 100pF. Có điều đặc biệt là không thể giảm nhỏ được các giá trị tụ chặn về 0 nên chúng thường được chỉ định với các giá trị tụ điện tối thiểu, khoảng từ 2 tới 10 pF.

## Cuộn cảm

**INDUCTOR COLOR GUIDE**  
Result Is In  $\mu\text{H}$

4-BAND-CODE  $270\mu\text{H} \pm 5\%$

| COLOR  | 1st BAND | 2nd BAND | MULTIPLIER          | TOLERANCE           |
|--------|----------|----------|---------------------|---------------------|
| BLACK  | 0        | 0        | 1                   | $\pm 20\%$          |
| BROWN  | 1        | 1        | 10                  | Military $\pm 1\%$  |
| RED    | 2        | 2        | 100                 | Military $\pm 2\%$  |
| ORANGE | 3        | 3        | 1,000               | Military $\pm 3\%$  |
| YELLOW | 4        | 4        | 10,000              | Military $\pm 4\%$  |
| GREEN  | 5        | 5        |                     |                     |
| BLUE   | 6        | 6        |                     |                     |
| VIOLET | 7        | 7        |                     |                     |
| GREY   | 8        | 8        |                     |                     |
| WHITE  | 9        | 9        |                     |                     |
| NONE   |          |          |                     | Military $\pm 20\%$ |
| GOLD   |          |          | 0.1 / Mil. Dec. Pt. | Both $\pm 5\%$      |
| SILVER |          |          | 0.01                | Both $\pm 10\%$     |

Military Identifier  $6.8\mu\text{H} \pm 10\%$   
MILITARY CODE

**Electronix Express / RSR**  
<http://www.elexp.com>

**1-800-972-2225**  
**In NJ 732-381-8020**

Tương tự như đối với điện trở, trên thế giới có một số loại cuộn cảm có cấu trúc tương tự như điện trở. Quy định màu và cách đọc màu đều tương tự như đối với các điện trở.

Tuy nhiên, do các giá trị của các cuộn cảm thường khá linh động đối với yêu cầu thiết kế mạch cho nên các cuộn cảm thường được tính toán và quấn theo số vòng dây xác định.

Với mỗi loại dây, với mỗi loại lõi khác nhau thì giá trị cuộn cảm sẽ khác nhau. Trong phần giáo trình này không đề cập cụ thể tới cách tính toán và quấn các cuộn cảm khác nhau. Phần này sẽ được đề cập cụ thể trong phần sách sau này.

## Một số các phương pháp kiểm tra thông thường

Để kiểm tra các giá trị tụ điện, cuộn cảm hoặc điện trở thì thông thường mọi người sử dụng các đồng hồ đo đa năng. Hiện nay, có các loại đồng hồ đo đa năng có chức năng đo chính xác các giá trị cuộn cảm, tụ điện và điện trở, điện áp, dòng điện, thậm chí xác định transistor và điốt. Chính vì thế, trong phần này, tôi không đề cập tới các phương pháp kiểm tra cũ (khi dùng đồng hồ cơ/kim) như trước đây.

## Tóm tắt chương

Trong chương này, các linh kiện điện tử cơ bản đã được trình bày một cách tương đối cụ thể. Yêu cầu duy nhất đối với người đọc đó là sau khi đọc chương này có thể nắm bắt được và nhận biết được các linh kiện điện tử cơ bản trước khi tìm hiểu và đi sâu hơn vào lĩnh vực điện tử. Yêu cầu nắm vững của phần chương này đó là phân biệt được các linh kiện cơ bản như điện trở, tụ điện, các phương pháp đọc điện trở và cao hơn nữa đó chính là khả năng đọc được giá trị của điện trở, tụ điện,...mà không cần phải tra cứu. Để đạt được điều này, yêu cầu đối với người đọc là phải thực hành so sánh và đọc giá trị các linh kiện thường xuyên.

## Tài liệu tham khảo

1. Electronix express

2. Công cụ đọc điện trở nhanh trên web

[http://www.samengstrom.com/nx1/3660/4\\_band\\_resistor\\_color\\_code\\_page.en.html](http://www.samengstrom.com/nx1/3660/4_band_resistor_color_code_page.en.html)

# Giáo trình Điện tử cơ bản/Cơ bản về bán dẫn

Bài từ Tủ sách Khoa học VLOS.

Cập nhật 23:39, 13/9/2009, bởi [Nguyen Phan Kien](#)

Jump to: [navigation](#), [search](#)

## Mục lục

[[giấu](#)]

- [1 Các khái niệm cơ bản về bán dẫn](#)
  - [1.1 Bản chất dòng điện trong chất bán dẫn](#)
  - [1.2 Bán dẫn tạp chất và bản chất dòng điện](#)

- [1.3 Điốt bán dẫn- Phần tử một mặt ghép p-n](#)
  - [1.3.1 Phân cực thuận](#)
  - [1.3.2 Phân cực ngược](#)
  - [1.3.3 Đánh thủng](#)
- [1.4 Lý thuyết về điốt](#)
  - [1.4.1 Phân loại điốt](#)
  - [1.4.2 Cách kiểm tra Điốt](#)
  - [1.4.3 Một số loại Điốt thông dụng](#)
- [2 Bán dẫn nhiều lớp](#)
  - [2.1 Transistor](#)
    - [2.1.1 Hai loại transistor cơ bản](#)
    - [2.1.2 I. Transistor lưỡng cực \(BJT\)](#)
      - [2.1.2.1 Đọc xong phần này bạn nên có thể:](#)
      - [2.1.2.2 I.1 Transistor chưa phân cực](#)
      - [2.1.2.3 I.1 Transistor đã phân cực](#)
    - [2.1.3 II. Transistor hiệu ứng trường \( FET \)](#)
    - [2.1.4 Cách kiểm tra transistor](#)
    - [2.1.5 Một số ứng dụng của Transistor](#)
  - [2.2 Thyristor](#)
- [3 Tóm tắt chương](#)
- [4 Câu hỏi tự đánh giá](#)
- [5 Tài liệu tham khảo](#)
  - [5.1 Bản quyền](#)

## Các khái niệm cơ bản về bán dẫn

Trong quá trình phân loại vật chất đối với quá trình dẫn điện, người ta chia các vật liệu ra thành ba loại. Đó chính là các vật liệu dẫn điện (như kim loại) và các vật liệu không dẫn điện/cách điện và loại thứ ba là các vật liệu bán dẫn. Các vật liệu dẫn điện là các vật liệu cho phép các dòng điện truyền qua còn các vật liệu cách điện hay không dẫn điện là các vật liệu không cho dòng điện truyền qua.

Chất bán dẫn chủ yếu được cấu tạo từ các nguyên tử có 4 electron lớp ngoài trong cấu trúc nguyên tử của chúng. Như vậy, về bản chất, các chất bán dẫn có 4 electron lớp ngoài cùng mà đặc trưng là 2 chất bán dẫn Ge và Si.

Ở dạng rắn, các nguyên tử cấu tạo nên chất bán dẫn được sắp xếp theo một cấu trúc có thứ tự mà chúng ta gọi là dạng tinh thể. Mỗi nguyên tử chia sẻ các electron của chúng với các nguyên tử ngay cạnh để tạo nên một cấu trúc bền vững có 8 electron lớp ngoài cho nguyên tử nằm tại vị trí trung tâm. Như vậy, mỗi nguyên tử xung quanh nguyên tử trung tâm sẽ chia sẻ 1 electron với nguyên tử trung tâm để tạo thành một cấu trúc bền vững có 8 electron lớp ngoài (đối với nguyên tử trung tâm). Như vậy có thể nói, liên kết giữa nguyên tử trung tâm với 4 nguyên tử xung quanh sẽ dựa trên chủ yếu 4 liên kết hóa trị. Dưới tác dụng của nhiệt, các nguyên tử sẽ tạo ra các dao động xung quanh vị trí cân bằng và tại một giá trị xác định nào đó, nhiệt độ có thể phá vỡ các liên kết hóa trị và tạo ra các electron tự do. Tại vị trí của các electron tự do vừa bứt ra sẽ thiếu 1 electron và trở thành các lỗ trống. Lỗ trống này có xu hướng nhận thêm 1 electron nhằm tạo lại sự cân bằng.

## **Bản chất dòng điện trong chất bán dẫn**

Như đã nói ở trên, trong cấu trúc vật liệu của bản thân chất bán dẫn, dưới tác dụng của nhiệt độ môi trường cũng luôn tồn tại hai dạng điện tích. Một là điện tích âm do electron và hai là điện tích dương do lỗ trống tạo ra. Dưới tác dụng của điện trường, các electron có xu hướng di chuyển về phía phía có năng lượng điện tích cao hơn. Do đó, lúc này, trong bản chất chất bán dẫn sẽ có 2 thành phần cân bằng. Một là electron tự do bứt ra khỏi liên kết hóa trị và hai là lỗ trống sinh ra do electron bứt ra. Electron bứt ra khỏi cấu trúc tinh thể sẽ di chuyển về phía điện trường có điện thế lớn. Đồng thời, lỗ trống cũng có xu hướng hút các electron ở xung quanh để điền đầy và đi về phía điện trường có điện thế nhỏ hơn. Như vậy, bản chất dòng điện trong chất bán dẫn được sinh ra bởi 2 dòng chuyển dời: dòng chuyển dời của các electron tự do và dòng chuyển dời của các lỗ trống. Các electron và các lỗ trống thường được gọi chung với một cái tên là hạt mang điện bởi chúng mang năng lượng điện tích dịch chuyển từ điểm này đến điểm khác.

## **Bán dẫn tạp chất và bản chất dòng điện**

Như đã biết, bán dẫn tạp chất được tạo ra bởi việc cung cấp các chất tạp chất thuộc nhóm 3 và nhóm 5 bảng tuần hoàn Mendeleev đưa vào trong cấu trúc tinh thể chất bán dẫn thuần.

Để tăng số lượng các electron tự do, thông thường, người ta thêm các tạp chất thuộc nhóm 5 trong bảng tuần hoàn Mendeleev vào. Khi đó, các thành phần tạp chất này sẽ tham gia xây dựng cấu trúc tinh thể của vật chất. Tương tự như giải thích về phần cấu tạo nguyên tử, khi 1 nguyên tử tạp chất đứng cạnh các nguyên tử bán dẫn thuần thì chúng cũng sẽ chia sẻ 1 electron với nguyên tử bán dẫn thuần, do đó sẽ còn 4 electron tại lớp ngoài cùng phân tử. Trong số 4 electron này chỉ có 3 electron tiếp tục tham gia tạo mạng tinh thể và 1 electron sẽ có xu hướng tách ra và trở thành các electron tự do. Do đó, khi so sánh với cấu trúc mạng tinh thể bán dẫn thuần, cấu trúc bán dẫn tạp chất loại này có nhiều các electron tự do hơn. Loại bán dẫn tạp chất này được gọi là bán dẫn loại n (n bản chất tiếng Anh là negative chỉ đặc trưng bản chất của việc thừa electron). Như vậy trong bán dẫn loại n sẽ tồn tại 2 loại hạt mang điện. Hạt đa số chính là các electron tự do tích điện âm và hạt thiểu số là các lỗ trống (mang điện tích dương).

Tương tự nhưng với hướng ngược lại, người ta thêm tạp chất thuộc nhóm 3 trong bảng tuần hoàn Mendeleev vào trong cấu trúc tinh thể chất bán dẫn thuần. Các thành phần tạp chất này cũng tham gia xây dựng cấu trúc tinh thể của chất bán dẫn, nhưng do chỉ có 3 electron lớp ngoài nên trong cấu trúc nguyên tử sẽ có một vị trí không có electron tham gia xây dựng các liên kết. Các vị trí thiếu này vô hình chung đã tạo nên các lỗ trống. Do đó, trong cấu trúc tinh thể của loại bán dẫn tạp chất này sẽ có nhiều vị trí khuyết electron hơn hay còn gọi là các lỗ trống hơn. Loại bán dẫn này được gọi là bán dẫn loại p (p đặc trưng cho từ positive). Hạt đa số chính là các lỗ trống và hạt thiểu số sẽ là các electron. Tóm lại, bán dẫn loại n có nhiều electron tự do hơn và bán dẫn loại p có nhiều lỗ trống hơn. Do đó, n có khả năng cho electron và p có khả năng nhận electron.

## **Điốt bán dẫn- Phần tử một mặt ghép p-n**

Trong công nghệ chế tạo phân tử 1 mặt ghép p-n, người ta thực hiện pha trộn hai loại bán dẫn tạp chất lên trên một phiến đế tinh thể bán dẫn thuần với một bên là bán dẫn loại p và 1 bên là bán dẫn loại n. Do lực hút lẫn nhau, các electron tự do bên phía bán dẫn loại n có xu hướng khuếch tán theo mọi hướng. Một vài electron tự do khuếch tán vượt qua bề mặt ghép p-n. Khi một electron tự do của bán dẫn loại n đi vào vùng của bán dẫn loại p, nó trở thành hạt thiếu số. Do có một lượng lớn các lỗ trống nên các electron này sẽ nhanh chóng liên kết với lỗ trống để tinh thể trở về trạng thái cân bằng và đồng thời làm lỗ trống biến mất.

Mỗi lần một electron khuếch tán vượt qua vùng tiếp giáp thì nó tạo ra một cặp các ion. Khi một electron rời khỏi miền n thì nó để lại cho cấu trúc nguyên tử tạp chất một (thuộc nhóm 5 bảng tuần hoàn Mendeleev) sang trạng thái mới, trạng thái thiếu một electron. Nguyên tử tạp chất lúc này lại trở thành 1 ion dương. Nhưng đồng thời, khi đi sang miền p và kết hợp với một lỗ trống thì nó vô hình đã làm nguyên tử tạp chất (thuộc nhóm 3 bảng tuần hoàn Mendeleev) trở thành ion âm.

Quá trình này diễn ra liên tục và làm cho vùng tiếp xúc của chất bán dẫn lần lượt có ngày càng nhiều cặp ion dương và âm tương ứng ở miền n và miền p. Các cặp ion này sau khi hình thành sẽ tạo nên một vùng tại miền tiếp xúc bán dẫn mà ta gọi là miền tiếp xúc, có điện trường ngược lại với chiều khuếch tán tự nhiên của các electron tự do và các lỗ trống. Quá trình khuếch tán sẽ dừng khi số lượng các cặp ion sinh ra đủ lớn để cản trở sự khuếch tán tự do của các electron từ n sang p.

Như vậy, ký hiệu âm và dương tại miền tiếp xúc p-n chính là ký hiệu của các cặp ion sinh ra trong quá trình khuếch tán.

## **Phân cực thuận**

## **Phân cực ngược**

## **Đánh thủng**

## **Lý thuyết về điốt**

## **Phân loại điốt**

Xem thêm mục [Điốt](#)

## **Cách kiểm tra Điốt**

Để kiểm tra một điốt còn khả năng hoạt động hay không, chúng ta có thể sử dụng các đồng hồ đo, đặt chế độ đo điện trở để đo khả năng dẫn dòng điện hay hạn chế dòng điện của điốt. Thông qua đó, chúng ta sẽ biết được điốt còn khả năng sử dụng hay không.

### **Chú ý:**

- Đối với một số loại Ohm kế cũ, dòng hoặc áp của Ohm kế có thể phá hủy 1 số loại diode sử dụng trong các mạch tần số cao.



- Giá trị của thang đo Ohm để xác định khả năng hoạt động của diode thường để khoảng vài trăm KiloOhm.

- Với các đồng hồ Digital Multimeter có chức năng kiểm tra diode, ta có thể sử dụng chức năng này để kiểm tra.

## Một số loại Điốt thông dụng

# Bán dẫn nhiều lớp

## Transistor

Tín hiệu radio hay vô tuyến thu được từ ăng-ten yếu đến mức nó không đủ để chạy một cái loa hay một đèn điện tử ở tivi. Đây là lý do chúng ta phải khuếch đại tín hiệu yếu để nó có đủ năng lượng để trở nên hữu dụng. Trước năm 1951, ống chân không là thiết bị chính dùng trong việc khuếch đại các tín hiệu yếu. Mặc dù khuếch đại khá tốt, nhưng ống chân không lại có một số nhược điểm. Thứ nhất, nó có một sợi nung bên trong, nó đòi hỏi năng lượng 1 W hoặc hơn. Thứ hai, nó chỉ sống được vài nghìn giờ, trước khi sợi nung hỏng. Thứ ba, nó tốn nhiều không gian. Thứ tư, nó tỏa nhiệt, làm tăng nhiệt độ của các thiết bị điện tử.

Năm 1951, Shockley đã phát minh ra tranzitor có mặt tiếp giáp đầu tiên, một dụng cụ bán dẫn có khả năng khuếch đại các tín hiệu radio và vô tuyến. Các ưu điểm của tranzito khắc phục được các khuyết điểm của ống chân không. Thứ nhất, nó không có sợi nung hay vật làm nóng nào, do đó nó cần ít năng lượng hơn. Thứ hai, do nó là dụng cụ bán dẫn nên có thể sống vô hạn định. Thứ ba, do nó rất nhỏ nên cần ít không gian. Thứ tư, do nó sinh ra ít nhiệt hơn, vì vậy nhiệt độ của các thiết bị điện tử sẽ thấp hơn.

Tranzito đã dẫn tới nhiều phát minh khác, bao gồm: mạch tích hợp (IC), một thiết bị nhỏ chứa hàng ngàn tranzito. Nhờ IC mà máy vi tính và các thiết bị điện tử kỳ diệu khác có thể thực hiện được.

## Hai loại transistor cơ bản

Transistor được chia làm 2 loại là transistor lưỡng cực (BJT -Bipolar Junction Trasistor) và transistor hiệu ứng trường (FET- Field Effect Transistor).

### I. Transistor lưỡng cực (BJT)

#### *Đọc xong phần này bạn nên có thể:*

- Trình bày những hiểu biết về mối quan hệ giữa các dòng điện bazơ, emitơ và collectơ của một transistor lưỡng cực.
- Vẽ sơ đồ của mạch CE và đánh dấu các cực, điện áp và điện trở.
- Vẽ một đường cong bazơ giả thuyết và tập hợp các đường cong emitơ, ghi tên các trục.
- Thảo luận về các đặc tính của transistor lý tưởng và transistor xấp xỉ lần hai.

- Kể ra vài thông số đặc trưng của transistor hữu dụng đối với các nhà kỹ thuật.

### ***1.1 Transistor chưa phân cực***

Một transistor có ba miền pha tạp như trong hình 6.1. Miền dưới cùng được gọi là *emitor*, miền giữa được gọi là *bazo*, miền trên cùng là *collector*. Loại transistor cụ thể ở đây là một thiết bị npn. Transistor còn có thể được sản xuất như các thiết bị pnp.

#### **Diode emitor và collector**

Transistor ở hình 6.1 có 2 tiếp giáp: một giữa emitor và bazo và cái kia là giữa bazo và collector. Do đó transistor tương tự hai diode. emitor và bazo tạo một diode, bazo và collector tạo thành một diode khác. Từ giờ, chúng ta sẽ gọi mấy diode này là *diode emitor* (cái dưới) và *diode collector* (cái trên).

#### **Trước và sau sự khuếch tán**

Hình 6.1 chỉ ra các miền của transistor trước khi sự khuếch tán xảy ra. Như đã nói đến ở phần trước, electron tự do ở miền n khuếch tán qua vùng tiếp giáp và kết hợp với lỗ trống ở miền p. Hình dung các electron ở mỗi miền n ngang qua phân tiếp giáp và kết hợp với các lỗ trống. Kết quả là hai vùng nghèo như hình 6.2, Mỗi vùng nghèo này hàng rào thế xấp xỉ 0.7 V ở 25°C. Như đã nói, chúng ta nhấn mạnh đến các thiết bị silic vì chúng được sử dụng rộng rãi hơn các thiết bị bằng germani.

### ***1.1 Transistor đã phân cực***

## **II. Transistor hiệu ứng trường ( FET )**

### **1. Giới thiệu chung về FET**

a.FET hoạt động dựa trên hiệu ứng trường có nghĩa là điện trở của bán dẫn được điều khiển bởi điện trường bên ngoài, dòng điện trong FET chỉ do 1 loại hạt dẫn là electron hoặc lỗ trống tạo nên.

b.Phân loại: FET có 2 loại chính:

- JFET: Transistor trường điều khiển bằng tiếp xúc N-P.
- IGFET: Transistor có cực cửa cách điện, thông thường lớp cách điện này được làm bằng 1 lớp oxit nên có tên gọi khác là MOSFET ( Metal Oxide Semiconductor FET ).

Mỗi loại FET đều có 2 loại kênh N và kênh P. FET có 3 cực là cực Nguồn ( source - S ), cực Máng ( drain - D ), cực Cổng ( gate - G ).

### **2. JFET**

a. Cấu tạo:

JFET được cấu tạo bởi 1 miếng bán dẫn mỏng ( loại N hoặc loại P ) 2 đầu tương ứng là D và S, miếng bán dẫn này được gọi là kênh dẫn điện. 2 miếng bán dẫn ở 2 bên kênh dẫn được nối với cực G, lưu ý, cực G được tách ra khỏi kênh nhờ tiếp xúc N-P.

Đa phần các JFET có cấu tạo đối xứng nên có thể đổi chỗ cực D và S mà tính chất không thay đổi.

## b. Nguyên lý hoạt động

Muốn cho JFET hoạt động ta phải cung cấp  $U_{GS}$  sao cho cả 2 tiếp xúc N-P đều phân cực ngược, nguồn  $U_{DS}$  sao cho dòng hạt dẫn dịch chuyển từ cực S qua kênh tới cực D tạo thành dòng  $I_D$ .

### - Khả năng điều khiển điện áp $I_D$ của $U_{GS}$ :

Giả sử với JFET kênh N,  $U_{DS} = \text{const}$ . Khi đặt  $U_{GS} = 0$ , tiếp giáp PN bắt đầu phân cực ngược mạnh dần, kênh hẹp dần từ S về D, nhưng lúc này độ rộng kênh là lớn nhất do vậy dòng qua kênh là lớn nhất kí hiệu là  $I_{D0}$ .

Khi  $U_{GS} < 0$ , PN phân cực ngược mạnh hơn do vậy bề rộng của kênh dẫn hẹp dần, tại thời điểm  $U_{GS} = U_{ngắt}$  thì 2 tiếp giáp PN phủ lên nhau, che lấp hết kênh, dòng  $I_D = 0$ . Dòng  $I_D$  được tính theo công thức:  $I_D = I_{D0} (1 - U_{GS}/U_{ngắt})^2$

Chú ý : giá trị của  $U_{ngắt}$  và  $I_{D0}$  phụ thuộc vào  $U_{DS}$ .

## Cách kiểm tra transistor

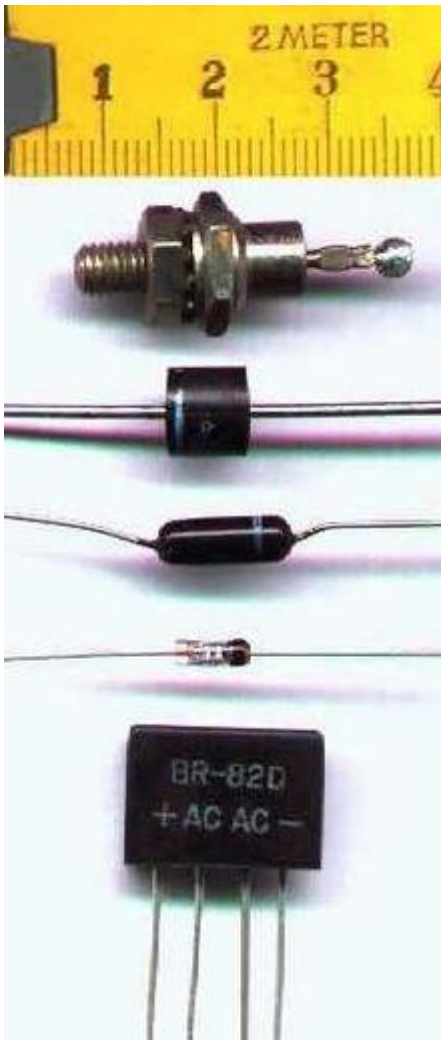
Đối với transistor nói chung, do cấu tạo của transistor gồm 2 tiếp xúc P-N nên có thể coi là 2 diode nối tiếp nhau từ đó có thể kiểm tra sự hoạt động của transistor tương tự như kiểm tra diode.

## Một số ứng dụng của Transistor

# Điốt bán dẫn

Bách khoa toàn thư mở Wikipedia

Bước tới: [menu](#), [tìm kiếm](#)



Một số loại diốt bán dẫn

Hình bên trái là một cầu chỉnh lưu bao gồm bốn diốt bán dẫn ghép lại.

Hình bên phải là đặc trưng của loại diốt sử dụng chất bán dẫn Ge

*Xem các bài liên quan Diốt tại [Diốt \(định hướng\)](#)*

**Diốt bán dẫn** là các [linh kiện điện tử thu đông](#) và [phi tuyến](#), cho phép dòng điện đi qua nó theo một chiều mà không theo chiều ngược lại, sử dụng các tính chất của các chất [bán dẫn](#).

Có nhiều loại diốt bán dẫn, như [điốt chỉnh lưu](#) thông thường, [điốt Zener](#), [LED](#). Chúng đều có nguyên lý cấu tạo chung là một khối [bán dẫn loại P](#) ghép với một khối [bán dẫn loại N](#).

## Mục lục

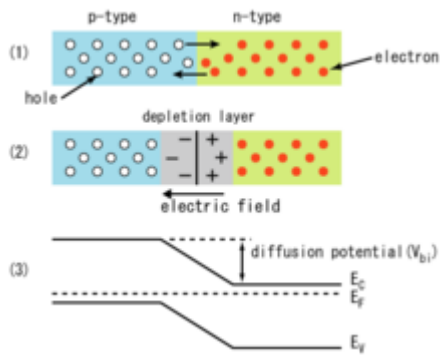
[\[ẩn\]](#)

- [1 Hoạt động](#)
- [2 Tính chất](#)
  - [2.1 Đặc tuyến Volt-Ampere](#)

## [sửa] Hoạt động

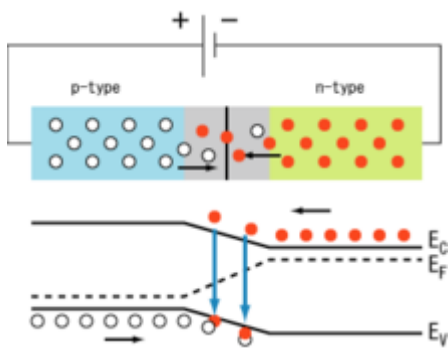
Khối bán dẫn loại P chứa nhiều [lỗ trống](#) tự do mang [điện tích dương](#) nên khi ghép với khối bán dẫn N (chứa các [điện tử](#) tự do) thì các lỗ trống này có xu hướng chuyển động khuếch tán sang khối N. Cùng lúc khối P lại nhận thêm các điện tử (điện tích âm) từ khối N chuyển sang. Kết quả là khối P tích điện âm (thiếu hụt lỗ trống và dư thừa điện tử) trong khi khối N tích điện dương (thiếu hụt điện tử và dư thừa lỗ trống).

Ở biên giới hai bên mặt tiếp giáp, một số điện tử bị lỗ trống thu hút và khi chúng tiến lại gần nhau, chúng có xu hướng kết hợp với nhau tạo thành các nguyên tử trung hòa. Quá trình này có thể giải phóng năng lượng dưới dạng [ánh sáng](#) (hay các [bức xạ điện từ](#) có bước sóng gần đó).



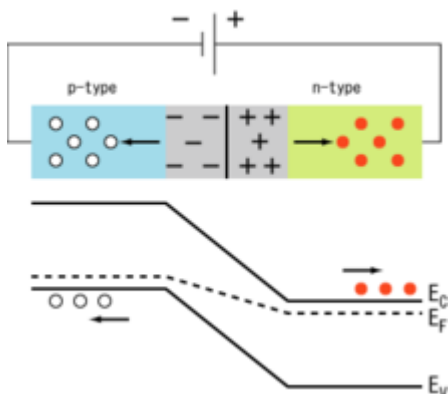
Điện áp tiếp xúc hình thành.

Sự tích điện âm bên khối P và dương bên khối N hình thành một [điện áp](#) gọi là điện áp tiếp xúc (UTX). [Điện trường](#) sinh ra bởi điện áp có hướng từ khối n đến khối p nên cản trở chuyển động khuếch tán và như vậy sau một thời gian kể từ lúc ghép 2 khối bán dẫn với nhau thì quá trình chuyển động khuếch tán chấm dứt và tồn tại điện áp tiếp xúc. Lúc này ta nói tiếp xúc P-N ở trạng thái cân bằng. Điện áp tiếp xúc ở trạng thái cân bằng khoảng 0.6V đối với điốt làm bằng bán dẫn [Si](#) và khoảng 0.3V đối với điốt làm bằng bán dẫn [Ge](#).



Điện áp ngoài ngược chiều điện áp tiếp xúc tạo ra dòng điện.

Hai bên mặt tiếp giáp là vùng các điện tử và lỗ trống dễ gặp nhau nhất nên quá trình tái hợp thường xảy ra ở vùng này hình thành các nguyên tử trung hòa. Vì vậy vùng biên giới ở hai bên mặt tiếp giáp rất hiếm các hạt dẫn điện tự do nên được gọi là vùng nghèo. Vùng này không dẫn điện tốt, trừ phi điện áp tiếp xúc được cân bằng bởi điện áp bên ngoài. Đây là cốt lõi hoạt động của điốt.



Điện áp ngoài cùng chiều điện áp tiếp xúc ngăn dòng điện.

Nếu đặt điện áp bên ngoài ngược với điện áp tiếp xúc, sự khuếch tán của các điện tử và lỗ trống không bị ngăn trở bởi điện áp tiếp xúc nữa và vùng tiếp giáp dẫn điện tốt. Nếu đặt điện áp bên ngoài cùng chiều với điện áp tiếp xúc, sự khuếch tán của các điện tử và lỗ trống càng bị ngăn lại và vùng nghèo càng trở nên nghèo hạt dẫn điện tự do. Nói cách khác điốt chỉ cho phép dòng điện qua nó khi đặt điện áp theo một hướng nhất định.

## [sửa] Tính chất

Điốt chỉ dẫn điện theo một chiều từ a-nốt sang ca-tốt. Theo nguyên lý dòng điện chảy từ nơi có điện thế cao đến nơi có điện thế thấp, muốn có dòng điện qua điốt theo chiều từ nơi có điện thế cao đến nơi có điện thế thấp, cần phải đặt ở a-nốt một điện thế cao hơn ở ca-tốt. Khi đó ta có  $U_{AK} > 0$  và ngược chiều với điện áp tiếp xúc ( $U_{TX}$ ). Như vậy muốn có dòng điện qua điốt thì điện trường do  $U_{AK}$  sinh ra phải mạnh hơn điện trường tiếp xúc, tức là:  $U_{AK} > U_{TX}$ . Khi đó một phần của điện áp  $U_{AK}$  dùng để cân bằng với điện áp tiếp xúc (khoảng 0.6V), phần còn lại dùng để tạo dòng điện thuận qua điốt.

Khi  $U_{AK} > 0$ , ta nói điốt phân cực thuận và dòng điện qua điốt lúc đó gọi là dòng điện thuận (thường được ký hiệu là IF tức IFORWARD hoặc ID tức IDIODE). Dòng điện thuận có chiều từ a-nốt sang ca-tốt.

Khi  $U_{AK}$  đã đủ cân bằng với điện áp tiếp xúc thì điốt trở nên dẫn điện rất tốt, tức là điện trở của điốt lúc đó rất thấp (khoảng vài chục [Ohm](#)). Do vậy phần điện áp để tạo ra dòng điện thuận thường nhỏ hơn nhiều so với phần điện áp dùng để cân bằng với  $U_{TX}$ . Thông thường phần điện áp dùng để cân bằng với  $U_{TX}$  cần khoảng 0.6V và phần điện áp tạo dòng thuận khoảng 0.1V đến 0.5V tùy theo dòng thuận vài chục mA hay lớn đến vài [Ampere](#). Như vậy giá trị của  $U_{AK}$  đủ để có dòng qua điốt khoảng 0.6V đến 1.1V. Ngưỡng 0.6V là ngưỡng điốt bắt đầu dẫn và khi  $U_{AK} = 0.7V$  thì dòng qua Diode khoảng vài chục mA.

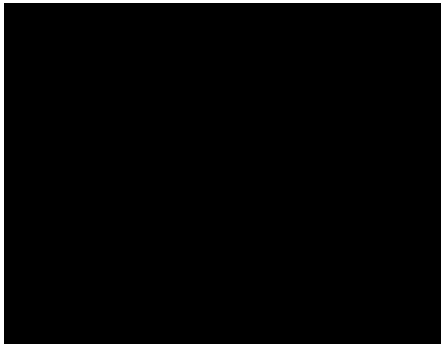
Nếu Diode còn tốt thì nó không dẫn điện theo chiều ngược ca-tốt sang a-nốt. Thực tế là vẫn tồn tại dòng ngược nếu điốt bị phân cực ngược với hiệu điện thế lớn. Tuy nhiên dòng điện ngược rất nhỏ (cỡ  $\mu\text{A}$ ) và thường không cần quan tâm trong các ứng dụng công nghiệp. Mọi điốt chỉnh lưu đều không dẫn điện theo chiều ngược nhưng nếu điện áp ngược quá lớn (VBR là ngưỡng chịu đựng của Diode) thì điốt bị đánh thủng, dòng điện qua điốt tăng nhanh và đốt cháy điốt. Vì vậy khi sử dụng cần tuân thủ hai điều kiện sau đây:


- Dòng điện thuận qua điốt không được lớn hơn giá trị tối đa cho phép (do nhà sản xuất cung cấp, có thể tra cứu trong các tài liệu của hãng sản xuất để xác định).
- Điện áp phân cực ngược (tức UKA) không được lớn hơn VBR (ngưỡng đánh thủng của điốt, cũng do nhà sản xuất cung cấp).

Ví dụ điốt 1N4007 có thông số kỹ thuật do hãng sản xuất cung cấp như sau:  
VBR=1000V, IFMAX = 1A, VF $\rightarrow$  = 1.1V khi IF = IFMAX. Những thông số trên cho biết:

- Dòng điện thuận qua điốt không được lớn hơn 1A.
- Điện áp ngược cực đại đặt lên điốt không được lớn hơn 1000V.
- Điện áp thuận (tức UAK) có thể tăng đến 1.1V nếu dòng điện thuận bằng 1A. Cũng cần lưu ý rằng đối với các điốt chỉnh lưu nói chung thì khi UAK = 0.6V thì điốt đã bắt đầu dẫn điện và khi UAK = 0.7V thì dòng qua điốt đã đạt đến vài chục mA.

### [sửa] Đặc tuyến Volt-Ampere



 Đặc tuyến Volt-Ampere của một điốt bán dẫn lý tưởng.

Đặc tuyến Volt-Ampere của Diode là đồ thị mô tả quan hệ giữa dòng điện qua điốt theo điện áp UAK đặt vào nó. Có thể chia đặc tuyến này thành hai giai đoạn:

- Giai đoạn ứng với UAK = 0.7V > 0 mô tả quan hệ dòng áp khi điốt phân cực thuận.
- Giai đoạn ứng với UAK = 0.7V < 0 mô tả quan hệ dòng áp khi điốt phân cực nghịch.

*(UAK lấy giá trị 0,7V chỉ đúng với các điốt Si, với điốt Ge thông số này khác)*

Khi điốt được phân cực thuận và [dẫn điện](#) thì dòng điện chủ yếu phụ thuộc vào [điện trở](#) của mạch ngoài (được mắc [nối tiếp](#) với điốt). Dòng điện phụ thuộc rất ít vào điện trở

thuận của điốt vì điện trở thuận rất nhỏ, thường không đáng kể so với điện trở của mạch điện.

## [sửa] Ứng dụng

Vì điốt có đặc tính chỉ dẫn điện theo một chiều từ a-nôt đến ca-tốt khi phân cực thuận nên điốt được dùng để chỉnh lưu dòng [điện xoay chiều](#) thành dòng [điện một chiều](#).

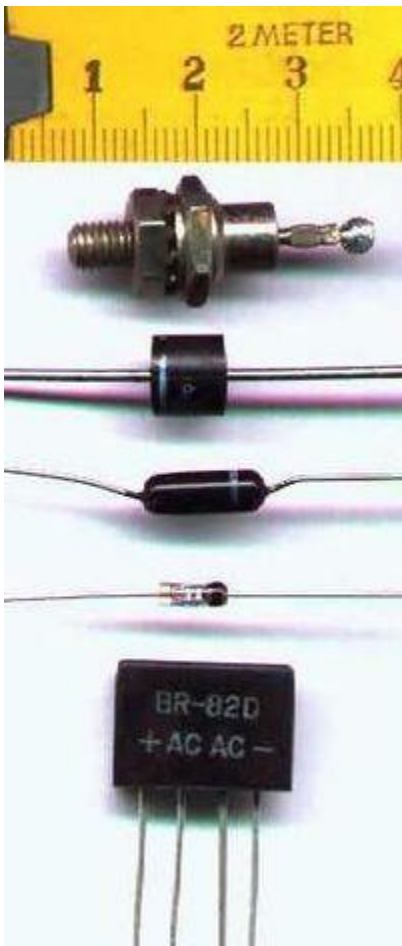
Ngoài ra điốt có nội trở thay đổi rất lớn, nếu phân cực thuận RD 0 (nối tắt), phân cực nghịch RD (hở mạch), nên điốt được dùng làm các công tắc điện tử, đóng ngắt bằng điều khiển mức [điện áp](#). Điốt chỉnh lưu dòng điện, giúp chuyển dòng điện xoay chiều thành dòng điện một chiều, điều đó có ý nghĩa rất lớn trong kỹ thuật điện tử. Vì vậy điốt được ứng dụng rộng rãi trong kỹ thuật điện và điện tử.

## Một số loại Điốt

Điốt được chia ra nhiều thể loại tùy theo vùng hoạt động của Điốt  
Phân loại theo sự phân cực:

- **Điốt phân cực thuận** Chỉ cần một điện áp dương đủ để cho Điốt dẫn điện . Điốt sẽ cho dòng điện đi qua theo **một chiều** từ Cực DƯƠNG đến Cực ÂM và sẽ cản dòng điện đi theo chiều ngược lại. Thí dụ : Điốt Bán dẫn, LED...
- **Điốt phân cực nghịch** Chỉ cần một điện áp âm đủ để cho Điốt dẫn điện (điện áp này gọi là điện áp đánh thủng của diode). Điốt sẽ cho dòng điện đi qua theo chiều phân cực nghịch của diode. Thông thường, dẫn điện tốt hơn trong chiều nghịch. Thí dụ : Điốt Zener, Điốt biến dung





Một số loại diốt thông dụng (*Riêng hình dưới cùng là một cầu chỉnh điện được tích hợp từ bốn diốt để dẫn điện xoay chiều thành một chiều*)



Điốt phát quang (LED)

Các Điốt thường thấy:

- **Điốt bán dẫn:** cấu tạo bởi chất bán dẫn Silic hoặc Gecmani có pha thêm một số chất để tăng thêm electron tự do. Loại này dùng chủ yếu để **chỉnh lưu** dòng điện hoặc trong mạch tách sóng.
- **Điốt Schottky:** Ở **tần số** thấp, diốt thông thường có thể dễ dàng khóa lại (ngưng dẫn) khi chiều phân cực thay đổi từ thuận sang nghịch, nhưng khi tần số tăng đến một ngưỡng nào đó, sự ngưng dẫn không thể đủ nhanh để ngăn chặn dòng điện suốt một phần của bán kỳ ngược. Điốt Schottky khắc phục được hiện tượng này.
- **Điốt Zener,** còn gọi là "điốt đánh thủng" hay "điốt ổn áp": là loại diốt được chế tạo tối ưu để hoạt động tốt trong miền đánh thủng. Khi sử dụng diốt này mắc

ngược chiều lại, nếu điện áp tại mạch lớn hơn điện áp định mức của điốt thì điốt sẽ cho dòng điện đi qua (và ngắt mạch xuống đất bảo vệ mạch điện cần ổn áp) và đến khi điện áp mạch mắc bằng điện áp định mức của điốt - Đây là cốt lõi của mạch ổn áp.

- **Điốt phát quang** hay còn gọi là LED (*Light Emitting Diode*) là các điốt có khả năng phát ra ánh sáng hay tia hồng ngoại, tử ngoại. Cũng giống như điốt bán dẫn, LED được cấu tạo từ một khối bán dẫn loại p ghép với một khối bán dẫn loại n.
- **Điốt quang** (*photodiode*): là loại nhạy với ánh sáng, có thể biến đổi ánh sáng vào thành đại lượng điện, thường sử dụng ở các máy ảnh (đo cường độ sáng), sử dụng trong các mạch điều khiển (kết hợp một điốt phát quang và một điốt quang thành một cặp), các modul đầu ra của các PLC...
- **Điốt biến dung** (varicap): Có tính chất đặc biệt, đó là khi phân cực nghịch, điốt giống như một **tu điện**, loại này được dùng nhiều cho máy thu hình, máy thu sóng FM và nhiều thiết bị truyền thông khác.
- **Điốt ổn định dòng điện**: là loại điốt hoạt động ngược với Điốt Zener. Trong mạch điện điốt này có tác dụng duy trì dòng điện không đổi.
- **Điốt step-recovery**: Ở bán kỳ dương, điốt này dẫn điện như loại điốt Silic thông thường, nhưng sang bán kỳ âm, dòng điện ngược có thể tồn tại một lúc do có lưu trữ điện tích, sau đó dòng điện ngược đột ngột giảm xuống còn 0.
- **Điốt ngược**: Là loại điốt có khả năng dẫn điện theo hai chiều, nhưng chiều nghịch tốt hơn chiều thuận.
- **Điốt xuyên hầm**: Nếu tăng nồng độ tạp chất của điốt ngược, có thể làm cho hiện tượng đánh thủng xảy ra ở **0V**, hơn nữa, nồng độ tạp chất sẽ làm biến dạng đường cong thuận chiều, điốt đó gọi là *điốt xuyên hầm*.

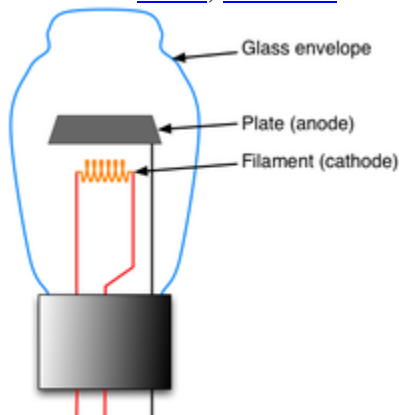
## [sửa] Tham khảo

- *Giáo trình mạch điện tử kỹ thuật tương tự*, phần viết về *Chất bán dẫn Diode và Transistor*. Nhà xuất bản thống kê năm 2002

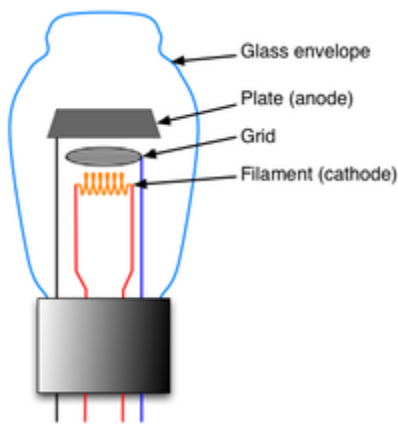
# Đèn điện tử chân không


Bách khoa toàn thư mở Wikipedia

Bước tới: [menu](#), [tìm kiếm](#)



Đèn điện tử chân không hai cực



 Đèn điện tử chân không ba cực

Trước đây, **đèn điện tử chân không** (*vacuum tube*, còn được gọi tắt là *tube* hay *valve*) còn thường được gọi là **đèn điện tử** hoặc **bóng điện tử** là một [linh kiện điện tử](#). Ngày nay, nhờ ứng dụng tính chất của [chất bán dẫn](#), phần lớn các đèn này được thay thế bằng các linh kiện điện tử khác nhỏ và rẻ hơn nhiều. Đầu [thế kỉ 21](#), có sự quan tâm trở lại của đèn điện tử chân không, vào thời điểm này có sự hình thành của [vi ống phát ra trường](#).

Bản chất của đèn điện tử có kích thước lớn, khi hoạt động toả ra nhiều nhiệt. Hiện nay hầu hết các thiết bị điện tử đã không còn dùng đèn này nữa mà dùng các [linh kiện bán dẫn](#) để thay thế (*transistor*, IC...). Tuy nhiên trong lĩnh vực chế tạo ampli cho giới sành nhạc, người ta vẫn rất chuộng ampli đèn, lý do là vì ampli đèn có khả năng tạo ra âm thanh trung thực bởi tính chất của nó (*trình bày ở phần sau*).

## Mục lục

[\[ân\]](#)

- [1 Nguyên lý hoạt động](#)
- [2 Ứng dụng](#)
- [3 Phân loại](#)
- [4 Xem thêm](#)

## [\[sửa\]](#) Nguyên lý hoạt động

Đèn điện tử là một loại thiết bị dựa vào sự khống chế luồng điện tử phát xạ để thực hiện những yêu cầu kỹ thuật phức tạp.

Khi hoạt động, các đèn điện tử cần đốt nóng các sợi đốt (một sợi ở đèn hai cực, ba cực đơn hoặc nhiều sợi ở các đèn điện tử kép), khi nhiệt độ các sợi đốt đạt đến một mức độ nào đó, động năng của chúng thắng sự liên kết của kim loại và sẵn sàng nhảy ra khỏi bề mặt kim loại của sợi đốt.

Để điều khiển các đèn điện tử chân không, giữa các cực cần có một điện trường, chính các điện trường này đã tạo ra dòng điện trong chân không: điện tử di chuyển đến a-nốt.

- Nếu là đèn điện tử hai cực: Dòng điện tử đơn thuần di chuyển từ ca-tốt đến a-nốt với cường độ phụ thuộc vào điện trường tạo ra (cùng các thông số khác của đèn ảnh hưởng đến)
- Nếu là đèn điện tử ba cực, dòng điện này phụ thuộc vào cực điều khiển (như hình), điện trường cực điều khiển sẽ quyết định đến cường độ dòng điện đi đến a-nốt.

Do điện tử có khối lượng rất nhỏ, chuyển động hầu như không có quán tính nên sự không chế luồng điện tử này có thể tạo nên những luồng điện tức thời. Điện tử lại có điện tích rất nhỏ cho nên không chế luồng điện tử về mặt số lượng có thể tạo được những dòng điện rất nhỏ cho những dụng cụ cần độ nhạy cao, những biến thiên rất nhỏ cũng được cảm nhận, có thể tập trung để tạo được dòng điện rất lớn cho những dụng cụ cần có công suất mạnh. Đây chính là ưu điểm của đèn điện tử chân không so với các transistor điện tử bán dẫn khiến cho chúng còn được sử dụng trong các bộ ampli công suất để khuếch đại tín hiệu tương tự. (Ổ transistor có thể không "mở" khi mức độ tín hiệu (tương tự) thấp hơn một giá trị nhất định nào đó, dẫn đến sự khuếch đại bị thất thoát, làm ảnh hưởng đến âm thanh được khuếch đại)

Như vậy về mặt tần số, có những dụng cụ điện tử làm việc tới 10 mũ 12 Hz, về mặt công suất có những đèn phát tới vài trăm kw.

Năng lượng điện là loại năng lượng dễ chuyển hóa thành các dạng năng lượng khác nên dụng cụ điện tử rất tiện dụng cho những quá trình vật lý phức tạp như những biến đổi quang – điện, nhiệt – điện, bức xạ...

## [sửa] Ứng dụng

Với những ưu điểm đó, dụng cụ điện tử có thể thực hiện được nhiều chức năng kỹ thuật từ đơn giản đến phức tạp như:

- Đèn điện tử hai cực (tương đương điốt): nắn điện, tách sóng.
- Đèn điện tử chân không ba cực (tương tự các transistor bán dẫn): khuếch đại, tạo sóng, biến tần, hiện sóng, chỉ thị báo hiệu, truyền hình, đo lường, tự động...

## [sửa] Phân loại

Đèn điện tử có rất nhiều loại, nhiều công dụng khác nhau nên có rất nhiều cách phân loại. Về mặt công dụng có thể chia làm đèn khuếch đại, đèn nắn điện, đèn tách sóng, đèn đổi tần, đèn phát, đèn tạo sóng, đèn chỉ thị...

Về mặt chế độ công tác có thể chia làm đèn làm việc theo chế độ liên tục, đèn làm việc theo chế độ xung.

Về mặt tần số có thể chia làm đèn âm tần, đèn cao tần, đèn siêu cao tần.

Về mặt kết cấu nội bộ đèn có thể chia làm đèn 2 cực, đèn 3 cực, 4 cực, năm cực, nhiều cực, đèn ghép, đèn kép, đèn nung trực tiếp, đèn nung gián tiếp đèn ca tốt lạnh. Về mặt kết cấu ngoại hình có thể làm đèn vỏ thủy tinh, đèn vỏ kim loại, gốm.

Về mặt làm nguội có thể chia làm đèn làm nguội tự nhiên, làm nguội bằng gió, làm nguội bằng nước chảy đối lưu, làm nguội bằng cách bay hơi.

Về cách bố trí các chân đèn để sử dụng để đèn có thể chia làm loại 8 chân (octal), 9 chân tãm (noval), Rimlock, chân chia...

Người ta còn chia làm loại đèn chân không và đèn có khí, trong đó có đèn gazotron, thyratron, đèn ổn áp (Stabilitron).

Về nguyên lý công tác, đèn điện tử còn có các loại manhêtron, klystron, đèn sóng chạy dùng cho lĩnh vực siêu cao.

Về hiệu ứng sử dụng còn có các loại đèn tia âm cực dùng cho máy hiện sóng, máy thu hình áp dụng tính năng điện – quang để xem sóng, xem hình có các loại đèn quang điện (tế bào quang điện) đèn nhãn quang điện để thể hiện sự biến đổi ánh sáng thành sự biến đổi của dòng điện dùng cho âm thanh chiếu bóng hoặc trong thiết bị kiểm tra tự động.

Tóm lại, có rất nhiều cách phân loại đèn điện tử và có rất nhiều loại đèn điện tử thực hiện được nhiều yêu cầu kỹ thuật phức tạp và ứng dụng rất nhiều trong các lĩnh vực khác nhau của kỹ thuật.