

**ỦY BAN NHÂN DÂN QUẬN 5**  
**TRƯỜNG TRUNG CẤP NGHỀ KỸ THUẬT CÔNG NGHỆ HÙNG VƯƠNG**

---



**GIÁO TRÌNH**  
**Kỹ thuật số**  
**Nghề: Điện công nghiệp**  
**TRÌNH ĐỘ TRUNG CẤP**



2

MỤC LỤC

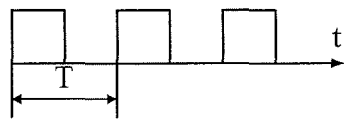
STT	Tên bài	Trang
	<b>KỸ THUẬT XUNG</b>	
1	Các khái niệm cơ bản	
2	Mạch dao động đa hài	
3	Mạch hạn chế biên độ và ghim	
	<b>KỸ THUẬT SỐ</b>	
4	Đại cương	
5	FLIP - FLOP	
6	Mạch logic MSI	
7	Mạch đếm và thanh ghi	
8	Họ vi mạch TTL - CMOS	
9	Bộ nhớ	
10	Kỹ thuật ADC - DAC	

# CHƯƠNG 1

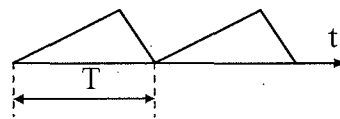
## CÁC KHÁI NIỆM CƠ BẢN

### KHÁI NIỆM

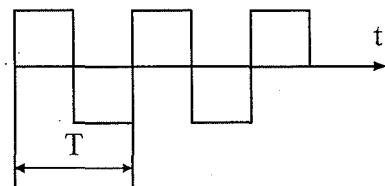
- Các tín hiệu điện có biên độ thay đổi theo thời gian được chia làm hai loại cơ bản là tín hiệu liên tục và tín hiệu rời rạc.
  - Tín hiệu liên tục (còn được gọi là tín hiệu tuyến tính hay tương tự) là tín hiệu có biên độ biến thiên liên tục theo thời gian.
  - Tín hiệu rời rạc (còn được gọi là tín hiệu xung hay số) là tín hiệu có biên độ biến thiên không liên tục theo thời gian.
- Tín hiệu xung: là tín hiệu rời rạc theo thời gian.
- Đặc điểm chung: là thời gian tồn tại xung rất ngắn hay sự biến thiên biên độ từ thấp lên cao hay từ cao xuống thấp xảy ra rất nhanh.
- Bản chất vật lý: dòng điện, điện áp, ánh sáng....
- Hình dạng: vuông, tam giác, răng cưa, nh?n, hình thang...



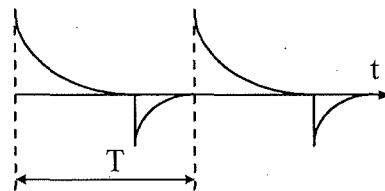
a. Xung vuông đơn cực



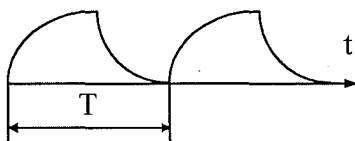
d. Xung tam giác



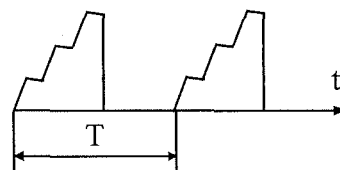
b. Xung vuông lưỡng cực



e. Xung nhọn lưỡng cực



c. Xung răng cưa



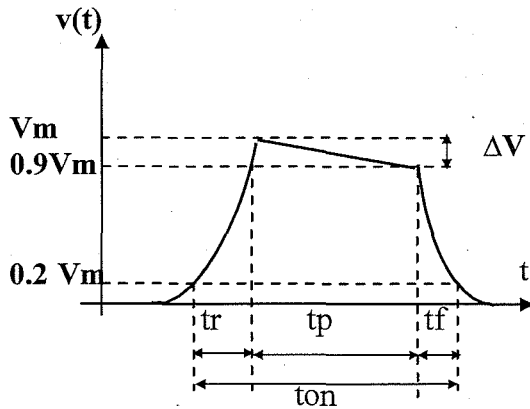
f. Xung nấc thang

**Hình 1.1** Các dạng tín hiệu xung

## CÁC THÔNG SỐ CƠ BẢN CỦA TÍN HIỆU XUNG

### Xung đơn

- Khái niệm: xung đơn là chỉ có một xung riêng biệt.



Trong đó:

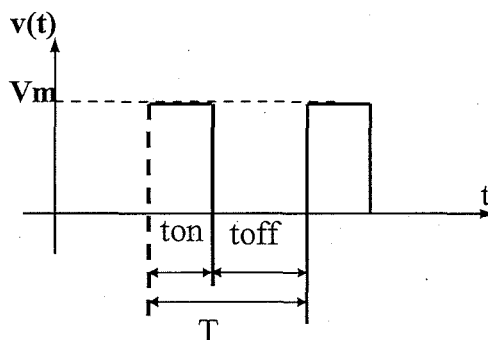
- $V_m$  : Biên độ xung.
- $\Delta V$  : Độ sụt áp đỉnh xung.
- $t_r$  : Độ rộng sườn trước.
- $t_p$  : Độ rộng đỉnh xung.
- $t_f$  : Độ rộng sườn sau.
- $t_{on}$  : Độ rộng xung thực tế.

**Hình 1.2** Xung vuông đơn

- Độ rộng sườn trước, độ rộng sườn sau là thời gian biên độ xung tăng hay giảm trong khoảng  $0,1V_m$  đến  $0,9V_m$ .
- Độ rộng đỉnh xung là thời gian xung có biên độ nằm trong khoảng từ  $0,9 V_m$  đến  $V_m$  ứng với đoạn đỉnh.
- Độ rộng xung thực tế là:
 
$$t_{on} = t_r + t_p + t_f$$
- Độ sụt áp đỉnh xung  $\Delta V$  là độ giảm biên độ ở phần đỉnh xung.

### Dãy xung

- Khái niệm: dãy xung là tín hiệu gồm nhiều xung đơn.  
Dãy xung có thể tuần hoàn hoặc không tuần hoàn.



Trong đó:

- $V_m$  : Biên độ xung.
- $t_{on}$  : Độ rộng xung.
- $t_{off}$  : Thời gian không có xung.
- $T$  : Chu kỳ

**Hình 1.3** Dãy xung vuông tuần hoàn

- Độ rộng của xung là thời gian ứng với điện áp cao gọi là  $t_{on}$  (hay  $t_x$ ).
- Thời gian không có xung ứng với điện áp thấp gọi là  $t_{off}$  (hay thời gian nghỉ  $t_{ng}$ ).
- Chu kỳ xung là:  $T = t_{on} + t_{off}$  (s)
- Xung vuông đối xứng:  $t_{on} = t_{off}$

- Tần số là số xung xuất hiện trong một đơn vị thời gian, được tính theo công thức:

$$f = \frac{1}{T} \quad (\text{Hz})$$

- Độ rộng của xung là tỉ số giữa chu kỳ T và độ rộng xung ton:

$$Q = \frac{T}{t_{\text{on}}}$$

- Nghịch đảo của độ rộng Q được gọi là hệ số đầy xung:

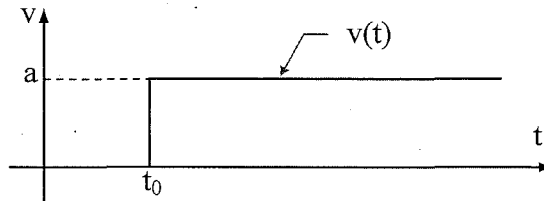
$$\eta = \frac{t_{\text{on}}}{T}$$

Để phân tích tín hiệu xung cần đưa về các dạng hàm cơ bản: hàm đột biến, hàm tuyến tính, hàm mũ giảm, hàm mũ tăng.

### CÁC DẠNG HÀM CƠ BẢN

**Hàm đột biến:**  $v(t) = a \cdot 1(t - t_0)$ .

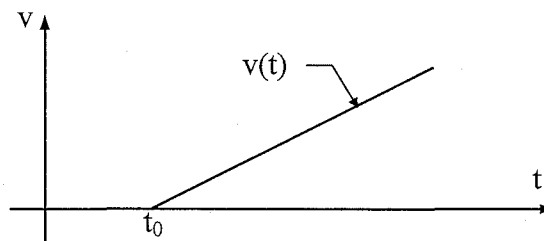
- Đột biến xảy ra tại thời điểm  $t = t_0$  với biên độ là a.
- $1(t - t_0)$ : Hàm đột biến đơn vị.
- Khi  $t < t_0$ :  $v = 0$
- Khi  $t \geq t_0$ :  $v = a$



**Hình 1.4** Hàm đột biến

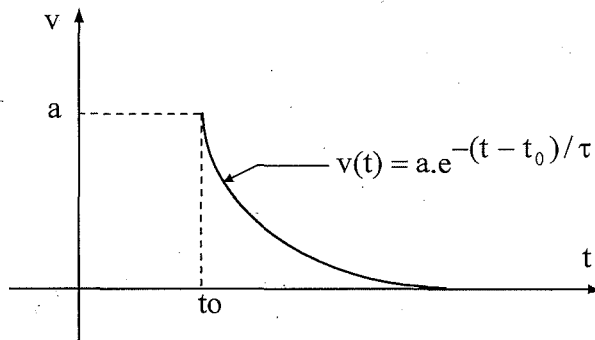
**Hàm tuyến tính:**  $v(t) = k(t - t_0)$ .

- k: Độ dốc của hàm.



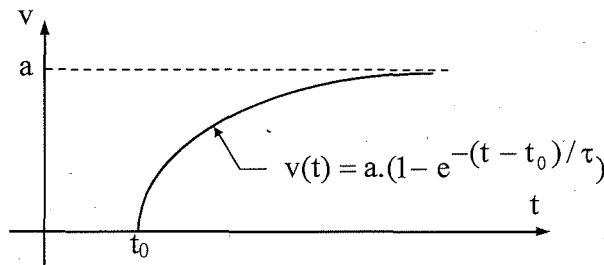
**Hình 1.5** Hàm tuyến tính.

**Hàm mũ giảm:**  $v(t) = a \cdot e^{-(t-t_0)/\tau}$



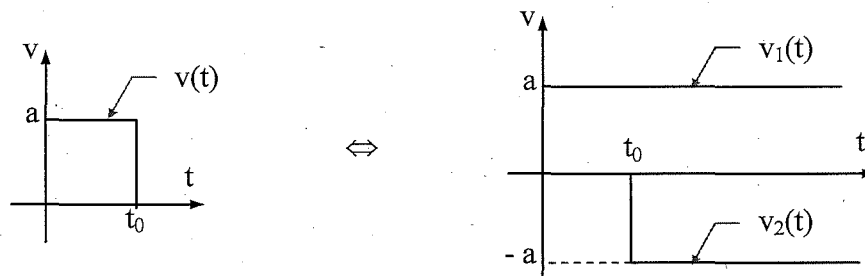
**Hình 1.6** Hàm mũ giảm

**1.3.4. Hàm mũ tăng:**  $v(t) = a \cdot (1 - e^{-(t-t_0)/\tau})$



**Hình 1.6** Hàm mũ tăng.

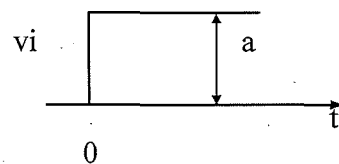
- ❖ Để phân tích 1 tín hiệu xung, phải đưa về dạng tổng các hàm cơ bản.  
Ví dụ:



Suy ra:  $v(t) = v_1(t) + v_2(t) = a \cdot 1(t) - a \cdot 1(t - t_0)$

**MẠCH RC**

**Phản ứng với hàm đột biến điện áp:**  $v_i = a \cdot 1(t)$



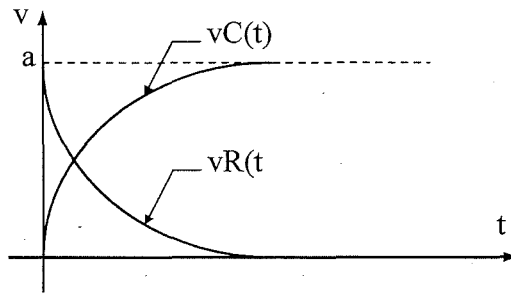
•  $t < 0:$   $v_i = 0, i =$

$0$   
 $v_R = 0, v_C = 0$

- $t = 0$  :  $v_i = a$   
 Mặt khác:  $v_i = v_C + v_R$   
 Mà  $v_C = 0$  (Điện áp trên tụ không đột biến).  
 Do đó:  $v_R = a \Rightarrow i = \frac{v_R}{R} = \frac{a}{R}$
- $t > 0$  : Tụ C nạp bằng dòng điện  $i = \frac{v_i - v_C}{R}$   
 $v_C$  tăng dần,  $\Rightarrow i$  giảm dần,  $v_R$  giảm dần.
- $t = \infty$  : Mạch xác lập.  
 Tụ C nạp đầy với  $v_C = v_i = a$ ,  $v_R = 0$ ,  $i = 0$

$$\Rightarrow \begin{cases} v_R = a \cdot e^{-t/\tau} \\ v_C = a \cdot (1 - e^{-t/\tau}) \end{cases}$$

$\tau = RC$  : Được gọi là thời hằng.

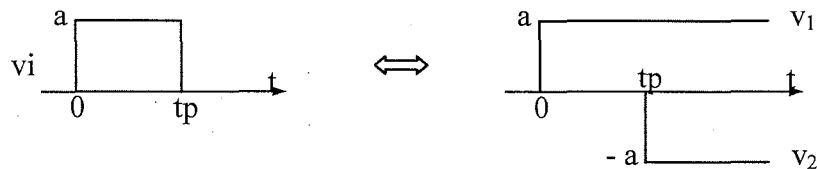


$\tau$  đặc

trung cho tốc độ diễn

ra quá trình quá độ.  $\tau$  càng lớn, quá trình quá độ càng kéo dài, mạch lâu xác lập.

**Phản ứng với xung vuông:**



❖ Phân tích  $v_i$  thành tổng các hàm cơ bản, ta có:

$$v_i = v_1 + v_2$$

Với :

$$v_1 = a \cdot 1(t)$$

$$v_2 = -a \cdot 1(t - t_p)$$

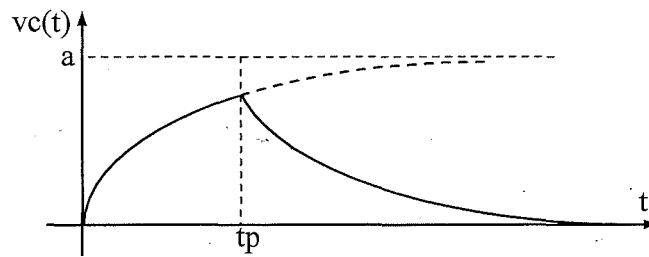
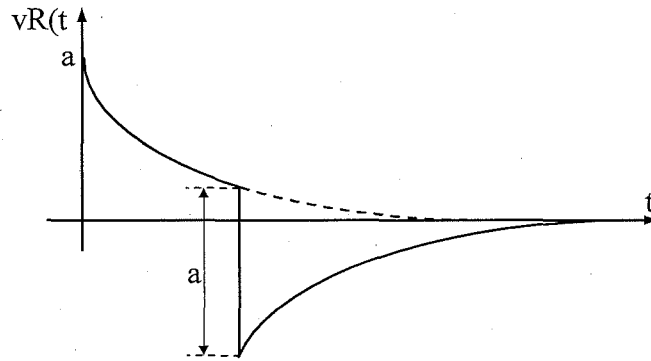
$$\Rightarrow \begin{cases} v_R = a \cdot e^{-t/\tau} - a \cdot e^{-(t-t_p)/\tau} \\ v_C = a \cdot (1 - e^{-t/\tau}) - a \cdot (1 - e^{-(t-t_p)/\tau}) \end{cases}$$

❖ Quá trình vật lý trong mạch:

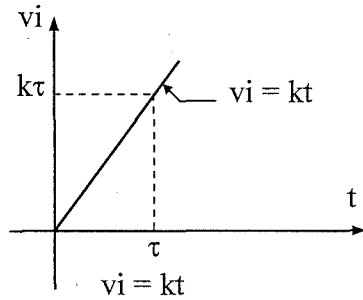
- $t < 0$  :  $v_i = 0$ ,  $i = 0$ ,  $v_C = 0$ ,  $v_R = 0$



- $0 \leq t < t_p$  :  $v_i = a$  ; tụ C nạp điện bằng dòng  $i = \frac{v_i - v_C}{R}$ .  
 $v_C$  tăng dần,  $\Rightarrow i$  giảm dần,  $v_R$  giảm.
- $t_p \leq t$  :  $v_i = 0$ , tụ C phóng điện qua R, với dòng  $i = -\frac{v_C}{R}$ .  
Điện áp  $v_R$  giảm dần đến 0.

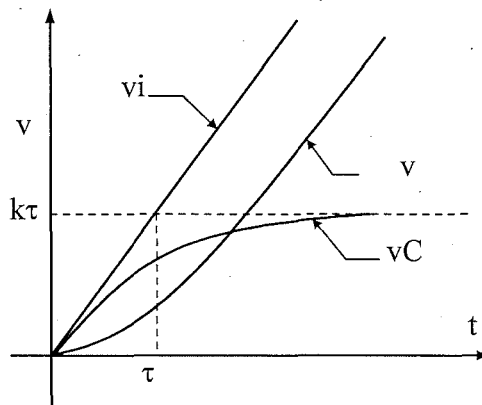


**Phản ứng với hàm tuyến tính:**



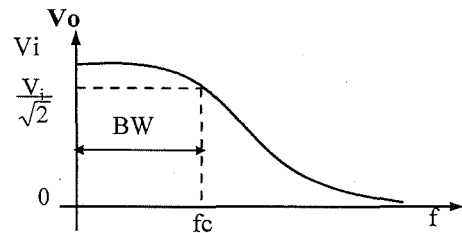
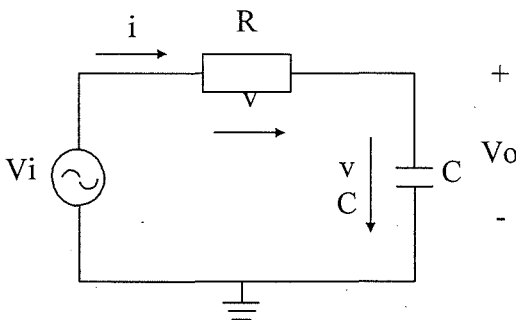
Bằng cách lý luận tương tự, ta có:

$$\begin{cases} v_R = k\tau.(1 - e^{-t/\tau}) \\ v_C = v_i - v_R = kt - k\tau.(1 - e^{-t/\tau}) \end{cases}$$



- ✓ Mạch RC lấy tín hiệu ra trên tải C thì được gọi là mạch thông thấp(hạ thông ). Nếu mạch hạ thông có thời hằng rất lớn thì được gọi là mạch tích phân (Mạch lọc thông thấp).
- ✓ Mạch RC lấy tín hiệu ra trên tải R thì được gọi là mạch thông cao(thượng thông ). Nếu mạch thượng thông có thời hằng rất nhỏ thì được gọi là mạch vi phân (Mạch lọc thông cao).

**Mạch thông thấp RC**



**Hình 1.8a** Mạch thông thấp RC

**Hình 1.8b** Đáp ứng tần số

- Tín hiệu ra lấy trên C.
- Mạch thông thấp cho các tín hiệu có tần số nhỏ hơn tần số cắt qua hoàn toàn, tín hiệu có tần số cao bị suy giảm biên độ. Tín hiệu ra trễ pha so với tín hiệu vào.

• Tần số cắt  $f_c = \frac{1}{2\pi RC}$

Tại tần cắt điện áp ra có biên độ  $V_o = \frac{V_i}{\sqrt{2}}$

#### a. Mạch tích phân RC :

**ĐN:** Mạch tích phân là mạch mà điện áp ra  $V_o(t)$  tỉ lệ với tích phân theo thời gian của điện áp vào  $V_i(t)$ :

$$V_o(t) = K \int V_i(t) dt$$

Trong đó K là hệ số tỉ lệ.

Mạch tích phân RC chính là mạch lọc thông thấp RC khi tín hiệu vào có tần số  $f_i$  rất lớn so với tần số cắt  $f_c$  của mạch.

#### • Thiết lập công thức:

Từ hình 1.8a ta có:  $V_i(t) = V_R(t) + V_C(t)$  (1)

Từ điều kiện tần số  $f_i$  rất lớn so với tần số cắt  $f_c$  ta có:

$$f_i \gg f_c = \frac{1}{2\pi RC}$$

$$\Rightarrow R \gg X_C = \frac{1}{2\pi f_i C}$$

$$\Rightarrow V_R(t) \gg V_C(t) \quad (2) \quad (\text{vì dòng } i(t) \text{ qua R và C bằng nhau})$$

Từ (1) và (2) ta có:  $V_i(t) \approx V_R(t) = R \cdot i(t)$

$$\Rightarrow i(t) = \frac{V_i(t)}{R} \quad (3)$$

Điện áp ra  $V_o(t)$ :

$$V_o(t) = V_C(t) = \frac{1}{C} \int i(t) dt$$

$$\Rightarrow V_o(t) = \frac{1}{C} \int \frac{V_i(t)}{R} dt$$

$$\Rightarrow V_o(t) = \frac{1}{RC} \int V_i(t) dt$$

Như vậy, điện áp ra  $V_o(t)$  tỉ lệ với tích phân theo thời gian của điện áp vào  $V_i(t)$  với hệ số tỉ lệ K là  $K = \frac{1}{RC}$  khi tần số  $f_i$  rất lớn so với  $f_c$ .

#### • Điều kiện mạch tích phân:

$$f_i \gg f_c \quad \Leftrightarrow \quad f_i \gg \frac{1}{2\pi RC}$$

$$RC \gg \frac{1}{2\pi f_i} \Leftrightarrow \tau \gg \frac{1}{2\pi f_i} = \frac{T_i}{2\pi}$$

Trong đó:  $\tau = R.C$  là hằng số thời gian.  
 $T_i$  là chu kỳ của tín hiệu vào.

**VD:** Trường hợp điện áp vào  $V_i(t)$  là tín hiệu hình sin qua mạch tích phân:

$$V_i(t) = V_m \sin \omega(t)$$

$$\text{Điện áp ra: } V_0(t) = \frac{1}{RC} \int V_m \sin \omega t dt = -\frac{V_m}{\omega RC} \cos \omega t$$

$$V_0(t) = \frac{V_m}{\omega RC} \sin(\omega t - 90^\circ)$$

Như vậy, nếu thỏa mãn điều kiện của mạch tích phân như trên thì điện áp ra bị trễ pha  $90^\circ$  và biên độ bị giảm xuống với hệ số tỉ lệ là  $\frac{1}{\omega RC}$ .

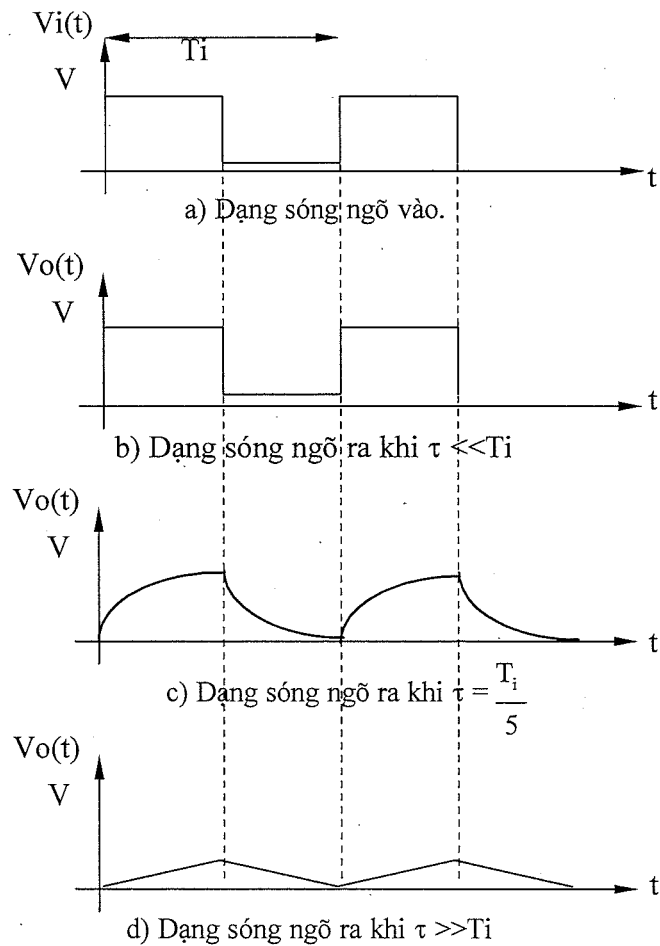
### b. Điện áp vào là tín hiệu xung vuông :

Khi điện áp vào là tín hiệu xung vuông có chu kỳ  $T_i$  thì có thể xét tỉ lệ hằng số thời gian  $\tau = RC$  so với  $T_i$  để giải thích các dạng sóng ra theo hiện tượng nạp xả của tụ.

Giả sử điện áp ngõ vào là tín hiệu xung vuông đối xứng có chu kỳ  $T_i$  (hình 1.9a).

- Nếu mạch tích phân có hằng số thời gian  $\tau = RC$  rất nhỏ so với  $T_i$  thì tụ nạp và xả rất nhanh nên điện áp ngõ ra  $V_0(t)$  có dạng giống như dạng điện áp vào  $V_i(t)$  (hình 1.9b).
- Nếu mạch tích phân có hằng số thời gian  $\tau = \frac{T_i}{5}$  thì tụ nạp và xả điện áp theo dạng hàm số mũ, biên độ đỉnh của điện áp ra nhỏ hơn  $V_p$  (hình 1.9c)
- Nếu mạch tích phân có hằng số thời gian  $\tau$  rất lớn so với  $T_i$  thì tụ C nạp rất chậm nên điện áp ra có biên độ rất thấp (hình 1.9d) nhưng đường tăng giảm điện áp gần như đường thẳng.

Như vậy, mạch tích phân chọn trị số  $RC$  thích hợp thì có thể sửa dạng xung vuông ở ngõ vào thành dạng sóng tam giác ở ngõ ra. Nếu xung vuông đối xứng thì xung tam giác ra là tam giác cân.



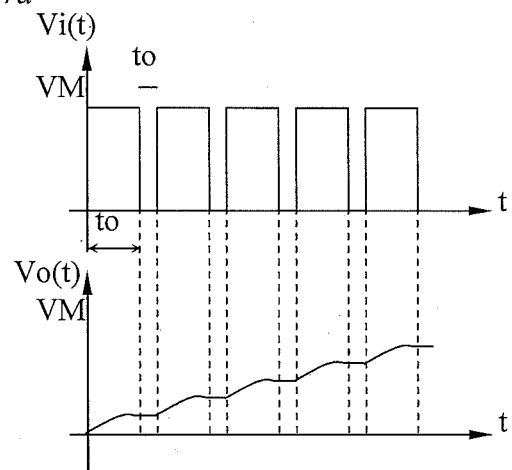
**Hình 1.9:** Dạng sóng vào và ra

Trường hợp ngõ vào là một chuỗi xung vuông không đối xứng với  $t_{on} > t_{off}$  qua mạch tích phân.

Trong thời gian  $t_{on}$  ở ngõ vào có điện áp cao nên tụ C nạp điện.

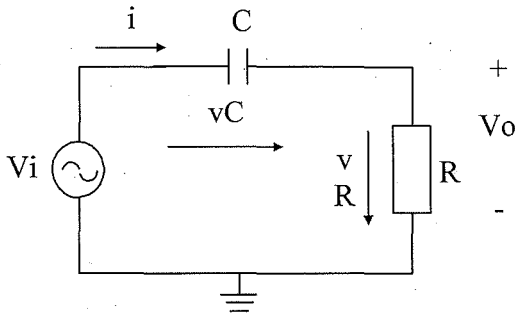
Trong thời gian  $t_{off}$  ngõ vào có điện áp  $0^V$  nên tụ xả điện nhưng do thời gian  $t_{off}$  nhỏ hơn  $t_{on}$  nên tụ chưa xả điện hết thì lại nạp điện tiếp làm cho điện áp của tụ tăng dần (hình 1.10).

Như vậy, tín hiệu ra có dạng xung nấc thang

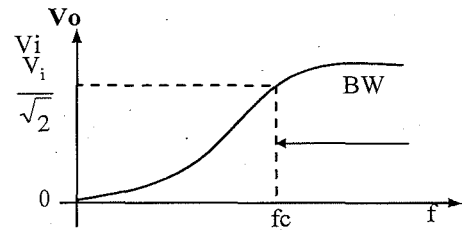


**Hình 1.10:** Chuỗi xung vuông vào

### Mạch thông cao RC



**Hình 1.11a** Mạch thông cao RC



**Hình 1.11b** Đáp ứng tần số

- Tín hiệu ra lấy trên R.
- Mạch thông cao cho các tín hiệu có tần số cao hơn tần số cắt qua hoàn toàn, tín hiệu có tần số thấp bị suy giảm biên độ. Tín hiệu ra sớm pha so với tín hiệu vào.

- Tần số cắt  $f_c = \frac{1}{2\pi RC}$

Tại tần cắt điện áp ra có biên độ  $V_o = \frac{V_i}{\sqrt{2}}$

#### a. Mạch vi phân RC :

**ĐN:** Mạch vi phân là mạch có điện áp ngõ ra  $V_o(t)$  tỉ lệ với đạo hàm theo thời gian của điện vào  $V_i(t)$ .

Ta có:  $V_o(t) = K \frac{d}{dt} V_i(t)$

Trong đó K là hệ số tỉ lệ.

Mạch vi phân RC chính là mạch lọc thông cao RC khi tín hiệu vào có tần số  $f_i$  rất thấp so với tần số cắt  $f_c$  của mạch.

Trong kỹ thuật xung, mạch vi phân có tác dụng thu hẹp độ rộng xung tạo ra các xung nhọn để kích các linh kiện điều khiển hay linh kiện công suất khác như SCR, Triac...

- **Thiết lập công thức:**

Từ hình 1.11a ta có:  $V_i(t) = V_R(t) + V_C(t)$  (1)

Từ điều kiện tần số  $f_i$  rất thấp so với tần số cắt  $f_c$  ta có:

$$f_i \ll f_c = \frac{1}{2\pi RC}$$

$$\Rightarrow R \ll X_c = \frac{1}{2\pi f_i C}$$

$$\Rightarrow V_R(t) \ll V_C(t) \quad (2) \text{ (vì dòng } i(t) \text{ qua R và C bằng nhau)}$$

Từ (1) và (2) ta có:  $V_i(t) \approx V_C(t)$

Đối với tụ C, điện áp trên tụ còn được tính theo công thức:

$$V(t) = \frac{q(t)}{C}$$

Trong đó  $q(t)$  là điện tích nạp vào tụ.

Từ đó ta có :

$$\frac{dV_i(t)}{dt} = \frac{dV_c(t)}{dt} = \frac{1}{C} \frac{dq(t)}{dt} = \frac{1}{C} i(t)$$

Hay là:  $i(t) = C \frac{dV_i(t)}{dt}$  (3)

Điện áp ra  $V_0(t)$ :

$$\begin{aligned} V_0(t) &= VR(t) = R \cdot i(t) \\ \Rightarrow V_0(t) &= RC \frac{dV_i(t)}{dt} \end{aligned}$$

Như vậy, điện áp ra  $V_0(t)$  tỉ lệ với vi phân ( đạo hàm) theo thời gian của điện áp vào với hệ số tỉ lệ  $K$  là  $K = RC$  khi tần số  $f_i$  rất thấp so với  $f_C$ .

• **Điều kiện mạch vi phân:**

$$\begin{aligned} f_i \ll f_C &\Leftrightarrow f_i \ll \frac{1}{2\pi RC} \\ RC \ll \frac{1}{2\pi f_i} &\Leftrightarrow \tau \ll \frac{1}{2\pi f_i} = \frac{T_i}{2\pi} \end{aligned}$$

Trong đó:  $\tau = R \cdot C$  là hằng số thời gian.  
 $T_i$  là chu kỳ của tín hiệu vào.

**VD:** Trường hợp điện áp vào  $V_i(t)$  là tín hiệu hình sin qua mạch vi phân:

$$V_i(t) = V_m \cdot \sin \omega(t)$$

$$\begin{aligned} \text{Điện áp ra là: } V_0(t) &= RC \frac{d}{dt} (V_m \cdot \sin \omega t) \\ &= \omega R \cdot C \cdot V_m \cos \omega t \\ &= \omega R \cdot C \cdot V_m \cdot \sin(\omega t + 90^\circ) \end{aligned}$$

Như vậy, nếu thỏa mãn điều kiện của mạch vi phân như trên thì điện áp ra bị sớm pha  $90^\circ$  và biên độ nhân với hệ số tỉ lệ là  $\omega RC$ .

**b. Điện áp vào là tín hiệu xung vuông:**

Khi điện áp vào là tín hiệu xung vuông có chu kỳ  $T_i$  thì có thể xét tỉ lệ hằng số thời gian  $\tau = RC$  so với  $T_i$  để giải thích các dạng sóng ra theo hiện tượng nạp xả của tụ. Giả sử điện áp ngõ vào là tín hiệu xung vuông đối xứng có chu kỳ  $T_i$  (hình 1.12a).

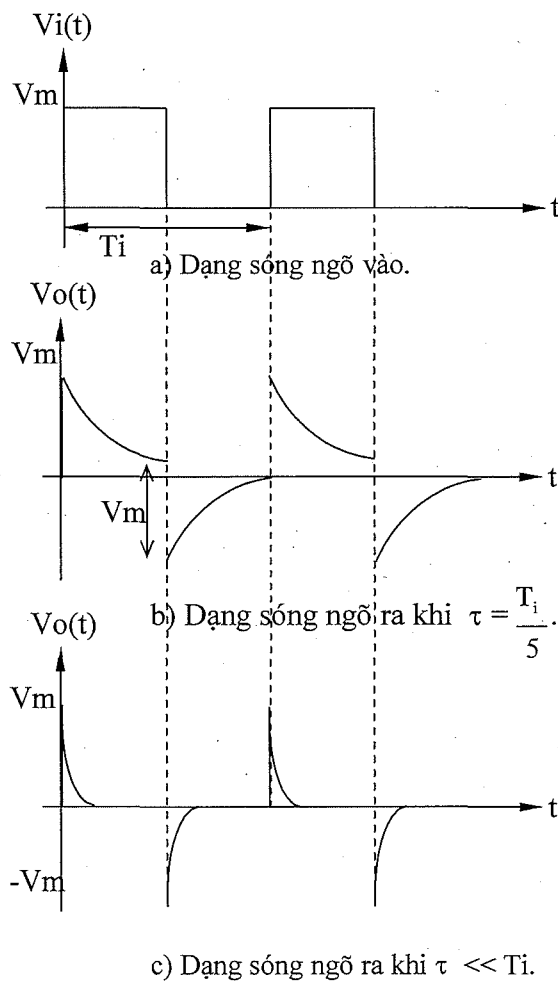
- Nếu mạch vi phân có hằng số thời gian  $\tau = \frac{T_i}{5}$  thì tụ nạp và xả điện tạo dòng

$i(t)$  qua điện trở  $R$  tạo ra điện áp giảm theo hàm số mũ. Khi điện áp ngõ vào bằng  $0^V$  thì đầu dương của tụ nối mass và tụ sẽ xả điện áp âm trên điện trở  $R$ . Ở ngõ ra sẽ có hai xung ngược nhau có biên độ giảm dần (1.12b)

- Nếu mạch vi phân có hằng số thời gian  $\tau$  rất nhỏ so với  $T_i$  thì tụ sẽ nạp xả điện rất nhanh nên cho ra hai xung ngược dấu nhưng có độ rộng xung rất hẹp được gọi là xung nhọn.

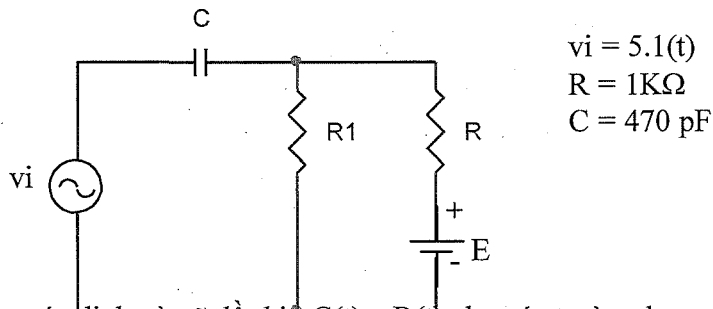
Như vậy, nếu thỏa điều kiện của mạch vi phân thì mạch RC sẽ đổi tín hiệu từ xung vuông đơn cực ra 2 xung nhọn lưỡng cực.





Hình 1.12: Dạng sóng vào và ra của mạch vi phân nhận xung vuông.

Ví dụ 1: Cho mạch như hình vẽ:



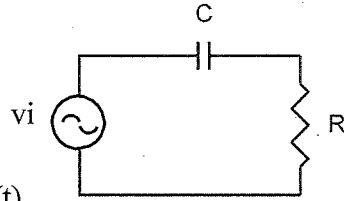
Hãy xác định và vẽ đồ thị  $v_C(t)$ ,  $v_R(t)$  cho các trường hợp sau:

- $E = 0, R_1 = \infty$
- $E = 1V, R_1 = \infty$
- $E = 1V, R_1 = 2K\Omega$

Bài giải

a.  $E = 0, R1 = \infty$

Mạch tương đương:

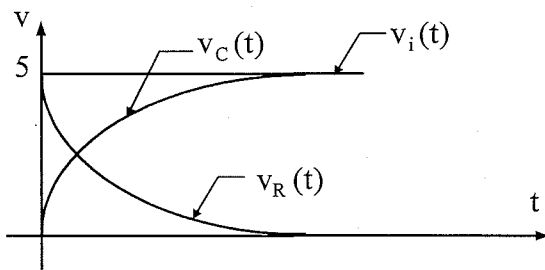


$$v_i = 5.1(t)$$

$$v_R = 5.e^{-t/\tau}$$

$$v_C = 5.(1 - e^{-t/\tau})$$

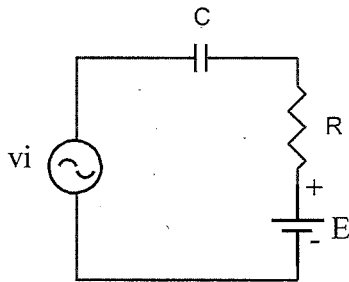
$$\text{Với } \tau = RC = 470.10^{-12}.1.10^3 = 470.10^{-9} = 0,47 (\mu s)$$



b.  $E$

$= 1V, R1 = \infty$

Mạch tương đương:



✓ Xét tác dụng của nguồn  $E$ :

$$i_E = 0$$

$$v_R^E = i_E.R = 0$$

$$v_C^E = i.R = 0$$

✓ Xét tác dụng của nguồn  $v_i$ :

$$v_R^E = i.R = 0$$

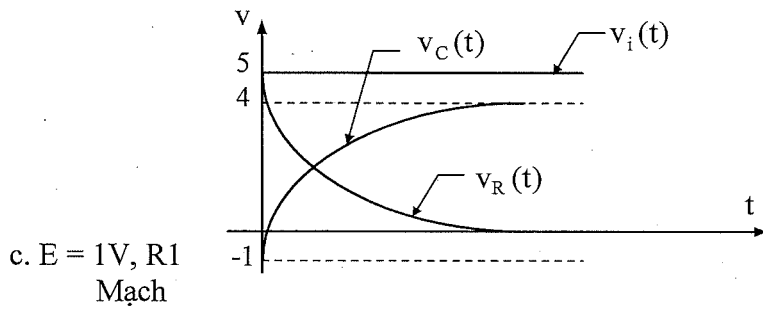
$$v_C^i = 5.(1 - e^{-t/\tau})$$

$$\tau = RC = 0.47 \mu s$$

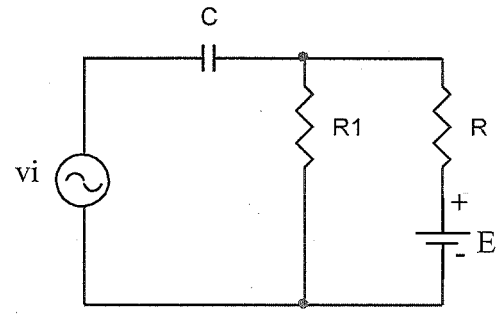
✓ Cộng tác dụng của 2 nguồn, suy ra:

$$v_R = 5.e^{-t/\tau}$$

$$v_C = 5.(1 - e^{-t/\tau}) - 1$$

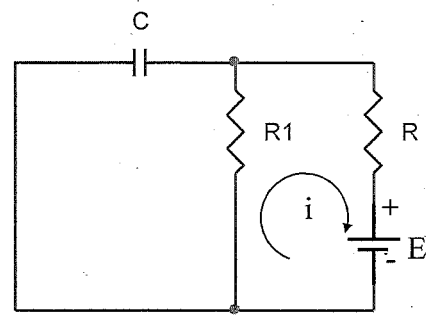


$= 2K\Omega$   
tương đương:



nguồn E:

✓ Xét tác dụng của



Ta có:  $i = -\frac{E}{R+R1}$

$v_R^E = i.R = -E \frac{R}{R+R1} = -\frac{1}{3} (V)$

$v_C^E = -v_{R1}^E = i.R1 = -\frac{2}{3} (V)$

✓ Xét tác dụng của nguồn v<sub>i</sub>:

$R_{td} = (R1 // R) = \frac{2}{3} (K\Omega)$

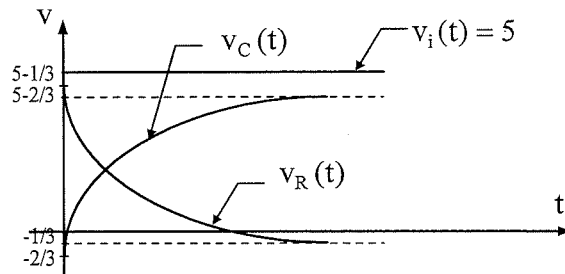
$v_R^i = v_{Rtd} = 5.e^{-t/\tau}$

$v_C^i = 5.(1 - e^{-t/\tau})$

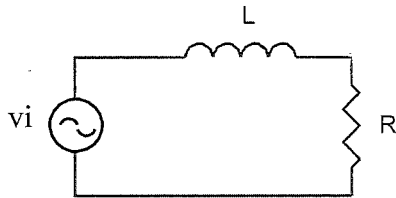
$\tau = R_{td}.C = 0,313 \mu s$

✓ Cộng tác dụng của 2 nguồn:

$$\Rightarrow \begin{cases} v_R = 5 \cdot e^{-t/\tau} - \frac{1}{3} \\ v_C = 5 \cdot (1 - e^{-t/\tau}) - \frac{2}{3} \end{cases}$$



**Mạch RL:**

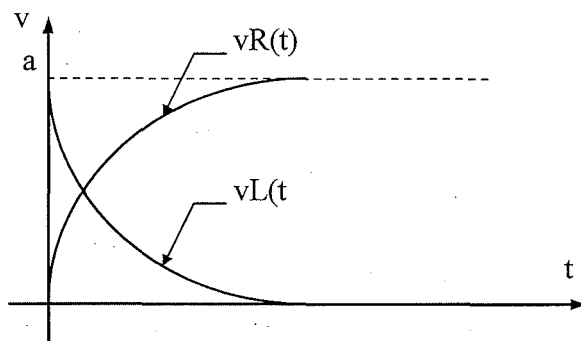


**Phản ứng với hàm đột biến điện áp:  $v_i = a \cdot 1(t)$**

- $t < 0$  :  $v_i = 0$   
Suy ra:  $v_R = 0, v_L = 0$
- $t = 0$  :  $v_i = a$   
Suy ra:  $i = 0$  ( dòng qua cuộn dây không đột biến ).  
 $v_R = 0$   
 $v_L = v_i - v_R = a$
- $t > 0$  : Dòng qua cuộn dây tăng dần,  $v_R$  tăng,  $v_L$  giảm.
- $t = \infty$  : Mạch xác lập.  
 $v_L = 0$   
 $v_R = a$

$$\Rightarrow \begin{cases} v_R = a \cdot (1 - e^{-t/\tau}) \\ v_L = a \cdot \frac{e^{-t/\tau}}{e} \end{cases}$$

Với  $\tau = L/R$  được gọi là thời hằng.

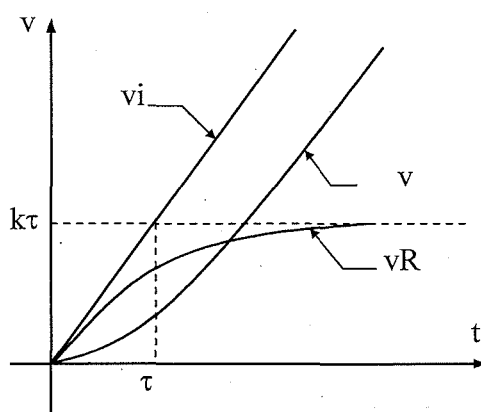


**Phản ứng với hàm tuyến tính:  $v_i = kt$**

Tương tự ta có được:

$$\begin{cases} v_L = k\tau \cdot (1 - e^{-t/\tau}) \\ v_R = kt - k\tau \cdot (1 - e^{-t/\tau}) \end{cases}$$

Với  $\tau = L/R$ .



✓ Mạch RL lấy tín

hiệu ra trên tải R thì được gọi là mạch hạ thông (thông thấp).

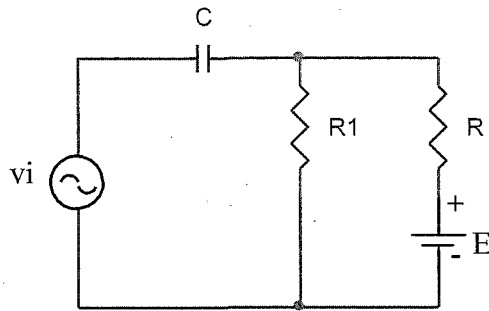
✓ Mạch RL lấy tín hiệu ra trên tải L thì được gọi là mạch thượng thông (thông cao).

❖ **Nhận xét:**

- Phản ứng của mạch RL thông cao giống phản ứng của mạch RC thông cao.
- Phản ứng của mạch RL thông thấp giống phản ứng của mạch RC thông thấp.

## BÀI TẬP CHƯƠNG I

**Bài 1:** Cho mạch như hình vẽ:



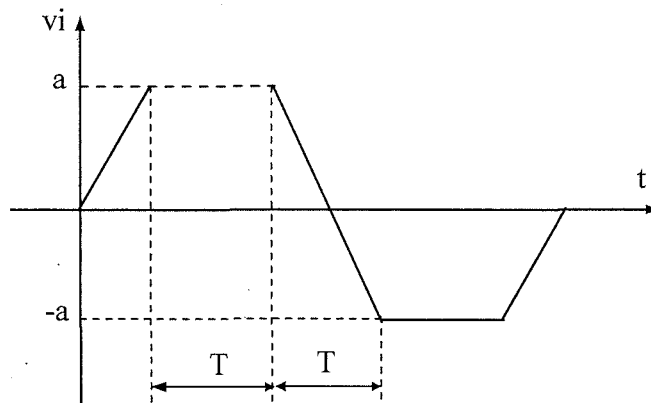
$$R = 1K\Omega$$

$$C = 470 \text{ pF}$$

Hãy xác định và vẽ đồ thị  $v_i(t)$ ,  $v_C(t)$ ,  $v_R(t)$  cho các trường hợp sau:

- $v_i(t) = 5.1(t) - 5.1(t - t_0)$ ;  $t_0 = 10 \mu\text{s}$ ;  $R1 = \infty$ ;  $E = 0$
- $v_i(t) = 5.1(t) - 5.1(t - t_0)$ ;  $t_0 = 10 \mu\text{s}$ ;  $R1 = 5,6K\Omega$ ;  $E = 2V$
- $v_i(t) = 5.1(t) - 7.1(t - t_0)$ ;  $t_0 = 10 \mu\text{s}$ ;  $R1 = 5,6K\Omega$ ;  $E = 2V$

**Bài 2:** Cho  $v_i$  như hình vẽ:

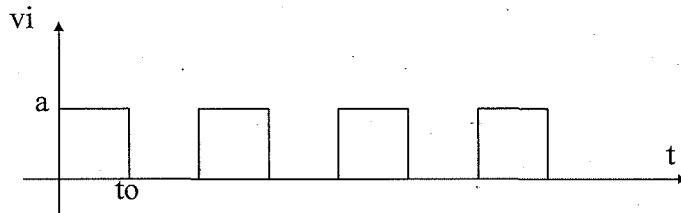


- Phân tích  $v_i(t)$  thành dạng tổng các hàm cơ bản.
- Khi đặt  $v_i$  ở ngõ vào của mạch RC thượng thông, hãy xác định và vẽ  $v_{OUT}$

$$\text{khi } T = \frac{RC}{10}.$$

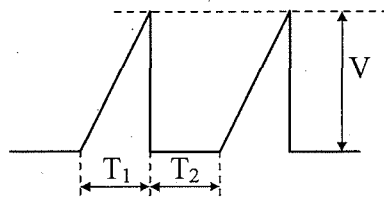
**Bài 3:** Xác định và vẽ  $v_R$ ,  $v_C$  của mạch RC khi  $v_i$  là chuỗi xung vuông có tần số là  $T = 2t_0$  trong các trường hợp:

- $\tau \gg t_0$ .
- $\tau \ll t_0$ .

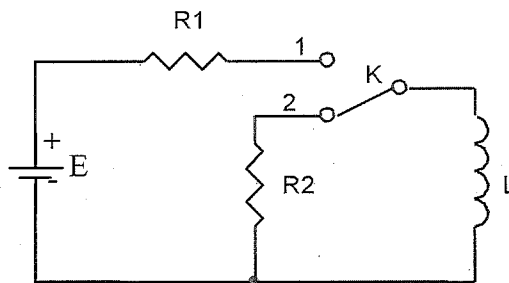


**Bài 4:**

- Tín hiệu  $v_i$  như hình vẽ được đưa vào mạch hạ thông RC. Hãy xác định và vẽ dạng tín hiệu ngõ ra.
- Cho  $T_1 = T_2 = RC$ , Hãy xác định giá trị lớn nhất và giá trị nhỏ nhất của tín hiệu ngõ ra; Vẽ dạng tín hiệu này.



**Bài 5:** Cho mạch như hình vẽ:



- Khi  $t < 0$  : K ở vị trí số 2.  
 $t = 0$  : K sang vị trí số 1.  
 $t = 30 \mu s$  : K trở lại vị trí số 2.

Hãy xác định và vẽ  $i_L$ ,  $v_L$  trong các trường hợp sau:

- $E = 10 \text{ V}$        $R_1 = 1\text{K}$        $R_2 = 1\text{K}$        $L = 1\text{mH}$
- $E = 10 \text{ V}$        $R_1 = 10\text{K}$        $R_2 = 10\text{K}$        $L = 10\text{mH}$

# CHƯƠNG 2

## MẠCH DAO ĐỘNG ĐA HÀI

### MẠCH XÉN

Chức năng: Giới hạn biên độ tín hiệu.

Phân loại theo chức năng có 3 loại:

- Mạch xén trên (xén dương).
- Mạch xén dưới (xén âm).
- Mạch xén hai mức (xén dương và xén âm)

Phân loại theo cấu trúc có 2 loại:

- Mạch xén nối tiếp: phần tử xén mắc nối tiếp với tải
- Mạch xén song song: phần tử xén mắc song song với tải

#### Mạch xén dùng Diode

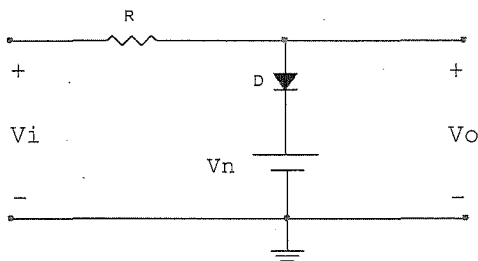
Đn: là một dạng mạch sửa dạng sóng rất phổ biến trong thực tế.

Giả sử các diode lý tưởng:

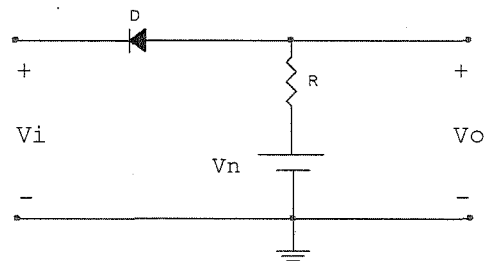
- $$\left\{ \begin{array}{l} V_\gamma = 0 : \text{điện áp bắt đầu dẫn} \\ I_s = 0 : \text{dòng rỉ khi phân cực nghịch} \\ r_f = 0 : \text{điện trở thuận} \\ r_s = \infty : \text{điện trở nghịch} \end{array} \right.$$

#### a. Mạch xén trên :

Xén phần tín hiệu lớn hơn giá trị VN.



Hình 2.1a. Mạch xén song



Hình 2.1b. Mạch xén nối tiếp

$V_i > V_N$  : D dẫn

$$\Rightarrow V_0 = V_N$$

$V_i \leq V_N$  : D tắt

$$\Rightarrow V_0 = V_i$$

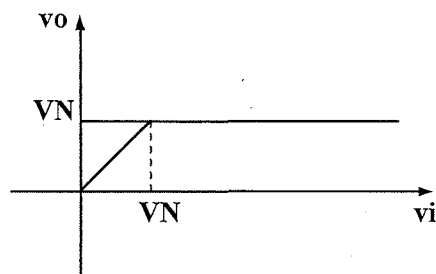
$V_i \geq V_N$  : D tắt

$$\Rightarrow V_0 = V_N$$

$V_i < V_N$  : D dẫn

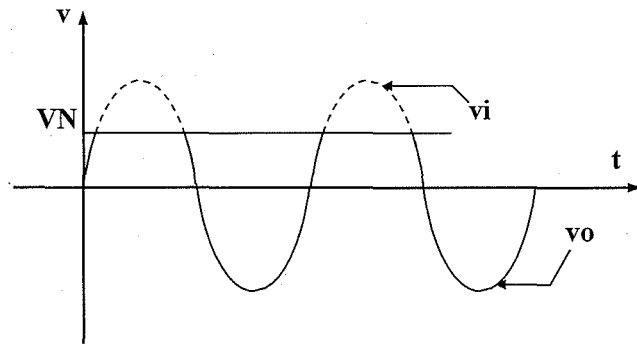
$$\Rightarrow V_0 = V_i$$

❖ Đặc tuyến hàm truyền đạt: Biểu diễn sự phụ thuộc giữa tín hiệu ngõ vào và tín hiệu ngõ ra.



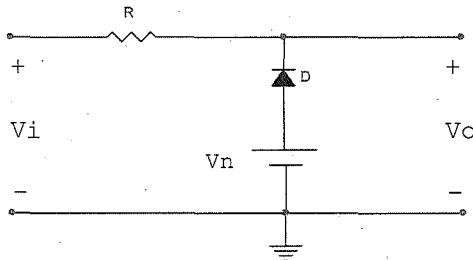


❖ Dạng tín hiệu ra khi tín hiệu vào là tín hiệu sin:

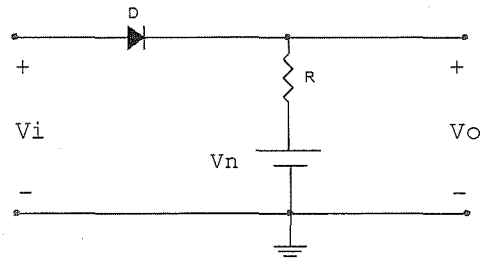


**b. Mạch xén dưới :**

Xén phần tín hiệu nhỏ hơn giá trị VN.



Hình 2.2a. Xén dưới song song

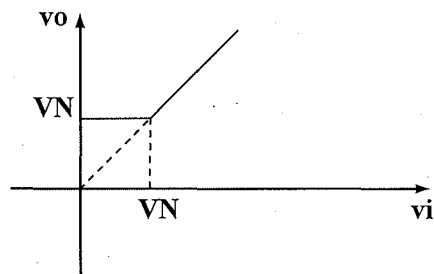


Hình 2.2 b. Xén dưới nối tiếp

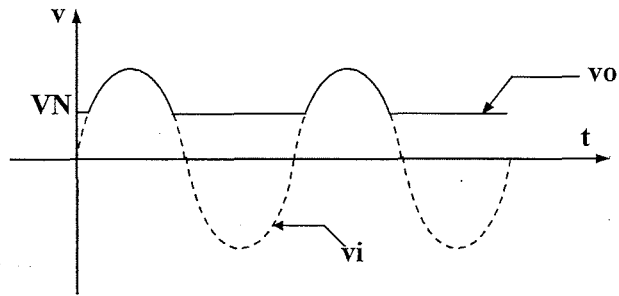
- $V_i \geq V_N$  : D tắt  
 $\Rightarrow V_o = V_i$
- $V_i < V_N$  : D dẫn  
 $\Rightarrow V_o = V_N$

- $V_i > V_N$  : D dẫn  
 $\Rightarrow V_o = V_i$
- $V_i \leq V_N$  : D tắt  
 $\Rightarrow V_o = V_N$

❖ Đặc tuyến hàm truyền đạt: Biểu diễn sự phụ thuộc giữa tín hiệu ngõ vào và tín hiệu ngõ ra.



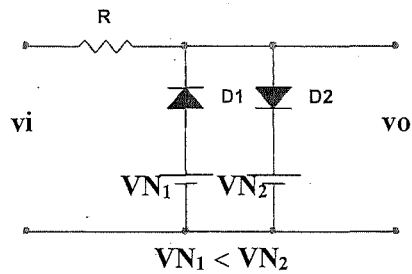
❖ Dạng tín hiệu ra khi tín hiệu vào là tín hiệu sin:



**c. Mạch xén 2 mức:**

Kết hợp mạch xén trên và xén dưới.

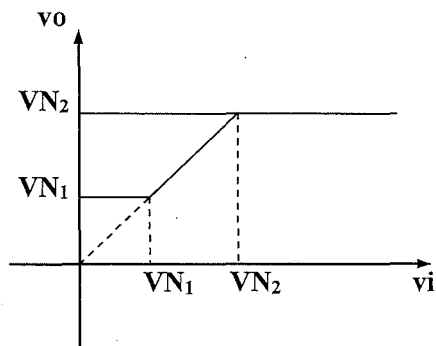
- Xén 2 mức nối tiếp.
- Xén 2 mức song song.
- Xén 2 mức nối tiếp và song song.



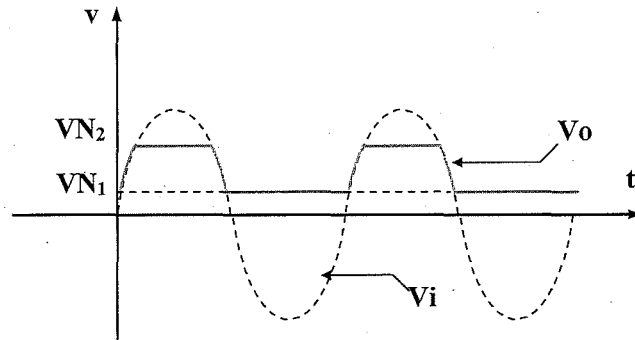
**Hình 2.3. Mạch song song**

- $V_i < VN_1 < VN_2$  : D<sub>1</sub> dẫn, D<sub>2</sub> tắt  $\Rightarrow V_o = VN_1$
- $VN_1 < V_i < VN_2$  : D<sub>1</sub> tắt, D<sub>2</sub> tắt  $\Rightarrow V_o = V_i$
- $V_i > VN_2 > VN_1$  : D<sub>1</sub> tắt, D<sub>2</sub> dẫn  $\Rightarrow V_o = VN_2$

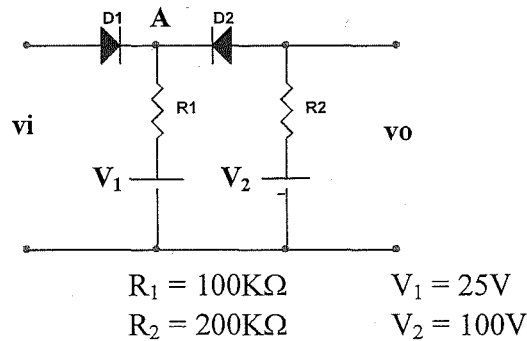
❖ Đặc tuyến hàm truyền đạt:



❖ Dạng tín hiệu ngõ ra khi tín hiệu ngõ vào có dạng hình sin:



Ví dụ: Cho mạch xén như hình vẽ.



Các diode  $D_1$  và  $D_2$  là diode lý tưởng.  
 Điện áp ngõ vào  $v_i$  biến thiên từ 0V đến 150V.  
 Hãy xác định và vẽ hàm truyền đạt.

Giải:

- $v_i = 0$  :  $D_1$  off,  $D_2$  on  
 Suy ra:  $v_o = V_A$

$$i = \frac{V_2 - V_1}{R_2 + R_1}$$

$$V_A = V_1 + iR_1$$

$$= V_1 + i \cdot R_1$$

$$= V_1 + \frac{V_2 - V_1}{R_2 + R_1} \cdot R_1$$

$$= 25 + \frac{100 - 25}{100 + 200} \cdot 100$$

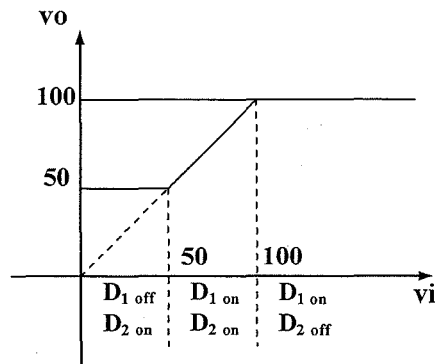
$$= 50 \text{ (V)}$$

- $V_2 > v_i > V_A$  :  $D_1$  on,  $D_2$  on  
 $v_o = V_A = v_i$

- $v_i \geq V_2$  :  $D_1$  on  
 Suy ra  $V_A = v_i$ , do đó  $D_2$  off

Dòng  $iR_2 = 0 \Rightarrow v_o = V_2 = 100$  (V).

- Đặc tuyến hàm truyền đạt:



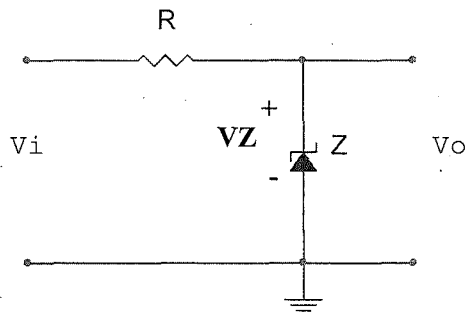
### Mạch xén 2 mức dùng diode zener:

Để giới hạn biên độ xung người ta còn dùng diode zener thay cho diode D và nguồn chuẩn VR. Điện áp  $V_Z$  do hiệu ứng zener sẽ là điện áp giới hạn biên độ xung.

Giả sử các diode zener lý tưởng:

- $V_\gamma = 0$  : điện áp bắt đầu dẫn
- $I_s = 0$  : dòng rỉ khi phân cực nghịch
- $r_f = 0$  : điện trở thuận
- $V_Z$  : điện áp ổn định củ Z khi phân cực nghịch

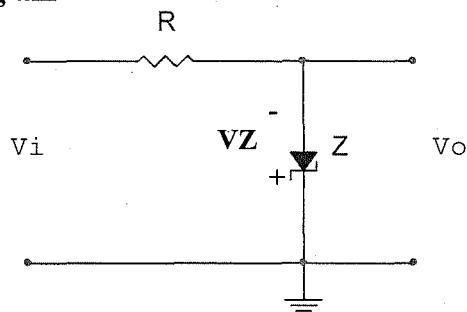
#### a. Giới hạn xung dương



Hình 2.4. Mạch giới hạn xung dương

- $V_i < 0$  : Z phân cực thuận, Z dẫn giống Diode  $\Rightarrow V_o = 0$
- $V_i \geq 0$  : Z phân cực nghịch
  - +  $V_i < V_Z$  : Z tắt  $\Rightarrow V_o = V_i$
  - +  $V_i > V_Z$  : Z dẫn ổn áp  $\Rightarrow V_o = V_Z$

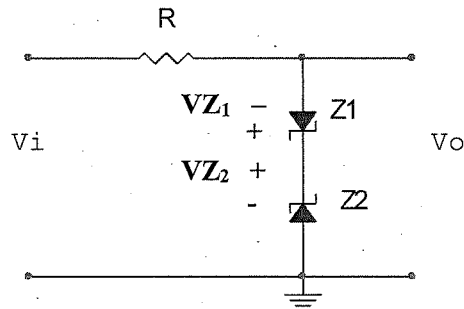
#### b. Giới hạn xung âm



Hình 2.5. Mạch giới hạn xung âm

- $V_i < 0$  : Z phân cực nghịch  
 $+ V_i > -V_Z$  : Z tắt  $\Rightarrow V_0 = V_i$   
 $+ V_i < -V_Z$  : Z dẫn ổn áp  $\Rightarrow V_0 = -V_Z$
- $V_i > 0$  : Z phân cực thuận, Z dẫn giống Diode  $\Rightarrow V_0 = 0$

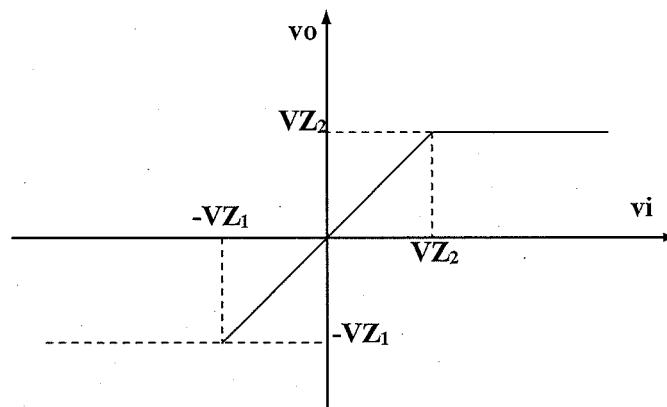
**c. Giới hạn xung âm dương**



**Hình 2.5. Mạch giới hạn xung âm dương**

- $V_i < -V_{Z1} < 0$  :  $Z_1$  phân cực nghịch,  $Z_1$  dẫn ổn áp  
 $Z_2$  phân cực thuận,  $Z_2$  dẫn bình thường  
 $\Rightarrow V_0 = -V_{Z1}$
- $-V_{Z1} < V_i < 0$  :  $Z_1$  phân cực nghịch,  $Z_1$  tắt  
 $\Rightarrow V_0 = V_i$
- $V_{Z2} > V_i > 0$  :  $Z_1$  phân cực thuận,  $Z_1$  dẫn bình thường  
 $Z_2$  phân cực nghịch,  $Z_2$  tắt  
 $\Rightarrow V_0 = V_i$
- $V_i > V_{Z2} > 0$  :  $Z_1$  phân cực thuận,  $Z_1$  dẫn bình thường  
 $Z_2$  phân cực nghịch,  $Z_2$  dẫn ổn áp  
 $\Rightarrow V_0 = V_{Z2}$

❖ **Đặc tuyến hàm truyền đạt:**



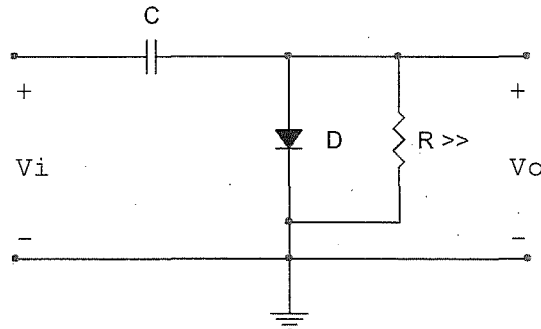
### MẠCH GHIM (MẠCH KẸP)

ĐN: Mạch ghim là mạch cố định đỉnh trên hay đỉnh dưới của tín hiệu ở một giá trị điện áp nhất định.

Mạch ghim không làm thay đổi biên độ đỉnh đỉnh của tín hiệu .

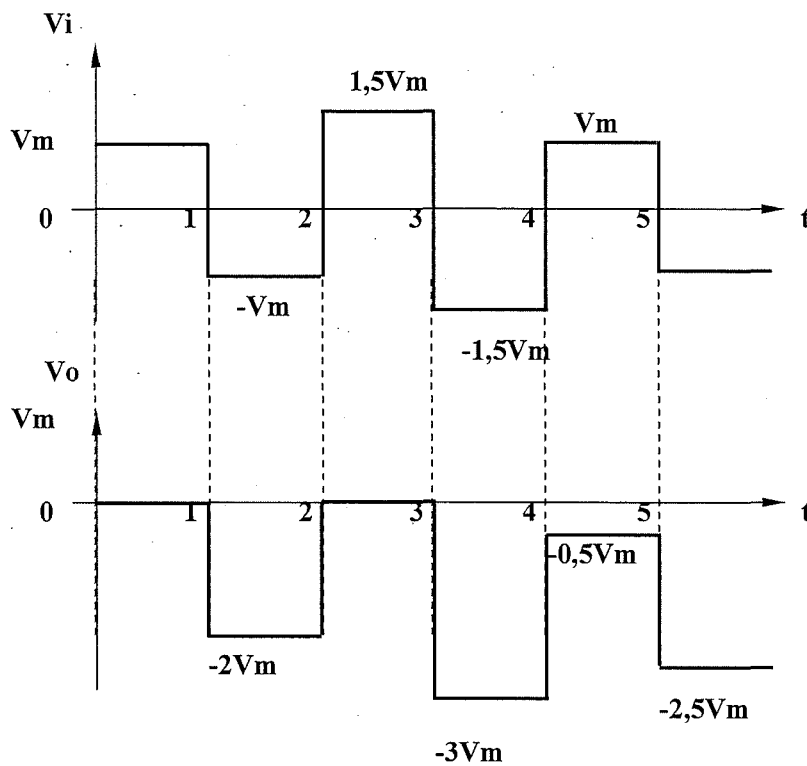
Có hai loại mạch ghim đỉnh trên và mạch ghim đỉnh dưới.

#### Mạch ghim đỉnh trên ở 0V



Hình 2.6. Mạch ghim đỉnh trên ở 0V

VD:



$=V_m, D \text{ dẫn} \Rightarrow V_o = 0$

•  $t = (0 \div 1): V_i$

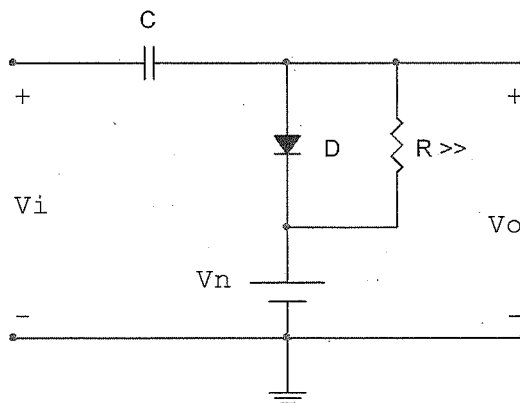
Đồng thời tụ C nạp nhanh  $V_c = V_m$

- $t = (1 \div 2)$ :  $V_i = -V_m$ , D tắt  $\Rightarrow$  tụ C không xả được  $\Rightarrow V_c = V_m = \text{const} \Rightarrow$  điện áp ra:  $V_o = V_i - V_c = -V_m - V_m = -2V_m$
- $t = (2 \div 3)$ :  $V_i$  tăng biên độ,  $V_i = 1,5V_m$ , D dẫn  $\Rightarrow V_o = 0$  (Mạch vẫn còn khả năng ghim áp khi  $V_i$  tăng)  
Đồng thời tụ C nạp tiếp  $V_c = 1,5V_m$
- $t = (3 \div 4)$ :  $V_i = -1,5 V_m$ , D tắt  $\Rightarrow$  tụ C không xả được  $\Rightarrow V_c = 1,5V_m = \text{const} \Rightarrow$  điện áp ra:  $V_o = V_i - V_c = -1,5V_m - 1,5V_m = -3V_m$
- $t = (4 \div 5)$ :  $V_i$  giảm biên độ,  $V_i = V_m$ , D tắt  $\Rightarrow$  tụ C không xả được  $\Rightarrow V_c = 1,5V_m = \text{const} \Rightarrow$  điện áp ra:  $V_o = V_i - V_c = V_m - 1,5V_m = -0,5V_m$   
(Mạch mất khả năng ghim áp khi  $V_i$  giảm).

\* Như vậy mạch sẽ mất khả năng ghim đỉnh trên ở mức 0V khi biên độ điện áp giảm. Để khắc phục tình trạng này ta mắc điện trở R rất lớn song song với Diode, tụ C sẽ xả bớt điện tích qua R sau một vài chu kỳ mạch sẽ hồi phục lại khả năng ghim

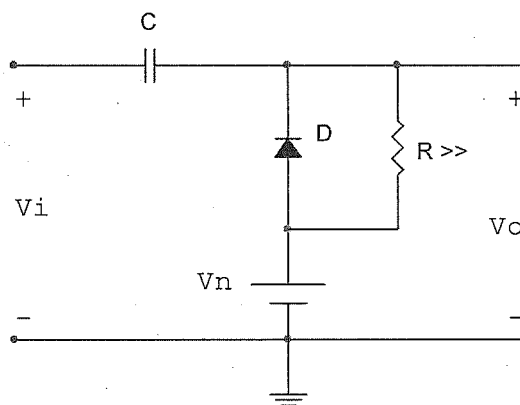
\* Chú ý : Khi phân tích mạch ghim ta bắt đầu tại bán kỳ mà tụ C nạp điện.

#### Mạch ghim đỉnh trên ở VN



Hình 2.5. Mạch ghim đỉnh trên ở VN

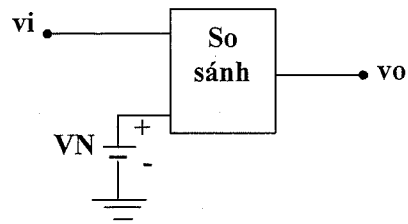
#### Mạch ghim đỉnh dưới ở VN



Hình 2.5. Mạch ghim đỉnh dưới ở VN

## MẠCH SO SÁNH:

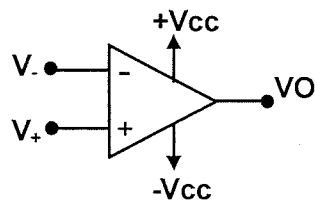
- Nguyên lý:



$$\begin{cases} vi < VN : vo = -V_{cc}. \\ vi > VN : vo = V_{cc}. \end{cases}$$

### Mạch so sánh dùng khuếch đại thuật toán (op\_amp):

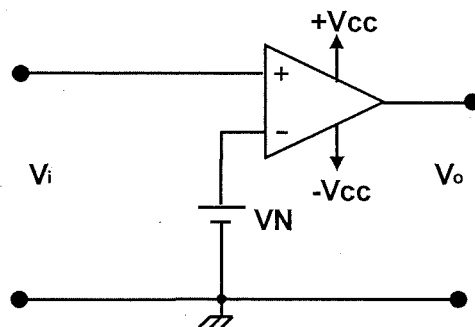
Khuếch đại thuật toán ở chế độ xung tồn tại chủ yếu ở trạng thái bão hòa.



$$\begin{aligned} V_+ > V_- & : V_o = +V_{c_c} : \text{bão hòa dương.} \\ V_+ < V_- & : V_o = -V_{c_c} : \text{bão hòa âm.} \end{aligned}$$

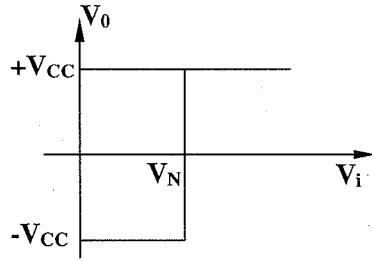
### Mạch so sánh không đảo

Điện áp  $V_i$  vào ngõ không đảo  $V_+$   
 $V_R$  vào ngõ đảo  $V_-$ .



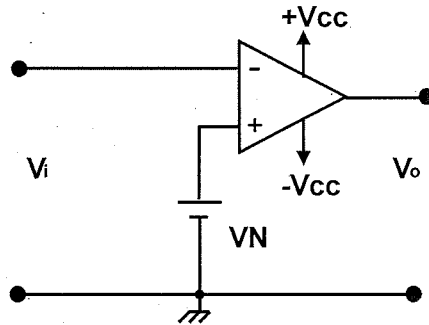


❖ Đặc tuyến hàm truyền đạt:

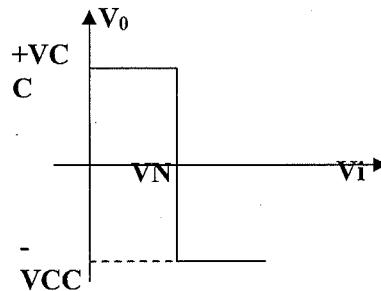


- $V_i < V_N$  (hay  $V_+ < V_-$ ) thì  $V_0 = -V_{CC}$
- $V_i > V_N$  (hay  $V_+ > V_-$ ) thì  $V_0 = +V_{CC}$ .

**Mạch so sánh không đảo Điện áp**  
 $V_i$  vào ngõ đảo  $V_-$ ,  $V_R$  vào ngõ không đảo  $V_+$



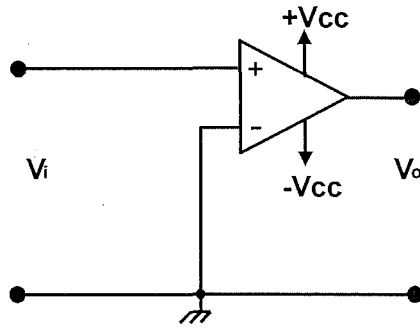
❖ Đặc tuyến hàm truyền đạt:



- $V_i < V_N$  (hay  $V_+ > V_-$ ) thì  $V_0 = +V_{CC}$
- $V_i > V_N$  (hay  $V_+ < V_-$ ) thì  $V_0 = -V_{CC}$

\* Chú ý: Nếu op-amp được cung cấp nguồn đơn thì khi bão hoà âm  $V_0 = 0$

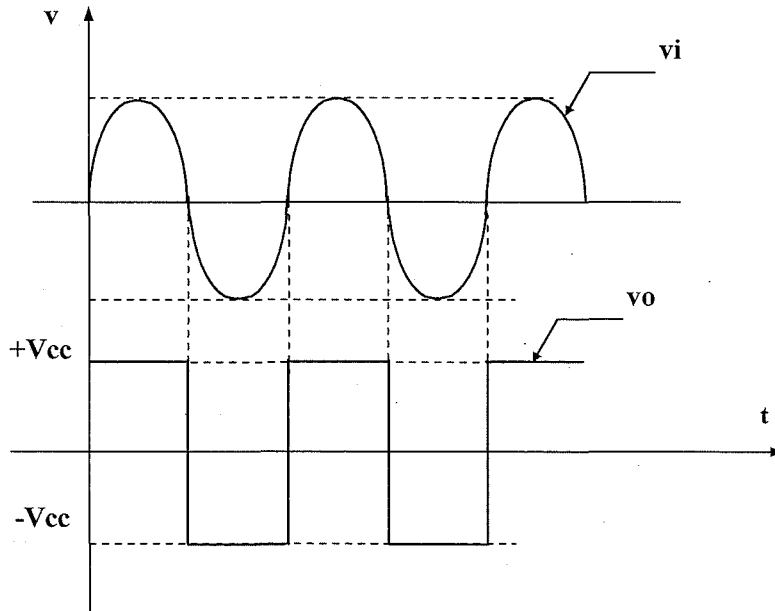
**Ví dụ:**



$$\begin{cases} v_+ = v_i \\ v_- = 0 \end{cases}$$

**Khi  $v_i > 0 \Rightarrow V_+ > V_- \Rightarrow v_o = +V_{cc}$ .**

**Khi  $v_i < 0 \Rightarrow V_+ < V_- \Rightarrow v_o = -V_{cc}$ .**



# CHƯƠNG 3

## MẠCH HẠN CHẾ BIÊN ĐỘ VÀ GHIM

### KHÁI NIỆM CHUNG

Các mạch tạo xung cơ bản nhất là các mạch tạo xung vuông được gọi chung là mạch dao động đa hài. Có ba loại mạch dao động đa hài là:

- Dao động đa hài lưỡng ổn (bistable – multivibrator) ( còn gọi là mạch Flip-Flop, mạch lật hay bấp bênh): mạch có hai trạng thái và hai trạng thái đều ổn định.
- Dao động đa hài đơn ổn ( Monostable Multivibrator) (còn gọi là mạch định thì): mạch có hai trạng thái, trong đó một trạng thái ổn định và một trạng thái không ổn định gọi là trạng thái tạo xung
- Dao động đa hài phi ổn (astable Multivibrator): mạch có hai trạng thái và cả hai trạng thái đều không ổn định còn gọi là mạch tự dao động.

Mạch dao động đa hài dùng BJT dựa vào sự nạp điện và sự xả điện của tụ điện kết hợp với đặc tính chuyển mạch của Transistor.

Ngoài ra mạch dao động đa hài được tạo ra từ các linh kiện như op-amp, IC555, các công logic, ....

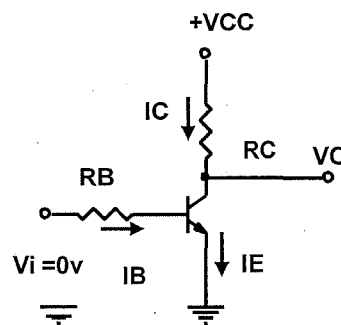
### TRẠNG THÁI NGẮT (TẮT) DẪN CỦA TRANSISTOR.

Transistor có 3 trạng thái:

- Trạng thái ngắt (tắt)
- Trạng thái dẫn khuếch đại
- Trạng thái dẫn bão hòa

Trong kỹ thuật xung transistor dùng để tạo xung vuông nên chỉ hoạt động ở hai trạng thái tắt và bão hòa. Lúc này transistor làm việc như một khóa điện tử để đóng và ngắt mạch điện.

Trạng thái đóng hay ngắt của mạch transistor tùy thuộc vào mức điện áp phân cực cho cực B của nó



Hình 3.1: Transistor ngắt

### Trạng thái ngắt (tắt)

Trong hình 3.1 transistor có điện áp

$$V_i = 0^V \Rightarrow V_B = 0^V,$$

transistor không được phân cực nên ngưng dẫn.

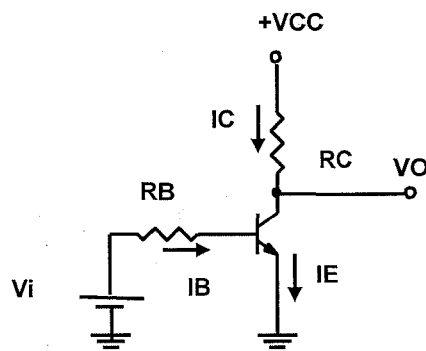
$$\Rightarrow I_B = 0 \text{ và } I_C = 0.$$

Điện áp ngõ ra ở cực C của transistor là:

$$V_0 = V_C = V_{CC} - I_C \cdot R_C$$

$$\Rightarrow V_0 = V_{CC}$$

Như vậy ngõ vào  $V_i$  có mức thấp, ngõ ra  $V_0$  có mức cao.



Hình 3.2: Transistor bão hòa

### Trạng thái dẫn bão hòa

Để transistor chuyển từ trạng thái ngắt như hình 3.1 sang trạng thái bão hòa thì ngõ vào phải được cấp một

điện áp  $V_i$  đủ lớn sao cho  $V_B$  lớn hơn một mức ngưỡng để transistor được phân cực bão hòa. Điện áp này được gọi là  $V_{BEsat}$  có trị số tùy thuộc chất bán dẫn chế tạo transistor.

Ta có  $V_{BEsat} = 0,7 \text{ V} \div 0,8 \text{ V}$  (Transistor chất silicium)  
 $V_{BEsat} = 0,3 \text{ V}$  (Transistor chất Germanium)

Trong mạch điện hình 3.2, điện trở  $R_C$  được coi là điện trở tải để xác định dòng điện  $I_C$  qua transistor. Khi transistor chạy ở trạng thái bão hòa thì điện cực C có điện áp ra:

$$V_0 = V_C = V_{CEsat} \approx 0,1 \text{ V} \div 0,2 \text{ V}$$

Như vậy ngõ vào  $V_i$  có mức thấp, ngõ ra  $V_0$  có mức cao.

Dòng điện  $I_C$  được tính theo công thức:

$$I_C = \frac{V_{CC} - V_{CEsat}}{R_C}$$

Khi có dòng điện tải  $I_C$  phải tính dòng điện cần thiết cung cấp cho cực B để chọn trị số điện trở  $R_B$  thích hợp. Thông thường ta có:

$$I_B = \frac{I_C}{\beta} \quad (\beta : \text{độ khuếch đại dòng điện})$$

Trường hợp cần cho transistor chạy bão hòa vững (bão hòa sâu) thì chọn

$$I_B \geq \frac{I_C}{\beta} \quad (\text{điều kiện bão hòa})$$

hay  $I_B = k \cdot \frac{I_C}{\beta}$  với  $k$  là hệ số bão hòa sâu ( $k \approx 2 \div 5$ )

Điện trở  $R_B$  được chọn theo công thức:

$$R_B = \frac{V_i - V_{BEsat}}{I_B}$$

Ví dụ: Mạch điện hình 3.2 có các thông số sau :  $+V_{CC} = 12 \text{ V}$ ,  $R_C = 1,2 \text{ k}\Omega$ , transistor chất Si và có  $\beta = 100$ , điện áp vào  $V_i = 1,5 \text{ V}$ . Tìm  $R_B$  để transistor hoạt động ở trạng thái bão hòa.

Trước hết phải tính dòng điện tải:

$$I_C = \frac{V_{CC} - V_{CEsat}}{R_C} = \frac{12 \text{ V} - 0,2 \text{ V}}{1,2 \text{ k}} \approx 10 \text{ mA}$$

Chọn hệ số bão hòa sâu  $K=3$  ta có:

$$I_B = \frac{I_C}{\beta} = 3 \cdot \frac{10 \text{ mA}}{100} = 0,3 \text{ mA}$$

Điện trở  $R_B$  được chọn có giá trị

$$R_B = \frac{V_i - V_{BEsat}}{I_B} = \frac{1,5 \text{ V} - 0,8 \text{ V}}{0,3 \text{ mA}} = 2,33 \text{ k}$$

Chọn điện trở theo tiêu chuẩn là  $R_B = 2,4 \text{ k}$

## MẠCH DAO ĐỘNG ĐA HÀI LƯỢNG ỔN DÙNG BJT.

### 3.3.1 Mạch đảo.

Một Transistor có thể làm chức năng của mạch đảo như hình 5.1.

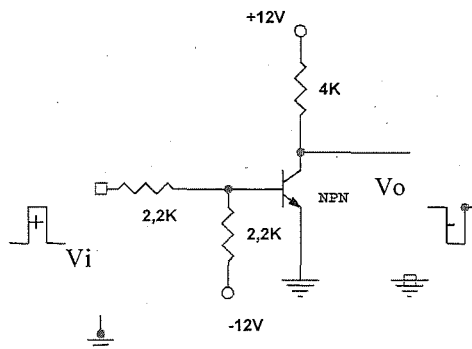
- Khi  $V_i$  ở mức điện áp cao thì Transistor chạy bão hòa và dòng  $I_c$  qua  $R_c$  tạo sụt áp  
 $\Rightarrow V_o \approx 0,2V$  ( $V_{CESat}$ ) ứng với mức điện áp thấp.
  - Khi  $V_i$  ở mức điện áp thấp thì Transistor bị phân cực ngược ở ngõ vào nên ngưng dẫn, dòng  $I_c = 0$  nên không giảm áp qua  $R_C$   
 $\Rightarrow V_o \approx V_{CC}$  ứng với mức điện áp cao ra.
- Như vậy, điện áp ra  $V_o$  và điện áp vào  $V_i$  ngược pha nhau

### Mạch lưỡng ổn (flip-flop) cơ bản.

Mạch dao động đa hài lưỡng ổn được tạo ra bằng cách ghép hai mạch đảo sao cho điện áp ra của mạch đảo này là ngõ vào của mạch đảo kia.

#### a. Sơ đồ

Mạch lưỡng ổn được Trong sơ đồ dùng 2 nguồn  $+V_{CC}$  để cấp dòng  $I_B$  và  $I_C$  cho Transistor dẫn bão hòa và nguồn  $-V_{BB}$  để phân cực ngược cho cực B của Transistor ngưng dẫn.



Hình 3.3: Transistor làm mạch đảo

#### b. Nguyên lý hoạt động.

Mạch có 2 trạng thái, trong mỗi trạng thái một transistor tắt và một transistor bão hòa. Giả thiết có mạch Flip-Flop đối xứng ( $T_1$  và  $T_2$  cùng tên, các điện trở phân cực cho hai Transistor cùng trị số) nhưng hai transistor không thể cân bằng một cách tuyệt đối nên sẽ có một Transistor dẫn mạnh hơn và một Transistor dẫn yếu hơn.

Giả thiết Transistor  $T_1$  dẫn mạnh hơn  $T_2$  nên dòng điện  $I_{C1}$  lớn hơn qua  $R_{C1}$  làm điện áp  $V_{C1}$  giảm. Điện áp  $V_{C1}$  qua điện trở  $R_2$  phân cực cho  $T_2$  sẽ làm  $V_{B2}$  giảm và điều này làm cho  $T_2$  chạy yếu hơn. Khi  $T_2$  chạy yếu thì dòng điện  $I_{C2}$  nhỏ hơn qua  $R_{C2}$  làm điện áp  $V_{C2}$  tăng lên. Điện áp  $V_{C2}$  qua điện trở  $R_1$  phân cực cho  $T_1$  sẽ làm  $V_{B1}$  tăng làm  $T_1$  chạy mạnh hơn nữa và cuối cùng  $T_1$  sẽ tiến đến trạng thái bão hòa  $T_2$  tiến đến ngưng dẫn. Nếu không có một tác động nào khác thì mạch điện sẽ ở trạng thái này. Đây là một trạng thái của mạch Flip-Flop.

Ngược lại, nếu giả thiết Transistor  $T_2$  dẫn nhanh hơn  $T_1$  và lý luận tương tự thì cuối cùng sẽ có  $T_2$  tiến đến trạng thái bão hòa và  $T_1$  tiến đến ngưng dẫn và mạch điện cũng ở mãi trạng thái này nếu không có một tác động nào khác. Đây là trạng thái thứ hai của Flip-Flop.

Mạch Flip-Flop sẽ ở một trong hai trạng thái trên nên được gọi là mạch lưỡng ổn.

Tuy nhiên, phải chọn các điện trở và nguồn điện thích hợp mới đạt được nguyên lý trên.

#### c. Phân tích mạch:

Để thấy rõ hơn nguyên lý của mạch Flip-Flop ta có thể phân tích dòng điện và điện áp trong mạch Flip-Flop tiêu biểu như trong mạch điện hình 3.5 với các trị số điện trở và nguồn cụ thể.

Theo giả thiết, khi  $T_1$  bão hòa ta có:

$$V_{C1} = V_{CCSat} \approx 0,2V$$

$$V_{B1} = V_{BEsat} \approx 0,8^V$$

Suy ra dòng điện  $I_{C1}$  và  $I_{B1}$  theo công thức:

$$I_{C1} = \frac{V_{CC} - V_{CEsat}}{R_{C1}}$$

$$I_{C1} = \frac{12^V - 0,2^V}{1,8^K} \approx 6,5\text{mA}$$

$$I_{B1} = \frac{V_{CC} - V_{BEsat}}{R_{C2} + R_{B1}} - \frac{V_{BEsat} + V_B}{R_{B1}} = \frac{12^V - 0,8^V}{1,8^K + 18^K} - \frac{0,8^V + 6^V}{47^K} = 0,41\text{mA}$$

Ở trạng thái bão hòa Transistor thường có  $\beta$  nhỏ, chọn  $\beta = 50$ . Ta có thể nghiệm lại điều kiện bão hòa của  $T_1$  như sau:

$$\text{Thông thường } I_B = \frac{I_C}{\beta} = \frac{6,5^{\text{mA}}}{50} = 0,13\text{mA}$$

$$\text{Mạch điện có: } I_{B1} = 0,41^{\text{mA}} \quad (I_{B1} > I_B)$$

Như vậy:  $T_1$  đủ điều kiện để bão hòa vì  $I_{B1} > \frac{I_{C1}}{\beta}$

Xét  $T_2$  lúc đó ở trạng thái ngưng ta có:

$$V_{C2} = V_{CC} - I_{C2} R_{C2} = V_{CC} - (I_{B1} + I_R) R_{C2}$$

$$V_{C2} = V_{CC} - \left( \frac{V_{CC} - V_{BEsat}}{R_{C2} + R_{B1}} \right) R_{C2}$$

$$= 12^V - \left( \frac{12^V - 0,8^V}{1,8^K + 18^K} \right) 1,8^K \approx 11^V$$

$$V_{B2} = (V_{C1} + V_{BB}) \frac{R_{B2}}{R_2 + R_{B2}} - V_{BB}$$

$$= (0,2^V + 6^V) \frac{47^K}{18^K + 47^K} - 6^V \approx -1,5^V$$

$T_2$  là loại transistor NPN có  $V_{B2} = -1,5^V$  ( $V_{B2} < 0^V$ ) nên  $T_2$  phải ngưng dẫn.

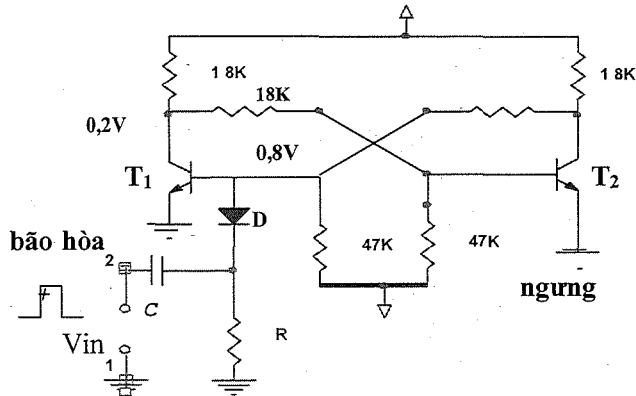
Nếu ở trạng thái ngưng lại thì hai transistor sẽ có dòng điện và điện áp ở các chân cực ngưng lại với phân tích trên.

Điện áp nguồn âm ( $-V_{BB}$ ) có tác dụng phân cực cho  $T_2$  để  $T_2$  ổn định ở trạng thái ngưng tránh tác động của nhiễu có thể làm cho  $T_2$  đổi trạng thái. Trường hợp không cần thiết chống nhiễu thì có thể không dùng nguồn  $-V_{BB}$ , lúc đó hai điện trở  $R_{B1} - R_{B2}$  được nối mass hay có thể không cần dùng cũng được.

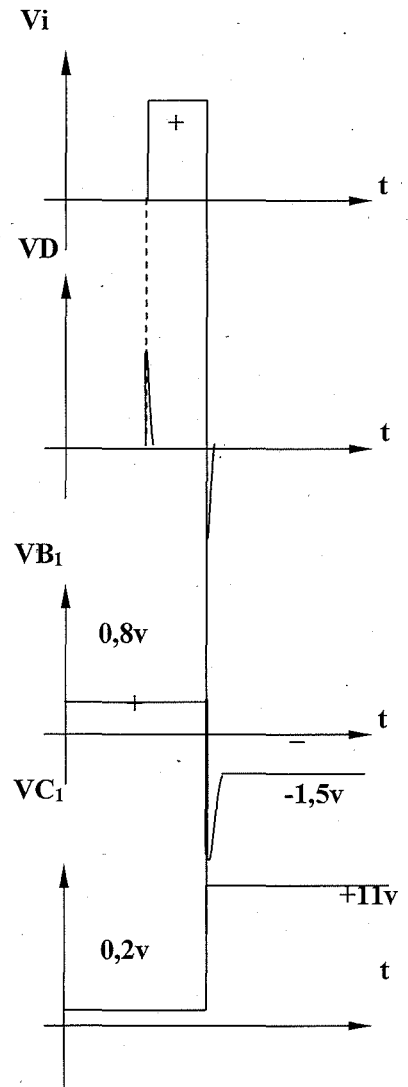
### Các phương pháp kích đổi trạng thái của flip-flop.

Trường hợp  $T_1$  đang bão hòa,  $T_2$  đang ngưng dẫn như mạch hình 3.6, muốn đổi trạng thái của Flip-Flop thì ta có thể cho một xung âm vào cực  $B_1$  (hoặc là cho một xung dương vào cực  $B_2$ ). Muốn đổi trở lại trạng thái cũ thì phải cho một xung dương vào cực  $B_1$  (hoặc là cho một xung âm vào cực  $B_2$ ). Để giản đơn người ta thường dùng một loại xung.

a. Mạch kích một bên



Hình 3.5. Mạch kích một bên.



Hình 3.6. Dạng sóng ở các chân.

Sơ đồ hình 3.6 là mạch Flip- Flop với mạch kích một bên. Xung kích điều khiển là xung vuông qua mạch vi phân RC để đổi từ xung vuông ra hai xung nhọn (xung nhọn dương ứng với cạnh lên và xung nhọn âm ứng với cạnh xuống). Diode D có tác dụng loại bỏ xung nhọn dương và chỉ đưa xung nhọn âm vào cực B<sub>1</sub> để đổi trạng thái T<sub>1</sub> từ bão hòa sang ngưng dẫn.

Giả thiết mạch có trạng thái như hình 3.6 là T<sub>1</sub> đang bão hòa và T<sub>2</sub> đang ngưng dẫn. Khi ngõ vào nhận xung vuông (Vin) qua mạch vi phân RC tạo điện áp Vi trên điện trở R là hai xung nhọn. Khi có xung nhọn dương thì diode D bị phân cực ngược nên ngưng dẫn và mạch Flip –Flop vẫn giữ nguyên trạng thái đang có. Khi có xung nhọn âm thì diode D được phân cực thuận coi như nối tắt làm điện áp VB<sub>1</sub> giảm xuống dưới 0V. Lúc đó T<sub>1</sub> ngưng dẫn nên Ib<sub>1</sub>= 0, Ic<sub>1</sub>= 0 nên Vc<sub>1</sub> tăng cao sẽ tạo phân cực đủ mạnh

cho cực B<sub>2</sub> và T<sub>2</sub> chạy bão hòa. Khi T<sub>2</sub> đã bão hòa thì V<sub>C2</sub> ≈ 0,2 V nên T<sub>1</sub> không được phân cực sẽ tiếp tục ngưng dẫn mặc dầu đã hết xung âm.

Như vậy, mạch Flip-Flop đã chuyển từ trạng thái T<sub>1</sub> bão hòa, T<sub>2</sub> ngưng sang trạng thái T<sub>1</sub> ngưng - T<sub>2</sub> bão hòa. Khi mạch đã ổn định ở trạng thái này thì mạch sẽ không bị tác động đổi trạng thái bởi xung kích vào cực B<sub>1</sub> nữa. Bây giờ muốn đổi trạng thái của mạch trở lại trạng thái cũ thì phải cho xung vuông tiếp theo qua mạch vi phân và diod D vào cực B<sub>2</sub> (vì T<sub>2</sub> đang ở trạng thái bão hòa)

### b. Mạch kích đếm:

Đối với mạch kích một bên thì mạch Flip-Flop phải được kích lần lượt, luân phiên vào cực B<sub>1</sub> và B<sub>2</sub> thông qua hai mạch vi phân và hai Diod. Để đổi trạng thái mạch Flip-Flop bằng một thứ xung kích vào một ngõ chúng ta có thể dùng mạch kích đếm. Mạch điện hình 3.8 là sơ đồ mạch Flip-Flop có ngõ kích đếm nhận xung kích là xung vuông. Theo sơ đồ này, mạch đang ở trạng thái T<sub>1</sub> bão hòa, T<sub>2</sub> ngưng dẫn. Hai điện trở 10k thêm vào mạch ra hai điểm A và B và hai điểm này có điện áp gần giống như điện áp của hai cực C<sub>1</sub> và C<sub>2</sub>.

Ta có: V<sub>A</sub> ≈ V<sub>C1</sub> = 0,2 V

( T<sub>1</sub> đang bão hòa)

V<sub>B</sub> ≈ V<sub>C2</sub> = 11V

( T<sub>2</sub> đang ngưng dẫn)

Khi có xung vuông ở ngõ vào ( V<sub>in</sub> ) thì qua hai tụ C<sub>1</sub> - C<sub>2</sub> sẽ có hai xung nhọn dương ứng với cạnh lên xung của vuông và có 2 xung nhọn âm ứng với cạnh xuống của xung vuông tại điểm A và B. Thời điểm có xung nhọn dương cả hai diode D<sub>1</sub> - D<sub>2</sub> đều bị phân cực ngược nên không có tác dụng với mạch Flip-Flop. Khi có xung nhọn âm tại hai điểm A và B thì tại hai điểm này sẽ có hai mức biên đổi khác nhau.

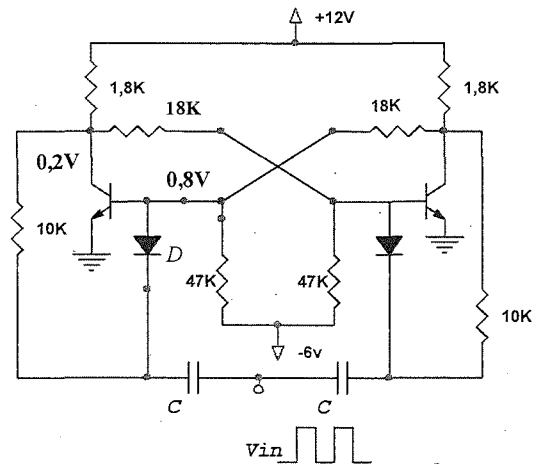
Do V<sub>A</sub> ≈ 0,2 V nên khi có xung nhọn âm thì xung âm sẽ làm giảm điện áp V<sub>A</sub> và diod V<sub>1</sub> được phân cực thuận. Điều này sẽ làm đổi trạng thái T<sub>1</sub> từ bão hòa sang ngưng dẫn và đổi trạng thái T<sub>2</sub> từ ngưng dẫn sang bão hòa. Lúc đó do V<sub>B</sub> = 11V rất cao so với xung âm nên khi có xung nhọn âm thì điện áp V<sub>B</sub> vẫn ở mức dương cao nên D<sub>2</sub> vẫn bị phân cực ngược và xung âm không có tác dụng với T<sub>2</sub>.

Khi có xung vuông thứ hai đến ngõ vào thì lần này xung nhọn âm chỉ có tác dụng đổi với T<sub>2</sub> là transistor đang bão hòa nên mạch Flip-Flop lại trở về trạng thái cũ.

### Các điểm cần lưu ý trong thiết kế .

- Mạch vi phân ở ngõ vào được chọn trị số cao cho thỏa các yêu cầu sau:
  - Xung âm phải có biên độ đủ cao và độ rộng đủ lớn để đủ kích đổi trạng thái của transistor đang bão hòa sang ngưng .
  - Nếu hằng số thời gian  $\tau = RC$  sẽ làm giới hạn tần số xung kích ( theo điều kiện của mạch vi phân trong chương.
  - Nếu hằng số thời gian  $\tau = RC$  nhỏ hơn sẽ làm giảm độ rộng xung và có thể không đủ đổi trạng thái của transistor

- Khi mạch Flip-Flop làm việc với các tín hiệu xung kích tần số cao nên chọn loại transistor có kết cấu Epiplanar để có đáp ứng nhanh.



Hình 3.8: Mạch F-F có ngõ kích đếm.



- c. Để mạch chuyển trạng thái tốt, tốc độ làm việc nhanh nên chọn mức điện áp nguồn thấp điều này còn phụ thuộc vào yêu cầu của tải nếu tải là  $R_c$ .
- d. Trong các mạch đơn giản người ta có thể không cần dùng âm  $-V_{BB}$ . Tuy nhiên, khi không có nguồn âm thì tính ngưng dẫn của transistor không tốt và khả năng chống nhiễu của mạch kém

### DAO ĐỘNG ĐA HÀI ĐƠN ỔN.

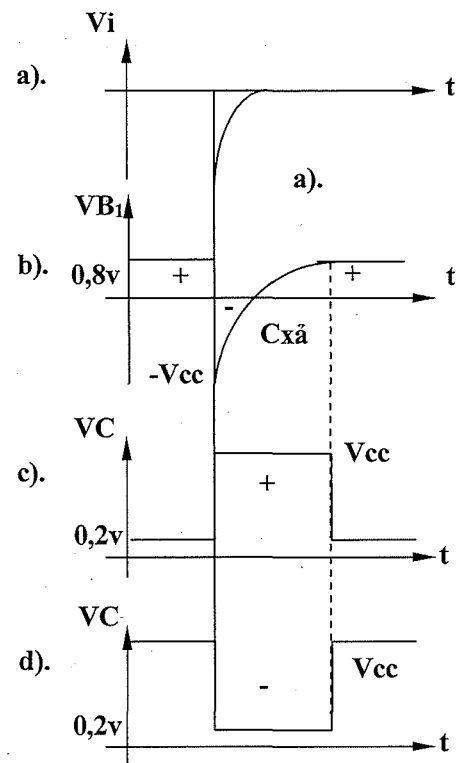
#### Giới thiệu.

Mạch dao động đa hài đơn ổn cũng có hai trạng thái ( $T_1$  bão hòa  $T_2$  ngưng hay  $T_1$  ngưng  $T_2$  bão hòa) nhưng trong hai trạng thái đó có một trạng thái ổn định và một trạng thái không ổn định gọi là trạng thái tạo xung.

Bình thường khi mạch đơn ổn được cấp nguồn sẽ ở trạng thái ổn định và ở mãi trạng thái này nếu không có tác động từ bên ngoài vào. Khi ngõ vào nhận được một xung kích thì mạch đơn ổn sẽ đổi trạng thái tạo xung ở ngõ ra và độ rộng xung ra sẽ tùy thuộc các thông số RC thiết kế trong mạch. Sau thời gian có xung ra ở mạch đơn ổn sẽ trở về trạng thái ổn định ban đầu.

Mạch dao động đa hài đơn ổn còn được gọi là mạch định thì vì thời gian có xung ra có thể định trước nhờ các thông số trong mạch. Mạch đơn ổn rất thông dụng trong lĩnh vực điều khiển tự động trong các thiết bị điện tử và điện tử công nghiệp.

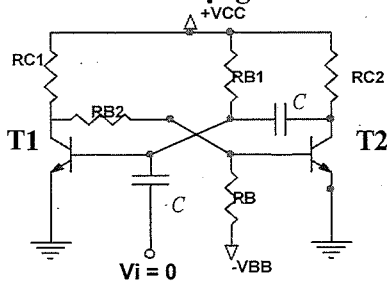
Mạch đơn ổn có thể thực hiện bằng nhiều cách: dùng transistor, op-amp vi mạch định thì hay các cổng logic. Phần này chỉ giới thiệu và phân tích mạch đơn ổn dùng transistor, các mạch dạng khác được giới thiệu trong chương sau.



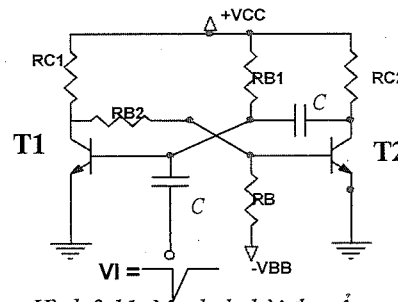
Hình 3.12. Dạng sóng vào và ra của mạch đơn ổn.

#### Mạch đơn ổn cơ bản.

##### a. Sơ đồ ở hai trạng thái.



Hình 3.10. Mạch đa hài đơn ổn



Hình 3.11. Mạch đa hài đơn ổn

##### b. Nguyên lý.

###### \* Trạng thái ổn định của mạch đơn ổn.(hình 3.10).

Khi mở điện, tụ C tức thời nạp điện qua điện trở  $R_{c2}$  tạo dòng điện đủ lớn cấp cho cực  $B_1$  nên  $T_1$  sẽ chạy ở trạng thái bão hòa. Lúc đó, dòng  $I_{C1}$  qua  $R_{C1}$  đủ lớn để tạo

sụt áp và  $V_{C1} = V_{CEsat} \approx 0,2V$ . Cầu phân áp  $R_{B2}$  và  $R_B$  sẽ tạo ra điện áp phân cực cho  $T_2$  ngưng dẫn vì  $V_{B2} < 0V$ . sau khi tụ nạp đầy sẽ có như hình vẽ 5.9. Điện áp nạp trên tụ có giá trị khoảng :  $V_C = V_{CC} - V_{BEsat} \approx V_{CC}$ .

Khi tụ nạp đầy thì dòng nạp bên tụ bằng 0 nhưng tụ  $T_1$  vẫn chạy ở trạng thái bão hòa vì vẫn còn dòng  $I_{B1}$  qua  $R_{B1}$  cấp phân cực cho cực  $B_1$ .

Hai Transistor sẽ chạy ổn định ở trạng thái này nếu không có tác động gì từ bên ngoài.

**\* Trạng thái tạo xung của mạch đơn ổn (hình 3.11).**

Khi ngõ vào  $V_i$  nhận xung kích âm qua tụ  $C_1$  sẽ làm điện áp  $V_{B1}$  giảm và  $T_1$  đang chạy bão hòa chuyển sang trạng thái ngưng dẫn. Lúc đó  $I_{C1} = 0$  điện áp vào  $V_{C1}$  tăng cao qua cầu phân áp  $R_{B2} - R_B$  sẽ phân cực cho  $T_2$  chạy bão hòa. Khi  $T_2$  chạy bão hòa  $V_{C2} = V_{BEsat} \approx 0,2V$  điều này làm cho tụ  $C$  có chân mang điện áp dương coi như nổi mass và chân kia có điện áp âm so với mass nên điện áp âm này sẽ phân cực ngược cho cực  $B_1$  làm  $T_1$  tiếp tục ngưng mặc dầu đã hết xung kích. Lúc đó tụ  $C$  xả điện qua điện trở  $R_{B1}$  và transistor  $T_2$  từ  $C$  xuống  $E$ . Trong thời gian này  $T_1$  ngưng dẫn  $T_2$  bão hòa nên điện áp ở các chân  $C$  và  $B$  của transistor đổi ngược lại chính là xung điện ở ngõ ra. Sau khi tụ xả xong làm mất điện áp âm đặt vào cực  $B_1$  và  $T_1$  sẽ hết trạng thái ngưng dẫn và chuyển sang trạng thái bão hòa như lúc ban đầu. Khi  $T_1$  trở lại trạng thái bão hòa thì  $V_{C1} = V_{CEsat} \approx 0,2 V$  nên  $T_2$  mất phân cực sẽ ngưng dẫn như lúc ban đầu.

Thời gian tạo xung của mạch đơn ổn chính là thời gian xả điện của tụ  $C$  qua  $R_{B1}$  Sau thời gian này mạch tụ trở lại trạng thái ban đầu là trạng thái ổn định.

**c. Dạng sóng ở các chân.**

Hình 3.12 cho thấy dạng sóng ở các chân của mạch đơn ổn Trong đó hình 3.12 a là áp ngõ vào  $V_i$ , trước thời điểm có xung kích là trạng thái ổn định. Khi có xung nhọn âm thì mạch đơn ổn bắt đầu chuyển sang trạng thái tạo xung.

Hình 3.12 b là dạng điện áp  $V_{B1}$ , khi có xung kích là  $T_1$  ngưng, tụ  $C$  xả điện áp âm nên  $V_{B1}$  có điện áp âm  $\approx -V_{CC}$  và tụ  $C$  xả điện qua  $R_{B1}$  làm điện áp âm giảm dần theo hàm số mũ. Thời gian xả của tụ  $C$  chính là thời gian tạo xung ở ngõ ra.

Ở trạng thái ổn định  $V_{C1} = 0,2V$  (bão hòa), ở trạng thái tạo xung  $V_{C1} = V_{CC}$  (ngưng dẫn) nên  $T_1$  có xung vuông dương ra. Ngược lại  $T_2$  có xung vuông âm ra, độ rộng xung là  $T_x$ .

**d. Điều kiện và thông số kỹ thuật của mạch đơn ổn.**

Để cho mạch đơn ổn hoạt động đúng theo nguyên lý phải thỏa mãn điều kiện là  $T_1$  bão hòa với :

$$I_{C1} = \frac{V_{CC} - V_{CEsat}}{R_{C1}} \approx \frac{V_{CC}}{R_{C1}} \quad (1) \text{ (với } V_{CEsat} \approx 0,2V)$$

$$I_{B1} = \frac{V_{CC} - V_{BEsat}}{R_{B1}} \approx \frac{V_{CC}}{R_{B1}} \quad (2) \text{ (với } V_{BEsat} \approx 0,2V)$$

Muốn cho  $T_1$  bão Hòa phải có:

$$I_{B1} > \frac{I_{C1}}{\beta_{sat}} \quad (3) \text{ (sat saturation : bão hòa)}$$

Thường chọn:  $I_{B1} = K \cdot \frac{I_{C1}}{\beta_{sat}}$

Trong đó :  $K$  là hệ số bão hòa sâu và  $K = 2 \div 5$

**\*. Cách tính độ rộng xung:**

Trong thời gian ổn định tụ C nạp điện qua  $RC_1$  với hằng số thời gian nạp là

$$\tau_{\text{nạp}} = RC_1 \cdot C$$

Điện áp nạp trên tụ tăng theo hàm số mũ bởi công thức :

$$\begin{aligned} VC(t) &= VCC(1 - e^{-\frac{t}{\tau}}) \\ &= VCC - VCC \cdot e^{-\frac{t}{\tau}} \end{aligned}$$

Điện áp trên tăng từ 0V lên VCC. Khi có xung âm vào cực  $B_1$  thì tụ C xả điện qua  $RB_1$  với hằng số thời gian xả là:

$$\tau_{\text{xả}} = RB_1 \cdot C$$

Điện áp trên tụ khi xả giảm theo hàm số mũ bởi công thức:

$$VC(t) = VCC \cdot e^{-\frac{t}{\tau}}$$

Do chân dương của tụ C coi như nối mass qua chân  $C_2$  khi  $T_2$  bão hòa nên tụ xả điện âm ( $-VCC$ ) và điện áp trên tụ tăng từ  $-VCC$  lên 0V rồi sau đó nạp tiếp tục từ 0V lên  $+VCC$ . Như vậy đường xả điện và nạp điện của tụ sẽ biến thiên như hình 3.13 được giới hạn từ  $-VCC$  lên  $+VCC$ . Đường biểu diễn điện áp trên tụ sẽ được tính theo công thức:

$$VC(t) = VCC - 2VCC \cdot e^{-\frac{t}{\tau}}$$

Khi  $VC(t) = 0^V$  là hết thời gian xả của tụ và mạch trở lại trạng thái ổn định thời gian này chính là thời gian tạo xung ở ngõ ra và còn gọi là độ rộng xung  $t_x$ .

Ta có :  $VCC = 2VCC \cdot e^{-\frac{t_x}{\tau}}$

$$e^{-\frac{t_x}{\tau}} = \frac{1}{2} \text{ hay } e^{-\frac{t_x}{\tau}} = 2.$$

Suy ra  $\frac{t_x}{\tau} = \text{Ln}2 \Rightarrow t_x = \tau \cdot \text{Ln}2.$

Thay  $\tau = RB_1 \cdot C$  và  $\text{Ln}2 = 0,69$ .

Suy ra:  $t_x = 0,69 RB_1 \cdot C$

Muốn thay đổi độ rộng xung  $t_x$  ta có thể thay đổi  $RB_1$  hay trị số của tụ C trong đó  $RB_1$  bị giới hạn bởi điều kiện nên thường người ta chỉ thay đổi tụ C.

**\* Biên độ xung ra:**

Ở trạng thái ổn định  $T_1$  bão hòa  $T_2$  ngưng :

$$VC_1 = VCE_{\text{sat}} \approx 0,2^V, VC_2 \approx VCC$$

Ở trạng thái ngưng  $T_1$  ngưng  $T_2$  bão hòa.

$$VC_1 \approx VCC \cdot \frac{R_{B2}}{R_{C1} + R_{B2}} = V_x \text{ (do mạch phân}$$

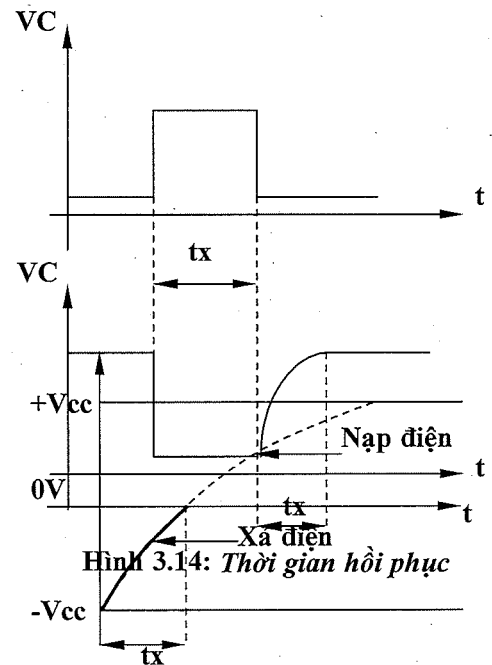
áp)

$$VC_2 = VCE_{\text{sat}} \approx 0,2^V$$

Như vậy biên độ xung vuông dương cho  $T_1$  tạo ra là:

$$V_{01} = V_x - 0,2V \approx V_x$$

Biên độ xung vuông âm do  $T_2$  tạo ra là:



Hình 3.14: Thời gian hồi phục

Hình 3.13: Đường xả và nạp điện trên tụ C

$$V_{02} = V_X - 0,2V \approx VCC$$

**\* Thời gian hồi phục :**

Theo sơ đồ mạch đơn ổn cơ bản: trạng thái ổn định là trạng thái  $T_1$  bão hòa, ngưng dẫn, trạng thái tạo xung là trạng thái  $T_1$  ngưng,  $T_2$  bão hòa. Sau khi song thời gian tạo xung  $t_x$  thì  $T_2$  sẽ trở lại trạng thái ngưng dẫn. Trong thực tế mạch chưa trở lại trạng thái ổn định ngay vì lúc đó tụ  $C$  lại nạp điện qua  $RC_2$  làm  $V_{C_2}$  tăng lên theo hàm số mũ chứ không tăng tức thời như hình vuông. Thời gian này được gọi là thời gian hồi phục  $t_h$

Hằng số thời gian nạp của tụ là:

$$\tau_{nạp} = RC_2 \cdot C$$

Tụ nạp đầy trong thời gian  $5\tau$  nhưng thường chỉ tính :

$$t_h \approx 4\tau_{nạp} = 4RC_2 \cdot C$$

**\* Thời gian phân cách:**

Do có thời gian hồi phục  $t_h$  để mạch đơn ổn trở lại trạng thái ổn định nên nếu tín hiệu xung kích ở ngõ vào là những tín hiệu liên tiếp nhau có tần số xung kích  $f_i$  chu kỳ xung kích  $T_i$  thì chu kỳ  $T_i$  phải thỏa điều kiện là:

$$T_i > t_x + t_h$$

Điều kiện này có nghĩa là khoảng cách ngắn nhất giữa hai xung kích phải lớn hơn độ rộng và thời gian hồi phục  $t_h$  thời gian  $t_x = t_h$  gọi là thời gian phân cách  $t_f$ .

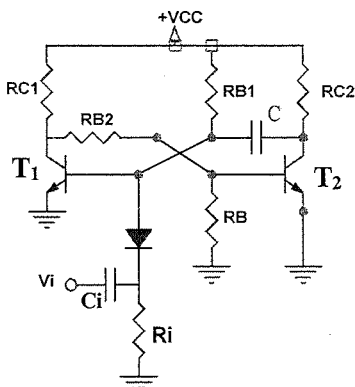
Ta có:  $T_i > t_f$  với  $t_f = t_x + t_h$

**Các mạch đơn ổn cải tiến .**

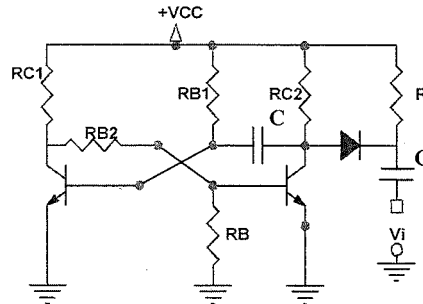
**a. Mạch đơn ổn dùng 1 nguồn.**

Trong các mạch đơn ổn người ta có thể không dùng nguồn  $-V_{BB}$  và điện trở  $R_B$  được nối mass – lúc đó  $R_B$  được chọn lại với trị số khác. Trường hợp này mạch có khả năng chống nhiễu kém...

Sơ đồ mạch đơn ổn hình 3.15, ngõ vào là mạch vi phân  $R_i-C_i$  để đổi xung vuông ra hai xung nhọn và diod  $D$  chỉ nhận xung nhọn âm đưa vào cực  $B_1$ .



Hình 3.15: Mạch đơn ổn dùng 1 nguồn



Hình 3.16. Mạch đơn ổn có xung kích vào cực  $C_2$ .

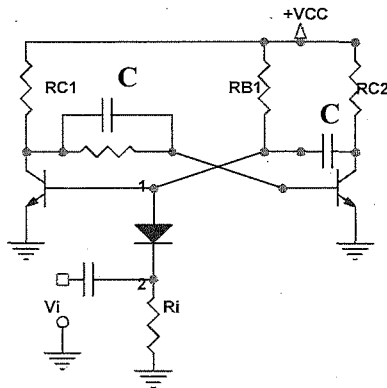
**b. Mạch đơn ổn có xung kích vào cực  $C_2$ .**

Ở trạng thái ổn định  $T_1$  bão hòa  $T_2$  ngưng, tụ nạp điện có điện áp như hình vẽ (hình 3.16). Khi có xung nhọn âm làm diod  $D$  được phân cực thuận thì tụ  $C$  có chân nạp

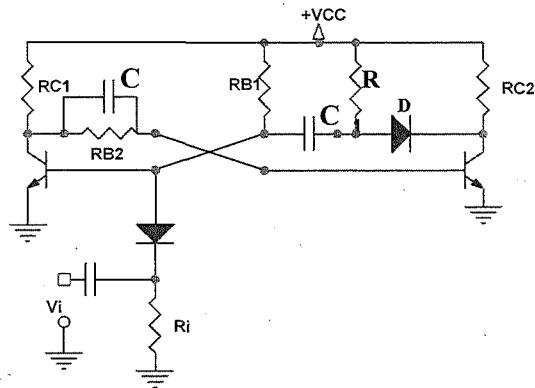
điện áp dương nối mass nên chân nạp điện áp âm sẽ làm phân cực ngược cực  $B_1$  và  $T_1$  ngưng dẫn. Lúc đó,  $VC_1$  tăng cao làm  $T_2$  tăng được phân cực bão hòa và  $VC_2 \approx 0,2V$  nên tụ  $C$  tiếp tục xả điện qua  $RB_1$  và mạch sẽ duy trì trạng thái  $T_1$  ngưng,  $T_2$  bão hòa cho đến khi tụ xả xong. Sau thời gian tạo xung tx thì mạch lại trở về trạng thái ổn định.

### c. Mạch đơn ổn dùng tụ gia tốc .

Để chuyển nhanh trạng thái của  $T_2$  từ ngưng dẫn sang bão hòa khi có xung kích âm vào cực  $B_1$ . Ta có thể dùng tụ tốc  $C_j$  ghép song song  $RB_2$ . Khi có xung kích âm vào cực  $B_1$ , transistor  $T_1$  đang bão hòa chuyển sang ngưng làm  $VC_1$  tăng ở trạng thái chuyển tiếp tụ  $C_j$  coi như nối tắt nên điện áp  $VC_1$  phân cực nhanh cho cực  $B_2$  làm  $T_2$  bão hòa nhanh. Điều này có tác dụng làm xung vuông ra ở cực  $C_2$  có cạnh xuống được thẳng đứng, sửa lại độ dốc trước xung ra (hình 3.17).



Hình 3.17 Dùng tụ gia tốc  $C_j$ .



Hình 3.18. Dùng diode cách ly D

### d. Dùng diode cách ly sửa độ dốc sau.

Trong phần các thông số của mạch đơn ổn có xét đến thời gian hồi phục th của xung ra trên cực  $C_2$  là do tụ c nạp điện qua điện trở  $R_2$  làm điện áp  $VC_2$  tăng chậm, độ dốc sau của xung dài ra.

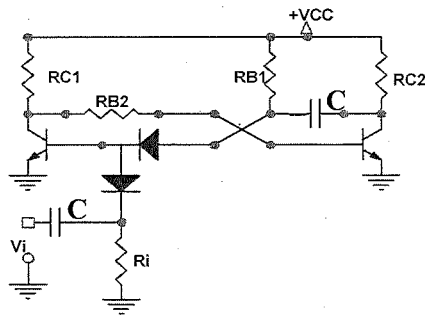
Để đảm bảo thời gian hồi phục ở ngõ ra, làm độ dốc sau được thẳng đứng người ta dùng thêm diode D và điện trở  $RD$  (hình 3.18). Khi  $T_2$  ngưng, điện áp  $VC_2$  làm phân cực ngược diode D và tụ  $C$  chỉ nạp điện qua  $RD$  nên điện áp  $VC_2$  tăng nhanh. Điều cần lưu ý trong mạch này là khi  $T_2$  bão hòa,  $VC_2$  giảm nên diod D được phân cực thuận, điện trở ở cực  $C_2$  là  $RC_2$  song song  $RD$

Thường chọn :  $RC_2 = RD = 2.RC_1$ .

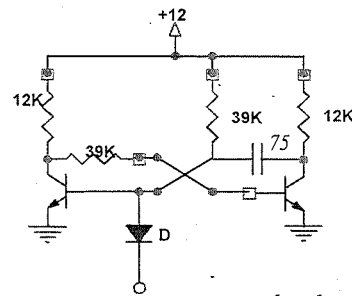
### e. Dùng diod cách ly bảo vệ mối nối $BE_1$ .

Khi mạch vi phân cho ra xung nhọn âm làm phân cực diod Di đưa đến  $T_1$  ngưng dẫn,  $T_2$  bão hòa. Lúc đó tụ  $C$  sẽ xả điện và điện áp đang nạp trên tụ đưa vào cực  $B_1$  với trị số khoảng  $VCC$ , điện áp này có thể làm hư mối nối  $BE_1$  vì điện áp đánh thủng mối nối  $BE$  ( $BVEBO$ ) thường có trị số không cao ( khoảng vài volt)

Để tránh hiện tượng trên người ta đặt thêm 1 diod giữa tụ  $C$  và cực  $B_1$  như hình vẽ. Khi tụ xả điện thì diode D sẽ chịu điện áp ngược thay cho mối nối  $BE$  mà điện áp ngược của diod thường cao nên diod không bị hư (hình 3.19).



Hình 3.19. Diode bảo vệ mỗi nối BE<sub>1</sub>



Hình 3.20. Mạch thiết kế.

### Bài tập áp dụng .

Thiết kế mạch đa hài đơn ổn theo các yêu cầu và thông số kỹ thuật sau:

-Nguồn VCC = 12V, transistor có  $\beta = 100$ , dòng tải IC = 10mA độ rộng xung tx = 2 giây

#### \* Tính điện trở tải RC<sub>1</sub> = RC<sub>2</sub> :

Khi transistor dẫn bão hòa ta có:  $V_{CEsat} \approx 0,2V$

$$\text{Suy ra: } RC_1 = RC_2 = \frac{V_{CC} - V_{CEsat}}{I_C} = \frac{12V - 0,2V}{10mA} \approx 1,2K\Omega$$

#### \* Tính điện trở phân cực RB<sub>1</sub> = RB<sub>2</sub>:

Để transistor dẫn bão hòa thường chọn hệ số bão hòa sâu là K = 3.

$$\text{Ta có: } I_B = K \frac{I_C}{\beta} = 3 \cdot \frac{10mA}{100} = 0,3mA$$

Điện trở RB được tính với mức điện áp phân cực bão hòa là V<sub>BEsat</sub> = 0,8V .

$$RB_1 = RB_2 = \frac{V_{CC} - V_{BEsat}}{I_B} = \frac{12V - 0,8V}{0,3mA} = 37 K\Omega$$

Chọn trị số tiêu chuẩn là: RB = 39K $\Omega$ .

#### \* Tính trị số tụ C :

Theo yêu cầu độ rộng xung là tx = 2 giây

$$\text{Ta có: } tx = 0,69 \cdot R_B \cdot C$$

$$\text{Suy ra: } C = \frac{tx}{0,69 \cdot R_B} = \frac{2}{0,69 \cdot 39 \cdot 10^3} \approx 75\mu F$$

Trong mạch này không yêu cầu thiết kế mạch vi phân. Muốn tính trị số R và C của mạch vi phân phải biết tần số hay độ rộng của xung vuông fi. Điều kiện của mạch vi phân là

$$R \cdot C \ll \frac{1}{2\pi f_i}$$

(Xem lại chương 1)

## MẠCH DAO ĐỘNG ĐA HÀI PHI ỔN.

### 3.5.1 Giới thiệu .

Mạch đa hài lưỡng ổn có hai trạng thái ổn định, muốn đổi trạng thái của mạch từ trạng thái ổn định này sang trạng thái ổn định khác thì phải có xung kích từ bên ngoài.

Mạch đa hài đơn ổn có một trạng thái ổn định và một trạng thái không ổn định trạng thái không ổn định chính là trạng thái tạo xung. Bình thường, mạch đơn ổn sẽ ở trạng thái ổn định, muốn tạo xung thì phải có xung kích từ bên ngoài.

Mạch đa hài phi ổn khác với hai mạch trên, mạch đơn ổn sẽ tạo ra sóng vuông liên tục mà không cần xung kích bên ngoài. Mạch đa hài phi ổn hoạt động theo đúng nguyên lý của mạch dao động là loại mạch tự phát sinh tín hiệu mà không cần tín hiệu điều khiển ở ngõ vào.

### Mạch đa hài phi ổn cơ bản.

#### a. Sơ đồ

Thông thường, mạch đa hài phi ổn là mạch đối xứng nên hai Transistor có cùng tên và các linh kiện điện trở, tụ điện có cùng trị số.

#### b. Nguyên lý hoạt động :

Tuy là hai Transistor cùng tên, các linh kiện cùng trị số nhưng không thể giống nhau một cách tuyệt đối. Điều này sẽ làm cho hai Transistor mạch dẫn điện không bằng nhau, khi mở điện sẽ có một Transistor dẫn điện mạnh hơn và một Transistor dẫn điện yếu hơn. Nhờ tác dụng của mạch hồi tiếp dương từ cực  $C_2$  về cực  $B_1$  và từ cực  $C_1$  về cực  $B_2$  sẽ làm cho Transistor dẫn mạnh hơn tiến dần đến bão hòa, Transistor dẫn điện yếu hơn tiến dần đến ngưng dẫn.

Giả thiết  $T_1$  dẫn điện mạnh hơn, tụ  $C_1$  nạp điện qua  $RC_2$  làm cho dòng  $IB_1$  tăng cao nên  $T_1$  tiến đến bão hòa. Khi  $T_1$  bão hòa, dòng  $IC_1$  tăng cao và  $V_{C_1} \approx V_{CE_{sat}} \approx 0,2V$ , tụ  $C_2$  xả điện qua  $RB_2$  và qua  $T_1$ . Khi tụ  $C_2$  xả điện, điện áp âm trên tụ  $C_2$  đưa vào cực  $B_2$  làm  $T_2$  ngưng ( hình 3.21)

Thời gian ngưng dẫn của  $T_2$  chính là thời gian tụ  $C_2$  xả điện qua  $RB_2$ . Sau khi tụ  $C_2$  xả xong, cực  $B_2$  lại được phân cực nhờ  $RB_2$  nên  $T_2$  dẫn bão hòa làm  $V_{C_2} = V_{CE_{sat}} \approx 0,2V$ . Điều này làm tụ  $C_1$  xả điện qua  $RB_1$  và điện áp âm trên tụ  $C_1$  đưa vào cực  $B_1$  làm cho  $T_1$  ngưng. Lúc đó tụ  $C_2$  lại nạp điện qua  $RC_1$  làm cho dòng  $IB_2$  tăng cao và  $T_2$  bão hòa nhanh.

Thời gian ngưng dẫn của  $T_1$  chính là thời gian tụ  $CC_1$  xả điện qua  $RB_1$ . Sau khi tụ  $C_1$  xả điện xong, cực  $B_1$  lại được phân cực nhờ  $RB_1$  nên  $T_1$  trở lại trạng thái dẫn bão hòa như trạng thái giả thiết ban đầu. Hiện tượng này được lặp lại tuần hoàn .

#### c. Dạng sóng ở các chân:

Xét cực  $B_1$  khi  $T_1$  bão hòa  $V_B \approx 0,8V$ . Khi  $T_1$  ngưng cho tụ  $C_1$  xả điện làm cực  $B_1$  có điện áp âm ( khoảng-  $V_{CC}$  ) và điện áp âm này giảm dần theo hàm số mũ.

Xét cực  $C_1$  : khi  $T_1$  bão hòa  $V_{C_1} \approx 0,2V$ , khi  $T_1$  ngưng  $V_{C_1} \approx +V_{CC}$  . Dạng sóng ra ở cực  $C$  là dạng sóng vuông. Hình 3.23.

Tương tự khi xét cực  $B_2$  và cực  $C_2$  . Dạng sóng ở hai cực này cùng dạng với dạng sóng ở cực  $B_1$  và  $C$ , nhưng đảo pha nhau.

Chu kỳ của tín hiệu hình vuông là:

$$T = t_1 + t_2.$$

Trong đó:

- $t_1$  là thời gian tụ  $C_1$  xả điện qua  $R_{B1}$  từ điện áp  $-V_{CC}$  lên nguồn  $+V_{CC}$  nên điện áp tức thời của tụ ( lấy mức  $-V_{CC}$  làm gốc ) là:

$$V_{C1}(t) = 2V_{CC} \cdot e^{-\frac{t}{R_{B1} \cdot C_1}}$$

Thời gian để tụ  $C_1$  xả qua  $R_{B1}$  từ  $-V_{CC}$  lên  $0^V$  cho bởi công thức:

$$V_{CC} = 2V_{CC} \cdot e^{-\frac{t_1}{R_{B1} \cdot C_1}}$$

Suy ra:  $e^{-\frac{t_1}{R_{B1} \cdot C_1}} = 2$

$$\Rightarrow \frac{t_1}{R_{B1}} = \ln 2$$

$$\Rightarrow t_1 = R_{B1} \cdot C_1 \ln 2 \approx 0,69 R_{B1} \cdot C_1$$

Tương tự, thời gian  $t_2$  để tụ  $C_2$  xả điện qua  $R_{B2}$  từ  $-V_{CC}$  lên  $0V$  là:

$$t_2 \approx 0,69 R_{B2} \cdot C_2$$

Chu kỳ dao động là:

$$T = t_1 + t_2 = 0,69 ( R_{B1} \cdot C_1 + R_{B2} \cdot C_2 )$$

Trong mạch đa hài phi ổn đối xứng ta có :

$$R_{B1} = R_{B2} = R_B$$

$$C_1 = C_2 = C$$

Chu kỳ dao động là:

$$T = 2 \times 0,69 R_B \cdot C = 1,4 R_B \cdot C$$

Tần số của xung vuông là:

$$f = \frac{1}{T} = \frac{1}{0,69(R_{B1} \cdot C_1 + R_{B2} \cdot C_2)}$$

Nếu là mạch đa hài phi ổn đối xứng ta có :

$$f = \frac{1}{T} = \frac{1}{1,4 R_B \cdot C}$$

#### d. Thiết kế mạch :

Thiết kế mạch đa hài phi ổn theo các thông số kỹ thuật sau  $V_{CC} = 12V$ , dòng điện tải qua cực là  $I_L = 10mA$  transistor có  $\beta = 100$  tần số dao động là  $f = 1000 Hz$ .

##### Bài giải

Mạch đa hài phi ổn là loại đối xứng có sơ đồ như mạch đa hài cơ bản ( hình 3.21 và hình 3.22)

-Tính điện trở RC :

Khi transistor chạy bão hòa sẽ có:

$$V_C = V_{CEsat} \approx 0,2^V$$

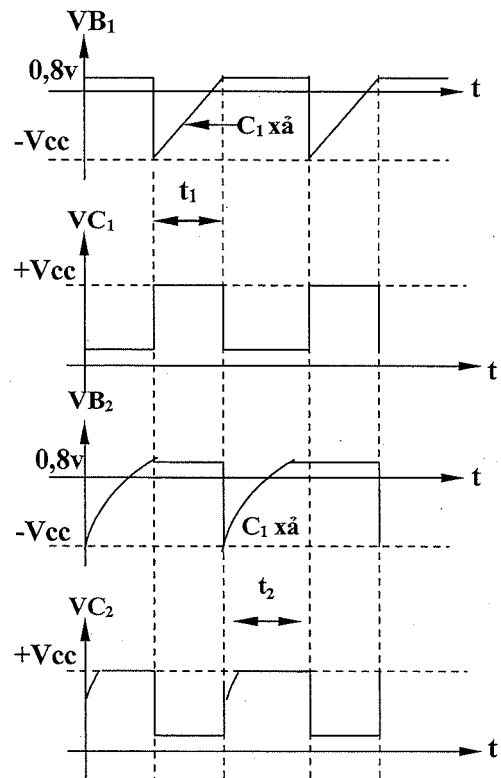
$$I_C = I_L = 10mA$$

Điện trở RC được tính theo công thức:

$$RC = \frac{V_{CC} - V_{CEsat}}{I_C} = \frac{12 - 0,2}{10mA} \approx 1,2^{k\Omega}$$

-Tính điện trở RB

Để transistor chạy bão hòa sâu thường chọn hệ số bão hòa sâu là  $K = 3$ .



Hình 3. 23. Dạng sóng ở các chân.



Ta có :  $I_B = K \frac{I_C}{\beta} = 3 \cdot \frac{10\text{mA}}{100} = 0,3\text{mA}$ .

Điện áp phân cực cho transistor chạy bão hòa là  $V_B = V_{BEsat} = 0,8\text{V}$ .

$$R_{B1} = R_{B2} = \frac{V_{CC} - V_{BEsat}}{I_B} = \frac{12\text{V} - 0,8\text{V}}{0,3\text{mA}} = 37 \text{ K}\Omega.$$

Chọn trị số tiêu chuẩn là:  $R_B = 39\text{K}\Omega$ .

-Tính trị số tụ C:

Từ công thức tần số của mạch đa hài phi ổn đối xứng là:

$$f = \frac{1}{1,4R_B \cdot C}$$

Suy ra:  $C = \frac{1}{1,4R_B \cdot f} = \frac{1}{1,4 \cdot 39 \cdot 10^3 \cdot 10^3} = 0,018\mu\text{F}$

### Mạch phi ổn thay đổi tần số :

#### a. Sơ đồ mạch

Từ công thức tính tần số của mạch đa hài phi ổn định cho thấy tần số dao động có thể thay đổi bằng cách thay đổi trị số điện trở  $R_B$  hay thay đổi giá trị tụ điện C.

Thông thường người ta dùng biến trở VR để thay trị số  $R_B$  như hình 3.23.

#### b. Nguyên lý hoạt động:

Biến trở VR là phần điện trở phân cực chung cho hai cực B của hai transistor. Điều kiện của mạch là khi điều chỉnh biến trở VR sẽ không làm thay đổi nguyên lý hoạt động của mạch, khi dẫn điện transistor vẫn phải ở trạng thái bão hòa.

Khi điều chỉnh biến trở VR sẽ làm thay đổi trị số điện trở  $R_{B1}$  và  $R_{B2}$  trong khoảng :

$$R_{B1\text{max}} = R_1 + VR \quad \text{hay} \quad R_{B2\text{min}} = R_2 + VR$$

$$R_{B1\text{min}} = R_1 \quad \text{hay} \quad R_{B2\text{max}} = R_2$$

Giới hạn trên sẽ cho ra khoảng tần số mà mạch dao động có thể cho ra được.

#### c. Thiết kế mạch :

Giả thiết mạch đa hài phi ổn được thiết kế trong phần trên có tần số điều chỉnh được từ  $f_{\text{min}} = 500\text{Hz}$  đến  $f_{\text{max}} = 1500\text{Hz}$  thì phần tính toán được giải theo trình tự sau:

\*. Đầu tiên ta giả thiết mạch dao động đa hài phi ổn có tần số dao động không đổi là tần số trung bình của  $f_{\text{min}}$  và  $f_{\text{max}}$  :

$$\bar{f} = \frac{f_{\text{min}} + f_{\text{max}}}{2} = \frac{500\text{Hz} + 1500\text{Hz}}{2} = 1000\text{Hz}$$

\*. Với tần số không đổi là  $f = 1000\text{Hz}$  bài toán đã trở về dạng thiết kế mạch đa hài phi ổn cơ bản như trên và ta đã có kết quả:

$$R_C = R_{C1} = R_{C2} = 1,2\text{K}\Omega$$

$$R_B = R_{B1} = R_{B2} = 39\text{K}\Omega$$

$$C = C_1 = C_2 = 0,018\mu\text{F}$$

\*. Sau khi có kết quả trên ta giữ trị số tụ C không đổi và thay trị số điện trở  $R_B$  để thay đổi tần số f.

Ta có:  $f = \frac{1}{1,4R_B \cdot C}$

Suy ra:  $R_B = \frac{1}{1,4f \cdot C}$

Trị số RB tỉ lệ nghịch với tần số f nên ta có hai trường hợp :

- Tần số là  $f_{min}$  khi  $R_{Bmax}$ .
- Tần số là  $f_{max}$  khi  $R_{Bmin}$ .

\* Tính trị số điện trở RB.

$$R_{Bmax} = \frac{1}{1,4f_{min}C} = \frac{1}{1,4.500.0,018.10^{-6}} = 80K\Omega.$$

$$R_{Bmax} = \frac{1}{1,4f_{min}C} = \frac{1}{1,4.500.0,018.10^{-6}} = 80K\Omega$$

$$R_{Bmin} = \frac{1}{1,4f_{max}C} = \frac{1}{1,4.1500.0,018.10^{-6}} = 27K\Omega.$$

Trong phần nguyên lý ta có:

$$R_{Bmin} = R_1 = R_2 = 27k\Omega$$

$$R_{Bmax} = R_1 + VR = R_2 + VR = 80K\Omega$$

$$VR = R_{Bmax} - R_{Bmin} = 80K\Omega - 27k\Omega = 53K\Omega$$

Chọn biến trở  $VR = 50K\Omega$  theo tiêu chuẩn.

Sơ đồ mạch hình 5.24 là mạch đa hài phi ổn thay đổi tần số được thiết kế.

\* Kiểm tra điều kiện bão hòa. Điều kiện của mạch đa hài phi ổn là khi dẫn điện phải ở trạng thái bão hòa. Khi thay đổi biến trở VR sẽ làm thay đổi RB và dòng điện IB nên cần kiểm tra lại trạng thái dẫn của transistor khi có  $R_{Bmax}$ .

$$\text{Ta có : } I_{Bmax} = \frac{V_{CC} - V_{BEsat}}{R_{Bmax}} = \frac{12^V - 0,8^V}{80K\Omega} = 0,14mA$$

Do dòng điện  $I_C = 10$  với  $\beta = 100$  thì ở trạng thái khuếch đại ta có:

$$I_B = \frac{I_C}{\beta} = \frac{10mA}{100} = 0,1mA$$

Dòng điện  $I_{Bmin} = 0,14mA$  vẫn lớn hơn  $I_B = 0,1mA$  nên vẫn đảm bảo transistor dẫn điện bão hòa. Trường hợp không đạt điều kiện này thì phải chọn transistor có  $\beta$  lớn hơn hay dùng transistor ráp kiểu Darlington.

### Mạch thay đổi chu trình làm việc.

Trong chương 1 khái niệm cơ bản về kỹ thuật xung, phần các thông số của tín hiệu xung có khái niệm về chu kỳ T của tín hiệu xung là:

$$T = t_{on} + t_{of}$$

Trong đó  $t_{on}$  là thời gian tín hiệu xung có điện áp cao  $t_{of}$  là thời gian xung có điện áp thấp.

Từ khái niệm trên người ta đưa ra hai khái niệm khác là độ rộng Q và hệ số đầy  $\eta$  của xung.

Độ rộng của xung được tính theo công thức:

$$Q = \frac{T}{t_{on}}$$

Nghịch đảo của độ rộng xung là hệ số đầy được tính theo công thức:

$$\frac{t_{on}}{T}$$

Hệ số đầy còn được gọi tên bằng một khái niệm kỹ thuật khác là chu trình làm việc D

(Duty Cycle)

Như vậy : 
$$D = \frac{t_{on}}{T} 100\%$$

Trong mạch dao động đa hài phi ổn đối xứng ta có thời gian xả của tụ  $C_1$  bằng thời gian xả của tụ  $C_2$  nên:

$$t_1 = t_2 \Rightarrow t_{on} = t_{off} = \frac{1}{2} T$$

Chu trình làm việc của mạch đa hài đối xứng là:

$$D = \frac{t_{on}}{T} 100\% = 50\%$$

Để thay đổi chu trình làm việc D người ta phải thay đổi  $t_{on}$  hoặc  $t_{off}$  nhưng phải giữ nguyên chu kỳ T.

### a. Sơ đồ mạch thay đổi chu trình làm việc :

Mạch điện hình 5.25 có biến trở VR dùng để thay đổi chu trình làm việc D.

$$VR = R_1 + R_2.$$

Điện trở R là phần điện trở RB dùng chung cho cả hai transistor.

$$\text{Ta có : } RB_1 = R + R_1.$$

$$RB_2 = R + R_2.$$

Khi điều chỉnh biến trở theo hướng tăng trị số  $R_1$  sẽ làm giảm trị số  $R_2$  và ngược lại.

Điều này có nghĩa là khi  $RB_1$  tăng thì giảm trị số  $RB_2$  và ngược lại.

Ta vẫn có thời gian xả của hai tụ  $C_1$  và  $C_2$  tính theo công thức sau:

$$t_1 = 0,69 RB_1 \cdot C_1 = 0,69 (R + R_1) C_1$$

$$t_2 = 0,69 RB_2 \cdot C_2 = 0,69 (R + R_2) C_2$$

Giả thiết  $C_1 = C_2 = C$  ta có chu kỳ T của tín hiệu xung vuông là:

$$T = t_1 + t_2 = 0,69 (R + R_1) C_1 + 0,69 (R + R_2) C_2$$

$$T = 0,69 [(R + R_1) + (R + R_2)] C$$

$$T = 0,69 [(R + R_1) + (R + R_2)] C$$

$$T = 0,69 (2R + R_1 + R_2) C$$

$$T = 0,69 (2R + VR) C$$

Như vậy, khi điều chỉnh biến trở VR sẽ làm không thay đổi chu kỳ T tức là giữ nguyên tần số f mà chỉ làm thay đổi thời gian  $t_1$ ,  $t_2$  tức là thời gian  $t_{on}$ ,  $t_{off}$  sẽ làm thay đổi chu trình làm việc D.

### b. Nguyên lý thiết kế:

Biến trở VR là phần điện trở phân cực chung cho hai cực B của hai transistor. Khi điều chỉnh biến trở đúng vị trí giữa điện trở phân cực cho hai transistor bằng nhau là:

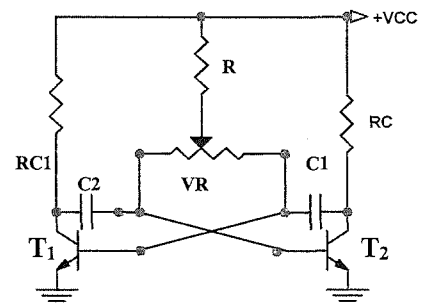
$$R_{B1} = R_{B2} = R + R = R + R = R + \frac{1}{2} VR$$

Khi thay đổi vị trí của biến trở VR sang phải hay sang trái làm tăng điện trở phân cực  $RB_1$ , giảm điện trở phân cực  $RB_2$  và ngược lại. Khi  $RB_1$  cực tiểu thì  $RB_2$  cực đại và ngược lại

$$\text{Ta có : } RB_{1min} = RB_{2min} = R$$

$$RB_{1max} = RB_{2max} = R + VR$$

Giả thiết mạch đa hài phi ổn được thiết kế trong phần trên có tần số dao động là 1000Hz nhưng chu trình làm việc thay đổi được từ 40% đến 60% thì phần tính toán được giải theo trình tự sau:



Hình 3.25: Mạch thay đổi chu trình

\* Đầu tiên ta giả thiết mạch dao động đa hài phi ổn có tần số là  $f = 1000\text{Hz}$  và chu trình làm việc không đổi là 50% (mạch phi ổn đôi xứng).

\* Với giả thiết bài toán đã trở về dạng thiết kế mạch cơ bản như trên và đã có kết quả:

$$RC = RC_1 = RC_2 = 1,2\text{K}\Omega$$

$$\bar{R}_B = RB_1 = RB_2 = 39\text{K}\Omega$$

(Trị số  $RB$  trung bình ứng với biến trở  $VR$  ở vị trí giữa)

$$C = C_1 = C_2 = 0,018\mu\text{F}$$

\* Sau khi có kết quả trên ta giữ trị số tụ  $C$  không đổi và thay đổi trị số điện trở  $RB_1, RB_2$  để thay đổi  $t_1, t_2$  tức là thay đổi chu trình làm việc.

Ta chỉ cần tính cho  $t_1$  sẽ suy ra tương tự cho  $t_2$

Từ tần số  $f = 1000\text{Hz}$  suy ra chu kỳ  $T$  là:

$$T = \frac{1}{f} = \frac{1}{1000} = 1\text{ms}$$

Khi chu trình làm việc là  $D = 40\%$  thì thời gian  $t_1$  là:

$$t_1 = \frac{40}{100} T = 0,4\text{ms}$$

và  $t_1 = 0,69RB_{1\min} \cdot C = 0,4\text{ms}$

$$\text{Suy ra: } RB_{1\min} = \frac{0,4\text{ms}}{0,69 \cdot 0,018 \cdot 10^{-6}} = 32,2\text{K}\Omega$$

$$\Rightarrow R = RB_{\min} = 32,2\text{K}\Omega \text{ (chọn } R = 33\text{K}\Omega)$$

Khi chu trình làm việc là  $D = 60\%$  thì thời gian  $t_1$  là:

$$t_1 = \frac{60}{100} T = 0,6\text{ms}$$

và  $t_1 = 0,69RB_{1\max} \cdot C = 0,6\text{ms}$

$$\text{Suy ra: } RB_{1\max} = \frac{0,6\text{ms}}{0,69 \cdot 0,018 \cdot 10^{-6}} = 48,3\text{K}\Omega.$$

$$\Rightarrow RB_{1\max} = R + VR.$$

Như vậy:  $VR = RB_{1\max} - R = 48,3\text{K}\Omega - 33\text{K}\Omega = 15,3\text{K}\Omega.$

Chọn biến trở  $VR = 15\text{K}\Omega.$

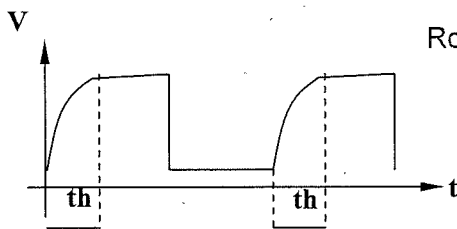
### Mạch đa hài phi ổn ở các dạng khác.

#### a. Mạch đa hài phi ổn dùng diode sửa dạng sóng:

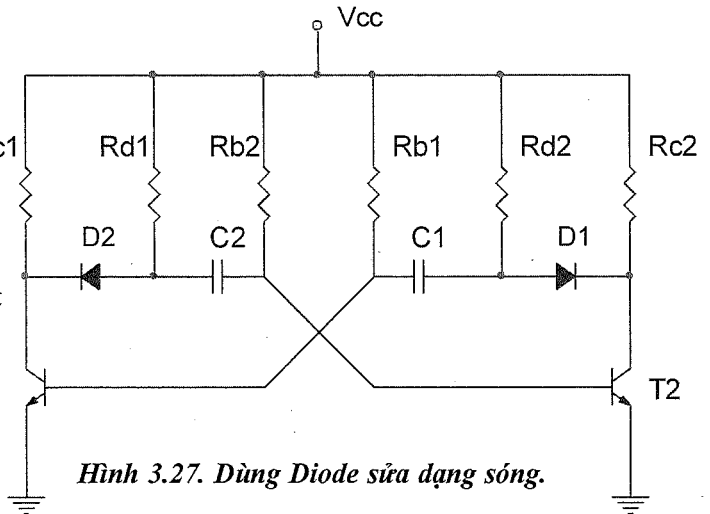
Khi transistor trong mạch phi ổn đổi trạng thái từ bão hòa ( $VC = V_{CEsat} \approx 0,2\text{V}$ ) sang ngưng dẫn ( $VC \approx V_{CC}$ ) thì điện áp ra không tăng lên tức thời theo dạng sóng vuông được vì lúc đó tụ  $C$  nạp qua  $RC$  làm điện áp ra tăng lên theo hàm số mũ. Thời gian điện áp ra tăng lên theo hàm số mũ gọi là thời gian hồi phục th. Thời gian này tụ thuộc hằng số thời gian nạp của tụ  $C$  và điện trở  $RC$ , th được tính theo công thức:

$$th = 3\tau_{\text{nạp}} = 3RC \cdot C$$

Để sửa chữa dạng sóng ra người ta dùng hai diod  $D_1-D_2$  để cách ly các tụ  $C_1-C_2$  với hai cực  $C_1-C_2$  Khi  $T_2$  ngưng dẫn tụ  $C_1$  sẽ không nạp điện qua  $RC_2$  vì  $D_2$  bị phân cực ngược, do đó tụ  $C_1$  sẽ nạp điện qua  $RD_2$ . Như vậy điện áp  $VC_2$  sẽ tăng nhanh cho ra dạng sóng vuông. Suy luận tương tự cho tụ  $C_2$  và điện áp  $VC_1$  khi  $T_1$  ngưng dẫn.



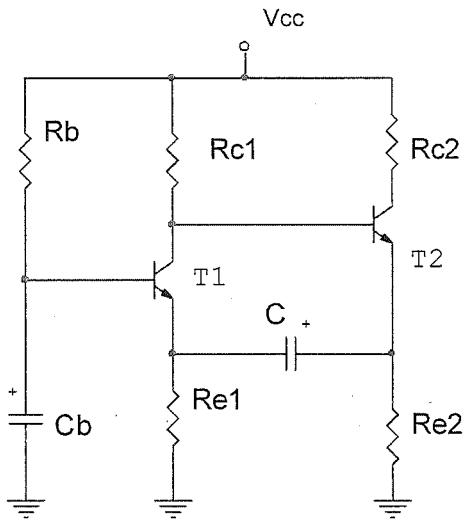
Hình 3.26. Dạng sóng vuông do  $T_1$  thời gian hồi phục.



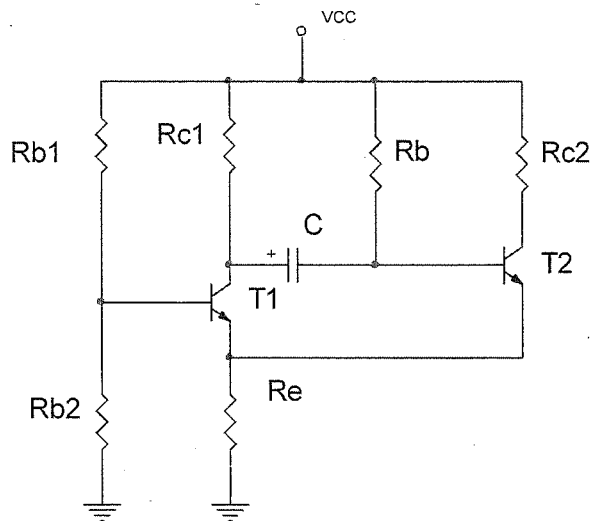
Hình 3.27. Dùng Diode sửa dạng sóng.

**b. Mạch đa hài phi ổn hồi tiếp về cực E :**

Mạch điện hình 5.28 có tụ hồi tiếp C nối giữa hai cực E, trong khi tụ CB loại tụ hóa có trị số lớn sẽ có tác dụng lọc bỏ thành phần xoay chiều xuống mass. Transistor  $T_1$  được ráp như kiểu cực B chung nên tín hiệu cực E và ra ở cực  $C_1$ .



Hình 3.28. Mạch phi ổn hồi tiếp về cực E.



Hình 3.29. Mạch phi ổn dùng chung Re.

Mạch điện 3.28 chạy theo nguyên lý sau.

Khi mới mở điện, tụ CB nạp nên  $V_{B1} = 0V$  làm  $T_1$  ngưng và  $V_{C1}$  tăng cao làm cho  $T_2$  bão hòa, dòng  $I_{C2}$  tăng ca.

Dòng  $I_{E2}$  sẽ qua  $R_{E2}$  làm  $V_{E2}$  tăng, tụ C nạp điện qua  $R_{C2}$  và  $R_{E1}$ . Dòng nạp vào tụ C làm  $V_{E1}$  tăng nên transistor  $T_1$  càng dễ ngưng dẫn.

Khi tụ C nạp đầy thì  $I_{E1} \approx 0$  làm  $V_{E2}$  giảm trong khi đó tụ CB đã nạp đầy nên  $V_{B1}$  cao làm  $T_1$  bão hòa. Khi  $T_1$  bão hòa có dòng  $I_{C1}$  qua  $R_{C1}$  nên  $V_{C1}$  giảm làm  $V_{B2}$  giảm và  $T_2$  ngưng.

Khi  $T_1$  bão hòa,  $T_2$  ngưng thì điện áp trên tụ C sẽ nối tiếp với điện áp nguồn và tụ C sẽ xả qua  $R_{C1}$  và  $R_{E2}$ . Khi tụ C xả điện qua  $R_{E2}$  sẽ làm  $V_{E2}$  giảm dần đến mức đủ nhỏ thì  $T_2$  lại dẫn điện tạo dòng điện nạp vào tụ C nên  $V_{E1}$  lại tăng cao làm  $T_1$  ngưng dẫn. Mạch đã trở lại trạng thái ban đầu. Hiện tượng trên lại tiếp tục và tuần hoàn.

Mạch điện hình 3.29 có tụ C ghép giữa cực  $C_1$  và cực  $B_2$ , mạch hồi tiếp được thực hiện nhờ hai transistor có chung điện trở RE. Khi mở điện tụ C nạp nhanh qua  $RC_1$ - $T_2$ -RE làm  $T_2$  dẫn và cho ra  $IE_2$  qua RE nên  $VE_1 = VE_2$  tăng cao và làm  $T_1$  ngưng. Khi tụ C nạp đầy làm mất dòng  $IB_2$  nên  $T_2$  ngưng dẫn.  $IE_2 = 0$  làm cho  $T_1$  dẫn bão hòa nhờ có cầu phân áp  $RB_1$  và  $RB_2$ . Lúc đó điện áp trên tụ C nối tiếp với điện áp nguồn sẽ xả qua RB và  $T_1$ -RE. Sau khi tụ xả xong làm mất điện áp âm đặt vào cực  $B_2$  nên  $T_2$  dẫn điện trở lại như trạng thái ban đầu.

Trong mạch điện hình 3.28 tụ C nạp điện qua  $RC_2$  - RE<sub>1</sub> và xả điện qua  $RC_1$ - RE<sub>2</sub>.

Ta có :  $\tau_{\text{nạp}} = (RC_2 + RE_1) \cdot C$  và  $\tau_{\text{xả}} = (RC_1 + RE_2) \cdot C$ .

Nếu mạch được thiết kế có  $RC_1 = RC_2$  và  $RE_1 = RE_2$  thì xung vuông ở ngõ ra là tín hiệu vuông đối xứng.

Trong mạch điện hình 3.29 tụ C nạp qua  $RC_1$ -RE, xả điện qua RB- RE.

Ta có :  $\tau_{\text{nạp}} = (RC_1 + RE) \cdot C$  và  $\tau_{\text{xả}} = (RB + RE) \cdot C$

Do điện trở RB thường rất lớn so với  $RC_1$  nên xung ra là một tín hiệu vuông không đối xứng.

### c. Mạch đa hài phi ổn dùng hai Transistor khác loại:

Trong mạch điện hình 4.30 dùng hai Transistor khác loại để tạo mạch đa hài phi ổn gồm  $T_1$  loại NPN và  $T_2$  loại PNP. Mạch có nguyên lý làm việc như sau :

Khi mở điện  $T_1$  dẫn do được phân cực cầu phân thế  $RB_1$ -  $RB_2$ . Lúc đó dòng  $IC_1$  làm giảm điện áp  $VC_1$  nên cũng làm giảm  $VB_2$ .

Hiện tượng này sẽ làm cho  $T_2$  cũng được phân cực nên sẽ dẫn điện ( vì  $T_2$  là loại PNP)

Khi  $T_2$  dẫn có dòng  $IC_2$  qua  $RC_2$  làm  $VC_2$  tăng và tụ C nạp điện qua RE<sub>2</sub> -  $T_2$  và  $RB_2$  xuống mass. Dòng nạp này qua  $RB_2$  làm  $VB_1$  tăng và  $T_1$  chạy bão hòa kéo  $T_2$  bão hòa theo.

Khi tụ C nạp đầy làm mất dòng điện qua  $RB_2$  nên  $VB_2$  giảm làm  $T_1$  chạy yếu dẫn theo  $T_2$  chạy yếu. Lúc đó  $VC_2$  giảm nhỏ do  $T_2$  dẫn yếu làm điện áp trên tụ C ghép nối tiếp với điện áp nguồn và tụ C sẽ xả điện qua  $RB_1$  và  $RC_2$ . Lúc đó điện áp âm trên tụ C sẽ làm  $VB_1$  âm, nên  $T_1$  ngưng kéo theo  $T_2$  ngưng theo.

Khi tụ C xả điện xong thì mạch trở lại trạng thái ban đầu và hiện tượng trên được tiếp diễn liên tục tuần hoàn.

Thời gian nạp của tụ qua  $RE_2$  và  $RB_2$  có trị số nhỏ nên ngắn hơn so với thời gian xả của tụ  $RB_1$  và  $RC_2$ . Do đó, tín hiệu xung ra có dạng xung vuông không đối xứng.

### d. Mạch đa hài phi ổn cho ra tần số thấp:

Mạch đa hài phi ổn cơ bản có công thức tính tần số dao động là:

$$f = \frac{1}{1,4R_B \cdot C}$$

Trong công nghiệp có những trường hợp cần tạo ra tín hiệu xung bỏ có tần số f rất thấp ( $f \ll 1\text{Hz}$ ) thí dụ:  $f = 0,01\text{Hz}$  Nói cách khác, xung vuông có chu kỳ T rất dài ( $T \gg 1\text{gy}$ ).

Theo công thức trên, để có f rất thấp thì trị số  $RB$  và tụ C phải rất lớn. Trong thực tế trị số tụ C loại tụ hóa cũng có mức giới hạn không thể quá lớn, điện trở RB nếu quá lớn sẽ không thỏa điều kiện bão hòa sâu vì lúc đó dòng  $IB$  sẽ nhỏ theo công thức:

$$IB = K \cdot \frac{I_C}{\beta}$$

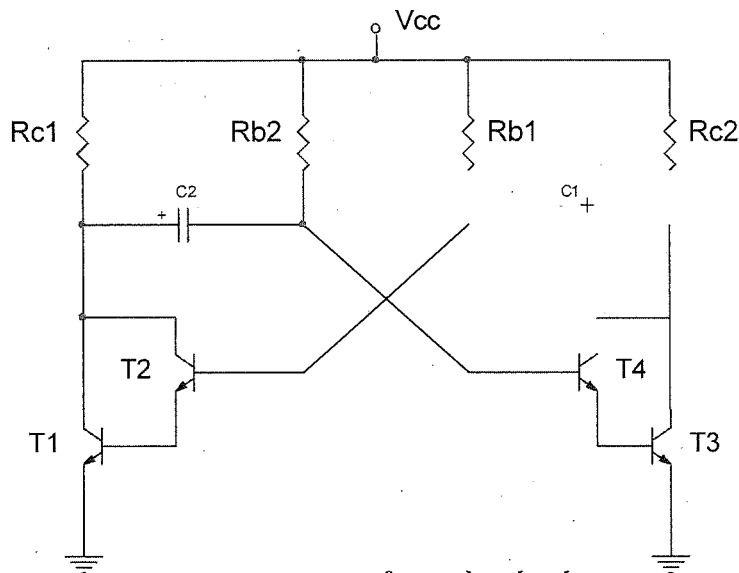
Để giải quyết cho trường hợp trên ta có thể dùng hai transistor ráp kiểu Darlington để có độ khuếch đại dòng lớn. Trong mạch điện hình 4.31, hai transistor  $T_{1A}-T_{1B}$  ráp kiểu Darlington,  $T_{2A}-T_{2B}$  ráp kiểu Darlington để tạo ra mạch đa hài đối xứng với cách ráp kiểu Darlington thì độ khuếch đại dòng chung cho hai transistor là:  $\beta_{\text{chung}} = \beta_A \cdot \beta_B$  ( $\approx$  vài ngàn ÷ vài chục ngàn)

Điều kiện bảo hòa sâu bây giờ sẽ là:

$$I_B = K \frac{I_C}{\beta_A \beta_B}$$

Với  $\beta$  chung rất lớn thì dòng điện  $I_B$  sẽ có trị số rất nhỏ và điều này giúp cho việc chọn trị số  $R_B$  có thể lớn theo công thức:

$$R_B = \frac{V_{CC} - 2V_{BE}}{I_B}$$



Hình 3.28. Mạch phi ổn có tần số thấp



# CHƯƠNG 4

## CÁC MẠCH DAO ĐỘNG KHÁC

### MẠCH DAO ĐỘNG ĐA HÀI DÙNG OP-AMP

Trong phần trước chúng ta đã khảo sát các mạch tạo xung dùng transistor rời, trong phần này chúng ta sẽ khảo sát các ứng dụng của OP-AMP trong kỹ thuật xung.

#### Mạch lưỡng ổn (FLIP – FLOP) dùng OP-AMP:

Mạch F/F dùng OP-AMP như sơ đồ hình 4.1 có hai OP-AMP làm việc như hai mạch khuếch đại so sánh. Hai OP-AMP sẽ ở trạng thái bão hòa dương nếu có:

$$V_{i^+} > V_{i^-} \Rightarrow V_0 = +V_{CC}$$

hay ở trạng thái bão hòa âm nếu có:

$$V_{i^-} > V_{i^+} \Rightarrow V_0 = +0^V$$

Giả thuyết mạch có trạng thái như hình vẽ với  $V_{01} = +V_{CC}$  và  $V_{02} = 0^V$

OP-AMP (1) được hồi tiếp từ  $V_{02} = 0^V$  về ngõ  $In^-$  qua điện trở 10K nên vẫn có:

$V_{i^+} > V_{i^-}$  và  $V_{01} = +V_{CC}$  ổn định

OP-AMP (2) được hồi tiếp từ  $V_{01} = +V_{CC}$  về ngõ  $In^-$  qua điện trở 10K ( $10K \ll 100K$ ) nên vẫn có:  $V_{i^-} > V_{i^+}$  và  $V_{01} = 0^V$  ổn định Đây là trạng thái ổn định thứ nhất của mạch F/F, OP-AMP(1) ở trạng thái bão hòa dương và OP-AMP(2) ở trạng thái bão hòa âm. Để đổi trạng thái của F/F, cho công tắc S nối vào ngõ  $In^-$  của OP-AMP(2) đang bão hòa âm. Lúc đó  $V_{i^-} = 0V$  và  $V_{i^+} > V_{i^-}$  nên OP-AMP(2) chuyển sang bão hòa dương,  $V_{02} = +V_{CC}$  qua điện trở hồi tiếp 10K sẽ làm đổi trạng thái cũ OP-AMP(1) từ bão hòa dương sang bão hòa âm vì lúc đó OP-AMP(1) có  $V_{i^-} > V_{i^+}$ .

Lưu ý: Điện trở hồi tiếp phải có trị số khá nhỏ so với điện trở nối ngõ  $In^+$  lên nguồn

+VCC Công tắc S có điểm chung nối masse

xem như xung âm kích điều khiển F/F.

Công tắc điều khiển lên nguồn +VCC

qua điện trở để kích đổi trạng thái của

F/F như xung dương kích điều khiển F/F.

Trường hợp này xung dương phải được đưa vào OP-AMP (1) đang bão hòa dương.

Sơ đồ hình là hình 4.3 mạch F/F đổi trạng thái bằng xung dương.

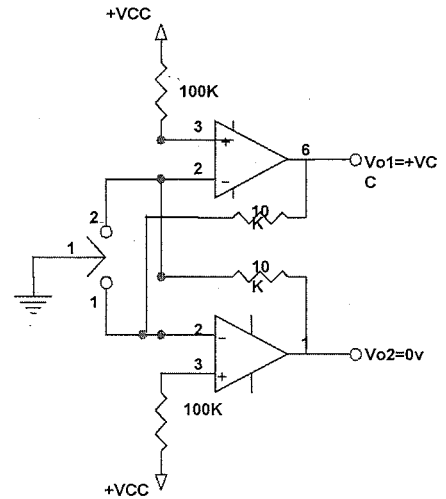
Như vậy, để đổi trạng thái bằng xung dương.

- Cho xung âm (hay mức điện thế thấp) vào ngõ  $In^-$  của OP-AMP đang bão hòa âm.

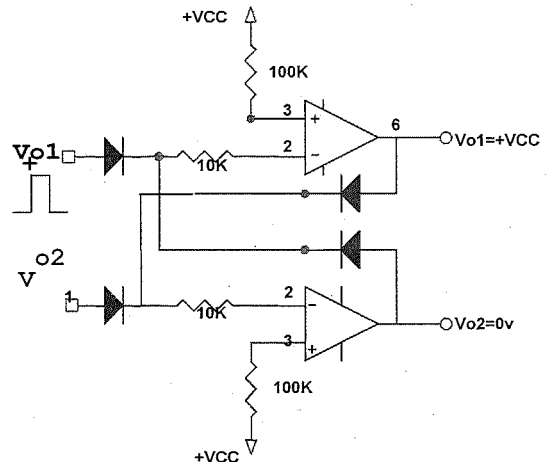
- Cho xung dương (hay mức điện thế cao) vào ngõ  $In^-$  của OP-AMP đang bão hòa dương.

#### Mạch flip-flop hồi tiếp bằng diode:

Mạch Flip-Flop hình 4.3 dùng hai diode



Hình 4.1: Mạch F/F dùng OP-AMP kích đổi trạng thái bằng xung âm



Hình 4.3: Mạch F/F hồi tiếp bằng D

D<sub>1</sub>-D<sub>2</sub> để nhận xung kích ở ngõ vào và hai diod D<sub>3</sub>-D<sub>4</sub> để lấy điện áp hồi tiếp.

Giả sử mạch đang có trạng thái ổn định như hình vẽ, OP-AMP(1) đang bão hòa dương, V<sub>01</sub>=+VCC; OP-AMP (2) đang bão hòa âm, V<sub>01</sub>̄=0V.

Trường hợp này nếu cho xung âm vào ngõ V<sub>12</sub> của OP-AMP đang bão hòa âm thì diode D<sub>2</sub> bị phân cực ngược nên xung âm không tác động được vào mạch F/F phải cho xung dương vào ngõ V<sub>11</sub> của OP-Amp đang bão hòa dương. Lúc đó, diode D<sub>1</sub> được phân cực thuận sẽ cho xung dương vào mạch F/F và làm cho mạch đổi trạng thái.

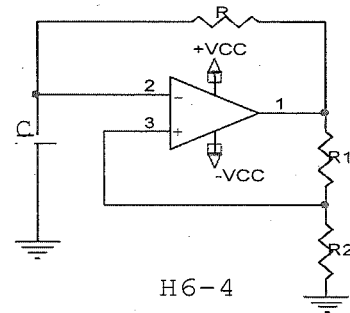
Như vậy, khi sử dụng diode như hình 4.3 thì 2 mạch F/F chỉ còn một cách kích đổi trạng thái là cho xung dương (hay mức điện thế cao) vào ngõ In<sup>-</sup> của OP-AMP (1) đang bão hòa dương.

#### Mạch dao động tích thoát:

##### a. Nguyên lý:

Mạch điện hình 4.4 là sơ đồ mạch dao động tích thoát dùng OP-AMP để cho ra tín hiệu vuông.

Sơ đồ có hai mạch hồi tiếp từ ngõ ra về hai ngõ vào. C là tụ phân áp RC hồi tiếp về ngõ In<sup>-</sup>, cầu phân áp R<sub>1</sub>-R<sub>2</sub> hồi tiếp về ngõ In<sup>+</sup>.



Hình 4.4. Mạch phi ổn hồi tiếp về cực E.

Để giải thích nguyên lý mạch ta giả sử tụ C chưa nạp điện và OP-AMP đang ở trạng thái bão hòa dương. Lúc này cầu phân áp R<sub>1</sub>-R<sub>2</sub> đưa điện áp dương về ngõ In<sup>+</sup> với mức điện áp là: V<sub>0</sub> = +VCC

$$V_{in}^+ = +V_{cc} \cdot \frac{R_2}{R_1 + R_2} = VA \quad (V_{in}^+ > 0V)$$

Trong khi đó, ở ngõ In<sup>-</sup> có điện áp tăng dần lên từ 0V, điện áp tăng do tụ C nạp qua R theo quy luật hàm số mũ với hằng số thời gian là  $\tau = R.C$

Khi tụ C nạp có V<sub>in</sub><sup>-</sup> < V<sub>in</sub><sup>+</sup> thì OP-AMP vẫn ở trạng thái bão hòa dương. Khi tụ C nạp đến mức điện áp V<sub>in</sub><sup>-</sup> > V<sub>in</sub><sup>+</sup> thì OP-AMP đổi thành trạng thái bão hòa âm, ngõ ra có V<sub>0</sub> = -VCC. Lúc này cầu phân áp R<sub>1</sub>-R<sub>2</sub> đưa điện áp âm về ngõ In<sup>+</sup> với mức điện áp là:

$$V_{in}^+ = -V_{cc} \cdot \frac{R_2}{R_1 + R_2} = VB \quad (V_{in}^+ < 0V)$$

Trong khi đó ở ngõ In<sup>-</sup> vẫn còn đang ở mức điện áp dương với trị số:

$$V_{in}^- = +V_{cc} \cdot \frac{R_2}{R_1 + R_2}$$

do tụ C đang còn nạp điện. Như vậy OP-Amp sẽ chuyển sang trạng thái bão hòa âm nhanh cho cạnh vông thẳng đứng. Tụ C bây giờ sẽ xả điện áp dương đang nạp trên tụ qua R<sub>1</sub> và tải ở ngõ ra xuống mass.̄ Khi tụ C xả điện áp dương đang có thì V<sub>in</sub><sup>+</sup> vẫn ở mức điện áp âm nên OP-AMP vẫn ở trạng thái bão hòa âm. Khi tụ C xả hết điện áp dương sẽ nạp điện qua R để có điện áp âm đang có do ngõ ra đang ở trạng thái bão hòa âm chiều nạp bây giờ ngược với chiều dòng điện nạp trên hình vẽ.

Khi tụ C nạp điện áp âm đến mức V<sub>in</sub><sup>-</sup> < V<sub>in</sub><sup>+</sup> (ngõ In<sup>-</sup> nhỏ hơn ngõ In<sup>+</sup>) thì OP-AMP lại đổi thành trạng thái bão hòa dương về ngõ ra có V<sub>0</sub> = +VCC.

Mạch đã trở lại trạng thái giả thiết ban đầu và hiện tượng trên cứ tiếp diễn liên tục tuần hoàn.

**b. Dạng sóng ở các chân:**

Mức giới hạn điện áp ngõ ra là:  $V_{0max} \approx +V_{CC}$   
 $V_{0min} \approx -V_{CC}$

Mức giới hạn điện áp ở hai ngõ

vào là:  $V_A = +V_{CC} \cdot \frac{R_2}{R_1 + R_2}$

$V_B = -V_{CC} \cdot \frac{R_2}{R_1 + R_2}$

Dạng điện ở ngõ vào  $In^-$  là dạng tan giác. Thời gian điện áp ở ngõ vào  $In^-$  tăng từ VB lên VA là OP-AMP ở trạng thái bão hòa dương, Thời gian điện áp ở ngõ vào  $In^-$  giảm từ VA xuống VB là OP-AMP ở trạng thái bão hòa dương. Dạng điện áp ở ngõ  $In^+$  và ngõ ra là trạng thái xung vuông đối xứng. Chu kỳ của tín hiệu được tính theo công thức

$$T = 2R \cdot C \ln \frac{R_1 + 2R_2}{R_1}$$

Suy ra tần số của tín hiệu xung được tính theo công thức

$$f = \frac{1}{T}$$

Trường hợp đặc biệt :

$$R_1 = 2R_2 \Rightarrow T = 2 \cdot R \cdot C \ln 2 \quad (\ln 2 = 0,69)$$

$$= 2 \cdot R \cdot C \cdot 0,69$$

$$\Rightarrow f = \frac{1}{2 \cdot 0,69 \cdot R \cdot C} \approx \frac{1}{1,4 \cdot R \cdot C}$$

$$R_1 = R_2 \Rightarrow T = 2 \cdot R \cdot C \ln 3 \quad (\ln 3 = 1,1)$$

$$= 2 \cdot R \cdot C \cdot 1,1$$

$$\Rightarrow f = \frac{1}{2,2 \cdot R \cdot C}$$

**c. Mạch đổi tần số:**

Theo công thức tính chu kỳ và tần số dao động như trên ta có thể đổi tần số dao động bằng các phương pháp sau:

Thay đổi tỉ số của cầu phân thế trong mạch hồi tiếp chương ( $R_1$  và  $R_2$ )

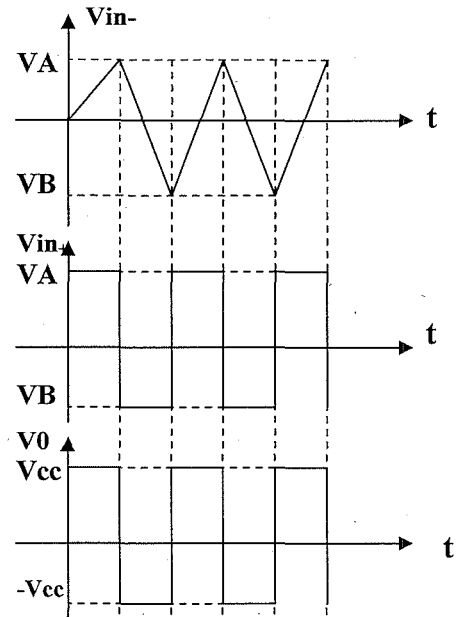
Thay đổi trị số điện trở R hay tụ C trong mạch hồi tiếp âm.

Tần số của mạch dao động tính thoát hình 6-6 được tính theo công thức:

$$T = 2 (R + VT_1) \cdot C \ln \frac{R_1 + 2(R_2 + VR_2)}{R_1}$$

**d. Mạch đổi chu trình làm việc:**

Trong sơ đồ mạch dao động tích thoát cơ bản dùng OP-AMP, tụ C nạp điện và xả điện đều qua điện trở R nên hằng số thời gian nạp và xả bằng nhau. Điều này có nghĩa là thời gian xung vuông có điện áp cao và thời gian xung vuông có điện áp thấp dài bằng nhau. Xung vuông ra là xung không đối xứng có chu trình làm việc là  $D = 50\%$ .



Hình 4.5. Dạng sóng ở các chân.

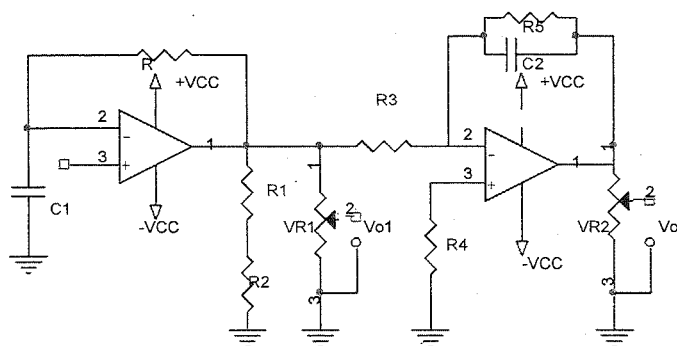
Để thay đổi chu trình làm việc, mạch dao động tích thoát có sơ đồ như hình 4.47, biến trở VR sẽ làm thay đổi thời gian nạp và thời gian xả của tụ theo hai hướng ngược nhau, nên tăng thời gian nạp sẽ làm giảm thời gian xả và ngược lại

Khi tụ C nạp điện áp dương từ ngõ ra sẽ nạp qua điện trở R, biến trở VR (phần dưới) và qua diode D<sub>2</sub>. Khi tụ C xả điện áp dương và sau đó nạp điện áp âm sẽ xả qua R < biến trở VR (phần trên) và qua diode D<sub>1</sub>.

Khi điều chỉnh biến trở VR chỉ làm thay đổi chu trình làm việc mà vẫn giữ nguyên tần số dao động.

#### 4.1.4 Mạch tạo xung vuông và tam giác:

Mạch dao động tích thoát cơ bản tạo xung đối xứng ở ngõ ra. Nếu kết hợp mạch tích phân tích cực dùng OP-AMP thì mạch có thể cho ra xung tam giác.



Hình 4.8: Mạch tạo xung vuông và tam giác

OP-AMP (1) là mạch dao động tích thoát để tạo xung vuông theo nguyên lý trên. Xung vuông được lấy trên biến trở VR<sub>1</sub> để thay đổi biên độ ngõ ra

OP-AMP (2) là mạch tích phân tích cực nhận xung vuông từ ngõ ra của OP-AMP(1) đổi thành dạng xung tam giác. Xung tam giác được lấy trên biến trở VR<sub>2</sub> để thay đổi biên độ ngõ ra.

#### MẠCH DAO ĐỘNG ĐA HÀI DÙNG VI MẠCH 555.

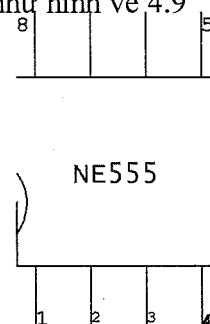
Vi mạch định thì 555 là mạch tích hợp Analog – Digital nên có nhiều tác giả xếp vào một chương trong giáo trình Mạch tương tự, có nhiều tác giả lại xếp vào một chương trong giáo trình Mạch số

Vi mạch định thì 555 và họ của nó được ứng dụng rất rộng rãi trong thực tế, đặc biệt trong lĩnh vực điều khiển vì nếu kết hợp với các linh kiện RC thì nó có thể thực hiện nhiều chức năng như định thì, tạo xung chuẩn, tạo tín hiệu kích hay điều khiển các linh kiện bán dẫn công suất như Transistor, SCR, Triac... Do đó chúng tôi xếp Vi mạch định thì 555 là một chương trình "Kỹ thuật xung".

##### Sơ đồ chân và cấu trúc:

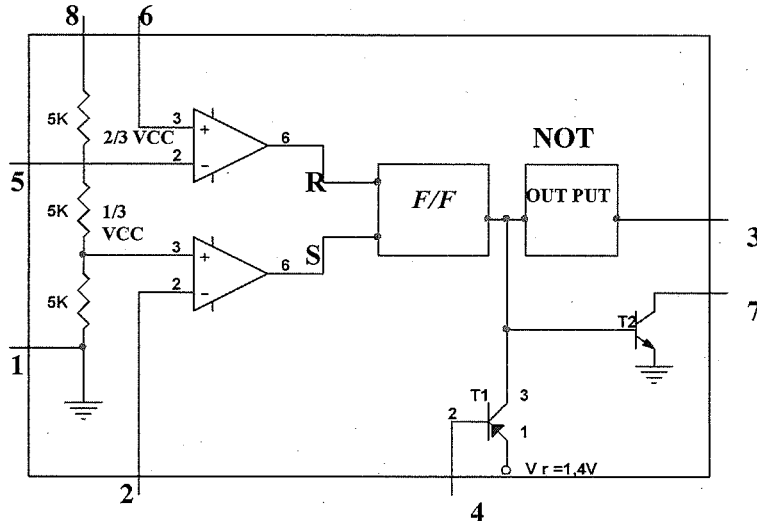
Vi mạch 555 được chế tạo thông dụng nhất là dạng vỏ Plastic như hình vẽ 4.9

- Chân1: GND (nối đất)
- Chân2: Trigger Input (ngõ vào xung nảy)
- Chân3: Output (ngõ ra)
- Chân4: Reset (hồi phục)
- Chân5: Control Voltage (điện áp điều khiển)
- Chân6: Threshold (Thêm –ngưỡng)
- Chân7: Discharge (xả điện)



Chân8: +VCC ( nguồn dương)

Bên trong vi mạch 555 có hơn 20 Transistor và nhiều điện trở thực hiện chức năng như trong hình 4.10 gồm có:



Hình 4.10 Cấu trúc của IC 555.

1)- Cầu phân áp gồm 3 điện trở  $5K\Omega$  nối từ nguồn  $=U_{CC}$  xuống mass cho ra 2 áp thế chuẩn là  $1/3 U_{CC}$  và  $2/3 U_{CC}$ .

2)- OP-AMP (1) là mạch khuếch đại so sánh có ngõ  $I_n$  nhận điện áp chuẩn  $2/3 U_{CC}$  còn ngõ  $I_n^+$  thì nối ra ngoài chân 6. Tùy thuộc điện áp của chân 6 so với điện áp chuẩn  $2/3 U_{CC}$  mà OP-AMP (1) có điện áp mức cao hay thấp để làm tín hiệu R (Reset) điều khiển Flip-Flop (F/F).

3)- OP-AMP (2) là mạch khuếch đại so sánh có ngõ  $I_n^+$  nhận điện áp chuẩn  $1/3 U_{CC}$  còn ngõ  $I_n$  thì nối ra ngoài chân 2. Tùy thuộc điện áp chân 2 so với điện áp chuẩn  $1/3 U_{CC}$  mà OP-AMP (2) có áp thế ra mức cao hay thấp để làm tín hiệu S (Set) điều khiển Flip-Flop (F/F).

4)- Mạch Flip-Flop (F/F) là loại mạch lưỡng ổn kích một bên. Khi chân Set (S) có điện áp cao thì điện áp này kích đổi trạng thái của F/f là ngõ Q lên mức cao và ngõ  $\bar{Q}$  của xuống mức thấp. Khi ngõ Set đang ở mức cao xuống thấp thì mạch F/F không đổi trạng thái. Khi chân Reset (R) có điện áp cao thì điện áp này kích đổi trạng thái. Khi chân Reset (R) có điện áp cao thì điện áp này kích đổi trạng thái của F/f làm ngõ  $\bar{Q}$  lên mức cao và ngõ Q xuống mức thấp. Khi ngõ Reset đang ở mức cao xuống thấp thì mạch F/F không đổi trạng thái.

5)- Mạch OUTPUT là mạch khuếch đại ngõ ra để tăng độ khuếch đại dòng cấp cho tải. Đây là mạch khuếch đại đảo có ngõ vào là chân  $\bar{Q}$  của F/F nên khi  $\bar{Q}$  ở mức cao thì ngõ ra chân 3 của IC sẽ có điện áp thấp ( $\approx 0V$ ) và ngược lại khi  $\bar{Q}$  ở mức thấp thì ngõ ra chân 3 của IC sẽ có điện áp cao ( $\approx U_{CC}$ ).

6)- Transistor  $T_1$  có chân e nối vào 1 điện áp chuẩn khoảng 1,4 V và loại Transistor. PNP nên khi cực b nối ra ngoài bởi chân 4 có điện áp cao hơn 1,4V thì  $T_1$  ngưng dẫn nên  $T_1$  không ảnh hưởng đến hoạt động của mạch khi chân 4 có điện trở trị số nhỏ thích hợp nối mass thì  $T_1$  dẫn bão hòa đồng thời làm mạch OUTPUT cũng dẫn bão hòa và ngõ ra xuống thấp. Chân 4 được gọi là chân Reset có nghĩa là nó Reset IC

555 bắt chấp tình trạng ở các ngõ vào khác, do đó chân Reset dùng để kết thúc xung ra sớm khi cần. Nếu không dùng chức năng Reset thì nối chân 4 lên UCC để tránh mạch bị Reset do nhiễu.

7)- Transistor  $T_2$  là Transistor có cực C để hở nối ra chân 7 (Discharge = xả). Do cực B cực phân cực bởi mức điện áp ra  $\overline{Q}$  của F/F nên khi  $\overline{Q}$  ở mức cao thì  $T_2$  bão hòa và cực C của  $T_2$  coi như nối mass, lúc đóng mở ra chân 3 cũng ở mức thấp. Khi  $\overline{Q}$  ở mức thấp thì  $T_2$  ngưng dẫn cực C của  $T_2$  bị hở, lúc đó ngõ ra chân 3 có điện áp cao. Theo nguyên lý trên cực C của  $T_2$  ra chân 7 có thể làm ngõ ra phụ có mức điện áp giống mức điện áp của ngõ ra chân 4. Hình 4.10 là sơ đồ cấu trúc bên trong của IC 555 vẽ theo kiểu sơ đồ chức năng.

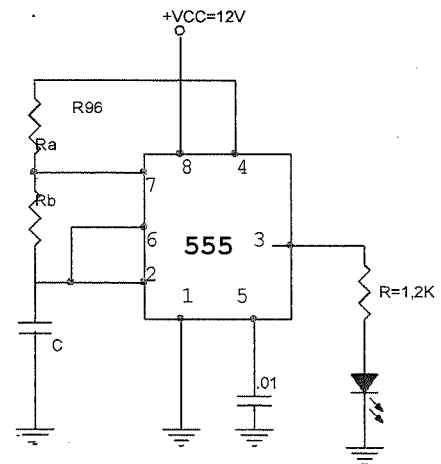
### Các mạch dao động.

#### a. Mạch phi ổn cơ bản.

Sơ đồ mạch hình 4.11 là ứng dụng của IC 555 là mạch đa hài phi ổn để tạo xung vuông.

Trong mạch chân ngưỡng (Threshoide) số 6 được nối với chân này (Trigger) số 2 nên hai chân này có chung điện áp là điện áp trên tụ C để so với điện áp chuẩn  $2/3 VCC$  và  $1/3 VCC$  bởi OP- AMP (1) và OP- AMP (2). Chân 5 có tụ nhỏ, 01 nối mass để lọc nhiễu tần số cao có thể làm ảnh hưởng điện áp chuẩn  $2/3 VCC$ .

Chân 4 nối nguồn + VCC nên không dùng chức năng Reset, chân 7 xả điện được nối vào giữa 2 điện



Hình 4.11: Mạch đa hài phi ổn

trở RA và RB tạo đường xả điện cho tụ. Ngõ ra chân 3 có điện trở giới hạn dòng 1,2 K và Led để biểu thị mức điện áp ra – chỉ có thể dùng trong trường hợp

tần số dao động có trị số thấp từ 20HZ trở xuống thì ở tần số cao hơn 40HZ trạng thái sáng + tắt của Led khó có thể nhận biết bằng mắt thường.

Để phân tích nguyên lý của mạch cần phối hợp mạch ứng dụng hình 4.51 và sơ đồ cấu trúc hình 4.52

Khi mới đóng điện tụ C bắt đầu nạp từ 0V lên nên:

-OP-AMP (1) có  $V_I^+ < V_I^-$  nên ngõ ra có  $V_{01} =$  mức thấp, ngõ R= 0 (mức thấp)

- OP-AMP (2) có  $V_I^+ > V_I^-$  nên ngõ ra có  $V_{02} =$  mức cao , ngõ S=1(mức cao )

Mạch F|F có ngõ S=1 nên  $\overline{Q}=1$  và  $Q = 0$  . Lúc đó ngõ ra chân 3 có  $V_0 \approx VCC$

(Do qua mạch đảo ) làm Led sáng .

Transistor  $T_2$  có  $V_{B2} = 0$  do  $\overline{Q}=0$  nên  $T_2$  ngưng dẫn để tụ C được nạp

điện Tụ C nạp điện qua  $R_A$  và  $R_B$  với hằng số thời gian khi nạp là:

$$\tau_{nạp} = (R_A + R_B) C$$

Khi điện áp trên tụ tăng đến mức 1/3 VCC thì OP-AMP (2) đổi trạng thái, ngõ ra có  $V_{02} =$  mức thấp, ngõ S = 0 ( mức thấp ). Khi S xuống mức thấp thì F|F không đổi trạng thái nên điện áp ngõ ra vẫn ở mức cao, Led vẫn sáng .

Khi điện áp trên tụ tăng đến mức 2/3 VCC thì OP-AMP (1) đổi trạng thái, ngõ ra có  $V_{01} =$  mức cao, ngõ R = 1.

Mạch F|F có ngõ R= 1 nên  $Q= 1$ .Lúc đó ra chân 3 có  $V_0 \approx 0V$  làm Led tắt. Khi ngõ  $Q=1$ . sẽ làm  $T_2$  dẫn bão hòa và chân 7 nối mass, làm tụ C không nạp tiếp điện áp được

mà phải xả điện qua  $R_B$  và Transistor  $T_2$  xuống mass. Tụ C xả điện qua  $R_B$  với hằng số thời gian là:

$$\tau_{xả} = R_B . C$$

Khi điện áp trên tụ - tức là điện áp chân 2 và chân 6- giảm xuống dưới 2/3 VCC thì OP-AMP(1) đổi trở lại trạng thái cũ là  $V_{01} =$  mức thấp, ngõ R = 0. Khi R xuống mức thấp thì F|F không đổi trạng thái nên điện áp ngõ ra vẫn ở mức thấp, Led vẫn tắt. Khi điện áp trên tụ giảm xuống đến mức 1/3 VCC thì OP-AMP(2) lại có  $V_I^+ > V_I^-$  nên ngõ ra có  $V_{02} =$  mức cao, ngõ  $S_1 = 1$ . Mạch F|F có ngõ S=1 và  $\overline{Q} = 0$  , ngõ ra chân 3 qua mạch đảo có  $V_0 \approx + VCC$  làm Led lại sáng, đồng thời lúc đó  $T_2$  mất phân cực do  $\overline{Q} = 0$  nên ngưng dẫn và chấm dứt giai đoạn xả điện của tụ. Như vậy mạch đã trở lại trạng thái ban đầu và tụ lại nạp điện từ mức 1/3 VCC nên đến 2/3 VCC , hiện tượng này sẽ tiếp diễn liên tục và tuần hoàn.

Lưu ý: Khi mới mở điện tụ C sẽ nạp điện từ 0V lên 2/3 VCC rồi sau đó tụ xả điện là 2/3 VCC xuống 1/2 VCC chứ không xả xuống 0v. Những chu kỳ sau tụ sẽ nạp từ 1/3VCC lên 2/3 chứ không nạp từ 0v nữa.

Thời gian tụ nạp là thời gian  $V_0 \approx + VCC$  , Led sáng. Thời gian tụ xả là thời gian  $V_0 \approx 0v$  , Led tắt.

Thời gian nạp và xả tụ được tính theo công thức:

$$\begin{aligned} \text{Thời gian nạp : } t_{nạp} &= 0,69 . \tau_{nạp} \\ &: t_{nạp} = 0,69 . ( R_A + R_B ) C \end{aligned}$$

$$\begin{aligned} \text{Thời gian xả : } t_{xả} &= 0,69 . \tau_{xả} \\ &. t_{xả} = 0,69 . R_B . C \end{aligned}$$

Điện áp ở ngõ ra chân 3 có dạng hình vuông với chu kỳ là:

$$\begin{aligned} T &= t_{nạp} + t_{xả} \\ T &= 0,69 ( R_A + 2R_B ) C \end{aligned}$$

Do thời gian nạp và thời gian xả không bằng nhau (  $t_{nạp} > t_{xả}$  ) nên tín hiệu hình

vuông ra không đối xứng. Tần số của tín hiệu hình vuông là:

$$f = \frac{1}{T} = \frac{1}{0,69(R_A + 2R_B)C}$$

**b. Mạch phi ổn đối xứng :**

Trong mạch phi ổn cơ bản , do thời gian nạp và thời gian xả của tụ không bằng nhau nên dạng điện áp ở ngõ ra không đối xứng.

$$\text{Ta có :} \quad t_{\text{nạp}} = 0,69 (R_A + R_B) . C$$

$$t_{\text{xả}} = 0,69 R_B . C$$

Để cho dạng sóng vuông ở ngõ ra đối xứng người ta có thể thực hiện bằng nhiều cách

Cách thứ 1: Chọn điện trở  $R_A$  có trị số rất nhỏ so với  $R_B$  thì lúc đó sai số giữa  $t_{\text{nạp}}$  và  $t_{\text{xả}}$  coi như không đáng kể. Điều này khó thực hiện nếu làm việc ở tần số cao. Điện trở  $R_A$  có trị số tối thiểu khoảng vài  $K\Omega$  thì  $R_B$  phải có trị số rất lớn khoảng vài trăm  $K\Omega$ . Với các trị số điện trở này thì tần số dao động không thể cao được.

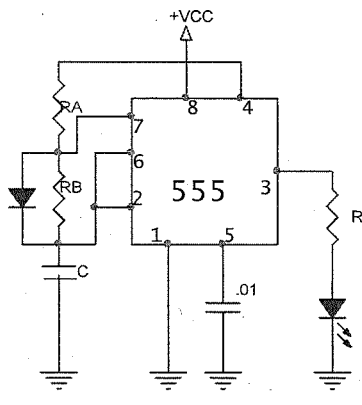
Cách thứ 2 : Dùng Diode  $D$  ghép song song  $R_B$  theo chiều hướng xuống như hình vẽ 4.12 khi có Diode  $D$ , thời gian tụ  $C$  nạp làm Diode  $D$  được phân cực thuận có điện trở rất nhỏ nên coi như nối tắt  $R_B$ . Thời gian nạp điện của tụ được tính theo công thức :

$$t_{\text{nạp}} \approx 0,69 R_A . C$$

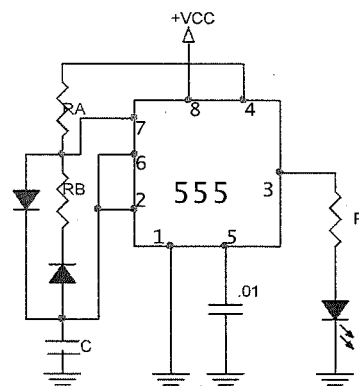
Khi tụ  $C$  xả điện thì Diode được phân cực ngược nên tụ vẫn xả điện qua  $R_B$ . Thời gian xả điện của tụ được tính theo công thức :  $t_{\text{xả}} = 0,69 R_B . C$

Nếu chọn trị số  $R_A$  bằng  $R_B$  thì mạch sẽ tạo ra tín hiệu hình vuông đối xứng .





Hình 4.12a



Hình 4.12b.

Thật ra trong mạch điện hình 4.12a khi tụ nạp dòng điện phải qua RA và điện trở thuận của Diode nên thời gian nạp vẫn lớn hơn thời gian xả một ít. Để cho tín hiệu thật đối xứng thì ghép thêm Diode nối tiếp với điện trở RB như hình 4.12b, như vậy cả hai trường hợp nạp và xả đều có Diode. Điều kiện của hai mạch trên là RB phải có trị số khá lớn so với điện trở thuận của Diode.

**c. Mạch phi ổn điều chỉnh tần số và chu trình làm việc:**

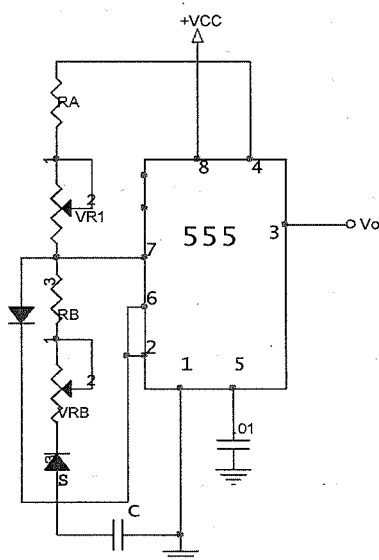
Hai yêu cầu thường có trong thiết kế của mạch đa hài phi ổn là:

Thay đổi tần số f mà vẫn giữ nguyên chu trình làm việc ( đối xứng )

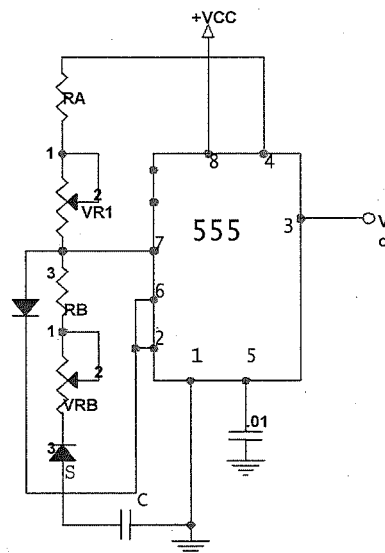
Thay đổi chu trình làm việc mà vẫn giữ nguyên tần số f.

Để có thể thay đổi tần số của tín hiệu hình vuông ra mà vẫn có tín hiệu đối xứng thì hai điện trở RA và RB phải được điều chỉnh sao cho cùng tăng hay giảm trị số. Lúc đó trong mạch điện có thêm hai biến trở VRA và VRB ghép nối tiếp như trong sơ đồ hình 4.46a.

Để thay đổi chu trình làm việc – tức là thay đổi tỉ lệ thời gian tín hiệu có điện áp cao và thời gian tín hiệu có điện áp thấp hay là thời gian nạp và thời gian xả của tụ – nhưng vẫn giữ nguyên tần số nghĩa là chu kỳ T bằng hằng số, hai điện trở RA và RB giảm cùng một giá trị thay đổi. Lúc đó trong mạch có thêm hai biến trở VRA và VRB ghép nối tiếp như sơ đồ hình 4.13b nhưng hai biến trở được điều chỉnh ngược hướng



Hình 4.13a



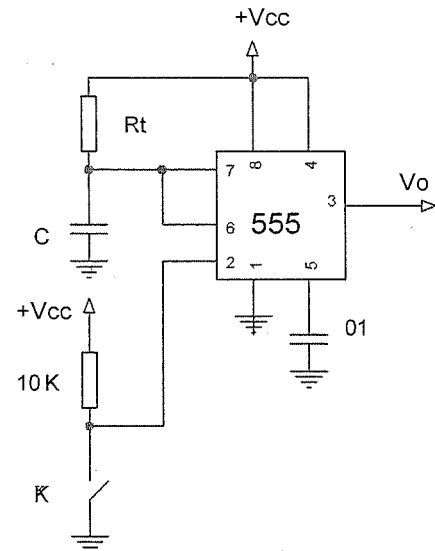
Hình 4.13b

## 2. Mạch đa hài đơn ổn dùng 555:

### a. Mạch đa hài đơn ổn cơ bản:

Để có thể phân tích nguyên lý của mạch đơn ổn một cách rõ ràng, để hiểu chúng ta sử dụng sơ đồ hình 4.14. Sơ đồ hình vẽ mạch áp dụng IC 555 làm mạch đơn ổn, sơ đồ hình 4.48 kết hợp với sơ đồ cấu trúc bên trong IC.

Trong mạch này chân ngưỡng số 6 và chân xả số 7 được nối vào điểm chung của mạch định thì RTC. Chân nhận xung kích số 2 được nối lên nguồn +VCC qua điện trở 10K sao cho chân này có điện áp lớn hơn  $1/3V_{CC}$ .



H  
Hình 4.14

Đặc điểm của mạch đơn ổn là khi có xung âm hẹp tác động tức thời ở ngõ vào Trigger chân hai mạch sẽ đổi trạng thái và tại ngõ ra chân 3 sẽ có xung dương ra. Độ rộng xung ở ngõ ra có thời gian dài hay ngắn tùy thuộc mạch định thì RTC, sau đó mạch sẽ trở lại trạng thái ban đầu.

Nguyên lý mạch đơn ổn được giải thích như sau:

Khi mở điện tụ C nối chân 6 và 7 xuống masse làm OP-AMP (1) có ngõ  $In^+$  nhỏ hơn ngõ  $In^-$  nên ngõ ra  $V_{01} = 0V$ , ngõ R ở mức thấp. Lúc đó OP-AMP (2) có ngõ  $In^+$  cũng nhỏ hơn ngõ  $In^-$  nên ngõ ra  $V_{02} = 0V$ , ngõ S cũng thấp. Mạch F|F có hai ngõ R và S đều ở mức thấp và nhờ cấu trúc của mạch chi tiết nên F|F có ngõ ra  $\bar{Q}$  ở mức cao, qua mạch đảo ngõ ra chân 3 sẽ có mức thấp gần  $0V$ . khi  $\bar{Q}$  ở mức cao tạo phân cực bão hòa cho  $T_2$  làm  $T_2$  dẫn nối chân 7 xuống mass, chân 6 cũng bị nối mass nên tụ C không nạp điện được, mạch sẽ ổn định ở trạng thái này nếu không có tác động khác ở bên ngoài.

Khi đóng khóa K sẽ có xung âm kích vào chân Trigger số 2 làm OP-AMP (2) đổi trạng thái ngõ S lên mức cao. Mức cao của ngõ S điều khiển làm F|F đổi trạng thái, làm ngõ  $\bar{Q}$  xuống mức thấp, ngõ ra qua mạch đảo sẽ tăng lên mức cao và xung dương ra. Lúc đó  $\bar{Q}$  ở mức thấp nên  $T_2$  ngưng dẫn để tụ C nạp điện qua  $R_T$ . Trong thời gian tụ C nạp điện mạch vẫn giữ trạng thái này nên ngõ ra tiếp tục ở ngõ cao.

Điện áp nạp trên tụ có trị số tăng theo hàm số mũ và khi điện áp đạt giá trị  $2/3 V_{CC}$  thì OP-AMP (1) đổi trạng thái, ngõ R tăng lên mức cao. Ngõ R có mức cao sẽ điều khiển F|F trở lại trạng thái cũ, ngõ  $\bar{Q}$  lên mức cao làm ngõ ra qua mạch đảo sẽ xuống mức thấp chấm dứt xung dương ra. Đồng thời lúc đó  $T_2$  được phân cực bão hòa nên chân 7 nối mass làm tụ C xả điện, mạch sẽ ổn định ở trạng thái này cho đến khi nào có xung âm khác tác động vào chân Trigger ( số 2 )

Thời gian xung dương ra tức là thời gian nạp điện từ  $0V$  lên  $2/3 V_{CC}$  được tính theo như sau:

Điện áp nạp trên tụ tăng theo hàm số mũ là:

$$VC = VCC (1 - e^{-\frac{t_x}{\tau}})$$

Trong đó  $\tau = RT \cdot C$

Thời gian tụ nạp được điện thế từ 0V lên  $2/3 VCC$  là  $t_x$  được tính bởi :

$$VC = VCC (1 - e^{-\frac{t_x}{\tau}}) = 2/3 VCC$$

$$\text{Suy ra : } (1 - e^{-\frac{t_x}{\tau}}) = \frac{2}{3} \text{ hay là } 1 - \frac{2}{3} = e^{-\frac{t_x}{\tau}}$$

$$\Rightarrow \frac{1}{3} = e^{-\frac{t_x}{\tau}} = \frac{1}{e^{\frac{t_x}{\tau}}}$$

$$\Rightarrow e^{\frac{t_x}{\tau}} = 3$$

Cuối cùng ta có hàm số ngược của hàm số mũ là Ln.

Như vậy :  $t_x = \tau \cdot \text{Ln}3$  ( $\text{Ln}3 = 1,1$ )

$$t_x = 1,1 RT \cdot C$$

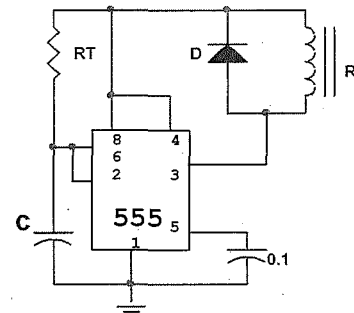
Bình thường chân 2 phải có điện áp lớn hơn  $1/3 VCC$ , khi có xung âm thì biên độ xung phải làm điện áp chân 2 nhỏ hơn  $1/3 VCC$ .

Khi vừa có xung âm ở chân 2 thì ngõ ra bắt đầu có xung dương và tụ C bắt đầu nạp điện

Thời gian xung dương ra  $t_x$  không tùy thuộc độ xung âm ở ngõ vào mà chỉ tùy thuộc hằng số thời gian  $\tau$  của mạch định thì. Nếu dùng biến trở thay VR thay cho RT ta có thể thay đổi độ rộng xung ra, cách khác là thay đổi tụ C bằng các điện dung có trị số khác nhau.

### b. Mạch trì hoãn dùng kiểu đơn ổn.

Mạch đơn ổn được ứng dụng rất rộng rãi trong lĩnh vực tự động điều khiển và đặc biệt là mạch trì hoãn. Trong thực tế người ta không cần tạo xung điều khiển cho vào chân số 2 (Trigger) mà mạch tự tạo xung khi mở điện ở ngõ ra cũng bắt đầu có xung ra. Mạch điện hình 4.49 là sơ đồ mạch tự tạo xung khi mở điện. Trong sơ đồ chân số 2 (Trigger) được nối đến chân số 6 (Threshold = thêm) nên sẽ có chung điện áp giữa mạch nạp RT.C để so với hai điện áp chuẩn trong IC là  $1/3 VCC$  và  $2/3 VCC$ .



Hình 4.15 Mạch trì hoãn kiểu đơn ổn

Khi mở điện tụ C bắt đầu nạp từ 0v lên nên OP- AMP (2) có ngõ  $In^+$  Lớn hơn ngõ  $In^-$  nên ngõ ra  $V_{02}$  ở mức cao, ngõ S cũng ở mức cao, mạch F/F có ngõ  $\bar{Q}$  ở mức thấp và ngõ ra của IC có  $V_0 \approx VCC$  có nghĩa là tức thời có xung ra. Lúc đó OP- AMP (1) có ngõ  $In^+$  nhỏ hơn  $In^-$  nên ngõ ra  $V_{01}$  ở mức thấp, ngõ R cũng ở mức thấp.

Khi tụ nạp đi đến mức  $1/3 VCC$  thì OP-AMP(2) đổi trạng thái, ngõ S xuống mức thấp nhưng mạch F/F cũng đổi trạng thái, ngõ  $\bar{Q}$  tăng lên mức cao làm ngõ ra của IC giảm xuống mức thấp  $V_0 = 0v$  và chấm dứt xung ở ngõ ra

Thời gian có xung ra hay độ rộng xung chính là thời gian tụ C nạp từ 0v đến  $2/3 VCC$  và cũng được tính theo công thức:

$$t_x = 1,1 RT.C$$

Trong mạch này chân 7 (Discharge = xả điện) để trống không nối vào mạch nạp RT.

C nên tụ sẽ không xả điện và mạch sẽ giữ mãi ở trạng thái này. Muốn có xung ra tiếp thì phải tắt điện rồi mở lại.

Trị số điện trở RT và tụ C được giới hạn trong khoảng :

$$RT = 10 \text{ K}\Omega \div 14 \text{ M}\Omega$$

$$C = 100\text{pF} \div 1000 \text{ MF}$$

Với các trị số trên Rt và C, mạch đơn ổn có thể cho ra xung có độ rộng ngắn nhất vài  $\mu\text{s}$  đến vài giờ.

Như vậy, khi mở điện rơ-le RY không có điện do  $V_0 \approx V_{CC}$ , sau khi chấm dứt xung tức là sau thời gian tx thì rơ-le RY có điện vì  $V_0 \approx 0\text{v}$ . khi rơ-le có điện sẽ đóng hay mở các tiếp điểm để điều khiển mạch khác thường là mạch công suất.

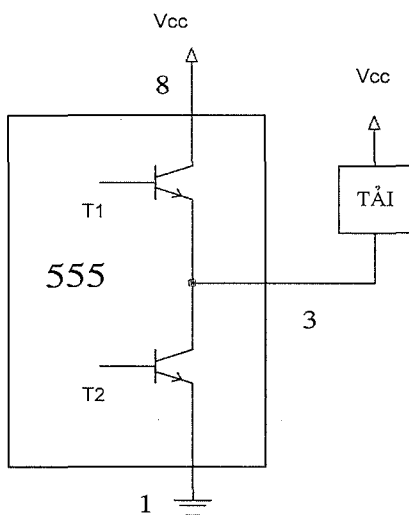
Thật ra IC555 và họ IC định thì của nó có ứng dụng rất đa dạng, trong chương này chỉ giới thiệu hai ứng dụng cơ bản nhất của nó là mạch đa hài phi ổn và đa hài đơn ổn

### c. IC555 giao tiếp với các loại tải:

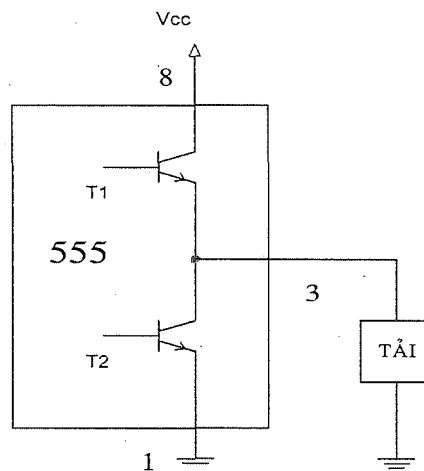
IC555 có thể giao tiếp với nhiều loại tải khác nhau và tùy trường hợp mỗi loại tải đều có thể mắc theo 2 cách

- Tải được cấp điện khi ngõ ra có điện áp thấp. Lúc đó IC sẽ cấp dòng điện cho tải theo chiều từ nguồn qua IC rồi ra tải. Dòng điện tải trường hợp này được gọi là I nhận

- Tải được cấp điện khi ngõ ra có điện áp cao. Lúc đó IC sẽ cấp dòng điện cho tải theo chiều từ nguồn qua IC rồi ra tải. Dòng điện tải trường hợp này được gọi là I nguồn.



Hình 4-16a : Điện áp ra mức thấp



Hình 4-17b: Điện áp ra mức cao

Khả năng cung cấp dòng điện và điện áp của IC555 như sau:

\* Điện áp ở mức thấp :

Với  $V_{CC} = 15\text{V}$ . Điện áp  $V_0$  chính là  $U_{CE2}$  khi T<sub>2</sub> bão hòa

$$I_L = 10\text{mA} \Rightarrow U_0 = 0,1\text{v}$$

$$I_L = 50\text{mA} \Rightarrow U_0 = 0,4\text{v}$$

$$I_L = 100\text{mA} \Rightarrow U_0 = 2\text{v}$$

$$I_L = 200\text{mA} \Rightarrow U_0 = 2,5$$

\* Điện áp ra ở mức cao:

Với  $U_{CC} = 15\text{v}$ . Điện áp  $U_0$  chính là  $U_{CC} - U_{CE_2}$ . Khi  $T_1$  bão hòa

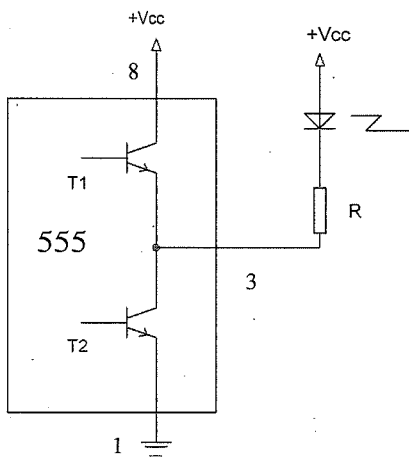
$$I_L = 100\text{mA} \Rightarrow U_0 = 13,3\text{v} \Rightarrow U_{CE_2} = 1,7\text{v}$$

$$I_L = 200\text{mA} \Rightarrow U_0 = 12,5\text{v} \Rightarrow U_{CE_2} = 2,5\text{v}$$

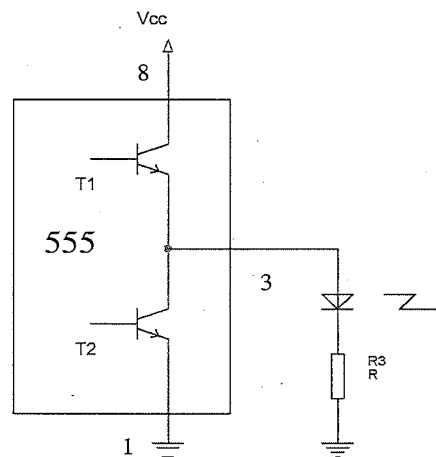
\* **Tải là Led**

Nếu tải là Led thì phải dùng điện trở ghép nối tiếp với Led để giới hạn dòng qua Led

Tùy theo cách mắc tải mà dòng điện qua Led có công thức tính khác nhau



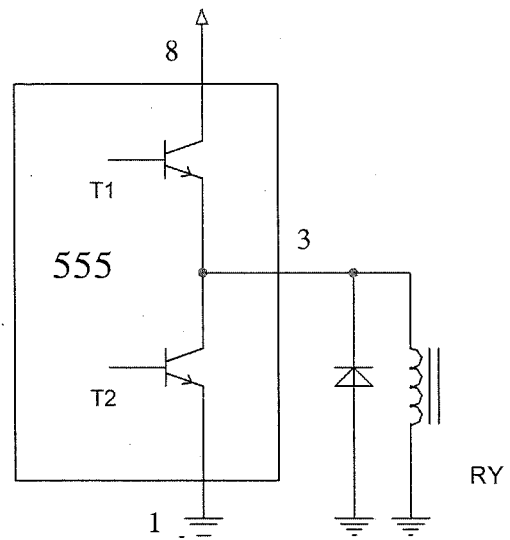
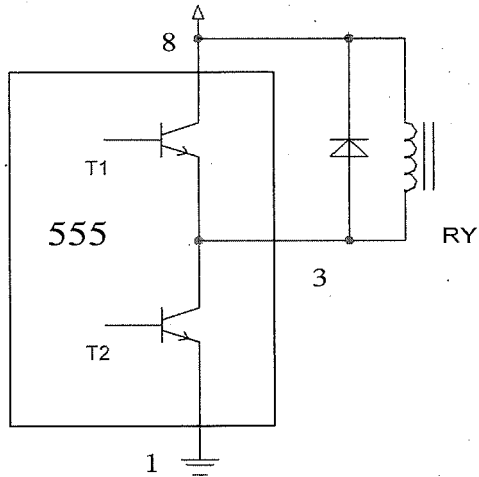
Hình 4-17a: Điện áp ra mức thấp



Hình 4-17b: Điện áp ra mức cao

**\* Tải là Rơ-le :**

IC 555 có thể giao tiếp với các loại Rơ-le điều khiển . Các loại Rơ-le này có dòng thường nhỏ dưới 100mA và điện áp cũng thường ở mức thấp như 6v -12v-24v .



# CHƯƠNG 1: ĐẠI CƯƠNG

Nhu cầu về định lượng trong quan hệ giữa con người với nhau, nhất là trong những trao đổi thương mại, đã có từ khi xã hội hình thành. Đã có rất nhiều cố gắng trong việc tìm kiếm các vật dụng, các ký hiệu . . . dùng cho việc định lượng này như các que gỗ, vỏ sò, số La mã . . . Hiện nay số Ả rập tỏ ra có nhiều ưu điểm khi được sử dụng trong định lượng, tính toán. . . .

Việc sử dụng hệ thống số hằng ngày trở nên quá quen thuộc khiến chúng ta có thể đã quên đi sự hình thành và các qui tắc để viết các con số.

Chương này nhắc lại một cách sơ lược nguyên lý của việc viết số và giới thiệu các hệ thống số khác ngoài hệ thống thập phân quen thuộc, phương pháp biến đổi qua lại của các số trong các hệ thống khác nhau. Chúng ta sẽ đặc biệt quan tâm đến hệ thống nhị phân là hệ thống được dùng trong lãnh vực điện tử-tin học như là một phương tiện để giải quyết các vấn đề mang tính logic.

Phần cuối của chương sẽ giới thiệu các loại mã thông dụng để chuẩn bị cho các chương kế tiếp.

## Nguyên lý của việc viết số

Một số được viết bằng cách đặt kề nhau các **ký hiệu**, được chọn trong một tập hợp xác định. Mỗi ký hiệu trong một số được gọi là **số mã** (số hạng, digit).

Thí dụ, trong hệ thống thập phân (ơ số 10) tập hợp này gồm 10 ký hiệu rất quen thuộc, đó là các con số từ 0 đến 9:

$$S_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

Khi một số gồm nhiều số mã được viết, giá trị của các số mã tùy thuộc vị trí của nó trong số đó. Giá trị này được gọi là **trọng số** của số mã.

Thí dụ số 1998 trong hệ thập phân có giá trị xác định bởi triển khai theo đa thức của

1  
0  
:

$$1998_{10} = 1 \times 10^3 + 9 \times 10^2 + 9 \times 10^1 + 8 \times 10^0 = 1000 + 900 + 90 + 8$$

Trong triển khai, số mũ của đa thức chỉ vị trí của một ký hiệu trong một số với qui ước



vị trí của hàng đơn vị là 0, các vị trí liên tiếp về phía trái là 1, 2, 3, ... . Nếu có phần lẻ, vị trí

đầu tiên sau dấu phẩy là -1, các vị trí liên tiếp về phía phải là -2, -3, ... .

Ta thấy, số 9 đầu tiên (sau số 1) có trọng số là 900 trong khi số 9 thứ hai chỉ là 90.

Có thể nhận xét là với 2 ký hiệu giống nhau trong hệ 10, ký hiệu đứng trước có trọng số gấp 10 lần ký hiệu đứng ngay sau nó. Điều này hoàn toàn đúng cho các hệ khác, thí dụ, đối với hệ nhị phân ( cơ số 2) thì tỉ lệ này là 2.

Tổng quát, một hệ thống số được gọi là **hệ b** sẽ gồm b ký hiệu trong một tập hợp:

$$S_b = \{S_0, S_1, S_2, \dots, S_{b-1}\}$$

Một số N được viết:

$$N = (a_n a_{n-1} a_{n-2} \dots a_i \dots a_0, a_{-1} a_{-2} \dots a_{-m})_b \text{ với } a_i \in S_b$$

Sẽ có giá trị:

$$N = a_n b^n + a_{n-1} b^{n-1} + a_{n-2} b^{n-2} + \dots + a_i b^i + \dots + a_0 b^0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \dots + a_{-m} b^{-m} \\ = \sum_{i=-m}^n a_i b^i$$

$a_i b^i$  chính là trọng số của một ký hiệu trong  $S_b$  ở vị trí thứ i.

## Các hệ thống số

### *Hệ cơ số 10 (thập phân, Decimal system)*

Hệ thập phân là hệ thống số rất quen thuộc, gồm 10 số mã như nói trên.

Dưới đây là vài ví dụ số thập phân:

$$N = 1998_{10} = 1 \times 10^3 + 9 \times 10^2 + 9 \times 10^1 + 8 \times 10^0 = 1 \times 1000 + 9 \times 100 + 9 \times 10 + 8 \times 1$$

$$N = 3,14_{10} = 3 \times 10^0 + 1 \times 10^{-1} + 4 \times 10^{-2} = 3 \times 1 + 1 \times 1/10 + 4 \times 1/100$$

### *Hệ cơ số 2 (nhị phân, Binary system)*

Hệ nhị phân gồm hai số mã trong tập hợp

$$S_2 = \{0, 1\}$$

Mỗi số mã trong một số nhị phân được gọi là một **bit** (viết tắt của binary digit).

Số N trong hệ nhị phân:

$$N = (a_n a_{n-1} a_{n-2} \dots a_i \dots a_0, a_{-1} a_{-2} \dots a_{-m})_2 \quad (\text{với } a_i \in S_2)$$

Có giá trị là:

$$N = a_n 2^n + a_{n-1} 2^{n-1} + \dots + a_i 2^i + \dots + a_0 2^0 + a_{-1} 2^{-1} + a_{-2} 2^{-2} + \dots + a_{-m} 2^{-m}$$

$a_n$  là bit có **trọng số lớn nhất**, được gọi là bit **MSB** (Most significant bit) và  $a_{-m}$  là bit có **trọng số nhỏ nhất**, gọi là bit **LSB** (Least significant bit).

$$\text{Thí dụ: } N = 1010,1_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} = 10,5_{10}$$

### *Hệ cơ số 8 (bát phân, Octal system)*

Hệ bát phân gồm tám số trong tập hợp

$$S_8 = \{0, 1, 2, 3, 4, 5, 6, 7\}.$$

Số N trong hệ bát phân:

$$N = (a_n a_{n-1} a_{n-2} \dots a_i \dots a_0, a_{-1} a_{-2} \dots a_{-m})_8 \quad (\text{với } a_i \in S_8)$$

Có giá trị là:

$$N = a_n 8^n + a_{n-1} 8^{n-1} + a_{n-2} 8^{n-2} + \dots + a_i 8^i + \dots + a_0 8^0 + a_{-1} 8^{-1} + a_{-2} 8^{-2} + \dots + a_{-m} 8^{-m}$$

$$\text{Thí dụ: } N = 1307,1_8 = 1 \times 8^3 + 3 \times 8^2 + 0 \times 8^1 + 7 \times 8^0 + 1 \times 8^{-1} = 711,125_{10}$$

### Hệ cơ số 16 (thập lục phân, Hexadecimal system)

Hệ thập lục phân được dùng rất thuận tiện để con người giao tiếp với máy tính, hệ này gồm mười sáu số trong tập hợp

$$S_{16} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F\}$$

(A tương đương với  $10_{10}$ , B =  $11_{10}$ , ..., F =  $15_{10}$ ).

Số N trong hệ thập lục phân:

$$N = (a_n a_{n-1} a_{n-2} \dots a_i \dots a_0, a_{-1} a_{-2} \dots a_{-m})_{16} \quad (\text{với } a_i \in S_{16})$$

Có giá trị là:

$$N = a_n 16^n + a_{n-1} 16^{n-1} + a_{n-2} 16^{n-2} + \dots + a_i 16^i + \dots + a_0 16^0 + a_{-1} 16^{-1} + a_{-2} 16^{-2} + \dots + a_{-m} 16^{-m}$$

Người ta thường dùng chữ H (hay h) sau con số để chỉ số thập lục phân.

$$\begin{aligned} \text{Thí dụ: } N = 20EA,8H &= 20EA,8_{16} = 2 \times 16^3 + 0 \times 16^2 + 14 \times 16^1 + 10 \times 16^0 + 8 \times 16^{-1} \\ &= 4330,5_{10} \end{aligned}$$

### Biến đổi qua lại giữa các hệ thống số

Khi đã có nhiều hệ thống số, việc xác định giá trị tương đương của một số trong hệ này so với hệ kia là cần thiết. Phần sau đây cho phép ta biến đổi qua lại giữa các số trong bất cứ hệ nào sang bất cứ hệ khác trong các hệ đã được giới thiệu.

#### Đổi một số từ hệ b sang hệ 10

Để đổi một số từ hệ b sang hệ 10 ta triển khai trực tiếp đa thức của b

Một số N trong hệ b:

$$N = (a_n a_{n-1} a_{n-2} \dots a_i \dots a_0, a_{-1} a_{-2} \dots a_{-m})_b \quad \text{với } a_i \in S_b$$

Có giá trị tương đương trong hệ 10 là:

$$N = a_n b^n + a_{n-1} b^{n-1} + \dots + a_i b^i + \dots + a_0 b^0 + a_{-1} b^{-1} + a_{-2} b^{-2} + \dots + a_{-m} b^{-m}$$

Thí dụ:

\* Đổi số  $10110,11_2$  sang hệ 10

$$10110,11_2 = 1 \times 2^4 + 0 + 1 \times 2^2 + 1 \times 2 + 0 + 1 \times 2^{-1} + 1 \times 2^{-2} = 22,75_{10}$$

\* Đổi số  $4BE,ADH$  sang hệ 10

$$4BE,ADH = 4 \times 16^2 + 11 \times 16^1 + 14 \times 16^0 + 10 \times 16^{-1} + 13 \times 16^{-2} = 1214,675_{10}$$

#### Đổi một số từ hệ 10 sang hệ b

Đây là bài toán tìm một dãy ký hiệu cho số N viết trong hệ b.

Tổng quát, một số N cho ở hệ 10, viết sang hệ b có dạng:

$$N = (a_n a_{n-1} \dots a_0, a_{-1} a_{-2} \dots a_{-m})_b = (a_n a_{n-1} \dots a_0)_b + (0, a_{-1} a_{-2} \dots a_{-m})_b$$

Trong đó

$$(a_n a_{n-1} \dots a_0)_b = PE(N) \text{ là phần nguyên của } N$$

$$\text{và } (0, a_{-1} a_{-2} \dots a_{-m})_b = PF(N) \text{ là phần lẻ của } N$$

Phần nguyên và phần lẻ được biến đổi theo hai cách khác nhau:

☒ **Phần nguyên:**

Giá trị của phần nguyên xác định nhờ triển khai:

$$PE(N) = a_n b^n + a_{n-1} b^{n-1} + \dots + a_1 b^1 + a_0 b^0$$

Hay có thể viết lại

$$PE(N) = (a_n b^{n-1} + a_{n-1} b^{n-2} + \dots + a_1) b + a_0$$

Với cách viết này ta thấy nếu chia PE(N) cho b, ta được **thương số** là PE'(N) =  $(a_n b^{n-1} + a_{n-1} b^{n-2} + \dots + a_1)$  và **số dư** là  $a_0$ .

Vậy **số dư của lần chia thứ nhất** này chính là **số mã có trọng số nhỏ nhất** ( $a_0$ ) của phần nguyên.

Lặp lại bài toán chia PE'(N) cho b:

$$PE'(N) = a_n b^{n-1} + a_{n-1} b^{n-2} + \dots + a_1 = (a_n b^{n-2} + a_{n-1} b^{n-3} + \dots + a_2) b + a_1$$

Ta được **số dư thứ hai**, chính là số mã có trọng số lớn hơn kế tiếp ( $a_1$ ) và thương số là PE''(N) =  $a_n b^{n-2} + a_{n-1} b^{n-3} + \dots + a_2$ .

Tiếp tục bài toán chia thương số có được với b, cho đến khi được **số dư của phép chia cuối cùng**, đó chính là số mã có trọng số lớn nhất ( $a_n$ )

☒ **Phần lẻ:**

Giá trị của phần lẻ xác định bởi:

$$PF(N) = a_{-1} b^{-1} + a_{-2} b^{-2} + \dots + a_{-m} b^{-m}$$

Hay viết lại

$$PF(N) = b^{-1} (a_{-1} + a_{-2} b^{-1} + \dots + a_{-m} b^{-m+1})$$

Nhân PF(N) với b, ta được :  $bPF(N) = a_{-1} + (a_{-2} b^{-1} + \dots + a_{-m} b^{-m+1}) = a_{-1} + PF'(N)$ .

Vậy lần nhân thứ nhất này ta được **phần nguyên của phép nhân**, chính là **số mã có trọng số lớn nhất của phần lẻ** ( $a_{-1}$ ) (số  $a_{-1}$  này có thể vẫn là số 0).

PF'(N) là phần lẻ xuất hiện trong phép nhân.

Tiếp tục nhân PF'(N) với b, ta tìm được  $a_{-2}$  và **phần lẻ PF''(N)**.

Lặp lại bài toán nhân phần lẻ với b cho đến khi kết quả có phần lẻ bằng không, ta sẽ tìm được dãy số  $(a_{-1} a_{-2} \dots a_{-m})$ .

**Chú ý:** Phần lẻ của số N khi đổi sang hệ b có thể gồm vô số số hạng (do kết quả của phép nhân luôn khác 0), điều này có nghĩa là ta không tìm được một số trong hệ b có **giá trị đúng bằng phần lẻ của số thập phân**, vậy tùy theo yêu cầu về độ chính xác khi chuyển đổi mà người ta lấy một số số hạng nhất định.

Thí dụ:

\* Đổi  $25,3_{10}$  sang hệ nhị phân

$$\text{Phần nguyên: } 25 : 2 = 12 \text{ dư } 1 \quad \Rightarrow a_0 = 1$$

$$12 : 2 = 6 \text{ dư } 0 \quad \Rightarrow a_1 = 0$$

$$6 : 2 = 3 \text{ dư } 0 \quad \Rightarrow a_2 = 0$$

$$3 : 2 = 1 \text{ dư } 1 \quad \Rightarrow a_3 = 1$$

thương số cuối cùng là 1 cũng chính là bit  $a_4$ :

$$\Rightarrow a_4 = 1$$

Vậy PE(N) = 11001

$$\text{Phần lẻ: } 0,3 * 2 = 0,6 \quad \Rightarrow a_{-1} = 0$$

$$0,6 * 2 = 1,2 \quad \Rightarrow a_{-2} = 1$$

$$0,2 * 2 = 0,4 \quad \Rightarrow a_{-3} = 0$$

$$0,4 * 2 = 0,8 \quad \Rightarrow a_{-4} = 0$$

$$0,8 * 2 = 1,6 \quad \Rightarrow a_{-5} = 1 \dots$$

Nhận thấy kết quả của các bài toán nhân luôn khác không, do phần lẻ của lần nhân cuối cùng là 0,6, đã lặp lại kết quả của lần nhân thứ nhất, như vậy bài toán không thể kết thúc với kết quả đúng bằng 0,3 của hệ 10.

Giả sử bài toán yêu cầu lấy 5 số lẻ thì ta có thể dừng ở đây và

$$PF(N) = 0,01001.$$

Kết quả cuối cùng là:

$$25,3_{10} = 11001,01001_2$$

\* Đổi  $1376,85_{10}$  sang hệ thập lục phân

$$\text{Phần nguyên: } 1376 : 16 = 86 \text{ số dư } = 0 \Rightarrow a_0 = 0$$

$$86 : 16 = 5 \text{ số dư } = 6 \Rightarrow a_1 = 6 \ \& \Rightarrow a_2 = 5$$

$$1376_{10} = 560H$$

$$\text{Phần lẻ: } 0,85 * 16 = 13,6 \Rightarrow a_{-1} = 13_{10} = DH$$

$$0,6 * 16 = 9,6 \Rightarrow a_{-2} = 9$$

$$0,6 * 16 = 9,6 \Rightarrow a_{-3} = 9$$

$$\text{Nếu chỉ cần lấy 3 số lẻ: } 0,85_{10} = 0,D99H$$

Và kết quả cuối cùng:

$$1376,85_{10} = 560,D99H$$

### Đổi một số từ hệ b sang hệ $b^k$ và ngược lại

Từ cách triển khai đa thức của số N trong hệ b, ta có thể nhóm thành từng k số hạng từ dấu phẩy về hai phía và đặt thành thừa số chung

$$N = a_n b^n + \dots + a_5 b^5 + a_4 b^4 + a_3 b^3 + a_2 b^2 + a_1 b^1 + a_0 b^0 + a_{-1} b^{-1} + a_{-2} b^{-2} + a_{-3} b^{-3} \dots + a_{-m} b^{-m}$$

Để dễ hiểu, chúng ta lấy thí dụ  $k=3$ , N được viết lại bằng cách nhóm từng 3 số hạng, kể từ dấu phẩy về 2 phía

$$N = \dots + (a_5 b^2 + a_4 b^1 + a_3 b^0) b^3 + (a_2 b^2 + a_1 b^1 + a_0 b^0) b^0 + (a_{-1} b^2 + a_{-2} b^1 + a_{-3} b^0) b^{-3} + \dots$$

Phần chứa trong mỗi dấu ngoặc luôn luôn nhỏ hơn  $b^3$ , vậy số này tạo nên một số trong hệ  $b^3$  và lúc đó được biểu diễn bởi ký hiệu tương ứng trong hệ này.

Thật vậy, số N có dạng:

$$N = \dots + A_2 B^2 + A_1 B^1 + A_0 B^0 + A_{-1} B^{-1} + \dots$$

Trong đó:

$$B = b^3 (B^0 = b^0; B^1 = b^3; B^2 = b^6; B^{-1} = b^{-3} \dots)$$

$$A_2 = a_5 b^2 + a_4 b^1 + a_3 b^0 = b^3 (a_5 b^{-1} + a_4 b^{-2} + a_3 b^{-3}) < B = b^3$$

$$A_1 = a_2 b^2 + a_1 b^1 + a_0 b^0 = b^3 (a_2 b^{-1} + a_1 b^{-2} + a_0 b^{-3}) < B = b^3$$

$$A_0 = a_{-1} b^2 + a_{-2} b^1 + a_{-3} b^0 = b^3 (a_{-1} b^{-1} + a_{-2} b^{-2} + a_{-3} b^{-3}) < B = b^3$$

Các số  $A_i$  luôn luôn nhỏ hơn  $B = b^3$  như vậy nó chính là một phần tử của tập hợp số tạo nên hệ  $B = b^3$

Ta có kết quả biến đổi tương tự cho các hệ số k khác.

Tóm lại, để đổi một số từ hệ b sang hệ  $b^k$ , từ dấu phẩy đi về hai phía, ta nhóm từng k số hạng, giá trị của mỗi k số hạng này (tính theo hệ b) chính là số trong hệ  $b^k$ .

Thí dụ:

\* Đổi số  $N = 10111110101, 01101_2$  sang hệ  $8 = 2^3$

Từ dấu phẩy, nhóm từng 3 số hạng về hai phía (nếu cần, thêm số 0 vào ở nhóm đầu và cuối để đủ 3 số hạng mà không làm thay đổi giá trị của số N):

$$N = 010\ 111\ 110\ 101, 011\ 010_2$$

Ghi giá trị tương ứng của các số 3 bit, ta được số N trong hệ 8

$$N = 2\ 7\ 6\ 5, 3\ 2\ 8$$

\* Đổi số N trên sang hệ  $16 = 2^4$

Cũng như trên nhưng nhóm từng 4 số hạng

$$N = 0101\ 1111\ 0101, 0110\ 1000_2$$

$$N = 5\ F\ 5, 6\ 8_{16}$$

Từ kết quả của phép đổi số từ hệ b sang hệ  $b^k$ , ta có thể suy ra cách biến đổi ngược một cách dễ dàng: Thay mỗi số hạng của số trong hệ  $b^k$  bằng một số gồm k số hạng trong hệ b.

Thí dụ để đổi số  $N = 5\ F5, 68_{16}$  (hệ  $2^4$ ) sang hệ nhị phân (2) ta dùng 4 bit để viết cho mỗi số hạng của số này:

$$N = 0101\ 1111\ 0101, 0110\ 1000_2$$

### **Đổi một số từ hệ $b^k$ sang hệ $b^p$**

Qua trung gian của hệ b, ta có thể đổi từ hệ  $b^k$  sang hệ  $b^p$ . Muốn đổi số N từ hệ  $b^k$  sang hệ  $b^p$ , trước nhất đổi số N sang hệ b rồi từ hệ b tiếp tục đổi sang hệ  $b^p$ .

Thí dụ:

- Đổi số  $1234,67_8$  sang hệ 16

$$1234,67_8 = 001\ 010\ 011\ 100,110\ 111_2 = 0010\ 1001\ 1100,1101\ 1100_2 = 29C,DCH$$

- Đổi số ABCD,EFH sang hệ 8

$$ABCD,EFH = 1010\ 1011\ 1100\ 1101,1110\ 1111_2 = 1\ 010\ 101\ 111\ 001\ 101,111\ 011\ 110_2 = 125715,736_8$$

Dưới đây là bảng kê các số đầu tiên trong các hệ khác nhau:

Thập phân	Nhi phân	Bát phân	Thập lục phân	Thập phân	Nhi phân	Bát phân	Thập lục phân
0	0	0	0	13	1101	15	D
1	1	1	1	14	1110	16	E
2	10	2	2	15	1111	17	F
3	11	3	3	16	10000	20	10
4	100	4	4	17	10001	21	11
5	101	5	5	18	10010	22	12
6	110	6	6	19	10011	23	12
7	111	7	7	20	10100	24	14
8	1000	10	8	21	10101	25	15
9	1001	11	9	22	10110	26	16
10	1010	12	A	23	10111	17	17
11	1011	13	B	24	11000	30	18
12	1100	14	C	25	11001	31	19

Bảng 1.1

### Các phép tính trong hệ nhị phân

Các phép tính trong hệ nhị phân được thực hiện tương tự như trong hệ thập phân, tuy nhiên cũng có một số điểm cần lưu ý

### Phép cộng

Là phép tính làm cơ sở cho các phép tính khác.

Khi thực hiện phép cộng cần lưu ý:

$$0 + 0 = 0 ;$$

$$0 + 1 = 1 ;$$

$$1 + 1 = 0 \text{ nhớ } 1 \text{ (đem qua bit cao hơn).}$$

Ngoài ra nếu cộng nhiều số nhị phân cùng một lúc ta nên nhớ :

- Nếu số bit 1 chẵn, kết quả là 0;

- Nếu số bit 1 lẻ kết quả là 1

- Và cứ 1 cặp số 1 cho 1 số nhớ (bỏ qua số 1 dư, thí dụ với 5 số 1 ta kể là 2 cặp)

**Thí dụ:** Tính  $011 + 101 + 011 + 011$

$$\begin{array}{r} 11 \leftarrow \text{số nhớ} \\ 111 \leftarrow \text{số nhớ} \\ 011 \\ + 101 \\ 011 \\ 011 \\ \hline 1110 \end{array}$$

### Phép trừ

Cần lưu ý:

$$0 - 0 = 0 ;$$

$$1 - 1 = 0 ;$$

$$1 - 0 = 1 ;$$

$$0 - 1 = 1 \text{ nhớ } 1 \text{ cho bit cao hơn}$$

**Thí dụ:** Tính  $1011 - 0101$

$$\begin{array}{r} 1 \leftarrow \text{số nhớ} \\ 1011 \\ - 0101 \\ \hline 0110 \end{array}$$

### Phép nhân

Cần lưu ý:

$$0 \times 0 = 0 ;$$

$$0 \times 1 = 0 ;$$

$$1 \times 1 = 1$$

**Thí dụ:** Tính  $1101 \times 101$

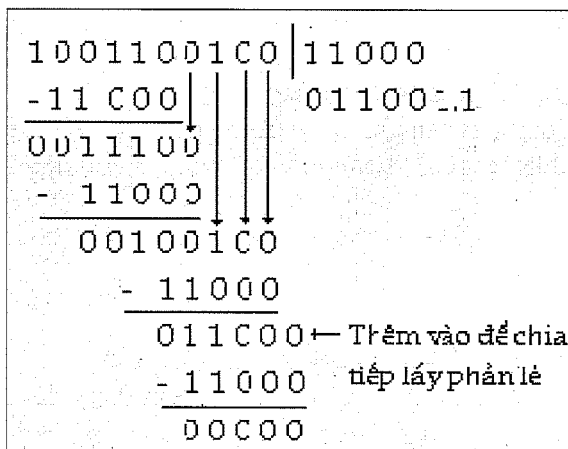
$$\begin{array}{r} 1101 \\ \times 101 \\ \hline 1101 \\ 0000 \\ 1101 \\ \hline 1101 \end{array}$$

1 0 0 0 0 1

### Phép chia

**Thí dụ:** Chia 1001100100 cho 11000

Lần chia đầu tiên, 5 bit của số bị chia nhỏ hơn số chia nên ta được kết quả là 0, sau đó ta lấy 6 bit của số bị chia để chia tiếp (tương ứng với việc dịch phải số chia 1 bit trước khi thực hiện phép trừ)



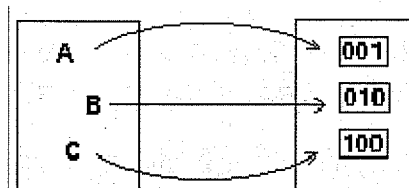
Kết quả :  $(11001.1)_2 = (25.5)_{10}$

### Mã hóa

#### Tổng quát

Mã hóa là gán một ký hiệu cho một đối tượng để thuận tiện cho việc thực hiện một yêu cầu cụ thể nào đó.

Một cách toán học, mã hóa là một phép áp một đối một từ một **tập hợp nguồn** vào một tập hợp khác gọi là **tập hợp đích**.



(H 1.1)

Tập hợp nguồn có thể là tập hợp các số, các ký tự, dấu, các lệnh dùng trong truyền dữ liệu... và tập hợp đích thường là tập hợp chứa các tổ hợp thứ tự của các số nhị phân.

Một tổ hợp các số nhị phân tương ứng với một số được gọi là **từ mã**. Tập hợp các từ mã được tạo ra theo một qui luật cho ta một **bộ mã**. Việc chọn một bộ mã tùy vào mục đích sử dụng.

Thí dụ để biểu diễn các chữ và số, người ta có **mã ASCII** (American Standard Code for Information Interchange), **mã Baudot**, **EBCDIC**... Trong truyền dữ liệu ta có **mã dò lỗi, dò và sửa lỗi, mật mã**....

Vấn đề ngược lại mã hóa gọi là **giải mã**.

Cách biểu diễn các số trong các hệ khác nhau cũng có thể được xem là một hình thức mã hóa, đó là các mã thập phân, nhị phân, thập lục phân . . . và việc chuyển từ mã này sang mã khác cũng thuộc loại bài toán mã hóa.

Trong kỹ thuật số ta thường dùng các mã sau đây:

### ***Mã BCD (Binary Coded Decimal)***

Mã BCD dùng **số nhị phân 4 bit** có giá trị tương đương thay thế cho **từng số hạng** trong số thập phân.

Thí dụ:

Số  $625_{10}$  có mã BCD là 0110 0010 0101.

Mã BCD dùng rất thuận lợi : mạch điện tử đọc các số BCD và hiển thị ra bằng đèn bảy đoạn (led hoặc LCD) hoàn toàn giống như con người đọc và viết ra số thập phân.

### ***Mã Gray***

Mã Gray hay còn gọi là mã **cách khoảng đơn vị**.

Nếu quan sát thông tin ra từ một máy đếm đang đếm các sự kiện tăng dần từng đơn vị, ta sẽ được các số nhị phân dần dần thay đổi. Tại thời điểm đang quan sát có thể có những lỗi rất quan trọng. Thí dụ giữa số 7(0111) và 8 (1000), các phần tử nhị phân đều phải thay đổi trong quá trình đếm, nhưng sự giao hoán này không bắt buộc xảy ra đồng thời, ta có thể có các trạng thái liên tiếp sau:

0111 → 0110 → 0100 → 0000 → 1000

Trong một quan sát ngắn các kết quả thấy được khác nhau. Để tránh hiện tượng này, người ta cần mã hóa mỗi số hạng sao cho hai số liên tiếp chỉ khác nhau một phần tử nhị phân (1 bit) gọi là mã **cách khoảng đơn vị** hay mã **Gray**.

Tính kề nhau của các tổ hợp mã Gray (tức các mã liên tiếp chỉ khác nhau một bit) được dùng rất có hiệu quả để **rút gọn hàm logic tới mức tối giản**.

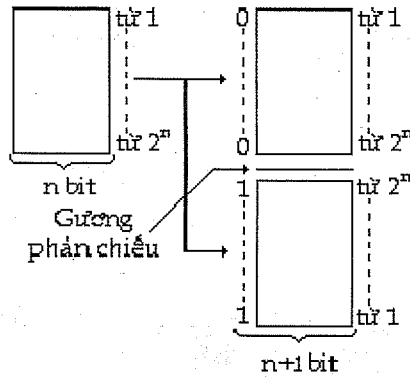
Ngoài ra, mã Gray còn được gọi là mã **phản chiếu** (do tính đối xứng của các số hạng trong tập hợp mã, giống như phản chiếu qua gương)

Người ta có thể thiết lập mã Gray bằng cách dựa vào tính đối xứng này:

- Giả sử ta đã có tập hợp  $2^n$  từ mã của số  $n$  bit thì có thể suy ra tập hợp  $2^{n+1}$  từ mã của số  $(n+1)$  bit bằng cách:

- Viết ra  $2^n$  từ mã theo thứ tự từ nhỏ đến lớn
- Thêm số 0 vào trước tất cả các từ mã đã có để được một phần của tập hợp từ mã mới
- Phần thứ hai của tập hợp gồm các từ mã giống như phần thứ nhất nhưng trình bày theo thứ tự ngược lại (giống như phản chiếu qua gương) và phía trước thêm vào số 1 thay vì số 0 (H 1.2).

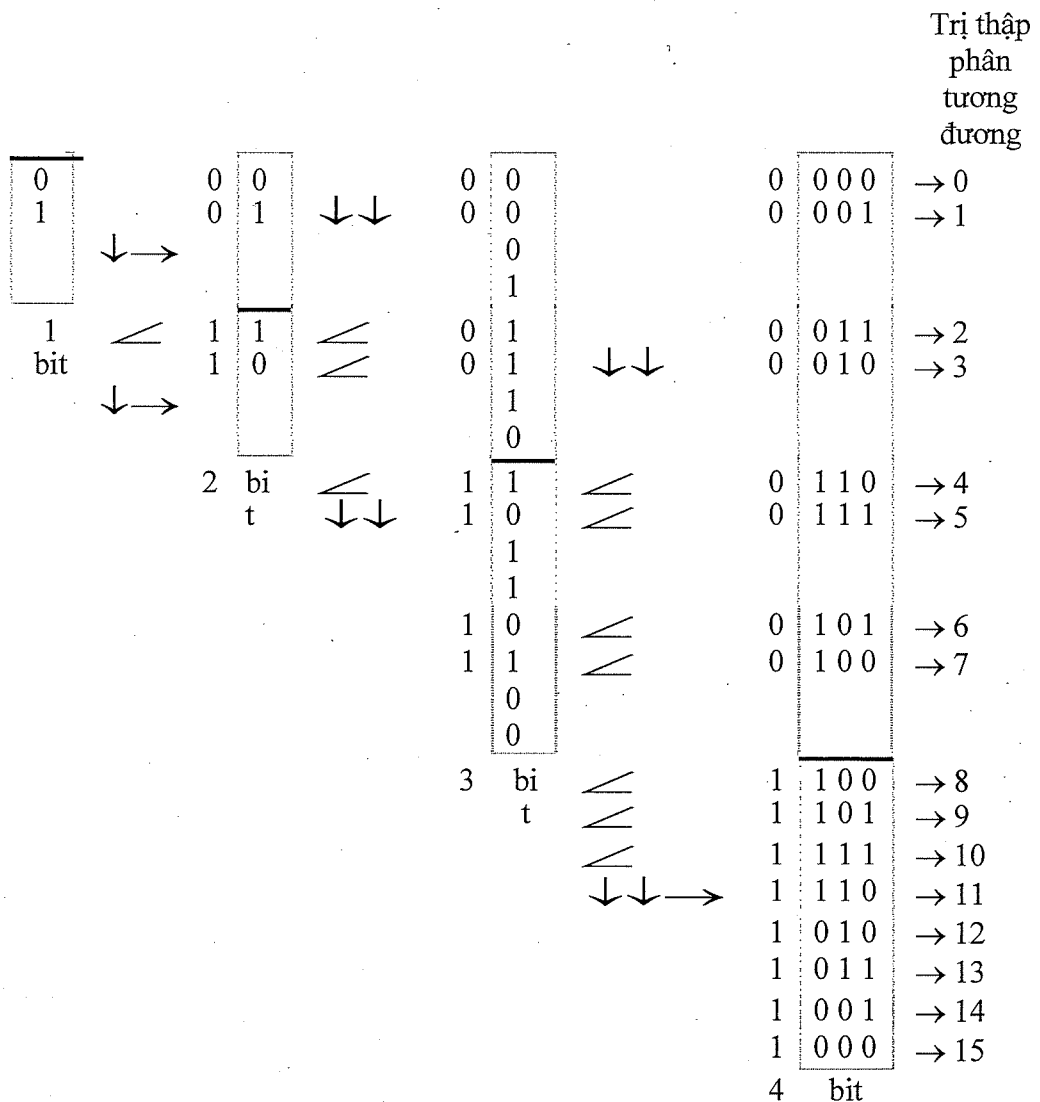




(H 1.2)

Để thiết lập mã Gray của số nhiều bit ta có thể thực hiện các bước liên tiếp từ tập hợp đầu tiên của số một bit (gồm hai bit 0, 1).

Dưới đây là các bước tạo mã Gray của số 4 bit. Cột bên phải của bảng mã 4 bit cho giá trị tương đương trong hệ thập phân của mã Gray tương ứng (H 1.3).



(H 1.3)

Nhận xét các bảng mã của các số Gray (1 bit, 2 bit, 3 bit và 4 bit) ta thấy các số gần nhau luôn luôn khác nhau một bit, ngoài ra, trong từng bộ mã, các số đối xứng nhau qua gương cũng khác nhau một bit.

### Bài Tập

1. Đổi các số thập phân dưới đây sang hệ nhị phân và hệ thập lục phân :

a/ 12    b/ 24    c/ 192    d/ 2079    e/ 15492  
f/ 0,25    g/ 0,375    h/ 0,376    i/ 17,150    j/ 192,1875

2. Đổi sang hệ thập phân và mã BCD các số nhị phân sau đây:

a/ 1011    b/ 10110    c/ 101,1    d/ 0,1101  
e/ 0,001    f/ 110,01    g/ 1011011    h/ 10101101011

3. Đổi các số thập lục phân dưới đây sang hệ 10 và hệ 8:

a/ FF    b/ 1A    c/ 789    d/ 0,13    e/ ABCD,EF

4. Đổi các số nhị phân dưới đây sang hệ 8 và hệ 16:

a/ 111001001,001110001    b/ 10101110001,00011010101  
c/ 1010101011001100,1010110010101    d/ 1111011100001,01010111001

5. Mã hóa số thập phân dưới đây dùng mã BCD :

a/ 12    b/ 192    c/ 2079    d/ 15436    e/ 0,375    f/ 17,250

## CHƯƠNG 2: FLIP - FLOP

Năm 1854 Georges Boole, một triết gia đồng thời là nhà toán học người Anh cho xuất bản một tác phẩm về lý luận logic, nội dung của tác phẩm đặt ra những mệnh đề mà để trả lời người ta chỉ phải dùng một trong hai từ đúng (có, yes) hoặc sai (không, no).

Tập hợp các thuật toán dùng cho các mệnh đề này hình thành môn Đại số Boole. Đây là môn toán học dùng hệ thống số nhị phân mà ứng dụng của nó trong kỹ thuật chính là các mạch logic, nền tảng của kỹ thuật số.

Chương này không có tham vọng trình bày lý thuyết Đại số Boole mà chỉ giới hạn trong việc giới thiệu các **hàm logic cơ bản** và các **tính chất cần thiết** để giúp sinh viên hiểu vận hành của một hệ thống logic.

### HÀM LOGIC CƠ BẢN

#### *Một số định nghĩa*

- **Trạng thái logic:** trạng thái của một thực thể. Xét về mặt logic thì một thực thể chỉ tồn tại ở một trong hai trạng thái. Thí dụ, đối với một bóng đèn ta chỉ quan tâm nó đang ở trạng thái nào: tắt hay cháy. Vậy tắt / cháy là 2 trạng thái logic của nó.

- **Biến logic** dùng đặc trưng cho các trạng thái logic của các thực thể. Người ta biểu diễn biến logic bởi một ký hiệu (chữ hay dấu) và nó chỉ nhận 1 trong 2 giá trị: 0 hoặc 1.

Thí dụ trạng thái logic của một công tắc là đóng hoặc mở, mà ta có thể đặc trưng bởi trị 1 hoặc 0.

- **Hàm logic** diễn tả bởi một nhóm biến logic liên hệ nhau bởi các phép toán logic. Cũng như biến logic, hàm logic chỉ nhận 1 trong 2 giá trị: 0 hoặc 1 tùy theo các điều kiện liên quan đến các biến.

Thí dụ, một mạch gồm một nguồn hiệu thế cấp cho một bóng đèn qua hai công tắc mắc nối tiếp, bóng đèn chỉ cháy khi cả 2 công tắc đều đóng. Trạng thái của bóng đèn là một hàm theo 2 biến là trạng thái của 2 công tắc.

Gọi A và B là tên biến chỉ công tắc, công tắc đóng ứng với trị 1 và hở ứng với trị 0. Y là hàm chỉ trạng thái bóng đèn, 1 chỉ đèn cháy và 0 khi đèn tắt. Quan hệ giữa hàm Y và các biến A, B được diễn tả nhờ bảng sau:

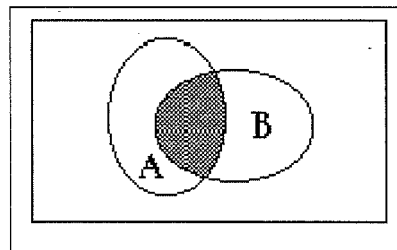
A	B	Y=f(A,B)
0 (hở)	0 (hở)	0 (tắt)
0 (hở)	1 (đóng)	0 (tắt)
1 (đóng)	0 (hở)	0 (tắt)
1 (đóng)	1 (đóng)	1 (cháy)

### Biểu diễn biến và hàm logic

#### Giản đồ Venn

Còn gọi là giản đồ Euler, đặc biệt dùng trong lãnh vực tập hợp. Mỗi biến logic chia không gian ra 2 vùng không gian con, một vùng trong đó giá trị biến là đúng (hay=1), và vùng còn lại là vùng phụ trong đó giá trị biến là sai (hay=0).

**Thí dụ:** Phân giao nhau của hai tập hợp con A và B (gạch chéo) biểu diễn tập hợp trong đó A và B là đúng (A AND B) (H 2.1)



(H 2.1)

**B  
ả  
n  
g  
sự  
th  
ật**

Nếu hàm có n biến, bảng sự thật có n+1 cột và  $2^n + 1$  hàng. Hàng đầu tiên chỉ tên biến và hàm, các hàng còn lại trình bày các tổ hợp của n biến trong  $2^n$  tổ hợp có thể có. Các cột đầu ghi giá trị của biến, cột cuối cùng ghi giá trị của hàm tương ứng với tổ hợp biến trên cùng hàng (gọi là trị riêng của hàm).

**Thí dụ:** Hàm OR của 2 biến A, B:  $f(A,B) = (A \text{ OR } B)$  có bảng sự thật tương ứng.

A	B	f(A,B) = A OR B
0	0	0
0	1	1
1	0	1
1	1	1

#### Bảng Karnaugh

Đây là cách biểu diễn khác của bảng sự thật trong đó mỗi hàng của bảng sự thật được thay thế bởi một ô mà tọa độ (gồm hàng và cột) xác định bởi tổ hợp đã cho của biến.

Bảng Karnaugh của n biến gồm  $2^n$  ô. Giá trị của hàm được ghi tại mỗi ô của bảng.

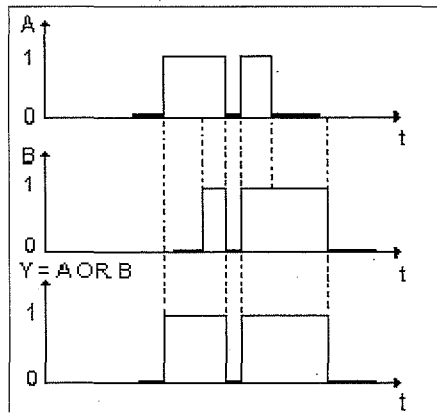
Bảng Karnaugh rất thuận tiện để đơn giản hàm logic bằng cách nhóm các ô lại với nhau.  
**Thí dụ:** Hàm OR ở trên được diễn tả bởi bảng Karnaugh sau đây

A \ B	0	1
<b>0</b>	0	1
<b>1</b>	1	1

## Giản đồ thời gian

Dùng để diễn tả quan hệ giữa các hàm và biến theo thời gian, đồng thời với quan hệ logic.

**Thí dụ:** Giản đồ thời gian của hàm OR của 2 biến A và B, tại những thời điểm có một (hoặc 2) biến có giá trị 1 thì hàm có giá trị 1 và hàm chỉ có giá trị 0 tại những thời điểm mà cả 2 biến đều bằng 0.



(H 2.2)

### Qui ước

Khi nghiên cứu một hệ thống logic, cần xác định qui ước logic. Qui ước này không được thay đổi trong suốt quá trình nghiên cứu.

Người ta dùng 2 mức điện thế thấp và cao để gán cho 2 trạng thái logic 1 và 0.

Qui ước **logic dương** gán điện thế thấp cho logic 0 và điện thế cao cho logic 1

Qui ước **logic âm** thì ngược lại.

### Hàm logic cơ bản (Các phép toán logic)

**Hàm NOT (đảo, bù) :**

$$Y = \bar{A}$$

Bảng sự thật

A	$Y = \bar{A}$
0	1
1	0

**Hàm AND [tích logic, toán tử (.)] :**  $Y = A.B$

Bảng sự thật

A	B	$Y=A.B$
0	0	0
0	1	0
1	0	0
1	1	1

0	0	0
0	1	0
1	0	0

1	1	1
---	---	---

*Nhận xét:* Tính chất của hàm AND có thể được phát biểu như sau:

- Hàm AND của 2 (hay nhiều) biến chỉ có giá trị 1 khi tất cả các biến đều bằng 1 hoặc
- Hàm AND của 2 (hay nhiều) biến có giá trị 0 khi có một biến bằng 0.

**Hàm OR [tổng logic, toán tử (+)] :  $Y = A + B$**

Bảng sự thật

A	B	$Y=A + B$
0	0	0
0	1	1
1	0	1
1	1	1

*Nhận xét:* Tính chất của hàm OR có thể được phát biểu như sau:

- Hàm OR của 2 (hay nhiều) biến chỉ có giá trị 0 khi tất cả các biến đều bằng 0 hoặc
- Hàm OR của 2 (hay nhiều) biến có giá trị 1 khi có một biến bằng 1.

**Hàm EX-OR (OR loại trừ)  $Y = A \oplus B$**

Bảng sự thật

A	B	$Y = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

*Nhận xét:* Một số tính chất của hàm EX - OR:

- Hàm EX - OR của 2 biến chỉ có giá trị 1 khi hai biến khác nhau và ngược lại. Tính chất này được dùng để so sánh 2 biến.
- Hàm EX - OR của 2 biến cho phép thực hiện cộng hai số nhị phân 1 bit mà không quan tâm tới số nhớ.
- Từ kết quả của hàm EX-OR 2 biến ta suy ra bảng sự thật cho hàm 3 biến

A	B	C	$Y = A \oplus B \oplus C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



- Trong trường hợp 3 biến (và suy rộng ra cho nhiều biến), hàm EX - OR có giá trị 1 khi số biến bằng 1 là số lẻ. Tính chất này được dùng để nhận dạng một chuỗi dữ liệu có số bit 1 là chẵn hay lẻ trong thiết kế mạch phát chẵn lẻ.

### Tính chất của các hàm logic cơ bản:

#### Tính chất cơ bản:

- ♦ Có một phần tử trung tính duy nhất cho mỗi toán tử (+) và (.):

$$A + 0 = A; 0 \text{ là phần tử trung tính của hàm OR}$$

$$A \cdot 1 = A; 1 \text{ là phần tử trung tính của hàm AND}$$

- ♦ Tính giao hoán:

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

- ♦ Tính phối hợp:

$$(A + B) + C = A + (B + C) = A + B + C \quad (A \cdot B) \cdot C =$$

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

- ♦ Tính phân bố:

$$\text{- Phân bố đối với phép nhân: } A \cdot (B + C) = A \cdot B + A \cdot C$$

$$\text{- Phân bố đối với phép cộng: } A + (B \cdot C) = (A + B) \cdot (A + C)$$

Phân bố đối với phép cộng là một tính chất đặc biệt của phép toán logic

- ♦ Không có phép tính lũy thừa và thừa số:

$$A + A + \dots + A = A \cdot A \cdot A \dots$$

$$\dots A = A$$

- ♦ Tính bù:

$$\overline{\overline{A}} = A$$

$$A + \overline{A} = 1$$

$$A \cdot \overline{A} = 0$$

#### Tính song đôi (duality):

Tất cả biểu thức logic vẫn đúng khi [thay phép toán (+) bởi phép (.) và 0 bởi 1] hay ngược lại. Điều này có thể chứng minh dễ dàng cho tất cả biểu thức ở trên.

$$\begin{aligned} \text{Thí dụ:} \quad A + B &= B + A & \Leftrightarrow & \quad A \cdot B = B \cdot A \\ A + \overline{A \cdot B} &= A + B & \Leftrightarrow & \quad A(A + B) = A \cdot B \\ A + 1 &= 1 & \Leftrightarrow & \quad A \cdot 0 = 0 \end{aligned}$$

#### Định lý De Morgan

Định lý De Morgan được phát biểu bởi hai biểu thức:

đảo.

A g và nhân nhờ vào phép

+

B

+

C

=

A

·

B

·

C

≡

A

·

B

·

C

=

A

+

B

+

C

Định  
h lý  
De  
Mo  
rga  
n  
cho  
phé  
p  
biế  
n  
đổi  
qua  
lại  
giữ  
a  
hai  
phé  
p  
cộn

Định lý De Morgan được chứng minh bằng cách lập bảng sự thật cho tất cả trường hợp có thể có của các biến A, B, C với các hàm AND, OR và NOT của chúng.

### Sự phụ thuộc lẫn nhau của các hàm logic cơ bản

Định lý De Morgan cho thấy các hàm logic không độc lập với nhau, chúng có thể biến đổi qua lại, sự biến đổi này cần có sự tham gia của hàm NOT. Kết quả là ta có thể dùng hàm (AND và NOT) hoặc (OR và NOT) để diễn tả tất cả các hàm.

Thí dụ:

Chỉ dùng hàm AND và NOT để diễn tả hàm sau:  $Y = A.B + B.C + \bar{A}.C$

Chỉ cần đảo hàm Y hai lần, ta được kết quả:

$$Y = Y = \overline{\overline{A.B + B.C + \bar{A}.C}} = \overline{\overline{A.B} . \overline{B.C} . \overline{\bar{A}.C}}$$

Nếu dùng hàm OR và NOT để diễn tả hàm trên làm như sau:

$$Y = \overline{\overline{A.B} . \overline{B.C} . \overline{\bar{A}.C}} = \overline{\overline{A} + \overline{B} + \overline{B} + \overline{C} + \overline{A} + \overline{C}}$$

### CÁC DẠNG CHUẨN CỦA HÀM LOGIC

Một hàm logic được biểu diễn bởi một tổ hợp của những tổng và tích logic.

♦ Nếu biểu thức là tổng của những tích, ta có dạng tổng

*Thí dụ:*  $f(X, Y, Z) = XY + XZ + YZ$

♦ Nếu biểu thức là tích của những tổng, ta có dạng tích

*Thí dụ:*  $f(X, Y, Z) = (X + Y).(X + Z).(Y + \bar{Z})$

Một hàm logic được gọi là hàm chuẩn nếu mỗi số hạng chứa đầy đủ các biến, ở dạng nguyên hay dạng đảo của chúng.

*Thí dụ:*  $f(X, Y, Z) = XYZ + X\bar{Y}Z + XY\bar{Z}$  là một tổng chuẩn.

Mỗi số hạng của tổng chuẩn được gọi là **minterm**.

$f(X, Y, Z) = (X + Y + Z).(X + \bar{Y} + Z).(\bar{X} + Y + \bar{Z})$  là một tích chuẩn.

Mỗi số hạng của tích chuẩn được gọi là **maxterm**.

Phần sau đây cho phép chúng ta viết ra một hàm dưới dạng tổng chuẩn hay tích chuẩn khi có bảng sự thật diễn tả hàm đó.

### Dạng tổng chuẩn

Để có được hàm logic dưới dạng chuẩn, ta áp dụng các **định lý triển khai của Shanon**.

Dạng tổng chuẩn có được từ triển khai theo **định lý Shanon thứ nhất**:

Tất cả các hàm logic có thể triển khai theo một trong những biến dưới dạng tổng của hai tích như sau:

$$f(A, B, \dots, Z) = A.f(1, B, \dots, Z) + \bar{A}.f(0, B, \dots, Z) \quad (1)$$

Hệ thức (1) có thể được chứng minh rất dễ dàng bằng cách lần lượt cho A bằng 2 giá trị 0 và 1, ta có kết quả là 2 vế của (1) luôn luôn bằng nhau. Thật vậy

Cho A=0:  $f(0, B, \dots, Z) = 0.f(1, B, \dots, Z) + 1.f(0, B, \dots, Z) = f(0, B, \dots, Z)$

Cho A=1:  $f(1, B, \dots, Z) = 1.f(1, B, \dots, Z) + 0.f(0, B, \dots, Z) = f(1, B, \dots, Z)$



$$f(A,B) = A.f(1,B) + \bar{A}.f(0,B)$$

Mỗi hàm trong hai hàm vừa tìm được lại có thể triển khai theo biến B

$$f(1,B) = B.f(1,1) + \bar{B}.f(1,0) \quad \& \quad f(0,B) = B.f(0,1) + \bar{B}.f(0,0)$$

Vậy:  $f(A,B) = AB.f(1,1) + \bar{A}.B.f(0,1) + A\bar{B}.f(1,0) + \bar{A}\bar{B}.f(0,0)$

$f(i,j)$  là giá trị riêng của  $f(A,B)$  khi  $A=i$  và  $B=j$  trong bảng sự thật của hàm.

Với 3 biến, trị riêng của  $f(A, B, C)$  là  $f(i, j, k)$  khi  $A=i, B=j$  và  $C=k$  ta được:

$$f(A,B,C) = \bar{A}.B.C.f(1,1,1) + \bar{A}.B.\bar{C}.f(1,1,0) + A.\bar{B}.C.f(1,0,1) + A.\bar{B}.\bar{C}.f(1,0,0) + \bar{A}.B.C.f(0,1,1) + \bar{A}.B.\bar{C}.f(0,1,0) + A.\bar{B}.C.f(0,0,1) + A.\bar{B}.\bar{C}.f(0,0,0)$$

Khi triển khai hàm 2 biến ta được tổng của  $2^2 = 4$  số hạng

Khi triển khai hàm 3 biến ta được tổng của  $2^3 = 8$  số hạng

Khi triển khai hàm n biến ta được tổng của  $2^n$  số hạng

Mỗi số hạng là tích của một tổ hợp biến và một trị riêng của hàm. Hai trường hợp có thể xảy ra:

- Giá trị riêng = 1, số hạng thu gọn lại chỉ còn các biến:

$$\bar{A} . \bar{B} . C . f(0,0,1) = \bar{A} . \bar{B} . C \text{ nếu } f(0,0,1) = 1$$

- Giá trị riêng = 0, tích bằng 0 :

$$\bar{A} . \bar{B} . \bar{C} . f(0,0,0) = 0 \text{ nếu } f(0,0,0) = 0$$

và số hạng này biến mất trong biểu thức của tổng chuẩn.

Thí dụ:

Cho hàm 3 biến A,B,C xác định bởi bảng sự thật:

Hàng	A	B	C	Z=f(A,B,C)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Với hàm Z cho như trên ta có các trị riêng  $f(i, j, k)$  xác định bởi:

$$f(0,0,1) = f(0,1,0) = f(0,1,1) = f(1,0,1) = f(1,1,1) = 1$$

$$f(0,0,0) = f(1,0,0) = f(1,1,0) = 0$$

- Hàm Z có trị riêng  $f(0,0,1)=1$  tương ứng với các giá trị của tổ hợp biến ở hàng (1) là  $A=0, B=0$  và  $C=1$  đồng thời, vậy  $\bar{A} . \bar{B} . C$  là một số hạng trong tổng chuẩn

- Tương tự với các tổ hợp biến tương ứng với các hàng (2), (3), (5) và (7) cũng là các số hạng của tổng chuẩn, đó là các tổ hợp:  $\bar{A} . \bar{B} . C, \bar{A} . B . C, A . \bar{B} . C$  và  $A . B . C$

- Với các hàng còn lại (hàng 0,4,6), trị riêng của  $f(A,B,C) = 0$  nên không xuất hiện trong triển khai.

Tóm lại ta có:  $Z = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot B \cdot C$

### - Ý nghĩa của định lý Shannon thứ nhất:

Nhắc lại tính chất của các hàm AND và OR:  $b_1 \cdot b_2 \dots b_n = 1$  khi  $b_1, b_2, \dots, b_n$  đồng thời bằng 1 và để  $a_1 + a_2 + \dots + a_p = 1$  chỉ cần ít nhất một biến  $a_1, a_2, \dots, a_p$  bằng 1

Trở lại thí dụ trên, biểu thức logic tương ứng với hàng 1 ( $A=0, B=0, C=1$ ) được viết  $\bar{A} \cdot \bar{B} \cdot C = 1$  vì  $\bar{A} = 1, B = 1, C = 1$  đồng thời.

Biểu thức logic tương ứng với hàng 2 là  $\bar{A} \cdot B \cdot \bar{C} = 1$  vì  $A=0 (\bar{A} = 1), B=1, C=0 (\bar{C} = 1)$  đồng thời

Tương tự, với các hàng 3, 5 và 7 ta có các kết quả:  $A \cdot \bar{B} \cdot C, A \cdot B \cdot \bar{C}$  và  $A \cdot B \cdot C$

Như vậy, trong thí dụ trên

$Z =$  hàng 1 + hàng 2 + hàng 3 + hàng 5 + hàng 7

$Z = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot \bar{C} + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C$

Tóm lại, từ một hàm cho dưới dạng bảng sự thật, ta có thể viết ngay biểu thức của hàm dưới dạng tổng chuẩn như sau:

- Số số hạng của biểu thức bằng số giá trị 1 của hàm thể hiện trên bảng sự thật

- **Mỗi số hạng trong tổng chuẩn là tích của tất cả các biến tương ứng với tổ hợp mà hàm có trị riêng bằng 1, biến được giữ nguyên khi có giá trị 1 và được đảo nếu giá trị của nó = 0.**

## Dạng tích chuẩn

Đây là dạng của hàm logic có được từ triển khai theo định lý Shannon thứ hai:

Tất cả các hàm logic có thể triển khai theo một trong những biến dưới dạng tích của hai tổng như sau:

$$f(A, B, \dots, Z) = [\bar{A} + f(1, B, \dots, Z)] \cdot [A + f(0, B, \dots, Z)] \quad (2)$$

Cách chứng minh định lý Shannon thứ hai cũng giống như đã chứng minh định lý Shannon thứ nhất.

Với hai biến, hàm  $f(A, B)$  có thể triển khai theo biến A

$$f(A, B) = [\bar{A} + f(1, B)] \cdot [A + f(0, B)]$$

Mỗi hàm trong hai hàm vừa tìm được lại có thể triển khai theo biến B

$$f(1, B) = [\bar{B} + f(1, 1)] \cdot [B + f(1, 0)] \quad \& \quad f(0, B) = [\bar{B} + f(0, 1)] \cdot [B + f(0, 0)]$$

$$f(A, B) = \bar{A} + [\bar{B} + f(1, 1)] \cdot [B + f(1, 0)] \leftarrow \cdot \bar{A} + [B + f(0, 1)] \cdot [B + f(0, 0)] \leftarrow$$

$$\text{Vậy: } f(A, B) = [\bar{A} + \bar{B} + f(1, 1)] \cdot [\bar{A} + B + f(1, 0)] \cdot [A + B + f(0, 1)] \cdot [A + B + f(0, 0)]$$

Cũng như dạng chuẩn thứ nhất,  $f(i, j)$  là giá trị riêng của  $f(A, B)$  khi  $A=i$  và  $B=j$  trong bảng sự thật của hàm.

Với hàm 3 biến:

$$f(A, B, C) = [\bar{A} + \bar{B} + \bar{C} + f(1, 1, 1)] \cdot [\bar{A} + \bar{B} + C + f(1, 1, 0)] \cdot [\bar{A} + B + \bar{C} + f(1, 0, 1)] \cdot [\bar{A} + B + C + f(1, 0, 0)] \cdot [A + \bar{B} + \bar{C} + f(0, 1, 1)] \cdot [A + \bar{B} + C + f(0, 1, 0)] \cdot [A + B + \bar{C} + f(0, 0, 1)] \cdot [A + B + C + f(0, 0, 0)]$$

Số số hạng trong triển khai n biến là  $2^n$ . Mỗi số hạng là tổng (OR) của các biến và trị riêng của hàm.

- Nếu trị riêng bằng 0 số hạng được rút gọn lại chỉ còn các biến (0 là trị trung tính của phép cộng logic)

$$A + B + C + f(0, 0, 0) = A + B + C \quad \text{nếu } f(0, 0, 0) = 0$$

- Nếu trị riêng bằng 1, số hạng triển khai = 1

$$A + B + C + f(0,0,1) = 1 \quad \text{nếu } f(0,0,1) = 1$$

và biến mất trong biểu thức của tích chuẩn.  
Lấy lại thí dụ trên:

Hàng	A	B	C	Z=f(A,B,C)
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Các trị riêng của hàm đã nêu ở trên.

- Hàm Z có giá trị riêng  $f(0,0,0) = 0$  tương ứng với các giá trị của biến ở hàng 0 là  $A=B=C=0$  đồng thời, vậy  $A+B+C$  là một số hạng trong tích chuẩn.

- Tương tự với các hàng (4) và (6) ta được các tổ hợp  $\bar{A}+B+C$  và  $\bar{A}+\bar{B}+C$ .

- Với các hàng còn lại (hàng 1, 2, 3, 5, 7), trị riêng của  $f(A,B,C) = 1$  nên không xuất hiện trong triển khai.

Tóm lại, ta có:  $Z = (A + B + C).(\bar{A} + B + C).(\bar{A} + \bar{B} + C)$

- Ý nghĩa của định lý thứ hai:

Nhắc lại tính chất của các hàm AND và OR: Để  $b_1.b_2.... b_n = 0$  chỉ cần ít nhất một biến trong  $b_1, b_2, ..., b_n = 0$  và  $a_1 + a_2 + ... + a_p = 0$  khi các biến  $a_1, a_2, ..., a_p$  đồng thời bằng 0.

Như vậy trong thí dụ trên:

$$Z = (\text{hàng 0}).(\text{hàng 4}).(\text{hàng 6})$$

$$Z = (A + B + C).(\bar{A} + B + C).(\bar{A} + \bar{B} + C)$$

Thật vậy, ở hàng 0 tất cả biến = 0:  $A=0, B=0, C=0$  đồng thời nên có thể viết  $(A+B+C) =$

0. Tương tự cho hàng (4) và hàng (6).

Tóm lại,

Biểu thức tích chuẩn gồm các thừa số, mỗi thừa số là tổng các biến tương ứng với tổ hợp có giá trị riêng =0, một biến giữ nguyên nếu nó có giá trị 0 và được đảo nếu có giá trị 1. Số thừa số của biểu thức bằng số số 0 của hàm thể hiện trên bảng sự thật.

### Đổi từ dạng chuẩn này sang dạng chuẩn

**khác:** Nhờ định lý De Morgan, hai định lý trên có thể chuyển đổi qua lại. Trở lại thí dụ trên, thêm cột Z vào bảng sự thật

Hàng	A	B	C	Z=f(A,B,C)	$\bar{Z}$
------	---	---	---	------------	-----------



0	0	0	0	0	1
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	0	1
7	1	1	1	1	0

Diễn tả  $\bar{Z}$  theo dạng tổng chuẩn:

$$\bar{Z} = \bar{A}BC + A\bar{B}C + ABC$$

Lấy đảo hai vế:

$$Z = \overline{\bar{A}BC + A\bar{B}C + ABC} = \overline{\bar{A}BC} \cdot \overline{A\bar{B}C} \cdot \overline{ABC}$$

Dùng định lý De Morgan một lần nữa cho từng thừa số trong biểu thức, ta được:

$$Z = (A + B + C) \cdot (\bar{A} + \bar{B} + \bar{C}) \cdot (\bar{A} + \bar{B} + \bar{C})$$

Diễn tả  $\bar{Z}$  theo dạng tích chuẩn:

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

Lấy đảo hai vế:

$$Z = \overline{(A + B + \bar{C})(A + \bar{B} + C)(A + \bar{B} + \bar{C})(\bar{A} + B + \bar{C})(\bar{A} + \bar{B} + C)}$$

$$\begin{aligned} Z &= \overline{A + B + \bar{C}} \cdot \overline{A + \bar{B} + C} \cdot \overline{A + \bar{B} + \bar{C}} \cdot \overline{\bar{A} + B + \bar{C}} \cdot \overline{\bar{A} + \bar{B} + C} \\ &= \bar{A}BC + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + A\bar{B}C + ABC \end{aligned}$$

### Dạng số

Để đơn giản cách viết người ta có thể diễn tả một hàm **Tổng chuẩn** hay **Tích chuẩn** bởi tập hợp các số dưới dấu tổng ( $\Sigma$ ) hay tích ( $\Pi$ ). Mỗi tổ hợp biến được thay bởi một số thập phân tương đương với trị nhị phân của chúng. Khi sử dụng cách viết này trọng lượng các biến phải được chỉ rõ.

**Thí dụ:** Cho hàm  $Z$  xác định như trên, tương ứng với dạng chuẩn thứ nhất, hàm này lấy giá trị của các hàng 1, 2, 3, 5, 7, ta viết  $Z=f(A,B,C) = \Sigma(1,2,3,5,7)$ . Tương tự, nếu dùng dạng chuẩn thứ hai ta có thể viết  $Z=f(A,B,C) = \Pi(0,4,6)$ .

**Chú ý:** Khi viết các hàm theo dạng số ta phải chỉ rõ trọng số của các bit, thí dụ ta có thể ghi kèm theo hàm  $Z$  ở trên 1 trong 3 cách như sau:  $A=MSB$  hoặc  $C=LSB$  hoặc  $A=4, B=2, C=1$

### RÚT GỌN HÀM LOGIC

Để thực hiện một hàm logic bằng mạch điện tử, người ta luôn luôn nghĩ đến việc sử dụng lượng linh kiện ít nhất. Muốn vậy, hàm logic phải ở dạng tối giản, nên vấn đề rút gọn hàm logic là **bước đầu tiên phải thực hiện** trong quá trình thiết kế. Có 3 phương pháp rút gọn hàm logic:

- Phương pháp đại số
- Phương pháp dùng bảng Karnaugh
- Phương pháp Quine Mc. Cluskey

## Phương pháp đại số

Phương pháp này bao gồm việc áp dụng các tính chất của hàm logic cơ bản. Một số đẳng thức thường được sử dụng được nhóm lại như sau:

$$(1) \quad AB + \bar{B} = B \quad (A+B).(\bar{A}+B) = B \quad (1')$$

$$(2) \quad A + AB = A \quad A.(A+B) = A \quad (2')$$

$$(3) \quad A + \bar{B} = A + B \quad A.(\bar{A}+B) = A.B \quad (3')$$

Chứng minh các đẳng thức 1, 2, 3:

$$(1) \quad AB + \bar{B} = B(A + \bar{B}) = B.1 = B$$

$$(2) \quad A + AB = A(1+B) = A$$

$$(3) \quad A + \bar{B} = (A + \bar{B}).(A+B) = A+B$$

Các đẳng thức (1'), (2'), (3') là song đối của (1), (2), (3).

Các qui tắc rút gọn:

- **Qui tắc 1:** Nhờ các đẳng thức trên nhóm các số hạng lại.

**Thí dụ:** Rút gọn biểu thức  $ABC + AB\bar{C} + A\bar{B}CD$

$$\text{Theo (1)} \quad ABC + AB\bar{C} = AB$$

$$\text{Vậy} \quad ABC + AB\bar{C} + A\bar{B}CD = AB + A\bar{B}CD = A(B + \bar{B}CD)$$

$$\text{Theo (3)} \quad B + \bar{B}CD = B + CD$$

$$\text{Và kết quả cuối cùng: } ABC + AB\bar{C} + A\bar{B}CD = A(B+CD)$$

- **Qui tắc 2:** Ta có thể thêm một số hạng đã có trong biểu thức logic vào biểu thức mà không làm thay đổi biểu thức.

**Thí dụ:** Rút gọn biểu thức:  $ABC + \bar{A}BC + A\bar{B}C + AB\bar{C}$

$$\text{Thêm } ABC \text{ vào để được: } (ABC + \bar{A}BC) + (ABC + A\bar{B}C) + (ABC + AB\bar{C})$$

$$\text{Theo (1) các nhóm trong dấu ngoặc rút gọn thành: } BC + AC + AB$$

$$\text{Vậy: } ABC + \bar{A}BC + A\bar{B}C + AB\bar{C} = BC + AC + AB$$

- **Qui tắc 3:** Có thể bỏ số hạng chứa các biến đã có trong số hạng khác

**Thí dụ 1:** Rút gọn biểu thức  $AB + \bar{B}C + AC$

Biểu thức không đổi nếu ta nhân một số hạng trong biểu thức với 1, ví dụ  $(B + \bar{B})$ :

$$AB + \bar{B}C + AC = AB + BC + AC(B + \bar{B})$$

Triển khai số hạng cuối cùng của vế phải, ta được:

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

$$\text{Thừa số chung: } AB(1+C) + \bar{B}C(1+A) = AB + \bar{B}C$$

$$\text{Tóm lại: } AB + \bar{B}C + AC = AB + \bar{B}C.$$

Trong bài toán này ta đã đơn giản được số hạng AC.

**Thí dụ 2:** Rút gọn biểu thức  $(A+B).(\bar{B}+C).(A+C)$

Biểu thức không đổi nếu ta thêm vào một thừa số có trị = 0, ví dụ B.B

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

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

$$\text{Theo (2')} \quad (A+B).(A + B + C) = (A+B) \text{ và } (\bar{B}+C).(A + \bar{B}+C) = (\bar{B}+C)$$

$$\text{Vậy: } (A+B).(\bar{B}+C).(A+C) = (A+B).(\bar{B}+C)$$

Trong bài toán này ta đã bỏ số hạng A+C

- **Qui tắc 4:** Có thể đơn giản bằng cách dùng hàm chuẩn tương đương có số hạng ít nhất  $n$

**Thí dụ:** Hàm  $f(A,B,C) = \Sigma(2,3,4,5,6,7)$  với trọng lượng  $A=4, B=2, C=1$

Hàm đảo của f:  $f(A, B, C) = \Sigma(0, 1) = A.B.C + A.B.C = A.B = A + B$   
 Vậy  $f(A,B,C) = A+B$

**Dùng bảng Karnaugh**

Dùng bảng Karnaugh cho phép rút gọn dễ dàng các hàm logic chứa từ 3 tới 6 biến.

**Nguyên tắc**

Xét hai tổ hợp biến  $AB$  và  $\bar{A}B$ , hai tổ hợp này chỉ khác nhau một bit, ta gọi chúng là hai tổ hợp kề nhau.

Ta có:  $AB + A\bar{B} = A$ , biến B đã được đơn giản.

Phương pháp của bảng Karnaugh dựa vào việc nhóm các tổ hợp kề nhau trên bảng để đơn giản biến có giá trị khác nhau trong các tổ hợp này.

Công việc rút gọn hàm được thực hiện theo bốn bước:

Vẽ bảng Karnaugh theo số biến của hàm

Chuyển hàm cần đơn giản vào bảng Karnaugh

Gom các ô chứa các tổ hợp kề nhau lại thành các nhóm sao cho có thể rút gọn hàm tới mức tối giản

Viết kết quả hàm rút gọn từ các nhóm đã gom được.

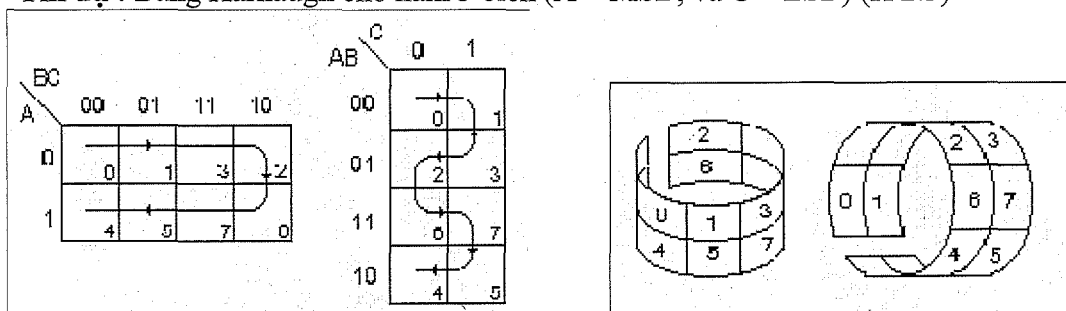
**2.3.2.2 Vẽ bảng Karnaugh**

- Bảng Karnaugh thực chất là một dạng khác của bảng sự thật, trong đó mỗi ô của bảng tương đương với một hàng trong bảng sự thật.

Để vẽ bảng Karnaugh cho n biến, người ta chia số biến ra làm đôi, phân nửa dùng để tạo  $2^{n/2}$  cột, phân nửa còn lại tạo  $2^{n/2}$  hàng (nếu n là số lẻ, người ta có thể cho số lượng biến trên cột lớn hơn số lượng biến cho hàng hay ngược lại cũng được). Như vậy, với một hàm có n biến, bảng Karnaugh gồm  $2^n$  ô, mỗi ô tương ứng với tổ hợp biến này. Các ô trong bảng được sắp đặt sao cho hai ô kề nhau chỉ khác nhau một đơn vị nhị phân (khác nhau một bit), điều này cho thấy rất thuận tiện nếu chúng ta dùng mã Gray. Chính sự sắp đặt này cho phép ta đơn giản bằng cách nhóm các ô kề nhau lại.

Với 2 biến AB, sự sắp đặt sẽ theo thứ tự: AB = 00, 01, 11, 10 (đây là thứ tự mã Gray, nhưng để cho dễ ta dùng số nhị phân tương ứng để đọc thứ tự này: 0, 1, 3, 2)

**Thí dụ :** Bảng Karnaugh cho hàm 3 biến (A = MSB, và C = LSB) (H 2.3)



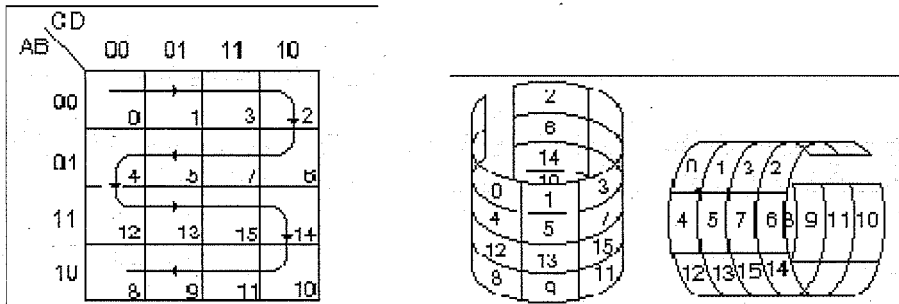
(H 2.3)

Với 3 biến ABC, ta được: ABC = 000, 001, 011, 010, 110, 111, 101, 100 (số nhị phân tương ứng: 0, 1, 3, 2, 6, 7, 5, 4)

Lưu ý là ta có thể thiết lập bảng Karnaugh theo chiều nằm ngang hay theo chiều đứng.

Do các tổ hợp ở các bìa trái và phải kề nhau nên ta có thể coi bảng có dạng hình trụ thẳng đứng và các tổ hợp ở bìa trên và dưới cũng kề nhau nên ta có thể coi bảng có dạng hình trụ trục nằm ngang. Và 4 tổ hợp biến ở 4 góc cũng là các tổ hợp kề nhau.

Hình (H 2.4) là bảng Karnaugh cho 4 biến.



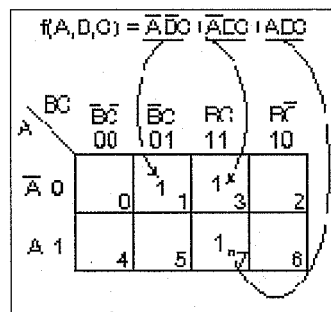
(H 2.4)

### Chuyển hàm logic vào bảng Karnaugh.

Trong mỗi ô của bảng ta đưa vào giá trị của hàm tương ứng với tổ hợp biến, để đơn giản chúng ta có thể chỉ ghi các trị 1 mà bỏ qua các trị 0 của hàm. Ta có các trường hợp sau:

◆ Từ hàm viết dưới dạng tổng chuẩn:

**Thí dụ 1 :**  $f(A,B,C) = \bar{A} \cdot \bar{B} \cdot C + \bar{A} \cdot B \cdot C + A \cdot B \cdot C$



(H 2.5)

◆ **Nếu hàm không phải là dạng chuẩn**, ta phải đưa về dạng chuẩn bằng cách thêm vào các số hạng sao cho hàm vẫn không đổi nhưng các số hạng chứa đủ các biến.

**Thí dụ 2 :**  $Y = \bar{A}BC + AB\bar{D} + A\bar{B}C + A\bar{C}D$

Hàm này gồm 4 biến, nên để đưa về dạng tổng chuẩn ta làm như sau:

$$Y = \bar{A}BC(D+\bar{D}) + AB\bar{D}(C+\bar{C}) + A\bar{B}C(D+\bar{D}) + A\bar{C}D(B+\bar{B})$$

$$Y = \bar{A}BCD + \bar{A}BC\bar{D} + ABC\bar{D} + ABCD + A\bar{B}C\bar{D} + A\bar{B}CD + A\bar{B}C\bar{D} + A\bar{B}CD$$

Và Hàm Y được đưa vào bảng Karnaugh như sau (H 2.6):

	00	01	11	10
AB				
$\overline{A}\overline{B}$				
$\overline{A}B$			1	1
$A\overline{B}$	1	1		1
$AB$		1	1	1

(H 2.6)

♦ Từ dạng số thứ nhất, với các trọng lượng tương ứng A=4, B=2, C=1

**Thí dụ 3 :**  $f(A,B,C) = \Sigma(1,3,7)$ . Hàm số sẽ lấy giá trị 1 trong các ô 1,3 và 7.

♦ Từ dạng tích chuẩn: Ta lấy hàm đảo để có dạng tổng chuẩn và ghi trị 0 vào các ô tương ứng với tổ hợp biến trong tổng chuẩn này. Các ô còn lại chứa số 1.

**Thí dụ 4 :**  $Y = f(A,B,C) = (A+B+C).(A+\overline{B}+C).(\overline{A}+B+C).(\overline{A}+B+\overline{C}).(\overline{A}+\overline{B}+C)$

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

Và bảng Karnaugh tương ứng (H 2.7).

	00	01	11	10
BC				
$\overline{A}$ 0	0	1	1	0
A 1	0	1	1	0

(H 2.7)

♦ Từ dạng số thứ hai:

**Thí dụ 5 :**  $f(A,B,C) = \Pi(0,2,4,5,6)$

Hàm sẽ lấy các trị 0 ở các ô 0, 2, 4, 5, 6. Dĩ nhiên là ta phải ghi các giá trị 1 trong các ô còn lại (H 2.7).

♦ Từ bảng sự thật:

**Thí dụ 6 :** Hàm  $f(A,B,C)$  cho bởi bảng sự thật

N	A	B	C	$f(A,B,C)$
---	---	---	---	------------

0	0	0	0	0
1	0	0	1	1
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

Ta ghi 1 vào các ô tương ứng với các tổ hợp biến ở hàng 1, 3 và 7, kết quả giống như ở thí dụ 1.

♦ **Trường hợp có một số tổ hợp cho giá trị hàm không xác định:** nghĩa là ứng với các tổ hợp này hàm có thể có giá trị 1 hoặc 0, do đó, ta ghi dấu X vào các ô tương ứng với các tổ hợp này, lúc gom nhóm ta sử dụng nó như số 1 hay số 0 một cách tùy ý sao cho có được kết quả rút gọn nhất.

**Thí dụ 7:**  $f(A,B,C,D) = \Sigma(3,4,5,6,7)$  với các tổ hợp từ 10 đến 15 cho hàm có trị bất kỳ (không xác định) (H 2.8).

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

(H 2.8)

### Qui tắc gom nhóm

Các tổ hợp biến có trong hàm logic hiện diện trong bảng Karnaugh dưới dạng các số 1 trong các ô, vậy việc gom thành nhóm các tổ hợp kề nhau được thực hiện theo qui tắc sau:

- Gom các số 1 kề nhau thành từng nhóm sao cho số nhóm càng ít càng tốt. Điều này có nghĩa là số số hạng trong kết quả sẽ càng ít đi.

- Tất cả các số 1 phải được gom thành nhóm và một số 1 có thể ở nhiều nhóm.

- Số số 1 trong mỗi nhóm càng nhiều càng tốt nhưng phải là bội của  $2^k$  (mỗi nhóm có thể có 1, 2, 4, 8 ... số 1). Cứ mỗi nhóm chứa  $2^k$  số 1 thì tổ hợp biến tương ứng với nhóm đó giảm đi k số hạng.

- Kiểm tra để bảo đảm số nhóm gom được không thừa.

### Qui tắc rút gọn

- Kết quả cuối cùng được lấy như sau:

Hàm rút gọn là tổng của các tích: Mỗi số hạng của tổng tương ứng với một nhóm các

số 1 nói trên và số hạng này là tích của các biến, biến A (hay ) là thừa số của tích khi tất cả các



số 1 của nhóm chỉ chứa trong phân nửa bảng trong đó biến A có giá trị 1 (hay 0). Nói cách khác nếu các số 1 của nhóm đồng thời nằm trong các ô của biến A và thì biến A sẽ được đơn giản. Hình dưới đây minh họa việc lấy các thừa số trong tích

**Thí dụ** đối với bảng (H 2.9) ta có kết quả như sau:

- Hàm Y là hàm 4 biến A,B,C,D

		CD				
AB		00	01	11	10	
00				1	1	nhóm 2
01		1	1	1	1	
11						nhóm 3
10						
	nhóm 1					

(H 2.9)

- Nhóm 1 chứa 2 số 1 ( $k=1$ ), như vậy nhóm 1 sẽ còn 3 biến, theo hàng, 2 số 1 này ở 2 ô ứng với B và AB, biến A sẽ được đơn giản và theo cột thì 2 ô này ứng với tổ hợp C.D.

Kết quả ứng với nhóm 1 là: B.C.D

- Nhóm 2 chứa 4 số 1 ( $4=2^2$ ,  $k=2$ ), như vậy nhóm 2 sẽ còn 2 biến, theo hàng, 4 số 1 này ở 2 ô ứng với tổ hợp B và B, biến B sẽ được đơn giản và theo cột thì 4 ô này ứng với tổ hợp CD và C D, cho phép đơn giản biến D.

Kết quả ứng với nhóm 2 là: A C.

- Nhóm 3 chứa 4 số 1 ( $4=2^2$ ,  $k=2$ ), như vậy nhóm 2 sẽ còn 2 biến, theo hàng, 4 số 1 này ở ô ứng với tổ hợp B, theo cột 4 số 1 này chiếm hết 4 cột nên 2 biến C và D được đơn giản.

**Kết quả ứng với nhóm 3 là: A B.**

Và hàm Y rút gọn là:  $Y = \overline{B}C\overline{D} + \overline{A}C + \overline{A}B$

Dưới đây là một số thí dụ

**Thí dụ 1 :** Rút gọn hàm  $Y = f(A,B,C) = \overline{A}.\overline{B}.C + \overline{A}.B.C + A.\overline{B}.\overline{C} + A.\overline{B}.C + A.B.C$

		C	
AB		0	1
00			1
01			1
11			1
10		1	1

(H 2.10)

(H 2.10) cho  $Y = A\overline{B} + C$

**Thí dụ 2 :** Rút gọn hàm  $Y = f(A,B,C,D) = \Sigma(0,2,4,5,8,10,12,13)$  với A=MSB

		CD			
AB		00	01	11	10
00		1			1
01		1	1		
11		1	1		
10		1			1

(H2.11)

(H 2.11) cho  $Y = B\bar{C} + \bar{B}\bar{D}$

**Thí dụ 3 :** Rút gọn hàm S cho bởi bảng sự thật:

N	A	B	C	D	S
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	0
10→15					x (Không xác định)

Bảng Karnaugh: (H 2.12)

		CD			
		00	01	11	10
AB	00			1	1
	01	1	1		
	11	x	x	x	x
	10			x	x

(H 2.12)

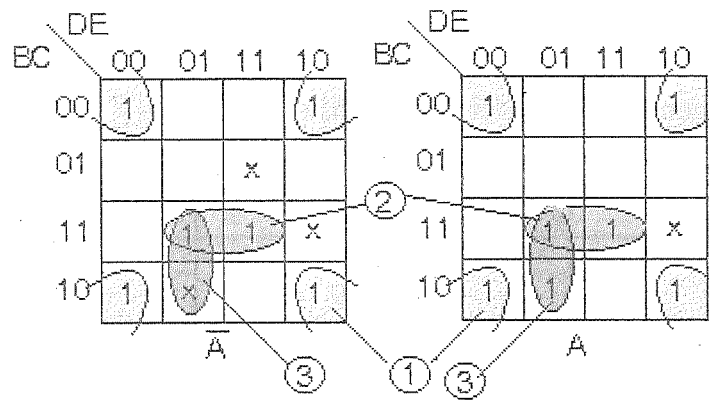
Kết quả :  $S = B\bar{C} + \bar{B}C$

**Rút gọn các hàm nhiều biến bằng cách dùng bảng Karnaugh 4 biến:**

Để rút gọn các hàm nhiều biến (5 và 6 biến) người ta có thể dùng bảng Karnaugh 4 biến. Dưới đây là vài thí dụ:

**Thí dụ 4 :** Rút gọn hàm  $f(A,B,C,D,E) = \sum (0,2,8,10,13,15,16,18,24,25,26,29,31)$  với (7,9,14,30) không xác định

- Trước nhất vẽ 2 bảng Karnaugh cho 4 biến BCDE, một ứng với  $\bar{A}$  và một với A
- Bảng ứng với  $\bar{A}$  dùng cho các số từ 0 đến 15
- Bảng ứng với A dùng cho các số từ 16 đến 31
- Nhóm các số 1 có cùng vị trí ở hai bảng, kết quả sẽ đơn giản biến A
- Nhóm các số 1 của từng bảng cho đến hết, kết quả được xác định như cách làm thông thường, nhớ A và  $\bar{A}$  trong từng nhóm (H 2.13).



(H 2.13)

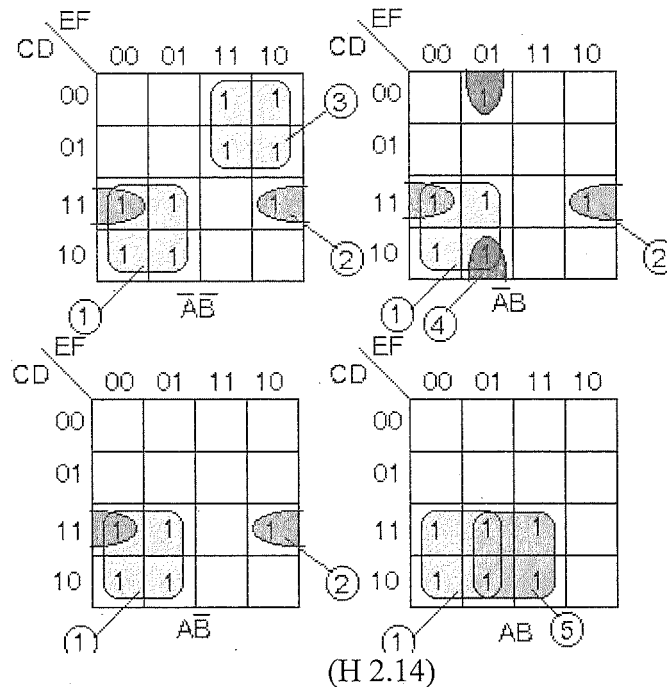
nhóm (1) cho :  $\overline{CE}$  ;            (2) cho :  $BCE$  ;            (3) cho :  $B\overline{DE}$   
 Vậy  $f(A,B,C,D,E) = \overline{CE} + BCE + B\overline{DE}$

**Thí dụ 5 :** Rút gọn hàm

$$f(A,B,C,D,E,F) = \Sigma(2,3,6,7,8,9,12,13,14,17,24,25,28,29,30,40,41,44,45,46,56,57,59,60,61,63)$$

Tương tự như trên nhưng phải vẽ 4 bảng cho:

$\overline{B}$  cho các số (0-15);             $\overline{B}$  cho các số (16-31);  
 $AB$  cho các số (48-63) và             $A\overline{B}$  cho các số (32-47).



(H 2.14)

Kết quả: (1) cho  $\overline{CE}$ ;     $\overline{A}C\overline{D}\overline{F} + \overline{B}C\overline{D}\overline{F}$  ; (3)  $\overline{A}\overline{B}C\overline{E}$ ; (4)  $\overline{A}\overline{B}D\overline{E}\overline{F}$  ; (5)  $ABC\overline{F}$   
 (2)

Vậy:

$$f(A,B,C,D,E,F) = \overline{CE} + \overline{A}\overline{B}C\overline{E} + ABC\overline{F} + \overline{A}C\overline{D}\overline{F} + \overline{B}C\overline{D}\overline{F} + \overline{A}\overline{B}D\overline{E}\overline{F}$$

## Phương pháp Quine-Mc. Cluskey

Phương pháp Quine-Mc. Cluskey cũng dựa trên tính kế của các tổ hợp biến để đơn giản số biến trong các số hạng của biểu thức dạng tổng (minterm). Trong quá trình đơn giản này có thể xuất hiện các số hạng giống nhau mà ta có thể bỏ bớt được.

### Phương pháp được thực hiện qua 2 giai đoạn:

**Giai đoạn 1:** Dựa trên tính kế của các tổ hợp biến để đơn giản số biến trong các số hạng của biểu thức dạng tổng (minterm).

**Giai đoạn 2:** Kiểm tra và thực hiện việc tối giản.

Thí dụ dưới đây minh họa cho việc thực hiện phương pháp để rút gọn một hàm logic.

**Thí dụ 1:** Rút gọn hàm  $f(A,B,C,D) = \Sigma(1,2,4,5,6,10,12,13,14)$

#### ♣ Giai đoạn 1

- Các minterm được nhóm lại theo số số 1 có trong tổ hợp và ghi lại trong bảng theo thứ tự số 1 tăng dần:

Trong thí dụ này có 3 nhóm:

Nhóm chứa **một** số 1 gồm các tổ hợp 1, 2, 4

Nhóm chứa **hai** số 1 gồm các tổ hợp 5, 6, 10, 12

Nhóm chứa **ba** số 1 gồm các tổ hợp 13, 14

Bảng 1:

		A	B	C	D
x	1	0	0	0	1
x	2	0	0	1	0
x	4	0	1	0	0
x	5	0	1	0	1
x	6	0	1	1	0
x	10	1	0	1	0
x	12	1	1	0	0
x	13	1	1	0	1
x	14	1	1	1	0

- Mỗi tổ hợp trong một nhóm sẽ được so sánh với mỗi tổ hợp trong nhóm kế cận. Nếu 2 tổ hợp chỉ khác nhau một biến, ta có thể dùng biểu thức  $\overline{AB} + B = B$  để đơn giản được 1 biến. Biến đã đơn giản được thay bởi dấu -. Đánh dấu x vào các tổ hợp đã xét để tránh sai sót

Như vậy, tổ hợp thứ nhất của nhóm thứ nhất **0001** so sánh với tổ hợp thứ nhất của nhóm

thứ hai **0101** vì chúng chỉ khác nhau ở biến B, vậy chúng có thể đơn giản thành **0-01**. Hai số hạng 1 và 5 đã được gom lại thành nhóm (1,5) và được ghi vào bảng 2.

Tiếp tục so sánh tổ hợp **0001** này với các tổ hợp còn lại của nhóm 2 (**0110, 1010, 1100**), vì chúng khác nhau nhiều hơn 1 bit nên ta không được kết quả nào khác. Như vậy, ta đã so sánh xong tổ hợp thứ nhất, đánh dấu x trước tổ hợp này để ghi nhớ.

Công việc tiến hành tương tự cho nhóm thứ hai và thứ ba.

**Lưu ý:** Nhận xét về việc so sánh các tổ hợp với nhau ta thấy có thể thực hiện nhanh được

bằng cách làm bài toán trừ 2 số nhị phân tương ứng của 2 tổ hợp, nếu kết quả là một số có trị  $= 2^k$  (1, 2, 4, 8 ...) thì 2 tổ hợp đó so sánh được và biến được đơn giản chính là biến có trọng

số  $= 2^k$  (thí dụ 2 tổ hợp 1 và 5 có hiệu số là 4 nên đơn giản được biến B), nếu hiệu số  $\neq 2^k$  thì 2 tổ hợp đó không so sánh được, tức không có biến được đơn giản.

Kết quả cho bảng thứ hai

- Bảng thứ hai gồm các tổ hợp đã được rút gọn và chỉ còn lại 2 nhóm (giảm một nhóm so với bảng 1).

Bảng 2

	A	B	C	D
x 1,5	0	-	0	1
x 2,6	0	-	1	0
x 2,10	-	0	1	0
x 4,5	0	1	0	-
x 4,6	0	1	-	0
x 4,12	-	1	0	0
x 5,13	-	1	0	1
x 6,14	-	1	1	0
x 10,14	1	-	1	0
x 12,13	1	1	0	-
x 12,14	1	1	-	0

Thực hiện công việc tương tự như trên với hai nhóm trong bảng thứ hai này, các số hạng sẽ được nhóm lại nếu chúng chỉ khác nhau một biến và có vị trí dấu - trùng nhau. Ta được bảng thứ 3.

Bảng 3:

	A	B	C	D
2,6 ; 10,14	-	-	1	0
2,10 ; 6,14	-	-	1	0
4,5 ; 12,13	-	1	0	-
4,6 ; 12,14	-	1	-	0
4,12 ; 5,13	-	1	0	-
4,12 ; 6,14	-	1	-	0

Quan sát bảng thứ 3 ta thấy có các tổ hợp giống nhau, như vậy ta có thể loại bỏ bớt các tổ hợp này và chỉ giữ lại một.

Kết quả của hàm rút gọn gồm tổng các số hạng tương ứng với các tổ hợp không gom thành nhóm trong các bảng đầu tiên, đó là tổ hợp (1,5) trong bảng 2, trị tương ứng là  $\overline{C}D$  với các tổ hợp còn lại trong bảng cuối cùng, đó là các tổ hợp (2,6 ; 10,14) mà trị tương ứng là

$\overline{CD}$ , (4,5 ; 12,13) cho  $\overline{BC}$  và (4,6 ; 12,14) cho  $\overline{BD}$  trong bảng 3. Vậy:

$$f(A,B,C,D) = \overline{A} \overline{C} D + \overline{CD} + \overline{BC} + \overline{BD}$$

Đến đây, nếu quan sát các tổ hợp cho các kết quả trên, ta thấy các tổ hợp còn chứa các số hạng giống nhau (số 4 và số 12 chẳng hạn), như vậy kết quả trên có thể là chưa tối giản.

♣ Giai đoạn 2:

Để có thể rút gọn hơn nữa ta lập một bảng như sau:

Cột bên trái ghi lại các tổ hợp đã chọn được trong giai đoạn 1, các cột còn lại ghi các trị thập phân có trong hàm ban đầu.



Trên cùng hàng của tổ hợp ta đánh dấu \* dưới các cột có số tương ứng (ví dụ hàng chứa tổ hợp 1,5 có các dấu \* ở cột 1 và 5). Tương tự cho các tổ hợp khác.

Bảng 4

		1	2	4	5	6	10	12	13	14
1,5	←	*↓			*↓					
2,6 ; 10,14	←		*↓			*↓	*↓			*↓
4,5 ; 12,13	←			*↓	*			*↓	*↓	
4,6 ; 12,14				*		*		*		*
		X	X	X	X	X	X	X	X	X

Xét các cột chỉ chứa một dấu \*, đó là các cột 1,2,10 và 13, các tổ hợp ở cùng hàng với các dấu \* này sẽ được chọn, đó là các tổ hợp (1,5), (2,6 ; 10,14), (4,5 ; 12,13), tương ứng với  $\overline{CD} + \overline{CD} + BC$ . Đánh dấu X dưới các cột tương ứng với các số có trong các tổ hợp đã chọn. Nếu tất cả các cột đều được đánh dấu thì các tổ hợp đã chọn đủ để diễn tả hàm ban đầu.

Trong trường hợp của bài toán này, sau khi chọn các tổ hợp nói trên thì tất cả cột đã được đánh dấu do đó kết quả cuối cùng là (sau khi loại bỏ tổ hợp BD):

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

**Thí dụ 2:** Rút gọn hàm  $f(A,B,C,D) = \Sigma(3,4,6,7,8,11,12,15)$

✧ Giai đoạn 1

Bảng 1:

		A	B	C	D
X	4	0	1	0	0
X	8	0	0	1	0
X	3	0	0	1	1
X	6	0	1	1	0
X	12	1	1	0	0
X	7	0	1	1	1
X	11	1	0	1	1
X	15	1	1	1	1

So sánh các tổ hợp của 2 nhóm gần nhau ta được kết quả cho bảng thứ hai

- Bảng thứ hai gồm các tổ hợp đã được rút gọn và chỉ còn lại 3 nhóm (giảm một nhóm so với bảng 1).

Bảng 2

		A	B	C	D
	4,6	0	1	-	0
	4,12	-	1	0	0
	8,12	1	-	0	0
X	3,7	0	-	1	1
X	3,11	-	0	1	1
	6,7	0	1	1	-
X	7,15	-	1	1	1
X	11,15	1	-	1	1

Bảng 3:

	A	B	C	D
3,7 ; 11,15	-	-	1	1
3,11 ; 7,15	-	-	1	1

Kết quả của hàm rút gọn gồm tổng các số hạng tương ứng với các tổ hợp không gom thành nhóm: (4,6), (4,12), (8,12), (6,7) và (3,7;11,15)

$$f(A,B,C,D) = CD + \overline{A}BD + BC\overline{D} + AC\overline{D} + \overline{A}BC$$

♣ **Giai đoạn 2:**

Bảng 4

		3	4	6	7	8	11	12	15
3,7;11,15	←	*↓			*↓		*↓		*↓
4,6			*	*					
4,12			*					*	
8,12	←					*↓		*↓	
6,7				*	*				
		X			X	X	X	X	X

Các cột 3, và 8 chỉ chứa một dấu \*, các tổ hợp ở cùng hàng với các dấu \* này sẽ được chọn, đó là các tổ hợp (3,7;11,15) và (8,12), tương ứng với  $\overline{C}D$  và  $AC\overline{D}$ .

Đánh dấu X dưới các cột tương ứng với các số có trong các tổ hợp đã chọn.

Đến đây ta thấy còn 2 cột 4 và 6 chưa có dấu X, trong lúc chúng ta còn đến 3 tổ hợp để chọn. Dĩ nhiên trong trường hợp này ta chỉ cần chọn tổ hợp  $\overline{(4,6)}$  ( $\overline{A}BD$ ) thay vì chọn (4,12) và (6,7) thì đủ dấu X để lấp đầy các cột.

Tóm lại:

$$f(A,B,C,D) = CD + \overline{A}BD + AC\overline{D}$$

Thí dụ về bài toán đầy đủ:

Thí dụ 1:

Cho hàm logic  $F(A, B, C)$  thỏa tính chất:  $F(A,B,C) = 1$  nếu có một và chỉ một biến bản

g 1 a- Lập bảng sự thật cho hàm F.

b- Rút gọn hàm F.

c- Diễn tả hàm F chỉ dùng hàm AND và NOT

Giải

a. Dựa vào điều kiện của bài toán ta có bảng sự thật của hàm F:

A	B	C	F(A,B,C)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0

1	1	1	0
---	---	---	---

b. Rút gọn hàm F  
Bảng Karnaugh

		BC			
		00	01	11	10
A	0		1		1
	1	1			

$$F(A,B,C) = \overline{A}BC + \overline{A}B\overline{C} + A\overline{B}\overline{C}$$

c. Diễn tả hàm F chỉ dùng hàm AND và NOT  
Dùng định lý De Morgan, lấy đảo 2 lần hàm F:

$$F(A,B,C) = F(A,B,C) = \overline{\overline{\overline{A}BC + \overline{A}B\overline{C} + A\overline{B}\overline{C}}} = \overline{\overline{A}BC} \cdot \overline{\overline{A}B\overline{C}} \cdot \overline{A\overline{B}\overline{C}}$$

Thí dụ 2:

Cho hàm logic F(A, B, C, D) thỏa tính chất: F(A,B,C,D) = 1 khi có ít nhất 3 biến

bản

g 1 a- Rút gọn hàm F.

b- Diễn tả hàm F chỉ dùng hàm OR và NOT

Giải

a- Rút gọn hàm F

Ta có thể đưa hàm vô bảng Karnaugh mà không cần vẽ bảng sự thật.

Ta đưa số 1 vào tất cả các ô chứa 3 trở lên

		CD			
		00	01	11	10
AB	00				
	01			1	
	11	1	1	1	
	10			1	

Và kết quả của hàm rút gọn là:

$$F(A,B,C,D) = ABC + ABD + ACD + BCD$$

b- Diễn tả hàm F chỉ dùng hàm OR và NOT

Dùng định lý De Morgan cho từng số hạng trong tổng

Viết lại hàm F:

$$F(A,B,C,D) = \overline{\overline{ABC}} + \overline{\overline{ABD}} + \overline{\overline{ACD}} + \overline{\overline{BCD}}$$

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

## BÀI TẬP

1. Diễn tả mỗi mệnh đề dưới đây bằng một biểu thức logic:

a/ Tất cả các biến A,B,C,D đều bằng 1

b/ Tất cả các biến A,B,C,D đều bằng 0

- c/ Ít nhất 1 trong các biến X,Y,Z,T bằng 1  
 d/ Ít nhất 1 trong các biến X,Y,Z,T bằng 0  
 e/ Các biến A,B,C,D lần lượt có giá trị 0,1,1,0

2. Tính đảo của các hàm sau:

- a/  $f_1 = (A + B)(\bar{A} + \bar{B})$   
 b/  $f_2 = (A + \bar{B} + \bar{C})(B + \bar{C} + D)(\bar{A} + C + D)$   
 c/  $f_3 = A(C + D) + (\bar{A} + C)(\bar{B} + C + D)$   
 d/  $f_4 = (AB + C)(BC + D) + \bar{A}BC + \bar{C}D$   
 e/  $f_5 = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A(BC + \bar{B}\bar{C})$

3. Chứng minh bằng đại số các biểu thức sau:

- a/  $\overline{A.B + \bar{A}.B} = \bar{A}.B + A.\bar{B}$   
 b/  $\overline{A.B + \bar{A}.C} = (A + C)(\bar{A} + B)$   
 c/  $\overline{A.C + B.\bar{C}} = \bar{A}.C + \bar{B}.\bar{C}$   
 d/  $\overline{(A + B)(\bar{A} + C)(B + C)} = (A + B)(\bar{A} + \bar{C})$   
 e/  $\overline{(A + C)(B + \bar{C})} = (\bar{A} + C)(\bar{B} + \bar{C})$

4. Viết dưới dạng tổng chuẩn các hàm xác định bởi:

- a/  $f(A,B,C) = 1$  nếu số nhị phân  $(ABC)_2$  là số chẵn  
 b/  $f(A,B,C) = 1$  nếu có ít nhất 2 biến số = 1  
 c/  $f(A,B,C) = 1$  nếu số nhị phân  $(ABC)_2 > 5$   
 d/  $f(A,B,C) = 1$  nếu số biến số 1 là số chẵn  
 e/  $f(A,B,C) = 1$  nếu có 1 và chỉ 1 biến số = 1

5. Viết dưới dạng tích chuẩn các hàm ở bài tập 4

6. Viết dưới dạng số các bài tập 4

7. Viết dưới dạng số các bài tập 5

8. Rút gọn các hàm dưới đây bằng phương pháp đại số (A = MSB)

- a/  $f_1 = ABC + A\bar{B}C + AB\bar{C}D$   
 b/  $f_2 = (A+BC) + \bar{A}(\bar{B}+\bar{C})(AD+C)$   
 c/  $f_3 = (A+B+C)(A+B+\bar{C})(\bar{A}+B+C)(\bar{A}+B+\bar{C})$   
 d/  $f_4(A,B,C,D) = \Sigma(0,3,4,7,8,9,14,15)$   
 e/  $f_5 = \bar{A}B + AC + BC$   
 f/  $f_6 = (A+\bar{C})(B+C)(A+B)$

9. Dùng bảng Karnaugh rút gọn các hàm sau: (A = MSB)

- a/  $f(A,B,C) = \Sigma(1,3,4)$   
 b/  $f(A,B,C) = \Sigma(1,3,7)$   
 c/  $f(A,B,C) = \Sigma(0,3,4,6,7)$   
 d/  $f(A,B,C) = \Sigma(1,3,4)$ . Các tổ hợp biến 6,7 cho hàm không xác định  
 e/  $f(A,B,C) = \bar{A}.\bar{B}.C + \bar{A}.B.\bar{C} + A.\bar{B}.\bar{C} + A.B.C$   
 f/  $f(A,B,C,D) = \Sigma(5,7,13,15)$   
 g/  $f(A,B,C,D) = \Sigma(0,4,8,12)$   
 h/  $f(A,B,C,D) = \Sigma(0,2,8,10)$   
 i/  $f(A,B,C,D) = \Sigma(0,2,5,6,9,11,13,14)$   
 j/  $f(A,B,C,D) = \Pi(0,1,5,9,10,15)$   
 k/  $f(A,B,C,D) = \Pi(0,5,9,10)$  với các tổ hợp biến (2,3,8,15) cho hàm không xác định  
 l/  $f(A,B,C,D,E) = \Sigma(2,7,9,11,12,13,15,18,22,24,25,27,28,29,31)$

m/ f(A,B,C,D,E) =  $\Sigma(0,2,8,10,13,15,16,18,24,25,26,29,31)$  với các tổ hợp biến (7,9,14,30) cho hàm không xác định

n/ f(A,B,C,D,E,F) =

$\Sigma(2,3,6,7,8,9,12,13,14,17,24,25,28,29,30,40,41,44,45,46,56,57,59,60,61,63)$

o/ f(A,B,C,D,E,F) =

$\Sigma(9,11,13,15,16,18,20,22,25,27,29,31,32,34,36,38,41,43,45,47,48,50,52,54)$

10. Làm lại các bài tập từ 9f bằng phương pháp Quine-Mc Cluskey.

## CHƯƠNG 3 MẠCH LOGIC MSI

Cổng logic là tên gọi chung của các mạch điện tử có chức năng thực hiện các hàm logic. Cổng logic có thể được chế tạo bằng các công nghệ khác nhau (Lưỡng cực, MOS), có thể được tổ hợp bằng các linh kiện rời nhưng thường được chế tạo bởi công nghệ tích hợp IC (Integrated circuit).

Chương này giới thiệu các loại cổng cơ bản, các họ IC số, các tính năng kỹ thuật và sự giao tiếp giữa chúng.

### CÁC KHÁI NIỆM LIÊN QUAN

#### *Tín hiệu tương tự và tín hiệu số*

Tín hiệu tương tự là tín hiệu có biên độ biến thiên liên tục theo thời gian. Nó thường do các hiện tượng tự nhiên sinh ra. Thí dụ, tín hiệu đặc trưng cho tiếng nói là tổng hợp của các tín hiệu hình sin trong dải tần số thấp với các họa tần khác nhau.

Tín hiệu số là tín hiệu có dạng xung, gián đoạn về thời gian và biên độ chỉ có 2 mức rõ rệt: mức cao và mức thấp. Tín hiệu số chỉ được phát sinh bởi những mạch điện thích hợp. Để có tín hiệu số người ta phải số hóa tín hiệu tương tự bằng các mạch biến đổi tương tự sang số (ADC)

#### *Mạch tương tự và mạch số*

Mạch điện tử xử lý các tín hiệu tương tự được gọi là mạch tương tự và mạch xử lý tín hiệu số được gọi là mạch số.

Một cách tổng quát, mạch số có nhiều ưu điểm so với mạch tương tự:

- Dễ thiết kế và phân tích. Vận hành của các cổng logic dựa trên tính chất dẫn điện (bảo hòa) hoặc ngưng dẫn của transistor. Việc phân tích và thiết kế dựa trên chức năng và đặc tính kỹ thuật của các IC và các khối mạch chứ không dựa trên từng linh kiện rời

- Có thể hoạt động theo chương trình lập sẵn nên rất thuận tiện trong điều khiển tự động, tính toán, lưu trữ dữ liệu và liên kết với máy tính.

- Ít bị ảnh hưởng của nhiễu tức có khả năng dung nạp tín hiệu nhiễu với biên độ lớn hơn rất nhiều so với mạch tương tự.

- Dễ chế tạo thành mạch tích hợp và có khả năng tích hợp với mật độ cao.

Dựa vào số cổng trong một chip, người ta phân loại IC số như sau:

- Số cổng < 10: SSI (Small Scale Integrated), mức độ tích hợp nhỏ.

-  $10 < \text{Số cổng} < 100$ : MSI (Medium Scale Integrated), mức độ tích hợp trung bình.

-  $100 < \text{Số cổng} < 1000$ : LSI (Large Scale Integrated), mức độ tích hợp lớn.

-  $1000 < \text{Số cổng} < 10000$ : VLSI (Very Large Scale Integrated), mức độ tích hợp rất

l  
ớ  
n

- Số cổng > 10000: ULSI (Ultra Large Scale Integrated), mức độ tích hợp siêu lớn.

### Biểu diễn các trạng thái Logic 1 và 0

Trong hệ thống mạch logic, các trạng thái logic được biểu diễn bởi các mức điện thế. Với qui ước **logic dương**, điện thế cao biểu diễn logic 1, điện thế thấp biểu diễn logic 0. Ngược lại ta có qui ước **logic âm**. Trong thực tế, mức 1 và 0 tương ứng với một khoảng điện thế xác định và có một khoảng chuyển tiếp giữa mức cao và thấp, ta gọi là khoảng **không xác định**. Khi điện áp của tín hiệu rơi vào khoảng này, mạch sẽ không nhận ra là mức 0 hay 1. Khoảng này tùy thuộc vào họ IC sử dụng và được cho trong bảng thông số kỹ thuật của linh kiện. (H 3.1) là giản đồ điện thế của các mức logic của một số cổng logic thuộc họ TTL.



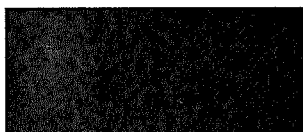
(H 3.1)

## CÔNG LOGIC CƠ BẢN

### Cổng NOT

- Còn gọi là cổng đảo (Inverter), dùng để thực hiện hàm đảo  $Y = \bar{A}$

- Ký hiệu (H 3.2), mũi tên chỉ chiều di chuyển của tín hiệu và vòng tròn là ký hiệu đảo. Trong những trường hợp không thể nhầm lẫn về chiều này, người ta có thể bỏ mũi tên.



(H 3.2)

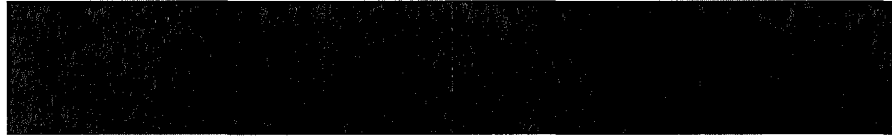


Bảng sự thật



### Cổng AND

- Dùng thực hiện hàm AND 2 hay nhiều biến.
- Cổng AND có số ngõ vào tùy thuộc số biến và một ngõ ra. Ngõ ra của cổng là hàm AND của các biến ngõ vào.
- Ký hiệu cổng AND 2 ngõ vào cho 2 biến (H 3.3a)



(a)

(H 3.3)

(b)

A	B	Y=A.B
0	0	0
0	1	0
1	0	0
1	1	1

Hoặc

A	B	Y=A.B
x	0	0
x	1	A

- Nhận xét:

- Ngõ ra cổng AND chỉ ở mức cao khi tất cả ngõ vào lên cao.
- Khi có một ngõ vào = 0, ngõ ra = 0 bất chấp các ngõ vào còn lại.
- Khi có một ngõ vào = 1, ngõ ra = AND của các ngõ vào còn lại.

Vậy với cổng AND 2 ngõ vào ta có thể dùng 1 ngõ vào làm ngõ kiểm soát (H 3.3b), khi ngõ kiểm soát = 1, **cổng mở** cho phép tín hiệu logic ở ngõ vào còn lại qua cổng và khi ngõ kiểm soát = 0, **cổng đóng**, ngõ ra luôn bằng 0, bất chấp ngõ vào còn lại.

Với cổng AND có nhiều ngõ vào hơn, khi có một ngõ vào được đưa lên mức cao thì ngõ ra bằng AND của các biến ở các ngõ vào còn lại.

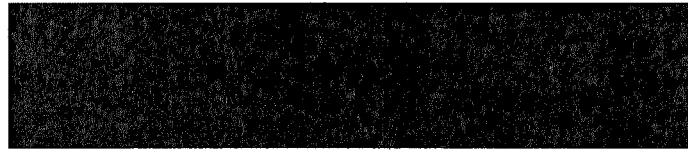
Hình (H 3.4) là giản đồ thời gian của cổng AND hai ngõ vào. Trên giản đồ, ngõ ra Y chỉ lên mức 1 khi cả A và B đều ở mức 1.



(H 3.4)

### Cổng OR

- Dùng để thực hiện hàm OR 2 hay nhiều biến.
- Cổng OR có số ngõ vào tùy thuộc số biến và một ngõ ra.
- Ký hiệu cổng OR 2 ngõ vào



(H 3.5)

- Bảng sự thật

A	B	Y=A+B
0	0	0
0	1	1
1	0	1
1	1	1

Hoặc

A	B	Y=A+B
x	1	1
x	0	A

- **Nhận xét:** - Ngã ra công OR chỉ ở mức thấp khi cả 2 ngã vào xuống thấp.
  - Khi có một ngã vào =1, ngã ra = 1 bất chấp ngã vào còn lại.
  - Khi có một ngã vào =0, ngã ra = OR các ngã vào còn lại.

Vậy với công OR 2 ngã vào ta có thể dùng 1 ngã vào làm ngã kiểm soát, khi ngã kiểm soát = 0, **công mở**, cho phép tín hiệu logic ở ngã vào còn lại qua công và khi ngã kiểm soát = 1, **công đóng**, ngã ra luôn bằng 1.

Với công OR nhiều ngã vào hơn, khi có một ngã vào được đưa xuống mức thấp thì ngã ra bằng OR của các biên ở các ngã vào còn lại.

### Công BUFFER

Còn gọi là công đệm. Tín hiệu số qua công BUFFER không đổi trạng thái logic. Công BUFFER được dùng với các mục đích sau:

- Sửa dạng tín hiệu.
- Đưa điện thế của tín hiệu về đúng chuẩn của các mức logic.
- Nâng khả năng cấp dòng cho mạch.
- Ký hiệu của công BUFFER.

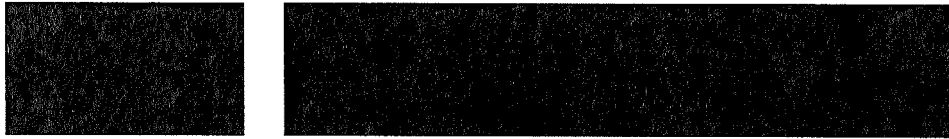


(H 3.6)

Tuy công đệm không làm thay đổi trạng thái logic của tín hiệu vào công nhưng nó giữ vai trò rất quan trọng trong các mạch số.

### Công NAND

- Là kết hợp của công AND và công NOT, thực hiện hàm  $Y = \overline{A.B}$  (Ở đây chỉ xét công NAND 2 ngã vào, độc giả tự suy ra trường hợp nhiều ngã vào).
- Ký hiệu của công NAND (Gồm AND và NOT, công NOT thu gọn lại một vòng tròn)
- Tương tự như công AND, ở công NAND ta có thể dùng 1 ngã vào làm ngã kiểm soát. Khi ngã kiểm soát = 1, công mở cho phép tín hiệu logic ở ngã vào còn lại qua công và bị đảo, khi ngã kiểm soát = 0, công đóng, ngã ra luôn bằng 1.
- Khi nối tất cả ngã vào của công NAND lại với nhau, nó hoạt động như một công đảo



(H 3.7)

### Cổng NOR

- Là kết hợp của cổng OR và cổng NOT, thực hiện hàm  $Y = \overline{A + B}$ .

Ký hiệu của cổng NOR (Gồm cổng OR và NOT, nhưng cổng NOT thu gọn lại một vòng tròn)



(H 3.8)

Các bảng sự thật và các giản đồ thời gian của các cổng BUFFER, NAND, NOR, sinh viên có thể tự thực hiện lấy

### Cổng EX-OR

- Dùng để thực hiện hàm EX-OR.  $Y = A \oplus B = \overline{A}B + A\overline{B}$

- Cổng EX-OR chỉ có 2 ngõ vào và 1 ngõ ra

- Ký hiệu (H 3.9a)

- Một tính chất rất quan trọng của cổng EX-OR:

+ Tương đương với một cổng đảo khi có một ngõ vào nối lên mức cao, (H 3.9b)

+ Tương đương với một cổng đệm khi có một ngõ vào nối xuống mức thấp, (H 3.9c)



(a)



(b)



(c)

(H 3.9)

### Cổng EX-NOR

- Là kết hợp của cổng EX-OR và cổng NOT

- Cổng EX-NOR có 2 ngõ vào và một ngõ ra

- Hàm logic ứng với cổng EX-NOR là

$$Y = \overline{A \oplus B} = \overline{A}B + A\overline{B}$$

- Ký hiệu (H 3.10)

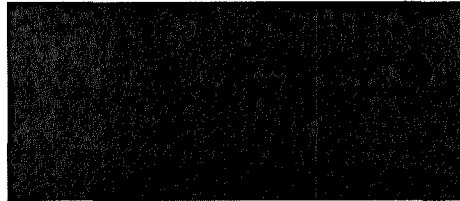
- Các tính chất của cổng EX-NOR giống cổng EX-OR nhưng có ngõ ra đảo lại.



(H 3.10)

### **Cổng phức AOI (AND-OR-INVERTER)**

Ứng dụng các kết quả của Đại số BOOLE, người ta có thể kết nối nhiều cổng khác nhau trên một chip IC để thực hiện một hàm logic phức tạp nào đó. Cổng AOI là một kết hợp của 3 loại cổng AND (A), OR (O) và INVERTER (I). Thí dụ để thực hiện hàm logic  $Y = A.B.C + D.E$ , ta có cổng phức sau:



(H 3.11)

### **Biến đổi qua lại giữa các cổng logic**

Trong chương Hàm Logic chúng ta đã thấy tất cả các hàm logic có thể được thay thế bởi 2 hàm duy nhất là hàm AND (hoặc OR) kết hợp với hàm NOT. Các cổng logic có chức năng thực hiện hàm logic, như vậy chúng ta chỉ cần dùng 2 cổng AND (hoặc OR) và NOT để thực hiện tất cả các hàm logic. Tuy nhiên, vì cổng NOT cũng có thể tạo ra từ cổng NAND (hoặc NOR). Như vậy, tất cả các hàm logic có thể được thực hiện bởi một cổng duy nhất, đó là cổng NAND (hoặc NOR). Hàm ý này cho phép chúng ta biến đổi qua lại giữa các cổng với nhau.

Quan sát Định lý De Morgan chúng ta rút ra qui tắc biến đổi qua lại giữa các cổng AND, NOT và OR, NOT như sau:

Chỉ cần thêm các cổng đảo ở ngã vào và ngã ra khi biến đổi từ AND sang OR hoặc ngược lại. Dĩ nhiên nếu ở các ngã đã có đảo rồi thì đảo này sẽ mất đi.

**Thí dụ 1:** Ba mạch dưới đây tương đương nhau:

(H 3.12b) có được bằng cách đổi AND - OR thêm các đảo ở các ngã vào và ra. Từ (H 3.12b) đổi sang (H 3.12c) ta bỏ 2 cổng đảo nối từ ngã ra cổng NOR đến ngã vào cổng AND



(a)

(b)

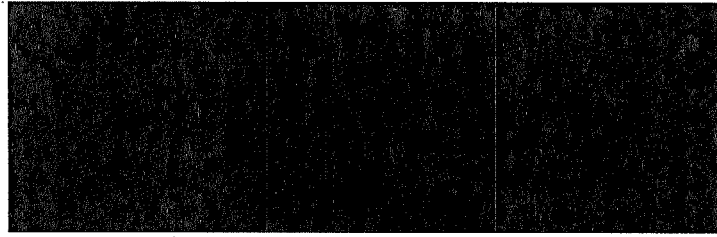
(c)

(H 3.12)

**Thí dụ 2:** Vẽ mạch tương đương của cổng EX-OR dùng toàn cổng NAND  
Dùng định lý De-Morgan, biểu thức hàm EX-OR viết lại:

$$Y = \overline{AB} + \overline{AB} = \overline{\overline{AB} \cdot \overline{AB}}$$

Và mạch tương đương cho ở (H 3.13)



(H 3.13)

## THÔNG SỐ KỸ THUẬT CỦA IC SỐ

Để sử dụng IC số có hiệu quả, ngoài sơ đồ chân và bảng sự thật của chúng, ta nên biết qua một số thuật ngữ chỉ các thông số cho biết các đặc tính của IC.

### *Các đại lượng điện đặc trưng*

-  $V_{CC}$ : Điện thế nguồn (power supply): khoảng điện thế cho phép cấp cho IC để hoạt động tốt. Thí dụ với IC số họ TTL,  $V_{CC}=5\pm 0,5$  V, họ CMOS  $V_{DD}=3-15$ V (Người ta thường dùng ký hiệu  $V_{DD}$  và  $V_{SS}$  để chỉ nguồn và mass của IC họ MOS)

-  $V_{IH}(\min)$ : Điện thế ngõ vào mức cao (High level input voltage): Đây là điện thế ngõ vào nhỏ nhất còn được xem là mức 1

-  $V_{IL}(\max)$ : Điện thế ngõ vào mức thấp (Low level input voltage): Điện thế ngõ vào lớn nhất còn được xem là mức 0.

-  $V_{OH}(\min)$ : Điện thế ngõ ra mức cao (High level output voltage): Điện thế nhỏ nhất của ngõ ra khi ở mức cao.

-  $V_{OL}(\max)$ : Điện thế ngõ ra mức thấp (Low level output voltage): Điện thế lớn nhất của ngõ ra khi ở mức thấp.

-  $I_{IH}$ : Dòng điện ngõ vào mức cao (High level input current): Dòng điện lớn nhất vào ngõ vào IC khi ngõ vào này ở mức cao.

-  $I_{IL}$ : Dòng điện ngõ vào mức thấp (Low level input current) : Dòng điện ra khỏi ngõ vào IC khi ngõ vào này ở mức thấp

-  $I_{OH}$ : Dòng điện ngõ ra mức cao (High level output current): Dòng điện lớn nhất ngõ ra có thể cấp cho tải khi nó ở mức cao.

-  $I_{OL}$ : Dòng điện ngõ ra mức thấp (Low level output current): Dòng điện lớn nhất ngõ ra có thể nhận khi ở mức thấp.

-  $I_{CCH}, I_{CCL}$ : Dòng điện chạy qua IC khi ngõ ra lần lượt ở mức cao và thấp.

Ngoài ra còn một số thông số khác được nêu ra dưới đây

### *Công suất tiêu tán (Power requirement)*

Mỗi IC khi hoạt động sẽ tiêu thụ một công suất từ nguồn cung cấp  $V_{CC}$  (hay  $V_{DD}$ ). Công suất tiêu tán này xác định bởi điện thế nguồn và dòng điện qua IC. Do khi hoạt động dòng qua IC thường xuyên thay đổi giữa hai trạng thái cao và thấp nên công suất tiêu tán sẽ được tính từ dòng trung bình qua IC và công suất tính được là công suất tiêu tán trung bình

$$P_D(\text{avg}) = I_{CC}(\text{avg}) \cdot V_{CC}$$

Trong đó

$$I_{CC}(\text{avg}) = \frac{I_{CCH} + I_{CCL}}{2}$$

Đối với các cổng logic họ TTL, công suất tiêu tán ở hàng mW và với họ MOS thì chỉ ở hàng nW.

### **Fan-Out:**

Một cách tổng quát, ngõ ra của một mạch logic đòi hỏi phải cấp dòng cho một số ngõ vào các mạch logic khác. Fan Out là số ngõ vào lớn nhất có thể nối với ngõ ra của một IC cùng loại mà vẫn bảo đảm mạch hoạt động bình thường. Nói cách khác Fan Out chỉ khả năng chịu tải của một cổng logic

Ta có hai loại Fan-Out ứng với 2 trạng thái logic của ngõ ra:

$$\text{Fan-Out}_H = \frac{I_{OH}}{I_{IH}}$$

$$\text{Fan-Out}_L = \frac{I_{OL}}{I_{IL}}$$

Thường hai giá trị Fan-Out này khác nhau, khi sử dụng, để an toàn, ta nên dùng trị nhỏ nhất trong hai trị này.

Fan-Out được tính theo đơn vị Unit Load UL (tải đơn vị).

### **Thời trễ truyền (Propagation delays)**

Tín hiệu logic khi truyền qua một cổng luôn luôn có một thời gian trễ.

Có hai loại thời trễ truyền: Thời trễ truyền từ thấp lên cao  $t_{PLH}$  và thời trễ truyền từ cao xuống thấp  $t_{PHL}$ . Hai giá trị này thường khác nhau. Sự thay đổi trạng thái được xác định ở tín hiệu ra. Thí dụ tín hiệu qua một cổng đảo, thời trễ truyền được xác định như ở (H 3.14)

Tùy theo họ IC, thời trễ truyền thay đổi từ vài ns đến vài trăm ns. Thời trễ truyền càng lớn thì tốc độ làm việc của IC càng nhỏ.



(H 3.14)

### **Tích số công suất-vận tốc (speed- power product)**

Để đánh giá chất lượng IC, người ta dùng đại lượng tích số công suất-vận tốc đó là tích số công suất tiêu tán và thời trễ truyền. Thí dụ họ IC có thời trễ truyền là 10 ns và công suất tiêu tán trung bình là 50 mW thì tích số công suất-vận tốc là:

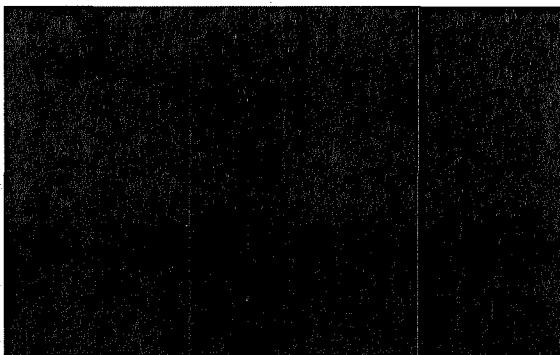
$$10 \text{ ns} \times 50 \text{ mW} = 10 \cdot 10^{-9} \times 50 \cdot 10^{-3} = 500 \cdot 10^{-12} \text{ watt-sec} = 500 \text{ picojoules (pj)}$$

Trong quá trình phát triển của công nghệ chế tạo IC người ta luôn muốn đạt được các IC có công suất tiêu tán và thời trễ truyền càng nhỏ càng tốt. Như vậy một IC có chất lượng càng tốt khi tích số công suất-vận tốc càng nhỏ. Tuy nhiên trên thực tế hai giá trị này thay đổi theo chiều ngược với nhau, nên ta khó mà đạt được các giá trị theo ý muốn, dù sao trong quá trình phát triển của công nghệ chế tạo linh kiện điện tử trị số này luôn được cải thiện.

### **Tính miễn nhiễu (noise immunity)**

Các tín hiệu nhiễu như tia lửa điện, cảm ứng từ có thể làm thay đổi trạng thái logic của tín hiệu do đó ảnh hưởng đến kết quả hoạt động của mạch.

Tính miễn nhiễu của một mạch logic tùy thuộc khả năng dung nạp hiệu thế nhiễu của mạch và được xác định bởi lề nhiễu. Lề nhiễu có được do sự chênh lệch của các điện thế giới hạn (còn được gọi là ngưỡng logic) của mức cao và thấp giữa ngõ ra và ngõ vào của các cổng (H 3.15).



(H 3.15)

Tín hiệu khi vào mạch logic được xem là mức 1 khi có trị  $>V_{IH}(\min)$  và là mức 0 khi  $<V_{IL}(\max)$ . Điện thế trong khoảng giữa không ứng với một mức logic nào nên gọi là vùng bất định. Do có sự khác biệt giữa  $V_{OH}(\min)$  với  $V_{IH}(\min)$  và  $V_{OL}(\max)$  với  $V_{IL}(\max)$  nên ta có 2 giá trị lề nhiễu:

$$\text{Lề nhiễu mức cao: } V_{NH} = V_{OH}(\min) - V_{IH}(\min)$$

$$\text{Lề nhiễu mức thấp: } V_{NL} = V_{IL}(\max) - V_{OL}(\max)$$

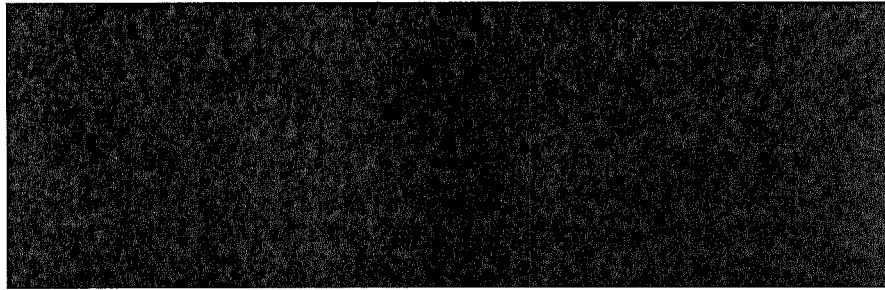
Khi tín hiệu ra ở mức cao đưa vào ngõ vào, bất cứ tín hiệu nhiễu nào có giá trị âm và biên độ  $>V_{NH}$  đều làm cho điện thế ngõ vào rơi vào vùng bất định và mạch không nhận ra được tín hiệu thuộc mức logic nào. Tương tự cho trường hợp ngõ ra ở mức thấp tín hiệu nhiễu có trị dương biên độ  $>V_{NL}$  sẽ đưa mạch vào trạng thái bất định.

### **Logic cấp dòng và logic nhận dòng**

Một mạch logic thường gồm nhiều tầng kết nối với nhau. Tầng cấp tín hiệu gọi là **tầng thúc** và tầng nhận tín hiệu gọi là **tầng tải**. Sự trao đổi dòng điện giữa hai tầng thúc và tải thể hiện bởi logic cấp dòng và logic nhận dòng.

(H 3.16a) cho thấy hoạt động gọi là cấp dòng: Khi ngõ ra mạch logic 1 ở mức cao, nó cấp dòng  $I_{IH}$  cho ngõ vào của mạch logic 2, vai trò như một tải nối mass. Ngõ ra cổng 1 như là một nguồn dòng cấp cho ngõ vào cổng 2

(H 3.16b) cho thấy hoạt động gọi là nhận dòng: Khi ngõ ra mạch logic 1 ở mức thấp, nó nhận dòng  $I_{IL}$  từ ngõ vào của mạch logic 2 xem như nối với nguồn  $V_{CC}$ .



(a)

(b)

(H 3.16)

Thường dòng nhận của tầng thúc khi ở mức thấp có trị khá lớn so với dòng cấp của nó khi ở mức cao, nên người ta hay dùng trạng thái này khi cần gánh những tải tương đối nhỏ, ví dụ khi chỉ cần thúc cho một led, người ta có thể dùng mạch (H 3.17a) mà không thể dùng mạch (H 3.17b).



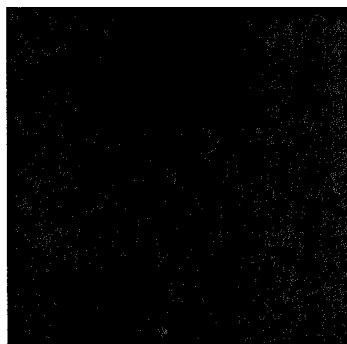
(a)

(H 3.17)

(b)

### ***Tính Schmitt Trigger***

Trong phần giới thiệu về lều, ta thấy còn một khoảng điện thế nằm giữa các ngưỡng logic, đây chính là khoảng điện thế ứng với transistor làm việc trong vùng tác động. Khoảng cách này xác định lều nhiều và có tác dụng làm giảm độ rộng sườn xung (tức làm cho đường dốc lên và dốc xuống của tín hiệu ra dốc hơn) khi qua mạch. Lều nhiều càng lớn khi vùng chuyển tiếp của ngõ vào càng nhỏ, tín hiệu ra thay đổi trạng thái trong một khoảng thời gian càng nhỏ nên sườn xung càng dốc. Tuy nhiên vẫn còn một khoảng sườn xung nằm trong vùng chuyển tiếp nên tín hiệu ra không vuông hoàn toàn. (H 3.18a) và (H 3.18b) minh họa điều đó



(a)

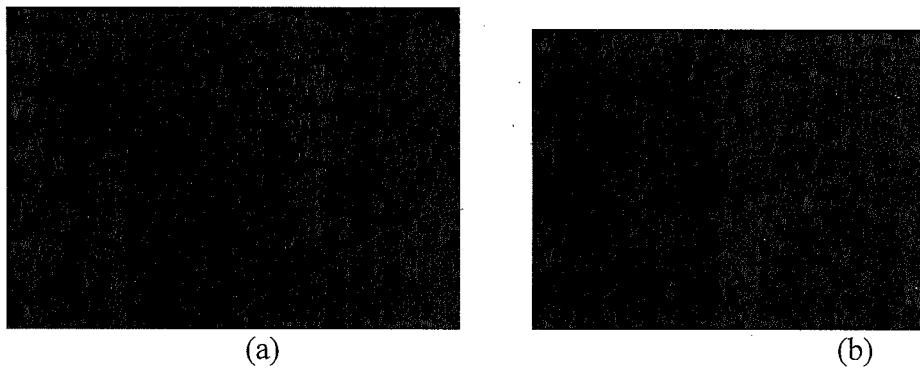


(b)

(H 3.18)

Để cải thiện hơn nữa dạng tín hiệu ngõ ra, bảo đảm tính miễn nhiễu cao, người ta chế tạo các công có tính trễ điện thế (H 3.19a), được gọi là công Schmitt Trigger (H 3.19b) mô tả mối quan hệ giữa  $V_{out}$  và  $V_{in}$  của một công đảo Schmitt Trigger.





(a)

(H 3.19)

(b)

(H 3.20a&b) là ký hiệu các cổng Schmitt Trigger.



(a)

(b)

(H 3.20)

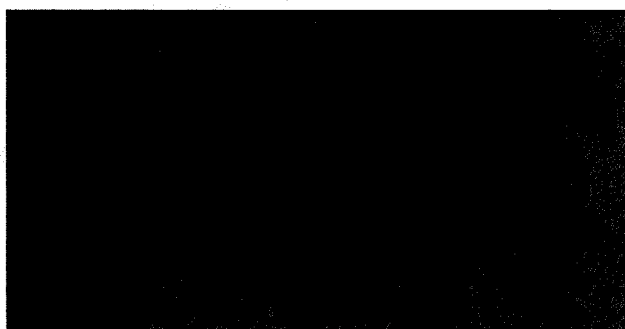
## HỌ TTL

Trong quá trình phát triển của công nghệ chế tạo mạch số ta có các họ: RTL (Resistor-transistor logic), DCTL (Direct couple-transistor logic), RCTL (Resistor-Capacitor-transistor logic), DTL (Diod-transistor logic), ECL (Emitter-couple logic) v.v.... Đến bây giờ tồn tại hai họ có nhiều tính năng kỹ thuật cao như thời trễ truyền nhỏ, tiêu hao công suất ít, đó là họ TTL (transistor-transistor logic) dùng công nghệ chế tạo BJT và họ MOS (Công nghệ chế tạo MOS)

Dưới đây, lần lượt khảo sát các cổng logic của hai họ TTL và MOS

### **Cổng cơ bản họ TTL**

Lấy cổng NAND 3 ngõ vào làm thí dụ để thấy cấu tạo và vận hành của một cổng cơ bản



(H 3.21)

Khi một trong các ngõ vào A, B, C xuống mức không  $T_1$  dẫn đưa đến  $T_2$  ngưng,  $T_3$  ngưng, ngõ ra Y lên cao; khi cả 3 ngõ vào lên cao,  $T_1$  ngưng,  $T_2$  dẫn,  $T_3$  dẫn, ngõ ra Y xuống thấp. Đó chính là kết quả của cổng NAND.

Tụ  $C_L$  trong mạch chính là tụ ký sinh tạo bởi sự kết hợp giữa ngõ ra của mạch (tăng thúc) với ngõ vào của tầng tải, khi mạch hoạt động tụ sẽ nạp điện qua  $R_4$  (lúc  $T_3$  ngưng) và phóng qua  $T_3$  khi transistor này dẫn do đó thời trễ truyền của mạch quyết định bởi  $R_4$  và  $C_L$ , khi  $R_4$  nhỏ mạch hoạt động nhanh nhưng công suất tiêu thụ lúc đó lớn, muốn giảm công suất phải tăng  $R_4$  nhưng như vậy thời trễ truyền sẽ lớn hơn (mạch giao hoán chậm hơn). Để giải quyết khuyết điểm này đồng thời thỏa mãn một số yêu cầu khác, người ta đã chế tạo các cổng logic với các kiểu ngõ ra khác nhau.

### Các kiểu ngõ ra

@ Ngõ ra totempole



(H 3.22)

$R_4$  trong mạch cơ bản được thay thế bởi cụm  $T_4$ ,  $R_C$  và Diode  $D$ , trong đó  $R_C$  có trị rất nhỏ, không đáng kể.  $T_2$  bây giờ giữ vai trò mạch đảo pha: khi  $T_2$  dẫn thì  $T_3$  dẫn và  $T_4$  ngưng,  $Y$  xuống thấp, khi  $T_2$  ngưng thì  $T_3$  ngưng và  $T_4$  dẫn, ngõ ra  $Y$  lên cao. Tụ  $C_L$  nạp điện qua  $T_4$  khi  $T_4$  dẫn và phóng qua  $T_3$  (dẫn), thời hằng mạch rất nhỏ và kết quả là thời trễ truyền nhỏ. Ngoài ra do  $T_3$  &  $T_4$  luân phiên ngưng tương ứng với 2 trạng thái của ngõ ra nên công suất tiêu thụ giảm đáng kể. Diode  $D$  có tác dụng nâng điện thế cực B của  $T_4$  lên để bảo đảm khi  $T_3$  dẫn thì  $T_4$  ngưng.

Mạch này có khuyết điểm là không thể nối chung nhiều ngõ ra của các cổng khác nhau vì có thể gây hư hỏng khi các trạng thái logic của các cổng này khác nhau.

@ Ngõ ra cực thu để hở



(H 3.23)

Ngõ ra cực thu để hở có một số lợi điểm sau:

- Cho phép kết nối các ngõ ra của nhiều cổng khác nhau, nhưng khi sử dụng phải mắc một điện trở từ ngõ ra lên nguồn  $V_{cc}$ , gọi là **điện trở kéo lên**, trị số của điện trở này có thể được chọn lớn hay nhỏ tùy theo yêu cầu có lợi về mặt công suất hay tốc độ làm việc.

Điểm nổi chung của các ngõ ra có tác dụng như một cổng AND nên ta gọi là điểm AND (H 3.24)

- Người ta cũng chế tạo các IC ngõ ra có cực thu để hở cho phép điện trở kéo lên mắc vào nguồn điện thế cao, dùng cho các tải đặc biệt hoặc dùng tạo sự giao tiếp giữa họ TTL với CMOS dùng nguồn cao.

Thí dụ IC 7406 là loại cổng đảo có ngõ ra cực thu để hở có thể mắc lên nguồn 24 V (H 3.25)

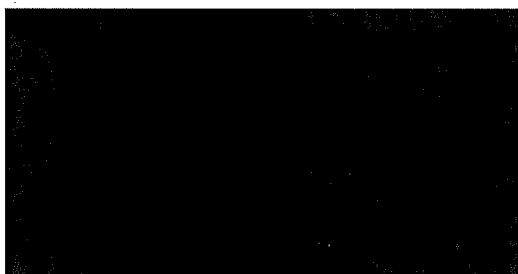


(H 3.24)



(H 3.25)

@ Ngõ ra ba trạng thái



(H 3.26)



(H 3.27)

Mạch (H 3.26) là một cổng đảo có ngõ ra 3 trạng thái, trong đó  $T_4$  &  $T_5$  được mắc Darlington để cấp dòng ra lớn cho tải. Diod D nối vào ngõ vào C để điều khiển. Hoạt động của mạch giải thích như sau:

- Khi  $C=1$ , Diod D ngưng dẫn, mạch hoạt động như một cổng đảo
- Khi  $C=0$ , Diod D dẫn, cực thu  $T_2$  bị ghim áp ở mức thấp nên  $T_3$ ,  $T_4$  &  $T_5$  đều ngưng, ngõ ra mạch ở trạng thái tổng trở cao.

Ký hiệu của cổng đảo ngõ ra 3 trạng thái, có ngõ điều khiển C tác động mức cao và bảng sự thật cho ở (H 3.27)

Cũng có các cổng đảo và cổng đệm 3 trạng thái với ngõ điều khiển C tác động mức thấp mà SV có thể tự vẽ ký hiệu và bảng sự thật.

(H 3.28) là một ứng dụng của cổng đệm có ngõ ra 3 trạng thái: Mạch chọn dữ liệu



Vận chuyển: Ứng với một giá trị địa chỉ AB, một ngõ ra mạch giải mã địa chỉ được tác động (lên cao) cho phép một công mở và dữ liệu ở ngõ vào công đó được truyền ra ngõ ra. Thí dụ khi AB = 00,  $Y_0 = 1$  ( $Y_1=Y_2=Y_3=0$ )  $G_1$  mở,  $D_0$  truyền qua  $G_1$  đến ngõ ra, trong lúc  $G_2, G_3, G_4$  đóng, có ngõ ra ở trạng thái Z cao, không ảnh hưởng đến hoạt động của mạch.

### Đặc tính các loạt TTL

Các IC số họ TTL được sản xuất lần đầu tiên vào năm 1964 bởi hãng Texas Instrument Corporation của Mỹ, lấy số hiệu là 74XXXX & 54XXXX. Sự khác biệt giữa 2 họ 74XXXX và 54XXXX chỉ ở hai điểm:

74:  $V_{CC}=5 \pm 0,5$  V và khoảng nhiệt độ hoạt động từ  $0^\circ$  C đến  $70^\circ$  C

54:  $V_{CC}=5 \pm 0,25$  V và khoảng nhiệt độ hoạt động từ  $-55^\circ$  C đến  $125^\circ$

C Các tính chất khác hoàn toàn giống nhau nếu chúng có cùng số.

Trước số 74 thường có thêm ký hiệu để chỉ hãng sản xuất. Thí dụ SN của hãng Texas, DM của National Semiconductor, S của Signetics

Ngoài ra trong quá trình phát triển, các thông số kỹ thuật (nhất là tích số công suất vận tốc) luôn được cải tiến và ta có các loạt khác nhau: 74 chuẩn, 74L (Low power), 74 H (High speed), 74S (Schottky), 74LS (Low power Schottky), 74AS (Advance Schottky), 74ALS (Advance Low power Schottky), 74F (Fast, Fair Child).

Bảng 3.1 cho thấy một số tính chất của các loạt kể trên:

Thông số kỹ thuật	74	74L	74H	74S	74LS	74AS	74ALS	74F
Thời trễ truyền (ns)	9	33	6	3	9,5	1,7	4	3
Công suất tiêu tán (mW)	10	1	23	20	2	8	1,2	6
Tích số công suất vận tốc (pJ)	90	33	138	60	19	13,6	4,8	18
Tần số xung $C_K$ max (MHz)	35	3	50	125	45	200	70	100
Fan Out (cùng loạt)	10	20	10	20	20	40	20	33
<b>Điện thế</b>								
$V_{OH}(\min)$	2,4	2,4	2,4	2,7	2,7	2,5	2,5	2,5
$V_{OL}(\max)$	0,4	0,4	0,4	0,5	0,5	0,5	0,4	0,5
$V_{IH}(\min)$	2,0	2,0	2,0	2,0	2,0	2,0	2,0	2,0
$V_{IL}(\max)$	0,8	0,7	0,8	0,8	0,8	0,8	0,8	0,8

Bảng 3.1

- Loạt 74S: Các transistor trong mạch được mắc thêm một Diod Schottky giữa hai cực CB với mục đích giảm thời gian chuyển trạng thái của transistor do đó làm giảm thời trễ truyền.

- Loạt 74AS và 74ALS là cải tiến của 74S để làm giảm hơn nữa giá trị tích số Công suất - Vận tốc.

- Loạt 74F: Dùng kỹ thuật đặc biệt làm giảm diện dung ký sinh do đó cải thiện thời trễ truyền của công.

### HOMOS

Gồm các IC số dùng công nghệ chế tạo của transistor MOSFET loại tăng, kênh N và kênh P. Với transistor kênh N ta có NMOS, transistor kênh P ta có PMOS và nếu dùng cả hai loại transistor kênh P & N ta có CMOS. Tính năng kỹ thuật của loại NMOS

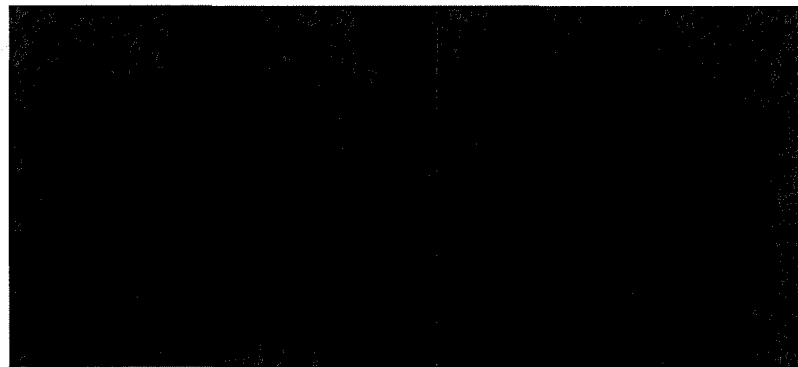
và PMOS có thể

nói là giống nhau, trừ nguồn cấp điện có chiều ngược với nhau do đó ta chỉ xét loại NMOS và CMOS.

Các transistor MOS dùng trong IC số cũng chỉ hoạt động ở một trong 2 trạng thái: dẫn hoặc ngưng.

- Khi dẫn, tùy theo nồng độ pha của chất bán dẫn mà transistor có nội trở rất nhỏ (từ vài chục  $\Omega$  đến hàng trăm  $\text{k}\Omega$ ) tương đương với một khóa đóng.
- Khi ngưng, transistor có nội trở rất lớn (hàng  $10^{10}\Omega$ ), tương đương với một khóa hở.

### Cổng cơ bản NMOS



(a) (b) (c)  
(H 3.29)

(H 3.29a), (H 3.29b) và (H3.29c) là các cổng NOT, NAND và NOR dùng NMOS. Bảng 3.2 cho thấy quan hệ giữa các điện thế của các ngõ vào, ra cổng NOT

$V_{in}$	$T_1$	$T_2$	$V_{out}$
0V (logic 0)	$R_{ON} =$	$R_{OFF} = 10^{10}$	+5V (logic 1)
+5V (logic 1)	$R_{ON} =$	$R_{ON} =$	0,05V (logic 0)

Bảng 3.2

Ngoài ra vận hành của cổng NAND và NOR được giải thích như sau:

#### ❖ Cổng NAND:

- Khi 2 ngõ vào nối lên mức cao,  $T_2$  và  $T_3$  dẫn, ngõ ra xuống thấp.
  - Khi có 1 ngõ vào nối xuống mức thấp, một trong 2 transistor  $T_2$  hoặc  $T_3$  ngưng, ngõ ra lên cao.
- Đó chính là kết quả của cổng NAND 2 ngõ vào.

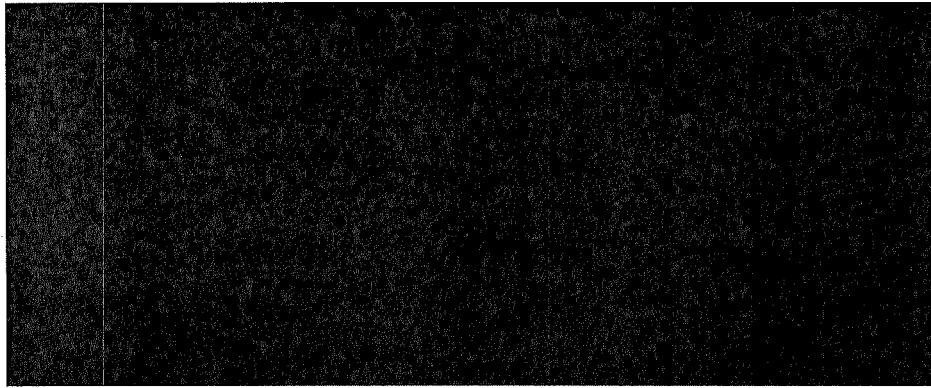
#### ❖ Cổng NOR:

- Khi 2 ngõ vào nối xuống mức thấp,  $T_2$  và  $T_3$  ngưng, ngõ ra lên cao.
- Khi có 1 ngõ vào nối lên mức cao, một trong 2 transistor  $T_2$  hoặc  $T_3$  dẫn, ngõ ra xuống thấp.

Đó chính là kết quả của cổng NOR 2 ngõ vào.

### Cổng cơ bản CMOS

Họ CMOS sử dụng hai loại transistor kênh N và P với mục đích cải thiện tích số công suất vận tốc, mặc dù khả năng tích hợp thấp hơn loại N và P. (H 3.30a), (H 3.30b) và (H 3.30c) là các cổng NOT, NAND và NOR họ CMOS



(a) (b)(c) (H 3.30)

Bảng 3.3 cho thấy quan hệ điện thế của các ngõ vào, ra cổng NOT

$V_{in}$	$T_1$	$T_2$	$V_{out}$
$V_{DD}$ (logic1)	$R_{OFF}=10^{10} \text{ } \wedge$	$R_{ON} =$	0V (logic 0)
0V (logic0)	$R_{ON} = 1 \text{ K} \wedge$	$R_{OFF}=10^{10}$	$V_{DD}$ (logic 1)

Bảng 3.3

Ngoài ra vận hành của cổng NAND và NOR được giải thích như sau:

❖ Cổng NAND:

- Khi 2 ngõ vào nổi lên mức cao,  $T_1$  và  $T_2$  ngưng,  $T_3$  và  $T_4$  dẫn, ngõ ra xuống thấp.
  - Khi có 1 ngõ vào nổi xuống mức thấp, một trong 2 transistor  $T_3$  hoặc  $T_4$  ngưng, một trong 2 transistor  $T_1$  hoặc  $T_2$  dẫn, ngõ ra lên cao.
- Đó chính là kết quả của cổng NAND 2 ngõ vào.

❖ Cổng NOR:

- Khi 2 ngõ vào nổi xuống mức thấp,  $T_1$  và  $T_2$  dẫn,  $T_3$  và  $T_4$  ngưng, ngõ ra lên cao.
  - Khi có 1 ngõ vào nổi lên mức cao, một trong 2 transistor  $T_3$  hoặc  $T_4$  dẫn, một trong 2 transistor  $T_1$  hoặc  $T_2$  ngưng, ngõ ra xuống thấp.
- Đó chính là kết quả của cổng NOR 2 ngõ vào.

**Các cổng CMOS khác**

Người ta cũng sản xuất các cổng CMOS với cực Drain để hở và ngõ ra 3 trạng thái để sử dụng trong các trường hợp đặc biệt như họ TTL



(a)



(H 3.31)

(b)

(H 3.31a) là một cổng NOT có cực D để hở, khi sử dụng phải có điện trở kéo lên

(H 3.31b) là một cổng NOT có ngõ ra 3 trạng thái:

- Khi ngõ vào Enable = 1,  $T_1$  và  $T_4$  dẫn, mạch hoạt động như là cổng đảo,
- Khi ngõ vào Enable = 0,  $T_1$  và  $T_4$  đều ngưng đưa mạch vào trạng thái Z cao.

Ngoài ra lợi dụng tính chất của transistor MOS có nội trở rất nhỏ khi dẫn, người ta cũng chế tạo các mạch có khả năng truyền tín hiệu theo 2 chiều, gọi là khóa 2 chiều. (H 3.32) là một khóa 2 chiều với A là ngõ vào điều khiển. Khi  $A = 0$  khóa hở, khi  $A = 1$ , khóa đóng cho tín hiệu truyền qua theo 2 chiều



(H 3.32)



A	X to Y	Y to X
0	OFF	OFF
1	ON	ON

Vận hành:  $T_3$  và  $T_4$  vai trò là một cổng đảo

- Khi  $A = 0$ , cực G của  $T_2$  ở mức thấp nên  $T_2$  (kênh N) ngưng, cực G của  $T_1$  (kênh P) ở mức cao nên  $T_1$  ngưng, mạch tương đương với khóa hở.
- Khi  $A = 1$ , cực G của  $T_2$  ở mức cao nên  $T_2$  dẫn, cực G của  $T_1$  ở mức thấp nên  $T_1$  dẫn, mạch tương đương với khóa đóng. Tín hiệu truyền qua một chiều nhờ  $T_1$  (loại P) và theo chiều ngược lại nhờ  $T_2$  (loại N)

Biên độ của tín hiệu  $V_i$  truyền qua khóa phải thỏa điều kiện  $0 < V_i < V_{DD}$ .

Như vậy nếu ta sử dụng nguồn  $\pm V_{DD}$  thì khóa cho tín hiệu xoay chiều đi qua.

### **Đặc tính của họ MOS**

Một số tính chất chung của các cổng logic họ MOS (NMOS, PMOS và CMOS) có thể kể ra như sau:

- Nguồn cấp điện :  $V_{DD}$  từ 3V đến 15V
- Mức logic:  $V_{OL}(\max) = 0V$   $V_{OH}(\min) = V_{DD}$   
 $V_{IL}(\max) = 30\% V_{DD}$   $V_{IH}(\min) = 70\% V_{DD}$
- Lệ nhiễu :  $V_{NH} = 30\% V_{DD}$   $V_{NL} = 30\% V_{DD}$

Với nguồn 5V, lệ nhiễu khoảng 1,5V, rất lớn so với họ TTL.

- Thời trễ truyền tương đối lớn, khoảng vài chục ns, do điện dung ký sinh ở ngã vào và tổng trở ra của transistor khá lớn.

- Công suất tiêu tán tương đối nhỏ, hàng nW, do dòng qua transistor MOS rất nhỏ.
- Số Fan Out: 50 UL

Do tổng trở vào của transistor MOS rất lớn nên dòng tải cho các cổng họ MOS rất nhỏ, do đó số Fan Out của họ MOS rất lớn, tuy nhiên khi mắc nhiều tầng tải vào một tầng thúc thì điện dung ký sinh tăng lên (gồm nhiều tụ mắc song song) ảnh hưởng đến thời gian giao hoán của mạch nên khi dùng ở tần số cao người ta giới hạn số Fan Out là 50, nghĩa là một cổng MOS có thể cấp dòng cho 50 cổng tải cùng loạt.

- Như đã nói ở trên, CMOS có cải thiện thời trễ truyền so với loại NMOS và PMOS, tuy nhiên mật độ tích hợp của CMOS thì nhỏ hơn hai loại này. Dù sao so với họ TTL thì mật độ tích hợp của họ MOS nói chung lớn hơn rất nhiều, do đó họ MOS rất thích hợp để chế tạo dưới dạng LSI và VLSI.

### **Các loạt CMOS**

CMOS có hai ký hiệu: 4XXX do hãng RCA chế tạo và 14XXX của hãng MOTOROLA, có hai loạt 4XXXA (14XXXA) và 4XXXB (14XXXB), loạt B ra đời sau có cải thiện dòng ra.

Ngoài ra còn có các loạt :

- 74C : CMOS có cùng sơ đồ chân và chức năng với IC TTL nếu có cùng số. Thí dụ IC 74C74 là IC gồm 2 FF D tác động bởi cạnh xung đồng hồ giống như IC 7474 của TTL. Hầu hết (nhưng không tất cả) các thông số của loạt 74C giống với 74 TTL nên ta có thể thay thế 2 loại này cho nhau được.

- 74HC (High speed CMOS), 74HCT: Đây là loạt cải tiến của 74C, tốc độ giao hoán có thể so sánh với 74LS, riêng 74HCT thì hoàn toàn tương thích với TTL kể cả các mức logic. Đây là loạt IC CMOS được dùng rộng rãi.

- 74AC và 74ACT (Advance CMOS) cải tiến của 74 HC và HCT về mặt nhiễu bằng cách sắp xếp lại thứ tự các chân, do đó nó không tương thích với TTL về sơ đồ chân.

## **GIAO TIẾP GIỮA CÁC HỌ IC SỐ**

Giao tiếp là thực hiện việc kết nối ngã ra của một mạch hay hệ thống với ngã vào của mạch hay hệ thống khác. Do tính chất về điện khác nhau giữa hai họ TTL và CMOS nên việc giao tiếp giữa chúng trong nhiều trường hợp không thể nối trực tiếp được mà phải nhờ một mạch trung gian nối giữa tầng thúc và tầng tải sao cho điện thế tín hiệu ra ở tầng thúc phù hợp với tín hiệu vào của tầng tải và dòng điện tầng thúc phải đủ cấp cho tầng tải.

---

CMOS ( $V_{DD} =$

TTL



Thông số	5V)						
	4000B	74HC	74HCT	74	74LS	74AS	74ALS
$V_{IH}(\text{min})$	3,5V	3,5V	2,0V	2,0V	2,0V	2,0V	2,0V
$V_{IL}(\text{max})$	1,5V	1,0V	0,8V	0,8V	0,8V	0,8V	0,8V
$V_{OH}(\text{min})$	4,95V	4,9V	4,9V	2,4V	2,7V	2,7V	2,7V
$V_{OL}(\text{max})$	0,05V	0,1V	0,1V	0,4V	0,5V	0,5V	0,4V
$I_{IH}(\text{max})$	1 $\mu$ A	1 $\mu$ A	1 $\mu$ A	40 $\mu$ A	20 $\mu$ A	200 $\mu$ A	20 $\mu$ A
$I_{IL}(\text{max})$	1 $\mu$ A	1 $\mu$ A	1 $\mu$ A	1,6 mA	0,4 mA	2 mA	100 $\mu$ A
$I_{OH}(\text{max})$	0,4 mA	4 mA	4 mA	0,4 mA	0,4 mA	2 mA	0,4 mA
$I_{OL}(\text{max})$	0,4 mA	4 mA	4 mA	16 mA	8 mA	20 mA	8 mA

Bảng 3.4

Có thể nói điều kiện để thúc trực tiếp

- Khi dòng điện ra của tầng thúc lớn hơn hoặc bằng dòng điện vào của tầng tải ở cả hai trạng thái thấp và cao.

- Khi hiệu thế ngã ra của tầng thúc ở hai trạng thái thấp và cao phù hợp với điện thế vào của tầng tải.

Như vậy, trước khi xét các trường hợp cụ thể ta xem qua bảng kê các thông số của hai họ IC

## TTL thúc CMOS

- TTL thúc CMOS dùng điện thế thấp ( $V_{DD} = 5V$ ):

Từ bảng 3.4 dòng điện vào của CMOS có trị rất nhỏ so với dòng ra của các loại TTL, vậy về dòng điện không có vấn đề

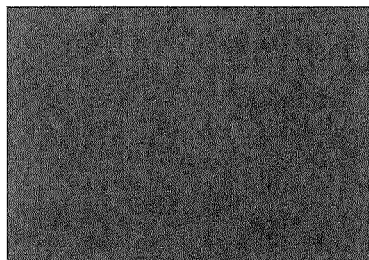
Tuy nhiên khi so sánh hiệu thế ra của TTL với hiệu thế vào của CMOS ta thấy  $V_{OH}(\text{max})$  của tất cả các loại TTL đều khá thấp so với  $V_{IH}(\text{min})$  của TTL, như vậy phải có biện pháp nâng hiệu thế ra của TTL lên. Điều này thực hiện được bằng một điện trở kéo lên mắc ở ngã ra của IC TTL (H 3.33)

- TTL thúc 74 HCT:

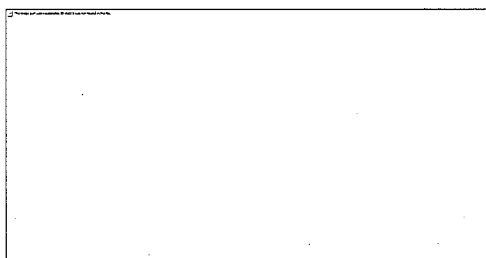
Như đã nói trước đây, riêng loạt 74HCT là loạt CMOS được thiết kế tương thích với TTL nên có thể thực hiện kết nối mà không cần điện trở kéo lên.

- TTL thúc CMOS dùng nguồn cao ( $V_{DD} = +10V$ )

Ngay cả khi dùng điện trở kéo lên, điện thế ngã ra mức cao của TTL vẫn không đủ cấp cho ngã vào CMOS, người ta phải dùng một công đệm có ngã ra để hở có thể dùng nguồn cao (Thí dụ IC 7407) để thực hiện sự giao tiếp (H 3.34)



(H 3.33)



(H 3.34)

### CMOS thúc TTL

- CMOS thúc TTL ở trạng thái cao:

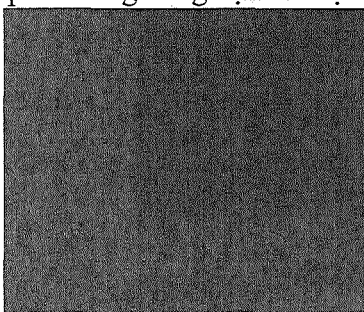
Bảng 3.4 cho thấy điện thế ra và dòng điện ra mức cao của CMOS đủ để cấp cho TTL. Vậy không có vấn đề ở trạng thái cao

- CMOS thúc TTL ở trạng thái thấp:

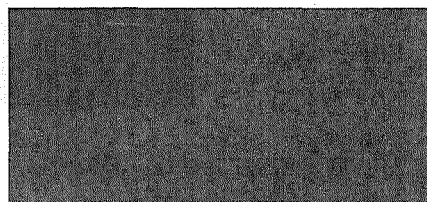
Dòng điện vào ở trạng thái thấp của TTL thay đổi trong khoảng từ  $100 \mu\text{A}$  đến  $2 \text{ mA}$ . Hai loạt 74HC và 74HCT có thể nhận dòng  $4 \text{ mA}$ . Vậy hai loạt này có thể giao tiếp với một IC TTL mà không có vấn đề. Tuy nhiên, với loạt 4000B,  $I_{OL}$  rất nhỏ không đủ để giao tiếp với ngay cả một IC TTL, người ta phải dùng một cổng đệm để nâng dòng tải của loạt 4000B trước khi thúc vào IC 74LS (H 3.35)

- CMOS dùng nguồn cao thúc TTL:

Có một số IC loạt 74LS được chế tạo đặc biệt có thể nhận điện thế ngã vào cao khoảng  $15\text{V}$  có thể được thúc trực tiếp bởi CMOS dùng nguồn cao, tuy nhiên đa số IC TTL không có tính chất này, vậy để có thể giao tiếp với CMOS dùng nguồn cao, người ta phải dùng cổng đệm để hạ điện thế ra xuống cho phù hợp với IC TTL (H 3.36)



(H 3.35)



(H 3.36)

Vài thí dụ dùng cổng thiết kế mạch

1. Dùng cổng NAND 2 ngã vào thiết kế mạch tạo hàm  $Y = f(A,B,C) = 1$  khi thỏa các điều kiện sau:

- a.  $A=0, B=1$  và  $C=1$
- b.  $A=1, B=1$  bất chấp  $C$

Giải

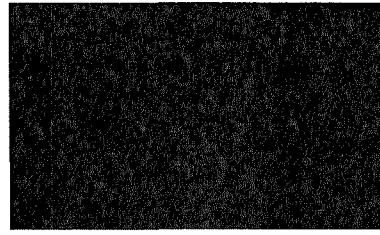
Dựa vào điều kiện của bài toán ta có bảng sự thật của hàm  $Y$

A	B	C	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Rút gọn hàm:



$$Y = AB + BC$$



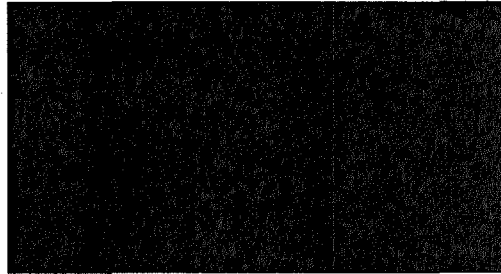
(H 3.37)

Để dùng toàn cổng NAND tạo hàm, ta dùng định lý De Morgan, biến đổi hàm Y:

$$Y = \overline{\overline{AB + BC}} = \overline{\overline{AB} \cdot \overline{BC}}$$

Và mạch có dạng (H 3.37)

2. Cho mạch



(H P3.38)

- Viết biểu thức hàm Y theo các biến A, B, C.
- Rút gọn hàm logic này
- Thay thế mạch trên bằng một mạch chỉ gồm cổng NAND 2 ngõ vào

Giải

- Ta có  $Y = \overline{A} \cdot \overline{B} \cdot \overline{C} + \overline{A} \cdot \overline{B} \cdot C + \overline{A} \cdot B \cdot \overline{D}$
- Rút gọn

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

- Vẽ mạch thay thế dùng cổng NAND 2 ngõ vào  
Trước nhất ta vẽ mạch tương ứng hàm rút gọn, sau đó dùng biến đổi cổng



(H P3.39)

000

## BÀI TẬP

1. Thiết kế mạch thực hiện các hàm sau đây dùng toàn cổng NAND 2 ngõ vào:

- $f(A,B,C) = 1$  nếu  $(ABC)_2$  là số chẵn.
- $f(A,B,C) = 1$  nếu có ít nhất 2 biến = 1.
- $f(A,B,C) = 1$  nếu số nhị phân  $(ABC)_2 > 5$ .
- $f(A,B,C) = 1$  nếu số biến có giá trị 1 là số chẵn.
- $f(A,B,C) = 1$  nếu có một và chỉ một biến = 1.

2. Thiết kế mạch gồm 2 ngõ vào D, E và 2 ngõ ra P, C thỏa các điều kiện sau đây:

- Nếu  $E = 1$   $D = 0 \Rightarrow P = 1, C = 0$
- Nếu  $E = 1$   $D = 1 \Rightarrow P = 0, C = 1$
- Nếu  $E = 0$   $D$  bất kỳ  $\Rightarrow P = 1, C = 1$

3. Hàm logic  $F(A, B, C)$  thỏa tính chất sau đây :

$F(A,B,C) = 1$  nếu có một và chỉ một biến bằng 1

- Lập bảng sự thật cho hàm F.
- Vẽ mạch logic tạo hàm F.

4. Thiết Kế mạch tạo hàm  $Y = \overline{A}.\overline{B}.\overline{C} + A\overline{B}.\overline{C} + \overline{A}.\overline{B}.C$  bằng các cổng NAND 2 ngõ vào

5. Hàm  $F(A,B,C)$  xác định bởi bảng sự thật

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

- Dùng bản đồ Karnaugh rút gọn hàm F.
- Vẽ sơ đồ mạch logic thực hiện hàm F.
- Vẽ lại mạch chỉ dùng cổng NOR hai ngõ vào.

6. Rút gọn hàm logic :

$f(A,B,C,D) = \Sigma(0,1, 2, 4, 5, 8)$ , A = MSB. Hàm không xác định với các tổ hợp biến (3,

7,

10 Dùng số cổng NOR ít nhất để thực hiện mạch tạo hàm trên.

).

7. Hàm  $f(A,B,C) = 1$  khi số biến = 1 là số chẵn

- Viết biểu thức logic của hàm  $f(A,B,C)$  theo tổ hợp biến A,B,C.
- Dùng các cổng EX-OR để thực hiện mạch tạo hàm trên.

8. Một mạch tổ hợp nhận vào một số nhị phân  $A=A_3A_2A_1A_0$  ( $A_0$  là LSB) tạo ra ở ngõ ra Y ở mức cao khi và chỉ khi  $0010 < A < 1000$ . Hãy thiết kế mạch với:

- Cấu trúc NAND-NAND.
- Toàn cổng NAND 2 ngõ vào.

9. Một mạch tổ hợp nhận vào một số BCD, có tên là X. Ngã ra của mạch lên 1 khi thỏa điều kiện  $1_{10} \leq X \leq 5_{10}$ .

Hãy thiết kế mạch tổ hợp trên, dùng toàn cổng NAND 2 ngõ vào.

10. Hàm  $f(A,B,C,D) = 1$  khi có ít nhất 3 biến = 1

- Viết biểu thức logic của hàm  $f(A,B,C,D)$  theo tổ hợp biến A,B,C,D.
- Dùng các cổng NAND 2 ngõ vào (số cổng ít nhất) để thực hiện mạch tạo hàm trên.

# CHƯƠNG 4: MẠCH ĐẾM VÀ THANH GHI MẠCH ĐẾM

Các mạch số được chia ra làm hai loại: Mạch tổ hợp và Mạch tuần tự.

- Mạch tổ hợp: Trạng thái ngõ ra chỉ phụ thuộc vào tổ hợp các ngõ vào khi tổ hợp này đã ổn định. Ngõ ra Q của mạch tổ hợp là hàm logic của các biến ngõ vào A, B, C . . .

$$Q = f(A, B, C \dots)$$

- Mạch tuần tự: Trạng thái ngõ ra không những phụ thuộc vào tổ hợp các ngõ vào mà còn phụ thuộc trạng thái ngõ ra trước đó. Ta nói mạch tuần tự có tính nhớ. Ngõ ra  $Q_+$  của mạch tuần tự là hàm logic của các biến ngõ vào A, B, C . . . và ngõ ra Q trước đó.

$$Q_+ = f(Q, A, B, C \dots)$$

Chương này nghiên cứu một số mạch tổ hợp thông dụng thông qua việc thiết kế một số mạch đơn giản và khảo sát một số IC trên thực tế.

## MẠCH MÃ HÓA

Mã hóa là gán các ký hiệu cho các đối tượng trong một tập hợp để thuận tiện cho việc thực hiện một yêu cầu cụ thể nào đó. Thí dụ mã BCD gán số nhị phân 4 bit cho từng số mã của số thập phân (từ 0 đến 9) để thuận tiện cho máy đọc một số có nhiều số mã; mã Gray dùng tiện lợi trong việc tối giản các hàm logic . . . Mạch chuyển từ mã này sang mã khác gọi là mạch chuyển mã, cũng được xếp vào loại mạch mã hóa. Thí dụ mạch chuyển số nhị phân 4 bit sang số Gray là một mạch chuyển mã.

### 4.1.1 Mạch mã hóa $2^n$ đường sang n đường

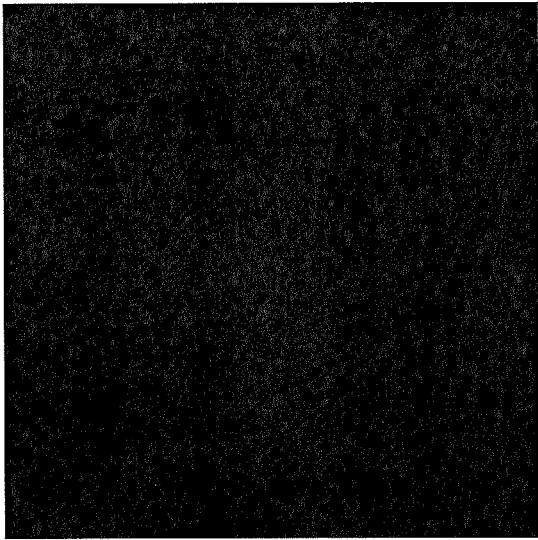
Một số nhị phân n bit cho  $2^n$  tổ hợp số khác nhau. Vậy ta có thể dùng số n bit để mã cho  $2^n$  ngõ vào khác nhau, khi có một ngõ vào được chọn bằng cách đưa nó lên mức tác động, ở ngõ ra sẽ chỉ báo số nhị phân tương ứng. Đó là mạch mã hóa  $2^n$  đường sang n đường.

(H 4.1) là mô hình một mạch mã hóa  $2^n$  đường sang n đường.

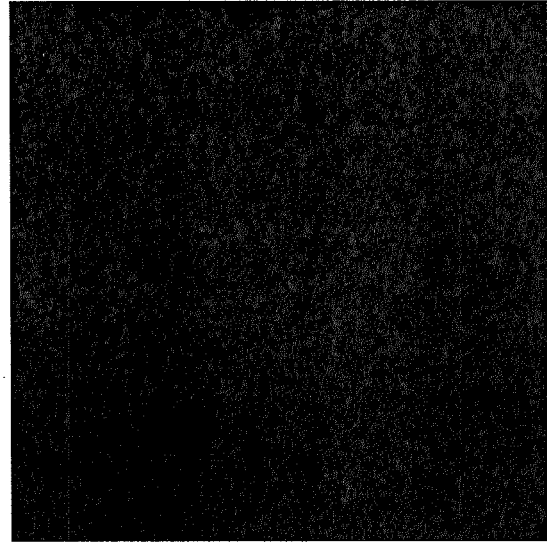
- (H 4.1a) là mạch có ngõ vào và ra tác động cao: Khi các ngõ vào đều ở mức thấp, mạch chưa hoạt động, các ngõ ra đều ở mức thấp. Khi có một ngõ vào được tác động bằng cách ấn khóa K tương ứng để đưa ngõ vào đó lên mức cao, các ngõ ra sẽ cho số nhị phân tương ứng.

- (H 4.1b) là mạch có ngõ vào và ra tác động thấp. Hoạt động tương tự như mạch trên nhưng có mức tác động ngược lại. (trong mô hình (H 4.1b) ký hiệu dấu 0 ở ngõ ra để chỉ mức tác động thấp, còn ở ngõ vào không có dấu 0 vì là mạch thật)

Trong trường hợp ngõ ra có mức tác động thấp, muốn đọc đúng số nhị phân ở ngõ ra, ta phải đảo các bit để đọc.



(a)



(b)

(H 4.1)

Dĩ nhiên, người ta cũng có thể thiết kế theo kiểu ngã vào tác động thấp và ngã ra tác động cao hay ngược lại. Trên thực tế, ta có thể có bất cứ loại ngã vào hay ra tác động theo bất cứ kiểu nào (mức cao hay thấp).

Ngoài ra, để tránh trường hợp mạch cho ra một mã sai khi người sử dụng vô tình (hay cố ý) tác động đồng thời vào hai hay nhiều ngã vào, người ta thiết kế các mạch mã hóa ưu tiên: là mạch chỉ cho ra một mã duy nhất có tính ưu tiên khi có nhiều ngã vào cùng được tác động.

### Mã hóa ưu tiên 4 đường sang 2 đường

Thiết kế mạch mã hóa 4 đường sang 2 đường, ưu tiên cho mã có trị cao, ngã vào và ra tác động cao

Bảng sự thật và sơ đồ mạch (H 4.2)

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

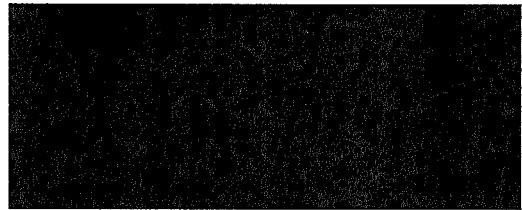


x	1	0	0	0	1
x	x	1	0	1	0
x	x	x	1	1	1

Bảng 4.1

Nhận thấy biến 0 trong bảng sự thật không ảnh hưởng đến kết quả nên ta chỉ vẽ bảng Karnaugh cho 3 biến 1, 2 và 3. Lưu ý là do trong bảng sự thật có các trường hợp bất chấp của biến nên ứng với một trị riêng của hàm ta có thể có đến 2 hoặc 4 số 1 trong bảng Karnaugh. Thí dụ với trị 1 của cả 2 hàm  $A_1$  và  $A_0$  ở dòng cuối cùng đưa đến 4 số 1 trong các ô 001, 011, 101 và 111 của 3 biến 123.

Từ bảng Karnaugh, ta có kết quả và mạch tương ứng. Trong mạch không có ngã vào 0, điều này được hiểu là mạch sẽ chỉ báo số 0 khi không tác động vào ngã vào nào.



(H 4.2)

### Mã hóa 8 đường sang 3 đường

Chúng ta sẽ khảo sát một IC mã hóa 8 đường sang 3 đường.

Trên thực tế khi chế tạo một IC, ngoài các ngã vào/ra để thực hiện chức năng chính của nó, người ta thường dự trù thêm các ngã vào và ra cho một số chức năng khác như cho phép, nối mạch để mở rộng hoạt động của IC.

IC 74148 là IC mã hóa ưu tiên 8 đường sang 3 đường, vào/ ra tác động thấp, có các ngã nối mạch để mở rộng mã hóa với số ngã vào nhiều hơn.

Dưới đây là bảng sự thật của IC 74148, trong đó  $E_i$  ngã vào nối mạch và cho phép,  $E_o$  là ngã ra nối mạch và  $G_s$  dùng để mở rộng cho số nhị phân ra.

Dựa vào bảng sự thật, ta thấy IC làm việc theo 10 trạng thái:

- Các trạng thái từ 0 đến 7: IC mã hóa cho ra số 3 bit
- Các trạng thái 8 và 9: dùng cho việc mở rộng, sẽ giải thích rõ hơn khi nối 2 IC để mở rộng mã hóa cho số 4 bit

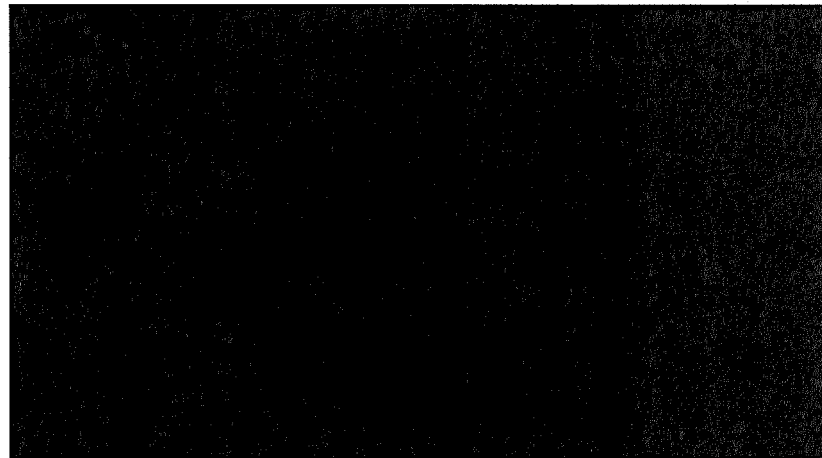
Trạng thái	Ngã vào								Ngã ra				
	$E_i$	0	1	2	3	4	5	6	$A_2$	$A_1$	$A_0$	$G_s$	$E_o$
9	1	x	x	x	x	x	x	x	1	1	1	1	1
8	0	x							1	1	1	1	0
7	0	1	1	1	1	1	1	1	0	0	0	0	1
6	0	1							0	0	1	0	1
5	0	x	x	x	x	x	x	x	0	1	0	0	1
4	0	0							0	1	1	0	1

---

3	0	x	x	x	x	x	x	0	1	0	0	0	1
2	0	1							1	0	1	0	1
1	0	x	x	x	x	x	0	1	1	1	0	0	1
0	0	1							1	1	1	0	1
		x	x	x	x	0	1	1					
		1											
		x	x	x	0	1	1	1					
		1											
		x	x	0	1	1	1	1					
		1											
		x	0	1	1	1	1	1					
		1											
		0	1	1	1	1	1	1					
		1											

Bảng 4.2

(H 4.3) là cách nối 2 IC để thực hiện mã hóa 16 đường sang 4 đường



(H 4.3)

- IC2 có  $E_i = 0$  nên hoạt động theo các trạng thái từ 0 đến 8, nghĩa là mã hóa từ 0 đến 7 cho các ngõ ra  $A_2A_1A_0$ .

- IC1 có  $E_i$  nối với  $E_o$  của IC2 nên IC1 chỉ hoạt động khi tất cả ngõ vào dữ liệu của IC2 lên mức 1 (IC2 hoạt động ở trạng thái 8)

\* Để mã hóa các số từ 0 đến 7, cho các ngõ vào 8 đến 15 (tức các ngõ vào dữ liệu của IC2) lên mức 1, IC2 hoạt động ở trạng thái 8.

Lúc đó  $E_{i1} = E_{o2} = 0$ : kết quả là IC1 sẽ hoạt động ở trạng thái từ 0 đến 7, cho phép tạo mã các số từ 0 đến 7 (từ 111 đến 000) và IC2 hoạt động ở trạng thái 8 nên các ngõ ra  $(A_2A_1A_0)_2 = 111$ , đây là điều kiện mở các cổng AND để cho mã số ra là  $B_2B_1B_0 = A_2A_1A_0$  của IC1, trong lúc đó  $B_3 = G_{s2} = 1$ , ta được kết quả từ 1111 đến 1000, tức từ 0 đến 7 (tác động thấp).

Thí dụ để mã số 4, đưa ngõ vào 4 xuống mức 0, các ngõ vào từ 5 đến 15 lên mức 1, bất chấp các ngõ vào từ 0 đến 3, mã số ra là  $B_3B_2B_1B_0 = G_{s2}B_2B_1B_0 = 1011$ , tức số 4

\* Để mã hóa các số từ 8 đến 15, cho IC2 hoạt động ở trạng thái từ 0 đến 7 (đưa ngõ vào ứng với số muốn mã xuống thấp, các ngõ vào cao hơn lên mức 1 và các ngõ vào thấp hơn xuống mức 0), bất chấp các ngõ vào dữ liệu của IC1 (cho IC1 hoạt động ở trạng thái 9), nên các ngõ ra  $(A_2A_1A_0)_1 = 111$ , đây là điều kiện mở các cổng AND để cho mã số ra là  $B_2B_1B_0 =$

$A_2A_1A_0$  của IC2, , trong lúc đó  $B_3 = G_{s2} = 0$ , ta được kết quả từ 0111 đến 0000, tức từ 8 đến 15.

Thí dụ để mã số 14, đưa ngã vào 14 xuống mức 0, đưa ngã vào 15 lên mức 1, bắt chập các ngã vào từ 0 đến 13, mã số ra là  $B_3B_2B_1B_0 = G_{s2}B_2B_1B_0 = 0001$ , tức số 14

Muốn có ngã ra chỉ số nhị phân đúng với ngã vào được tác động mà không phải đảo các bit ta có thể thay các cổng AND bằng cổng NAND

### Mạch tạo mã BCD cho số thập phân

Mạch gồm 10 ngã vào tương trưng cho 10 số thập phân và 4 ngã ra là 4 bit của số BCD. Khi một ngã vào (tương trưng cho một số thập phân) được tác động bằng cách đưa lên mức cao các ngã ra sẽ cho số BCD tương ứng

Bảng sự thật của mạch:

Trạng thái các ngã vào								Mã số ra		
9	8	7	6	5	4	3	2	$A_3$	$A_2$	$A_1$
1	0							$A_0$		
0	0	0	0	0	0	0	0	0	0	0
0	1							0		
0	0	0	0	0	0	0	0	0	0	0
1	0							1		
0	0	0	0	0	0	0	1	0	0	1
0	0							0		
0	0	0	0	0	0	1	0	0	0	1
0	0							1		
0	0	0	0	0	1	0	0	0	1	0
0	0							0		
0	0	0	0	1	0	0	0	0	1	0
0	0							1		
0	0	0	1	0	0	0	0	0	1	1
0	0							0		
0	0	1	0	0	0	0	0	0	1	1
0	0							1		
0	1	0	0	0	0	0	0	1	0	0
0	0							0		
1	0	0	0	0	0	0	0	1	0	0
0	0							1		

Bảng 4.3

Không cần bảng Karnaugh ta có thể viết ngay các hàm xác định các ngã ra:

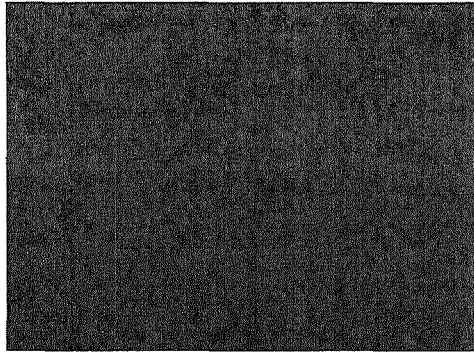
$$A_0 = 1 + 3 + 5 + 7 + 9$$

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

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

$$A_3 = 8 + 9$$

Mạch cho ở (H 4.4)



(H 4.4)

Để tạo mã BCD ưu tiên cho số lớn, ta viết lại bảng sự thật và dùng phương pháp đại số để đơn giản các hàm xác định các ngõ ra  $A_3, A_2, A_1, A_0$

Trạng thái các ngõ vào								Mã số ra		
9	8	7	6	5	4	3	2	$A_3$	$A_2$	$A_1$
1	0							$A_0$		
0	0	0	0	0	0	0	0	0	0	0
0	1							0		
0	0	0	0	0	0	0	0	0	0	0
1	x							1		
0	0	0	0	0	0	0	1	0	0	1
x	x							0		
0	0	0	0	0	0	1	x	0	0	1
x	x							1		
0	0	0	0	0	1	x	x	0	1	0
x	x							0		
0	0	0	0	1	x	x	x	0	1	0
x	x							1		
0	0	0	1	x	x	x	x	0	1	1
x	x							0		
0	0	1	x	x	x	x	x	0	1	1
x	x							1		
0	1	x	x	x	x	x	x	1	0	0
x	x							0		
1	x	x	x	x	x	x	x	1	0	0
x	x							1		

Bảng 4.4

$$A_3 = \overline{9} \cdot 8 + 9 = 9 + 8$$

$$A_2 = \overline{7} \cdot \overline{8} \cdot 9 + 6 \cdot \overline{7} \cdot \overline{8} \cdot 9 + 5 \cdot \overline{6} \cdot \overline{7} \cdot \overline{8} \cdot 9 + 4 \cdot \overline{5} \cdot \overline{6} \cdot \overline{7} \cdot \overline{8} \cdot 9 = (7 + 6 \cdot \overline{7} + 5 \cdot \overline{6} \cdot \overline{7} + 4 \cdot \overline{5} \cdot \overline{6} \cdot \overline{7}) \cdot 8 \cdot 9$$

$$A_2 = (7 + 6 + 5 + 4) \cdot \overline{8} \cdot 9 = (7 + 6 + 5 + 4) \cdot (8 + 9)$$

$$A_1 = \overline{7} \cdot \overline{8} \cdot 9 + 6 \cdot \overline{7} \cdot \overline{8} \cdot 9 + 3 \cdot \overline{4} \cdot \overline{5} \cdot \overline{6} \cdot \overline{7} \cdot \overline{8} \cdot 9 + 2 \cdot \overline{3} \cdot \overline{4} \cdot \overline{5} \cdot \overline{6} \cdot \overline{7} \cdot \overline{8} \cdot 9 = (7 + 6 \cdot \overline{7} + 3 \cdot \overline{4} \cdot \overline{5} \cdot \overline{6} \cdot \overline{7} + 2 \cdot \overline{3} \cdot \overline{4} \cdot \overline{5} \cdot \overline{6} \cdot \overline{7}) \cdot 8 \cdot 9$$

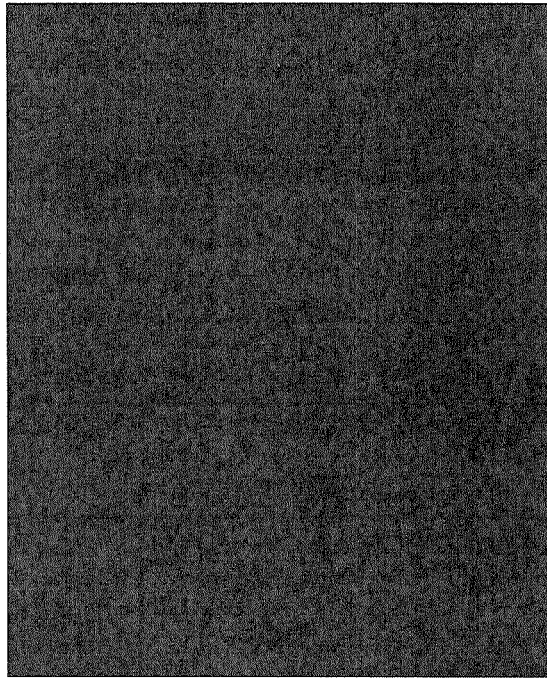
$$A_1 = (7 + 6 + 3 \cdot \overline{4} \cdot \overline{5} + 2 \cdot \overline{3} \cdot \overline{4} \cdot \overline{5}) \cdot 8 \cdot 9 = (7 + 6 + 3 \cdot \overline{4} \cdot \overline{5} + 2 \cdot \overline{4} \cdot \overline{5}) \cdot (8 + 9)$$

$$A_0 = 9 + \overline{7} \cdot \overline{8} \cdot 9 + 5 \cdot \overline{6} \cdot \overline{7} \cdot \overline{8} \cdot 9 + 3 \cdot \overline{4} \cdot \overline{5} \cdot \overline{6} \cdot \overline{7} \cdot \overline{8} \cdot 9 + 1 \cdot \overline{2} \cdot \overline{3} \cdot \overline{4} \cdot \overline{5} \cdot \overline{6} \cdot \overline{7} \cdot \overline{8} \cdot 9$$

$$= 9 + (7 + 5.6.7 + 3.4.5.6.7 + 1.2.3.4.5.6.7)8.9$$

$$A_0 = 9 + (7 + 5.6 + 3.4.6 + 1.2.4.6)8.9 = 9 + (7 + 5.6 + 3.4.6 + 1.2.4.6)(8 + 9)$$

Mạch cho ở (H 4.5)



(H 4.5)

### *Mạch chuyển mã*

Mạch chuyển từ một mã này sang một mã khác cũng thuộc loại mã hóa.

☛ Mạch chuyển mã nhị phân sang Gray

Thử thiết kế mạch chuyển từ mã nhị phân sang mã Gray của số 4 bit.

Trước tiên viết bảng sự thật của số nhị phân và số Gray tương ứng. Các số nhị phân là các biến và các số Gray sẽ là hàm của các biến đó.

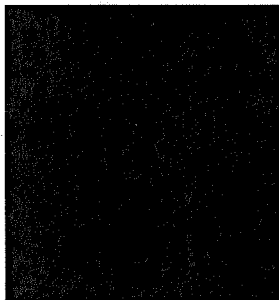
A	B	C	D		→	X	Y	Z	T
0	0	0	0	→	0	0	0	0	
0	0	0	1	→	0	0	0	1	
0	0	1	0	→	0	0	1	1	
0	0	1	1	→	0	0	1	0	
0	1	0	0	→	0	1	1	0	
0	1	0	1	→	0	1	1	1	
0	1	1	0	→	0	1	0	1	
0	1	1	1	→	0	1	0	0	
1	0	0	0	→	1	1	0	0	
1	0	0	1	→	1	1	0	1	
1	0	1	0	→	1	1	1	1	
1	0	1	1	→	1	1	1	0	
1	1	0	0	→	1	0	1	0	
1	1	0	1	→	1	0	1	1	
1	1	1	0	→	1	0	0	1	
1	1	1	1	→	1	0	0	0	

Bảng 4.5

Dùng bảng Karnaugh để xác định X, Y, Z, T theo A, B, C, D

Quan sát bảng sự thật ta thấy ngay:  $X = A$ ,

Vậy chỉ cần lập 3 bảng Karnaugh cho các biến Y, Z, T (H 4.6 a,b,c) và kết quả cho ở (H 4.6 d)



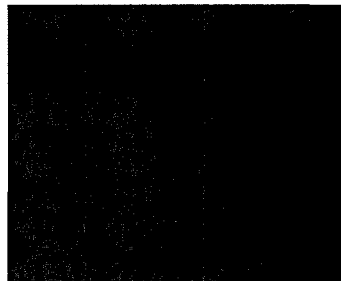
(a)



(b)



(c)



(H 4.6) (d)

## MẠCH GIẢI MÃ

Giải mã  $n$  đường sang  $2^n$  đường

Giải mã 2 đường sang 4 đường:

Thiết kế mạch Giải mã 2 đường sang 4 đường có ngõ vào cho phép (cũng được dùng để nối mạch)

Để đơn giản, ta xét mạch giải mã 2 đường sang 4 đường có các ngõ vào và ra đều tác động cao.

Bảng sự thật, các hàm ngõ ra và sơ đồ mạch:

G	Vào		R a			
	A <sub>1</sub>	A <sub>0</sub>	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

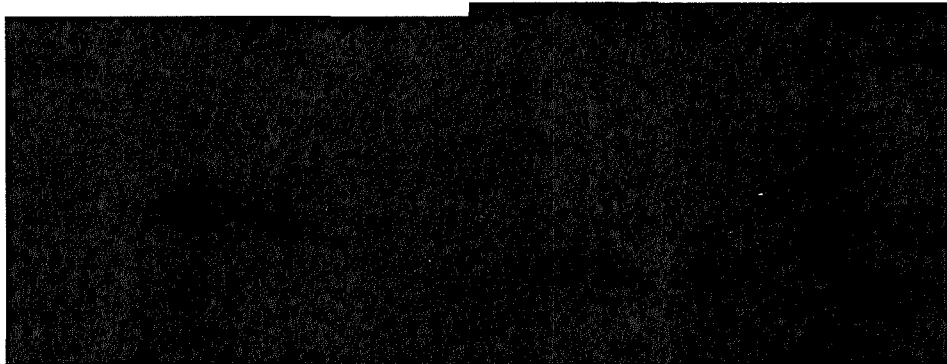
$$Y_0 = G \cdot \bar{A}_1 \bar{A}_0$$

$$Y_1 = G \cdot \bar{A}_1 A_0$$

$$Y_2 = G \cdot A_1 \bar{A}_0$$

$$Y_3 = G \cdot A_1 A_0$$

00



(H 4.7)

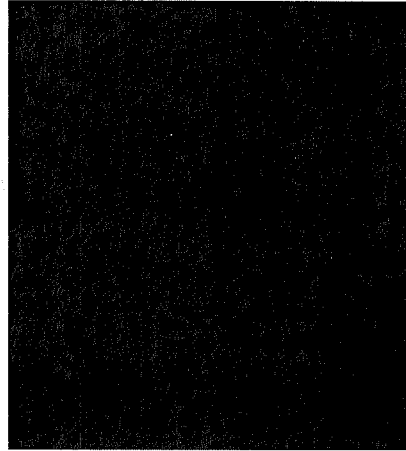
Giải mã 3 đường sang 8 đường

Dùng 2 mạch giải mã 2 đường sang 4 đường để thực hiện mạch giải mã 3 đường sang 8 đường (H 4.8)

Vào			R a							
A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	Y <sub>0</sub>	Y <sub>1</sub>	Y <sub>2</sub>	Y <sub>3</sub>	Y <sub>4</sub>	Y <sub>5</sub>	Y <sub>6</sub>	Y <sub>7</sub>
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1



Quan sát bảng sự thật ta thấy: Trong các tổ hợp số 3 bit có 2 nhóm trong đó các bit thấp  $A_1A_0$  hoàn toàn giống nhau, một nhóm có bit  $A_2 = 0$  và nhóm kia có  $A_2 = 1$ . Như vậy ta có thể dùng ngõ vào G cho bit  $A_2$  và mắc mạch như sau.



(H 4.8)

Khi  $A_2=G=0$ , IC1 giải mã cho 1 trong 4 ngõ ra thấp và khi  $A_2=G=1$ , IC2 giải mã cho 1 trong 4 ngõ ra cao

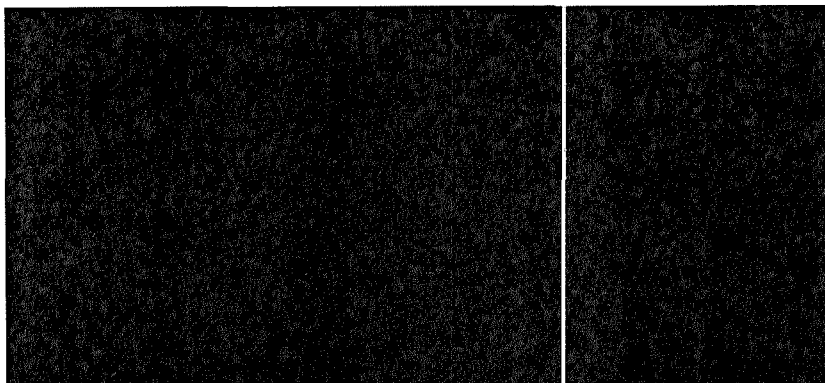
Trên thị trường hiện có các loại IC giải mã như:

- 74139 là IC chứa 2 mạch giải mã 2 đường sang 4 đường, có ngõ vào tác động cao, các ngõ ra tác động thấp, ngõ vào cho phép tác động thấp.
- 74138 là IC giải mã 3 đường sang 8 đường có ngõ vào tác động cao, các ngõ ra tác động thấp, hai ngõ vào cho phép  $G_{2A}$  và  $G_{2B}$  tác động thấp,  $G_1$  tác động cao.
- 74154 là IC giải mã 4 đường sang 16 đường có ngõ vào tác động cao, các ngõ ra tác động thấp, 2 ngõ vào cho phép  $E_1$  và  $E_2$  tác động thấp

Dưới đây là bảng sự thật của IC 74138 và cách nối 2 IC để mở rộng mạch giải mã lên 4 đường sang 16 đường (H 4.9)

Vào					Ra							
Ch phép		Dữ liệu										
$G_1$	$G_2$	C	B	A	$Y_0$	$Y_1$	$Y_2$	$Y_3$	$Y_4$	$Y_5$	$Y_6$	$Y_7$
x	H	x	x	x	H	H	H	H	H	H	H	H
L	x	x	x	x	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

Ghi chú  $G_2 = G_{2A} + G_{2B}$ , H = 1, L = 0, x: bất chấp



(H 4.9)

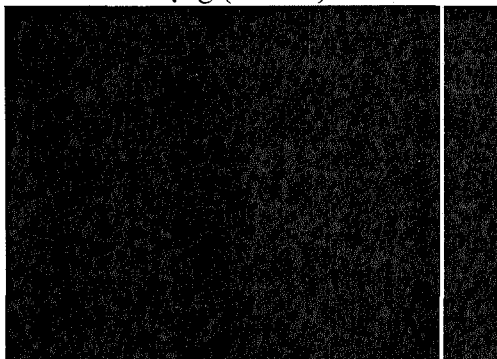
Một ứng dụng quan trọng của mạch giải mã là dùng giải mã địa chỉ cho bộ nhớ bán dẫn

Ngoài ra, mạch giải mã kết hợp với một cổng OR có thể tạo được hàm logic.

Thí dụ, thiết kế mạch tạo hàm  $Y=f(A,B,C)=\overline{A}BC + A\overline{B}C + \overline{A}\overline{B}C + ABC$

Với hàm 3 biến, ta dùng mạch giải mã 3 đường sang 8 đường. 8 ngõ ra mạch giải mã tương ứng với 8 tổ hợp biến của 3 biến, các ngõ ra tương ứng với các tổ hợp biến có trong hàm sẽ lên mức 1. Với một hàm đã viết dưới dạng tổng chuẩn, ta chỉ cần dùng một cổng OR có số ngõ vào bằng với số tổ hợp biến trong hàm nối vào các ngõ ra tương ứng của mạch giải mã để cộng các tổ hợp biến có trong hàm lại ta sẽ được hàm cần tạo.

Như vậy, mạch tạo hàm trên có dạng (H 4.10)



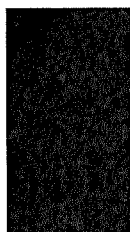
(H 4.10)

Dĩ nhiên, với những hàm chưa phải dạng tổng chuẩn, chúng ta phải chuẩn hóa. Và nếu bài toán có yêu cầu ta phải thực hiện việc đổi cổng, bằng cách dùng định lý De Morgan.

### Giải mã BCD sang 7 đoạn

#### Đèn 7 đoạn

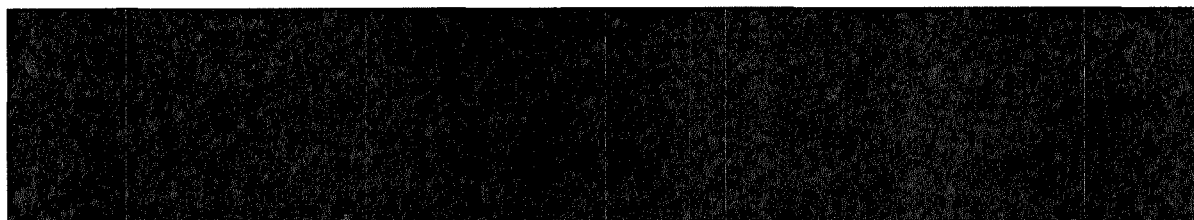
Đây là loại đèn dùng hiển thị các số từ 0 đến 9, đèn gồm 7 đoạn a, b, c, d, e, f, g, bên dưới mỗi đoạn là một led (đèn nhỏ) hoặc một nhóm led mắc song song (đèn lớn). Qui ước các đoạn cho bởi (H 4.11).



(H 4.11)

Khi một tổ hợp các đoạn cháy sáng sẽ tạo được một con số thập phân từ 0 - 9.

(H 4.12) cho thấy các đoạn nào cháy để thể hiện các số từ 0 đến 9



0 1 2 3 4 5 6 7 8 9

(H 4.12)

Đèn 7 đoạn cũng hiển thị được một số chữ cái và một số ký hiệu đặc biệt.

Có hai loại đèn 7 đoạn:

- Loại catod chung (H 4.13a), dùng cho mạch giải mã có ngõ ra tác động cao.
- Loại anod chung (H 4.13b), dùng cho mạch giải mã có ngõ ra tác động thấp.



(a)

(H 4.13)

(b)

### Mạch giải mã BCD sang 7 đoạn :

Mạch có 4 ngõ vào cho số BCD và 7 ngõ ra thích ứng với các ngõ vào a, b, c, d, e, f, g của led 7 đoạn, sao cho các đoạn cháy sáng tạo được số thập phân đúng với mã BCD ở ngõ vào.

Bảng sự thật của mạch giải mã 7 đoạn, có ngõ ra tác động thấp:

Số TP	Ngõ vào				Ngõ ra						
	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0

Bảng 4.6

Dùng Bảng Karnaugh hoặc có thể đơn giản trực tiếp với các hàm chứa ít tổ hợp, ta có kết quả:

$$a = \overline{DB}(C\overline{A} + \overline{CA})$$

$$b = C\overline{BA} + C\overline{BA}$$

$$c = \overline{DCBA}$$

$$d = \overline{DCBA} + C\overline{BA} + CBA$$

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

$$f = \overline{CB} + BA + \overline{DCA}$$

$$g = \overline{DCB} + CBA$$

Từ các kết quả ta có thể vẽ mạch giải mã 7 đoạn dùng các cổng logic.

Hai IC thông dụng dùng để giải mã BCD sang 7 đoạn là:

- CD 4511 (loại CMOS, ngã ra tác động cao và có đệm)
- 7447 (loại TTL, ngã ra tác động thấp, cực thu để hở)

Chúng ta khảo sát một IC giải mã BCD sang 7 đoạn : IC 7447

Bảng sự thật của 7447:

Số / Hàm	Vào							Ra						
	LT	RB I	D	C	B	A	BI (1)	a	b	c	d	e	f	g
							RBO							
0	1	1	0	0	0	0	1	0	0	0	0	0	0	1
1	1	x	0	0	0	1	1	1	0	0	1	1	1	1
2	1	x	0	0	1	0	1	0	0	1	0	0	1	0
3	1	x	0	0	1	1	1	0	0	0	0	1	1	0
4	1	x	0	1	0	0	1	1	0	0	1	1	0	0
5	1	x	0	1	0	1	1	0	1	0	0	1	0	0
6	1	x	0	1	1	0	1	1	1	0	0	0	0	0
7	1	x	0	1	1	1	1	0	0	0	1	1	1	1
8	1	x	1	0	0	0	1	0	0	0	0	0	0	0
9	1	x	1	0	0	1	1	0	0	0	1	1	0	0
10	1	x	1	0	1	0	1	1	1	1	0	0	1	0
11	1	x	1	0	1	1	1	1	1	0	0	1	1	0
12	1	x	1	1	0	0	1	1	0	1	1	1	0	0
13	1	x	1	1	0	1	1	0	1	1	0	1	0	0
14	1	x	1	1	1	0	1	1	1	1	0	0	0	0
15	1	x	1	1	1	1	1	1	1	1	1	1	1	1
(2)	x	x	x	x	x	x	0	1	1	1	1	1	1	1
(3)	1	0	0	0	0	0	0	1	1	1	1	1	1	1
(4)	0	x	x	x	x	x	1	0	0	0	0	0	0	0

Ghi chú:

1. BI/RBO được nối theo kiểu điểm AND bên trong IC và được dùng như ngã vào xóa (Blanking Input, BI) và/hoặc ngã ra xóa dọn sóng (Ripple Blanking Output, RBO). Ngã vào BI phải được để hở hay giữ ở mức cao khi cần thực hiện giải mã cho số ra. Ngã vào xóa dọn sóng (Ripple Blanking Input, RBI) phải để hở hay ở mức cao khi muốn đọc số 0.

2. Khi đưa ngõ vào BI xuống thấp, ngõ ra lên 1 (không tác động) bắt chấp các ngõ vào còn lại. Ta nói IC làm việc dưới điều kiện bị ép buộc và đây là trường hợp duy nhất BI giữ vai trò ngõ vào.

3. Khi ngõ vào RBI ở mức 0 và  $A=B=C=D=0$ , tất cả các ngõ ra kể cả RBO đều xuống 1. Ta nói IC làm việc dưới điều kiện đáp ứng.

4. Khi BI/RBO để hở hay được giữ ở mức 1 và ngõ vào thử đèn (Lamp test, LT) xuống 0, tất cả các led đều cháy (ngõ ra xuống 0).

Dựa vào bảng sự thật và các ghi chú 7447 là IC giải mã BCD sang 7 đoạn có đầy đủ các chức năng khác như : thử đèn, xóa số 0 khi nó không có nghĩa. Ta có thể hiểu rõ hơn chức năng này với thí dụ mạch hiển thị một kết quả có 3 chữ số sau đây: (H 4.14)



(H 4.14)

Vận hành của mạch có thể giải thích như sau:

- IC hàng đơn vị có ngõ vào RBI đưa lên mức cao nên đèn số 0 hàng đơn vị luôn luôn được hiển thị (dòng 0 trong bảng sự thật), điều này là cần thiết để xác nhận rằng mạch vẫn chạy và kết quả giải mã là số 0.

- IC hàng chục có ngõ vào RBI nối với ngõ ra RBO của IC hàng trăm nên số 0 hàng chục chỉ được hiển thị khi số hàng trăm khác 0 ( $RBO=1$ ) (dòng 0 đến 15).

- IC hàng trăm có ngõ vào RBI đưa xuống mức thấp nên số 0 hàng trăm luôn luôn tắt (dòng ghi chú 3).

### Hiển thị 7 đoạn bằng tinh thể lỏng (liquid crystal displays, LCD)

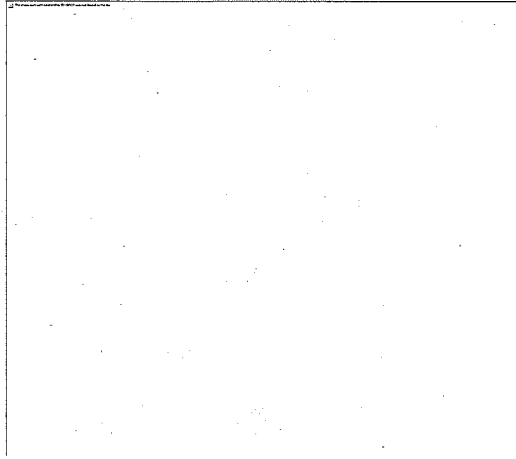
LCD gồm 7 đoạn như led thường và có chung một cực nền (backplane). Khi có tín hiệu xoay chiều biên độ khoảng 3 - 15  $V_{RMS}$  và tần số khoảng 25 - 60 Hz áp giữa một đoạn và cực nền, thì đoạn đó được tác động và sáng lên.

Trên thực tế người ta tạo hai tín hiệu nghịch pha giữa nền và một đoạn để tác động cho đoạn đó cháy.

Để hiểu được cách vận chuyển ta có thể dùng IC 4511 kết hợp với các cổng EX-OR để thúc LCD (H 4.15). Các ngõ ra của IC 4511 (Giải mã BCD sang 7 đoạn, tác động cao) nối vào các ngõ vào của các cổng EX-OR, ngõ vào còn lại nối với tín hiệu hình vuông tần số khoảng 40 Hz (tần số thấp có thể gây ra nhấp nháy), tín hiệu này đồng thời được đưa vào nền. Khi một ngõ ra mạch giải mã lên cao, ngõ ra cổng EX-OR cho một tín hiệu đảo pha với tín hiệu ở nền, đoạn tương ứng xem như nhận được tín hiệu có biên độ gấp đôi và sẽ sáng lên. Với các ngõ ra mạch giải mã ở mức thấp, ngõ ra cổng EX-OR cho một tín hiệu cùng pha với tín hiệu ở nền nên đoạn tương ứng không sáng.

Người ta thường dùng IC CMOS để thúc LCD vì hai lý do:

- CMOS tiêu thụ năng lượng rất thấp phù hợp với việc dùng pin cho các thiết bị dùng L
  - Mức thấp của CMOS đạt trị 0 và tín hiệu thúc LCD sẽ không chứa thành phần một C D.
- chiều, tuổi thọ LCD được kéo dài. (Mức thấp của TTL khoảng 0,4 V, thành phần DC này làm giảm tuổi thọ của LCD).



(H 4.15)

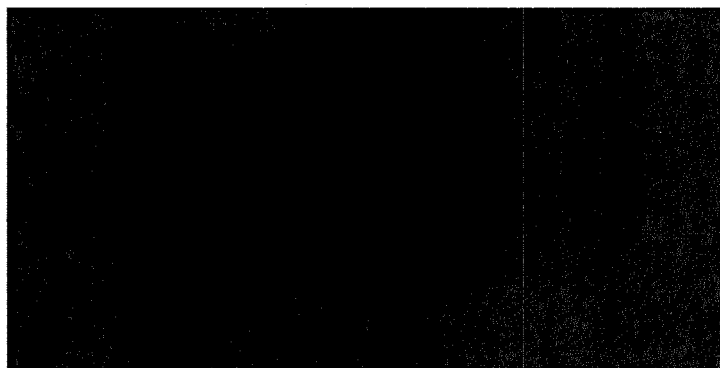
## MẠCH ĐA HỢP VÀ MẠCH GIẢI ĐA HỢP

### 4.3.1. Khái niệm

Trong truyền dữ liệu, để tiết kiệm đường truyền, người ta dùng một đường dây để truyền nhiều kênh dữ liệu, như vậy phải thực hiện việc chọn nguồn dữ liệu nào trong các nguồn khác nhau để truyền.

Mạch đa hợp hay còn gọi là mạch chọn dữ liệu sẽ làm công việc này.

Ở nơi thu, dữ liệu nhận được phải được chuyển tới các đích khác nhau, ta cần mạch phân bố dữ liệu hay giải đa hợp (H 4.16).



(H 4.16)

### 4.3.2 Mạch đa hợp

Còn được gọi là mạch chọn dữ liệu, gồm  $2^n$  ngõ vào dữ liệu,  $n$  ngõ vào địa chỉ (hay điều khiển) và một ngõ ra. Khi có một địa chỉ được tác động dữ liệu ở ngõ vào tương ứng với địa chỉ đó sẽ được chọn.

- Thiết kế mạch đa hợp 4→1

Mạch có 4 ngõ vào dữ liệu  $D_0 \dots D_3$ , 2 ngõ vào điều khiển AB và ngõ ra Y

Bảng sự thật:

A	B	Y
0	0	D <sub>0</sub>
0	1	D <sub>1</sub>
1	0	D <sub>2</sub>
1	1	D <sub>3</sub>

Từ bảng sự thật ta có hàm Y như sau:

$$Y = \overline{A} \cdot \overline{B} D_0 + \overline{A} B D_1 + A \overline{B} D_2 + A B D_3$$

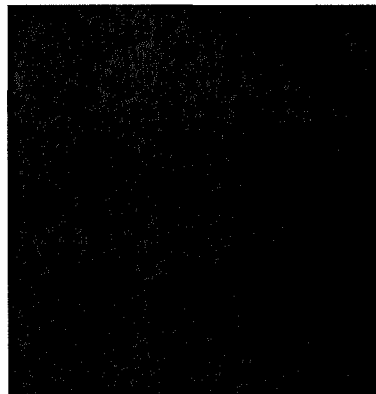
Và mạch có dạng (H 4.17)



(H 4.17)

Nếu chịu khó quan sát ta sẽ thấy mạch đa hợp 4→1 có thể được thiết kế từ mạch giải mã 2 đường sang 4 đường trong đó ngõ vào cho phép G đã được tách riêng ra để làm ngõ vào dữ liệu (D<sub>0</sub> . . . . D<sub>3</sub>) và ngõ vào dữ liệu của mạch giải mã đã trở thành ngõ vào điều khiển của mạch đa hợp (A, B)

(H 4.18) là ký hiệu một mạch đa hợp với 8 ngõ vào dữ liệu, 3 ngõ vào điều khiển và 1 ngõ ra, ta gọi là đa hợp 8 → 1.



(H 4.18)

Bảng sự thật:

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

Một đa hợp 8 → 1 có ngõ ra Y quan hệ với các ngõ vào dữ liệu và điều khiển theo hàm

$$Y = \overline{A} \cdot \overline{B} \cdot \overline{C} D_0 + \overline{A} \cdot \overline{B} \cdot C D_1 + \overline{A} B \overline{C} D_2 + \overline{A} B C D_3 + A \overline{B} \overline{C} D_4 + A \overline{B} C D_5 + A B \overline{C} D_6 + A B C D_7$$

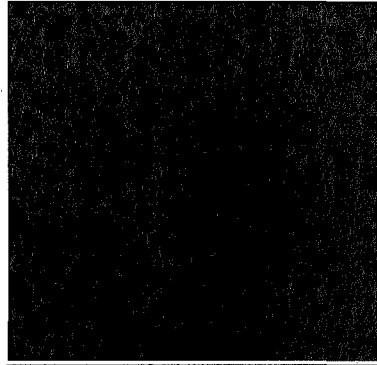
### Ứng dụng mạch đa hợp

Ngoài chức năng chọn dữ liệu mạch đa hợp còn được dùng để:



### Biến chuỗi dữ liệu song song thành nối tiếp:

Một mạch đa hợp kết hợp với một mạch đếm sẽ biến chuỗi dữ liệu song song ở ngõ vào thành chuỗi dữ liệu nối tiếp ở ngõ ra (H 4.19)



(H 4.19)

### Tạo chuỗi xung tuần hoàn :

Nếu cho dữ liệu vào tuần hoàn, dữ liệu ra nối tiếp cũng tuần hoàn, như vậy chỉ cần đặt trước các ngõ vào thay đổi theo một chu kỳ nào đó ta sẽ được chuỗi xung tuần hoàn ở ngõ ra.

### Tạo hàm:

⑦ Một đa hợp  $2^n \rightarrow 1$  có thể tạo hàm n biến bằng cách cho các biến vào ngõ vào điều khiển và cho trị riêng của hàm vào các ngõ vào dữ liệu.

Thí dụ: Để tạo hàm 3 biến bằng đa hợp  $8 \rightarrow 1$  ta viết lại biểu thức của đa hợp

$$Y = \overline{A}.\overline{B}.\overline{C}D_0 + \overline{A}.\overline{B}.C.D_1 + \overline{A}.B.\overline{C}D_2 + \overline{A}.B.C.D_3 + A.\overline{B}.\overline{C}D_4 + A.\overline{B}.C.D_5 + A.B.\overline{C}D_6 + A.B.C.D_7$$

So sánh với biểu thức của hàm viết dưới dạng triển khai theo định lý Shannon thứ nhất

$$f(A,B,C) = \overline{A}.\overline{B}.\overline{C}f(0,0,0) + \overline{A}.\overline{B}.Cf(0,0,1) + \overline{A}.B.\overline{C}f(0,1,0) + \overline{A}.B.Cf(0,1,1) + A.\overline{B}.\overline{C}f(1,0,0) + A.\overline{B}.Cf(1,0,1) + A.B.\overline{C}f(1,1,0) + A.B.Cf(1,1,1)$$

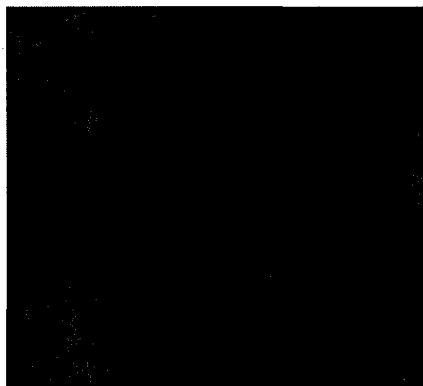
Ta được kết quả:

$$D_0 = f(0,0,0) ; D_1 = f(0,0,1) , \dots \dots \dots D_6 = f(1,1,0) \text{ và } D_7 = f(1,1,1)$$

Thí dụ: Tạo hàm:

$$Y = f(A, B, C) = \overline{A}.\overline{B}.\overline{C} + \overline{A}BC + \overline{A}B\overline{C} + A\overline{B}C + ABC$$

Ta thấy  $D_0=D_2=D_3=D_5=D_7=1$  nên các ngõ vào này được nối lên nguồn, các ngõ vào còn lại  $D_1=D_4=D_6=0$  nên được đưa xuống mass (H 4.20).



(H 4.20)

⑦ Một đa hợp  $2^n \rightarrow 1$  kết hợp với một cổng NOT có thể tạo hàm  $(n+1)$  biến. *Thí dụ* :  
 Tạo hàm  $F_1 = \overline{AB} + \overline{ABC} + \overline{BC} + \overline{AC}$  dùng đa hợp  $4 \rightarrow 1$  và cổng NOT  
 Giải

Đa hợp 4 sang 1 thực hiện  
 hàm:

$$Y = \overline{ABD}_0 + \overline{ABD}_1 + \overline{ABD}_2 + \overline{ABD}_3$$

Chuẩn hóa hàm  $F_1$  :

$$F_1 = \overline{ABC} + \overline{ABC} + \overline{ABC} + \overline{ABC} + \overline{ABC}$$

Để  $Y = F_1$  ta phải có:

$$D_0 = C; D_1 = \overline{C}; D_2 = 1; D_3 = C$$

(H 4.21)

Trên thực tế, ta có đủ các loại mạch đa hợp từ  $2 \rightarrow 1$  (IC 74157),  $4 \rightarrow 1$  (IC 74153),  $8 \rightarrow 1$  (IC 74151) và  $16 \rightarrow 1$  (74150) . . . .

Ngoài ra, để chọn dữ liệu là các nguồn tín hiệu tương tự, ta cũng có các đa hợp tương tự với tên gọi khóa tương tự (analog switch), được chế tạo theo công nghệ MOS như IC 4051 (8 kênh) IC 4053 (2 kênh). . . . Cũng có loại khóa sử dụng được cho cả tín hiệu tương tự và số (bilateral switches) như IC 4016, IC 4066, . . mà sinh viên có thể tìm hiểu, sử dụng dễ dàng khi có bảng tra kỹ thuật.

### Mạch giải đa hợp

Mạch giải đa hợp thực chất là mạch giải mã trong đó ngõ vào cho phép trở thành ngõ vào dữ liệu và ngõ vào của tổ hợp số nhị phân trở thành ngõ vào địa chỉ.

Trên thị trường, người ta chế tạo mạch giải mã và giải đa hợp chung trong một IC, tùy theo điều kiện mà sử dụng. Thí dụ IC 74138 là IC Giải mã 3 sang 8 đường đồng thời là mạch giải đa hợp  $1 \rightarrow 8$ .

Khi sử dụng IC 74138 làm mạch giải đa hợp, người ta dùng một ngõ vào cho phép làm ngõ vào dữ liệu và các ngõ vào số nhị phân làm ngõ vào địa chỉ. (H 4.22a) là IC 74138 dùng

giải đa hợp với ngõ vào dữ liệu  $\overline{G}_{2A}$ . (H 4.22b) là dạng dữ liệu vào  $\overline{G}_{2A}$  và ra ở  $\overline{Y}_0$  (vì

CBA=000), các ngõ ra khác ( $\overline{Y}_1$  ) ở mức cao.

-  $\overline{Y}_7$



(a)

(H 4.22)

(b)

## MẠCH SO SÁNH

### *Mạch so sánh 2 số 1 bit*

Trước tiên ta thiết kế mạch so sánh hai số 1 bit.

Bảng sự thật của mạch so sánh một bit có ngõ vào cho phép (nối mạch) G :

G	a	b	S (a>b)	I (a<b)	E (a=b)
0	x	x	0	0	0
1	0	0	0	0	1
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	0	0	1

Bảng 4.7



(H 4.23)

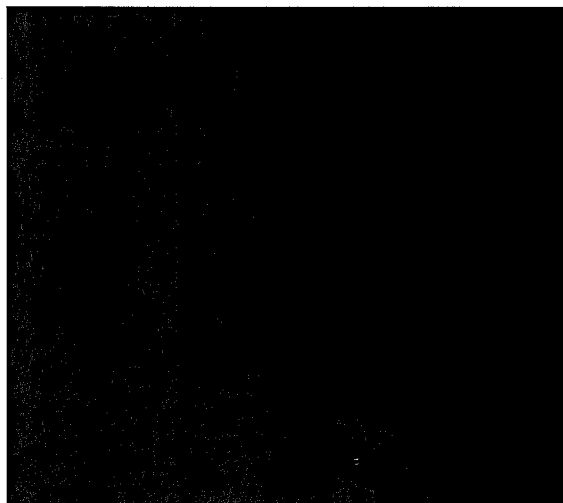
Từ mạch so sánh 1 bit ta có thể mở rộng để so sánh nhiều bit.

### *Mạch so sánh 2 số nhiều bit*

Để so sánh 2 số nhiều bit, trước tiên người ta so sánh 2 bit cao nhất (MSB), kết quả lớn hoặc nhỏ hơn do 2 bit này quyết định, nếu 2 bit MSB bằng nhau người ta so sánh 2 bit có trọng số thấp hơn tiếp theo và kết quả được quyết định theo cách tương tự như ở 2 bit MSB. .

... Sự so sánh được lặp lại cho đến bit LSB để được kết cuối cùng.

Dưới đây là sơ đồ mạch so sánh 3 bit (H 4.24).



(H 4.24)

- IC 1 so sánh 2 bit cao ( $a_3$  &  $b_3$ ) nên ngõ vào cho phép được đưa lên mức cao, nếu kết quả bằng nhau, ngõ ra E của nó lên cao, cho phép IC 2 so sánh, nếu kết quả lại bằng nhau, ngõ ra E của IC 2 lên cao cho phép IC 3 so sánh, kết quả bằng nhau cuối cùng chỉ bởi ngõ ra E của IC 3.

- Các ngõ vào cổng OR nhận tín hiệu từ các ngõ ra S (hoặc I) sẽ cho kết quả lớn hơn (hoặc nhỏ hơn) tùy vào kết quả so sánh ở bất cứ bit nào. Thật vậy khi có một kết quả lớn hơn (hoặc nhỏ hơn) thì S (hoặc I) ở một IC lên cao, các ngõ ra E và I (hoặc S) của các IC khác bằng 0, đây là điều kiện mở cổng OR để cho kết quả so sánh xuất hiện ở một trong các cổng OR này.

Trên thị trường có sẵn loại IC so sánh 4 bit 7485 có ngõ nối mạch để mở rộng việc so sánh cho số nhiều bit hơn.

Bảng sự thật của IC 7485

Trạng thái	Ngõ vào so sánh				Vào nối mạch			ra		
	$A_3, B_3$	$A_2, B_2$	$A_1, B_1$	$A_0, B_0$	$A' > B'$	$A' < B'$	$A' = B'$	$A > B$	$A < B$	$A = B$
1	$A_3 > B_3$	x	x	x	x	x	x	1	0	0
2	$A_3 <$	x	x	x	x	x	x	0	1	0
3	$B_3$	$A_2 > B_2$	x	x	x	x	x	1	0	0
4	$A_3 =$	$A_2 <$	x	x	x	x	x	0	1	0
5	$B_3$	$B_2$	$A_1 > B_1$	x	x	x	x	1	0	0
6	$A_3 =$	$A_2 =$	$A_1 <$	x	x	x	x	0	1	0
7	$B_3$	$B_2$	$B_1$	$A_0 > B_0$	x	x	x	1	0	0
8	$A_3 =$	$A_2 =$	$A_1 =$	$A_0 <$	x	x	x	0	1	0
9	$B_3$	$B_2$	$B_1$	$B_0$	0	0	1	0	0	1
10	$A_3 =$	$A_2 =$	$A_1 =$	$A_0 =$	1	0	0	1	0	0
11	$B_3$	$B_2$	$B_1$	$B_0$	0	1	0	0	1	0
	$A_3 =$	$A_2 =$	$A_1 =$	$A_0 =$						
	$B_3$	$B_2$	$B_1$	$B_0$						
	$A_3 =$	$A_2 =$	$A_1 =$	$A_0 =$						
	$B_3$	$B_2$	$B_1$	$B_0$						
	$A_3 =$	$A_2 =$	$A_1 =$	$A_0 =$						
	$B_3$	$B_2$	$B_1$	$B_0$						
	$A_3 =$	$A_2 =$	$A_1 =$	$A_0 =$						
	$B_3$	$B_2$	$B_1$	$B_0$						

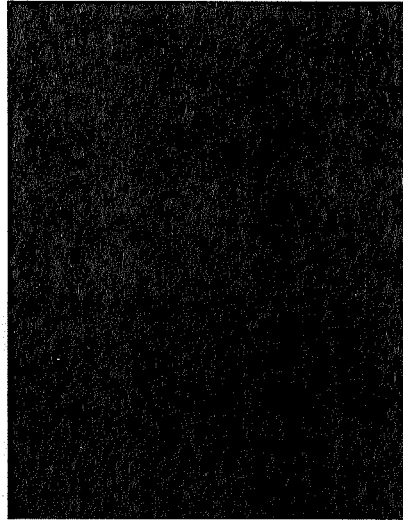
Bảng 4.8

Dựa vào bảng sự thật, ta thấy:

- Khi dùng IC 7485 để so sánh 2 số 4 bit ta phải giữ ngõ vào nối mạch  $A' = B'$  ở mức cao, hai ngõ vào nối mạch còn lại ở mức thấp, như vậy IC mới thể hiện được kết quả của trạng thái 9.

- Khi so sánh 2 số nhiều bit hơn ta phải dùng nhiều IC 7485 và nối ngõ ra của IC so sánh bit thấp vào ngõ vào nối mạch tương ứng của các IC so sánh các bit cao hơn và IC so sánh các bit thấp nhất có ngõ vào nối mạch được mắc như khi dùng riêng lẻ. Để đọc được kết quả so sánh ta phải quan tâm tới các trạng thái 9, 10 và 11 trong bảng sự thật.

(H 4.25) cho ta cách mắc 2 IC 7485 để so sánh 2 số nhị phân 8 bit:



(H 4.25)

Thí dụ :

a. So sánh hai số  $A_7 \dots A_0 = 10101111$  và  $B_7 \dots B_0 = 10110001$

IC 2 so sánh các bit cao  $A_7 \dots A_4 = 1010$  và  $B_7 \dots B_4 = 1011$  có  $A_7 = B_7$ ,  $A_6 = B_6$ ,  $A_5 = B_5$  và  $A_4 < B_4$  cho ngã ra  $A < B = 1$  bất chấp trạng thái của các ngã vào nối mạch (trạng thái 8). Điều này có nghĩa là khi IC so sánh bit cao thấy có kết quả khác nhau giữa 2 số bit cao thì không quan tâm tới kết quả của bit thấp.

b. So sánh hai số  $A_7 \dots A_0 = 10101111$  và  $B_7 \dots B_0 = 10101001$

Trong trường hợp này kết quả hai số bit cao bằng nhau nên IC 2 nhìn vào ngã vào nối mạch để xem kết quả so sánh của IC1 (so sánh bit thấp),  $A_3A_2A_1A_0 = 1111 > B_3B_2B_1B_0 = 1001$  nên ngã ra  $A > B = 1$  để chỉ kết quả so sánh của 2 số 8 bit (trạng thái 10).

## MẠCH KIỂM / PHÁT CHẴN LẼ

Do yêu cầu kiểm sai trong truyền dữ liệu, người ta có phương pháp kiểm tra chẵn lẻ. Trong phương pháp này, ngoài các bit dữ liệu, người ta thêm vào 1 bit kiểm tra sao cho tổng số bit 1 kể cả bit kiểm tra là số chẵn (KT chẵn) hoặc lẻ (KT lẻ)

1	0	1	1	0	0	1	1
---	---	---	---	---	---	---	---

 ← Bit chẵn lẻ thêm vào (KT lẻ)

1	1	0	0	1	0	1	0
---	---	---	---	---	---	---	---

 ← Bit chẵn lẻ thêm vào (KT chẵn)

Ở nơi thu, mạch kiểm tra chẵn lẻ sẽ kiểm tra lại số số 1 có trên tất cả các bit để biết dòng dữ liệu nhận được đúng hay sai.

Với phương pháp này máy thu sẽ có kết luận đúng khi số bit lỗi là số lẻ. Như vậy phương pháp chỉ cho kết quả đúng với xác suất 50%, tuy nhiên vì xác suất để một lỗi xảy ra là rất nhỏ nên phương pháp vẫn được sử dụng phổ biến trong một số hệ truyền thông.

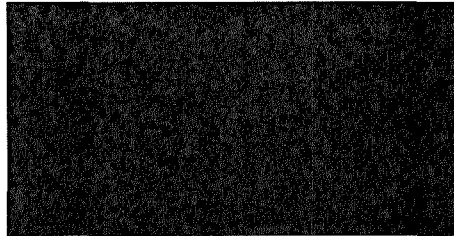
### Mạch phát chẵn lẻ (Parity Generator, PG)

Ta sẽ xét trường hợp mạch có 4 bit dữ liệu.

Mạch có 4 ngã vào dữ liệu A, B, C, D và 1 ngã vào chọn chẵn lẻ S

- Giai đoạn 1: Thiết kế mạch ghi nhận số bit 1 là chẵn hay lẻ  
 Giả sử ta muốn có mạch báo kết quả  $Y=1$  khi số bit 1 là lẻ và  $Y=0$  khi ngược lại.  
 Lợi dụng tính chất của cổng EX-OR có ngõ ra =1 khi số số 1 ở ngõ vào là lẻ. Với 4 ngõ vào, ta dùng 3 cổng EX-OR để thực hiện mạch ghi nhận này:

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



(H 4.26)

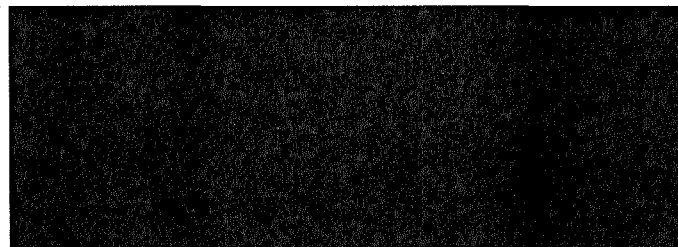
- Giai đoạn 2: Thiết kế phần mạch tạo bit chẵn lẻ P theo sự điều khiển của ngõ vào S  
 Giả sử ta muốn có  
 Tổng số bit 1 của A, B, C, D, P là lẻ khi  $S = 1$  và chẵn khi  $S = 0$

S	Số bit 1 của ABCD	Y	P
0	Lẻ	1	1
0	Chẵn	0	0
1	Lẻ	1	0
1	Chẵn	0	1

Bảng 4.9

Bảng 4.9 cho kết quả:  
 Vậy mạch có dạng

$$P = S \oplus Y$$



(H 4.27)

### Mạch kiểm chẵn lẻ (Parity checker, PC)

Nếu ta xem mạch phát ở (H 4.27) như là mạch có 5 ngõ vào thì ngõ ra P quan hệ với số lượng bit 1 ở các ngõ vào đó có thể được suy ra từ bảng 4.9



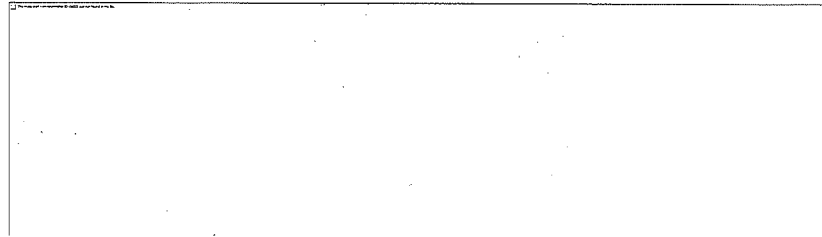
Số bit 1 của ABCDS	P
Lẻ	1
Chẵn	0

Bảng 4.9

Như vậy, ta có thể dùng mạch phát ở trên để làm mạch kiểm tra chẵn lẻ.

Tóm lại, một hệ thống gồm mạch phát và kiểm chẵn lẻ được mắc như (H 4.28)

Khi ngõ vào S của mạch phát đưa xuống mức 0, nếu bản tin nhận đúng thì ngõ ra P ở mạch kiểm cũng xuống 0.



(H 4.28)

Trên thị trường có các IC kiểm/phát chẵn lẻ như 74180 (9 bit) 74280 (9 bit), loại CMOS có 40101 (9 bit), 4531 (13 bit).

## BÀI TẬP

1. Thiết kế mạch mã hóa 32 đường sang 5 đường dùng IC 74148 và cổng logic.
2. Thiết kế mạch giải mã 4 đường sang 16 đường từ mạch giải mã 2 đường sang 4 đường có ngõ vào cho phép.
3. Thiết kế mạch so sánh 4 bit từ mạch so sánh 1 bit
4. Thiết kế mạch chuyển từ mã Gray sang mã nhị phân.
5. Thiết kế mạch chuyển từ mã BCD sang mã Excess-3 của các số từ 0 đến 9.  
(Mã Excess-3 của 1 số có được từ trị nhị phân tương ứng cộng thêm 3, thí dụ mã số 0 là 0011, mã số 9 là 1100)
6. Dùng một mạch giải mã 3 sang 8 đường, 2 cổng NAND 3 ngõ vào và 1 cổng AND 2 ngõ vào thực hiện các hàm sau:

$$F_1 = \Sigma(1,2,3) ; F_2 = \Sigma(4,5,7) ; F_3 = \Sigma(1,2,3,4,5,7)$$

7. Cài đặt các hàm sau dùng bộ dồn kênh (multiplexer)  $4 \rightarrow 1$  (Dùng thêm cổng logic nếu cần)

$$F_1 = \overline{A}B + \overline{A}BC + \overline{B}C + AC$$

$$F_2 = A \oplus (\overline{BC})$$

$$F_3 = \prod(1,3,6)$$

8. Thiết kế mạch MUX  $4 \rightarrow 1$  từ các MUX  $2 \rightarrow 1$

9. Dùng 2 MUX  $2 \rightarrow 1$  để thực hiện 1 MUX  $3 \rightarrow 1$  như sau:

AB = 00 chọn C

AB = 01 chọn D

AB = 1X chọn E (Trường hợp này B không xác định).

10. Thực hiện hàm  $Z = AB + BC + CA$

- Giải mã 3 sang 8 đường (dùng thêm cổng logic nếu cần).
- Đa hợp  $4 \rightarrow 1$  (dùng thêm cổng logic nếu cần).
- Hai mạch cộng bán phần và một cổng OR.

## CHƯƠNG 5      Họ vi mạch TTL - CMOS

### FLIP FLOP

Mạch flipflop (FF) là mạch dao động đa hài lưỡng ổn tức mạch tạo ra sóng vuông và có hai trạng thái ổn định. Trạng thái của FF chỉ thay đổi khi có xung đồng hồ tác động.

Một FF thường có:

- Một hoặc hai ngõ vào dữ liệu, một ngõ vào xung  $C_K$  và có thể có các ngõ vào với các chức năng khác.

- Hai ngõ ra, thường được ký hiệu là  $Q$  (ngõ ra chính) và  $\bar{Q}$  (ngõ ra phụ). Người ta thường dùng trạng thái của ngõ ra chính để chỉ trạng thái của FF. Nếu hai ngõ ra có trạng thái giống nhau ta nói FF ở **trạng thái cấm**.

Flipflop có thể được tạo nên từ mạch chốt (latch)

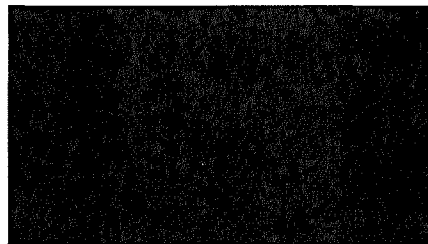
Điểm khác biệt giữa một mạch chốt và một FF là: FF chịu tác động của xung đồng hồ còn mạch chốt thì không.

Người ta gọi tên các FF khác nhau bằng cách dựa vào tên các ngõ vào dữ liệu của chúng.

#### Chốt RS

##### Chốt RS tác động mức cao:

(H 5.1) là chốt RS có các ngõ vào R và S tác động mức cao.



(H 5.1)

Các trạng thái logic của mạch cho ở bảng 5.1:

(Đối với mạch chốt vì không có tác động của xung đồng hồ nên ta có thể hiểu trạng thái trước là trạng thái giả sử, còn trạng thái sau là trạng thái khi mạch ổn định).

R	S	Q	$Q_+$
0	0	0	$0 \leftrightarrow$ Tác dụng nhớ
0	0	1	$1 \uparrow Q_+ = Q$
0	1	0	$1 \leftrightarrow$ Đặt (Set)
0	1	1	$1 \uparrow Q_+ = 1$
1	0	0	$0 \leftrightarrow$ Đặt lại (Reset)
1	0	1	$0 \uparrow Q_+ = 0$
1	1	0	$\leftrightarrow Q_+ = Q_+ = 0$
1	1	1	(Cấm)

R	S	$Q_+$
0	0	Q
0	1	1
1	0	0
1	1	Cấm

Bảng 5.1

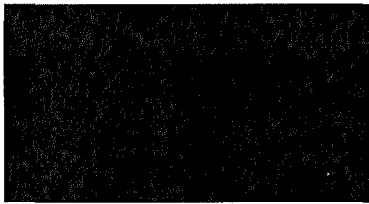
Bảng 5.2

Từ Bảng 5.1 thu gọn lại thành Bảng 5.2 và tính chất của chốt RS tác động mức cao được tóm tắt như sau:

- Khi  $R=S=0$  (cả 2 ngõ vào đều không tác động), ngõ ra không đổi trạng thái.
- Khi  $R=0$  và  $S=1$  (ngõ vào S tác động), chốt được Set (tức đặt  $Q_+=1$ ).
- Khi  $R=1$  và  $S=0$  (ngõ vào R tác động), chốt được Reset (tức đặt lại  $Q_+=0$ ).
- Khi  $R=S=1$  (cả 2 ngõ vào đều tác động), chốt rơi vào trạng thái cấm

**Chốt RS tác động mức thấp:**

(H 5.2) là chốt RS có các ngõ vào R và S tác động mức thấp. Các trạng thái logic cho bởi Bảng 5.3

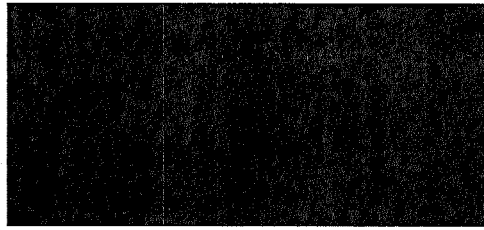


(H 5.2)

S	R	$Q_+$
0	0	Cấm
0	1	1
1	0	0
1	1	Q

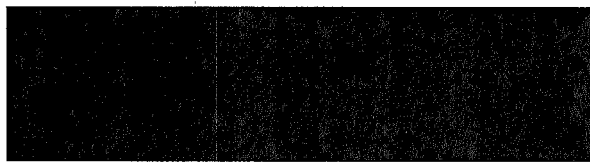
Bảng 5.3

Để có chốt RS tác động mức cao dùng cổng NAND, người ta thêm vào 2 cổng đảo ở các ngõ vào của mạch (H 5.2)



(H 5.3)

(H 5.4a) là ký hiệu chốt RS tác động cao và (H 5.4b) là chốt RS tác động thấp.



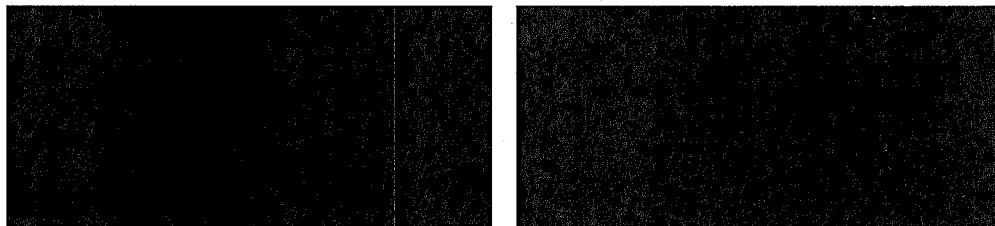
(a)

(b)

(H 5.4)

### Flip Flop RS

Trong các phần dưới đây, ta luôn sử dụng chốt RS tác động mức cao dùng cổng NAND. Khi thêm ngõ vào xung  $C_K$  cho chốt RS ta được FF RS. (H 5.5a) là FF RS có các ngõ vào R, S và xung đồng hồ  $C_K$  đều tác động mức cao.



(a)

(H 5.5)

(b)

Hoạt động của FF (H 5.5a) cho bởi Bảng sự thật: (Bảng 5.4)

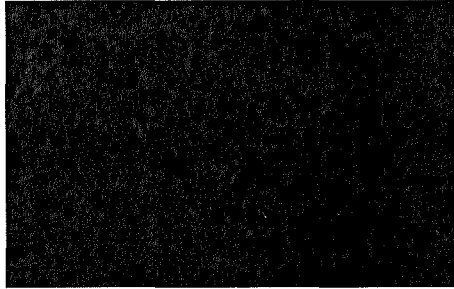
$C_K$	Vào		Ra
	S	R	$Q_+$
0	x	x	Q
1	0	0	Q
1	0	1	0
1	1	0	1
1	1	1	Cấm

Bảng 5.4

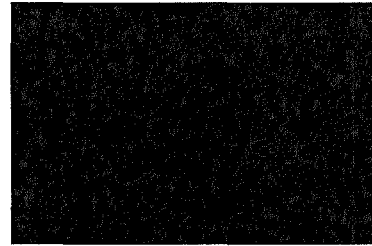
Để có FF RS có xung đồng hồ tác động thấp chỉ cần thêm một cổng đảo cho ngõ vào  $C_K$  (H 5.5b). Ta có bảng sự thật giống Bảng 5.4, trừ ngõ vào  $C_K$  phải đảo lại

### Flipflop RS có ngõ vào Preset và Clear:

Tính chất của FF là có trạng thái ngõ ra bất kỳ khi mở máy. Trong nhiều trường hợp, có thể cần đặt trước ngõ ra  $Q=1$  hoặc  $Q=0$ , muốn thế, người ta thêm vào FF các ngõ vào Preset (đặt trước  $Q=1$ ) và Clear (Xóa  $Q=0$ ), mạch có dạng (H 5.6a) và (H 5.6b) là ký hiệu của FF RS có ngõ vào Preset và Clear tác động mức thấp.



(a)



(b)

(H 5.6)

Thay 2 cổng NAND cuối bằng hai cổng NAND 3 ngõ vào, ta được FF RS có ngõ vào Preset (Pr) và Clear (Cl).

- Khi ngõ Pr xuống thấp (tác động) và ngõ Cl lên cao ngõ ra Q lên cao bất chấp các ngõ vào còn lại.

- Khi ngõ Cl xuống thấp (tác động) và ngõ Pr lên cao ngõ ra Q xuống thấp bất chấp các ngõ vào còn lại.

- Ngoài ra 2 ngõ vào Pr và Cl còn được đưa về 2 ngõ vào một cổng AND, nơi đưa tín hiệu  $C_K$  vào, mục đích của việc làm này là khi một trong 2 ngõ vào Pr hoặc Cl tác động thì mức thấp của tín hiệu này sẽ khóa cổng AND này, vô hiệu hóa tác dụng của xung  $C_K$ .

Bảng sự thật của FF RS có Preset và Clear (tác động thấp) cho ở bảng 5.5

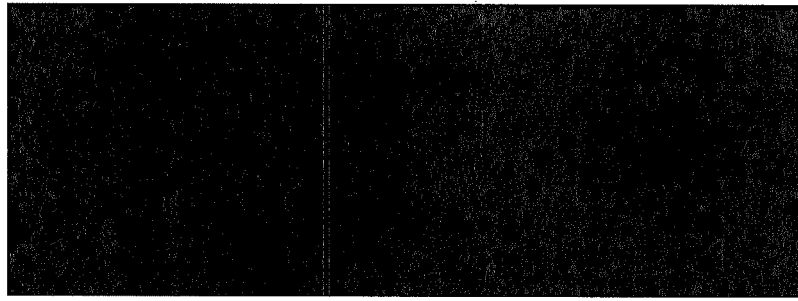
Pr	Cl	$C_K$	S	R	$Q_+$
0	0	x	x	x	Cấm
0	1	x	x	x	1
1	0	x	x	x	0
1	1	0	x	x	Q
1	1	1	0	0	Q
1	1	1	0	1	0
1	1	1	1	0	1
1	1	1	1	1	Cấm

Bảng 5.5

**Lưu ý:** Trên bảng 5.5, dòng thứ nhất tương ứng với trạng thái cấm vì hai ngõ vào Pr và Cl đồng thời ở mức tác động, 2 cổng NAND cuối cùng đều đóng, nên  $Q_+=Q=1$ .

### Flipflop RS chủ tớ:

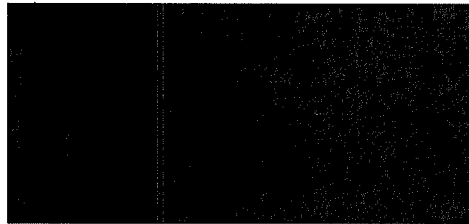
Kết nối thành chuỗi hai FF RS với hai ngõ vào xung  $C_K$  của hai FF có mức tác động trái ngược nhau, ta được FF chủ tớ (H 5.7).



(H 5.7)

Hoạt động của FF được giải thích như sau:

- Do  $C_{KS}$  của tầng tứ là đảo của  $C_{KM} = C_K$  của tầng chủ nên khi  $C_K=1$ , tầng chủ giao hoán thì tầng tứ ngưng. Trong khoảng thời gian này, dữ liệu từ ngõ vào R và S được đưa ra và ổn định ở ngõ ra R' và S' của tầng chủ, tại thời điểm xung  $C_K$  xuống thấp, R' và S' được truyền đến ngõ ra Q và  $\bar{Q}$  (H 5.8)



(H 5.8)

- Đối với trường hợp R = S = 1 khi  $C_K=1$  thì R' = S' = 1, nhưng khi  $C_K$  xuống thấp thì một trong hai ngõ ra này xuống thấp, do đó mạch thoát khỏi trạng thái cấm, nhưng S' hay R' xuống thấp trước thì không đoán trước được nên mạch rơi vào trạng thái bất định, nghĩa là Q+ có thể = 1 có thể = 0, nhưng khác với  $\bar{Q}_+$ . Ta có bảng sự thật:

S	R	$C_K$	Q+
0	0	↓	Q
0	1	↓	0
1	0	↓	1
1	1	↓	Bất định

Bảng 5.6

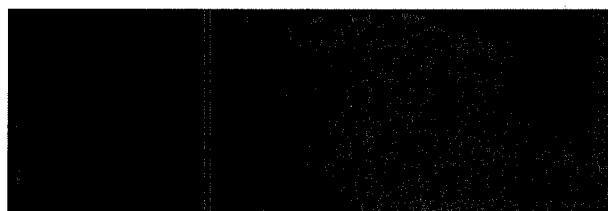
Tóm lại, FF RS chủ tứ đã thoát khỏi trạng thái cấm nhưng vẫn rơi vào trạng thái bất định, đồng thời ta được FF có ngõ vào xung đồng hồ tác động bởi cạnh xuống của tín hiệu  $C_K$ .

Để có FF RS có ngõ vào xung đồng hồ tác động bởi cạnh lên của tín hiệu  $C_K$  ta có thể dời cổng NOT đến ngõ vào FF chủ và cho tín hiệu  $C_K$  vào thẳng FF tứ.

Mặc dù thoát khỏi trạng thái cấm nhưng FF RS chủ tứ vẫn còn trạng thái bất định nên người ta ít sử dụng FF RS trong trường hợp R=S.

### Flipflop JK

FF JK được tạo ra từ FF RS theo sơ đồ như (H 5.9a).



(a)  
(H 5.9)

(b)

(H 5.9b) là ký hiệu FF JK có ngõ vào Pr và Cl tác động thấp.  
Bảng sự thật 5.7 (Để đơn giản, ta bỏ qua các ngõ vào Pr và Cl)

J	K	Q	$\bar{Q}$	S=J $\bar{Q}$	R=KQ	C <sub>K</sub>	Q <sub>+</sub>
0	0	0	1	0	0	↓	Q
0	0	1	0	0	0	↓	Q
0	1	0	1	0	0	↓	Q=0
0	1	1	0	0	1	↓	0
1	0	0	1	1	0	↓	1
1	0	1	0	0	0	↓	Q=1
1	1	0	1	1	0	↓	1
1	1	1	0	0	1	↓	0

J	K	C <sub>K</sub>	Q <sub>+</sub>
0	0	↓	Q
0	1	↓	0
1	0	↓	1
1	1	↓	Q

Bảng 5.8

Bảng 5.7

Bảng 5.8 là bảng rút gọn, suy ra từ bảng 5.7

Kết quả từ bảng 5.8 cho thấy:

FF JK đã thoát khỏi trạng thái cấm và thay vào đó là **trạng thái đảo** (khi J=K=1 thì Q<sub>+</sub>= $\bar{Q}$ ). Người ta lợi dụng trạng thái đảo này để thiết kế mạch đếm

### FlipFlop D

Thiết kế từ FF RS (hoặc JK) bằng cách nối một công đảo từ S qua R (hoặc từ J qua K). Dữ liệu được đưa vào ngõ S (J) mà bây giờ gọi là ngõ vào D (H 5.10a&b) và bảng 5.9 cho thấy các trạng thái của FF, cụ thể là mỗi khi có xung C<sub>K</sub> tác động dữ liệu từ ngõ vào sẽ xuất hiện ở ngõ ra.



(a)

(b)  
(H 5.10)

(c)

D	C <sub>K</sub>	Q <sub>+</sub>
0	↓	0
1	↓	1

Bảng 5.9

T	C <sub>K</sub>	Q <sub>+</sub>
0	↓	Q
1	↓	$\bar{Q}$

Bảng 5.10

### FlipFlop T

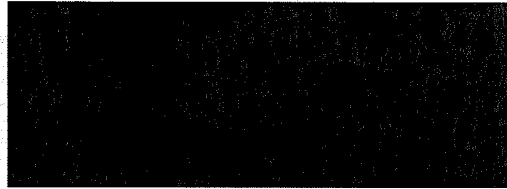
Nối chung hai ngõ vào J và K của FF JK ta được FF T (H 5.10c). Tính chất của FF T thể hiện trong bảng sự thật 5.10:

- Khi T=0, **FF không đổi trạng thái** dù có tác động của C<sub>K</sub>.
- Khi T=1, **FF đổi trạng thái** mỗi lần có xung C<sub>K</sub> tác động.



### Mạch chốt D

Mạch chốt D hoạt động giống FF D, chỉ khác ở điểm ngã vào xung đồng hồ  $C_K$  được thay bằng ngã vào cho phép G, và tác động bằng mức chứ không bằng cạnh (H 5.11) và Bảng 5.11.

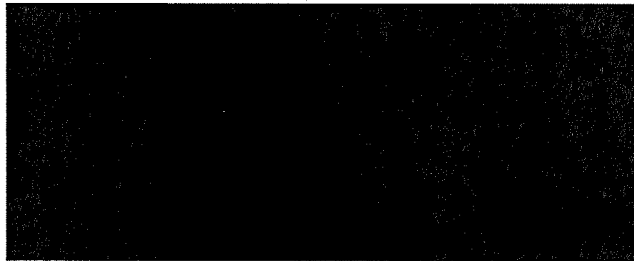


(H 5.11)

Bảng 5.11

### MẠCH GHI DỊCH

#### Sơ đồ nguyên tắc và vận chuyển (H 5.12)



(H 5.12)

(H 5.12) là sơ đồ một mạch ghi dịch 4 bit đơn giản, mạch gồm 4 FF D nối thành chuỗi (ngã ra Q của FF trước nối vào ngã vào D của FF sau) và các ngã vào  $C_K$  được nối chung lại (các FF chịu tác động đồng thời). Mạch ghi dịch này có khả năng dịch phải.

Ngã vào  $D_A$  của FF đầu tiên được gọi là **ngã vào dữ liệu nối tiếp**, các ngã ra  $Q_A, Q_B, Q_C, Q_D$  là các **ngã ra song song**, ngã ra của FF cuối cùng (FF D) là **ngã ra nối tiếp**.

Trước khi cho mạch hoạt động, tác dụng một xung xóa vào các ngã vào  $\overline{C_I}$  (đưa các chân  $\overline{C_I}$  đã được nối chung xuống thấp rồi lên cao) để các ngã ra  $Q_A = Q_B = Q_C = Q_D = 0$ .

Cho dữ liệu vào  $D_A$ , sau mỗi xung đồng hồ, dữ liệu từ tầng trước lần lượt truyền qua tầng sau. (Giả sử  $D_A$  là chuỗi dữ liệu gồm 3 bit cao, 2 bit thấp rồi 1 cao và 1 thấp), trạng thái các ngã ra của các FF cho ở Bảng 5.12

C <sub>I</sub>	Vào		Ra			
	C <sub>K</sub>	D <sub>A</sub>	Q <sub>A</sub>	Q <sub>B</sub>	Q <sub>C</sub>	Q <sub>D</sub>
0	x	x	0	0	0	0
1	↓	1	1	0	0	0
1	↓	1	1	1	0	0
1	↓	1	1	1	1	0
1	↓	0	0	1	1	1
1	↓	0	0	0	1	1
1	↓	1	1	0	0	1
1	↓	0	0	1	0	0
	↓					

Bảng 5.12

Các mạch ghi dịch được phân loại tùy vào số bit (số FF), chiều dịch (phải/trái), các ngã vào/ra (nối tiếp/song song).

Để có mạch dịch trái, dữ liệu nối tiếp đưa vào ngõ vào D của FF cuối cùng và các ngõ ra của FF sau nối ngược trở lại ngõ vào của FF trước (H 5.13)



(H 5.13)

Cho dữ liệu nối tiếp vào ngõ vào D của FF 4, sau mỗi xung đồng hồ, dữ liệu truyền từ tầng sau ra tầng trước. Giả sử chuỗi dữ liệu giống như trên, trạng thái các ngõ ra của các FF cho ở bảng 5.13

Cl	Vào		Ra			
	C <sub>K</sub>	D <sub>4</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub>
0	x	x	0	0	0	0
1	↓	1	0	0	0	1
1	↓	1	0	0	1	1
1	↓	1	0	1	1	1
1	↓	0	1	1	1	0
1	↓	0	1	1	0	0
1	↓	1	1	0	0	1
1	↓	0	0	0	1	0

Bảng 5.13

### Vài IC ghi dịch tiêu biểu

Trên thị trường hiện có khá nhiều loại IC ghi dịch, có đầy đủ các chức năng dịch phải trái, vào/ra nối tiếp, song song. Sau đây, chúng ta khảo sát 2 IC tiêu biểu:

- IC 74164: dịch phải 8 bit;
- IC 7495: 4 bit, dịch phải, trái, vào/ra nối tiếp/song song.

#### 5.2.2.1. IC 74164:



(H 5.14)

MR : Master Reset, đây cũng là chân Clear của cả mạch, tác động thấp  
 CP: Clock pulse, ngõ vào xung đồng hồ tác động cạnh lên.

### 5.2.2.2. IC 7495:



(H 5.15)

Ý nghĩa các chân:      S: Mode control input      D<sub>s</sub>: Serial Data input  
P<sub>0</sub> - P<sub>3</sub> : Parrallel data inputs  
CP<sub>1</sub> : Serial Clock                      CP<sub>2</sub> : Parrallel clock  
Q<sub>0</sub> - Q<sub>3</sub> : Parrallel outputs

Dưới đây là các bước thao tác để thực hiện các chức năng của IC

#### ◆ Nạp dữ liệu song song

- Chuẩn bị dữ liệu ở các ngõ vào P<sub>0</sub> - P<sub>3</sub>
- Cho S = 1, dữ liệu được đưa vào các ngõ vào của các FF, CP<sub>1</sub> bị khóa, CP<sub>2</sub> là ngõ vào C<sub>K</sub>, dữ liệu xuất hiện ở ngõ ra Q<sub>0</sub> - Q<sub>3</sub> khi có cạnh xuống của C<sub>K</sub>

#### ◆ Dịch phải

- Sau khi đã nạp dữ liệu song song - Chuẩn bị dữ liệu nối tiếp.
- Cho S = 0
- Đưa dữ liệu nối tiếp vào ngõ vào D<sub>s</sub>, CP<sub>2</sub> bị khóa, CP<sub>1</sub> là ngõ vào C<sub>K</sub>, khi C<sub>K</sub> tác động, dữ liệu sẽ dịch phải từng bit một trên các ngõ ra Q<sub>0</sub> - Q<sub>3</sub>

#### ◆ Dịch trái

- Nối ngõ ra FF sau vào ngõ vào song song của FF trước - P<sub>3</sub> là ngõ vào nối tiếp
- S = 1 để cách ly ngõ ra FF trước với ngõ vào FF sau
- CP<sub>2</sub> là ngõ vào xung C<sub>K</sub>, dữ liệu sẽ dịch trái ứng với cạnh xuống của C<sub>K</sub>.

*Lưu ý: Mặc dù có 2 ngõ vào cho xung C<sub>K</sub> nhưng khi sử dụng chúng thường được nối chung lại, lý do là vì ứng với một trạng thái của tín hiệu điều khiển S chỉ có một trong hai cổng AND mở để cho tín hiệu C<sub>K</sub> đi qua.*

#### Ứng dụng của ghi dịch

Ghi dịch có khá nhiều ứng dụng:

- Một số nhị phân khi dịch trái 1 bit, giá trị được nhân lên gấp đôi và được chia hai khi dịch phải một bit.

Thí dụ số 1010.00 = 10<sub>10</sub> khi dịch trái thành 10100.0 = 20<sub>10</sub> và khi dịch phải thành 101.000 = 5<sub>10</sub>.

- Trong máy tính thanh ghi (tên thường gọi của mạch ghi dịch) là nơi lưu tạm dữ liệu để thực hiện các phép tính, các lệnh cơ bản như quay, dịch ....
- Ngoài ra, mạch ghi dịch còn những ứng dụng khác như: tạo mạch đếm vòng, biến đổi dữ liệu nối tiếp  $\leftrightarrow$  song song, dùng thiết kế các mạch đèn trang trí, quang báo. ....

## MẠCH ĐẾM

Lợi dụng tính đảo trạng thái của FF JK khi  $J=K=1$ , người ta thực hiện các mạch đếm.

Chức năng của mạch đếm là đếm số xung  $C_K$  đưa vào ngõ vào hoặc thể hiện số trạng thái có thể có của các ngõ ra.

Nếu xét khía cạnh tần số của tín hiệu thì mạch đếm có chức năng chia tần, nghĩa là tần số của tín hiệu ở ngõ ra là kết quả của phép chia tần số của tín hiệu  $C_K$  ở ngõ vào cho số đếm của mạch.

Ta có các loại: mạch đếm đồng bộ, không đồng bộ và đếm vòng.

### Mạch đếm đồng bộ

Trong mạch đếm đồng bộ các FF chịu tác động đồng thời của xung đếm  $C_K$ .

### Mạch đếm đồng bộ n tầng, đếm lên

Để thiết kế mạch đếm đồng bộ n tầng (lấy thí dụ  $n=4$ ), trước tiên lập bảng trạng thái, quan sát bảng trạng thái suy ra cách mắc các ngõ vào JK của các FF sao cho mạch giao hoán tạo các ngõ ra đúng như bảng đã lập. Giả sử ta dùng FF tác động bởi cạnh xuống của xung  $C_K$  (Thực ra, kết quả thiết kế không phụ thuộc vào chiều tác động của xung  $C_K$ , tuy nhiên điều này phải được thể hiện trên mạch nên ta cũng cần lưu ý). Với 4 FF mạch đếm được  $2^4=16$  trạng thái và số đếm được từ 0 đến 15. Ta có bảng trạng thái:

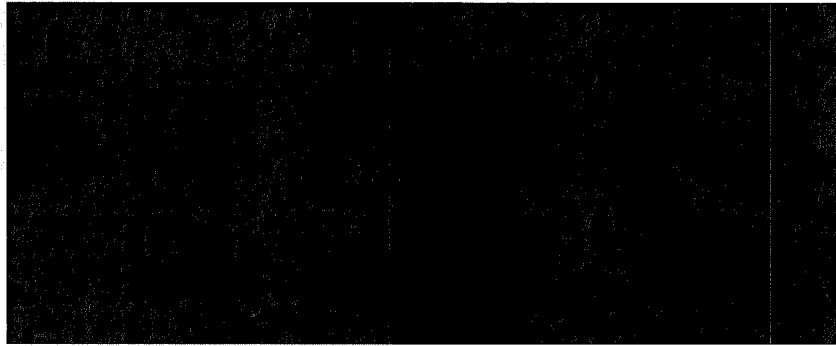
$C_K$	$Q_D$	$Q_C$	$Q_B$	$Q_A$	Số đếm
Xóa	0	0	0	0	0
1↓	0	0	0	1	1
2↓	0	0	1	0	2
3↓	0	0	1	1	3
4↓	0	1	0	0	4
5↓	0	1	0	1	5
6↓	0	1	1	0	6
7↓	0	1	1	1	7
8↓	1	0	0	0	8
9↓	1	0	0	1	9
10↓	1	0	1	0	10
11↓	1	0	1	1	11
12↓	1	1	0	0	12
13↓	1	1	0	1	13
14↓	1	1	1	0	14
15↓	1	1	1	1	15
16↓	0	0	0	0	0

Bảng 5.14

Nhận thấy:

- FF A đổi trạng thái sau từng xung  $C_k$ , vậy:  $T_A = J_A = K_A = 1$
- FF B đổi trạng thái nếu trước đó  $Q_A = 1$ , vậy:  $T_B = J_B = K_B = Q_A$
- FF C đổi trạng thái nếu trước đó  $Q_A = Q_B = 1$ , vậy:  $T_C = J_C = K_C = Q_A \cdot Q_B$
- FF D đổi trạng thái nếu trước đó  $Q_A = Q_B = Q_C = 1$ , vậy:  
 $T_D = J_D = K_D = Q_A \cdot Q_B \cdot Q_C = T_C \cdot Q_C$

Ta được kết quả ở (H 5.16).



(H 5.16)

### Mạch đếm đồng bộ n tầng, đếm xuống

Bảng trạng thái:

$C_k$	$Q_D$	$Q_C$	$Q_B$	$Q_A$	Số đếm
Xóa	0√	0√	0√	0	0
1↓	1	1	1	1	15
2↓	1	1	1√	0	14
3↓	1	1	0	1	13
4↓	1	1√	0√	0	12
5↓	1	0	1	1	11
6↓	1	0	1√	0	10
7↓	1√	0√	0√	0	9
8↓	0	1	1	1	8
9↓	0	1	1√	0	7
10↓	0	1	0	1	6
11↓	0	1√	0√	0	5
12↓	0	0	1	1	4
13↓	0	0	1√	0	3
14↓	0	0	0	1	2
15↓	0	0	0	0	1
16↓	0	0	0	0	0

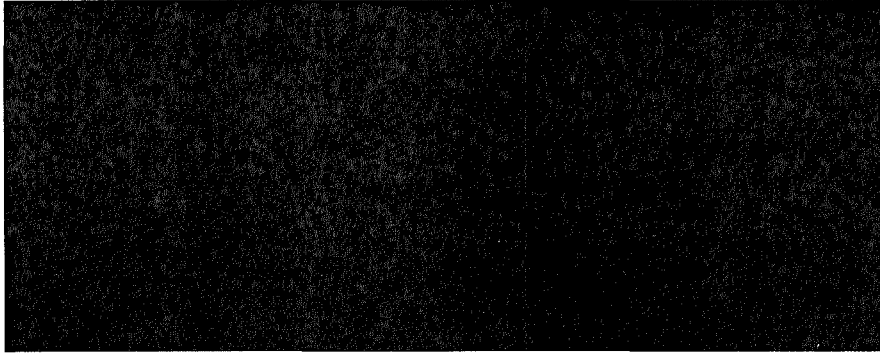
Bảng 5.15

Nhận thấy:

- FF A đổi trạng thái sau từng xung  $C_k$ , vậy:  $T_A = J_A = K_A = 1$
- FF B đổi trạng thái nếu trước đó  $Q_A = 0$ , vậy:  $T_B = J_B = K_B = \overline{Q_A}$
- FF C đổi trạng thái nếu trước đó  $Q_A = Q_B = 0$ , vậy:  $T_C = J_C = K_C = \overline{Q_A} \cdot \overline{Q_B}$
- FF D đổi trạng thái nếu trước đó  $Q_A = Q_B = Q_C = 0$ , vậy:

$$T_D = J_D = K_D = \overline{Q_A} \cdot \overline{Q_B} \cdot \overline{Q_C} = T_C \cdot \overline{Q_C}$$

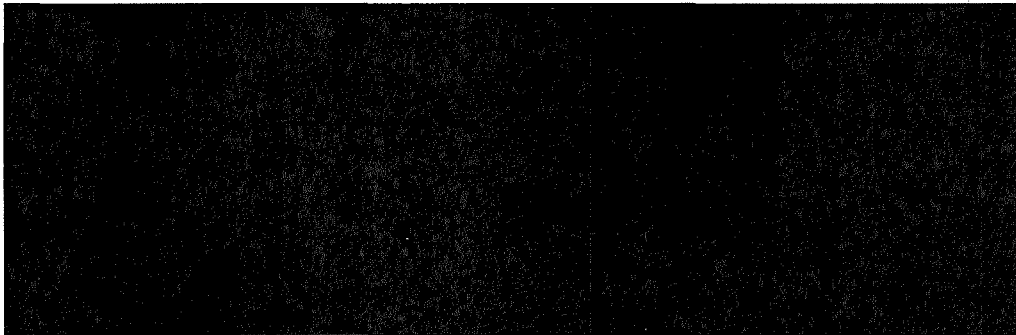
Ta được kết quả ở (H 5.17)



(H 5.17)

### Mạch đếm đồng bộ n tầng, đếm lên/ xuống

Để có mạch đếm n tầng, đếm lên hoặc xuống ta dùng một đa hợp 2→1 có ngõ vào điều khiển C để chọn Q hoặc  $\overline{Q}$  đưa vào tầng sau qua các cổng AND. Trong mạch (H 5.18) dưới đây khi C=1 mạch đếm lên và khi C=0 mạch đếm xuống.



(H 5.18)

### Tần số hoạt động lớn nhất của mạch đếm đồng bộ n tầng:

Trong mạch (H 5.16) ta cần 2 cổng AND. Trong trường hợp tổng quát cho n tầng, số cổng AND là (n-2) như vậy thời gian tối thiểu để tín hiệu truyền qua mạch là:

$$T_{\min} = T_{\text{PFF}} + T_{\text{P,AND}}(n-2)$$

Tần số cực đại xác định bởi:

$$f_{\max} = \frac{1}{T_{\min}} = \frac{1}{t_{\text{PFF}} + (n-2)t_{\text{P,AND}}}$$

Để gia tăng tần số làm việc của mạch, thay vì dùng các cổng AND 2 ngõ vào ta phải dùng cổng AND nhiều ngõ vào và mắc theo kiểu:

$$\begin{aligned} T_A = J_A = K_A &= 1 & T_B = J_B = K_B &= Q_A \\ T_C = J_C = K_C &= Q_A \cdot Q_B & T_D = J_D = K_D &= Q_A \cdot Q_B \cdot Q_C \end{aligned}$$

Như vậy tần số làm việc không phụ thuộc vào n và bằng:

$$f_{\max} = \frac{1}{T_{\text{PFF}} + T_{\text{P,AND}}}$$

### Mạch đếm đồng bộ Modulo - N ( $N \neq 2^n$ )

Để thiết kế mạch đếm modulo - N, trước nhất ta phải chọn số tầng.

Số tầng n phải thỏa điều kiện:

$$2^{n-1} < N < 2^n$$

Thí dụ thiết kế mạch đếm 10 ( $N = 10$ ).

$$2^{4-1} < 10 < 2^4$$

Vậy số tầng là 4

Có nhiều phương pháp thiết kế mạch đếm đồng bộ modulo-N.

Sau đây ta khảo sát hai phương pháp : dùng **hàm Chuyển** và **MARCUS**

#### ◆ Phương pháp dùng hàm Chuyển (Transfer function)

Hàm Chuyển là hàm cho thấy có sự thay đổi trạng thái của FF. Mỗi loại FF có một hàm Chuyển riêng của nó.

Hàm Chuyển được định nghĩa như sau: hàm có trị 1 khi có sự thay đổi trạng thái của FF ( $Q_+ \neq Q$ ) và trị 0 khi trạng thái FF không đổi ( $Q_+ = Q$ ).

Chúng ta chỉ thiết kế mạch đếm dùng FF JK do đó ta chỉ xác định hàm Chuyển của loại FF này.

Bảng trạng thái của FF JK (Bảng 5.16)

$C_K$	J	K	Q	$Q_+$	H
↓	0	0	0	0	0
↓	0	0	1	1	0
↓	0	1	0	0	0
↓	0	1	1	0	1
↓	1	0	0	1	1
↓	1	0	1	1	0
↓	1	1	0	1	1
↓	1	1	1	0	1

Bảng 5.16

Dùng Bảng Karnaugh ta suy ra được biểu thức của H:  $H = \overline{JQ} + KQ$

Để thiết kế mạch đếm cụ thể ta sẽ xác định hàm H cho từng FF trong mạch, so sánh với biểu thức của hàm H suy ra J, K của các FF. Dưới đây là một thí dụ.

Thiết kế mạch đếm 10 đồng bộ dùng FF JK

Bảng trạng thái của mạch đếm 10 và giá trị của các hàm H tương ứng:

$C_K$	$Q_D$	$Q_C$	$Q_B$	$Q_A$	$Q_D$	$Q_C$	$Q_B$	$Q_A$	$H_D$	$H_C$	$H_B$	$H_A$
					+	+	+	+				
1↓	0	0	0	0	0	0	0	1	0	0	0	1
2↓	0	0	0	1	0	0	1	0	0	0	1	1
3↓	0	0	1	0	0	0	1	1	0	0	0	1
4↓	0	0	1	1	0	1	0	0	0	1	1	1
5↓	0	1	0	0	0	1	0	1	0	0	0	1
6↓	0	1	0	1	0	1	1	0	0	0	1	1
7↓	0	1	1	0	0	1	1	1	0	0	0	1
8↓	0	1	1	1	1	0	0	0	1	1	1	1

9↓	1	0	0	0	1	0	0	1	0	0	0	1
10↓	1	0	0	1	0	0	0	0	1	0	0	1

Bảng 5.17

Từ bảng 5.17, ta thấy:

$$H_A = 1 = Q_A + \overline{Q_A} \Rightarrow J_A = K_A = 1$$

Để xác định  $H_B$ ,  $H_C$  và  $H_D$  ta phải vẽ bảng Karnaugh

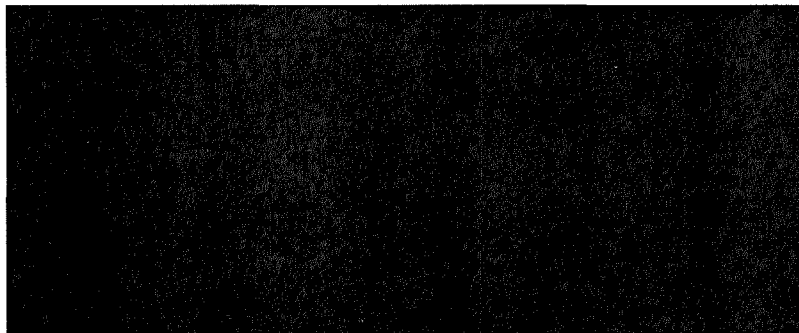


$$\begin{aligned}
 H_B &= \overline{Q_D} Q_A \overline{Q_B} + Q_D Q_A Q_B & H_C &= Q_B Q_A \overline{Q_C} + Q_B Q_A Q_C & H_D &= Q_C Q_B Q_A \overline{Q_D} + Q_A Q_D \\
 \Rightarrow J_B = K_B &= \overline{Q_D} Q_A & \Rightarrow J_C = K_C &= Q_B Q_A & \Rightarrow J_D = Q_C Q_B Q_A, K_D &= Q_A
 \end{aligned}$$

(H 5.19)

**Ghi chú:** Trong kết quả của hàm  $H$  ta muốn có chứa  $Q$  và  $\overline{Q}$  tương ứng để suy ra ngay các trị  $J$  và  $K$  nên ta đã chia bảng Karnaugh ra làm 2 phần chứa  $Q$  và  $\overline{Q}$  và nhóm riêng từng phần này.

Từ các kết quả này, ta vẽ được mạch (H 5.20)



(H 5.20)

Bây giờ ta có thể kiểm tra xem nếu như vì một lý do nào đó, số đếm rơi vào các trạng thái không sử dụng (tương ứng với số từ 10 đến 15) thì khi có xung đồng hồ trạng thái tiếp theo sẽ như thế nào? Mạch có quay về để đếm tiếp?

Áp dụng các hàm chuyển có được, ứng với mỗi trạng thái  $Q$  của từng FF trong các tổ hợp không sử dụng, ta tìm trị  $H$  tương ứng rồi suy ra  $Q_+$ , ta được bảng kết quả sau:

$C_K$	$Q_D$	$Q_C$	$Q_B$	$Q_A$	$H_D$	$H_C$	$H_B$	$H_A$	$Q_D$	$Q_C$	$Q_B$	$Q_A$
									+	+	+	+
↓	1	0	1	0	0	0	0	1	1	0	1	1
↓	1	0	1	1	1	1	0	1	0	1	1	0
↓	1	1	0	0	0	0	0	1	1	1	0	1
↓	1	1	0	1	1	0	0	1	0	1	0	0
↓	1	1	1	0	0	0	0	1	1	1	1	1
↓	1	1	1	1	0	1	0	1	0	0	1	0



Bảng 5.18

Từ bảng kết quả ta có kết luận:

- Khi ngã ra rơi vào trạng thái  $10_{10}$  (1010), nó sẽ nhảy tiếp vào trạng thái  $11_{10}$  (1011) rồi sau đó nhảy về  $6_{10}$  (0110) (Dòng 1 và 2)
- Khi ngã ra rơi vào trạng thái  $12_{10}$  (1100), nó sẽ nhảy tiếp vào trạng thái  $13_{10}$  (11 01) rồi sau đó nhảy về  $4_{10}$  (0100) (Dòng 3 và 4)
- Khi ngã ra rơi vào trạng thái  $14_{10}$  (1110), nó sẽ nhảy tiếp vào trạng thái  $15_{10}$  (1111) rồi sau đó nhảy về  $2_{10}$  (0010) (Dòng 5 và 6).

Tóm lại, nếu có một sự cố xảy ra làm cho số đếm rơi vào các trạng thái không sử dụng thì sau 1 hoặc 2 số đếm nó tự động quay về một trong các số đếm từ 0 đến 9 rồi tiếp tục đếm bình thường.

◆ Phương pháp MARCUS

Phương pháp MARCUS cho phép xác định các biểu thức của J và K dựa vào sự thay đổi của  $Q_+$  so với Q

Từ bảng trạng thái của FF JK (Bảng 5.7) ta có thể viết lại Bảng 5.19:

Q	$Q_+$	J	K
0	0	0	x
0	1	1	x
1	0	x	1
1	1	x	0

Bảng 5.19

Để thiết kế mạch, ta so sánh  $Q_+$  và Q để có được bảng sự thật cho J, K của từng FF, sau đó xác định J và K.

Thí dụ thiết kế lại mạch đếm 10 bằng phương pháp MARCUS

Bảng sự thật cho J, K của từng FF

$C_K$	$Q_D$	$Q_C$	$Q_B$	$Q_A$	$J_D$	$K_D$	$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
1↓	0	0	0	0	0	x	0	x	0	x	1	x
2↓	0	0	0	1	0	x	0	x	1	x	x	1
3↓	0	0	1	0	0	x	0	x	x	0	1	x
4↓	0	0	1	1	0	x	1	x	x	1	x	1
5↓	0	1	0	0	0	x	x	0	0	x	1	x
6↓	0	1	0	1	0	x	x	0	1	x	x	1
7↓	0	1	1	0	0	x	x	0	x	0	1	x
8↓	0	1	1	1	1	x	x	1	x	1	x	1
9↓	1	0	0	0	x	0	0	x	0	x	1	x
10↓	1	0	0	1	x	1	0	x	0	x	x	1

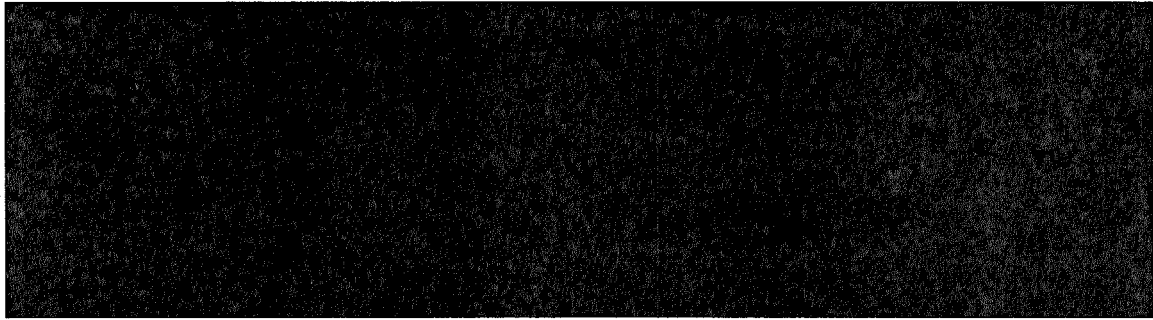
Bảng 5.20

**Ghi chú:** Trong bảng 5.20, không có các cột cho  $Q_+$ , tuy nhiên ta có thể thấy ngay là dòng bên dưới chính là  $Q_+$  của dòng bên trên, như vậy kết quả có được từ sự so sánh dòng trên và dòng ngay dưới nó.

Ta thấy ngay  $J_A = K_A = 1$

Dùng bảng Karnaugh để xác định các hàm còn lại

Nhận thấy các FF B và C có thể xác định chung cho J và K (cùng vị trí 1 và x), FF D được xác định J và K riêng



$$J_B = K_B = \overline{Q_D} Q_A \quad J_C = K_C = Q_B \cdot Q_A \quad J_D = Q_C \cdot Q_B \cdot Q_A \quad K_D = Q_A$$

(H 5.21)

Ta được lại kết quả trên.

Trên thị trường có khá nhiều IC đếm:

- 4 bit BCD: 74160, 74162, 74190, 74192, 4192, 4510, 4518. ...
- 4 bit nhị phân: 74161, 74163, 74191, 74193, 4193, 4516, 4520. ...
- 8 bit nhị phân: 74269, 74579, 74779. ...

### **Mạch đếm không đồng bộ**

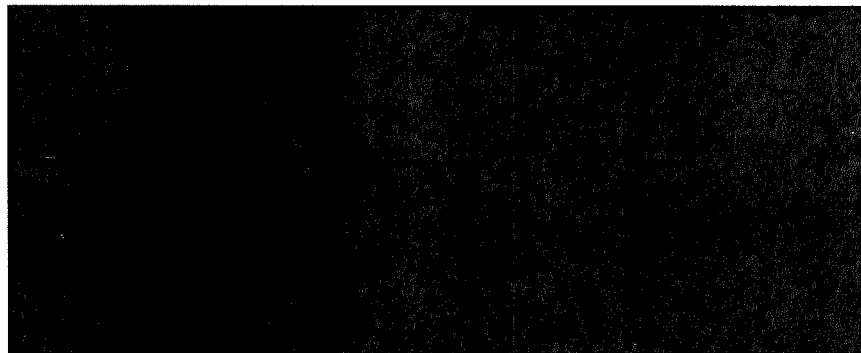
Là các mạch đếm mà các FF không chịu tác động đồng thời của xung  $C_K$ .

Khi thiết kế mạch đếm không đồng bộ ta phải quan tâm tới **chiều tác động của xung**

đồng hồ  $C_K$ .

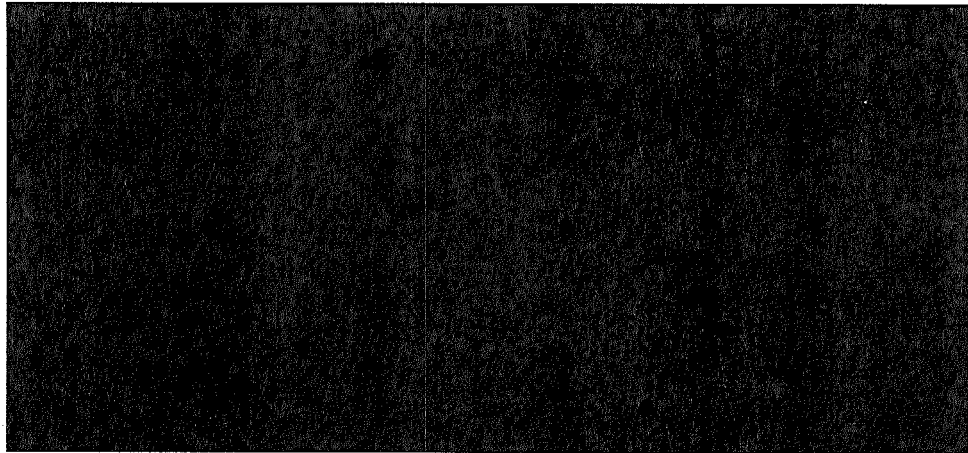
### **Mạch đếm không đồng bộ, n tầng, đếm lên (n=4):**

Từ bảng trạng thái 5.14 của mạch đếm 4 bit, ta thấy nếu dùng FF JK tác động bởi **cạnh xuống của xung đồng hồ** thì có thể lấy ngõ ra của tầng trước làm xung đồng hồ  $C_K$  cho tầng sau, với điều kiện các ngõ vào JK của các FF đều được đưa lên mức cao. Ta được mạch đếm không đồng bộ, 4 bit, đếm lên (H 5.22).



(H 5.22)

(H 5.23) là dạng tín hiệu xung  $C_K$  và các ngõ ra của các FF



(H 5.23)

Tổ hợp các số tạo bởi các ngõ ra các FF D, C, B, A là số nhị phân từ 0 đến 15

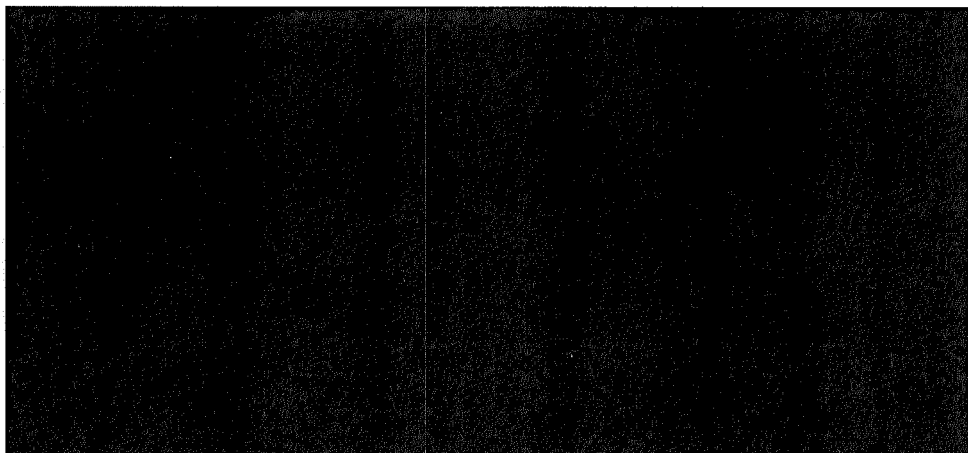
**Mạch đếm không đồng bộ, n tầng, đếm xuống (n=4):**

Để có mạch đếm xuống ta nối  $\bar{Q}$  (thay vì Q) của tầng trước vào ngõ vào  $C_K$  của tầng sau. (H 5.24) là mạch đếm xuống 4 tầng.

Dạng sóng ở ngõ ra các FF và số đếm tương ứng cho ở (H 5.25)



(H 5.24)



(H 5.25)

Quan sát tín hiệu ra ở các Flipflop ta thấy sau mỗi FF tần số của tín hiệu ra giảm đi một nửa, nghĩa là:

$$f_{QA} = \frac{f_{CK}}{2}$$

$$f_{Q_B} = \frac{f}{2} = \frac{f}{2^1} = \frac{f}{2}$$

$$f_{Q_C} = \frac{f}{4} = \frac{f}{2^2} = \frac{f}{4}$$

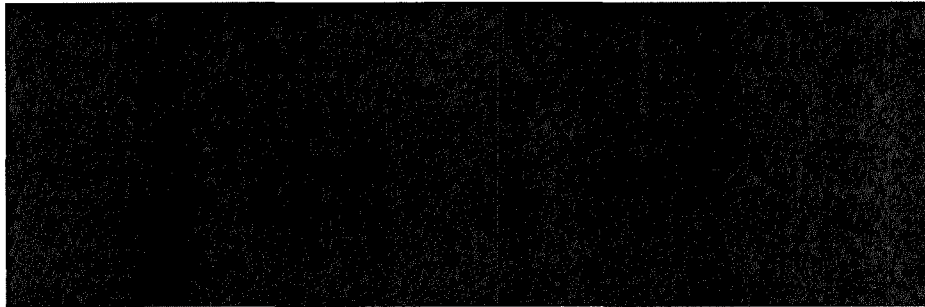
$$f_{Q_D} = \frac{f}{8} = \frac{f}{2^3} = \frac{f}{8}$$

Như vậy xét về khía cạnh tần số, ta còn gọi mạch đếm là mạch chia tần.

### Mạch đếm không đồng bộ, n tầng, đếm lên, xuống (n=4):

Để có mạch đếm lên hoặc đếm xuống người ta dùng các mạch đa hợp 2→1 với ngõ vào điều khiển C chung để chọn Q hoặc  $\bar{Q}$  của tầng trước nối vào  $C_K$  tầng sau tùy theo yêu cầu về cách đếm.

Trong (H 5.26), khi  $C=1$ , Q nối vào  $C_K$ , mạch đếm lên và khi  $C=0$ ,  $\bar{Q}$  nối vào  $C_K$ , mạch đếm xuống



$c = 0$  : đếm xuống       $c = 1$  : đếm lên  
(H 5.26)

Trên thực tế, để đơn giản, ta có thể thay đa hợp 2→1 bởi một cổng EX-OR, ngõ điều khiển C nối vào một ngõ vào cổng EX-OR, ngõ vào còn lại nối với ngõ ra Q của FF và ngõ ra của cổng EX-OR nối vào ngõ vào  $C_K$  của FF sau, mạch cũng đếm lên/xuống tùy vào  $C=0$  hay  $C=1$ .



$c = 1$  : đếm xuống       $c = 0$  : đếm lên  
(H 5.27)

### Mạch đếm không đồng bộ modulo - N (N=10)

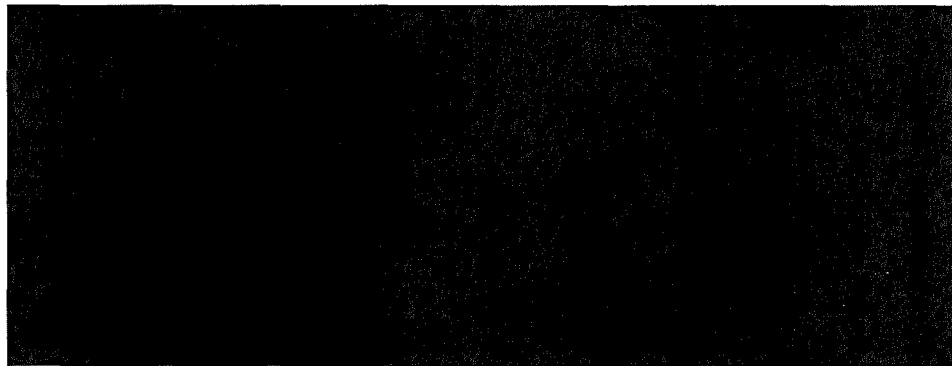
◆ Kiểu Reset:

Để thiết kế mạch đếm kiểu Reset, trước nhất người ta lập bảng trạng thái cho số đếm (Bảng 5.21)

Quan sát bảng 5.21 ta thấy ở xung thứ 10, nếu theo cách đếm 4 tầng thì  $Q_D$  và  $Q_B$  phải lên 1. Lợi dụng hai trạng thái này ta dùng một công NAND 2 ngõ vào để đưa tín hiệu về xóa các FF, ta được mạch đếm ở (H 5.28).

Số xung $C_K$ vào	Số $Q_D$	Nhi $Q_C$	Phân $Q_B$	Ra $Q_A$	Số thập phân tương ứng
Xóa	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	0(1)	0	0(1)	0	10

Bảng 5.21



(H 5.28)

Mạch đếm kiểu Reset có khuyết điểm như:

- Có một trạng thái trung gian trước khi đạt số đếm cuối cùng.
- Ngõ vào Cl không được dùng cho chức năng xóa ban đầu.

◆ Kiểu Preset:

Trong kiểu Preset các ngõ vào của các FF sẽ được đặt trước thế nào để khi mạch đếm đến trạng thái thứ N thì tất cả các FF tự động quay về không.

Để thiết kế mạch đếm không đồng bộ kiểu Preset, thường người ta làm như sau:

- Phân tích số đếm  $N = 2^n \cdot N'$  ( $N' < N$ ) rồi kết hợp hai mạch đếm n bit và  $N'$ . Việc thiết kế rất đơn giản khi số  $N' \ll N$

- Quan sát bảng trạng thái và kết hợp với phương pháp thiết kế mạch đếm đồng bộ (MARCUS hay hàm chuyển) để xác định JK của các FF.

Thí dụ, để thiết kế mạch đếm 10, ta phân tích  $10 = 2 \times 5$  và ta chỉ cần thiết kế mạch đếm 5 rồi kết hợp với một FF (đếm 2)

Bảng trạng thái của mạch đếm 5.

Số xung $C_K$ vào	Số Nhi $Q_D$	Phân $Q_C$	Ra $Q_B$	Số thập phân tương ứng
Xóa	0	0	0	0

1	0	0	1	1
2	0	1	0	2
3	0	1	1	3
4	1	0	0	4
5	0	0	0	0

Bảng 5.22

Giả sử dùng FF JK có xung  $C_K$  tác động cạnh xuống.

Từ bảng 5.21, ta thấy có thể dùng tín hiệu ngõ ra FF B làm xung đồng hồ cho FF C và đưa  $J_C$  và  $K_C$  lên mức cao:

$$C_{KC} = Q_B; \quad J_C = K_C = 1$$

Các FF B và D sẽ dùng xung  $C_K$  của hệ thống và các ngõ vào JK được xác định nhờ hàm chuyển:

$C_K$	$Q_D$	$Q_C$	$Q_B$	$H_D$	$H_B$
1↓	0	0	0	0	1
2↓	0	0	1	0	1
3↓	0	1	0	0	1
4↓	0	1	1	1	1
5↓	1	0	0	1	0
	0	0	0		

Bảng 5.23

Dùng bảng Karnaugh xác định  $H_D$  và  $H_B$  rồi suy ra các trị J, K của các FF.



$$H_D = Q_C \cdot Q_B \cdot Q_D + Q_D$$

$$\Rightarrow J_D = Q_C \cdot Q_B; \quad K_D = 1$$



$$H_B = Q_D \cdot Q_B + Q_B$$

$$\Rightarrow J_B = Q_D; \quad K_B = 1$$

(H 5.29)

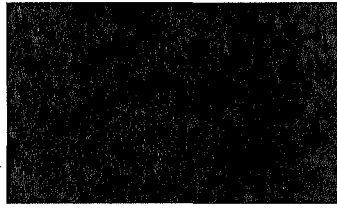
Có thể xác định J, K của các FF B và D bằng phương pháp MARCUS:

$C_K$	$Q_D$	$Q_C$	$Q_B$	$J_D$	$K_D$	$J_B$	$K_B$
1↓	0	0	0	0	x	1	x
2↓	0	0	1	0	x	x	1
3↓	0	1	0	0	x	1	x
4↓	0	1	1	1	x	x	1
5↓	1	0	0	x	1	0	x
	0	0	0				

Bảng 5.24

Ta có ngay  $K_D = K_B = 1$

Dùng bảng Karnaugh xác định  $J_D$  và  $J_B$



$$J_D = Q_C \cdot Q_B$$



$$J_B = \overline{Q_D}$$

(H 5.30)

(H 5.31) là mạch đếm 10 thiết kế theo kiểu đếm 2x5 với mạch đếm 5 có được từ kết quả trên.



(H 5.31)

IC 7490 là IC đếm 10, có cấu tạo như mạch (H 5.31) thêm các ngõ vào Reset 0 và Reset 9 có sơ đồ mạch (H 5.32)



(H 5.32)

Bảng 5.25 là bảng sự thật cho các ngõ vào Reset

Reset Inputs				Outputs			
R <sub>0</sub> (1)	R <sub>0</sub> (2)	R <sub>9</sub> (1)	R <sub>9</sub> (2)	Q <sub>D</sub>	Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>
1	1	0	x	0	0	0	0
1	1	x	0	0	0	0	0
0	x	1	1	1	0	0	1
x	0	1	1	1	0	0	1
x	0	x	0	Đếm	Đếm	Đếm	Đếm
0	x	0	x	nt	nt	nt	nt
0	x	x	0	nt	nt	nt	nt
x	0	0	x	nt	nt	nt	nt

Bảng 5.25

Dùng IC 7490, có thể thực hiện một trong hai cách mắc:

- ◆ Mạch đếm 2x5: Nối Q<sub>A</sub> vào ngõ vào B, xung đếm (C<sub>K</sub>) vào ngõ vào A

◆ Mạch đếm 5x2: Nối  $Q_D$  vào ngõ vào A, xung đếm ( $C_K$ ) vào ngõ vào B

Hai cách mắc cho kết quả số đếm khác nhau nhưng cùng một chu kỳ đếm 10. Tần số tín hiệu ở ngõ ra sau cùng bằng 1/10 tần số xung  $C_K$  (nhưng dạng tín hiệu ra khác nhau). Dưới đây là hai bảng trạng thái cho hai trường hợp nói trên.

$Q_D$	$Q_C$	$Q_B$	$Q_A$
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1

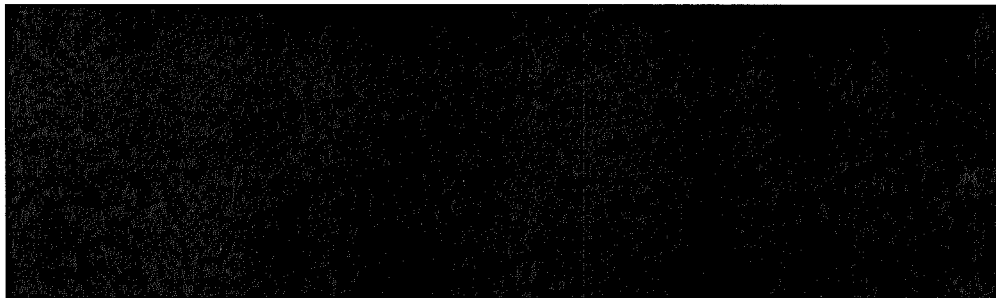
Bảng 5.26 : Đếm 2x5

$Q_D$	$Q_C$	$Q_B$	$Q_A$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1

Bảng 5.27 : Đếm 5x2

(H 5.33) cho thấy dạng sóng ở các ngõ ra của hai mạch cùng đếm 10 nhưng hai kiểu đếm khác nhau:

- Kiểu đếm 2x5 cho tín hiệu ra ở  $Q_D$  không đối xứng
- Kiểu đếm 5x2 cho tín hiệu ra ở  $Q_A$  đối xứng



(H 5.33)

### Mạch đếm vòng

Thực chất là mạch ghi dịch trong đó ta cho hồi tiếp từ một ngõ ra nào đó về ngõ vào để thực hiện một chu kỳ đếm. Tùy đường hồi tiếp mà ta có các chu kỳ đếm khác nhau. Sau đây ta khảo sát vài loại mạch đếm vòng phổ biến.

Hồi tiếp từ  $Q_D$  về  $J_A$  và  $Q_D$  về  $K_A$



(H 5.34)

Đối với mạch này, sự đếm vòng chỉ thấy được khi có đặt trước ngõ ra



- Đặt trước  $Q_A = 1$ , ta được kết quả như bảng 5.28.

$C_K$	$Q_D$	$Q_C$	$Q_B$	$Q_A$	Số TP
Preset	0	0	0	1	1
1↓	0	0	1	0	2
2↓	0	1	0	0	4
3↓	1	0	0	0	8
4↓	0	0	0	1	1
:	:	:	:	:	:

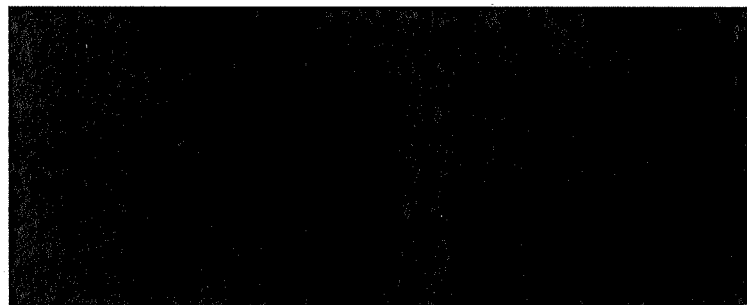
Bảng 5.28

Nếu đặt trước  $Q_A = Q_B = 1$  ta có bảng 5.29

$C_K$	$Q_D$	$Q_C$	$Q_B$	$Q_A$	Số TP
Preset	0	0	1	1	3
1↓	0	1	1	0	6
2↓	1	1	0	0	12
3↓	1	0	0	1	9
4↓	0	0	1	1	3
:	:	:	:	:	:

Bảng 5.29

### Hồi tiếp từ $\overline{Q_D}$ về $J_A$ và $Q_D$ về $K_A$ (H 5.35)



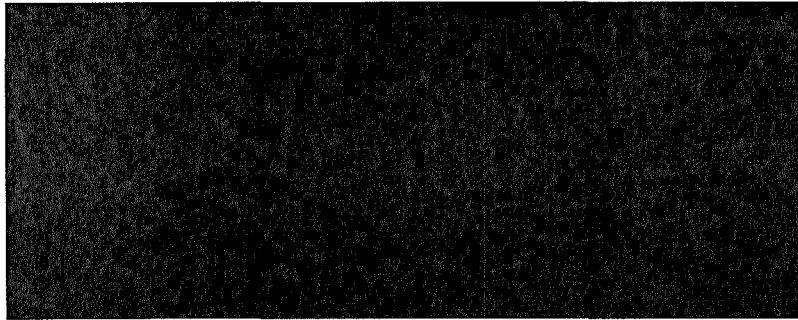
(H 5.35)

Mạch này còn có tên là mạch đếm Johnson. Mạch có một chu kỳ đếm mặc nhiên mà không cần đặt trước và nếu có đặt trước, mạch sẽ cho các chu kỳ khác nhau tùy vào tổ hợp đặt trước đó. Bảng 5.30 là chu kỳ đếm mặc nhiên.

$C_K$	$Q_D$	$Q_C$	$Q_B$	$Q_A$	Số TP
Preset	0	0	0	0	0
1↓	0	0	0	1	1
2↓	0	0	1	1	3
3↓	0	1	1	1	7
4↓	1	1	1	1	15
5↓	1	1	1	0	14
6↓	1	1	0	0	12
7↓	1	0	0	0	8
8↓	0	0	0	0	0

Bảng 5.30

Hồi tiếp từ  $\overline{Q_D}$  về  $J_A$  và  $Q_C$  về  $K_A$  (H 5.36)



(H 5.36)

$C_K$	$Q_D$	$Q_C$	$Q_B$	$Q_A$	Số TP
Preset	0	0	0	0	0
1↓	0	0	0	1	1
2↓	0	0	1	1	3
3↓	0	1	1	1	7
4↓	1	1	1	0	14
5↓	1	1	0	0	12
6↓	1	0	0	0	8
7↓	0	0	0	0	0

Bảng 5.31

Vài thí dụ thiết kế mạch đếm

1. Dùng FF JK thiết kế mạch đếm 6, đồng bộ

Bảng trạng thái và hàm chuyển mạch đếm 6:

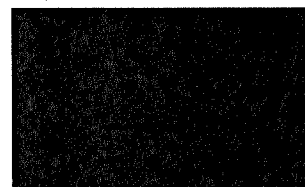
N	$Q_A$	$Q_B$	$Q_C$	$Q_{A+}$	$Q_{B+}$	$Q_{C+}$	$H_A$	$H_B$	$H_C$
0	0	0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0	1	1
2	0	1	0	0	1	1	0	0	1
3	0	1	1	1	0	0	1	1	1
4	1	0	0	1	0	1	0	0	1
5	1	0	1	0	0	0	1	0	1

Bảng 5.32

$H_C = 1 \Rightarrow J_C = K_C = 1$

Xác định  $J_A, K_A, J_B, K_B$

Bảng Karnaugh cho hai hàm chuyển  $H_A$  &  $H_B$



(H 5.37)

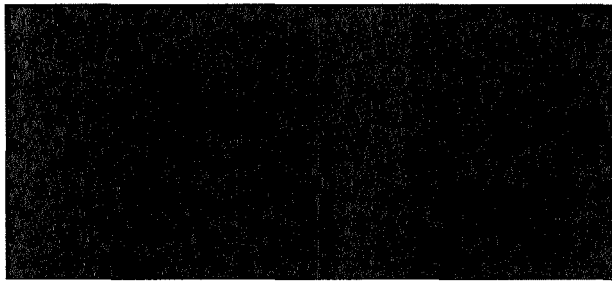
$$H_A = Q_B Q_C \overline{Q_A} + Q_C Q_A$$

$$\Rightarrow J_A = Q_B Q_C ; K_A = Q_C$$

$$H_B = \overline{Q_A} Q_C \overline{Q_B} + Q_C Q_B$$

$$\Rightarrow J_B = \overline{Q_A} Q_C ; K_B = Q_C$$

Mạch:



(H 5.38)

2. Thiết kế mạch đếm 7 không đồng bộ, dùng FF JK có ngõ vào xung đồng hồ tác động bởi cạnh lên của  $C_K$ .

Bảng trạng thái

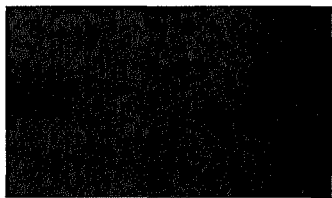
N	$Q_A$	$Q_B$	$Q_C$	$J_B$	$K_B$	$J_C$	$K_C$
0↑	0	0	0	0	x	1	x
1↑	0	0	1	1	x	x	1
2↑	0	1	0	x	0	1	x
3↑	0	1	1	x	1	x	1
4↑	1	0	0	0	x	1	x
5↑	1	0	1	1	x	x	1
6↑	1	1	0	x	1	0	x
	0	0	0				

Bảng 5.33

Nhận xét bảng trạng thái ta thấy mỗi lần  $Q_B$  thay đổi từ 1 xuống 0 thì  $Q_A$  đổi trạng thái, mà FF có xung đồng hồ tác động bởi cạnh lên nên ta có thể lấy  $\overline{Q_B}$  làm xung đồng hồ cho FFA và  $J_A=K_A=1$ .

FF B và FFC sẽ dùng xung đồng hồ hệ thống, dùng phương pháp MARCUS để xác định J & K của các FF này.

Ta thấy ngay  $K_C=1$

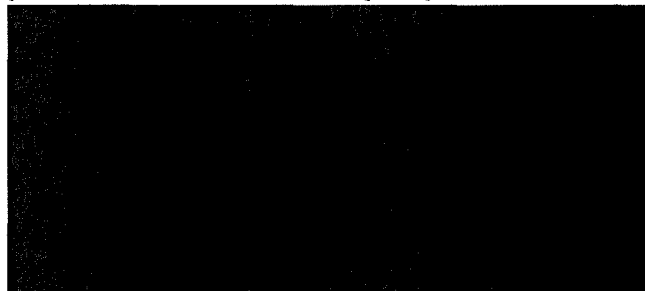


(H 5.39)

$$J_B=Q_C$$

$$K_B=Q_A+Q_C$$

$$J_C=\overline{Q_A} + \overline{Q_B}$$



(H 5.40)

## BÀI TẬP

1. Thiết kế bộ đếm đồng bộ có dãy đếm sau: 000, 010, 101, 110 và lặp lại.
2. Làm lại bài 1. Thêm điều kiện các trạng thái không sử dụng 001, 011, 100 và 111 phải luôn luôn nhảy về 000 ở xung đồng hồ kế tiếp.
3. Thiết kế bộ đếm đồng bộ dùng FF-JK với dãy đếm sau: 000, 001, 011, 010, 110, 111, 101, 100, 000 . . .
4.
  - a. Thiết kế một mạch đếm đồng bộ dùng FF-JK tác động cạnh xuống, có dãy đếm như sau: 000, 001, 011, 111, 110, 100, 001. . . Những trạng thái không sử dụng được đưa về trạng thái 000 ở xung đồng hồ kế tiếp. Vẽ sơ đồ mạch.
  - b. Mắc nối tiếp một bộ đếm 2 (Dùng FF-JK, tác động cạnh xuống) với bộ đếm đã được thiết kế ở câu a. Vẽ dạng sóng ở các ngõ ra của bộ đếm giả sử trạng thái ban đầu của các ngõ ra đều bằng 0. Xác định dãy đếm của mạch.
5. Thiết kế mạch đếm đồng bộ modulo-12 dùng FF JK.  
Dùng ngõ ra mạch đếm để điều khiển hệ thống đèn giao thông:
  - Đèn xanh cháy trong 40 s
  - Đèn vàng cháy trong 20s
  - Đèn đỏ cháy trong 40s
  - Đèn vàng và đỏ cùng cháy trong 20s. Chu kỳ lặp lạiCho chu kỳ xung đồng hồ là 10s.
6. Thiết kế mạch đếm đồng bộ dùng FF JK có ngõ vào điều khiển X:
  - Khi  $X=0$  mạch đếm theo thứ tự 0, 2, 4, 6 rồi trở về 0
  - Khi  $X=1$  mạch đếm 0, 6, 4, 2 rồi trở về 0.Các trạng thái không sử dụng trong hai lần đếm đều trở về 0 khi có xung đồng hồ

## CHƯƠNG 6: BỘ NHỚ

### Số bù

Cho số dương  $N$ ,  $n$  bit, các số bù của  $N$  được định nghĩa:

**Số bù 2:**  $(N)_2 = 2^n - N$  (số  $2^n$  gồm bit 1 và  $n$  bit 0 theo sau)

**Số bù 1:**  $(N)_1 = (N)_2 - 1 = 2^n - N - 1$

**Thí dụ 1:**  $N = 1010$

Số bù 2 của  $N$  là  $(N)_2 = 10000 - 1010 = 0110$

Và số bù 1 của  $N$  là  $(N)_1 = 0110 - 1 = 0101$

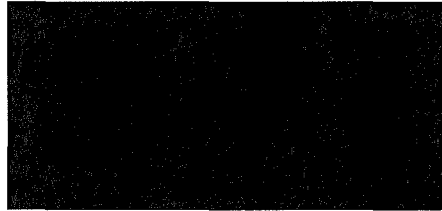
**Thí dụ 2:**  $N = 110010101100 \Rightarrow (N)_2 = 001101010100$  và  $(N)_1 = 001101010011$

Nhận xét:

- Để có số bù 2 của một số, bắt đầu từ bit LSB (tận cùng bên phải) đi ngược về bên trái, các bit sẽ giữ nguyên cho đến lúc gặp bit 1 đầu tiên, sau đó đảo tất cả các bit còn lại.

- Để có số bù 1 của một số, ta đảo tất cả các bit của số đó.

Từ các nhận xét trên ta có thể thực hiện một mạch tạo số bù 1 và 2 sau đây:



(H 6.1)

- Khi  $C=1$ ,  $B$  là số bù 1 của  $b$  ( $B_1$  và  $b_1$  là bit LSB)

- Khi  $C=0$ , B là số bù 2 của b.

Thật vậy, các biểu thức logic của B theo b và C là:

$$B_1 = b_1 \oplus C$$

$$B_2 = b_2 \oplus (C + b_1)$$

$$B_3 = b_3 \oplus (C + b_1 + b_2)$$

- Khi  $C=1$ , các ngõ ra cổng OR luôn bằng 1, các cổng EX - OR luôn có một ngõ vào bằng 1 nên ngõ ra là đảo của ngõ vào còn lại, ta được:

$$B_1 = b_1 \oplus 1 = \overline{b_1}$$

$$B_2 = b_2 \oplus (1 + b_1) = b_2 \oplus 1 = \overline{b_2}$$

$$B_3 = b_3 \oplus (1 + b_1 + b_2) = b_3 \oplus 1 = \overline{b_3}$$

- Khi  $C=0$

$$B_1 = b_1 \oplus 0 = b_1$$

$$B_2 = b_2 \oplus (0 + b_1) = b_2 \oplus b_1$$

$$= b_2 \text{ nếu } b_1=0 \text{ và } \overline{b_2} \text{ nếu } b_1 = 1$$

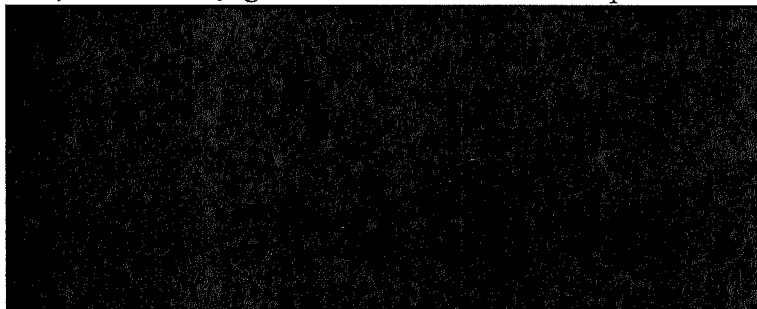
$$B_3 = b_3 \oplus (0 + b_1 + b_2) = b_3 \oplus (b_1 + b_2)$$

$$= b_3 \text{ nếu } b_1 \text{ và } b_2 \text{ đều } =0$$

$$= \overline{b_3} \text{ nếu } (b_1 \text{ và/hoặc } b_2 = 1)$$

Như vậy tất cả các bit sau bit 1 thứ nhất tính từ bit LSB đều bị đảo và B chính là số bù 2 của chúng ta cũng có thể thiết kế mạch tạo số bù hai bằng cách dùng FF RS, có ngõ vào b

R, S tác động mức cao, kết hợp với các cổng logic như (H 6.2). Mạch này dùng khá tiện lợi khi cần thực hiện bài toán cộng và trừ nhiều bit kiểu nối tiếp.



(H 6.2)

Bắt đầu, Preset mạch để ngõ ra  $Q = 1$ , cổng  $G_3$  đóng,  $G_2$  mở, cho số B đi qua mà không bị đảo cho đến khi có bit 1 đầu tiên đến, cổng  $G_1$  mở cho xung đồng hồ đi qua, FF RS được reset,  $Q = 0$ ,  $Q = 1$ ,  $G_2$  đóng,  $G_3$  mở, số B đi qua cổng  $G_2$  và bị đảo. Ở ngõ ra được số bù 2 của B.

Phép trừ số nhị phân dùng số bù 1:

Cho hai số dương A và B có n bit (nếu số bit khác nhau, ta thêm số 0 vào, mà không làm thay đổi trị, để cả hai có cùng số bit)

a/ -  $A \leq B$

Kết quả A-B là số 0 hoặc âm, phép tính được thực hiện như sau:

Tính A - B:

$$\begin{aligned}A - B &= A - B + 2^n - 1 - 2^n + 1 \\&= A + (2^n - B - 1) - 2^n + 1 \\&= A + (B)_1 - 2^n + 1 \\&= - \{2^n - [A + (B)_1] - 1\} \\&= - [A + (B)_1]_1\end{aligned}$$

Vậy A-B có được bằng cách cộng số bù 1 của B vào A rồi lấy bù 1 của tổng và thêm dấu trừ. Như vậy để thực hiện phép tính trừ ta chỉ cần dùng phép cộng và phép đảo

**Thí dụ 3 :** Tính 1001 - 11010 dùng số bù 1

Ta có  $A = 01001$  (thêm số 0 vào để có 5 bit như số B)

$$B = 11010 \Rightarrow (B)_1 = 00101$$

$$\begin{aligned}A - B &= - [A + (B)_1]_1 = - (01001 + 00101) = - (01110)_1 \\&= - (10001)\end{aligned}$$

Trong hệ thập phân, đây là bài toán  $9_{10} - 26_{10} = -17_{10}$

Để thấy dấu trừ được nhận ra như thế nào, ta viết lại phép toán:



Không có số tràn (hay số tràn =0) là dấu hiệu của kết quả âm (hoặc =0) và ta phải lấy bù 1, thêm dấu trừ để đọc kết quả cuối cùng:  $(01110)_1 = -10001$

**Thí dụ 4:** Tính 10110 - 10110

$$A = 10110 \text{ và } B = 10110 \Rightarrow (B)_1 = 01001$$



Trong phép cộng đầu tiên không có số tràn, kết quả xem như số âm (hoặc =0) lấy bù 1 của kết quả ta được A-B=00000.

b/ - A > B

Kết quả A-B là số dương, phép tính được thực hiện theo qui tắc sau:

**Cộng A với (B)<sub>1</sub> rồi thêm 1 và không quan tâm tới số nhớ cuối cùng**

**Thí dụ 5:** Tính 110101 - 100110 dùng số bù 1

$$A = 110101 \text{ và } B = 100110 \Rightarrow (B)_1 = 011001$$



Bỏ qua số nhớ cuối cùng, ta được kết quả A-B=001111.

Trong hệ thập phân đó là bài toán  $53_{10} - 38_{10} = 15_{10}$ .



Trong phép tính có số tràn chúng tỏ kết quả là số dương. Số 1 cộng thêm vào xem như lấy từ số nhớ đem qua.

Tóm lại, để thực hiện bài toán trừ,  $A-B$ , ta cộng  $A$  với bù 1 của  $B$ . Dựa vào sự có mặt hay không của số tràn mà có biện pháp xử lý kết quả:

- Nếu số tràn = 0, kết quả là số âm (hoặc = 0), ta phải lấy bù 1 của kết quả và thêm dấu - để đọc.

- Nếu số tràn = 1, ta cộng thêm 1 vào để có kết quả cuối cùng (bỏ qua bit tràn) là một số dương.

phép trừ số nhị phân dùng số bù 2:

Phép toán dùng số bù 1 có một bất tiện là ta phải thêm bài toán cộng 1 vào, để tránh việc này ta dùng phép toán với số bù 2

Cho hai số dương  $A$  và  $B$  có  $n$  bit

a/ -  $A < B$

Tính  $A-B$ :

$$\begin{aligned} A-B &= A-B+2^n-2^n \\ &= A+(2^n - B) - 2^n \\ &= A+(B)_2 - 2^n \\ &= - \{2^n - [A+(B)_2] \} \\ &= - [A+(B)_2]_2 \end{aligned}$$

Vậy  $A-B$  có được bằng cách cộng số bù 2 của  $B$  vào  $A$  rồi lấy bù 2 của tổng và thêm dấu trừ. Như vậy ta đã chuyển phép tính trừ thành phép cộng

**Thí dụ 6:** Tính  $1001 - 11010$  dùng số bù 2

Ta có  $A = 01001$  (thêm số 0 vào để có 5 bit như số  $B$ )

$$B = 11010 \Rightarrow (N_2)_2 = 00110$$

$$\begin{aligned} A-B &= - [A+(B)_2]_2 = - (01001+00110) = - (01111)_2 \\ &= - (10001) \end{aligned}$$

Ta được lại kết quả trên

Để thấy dấu trừ được nhận ra như thế nào, ta viết lại phép toán:



Không có số tràn là dấu hiệu của kết quả âm và ta phải lấy bù 2, thêm dấu trừ để đọc kết quả cuối cùng:  $(01111)_2 = -10001$

b/ -  $A \geq B$

Kết quả  $A-B$  là 0 hoặc số dương, phép tính được thực hiện theo qui tắc sau:

Cộng  $A$  với  $(B)_2$  và không quan tâm tới số nhớ ở vị trí  $2^n$

**Thí dụ 7:** Tính  $110101 - 100110$  dùng số bù 2

$$A = 110101 \text{ và } B = 100110 \Rightarrow (B)_2 = 011010$$



Có số tràn, kết quả là số dương. Bỏ qua số tràn và đọc ngay kết quả mà không phải biến đổi:  $001111 = 15_{10}$

**Thí dụ 8 :** Tính  $10110 - 10110$

$$A = 10110 \text{ và } B = 10110 \Rightarrow (B)_2 = 01010$$



Bỏ qua số tràn ta được  $A-B=00000$ .

### Phép toán với số có dấu

Cho tới giờ chúng ta thực hiện các phép toán với số không dấu và đôi khi xuất hiện dấu trừ trong kết quả. Trong máy tính, điều này có thể khắc phục được bằng cách dùng số có dấu.

Với qui ước số dương có bit dấu là 0 và số âm có dấu là 1

**Thí dụ 9:**  $+10_{10} = 01010$   $+15_{10} = 01111$   $+23_{10} = 010111$

$$-10_{10} = 10110 \quad -15_{10} = 10001 \quad -23_{10} = 101001$$

Có thể thấy rằng số âm của một số là số bù 2 của nó kể cả bit dấu.

Với cách biểu diễn số có dấu, phép toán trừ trở thành phép toán cộng:

$$A-B = A+(-B)$$

**Thí dụ 10:** Tính  $A-B=01110 - 01001$ ;  $B = 01001 = +9_{10} \Rightarrow -9_{10} = 10111$



Bit dấu =0 chỉ kết quả dương, bỏ bit tràn  $C'_2$ .

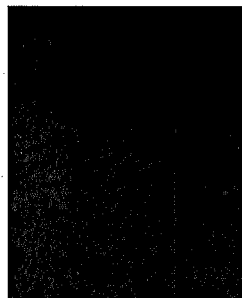
$$\text{Vậy } A-B = 00101 [(+14_{10})-(+9_{10})] = +5_{10}$$

Nếu A hoặc B đều dương hoặc âm, kết quả có thể cần thêm một bit do tràn số.

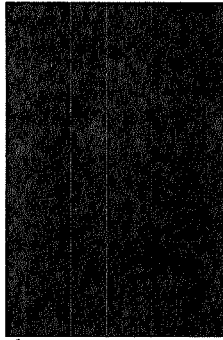
Trong trường hợp này bit tràn đầu tiên thuộc kết quả và  $C'_2$  là bit dấu

**Thí dụ 11:** Tính  $A+B$  với  $A = 01110 (+14_{10})$  và  $B = 01001 (+9_{10})$

Kết quả là  $010111 = +23_{10}$  với  $C'_2 = 0$  là bit dấu



**Thí dụ 12:** Tính A-B với A=10010 (-14<sub>10</sub>) và B=01001 (+9<sub>10</sub>)



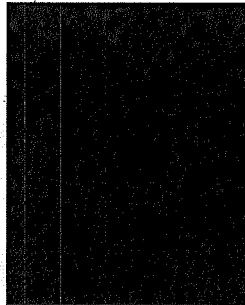
Một lần nữa C<sub>2</sub> chỉ bit dấu. Kết quả là 101001 = -23<sub>10</sub> (010111 = 23<sub>10</sub>)

Từ các kết quả trên, ta rút ra qui tắc sau đây:

Nếu C<sub>1</sub> = C<sub>2</sub> thì C<sub>2</sub> là bit tràn, bỏ đi và nếu C<sub>1</sub> ≠ C<sub>2</sub> thì C<sub>2</sub> là bit dấu.

**Thí dụ 13:** Tính A-B với A=011101 (+29<sub>10</sub>) và B=0110 (+6<sub>10</sub>)

B= 000110 = +6<sub>10</sub> ⇒ - 6<sub>10</sub> = 111010



**Ghi chú:** - Trong tất cả trường hợp, ta luôn luôn thực hiện phép cộng do đó có thể bỏ qua phép trừ

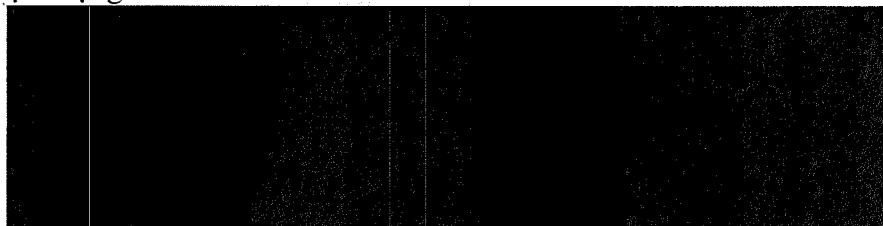
- Khi cộng hai số hạng cùng dấu thì có thể xảy ra hiện tượng tràn, lúc đó bit dấu dời về bên trái một bit. Trong các trường hợp khác thì dấu của kết quả ở cùng vị trí với dấu của các số hạng

- Ngoài ra kết quả còn được xử lý tùy vào kết quả so sánh sự khác nhau của hai số nhớ C<sub>1</sub> và C<sub>2</sub> (nhờ một công EX-OR).

Mạch cộng nhị phân:

**Mạch cộng bán phân (Half adder, HA):**

Là mạch cộng hai số 1 bit



Bảng sự thật      kết quả

Mạch  
(H 6.3)

Ký hiệu

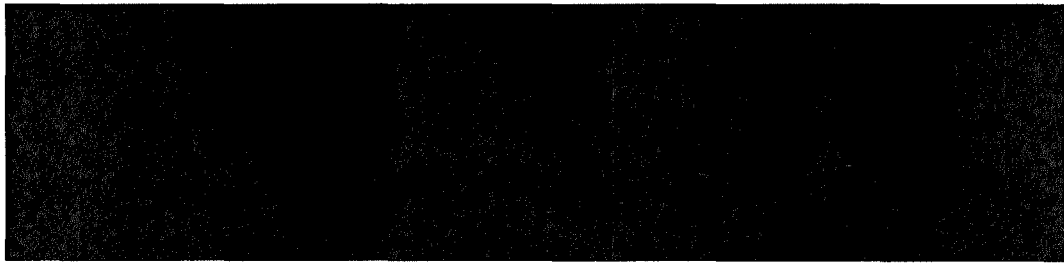
**Mạch cộng toàn phần (Full adder, FA) :**

Là mạch cộng hai bit ở cùng vị trí trong hai số nhị phân nhiều bit, nói cách khác, đây là mạch cộng hai bit, giả sử thứ n, và bit nhớ có được từ phép cộng hai bit thứ n-1 của hai số nhị phân đó. Ta có bảng sự thật

$C_{n-1}$	$B_n$	$A_n$	$S_n$	$C_n$
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

Dùng bảng Karnaugh ta xác định được  $S_n$  và  $C_n$  như sau:

$$S_n = C_{n-1} \oplus (A_n \oplus B_n)$$
$$C_n = A_n B_n + C_{n-1} (A_n \oplus B_n)$$



(H 6.4)

Có thể thấy một mạch cộng toàn phần gồm hai mạch cộng bán phần và một cổng OR

Cộng hai số nhị phân nhiều bit:

**Cộng nối tiếp**

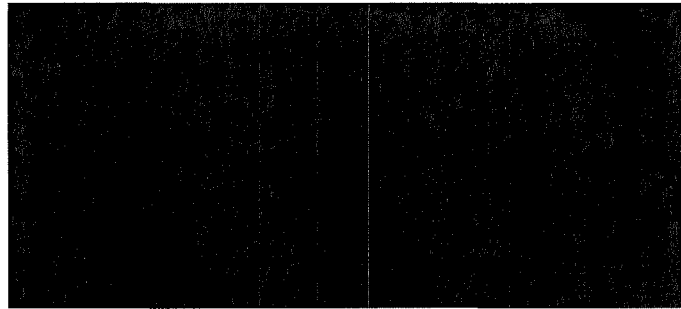
Trong cách cộng nối tiếp, người ta dùng các ghi dịch để chuyển các bit vào một mạch cộng toàn phần duy nhất, số nhớ từ ngõ ra  $C_n$  được làm trễ một bit nhờ FF D và đưa vào ngõ vào  $C_{n-1}$ . Như vậy tốc độ của phép cộng tùy thuộc vào tần số xung  $C_K$  và số bit phải thực hiện.



(H 6.5)

### **Cộng song song**

Trong cách cộng song song, các bit được đưa đồng thời vào các mạch cộng toàn phần và số nhớ của kết quả ở bit thấp được đưa lên bit cao hơn (H 6.6).



(H 6.6)

Chính vì phải chờ số nhớ mà tốc độ cộng còn hạn chế. Muốn nâng tốc độ cộng lên, người ta thực hiện phép cộng song song định trước số nhớ.

### **Mạch cộng song song định trước số nhớ**

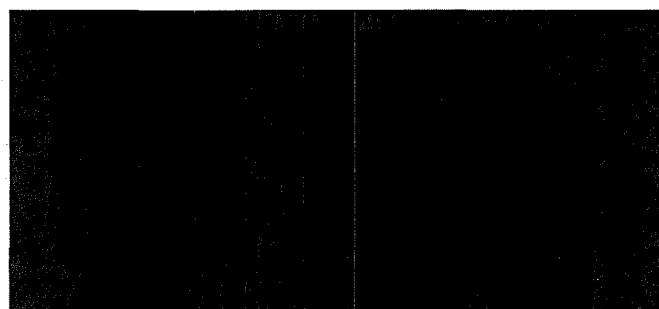
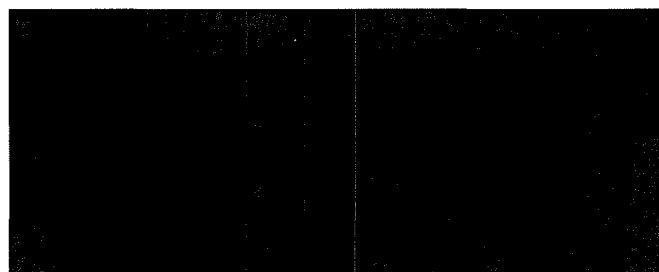
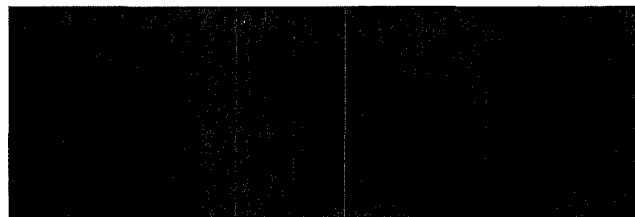
Để tăng tốc độ của mạch cộng song song, người ta tạo trước các số nhớ để đưa đồng thời vào mạch cộng

Từ biểu thức xác định số nhớ

$$C_n = A_n B_n + C_{n-1} (A_n \oplus B_n)$$

$$\text{Đặt } P_n = A_n B_n \quad \text{và} \quad G_n = A_n \oplus B_n$$

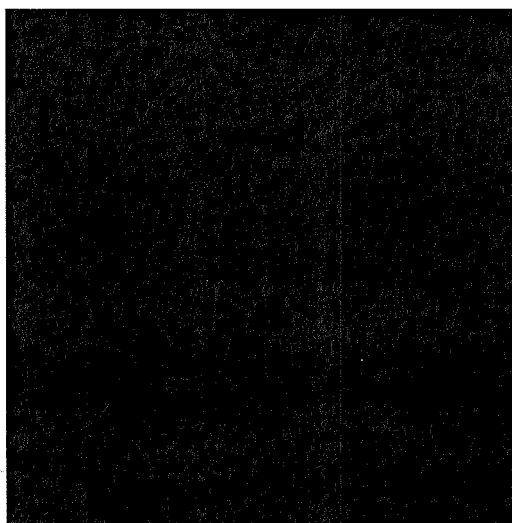
Ta xác định được  $C_1, C_2, C_3 \dots$  như sau:



(H 6.7)

Nhận thấy thời gian tính số nhớ giống nhau ở các tầng và bằng  $t_1+t_2$ .  $t_1$  là thời gian truyền đồng thời qua các cổng AND và  $t_2$  là thời gian truyền qua cổng OR.

Sơ đồ khối mạch cộng song song định trước số nhớ:



(H 6.8)

Trên thị trường hiện có IC 7483 (tương đương 4008 của CMOS) là IC cộng 4 bit theo kiểu định trước số nhớ.

### Cộng hai số BCD

Trên thị trường có các IC cộng số nhị phân, trong lúc trên thực tế nhiều khi chúng ta cần cộng các số BCD để cho kết quả là số BCD.

Chúng ta tìm cách dùng IC 7483 (4008) để cộng hai số BCD

Hai số BCD có trị từ  $0_{10}$  đến  $9_{10}$  khi cộng lại cho kết quả từ  $0_{10}$  đến  $18_{10}$ . Để đọc được kết quả dạng BCD ta phải hiệu chỉnh kết quả có được từ mạch cộng nhị phân.

Dưới đây là kết quả tương đương giữa 3 loại mã: thập phân, nhị phân và BCD

TP	Phân					B C D					BCD đọc theo NP
	S'=C' <sub>4</sub>	S' <sub>4</sub>	S' <sub>3</sub>	S' <sub>2</sub>	S' <sub>1</sub>	S=C <sub>4</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0	1	1
2	0	0	0	1	0	0	0	0	1	0	2
3	0	0	0	1	1	0	0	0	1	1	3
4	0	0	1	0	0	0	0	1	0	0	4
5	0	0	1	0	1	0	0	1	0	1	5
6	0	0	1	1	0	0	0	1	1	0	6
7	0	0	1	1	1	0	0	1	1	1	7
8	0	1	0	0	0	0	1	0	0	0	8
9	0	1	0	0	1	0	1	0	0	1	9
10	0	1	0	1	0	1	0	0	0	0	16
11	0	1	0	1	1	1	0	0	0	1	17
12	0	1	1	0	0	1	0	0	1	0	18
13	0	1	1	0	1	1	0	0	1	1	19
14	0	1	1	1	0	1	0	1	0	0	20
15	0	1	1	1	1	1	0	1	0	1	21
16	1	0	0	0	0	1	0	1	1	0	22
17	1	0	0	0	1	1	0	1	1	1	23

18	1	0	0	1	0	1	1	0	0	0	24
----	---	---	---	---	---	---	---	---	---	---	----

Nhận thấy:

- Khi kết quả <10 mã nhị phân và BCD hoàn toàn giống nhau
- Khi kết quả ≥10 để có được mã BCD ta phải cộng thêm 6 cho mã nhị phân

Để giải quyết vấn đề hiệu chỉnh này trước tiên ta sẽ thực hiện một mạch phát hiện kết quả trung gian của mạch cộng hai số nhị phân 4 bit. Mạch này nhận vào kết quả trung gian của phép cộng 2 số nhị phân 4 bit và cho ở ngõ ra  $Y = 1$  khi kết quả này ≥10, ngược lại,  $Y=0$ .

Bảng sự thật

TP	C'4	S'4	S'3	S'2	Y
0-1	0	0	0	0	0
2-3	0	0	0	1	0
4-5	0	0	1	0	0
6-7	0	0	1	1	0
8-9	0	1	0	0	0
10-11	0	1	0	1	1
12-13	0	1	1	0	1
14-15	0	1	1	1	1
16-17	1	0	0	0	1
18	1	0	0	1	1

Ta không dùng ngõ vào  $S'_1$  vì từng cặp trị có  $C'_4 S'_4 S'_3 S'_2$  giống nhau thì  $S'_1 = 0$  và  $S'_1 = 1$

Dùng bảng Karnaugh xác định được Y

$$Y = C'_4 + S'_4 (S'_3 + S'_2)$$

Và mạch cộng hai số BCD được thực hiện theo sơ đồ (H 6.9)



(H 6.9)

### Vận hành:

- IC thứ nhất cho kết quả trung gian của phép cộng hai số nhị phân.
- IC thứ hai dùng hiệu chỉnh để có kết quả là số BCD:
- Khi kết quả < 10, IC 2 nhận ở ngõ vào B số 0000 (do  $Y=0$ ) nên kết quả không thay đổi
- Khi kết quả trung gian ≥ 10, IC 2 nhận ở ngõ vào B số  $0110_2 = 6_{10}$  (do  $Y=1$ ) và kết quả được hiệu chỉnh như đã nói trên.

### Mạch cộng lưu số nhớ

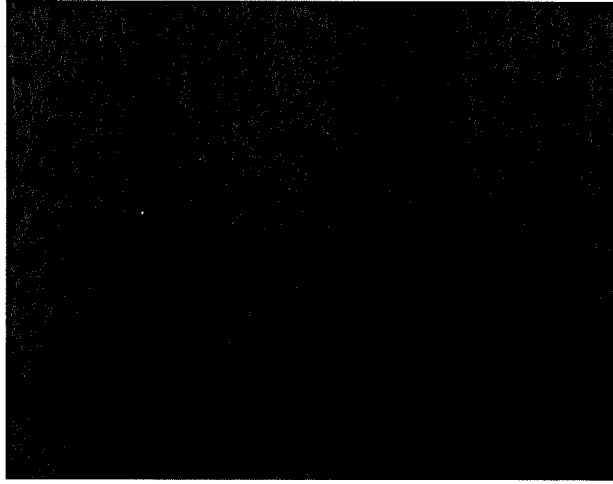
Nhắc lại, một mạch cộng toàn phần (FA) nhận 3-bit ở ngõ vào và cho 2 ngõ ra :

- Một là tổng của các bit có cùng trọng số với các bit ở ngõ vào
- Một là số nhớ có trọng số gấp đôi trọng số của các bit ở ngõ vào



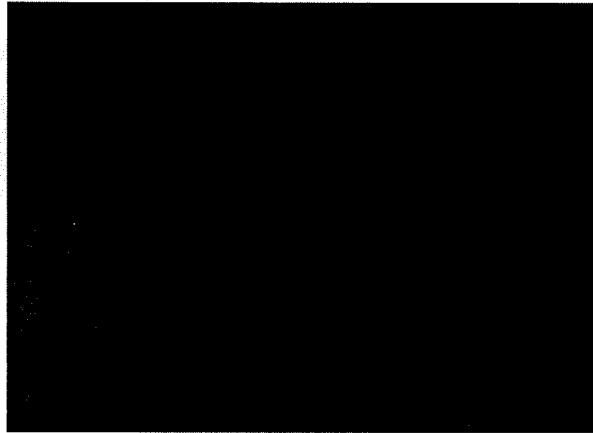
Để cộng một chuỗi số, nhiều mạch cộng toàn phần sẽ được sử dụng, số nhớ được lưu lại để đưa vào mạch cộng bit cao hơn.

**Thí dụ 14 :** Với 3 số 3-bit  $X (X_3X_2X_1)$ ,  $Y (Y_3Y_2Y_1)$ ,  $Z (Z_3Z_2Z_1)$  mạch cộng có dạng



(H 6.10)

Người ta dùng mạch cộng loại này để thực hiện bài toán nhân. Để có kết quả nhanh hơn, có thể dùng mạch (H 6.11)

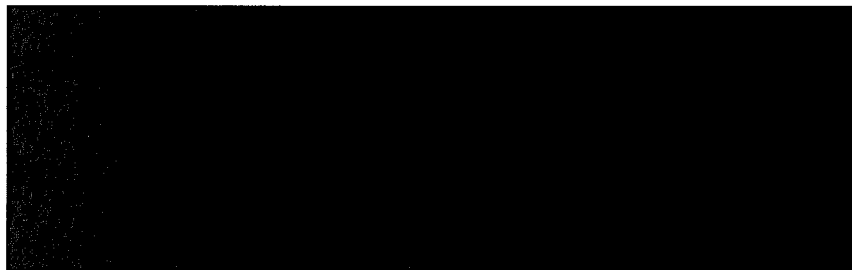


(H 6.11)

Mạch trừ nhị phân:

***Mạch trừ bán phần***

Là mạch trừ hai số 1 bit (H 6.12)



(H 6.12)

### Mạch trừ có số nhớ (mạch trừ toàn phần)

Là mạch trừ 2 bit có quan tâm tới số nhớ mang từ bit trước

$R_{n-1}$	$A_n$	$B_n$	$D_n$	$R_n$
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

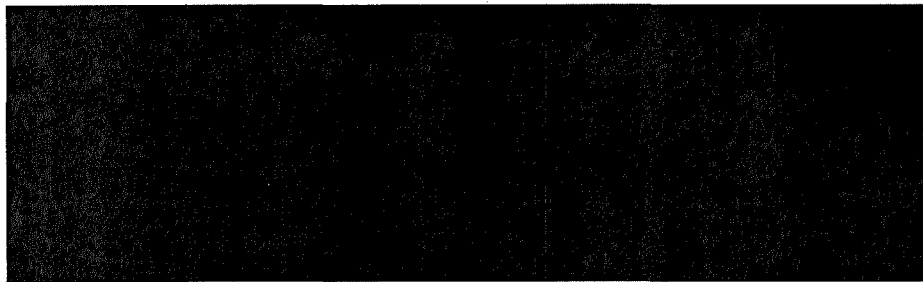
Bảng sự thật

Dùng bảng Karnaugh xác định được các hàm  $D_n$  và  $R_n$

$$D_n = R_{n-1} \oplus (A_n \oplus B_n)$$

$$R_n = \overline{A_n}B_n + R_{n-1}(A_n \oplus B_n)$$

Và mạch (H 6.13)



(H 6.13)

Nhận thấy cấu tạo mạch trừ giống như mạch cộng, chỉ khác ở mạch tạo số nhớ

### Trừ số nhiều bit

Ta có mạch trừ số nhiều bit bằng cách mắc song song các mạch trừ 1 bit (H 6.14)

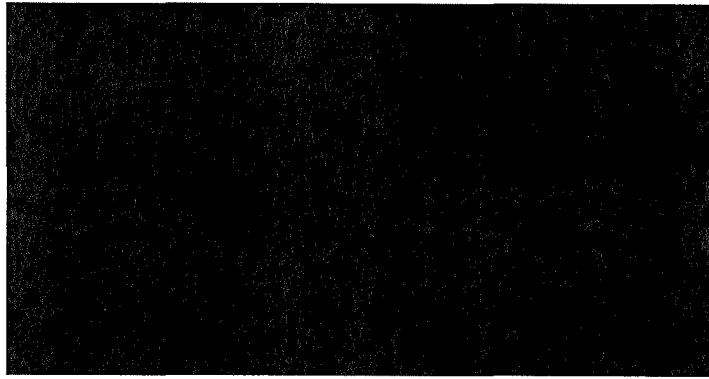


(H 6.14)

### Cộng và trừ số nhiều bit trong một mạch

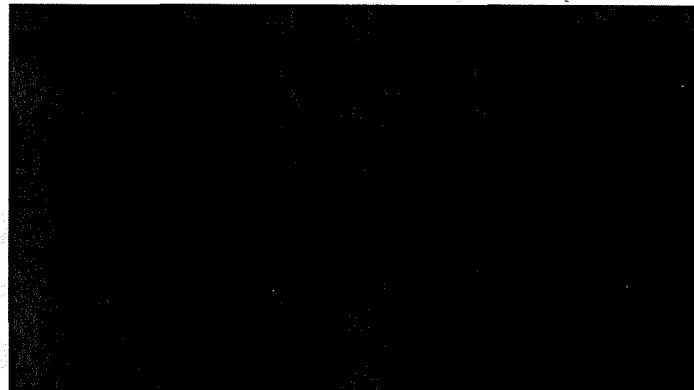
Nhắc lại để thực hiện phép toán trừ, người ta cộng với số bù 1 và cộng thêm 1 (hoặc cộng với số bù 2), như vậy để thực hiện phép trừ  $A - B$  ta tính  $A + (B)_1 + 1$ . Mạch (H 6.6) được sửa đổi để có thực hiện phép cộng và trừ tùy vào ngã điều khiển C (H 6.15)

- Khi  $C=0$ , ta có mạch cộng
- Khi  $C=1$ , ta có mạch trừ



(H 6.15)

Ta cũng có thể thực hiện mạch cộng trừ theo kiểu mắc nối tiếp (H 6.16)



(H 6.16)

Nếu hai số A, B là số 8 bit, có dấu, kết quả được xử lý bởi mạch dò số tràn, thiết kế dựa vào biểu thức:  $OV = C_7 \oplus C_8$ . Khi  $OV = 1$  nghĩa là có số tràn (tức  $C_7 \neq C_8$ ), thì số tràn  $C_8$  sẽ là bit dấu,  $S_8$  là một bit của kết quả và khi  $OV = 0$  (tức  $C_7 = C_8$ ), thì  $S_8$  là bit dấu.

### Mạch nhân

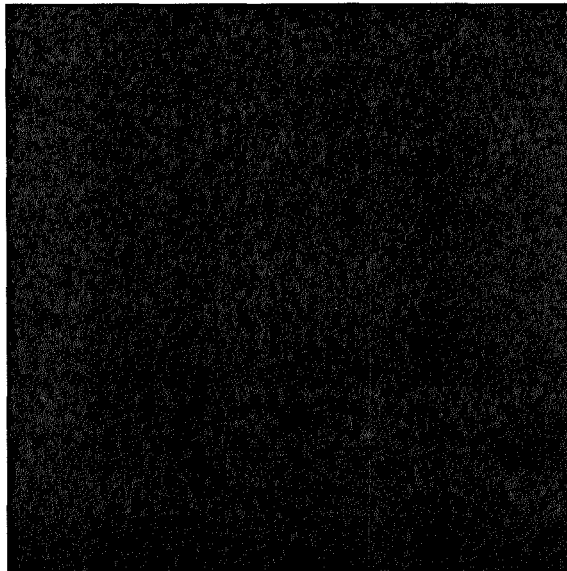
Lấy thí dụ bài toán nhân 2 số 4 bit

			$Y_4$	$Y_3$	$Y_2$	$Y_1$	Số bị nhân
			$X_4$	$X_3$	$X_2$	$X_1$	Số nhân
			$P_{14}$	$P_{13}$	$P_{12}$	$P_{11}$	Tích từng phần
		$P_{24}$	$P_{23}$	$P_{22}$	$P_{21}$		
	$P_{34}$	$P_{33}$	$P_{32}$	$P_{31}$			
$P_{44}$	$P_{43}$	$P_{42}$	$P_{41}$				$S_8$
$S_7$	$S_6$	$S_5$	$S_4$	$S_3$	$S_2$	$S_1$	Kết quả

### Mạch nhân cơ bản

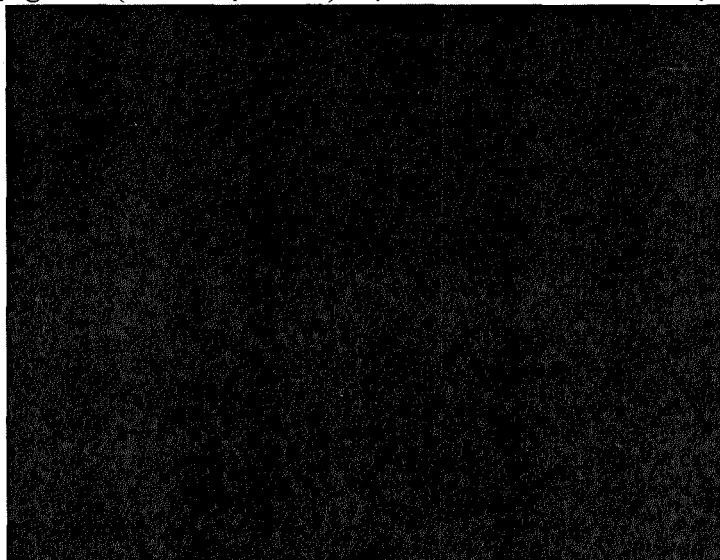
Việc thực hiện bài toán nhân có thể xem như gồm hai bước:

- Tính các tích từng phần: thực hiện bởi các công AND
- Tính tổng của các tích từng phần: Áp dụng bài toán tổng chuỗi số (H 6.17)



(H 6.17)

Dùng IC cộng 4 bit (7483 hoặc 4008) mạch nhân hai số 4 bit có dạng (H 6.18)



(H 6.18)

*Mạch nhân nối tiếp - song song đơn giản (H 6.19)*



(H 6.19)

Trong mạch này, một trong hai số được đưa nối tiếp vào mạch (trong trường hợp này là số bị nhân) và số còn lại đưa song song vào mạch.

- Số nhân ( $b_4b_3b_2b_1$ ) đưa song song vào mạch qua các cổng AND đồng thời kiểm soát các cổng này: ứng với bit 1 số bị nhân qua mạch để tới mạch cộng (cổng 2 và 4); ứng với bit 0 ngã ra cổng AND bằng không (cổng 1 và 3)

- Số bị nhân đưa nối tiếp vào mạch theo thứ tự từ bit LSB. Các FF D có tác dụng dịch kết quả của phép nhân (là các tích từng phần) trước khi đưa vào mạch cộng để cộng các tích từng phần này.

**Thí dụ 15 :** Xem bài toán nhân  $10 \times 14$ . Số nhân là  $1010 (10_{10})$  và số bị nhân là  $1110 (14_{10})$ . Quá trình nhân giải thích như sau:

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

$$10001100_2 = 140_{10}$$

Có thể thấy rằng ngõ ra A luôn luôn bằng 0 vì bit LSB của số nhân = 0. Ngõ ra B có giá trị của số bị nhân được làm trễ 1 bit (1 xung đồng hồ). Ngõ ra C được làm trễ 2 bit và luôn bằng 0 (Giống như A). Ngõ ra D giống như B nhưng trễ 3 bit. Điều này có thể so sánh với bài toán trên giấy

Số bị nhân					1	1	1	0
Số nhân					<u>1</u>	<u>0</u>	<u>1</u>	<u>0</u>
A					0	0	0	0
B						1	1	0
C				0	0	0	0	0
D			1	1	1	0	0	0
Tích	1	0	0	0	1	1	0	0

Muốn không sử dụng mạch cộng số nhiều bit, người ta dùng mạch (H 6.20)



(H 6.20)

Mạch (H 6.20) cần (n-1) mạch cộng và mạch trễ (FF D) cho số nhân n bit. Các cổng AND cho phép các bit của số bị nhân đi qua khi số nhân là 1, số bị nhân (với số bit bất kỳ) được cho vào mạch nối tiếp với bit LSB vào đầu tiên.

Ngõ ra cổng 4 sau 4 xung Clock là 1110. Ngõ ra cổng 3 luôn luôn bằng 0.

Mạch cộng A cộng số ngõ ra 3 và ngõ ra 4 bị trễ 1 bit:

	0	0	0	0
	1	1	1	0
	<u>1</u>	<u>1</u>	<u>1</u>	<u>0</u>
	1	1	1	0

Tương tự mạch cộng B cộng số bị nhân với kết quả ở A được làm trễ 1 bit.

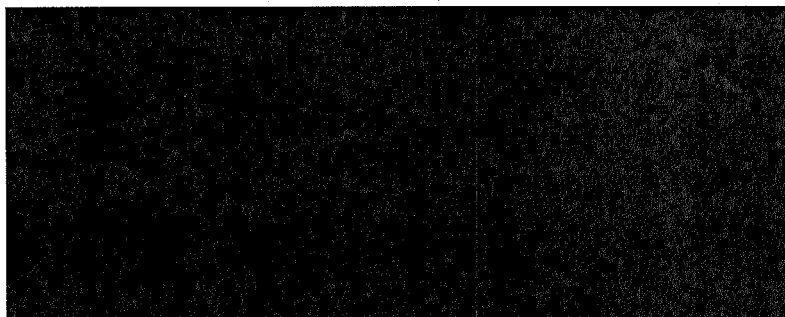
				1	1	1	0
		1	1	1	0	0	0
	<u>1</u>	<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>
	1	0	0	0	1	1	0

và mạch cộng C

				0	0	0	0
1	0	0	0	1	1	0	0
<hr/>							
1	0	0	0	1	1	0	0

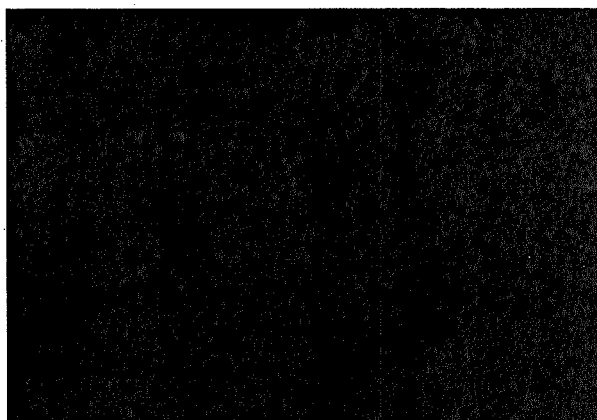
Lưu ý là ở mạch (H 6.20) kết quả cho ở ngõ ra mạch cộng C với bit LSB ra đầu tiên, tuy nhiên mạch này chưa quan tâm tới số nhớ.

Mạch (H 6.21) cho kết quả với số nhớ.



(H 6.21)

Và (H 6.22) là một mạch thực tế dùng ghi dịch 4 bit có ngõ vào/ra song song, một mạch cộng 4 bit và một chip 4 cổng AND 2 ngõ vào để thực hiện bài toán nhân.

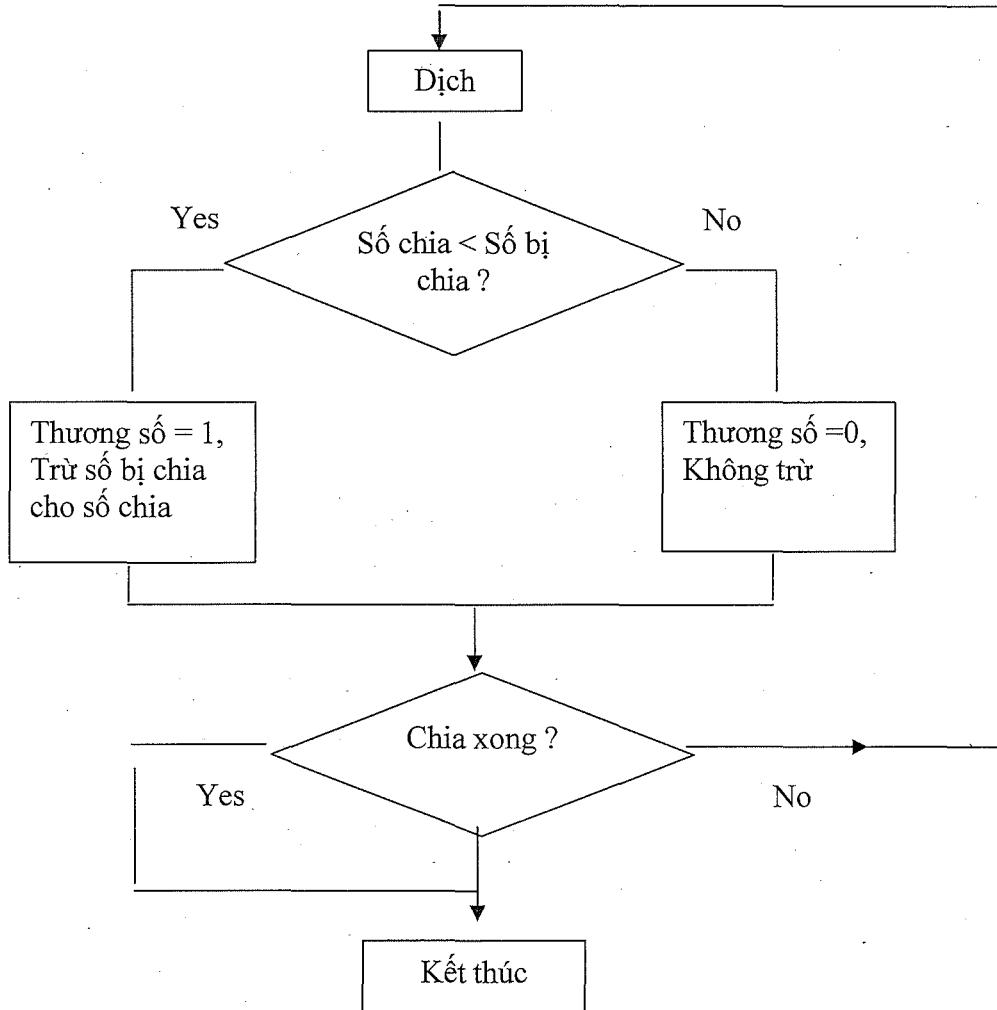


(H 6.22)

### Mạch chia

Nguyên tắc của phép chia số nhị phân là thực hiện phép so sánh một phần của số bị chia (số bit đầu tiên bằng với số bit của số chia) với số chia, nếu số bị chia lớn hơn số chia thì thương số =1, thực hiện phép trừ, ngược lại thì thương số =0, sau đó dịch trái phần còn lại của số bị chia một bit (hoặc dịch phải số chia 1 bit) rồi tiếp tục thực hiện bài toán so sánh giống như trên. Công việc được lặp lại cho đến khi chấm dứt.

Sơ đồ (H 6.23) tóm tắt giải thuật thực hiện bài toán chia

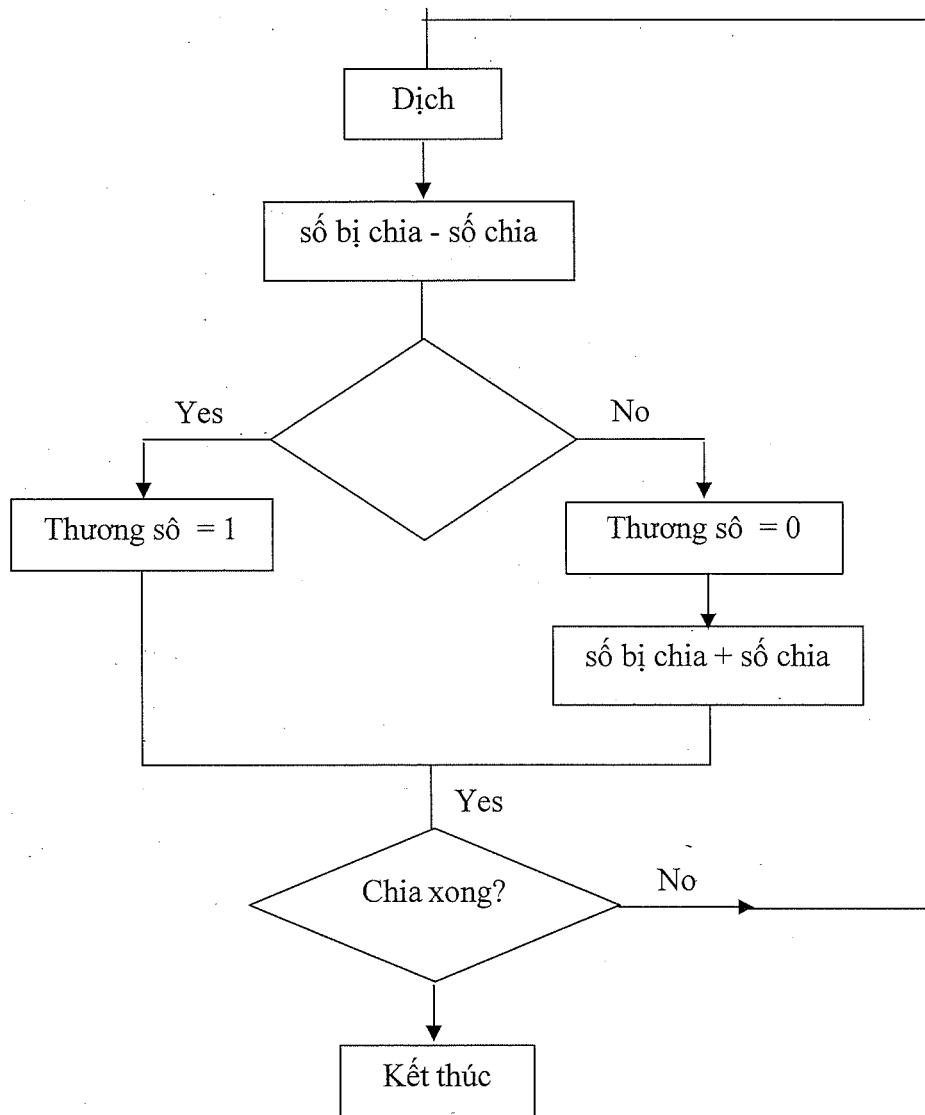


(H 6.23)

### ***Phép chia có phức hồi số bị chia***

Thay vì phải thực hiện việc so sánh, người ta làm phép tính trừ một phần số bị chia cho số chia, nếu kết quả dương, thương số là 1, nếu kết quả âm, thương số là 0, trong trường hợp này phải phức hồi lại số bị chia bằng cách cộng số bị chia cho số chia trước khi dịch số bị chia sang trái 1 bit (hoặc số chia sang phải) để tiếp tục lặp lại bài toán cho đến khi kết thúc.

(H 6.24) là sơ đồ giải thuật thực hiện phép chia có phức hồi số bị chia.



(H 6.24)

Để thực hiện phép chia theo sơ đồ trên, ngoài các thanh ghi để chứa các số bị chia, số chia, số thương người ta phải dùng thanh ghi chứa số bị chia được phục hồi.

### ***Phép chia không phục hồi số bị chia***

Hệ thống sẽ đơn giản hơn nếu chúng ta dùng phép chia không cần phục hồi số bị chia theo nguyên tắc như dưới đây.

Quan sát giản đồ (H 6.24) ta thấy có 2 trường hợp:



♦ **Số chia lớn hơn số bị chia** (nhánh bên phải)

Lưu ý là dịch số chia về bên phải 1 bit tương đương với chia số đó cho 2

Nhánh bên phải của sơ đồ trên gồm 2 bài toán:

- Cộng số bị chia với số chia.

- Trừ số bị chia cho 1/2 số chia (trừ bị chia cho số chia đã dịch phải)

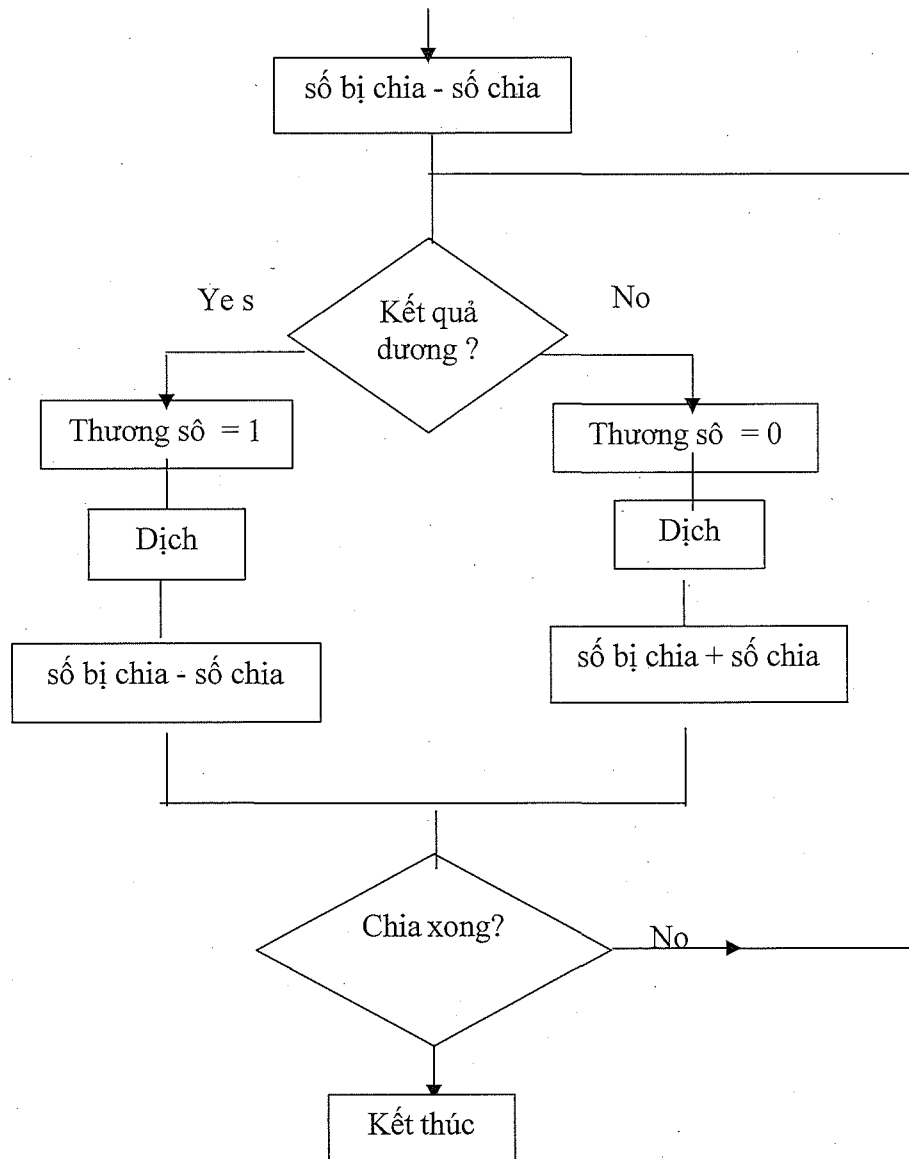
Hai bước này có thể gom lại thành một bước duy nhất như sau:

- Cộng số bị chia với số chia đã dịch phải.

♦ **Số chia nhỏ hơn số bị chia** (nhánh bên trái)

Sau khi lấy kết quả =1, lệnh kế tiếp thực hiện là trừ số chia đã dịch phải.

Từ các kết quả nhận xét trên có thể thay sơ đồ (H 6.24) bởi sơ đồ giải thuật thực hiện phép chia không cần phục hồi số bị chia (H 6.25)



(H 6.25)

Dựa vào sơ đồ (H 6.25), các bước thực hiện bài toán chia như sau:

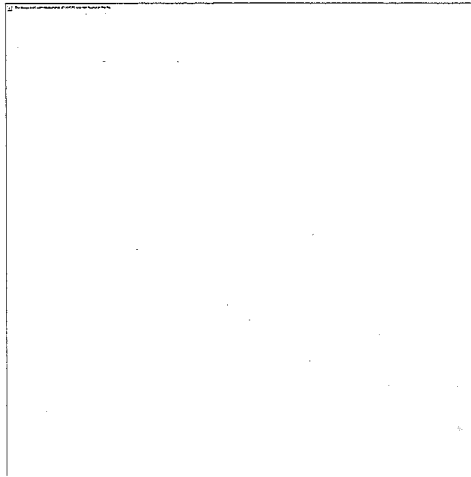
- Số chia (SC) lớn hơn số bị chia (SBC) ( $SBC - SC < 0$ ), thương số là 0, dịch phải số chia 1 bit (thực tế ta mang thêm 1 bit của số bị chia xuống), thực hiện bài toán cộng số chia và số bị chia

- Số chia nhỏ hơn số bị chia ( $SBC - SC > 0$ ), thương số là 1, dịch phải số chia 1 bit, thực hiện bài toán trừ (cộng số bù 2) số bị chia cho số chia

Để đơn giản, giả sử số chia và bị chia đều dương ( $MSB = 0$ ), số bị chia gồm 6 bit và số chia gồm 4 bit.

**Thí dụ 1:** Thực hiện bài toán chia  $21_{10} = 010101_2$  cho  $7_{10} = 0111_2$ .

Số bù 2 của  $0111$  là  $(0111)_2 = 1001$



**Ghi chú:**

(1) Số 1 trên mũi tên chỉ rằng kết quả phép toán trừ là số âm, bước kế tiếp là dời và cộng số chia

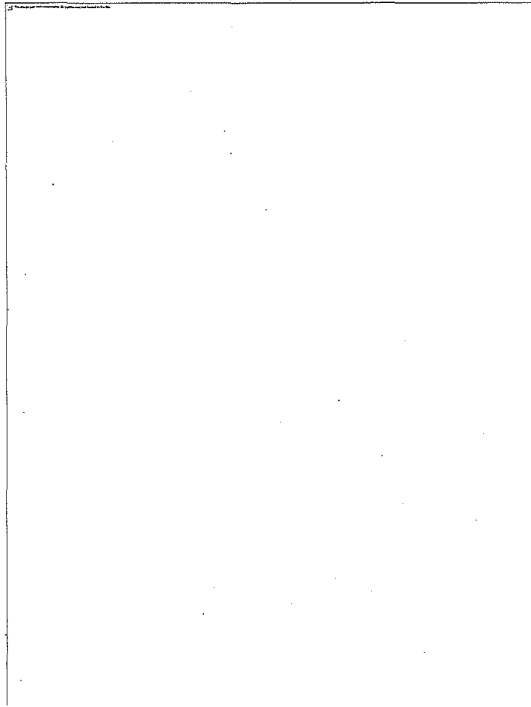
(2) Số 0 trên mũi tên chỉ rằng kết quả phép toán trừ là số dương, bước kế tiếp là dời và trừ số chia (cộng số bù 2)

Thương số có được từ các số trên mà trên phép tính ta ghi trong vòng tròn.

Kết quả: thương là  $011(=3)$  và số dư là  $0000(=0)$

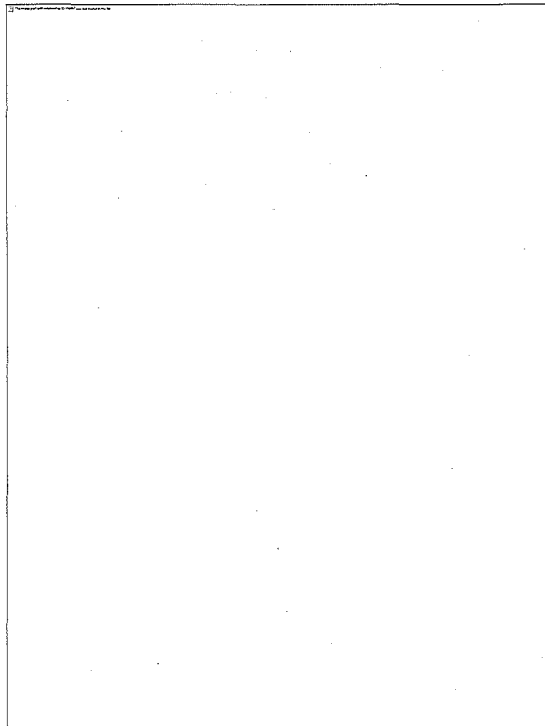
Bài toán trên cho kết quả với 3 bước cộng/trừ. Tuy nhiên nếu ta chia 21 cho 1 thì cần tới 6 bước cộng trừ để có thương số 6 bit. Một cách tổng quát số bước của bài toán bằng với số bit của số bị chia.

Ta có thể làm lại bài toán với 6 bước cộng/trừ ((thêm 3 bit 0 cho số bị chia)

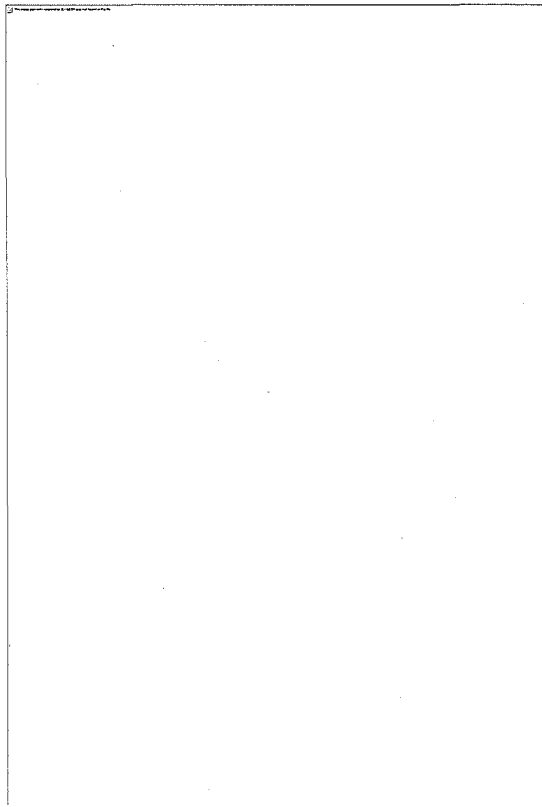


Thí dụ 2 và 3 dưới đây là bài toán 6 bước

**Thí dụ 2 :** Chia 21 cho 6 được kết quả 3 và số dư là 3



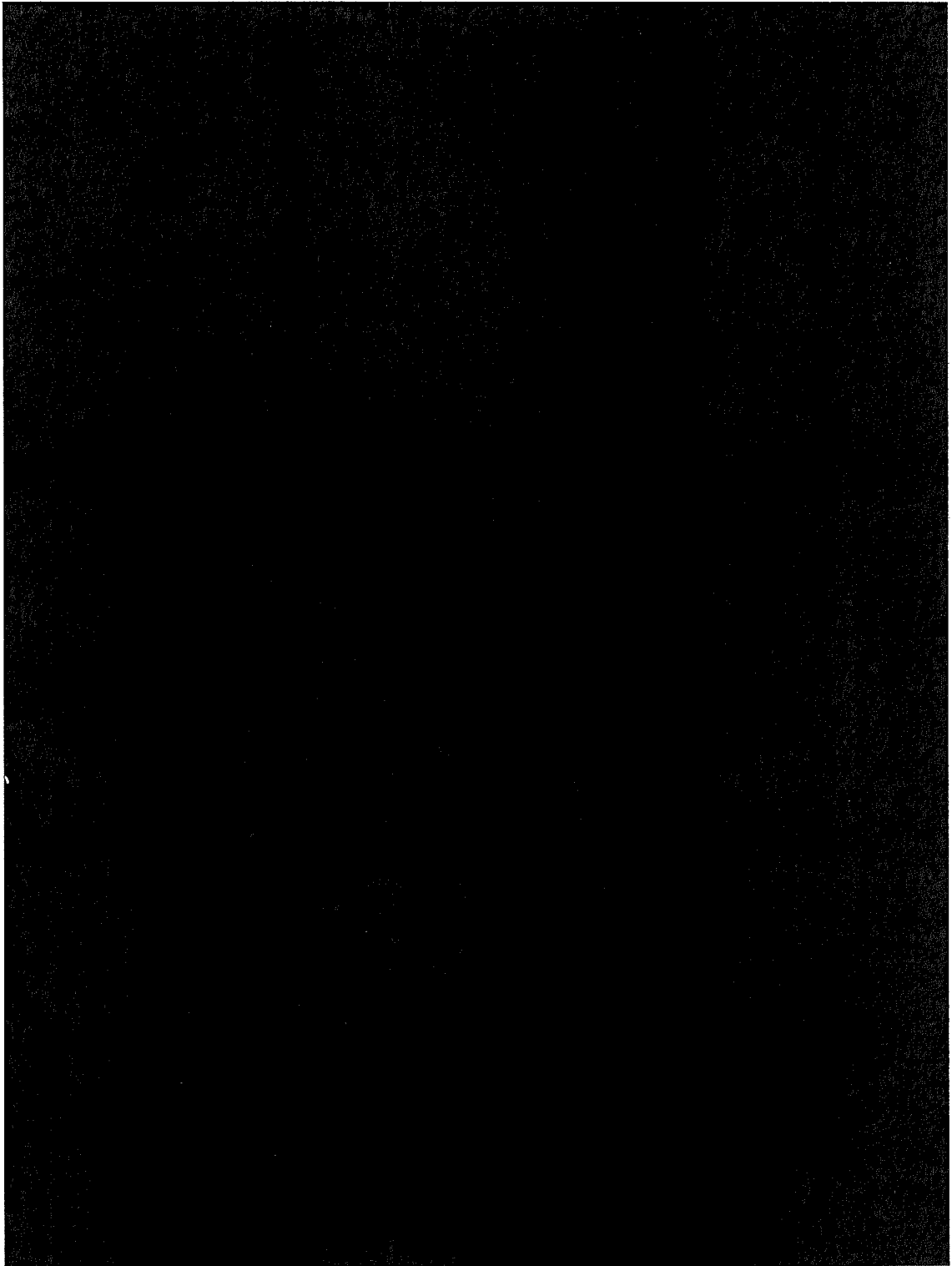
**Thí dụ 3 :** Chia 21 cho 5, được kết quả 4 và số dư là 1. Tuy nhiên trên phép toán ta thấy phép cộng với số chia cuối cùng cho kết quả âm (số 1100) nên để điều chỉnh số dư ta phải cộng số chia vào và bỏ qua số tràn.



(1) Cộng số chia vào để điều chỉnh số dư

Mạch thực hiện các bài toán này cho ở (H 6.26).

Trong (H 6.26) bước đầu tiên được thực hiện bởi các cổng EX-OR trên cùng có ngõ điều khiển = 1 để thực hiện bài toán trừ. Sau bước thứ nhất, bit thứ tư của mạch cộng ( $S_4$ ) sẽ quyết định phép toán sau đó là cộng ( $S_4=1$ ) hay trừ ( $S_4=0$ ) số bị chia với số chia. Số nhớ của bài toán cuối cùng (bước 6) là bit LSB của thương số. Và mạch cộng cuối cùng được thiết kế kết hợp với các cổng AND để xử lý kết quả của số dư như trong hai thí dụ 2 và 3. Nếu kết quả của bài toán ở bước 6 có  $S_4 = 1$  thì cổng AND mở để thực hiện bài toán cộng với số chia để điều chỉnh số dư.



(H 6.26)

## ● CHƯƠNG 7: KỸ THUẬT ADC - DAC

Tính ưu việt chủ yếu của các hệ thống số so với hệ thống tương tự là khả năng lưu trữ một lượng lớn thông tin số và dữ liệu trong những khoảng thời gian nhất định. Khả năng nhớ này là điều làm cho hệ thống số trở thành đa năng và có thể thích hợp với nhiều tình huống. Thí dụ trong một máy tính số, bộ nhớ trong chứa những lệnh mà theo đó máy tính có thể hoàn tất công việc của mình với sự tham gia ít nhất của con người.

Bộ nhớ bán dẫn được sử dụng làm **bộ nhớ chính** trong các máy tính nhờ vào khả năng thỏa mãn tốc độ truy xuất dữ liệu của bộ xử lý trung tâm (CPU).

Chúng ta đã quá quen thuộc với Fliflop, một linh kiện điện tử có tính nhớ. Chúng ta cũng đã thấy một nhóm các FF hợp thành thanh ghi để lưu trữ và dịch chuyển thông tin như thế nào. Các FF chính là các phần tử nhớ tốc độ cao được dùng rất nhiều trong việc điều hành bên trong máy tính, nơi mà dữ liệu dịch chuyển liên tục từ nơi này đến nơi khác.

Tiến bộ trong công nghệ chế tạo LSI và VLSI cho phép kết hợp một lượng lớn FF trong một chip tạo thành các bộ nhớ với các dạng khác nhau. Những bộ nhớ bán dẫn với công nghệ chế tạo transistor lưỡng cực (BJT) và MOS là những bộ nhớ nhanh nhất và giá thành của nó liên tục giảm khi các công nghệ LSI và VLSI ngày càng được cải tiến.

Dữ liệu số cũng có thể được lưu trữ dưới dạng điện tích của tụ điện, và một loại phần tử nhớ bán dẫn rất quan trọng đã dùng nguyên tắc này để lưu trữ dữ liệu với mật độ cao nhưng tiêu thụ một nguồn điện năng rất thấp.

Bộ nhớ bán dẫn được dùng như là **bộ nhớ trong** chính của máy tính, nơi mà việc vận hành nhanh được xem như ưu tiên hàng đầu và cũng là nơi mà tất cả dữ liệu của chương trình lưu chuyển liên tục trong quá trình thực hiện một tác vụ do CPU yêu cầu.

Mặc dù bộ nhớ bán dẫn có tốc độ làm việc cao, rất phù hợp cho bộ nhớ trong, nhưng giá thành tính trên mỗi bit lưu trữ cao khiến cho nó không thể là loại thiết bị có tính chất lưu trữ khối (mass storage), là loại thiết bị có khả năng lưu trữ hàng tỉ bit mà không cần cung cấp năng lượng và được dùng như là **bộ nhớ ngoài** (đĩa từ, băng từ, CD ROM . . .). Tốc độ xử lý dữ liệu ở bộ nhớ ngoài tương đối chậm nên khi máy tính làm việc thì dữ liệu từ bộ nhớ ngoài được chuyển vào bộ nhớ trong.

Băng từ và đĩa từ là các thiết bị lưu trữ khối mà giá thành tính trên mỗi bit tương đối thấp. Một loại bộ nhớ khối mới hơn là bộ nhớ **bọt từ (magnetic bubble memory, MBM)** là bộ nhớ điện tử dựa trên nguyên tắc từ có khả năng lưu trữ hàng triệu bit trong một chip. Với tốc độ tương đối chậm nó không được dùng như bộ nhớ trong.

Chương này nghiên cứu cấu tạo và tổ chức của các **bộ nhớ bán dẫn**.

## Thuật ngữ liên quan đến bộ nhớ

Để tìm hiểu cấu tạo, hoạt động của bộ nhớ chúng ta bắt đầu với một số thuật ngữ liên quan đến bộ nhớ

- **Tế bào nhớ**: là linh kiện hay một mạch điện tử dùng để lưu trữ một bit đơn (0 hay 1). Thí dụ của một tế bào nhớ bao gồm: mạch FF, tụ được tích điện, một điểm trên băng từ hay đĩa từ. . . .

- **Từ nhớ** : là một nhóm các bit (tế bào) trong bộ nhớ dùng biểu diễn các lệnh hay dữ liệu dưới dạng một số nhị phân. Thí dụ một thanh ghi 8 FF là một phần tử nhớ lưu trữ từ 8 bit. Kích thước của từ nhớ trong các máy tính hiện đại có chiều dài từ 4 đến 64 bit.

- **Byte** : từ 8 bit, đây là kích thước thường dùng của từ nhớ trong các máy vi tính.

- **Dung lượng** : chỉ số lượng bit có thể lưu trữ trong bộ nhớ. Thí dụ bộ nhớ có khả năng lưu trữ 4.096 từ nhớ 20 bit, dung lượng của nó là  $4096 \times 20$ , mỗi 1024 ( $=2^{10}$ ) từ nhớ được gọi là "1K", như vậy  $4096 \times 20 = 4K \times 20$ . Với dung lượng lớn hơn ta dùng "1M" hay 1meg để chỉ  $2^{20} = 1.048.576$  từ nhớ.

- **Địa chỉ** : là số nhị phân dùng xác định vị trí của từ nhớ trong bộ nhớ. Mỗi từ nhớ được lưu trong bộ nhớ tại một địa chỉ duy nhất. Địa chỉ luôn luôn được biểu diễn bởi số nhị phân, tuy nhiên để thuận tiện người ta có thể dùng số hex hay thập phân, bát phân

- **Tác vụ đọc** : (*Read*, còn gọi là *fetch*), một từ nhớ tại một vị trí nào đó trong bộ nhớ được truy xuất và chuyển sang một thiết bị khác.

- **Tác vụ viết** : (*ghi*, *Write*, còn gọi là *store*), một từ mới được đặt vào một vị trí trong bộ nhớ, khi một từ mới được viết vào thì từ cũ mất đi.

- **Thời gian truy xuất** (*access time*) : số đo tốc độ hoạt động của bộ nhớ, ký hiệu  $t_{ACC}$ . Đó là thời gian cần để hoàn tất một tác vụ đọc. Chính xác đó là thời gian từ khi bộ nhớ nhận một địa chỉ mới cho tới lúc dữ liệu khả dụng ở ngõ ra bộ nhớ

- **Bộ nhớ không vĩnh cửu** (*volatile*) : Bộ nhớ cần nguồn điện để lưu trữ thông tin. Khi ngắt điện, thông tin lưu trữ bị mất. Hầu hết bộ nhớ bán dẫn là loại không vĩnh cửu, trong khi bộ nhớ từ là loại vĩnh cửu (*nonvolatile*).

- **Bộ nhớ truy xuất ngẫu nhiên** (*Random-Access Memory, RAM*) : Khi cần truy xuất một địa chỉ ta tới ngay địa chỉ đó. Vậy thời gian đọc hay viết dữ liệu vào các vị trí nhớ khác nhau trong bộ nhớ không tùy thuộc vào vị trí nhớ. Nói cách khác, thời gian truy xuất như nhau đối với mọi vị trí nhớ. Hầu hết bộ nhớ bán dẫn và **nhân từ** (bộ nhớ trong của máy tính trước khi bộ nhớ bán dẫn ra đời) là loại truy xuất ngẫu nhiên.

- **Bộ nhớ truy xuất tuần tự** (*Sequential-Access Memory, SAM*) : Khi cần truy xuất một địa chỉ ta phải lướt qua các địa chỉ trước nó. Như vậy thời gian đọc và viết dữ liệu ở những vị trí khác nhau thì khác nhau. Những thí dụ của bộ nhớ này là băng từ, đĩa từ. Tốc độ làm việc của loại bộ nhớ này thường chậm so với bộ nhớ truy xuất ngẫu nhiên.

- **Bộ nhớ đọc/viết** (*Read/Write Memory, RWM*) : Bộ nhớ có thể viết vào và đọc ra.

- **Bộ nhớ chỉ đọc** (*Read-Only Memory, ROM*): là bộ nhớ mà tỉ lệ tác vụ đọc trên tác vụ ghi rất lớn. Về mặt kỹ thuật, một ROM có thể được ghi chỉ một lần ở nơi sản xuất và sau đó thông tin chỉ có thể được đọc ra từ bộ nhớ. Có loại ROM có thể được ghi nhiều lần nhưng tác vụ ghi khá phức tạp hơn là tác vụ đọc. ROM thuộc loại bộ nhớ vĩnh cửu và dữ liệu được lưu giữ khi đã cắt nguồn điện.

- **Bộ nhớ tĩnh** (*Static Memory Devices*) : là bộ nhớ bán dẫn trong đó dữ liệu đã lưu trữ được duy trì cho đến khi nào còn nguồn nuôi.

- **Bộ nhớ động** (*Dynamic Memory Devices*) : là bộ nhớ bán dẫn trong đó dữ liệu đã lưu trữ muốn tồn tại phải được **ghi lại** theo chu kỳ. Tác vụ ghi lại được gọi là **làm tươi** (*refresh*).

- **Bộ nhớ trong** (*Internal Memory*) : Chỉ bộ nhớ chính của máy tính. Nó lưu trữ các lệnh và dữ liệu mà CPU dùng thường xuyên khi hoạt động.

- **Bộ nhớ khối** (Mass Memory): Còn gọi là bộ nhớ phụ, nó chứa một lượng thông tin rất lớn ở bên ngoài máy tính. Tốc độ truy xuất trên bộ nhớ này thường chậm và nó thuộc loại vĩnh cửu.

### Đại cương về vận hành của bộ nhớ

#### *Các tác vụ và các nhóm chân của một IC nhớ*

Mặc dù mỗi loại bộ nhớ có hoạt động bên trong khác nhau, nhưng chúng có chung một số nguyên tắc vận hành mà chúng ta có thể tìm hiểu sơ lược trước khi đi vào nghiên cứu từng loại bộ nhớ.

Mỗi hệ thống nhớ luôn có một số yêu cầu ở các ngõ vào và ra để hoàn thành một số tác vụ:

- Chọn địa chỉ trong bộ nhớ để truy xuất (đọc hoặc viết)
- Chọn tác vụ đọc hoặc viết để thực hiện
- Cung cấp dữ liệu để lưu vào bộ nhớ trong tác vụ viết
- Gửi dữ liệu ra từ bộ nhớ trong tác vụ đọc
- Cho phép (Enable) (hay Không, Disable) bộ nhớ đáp ứng (hay không) đối với lệnh đọc/ghi ở địa chỉ đã gọi đến.

Từ các tác vụ kể trên, ta có thể hình dung mỗi IC nhớ có một số ngõ vào ra như sau:

- **Ngõ vào địa chỉ** : mỗi vị trí nhớ xác định bởi một địa chỉ duy nhất, khi cần đọc dữ liệu ra hoặc ghi dữ liệu vào ta phải tác động vào chân địa chỉ của vị trí nhớ đó. Một IC có  $n$  chân địa chỉ sẽ có  $2^n$  vị trí nhớ. Ký hiệu các chân địa chỉ là  $A_0$  đến  $A_{n-1}$  Một IC có 10 chân địa chỉ sẽ có 1024 (1K) vị trí nhớ.

- **Ngõ vào/ra dữ liệu**: Các chân dữ liệu là các ngõ vào/ra, nghĩa là dữ liệu luôn được xử lý theo hai chiều. Thường thì dữ liệu vào/ra chung trên một chân nên các ngõ này thuộc loại ngõ ra 3 trạng thái. Số chân địa chỉ và dữ liệu của một IC xác định dung lượng nhớ của IC đó. Thí dụ một IC nhớ có 10 chân địa chỉ và 8 chân dữ liệu thì dung lượng nhớ của IC đó là 1Kx8 (8K bit hoặc 1K Byte).

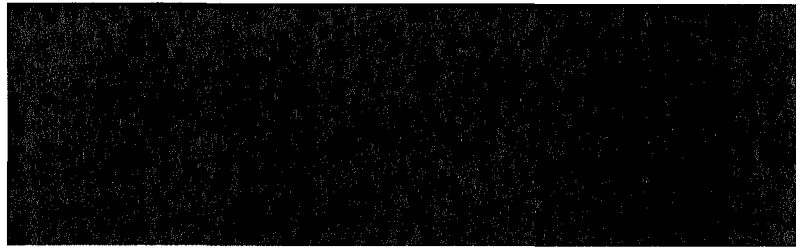
- **Các ngõ vào điều khiển**: Mỗi khi IC nhớ được chọn hoặc có yêu cầu xuất nhập dữ liệu các chân tương ứng sẽ được tác động. Ta có thể kể ra một số ngõ vào điều khiển:

- \*  $\overline{CS}$ : Chip select - Chọn chip - Khi chân này xuống thấp IC được chọn
- \*  $\overline{CE}$ : Chip Enable - Cho phép chip - Chức năng như chân  $\overline{CS}$
- \*  $\overline{OE}$ : Output Enable - Cho phép xuất - Dừng khi đọc dữ liệu
- \*  $R/\overline{W}$  : Read/Write - Đọc/Viết - Cho phép **ĐỌC** dữ liệu ra khi ở mức cao và **Ghi** dữ liệu vào khi ở mức thấp
- \*  $\overline{CAS}$  : Column Address Strobe - Chốt địa chỉ cột
- \*  $\overline{RAS}$ : Row Address Strobe - Chốt địa chỉ hàng.

Trong trường hợp chip nhớ có dung lượng lớn, để giảm kích thước của mạch giải mã địa chỉ bên trong IC, người ta chia số chân ra làm 2: địa chỉ hàng và địa chỉ cột. Như vậy phải dùng 2 mạch giải mã địa chỉ nhưng mỗi mạch nhỏ hơn rất nhiều. Thí dụ với 10 chân địa chỉ, thay vì dùng 1 mạch giải mã 10 đường sang 1024 đường, người ta dùng 2 mạch giải mã 5 đường sang 32 đường, hai mạch này rất đơn giản so với một mạch kia. Một vị trí nhớ bây giờ có 2 địa chỉ : hàng và cột, dĩ nhiên muốn truy xuất một vị trí nhớ phải có đủ 2 địa chỉ nhờ 2 tín hiệu  $\overline{RAS}$  và  $\overline{CAS}$ .

(H 7.1) cho thấy cách vẽ các nhóm chân của IC nhớ ( $m$  chân địa chỉ và  $n$  chân dữ liệu). (H 7.1b) và (H 7.1c) vẽ các chân địa chỉ và dữ liệu dưới dạng các Bus. (H 7.1b) được dùng trong các sơ đồ chi tiết và (H 7.1c) được dùng trong các sơ đồ khối.





(a)

(b)  
(H 7.1)

(c)

### ***Giao tiếp giữa IC nhớ và bộ xử lý trung tâm (CPU)***

Trong hệ thống mọi hoạt động có liên quan đến IC nhớ đều do bộ xử lý trung tâm (Central Processing Unit, CPU) quản lý. Giao tiếp giữa IC nhớ và CPU mô tả ở (H 7.2)



(H 7.2)

Một tác vụ có liên quan đến bộ nhớ được CPU thực hiện theo các bước:

- Đặt địa chỉ quan hệ lên bus địa chỉ.
- Đặt tín hiệu điều khiển lên bus điều khiển.
- Dữ liệu khả dụng xuất hiện trên bus dữ liệu, sẵn sàng để ghi vào hoặc đọc ra.

Để hoạt động của IC đồng bộ, các bước trên phải tuân thủ **giãn đồ thời gian** của từng IC nhớ (sẽ đề cập đến khi xét các loại bộ nhớ)

### **Các loại bộ nhớ bán dẫn**

Có 3 loại bộ nhớ bán dẫn :

- Bộ nhớ bán dẫn chỉ đọc : (Read Only Memory, ROM)
- Bộ nhớ truy xuất ngẫu nhiên : (Random Access Memory, RAM)

Thật ra ROM và RAM đều là loại bộ nhớ truy xuất ngẫu nhiên, nhưng RAM được giữ tên gọi này. Để phân biệt chính xác ROM và RAM ta có thể gọi ROM là **bộ nhớ chết** (nonvolatile, vĩnh cửu) và RAM là **bộ nhớ sống** (volatile, không vĩnh cửu) hoặc nếu coi ROM là **bộ nhớ chỉ đọc** thì RAM là **bộ nhớ đọc được - viết được** (Read-Write Memory)

- Thiết bị logic lập trình được : (Programmable Logic Devices, PLD) có thể nói điểm khác biệt giữa PLD với ROM và RAM là qui mô tích hợp của PLD thường không lớn như ROM và RAM và các tác vụ của PLD thì có phần hạn chế.

### ***ROM (Read Only Memory)***

Mặc dù có tên gọi như thế nhưng chúng ta phải hiểu là khi sử dụng ROM, tác vụ đọc được thực hiện rất nhiều lần so với tác vụ ghi. Thậm chí có loại ROM chỉ ghi một lần khi xuất xưởng.

Các tế bào nhớ hoặc từ nhớ trong ROM sắp xếp theo dạng ma trận mà mỗi phần tử chiếm một vị trí xác định bởi một địa chỉ cụ thể và nối với ngã ra một mạch giải mã địa chỉ

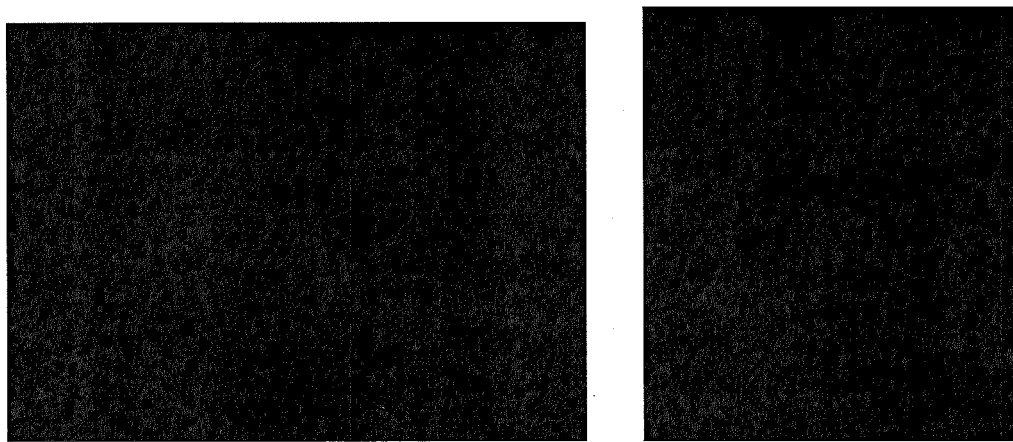
bên trong IC. Nếu mỗi vị trí chứa một tế bào nhớ ta nói ROM có **tổ chức bit** và mỗi vị trí là một từ nhớ ta có **tổ chức từ**.

Ngoài ra, để giảm mức độ cồng kềnh của mạch giải mã, mỗi vị trí nhớ có thể được xác định bởi 2 đường địa chỉ : đường địa chỉ hàng và đường địa chỉ cột và trong bộ nhớ có 2 mạch giải mã nhưng mỗi mạch có số ngõ vào bằng  $1/2$  số đường địa chỉ của cả bộ nhớ.

### ROM mặt nạ (Mask Programmed ROM, MROM)

Đây là loại ROM được chế tạo để thực hiện một công việc cụ thể như các bảng tính, bảng lượng giác , bảng logarit . . . ngay sau khi xuất xưởng. Nói cách khác, các tế bào nhớ trong ma trận nhớ đã được tạo ra theo một chương trình đã xác định trước bằng phương pháp mặt nạ: đưa vào các linh kiện điện tử nối từ **đường từ** qua **đường bit** để tạo ra một giá trị bit và để trống cho giá trị bit ngược lại.

- (H 7.3) là mô hình của một MROM trong đó các ô vuông là nơi chứa (hay không) một linh kiện (diod, transistor BJT hay MOSFET) để tạo bit. Mỗi ngõ ra của mạch giải mã địa chỉ gọi là đường từ và đường nối tế bào nhớ ra ngoài gọi là đường bit. Khi đường từ lên mức cao thì tế bào nhớ hoặc từ nhớ được chọn.

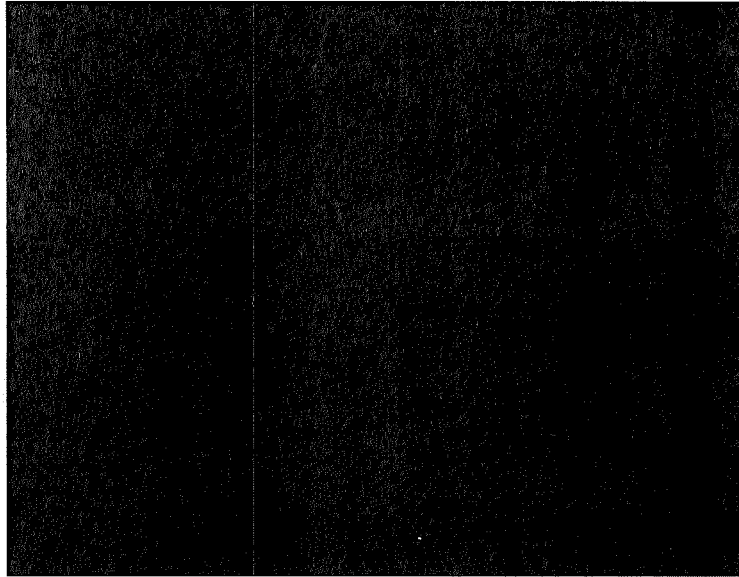


(H 7.3)

Nếu tế bào nhớ là Diod hoặc BJT thì sự hiện diện của linh kiện tương ứng với bit 1 (lúc này đường từ lên cao, Transistor hoặc diod dẫn, dòng điện qua điện trở tạo điện thế cao ở hai đầu điện trở) còn vị trí nhớ trống tương ứng với bit 0.

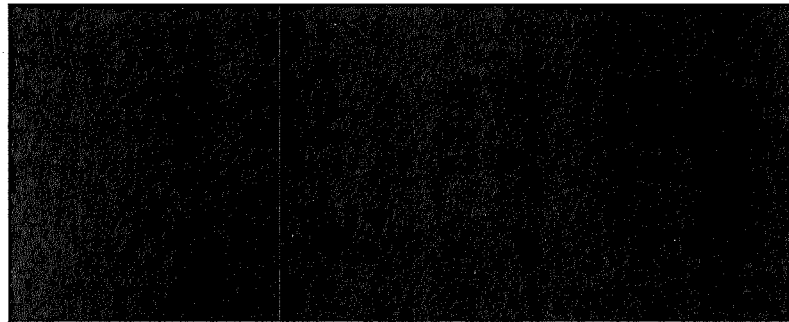
Đối với loại linh kiện MOSFET thì ngược lại, nghĩa là sự hiện diện của linh kiện tương ứng với bit 0 còn vị trí nhớ trống tương ứng với bit 1 (muốn có kết quả như loại BJT thì thêm ở ngõ ra các cổng đảo).

(H 7.4) là một thí dụ bộ nhớ MROM có dung lượng 16x1 với các mạch giải mã hàng và cột (các mạch giải mã 2 đường sang 4 đường của hàng và cột đều dùng Transistor MOS và có cùng cấu trúc).



(H 7.4)

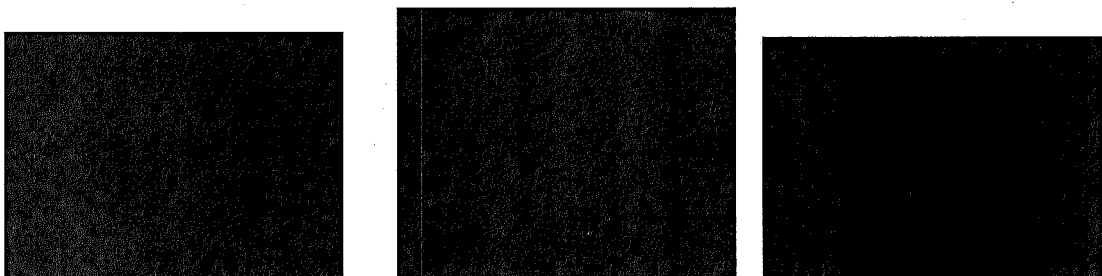
Trong thực tế, để đơn giản cho việc thực hiện, ở mỗi vị trí nhớ người ta đều cho vào một transistor MOS. Nhưng ở những vị trí ứng với bit 1 các transistor MOS được chế tạo với lớp SiO<sub>2</sub> dày hơn làm tăng điện thế ngưỡng của nó lên, kết quả là transistor MOS này luôn luôn không dẫn điện (H 7.5), Các transistor khác dẫn điện bình thường.



(H 7.5)

### ROM lập trình được (Programmable ROM, PROM)

Có cấu tạo giống MROM nhưng ở mỗi vị trí nhớ đều có linh kiện nối với cầu chì. Như vậy khi xuất xưởng các ROM này đều chứa cùng một loại bit (gọi là ROM trắng), lúc sử dụng người lập trình thay đổi các bit mong muốn bằng cách phá vỡ cầu chì ở các vị trí tương ứng với bit đó. Một khi cầu chì đã bị phá vỡ thì không thể nối lại được do đó loại ROM này cho phép lập trình một lần duy nhất để sử dụng, nếu bị lỗi không thể sửa chữa được (H 7.6).

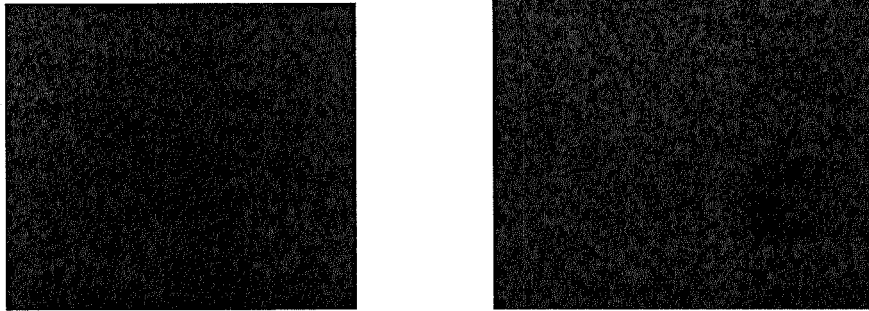


(H 7.6)

Người ta có thể dùng 2 diod mắc ngược chiều nhau, mạch không dẫn điện, để tạo bit 0, khi lập trình thì một diod bị phá hỏng tạo mạch nối tắt, diod còn lại dẫn điện cho bit 1

## ROM lập trình được, xóa được bằng tia U.V. (Ultra Violet Erasable Programmable ROM, U.V. EPROM)

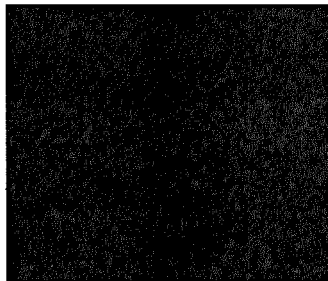
Đây là loại ROM rất tiện cho người sử dụng vì có thể dùng được nhiều lần bằng cách xóa và nạp lại. Cấu tạo của tế bào nhớ của U.V. EPROM dựa vào một transistor MOS có cấu tạo đặc biệt gọi là FAMOS (Floating Gate Avalanche Injection MOS)



(H 7.7)

Trên nền chất bán dẫn N pha loãng, tạo 2 vùng P pha đậm ( $P^+$ ) nổi ra ngoài cho 2 cực S (Source) và D (Drain). Trong lớp cách điện  $SiO_2$  giữa 2 cực người ta cho vào một thỏi Silicon không nối với bên ngoài và được gọi là **cổng nổi**. Khi nguồn  $V_{DD}$ , phân cực ngược giữa cực nền và Drain còn nhỏ, transistor không dẫn, nhưng nếu tăng  $V_{DD}$  đủ lớn, hiện tượng thác đổ (avalanche) xảy ra, electron đủ năng lượng chui qua lớp cách điện tới bám vào cổng nổi. Do hiện tượng cảm ứng, một điện lộ P hình thành nối hai vùng bán dẫn  $P^+$ , transistor trở nên dẫn điện. Khi cắt nguồn, transistor tiếp tục dẫn điện vì electron không thể trở về để tái hợp với lỗ trống.

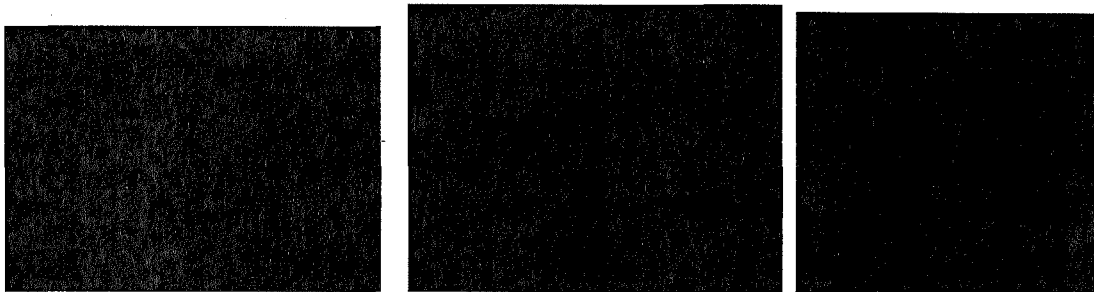
Để xóa EPROM, người ta chiếu tia U.V. vào các tế bào trong một khoảng thời gian xác định để electron trên cổng nổi nhận đủ năng lượng vượt qua lớp cách điện trở về vùng nền tái hợp với lỗ trống xóa điện lộ P và transistor trở về trạng thái không dẫn ban đầu.



(H 7.8)

Mỗi tế bào nhớ EPROM gồm một transistor FAMOS nối tiếp với một transistor MOS khác mà ta gọi là transistor chọn, như vậy vai trò của FAMOS giống như là một cầu chì nhưng có thể phục hồi được.

Để loại bỏ transistor chọn người ta dùng transistor SAMOS (Stacked Gate Avalanche Injection MOS) có cấu tạo tương tự transistor MOS nhưng có đến 2 cổng nằm chồng lên nhau, một được nối ra cực Gate và một để nổi. Khi cổng nổi tích điện sẽ làm gia tăng điện thế thêm khiến transistor trở nên khó dẫn điện hơn. Như vậy nếu ta chọn điện thế  $V_c$  ở khoảng giữa  $V_{T1}$  và  $V_{T2}$  là 2 giá trị điện thế thêm tương ứng với 2 trạng thái của transistor ( $V_{T1} < V_c < V_{T2}$ ) thì các transistor không được lập trình (không có lớp electron ở cổng nổi) sẽ dẫn còn các transistor được lập trình sẽ không dẫn.



(H 7.9)

Điểm bất tiện của U.V EPROM là cần thiết bị xóa đặc biệt phát tia U.V. và mỗi lần xóa tất cả tế bào nhớ trong một IC nhớ đều bị xóa. Như vậy người sử dụng phải nạp lại toàn bộ chương trình

### **ROM lập trình được và xóa được bằng xung điện (Electrically Erasable PROM, EEPROM hay Electrically Alterable PROM, EAPROM)**

Đây là loại ROM lập trình được và xóa được nhờ xung điện và đặc biệt là có thể xóa để sửa trên từng byte. Các tế bào nhớ EEPROM sử dụng transistor MNOS (Metal Nitride Oxide Semiconductor) có cấu tạo như (H 7.10).



(H 7.10)

Giữa lớp kim loại nối ra các cực và lớp  $\text{SiO}_2$  là một lớp mỏng chất Nitrua Silic ( $\text{Si}_3\text{N}_4$ ) - từ 40nm đến 650nm - Dữ liệu được nạp bằng cách áp một điện thế dương giữa cực G và S (khoảng 20 đến 25V trong 100ms). Do sự khác biệt về độ dẫn điện, electron tích trên bề mặt giữa 2 lớp  $\text{SiO}_2$  và  $\text{Si}_3\text{N}_4$ , các electron này tồn tại khi đã ngắt nguồn và làm thay đổi trạng thái dẫn điện của transistor. Bây giờ nếu áp một điện thế âm giữa cực G và S ta sẽ được một lớp điện tích trái dấu với trường hợp trước. Như vậy hai trạng thái khác nhau của Transistor có thể thiết lập được bởi hai điện thế ngược chiều nhau và như vậy các tế bào nhớ được ghi và xóa với 2 xung điện trái dấu nhau.

### **FLASH ROM**

EPROM là loại nonvolatile, có tốc độ truy xuất nhanh (khoảng 120ns), mật độ tích hợp cao, giá thành rẻ tuy nhiên để xóa và nạp lại phải dùng thiết bị đặc biệt và lấy ra khỏi mạch.

EEPROM cũng nonvolatile, cũng có tốc độ truy xuất nhanh, cho phép xóa và nạp lại ngay trong mạch trên từng byte nhưng có mật độ tích hợp thấp và giá thành cao hơn EPROM.

Bộ nhớ FLASH ROM tận dụng được các ưu điểm của hai loại ROM nói trên, nghĩa là có tốc độ truy xuất nhanh, có mật độ tích hợp cao nhưng giá thành thấp.

Hầu hết các FLASH ROM sử dụng cách xóa đồng thời cả khối dữ liệu nhưng rất nhanh (hàng trăm ms so với 20 min của U.V. EPROM). Những FLASH ROM thế hệ mới cho phép xóa từng sector (512 byte) thậm chí từng vị trí nhớ mà không cần lấy IC ra khỏi mạch. FLASH ROM có thời gian ghi khoảng  $10\mu\text{s}/\text{byte}$  so với  $100\mu\text{s}$  đối với EPROM và 5 ms đối với EEPROM

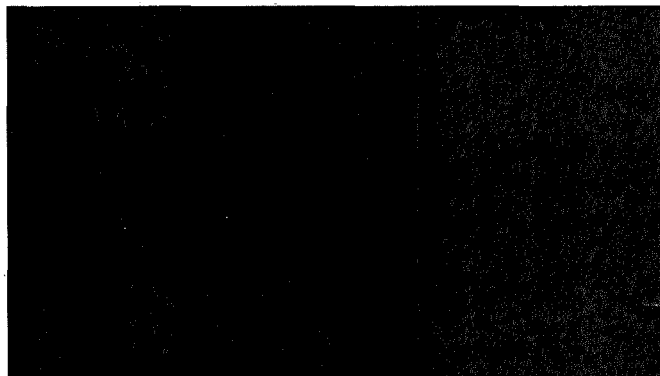
## Giản đồ thời gian của ROM

Ngoại trừ MROM chỉ dùng ở chế độ đọc, các loại ROM khác đều sử dụng ở hai chế độ đọc và nạp chương trình.

Như vậy ta có hai loại giản đồ thời gian: Giản đồ thời gian đọc và giản đồ thời gian nạp trình.

(H 7.11) là giản đồ thời gian tiêu biểu cho một chu kỳ đọc của ROM.

Các giá trị địa chỉ, các tín hiệu  $R/\overline{W}$  và  $\overline{CS}$  được cấp từ CPU khi cần thực hiện tác vụ đọc dữ liệu tại một địa chỉ nào đó. Thời gian để thực hiện một tác vụ đọc gọi là chu kỳ đọc rC. Trong một chu kỳ đọc có thể kể một số thời gian sau:



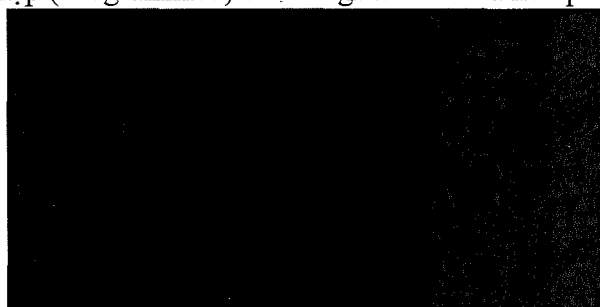
(H 7.11)

-  $t_{ACC}$ : Address Access time: Thời gian truy xuất địa chỉ: Thời gian tối đa từ lúc CPU đặt địa chỉ lên bus địa chỉ đến lúc dữ liệu có giá trị trên bus dữ liệu. Đối với ROM dùng BJT thời gian này khoảng từ 30 ns đến 90 ns, còn loại MOS thì từ 200 ns đến 900 ns.

-  $t_{ACS}$  ( $t_{ACE}$ ): Chip select (enable) access time: Thời gian thâm nhập chọn chip: Thời gian tối đa từ lúc tín hiệu  $\overline{CS}$  được đặt lên bus điều khiển đến lúc dữ liệu có giá trị trên bus dữ liệu. ROM BJT khoảng 20 ns, MOS 100 ns

-  $t_H$  (Hold time): Thời gian dữ liệu còn tồn tại trên bus dữ liệu kể từ lúc tín hiệu  $\overline{CS}$  hết hiệu lực

(H 7.12) là giản đồ thời gian của một chu kỳ nạp dữ liệu cho EPROM. Một chu kỳ nạp liệu bao gồm thời gian nạp (Programmed) và thời gian kiểm tra kết quả (Verify)



(H 7.12)

### ***Thiết bị logic lập trình được (Programmable logic devices, PLD)***

Là tên gọi chung các thiết bị có tính chất nhớ và có thể lập trình để thực hiện một công việc cụ thể nào đó

Trong công việc thiết kế các hệ thống, đôi khi người ta cần một số mạch tổ hợp để thực hiện một hàm logic nào đó. Việc sử dụng mạch này có thể lặp lại thường xuyên và sự thay đổi một tham số của hàm có thể phải được thực hiện để thỏa mãn yêu cầu của việc thiết kế. Nếu phải thiết kế từ các công logic cơ bản thì mạch sẽ rất cồng kềnh, tốn kém mạch in,

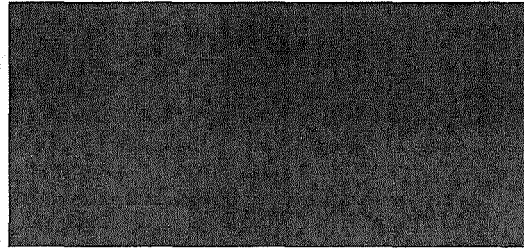
dây nối nhiều, kết quả là độ tin cậy không cao. Như vậy, sẽ rất tiện lợi nếu các mạch này được chế tạo sẵn và người sử dụng có thể chỉ tác động vào để làm thay đổi một phần nào chức năng của mạch bằng cách lập trình. Đó là ý tưởng cơ sở cho sự ra đời của thiết bị logic lập trình được. Các thiết bị này có thể được xếp loại như bộ nhớ và gồm các loại: PROM, PAL (Programmable Array Logic) và PLA (Programmable Logic Array).

Trước nhất, chúng ta xét qua một số qui ước trong cách biểu diễn các phần tử của PLD

Một biến trong các hàm thường xuất hiện ở dạng nguyên và đảo của nó nên chúng ta dùng ký hiệu đậm và đảo chung trong một công có 2 ngã ra.

Một **nối chết**, còn gọi là **nối cứng** (không thay đổi được) được vẽ bởi một chấm đậm (.) và một **nối sống**, còn gọi là **nối mềm** (dùng lập trình) bởi một dấu (x). Nối sống thực chất là một cầu chì, khi lập trình thì được phá bỏ.

Một công nhiều ngã vào thay thế bởi một ngã vào duy nhất với nhiều mối nối (H 7.13).



(H 7.13)

Chúng ta chỉ lấy thí dụ với mạch tương đối đơn giản để thấy được cấu tạo của các PLD, đó là các PLD chỉ thực hiện được 4 hàm mỗi hàm gồm 4 biến, như vậy mạch gồm 4 ngã vào và 4 ngã ra. Trên thực tế số hàm và biến của một PLD rất lớn.

## PROM

(H 7.14) là cấu tạo PROM có 4 ngã vào và 4 ngã ra.

Có tất cả 16 cổng AND có 4 ngã vào được nối chết với các ngã ra đảo và không đảo của các biến vào, ngã ra các cổng AND là 16 tổ hợp của 4 biến (Gọi là đường tích) Các cổng OR có 16 ngã vào được nối sống để thực hiện hàm tổng (đường tổng). Như vậy với PROM việc lập trình thực hiện ở các đường tổng.

Thí dụ dùng PROM này để tạo các hàm sau:

$$O_1 = A + \overline{DB} + \overline{DC} \quad O_2 = \overline{DCBA} + DCBA \quad O_3 = \overline{CBA} \quad O_4 = BA + \overline{DC}$$

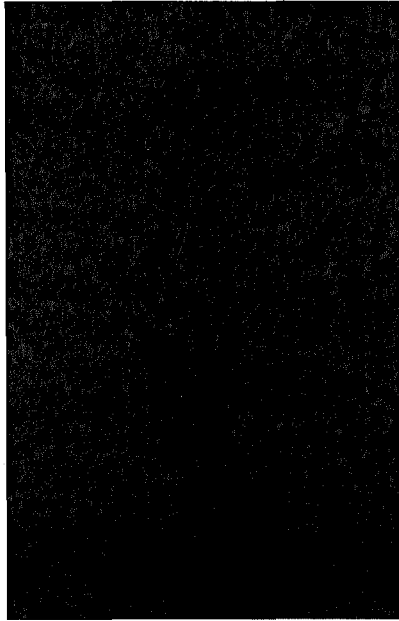
Ta phải chuẩn hóa các hàm chưa chuẩn

$$O_1 = \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA}$$

$$O_3 = \overline{CBA} = \overline{DCBA} + \overline{DCBA}$$

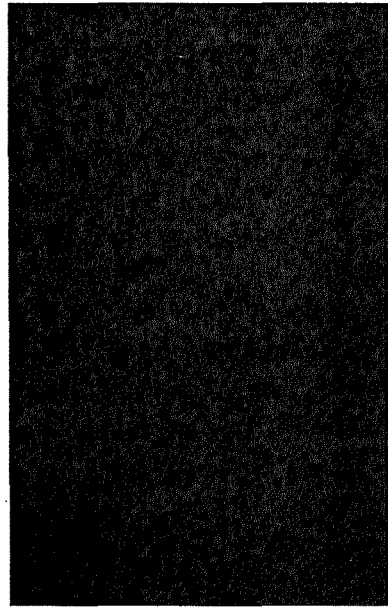
$$O_4 = BA + \overline{DC} = \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA} + \overline{DCBA}$$

Mạch cho ở (H 7.14b)



(a)

(H 7.14)



(b)

### PAL

Mạch tương tự với IC PROM, PAL có các cổng AND 8 ngõ vào được nối sống và 4 cổng OR mỗi cổng có 4 ngõ vào nối chết với 4 đường tích. Như vậy việc lập trình được thực hiện trên các đường tích

(H 7.15b) là IC PAL đã được lập trình để thực hiện các hàm trong thí dụ trên:

$$O_1 = A + \overline{DB} + \overline{DC} \quad O_2 = \overline{DCBA} + \overline{DCBA} \quad O_3 = \overline{CBA} \quad O_4 = \overline{BA} + \overline{DC}$$



(a)

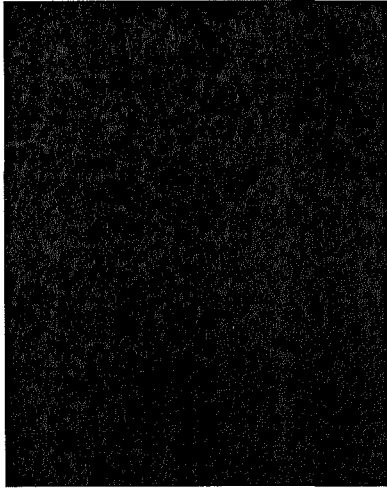


(b)

(H 7.15)

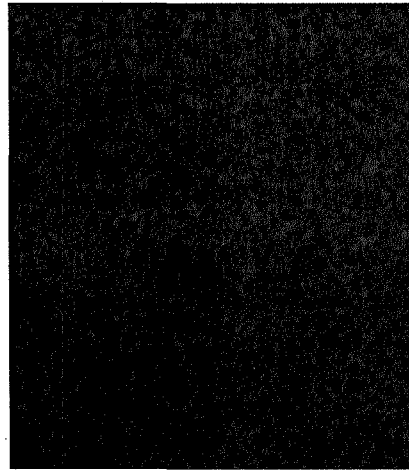






(a)

(H 7.17)



(b)

(H 7.17a) là một tế bào nhớ RAM tĩnh dùng transistor BJT với 2 đường địa chỉ hàng và cột.

Khi một trong hai đường địa chỉ hàng hoặc cột ở mức thấp các tế bào không được chọn vì cực E có điện thế thấp hai Transistor đều dẫn, mạch không hoạt động như một FF. Khi cả hai địa chỉ hàng và cột lên cao, mạch hoạt động như FF, hai trạng thái 1 và 0 của tế bào nhớ được đặc trưng bởi hai trạng thái khác nhau của 2 đường bit và bit .

Giả sử khi  $T_1$  dẫn thì  $T_2$  ngưng, đường bit có dòng điện chạy qua, tạo điện thế cao ở  $R_3$  trong khi đó đường bit không có dòng chạy qua nên ở  $R_4$  có điện thế thấp. Nếu ta qui ước trạng thái này tương ứng với bit 1 thì trạng thái ngược lại, là trạng thái  $T_1$  ngưng và  $T_2$  dẫn, hiệu thế ở điện trở  $R_3$  thấp và ở  $R_4$  cao, sẽ là bit 0.  $R_3$  và  $R_4$  có tác dụng biến đổi dòng điện ra điện thế.

Đối với tế bào nhớ dùng MOS, hai đường từ nối với  $T_5, T_6$  và  $T_7, T_8$  nên khi một trong hai đường từ ở mức thấp  $T_1$  và  $T_2$  bị cô lập khỏi mạch, tế bào nhớ không được chọn. Khi cả hai lên cao mạch hoạt động tương tự như trên. Trong mạch này  $R_1$  và  $R_2$  thay bởi  $T_3$  và  $T_4$  và không cần  $R_3$  và  $R_4$  như mạch dùng BJT.

(H 7.18) là mạch điều khiển chọn chip và thực hiện tác vụ đọc/viết vào tế bào nhớ.



(H 7.18)

OPAMP giữ vai trò mạch so sánh điện thế hai đường bit và bit cho ở ngã ra mức cao hoặc thấp tùy kết quả so sánh này (tương ứng với 2 trạng thái của tế bào nhớ) và dữ liệu được đọc ra khi cổng đệm thứ 2 mở ( $R/\overline{W}$  lên cao).

Khi cổng đệm thứ nhất mở ( $R/\overline{W}$  xuống thấp) dữ liệu được ghi vào tế bào nhớ qua cổng đệm 1. Cổng 3 tạo ra hai tín hiệu ngược pha từ dữ liệu vào. Nếu hai tín hiệu này cùng trạng thái với hai đường bit và bit của mạch trước đó, mạch sẽ không đổi trạng thái nghĩa là

nếu tế bào nhớ đang lưu bit giống như bit muốn ghi vào thì mạch không thay đổi. Bây giờ, nếu dữ liệu cần ghi khác với dữ liệu đang lưu trữ thì mạch FF sẽ thay đổi trạng thái cho phù hợp với 2 tín hiệu ngược pha được tạo ra từ dữ liệu. Bit mới đã được ghi vào.

- Chu kỳ đọc của SRAM

Giãn đồ thời gian một chu kỳ đọc của SRAM tương tự như giãn đồ thời gian một chu kỳ đọc của ROM (H 7.11) thêm điều kiện tín hiệu  $\overline{R/W}$  lên mức cao.

- Chu kỳ viết của SRAM

(H 7.19) là giãn đồ thời gian một chu kỳ viết của SRAM

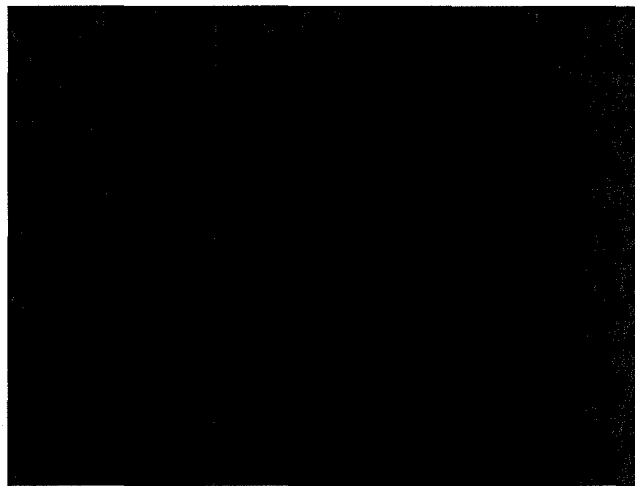
Một chu kỳ viết t<sub>wc</sub> bao gồm:

- **t<sub>AS</sub>** (Address Setup time): Thời gian thiết lập địa chỉ : Thời gian để giá trị địa chỉ ổn định trên bus địa chỉ cho tới lúc tín hiệu  $\overline{CS}$  tác động.

- **t<sub>w</sub>** (Write time): Thời gian từ lúc tín hiệu  $\overline{CS}$  tác động đến lúc dữ liệu có giá trị trên bus dữ liệu.

- **t<sub>DS</sub> và t<sub>DH</sub>**: Khoảng thời gian dữ liệu tồn tại trên bus dữ liệu bao gồm thời gian trước (t<sub>DS</sub>) và sau (t<sub>DH</sub>) khi tín hiệu  $\overline{CS}$  không còn tác động

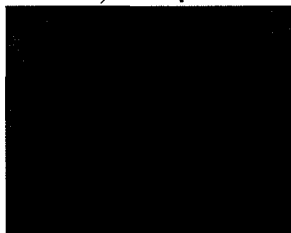
- **t<sub>AH</sub>** (Address Hold time): Thời gian giữ địa chỉ: từ lúc tín hiệu  $\overline{CS}$  không còn tác động đến lúc xuất hiện địa chỉ mới.



(H 7.19)

### RAM động (Dynamic RAM, DRAM)

(H 7.20a) là một tế bào nhớ của DRAM



(a)



(H 7.20)

(b)

(H 7.20b) là một cách biểu diễn tế bào nhớ DRAM trong đó đơn giản một số chi tiết được dùng để mô tả các tác vụ viết và đọc tế bào nhớ này.

Các khóa từ S<sub>1</sub> đến S<sub>4</sub> là các transistor MOS được điều khiển bởi các tín hiệu ra từ mạch giải mã địa chỉ và tín hiệu  $\overline{R/W}$ .

Để ghi dữ liệu vào tế bào, các khóa  $S_1$  và  $S_2$  đóng trong khi  $S_3$  và  $S_4$  mở. Bit 1 thực hiện việc nạp điện cho tụ C và bit 0 làm tụ C phóng điện. Sau đó các khóa sẽ mở để cô lập C với phần mạch còn lại. Một cách lý tưởng thì C sẽ duy trì trạng thái của nó vĩnh viễn nhưng thực tế luôn luôn có sự rỉ điện qua các khóa ngay cả khi chúng mở do đó C bị mất dần điện tích.

Để đọc dữ liệu các khóa  $S_2$ ,  $S_3$ ,  $S_4$  đóng và  $S_1$  mở, tụ C nối với một mạch so sánh với một điện thế tham chiếu để xác định trạng thái logic của nó. Điện thế ra mạch so sánh chính là dữ liệu được đọc ra. Do  $S_2$  và  $S_4$  đóng, dữ liệu ra được nối ngược lại tụ C để làm tươi nó. Nói cách khác, bit dữ liệu trong tế bào nhớ được làm tươi mỗi khi nó được đọc.

Sử dụng DRAM, được một thuận lợi là dung lượng nhớ khá lớn nhưng phải có một số mạch phụ trợ:

- Mạch đa hợp địa chỉ vì DRAM luôn sử dụng địa chỉ hàng và cột
- Mạch làm tươi để phục hồi dữ liệu có thể bị mất sau một khoảng thời gian ngắn nào đó.

### **a. Đa hợp địa chỉ**

Như đã nói trên, do dung lượng của DRAM rất lớn nên phải dùng phương pháp đa hợp để chọn một vị trí nhớ trong DRAM. Mỗi vị trí nhớ sẽ được chọn bởi 2 địa chỉ hàng và cột lần lượt xuất hiện ở ngã vào địa chỉ.

Thí dụ với DRAM có dung lượng  $16K \times 1$ , thay vì phải dùng 14 đường địa chỉ ta chỉ cần dùng 7 đường và mạch đa hợp  $14 \rightarrow 7$  (7 đa hợp  $2 \rightarrow 1$ ) để chọn 7 trong 14 đường địa chỉ ra từ CPU (H 7.21). Bộ nhớ có cấu trúc là một ma trận  $128 \times 128$  tế bào nhớ, sắp xếp

## ◆ CHƯƠNG 7 : BIẾN ĐỔI AD & DA

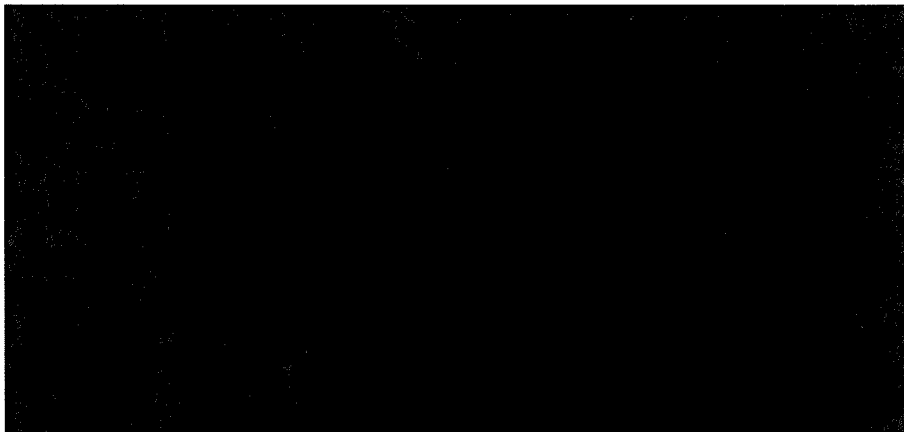
Có thể nói sự biến đổi qua lại giữa các tín hiệu từ dạng tương tự sang dạng số là cần thiết vì:

- Hệ thống số xử lý tín hiệu số mà tín hiệu trong tự nhiên là tín hiệu tương tự: cần thiết có mạch đổi tương tự sang số.

- Kết quả từ các hệ thống số là các đại lượng số: cần thiết phải đổi thành tín hiệu tương tự để có thể tác động vào các hệ thống vật lý và thể hiện ra bên ngoài (thí dụ tái tạo âm thanh hay hình ảnh) hay dùng vào việc điều khiển sau đó (thí dụ dùng điện thế tương tự để điều khiển vận tốc động cơ)

### **Biến đổi số - tương tự (digital to analog converter, ADC)**

*Mạch biến đổi DAC dùng mạng điện trở có trọng lượng khác nhau (Weighted resistor network)*



(H 8.1)

Trong mạch trên, nếu thay OP-AMP bởi một điện trở tải, ta có tín hiệu ra là dòng điện.

Như vậy OP-AMP giữ vai trò biến dòng điện ra thành điện thế ra, đồng thời nó là một mạch cộng

$$\begin{aligned} \text{Ta có } v_0 = -R_F \cdot I &= -(2^3 b_3 + 2^2 b_2 + 2b_1 + b_0) V_r \cdot R_F / 2^3 R \\ &= -(2^{n-1} b_{n-1} + 2^{n-2} b_{n-2} + \dots + 2b_1 + b_0) V_r \cdot R_F / 2^{n-1} \cdot R \end{aligned}$$

Nếu  $R_F = R$  thì:

$$v_0 = -(2^{n-1} b_{n-1} + 2^{n-2} b_{n-2} + \dots + 2b_1 + b_0) V_r / 2^{n-1}$$

Thí dụ:

1/ Khi số nhị phân là 0000 thì  $v_0 = 0$   
 1111 thì  $v_0 = -15V_r / 8$

2/ Với  $V_r = 5V$ ;  $R = R_F = 1k\Omega$

Ta có kết quả chuyển đổi như sau:

$b_3$	$b_2$	$b_1$	$b_0$	$v_0$ (V)
0	0	0	0	0
0	0	0	1	-0,625 ← LSB
0	0	1	0	-1,250
0	0	1	1	-1,875
0	1	0	0	-2,500
0	1	0	1	-3,125
0	1	1	0	-3,750
0	1	1	1	-4,375
1	0	0	0	-5,000
1	0	0	1	-5,625
1	0	1	0	-6,250
1	0	1	1	-6,875
1	1	0	0	-7,500
1	1	0	1	-8,125
1	1	1	0	-8,750
1	1	1	1	-9,375 ← Full Scale ( $V_{FS}$ )

Mạch có một số hạn chế:

- Sự chính xác tùy thuộc vào điện trở và mức độ ổn định của nguồn tham chiếu  $V_r$
- Với số nhị phân nhiều bit thì cần các điện trở có giá trị rất lớn, khó thực hiện.

### Mạch đổi DAC dùng mạng điện trở hình thang



(H 8.2)

Cho  $R_F = 2R$  và lần lượt

Cho  $b_3 = 1$  các bit khác = 0, ta được:  $v_0 = -8(V_r / 24)$

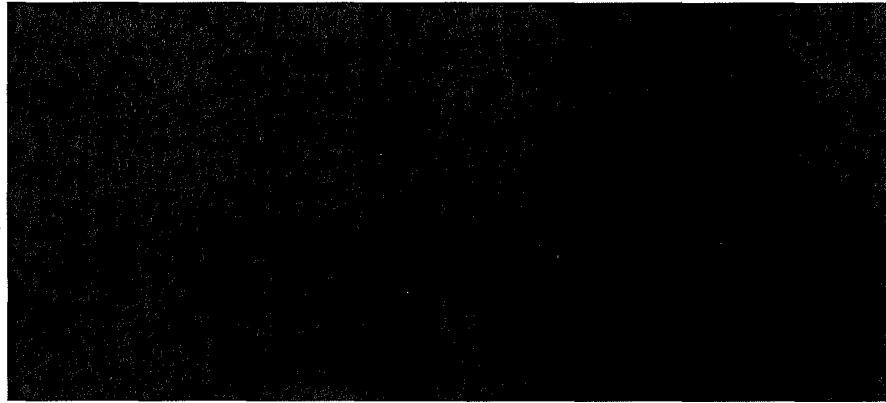
Cho  $b_2 = 1$  các bit khác = 0, ta được:  $v_0 = -4(V_r / 24)$

Cho  $b_1 = 1$  các bit khác = 0, ta được:  $v_0 = -2(V_r / 24)$

Cho  $b_0 = 1$  các bit khác = 0, ta được:  $v_0 = -(V_r / 24)$

Ta thấy  $v_0$  tỉ lệ với giá trị B của tổ hợp bit  $B = (b_3 b_2 b_1 b_0)_2 \Rightarrow v_0 = -B(V_r / 24)$

### Mạch đổi DAC dùng nguồn dòng có trọng lượng khác nhau



(H 8.3)

### Đặc tính kỹ thuật của mạch đổi DAC

#### Bit có ý nghĩa thấp nhất (LSB) và bit có ý nghĩa cao nhất (MSB)

Qua các mạch biến đổi DAC kể trên ta thấy vị trí khác nhau của các bit trong số nhị phân cho giá trị biến đổi khác nhau, nói cách khác trị biến đổi của một bit tùy thuộc vào trọng lượng của bit đó.

Nếu ta gọi trị toàn giai là  $V_{FS}$  thì bit LSB có giá trị là:

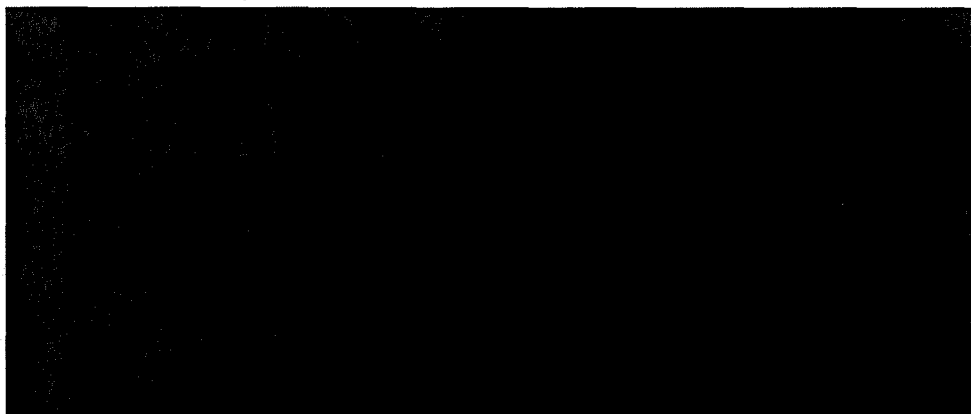
$$\text{LSB} = V_{FS} / (2^n - 1)$$

và bit

$$\text{MSB} = V_{FS} \cdot 2^{n-1} / (2^n - 1)$$

Điều này được thể hiện trong kết quả của thí dụ 2 ở trên.

(H 8.4) là đặc tuyến chuyển đổi của một số nhị phân 3 bit



(a)

(b)

(H 8.4)

(H 8.4a) là đặc tuyến lý tưởng, tuy nhiên, trong thực tế để đường trung bình của đặc tính chuyển đổi đi qua điểm 0 điện thế tương tự ra được làm lệch  $(1/2)\text{LSB}$  (H 8.4b). Như vậy điện thế tương tự ra được xem như thay đổi ở ngay giữa hai mã số nhị phân vào kế nhau. Thí

dụ khi mã số nhị phân vào là 000 thì điện thế tương tự ra là 0 và điện thế tương tự ra sẽ lên nấc kế 000+(1/2)LSB rồi nấc kế tiếp ở 001+(1/2)LSB.v.v....Trị tương tự ra ứng với 001 gọi tắt là 1LSB và trị toàn giai  $V_{FS} = 7\text{LSB}$  tương ứng với số 111

#### 8.1.4.2 Sai số nguyên lượng hóa (quantization error)

Trong sự biến đổi, ta thấy ứng với một giá trị nhị phân vào, ta có một khoảng điện thế tương tự ra. Như vậy có một sai số trong biến đổi gọi là sai số nguyên lượng hóa và  $= (1/2)\text{LSB}$

#### Độ phân giải (resolution)

Độ phân giải được hiểu là giá trị thay đổi nhỏ nhất của tín hiệu tương tự ra có thể có khi số nhị phân vào thay đổi. Độ phân giải còn được gọi là trị bước (step size) và bằng trọng lượng bit LSB.

Số nhị phân n bit có  $2^n$  giá trị và  $2^n - 1$  bước

Hiệu thế tương tự ra xác định bởi  $v_0 = k.(B)_2$

Trong đó k chính là độ phân giải và  $(B)_2$  là số nhị phân

Người ta thường tính phần trăm phân giải:

$$\%res = (k / V_{FS})100 \%$$

Với số nhị phân n bit

$$\%res = [1 / (2^n - 1)]100 \%$$

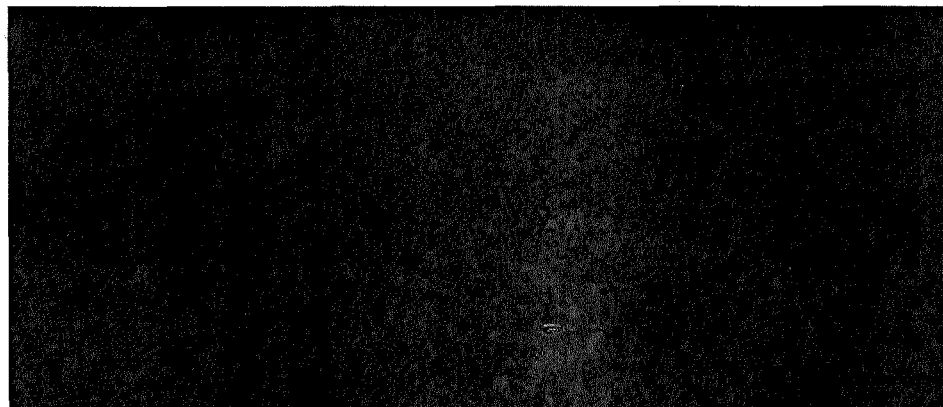
Các nhà sản xuất thường dùng số bit của số nhị phân có thể được biến đổi để chỉ độ phân giải. Số bit càng lớn thì độ phân giải càng cao (finer resolution)

#### Độ tuyến tính (linearity)

Khi điện thế tương tự ra thay đổi đều với số nhị phân vào ta nói mạch biến đổi có tính tuyến tính

#### Độ đúng (accuracy)

Độ đúng (còn gọi là độ chính xác) tuyệt đối của một DAC là hiệu số giữa điện thế tương tự ra và điện thế ra lý thuyết tương ứng với mã số nhị phân vào. Hai số nhị phân kế nhau phải cho ra hai điện thế tương tự khác nhau đúng 1LSB, nếu không mạch có thể tuyến tính nhưng không đúng (H 8.5)



a/ Tuyến tính

b/ Tuyến tính nhưng không đúng  
(H 8.5)

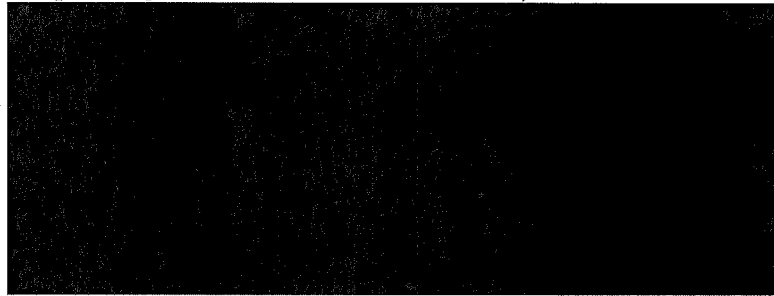


## Biến đổi tương tự - số (analog to digital converter, ADC)

### Mạch lấy mẫu và giữ (sample and hold)

Để biến đổi một tín hiệu tương tự sang tín hiệu số, người ta không thể biến đổi mọi giá trị của tín hiệu tương tự mà chỉ có thể biến đổi một số giá trị cụ thể bằng cách **lấy mẫu** tín hiệu đó theo một chu kỳ xác định nhờ một tín hiệu có dạng xung. Ngoài ra, mạch biến đổi cần một khoảng thời gian cụ thể (khoảng  $1\mu\text{s} - 1\text{ms}$ ) do đó **cần giữ mức tín hiệu** biến đổi trong khoảng thời gian này để mạch có thể thực hiện việc biến đổi chính xác. Đó là nhiệm vụ của mạch lấy mẫu và giữ.

(H 8.6) là dạng mạch lấy mẫu và giữ cơ bản: Điện thế tương tự cần biến đổi được lấy mẫu trong thời gian rất ngắn do tụ nạp điện nhanh qua tổng trở ra thấp của OP-AMP khi các transistor dẫn và giữ giá trị này trong khoảng thời gian transistor ngưng (tụ phóng rất chậm qua tổng trở vào rất lớn của OP-AMP)



(H 8.6)

### Nguyên tắc mạch biến đổi ADC

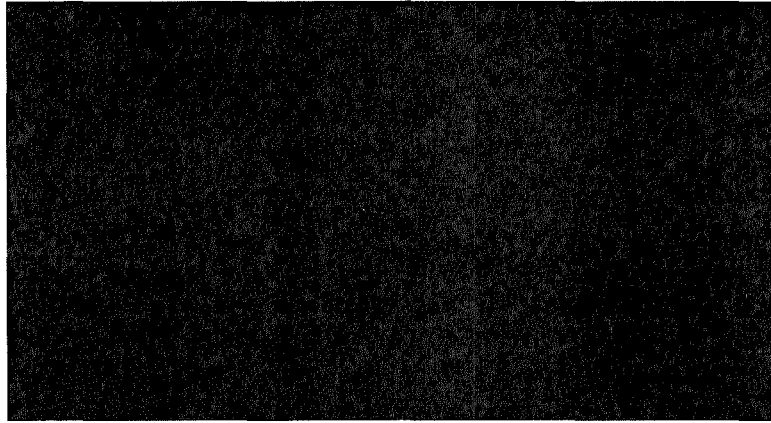
Mạch biến đổi ADC gồm bộ phận trung tâm là một mạch so sánh (H 8.7). Điện thế tương tự chưa biết  $v_a$  áp vào một ngõ vào của mạch so sánh, còn ngõ vào kia nối đến một điện thế tham chiếu thay đổi theo thời gian  $V_r(t)$ . Khi chuyển đổi điện thế tham chiếu tăng theo thời gian cho đến khi bằng hoặc gần bằng với điện thế tương tự (với một sai số nguyên lượng hóa). Lúc đó mạch tạo mã số ra có giá trị ứng với điện thế vào chưa biết. Vậy nhiệm vụ của mạch tạo mã số là thử một bộ số nhị phân sao cho hiệu số giữa  $v_a$  và trị nguyên lượng hóa sau cùng nhỏ hơn  $1/2 \text{ LSB}$

$$|v_a - (V_{FS} / 2^n - 1)(B)_2| < 1/2 \text{ LSB}$$



(H 8.7)

### Mạch đổi dùng điện thế tham chiếu nấc thang



(a) (H 8.8) (b)

Một cách đơn giản để tạo điện thế tham chiếu có dạng nấc thang là dùng một mạch DAC mà số nhị phân vào được lấy từ mạch đếm lên (H 8.8). Khi có xung bắt đầu FlipFlop và

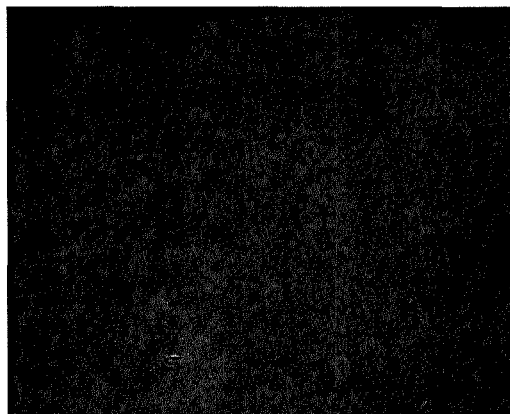
mạch đếm được đặt về 0 nên ngõ ra  $\overline{Q}$  của FF lên 1, mở cổng AND cho xung  $C_K$  vào mạch đếm. Ngõ ra mạch đếm tăng dần theo dạng nấc thang ( $V_{DAC}$ ), đây chính là điện thế tham chiếu, khi  $V_r$  còn nhỏ hơn  $v_a$ , ngõ ra mạch so sánh còn ở mức thấp và  $Q$  vẫn tiếp tục ở mức

cao, nhưng khi  $V_r$  vừa vượt  $v_a$  ngõ ra mạch so sánh lên cao khiến  $Q$  xuống thấp, đóng cổng AND không cho xung  $C_K$  qua và mạch đếm ngưng. Đồng thời ngõ ra  $Q$  lên cao báo kết thúc sự chuyển đổi. Số đếm ở mạch đếm chính là số nhị phân tương ứng với điện thế vào.

Gọi thời gian chuyển đổi là  $t_c$ . Thời gian chuyển đổi tùy thuộc điện thế cần chuyển đổi. Thời gian lâu nhất ứng với điện thế vào bằng trị toàn giai:

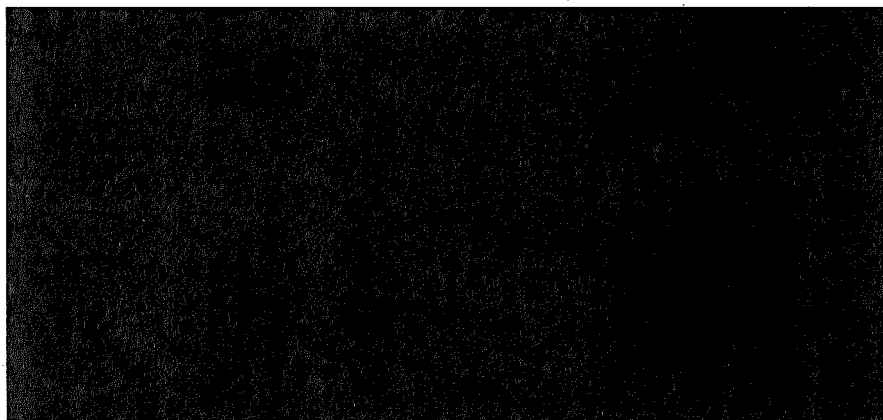
$$t_c(\max) = 2^n / f_{CK} = 2^n \cdot T_{CK}$$

Mạch đổi này có tốc độ chậm. Một cách cải tiến là thay mạch đếm lên bởi một mạch đếm lên/xuống (H 8.9). Nếu ngõ ra mạch so sánh cho thấy  $V_r$  nhỏ hơn  $v_a$ , mạch Logic sẽ điều khiển đếm lên và ngược lại thì mạch sẽ đếm xuống. Nếu  $v_a$  không đổi  $V_r$  sẽ dao động quanh trị  $v_a$  với hai trị số khác nhau 1 LSB



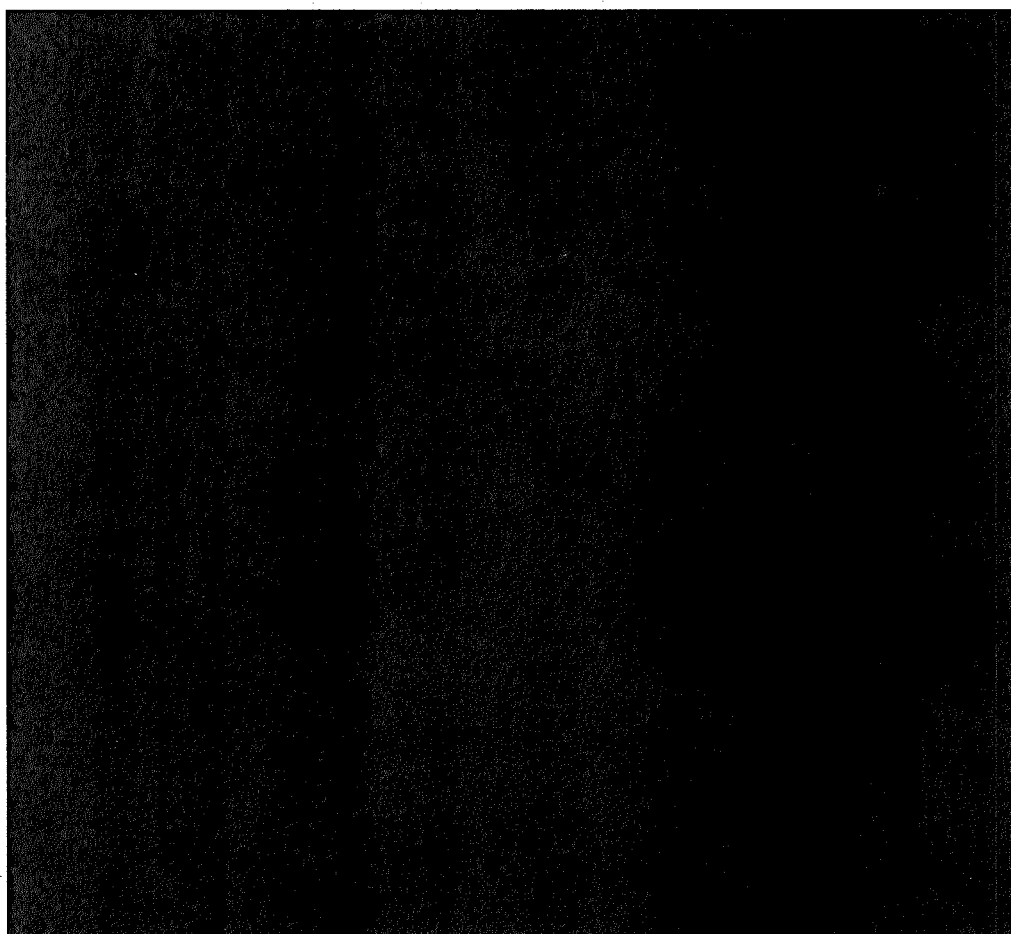
(H 8.9)

### 8.2.4 Mạch đổi lấy gần đúng kế tiếp (successive approximation converter)



(H 8.10)

Mạch đổi lấy gần đúng kế tiếp dùng cách tạo điện thế tham chiếu một cách có hiệu quả hơn khiến việc chuyển đổi ra mã số  $n$  bit chỉ tốn  $n$  chu kỳ xung  $C_K$ . Mạch này bao gồm: một mạch so sánh, một mạch ghi dịch đặc biệt (SAR) và một mạch DAC (H 8.11).



(H 8.11)

Mạch SAR (H 8.11) là mạch ghi dịch có kết hợp điều khiển Logic. Mạch gồm 6 FF D mắc thành chuỗi, ngõ ra FF cuối (F) hồi tiếp về FF đầu (A), khối điều khiển gồm 4 cổng AND và 4 FF RS có ngõ vào tác động mức cao, các ngõ ra Q của các FF RS được đưa vào mạch DAC để tạo điện thế tương tự  $V_r$  (dùng so sánh với điện thế ra từ mạch lấy

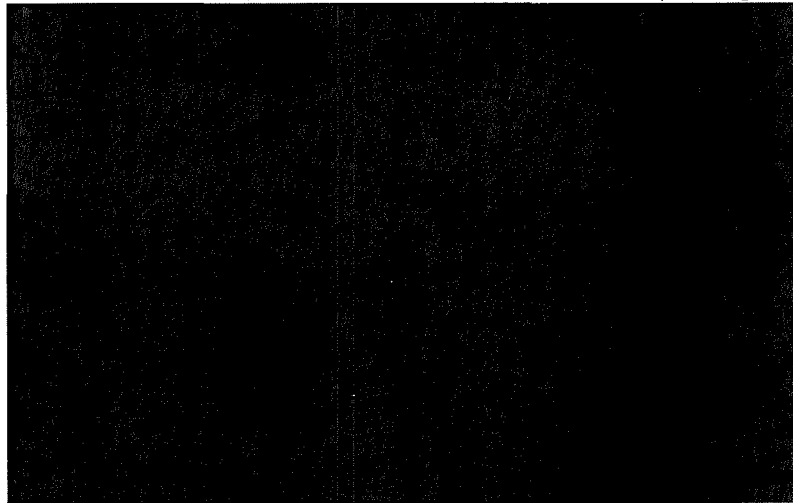
#### 8.2.4 Mạch đổi lấy gần đúng kế tiếp (successive approximation converter)

mẫu và giữ  $v_a$ ), đồng thời đây cũng là mã số ra khi sự biến đổi đã kết thúc.

**Vận hành:** Lúc có xung bắt đầu, mạch SAR được đặt về 0. Ngã ra DAC được làm lệch  $1/2$  LSB để tạo đặc tính chuyển đổi như đã nói trong phần trước, kể đó SAR đưa bit MSB lên cao (bằng cách preset FF A), các bit khác bằng 0, số này được đưa vào mạch DAC để tạo điện thế tham chiếu  $V_r$  để so sánh với  $v_a$ . Tùy theo kết quả so sánh, nếu  $V_r > v_a$  thì ngã ra mạch so sánh ở mức cao khiến SAR bỏ đi bit MSB khi có xung  $C_K$  kế tiếp xuất hiện, còn nếu  $V_r < v_a$  thì ngã ra mạch so sánh ở mức thấp, khiến SAR giữ bit MSB lại (FF RS 4 giữ nguyên trạng thái) đồng thời đưa bit có nghĩa kế tiếp lên cao (do FF 3 được set từ giá trị 1 ở ngã ra FF B, trị 1 này được chuyển từ FF A sang). Mạch so sánh tiếp tục làm việc và kết quả sẽ được quyết định theo cùng cách thức như đối với bit MSB.... Tiếp tục như vậy cho đến bit cuối cùng của SAR, lúc đó  $v_a$  gần  $V_r$  nhất và ta được kết quả chuyển đổi trong thời gian tối đa là  $n$  chu kỳ xung đồng hồ. Mạch chuyển đổi chấm dứt khi ngã ra FF F lên mức cao cho phép mở các đệm để cho mã số ra.

### **Mạch đổi dùng tín hiệu dốc đơn (single ramp converter)**

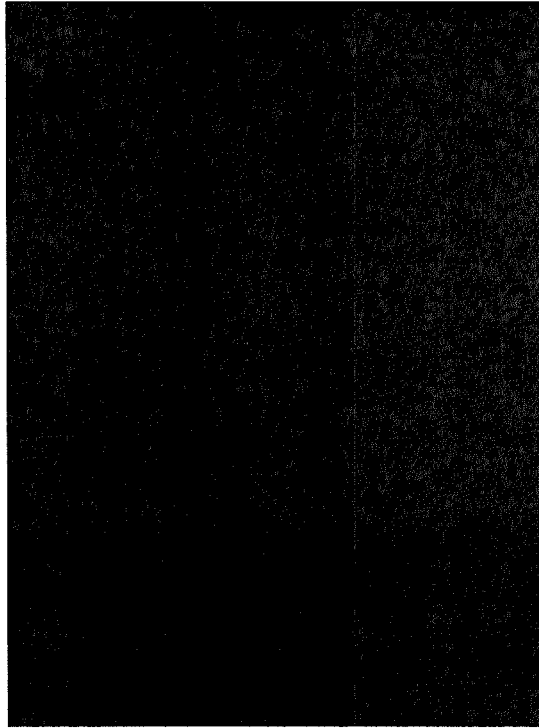
Điện thế chuẩn từng nấc tạo bởi mạch DAC có thể được thay thế bởi điện thế tham chiếu có dốc lên liên tục tạo bởi mạch tạo tín hiệu dốc lên (thường là mạch tích phân).



(H 8.12)

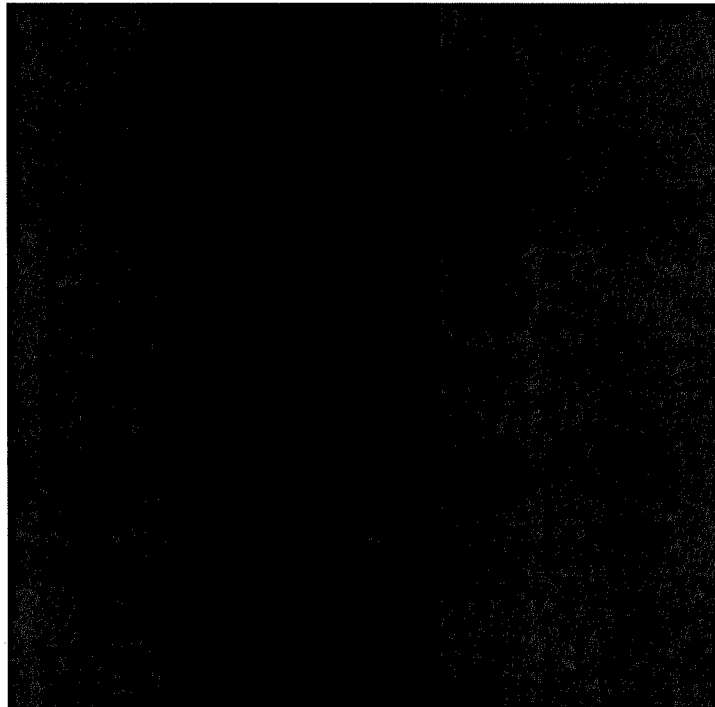
Xung bắt đầu đặt mạch đếm  $n$  bit về 0 và khởi động mạch tạo dốc lên để tạo  $V_r$ , từ một trị hơi âm, khi  $V_r$  cắt trục 0 ngã ra mạch so sánh 2 lên cao mở cổng AND cho xung  $C_K$  vào mạch đếm. Khi đường dốc đạt trị số bằng trị tương tự cần biến đổi ngã ra mạch so sánh 1

lên cao đưa ngã ra  $Q$  của FF xuống thấp, cổng AND đóng và kết thúc sự chuyển đổi. Số đếm được ở mạch đếm tỷ lệ với điện thế tương tự vào. Mạch có khuyết điểm là độ dốc của  $V_r$  tùy thuộc thông số RC của mạch tích phân nên không chính xác.



(H 8.13)

*Mạch đổi lấy tích phân (Integrating Converter)*



(H 8.14)

Mạch này giải quyết được khuyết điểm của mạch biến đổi dùng tín hiệu dốc đơn, nghĩa là độ chính xác không tùy thuộc RC.

Xung bắt đầu đưa mạch đếm về 0, mạch điều khiển mở khóa  $S_3$  của mạch tích phân, đóng khóa  $S_1$  để đưa tín hiệu tương tự  $v_a$  (giả sử âm) vào mạch tích phân đồng thời mở khóa  $S_2$ . Ngã ra mạch tích phân có trị âm nhỏ ban đầu. Tín hiệu tương tự vào được lấy tích phân, độ dốc  $-v_a / RC$ . Khi ngã ra mạch tích phân vượt trục 0, ngã ra mạch so sánh

VIII - 10

lên cao mở công

AND đưa xung  $C_K$  vào mạch đếm. Không kể lượng lệch âm ban đầu, hiệu thế ngã ra mạch tích phân là:

$$V_I(t) = \int -\frac{V_a}{RC} dt$$

Giả sử  $v_a$  không đổi trong thời gian chuyển đổi

$$V_I(t) = -(v_a \cdot t / RC)$$

Nếu  $v_a$  âm thì ngã ra mạch tích phân là đường dốc lên đều.

Khi mạch đếm tràn (tức đếm hết dung lượng và tự động quay về 0) mạch Logic điều khiển mở khóa  $S_1$  và đóng khóa  $S_2$  đưa điện thế tham chiếu  $V_r$  (dương) đến mạch lấy tích phân. Ngã ra mạch tích phân bây giờ là đường dốc xuống với độ dốc là  $-V_r / RC$ . Khi  $V_I$  xuống 0, mạch so sánh xuống thấp đóng cổng AND và kết thúc quá trình biến đổi. Số đếm sau cùng của mạch đếm tỷ lệ với điện thế tương tự vào.

Giả sử  $RC$  không đổi trong quá trình biến đổi, tích phân trong thời gian  $t_1$  bằng tích phân trong thời gian  $t_2$  nên ta có:

$$|v_a| t_1 = V_r \cdot t_2$$

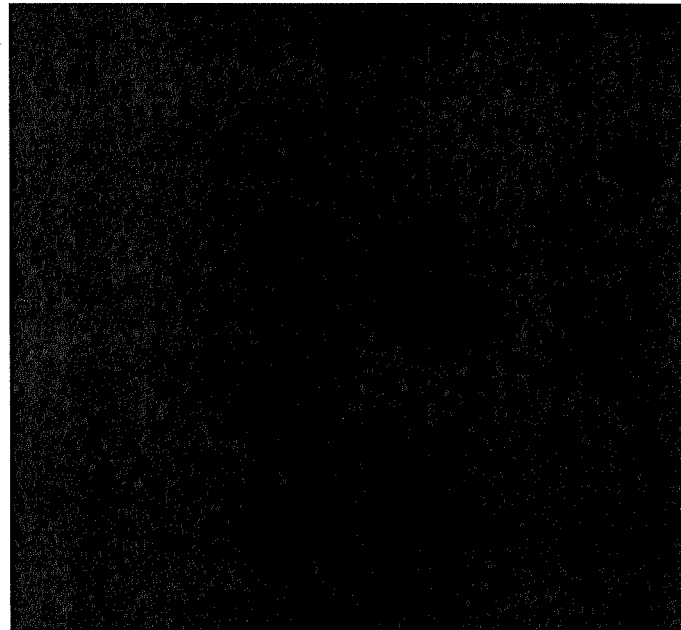
$t_1$  là thời gian đếm từ 0 cho đến khi tràn nên

$$t_1 = 2^n / f_{CK}$$

$$\text{và } t_2 = N / f_{CK}$$

$N$  là số đếm sau cùng.

Tóm lại ta thấy số đếm được không phụ thuộc  $RC$

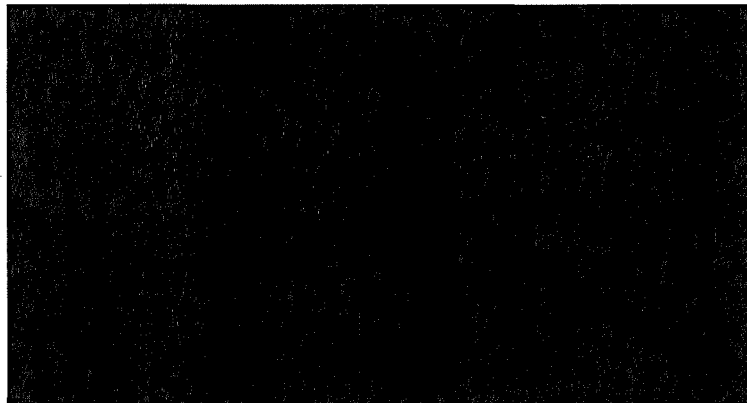


(H 8.15)

### *Mạch đổi lưỡng cực*

Một cách đơn giản để thực hiện chuyển đổi một tín hiệu tương tự lưỡng cực là dùng một mạch đảo tương tự và một mạch so sánh để xác định  $v_a$  âm hay dương để đảo hay không trước khi đưa vào mạch ADC đơn cực (H 8.16)



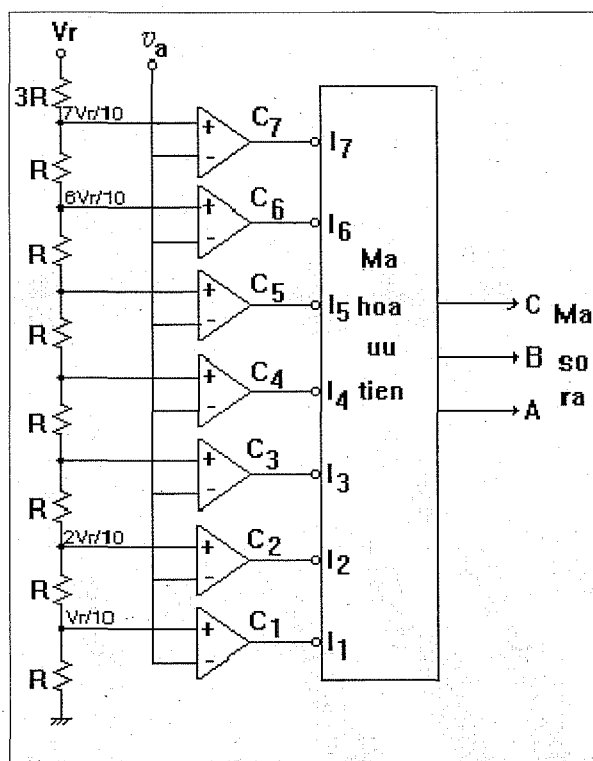


(H 8.16)

**Mạch đổi song song (parallel hay flash conversion)**

Đây là mạch đổi có tốc độ chuyển đổi rất nhanh, có thể đạt vài triệu lần trong một giây, áp dụng vào việc chuyển đổi tín hiệu hình trong kỹ thuật video. Thí dụ để có mạch đổi 3 bit, người ta dùng 7 mạch so sánh ở ngõ vào và một mạch mã hóa ưu tiên để tạo mã số nhị phân ở ngõ ra (H 8.17).

- Khi  $v_a < V_r / 10$ , các ngõ ra mạch so sánh đều lên cao khiến mã số ra là 000
- Khi  $V_r / 10 < v_a < 2V_r / 10$ , ngõ ra mạch so sánh 1 xuống thấp khiến mã số ra là 001
- Khi  $2V_r / 10 < v_a < 3V_r / 10$ , ngõ ra mạch so sánh 2 xuống thấp khiến mã số ra là 010 Cứ như thế, ta thấy mã số ra tỷ lệ với điện thế tương tự vào



(H 8.17)