

**TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP THÁI NGUYÊN**  
**KHOA ĐIỆN TỬ**

-----\*\*\*\*\*-----



# **GIÁO TRÌNH** **ĐIỆN TỬ SỐ**

**BỘ MÔN KỸ THUẬT ĐIỆN TỬ**

Thái Nguyên 2015

# Mục lục

	Trang
<b>Chương 1 Cơ sở đại số logic và các phần tử logic</b>	4
1.1 Đại số boole	4
1.1.1 Một số định nghĩa	4
1.1.2 Biểu diễn biến và hàm logic	4
1.1.3. Các hàm logic cơ bản	5
1.1.4. Một số tính chất của hàm nhân, cộng, phủ định	5
1.2 Biểu diễn giải tích các hàm logic	6
1.2.1 Dạng tuyển chính quy	7
1.2.2 Dạng hội chính quy	8
1.2.3 Biểu diễn tuyển chính quy, hội chính quy dưới dạng số	8
1.3 Tối thiểu hoá các hàm logic	9
1.3.1 Phương pháp đại số	9
1.3.2 Phương pháp bìa các nô	10
1.4 Các hệ thống số đếm thường sử dụng trong kỹ thuật số .	12
1.4.1 Hệ thập phân (Decimal Number System)	12
1.4.2 Hệ nhị phân (Binary number System)	12
1.4.3 Hệ cơ số 16: (Hexa Decimal System).	14
1.4.5 Chuyển đổi giữa các hệ đếm	14
1.4.6 Hệ BCD: (Hệ 2 - 10)	16
1.4.7 Số bù	16
1.5. Các phần tử logic cơ bản	17
1.5.1 Phần tử phủ định logic (phần tử đảo- NOT)	17
1.5.2 Phần tử ‘Và’ (AND)	19
1.5.3 Phần tử ‘Hoặc’ (OR)	21
1.5.4 Phần tử ‘Và - phủ định’ (NAND)	22
1.5.5 Phần tử ‘Hoặc - phủ định’ (NOR)	24
1.6 Một số phần tử logic thông dụng	26
1.6.1 Phần tử tương đương (cùng dấu)	26
1.6.2 Phần tử khác dấu (cộng modul 2)	27
1.7 Các thông số đặc trưng của phần tử logic	28

1.7.1	Các đặc trưng kỹ thuật tiêu biểu của những họ TTL	28
1.7.2	Thiết kế mạch với IC họ 74xxx	30
1.7.3	Các đặc trưng kỹ thuật tiêu biểu của những họ CMOS	31
1.7.4	Những chú ý cần thiết khi thiết kế mạch với IC CMOS	33
	<b>Chương 2 Các mạch số học</b>	35
2.1	Bộ cộng	35
2.1.1	Phần tử nửa tổng	35
2.1.2	Phần tử toàn tổng	35
2.1.3	Phép cộng hai số nhị phân n bit	36
2.1.4	Bộ cộng song song tính trước số nhớ	37
2.2	Bộ trừ	38
2.2.1	Bộ nửa trừ (bán hiệu)	38
2.2.2	Bộ trừ đầy đủ	39
2.2.3	Phép trừ hai số nhị phân n bit	40
2.2.4	Dùng phần tử toàn tổng thực hiện phép trừ hai số nhị phân	40
2.3	Bộ nhân	41
2.4	Bộ so sánh	42
	<b>Chương 3 Các mạch Trigo</b>	45
3.1	Trigo R-S	45
3.1.1	Trigo R-S không đồng bộ:	45
3.1.2	Trigo R-S đồng bộ	48
3.2	Trigo D (Delay)	48
3.3	Trigo chính – phụ (Trigo M - S)	50
3.4	Trigo vạn năng J-K	51
3.5	Trigo đếm T	54
3.6	Chuyển đổi Trigo vạn năng J-K thành các loại Trigo khác.	55
3.6.1	Chuyển đổi Trigo J-K thành Trigo R-S	55
3.6.2	Chuyển đổi Trigo J-K thành Trigo D	55
3.6.3	Chuyển đổi Trigo J-K thành Trigo đếm T	56
	<b>Chương 4 Bộ đếm</b>	57
4.1	Các bộ đếm nhị phân	57
4.1.1	Bộ đếm nhị phân kiểu nối tiếp (không đồng bộ)	58
4.1.2	Bộ đếm nhị phân kiểu song song (đồng bộ)	59
4.2	Bộ đếm nhị phân có mô đun đếm bất kỳ	61

4.3 Bộ đếm nhị phân ngược (trừ xung)	67
4.4 Bộ đếm nhị phân thuận ngược	68
<b>Chương 5 Một số hệ logic thông dụng</b>	71
5.1 Bộ ghi dịch	71
5.2 Bộ biến đổi mã và giải mã	74
5.2.1 Bộ biến đổi mã nhị phân sang mã "1 từ n"	74
5.2.2 Bộ giải mã nhị phân BCD - mã thập phân 7 dấu	77
5.3 Bộ chọn kênh, phân kênh	79
5.3.1 Bộ chọn kênh	79
5.3.2 Bộ phân kênh	83
<b>Chương 6 Bộ nhớ</b>	85
6.1 Các khái niệm	85
6.2 Bộ nhớ RAM	87
6.2.1 Bộ nhớ RAM dùng Tranzitor lưỡng cực	88
6.2.2 Bộ nhớ RAM dùng MOSFET	90
6.3 Các bộ nhớ ROM	102
6.3.1. Bộ nhớ ROM dùng điốt	102
6.3.2. Bộ nhớ ROM dùng tụ điện	104
6.3.3 Bộ nhớ ROM dùng Tranzitor lưỡng cực	104
6.3.4 Bộ nhớ ROM dùng MOSFET	105
6.3.5 Các bộ nhớ PROM, EPROM và EAROM	105
6.3.6 Ứng dụng của bộ nhớ ROM	109
<b>Chương 7 Biến đổi số - tương tự và biến đổi tương tự - số</b>	110
7.1 Giới thiệu chung	110
7.2 Bộ biến đổi số – tương tự (DAC)	110
7.2.1 Bộ biến đổi DAC điện trở hình T	111
7.2.2 Các chỉ tiêu kỹ thuật chủ yếu của bộ biến đổi DAC	116
7.2.3. Ví dụ IC 1 chip DAC	118
7.3 Bộ biến đổi tương tự – số (ADC)	121
7.3.1 Các bước chuyển đổi AD và định lý lấy mẫu	121
7.3.2 Bộ biến đổi ADC xấp xỉ tiệm cận (successive-approimation ADC)	124
7.3.4. Các chỉ tiêu kỹ thuật của ADC	126
7.3.5 Bộ biến đổi ADC 8 bit 0809	127

## Chương 1

### Cơ sở đại số logic và các phần tử logic

#### 1.1. Cơ sở đại số logic

##### 1.1.1 Cơ sở đại số logic

###### a. Đại số Boole

Trong mạch số các tín hiệu thường cho ở hai mức điện áp 0(v) và 5(v). những linh kiện điện tử dùng trong mạch số làm việc ở một trong hai trạng thái (tắt hoặc thông). Do vậy để mô tả mạch số người ta dùng hệ nhị phân (Binary) hai trạng thái trong mạch được mã hoá tương ứng là "1" hoặc "0". Hệ nhị phân thể hiện được trạng thái vật lý mà hệ thập phân không thể hiện được. Môn đại số mang tên người sáng lập ra nó - Đại số Boole còn được gọi là đại số logic.

###### b. Một số định nghĩa:

- Biến logic: Đại lượng biểu diễn bằng ký hiệu nào đó chỉ lấy giá trị "1" hoặc "0".

- Hàm logic: Biểu diễn nhóm các biến logic liên hệ với nhau thông qua các phép toán logic, một hàm logic cho dù là đơn giản hay phức tạp cũng chỉ nhận giá trị hoặc là "1" hoặc là "0".

- Các phép toán logic: có 3 phép toán cơ bản.

Phép nhân (và) - kí hiệu là AND.

Phép cộng (hoặc) - kí hiệu là OR.

Phép phủ định (đảo) - kí hiệu là NOT

##### 1.1.2 Các phép toán, các qui tắc, các định luật, các định lý trong đại số logic

- Tồn tại phần tử trung tính duy nhất cho phép "nhân", phép "cộng".

$A + 0 = A$ ;     0 - Phần tử trung tính cho phép tính "cộng".

$A.1 = A$ ;     1 - Phần tử trung tính cho phép "nhân".

- Hoán vị:  $A + B = B + A$ ;      $A. B = B. A$ .

- Kết hợp  $(A + B) + C = A + (B + C) = (A + C) + B$

$(A. B). C = A. (B. C) = (A. C). B$

- Phân phối :  $A.(B + C) = A.B + A.C$

- Không có số mũ, không có hệ số.

$$A + A + \dots + A = A ; \quad A.A \dots A = A.$$

- Bù :  $\overline{\overline{A}} = A ; \quad A + \overline{A} = 1 ; \quad A.\overline{A} = 0$

\* **Định lý Demorgan:**

Trường hợp tổng quát :  $\overline{f[x_i, \bullet, +]} = f[\overline{x_i}, +, \bullet]$

Thí dụ:  $\overline{X + Y} = \overline{X} . \overline{Y} ; \quad \overline{X . Y} = \overline{X} + \overline{Y}$

(Đảo của một tổng bằng tích các đảo, đảo của một tích bằng tổng các đảo)

- Đối ngẫu  $+ \leftrightarrow \bullet ; \quad 0 \leftrightarrow 1$

Thí dụ:  $A + B = B + A \leftrightarrow A.B = B.A$

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

$$A + 1 = 1 \leftrightarrow A.0 = 0.$$

## 1.2. Các phương pháp biểu diễn hàm logic

### 1.2.1 Bảng thật, bảng trạng thái:

\*Bảng thật : Quan hệ hàm ra với biến vào ở thời điểm hiện tại.

A	B	f(A,B)
0	0	0
0	1	1
1	0	1
1	1	1

Bảng thật  
 $f(A,B) = A + B$

\* Bảng trạng thái: Hàm ra không những phụ thuộc vào biến vào ở thời điểm hiện tại mà còn phụ thuộc vào (trạng thái) quá khứ của nó.

Giả sử có n biến thì bảng có (n+1) cột và  $2^n$  hàng

+ (n+1) cột  $\rightarrow$  (n) biến + (1) giá trị hàm.

+  $2^n$  hàng  $\rightarrow 2^n$  tổ hợp giá trị biến.

Ví dụ: Hàm có hai biến - bảng thật gồm có 3 cột, 4 hàng.

A	B	f(A,B)
0	0	0
0	1	1
1	0	1
1	1	1

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

### 1.2.2 Bìa Karnaugh (Các nô).

Biểu diễn tương đương bảng thật. Mỗi dòng của bảng thật ứng với một ô của bìa các nô. Toạ độ của ô được quy định bởi giá trị tổ hợp biến, giá trị của hàm tương ứng với tổ hợp biến được ghi trong ô đó.

A \ B	0	1
0	0	1
1	1	1

Bìa các nô  
 $f(A,B) = A + B$

### 1.2.3 Biểu diễn giải tích các hàm logic

Với các kí hiệu hàm, biến và các phép tính giữa chúng. Có hai dạng giải tích được sử dụng là.

- + Dạng tuyển: Hàm được cho dưới dạng tổng của tích các biến.
- + Dạng hội: Hàm được cho dưới dạng tích của tổng các biến.
- + Dạng tuyển chính quy: Nếu mỗi số hạng chứa đầy đủ mặt các biến.
- + Dạng tuyển không chính quy: Chỉ cần ít nhất một số hạng chứa không đầy đủ mặt các biến.
- + Hội chính quy: Nếu mỗi thừa số chứa đầy đủ mặt các biến.
- + Hội không chính quy: chỉ cần ít nhất một thừa số không chứa đầy đủ mặt các biến.

Thí dụ:  $f(X,Y,Z) = \bar{X}.\bar{Y}.\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}YZ + XYZ$  (tuyển chính quy)

$f(X,Y,Z) = \bar{X}.\bar{Y} + \bar{X}\bar{Y}Z + \bar{X}YZ + XZ$  (tuyển không chính quy)

$f(x,y,z) = (X + Y + Z).(X + \bar{Y} + Z).(\bar{X} + \bar{Y} + \bar{Z}).$  (hội chính quy).

$$f(x,y,z) = (X + Y + Z).(Y + Z).(Z + \bar{Y} + \bar{X}). \quad (\text{hội không chính quy}).$$

**a. Phương pháp biểu diễn dạng tuyển chính quy:**

- Chỉ quan tâm đến các tổ hợp biến tại đó hàm thành phần nhận trị "1".
- Số số hạng bằng số lần hàm thành phần nhận trị "1".
- Trong biểu thức logic các biến nhận trị "1" giữ nguyên, biến nhận trị "0" ta lấy phủ định.

STT	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

Thí dụ: Cho hàm 3 biến có bảng thật như hình trên thì:

$$Z = f(A,B,C) = \bar{A}.\bar{B}.C + \bar{A}.B.\bar{C} + \bar{A}.B.C + A.\bar{B}.C + A.B.C$$

**b. Phương pháp biểu diễn dạng hội chính quy:**

- Chỉ quan tâm đến các tổ hợp biến tại đó hàm thành phần nhận trị "0".
- Số thừa số bằng số lần hàm thành phần nhận trị "0".
- Trong biểu thức logic các biến nhận trị "0" giữ nguyên, các biến nhận trị "1" ta lấy phủ định.

Với bảng thật trên thì:

$$Z = f(A,B,C) = [A + B + C].[\bar{A} + \bar{B} + \bar{C}].[\bar{A} + \bar{B} + C].$$

**c. Biểu diễn tuyển chính quy, hội chính quy dưới dạng số:**

**- Tuyển chính quy dạng số.**

Từ thí dụ trước tuyển chính quy dạng số được cho:

$$Z = F(A, B, C) = \sum m(1,2,3,5,7)$$



(Tại các giá trị tổ hợp 1, 2, 3, 5, 7 của biến vào hàm nhận trị "1")

**- Hội chính quy dạng số:**

Cũng từ thí dụ trên hội chính quy dạng số được cho như sau:

$$Z = F(a,b,c) = \prod M(0,4,6).$$

(tại các tổ hợp biến 0, 4, 6 hàm logic nhận trị "0" )

### 1.2.4 Các hệ thống số đếm thường sử dụng trong kỹ thuật số.

#### a. Hệ thập phân (Decimal Number System)

\* **Cấu tạo** : Dùng 10 chữ số 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 để biểu diễn con số đếm và tính toán.

\* **Tính chất**: Khi có một số đếm được viết bởi hệ cơ số 10 mà có 2 chữ số bất kỳ kề nhau và giống nhau thì chữ số bên trái có giá trị gấp 10 lần chữ số bên phải.

\* **Khả năng thao tác toán học**: có thể thực hiện mọi phép toán +, -, \*, ÷ các phép toán đại số.

\* Mọi số đếm hệ 10 đều có thể tách thành tổng các số lũy thừa cơ số 10.

Ví dụ:

$$[2452,34]_D = [2 \cdot 10^3 + 4 \cdot 10^2 + 5 \cdot 10^1 + 2 \cdot 10^0 + 3 \cdot 10^{-1} + 4 \cdot 10^{-2}]_D$$

#### b. Hệ nhị phân (Binary number System)

\* **Cấu tạo**: Chỉ dùng hai chữ số "0" và "1" để biểu diễn số đếm và tính toán.

\* **Tính chất**: Khi có một số đếm được viết bởi hệ cơ số 2 có hai chữ số bất kỳ kề nhau và giống nhau thì chữ số bên trái có giá trị gấp 2 lần chữ số bên phải.

\* **Khả năng thao tác toán học**: Rất thích ứng với các phép toán trong đại số logic.

\* **Mọi số đếm hệ cơ số 2** (nhị phân) đều có thể tách thành tổng các số lũy thừa cơ số 2.

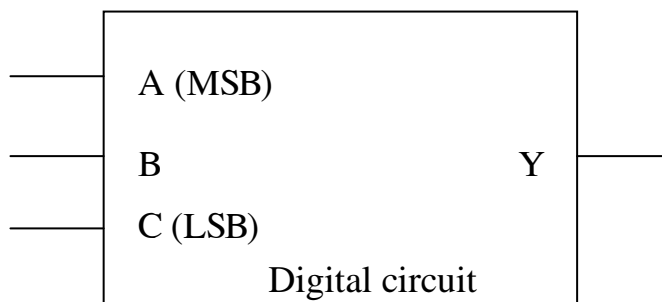
Ví dụ:

$$[10101]_2 = [1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0]_{10} = 1 \cdot 16 + 0 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 21$$

\* **Bit có trọng số lớn nhất MSB ( Most Significant Bit):** Bit nằm về phía ngoài cùng bên trái trong dãy bit nhị phân. Bit này là bit có trọng số lớn nhất.

\* **Bit có trọng số lớn nhất LSB ( Low Significant Bit):** Bit nằm về phía ngoài cùng bên phải trong dãy bit nhị phân. Bit này là bit có trọng số nhỏ nhất.

Ví dụ: một mạch số trong hộp đen có 3 đầu vào, mỗi đầu vào có 2 trạng thái cao và thấp. Trong đó A là bit MSB và C là bit LSB. Lập bảng trạng thái mô tả cho mạch số trên



Bảng trạng thái:

Logic Level			Binary Value			Decimal Equivalent
<i>A</i>	<i>B</i>	<i>C</i>	<i>A</i>	<i>B</i>	<i>C</i>	
L	L	L	0	0	0	0
L	L	H	0	0	1	1
L	H	L	0	1	0	2
L	H	H	0	1	1	3
H	L	L	1	0	0	4
H	L	H	1	0	1	5
H	H	L	1	1	0	6
H	H	H	1	1	1	7

\* **Ảnh hưởng của sự thay đổi MSB và LSB**

	<i>A</i>	<i>B</i>	<i>C</i>	Decimal	
<b>Original</b>	0	1	1	3	
<b>Change MSB</b>	1	1	1	7	Difference = 4
<b>Change LSB</b>	0	1	0	2	Difference = 1

\* **Phép cộng hai số nhị phân:**

$$0 + 1 = 1.$$

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$1 + 1 = 0 \rightarrow$  nhớ 1 gửi lên phép cộng bit cao hơn tiếp theo.

$1 + 1 + 1 = 1 \rightarrow$  nhớ 1 gửi lên phép cộng bit cao hơn tiếp theo

Thí dụ : Số nhớ 0010 1110

Số hạng thứ nhất 0101 0111

Số hạng thứ hai 1001 0101

Tổng 1110 1100

**\* Phép trừ hai số nhị phân:**

$$0 - 0 = 0.$$

$$1 - 1 = 0.$$

$$1 - 0 = 1.$$

$0 - 1 = 1 \rightarrow$  nhớ 1 (mượn) gửi lên phép trừ bit cao hơn tiếp theo

$1 - 1 - 1 = 1 \rightarrow$  nhớ 1 (mượn) gửi lên phép trừ bit cao hơn tiếp theo

\* Thí dụ: Số mượn 1 1 0 0 1. 0 0

Số bị trừ 1 0 0 1 1. 0 1

Số trừ 1 1 0 0. 1 0

Hiệu số 0 0 1 1 0. 1 1

**\* Phép nhân hai số nhị phân:**

$$0.0 = 0.$$

$$1.0 = 0$$

$$0.1 = 0$$

$$1.1 = 1$$

$$\begin{array}{r} 101 \\ \times 110 \\ \hline 000 \\ + 101 \\ \hline 101 \\ \hline 11110 \end{array}$$

**\* Phép chia hai số nhị phân:**

Ví dụ:  $35/5 = 7$

$$\begin{array}{r} 100011 \\ \underline{000} \\ 1000 \\ \underline{101} \\ 111 \\ \underline{101} \\ 101 \\ \underline{101} \\ 0 \end{array} \quad \begin{array}{r} 101 \\ \hline 0111 \end{array}$$

**\* Định nghĩa bit:**

Một chữ số hệ cơ số 2 là 1 bit:

$0_{(B)} \rightarrow 1 \text{ bit}$  ;  $1_{(B)} \rightarrow 1 \text{ bit}$  ;  $10_{(B)} \rightarrow 2 \text{ bit}$  ;  $101_{(B)} \rightarrow 3 \text{ bit}$

**\* Định nghĩa Byte:**

8 bit nối tiếp tạo nên một Byte.

**\* Chuyển đổi thập phân thành nhị phân****- Tổng các lũy thừa của 2**

Bạn có thể chuyển đổi các số nhị phân thành tổng các lũy thừa của 2.

Ví dụ: Chuyển đổi số 57 trong hệ thập phân thành số nhị phân.

32	16	8	4	2	1	
1						$57 - 32 = 25$
32	16	8	4	2	1	
1	1					$57 - (32 + 16) = 9$
32	16	8	4	2	1	
1	1	1				$57 - (32 + 16 + 8) = 1$
32	16	8	4	2	1	
1	1	1	0	0	1	$57 - (32 + 16 + 8 + 1) = 0$

$57_{10} = 111001_2$

**- Lặp lại các bước chia cho 2**

$$46/2 = 23 + \text{còn lại } 0 \text{ (LSB)}$$

$$23/2 = 11 + \text{còn lại } 1$$

Phép chia tiếp tục được thực hiện cho tới khi số chia được là 0.

$$11/2 = 5 + \text{còn lại } 1$$

$$5/2 = 2 + \text{còn lại } 1$$

$$2/2 = 1 + \text{còn lại } 0$$

$$1/2 = 0 + \text{còn lại } 1$$

Kết quả sẽ là  $46_{10} = 101110_2$

**c. Hệ cơ số 16 (Hexa Decimal System).****\*Cấu tạo.**

Dùng 16 ký hiệu : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F để biểu diễn con số đếm và tính toán.

Hex	Decimal	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

**\* Tính chất:**

Khi có một số đếm được viết bởi hệ cơ số 16 mà có 2 chữ số bất kỳ kề nhau và giống nhau thì chữ số bên trái có giá trị gấp 16 lần chữ số bên phải.

**\* Khả năng thao tác toán học:**

Thực hiện mọi phép toán +, -, \*, ÷ và các phép toán logic.

**\* Chuyển đổi hệ Hexa sang hệ thập phân**

Ví dụ :

$$[3A2F]_{16} = [3.16^3 + 10.16^2 + 2.16^1 + 15.16^0]_{10}.$$

**\* Chuyển đổi hệ thập phân sang hệ Hexa****- Tổng các lũy thừa của 16**

Ví dụ: chuyển đổi số  $35_{10}$  sang hệ hexa

$$35 = 32 + 3 = 2 \cdot 16^1 + 3 \cdot 16^0 = 23_H$$

**- Lặp lại các bước chia cho 16**

Ví dụ: chuyển đổi số  $31581_{10}$  sang hệ hexa

$$31581/16 = 1973 + \text{còn lại } 13 \text{ (LSB)}$$

$$1973/16 = 123 + \text{còn lại } 5$$

$$123/16 = 7 + \text{còn lại } 11$$

$$7/16 = 0 + \text{còn lại } 7 \text{ (MSB)}$$

Kết quả:  $31581_{10} = 7B5D_{16}$

**\*Ý nghĩa của hệ 16:**

Có khả năng biểu diễn, tính toán rất gọn vì 1 chữ số trong hệ 16 tương ứng với 4 bit nhị phân.

**d. Hệ BCD: (Hệ 2 - 10)**

Các số nhị phân có độ dài (số bit) lớn hơn 4 thường khó đọc. Theo thói quen thập phân thông thường người ta dùng mã nhị - thập phân (BCD). Dùng 4 bit nhị phân để diễn tả một chữ số hệ BCD, giá trị thập phân của 4 bit nhị phân này không lớn hơn 9.

Thí dụ:  $[9]_{10} \Leftrightarrow [1\ 0\ 0\ 1]_{\text{BCD}}$ .  $[19]_{10} \Leftrightarrow [0\ 0\ 0\ 1\ 1\ 0\ 0\ 1]_{\text{BCD}}$ .

$[1999]_{10} \Leftrightarrow [0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1]_{\text{BCD}}$ .

**1.4.7 Số bù:****a. Số bù 1.**

Trong hệ cơ số 2 khi cho trước một số, tìm được một số khác sao cho tổng của số đã cho và số vừa tìm được có các hàng đều bằng 1 thì số vừa tìm được gọi là số bù 1 của số đã cho.

Ví dụ:  $*[0]_2 \rightarrow$  Số bù 1 của nó là số  $[1]_2$ .

Vì  $\begin{array}{r} 0 \\ + \\ 1 \\ \hline 1 \end{array}$

và ngược lại số  $[1]_2$  thì số bù của nó là số  $[0]_2$

\* Cho số  $[101]_2 \rightarrow$  tìm số bù 1 của số đã cho.

$\begin{array}{r} 1\ 0\ 1 \\ + \\ \hline \end{array}$

$\begin{array}{r} x\ x\ x \\ 1\ 1\ 1 \\ \hline \end{array} \rightarrow$  Số bù 1 của số  $[101]_2$  là  $[0\ 1\ 0]_2$

Nhận xét: Số bù 1 là số đảo của số đã cho.

## b. Số bù 2

Trong hệ cơ số 2 nếu cho trước một số, tìm được một số khác sao cho tổng của số đã cho với số tìm được có kết quả của các hàng đều bằng 0 nhưng có 1 hàng vượt ra ngoài nhận giá trị 1. Số tìm được gọi là số bù 2 của số đã cho.

Thí dụ: Cho số  $[0]_2$  ta tìm số bù 2 của nó.

$$+ 0 \rightarrow x = [1\ 0]_2 ; [1\ 0]_2 \text{ là số bù 2 của } [0]_2$$

$$\begin{array}{r} \underline{x.} \\ 1\ 0 \end{array}$$

- Cho số  $[1]_2$  ta tìm số bù 2 của nó.

$$+ 1 \rightarrow x = [1]_2 ; [1]_2 \text{ là số bù 2 của } [1]_2.$$

$$\begin{array}{r} \underline{x.} \\ 1\ 0 \end{array}$$

Nhận xét: Số bù 2 của một số nhị phân chính bằng số bù 1 của nó cộng thêm 1.

## 1.3. Các phương pháp tối thiểu hoá hàm logic

Một hàm logic có thể có vô số cách biểu diễn giải tích tương đương. Tuy nhiên chỉ tồn tại 1 cách gọn nhất tối ưu về số biến, số số hạng hay thừa số và được gọi là tối giản. việc tối giản hàm logic mang ý nghĩa quan trọng về phương diện kinh tế, kỹ thuật. Để tối thiểu hoá các hàm logic người ta thường dùng phương pháp đại số và phương pháp bìa các nô.

### 1.3.1 Phương pháp đại số:

Biến đổi biểu thức logic dựa vào các tính chất của đại số **Boole**.

$$\text{Thí dụ : } A.B + \bar{A}.B = B ; \quad A + A.B = A ; \quad A + \bar{A}.B = A + B.$$

Ta chứng minh các đẳng thức trên, theo tính chất đối ngẫu:

$$A.B + \bar{A}.B = B \Leftrightarrow (A + B).(\bar{A} + B) = B.$$

$$A + A.B = A \Leftrightarrow A.(A + B) = A.$$

$$A + \bar{A}.B = A + B \Leftrightarrow A.(\bar{A} + B) = A.B.$$

### Quy tắc 1:

Nhóm các số hạng có thừa số chung.

$$\text{Thí dụ: } A.B.C + A.B.\bar{C} = A.B(C + \bar{C}) = A.B.$$

### Quy tắc 2:

Đưa số hạng đã có vào biểu thức logic.

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

### Quy tắc 3:

Có thể loại các số hạng thừa.

$$\begin{aligned} A.B + \bar{B}.C + A.C &= A.B + \bar{B}.C + A.C (B + \bar{B}). \\ &= A.B + \bar{B}.C + A.B.C + A.\bar{B}.C \\ &= A.B + \bar{B}.C \text{ (loại } A.C) \end{aligned}$$

## 1.3.2 Phương pháp bìa các nô.

### a) Cấu tạo:

- Gồm 1 đồ hình các ô vuông, hàm có n biến bảng có  $2^n$  ô (1 biến - 2 ô, 2 biến - 4 ô, 3 biến - 8 ô ...).
- Thứ tự của các ô do giá trị tổ hợp biến quy định
- Hai ô được gọi là kề nhau, hoặc đối xứng chỉ khác nhau 1 giá trị của biến.
- Giá trị của hàm tương ứng với tổ hợp biến được ghi ngay trong ô đó.
- Các ô tại đó giá trị của hàm không xác định được đánh bằng dấu "X".

### b) Nguyên tắc tối giản hàm logic trên bìa các nô

- Thực hiện nhóm các ô tại đó hàm nhận trị "1" hoặc "0" kề nhau hoặc đối xứng nhau, số ô trong một nhóm dán phải là số lũy thừa của 2 (khi viết hàm dạng tuyển ta nhóm các ô có giá trị "1", dạng hội nhóm các ô có giá trị "0").
- Trong một nhóm dán các biến có trị thay đổi ta loại, các biến có trị không đổi giữ nguyên, điều này có nghĩa là số ô trong nhóm dán càng nhiều thì số biến bị loại càng tăng (2 ô - loại 1 biến, 4 ô - loại 2 biến ...  $2^m$  ô - loại m biến).
- Biểu thức logic có số số hạng hay thừa số chính bằng số nhóm dán. Khi viết hàm logic dưới dạng tuyển các biến còn lại nhận trị "1" ta giữ nguyên, nhận trị "0" ta lấy phủ định, khi viết hàm logic dưới dạng hội thì ngược lại.
- Một ô có thể tham gia vào nhiều nhóm dán.



- Các ô tại đó giá trị hàm không xác định ta coi tại ô đó hàm có thể lấy giá trị "1" hoặc "0" tùy từng trường hợp cụ thể.

Chú ý: Phương pháp tối giản hàm logic trên bìa các nô chỉ thích hợp với hàm có số biến  $\leq 6$ . Trường hợp hàm có số biến lớn hơn 6, bảng các nô rất phức tạp.

F		BC			
		00	01	11	10
A	0	$\nearrow_0$	$\nearrow_1$	$\nearrow_3$	$\nearrow_2$
	1	$\nearrow_4$	$\nearrow_5$	$\nearrow_7$	$\nearrow_6$

4 cột - 2 hàng (hàm 3 biến)

F		C	
		0	1
AB	00	$\nearrow_0$	$\nearrow_1$
	01	$\nearrow_2$	$\nearrow_3$
	11	$\nearrow_4$	$\nearrow_5$
	10	$\nearrow_6$	$\nearrow_7$

2 cột - 4 hàng (hàm 3 biến)

F		CD			
		00	01	11	10
AB	00	$\nearrow_0$	$\nearrow_1$	$\nearrow_3$	$\nearrow_2$
	01	$\nearrow_4$	$\nearrow_5$	$\nearrow_7$	$\nearrow_6$
	11	$\nearrow_{12}$	$\nearrow_{13}$	$\nearrow_{15}$	$\nearrow_{14}$
	10	$\nearrow_8$	$\nearrow_9$	$\nearrow_{11}$	$\nearrow_{10}$

Hàm 4 biến (4 hàng - 4 cột - 16 ô)

Thí dụ: Cho hàm logic 4 biến  $F(a,b,c,d) = \sum m(0,1,2,4,6,8,9,10)$  và không xác định tại  $m = 5, 11, 13, 15$ . (Thí dụ này tương đương với việc cho hàm logic 4 biến  $F(A,B,C,D) = \prod M(3,7,12,14)$  và không xác định tại  $M = 5, 11, 13, 15$ )

Từ bài ra ta có bảng các nô như sau:

+ Biểu diễn dạng tuyến (3 nhóm dán)

- Nhóm 1: Các ô 0, 2, 8, 14  $\rightarrow$  kết quả:  $\overline{B}.\overline{D}$

- Nhóm 2: Các ô 0, 2, 4, 6  $\rightarrow$  kết quả:  $\overline{A}.\overline{D}$

- Nhóm 3: Các ô 1, 5, 9, 13  $\rightarrow$  kết quả:  $\overline{C}.D$

Hàm biểu diễn dưới dạng tuyến:

$$F(a,b,c,d) = \overline{B}.\overline{D} + \overline{A}.\overline{D} + \overline{C}.D$$

F		CD			
		00	01	11	10
AB	00	1	1	0	1
	01	1	X	0	1
	11	0	X	X	0
	10	1	1	X	1

+ Biểu diễn hàm logic dưới dạng hội (2 nhóm)

- Nhóm 1: Gồm các ô 3, 7, 11, 15 → kết quả:  $\overline{C} + \overline{D}$

- Nhóm 2: Gồm các ô 12, 13, 14, 15 → kết quả:  $\overline{A} + \overline{B}$

Hàm biểu diễn dưới dạng hội:  $F(A,B,C,D) = (\overline{C} + \overline{D}).(\overline{A} + \overline{B})$

## 1.4. Các phần tử logic cơ bản

Các phép toán cơ bản của đại số logic có thể được thực hiện bằng các mạch khoá điện tử (Tranzitor hoặc IC). Nét đặc trưng nhất của các phần tử logic ở đây là hai mức điện thế cao hoặc thấp của mạch khoá hoàn toàn cho một sự tương ứng đơn trị với hai trạng thái của biến hay hàm logic. Nếu sự tương ứng được quy ước là điện thế thấp - trị "0" và điện thế cao - trị "1" ta gọi đó là logic dương. Trong trường hợp ngược lại với quy ước mức thế thấp trị "1" và mức thế cao trị "0", ta có logic âm. Để đơn giản ở đây ta chỉ xét với mức logic dương.

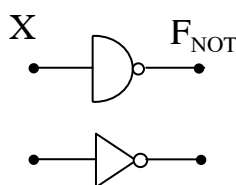
### 1.4.1 Phần tử phủ định logic (phần tử đảo- NOT).

- Phần tử phủ định có 1 đầu vào biến và 1 đầu ra thực hiện hàm phủ định logic:  $F_{\text{NOT}} = \overline{x}$

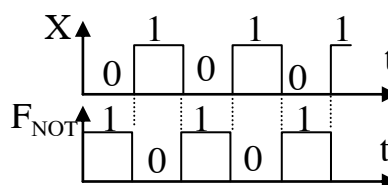
Tức là  $F_{\text{NOT}} = 1$  khi  $x = 0$  hoặc ngược lại  $F_{\text{NOT}} = 0$  khi  $x = 1$ . Bảng trạng thái, kí hiệu quy ước và giản đồ thời gian minh hoạ được cho trên hình 1a, b, c tương ứng:

X	$F_{\text{NOT}}$
0	1
1	0

a)



b)

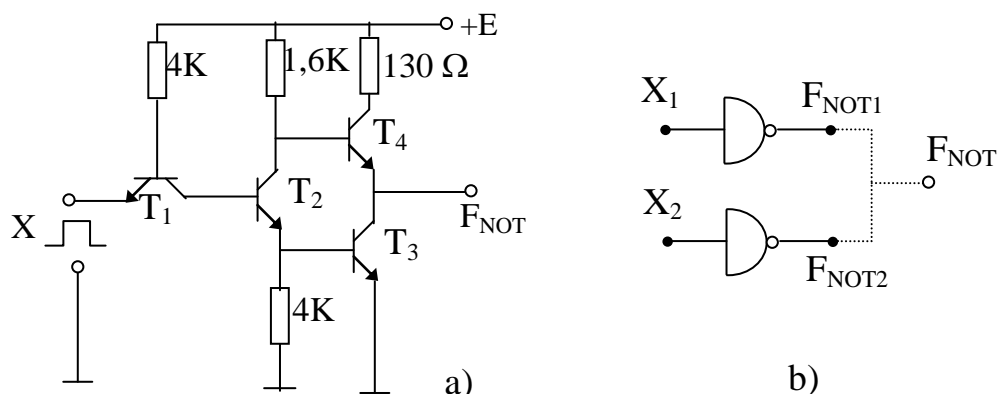


c)

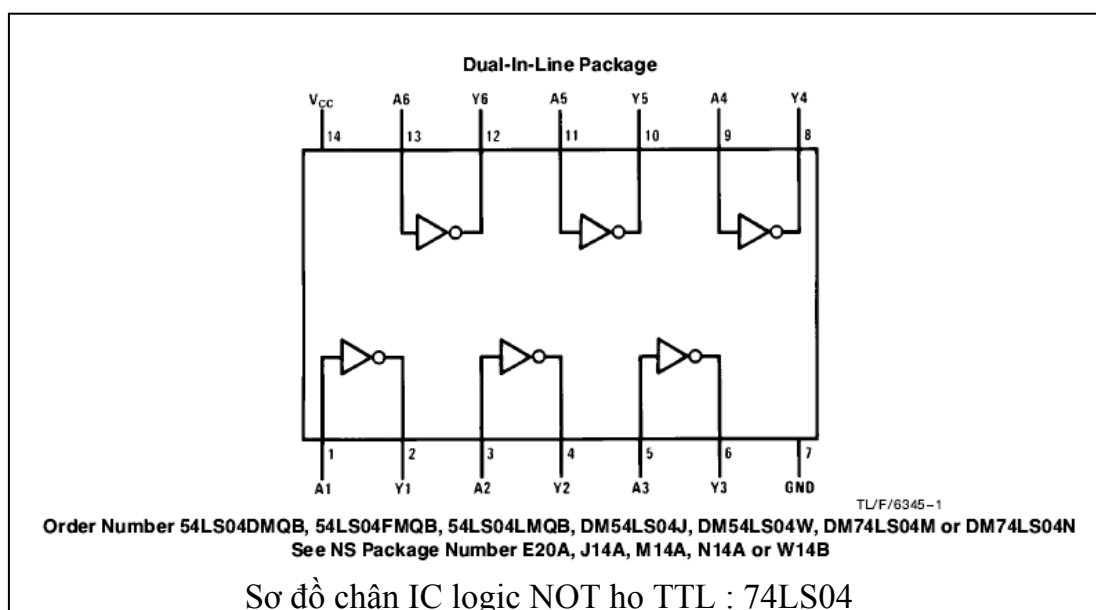
Hình 1: Bảng trạng thái (a); Kí hiệu quy ước (b); Giản đồ điện áp minh hoạ của phần tử NOT (c)

- Để thực hiện hàm  $F_{\text{NOT}}$ , có thể dùng một trong các sơ đồ mạch khoá (Tranzitor hay IC). Dựa trên tính chất đảo pha của một tầng EC đối với Tranzitor hay đối với đầu vào đảo của IC thuật toán. Mạch thực tế có phức tạp hơn để nâng cao khả năng làm việc tin cậy và chính xác. Hình 2 đưa ra một sơ đồ đảo kiểu TTL(Tranzitor - Tranzitor - logic) hoàn thiện trong 1 vỏ IC số. Mạch ra của sơ đồ gồm 2 Tranzitor  $T_3$  và  $T_4$  làm việc ngược pha nhau (ở chế độ

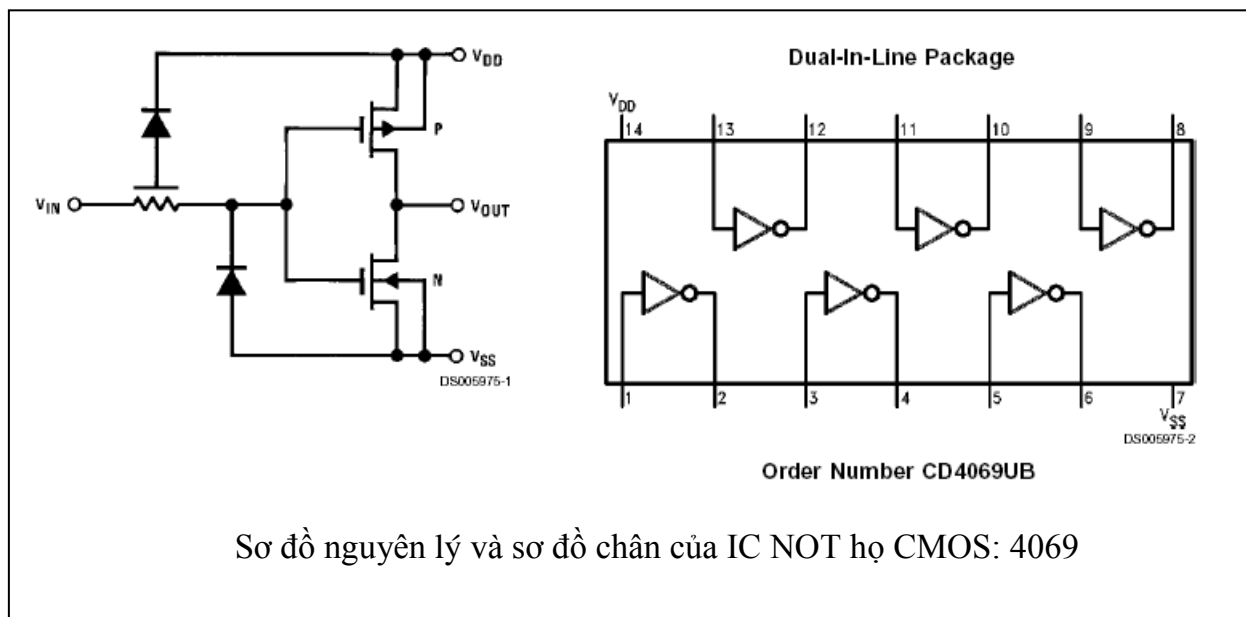
khoá) nhờ tín hiệu lấy trên các lối ra phân tải của  $T_2$ . Mạch vào của sơ đồ dùng Tranzitor  $T_1$  mắc kiểu BC và tín hiệu vào (x) được đưa tới cực emitor của  $T_1$  thể hiện là các xung điện áp cực tính dương (lúc  $x = 1$ ) có biên độ lớn hơn mức  $U_H$  hoặc không có xung (lúc  $x = 0$ ) điều khiển  $T_1$  khoá (lúc  $x = 1$ ) hay mở (lúc  $x = 0$ ). Nghĩa là khi  $x = 0$ ,  $T_1$  mở, điện thế  $U_{C1} = U_{B2}$  ở mức thấp  $\rightarrow T_2$  khoá, điều này làm  $T_3$  khoá (vì  $U_{E2}$  ở mức thấp) và  $T_4$  mở (vì  $U_{C2}$  ở mức cao), kết quả là tại đầu ra, điện thế tại điểm A ở mức cao hay  $F_{NOT} = 1$ . Nhờ  $T_4$  mở mức điện thế tại A được nâng lên xấp xỉ nguồn +E (ưu điểm hơn so với việc dùng điện trở  $R_{C3}$ ) nên  $T_4$  được gọi là Tranzitor "kéo lên", điều này còn làm tăng khả năng chịu tải nhỏ hay dòng lớn cho tầng ra. Khi  $x = 1$ , tình hình sẽ ngược lại  $T_1$  khoá  $T_2$  mở làm  $T_4$  khoá và  $T_3$  mở dẫn tới  $F_{NOT} = 0$ .



Hình 2: Bộ đảo TTL có hai đầu trạng thái kết cấu dưới dạng một vi mạch số (a); kiểu mắc chung sai đầu ra cho hai phần tử NOT (b).



Sơ đồ hình 3 được chế tạo trên công nghệ CMOS và có ưu điểm là dòng tĩnh lỗi vào cũng như lỗi ra gần bằng 0.



**1.4.2 Phần tử "Và" (AND):** Là phần tử có nhiều đầu vào biến và một đầu ra thực hiện hàm nhân logic tức là hàm  $F_{AND}$ .

$$F_{AND} = x_1 x_2 x_3 \dots x_n$$

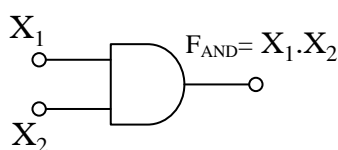
$F_{AND} = 1$  khi và chỉ khi tất cả các biến  $x_i$  nhận trị 1.

$F_{AND} = 0$  khi ít nhất 1 trong các biến  $x_i$  nhận trị 0

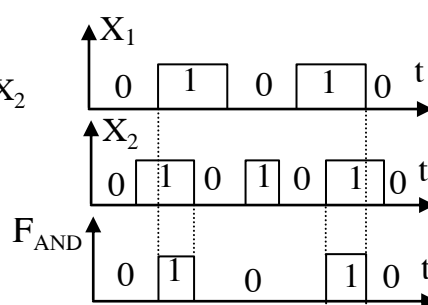
Bảng trạng thái, kí hiệu quy ước và giản đồ thời gian minh hoạ của  $F_{AND}$  cho trên hình 4 (với  $n=2$ ).

$X_1$	$X_2$	$F_{AND}$
0	0	0
0	1	0
1	0	0
1	1	1

a)



b)



c)

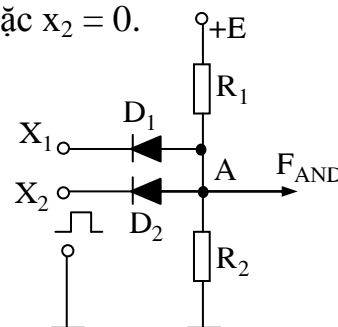
Hình 4: Bảng trạng thái (a), ký hiệu quy ước (b) và giản đồ thời gian (c) của phần tử NAND

Mạch điện thực hiện  $F_{AND}$  loại đơn giản nhất dựa trên các khoá điôt cho trên hình 5, bình thường khi  $x_1 = x_2 = 0$  nhờ E qua phân áp  $R_1 R_2$  có  $U_A > 0$  các điôt  $D_1 D_2$  đều mở, điện áp ra ở mức thấp (cỡ bằng sụt áp thuận của điôt)  $F_{AND}=0$  tình hình trên không thay đổi khi chỉ  $x_1 = 0$  hoặc  $x_2 = 0$ .

Khi  $x_1 = x_2 = 1$  (ứng với trạng thái các đầu vào có xung vuông biên độ lớn hơn  $U_A$ ), các điôt đều khoá các nhánh đầu vào, lúc đó

$$U_A = \frac{E}{R_1 + R_2} \cdot R_2 \text{ ở thế cao } F_{AND} = 1$$

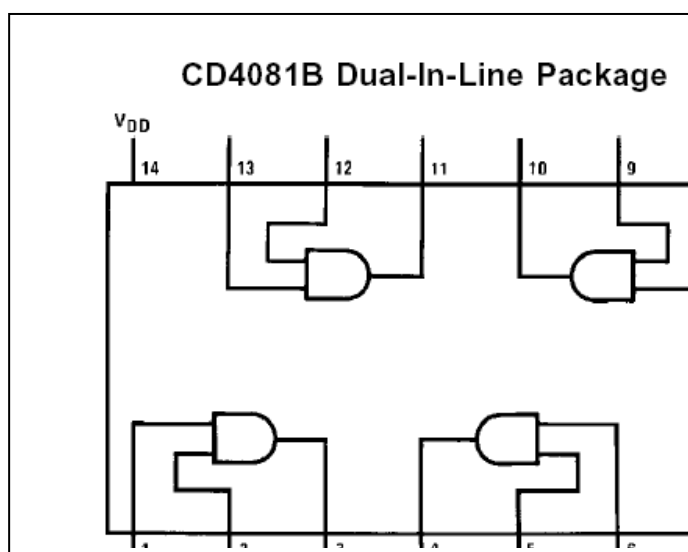
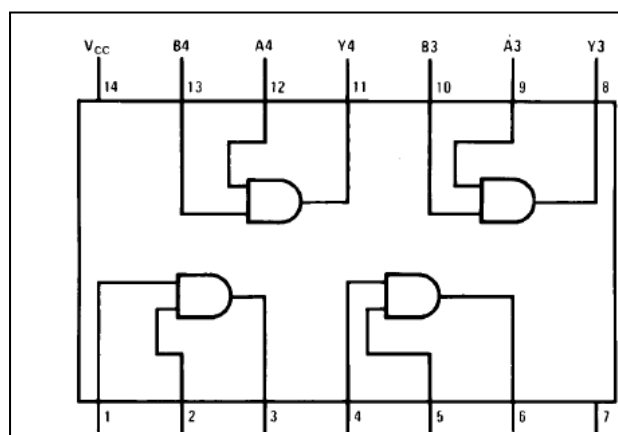
(khi  $R_2 \gg R_1$ )



Hình 5: Sơ đồ nguyên lý mạch AND dựa trên điôt

Lưu ý khi số lượng đầu vào nhiều hơn số biến, các đầu vào không dùng cần được nối với +E để nhánh tương ứng tách khỏi mạch (điôt khoá) tránh được nhiễu với các đầu khác đang làm việc.

Sơ đồ chân IC logic AND  
họ TTL:  
54LS08, 74LS08



Sơ đồ chân IC logic AND họ CMOS : 4081

**1.4.3 Phần tử “Hoặc” (OR):** Là phần tử có nhiều đầu vào biến, một đầu ra thực hiện hàm cộng logic:

$$F_{OR} = x_1 + x_2 + \dots + x_n$$

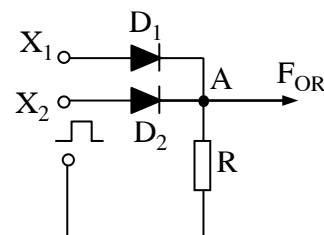
$F_{OR} = 1$  khi ít nhất một trong các biến  $x_i$  nhận trị 1

$F_{OR} = 0$  khi tất cả các biến nhận trị 0:  $x_1 = x_2 = \dots$

$$= x_n = 0$$

Bảng trạng thái kí hiệu quy ước và đồ thị thời gian minh hoạ của  $F_{OR}$  cho trên hình 6 (cho với  $n = 1$ ).

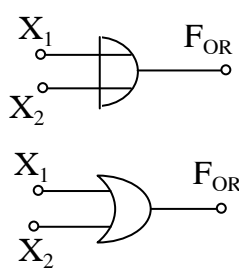
Có thể dùng khoá diôt thực hiện hàm  $F_{OR}$



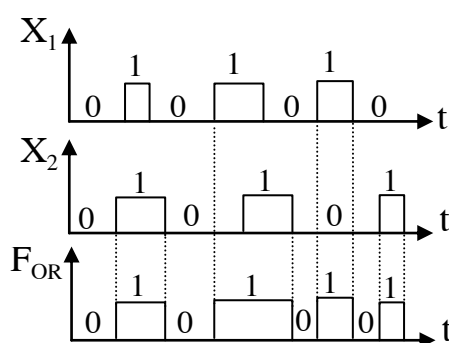
Hình 7: Sơ đồ nguyên lý mạch OR dựa trên diôt

$X_1$	$X_2$	$F_{OR}$
0	0	0
0	1	1
1	0	1
1	1	1

a)



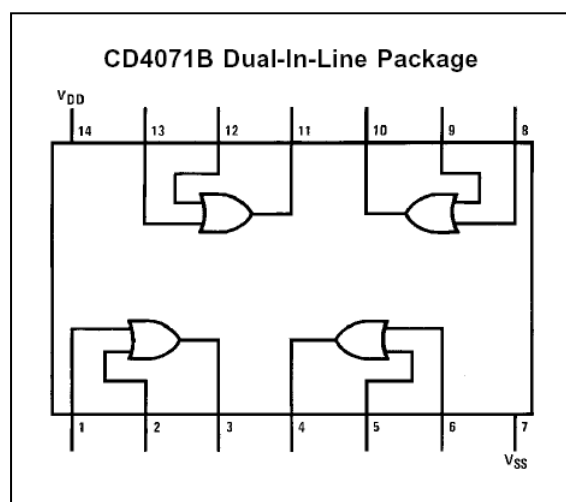
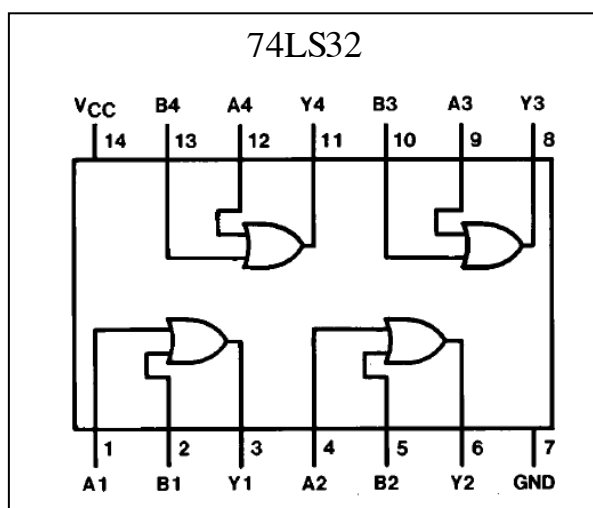
b)



c)

Hình 6: Bảng trạng thái (a); Ký hiệu quy ước (b); và giản đồ thời gian (c) của phần tử OR

Bình thường khi  $x_1 = x_2 = 0$  các diôt đều khoá trên R không có dòng điện  $U_{ra} = 0$ ,  $F_{OR} = 0$ . Khi ít nhất một đầu vào có xung dương diôt tương ứng mở tạo dòng trên R do đó  $U_A$  ở mức cao hay  $F_{OR} = 1$ . Khi đó đầu vào nhiều hơn số biến, đầu vào không dùng được nối mát để chống nhiễu.



Sơ đồ chân IC logic OR họ TTL : 74LS32 và họ CMOS : 4071

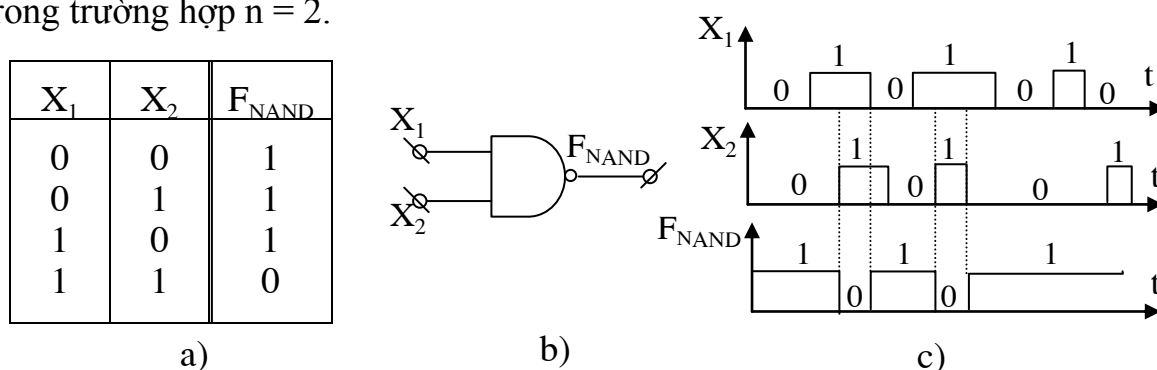
**1.4.4 Phần tử “Và - phủ định” (NAND):** Là phần tử nhiều đầu vào biến, một đầu ra thực hiện hàm logic Và - phủ định:

$$F_{\text{NAND}} = \overline{x_1 x_2 \dots x_n}$$

$F_{\text{NAND}} = 0$  khi tất cả các đầu vào biến có trị 1.

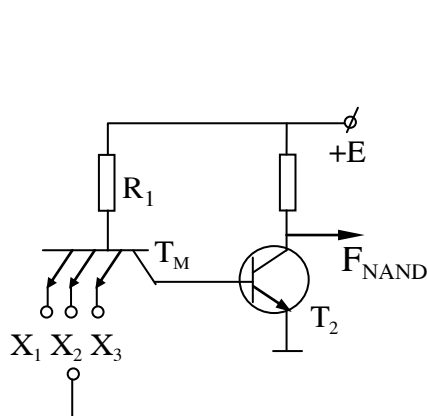
$F_{\text{NAND}} = 1$  trong các trường hợp còn lại.

Hình 8: Đưa ra bảng trạng thái, ký hiệu quy ước và đồ thị thời gian minh họa trong trường hợp  $n = 2$ .

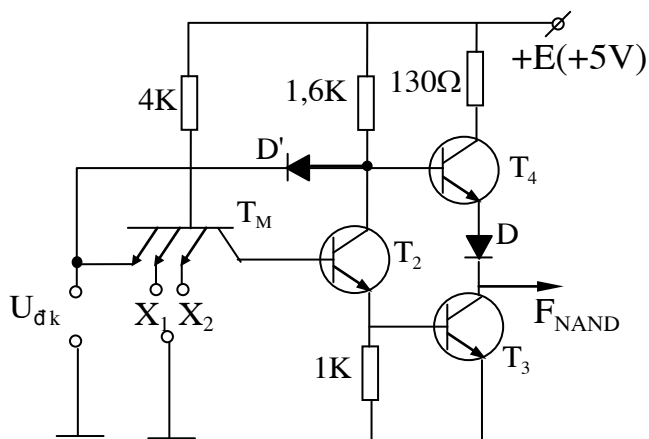


Hình 8: Bảng trạng thái (a), ký hiệu quy ước (b), và giản đồ thời gian (c) của phần tử NAND

Cũng như các phần tử NOT, OR, AND có thể thực hiện phần tử NAND bằng nhiều cách khác nhau dựa trên tất cả các công nghệ bán dẫn: loại điện trở Tranzitor - logic (RTL), loại điôt - Tranzitor - logic (DTL), loại Tranzitor - Tranzitor - logic (TTL) hay công nghệ MOS, CMOS ... Để minh họa hình 9 đưa ra một phần tử NAND dựa trên công nghệ TTL, sử dụng loại Tranzitor nhiều cực emitor, có ưu điểm là bảo đảm mức logic tác động nhanh và khả năng tải lớn.



Hình 9: Nguyên lý xây dựng phần tử NAND loại TTL

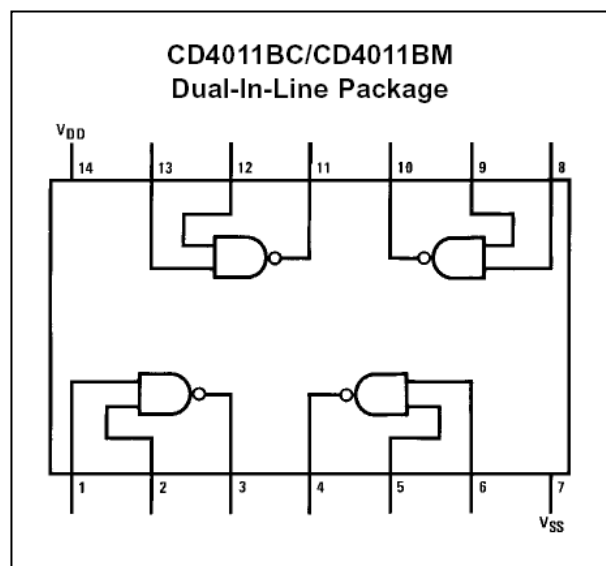
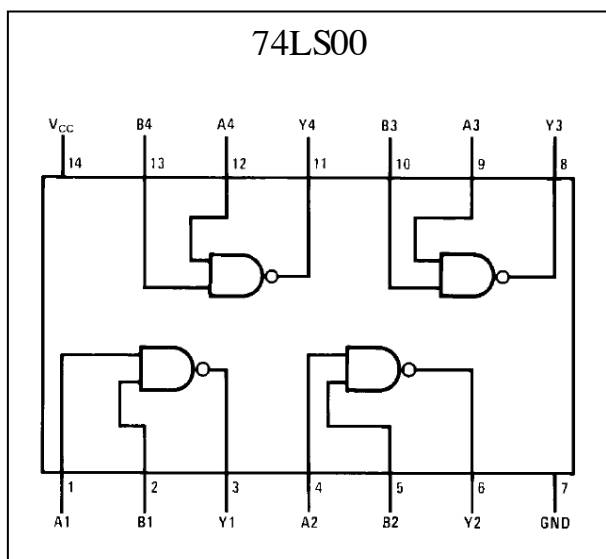


Hình 10: Phần tử logic NAND TTL thực tế có đầu vào điều khiển (loại ba trạng thái ra ổn định)

Với mạch hình 9, khi tất cả lỗi vào có điện áp cao ( $x_1 = x_2 = x_3 = 1$ ),  $T_M$  khoá,  $U_{CM} = U_{B2}$  ở mức cao làm  $T_2$  mở,  $F_{NAND} = 0$ . Nếu chỉ một trong các lỗi vào có mức điện áp thấp tiếp giáp emitter - bazơ tương ứng của  $T_M$  mở làm mất dòng  $I_{B2}$  nên  $T_2$  khoá :  $F_{NAND} = 1$  thực tế  $T_2$  được thay bằng 1 mạch ra (hình10) dạng đẩy kéo tương tự hình 2 cho dòng ra lớn tăng khả năng chịu tải và chống nhiễu. Khi  $T_2$  khoá  $T_3$  cũng khoá (do  $U_{E2} \approx 0$ )  $F_{NAND} = 1$  nhờ bộ lặp cực emitter  $T_4$  trở kháng ra thấp tăng khả năng chịu tải cho toàn mạch.

Khi  $T_2$  mở  $\rightarrow T_3$  mở  $T_4$  khoá, D tách nhánh  $T_4$  khỏi mạch ra  $F_{NAND} = 0$  (mức ra cỡ +0,1V).

Để điều khiển tăng ra có thể dùng một lỗi vào đặc biệt khi  $U_{dk} \approx 0$  (mức thấp)  $T_3, T_4$  đều khoá (trạng thái ổn định thứ 3 của sơ đồ còn gọi là trạng thái trở kháng cao). Khi  $U_{dk}$  ở mức cao điốt  $D'$  khoá, sơ đồ làm việc bình thường như đã phân tích ở trên với hai trạng thái ổn định còn lại. Tín hiệu  $U_{dk}$  được gọi là tín hiệu chọn vô (CS) tạo khả năng cho phép (lúc  $CS = 1$ ) hay không cho phép (lúc  $CS = 0$ ) mạch NAND làm việc, điều này đặc biệt thuận lợi khi điều khiển nhiều NAND làm việc chung với một lỗi ra.



Sơ đồ chân IC logic NAND họ TTL : 74LS00 và họ CMOS : 4011

**1.4.5 Phần tử “Hoặc - phủ định” (NOR):** Gồm nhiều đầu vào biến, một đầu ra thực hiện hàm logic hoặc - phủ định.



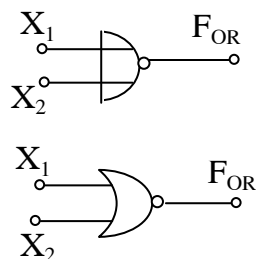
$$F_{\text{NOR}} = \overline{x_1 + x_2 + \dots + x_n}$$

$F_{\text{NOR}} = 1$  khi mọi biến vào có trị số "0" và  $F_{\text{NOR}} = 0$  trong các trường hợp còn lại.

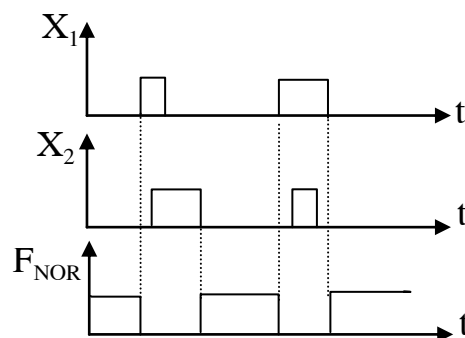
Bảng trạng thái, kí hiệu quy ước và giản đồ thời gian minh hoạ của  $F_{\text{NOR}}$  (với  $n = 2$ ) cho trên hình 11.

$X_1$	$X_2$	$F_{\text{OR}}$
0	0	1
0	1	0
1	0	0
1	1	0

a)



b)

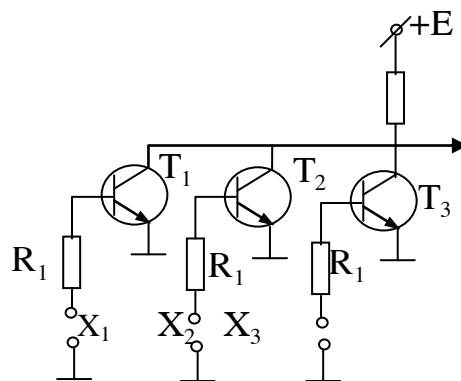


c)

Hình 11: Bảng trạng thái (a); Ký hiệu quy ước (b); và giản đồ thời gian (c) của phần tử NOR

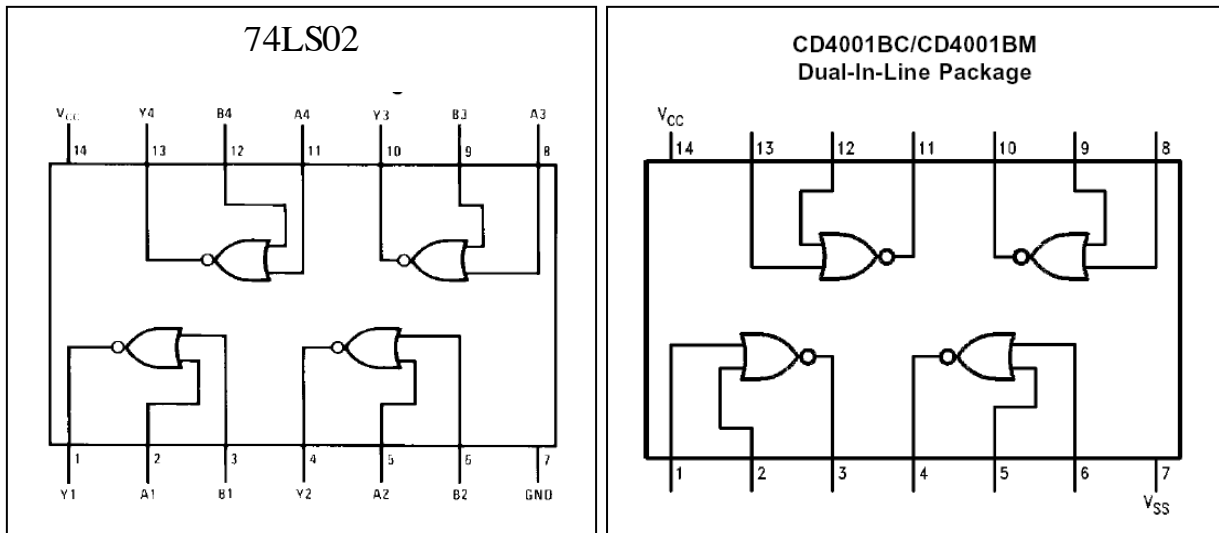
Hình 12 cho kết cấu thực hiện  $F_{\text{NOR}}$  trên công nghệ RTL.

Khi ít nhất một trong các cửa vào có xung dương mở, điện áp ra ở mức thấp  $F_{\text{NOR}} = 0$ , còn khi  $x_1 = x_2 = \dots = x_n = 0$ , do các Tranzitor được thiết kế ở chế độ thường khoá, tất cả các Tranzitor khoá  $\rightarrow F_{\text{NOR}} = 1$  (lưu ý nếu thiết kế các Tranzitor thường mở thì mạch hoạt động như một phần tử NAND với các xung vào cực tính âm điều khiển khoá các Tranzitor).



Hình 12: Phần tử NOR với cực collector hở

Có thể thực hiện phần tử NOR dựa trên công nghệ MOS hoặc CMOS (từng cặp  $\text{MOS}_n$  và  $\text{MOS}_p$  với mỗi đầu ra) với nhiều ưu điểm nổi bật: thời gian chuyển biến nhanh, không có dòng dò và tiêu thụ công suất cực bé.



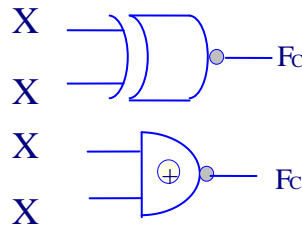
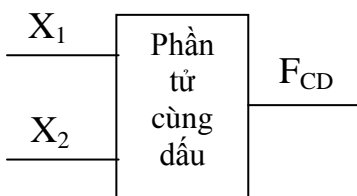
Sơ đồ chân IC logic NOR họ TTL : 74LS02 và họ CMOS : 4001

#### 1.4.6 Phần tử tương đương (cùng dấu).

Là phần tử logic có hai đầu vào biến ( $X_1, X_2$ ), một đầu ra ( $F_{CD}$ ). Quan hệ của hàm ra với các biến vào thỏa mãn ý tưởng sau:

- Khi  $X_1 = X_2$  (cùng nhận giá trị)  $\rightarrow$  hàm ra nhận trị “1”  $\rightarrow F_{CD} = 1$ .
- Khi  $X_1 \neq X_2$  (khác giá trị nhau)  $\rightarrow$  hàm ra nhận trị “0”  $\rightarrow F_{CD} = 0$ .

Hoạt động của phần tử cùng dấu được trình bày ở bảng trạng thái hình 3.



$X_1$	$X_2$	$F_{CD}$
0	0	1
0	1	0
1	0	0
1	1	1

Hình 1: Mạch mô phỏng.

Hình 2: Ký hiệu.

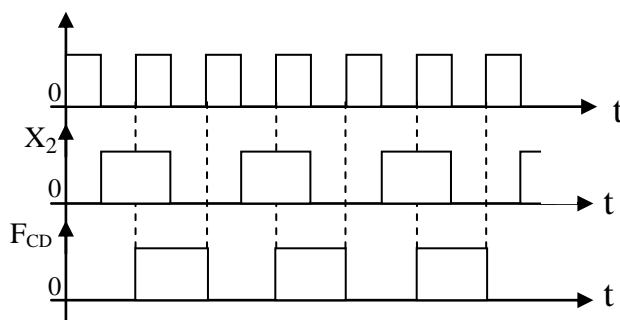
Hình 3: Bảng trạng thái.

Từ bảng trạng thái ta có:  $F_{CD} = \overline{X_1} \cdot \overline{X_2} + X_1 \cdot X_2 = \overline{X_1} \oplus \overline{X_2}$ . (1)

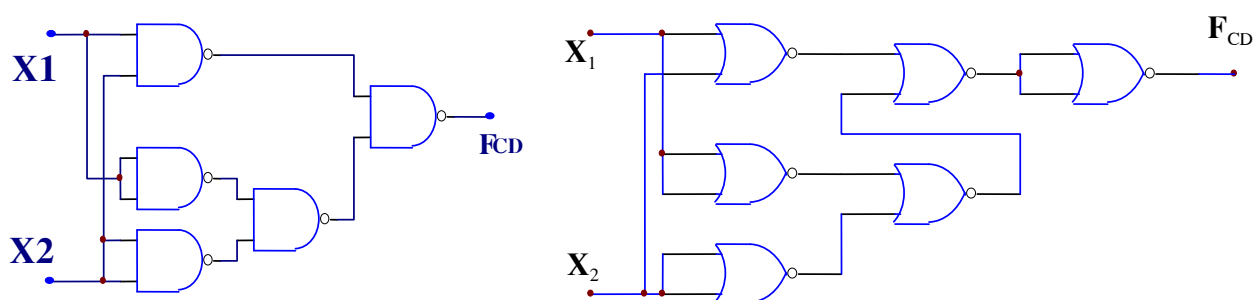
Biến đổi biểu thức (1) ta được:  $F_{CD} = \overline{\overline{\overline{X_1} \cdot \overline{X_2} + X_1 \cdot X_2}} = \overline{\overline{\overline{X_1} \cdot \overline{X_2}} \cdot \overline{X_1 \cdot X_2}}$  (2).

$$F_{CD} = \overline{\overline{\overline{X_1} \cdot \overline{X_2}} \cdot \overline{X_1 \cdot X_2}} = \overline{\overline{\overline{X_1} + X_2} \cdot \overline{X_1 + X_2}} \quad (3)$$

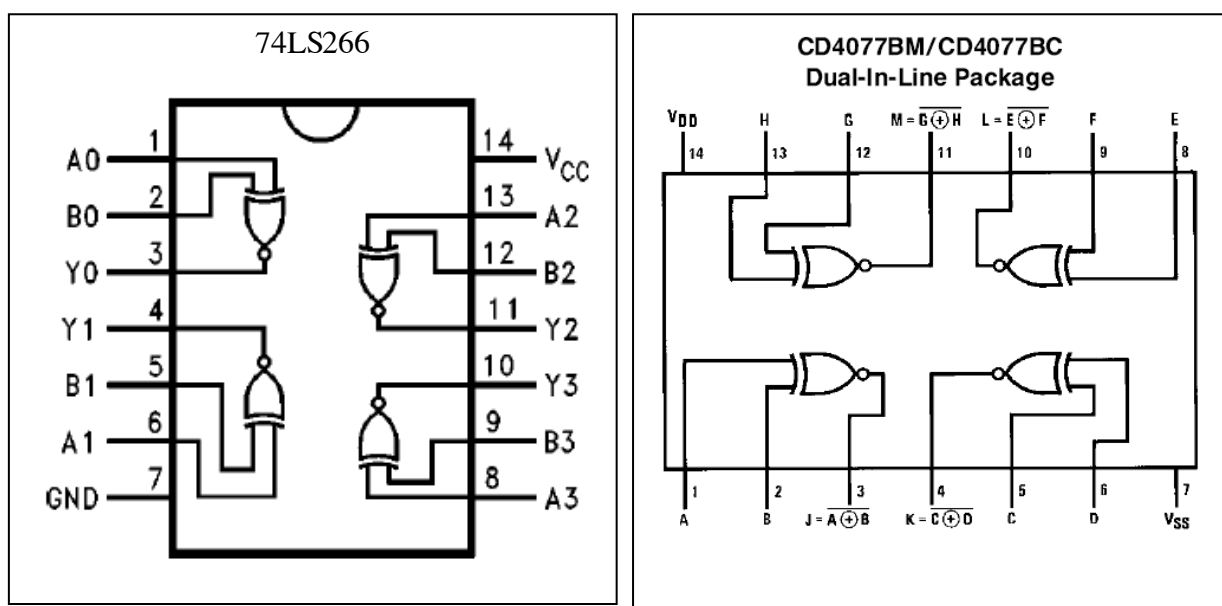
Từ biểu thức (2), (3) ta nhận được mạch điện của phần tử tương đương (cùng dấu) được xây dựng từ các phần tử NAND (NOR) có hai lối vào.



Hình 4: Giải đồ xung.



Hình 5: Phần tử tương đương (cùng dấu) cấu trúc từ phần tử NAND (a) hoặc phần tử NOR (b)



Sơ đồ chân IC cùng dấu họ TTL : 74LS266 và họ CMOS : 4077

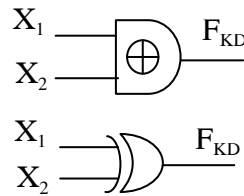
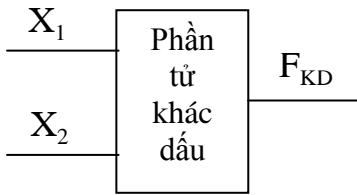
### 1.4.7 Phần tử khác dấu (cộng modul 2).

Là phần tử logic có hai đầu vào biến ( $X_1, X_2$ ), một đầu ra ( $F_{KD}$ ). Quan hệ của hàm ra với các biến vào như sau:

- Khi  $X_1 = X_2$  (cùng nhận giá trị)  $\rightarrow$  hàm ra nhận trị “0”  $\rightarrow F_{KD} = 0$ .

- Khi  $X_1 \neq X_2$  (khác giá trị nhau)  $\rightarrow$  hàm ra nhận trị “1”  $\rightarrow F_{KD} = 1$ .

Hoạt động của phần tử khác dấu được trình bày ở bảng trạng thái hình 8.



$X_1$	$X_2$	$F_{KD}$
0	0	0
0	1	1
1	0	1
1	1	0

Hình 6: Mạch mô phỏng.

Hình 7: Ký hiệu.

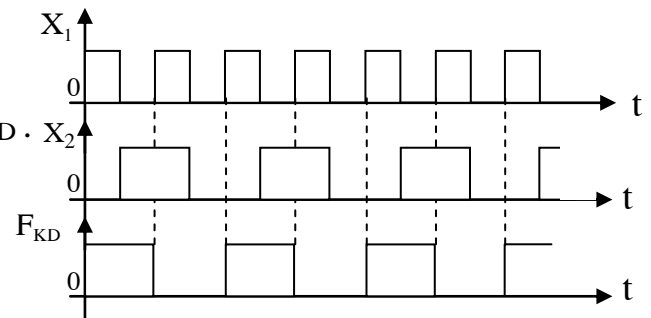
Hình 8: Bảng trạng thái.

Từ bảng trạng thái ta có:

$$F_{KD} = \bar{X}_1 \cdot X_2 + X_1 \cdot \bar{X}_2 = X_1 \oplus X_2 = \bar{F}_{CD} \cdot X_2$$

(1)

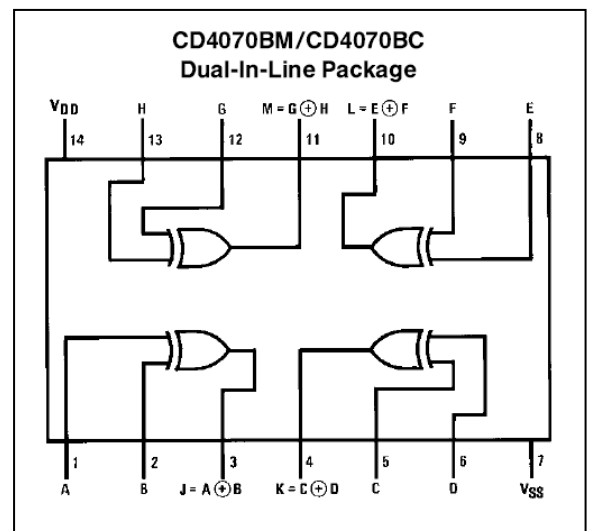
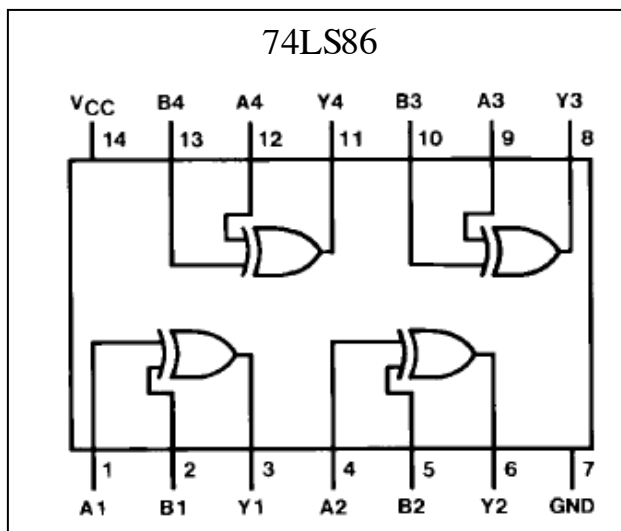
Mạch điện của phần tử khác dấu được suy ra từ mạch điện của phần tử cùng dấu bằng cách mắc thêm vào đầu ra phần tử đảo.



Hình 9: Giải đồ xung.

Phép cộng modul 2 có tính chất sau:

$$\text{Nếu } X_1 \oplus X_2 = X_3 \text{ thì } X_1 \oplus X_3 = X_2 \text{ và } X_3 \oplus X_2 = X_1$$



Sơ đồ chân IC khác dấu họ TTL : 74LS86 và họ CMOS : 4070

### 1.4.8 Các thông số đặc trưng của phần tử logic.

#### a. Các đặc trưng kỹ thuật tiêu biểu của những họ TTL

Sau đây là đặc trưng cơ bản của họ 74LSXX

- Công suất tiêu tán của 74 LSXX:  $P = 2 \text{ mW} + 0.25 \text{ mW/MHz}$  ( với  $C_{\text{tải}} = 15\text{pF}$  )
- Nguồn nuôi +5 V
- Logic “0” điện áp ra 0.2 V
- Logic “1” điện áp ra 3 V
- Khoảng an toàn 1 V
- Khoảng nhiệt độ làm việc  $0^{\circ}\text{C} - 70^{\circ}\text{C}$
- Khoảng nhiệt độ bảo quản  $-65^{\circ}\text{C} - +150^{\circ}\text{C}$
- Điện áp cao nhất cho phép +7 V
- Điện áp thấp nhất -0.5 V
- Điện áp cao nhất giữa hai ngả vào 5.5 V
- Điện áp cao nhất giữa ngả vào và đất +5.5 V
- Điện áp cao nhất giữa ngả ra và đất +5.5 V
- Điện áp thấp nhất giữa ngả vào và đất - 0.8 V
- Điện áp thấp nhất giữa ngả ra và đất - 0.8 V
- (Tốt nhất giữ dòng điện  $< 1\text{mA}$  )
- Nhiệt độ hàn với mỏ hàn  $265^{\circ}\text{C}$
- Nhiệt độ hàn cao nhất với bể thiếc / chì hàn  $240^{\circ}\text{C}$

Họ logic HCMOS ( High-speed-CMOS ) được chế tạo với kỹ thuật CMOS có các ĐTKT ( Có ba loại HCMOS : 74 HC , 74 HCT, và 74 HCU)

- Dòng điện cần cho IC rất bé .
- Không nhạy cảm với nhiễu, sự thay đổi nhiệt độ và hiệu ứng Latch-up.
- Vận tốc làm việc nhanh và chịu số tải giống như bipolar LS-TTL.
- Mức Logic ở ngả ra là 0.1 V và  $V_{\text{cc}} - 0.1 \text{ V}$
- Tất cả đều có đệm trừ loại 74HCU
- Thời gian trễ cho một cổng là 8 ns.
- Khoảng làm việc tới 60 MHz.
- HCMOS-Ics có thể chịu đựng đến 10 tải LS-TTL(4 mA) và loại ngả ra cho Bus Driver đến 15 tải LS-TTL(6 mA)
- Ngả vào có mạch bảo vệ chống tĩnh điện .

- HCMOS có chức năng và sơ đồ chân giống loại IC LS-TTL và CMOS-4000.
- Loại công tắc tín hiệu analog làm việc đến 10 V .
- Thời gian lên xuống của tín hiệu ở ngõ ra giống nhau .
- Khoảng làm việc từ  $-40^{\circ}$  đến  $+85^{\circ}$  hoặc  $-40^{\circ}$  đến  $+125^{\circ}$  C.
- Điện áp tối đa từ - 0.5V đến +7 V
- Công suất tiêu tán tối đa từ 400 mW đến 500 mW.

#### **b. Thiết kế mạch với IC họ 74xxx**

##### **Nguồn nuôi**

Để tổng trở của nguồn nuôi được giữ thấp cần phải có những tụ điện mắc giữa hai cực. Đó là các tụ gốm có trị số từ  $1\mu\text{F}$  đến 100 nF. Để các tụ đạt hiệu quả cao nên nối chúng càng gần mạch in càng tốt cứ 4 đến 6 IC thì cần một tụ . Các tụ này phải có tính cao tần tốt, mặt khác góc tổn hao của tụ điện phải khá lớn để các dao động được hình thành bởi các tụ và độ tự cảm của mạch in được suy giảm mạnh .

##### **Tải điện dung**

Tải điện dung cho những mạch TTL không được vượt quá 100pF. Mục đích để những ngõ ra không quá tải với dòng nạp và phóng điện, các mức logic được đảm bảo. Trường hợp cần làm trễ tín hiệu với những tụ điện khá lớn nên mắc vào trước dây điện trở = 270  $\Omega$  cho loại schottky 74S và R = 1K $\Omega$  cho loại low power schottky 54LS/74LS.

##### **Diode khoá ở ngõ vào ( Input Clamping Diode )**

ở ngõ vào cổng TTL đều có Diode khoá, trừ loại 74LS. Mục đích để triệt tiêu bớt các điện áp âm từ các tín hiệu phản hồi , do đó nhiễu của hệ thống bị ngăn chặn .

##### **Cách nối song song các ngõ ra**

Chỉ được nối cao nhất hai ngõ ra hoặc hai ngõ vào với nhau. Để biết chính xác sự phân bố dòng điện, chỉ nên nối hai cổng song song với nhau khi cả hai cổng ở trong cùng một IC.

---

**Dây đất**

Để tránh sự sụt áp trên dây đất, mạch điện phải được thiết kế sao cho mạch đất có điện trở bé và độ tự cảm bé. Mạch đất và mạch nối điện áp nguồn  $V_{cc}$  phải tự có độ rộng ít nhất 2.5 mm để sự ảnh hưởng của điện áp ngoài da được giữ thật thấp. Mạch điện với độ rộng từ 0.5 đến 1mm có độ tự cảm  $L=10\text{nH/cm}$

**Những ngã vào không dùng tới**

Những ngã vào của AND và NAND nên nối đất với  $V_{cc}$  qua một điện trở  $R=1\text{ K}\Omega$ . Những ngã vào của OR và NOR nên nối với đất. Ngã vào của các logic khác ( flip-flop, bộ đếm ... ) nên nối với đất .

**c. Các đặc trưng kỹ thuật tiêu biểu của những họ CMOS**

CMOS được viết tắt từ Complementary-Metall-Oxide-Silicon. Đầu tiên CMOS được nghiên cứu để sử dụng trong ngành kỹ thuật hàng không vũ trụ. Với đặc tính như không bị lệ thuộc vào lưới điện và miễn nhiễm ... ngày nay CMOS được sử dụng rộng rãi trong điện tử công nghiệp, điện tử y khoa, kỹ thuật xe hơi và cả trong kỹ thuật máy tính điện tử .

CMOS có một số đặc tính quan trọng :

- Công suất tiêu tán bé : 02.5nW per Gate (static).
- Điện áp làm việc từ 3 V đến 15 V , max 18 V.
- CMOS chống nhiễu tốt .
- Khoảng nhiệt độ làm việc :

Thương mại : - 40<sup>0</sup> C đến +85 C

Quân sự : - 55<sup>0</sup> C đến +125<sup>0</sup> C

Lúc không làm việc hai Transitor kênh P và N không bao giờ dẫn điện cùng một lúc. Vì thế dòng điện chạy từ  $V_{dd}$  đến  $V_{ss}$  chỉ là dòng điện rò khoảng 0,5 nA cho mỗi cổng. Với điện áp + 5 V công suất tiêu tán cho mỗi cổng chỉ có 2,5 nW. Công suất tiêu tán của CMOS tăng nhanh với tần số làm việc và cao hơn so với loại LS-TTL trong khoảng từ 100 KHz đến 2 MHz. Nhưng trên thực tế công suất tiêu tán của một IC CMOS trong lúc làm việc không bao giờ hơn công suất tiêu tán của IC TTL ngay trong lúc không làm việc ! Có hai lý do :

---

- Các cổng CMOS chỉ dẫn điện khi nó làm việc (khi các cổng Clock và data được kích ).

- Trên thực tế lúc làm việc với tần số cao, không phải cổng nào của IC CMOS cũng làm việc .

### **Điện áp**

CMOS có thể hoạt động từ 3 V đến 15 V. Tuy nhiên với điện áp nhỏ hơn 4,5 V thời gian trễ sẽ ra tăng (vận tốc làm việc chậm lại ), tổng trở ra cũng lớn hơn và đồng thời tính chống nhiễu sẽ giảm. Với điện áp lớn hơn 15 V ta cũng có những bất lợi :

- Công suất tiêu tán lúc CMOS hoạt động tăng cao .
- Với những xung nhiễu từ nguồn vượt quá điện áp đánh thủng (20 V), tạo ra hiệu ứng SCR-latch-up và làm hỏng IC nếu không được hạn chế từ bên ngoài .

Trường hợp bắt buộc phải dùng điện áp cao hơn 15 V, max 18 V cần phải có điện trở hạn dòng từ bên ngoài .

### **Thời gian trễ**

Điện áp càng cao thì CMOS hoạt động càng nhanh . Thời gian trễ gia tăng với nhiệt độ và tải điện dung .

### **Tính miễn nhiễu ( Noise immunity )**

CMOS chống nhiễu rất tốt, thường là 45 % điện áp cấp 2,25 V với điện áp 5 V và 4,5 V cho điện áp 10 V. Thời gian trễ của CMOS đóng vai trò như bộ lọc nhiễu. Xung 10 ns biến mất sau một chuỗi các cổng CMOS, còn sau một chuỗi các cổng TTL thì được khuếch đại. Vì tính chất đặc biệt này CMOS được dùng để thiết kế các mạch điện của các thiết bị công nghiệp phải hoạt động trong môi trường đầy nhiễu điện và điện từ .

Với điện áp cấp +5 V CMOS vẫn làm việc bình thường với sự mất ổn định của điện áp cấp hay điện áp nhiễu đến 1V

### **Giao tiếp với họ TTL**



Với điện áp 5V CMOS giao tiếp thẳng với TTL. Tổng trở vào của CMOS rất lớn, TTL có thể tải vô số cổng CMOS mà không làm mất Fan out ở trạng thái LOW.

Trạng thái High của TTL thường chỉ có 3,5V. Ta nên mắc vào ngả ra của TTL một điện trở nâng áp ( pull up resistor ) từ  $1k\Omega$  đến  $10k\Omega$  với điện áp 5V để trạng thái High được kéo lên đến 4,5V hay hơn. Khi CMOS hoạt động với điện áp lớn hơn 5V sự giao tiếp thẳng với TTL không thể thực hiện. IC 4104B (Quad level translator ) chuyển mức TTL lên cao 15V.

Các IC 4009, 4010, 4049 và 4050 tạo sự giao tiếp giữa mức điện CMOS và TTL khi CMOS dùng điện áp  $> 5V$ .

### **Điện dung ngả ra / vào**

Điện dung ngả vào của CMOS = 1.5pF đến 5pF và điện dung ngả ra = 3pF đến 7.5 pF .

### **Những chú ý khi sử dụng CMOS**

Vì ngả vào của CMOS có điện trở vào lên đến  $10^{14}\Omega$  , điện dung  $\approx 1pF$  rất bé với dòng điện rò không đáng kể (10pA ), nếu không có biện pháp bảo vệ thích hợp , lớp SiO<sub>2</sub> của Gate ở ngả vào có thể bị phá huỷ vì tĩnh điện nạp vượt quá một trị số cho phép  $\approx 100V$ . Tĩnh điện của Nylon  $\approx 2,2 kV$ ; 1 bao Plastic  $\approx 2$  đến 4kV có thể dễ dàng làm hỏng lớp Oxid của Gate .

Một số biện pháp sau đây nên thực hiện :

- IC CMOS chưa dùng nên đựng trong các ống nhựa antistatic hay bọc trong các giấy nhôm .
- Mở hàn nên dùng loại có nối đất .
- Bàn làm việc và người làm việc nên được nối đất ( qua một điện trở  $1M\Omega$  để đảm bảo an toàn lao động ).

Không nên tháo hay ráp IC vào mạch điện đang có điện áp .

- Những ngả vào không dùng đến nên nối với Vss hay VDD hay ngả ra của một mạch Logic. Nên dùng nguồn cấp điện bé hơn 12V nếu có thể được để giảm tỉ số IC CMOS bị hư hỏng.

---

**Công suất tiêu tán một cổng NAND 4000B**


---

$$P = 10nW + 1,5mW / MHz ; \quad (C_L = 15pF, V_{DD} = 10V)$$

$$P = 10nW + 3,38mW / MHz ; \quad (C_L = 15pF, V_{DD} = 15V)$$

**d. Những chú ý cần thiết khi thiết kế mạch với IC CMOS**

- Tất cả các chân không dùng nên nối với đất hay điện áp cấp để tránh loại logic “có lẽ ...” cho IC.

- Những tín hiệu vào thay đổi mức logic quá chậm sẽ làm IC CMOS dao động và IC bị trigger nhiều lần. Điện áp cấp cho IC ổn áp kém và không sạch sẽ đưa đến trường hợp này vì điện áp ngưỡng vào của IC tùy thuộc vào điện áp cấp. Với các xung đồng bộ có thời gian lên chậm IC CMOS cũng thường hiểu sai. Flip-flops, Register và Latch-ICs có thời gian giữ (hold time) khá lâu. Dữ liệu vào thường phải chờ sau khi sườn xung đồng bộ đã xuất hiện.

- IC CMOS cùng loại có đặc trưng kỹ thuật khác nhau với các nhà chế tạo khác nhau.

- Dòng ra của IC CMOS loại B cho toàn dải nhiệt độ làm việc khoảng 0,36mA đủ để thúc một cổng LS-TTL.

IC logic CMOS loại 4000A, 4000UB không có đệm; 4000A/B, 4000B có đệm .

Loại có đệm có những ưu điểm sau :

- Tổng trở ra không tùy thuộc vào điện áp ra, vì chế độ dốc xung lên hay xung xuống đều bằng nhau.
- Khoảng an toàn lớn hơn.

Loại không có đệm có thời gian trễ bé hơn.

IC CMOS với ký hiệu B (có đệm) hay UB (không có đệm) là những loại có đặc trưng kỹ thuật đạt những tiêu chuẩn tối thiểu của công nghiệp. Cổng và bộ đảo của loại UB chỉ được cấu tạo có một tầng. Với độ khuếch đại bé loại UB có độ an toàn và tính chất tuyến tính những bộ dao động, Monoflop hoặc bộ

---

khuyếch đại. ở ngõ ra ta được những tín hiệu sạch và ổn định. Và vì chỉ cấu tạo bởi một tầng, UB có vận tốc làm việc nhanh hơn. Chỉ có một số IC CMOS được chế tạo theo loại UB như 4000, 4001, 4002, 4023, 4025, 4041, 4049 và 4069. Còn lại đều là B.

## Chương 2: Các mạch số học

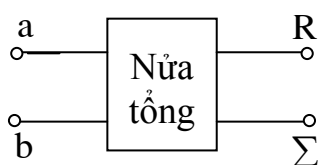
### 2.1 Bộ cộng

#### 2.1.1. Phần tử nửa tổng:

Là mạch logic có hai đầu vào biến  $a, b$ ; hai đầu ra  $\Sigma, R$  với:

- $a, b$ : hai số nhị phân 1 bit cần cộng.
- $\Sigma$ : kết quả của phép cộng hai số nhị phân  $a, b$ .
- $R$ : Kết quả nhớ của phép cộng hai số nhị phân nói trên.

Quá trình làm việc thỏa mãn bảng trạng thái hình 2 (bảng trạng thái được xây dựng dựa trên cơ sở quy tắc của phép cộng hai số nhị phân 1 bit).



Hình 1: Mạch mô phỏng

a	b	$\Sigma$	R
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

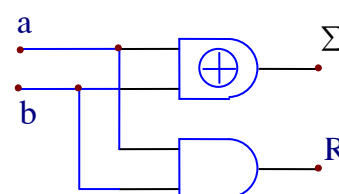
Hình 2: Bảng trạng thái

Từ bảng trạng thái ta có hệ sau:

$$\begin{cases} \Sigma = \bar{a}.b + a.\bar{b} = a \oplus b \\ R = a.b \end{cases} \quad (1)$$

Từ (1) ta có mạch điện hình 3.

Phần tử nửa tổng gồm một phần tử

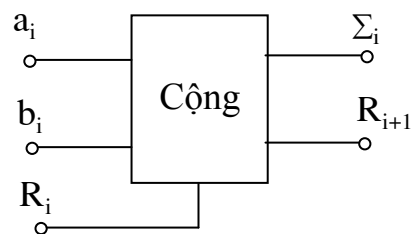


Hình 3: Cấu trúc của phần tử nửa tổng

cộng môđun 2 (thực hiện phép cộng) và 1 phần tử AND (kết quả nhớ của phép cộng)

#### 2.1.2. Phần tử toàn tổng.

Gồm 3 đầu vào (hai đầu vào biến, một đầu vào kết quả nhớ của nhịp trước gửi tới) và 2 đầu ra (thực hiện cộng môđun 2 và tạo nhớ cho nhịp sau) thỏa mãn bảng trạng thái Hình 5.



Hình 4: Mạch điện mô phỏng

+ $R_i$  : Kết quả nhớ của nhịp trước đến.

+  $R_{i+1}$  : Kết quả nhớ gửi đến nhịp sau.

$a_i$	$b_i$	$R_i$	$\Sigma_i$	$R_{i+1}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Hình 5: Bảng trạng thái

		$a_i b_i$			
$R_i$	$\Sigma_i$	00	01	11	10
		0	1	0	1
		1	0	1	0

a)

		$a_i b_i$			
$R_{i+1}$	$R_i$	00	01	11	10
		0	0	1	0
		1	0	1	1

b)

Hình 6: Bìa các nô các hàm ra  $\Sigma_i$ ,  $R_{i+1}$

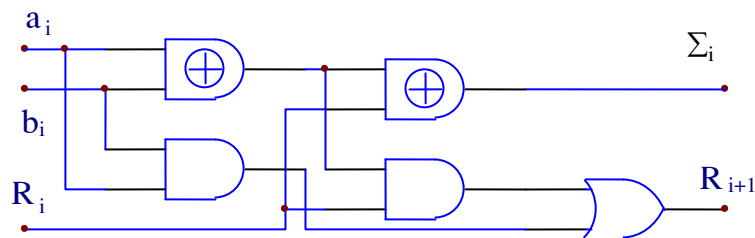
Từ hình 6a:

$$\begin{aligned}\Sigma_i &= \overline{R_i}(\overline{a_i}b_i + a_i\overline{b_i}) + R_i(\overline{a_i}\overline{b_i} + a_ib_i) = \overline{R_i}(a_i \oplus b_i) + R_i\overline{(a_i \oplus b_i)} \\ &= a_i \oplus b_i \oplus R_i\end{aligned}\quad (2)$$

Từ hình 6b:

$$R_{i+1} = a_ib_i + R_i(\overline{a_i}b_i + a_i\overline{b_i}) = a_ib_i + R_i(a_i \oplus b_i) \quad (3)$$

Từ (2) và (3) ta có mạch điện hình 7:



Hình 7: Bộ tổng đầy đủ được cấu trúc từ hai phần tử nửa tổng và phần tử OR

## 2.2 Phép cộng hai số nhị phân n bit

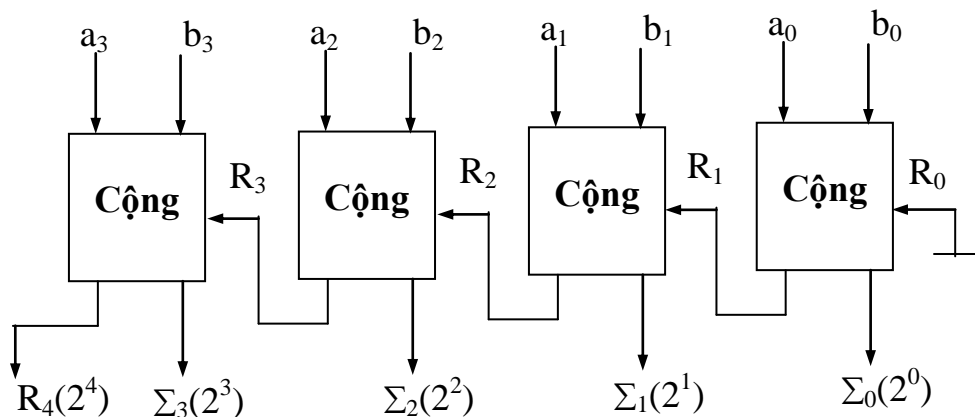
### 2.2.1 Bộ cộng hai số nhị phân n bit

Về nguyên tắc để thực hiện phép cộng hai số nhị phân n bit cần sử dụng n phần tử toàn tổng. Để đơn giản cho quá trình phân tích ta xét phép cộng hai số nhị phân 4 bit:  $A = a_3 a_2 a_1 a_0$  ;  $B = b_3 b_2 b_1 b_0$

Phải cần đến 4 phần tử cộng đủ để thực hiện phép tính này

+ Kết quả =  $R_4 \Sigma_3 \Sigma_2 \Sigma_1 \Sigma_0$

$$\begin{array}{cccc} & R_3 & R_2 & R_1 & R_0 \\ A=a_3 & \uparrow & \uparrow & \uparrow & \\ & a_2 & a_1 & a_0 & \\ B=b_3 & \uparrow & \uparrow & \uparrow & \\ & b_2 & b_1 & b_0 & \\ \hline R_4 & R_3 & R_2 & R_1 & \\ R_4 \Sigma_3 & \Sigma_2 & \Sigma_1 & \Sigma_0 & \end{array}$$



Hình 8: Bộ cộng song song hai số nhị phân 4 bit

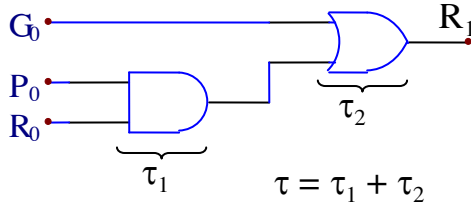
Với kết cấu hình 8 mạch có ưu điểm là đơn giản tuy nhiên có nhược điểm là độ tác động không nhanh vì phải chờ kết quả từ bit thấp đưa lên phép cộng bit cao hơn kề nó. Để khắc phục nhược điểm này ta dùng bộ cộng song song tính trước số nhớ.

**2.2.2 Bộ cộng song song tính trước số nhớ:** (thời gian trễ không phụ thuộc vào chiều dài của bit).

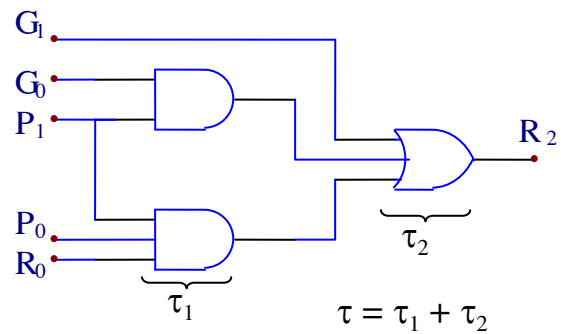
Từ (3)  $\rightarrow R_{i+1} = a_i b_i + R_i (a_i \oplus b_i)$ ; Đặt  $P_i = a_i \oplus b_i$   $G_i = a_i \cdot b_i$

Khi đó  $R_{i+1} = G_i + R_i P_i$  (4) Khi  $i = 0 \rightarrow R_1 = a_0 b_0 + R_0 P_0$  (5)

Từ (5) ta có mạch điện hình 9:



Hình 9:



Hình 10:

\* Khi  $i = 1 \rightarrow R_2 = G_1 + R_1P_1$  (6). Thay  $R_1$  từ (5) vào (6) ta có:

$$R_2 = G_1 + G_0P_1 + R_0P_0P_1 \quad (7)$$

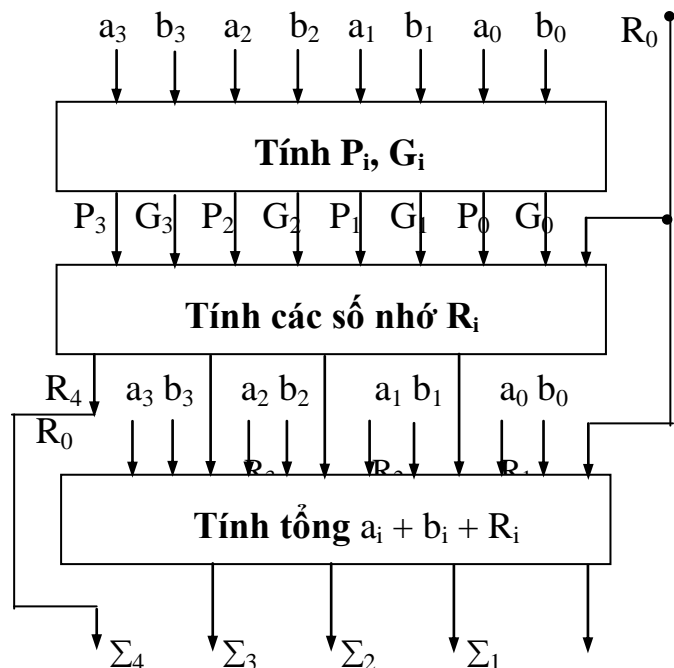
Từ (7)  $\rightarrow$  Mạch điện hình 10

So sánh kết quả (thời gian trễ  $\tau$ ) của hình 9, hình 10 ta có kết luận : thời gian trễ không phụ thuộc vào độ dài của bit.

Như vậy theo phương pháp này trình tự các bước được tiến hành như sau:

- Tính  $P_i, G_i$ .
- Tính  $R_i$ .
- Tính tổng  $a_i + b_i + R_i$

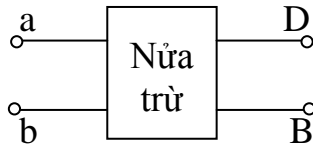
Hình 11: Lưu đồ các bộ cộng song song 2 số nhị phân 4 bit có tính trước số nhớ



## 2.3 Bộ trừ

**2.3.1 Bộ nửa trừ (bán hiệu):** Có hai đầu vào biến  $a$ ,  $b$  và hai đầu ra  $D$ ,  $B$  với:

$a$ : số bị trừ ;  $b$ : số trừ ;  $D$ : kết quả của phép trừ ;  $B$ : kết quả nhớ (vay của phép trừ)



Hình 1: Mạch mô phỏng

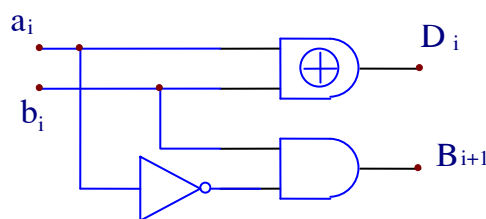
a	b	D	B
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

Hình 2: Bảng trạng thái

Từ bảng trạng thái ta có

$$\begin{cases} D = \bar{a}b + a\bar{b} = a \oplus b \\ B = \bar{a}.b \end{cases} \quad (1)$$

Từ (1) xây dựng mạch điện hình 3



Hình 3: Cấu trúc phân tử nửa trừ

**2.3.2 Bộ trừ đầy đủ:**

Gồm 3 đầu vào (2 đầu vào biến, một đầu vào nhớ kết quả của nhịp trước gửi tới) 2 đầu ra (thực hiện phép trừ và tạo nhớ nhịp sau) thỏa mãn bảng trạng thái hình 5.

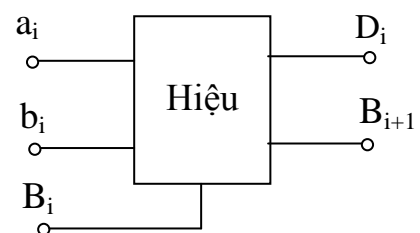
+  $a_i$  : số bị trừ.

+  $b_i$  : số trừ .

+  $D_i$  : hiệu số

+  $B_i$  : Kết quả nhớ của nhịp trước gửi đến.

+  $B_{i+1}$  : Kết quả nhớ gửi đến nhịp sau.



Hình 4: Mạch điện mô phỏng



$a_i$	$b_i$	$B_i$	$D_i$	$B_{i+1}$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Hình 5: Bảng trạng thái

a)

$D_i$	$a_i b_i$	00	01	11	10
$B_i$	0	0	1	0	1
	1	1	0	1	0

b)

$B_{i+1}$	$a_i b_i$	00	01	11	10
$B_i$	0	0	0	1	0
	1	0	1	1	1

Hình 6: Bìa các nô các hàm ra  $D_i$ ,  $B_{i+1}$

Bảng trạng thái được xây dựng dựa vào quy tắc của phép trừ 2 số nhị phân.

Từ hình 6a:  $D_i = \bar{B}_i (\bar{a}_i b_i + a_i \bar{b}_i) + B_i (\bar{a}_i \bar{b}_i + a_i b_i)$

$$= \bar{B}_i (a_i \oplus b_i) + B_i (\overline{a_i \oplus b_i})$$

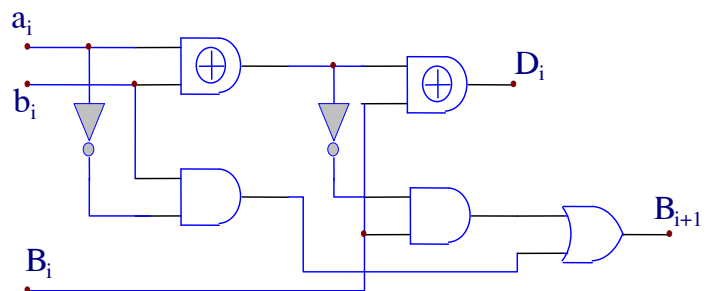
$$= (a_i \oplus b_i) \oplus B_i \quad (2)$$

Từ hình 6b:  $B_{i+1} = \bar{a}_i b_i + B_i (\bar{a}_i \bar{b}_i + a_i b_i)$

$$= \bar{a}_i b_i + B_i (a_i \oplus b_i) \quad (3)$$

Từ (2) và (3) ta có mạch điện hình 7:

Hình 7:  
Bộ trừ đầy đủ được cấu trúc từ hai phần tử nửa trừ và phần tử OR



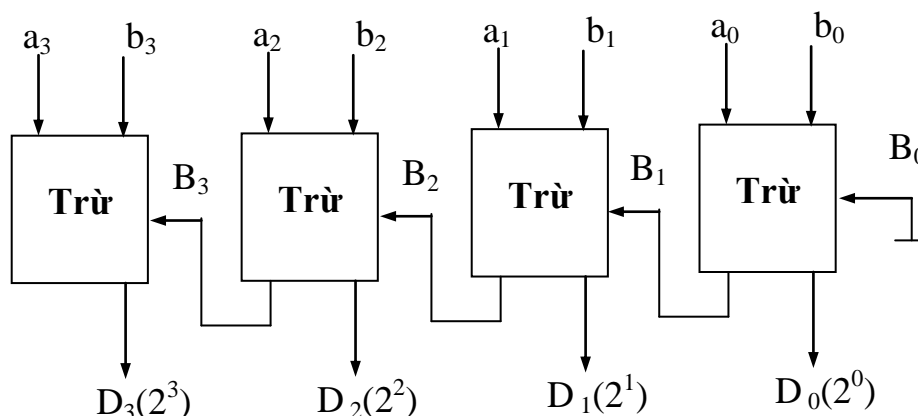
### 2.3.3 Phép trừ hai số nhị phân n bit.

Ta xét trường hợp phép trừ hai số nhị phân 4 bit.

Phải cần đến 4 phần tử trừ  
đủ để thực hiện phép tính trên

$$\begin{array}{r}
 A = a_3 \quad a_2 \quad a_1 \quad a_0 \\
 - B = b_3 \quad b_2 \quad b_1 \quad b_0 \\
 \hline
 B_3 \quad B_2 \quad B_1 \quad B_0 \\
 \hline
 B_4 \quad B_3 \quad B_2 \quad B_1 \\
 D_3 \quad D_2 \quad D_1 \quad D_0
 \end{array}$$

Kết quả =



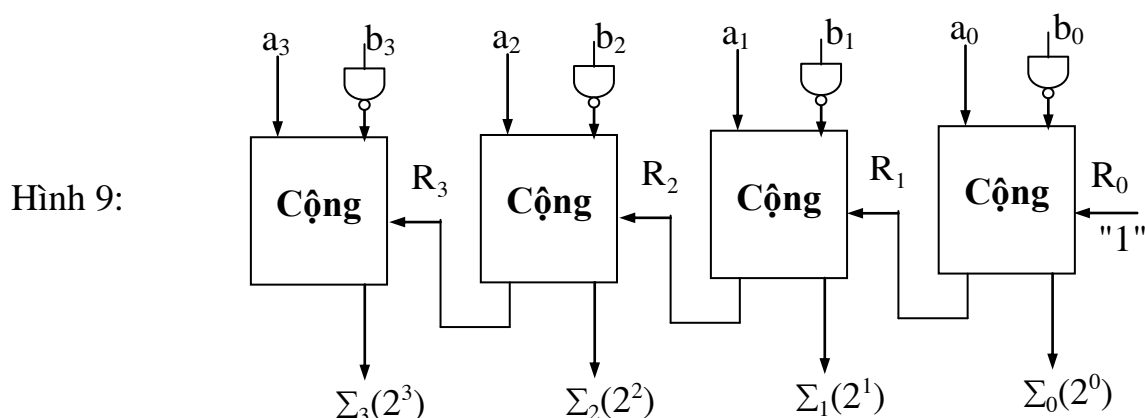
Hình 8: Bộ trừ hai số nhị phân 4 bit

### 2.3.4 Dùng phần tử toàn tổng thực hiện phép trừ hai số nhị phân.

Xuất phát:  $A - B = A + \text{bù 2 của } B = A + [(\text{bù 1 của } B) + 1]$ , ta có mạch điện sau:

Hình 9: Dùng 4 phần tử toàn tổng thực hiện phép trừ 2 số nhị phân 4 bit

Kết quả của phép trừ =  $\Sigma_3 \Sigma_2 \Sigma_1 \Sigma_0$  (bỏ kết quả nhớ  $R_4$ ).

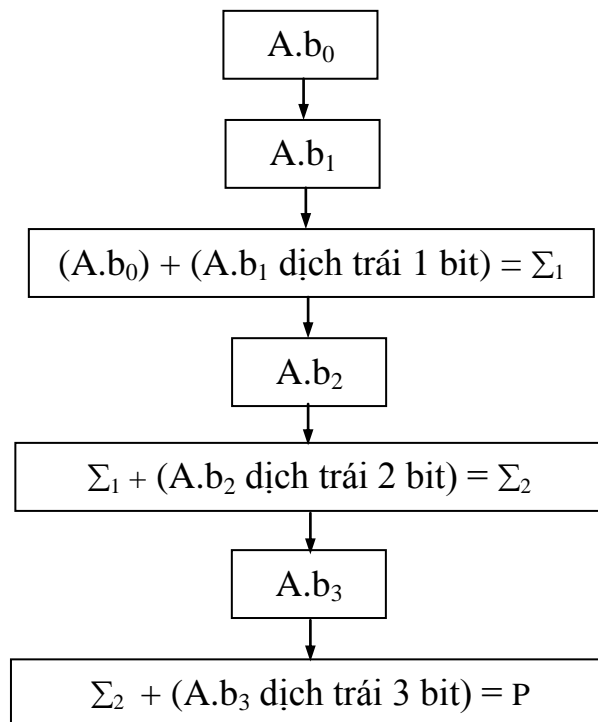


## 2.4 . Bộ nhân

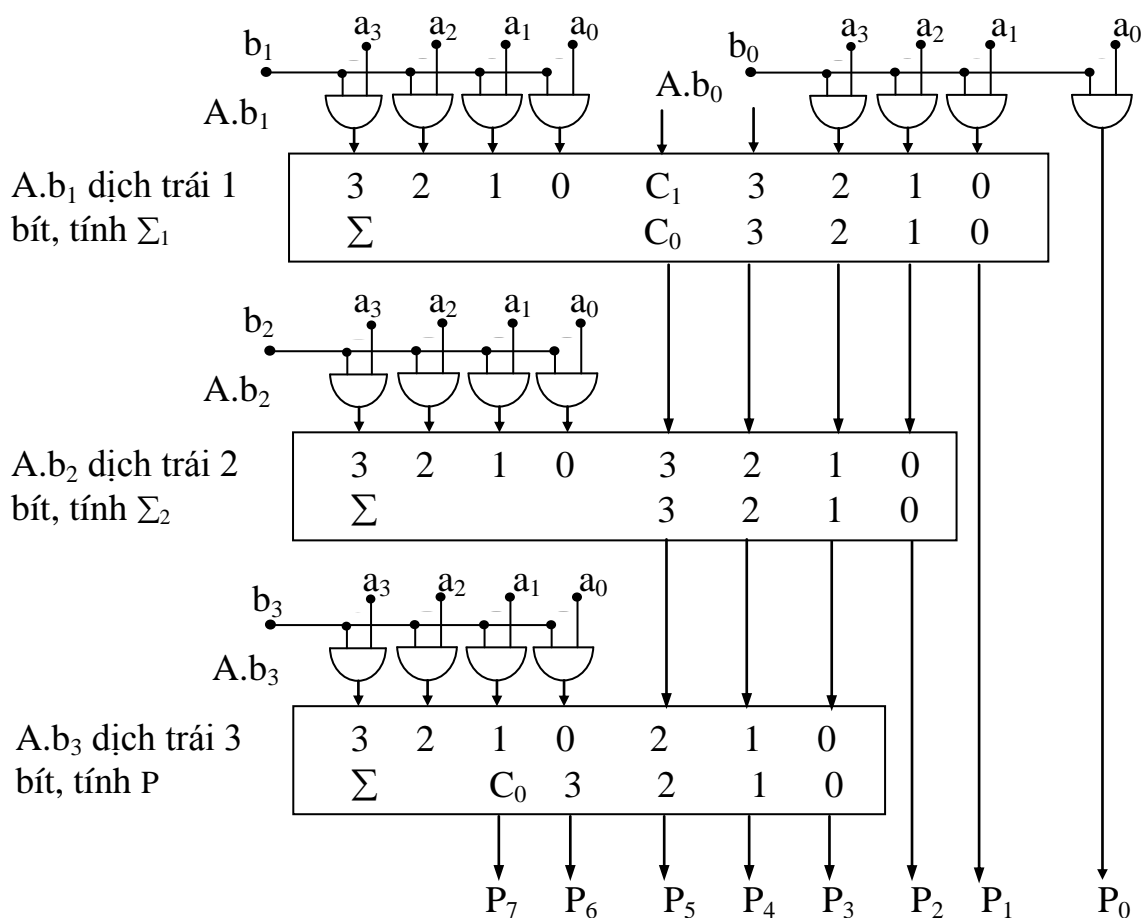
Ta thực hiện phép nhân hai số nhị phân 4 bit.

$$\begin{array}{r}
 A = a_3 a_2 a_1 a_0 \\
 B = b_3 b_2 b_1 b_0 \\
 \hline
 \begin{array}{r}
 A.b_0 \quad \quad \quad a_3b_0 \ a_2b_0 \ a_1b_0 \ a_0b_0 \\
 A.b_1 \text{ Dịch trái một bit} \quad \leftarrow + \quad \quad \quad a_3b_1 \ a_2b_1 \ a_1b_1 \ a_0b_1 \\
 A.b_2 \text{ Dịch trái một bit} \quad \leftarrow \quad \quad \quad a_3b_2 \ a_2b_2 \ a_1b_2 \ a_0b_2 \\
 A.b_3 \text{ Dịch trái một bit} \quad \leftarrow \quad \quad \quad a_3b_3 \ a_2b_3 \ a_1b_3 \ a_0b_3 \\
 \hline
 \begin{array}{cccccccc}
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
 P_7 & P_6 & P_5 & P_4 & P_3 & P_2 & P_1 & P_0
 \end{array}
 \end{array}
 \end{array}$$

Từ phép nhân trên ta đưa ra lưu đồ thuật toán như sau:



Từ lưu đồ thuật toán ta triển khai mạch điện cụ thể hơn.



## 2.5 Bộ so sánh

Thực hiện thuật toán so sánh hai số nhị phân:

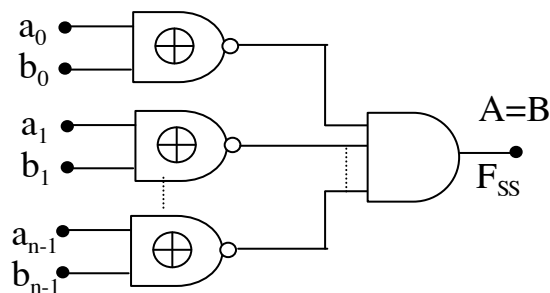
$$A_n = (a_{n-1}a_{n-2}...a_1a_0)_2 ; B_n = (b_{n-1}b_{n-2}...b_1b_0)_2$$

Kết quả so sánh phát hiện ra một trong ba trạng thái có thể xảy ra:

Hoặc  $A_n < B_n$  hoặc  $A_n = B_n$  hoặc  $A_n > B_n$ .

### 1- Phép so sánh bằng nhau:

Để thực hiện việc so sánh bằng nhau của 2 số nhị phân n bit ta dùng n phần tử tương đương cùng dấu kết hợp với một phần tử AND n lối vào. Hai bit có trọng số tương ứng (cùng chỉ số)



Hình 1: Mạch so sánh bằng nhau của hai số nhị phân n bit.

$F_{ss} = 1$  khi  $A = B$ ;  $F_{ss} = 0$  khi  $A \neq B$

được so sánh với nhau bởi một phần tử cùng dấu.

Khi tất cả các bit có trọng số tương ứng của hai số nhị phân cùng giá trị, các đầu ra của các phần tử cùng dấu đều nhận trị "1", qua mạch AND hàm  $F_{SS}$  nhận trị "1" ( $A = B$ ). Chỉ cần có ít nhất một sự sai khác về trị số của 2 bit có trọng số tương ứng thì hàm  $F_{SS}$  nhận trị "0" ( $A \neq B$ ).

## 2- Phần tử so sánh vạn năng hai số nhị phân một bit

Có 3 đầu vào  $E_i$ ,  $a_i$ ,  $b_i$ , 3 đầu ra  $S_i$ ,  $E_{i-1}$ ,  $I_i$ .

$E_i$ : Tín hiệu cho phép.

+  $E_i = 0$ : Không cho phép so sánh,

khi đó các đầu ra đều nhận trị "0".

+  $E_i = 1$ : Cho phép so sánh.

Khi  $E_i = 1$ :

-  $a_i > b_i \rightarrow S_i = "1"$ ;  $E_{i-1}$ ,  $I_i$  đều nhận trị "0".

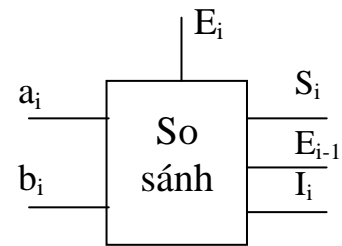
-  $a_i = b_i \rightarrow E_{i-1} = "1"$ ;  $S_i$ ,  $I_i$  đều nhận trị "0"

-  $a_i < b_i \rightarrow I_i = "1"$ ;  $S_i$ ,  $E_{i-1}$  đều nhận trị "0"

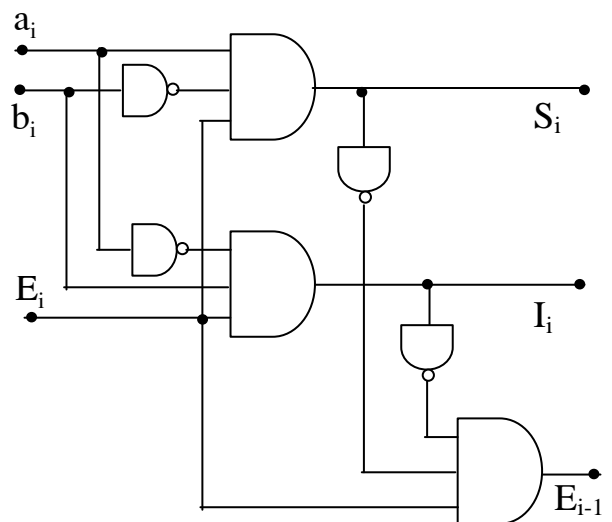
Qua các ý tưởng trên ta xây dựng bảng trạng thái hình 3.

$E_i$	$a_i$	$b_i$	$S_i$	$E_{i-1}$	$I_i$
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	0	1	0
1	0	1	0	0	1
1	1	0	1	0	0
1	1	1	0	1	0

Hình 3: Bảng trạng thái



Hình 2: Mạch mô phỏng



Hình 4: Mạch điện thực hiện

Từ bảng trạng thái ta xác định được quan hệ của các hàm ra với các biến vào như sau:

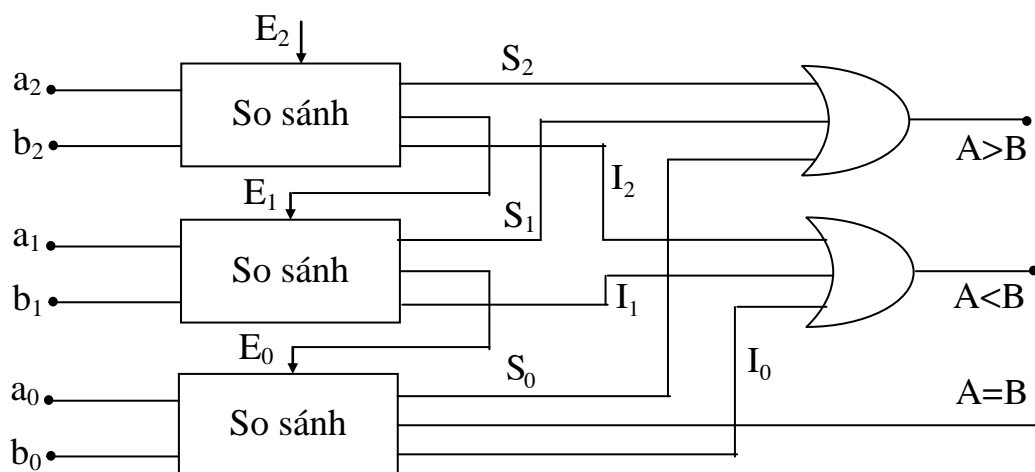
$$S_i = E_i \cdot a_i \cdot \bar{b}_i \quad (1) ; \quad I_i = E_i \cdot \bar{a}_i \cdot b_i \quad (2) ; \quad E_{i+1} = E_i \cdot (\bar{a}_i \cdot \bar{b}_i + a_i \cdot b_i) \quad (3) ;$$

$$E_{i+1} = E_i \cdot \overline{(a_i \oplus b_i)} = E_i \cdot \overline{(S_i + I_i)} = E_i \cdot \bar{S}_i \cdot \bar{I}_i \quad (4)$$

Từ các biểu thức trên cho phép ta xây dựng được mạch điện hình 4.

### 3- Mạch điện so sánh vạn năng hai số nhị phân n bit

Về nguyên tắc ta phải dùng n phần tử so sánh vạn năng 2 số nhị phân 1 bit, việc so sánh bắt đầu được thực hiện từ cặp bit có trị số lớn nhất. Để đơn giản cho quá trình phân tích ta xét mạch so sánh vạn năng (so sánh đủ) 2 số nhị phân 3 bit



Hình 5: Bộ so sánh vạn năng 2 số nhị phân 3 bit

Để bắt đầu quá trình so sánh ta cho  $E_2$  nhận trị "1", quá trình so sánh thực hiện xong, kết quả của quá trình so sánh được thể hiện ở các trạng thái logic của các đầu ra.

## Chương 3: Các mạch Trigo

Trigo (Flip - Flop) là phần tử cơ bản nhất để từ đó chế tạo ra các mạch dây (mạch logic có nhớ). Mạch Trigo thuộc loại mạch không đồng bộ có hai trạng thái ổn định bền theo thời gian ứng với hai mức logic "1" và "0". Trạng thái của Trigo có thể thay đổi khi tác động xung lên các đầu vào. Trạng thái tương lai của Trigo không những phụ thuộc vào các biến vào mà còn phụ thuộc vào trạng thái hiện tại. Khi ngừng tác động xung lên các đầu vào, trạng thái Trigo giữ nguyên, với đặc điểm này các mạch Trigo được dùng để lưu trữ thông tin dưới dạng mã nhị phân.

### 3.1 Trigo R-S

#### 3.1.1. Trigo R-S không đồng bộ:

Là loại Trigo cơ bản nhất để từ đó tạo ra các loại Trigo khác gồm có 2 đầu vào là R, S và hai đầu ra Q,  $\bar{Q}$  với:

- Q: Đầu ra chính thường được sử dụng.
- $\bar{Q}$ : Đầu ra phụ, luôn thoả mãn  $Q + \bar{Q} = 1$
- R (Reset): Đầu vào xoá.
- S (Set): Đầu vào thiết lập.

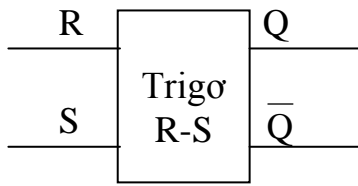
***ý tưởng thiết kế trigơ R-S không đồng bộ theo các điều kiện sau:***

- +  $R_n = S_n = 0$ ; trạng thái của trigơ giữ nguyên  $\rightarrow Q_{n+1} = Q_n$ .
- +  $R_n = 0$ ;  $S_n = 1$ ; đầu ra trigơ nhận giá trị "1"  $\rightarrow Q_{n+1} = 1$ .
- +  $R_n = 1$ ;  $S_n = 0$ ; đầu ra trigơ nhận giá trị "0"  $\rightarrow Q_{n+1} = 0$ .
- +  $R_n = 1$ ;  $S_n = 1$ ; đây là trạng thái cấm, trạng thái Trigo là không xác định, trong bảng trạng thái được đánh dấu bằng dấu "x".

Hoạt động của trigơ R-S tuân theo bảng trạng thái như hình vẽ.

- n: Trạng thái hiện tại
- n + 1: Trạng thái tương lai.
- "-": Giá trị tùy chọn - có thể lấy giá trị "1" hoặc "0".

- x: Trạng thái cấm tại đó giá trị của hàm ra là không xác định.



Hình 1: Sơ đồ mô phỏng

$R_n$	$S_n$	$Q_{n+1}$
0	0	$Q_n$
0	1	1
1	0	0
1	1	x

Hình 2: Bảng trạng thái

$Q_{n+1}$ $R_n, S_n$					
$Q_n$					
		00	01	11	10
0		0	1	x	0
1		1	1	x	0

Hình 3: Bảng chuyển tiếp.

$Q_n$	$Q_{n+1}$	$R_n$	$S_n$
0	0	-	0
0	1	0	1
1	0	1	0
1	1	0	-

Hình 4: Bảng đầu vào kích.

Từ bảng trạng thái hình 2 ta có bảng chuyển tiếp hình 3, bảng kích hình 4.  
Thực hiện nhóm các ô có giá trị “1” trong bảng trạng thái (dạng tuyển) ta có:

$$Q_{n+1} = S_n + \overline{R_n} \cdot Q_n \quad (1)$$

Nhóm các ô có giá trị “0” trong bảng trạng thái (dạng hội) ta có:

$$Q_{n+1} = \overline{R_n} \cdot (Q_n + S_n) \quad (2)$$

$$\text{Từ (1)} \rightarrow Q_{n+1} = \overline{\overline{S_n + Q_n \cdot R_n}} = \overline{\overline{S_n} \cdot \overline{Q_n \cdot R_n}} \quad (3)$$

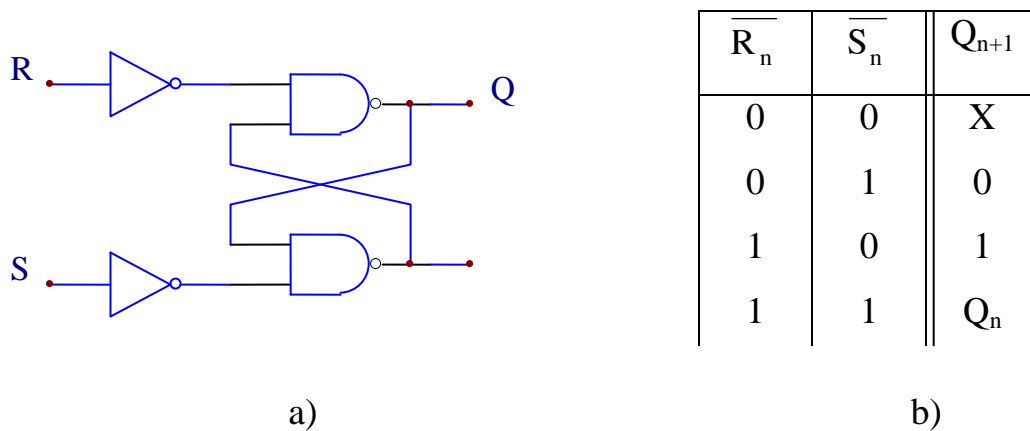
$$\begin{aligned} \text{Từ (2)} \rightarrow Q_{n+1} &= \overline{\overline{R_n} \cdot (Q_n + S_n)}} = \overline{\overline{R_n} \cdot Q_n \cdot S_n}} \\ &\rightarrow \overline{Q_{n+1}} = \overline{\overline{R_n} \cdot (Q_n \cdot S_n)}} \end{aligned} \quad (4)$$

$$\text{Từ (2)} \rightarrow Q_{n+1} = \overline{\overline{R_n} \cdot (Q_n + S_n)}} = \overline{R_n + (Q_n + S_n)} \quad (5)$$

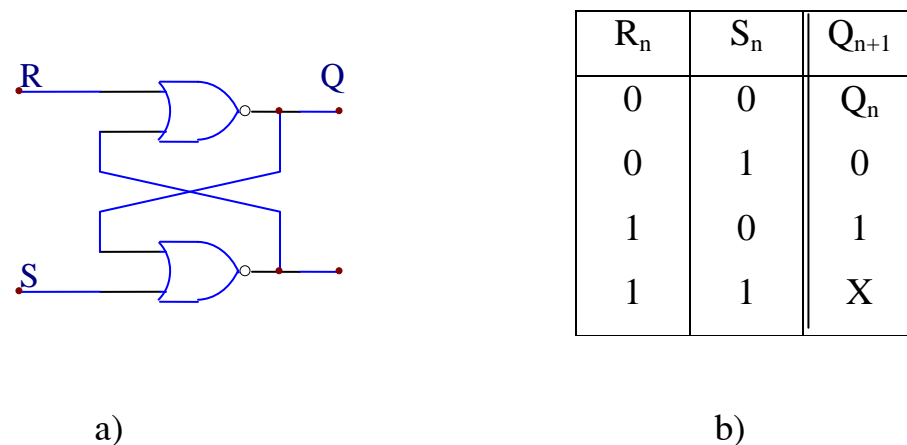
$$\begin{aligned} \text{Từ (1)} \rightarrow Q_{n+1} &= \overline{\overline{S_n + R_n \cdot Q_n}}} = \overline{S_n + \overline{Q_n} + R_n} \\ \overline{Q_{n+1}} &= \overline{S_n + \overline{Q_n} + R_n} \end{aligned} \quad (6)$$



Từ (3) và (4), (5) và (6) cho phép ta xây dựng Trilog RS không đồng bộ từ các phần tử NAND, các phần tử NOR hai lối vào.



Hình 5 : Trilog RS xây dựng từ NAND (a) và bảng trạng thái (b)

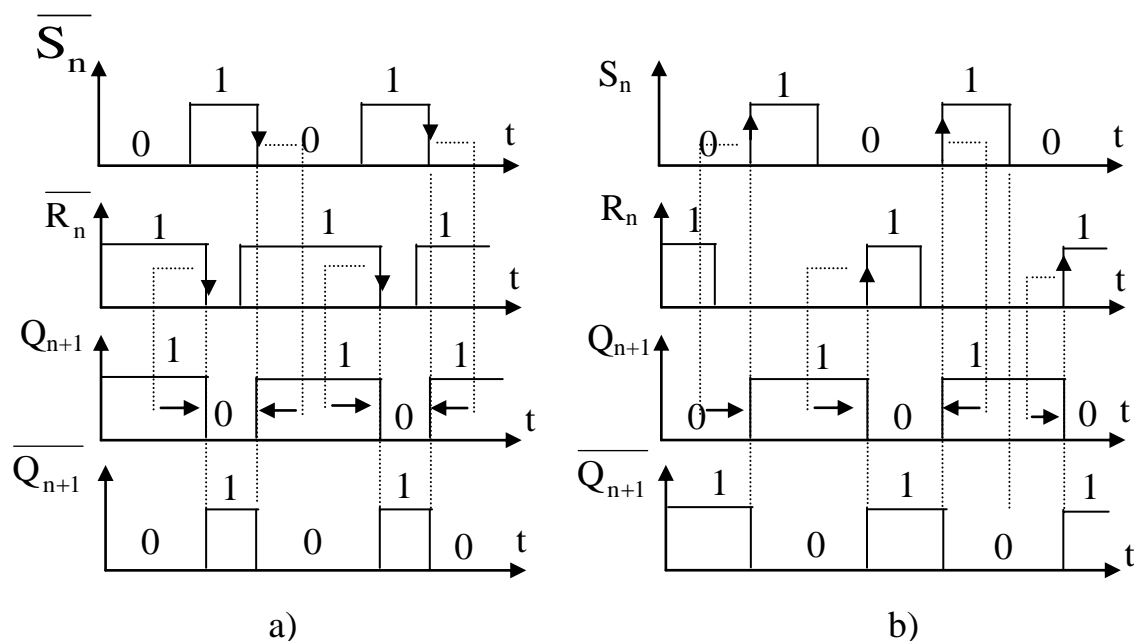


Hình 6 : Trilog RS xây dựng từ NOR (a) và bảng trạng thái (b)

Với các tín hiệu xung vào  $\overline{R}_n, \overline{S}_n$  và  $R_n, S_n$  đã cho kết hợp với các bảng trạng thái hình 5b, 6b ta vẽ được dạng xung ra của các Trilog R-S không đồng bộ (hình 7). Trên giản đồ xung ta có nhận xét:

- Đối với Trilog R-S được xây dựng từ các phần tử NAND trạng thái Trilog chỉ thay đổi (lật) ở các thời điểm xung điều khiển chuyển từ trị "1" về "0". Ta nói đó là loại Trilog chỉ phản ứng với các sườn âm của xung điều khiển đặt tới lối vào.

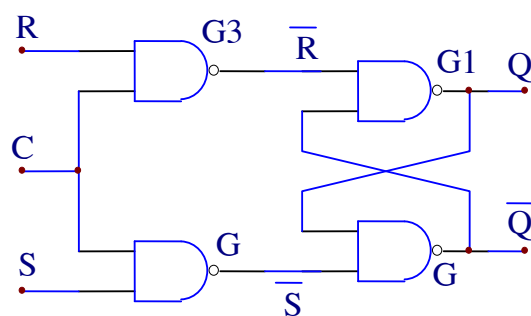
- Đối với Trigo được xây dựng từ các phần tử NOR trạng thái của Trigo chỉ thay đổi ở các thời điểm xung điều khiển chuyển từ trị "0" lên "1" ta nói đó là loại Trigo chỉ phản ứng với các sườn dương của xung điều khiển.



Hình 7: Giải đồ điện áp tín hiệu xung minh họa quá trình làm việc của Trigo R-S được xây dựng từ NAND (a), từ NOR (b)

### 3.1.2. Trigo R-S đồng bộ.

Người ta muốn Trigo chỉ phản ứng vào những thời điểm xác định, điều này được thực hiện bằng cách đưa thêm tới đầu vào tín hiệu phụ C được gọi là tín hiệu đồng bộ. Khi  $C = "0"$  thì  $\bar{R} = \bar{S} = 1$  trạng thái Trigo giữ nguyên còn  $C = "1"$  hoạt động của sơ đồ giống Trigo R-S không đồng bộ như đã phân tích ở phần trên.

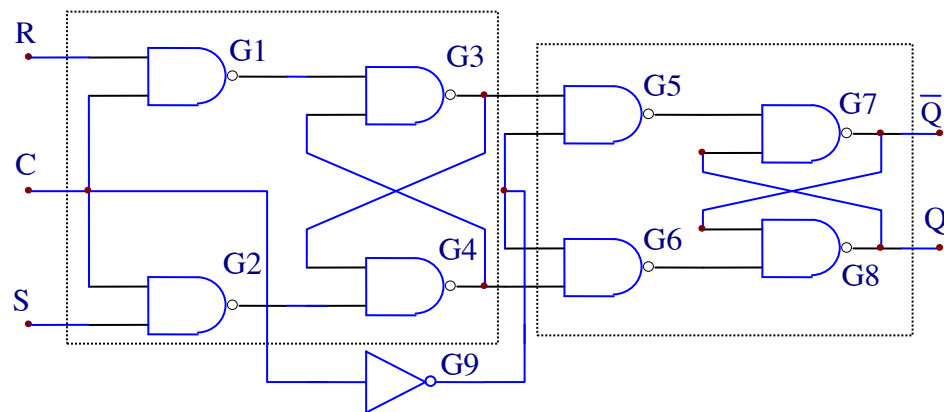


Hình 8: Trigo R-S đồng bộ

### 3.1.3. Trơ chính – phụ (Trơ M - S)

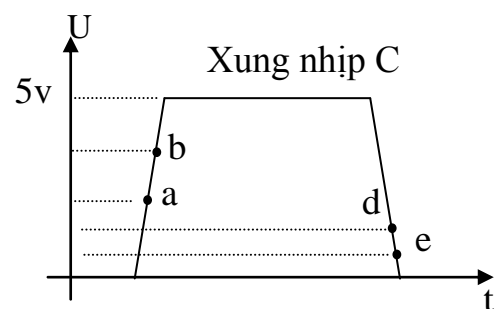
Thông thường để tránh ảnh hưởng của nhiễu, tăng độ tin cậy trong việc ghi đọc thông tin, từ R-S Trơ người ta xây dựng các M-S Trơ bằng cách ghép hai Trơ R-S đồng bộ liên tiếp nhau (hình 1). Khi đó việc ghi thông tin chỉ xảy ra khi lỗi ra bị khoá và ngược lại việc đọc thông tin chỉ xảy ra khi lỗi vào đã bị khoá.

Nhóm các phần tử  $G_1, G_2, G_3, G_4$  tạo nên R-S Trơ chính, nhóm  $G_5, G_6, G_7, G_8$  tạo nên R-S Trơ phụ. Hai nhóm làm việc với hai dãy xung nhịp C ngược pha nhau nhờ cửa đảo  $G_9$ . ở đây chỉ minh hoạ cho quá trình ghi và đọc thông tin qua giản đồ thời gian của xung nhịp C (hình 2).



Hình 1: M-S Trơ cấu trúc từ phần tử NAND

Tại sườn dương khi biên độ xung tăng tương ứng với điểm a Trơ phụ ngắt khỏi Trơ chính nhờ  $\bar{C} = "0"$  khoá  $G_5, G_6$ . Khi biên độ xung tăng tới mức b thông tin đặt tới lỗi vào sẽ được ghi vào Trơ chính. Tại sườn âm khi biên độ xung giảm tới mức d Trơ chính bị ngắt khỏi lỗi vào nhờ khi đó  $C = 0$  khoá  $G_1, G_2$ , lúc đạt tới mức e thông tin được chuyển từ Trơ chính sang Trơ phụ (do  $\bar{C} = "1"$ ) và đặt tới lỗi ra Q (sự thay đổi trạng thái ở lỗi ra chỉ xảy ra trên sườn âm của xung nhịp C)



Hình 2: Quá trình ghi, đọc thông tin được điều khiển bởi xung nhịp C

### 3.2 Trilog D (Delay)

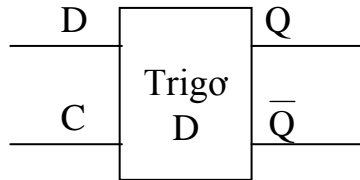
Gồm có hai đầu vào C, D<sub>n</sub> và hai đầu ra Q<sub>n</sub>,  $\overline{Q}_n$  với:

- C: Biến điều khiển (xung nhịp - xung đồng bộ)
- D<sub>n</sub>: Dữ liệu vào.

ý tưởng thiết kế trilog D tuân theo các điều kiện sau:

- + Khi C = 0, trạng thái của trilog giữ nguyên  $\rightarrow Q_{n+1} = Q_n$ .
- + Khi C = 1, giá trị đầu ra trilog nhận giá trị đưa đến đầu vào D  $\rightarrow Q_{n+1} = D_n$ .

Hoạt động của trilog D tuân theo bảng trạng thái như hình 2.



Hình 1: Sơ đồ mô phỏng trilog D.

C	D <sub>n</sub>	Q <sub>n+1</sub>
0	0	Q <sub>n</sub>
0	1	Q <sub>n</sub>
1	0	0
1	1	1

Hình 2: Bảng trạng thái của trilog D.

Q <sub>n+1</sub> \ C, D <sub>n</sub>					
Q <sub>n</sub>		C, D <sub>n</sub>			
		00	01	11	10
0		0	0	1	0
1		1	1	1	0

Hình 3: Bảng chuyển tiếp.

Q <sub>n</sub>	Q <sub>n+1</sub>	C	D
0	0	0	-
0	1	1	1
1	0	1	0
1	1	0	-

Hình 4: Bảng đầu vào kích.

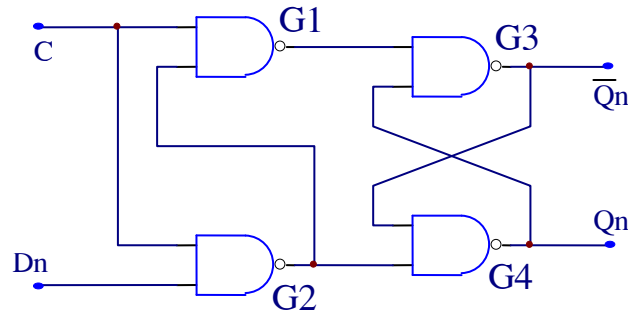
Từ bảng trạng thái hình 2 ta có bảng chuyển tiếp hình 3, bảng kích hình 4. Thực hiện nhóm các ô có giá trị “1” trong bảng trạng thái (dạng tuyển) ta có:

$$\begin{aligned}
 Q_{n+1} &= C.D_n + Q_n.\overline{C} + Q_n.D_n = C.D_n + Q_n[\overline{C} + D_n] = C.D_n + Q_n.\overline{C}.\overline{D}_n \\
 &= \overline{\overline{C.D_n + Q_n.\overline{C}.\overline{D}_n}} = \overline{\overline{C.D_n} . \overline{Q_n} . \overline{\overline{C}.\overline{D}_n}} \quad (1)
 \end{aligned}$$

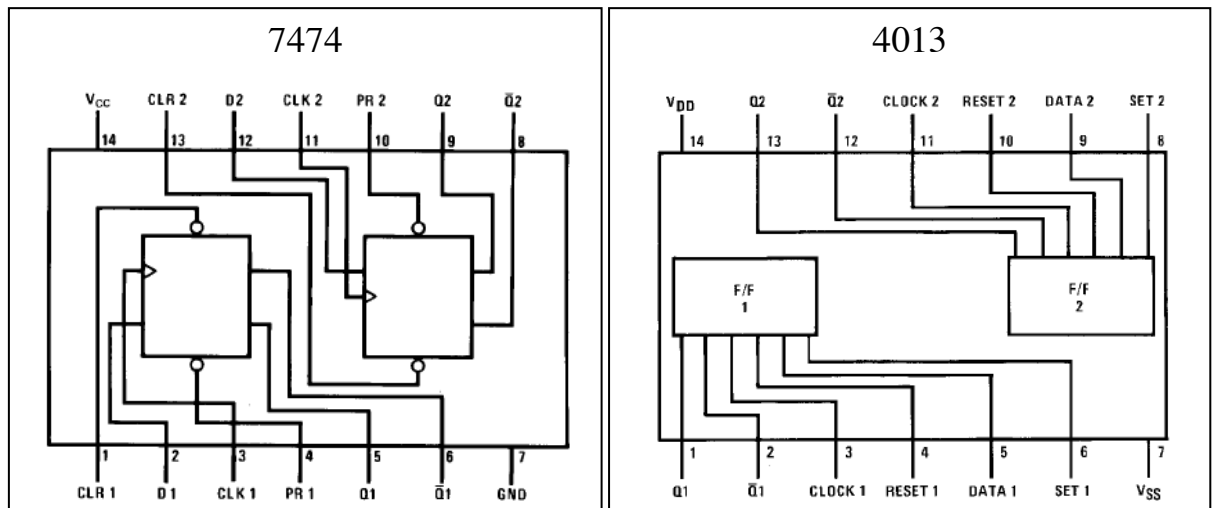
Thực hiện nhóm các ô có giá trị “0” trong bảng trạng thái (dạng hội) ta có:

$$\begin{aligned}
 Q_{n+1} &= [\bar{C} + D_n][Q_n + C][Q_n + D_n] = [\bar{C} + D_n][Q_n + Q_n \cdot D_n + Q_n \cdot C + C \cdot D_n] \\
 &= [\bar{C} + D_n][Q_n(1 + D_n + C) + C \cdot D_n] = (\bar{C} + D_n)(Q_n + C \cdot D_n) = \bar{C} \cdot D_n \cdot Q_n \cdot C \cdot D_n = \bar{C} \cdot C \cdot D_n \cdot Q_n \cdot C \cdot D_n \\
 &\Rightarrow \bar{Q}_{n+1} = \bar{C} \cdot C \cdot D_n \cdot Q_n \cdot C \cdot D_n. \quad (2)
 \end{aligned}$$

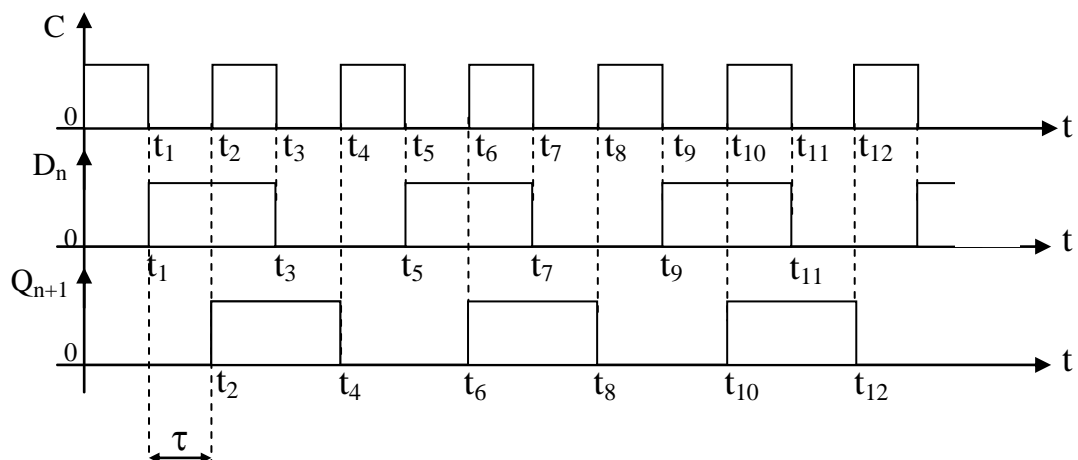
Từ các biểu thức (1) và (2) ta có mạch điện của trigơ D được xây dựng từ các phân tử NAND hai lối vào.



Hình 5: Trigo D được xây dựng từ các phân tử NAND.



Sơ đồ chân của Trigo D họ TTL 7474 và họ CMOS 4013



Hình 6: Giản đồ xung minh họa quá trình làm việc của Trigo D.

Trên giản đồ xung ta nhận thấy xung ra chậm sau so với xung vào một khoảng thời gian là  $\tau$  (chính vì đặc điểm này mà người ta gọi nó là trigơ trễ D-Delay), ở giản đồ trên độ rộng xung ra bằng chu kỳ của dãy xung C.

Trigơ D là phần tử cơ bản xây dựng nên các bộ ghi thông tin nhị phân, các bộ chốt đệm dữ liệu (khi  $C = 1$  thực hiện nhận dữ liệu – chế độ đệm,  $C = 0$  trạng thái trigơ không đổi – chốt dữ liệu).

### 3.3. Trigơ đếm T

Có hai đầu vào T, C, hai đầu ra Q,  $\bar{Q}$ .

-C: Xung đồng bộ (xung nhịp).

-T: Biến điều khiển, thoả mãn yêu cầu sau:

+ T = "0" trạng thái Trigơ giữ nguyên  $\rightarrow Q_{n+1} = Q_n$ .

+ T = "1" Trigơ lật trạng thái  $\rightarrow Q_{n+1} = \bar{Q}_n$ .

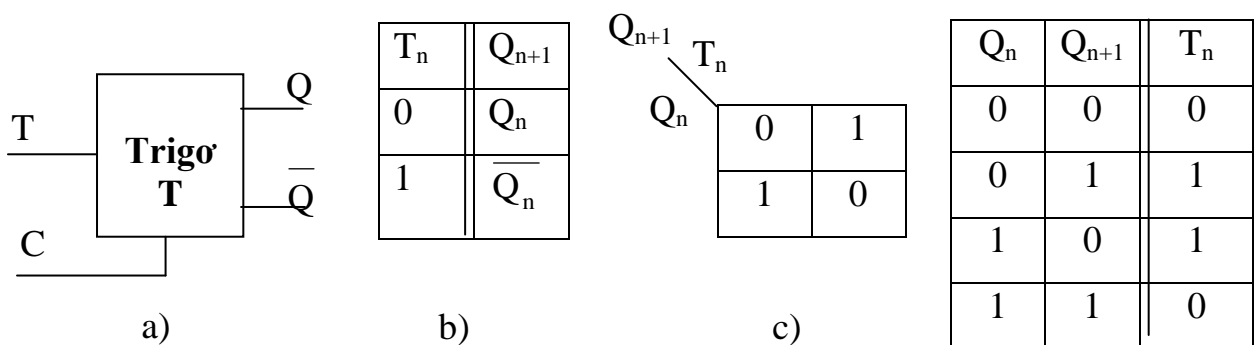
Như vậy hoạt động của Trigơ đếm T tuân theo bảng trạng thái như hình 1b

Từ bảng trạng thái ta có bảng chuyển tiếp hình 1c, bảng kích hình 1d.

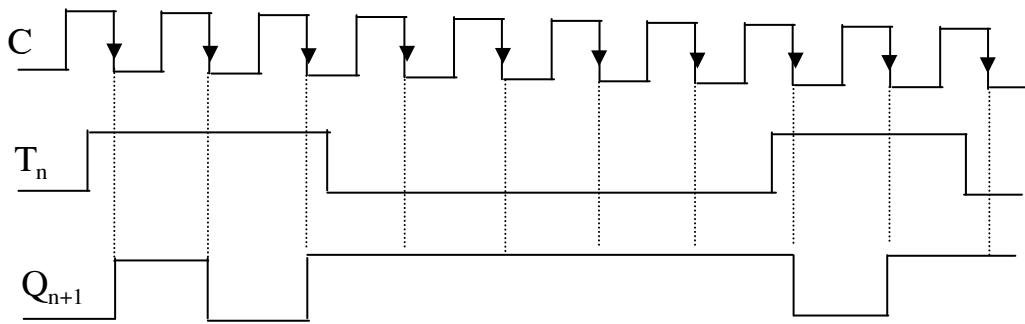
Từ bảng chuyển tiếp ta có:  $Q_{n+1} = T_n \cdot \bar{Q}_n + \bar{T}_n \cdot Q_n$  (1)

$$Q_{n+1} = (T_n + Q_n) \cdot (\bar{T}_n + \bar{Q}_n) \quad (2)$$

Biểu thức (1) và (2) được gọi là phương trình đặc tính của Trigơ đếm T.



Hình 1: a) Mạch mô phỏng; b) Bảng trạng thái;  
c) Bảng chuyển tiếp; d) Bảng kích



Hình 2: Giản đồ xung của Trơ đếm T

### 3.4. Trơ vạn năng j-k

Gồm có 3 đầu vào C,  $J_n$ ,  $K_n$  và hai đầu ra  $Q_n$ ,  $\overline{Q}_n$  với:

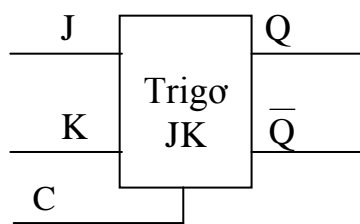
- C: Xung đồng bộ.
- $J_n$ ,  $K_n$ : Các đầu vào điều khiển.

ý tưởng thiết kế trơ vạn năng J, K theo các điều kiện sau:

- +  $K_n = J_n = 0$ , trạng thái của trơ giữ nguyên  $\rightarrow Q_{n+1} = Q_n$ .
- +  $K_n = 0$ ;  $J_n = 1$  đầu ra trơ nhận giá trị "1"  $\rightarrow Q_{n+1} = 1$ .
- +  $K_n = 1$ ;  $J_n = 0$  đầu ra trơ nhận giá trị "0"  $\rightarrow Q_{n+1} = 0$ .
- +  $K_n = 1$ ;  $J_n = 1$  trơ lật trạng thái  $\rightarrow Q_{n+1} = \overline{Q}_n$ .

(Các trạng thái trên xảy ra tại các thời điểm xung đồng bộ C chuyển trạng thái từ "1" về "0" nếu như Trơ được xây dựng từ các phần tử NAND)

Hoạt động của trơ vạn năng J-K tuân theo bảng trạng thái như hình vẽ.



Hình 1: Sơ đồ mô phỏng

$K_n$	$J_n$	$Q_{n+1}$
0	0	$Q_n$
0	1	1
1	0	0
1	1	$\overline{Q}_n$

Hình 2: Bảng trạng thái

		$Q_{n+1} \backslash \begin{matrix} K_n & J_n \end{matrix}$			
		$Q_n$	00	01	11
	0	0	1	1	0
	1	1	1	0	0

Hình 3: Bảng chuyển tiếp.

$Q_n$	$Q_{n+1}$	$K_n$	$J_n$
0	0	-	0
0	1	-	1
1	0	1	-
1	1	0	-

Hình 4: Bảng đầu vào kích.

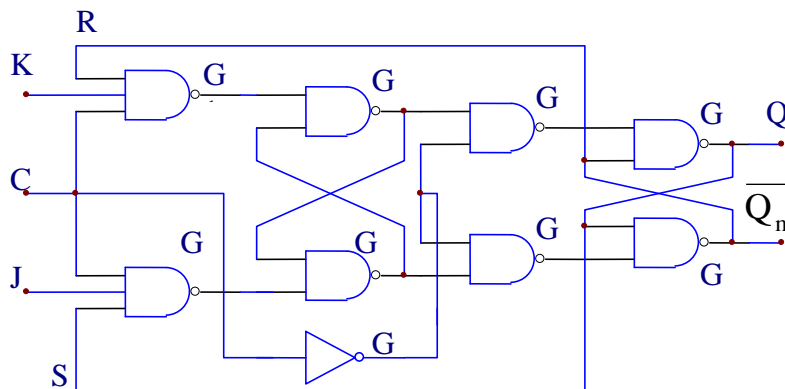
Từ bảng trạng thái hình 2 ta có bảng chuyển tiếp hình 3, bảng kích hình 4.  
Thực hiện nhóm các ô có giá trị “1” trong bảng trạng thái (dạng tuyển) ta có:

$$Q_{n+1} = \overline{K_n} \cdot Q_n + J_n \cdot \overline{Q_n} \quad (1)$$

Nhóm các ô có giá trị “0” trong bảng trạng thái (dạng hội) ta có:

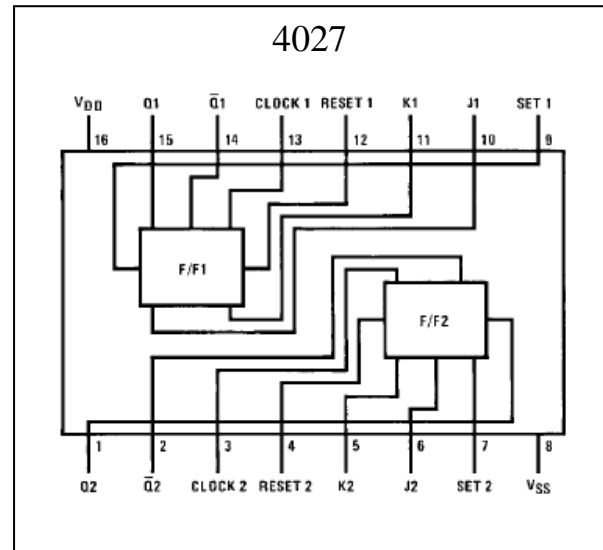
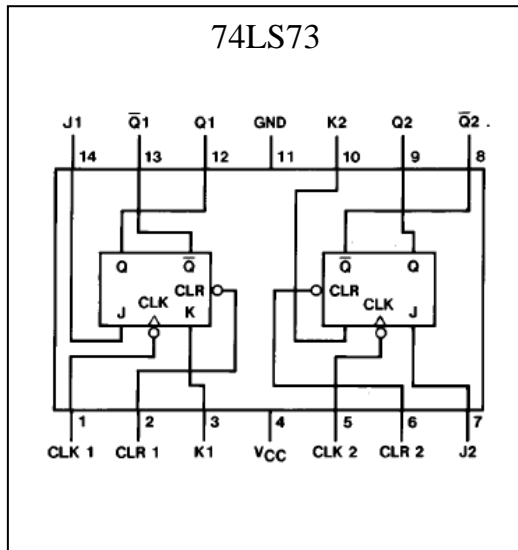
$$Q_{n+1} = (\overline{K_n} + \overline{Q_n}) \cdot (J_n + Q_n) \quad (2)$$

Các biểu thức (1) hoặc (2) được gọi là phương trình đặc tính của Trigo vậ năng J-K

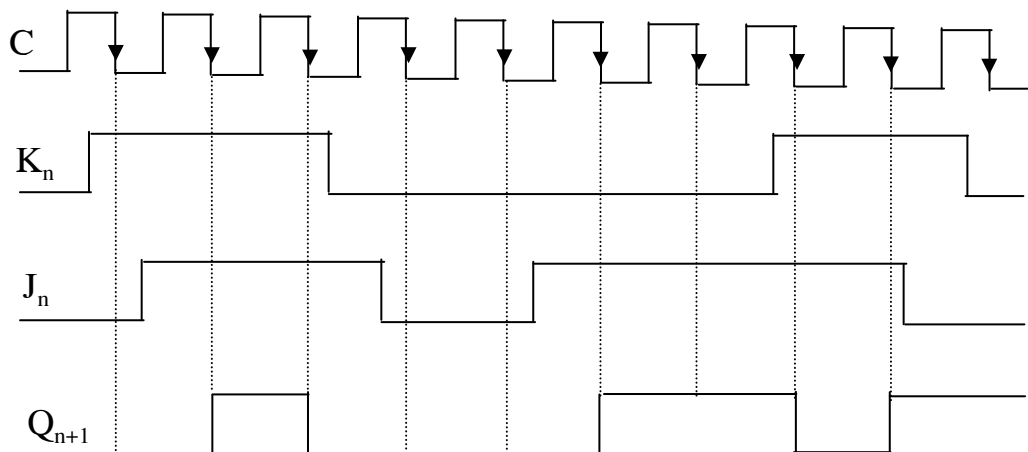


Hình 5: Trigo vậ năng J-K được xây dựng từ các phần tử NAND.





Sơ đồ chân của Trigo J-K họ TTL 74LS73 và họ CMOS 4027



Hình 6: Giải đồ xung minh họa quá trình làm việc của Trigo J-K

Nhờ hai mạch vòng hồi tiếp  $Q = R$ ,  $\bar{Q} = S$  nên khi  $J = K = 1$  thì tín hiệu ra bị đảo qua mỗi sườn âm của xung đồng bộ. Để tăng khả năng điều khiển của Trigo người ta chế tạo có nhiều đầu vào J, K, điều này được thực hiện bằng cách chọn các cổng logic G1, G2 có nhiều đầu vào, trong mạch điện sử dụng số đầu vào J, K không dùng đến sẽ được nối với mức logic "1" để chống nhiễu.

Được gọi là Trigo vạn năng là vì bằng cách thay đổi cách nối các đầu vào J, K và mức logic của chúng sẽ tạo ra các loại Trigo khác.

## 3.6 Chuyển đổi Trigo J-K thành các loại Trigo khác

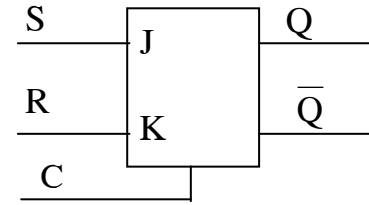
### 3.6.1. Chuyển đổi Trigo J-K thành Trigo R-S

Phương trình đặc tính của Trigo J-K:  $Q_{n+1} = \overline{K_n} \cdot Q_n + J_n \cdot \overline{Q_n}$  (1)

Phương trình đặc tính của Trigo R-S:  $Q_{n+1} = S_n + \overline{R_n} \cdot Q_n$  (2)

Từ (2)  $\rightarrow Q_{n+1} = S_n \cdot (\overline{Q_n} + Q_n) + \overline{R_n} \cdot Q_n = S_n \cdot \overline{Q_n} + S_n \cdot Q_n + \overline{R_n} \cdot Q_n$   
 $= S_n \cdot \overline{Q_n} + \overline{R_n} \cdot Q_n + S_n \cdot Q_n \cdot (\overline{R_n} + R_n) = S_n \cdot \overline{Q_n} + \overline{R_n} \cdot Q_n + S_n \cdot Q_n \cdot \overline{R_n}$   
 $= S_n \cdot \overline{Q_n} + \overline{R_n} \cdot Q_n (1 + S_n) + S_n \cdot Q_n \cdot \overline{R_n}$   
 $= S_n \cdot \overline{Q_n} + \overline{R_n} \cdot Q_n (1 + S_n) + S_n \cdot Q_n \cdot \overline{R_n}$  (3)

So sánh (1) và (3)  $\rightarrow S_n = J_n ; R_n = K_n$ .



Nối Trigo J-K thành R-S

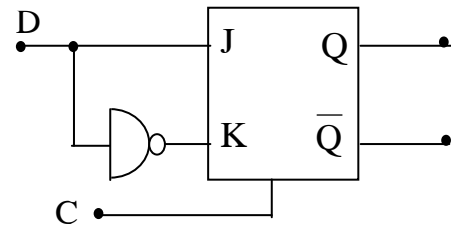
### 3.6.2. Chuyển đổi Trigo J-K thành Trigo D

Phương trình đặc tính của Trigo D:  $Q_{n+1} = D_n$  (4)

Từ (4)  $\rightarrow Q_{n+1} = D_n \cdot (Q_n + \overline{Q_n}) = D_n \cdot Q_n + D_n \cdot \overline{Q_n}$  (5)

So sánh (1) và (5)  $\rightarrow D_n = J_n = \overline{K_n}$

(Nhờ phần tử đảo mà ở sơ đồ bên luôn đảm bảo  $J \neq K$ , mặt khác từ bảng trạng thái của Trigo vận năng JK ta thấy khi  $J \neq K$  thì đầu ra Q luôn thay đổi theo J, mà  $D = J \rightarrow Q_{n+1} = D_n$ )



Nối Trigo J-K thành D

### 3.6.3. Chuyển đổi Trigo J-K thành Trigo đếm T

Phương trình đặc tính của Trigo đếm T :

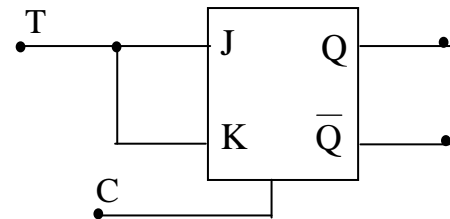
$$Q_{n+1} = \overline{T_n} \cdot Q_n + T_n \cdot \overline{Q_n} \quad (6)$$

So sánh (1) và (6)  $\rightarrow T_n = J_n = K_n$

+ Khi  $T = 0 \rightarrow J = K = 0$  trạng thái

Trigo giữ nguyên  $Q_{n+1} = Q_n$ ;

+ Khi  $T = 1 \rightarrow J = K = 1$  Trigo lật trạng thái :  $Q_{n+1} = \overline{Q_n}$



Nối Trigo J-K thành Trigo đếm T

## Chương 4: Bộ đếm

Tiến hành ghép liên tiếp các Trigơ đếm T ta được bộ đếm thực hiện chức năng đếm số lượng xung đưa tới đầu vào. Tùy thuộc vào kết cấu của bộ đếm mà có thể thực hiện đếm thuận (cộng số lượng xung đầu vào), đếm ngược (trừ xung) hoặc đếm thuận ngược. Tổng quát một bộ đếm đếm được  $(n - 1)$  xung còn xung thứ  $n$  sẽ đưa bộ đếm về trạng thái ban đầu, ta nói số lượng xung đếm được là  $(n - 1)$  xung, bộ đếm có mô đun đếm là  $n$ . Bộ đếm mô đun 2 (Trigơ đếm T) là bộ đếm cơ sở để từ đó xây dựng nên các bộ đếm có mô đun bất kỳ. Kết quả (số lượng xung đếm được) được lấy từ các đầu ra của bộ đếm và được biểu diễn dưới dạng mã nhị phân.

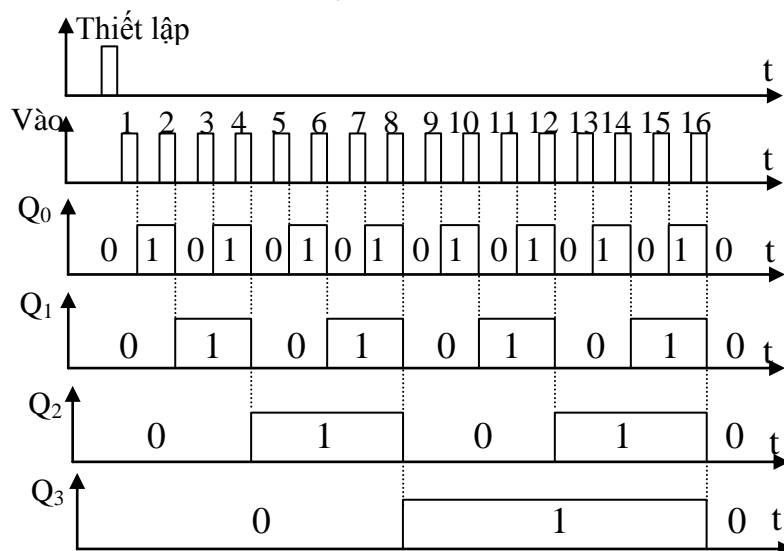
### 4.1. Các bộ đếm nhị phân

Ghép liên tiếp  $N$  trigơ đếm T ta được một bộ đếm nhị phân có mô đun đếm là  $2^N$ , dung lượng bộ đếm (số xung tối đa bộ đếm có thể đếm được) là  $2^N - 1$ . Trong thực tế với nhiều lý do khác nhau thường  $N = 4$  khi đó ta có bộ đếm nhị phân 4 bit, mô đun 16, dung lượng đếm là 15 xung. Giải đồ xung và bảng trạng thái minh họa quá trình làm việc của bộ đếm này được thể hiện ở các hình vẽ sau.

Từ bảng trạng thái (hình 2) ta có các nhận xét sau:

a. Trạng thái của một trigơ bất kỳ sẽ chỉ lật khi đầu ra  $Q$  của trigơ cấp thấp hơn kề nó chuyển giá trị từ 1 về 0.

Với nhận xét này cho phép ta chế tạo bộ đếm nhị phân không đồng bộ (nối tiếp).



Hình 1: Giải đồ thời gian minh họa hoạt động của bộ đếm nhị phân thuận 4 bit, mô đun 16 (lập với sườn âm)

b. Trạng thái của một trigơ bất kỳ sẽ chỉ lật khi tất cả các đầu ra Q của các trigơ cấp thấp hơn đều nhận trị 1.

Với nhận xét này cho phép ta xây dựng bộ đếm nhị phân đồng bộ (song song)

#### 4.1.1. Bộ đếm nhị phân kiểu nối tiếp (không đồng bộ).

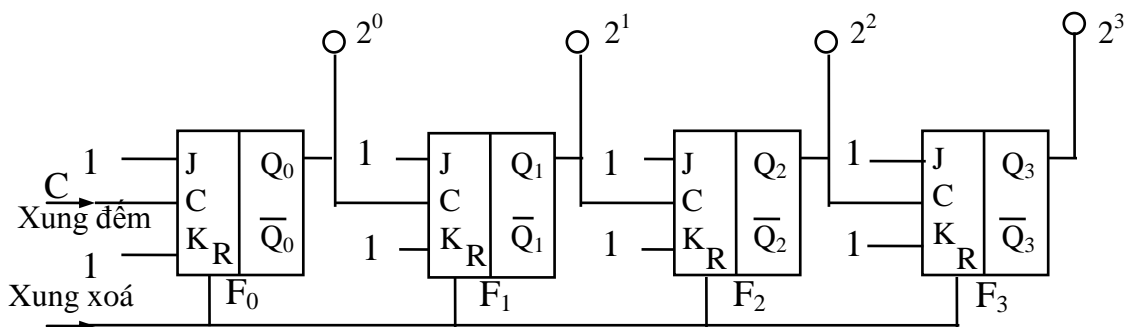
Hình 3 là cấu trúc của bộ đếm nhị phân thuận, môđun 16 kiểu nối tiếp dùng các trigơ JK dạng MS  $F_0 \div F_3$  (cấu tạo từ các phần tử NAND) nối thành trigơ đếm T thực hiện chức năng bảng trạng thái hình 2.

Các đặc điểm chính của bộ đếm hình 3 là:

- Xung cần đếm được đưa vào một cách tuần tự tại lối vào đồng bộ (cửa C) của trigơ đầu tiên  $F_0$ . Đầu ra Q của trigơ trước được nối với đầu vào đồng bộ C của trigơ tiếp theo cấp cao hơn ( $Q_i$  nối với  $C_{i+1}$ ).

Số xung vào	Trạng thái trigơ đếm			
	$(2^3)$	$(2^2)$	$(2^1)$	$(2^0)$
	$F_3$	$F_2$	$F_1$	$F_0$
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
16	0	0	0	0

Hình 2: Bảng trạng thái các trigơ đếm của bộ đếm nhị phân thuận 4 bit môđun 16.



Hình 3 : Bộ đếm nhị phân không đồng bộ (nối tiếp) thuận, môđun 16 dùng 4 Trigơ JK loại MS nối tiếp kiểu Trigơ đếm.

- Xung xoá phải xuất hiện trước dãy xung đếm để thiết lập trạng thái ban đầu  $Q_0 = Q_1 = Q_2 = Q_3 = "0"$ .

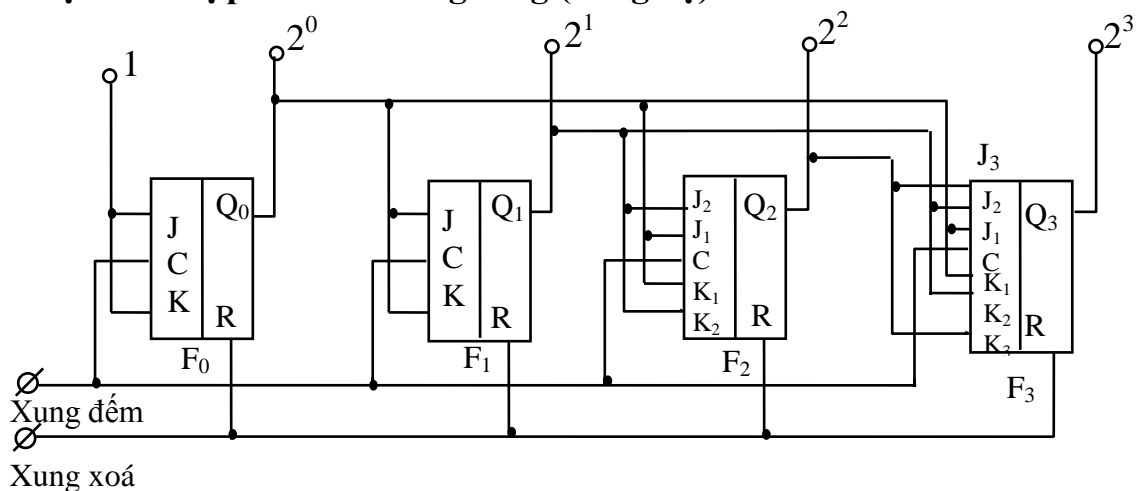
- Để trạng thái của trigơ bất kỳ chỉ lật khi đầu ra Q của trigơ cấp thấp hơn kề nó chuyển đổi từ "1" về "0" thì các đầu vào điều khiển của các trigơ phải cùng nhận trị "1" ( $J=K=1$ ).

- Qua mỗi trigơ  $F_i$  thực hiện chia đôi tần số của dãy xung vào.

- Để tạo ra bộ đếm có dung lượng lớn ta cần tăng số trigơ (số bit) khi đó do có hiện tượng trễ tích lũy giữa dãy xung vào và dãy xung ra làm giảm khả năng đếm nhanh khi số bit tăng dần, độ trễ tích lũy chung bằng tổng độ trễ do các trigơ tạo nên. Đây cũng chính là nhược điểm chính của bộ đếm nhị phân nối tiếp (không đồng bộ).

- Nếu dùng các trigơ được xây dựng từ các phần tử NOR (lật theo sườn dương của xung đồng bộ) thì từ hình 3 ta nhận được bộ đếm nhị phân lùi. Trường hợp dùng các trigơ được xây dựng từ NAND mà tạo ra bộ đếm nhị phân lùi cần nối đầu ra  $\overline{Q}_i$  với  $C_{i+1}$ .

#### 4.1.2. Bộ đếm nhị phân kiểu song song (đồng bộ)



Hình 4 : Bộ đếm nhị phân thuận đồng bộ 4 bit mô đun 16 dùng Trigơ JK .

Hình 4 là cấu trúc của bộ đếm nhị phân đồng bộ 4 bit dùng các trigơ JK loại MS, bộ đếm này có đặc điểm như sau:

a- Các xung đếm được đưa tới đồng thời các lối vào đồng bộ (cửa C) của các trigơ, các đầu vào JK được dùng để điều khiển quá trình lật của mỗi trigơ

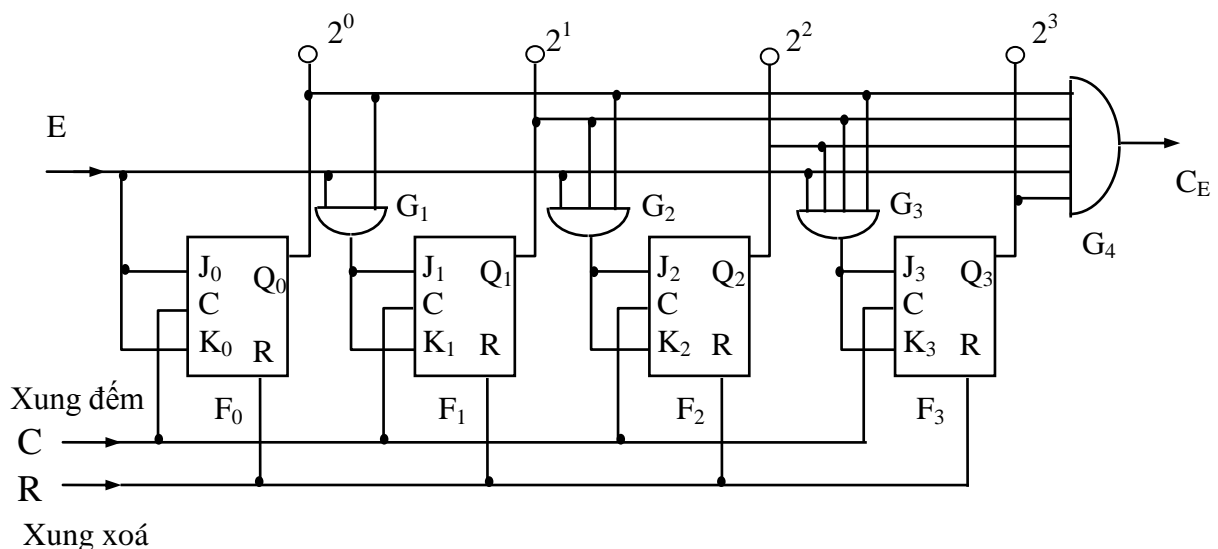
theo trình tự của bảng trạng thái *hình 2*. Do vậy thời gian trễ chung của bộ đếm chỉ bằng thời gian trễ của một trigơ gây nên, đây cũng chính là ưu điểm nổi bật so với bộ đếm nhị phân không đồng bộ.

b- Theo bảng trạng thái *hình 2*:

- + Trigơ  $F_0$  lật trạng thái ứng với mỗi xung nhịp (xung đếm) đưa tới muốn vậy đối với  $F_0$  thì  $J=K=1$ .
- + Trigơ  $F_1$  lật chỉ khi  $Q_0 = "1"$  đồng thời có xung đếm đưa tới.
- + Trigơ  $F_2$  lật chỉ khi  $Q_1 = Q_0 = "1"$  đồng thời có xung đếm đưa tới.
- + Trigơ  $F_3$  lật chỉ khi  $Q_2 = Q_1 = Q_0 = "1"$  đồng thời có xung đếm đưa tới.

Các đầu vào điều khiển JK được nối với các đầu ra  $Q_0, Q_1, Q_2$  cho phép thực hiện đồng thời các điều kiện trên.

Bộ đếm nhị phân đồng bộ có kết cấu *hình 4* với ưu điểm là đơn giản tuy nhiên tồn tại nhược điểm là khi bộ đếm cần dung lượng lớn thì các trigơ ở bit trọng số cao cần rất nhiều các đầu vào điều khiển JK điều này làm cho mạch điện phức tạp. Để khắc phục nhược điểm này ta dùng sơ đồ *hình 5*.



Hình 5: Bộ đếm nhị phân thuận 4 bit, mô đun 16 dùng các trigơ vạn năng JK kết hợp với các cổng logic, có logic tạo nhớ.

Từ *hình 5* ta thấy:

$$C_E = Q_0 \cdot Q_1 \cdot Q_2 \cdot Q_3 \cdot E ; \quad J_3 = K_3 = Q_0 \cdot Q_1 \cdot Q_2 \cdot E ; \quad J_2 = K_2 = Q_0 \cdot Q_1 \cdot E$$

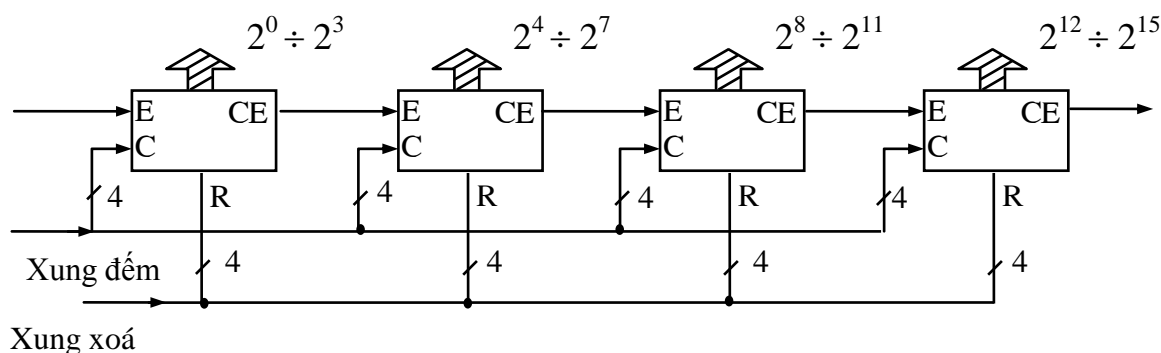
$$J_1 = K_1 = Q_0 \cdot E ; \quad J_0 = K_0 = E.$$

- E: là tín hiệu điều khiển, E = "1" cho phép bộ đếm làm việc, E = "0" trạng thái của bộ đếm không đổi.

- Trạng thái của một trigơ bất kỳ sẽ chỉ lật khi J = K = 1 và xung đồng bộ chuyển đổi từ "1" về "0" (giả thiết trigơ được xây dựng từ các phần tử NAND), điều này cũng có nghĩa là trạng thái của một trigơ bất kỳ sẽ chỉ lật khi tất cả các đầu ra Q của các trigơ cấp thấp hơn đều nhận trị "1".

- Tín hiệu nhớ  $C_E$  nhận trị "1" khi bộ đếm đã đầy ( $Q_0=Q_1=Q_2=Q_3=E="1"$ )

Khi cần bộ đếm có dung lượng lớn hơn người ta tiến hành ghép liên tiếp các mô đun đếm với mỗi mô đun là bộ đếm nhị phân đồng bộ 4 bit mô đun 16 như hình 5.



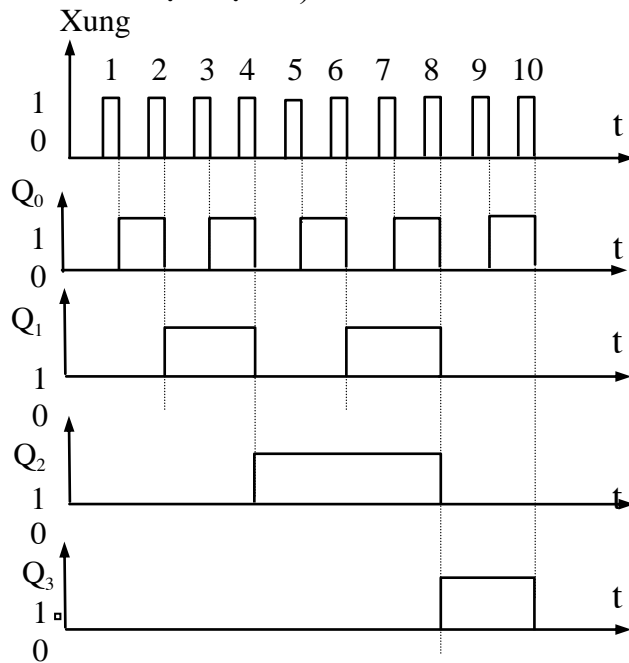
Hình 6: Bộ đếm nhị phân đồng bộ 16 bit

Hình 6 đưa ra cấu trúc của bộ đếm nhị phân đồng bộ 16 bit, dung lượng  $2^{16}-1$  xung được xây dựng bằng cách ghép liên tiếp 4 bộ đếm 4 bit có logic tạo nhớ

## 4.2. Bộ đếm nhị phân có mô đun đếm bất kỳ

Được xây dựng dựa trên cơ sở các bộ đếm tổng nhị phân (bộ đếm có mô đun đếm đầy đủ) sau khi đã loại bỏ các trạng thái dư. Số trạng thái dư với mô đun đếm n là  $S = 2^N - n$  (N - số trigơ hay số bit). Số trigơ đếm sử dụng được chọn theo số lượng cực tiểu của S. Việc loại bỏ các trạng thái dư được thực hiện cưỡng bức nhờ các mạch vòng hồi tiếp thích hợp, vào các thời điểm thích hợp theo ý định thiết kế. Cụ thể ta xét bộ đếm nhị phân mô đun 10 (còn gọi là bộ đếm nhị - thập phân hay BCD), nó được xây dựng dựa trên cơ sở bộ đếm nhị

phần 4 bit mô đun 16 sau khi đã loại 6 trạng thái dư, nhờ các mạch vòng hồi tiếp thích hợp mà đến xung đếm thứ 10 sẽ đưa bộ đếm về trạng thái ban đầu (các đầu ra đều nhận trị "0").

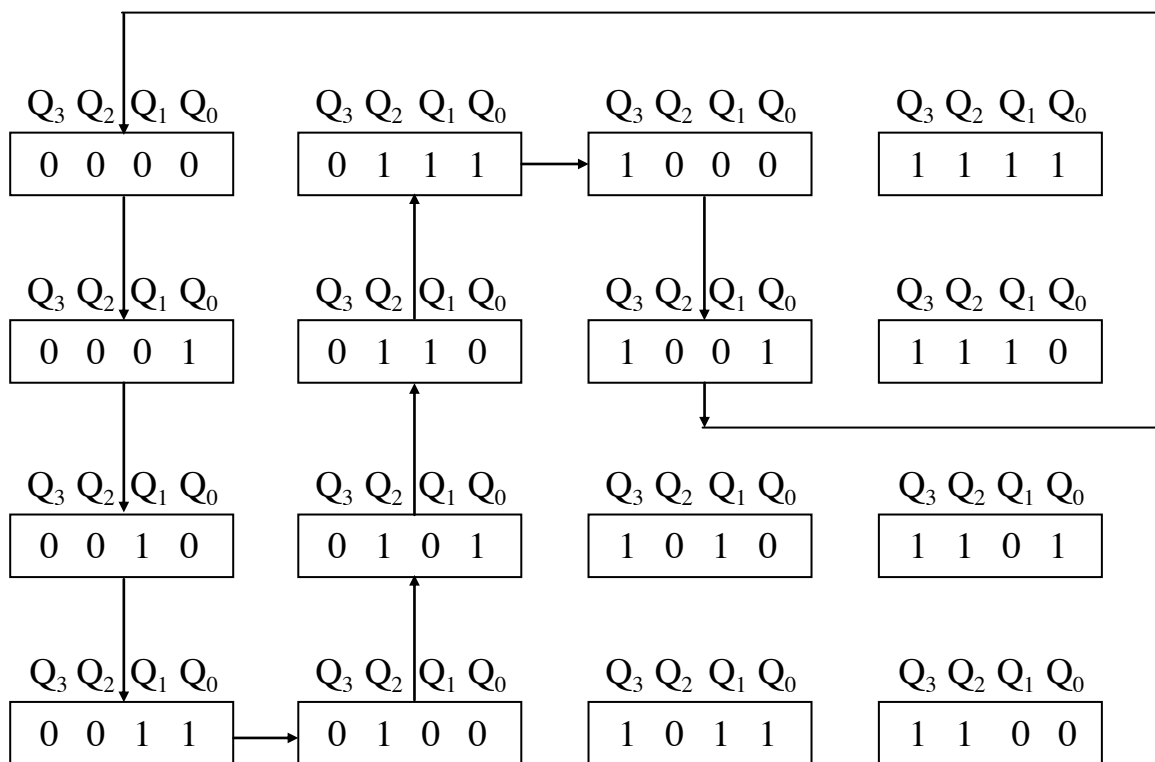


Hình 8: Giản đồ thời gian minh hoạ

Số Xung vào	Trạng thái các Trơ đếm			
	( $2^3$ ) F <sub>3</sub>	( $2^2$ ) F <sub>2</sub>	( $2^1$ ) F <sub>1</sub>	( $2^0$ ) F <sub>0</sub>
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

Hình 7 : Bảng trạng thái của bộ đếm nhị phân mô đun 10

Từ bảng trạng thái *hình 7* ta có đồ hình chuyển đổi trạng thái *hình 8*



Hình 9: Đồ hình chuyển đổi trạng thái của bộ đếm mô đun 10.



Để số trạng thái dư là ít nhất, ta dùng 4 trigơ vạn năng JK để xây dựng bộ đếm

Dựa vào bảng đầu vào kích của trigơ JK và bảng trạng thái *hình 7* ta đưa ra bảng trạng thái *hình 9* minh họa quá trình hoạt động của các trigơ (với từng trạng thái là các giá trị tương ứng của các đầu vào điều khiển J, K) trong bộ đếm bộ đếm nhị phân mô đun 10. Như đã biết trong bộ đếm tồn tại các mạch hồi tiếp với lượng vào là trạng thái ra của các trigơ, lượng ra điều khiển các đầu vào JK nhờ đó quá trình làm việc của bộ đếm sẽ tuân theo bảng trạng thái *hình 7*. Để xây dựng được các mạch hồi tiếp điều khiển sự hoạt động của các trigơ ta coi  $J_0 \div J_3$ ,  $K_0 \div K_3$  là các hàm ra,  $Q_0 \div Q_3$  là các biến vào. Để tìm quan hệ giữa các hàm ra với các biến vào đồng thời đưa chúng về dạng tối giản ta dùng phương pháp bìa các nô (có tận dụng 6 trạng thái dư để tối giản hàm).

XD	Trạng thái các trigơ đếm								Trạng thái các hàm đầu vào kích của các trigơ							
	Hiện tại				Tiếp theo											
	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	Q' <sub>3</sub>	Q' <sub>2</sub>	Q' <sub>1</sub>	Q' <sub>0</sub>	J <sub>3</sub>	K <sub>3</sub>	J <sub>2</sub>	K <sub>2</sub>	J <sub>1</sub>	K <sub>1</sub>	J <sub>0</sub>	K <sub>0</sub>
0	0	0	0	0	0	0	0	1	0	—	0	—	0	—	1	—
1	0	0	0	1	0	0	1	0	0	—	0	—	1	—	—	1
2	0	0	1	0	0	0	1	1	0	—	0	—	—	0	1	—
3	0	0	1	1	0	1	0	0	0	—	1	—	—	1	—	1
4	0	1	0	0	0	1	0	1	0	—	—	0	0	—	1	—
5	0	1	0	1	0	1	1	0	0	—	—	0	1	—	—	1
6	0	1	1	0	0	1	1	1	0	—	—	0	—	0	1	—
7	0	1	1	1	1	0	0	0	1	—	—	1	—	1	—	1
8	1	0	0	0	1	0	0	1	—	0	0	—	0	—	1	—
9	1	0	0	1	0	0	0	0	—	1	0	—	0	—	—	1

Hình 10: Bảng trạng thái minh họa quá trình làm việc của bộ đếm nhị phân mô đun 10 được xây dựng từ 4 trigơ vạn năng JK.

Chú ý: Trong bảng các ô ghi dấu "\_" , "X" tại đó giá trị của hàm là tùy chọn, không xác định. Trong quá trình tối giản hàm các ô được ghi như trên có thể lấy trị "1" hoặc "0" tùy từng trường hợp cụ thể.

$J_3$ $Q_3Q_2$		$Q_1Q_0$			
		00	01	11	10
00	00	0	0	0	0
01	01	0	0	1	0
11	11	x	x	x	x
10	10	-	-	x	x

$$J_3 = Q_1 \cdot Q_0 \cdot Q_2$$

$K_3$ $Q_3Q_2$		$Q_1Q_0$			
		00	01	11	10
00	00	-	-	-	-
01	01	-	-	-	-
11	11	x	x	x	x
10	10	0	1	x	x

$$K_3 = Q_0$$

$J_2$ $Q_3Q_2$		$Q_1Q_0$			
		00	01	11	10
00	00	0	0	1	0
01	01	-	-	-	-
11	11	x	x	x	x
10	10	0	0	x	x

$$J_2 = Q_1 \cdot Q_0$$

$K_2$ $Q_3Q_2$		$Q_1Q_0$			
		00	01	11	10
00	00	-	-	-	-
01	01	0	0	1	0
11	11	x	x	x	x
10	10	-	-	x	x

$$K_1 = Q_1 \cdot Q_0$$

$J_1$ $Q_3Q_2$		$Q_1Q_0$			
		00	01	11	10
00	00	0	1	-	-
01	01	0	1	-	-
11	11	x	x	x	x
10	10	0	0	x	x

$$J_1 = \overline{Q_3} \cdot Q_0$$

$K_1$ $Q_3Q_2$		$Q_1Q_0$			
		00	01	11	10
00	00	-	-	1	0
01	01	-	-	1	0
11	11	x	x	x	x
10	10	-	-	x	x

$$K_1 = Q_0$$



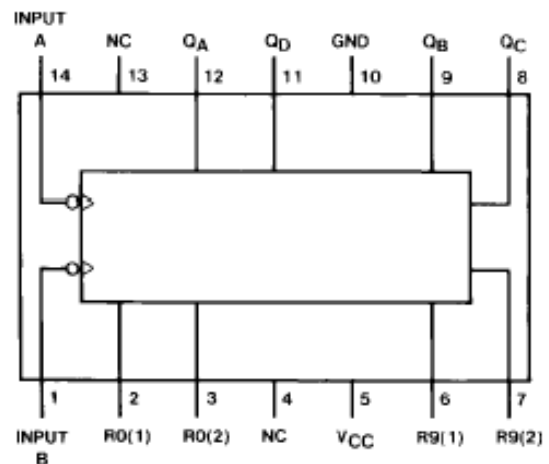
-  $J_0 = K_0 = 1 \rightarrow Q_0 = 0$  ; (Trigơ lật trạng thái).

Như vậy sau xung nhịp thứ 10 đưa bộ đếm về trạng thái ban đầu.

Do nhu cầu sử dụng bộ đếm modul 10 rất phổ biến, trên thị trường hiện nay đang sử dụng bộ đếm modul 10 được tích hợp trên một IC họ TTL - 74LS90. Dưới đây là sơ đồ chân của vi mạch 74LS90 và các bảng trạng thái của chúng.

Bảng thiết lập trạng thái đầu ra

Reset Inputs				Output			
R0(1)	R0(2)	R9(1)	R9(2)	Q <sub>D</sub>	Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>
H	H	L	X	L	L	L	L
H	H	X	L	L	L	L	L
X	X	H	H	H	L	L	H
X	L	X	L	COUNT			
L	X	L	X	COUNT			
L	X	X	L	COUNT			
X	L	L	X	COUNT			



Sơ đồ chân của 74LS90

Để đặt cho 74LS90 ở chức năng đếm thì các chân R0(1), R0(2), R9(1) và R9(2) thông thường sẽ được nối với chân GND.

Khi nối chân 12 (Q<sub>A</sub>) với chân 1 (INPUT B) ta được bảng trạng thái của 74LS90 như sau:

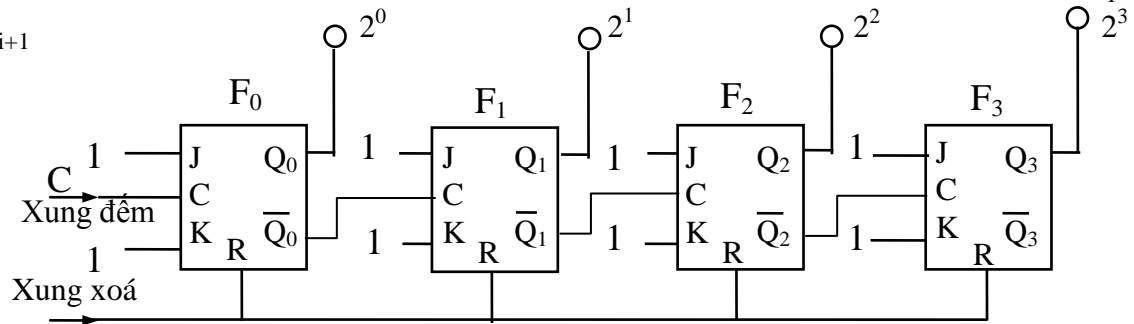
Count	Output			
	Q <sub>D</sub>	Q <sub>C</sub>	Q <sub>B</sub>	Q <sub>A</sub>
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H

Count	Output			
	Q <sub>A</sub>	Q <sub>D</sub>	Q <sub>C</sub>	Q <sub>B</sub>
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	H	L	L	L
6	H	L	L	H
7	H	L	H	L
8	H	L	H	H
9	H	H	L	L

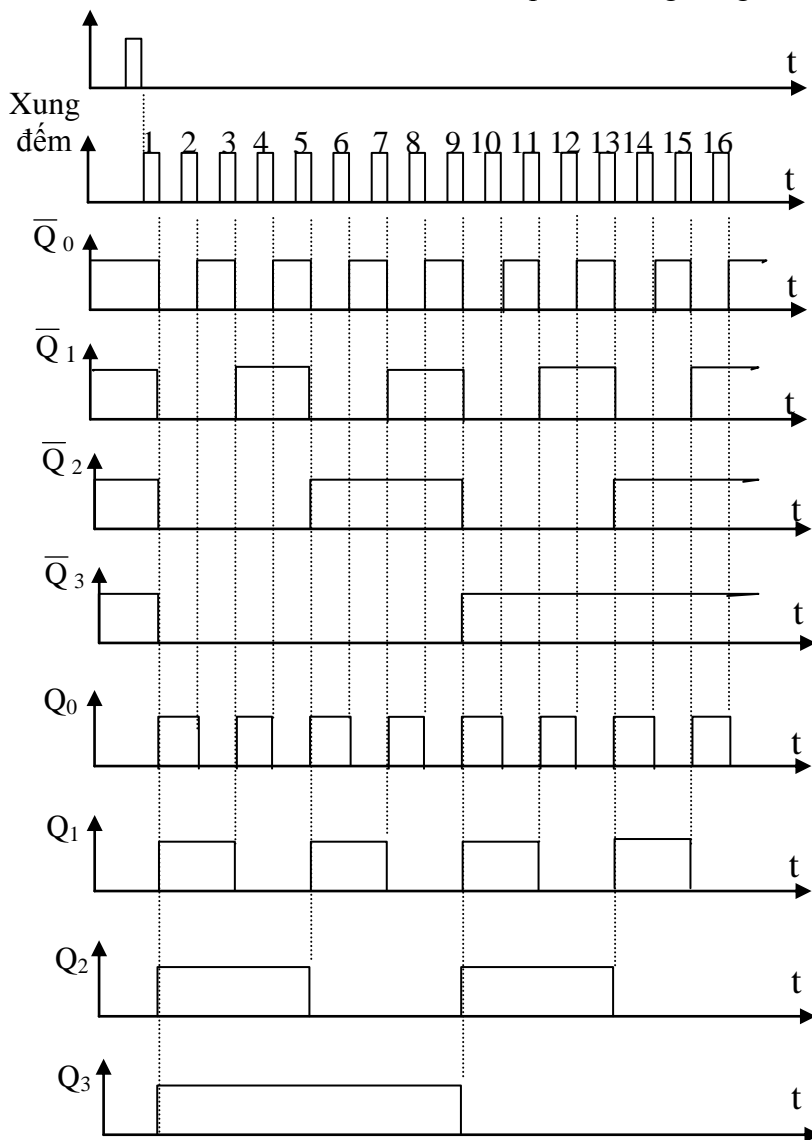
Khi nối chân 11 (Q<sub>D</sub>) với chân 14 (INPUT A) ta được bảng trạng thái của 74LS90 như sau:

### 4.3. Bộ đếm nhị phân ngược (trừ xung)

Nguyên lý làm việc tương tự như bộ đếm nhị phân thuận, giá trị nhị phân của bộ đếm giảm dần khi có xung đưa tới. ở bộ đếm nhị phân ngược nối tiếp mà các trigơ được xây dựng từ các phần tử NAND người ta thực hiện nối  $\bar{Q}_i$  với  $C_{i+1}$



Hình 11 : Bộ đếm nhị phân không đồng bộ lùi 4 bit mô đun 16



Hình 12: Giảm đồ thời gian minh họa hoạt động của bộ đếm nhị phân không đồng bộ lùi 4 bit.

Số xung vào	Trạng thái trigơ đếm			
	(2 <sup>3</sup> )	(2 <sup>2</sup> )	(2 <sup>1</sup> )	(2 <sup>0</sup> )
	F <sub>3</sub>	F <sub>2</sub>	F <sub>1</sub>	F <sub>0</sub>
0	0	0	0	0
1	1	1	1	1
2	1	1	1	0
3	1	1	0	1
4	1	1	0	0
5	1	0	1	1
6	1	0	1	0
7	1	0	0	1
8	1	0	0	0
9	0	1	1	1
10	0	1	1	0
11	0	1	0	1
12	0	1	0	0
13	0	0	1	1
14	0	0	1	0
15	0	0	0	1
16	0	0	0	0

Hình 13: Bảng trạng thái các trigơ đếm của bộ đếm nhị phân lùi 4 bit.

#### 4.4. Bộ đếm nhị phân thuận ngược

Người ta tạo ra biến điều khiển L với chức năng sau:

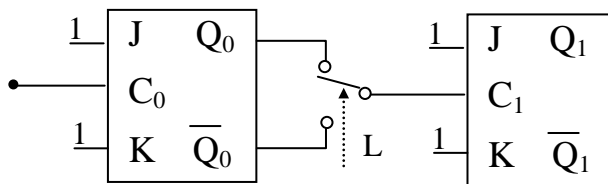
- L = 1 thực hiện đếm thuận  $\rightarrow Q_i$  nối với  $C_{i+1}$ .
- L = 0 thực hiện đếm ngược  $\rightarrow \overline{Q_i}$  nối với  $C_{i+1}$ .

ý tưởng trên được trình bày ở bảng trạng thái hình 14. Từ bảng trạng thái ta có:

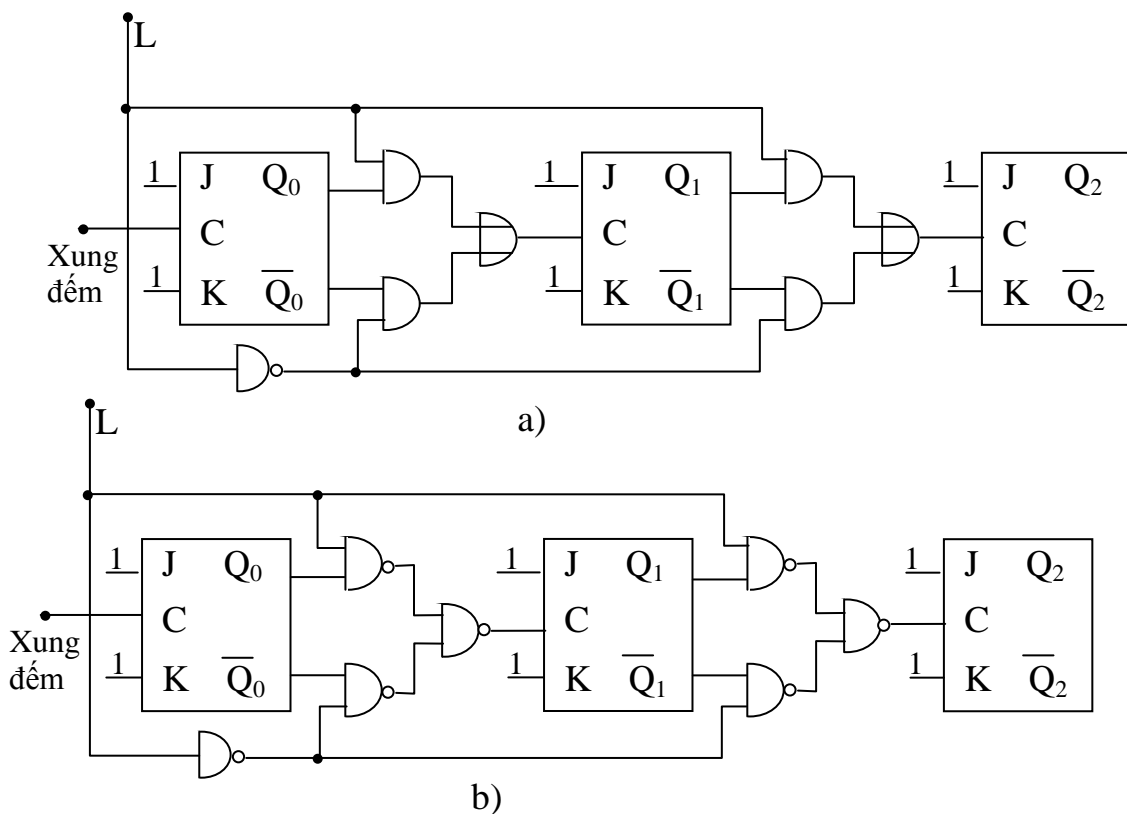
$$C_{i+1} = \overline{L} \cdot \overline{Q_i} + L \cdot Q_i = \overline{\overline{\overline{L} \cdot \overline{Q_i}} \cdot \overline{\overline{L} \cdot \overline{Q_i}}} \cdot \overline{\overline{L} \cdot \overline{Q_i}}$$

L	$C_{i+1}$
0	$\overline{Q_i}$
1	$Q_i$

Hình 14: Bảng trạng thái của mạch điều khiển đếm thuận, đếm ngược



Hình 15: Mô phỏng chuyển mạch

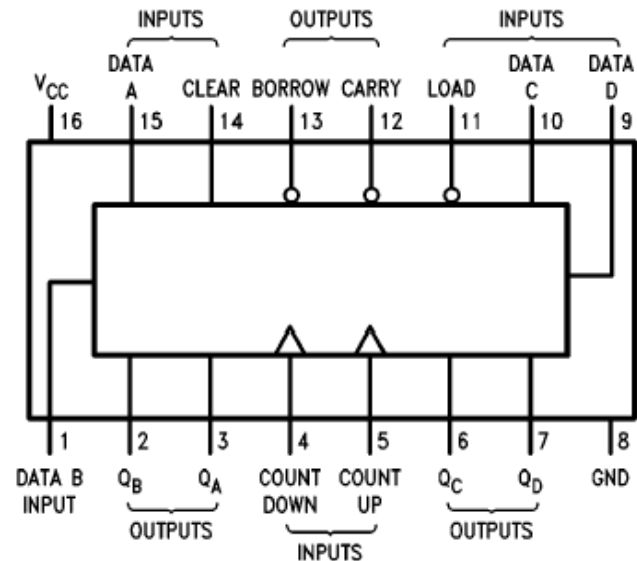


Hình 16: Bộ đếm nhị phân thuận ngược 3 bit mô đun 8 với chuyển mạch dùng các cổng logic hỗn hợp (a), chỉ dùng cổng NAND.

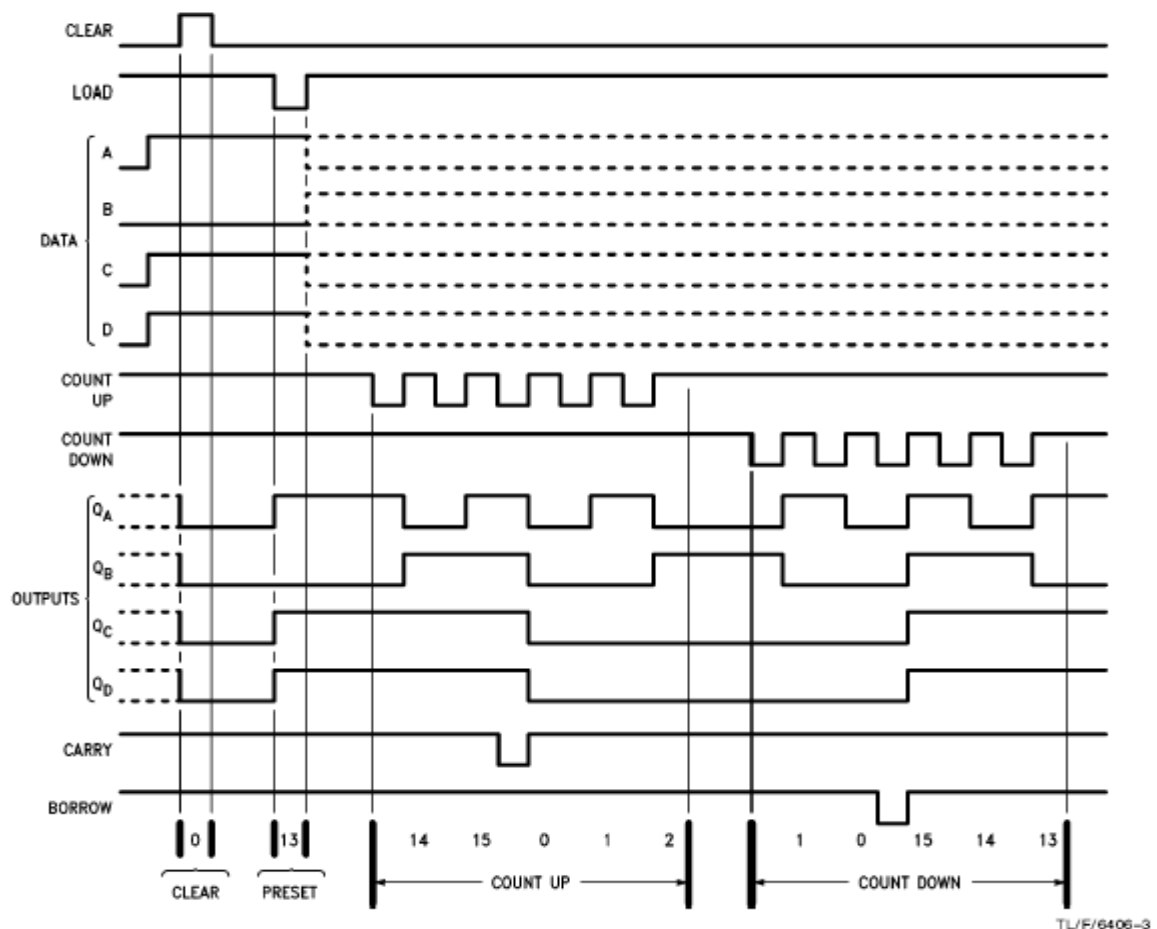
Sau đây là phần giới thiệu một loại bộ đếm nhị phân thuận/ngược 4 bit được tích hợp trên IC họ TTL: 74LS193

Chức năng các chân:

- COUNT DOWN : đầu vào của xung đếm lùi.
- COUNT UP : đầu vào của xung đếm tiến.
- A,B,C,D: 4 bit đầu vào để đặt giá trị ban đầu cho bộ đếm.
- LOAD: khi chân này chuyển về mức logic 0 sẽ đặt 4 bit đầu ra bằng giá trị của 4 bit A,B,C,D.
- $Q_A, Q_B, Q_C, Q_D$ : 4 bit nhị phân đầu ra.
- CARRY: chân này sẽ nhận mức “0” khi bộ đếm tiến tràn (4 bit đầu ra = 1111).
- BORROW: chân này sẽ nhận mức “0” khi bộ đếm lùi về 0 (4 bit đầu ra = 0000)

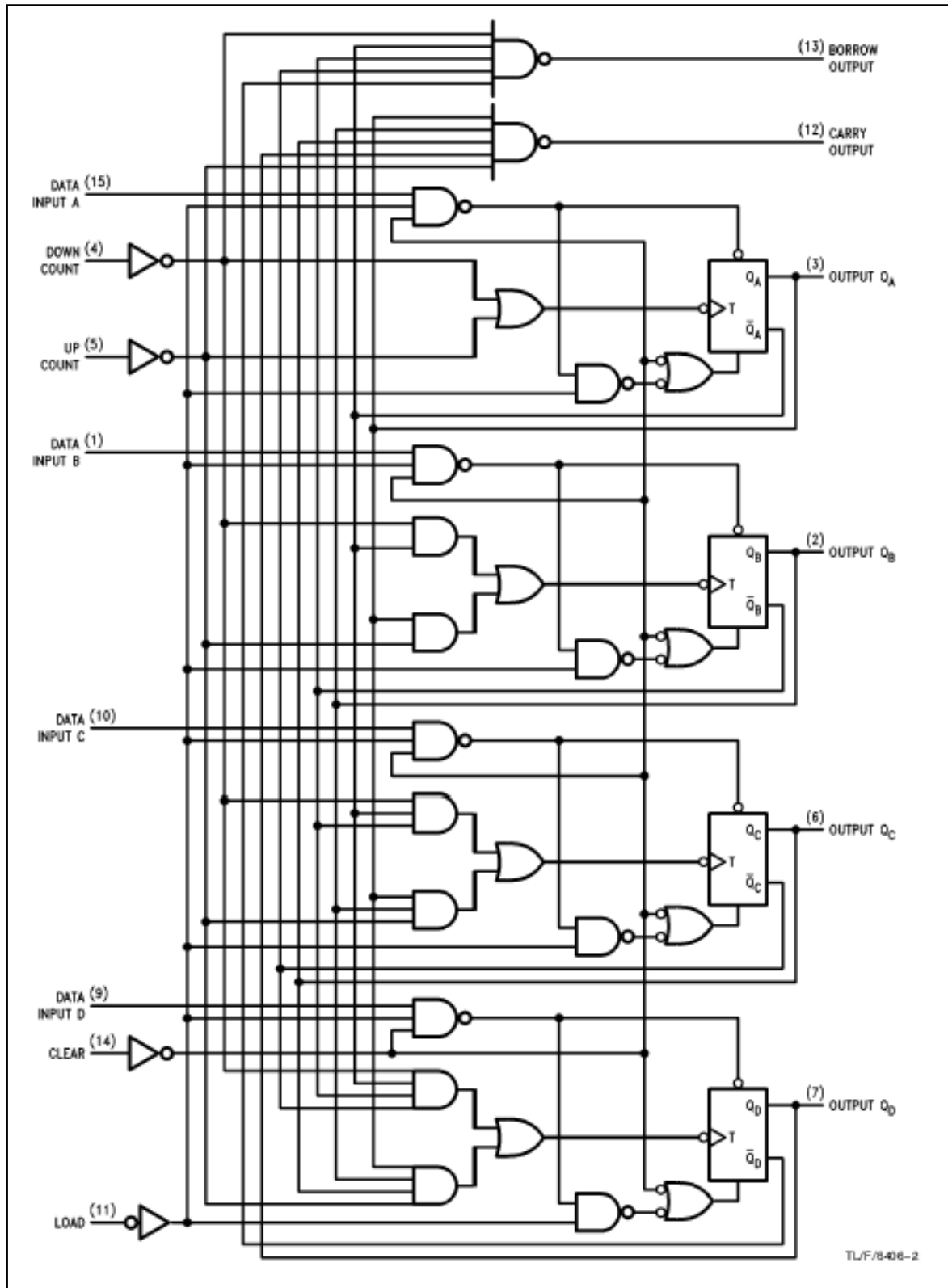


Sơ đồ chân của 74LS193



Giải đồ xung mô tả nguyên làm việc của 74LS193

### Sơ đồ logic của vi mạch 74LS193



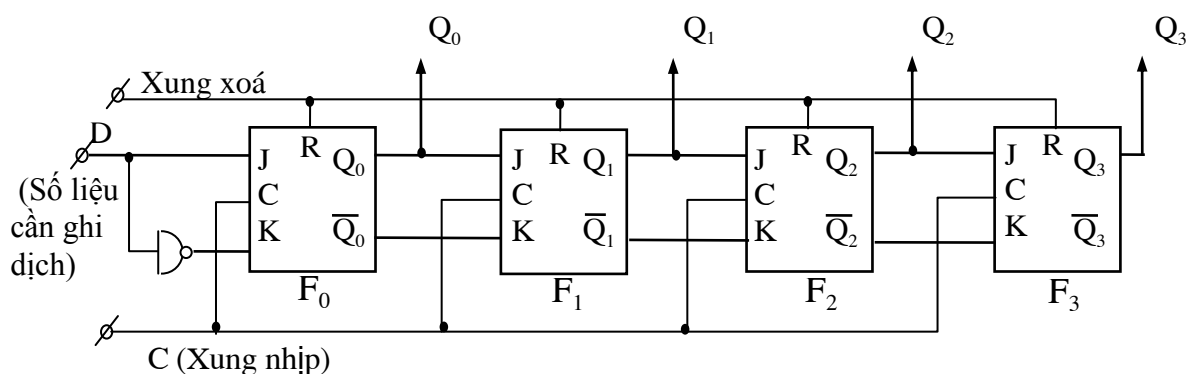


## CHƯƠNG 5: CÁC MẠCH LOGIC TỔ HỢP

### 5.1. Bộ ghi dịch

Các thông tin nhị phân có thể được lưu trữ nhờ các trigơ có vai trò như một ô nhớ số nhị phân. Đầu nối tiếp nhiều trigơ D ta nhận được một bộ ghi dịch: Mỗi khi có xung nhịp đặt vào cửa nhịp, thông tin vào cửa D sẽ được dịch từ một ô sang ô tiếp theo từ  $F_0 \div F_3$ , đây là phương pháp ghi nối tiếp thông tin vào các ô nhớ.

Bộ ghi dịch 4 bit nhị phân với khả năng ghi và nhớ 4 bit thông tin cấu tạo từ 4 trigơ JK nối kiểu trigơ D được cho trên hình 1. Có hai khả năng lấy thông tin ra khỏi bộ ghi dịch: lấy ra đồng thời ở  $Q_0 \div Q_3$  (kiểu song song) sau 4 nhịp ghi nối tiếp hoặc lấy ra lần lượt tại  $Q_3$  các thông tin vào trước đó 4 nhịp.



Hình 2 : Bộ ghi dịch đưa vào nối tiếp dùng Trigơ JK nối kiểu trigơ D

Ngoài ra còn cách ghi song song (đồng thời) vào các ô nhớ như hình 3.

-  $G_0 \div G_7$  là các cổng 3 trạng thái với đặc điểm: Tín hiệu điều khiển  $G = "1"$  đầu ra được nối với đầu vào,  $G = "0"$  đầu ra ở trạng thái trở kháng cao.

- Khi lệnh ghi nhận trị "1" thông tin nhị phân  $D_0 \div D_7$  được ghi vào các trigơ D ( $F_0 \div F_7$ ), kết thúc lệnh ghi (nhận trị "0")

Nhịp	$Q_0$	$Q_1$	$Q_2$	$Q_3$
0	0	0	0	0
1	$D_1$	0	0	0
2	$D_2$	$D_1$	0	0
3	$D_3$	$D_2$	$D_1$	0
4	$D_4$	$D_3$	$D_2$	$D_1$
5	$D_5$	$D_4$	$D_3$	$D_2$
6	$D_6$	$D_5$	$D_4$	$D_3$
7	$D_7$	$D_6$	$D_5$	$D_4$

Hình 2: Trạng thái ra của bộ ghi dịch 4 bit hình 1 theo trật tự xung nhịp

thông tin nhị phân được lưu trữ trong đó.

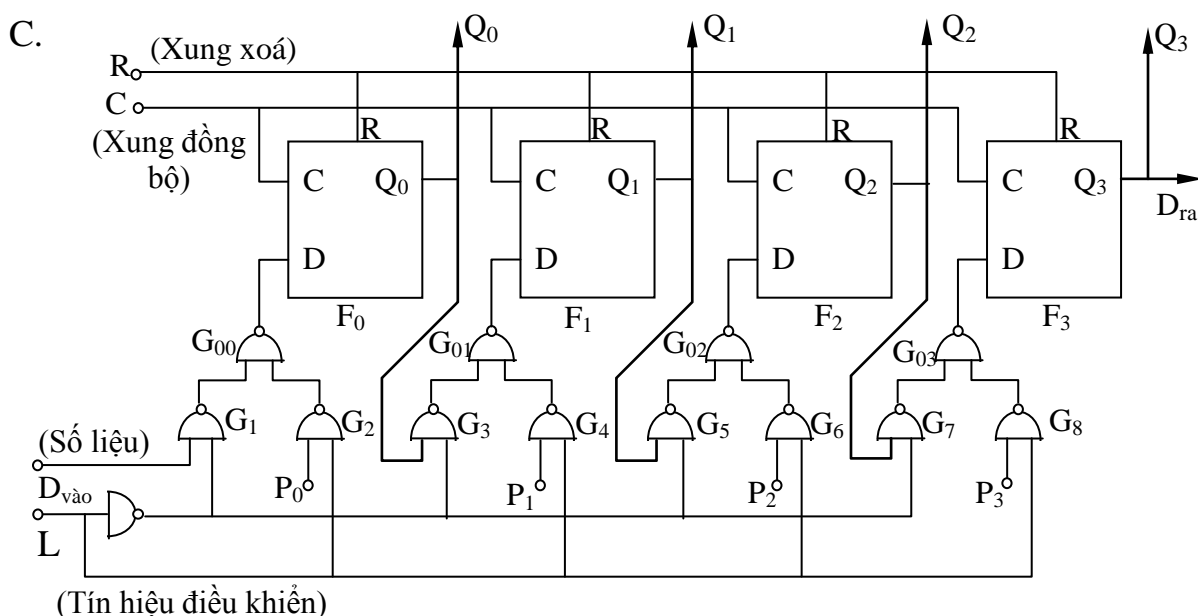
Khi có lệnh đọc (G nhận trị "1") các cổng 3 trạng thái được mở, thông tin nhị phân được gửi tới địa chỉ cần nhận

Các thao tác ghi - đọc được thực hiện đồng thời với cả 8 bit thông tin.

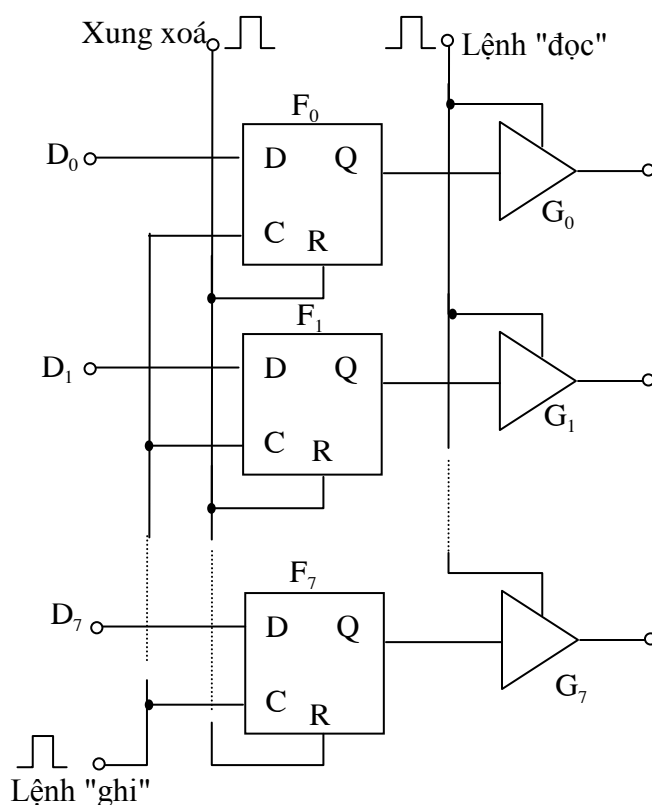
Ngoài ra người ta còn kết hợp phương pháp nối tiếp và song song trong một bộ ghi dịch để sử dụng linh hoạt các ưu thế của mỗi cách đồng thời tạo khả năng chuyển từ một dãy thông tin nối tiếp thành

dạng song song hoặc ngược lại. Hình 4 đưa ra cấu trúc một bộ ghi dịch 4 bit kiểu này, sử dụng 4 trigơ D kết hợp với các cổng logic phụ.

Số liệu đưa vào bộ ghi dịch hình 4 có thể tuần tự (kiểu nối tiếp) ở đầu vào D hay kiểu đồng thời ở các đầu  $P_0 \div P_3$  tùy theo xung điều khiển L và xung nhịp C.



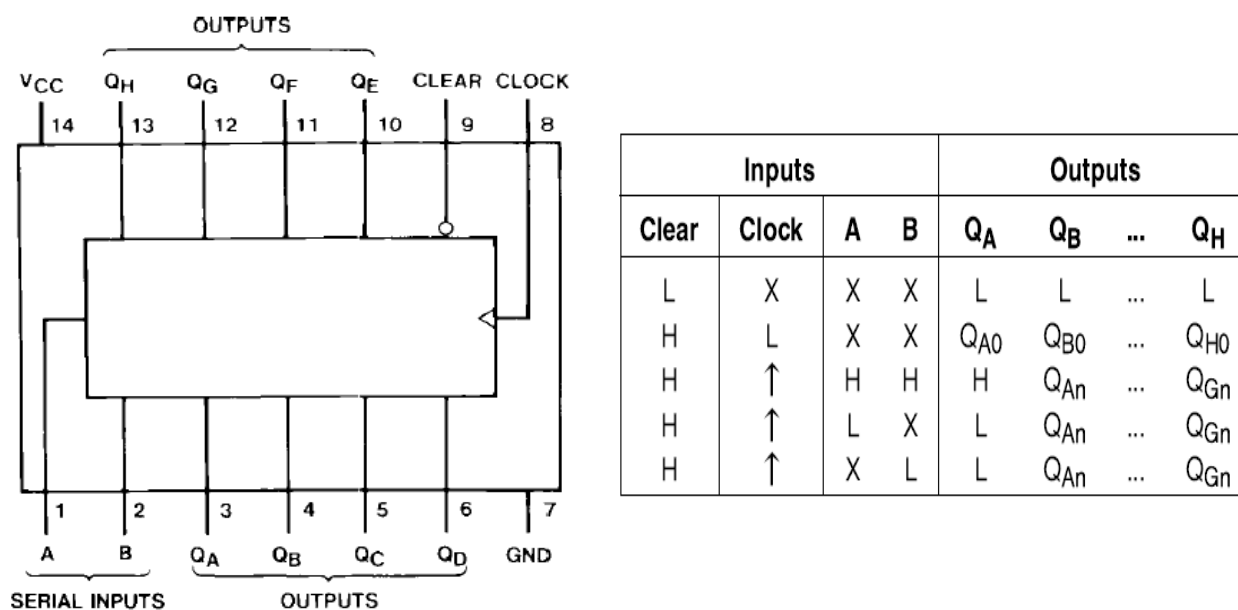
Hình 4: Bộ ghi dịch 4 bit hỗn hợp.



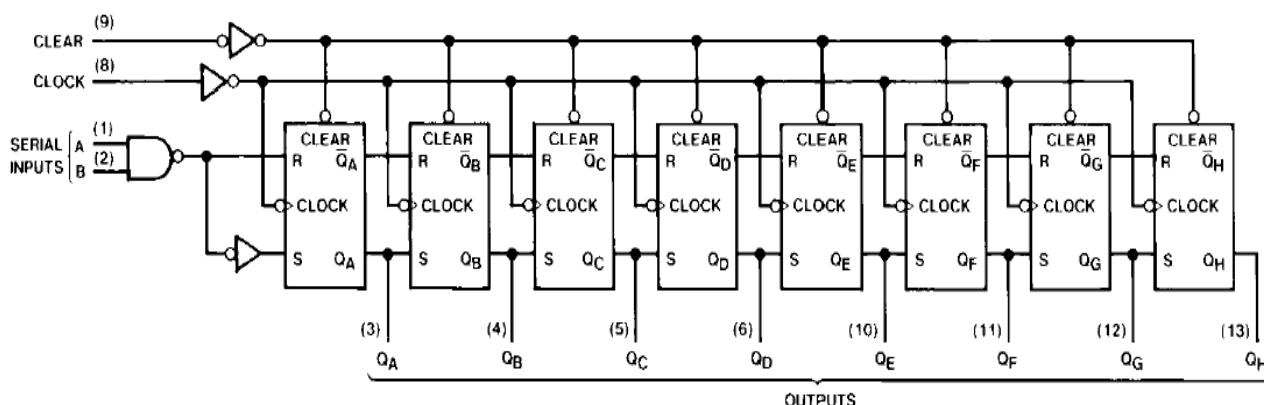
Hình 3: Bộ ghi cấu trúc vào ra song song (8 bit)

Khi  $L = 0$  thì với việc có xung nhịp  $C$ , thông tin  $D$  sẽ được dịch phải 1 bit hướng  $F_0 \rightarrow F_3$ . Lúc  $L = 1$  thì khi có xung nhịp  $C$ , thông tin  $P_0 \div P_3$  sẽ được đưa đồng thời vào  $F_0 \div F_3$ . Việc lấy số liệu ra cũng có thể đồng thời cả 4 bit trên các lối ra  $Q_0 \div Q_3$  hay tuần tự trên lối ra  $D_{ra}$  kiểu vào trước ra trước sau 4 nhịp của xung  $C$ .

Kết cấu hình 4 cho phép sử dụng linh hoạt và khai thác hết các ưu điểm của mỗi phương pháp ghi dịch kiểu tuần tự hay đồng thời.



Sơ đồ chân và bảng trạng thái của IC TTL 74LS164  
- bộ ghi dịch nối tiếp/song song 8bit



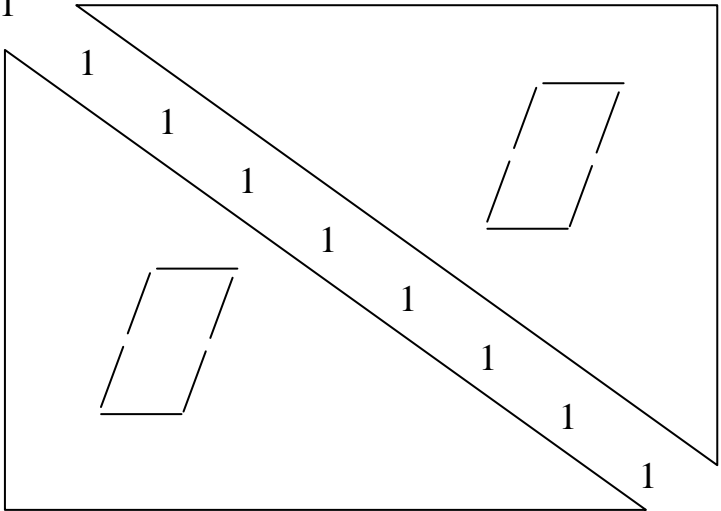
Sơ đồ logic của IC TTL 74LS164 - bộ ghi dịch nối tiếp/song song 8bit

## 5.2. Bộ biến đổi mã và giải mã

Các bộ biến đổi mã thực hiện việc chuyển đổi cách biểu diễn của một số nhị phân ở dạng này sang dạng khác nhờ đó quá trình gia công xử lý, ghi nhớ hay hiển thị kết quả thông tin được thuận lợi hơn. Trong phần này ta sẽ đề cập tới một số dạng biến đổi và giải mã điển hình nhất

### 5.2.1 Bộ biến đổi mã nhị phân sang mã "1 từ n".

Trạng thái của bộ biến đổi mã nhị - thập phân (BCD) sang mã "1 từ 10" được cho trên bảng hình 1

M	$2^3$ $X_3$	$2^2$ $X_2$	$2^1$ $X_1$	$2^0$ $X_0$	$y_0$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$	$y_6$	$y_7$	$y_8$	$y_9$
0	0	0	0	0	1									
1	0	0	0	1										
2	0	0	1	0										
3	0	0	1	1										
4	0	1	0	0										
5	0	1	0	1										
6	0	1	1	0										
7	0	1	1	1										
8	1	0	0	0										
9	1	0	0	1										

Hình 1: Bảng trạng thái bộ chuyển đổi mã nhị - thập phân sang mã "1 từ 10"

Để tìm quan hệ của các hàm ra với các biến vào đồng thời đưa chúng về dạng tối giản ta dùng phương pháp bìa các nô có tận dụng các trạng thái dư.

$y_0$	$x_3x_2$	$x_1x_0$	00	01	11	10
00	00		1	0	0	0
01	01		0	0	0	0
11	11		x	x	x	x
10	10		0	0	x	x

$$y_0 = \bar{x}_3\bar{x}_2\bar{x}_1\bar{x}_0$$

$y_9$	$x_3x_2$	$x_1x_0$	00	01	11	10
00	00		0	0	0	0
01	01		0	0	0	0
11	11		x	x	x	x
10	10		0	1	x	x

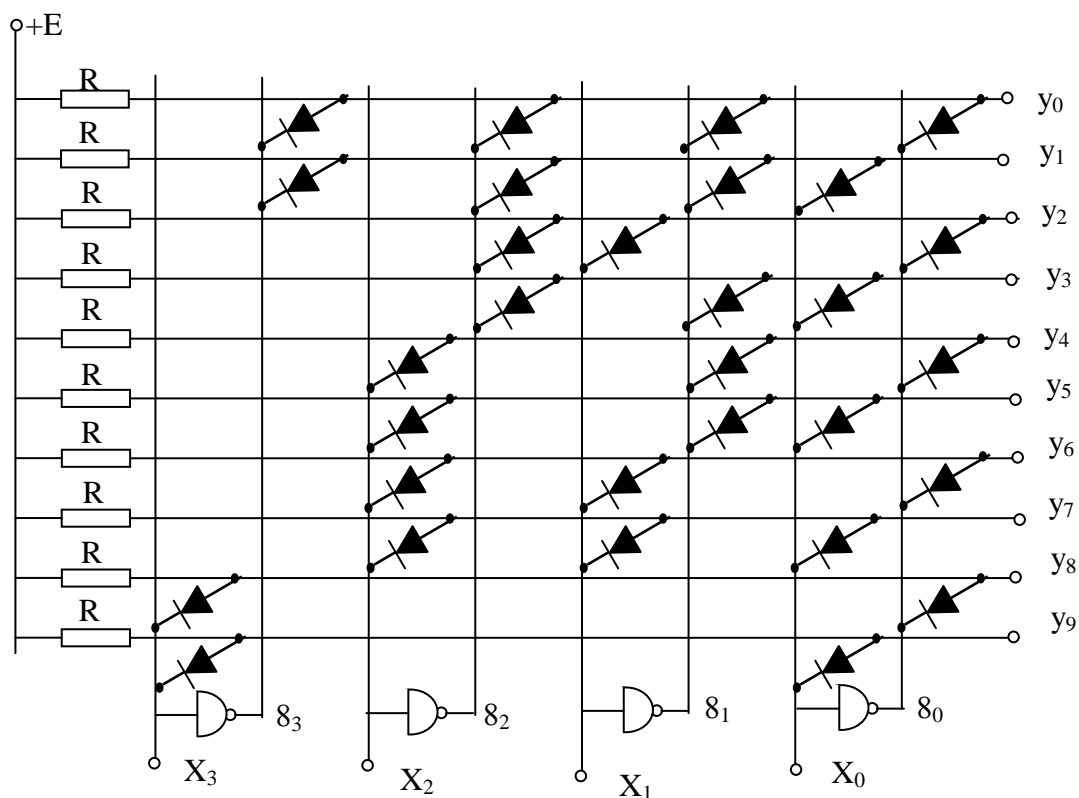
$$y_9 = x_3x_0$$

Kết quả như sau:

$$y_0 = \bar{x}_3 \bar{x}_2 \bar{x}_1 \bar{x}_0 \quad ; \quad y_1 = \bar{x}_3 \bar{x}_2 \bar{x}_1 x_0 \quad ; \quad y_2 = \bar{x}_3 x_1 \bar{x}_0 \quad ; \quad y_3 = \bar{x}_2 x_1 x_0$$

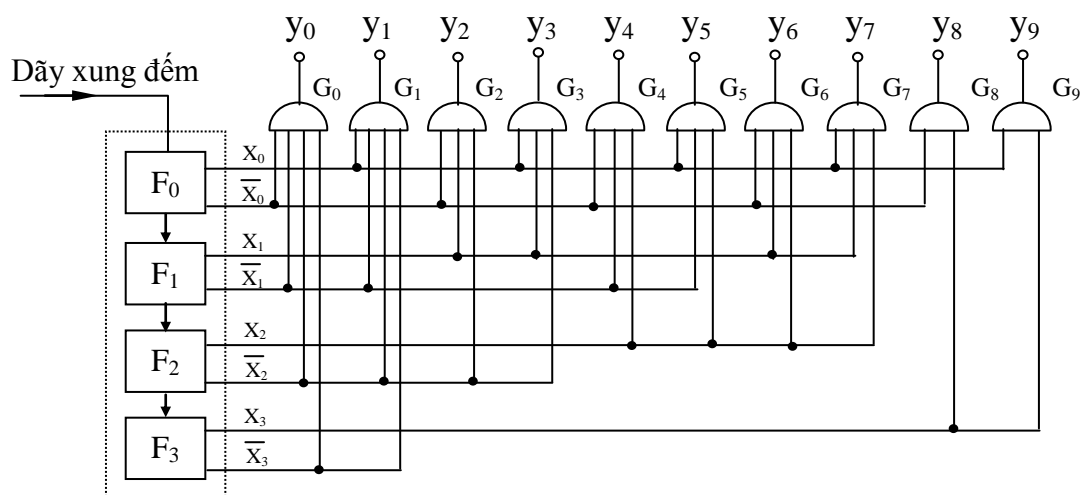
$$y_4 = x_2 \bar{x}_1 \bar{x}_0 \quad ; \quad y_5 = x_2 \bar{x}_1 x_0 \quad ; \quad y_6 = x_2 x_1 \bar{x}_0 \quad ; \quad y_7 = x_2 x_1 x_0$$

$$y_8 = x_3 \bar{x}_0 \quad ; \quad y_9 = x_3 x_0$$



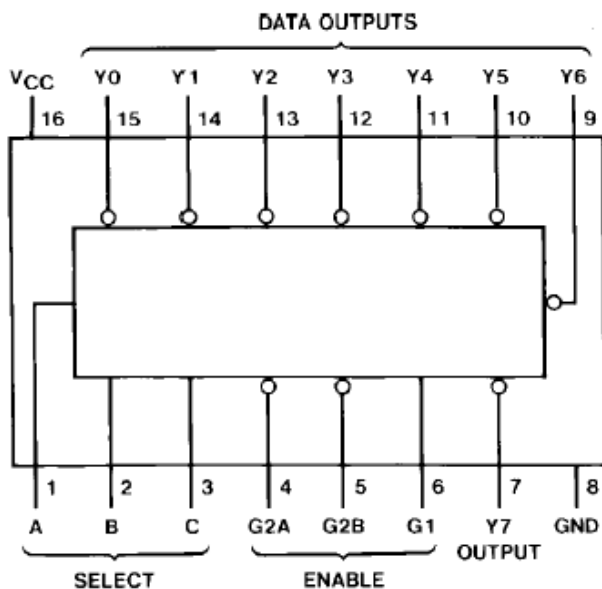
Hình 2: Bộ giải mã "1 từ 10" cấu trúc kiểu ma trận điôt điện trở

Hình 2 đưa ra một cấu trúc đơn giản nhất của bộ giải mã "1 từ 10" kiểu ma trận điôt - điện trở. Các hàng ngang tương ứng với các hàm ra, các cột tương ứng với các biến vào. Các điôt có các Anốt nối với hàng ngang tương ứng với hàm, Katốt nối với các cột ứng với các biến. Với một giá trị tổ hợp biến đầu vào chỉ duy nhất có một hàng các điôt đều khoá, hàm tương ứng nhận trị "1", còn các hàng khác có ít nhất một điôt thông do vậy các hàm còn lại đều nhận trị "0".

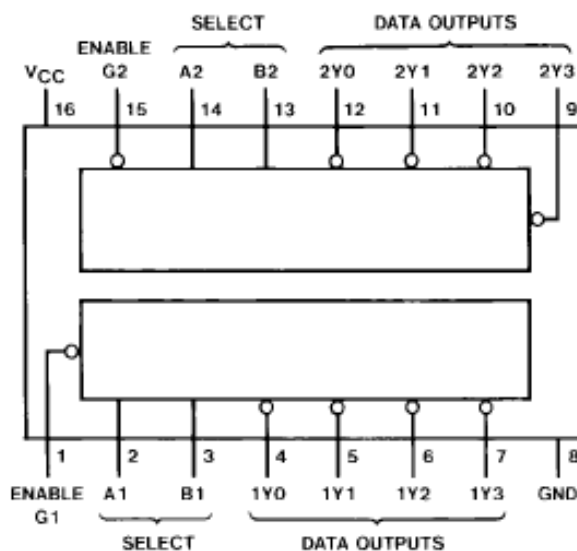


Bộ đếm nhị phân BCD

Hình 3: Bộ giải mã BCD - "1 từ 10".



Sơ đồ chân IC 74LS138  
– Bộ giải mã 1 từ 8



Sơ đồ chân IC 74LS139  
– Bộ giải mã 1 từ 4

Inputs		Outputs										
Enable	Select											
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
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	L	L	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H

\* G2 = G2A + G2B

H = High Level, L = Low Level, X = Don't Care

Inputs			Outputs			
Enable	Select					
G	B	A	Y0	Y1	Y2	Y3
H	X	X	H	H	H	H
L	L	L	L	H	H	H
L	L	H	H	L	H	H
L	H	L	H	H	L	H
L	H	H	H	H	H	L

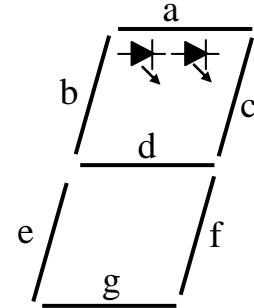
H = High Level, L = Low Level, X = Don't Care

Bảng trạng thái của IC 74LS138 và 74LS139

### 5.2.2. Bộ giải mã nhị phân BCD - mã thập phân 7 dấu.

Bộ chỉ thị 7 dấu kí hiệu được dùng phổ biến để biểu thị kết quả thông tin bằng số thập phân nhờ đặc điểm có cấu tạo điốt phát quang (LED) hay tinh thể lỏng bố trí như thể hiện trên hình 4.

M	Biến vào				Hàm ra						
	$x_3$	$x_2$	$x_1$	$x_0$	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	0	1	1	1
1	0	0	0	1	0	0	1	0	0	1	0
2	0	0	1	0	1	0	1	1	1	0	1
3	0	0	1	1	1	0	1	1	0	1	1
4	0	1	0	0	0	1	1	1	0	1	0
5	0	1	0	1	1	1	0	1	0	1	1
6	0	1	1	0	1	1	0	1	1	1	1
7	0	1	1	1	1	0	1	0	0	1	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1



Hình 4: Dạng bộ chỉ thị 7 thanh

Bảng biến đổi của bộ giải mã BCD loại Ktot - mã 7 dấu cho trên bảng hình 5. ở đây các biến logic đầu vào ký hiệu là  $x_0 \rightarrow x_3$  và các hàm ra là a b c d e f g.

Hình 5: Bảng trạng thái bộ giải mã

Bình thường các điốt phát quang a, b, c, d, e, f, g không phát sáng.

Ứng với mỗi tổ hợp nhị phân ở đầu vào, một vài trong số 7 thanh đầu ra nhận được tín hiệu "1" (thể cao) kích thích chúng phát sáng và hiện hình số thập phân tương ứng. Cấu trúc tổ hợp của bộ giải mã này được xây dựng xuất phát từ hệ các hàm a ÷ g sau khi đã được tối thiểu hoá.

a

$x_3x_2$	$x_1x_0$			
	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	x	x	x	x
10	1	1	x	x

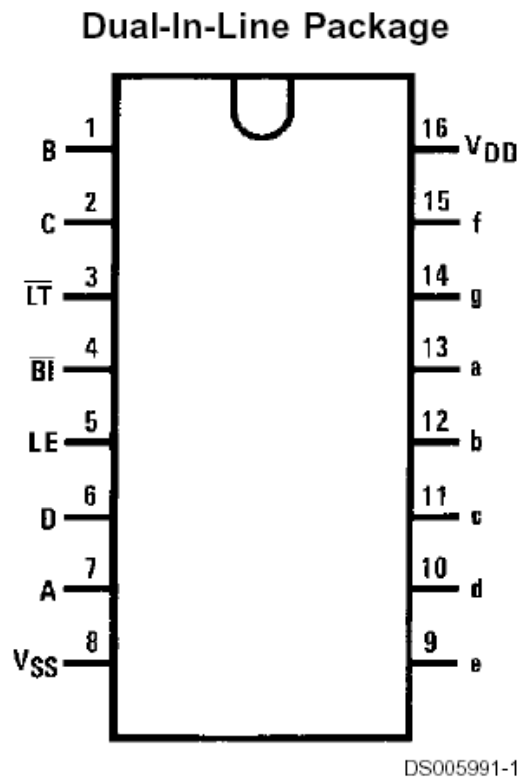
$$a = (x_3 + x_2 + x_1 + \overline{x_0})(\overline{x_2} + x_1 + x_0)$$

b

$x_3x_2$	$x_1x_0$			
	00	01	11	10
00	1	0	0	0
01	1	1	0	1
11	x	x	x	x
10	1	1	x	x

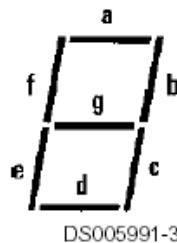
$$b = x_3 + \overline{x_1}\overline{x_0} + x_2\overline{x_1} + x_2\overline{x_0}$$

Thường chúng được kết cấu ở một dạng vi mạch có sẵn.



**Top View**  
**Order Number CD4511B**

**Segment Identification**

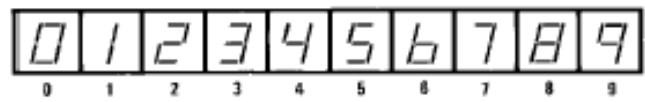


Inputs							Outputs							
E	$\overline{B}$	LT	D	C	B	A	a	b	c	d	e	f	g	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	B
X	0	1	X	X	X	X	0	0	0	0	0	0	0	
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	
0	1	1	1	0	1	1	0	0	0	0	0	0	0	
0	1	1	1	1	0	0	0	0	0	0	0	0	0	
0	1	1	1	1	1	0	0	0	0	0	0	0	0	
0	1	1	1	1	1	1	0	0	0	0	0	0	0	
1	1	1	X	X	X	X				*				*

X = Don't Care

\*Depends upon the BCD code applied during the 0 to 1 transition of LE.

**Display**



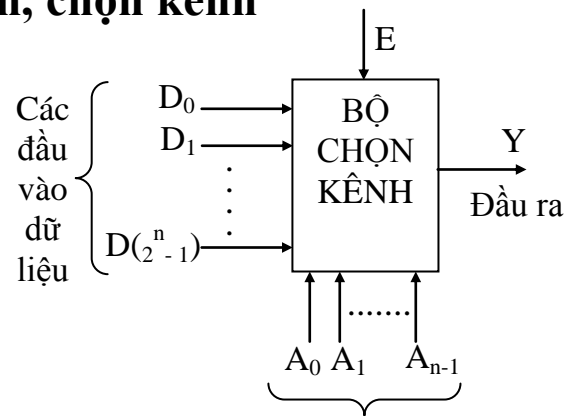
Sơ đồ chân và bảng trạng thái của IC 4511 – Bộ giải mã 7 thanh



## 5.3. Bộ phân kênh, chọn kênh

### 5.3.1. Bộ chọn kênh (Bộ dồn kênh)

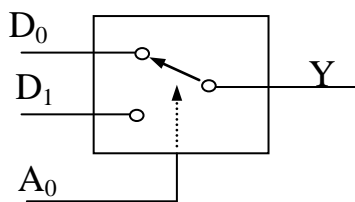
Bộ chọn kênh (Multiplexer) là mạch logic có nhiều đầu vào dữ liệu, một đầu ra, cho phép chọn một trong các đầu vào được nối với đầu ra.



Trong đó: +  $D_0 \div D_{(2^n-1)}$ : Các đầu vào dữ liệu (Data)  
 +  $A_0 \div A_{n-1}$ : Các đầu vào địa chỉ (Address)  
 + Y : Đầu ra; E (Enable): Tín hiệu cho phép

#### 5.3.1.1. Bộ chọn kênh 2 đầu vào:

Có hai đầu vào biến trạng thái, một đầu vào biến địa chỉ, một đầu ra thỏa mãn bảng trạng thái hình 2.



Hình 1: Mạch điện mô phỏng trạng thái

$A_0$	Y
0	$D_0$
1	$D_1$

Hình 2: Bảng trạng thái

$D_0, D_1$ : 2 đầu vào biến trạng thái.

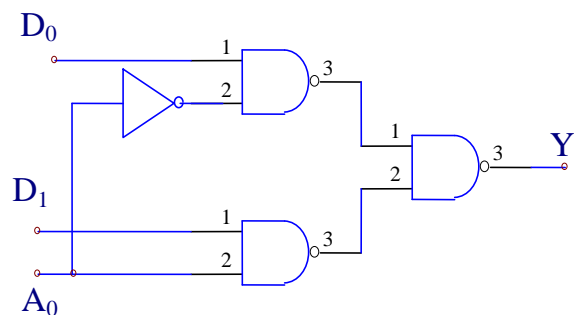
$A_0$ : Biến địa chỉ.

Y: Hàm ra.

Từ bảng trạng thái (hình 2) ta có:

$$Y = \overline{A_0} \cdot D_0 + A_0 \cdot D_1 = \overline{\overline{\overline{\overline{A_0} \cdot D_0}} \cdot \overline{\overline{\overline{A_0} \cdot D_1}}} \quad (*)$$

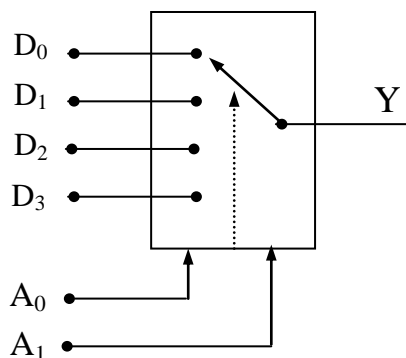
Từ (\*)  $\rightarrow$  Mạch điện hình 3:



Hình 3: Bộ chọn kênh 2 đầu vào

#### 5.3.1.2. Bộ chọn kênh 4 đầu vào:

Gồm 4 đầu vào biến trạng thái  $D_0, D_1, D_2, D_3$ , hai đầu vào biến địa chỉ  $A_1, A_0$  một đầu ra Y. Đầu ra Y sẽ được nối với một trong 4 đầu vào thông tin được chọn.



Hình 4: Mạch điện mô phỏng

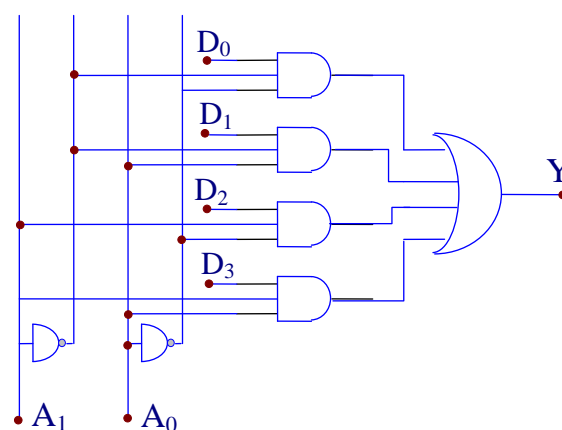
$\hat{A}_1$	$A_0$	Y
0	0	$D_0$
0	1	$D_1$
1	0	$D_2$
1	1	$D_3$

Hình 5: Bảng trạng thái

Từ bảng trạng thái ta xác định được quan hệ của hàm ra với các biến vào:

$Y = \overline{A_1} \cdot \overline{A_0} \cdot D_0 + \overline{A_1} \cdot A_0 \cdot D_1 + A_1 \cdot \overline{A_0} \cdot D_2 + A_1 \cdot A_0 \cdot D_3$  (\*\*). Từ (\*\*) ta có mạch điện hình 6:

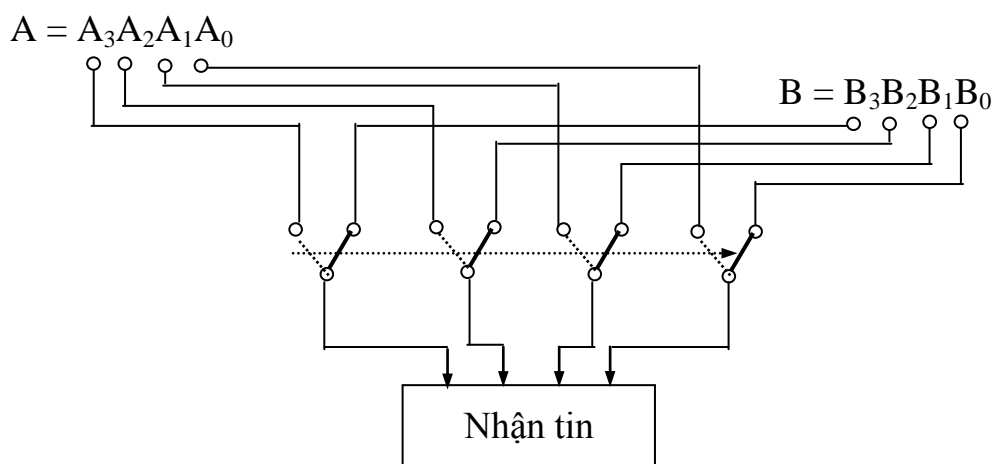
Tổng quát nếu có n biến địa chỉ từ  $A_{n-1} \div A_0$ , cho phép đầu ra được nối với 1 trong  $2^n$  đầu vào dữ liệu được chọn. Trường hợp các biến địa chỉ được lấy từ đầu ra của bộ đếm nhị phân thì bộ đếm có modul chính bằng số đầu vào của bộ chọn kênh, thí dụ bộ chọn kênh 12 đầu vào cần có 4 biến địa chỉ  $A_3, A_2, A_1, A_0$  được lấy từ đầu ra của bộ đếm nhị phân 4 bit mô đun 12.



Hình 6: Cấu trúc bộ chọn kênh 4 đầu vào

### 5.3.1.3. Một số ứng dụng của bộ chọn kênh:

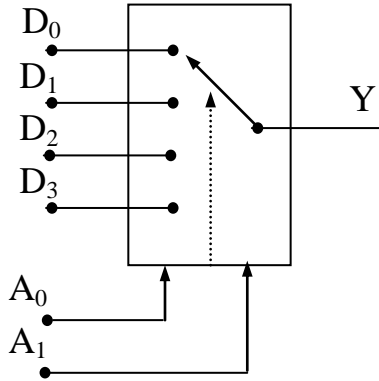
#### a. Chọn nguồn thông tin :



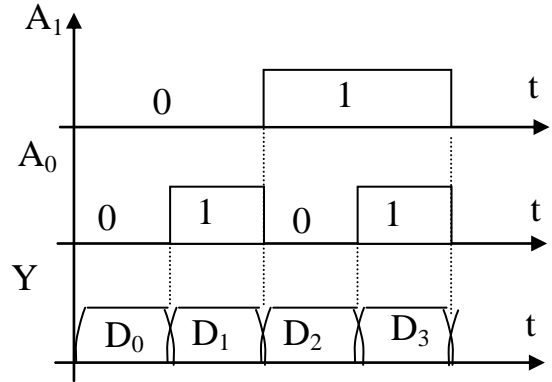
Hình 7: Truyền hai thông tin nhị phân 4 bit (hoặc chọn A hoặc chọn B).

### b. Chuyển đổi (tín hiệu) song song nối tiếp.

Thông tin nhị phân 4 bit  $D = D_3D_2D_1D_0$  đồng thời đưa tới các lối vào, dưới sự điều khiển của  $A_1, A_0$  (phù hợp như hình vẽ) tín hiệu ở đầu ra  $Y$  sẽ xuất hiện trình tự  $D_0 - D_1 - D_2 - D_3$ .



Hình 8: Mạch điện mô phỏng



Hình 9: Giải đồ minh họa quá trình làm việc chuyển đổi song song - nối tiếp

### c. Tạo hàm logic.

Theo định lí Shannon, hàm logic 2 biến dạng tuyến chính quy:

$$f_{(A,B)} = \overline{A}.\overline{B}.f_{(0,0)} + \overline{A}.B.f_{(0,1)} + A.\overline{B}.f_{(1,0)} + A.B.f_{(1,1)} \quad (1)$$

Với  $f_{(0,0)}, f_{(0,1)}, f_{(1,0)}, f_{(1,1)}$  được gọi là các hàm thành phần.

Theo kết quả bộ chọn kênh 4 đầu vào:

$$Y = \overline{A_1}.\overline{A_0}.D_0 + \overline{A_1}.A_0.D_1 + A_1.\overline{A_0}.D_2 + A_1.A_0.D_3 \quad (2)$$

Từ (1) và (2) ta thấy có sự tương ứng :

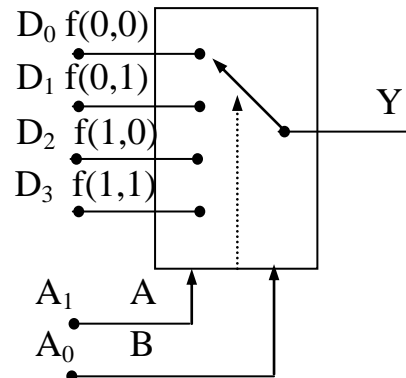
$$A_1 \leftrightarrow A; A_0 \leftrightarrow B; D_0 \leftrightarrow f(0,0); D_1 \leftrightarrow f(0,1);$$

$$D_2 \leftrightarrow f(1,0); D_3 \leftrightarrow f(1,1).$$

Do vậy bằng cách thay đổi trị logic của các đầu vào bộ chọn kênh sẽ biến đổi bộ chọn kênh thành các phần tử logic với chức năng theo mong muốn (mạch cộng, mạch nhân, phần tử khác dấu, cùng dấu.v.v...). Khi này người ta gọi đây là mạch tạo hàm logic có lập trình.

$A_1$	$A_0$	$Y$
0	0	$D_0$
0	1	$D_1$
1	0	$D_2$
1	1	$D_3$

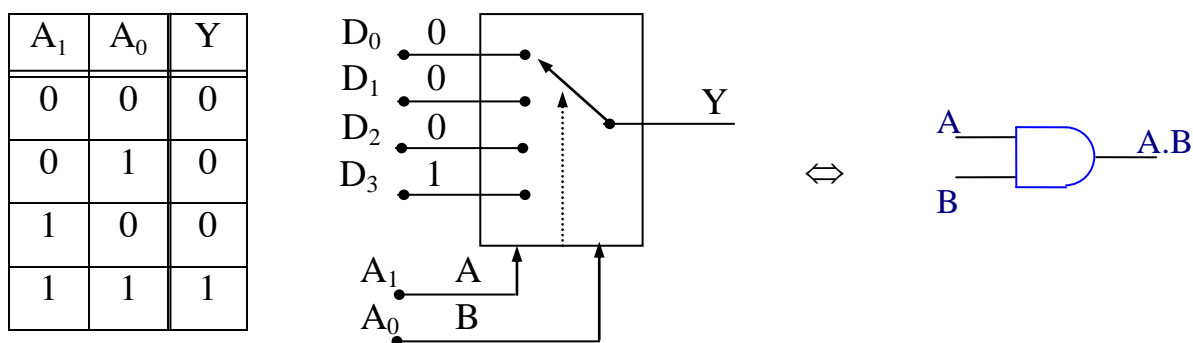
Hình 10: Bảng trạng thái



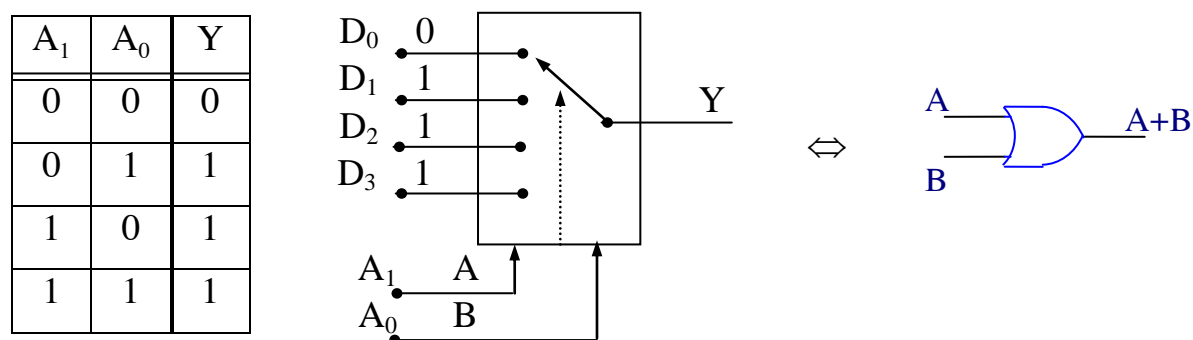
Hình 11: Mạch điện mô phỏng

**Giả sử tạo một số hàm logic hai biến cơ bản.**

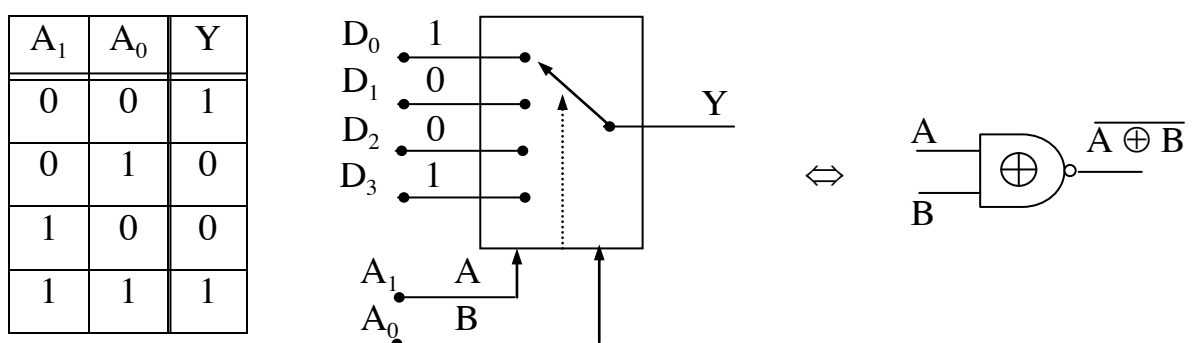
\* Tạo mạch nhân: Cho  $X_0 = 0$ ;  $X_1 = 0$ ;  $X_2 = 0$ ;  $X_3 = 1$ .



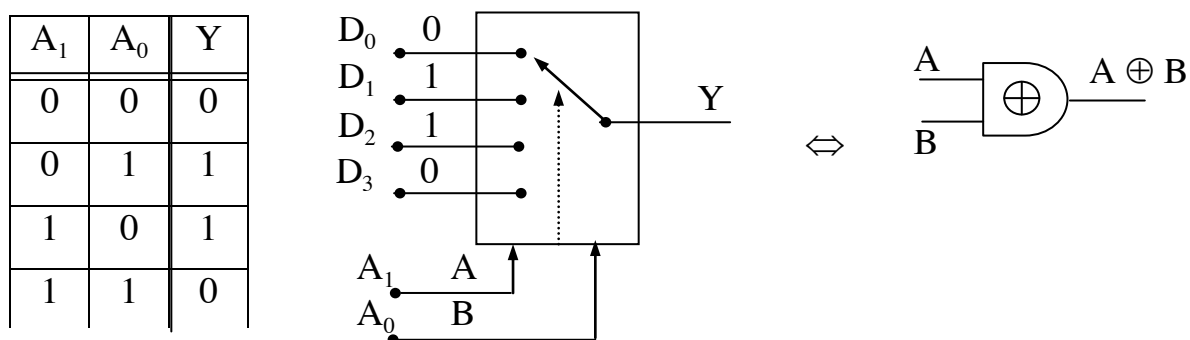
\* Tạo mạch cộng: Cho  $D_0 = 0$ ;  $D_1 = 1$ ;  $D_2 = 1$ ;  $D_3 = 1$ .



\* Tạo mạch tương đương cùng dấu : Cho  $D_0 = 1$ ;  $D_1 = 0$ ;  $D_2 = 0$ ;  $D_3 = 1$ .



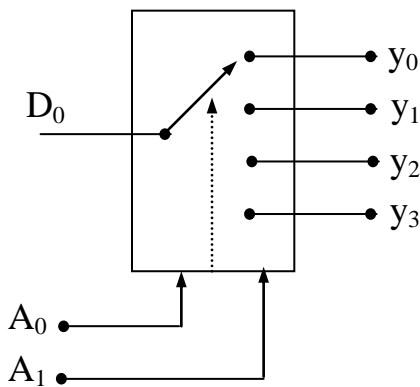
\* Tạo mạch khác dấu : Cho  $D_0 = 0$ ;  $D_1 = 1$ ;  $D_2 = 1$ ;  $D_3 = 0$ .



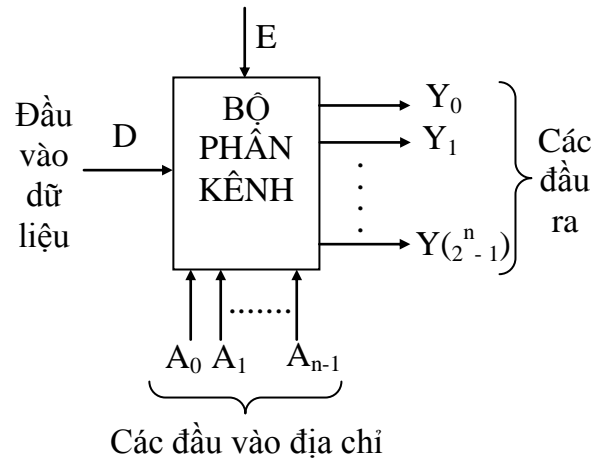
### 5.3.2. Bộ phân kênh (Bộ tách kênh)

Bộ phân kênh (Demultiplexer) là mạch logic có nhiều đầu ra, một đầu vào, cho phép nối tín hiệu vào đến một trong các đầu ra được chọn.

Ta xét bộ phân kênh 4 đầu ra (Hình 1) bao gồm 1 đầu vào dữ liệu  $D_0$ , 2 đầu vào biến địa chỉ  $A_1, A_0$ , 4 đầu ra  $y_0, y_1, y_2, y_3$  thỏa mãn bảng trạng thái (hình 2).



Hình 1: Mạch điện mô phỏng



$A_1$	$A_0$	$y_0$	$y_1$	$y_2$	$y_3$
0	0	$D_0$	0	0	0
0	1	0	$D_0$	0	0
1	0	0	0	$D_0$	0
1	1	0	0	0	$D_0$

Hình 2: Bảng trạng thái

Từ bảng trạng thái (Hình 2), ta có:  $y_0 = \overline{A_1} \cdot \overline{A_0} \cdot D_0$ ;  $y_1 = \overline{A_1} \cdot A_0 \cdot D_0$

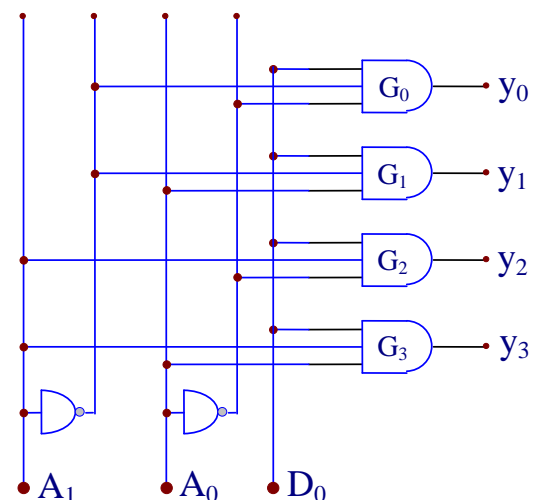
$$y_2 = A_1 \cdot \overline{A_0} \cdot D_0; \quad y_3 = A_1 \cdot A_0 \cdot D_0$$

Từ các hàm trên ta có mạch điện hình 3. Ứng với một giá trị tổ hợp của biến địa chỉ, đầu vào được nối với một đầu ra duy nhất được chọn.

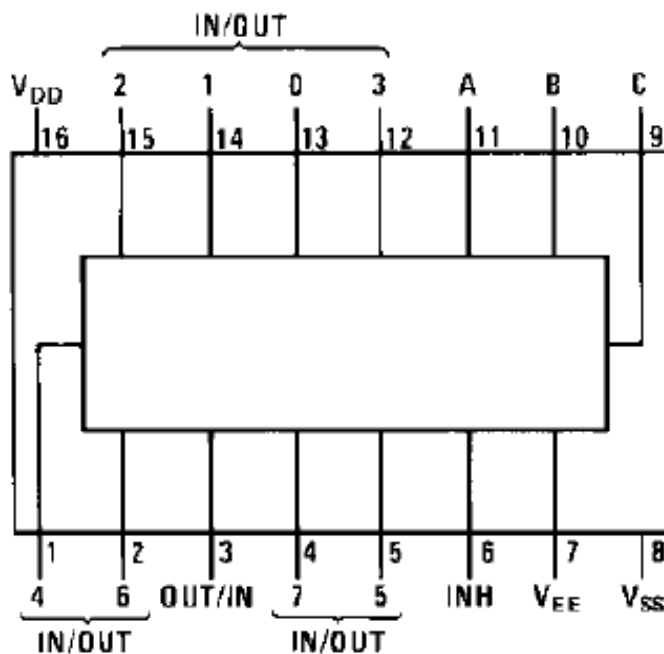
Trường hợp đầu vào  $D_0$  nhận trị "1" (nối với  $+U_{CC}$ ) bộ phân kênh làm việc như một bộ giải mã "1 từ n".

Tổng quát: nếu có n biến địa chỉ  $A_0, A_{n-1}$  thì có thể nối đầu vào với 1 trong  $2^n$  đầu ra.

Một trong các ứng dụng quan trọng của bộ phân kênh là được dùng để chuyển đổi (tín hiệu) nối tiếp - song song.



Hình 3: Bộ phân kênh 4 đầu ra



Sơ đồ chân IC 4051BM -  
Bộ chọn kênh/phân kênh tương tự 8 kênh

Input				"ON" Channel
Inh	C	B	A	
H	X	X	X	None
L	L	L	L	Y0
L	L	L	H	Y1
L	L	H	L	Y2
L	L	H	H	Y3
L	H	L	L	Y4
L	H	L	H	Y5
L	H	H	L	Y6
L	H	H	H	Y7

Bảng trạng thái của IC 4051BM-

## CHƯƠNG 6: BỘ NHỚ

### 6.1. Các khái niệm

Đối với các thiết bị số, khả năng chứa đựng được dữ liệu là một yêu cầu quan trọng. Chẳng hạn trong máy tính, các con số cần thiết trong phép toán phải được lưu trữ ngay trong máy. Còn các thiết bị điều khiển số thì lệnh điều khiển cũng phải được lưu trữ để thực hiện dần theo một trình tự nào đó. Vì vậy, bộ nhớ là một thành phần không thể thiếu được của các thiết bị số.

Khi nghiên cứu về bộ nhớ, thông tin hay dữ liệu lưu chuyển trong các thiết bị số đều phải dưới dạng mã hệ 2, tức là một chuỗi kết hợp bởi 0 và 1 và được biểu diễn bởi hai mức điện thế khác nhau.

Thông thường thông tin hay dữ liệu được tạo thành từ một đơn vị cơ bản gọi là từ (word). Một từ có chiều dài nhất định tùy theo loại máy, chẳng hạn 8 bit, 16 bit, 32 bit, .v.v... Từ là thành phần thông tin cơ bản nhất. Các bộ phận của thiết bị thường chỉ truyền đi hay nhận vào nguyên một từ (hay nguyên từ) chứ không phải vài bit của từ. Tuy nhiên, vì từ được tạo thành từ nhiều bit nên đơn vị cơ bản của bộ nhớ chính là đơn vị lưu trữ được một bit.

Khi so sánh các bộ nhớ người ta thường lưu ý đến các đặc tính sau:

+ Dung lượng.

Dung lượng (hay nói rõ hơn là dung lượng nhớ) là khối lượng thông tin hay dữ liệu có thể lưu trữ được trong bộ nhớ. Để xác định dung lượng ta thường dùng đơn vị là số bit (hoặc kilobit hoặc megabit). Dung lượng liên quan mật thiết đến giá thành của bộ nhớ. Giá thành này được đánh giá theo tiêu chuẩn: chi phí/ bit.

+ Thời gian thâm nhập (access time).

Thời gian này gồm có 2 phần: thứ nhất là thời gian cần thiết để xác định vị trí cần thiết của từ (thời gian tìm từ) trong bộ nhớ và thứ hai là thời gian cần thiết để lấy ra khỏi bộ nhớ. Thời gian thâm nhập (viết tắt là at) là một thông số quan trọng của bộ nhớ, nếu nó kéo dài thì sẽ làm giảm khả năng làm

việc của thiết bị vì thiết bị chỉ hoạt động được chừng nào chúng nhận được dữ liệu mà thôi.

Bộ nhớ thường được chia làm hai loại căn cứ vào hai tính chất vừa nêu trên là bộ nhớ chính và bộ nhớ phụ.

+ Bộ nhớ chính.

Bộ nhớ chính nằm gần các bộ xử lý dữ liệu và cần có at rất ngắn ( $\leq \mu\text{sec}$ ) với dung lượng không cần lớn lắm (vài chục kilobit là có thể được). Phần này chứa các dữ liệu, thông tin, hoặc các lệnh cần ngay cho công tác.

+ Bộ nhớ phụ.

Bộ nhớ phụ không cần thiết phải nằm gần thiết bị, at có thể lớn (tới ms) nhưng lại cần có dung lượng lớn (từ hàng chục kilobit trở lên, có thể đến hàng chục megabit) để lưu trữ các thông tin chưa cần ngay hoặc các kết quả vừa được xử lý xong. Chúng có tính chất như một thư viện.

Bộ nhớ chỉ có khả năng lưu trữ, khi sử dụng ta phải ghi dữ liệu vào (thường gọi là viết vào) hoặc lấy dữ liệu ra (thường gọi là đọc ra). Việc viết và đọc như vậy thường bao gồm cả việc xác định vị trí của mỗi từ trong bộ nhớ. Mỗi từ (trong bộ nhớ) phải có một vị trí riêng được xác định bởi một mã số gọi là địa chỉ của từ. Như vậy bộ nhớ cần phải có mạch để xác định địa chỉ của mỗi từ trước khi đọc hay viết.

Ta thường có thể xoá các dữ liệu cũ trong bộ nhớ để viết vào đó dữ liệu mới, loại bộ nhớ như vậy gọi là bộ nhớ đọc - viết (Read - Write Memory). Sở dĩ phải gọi như thế là vì cần phải phân biệt với một loại bộ nhớ khác có chứa sẵn các dữ liệu, khi sử dụng ta cần đọc ra mà không viết gì vào được. Với loại bộ nhớ này dữ liệu được ghi vào trong quá trình chế tạo, sau đó nội dung của bộ nhớ được lưu trữ vĩnh viễn trong bộ nhớ mà không thay đổi được. Loại này gọi là bộ nhớ chỉ đọc (Read Only Memory) ứng dụng của loại này thường là để chứa các lệnh điều khiển sự hoạt động của thiết bị mà các lệnh này không cần thay đổi trong quá trình làm việc khác nhau. Nằm giữa hai loại bộ nhớ nói trên còn có một loại bộ nhớ khác, chúng giống như bộ nhớ đọc viết ở



chỗ có thể viết vào bằng phương tiện đặc biệt và sau đó nội dung được lưu trữ vĩnh viễn cho đến khi ta muốn xóa đi (tất nhiên cũng bằng phương tiện đặc biệt), loại bộ nhớ này đôi khi được gọi là bộ nhớ bán cố định (Read Mostly Memory).

Sau đây chúng sẽ xét các loại bộ nhớ thông dụng nhất hiện nay.

## 6.2. Bộ nhớ Ram

Thuật ngữ RAM là viết tắt của từ tiếng anh Ramdom Access Memory, thường dùng để chỉ các bộ nhớ đọc-viết.

Bộ nhớ Ram thường được sử dụng trong các thiết bị tính để cất giữ các kết quả trung gian hay kết quả tạm thời khi thực hiện các chương trình điều khiển.

Như ta biết một mạch FF có hai trạng thái bền, rất tiện dụng để làm một đơn vị nhớ của bộ nhớ đọc viết. Tuy nhiên, trong quá khứ ý tưởng này không được thực hiện vì mạch FF làm bằng bộ phận rời có kích thước tương đối lớn, tiêu thụ công suất đáng kể và tốn kém. Nhưng từ khi kỹ thuật mạch tích hợp ra đời người ta chế tạo được các mạch tích hợp với nhiều bộ phận bán dẫn trên một diện tích Silic ngày càng bé. Từ đó bộ nhớ bán dẫn dùng FF ra đời.

Hiện nay có hai loại bộ nhớ bán dẫn RAM chính là dùng Tranzitor lưỡng cực (npn) và loại MOSFET.

+ Bộ nhớ Ram dùng Tranzitor lưỡng cực lấy FF làm đơn vị nhớ cơ bản nên vận tốc rất cao, at vào khoảng vài chục nanosec và công suất tiêu thụ vào khoảng 1 nanowatt/bit.

+ Bộ nhớ RAM dùng MOSFET được chia làm hai loại:

- Loại tĩnh (static) cũng lấy cấu trúc FF làm đơn vị nhớ cơ bản.
- Loại động (dynamic) lợi dụng điện dung kí sinh của cực cổng (gate) để chứa dữ liệu.

Sở dĩ người ta gọi tĩnh và động là vì loại dùng FF không cần xung kích thích để tăng cường thêm điện tích trong tụ điện (thao tác làm tươi bộ nhớ), nếu không điện tích này sẽ giảm đi và mất dữ liệu.

Nói chung bộ nhớ MOSFET chậm hơn bộ nhớ lưỡng cực nhưng bù lại đơn vị nhớ có kích thước nhỏ hơn nên có thể thu gọn nhiều đơn vị nhớ trong một diện tích silic nhỏ mà công suất tiêu thụ lại không đáng kể (chẳng hạn 250 mW cho 4096 bit đối với các bộ nhớ thế hệ đầu tiên).

Khi cần các bộ nhớ RAM có dung lượng lớn ta có thể mắc nhiều bộ nhớ nhỏ lại với nhau mà kích thước toàn thể không lớn lắm.

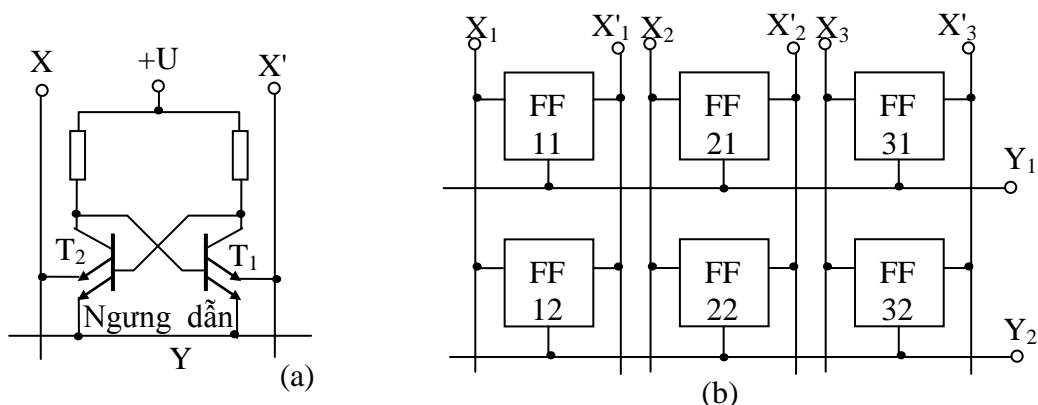
Sau đây ta sẽ khảo sát chi tiết hơn về bộ nhớ bán dẫn RAM.

### 6.2.1. Bộ nhớ RAM dùng Tranzitor lưỡng cực

#### a. Đơn vị nhớ.

Đơn vị nhớ (memory cell) là một FF nhưng được biến đổi đi đôi chút như sơ đồ trên hình 6-1.a : Tranzitor có nhiều cực phát (như TTL), mạch có hai đường  $XX'$  và một đường  $Y$ .

Mỗi đường  $Y$  dùng cho một từ, cặp  $XX'$  dùng cho từng bit của từ. Như vậy muốn chọn từ nào và bit nào ta sẽ kích thích các đường  $Y$ ,  $XX'$  tương ứng. Trên hình 6-1b là một mạch nhớ có 6 đơn vị nhớ được tạo thành từ 6 FF. Đường  $Y_1$  dùng cho từ thứ nhất (có 3 đơn vị nhớ FF 11, FF 21 và FF 31), còn đường  $Y_2$  dùng từ thứ hai (có 3 đơn vị nhớ FF 12, FF 22, và FF 32).



Hình 6-1: Đơn vị nhớ RAM dùng Tranzitor lưỡng

Bình thường thì đường Y ở điện thế thấp (chẳng hạn 0V) còn hai đường XX' ở điện thế cao (+U). Do đó hai cực phát nối vào dây XX' không dẫn điện, FF hoạt động nhờ hai cực phát nối vào dây Y và duy trì trạng thái bền nào đó, chẳng hạn  $T_1$  dẫn  $T_2$  ngưng.

+ Thao tác đọc.

Muốn đọc một từ ta nâng điện thế đường Y tương ứng lên cao hơn  $+U_x$  làm cho hai cực phát nối vào dây này ngưng dẫn và FF phải hoạt động nhờ hai cực phát nối vào XX'. Lúc này dòng điện sẽ chảy qua Tranzitor  $T_1$  (dẫn) để vào X' và ra ngoài. Sự hiện diện của dòng điện báo hiệu sự xuất hiện của tín hiệu:

- Nếu có dòng điện chảy trong đường X' ( $T_1$  dẫn) thì ta bảo FF chứa bit 0.

- Nếu có dòng điện chảy trong đường X ( $T_2$  dẫn) thì ta bảo FF chứa bit 1

Mạch ngoài sẽ có nhiệm vụ biến dòng điện thành điện thế thích hợp cho các cổng logic (thường là TTL). Sau khi đọc xong đường Y lại trở về điện thế thấp, như vậy ta chỉ đưa vào Y một xung điện thế để đọc và nhận một xung dòng điện chạy ra. Phép đọc như vậy không làm mất dữ liệu nên không cần phải viết lại.

+ Thao tác viết.

Muốn viết một từ ta phải qua hai giai đoạn: Trước hết ta phải nâng điện thế đường Y tương ứng để lựa chọn từ. Sau đó điều khiển các đường XX' để viết các bit như sau:

- Muốn viết bit 0 ta hạ điện thế đường X' thấp hơn đường X trong một thời gian ngắn làm  $T_1$  dẫn.

- Muốn viết bit 1 ta làm ngược lại, nghĩa là hạ điện thế đường X thấp hơn đường X' làm cho  $T_2$  dẫn.

Điều nói trên có nghĩa là ta sẽ cho vào X hay X' một xung hướng âm.

## **b. Đặc tính.**

Bộ nhớ RAM dùng Tranzitor lưỡng cực có các đặc tính sau:

- Vận tốc cao, thời gian thâm nhập bé (khoảng chừng vài chục nanosec) nên thuận tiện cho việc sử dụng trực tiếp với các bộ phận xử lý dữ liệu.
- Vì luôn có Tranzitor ở trạng thái dẫn nên luôn tiêu thụ công suất. So với loại MOSFET thì công suất tiêu thụ luôn luôn lớn hơn.
- Những mạch có dung lượng lớn thường có thêm mạch giải mã để tìm địa chỉ của từ và các mạch đầu ra để làm tăng hệ số mắc tải (fan out), tất cả được đặt lên cùng một phiên bán dẫn.

## 6.2.2. Bộ nhớ RAM dùng MOSFET

### a. Loại tĩnh (SRAM)

+ Đơn vị nhớ.

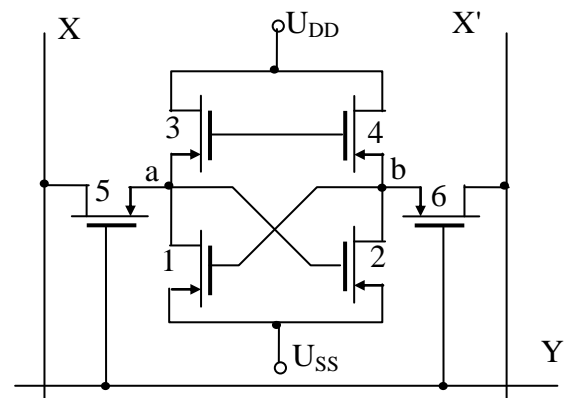
Đơn vị nhớ của loại RAM tĩnh dùng MOSFET cũng là FF nhưng hơi khác dạng đã đề cập ở trên (hình 6-2).

Ta vẫn có hai đường XX' và đường Y để xác định từ và các bit của từ. FF được tạo thành bởi các Tranzitor  $T_1$  và  $T_2$  còn  $T_3$  và  $T_4$  hoạt động như hai điện trở.  $T_5$  và  $T_6$  làm việc như hai khoá điện tử để đọc ra hay viết vào FF đã được chọn.

Nếu tất cả các Tranzitor đều là loại kênh P thì  $U_{DD}$  thường âm còn  $U_{SS}$  có thể bằng 0V. Các đường XX' thường có điện thế âm.

- Thao tác đọc.

Muốn đọc ta cho xung vào để làm giảm điện thế của đường Y xuống âm,  $T_5$  và  $T_6$  dẫn. Giả sử  $T_2$  dẫn thì cực máng của nó (nút b nối vào  $T_6$ ) ở điện thế thấp, vì vậy  $T_6$  sẽ dẫn một dòng điện chảy vào X' tạo ra dữ liệu. Trong khi đó vì  $T_1$  ngưng nên cực máng của nó ở điện thế âm cao bằng điện thế đường  $T_5$  không dẫn.



Hình 6-2: đơn vị nhớ RAM tĩnh dùng MOSFET

- Thao tác viết.

Muốn viết, ta cũng giảm điện thế của đường Y xuống để mở  $T_5$  và  $T_6$  (chọn từ) rồi sau đó nối đường X hoặc X' xuống 0V tùy theo ta muốn  $T_1$  hay  $T_2$  ngưng dẫn. Nếu  $T_2$  đang dẫn, nút b ở điện thế thấp gần 0V, nên khi nối đường X' xuống 0V thì sẽ không có gì thay đổi cả. Nhưng nút a đang ở điện thế âm nhiều nên nếu nối X xuống 0V thì sẽ làm hạ điện thế tại đây xuống dưới trị số ngưỡng nên  $T_2$  ngưng dẫn làm  $T_1$  trở nên dẫn.

+ Đặc tính.

- Người ta gọi loại bộ nhớ này là tĩnh bởi vì dữ liệu được lưu giữ vĩnh viễn

trong mạch chừng nào mà ta còn cung cấp nguồn điện cho nó. Các đường điều khiển không cần thêm một xung nào.

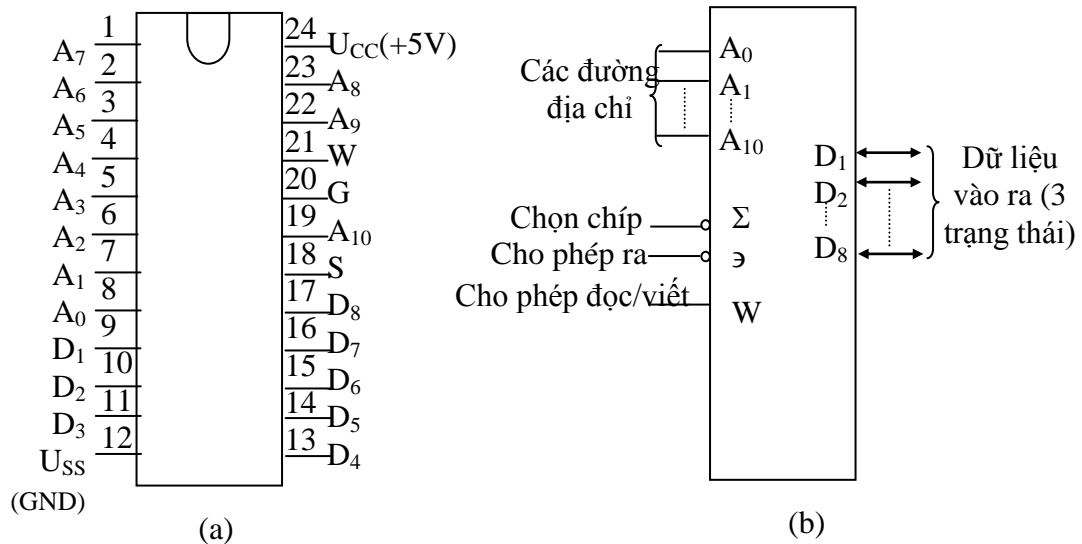
- Loại bộ nhớ này tương đối rẻ hơn loại Tranzitor lưỡng cực vì hai lý do: một là phương pháp chế tạo đơn giản và hai là kích thước của một đơn vị nhớ nhỏ nên cho phép chế tạo bộ nhớ có dung lượng lớn trên một diện tích miếng silic nhỏ.

- Thời gian thâm nhập lớn hơn loại lưỡng cực vì có điện dung ký sinh ở cực cổng của MOSFET.

- Có thể chế tạo theo phương pháp kênh P, kênh N hay CMOS.

+ Ví dụ 1:

Trên hình 6-4 giới thiệu sơ đồ chức năng (b), sơ đồ phân bố chân (a) và bảng chân lý (c) của mạch nhớ RAM tĩnh có ký hiệu là TMS 4016. đây là bộ nhớ có cấu trúc 2048x 8 bit.



$\overline{W}$	$\Sigma$	$\overline{G}$	$D_1$ đến $D_8$	Chế độ (mode)
0	0	X	Dữ liệu xác định	Viết
1	0	0	Dữ liệu ra	Đọc
X	1	X	Hi_Z	Cấm vi mạch
1	0	1	Hi_Z	Cấm ra

(c)

Hình 6-4: Vi mạch RAM tĩnh TMS 4016

Chân 20 (kí hiệu  $\Theta$ ) dùng để đưa tín hiệu cho phép đưa ra (output enable) hay không vào vi mạch. Khi  $\Theta$  ở mức cao (H) thì đầu ra ở trạng thái trở kháng cao (Hi\_Z). Khi  $\Theta$  ở mức thấp (L) thì đầu ra sẽ được nối thông với hệ thống.

Chân 18 (ký hiệu  $\overline{S}$ ) dùng để đưa tín hiệu chọn chip (chip select) vào vi mạch. Khi  $\overline{S}$  ở mức cao (H), thì vi mạch bị đóng và không thể đọc hay viết dữ liệu.

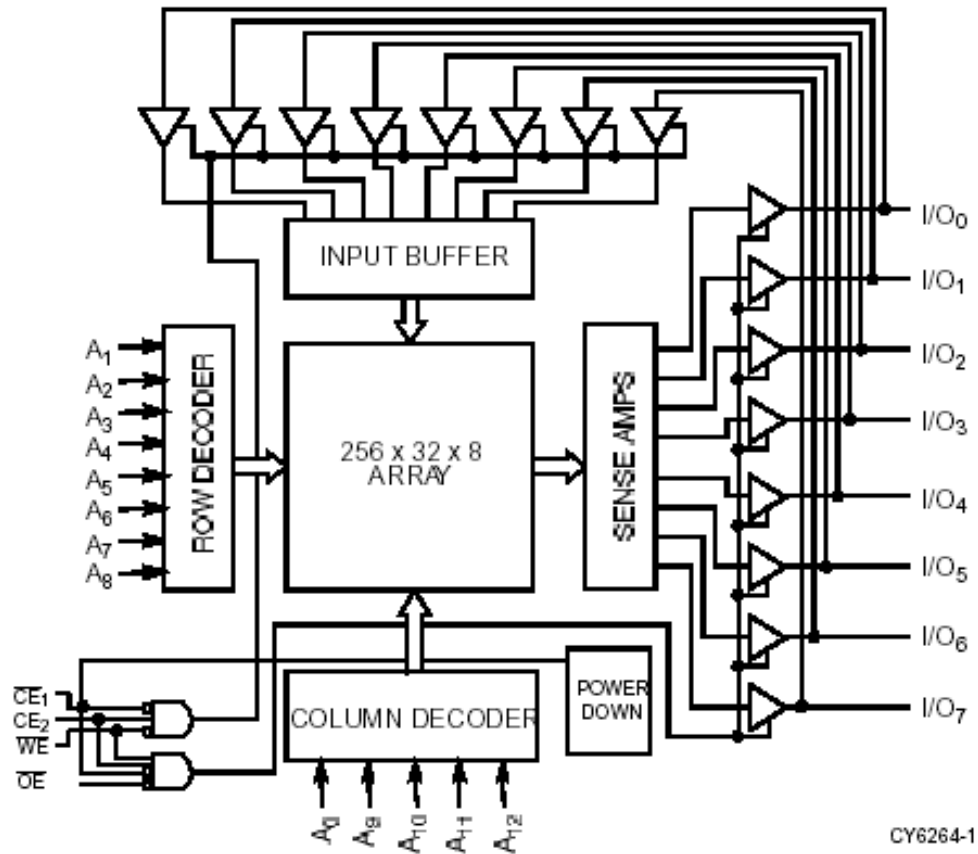
Chân 21 (ký hiệu  $\overline{W}$ ) dùng để đưa tín hiệu đọc - viết vào vi mạch,. Khi  $\overline{W}$  ở trạng thái cao là đọc (read), và khi  $\overline{W}$  ở trạng thái thấp là viết (write).

$A_0$  đến  $A_{10}$  là các đường địa chỉ. Số địa chỉ tối đa có thể sử dụng là  $2^{11} = 2048$  địa chỉ.  $D_1$  đến  $D_8$  là các đường dữ liệu. Vì vậy mà ở trên ta nói rằng vi mạch này có cấu trúc 2048x8 bit.

Bảng chân lý trên hình 6-4c cho biết rõ các hoạt động của vi mạch.

+ Ví dụ 2:

Ví dụ sau sẽ mô tả một loại SRAM CY6264 của hãng CYPRESS



Hình 6-5: Sơ đồ logic của SRAM 6264

Các chân chức năng:

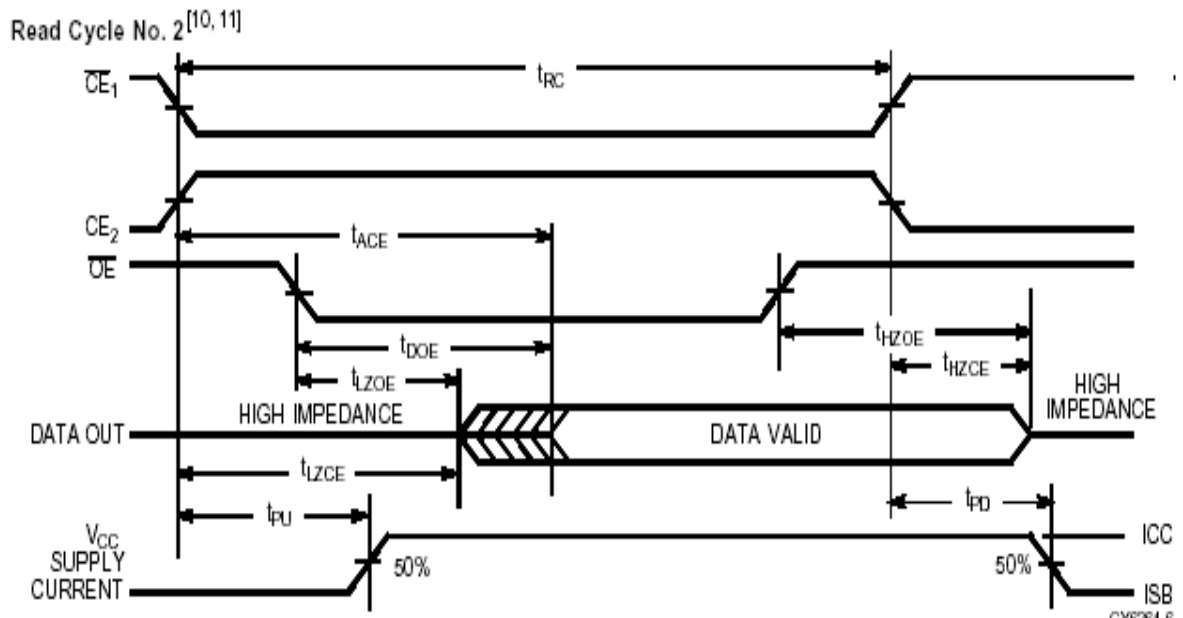
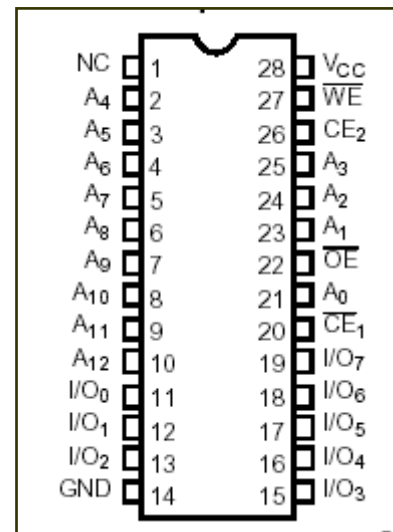
- A0...A12: Tín hiệu địa chỉ
- I/O0...I/O7 : Tín hiệu Data
- /CE1, CE2: Tín hiệu chọn chip
- /WE : Cho phép viết
- /OE: Cho phép đọc

Các thông số cơ bản:

- Tốc độ truy cập 55.70ns
- Chế tạo theo công nghệ CMOS:
  - Khi truy cập dòng điện đạt : 100mA
  - Khi không truy cập dẫn dòng : 20mA
- Dung lượng 4K x 8
- Phạm vi nhiệt độ làm việc: -50°C đến +150°C

Nguyên lý đọc dữ liệu được thể hiện qua hình 6-6:

Sơ đồ chân của SRAM 6264



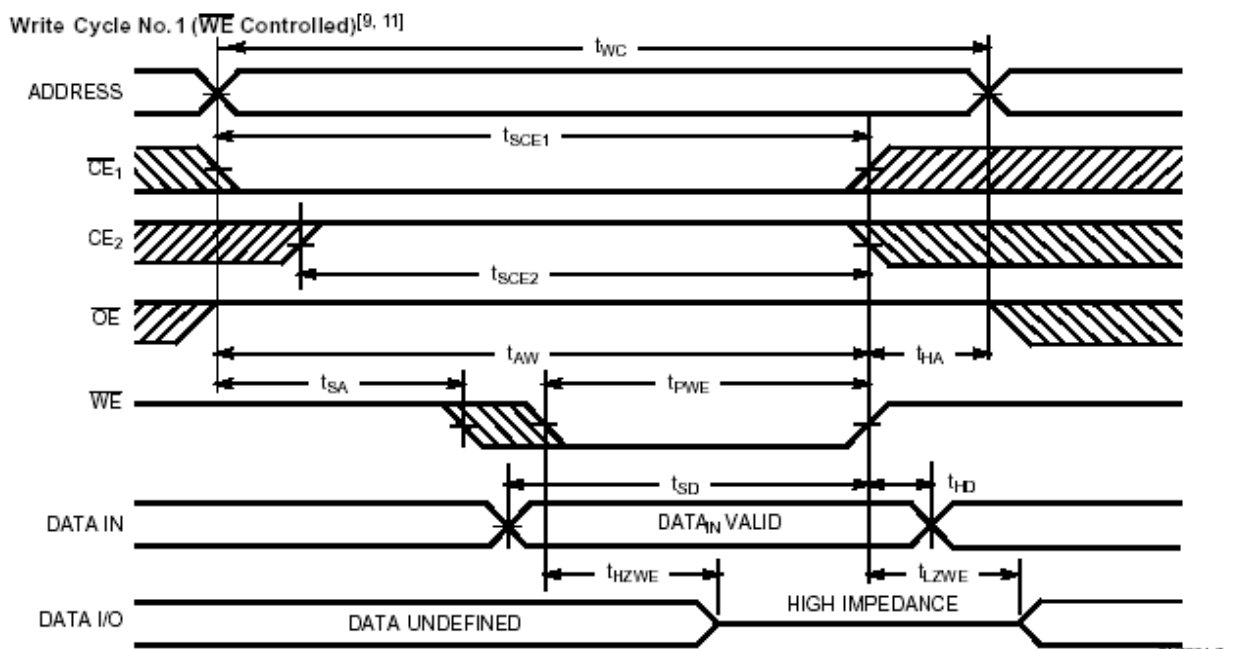
Hình 6-6: Một chu kỳ đọc của SRAM - 6264



Bảng thông số thời gian của một chu kỳ đọc dữ liệu:

Parameter	Description	6264-55		6264-70		Unit
		Min.	Max.	Min.	Max.	
READ CYCLE						
t <sub>RC</sub>	Read Cycle Time	55		70		ns
t <sub>AA</sub>	Address to Data Valid		55		70	ns
t <sub>OHA</sub>	Data Hold from Address Change	5		5		ns
t <sub>ACE1</sub>	$\overline{CE}_1$ LOW to Data Valid		55		70	ns
t <sub>ACE2</sub>	CE <sub>2</sub> HIGH to Data Valid		40		70	ns
t <sub>DOE</sub>	$\overline{OE}$ LOW to Data Valid		25		35	ns
t <sub>LZOE</sub>	$\overline{OE}$ LOW to Low Z	3		5		ns
t <sub>HZOE</sub>	$\overline{OE}$ HIGH to High Z <sup>[6]</sup>		20		30	ns
t <sub>LZCE1</sub>	$\overline{CE}_1$ LOW to Low Z <sup>[6]</sup>	5		5		ns
t <sub>LZCE2</sub>	CE <sub>2</sub> HIGH to Low Z	3		5		ns
t <sub>HZCE</sub>	$\overline{CE}_1$ HIGH to High Z <sup>[5, 6]</sup> CE <sub>2</sub> LOW to High Z		20		30	ns
t <sub>PU</sub>	$\overline{CE}_1$ LOW to Power-Up	0		0		ns
t <sub>PD</sub>	$\overline{CE}_1$ HIGH to Power-Down		25		30	ns

Nguyên lý viết dữ liệu được thể hiện qua hình 6-7:



Hình 6-7: Một chu kỳ viết của SRAM - 6264

Bảng thông số thời gian của một chu kỳ viết dữ liệu:

Parameter	Description	6264-55		6264-70		Unit
		Min.	Max.	Min.	Max.	
WRITE CYCLE <sup>[7]</sup>						
t <sub>WC</sub>	Write Cycle Time	50		70		ns
t <sub>SCE1</sub>	CE <sub>1</sub> LOW to Write End	40		60		ns
t <sub>SCE2</sub>	CE <sub>2</sub> HIGH to Write End	30		50		ns
t <sub>AW</sub>	Address Set-Up to Write End	40		55		ns
t <sub>HA</sub>	Address Hold from Write End	0		0		ns
t <sub>SA</sub>	Address Set-Up to Write Start	0		0		ns
t <sub>PWE</sub>	WE Pulse Width	25		40		ns
t <sub>SD</sub>	Data Set-Up to Write End	25		35		ns
t <sub>HD</sub>	Data Hold from Write End	0		0		ns
t <sub>HZWE</sub>	WE LOW to High Z <sup>[6]</sup>		20		30	ns
t <sub>LZWE</sub>	WE HIGH to Low Z	5		5		ns

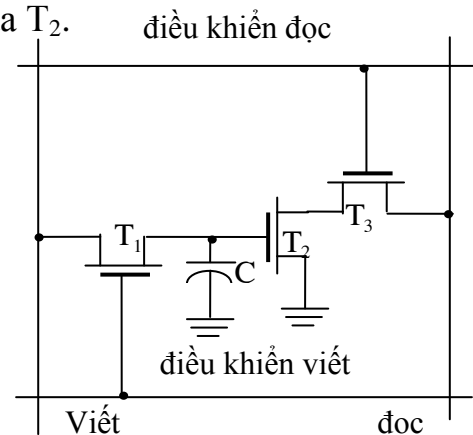
## b. Loại động (DRAM)

+ Đơn vị nhớ .

Với trở kháng vào vô cùng lớn của MOSFET người ta có thể chế tạo một loại đơn vị nhớ khác, sử dụng sự có mặt hay không có mặt của điện tích trên cổng của MOSFET để lưu trữ thông tin trong khoảng thời gian nhất định, gọi là RAM động dùng MOSFET. Trên hình 6-8 là một dạng thông dụng của loại bộ nhớ này.

Mạch sử dụng 3 MOSFET nhưng không nối theo kiểu FF.  $T_1$  và  $T_3$  hoạt động như các khoá điện tử,  $T_2$  có tác dụng như một mạch đảo. Dữ liệu được lưu trữ dưới dạng điện tích tại cực cổng của  $T_2$ .

Đơn vị nhớ có 4 đường: đường điều khiển đọc và điều khiển viết thuộc về một từ (word), đường viết để ghi dữ liệu vào và đường đọc để lấy dữ liệu ra. Ta giả sử Tranzitor là loại kênh P.



+ Thao tác viết.

Hình 6-8: Đơn vị nhớ RAM động dùng MOSFET

Khi muốn viết ta cho xung âm

vào đường điều khiển viết làm cho  $T_1$  dẫn. Nếu đường viết ở 0V, tụ C không có điện tích, ta ghi bit 0. Nếu đường bit ở -U thì tụ C có điện tích ta ghi bit 1.

+ Thao tác đọc.

Khi muốn đọc, ta cho xung âm vào đường điều khiển đọc và đường đọc. Nếu tụ C không có điện tích (bit 0) thì  $T_2$  ngưng dẫn làm  $T_3$  cũng ngưng dẫn nên trên đường đọc không có dòng điện ra. Ngược lại nếu tụ C có điện tích thì  $T_2$  dẫn làm  $T_3$  cũng dẫn cho nên trên đường đọc có xung dòng điện chảy, ở mạch ngoài sẽ có dòng điện và tạo ra một xung điện thế tương ứng.

Mạch có thể được làm đơn giản hơn bằng cách dùng một đường chung cho cả điều khiển đọc lẫn điều khiển viết. Lúc đó ta cần phải lưu ý rằng xung điều khiển viết thường phải bé hơn xung điều khiển đọc để tránh làm cho (khi đọc)  $T_1$  cũng dẫn điện khiến dữ liệu của tụ C thay đổi.

Bởi vì mạch luôn luôn bị mất dần điện tích trên tụ C nên dữ liệu không lưu trữ được vĩnh viễn. Do đó mạch phải được viết lại liên tục, người ta gọi đó là thao tác làm tươi bộ nhớ. Muốn viết lại trước hết ta phải đọc xem tụ C có điện tích hay không rồi căn cứ vào đó mà viết lại như cũ. Chu kỳ viết lại thường vào khoảng 2ms.

+Đặc tính

- Khi không đọc viết hay không viết lại, mạch hầu như không tiêu thụ công suất. Vì vậy nói chung loại RAM này tiêu thụ ít công suất so với các loại đã khảo sát ở trên.

- Vì chỉ sử dụng ít Tranzitor cho một đơn vị nhớ nên dễ chế tạo mạch với với dung lượng lớn trên một diện tích bán dẫn nhỏ.

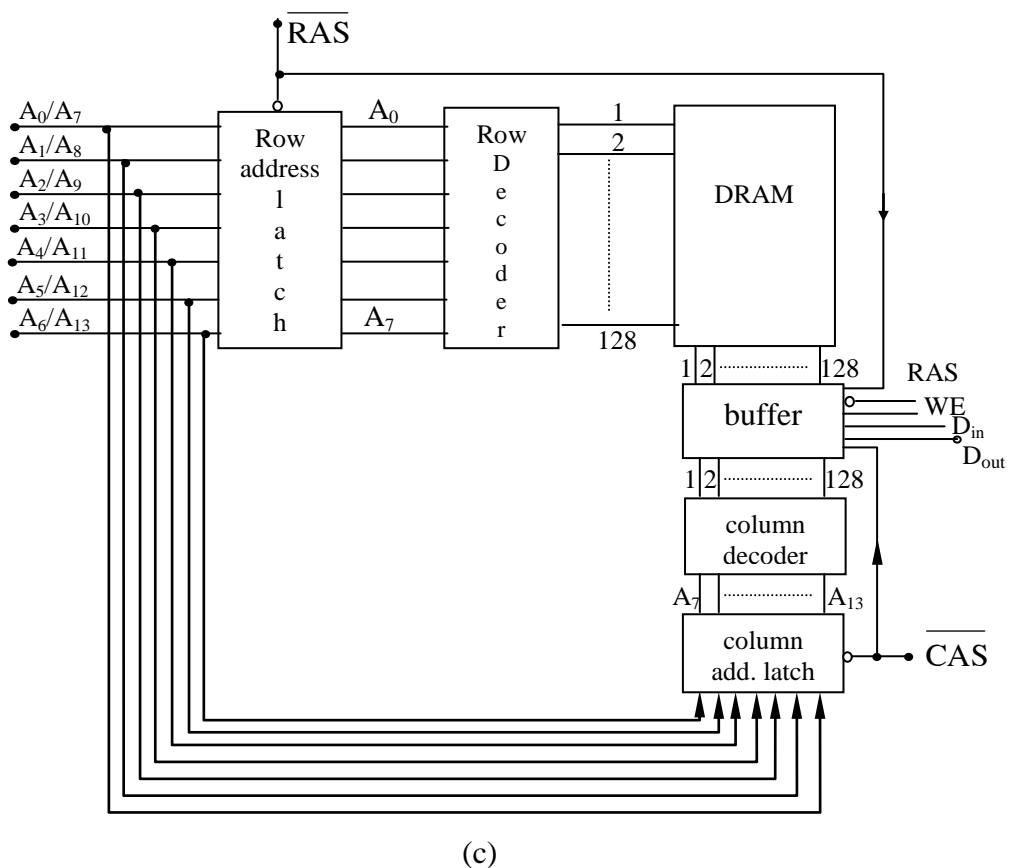
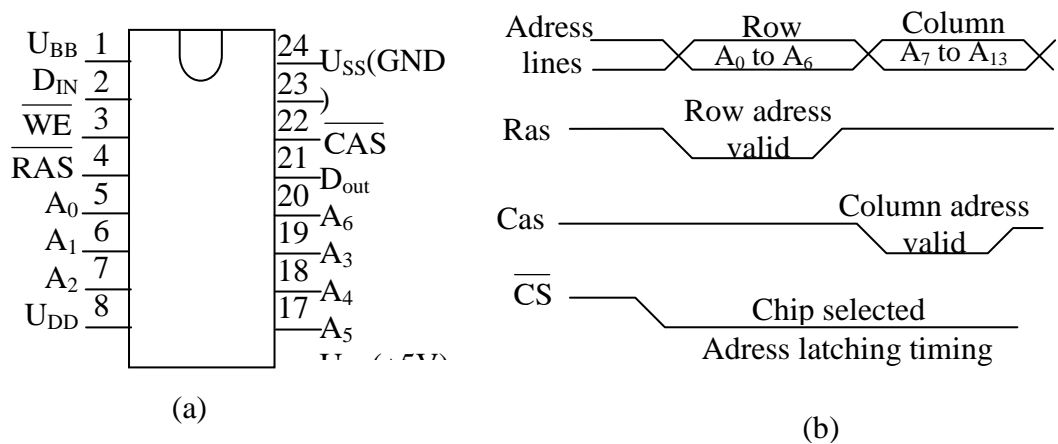
- Thời gian thâm nhập bé hơn các loại RAM tĩnh vì cấu trúc của mạch đơn giản hơn.

- Bộ nhớ loại này có một nhược điểm là luôn luôn cần phải làm tươi bộ nhớ khiến cho việc sử dụng trở nên rắc rối hơn và tín hiệu thường không phù hợp với TTL.

+ Ví dụ

Trên hình 6-9a là sơ đồ chân của bộ RAM động ký hiệu TMS 4116 thuộc loại MOSFET có cấu trúc 16K x 1 bit. Ta biết rằng 16K tương ứng với 16.384

vị trí nhớ trong thực tế, nên số đường địa chỉ yêu cầu phải là 14 đường (vì  $2^{14} = 16.384$ ). Để tiết kiệm số đường địa chỉ (và cũng chính là số chân của vi

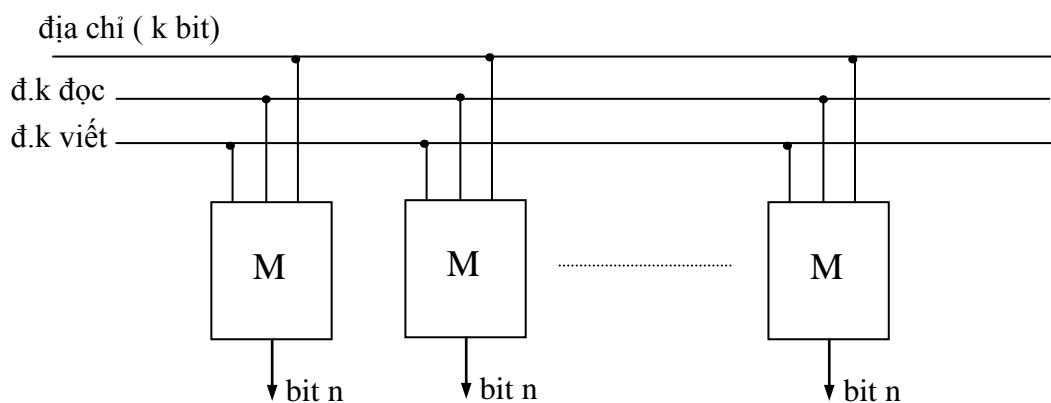


Hình 6-9: Vi mạch RAM động TMS 4116

mạch) hầu hết các loại RAM động đều sử dụng phương pháp kiểu chọn kênh nên thay vì cần 14 đường địa chỉ thì thực ra chỉ cần 7 đường. Trong quá trình đọc hay viết, đầu tiên 7 đường địa chỉ chứa các thông tin về hàng rồi sau đó lại chứa các thông tin về cột. Để đảm bảo cho sự hoạt động như vậy, người ta sử dụng hai tín hiệu là  $\overline{RAS}$  và  $\overline{CAS}$  có biểu đồ thời gian trên hình 6-9b. Khi  $\overline{RAS}$  thấp, thông tin trên các đường địa chỉ sẽ được mở thông đường qua mạch chốt địa chỉ hàng (row-address latch). Khi  $\overline{CAS}$  thấp thông tin trên các đường địa chỉ sẽ được mở thông đường qua mạch chốt địa chỉ cột(column-address latch). Các đường  $\overline{RAS}$  và  $\overline{CAS}$  không bao giờ được phép cùng thấp vì vậy sẽ làm rối loạn các đường địa chỉ. Đường  $\overline{WE}$  (write enable) dùng để đưa tín hiệu cho phép viết vào vi mạch. Khi viết thông tin vào vi mạch, đường  $\overline{WE}$  phải thấp, lúc đó thông tin trên đường  $D_{in}$  sẽ được viết vào địa chỉ đã được chọn. Còn khi muốn đọc ra đường  $\overline{WE}$  phải cao, lúc đó địa chỉ đã được chọn là thông tin sẽ được đưa ra đường  $D_{out}$ . Sơ đồ khối của vi mạch này được vẽ trên hình 6-9c.

### c. Bộ nhớ bán dẫn với dung lượng lớn.

Để có được các bộ nhớ bán dẫn với dung lượng lớn (và có thể rất lớn) người ta cần phải tổ chức bằng cách mắc các bộ nhớ có dung lượng nhỏ hơn (ở mức công nghệ cho phép) lại với nhau. Các bộ nhớ LSI thường được chế tạo như một mạch chứa M từ 1 bit. Do đó nếu muốn có M từ n bit ta mắc song song n mạch như vậy lại với nhau như trên hình 6-10.

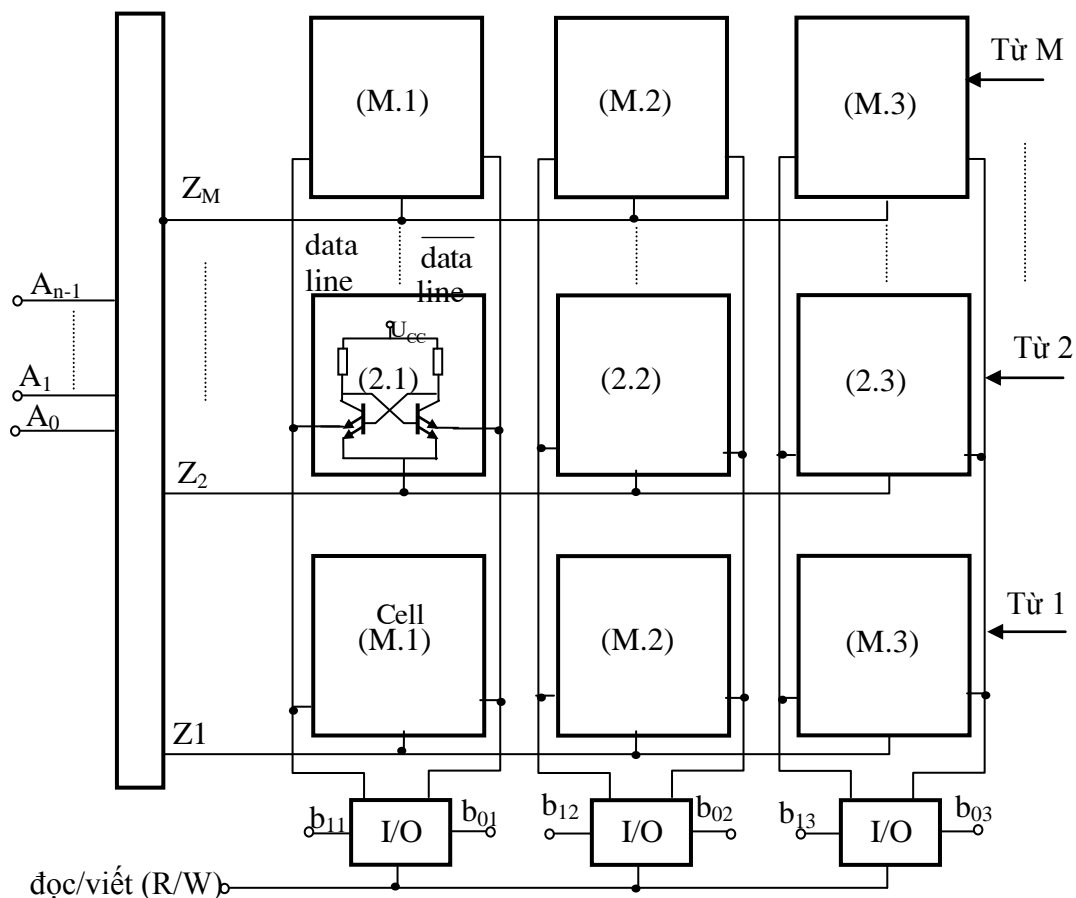


Hình 6-10: Mắc song song n mạch nhớ M từ 1 bit để tạo bộ nhớ chứa M từ n

Trên hình 6-11 là sơ đồ một bộ nhớ M từ 3 bit. Đơn vị nhớ (1,1) lưu trữ bit đầu tiên của từ thứ nhất, đơn vị (1,2) là bit thứ hai của từ thứ nhất ..v..v... Các đơn vị nhớ lưu trữ từ thứ nhất được định đường địa chỉ bằng cách nâng mức của đường  $Z_1$ . Nếu bộ nhớ lưu trữ M từ thì cần thiết M đường địa chỉ  $Z_1, Z_2, \dots, Z_M$ .

Việc xác định vị trí của sự lưu trữ trong bộ nhớ được thực hiện thông qua một bộ giải mã có  $\lambda$  bit địa chỉ  $A_0, A_1, \dots, A_{\lambda-1}$ .

Tất cả các đơn vị nhớ ( $C_{1,1}, C_{2,1} \dots C_{M,1}$ ) dùng để lưu trữ các bit đầu tiên của các từ được nối vào một cặp đường dữ liệu chung. Và cũng tương tự như vậy cho các bit thứ hai, thứ ba của từ. Các khối ký hiệu I/O là các khối điều khiển đọc/viết. Các bit  $b_{i,1}, b_{i,2}$  và  $b_{i,3}$  là các bit sẽ được viết vào bộ nhớ khi đường đọc viết (RW) chuyển lên mức logic 1. Các bit  $b_{0,1}, b_{0,2}$  và  $b_{0,3}$  là các bit sẽ được đọc ra từ bộ nhớ khi đường RW ở mức logic 0.

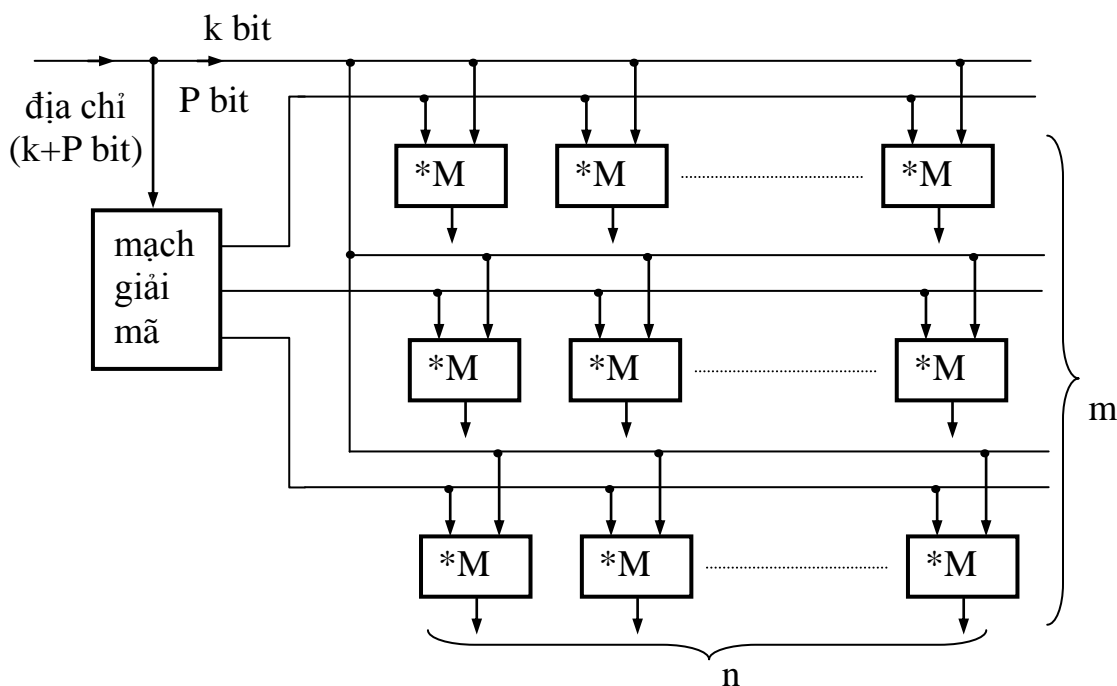


Hình 6-11: Bộ nhớ M từ 3 bit

Trong trường hợp muốn có số từ nhiều hơn  $M$  ta sử dụng cách mắc tổng hợp. Lúc này ta phải sử dụng thêm một mạch giải mã để chọn hàng ngang. đường ra từ mạch này được nối vào đầu vào điều khiển chọn mạch (CHIP SELECT) thường có sẵn trên một mạch LSI như trên hình 6-12. Trên hình này đầu vào có dấu \* là đầu CHIP SELECT.

Khi đường chọn mạch được kích thích thì mạch LSI mở cho tín hiệu xuất hiện ở đầu ra. Trong trường hợp nêu trên, nếu  $M=16$ ,  $n=8$ ,  $m=8$  ta sẽ có một bộ nhớ  $16 \times 8 = 128$  từ mỗi từ dài 8 bit. Muốn xác định địa chỉ ta cần một mã

dài 7 bit : 3 bit để chọn 8 hàng ( $m$ ) 4 bit để chọn 16 từ mỗi mạch LSI.

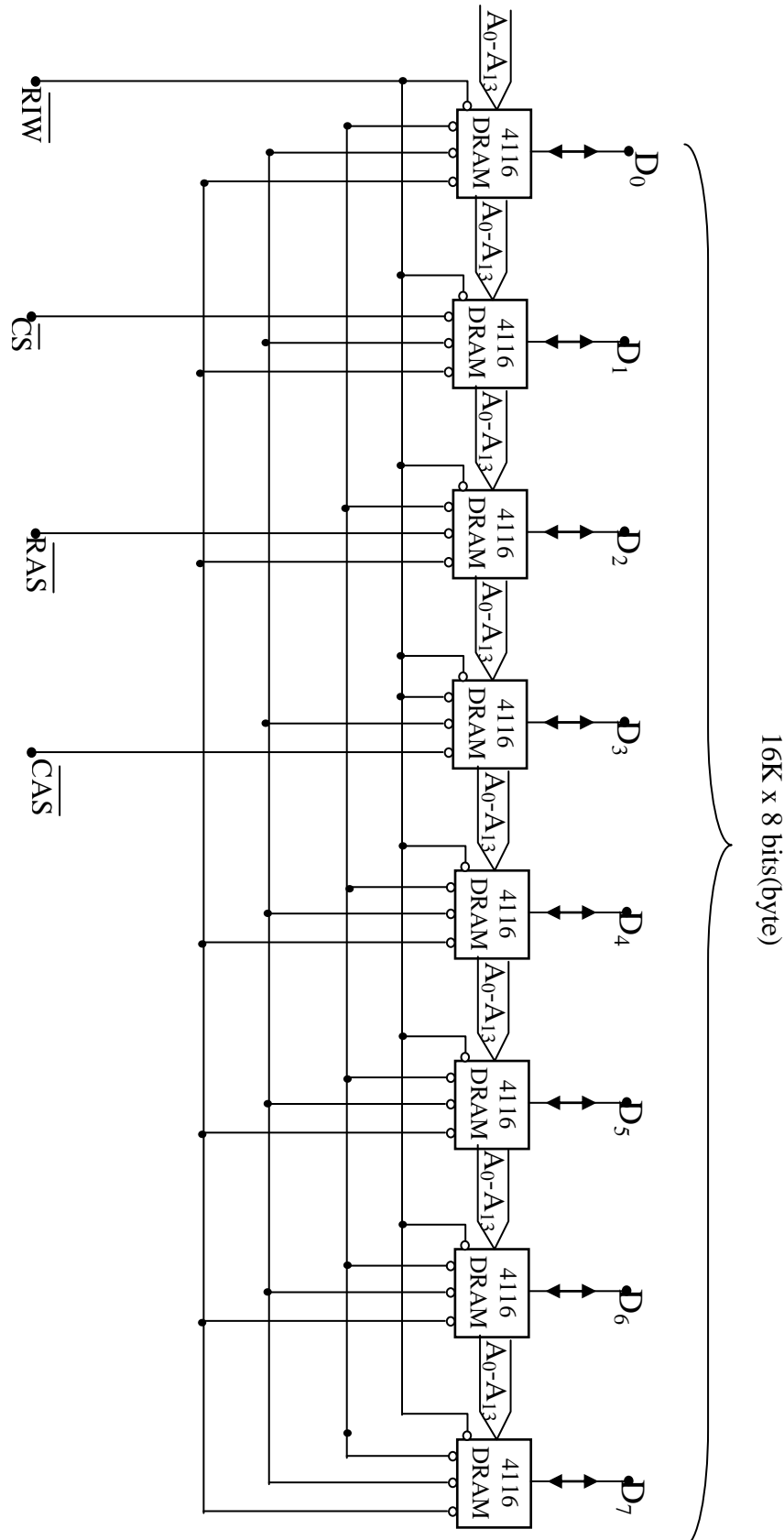


Hình 6-12: Cách mắc tổng hợp các bộ nhớ để có dung lượng

Hiện nay đã xuất hiện một bộ nhớ RAM có dung lượng hàng trăm kilobit trên một chip. thời gian thâm nhập và năng lượng tiêu thụ ngày càng bé.

+ Ví dụ: Trên hình 6-13 biểu diễn một bộ nhớ có dung lượng  $16K \times 8$  bit dùng 8 vi mạch loại TMS 4116 ( $16.384 \times 1$  bit). Trong trường hợp này ta lưu ý rằng tất cả các đường địa chỉ và đường điều khiển của vi mạch đều phải nối song song với nhau. Lúc đó nếu thao tác là viết thì tất cả các vi mạch đều

được đặt ở chế độ công tác viết, còn nếu thao tác là đọc thì tất cả các vi mạch đều được đặt ở chế độ công tác đọc. Các đường dữ liệu (8 bit) tạo thành một hệ thống nhớ có cỡ 8 bit (còn gọi là cỡ byte).



Hình 6-13: bộ nhớ dung lượng 16K x 8 bit dùng 8 vi mạch loại TMS 4116



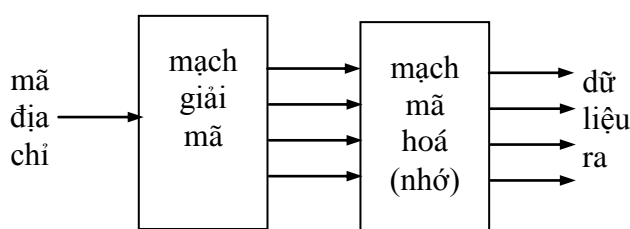
### 6.3. Các bộ nhớ ROM

Thuật ngữ ROM là viết tắt của từ tiếng Anh là Read Only Memory. Bộ nhớ ROM có đặc tính là chứa sẵn dữ liệu, khi cần ta chỉ đọc mà không viết ngay vào được. Dữ liệu trong bộ nhớ này đã được ghi vào từ trước và thường được lưu trữ một cách vĩnh viễn. Có loại ta không thể xoá được nội dung của nó và viết lại, có loại có thể xoá được nội dung của nó hay viết lại nhưng cần phải qua một thủ tục phức tạp (đây chính là các bộ nhớ bán dẫn cố định). Chính vì không cần viết ngay nên loại mạch này đơn giản hơn so với RAM vì không có các bộ phận cần cho thao tác viết vào.

Bộ nhớ ROM ngày càng thông dụng nhất là loại bộ nhớ bán dẫn. Cả hai công nghệ lưỡng cực và MOSFET đều được sử dụng chế tạo ROM. Thời gian thâm nhập (at) của bộ nhớ lưỡng cực vào khoảng 50 - 90 ns, còn đối với MOSFET thì gấp 10 lần (chậm hơn). ROM lưỡng cực nhanh hơn và có khả năng kích tốt hơn, trong khi đó bộ nhớ MOSFET có cùng dung lượng thì có kích thước nhỏ hơn và tiêu thụ công suất ít hơn.

Thực tế ROM chính là một mạch mã hoá do đó dưới dạng căn bản thì ROM thường bao gồm cả mạch giải mã như trên *hình 6-14*.

Dạng được giới thiệu trên là dạng phẳng trong đó mỗi địa chỉ vào xác định một đường của mạch giải mã, nghĩa là xác định một từ còn đầu ra là các bit của từ. Với cấu trúc như vậy mạch không khác gì một mạch mã hoá để chuyển từ một mã hệ 2 sang một mã nào đó.



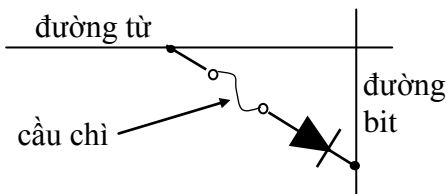
Hình 6-14: Dạng cơ bản của ROM

#### 6.3.1. Bộ nhớ ROM dùng diot.

Trước hết ta xét một ví dụ trên *hình 6-16*. Đây một mạch nhớ ROM gồm 8 từ, mỗi từ dài 4 bit. Vị trí của một bit là giao điểm của hai đường là đường từ

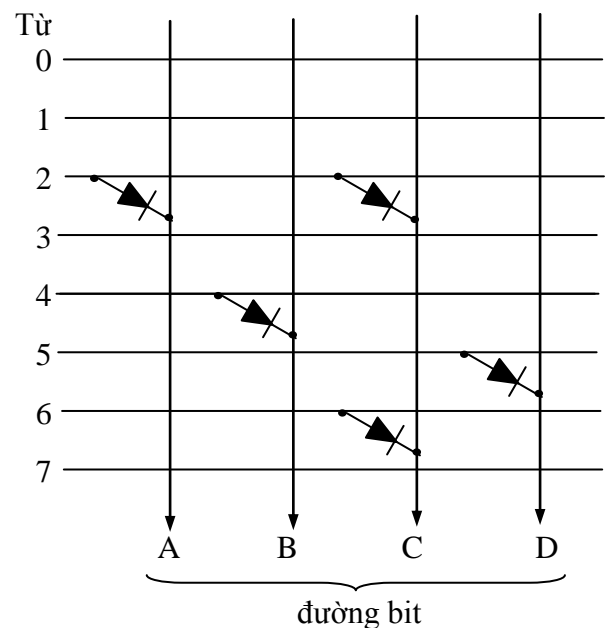
và đường bit. Tại giao điểm đó nếu có một điôt thì bit là 1, nếu không thì bit là 0. Muốn đọc thì ta cho địa chỉ, chẳng hạn 010 (2) vào mạch giải mã, tại đường của chữ số 2 sẽ có điện thế +U làm cho bít A và C bằng 1 (+U) còn bít B và D không có điôt nên không có tín hiệu ra tức là bằng 0. Như thế đường số 2 (hay địa chỉ 010) có dữ liệu 1010. Vậy có thể mắc điôt vào các giao điểm nào theo ý muốn và khi cần cũng có thể thay đổi vị trí các điôt để viết dữ liệu vào mạch nhớ.

Mạch như vậy rất đơn giản, gọn và khi không đọc thì hầu như không tiêu thụ năng lượng. Với dạng mạch này chắc chắn chúng ta có thể chế tạo theo phương pháp tích hợp trên quy mô lớn (LSI) tức là làm toàn bộ mạch nhớ trên một miếng silic nhỏ. Thực tế người ta chế tạo các bộ nhớ ROM dùng điôt với dung lượng 4096 từ, mỗi từ dài 1 bit trên một miếng silic khoảng vài  $\text{mm}^2$  và có at khoảng vài chục nanosec.



Hình 6-15: Nối cầu chì với điôt ở PROM

Trong trường hợp bộ nhớ LSI kiểu này người ta cũng còn có thể nối tiếp mỗi điôt một cầu chì nhỏ ở tất cả mọi giao điểm như trên hình 6-15 để tạo ra một loại ROM lập trình được (Programmable ROM hay



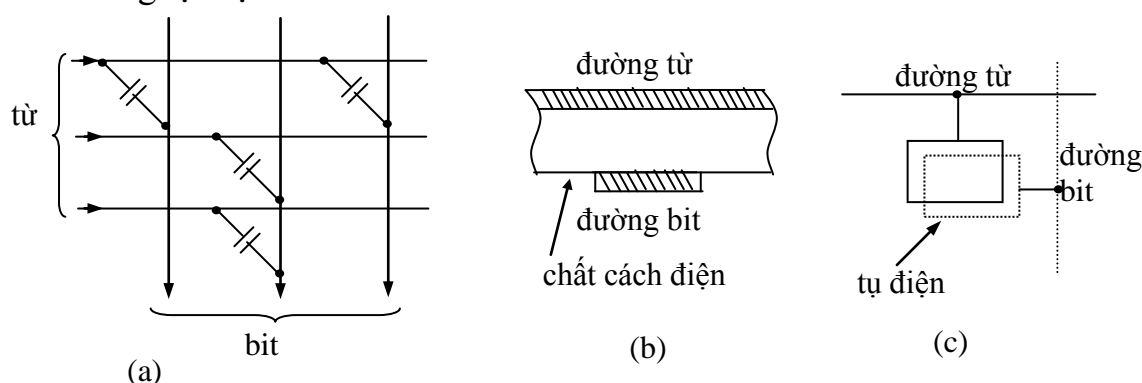
Hình 6-16: Bộ nhớ ROM dùng điôt

PROM). Khi sử dụng ta viết dữ liệu vào bộ nhớ bằng cách phóng vào bộ nhớ một xung dòng điện có cường độ đủ để cháy cầu chì tại đó, như vậy giao điểm này được viết bit 0. Còn giao điểm nào không bị làm đứt cầu chì thì sẽ là bit

1. Với loại này nhà chế tạo làm hàng loạt bộ nhớ LSI giống nhau và khi sử dụng thì người sử dụng sẽ ghi vào theo ý muốn của mình. Dĩ nhiên là sau khi viết xong ta không thể thay đổi được nội dung lần nữa.

### 6.3.2. Bộ nhớ ROM dùng tụ điện

Nếu thay đổi mạch nhớ ROM dùng diot bằng cách thay diot ở mỗi giao điểm bằng một tụ điện như trên hình 6-17a, ta lại có một loại ROM khác là ROM dùng tụ điện.

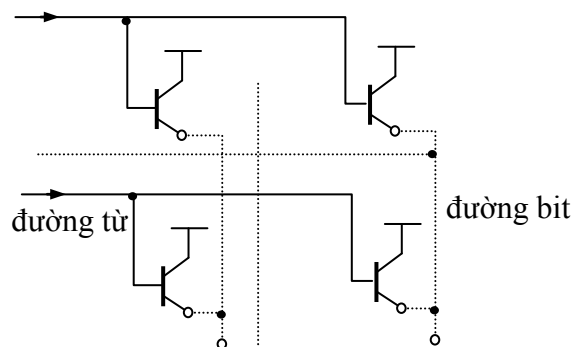


Hình 6-17: Bộ nhớ ROM dùng tụ điện.

Khi cho một xung vào đường từ, tại giao điểm nào có tụ điện thì đường bit tương ứng sẽ có xung ra (bit 1). Trong thực tế người ta chế tạo loại bộ nhớ này bằng phương pháp mạch in (hình 6-17b). Trong đó các đường từ và bit nằm cách nhau bởi một lớp cách điện. Tại giao điểm có tụ điện thì đường từ và bit có thêm hai miếng kim loại mỏng, cách nhau bởi chất cách điện (hình 6-17c). Với cấu trúc như vậy ta có thể chế tạo các bộ nhớ có cấu trúc tương đối lớn.

### 6.3.3 Bộ nhớ ROM dùng Tranzitor lưỡng cực.

Trong bộ nhớ ROM dùng Tranzitor lưỡng cực là phần tử nằm ở giao điểm của đường từ và đường bit. Một mạng nhớ (ma trận nhớ) dùng Tranzitor lưỡng cực tiêu biểu được biểu diễn như trên hình 6-18.



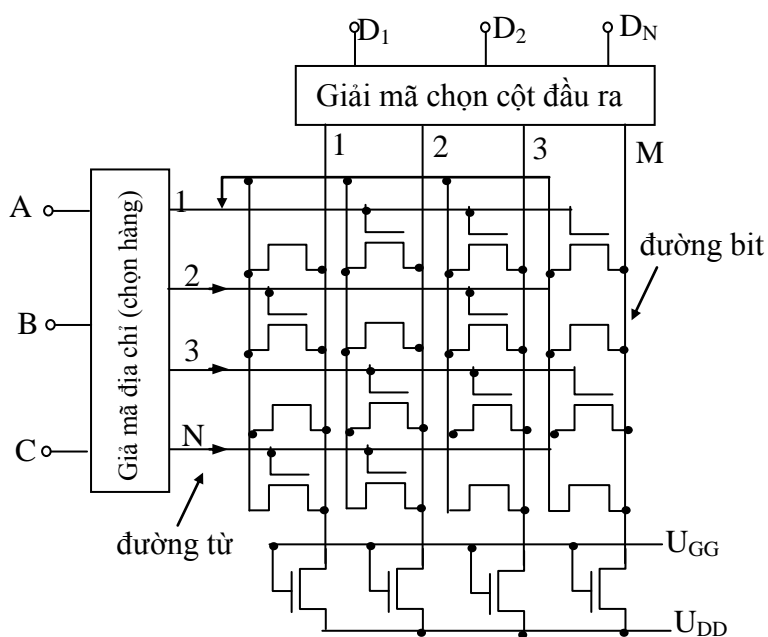
Hình 6-18: Bộ nhớ ROM dùng Tranzitor lưỡng cực

Một Tranzitor giao giữa đường từ và đường bit được tạo bởi đường nối với cực phát của Tranzitor. Việc nối được hoàn thành trong quá trình chế tạo các bộ khuếch đại nhạy cảm dòng điện ở cột được chọn để xác định mức logic 0 hay là 1.

### 6.3.4 Bộ nhớ ROM dùng MOSFET

MOSFET là công nghệ lý tưởng để chế tạo ROM nhờ mật độ linh kiện cao cho phép thiết kế các cấu trúc dạng ma trận chứa hàng nghìn phần tử MOSFET.

Trên hình 6-19 là sơ đồ mạch ROM sử dụng MOSFET kênh P. Sự có mặt của một MOSFET nối các đường từ, đường bit được xác định bởi lớp ôxit ở vị trí của mỗi Tranzitor. Trong quá trình chế tạo nếu lớp ôxit đó dày thì xem như không có Tranzitor, còn nếu lớp ôxit mỏng thì ứng với



Hình 6-19: Bộ nhớ ROM dùng MOSFET kênh P

một Tranzitor ở mức logic 1 cho vị trí ấy. Việc cấy hay khử các Tranzitor trong ma trận được tạo thành trong quá trình chế tạo mà người sử dụng không có thể can thiệp vào để thay đổi được.

### 6.3.5 Các bộ nhớ PROM, EPROM và EAROM

Các bộ nhớ PROM, EPROM và EAROM được viết bởi người sử dụng sau khi đã chế tạo.

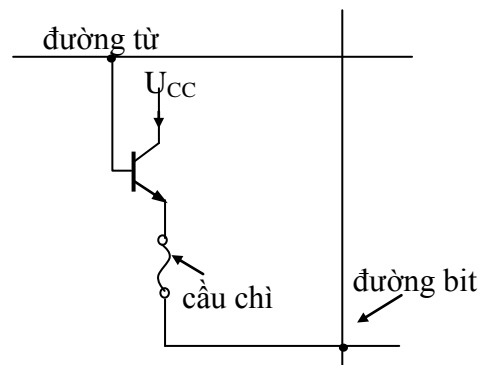
- Bộ nhớ PROM chỉ có thể viết được một lần. Việc viết vào PROM yêu cầu phải có những mạch cung cấp dòng điện đủ lớn để đốt cháy cầu chì ở những bit mong muốn. Những thiết bị viết này có thể do người sử dụng tự thiết kế hoặc mua sẵn.

- EPROM và EAROM mặc dù về giá thành thì đắt hơn PROM nhưng có thể viết nhiều lần nên tỏ ra thuận tiện trong việc thiết kế thử nghiệm hoặc sản xuất đối với khối lượng nhỏ. EPROM được xóa bằng tia cực tím còn EAROM thì được xóa bằng dòng điện nên khi xóa có thể để trong mạch điện mà không cần phải gỡ ra như đối với EPROM.

#### a. Bộ nhớ PROM nối bằng cầu chì.

Một bộ nhớ PROM tiêu biểu đã được giới thiệu trên *hình 6-15* trong đó chúng ta lưu ý đến sự có mặt của cầu chì nối tiếp với điôt. Người ta cũng sản xuất ra một loại PROM mà cầu chì được nối tiếp với một Tranzitor (thay vì điôt) như trên *hình 6-20*. Cầu chì được làm bằng Niken -Crome hay silic đa tinh thể. Khi đường từ được chọn Tranzitor mở. Nếu có cầu chì thì đường bit được đưa lên điện thế  $5V(U_{cc})$ . Nếu cầu chì đã làm bị đứt (trong quá trình viết vào) thì đường bit bị thả nổi.

Bộ nhớ PROM được xác định bằng chương trình làm đứt cầu chì ở các phần tử giao nhau hay không. Đối với loại PROM có cầu chì chế tạo bằng đa tinh thể silic xung làm đứt cầu chì có biên độ 20-30mA và độ rộng tăng dần cho đến khi cầu chì đứt.



Hình 6-20: Phần tử nhớ PROM nối bằng cầu chì

#### b. Bộ nhớ EPROM có cửa thả nổi.

Các bộ nhớ EPROM (Erasable PROM) có cửa thả nổi được sử dụng rất thuận tiện khi thiết kế các hệ thống đặc biệt là trong kỹ thuật vi xử lý. Vi mạch nhớ loại này đầu tiên do hãng Intel đưa ra năm 1971, đó là vi mạch 1702 có dung lượng 256 từ 8 bit sử dụng công nghệ PMOS và đóng trong vỏ 24 chân.

EPROM được thiết kế với các ô nhớ có thể xóa và viết lại được. Hiện nay EPROM đã được chế tạo dưới nhiều dạng với dung lượng khác nhau.

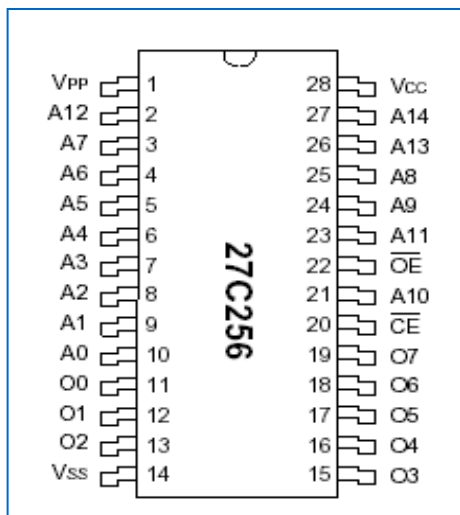
VD: Sau đây là một ví dụ về bộ nhớ EPROM 27C256 đang được sử dụng phổ biến:

Các thông số cơ bản:

Tốc độ truy cập 90ns

Công nghệ CMOS:

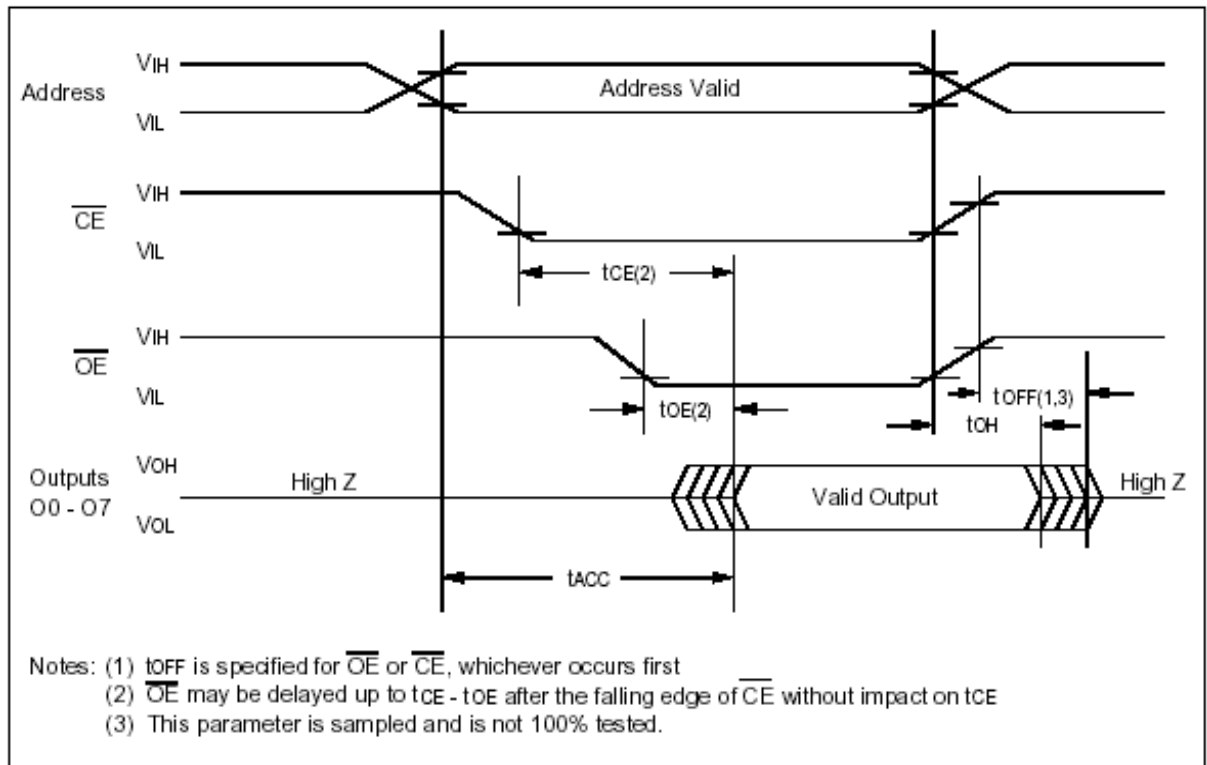
- Dòng khi truy cập : 20mA
- Dòng khi không truy cập : 100uA
- Dung lượng 32K x 8
- Độ bền dữ liệu khoảng 200 năm
- Phạm vi nhiệt độ làm việc: -40°C to +125°C



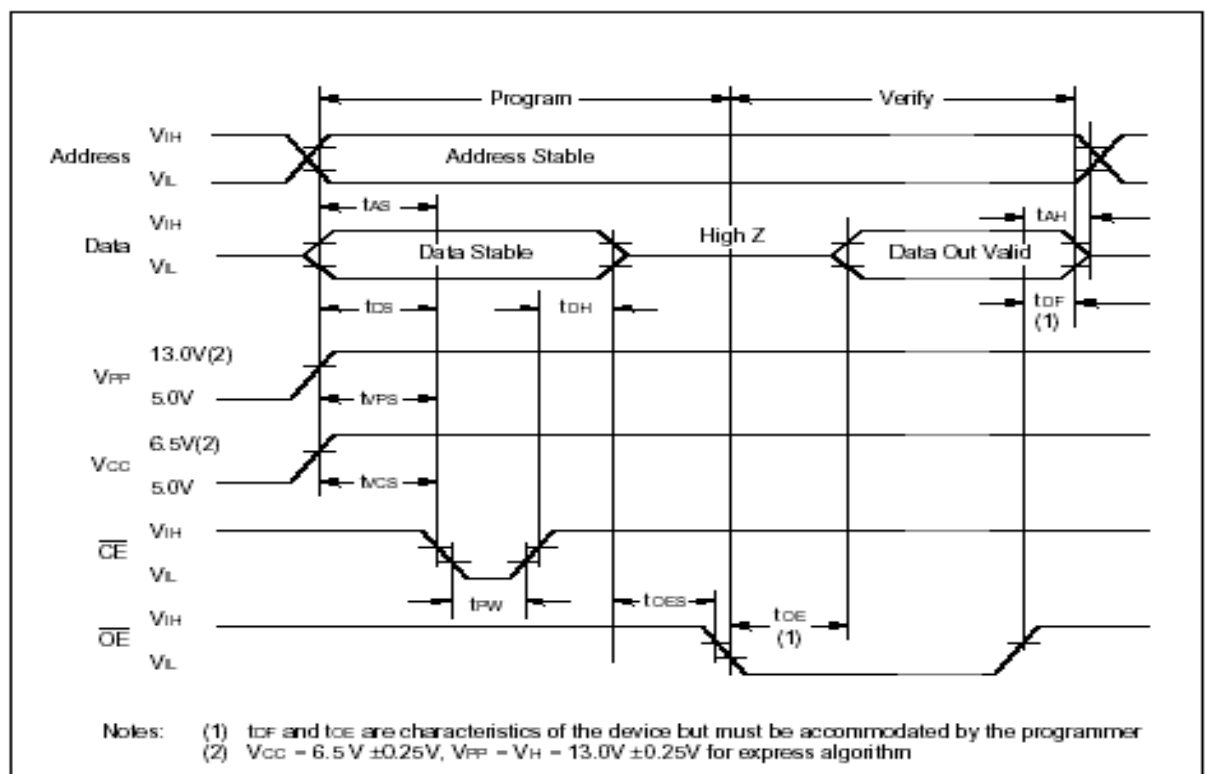
Hình 6-21: Sơ đồ chân và hình dáng của EPROM 27C256

Chức năng các chân:

- A0... A14 : Tín hiệu địa chỉ
- O0...O7 : 8 bit dữ liệu
- /CE: Chân chọn chip
- /OE : Chân tín hiệu cho phép ra
- Vpp : Chân cấp tín hiệu nạp chip



Hình 6-22: Giải đồ thời gian khi đọc dữ liệu



Hình 6-23: Giải đồ thời gian khi nạp dữ liệu

**b. Bộ nhớ EAROM.**

Bộ nhớ EAROM (Electrically Alterable ROM) thay đổi nội dung được nhờ các tín hiệu điện tương tự như đối với bộ nhớ RAM. Chúng có thời gian viết và xoá từng ô riêng biệt chẳng hạn vi mạch ER400 của hãng General Instruments có dung lượng 1024 từ 4 bit, thời gian viết là 1ms, thời gian xoá là 10 ms và thời gian thâm nhập là 900ns.

EAROM thường được sử dụng trong những ứng dụng đòi hỏi không những bộ nhớ bất biến mà nội dung còn có thể thay đổi khi cần thiết. Nhược điểm quan trọng của bộ nhớ EAROM là cần số lượng các mức điện áp khác nhau khi xoá, viết và đọc.

**6.3.6 Ứng dụng của bộ nhớ ROM.**

Bộ nhớ ROM có nhiều ứng dụng khác nhau. Dưới đây là một số ứng dụng tiêu biểu:

- Chuyển mã : khi cho vào đầu vào một từ mã nào đó thì đầu ra của bộ nhớ ROM sẽ là một từ dưới dạng mã khác.
- Chứa các lệnh điều khiển cho các phần tử của một thiết bị làm việc. Khi cần thực hiện một công việc gì, máy sẽ tìm ở một địa chỉ tương ứng mà tại đó có viết các chi tiết cần thực hiện.
- Chứa các dữ liệu mà máy thường không dùng đến, chẳng hạn các số  $\pi$ , e, ..v..v...



## CHƯƠNG 7: BIẾN ĐỔI SỐ - TƯƠNG TỰ VÀ BIẾN ĐỔI TƯƠNG TỰ - SỐ

### 7.1. Giới thiệu chung

Do sự phát triển nhanh chóng của kỹ thuật điện tử số, đặc biệt là sự ứng dụng phổ biến của máy tính điện tử số, nên thường phải dùng mạch số để xử lý tín hiệu tương tự.

Muốn dùng hệ thống xử lý tín hiệu tương tự, thì phải biến đổi tín hiệu tương tự thành tín hiệu số tương ứng, rồi đưa vào được hệ thống số (máy tính chẳng hạn) xử lý. Mặt khác, thường có yêu cầu biến đổi tín hiệu số (kết quả xử lý) thành tín hiệu tương tự tương ứng để đưa ra sử dụng. Chúng ta gọi sự chuyển đổi từ tín hiệu tương tự sang tín hiệu số là chuyển đổi AD, và mạch điện thực hiện công việc đó là ADC. Chúng ta gọi sự chuyển đổi từ tín hiệu số sang tín hiệu tương tự là chuyển đổi DA, mạch điện tương ứng là DAC.

Để có kết quả xử lý nhận được chính xác thì chuyển đổi AD, DA phải có độ chính xác đủ cao. Mặt khác muốn dùng hệ thống số điều khiển, giám sát các quá trình biến đổi nhanh thì ADC và DAC phải có tốc độ đủ lớn. Vậy độ chính xác và tốc độ chuyển đổi là các đặc trưng kỹ thuật chủ yếu để đánh giá chất lượng ADC và DAC.

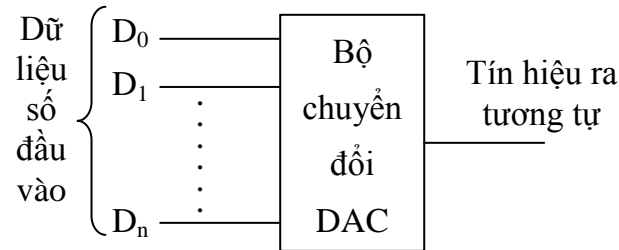
Những năm gần đây, sự phát triển của kỹ thuật chuyển đổi AD và DA là cực kỳ nhanh chóng. Đặc biệt để thích hợp yêu cầu sản xuất các IC một chip ADC, DAC người ta đã không ngừng nghiên cứu và ứng dụng rất nhiều phương pháp chuyển đổi mới và rất nhiều mạch điện chuyển đổi mới. Trong chương này, chúng ta chỉ nghiên cứu tìm hiểu DAC hình T và ADC xấp xỉ tiệm cận, là những bộ biến đổi điển hình được dùng nhiều.

### 7.2. Bộ chuyển đổi Số - Tương tự (DAC)

\* Bộ chuyển đổi số – tương tự được dùng để chuyển đổi các tín hiệu số thành tín hiệu tương tự.

\* Bản chất của quá trình chuyển đổi DAC là quá trình nhận một nhóm xung dưới dạng mã nhị phân hay mã nhị – thập phân (BCD) sau đó biến đổi

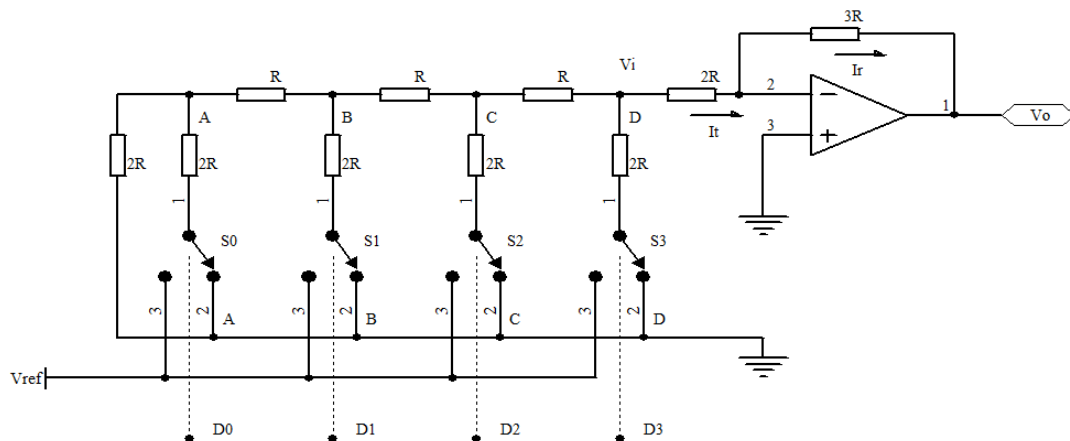
thành một mức điện thế hay cường độ dòng điện tương tự nào đó. Mức độ (hay độ lớn) của tín hiệu áp (dòng) này tỉ lệ với giá trị số đầu vào nhận được.



Hình 7-2-1: Sơ đồ khối của một bộ DAC

### 7.2.1. Bộ biến đổi DAC điện trở hình T

Hình 7-2-2 là sơ đồ DAC điện trở hình T 4 bit. Hai loại giá trị điện trở  $R$  và  $2R$  được mắc thành 4 cực hình T nối dây chuyền.  $S_3, S_2, S_1, S_0$  là các chuyển mạch tương tự. Bên phải hình có bộ khuếch đại đảo dùng khuếch đại thuật toán.  $V_{ref}$  là điện áp chuẩn tham khảo.  $D_3 D_2 D_1 D_0$  là mã nhị phân 4 bit đầu vào.  $V_o$  là điện áp tương tự đầu ra. Các chuyển mạch  $S_3, S_2, S_1, S_0$  chịu sự điều khiển của tín hiệu  $D_3 D_2 D_1 D_0$  tương ứng khi  $D_i = 1$  thì  $S_i$  đầu vào  $V_{ref}$ , khi  $D_i = 0$  thì  $S_i$  nối đất.



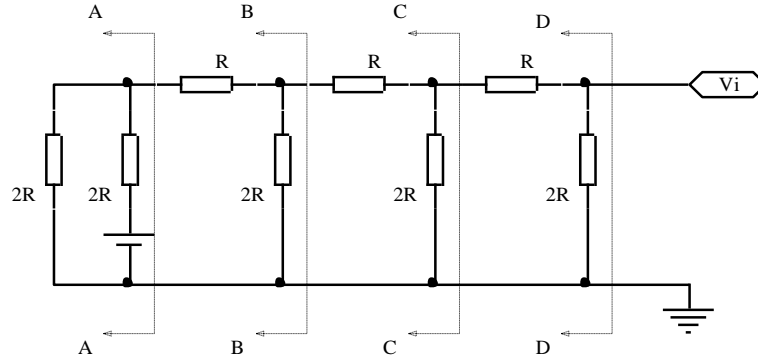
Hình 7-2-2 : Sơ đồ DAC điện trở hình T 4bit

#### 1. Nguyên lý làm việc

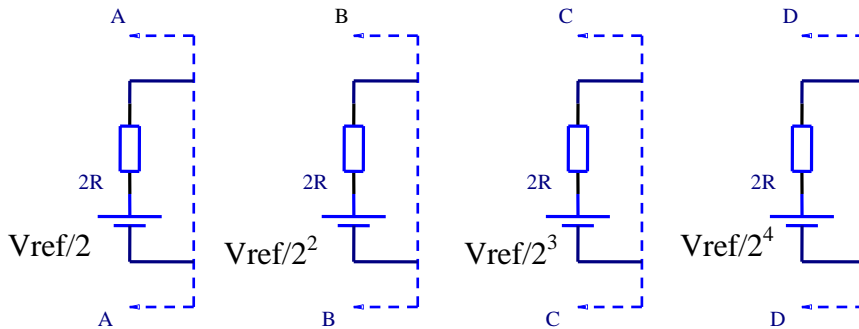
Để tiện thuyết minh nguyên lý làm việc của mạch hình 7-2-2, chúng ta hãy đơn giản hoá mạng điện trở hình T.

Nếu  $D_3 D_2 D_1 D_0 = 0001$  thì chỉ có  $S_0$  đầu vào  $V_{ref}$ ,  $S_3 S_2 S_1$  đều nối đất. Áp dụng định lý Thevenin tuần tự đơn giản hoá mạch từ đầu AA sang phải. Ta phải thấy rằng cứ qua mỗi mắt mạch (A,B,C,D) thì điện áp ra suy giảm đi

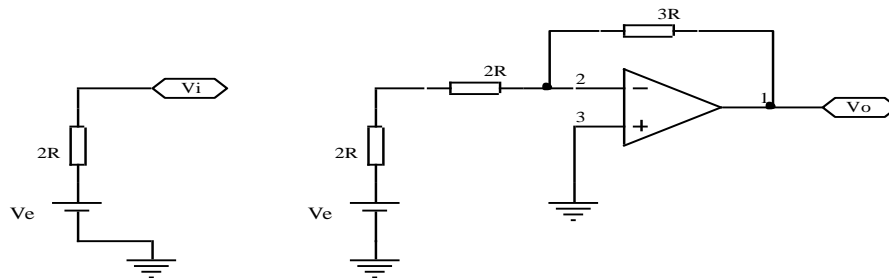
một nửa. Vậy nếu  $V_{ref}$  đầu vào  $S_0$  thì trên đầu ra DD chỉ còn  $V_{ref}/2^4$ . Cũng với phương pháp trên, xét riêng  $S_3 S_2 S_1$  đầu vào  $V_{ref}$  thì trên đầu ra tương ứng ( $D_3 D_2 D_1 D_0 = 0010, 0100, 1000$ ) có các điện áp  $V_{ref}/2^3, V_{ref}/2^2, V_{ref}/2^1$ . Điện trở tương đương của phần mạch bên trái DD bao giờ cũng là  $R$ .



Hình 7-2-3: a) Mạch điện trở hình T khi  $D_3 D_2 D_1 D_0 = 0001$



Hình 7-2-3: b) Mạch điện trở hình T khi  $D_3 D_2 D_1 D_0 = 0001$



Hình 7-2-3: c) Mạch tương đương của hình 7-2-2

Áp dụng nguyên lý xếp chồng đối với các giá trị điện áp trên, ta có mạng tương đương mạng điện trở hình T trên hình 7-2-3c. Trong đó nội trở tương đương là  $R$ , sức điện động nguồn tương đương là  $V_e$ :

$$V_e = (V_{ref}/2^4)(D_3 \cdot 2^3 + D_2 \cdot 2^2 + D_1 \cdot 2^1 + D_0 \cdot 2^0) \quad (7-2-1)$$

Hình (7-2-3c) là sơ đồ tương đương toàn mạch, theo lí thuyết mạch khuếch đại thuật toán, ta có điện áp tương tự đầu ra  $V_0$  là :

$$V_0 = -V_e = - (V_{ref}/2^4)(D_3*2^3 + D_2*2^2 + D_1*2^1 + D_0*2^0) \quad (7-2-2)$$

Biểu thức (7-2-2) chứng tỏ rằng biên độ điện áp đầu ra tỉ lệ thuận với giá trị tín hiệu số đầu vào. Có thể thấy rằng đối với DAC điện trở hình T n bit thì điện áp tương tự đầu ra  $V_0$  là

$$V_0 = (V_{ref}/2^n)(D_{n-1}*2^{n-1} + D_{n-2}*2^{n-2} + \dots + D_1*2^1 + D_0*2^0) \quad (7-2-3)$$

## 2. Sai số chuyển đổi

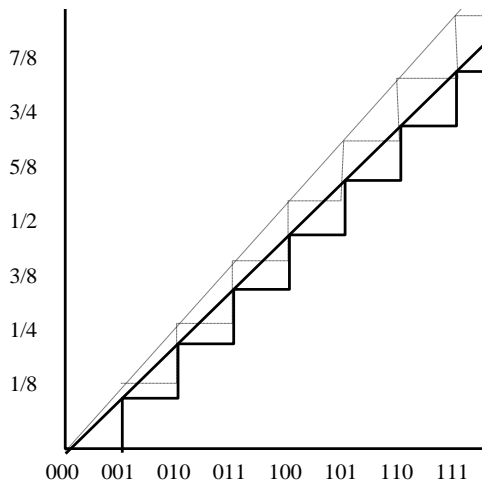
Các nguyên nhân dẫn đến sai số của DAC điện trở hình T là :

- Sai lệch điện áp chuẩn  $V_{ref}$ .
- Sự trôi điểm 0 của khuếch đại thuật toán .
- Điện áp rơi trên điện trở tiếp xúc của tiếp điểm chuyển mạch.
- Sai số của điện trở.

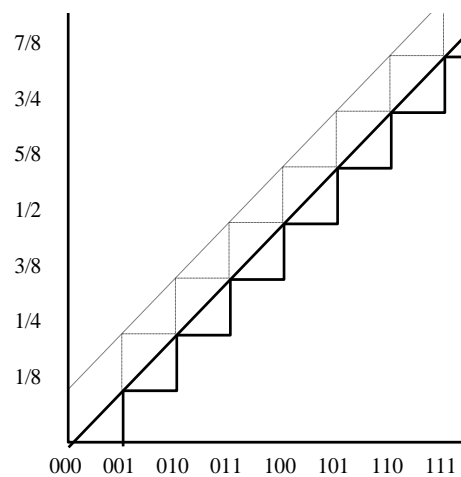
Từ (7-2-3) ta có sai số chuyển đổi DA do riêng sai lệch điện áp chuẩn  $V_{ref}$  gây ra :

$$\Delta V = (\Delta V_{ref}/2^n)(D_{n-1}*2^{n-1} + D_{n-2}*2^{n-2} + \dots + D_1*2^1 + D_0*2^0) \quad (7-2-4)$$

Biểu thức (7-2-4) chứng tỏ sai số của điện áp tương tự  $\Delta V$  tỉ lệ thuận với sai lệch  $\Delta V_{ref}$  và tỉ lệ thuận với giá trị tín hiệu số đầu vào. Hình 7-2-4 biểu thị bằng đường nét đứt sự biến đổi của  $V_0$  khi  $\Delta V_{ref}$  không đổi. Biểu thức (7-2-4) cũng chứng tỏ rằng nếu giá trị tín hiệu không đổi thì sai số  $\Delta V_0$  tỉ lệ thuận với sai lệch  $\Delta V_{ref}$ . Chúng ta gọi sai lệch này là sai lệch hệ số tỉ lệ.



Hình 7-2-4 : Sai số hệ số tỉ lệ



Hình 7-2-5: Sai số trôi

Sự trôi điểm 0 của bộ khuếch đại thuật toán ảnh hưởng như nhau tới các giá trị tín hiệu số được biến đổi; sai số  $\Delta V_0$  do trôi điểm 0 không phụ thuộc vào giá trị tín hiệu số. Hình 7-2-5 biểu thị bằng nét đứt đặc tuyến điện áp ra khi có sai số trôi, đặc tuyến này bị dịch ngang so với đặc tuyến lý tưởng.

Chúng ta gọi sai số này là sai số trôi (hay sai số dịch ngang). Các chuyển mạch không phải lý tưởng. Thực tế điện áp rơi khi nối thông của mạch điện chuyển mạch không thể tuyệt đối bằng 0. Vậy điện áp rơi này đóng vai trò tín hiệu sai số đưa đến đầu vào mạng điện trở hình T. Từ sự phân tích hình 7-2-3 ta thấy điện áp rơi của mỗi chuyển mạch sẽ gây ra sai số khác nhau đối với điện áp tương tự đầu ra. Gọi  $\Delta S_i$  là điện áp rơi của chuyển mạch  $S_i$  nối thông. Tổng sai số chuyển đổi do điện áp rơi các chuyển mạch là

$$|\Delta V_0| = (2^3 \Delta V_{s3} + 2^2 \Delta V_{s2} + 2^1 \Delta V_{s1} + 2^0 \Delta V_{s0}). 1/2^4$$

Cần lưu ý rằng  $\Delta V_0$  không phải là hằng số, cũng không tỉ lệ thuận với giá trị tín hiệu số. (nói chung  $\Delta V_{s0} \neq \Delta V_{s2} \neq \Delta V_{s3} \neq \Delta V_{s4}$ , hơn nữa điện áp rơi của cùng chuyển mạch cũng lấy các giá trị khác nhau trong hai trường hợp: nối vào  $V_{ref}$  hay nối đất). Chúng ta gọi sai số này là sai số phi tuyến.

Sai số giá trị điện trở cũng gây ra sai số phi tuyến sai số của các điện trở không như nhau, tác động gây sai số chuyển đổi DA của những điện trở khác nhau về vị trí là khác nhau.

Từ công thức 7-2-3 có thể thấy trên thực tế thì điện áp đầu ra chỉ phụ thuộc vào tỉ số giá trị điện trở chứ không phụ thuộc vào giá trị tuyệt đối của mỗi điện trở. Điều này tạo thuận lợi cho sản xuất IC. Trên mạch điện một chip của IC, rất khó bảo đảm độ chính xác cao của giá trị điện trở, nhưng dễ hơn là đảm bảo độ chính xác cao của tỉ số các giá trị điện trở đó.

Trong trạng thái động, có thể xem mạng điện trở hình T như một dây truyền dẫn. Vậy các tín hiệu xung sinh ra tại các chuyển mạch có thời gian truyền đến đầu vào bộ khuếch đại thuật toán không như nhau. Do đó sẽ sinh ra các xung nhọn biên độ đáng kể ở đầu ra. Lại thêm sai số thời gian chuyển mạch có thể kéo dài thời gian duy trì xung nhọn. Trong trạng thái động, giá trị tức thời của điện áp tương tự đầu ra có thể lớn hơn nhiều giá trị ổn định, nghĩa là sai số động có thể rất lớn. Giá trị đỉnh của xung nhọn sinh ra trong

trường hợp bit trọng số lớn nhất của tín hiệu số đầu vào từ 0 chuyển sang 1 trước, còn tất cả các bit khác vẫn ở 0. Lúc này giá trị điện áp tức thời đầu ra bằng giá trị điện áp tương tự đầu ra do chuyển đổi DA của tín hiệu số lớn nhất (các bit đều là 1).

Để khử bỏ sai số động ta có thể dùng mạch giữ lấy mẫu ở đầu ra DAC (xem 7-3-2), hơn nữa thời gian lấy mẫu chọn sau khi đã kết thúc quá trình quá độ. Vì lúc lấy mẫu thì xung nhọn đã qua rồi, nên sai số động không ảnh hưởng tới mẫu nữa.

### 3. Tốc độ chuyển đổi

DAC điện trở hình T công tác song song (các tín hiệu số đầu vào được đưa vào song song) nên có tốc độ chuyển đổi cao. Thời gian cần thiết cho một lần chuyển đổi gồm hai đoạn: thời gian trễ truyền đạt của bit tín hiệu vào xa nhất nào đó tới bộ khuếch đại thuật toán và thời gian cần thiết để bộ khuếch đại thuật toán ổn định tín hiệu. Hiện nay IC đơn chip DAC 10 tới 12 bit có thời gian chuyển đổi cỡ vài  $\mu s$ , trong đó thời gian trễ truyền đạt không quá  $1\mu s$ .

### 4. DAC điện trở hình T đảo

Để tránh khỏi xung nhọn xuất hiện trong quá trình động của DAC điện trở hình T, nhờ vậy nâng cao thêm tốc độ chuyển đổi, ta tìm cách duy trì dòng điện chạy qua mỗi nhánh trong mạch là không đổi. Dù tín hiệu số đầu vào là 1 hay là 0 thì dòng điện trong nhánh tương ứng với bit đó cũng không đổi. Vậy là có thể loại trừ cơ bản nguyên nhân tạo ra xung nhọn. *Hình 7-2-5 giới thiệu cách nối mạch bảo đảm mục đích đó. Chúng ta gọi mạch hình 7-2-5 là DAC điện trở hình T đảo.*

Nếu bit bất kì của tín hiệu số đầu vào là 1 thì chuyển mạch tương ứng sẽ nối điện trở nhánh xét vào đầu đảo bộ khuếch đại thuật toán. Nếu bit là 0 thì chuyển mạch sẽ nối điện trở xuống đất. Vậy dù trạng thái tín hiệu đầu vào thế nào thì dòng điện mỗi nhánh đều giữ trước sau không đổi. Dòng điện tổng lấy từ nguồn điện áp chuẩn do đó cũng không đổi. Dòng điện tổng từ nguồn điện áp chuẩn tham khảo do đó cũng không đổi :  $I = V_{ref}/R$

*Xét mạch hình 7-2-5, ta có:*

$$V_0 = -I_i R = -(D_3 2^3 + D_2 2^2 + D_1 2^1 + D_0 2^0) V_{ref} / 2^4 \quad 7-2-6$$

Tức là điện áp tương tự đầu ra tỉ lệ với giá trị tín hiệu số đầu vào. Ưu điểm nổi bật của mạch điện này là tốc độ cao và xung nhọn đầu trong quá trình động thì rất nhỏ. Dòng điện trong các nhánh của mạng điện trở hình T đảo nối trực tiếp vào đầu bộ khuếch đại thuật toán; vì vậy không có sai lệch thời gian truyền đặt giữa chúng, tức là giảm nhỏ sai số trạng thái động.

Trong quá trình chuyển đổi trạng thái, dòng điện trong từng nhánh vẫn không đổi, không cần thời gian thiết lập và ngắt bỏ của dòng điện (các chuyển mạch tương tự nói chung đều công tác theo yêu cầu trước khi ngắt khi chuyển đổi trạng thái)

Vì những nguyên nhân trên đây, mạch DAC điện trở hình T đảo là mạch có tốc độ chuyển đổi DA hạng cao nhất.

Sai số tĩnh của mạch T đảo không có gì khác mạch T trên đây.

### 5. Chuyển mạch tương tự

Các chuyển mạch tương tự được dùng trong DAC hình 7-2-2 và hình 7-2-6 cần có những đặc trưng công tác sau:

1 - Trạng thái của tín hiệu số đầu vào (mức logic 1 hoặc 0) điều khiển trạng thái thông ngắt của chuyển mạch. Đặc điểm này giống như yêu cầu mạch cổng đối với bộ đảo.

2 - Sự thông và ngắt của chuyển mạch không được ảnh hưởng tới giá trị tín hiệu tương tự thông qua nó. Nghĩa là yêu cầu chuyển mạch công tác lí tưởng: điện áp rơi tiếp điểm bằng 0, điện trở nội ngắt mạch bằng  $\infty$ .

Chúng ta biết rằng, trong mạch logic, chuyển mạch logic chỉ có yêu cầu phải thoả mãn là phân biệt mức logic (1 và 0). Yêu cầu này cho phép mức logic 1 và 0 có phạm vi biến đổi như đã quy định. Vậy yêu cầu chuyển mạch tương tự cao hơn nhiều yêu cầu so với chuyển mạch logic. Chúng ta gọi chuyển mạch có đặc điểm hai nói trên là chuyển mạch tương tự.

Chuyển mạch tương tự thường có hai loại: chuyển mạch tương tự lưỡng cực dùng tranzito lưỡng cực, chuyển mạch tương tự đơn cực dùng tranzito trường. Chuyển mạch lưỡng cực có nhiều có nhiều cấu trúc mạch khác nhau, nhưng có thể phân thành hai nhóm cấu trúc mạch chính: bão hoà và không bão hoà. Chuyển mạch tương tự lưỡng cực bão hoà có tốc độ chuyển mạch thấp vì tranzito làm việc ở chế độ bão hoà sâu nên thời gian tồn trữ hạt dẫn

đáng kể, chuyển mạch lưỡng cực không bão hoà có tốc độ cao vì tranzito không bão hoà khi nối thông.

### 7.2.2. Các chỉ tiêu kỹ thuật chủ yếu của bộ biến đổi DAC

#### 1. Độ phân giải

Độ phân giải là tỉ số giữa giá trị cực tiểu đối với giá trị cực đại của điện áp đầu ra, về trị số này tương ứng tỉ số giá trị cực tiểu đối với giá trị cực đại của tín hiệu đầu vào.

Ví dụ đối với DAC 10 bit, độ phân giải là:

$$0000000001/1111111111=1/[(2^{10}) - 1]=1/1023 \approx 0,001$$

Độ phân giải DAC biểu thị bằng số bit của tín hiệu số đầu vào.

#### 2. Độ tuyến tính

Độ tuyến tính của DAC biểu thị bằng sai số phi tuyến là số % của giá trị lệch cực đại khỏi đặc tính vào – ra lí tưởng so với giá trị đầu ra .

#### 3. Độ chính xác chuyển đổi

Độ chính xác chuyển đổi xác định bằng sai số chuyển đổi tĩnh cực đại. Sai số này bao gồm cả sai số phi tuyến, sai số tỉ lệ và sai số trôi ...trong tài liệu kĩ thuật của sản phẩm, đôi khi chỉ cho riêng từng sai số trên mà không cho sai số tổng hợp.

#### 4. Thời gian xác lập dòng, áp đầu ra

Thời gian xác lập ra là thời gian từ khi tín hiệu số được đưa vào đến khi dòng hoặc áp đầu ra đã ổn định. Cũng có tài liệu kĩ thuật của sản phẩm xem thời gian xác lập ra là thời gian cần thiết để mức đầu ra đạt đến số % quy định của mức đầu ra cực đại.

#### 5. Các mức logic cao, thấp và điện trở, điện dung của đầu vào

#### 6. Dải động, điện trở, điện dung của đầu ra

#### 7. Hệ số nhiệt độ

Hệ số nhiệt độ là số biến thiên mức đầu ra khi nhiệt độ tăng 1 độ C trong điều kiện mức đầu ra cực đại.

#### 8. Tỉ số phụ nguồn



DAC chất lượng cao yêu cầu sự ảnh hưởng của biến thiên điện áp nguồn đối với điện áp đầu ra vô cùng nhỏ (Nguồn xét để cung cấp cho chuyển mạch và khuếch đại thuật toán). Tỉ số phụ thuộc nguồn là tỉ số biến thiên mức biến thiên mức điện áp đầu ra với biến thiên điện áp nguồn gây ra nó.

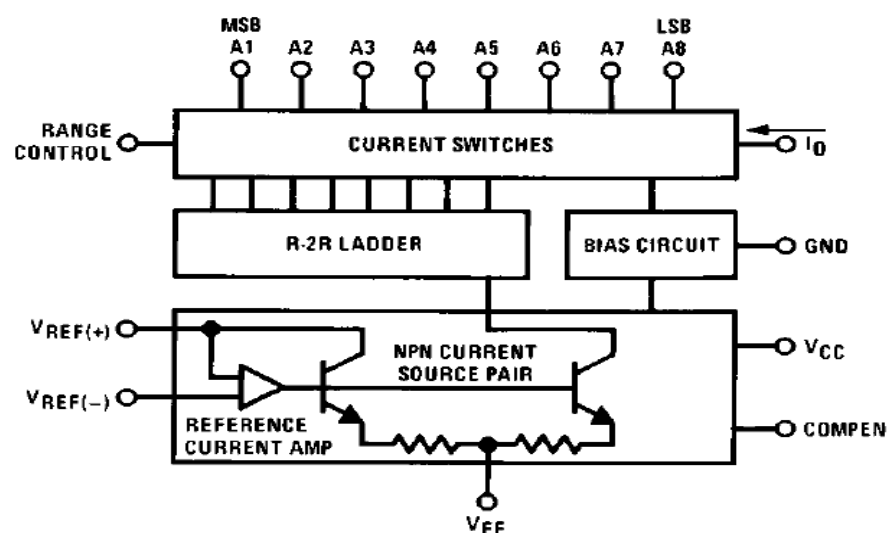
## 9. Công suất tiêu hao

### 7.2.3. Ví dụ IC 1 chip DAC

Bộ biến đổi DAC 0808 là một bộ DAC thông dụng và được sử dụng rộng rãi, bộ DAC 0808 có khả năng giao tiếp được với các mức logic TTL và CMOS có các thông số đặc trưng sau:

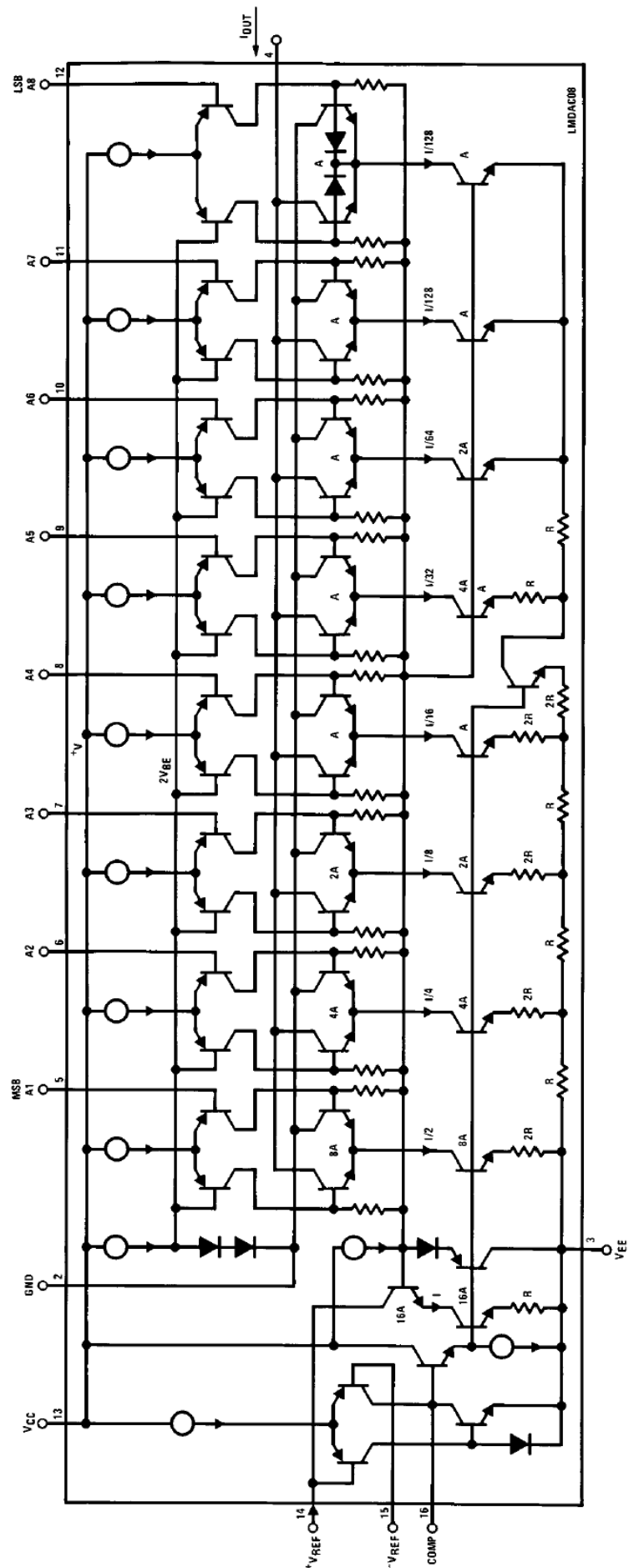
- Độ phân giải 8 bit
- Thời gian chuyển đổi 150ns
- Công suất tiêu thụ 1000mW
- Độ chính xác  $\pm 0.19\%$
- Dòng chuẩn lớn nhất 5mA
- Điện áp  $V_{CC}$  lớn nhất = 18VDC
- Điện áp  $V_{EE}$  lớn nhất = -18VDC
- Dải nhiệt  $-65^{\circ}\text{C} \div 150^{\circ}\text{C}$

Sơ đồ khối của DAC 0808 được chỉ ra trên hình 7-2-6

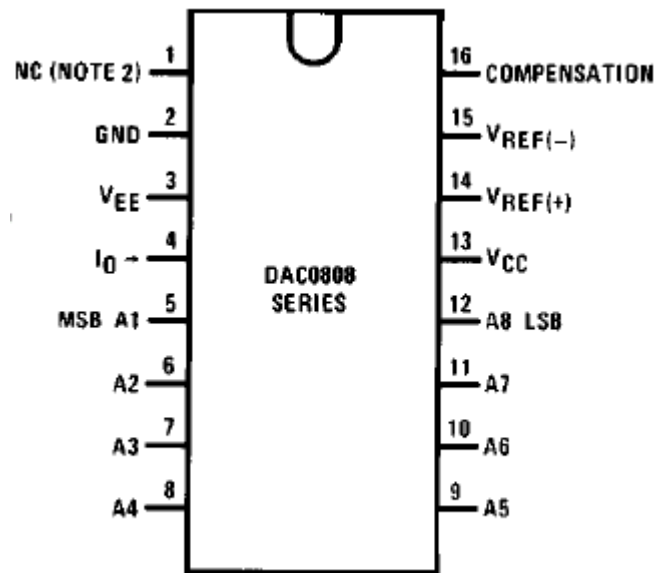


Hình 7-2-6: Sơ đồ khối của DAC 0808

Hình 7-2-7:  
Sơ đồ nguyên lý  
của DAC0808

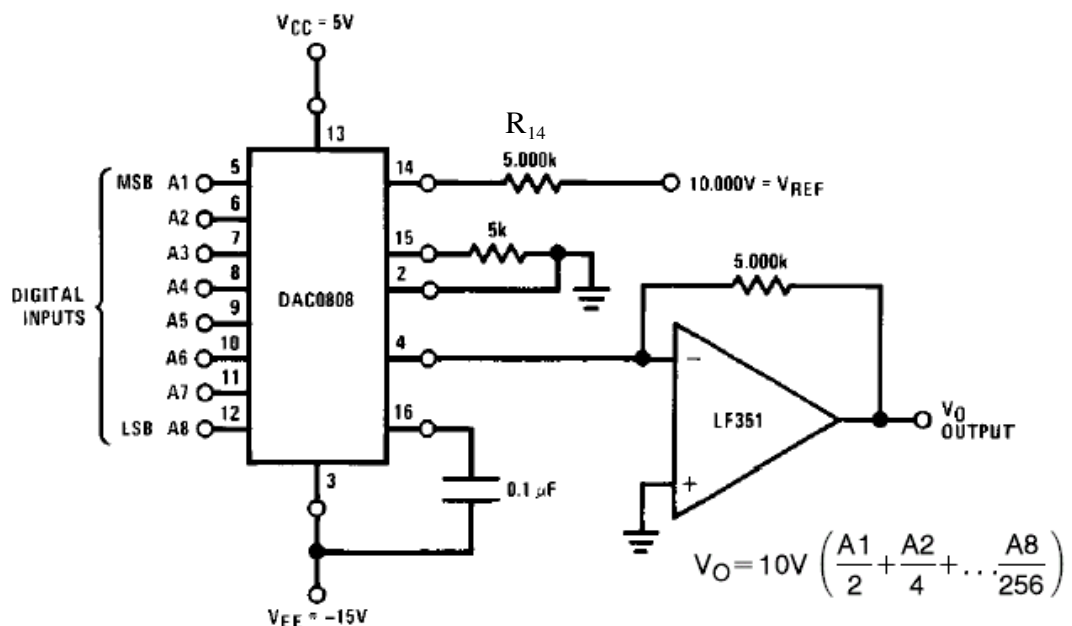


DAC 0808 đóng vỏ DIP (hai hàng chân song song) có 16 chân, xem hình 7-2-8.



Hình 7-2-8 : Sơ đồ chân của DAC 0808

Hình 7-2-9 giới thiệu sơ đồ nguyên lý của IC DAC 0808. Để có bộ DAC hoàn chỉnh cần phối ghép DAC0808 với bộ khuếch đại thuật toán ngoài.



Hình 7-2-9: Sơ đồ nguyên lý của bộ DAC 0808

Trong hình 7-2-9 bộ khuếch đại thuật toán LF351 có biên độ điện áp ra  $V_0 \leq 10V$ , thông thường điện trở  $R_{14}$  được chọn sao cho  $V_{REF}/R_{14} = 2mA$ .

Nếu  $V_{CC} = 5V$ ,  $V_{EE} = -15V$ ,  $V_{REF} = +10V$  và  $R_{14} = 5K$  thì:

- Khi đầu vào là 1111 1111, đầu ra là  $V_0 = 10V$
- Khi đầu vào là 1000 0000, đầu ra là  $V_0 = 5V$
- Khi đầu vào là 0000 0000, đầu ra là  $V_0 = 0V$

### 7.3 Bộ biến đổi Tương tự - Số (ADC)

Bộ biến đổi tương tự - số ADC đóng vai trò quan trọng trong hệ thống xử lý tín hiệu số khi mà các luồng tín hiệu đưa vào hệ số và hệ vi xử lý là các tín hiệu tương tự.

Trong ADC, tín hiệu tương tự đầu vào là liên tục, tín hiệu số mã hoá đầu ra là rời rạc. Sự chuyển đổi AD đòi hỏi phải lấy mẫu đối với tín hiệu tương tự ở đầu vào ở những thời điểm qui định, sau đó chuyển đổi các giá trị mẫu đó thành lượng số đầu ra. Vậy quá trình chuyển đổi AD nói chung thực hiện hai chức năng cơ bản là lượng tử hoá và mã hoá. Lượng tử hoá là gán những giá trị của một tín hiệu tương tự vào vùng các giá trị rời rạc có thể xảy ra trong quá trình lượng tử hoá. Mã hoá là gán những giá trị nhị phân cho từng giá trị rời rạc sinh ra trong quá trình lượng tử hoá. Đối với ADC ta cũng dùng các loại mã số như nhị phân, BCD, bù 2, bù 1.

#### 7.3.1 Các bước chuyển đổi AD và định lý lấy mẫu

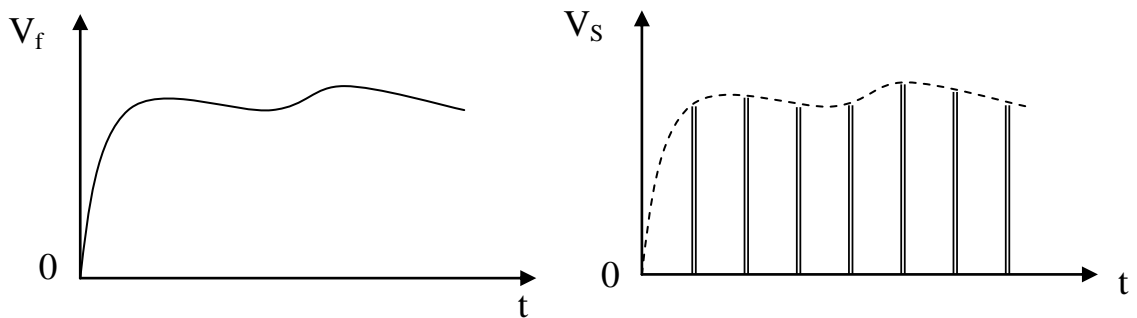
Trong ADC, tín hiệu tương tự đầu vào là liên tục, tín hiệu số mã hoá đầu ra là rời rạc. Sự chuyển đổi AD đòi hỏi phải lấy mẫu đối với tín hiệu tương tự ở đầu vào ở những thời điểm qui định, sau đó chuyển đổi các giá trị mẫu đó thành lượng số đầu ra. Vậy quá trình chuyển đổi AD nói chung có 4 bước: lấy mẫu, nhớ mẫu, lượng tử hoá mã hoá. Như chúng ta sẽ thấy sau đây, các bước trên đây luôn kết hợp nhau trong một quá trình thống nhất. Ví dụ lấy mẫu và nhớ mẫu là công việc liên tục của cùng một mạch điện, lượng tử hoá và mã hoá là công việc đồng thời thực hiện trong quá trình chuyển đổi với khoảng thời gian cần thiết là một phần của thời gian nhớ mẫu.

##### 1. Định lý lấy mẫu

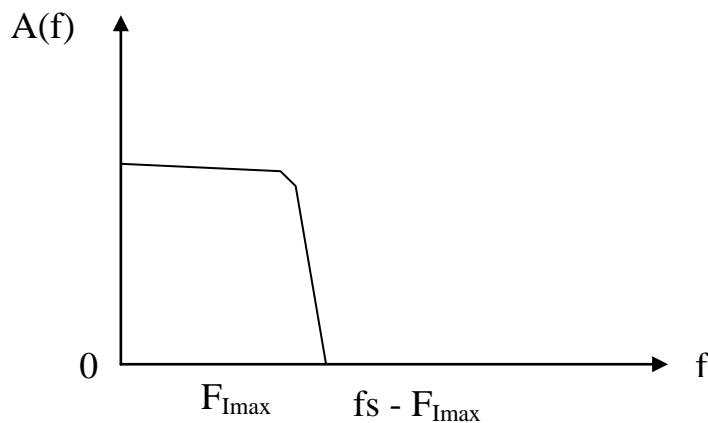
Người ta đã chứng minh rằng đối với tín hiệu tương tự  $V_1$  thì tín hiệu lấy mẫu  $V_s$  sau quá trình lấy mẫu có thể khôi phục trở lại  $V_1$  một cách trung thực, nếu điều kiện sau đây được thỏa mãn  $f_s \geq 2F_{\text{Imax}}$  (7-3-1)

$f_s$  là tần số lấy mẫu,  $F_{\text{Imax}}$  là giới hạn trên một giải tín hiệu tương tự (7-3-1) là nội dung định lý lấy mẫu.

Nếu (7-3-1) được thoả mãn, có thể dùng bộ lọc thông thấp để khôi phục  $V_I$  từ  $V_s$ . Hình 7-3-2 biểu thị đặc tính tần số của bộ lọc thông thấp xét: hệ số truyền đạt điện áp  $|A(f)|$  của bộ lọc trong phạm vi giải tần  $< F_{\text{Imax}}$  cần giữ không đổi còn trong phạm vi giải tần  $f_s - F_{\text{Imax}}$  phải nhanh chóng giảm xuống 0. Vậy định lý lấy mẫu quy định cho chúng ta giới hạn dưới của tần số trong chuyển đổi AD.



Hình 7-3-1: Lấy mẫu tín hiệu tương tự



Hình 7-3-2: Đặc tính tần số của bộ lọc khôi phục tín hiệu

Vì mỗi lần chuyển đổi điện áp lấy mẫu thành tín hiệu số tương ứng đều cần một thời gian nhất định, nên phải nhớ mẫu một khoảng thời gian cần thiết sau mỗi lần lấy mẫu. Điện áp tương tự đầu vào được thực hiện chuyển đổi AD, trên thực tế giá trị  $V_I$  đại diện, giá trị này là kết quả của mỗi lần lấy mẫu.

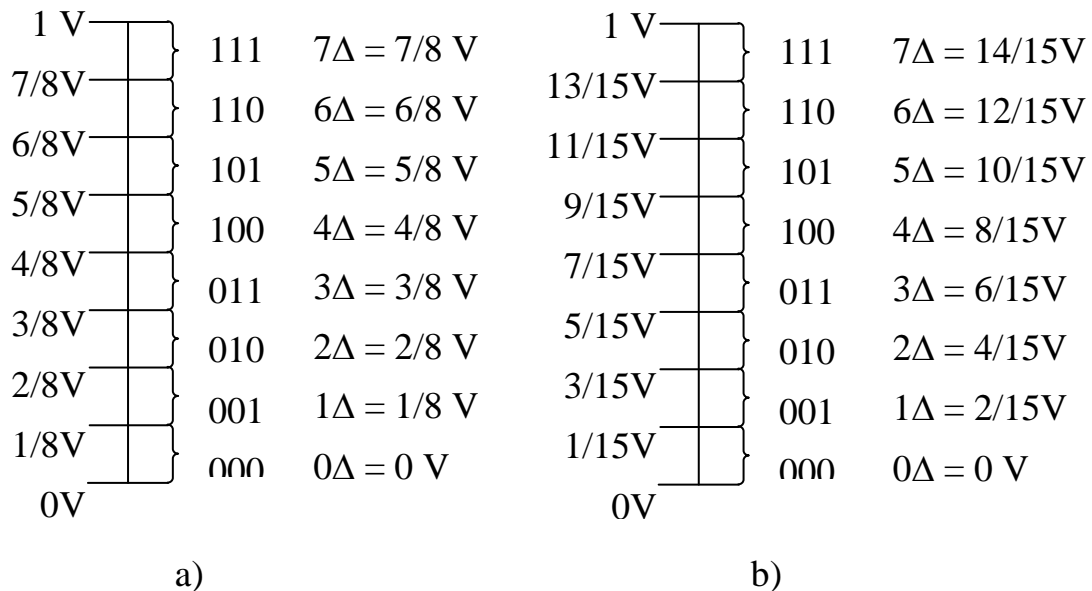
## 2. Lượng tử hoá và mã hoá

Ta đã biết, tín hiệu số không những là rời rạc trong thời gian, mà còn không liên tục trong biến đổi giá trị. Một giá trị bất kì của tín hiệu số đều

phải biểu thị bằng bội số nguyên lần giá trị đơn vị nào đó, giá trị này là nhỏ nhất được chọn. Nghĩa là, nếu dùng tín hiệu số biểu thị điện áp lấy mẫu, thì tất phải bắt điện áp lấy mẫu hoá thân thành bội số nguyên là giá trị đơn vị. Quá trình này gọi là lượng tử hoá. Đơn vị được chọn theo qui định này gọi là đơn vị lượng tử, kí hiệu là  $\Delta$ . Rõ ràng, giá trị bit 1 của LSB tín hiệu số bằng  $\Delta$ . Việc dùng mã nhị phân biểu thị giá trị tín hiệu số là mã hoá. Mã nhị phân có được sau quá trình trên chính là tín hiệu đầu ra của chuyển đổi AD.

Tín hiệu tương tự đã liên tục thì không nhất thiết phải là bội số nguyên lần của  $\Delta$ . Do đó ta không tránh khỏi sai số lượng tử hoá. Tồn tại những cách khác nhau phân chia các mức lượng tử dẫn đến những sai số lượng tử khác nhau.

Giả sử chuyển đổi tín hiệu điện áp tương tự 0÷1V thành tín hiệu số nhị phân 3 bit. Nếu chọn  $\Delta = V \cdot 1/8$ , đồng thời quy định rằng điện áp tương tự trong phạm vi 0÷ $V/8$  xem như là  $0 \cdot \Delta$ , thì tín hiệu tương ứng 000. Tương tự điện áp tương tự  $(1/8)V$  ÷  $(2/8)V$  là  $1 \cdot \Delta$  thì tương ứng với 001 v.v....



Hình 7-3-3: Hai phương pháp phân chia mức lượng tử

Hình 7-3-3b chỉ ra phương pháp phân chia mức lượng tử có thể giảm nhỏ hơn sai số lượng tử. Chọn  $\Delta = 2/15 V$

Quy định điện áp tương tự 0 ÷ 1/15 V tương ứng 000 ( $0 \div \Delta/2$ )

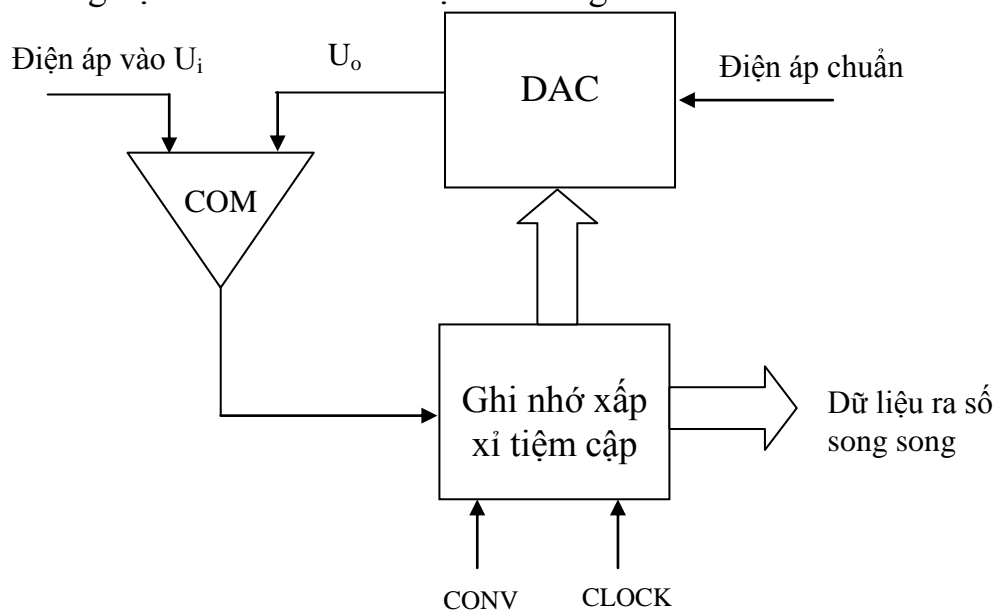
Quy định điện áp tương tự 1/15 V ÷ 3/15 V tương ứng 001 v.v...

Sai số lượng tử cực đại của phương pháp này là  $\Delta/2 = 1/15 \text{ V}$

Chúng ta thấy sai số lượng tử của phương pháp này giảm nhỏ là nhờ cách phân chia mức đặc biệt, trong đó, giá trị quy định của điện áp tương tự có cùng giá trị lượng tử. (điện áp tương tự trong phạm vi  $1/15 \text{ V} \div 3/15 \text{ V}$  đều có cùng giá trị lượng tử 001, mà 001 biểu diễn giá trị đúng  $2/15 \text{ V}$ , nằm đúng giữa phạm vi trên)

### 7.3.2. Bộ biến đổi ADC xấp xỉ tiệm cận ( Successive- approximation ADC)

Hình 7-3-4 là sơ đồ khối ADC xấp xỉ tiệm cận. Trong sơ đồ này thì có các khối sau: bộ so sánh (COM), DAC, điện áp chuẩn, bộ nhớ xấp xỉ tiệm cận, logic điều khiển, tín hiệu đồng hồ (clock)v.v... Trước khi thực hiện chuyển AD, bộ nhớ phải bị xoá về không. Bắt đầu chuyển đổi, xung đồng hồ lập bit MSB trong bộ nhớ mức 1, số liệu ra của bộ nhớ là 100..0. Tín hiệu số này được DAC chuyển đổi thành điện áp tương tự ứng  $U_0$ . COM so sánh  $U_i$  và  $U_0$ , nếu  $U_0 > U_i$  tín hiệu số quá lớn thì bit MSB bị xoá về không,  $U_0 < U_i$  tín hiệu số vẫn còn bé thì bit MSB duy trì giá trị 1. Tiếp theo, cùng một phương pháp như trên, xung đồng hồ thiết lập bit có trọng số bé hơn mức 1, sau khi so sánh, logic mạch xác định giá trị 1 này có duy trì hay không. Cứ thế tiếp tục mãi đến bit LSB thì xong. Sau quá trình so sánh tất cả các bit, dữ liệu trong bộ nhớ chính là tín hiệu số mong muốn.

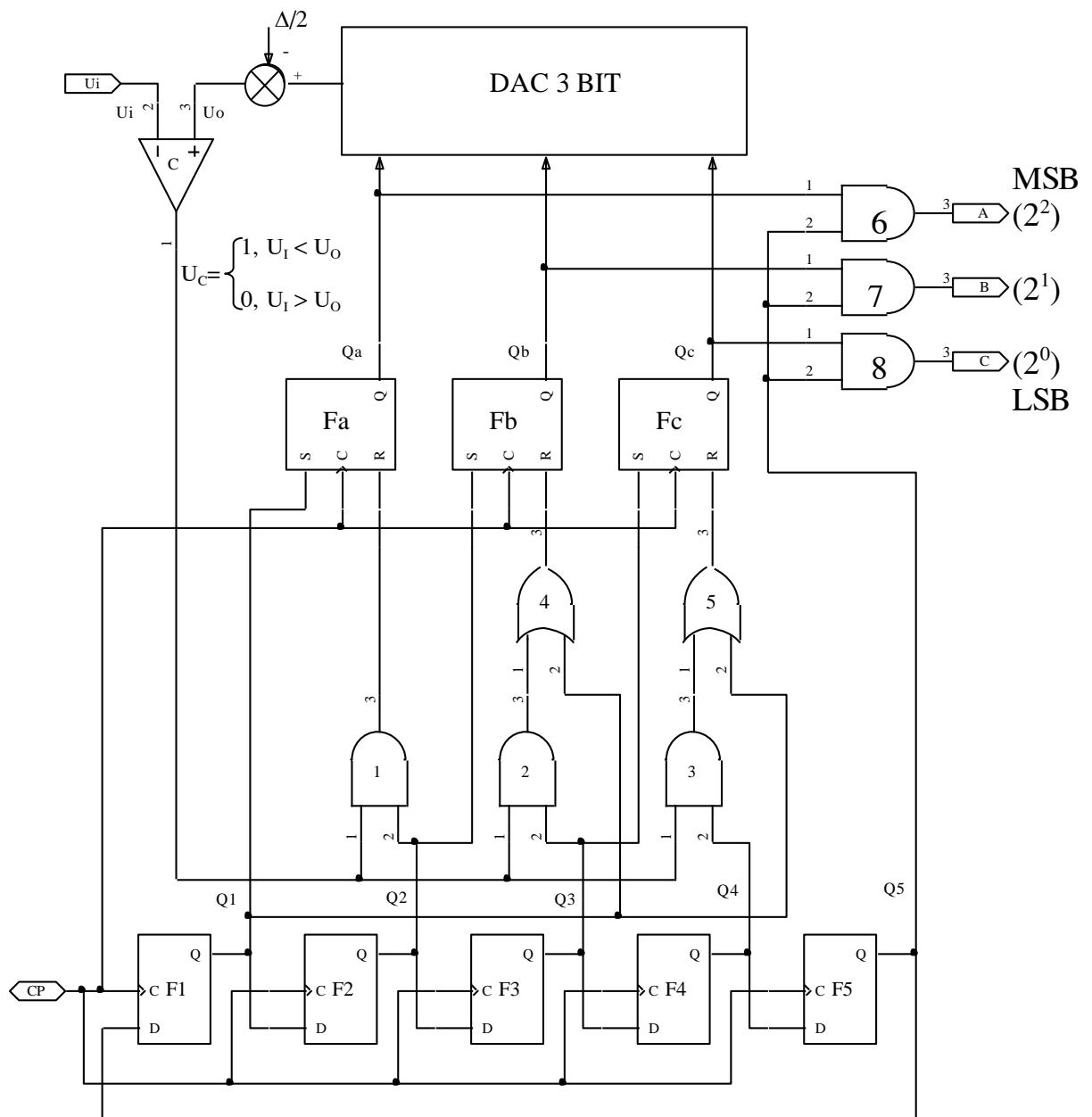


Tín hiệu điều khiển chuyển đổi

**Hình 7-3-4: ADC xấp xỉ tiệm cận**

Quá trình trên tương tự như quá trình cân một vật bằng cân bàn. Các quả cân được chọn lựa từ lớn đến bé, mỗi lần chọn lựa một quả cân bằng của lần chọn trước.

Xét mạch điện ADC xấp xỉ tiệm cận 3 bit hình 7-3-5. Fa, Fb, Fc là bộ nhớ xấp xỉ 3 bit và các cổng AND, OR là mạch logic điều khiển. F1÷F5 được đánh dấu thành bộ ghi dịch vòng. Trước khi bắt đầu chuyển đổi thiết lập  $Q_1 = Q_2 = Q_3 = Q_4 = 0, Q_5 = 1$ . Xung đồng hồ CP thứ nhất đến :  $Q_1=1, Q_2 = Q_3 = Q_4 = Q_5 = 0, Q_A = 1, Q_B = Q_C = 0$ . Tín hiệu số đưa đến DAC là 100, đầu ra là điện áp tương tự  $U_0$  tương ứng. Bộ so sánh C so sánh  $U_0$  và  $U_I$ , nếu  $U_I < U_0$  thì  $U_c = 1$ , nếu  $U_I \geq U_0$  thì  $U_c = 0$ .



Hình 7-3-5 : Mạch điện ADC xấp xỉ tiệm cận 3 bit



Trong nhịp đồng hồ thứ hai: bộ ghi dịch vòng dịch phải 1 bit, nghĩa là  $Q_2=1$ ,  $Q_2=Q_3=Q_4=Q_5=0$ . Nếu  $U_C=1$  thì cổng AND1 thông, Reset  $F_A$ ,  $Q_A=0$  (xoá MSB). Nếu  $U_C=1$  thì cổng AND1 vẫn ngắt,  $F_A$  duy trì trạng thái  $Q_A=1$ . Đồng thời  $Q_2=1$  thiết lập  $F_B=1$ ,  $Q_B=1$ .

Đến nhịp đồng hồ thứ ba: bộ ghi dịch vòng dịch phải 1 bit nữa, nghĩa là  $Q_3=1$ ,  $Q_2=Q_3=Q_4=Q_5=0$ . Kết quả  $F_C=1$ ,  $Q_C=1$ . Đồng thời cổng AND2 thông, tùy theo giá trị  $U_C$  đầu ra bộ so sánh C mà  $Q_B=0$  hay  $Q_B=1$ .

Ở nhịp đồng hồ thứ tư:  $Q_4=1$ ,  $Q_1=Q_2=Q_3=Q_5=0$  kết quả so sánh  $U_C$  quyết định  $Q_C=0$  hay  $Q_C=1$ .

Nhịp đồng hồ tiếp theo  $Q_5=1$ ,  $Q_1=Q_2=Q_3=Q_4=0$  kết quả chuyển đổi thể hiện ở trạng thái  $F_A$ ,  $F_B$ ,  $F_C$  được đưa ra đầu ra thông qua các cổng AND 6,7,8.

Bộ trừ  $\otimes$  trên hình 7-3-5 làm giảm giá trị đầu ra DAC một lượng  $\Delta/2$ , với mục đích giảm sai số lượng tử. Hình 7-3-5 cho ta biết rằng, để sai số lượng tử không lớn hơn  $\Delta/2$  thì mức so sánh thứ nhất không phải là  $\Delta$ , mà là  $\Delta/2$ , còn đối với các mức so sánh khác phải chênh nhau  $\Delta$ , nghĩa là cũng phải giảm đi  $\Delta/2$ . Nhờ bộ trừ mà tất cả các mức tín hiệu đầu ra DAC đều giảm bớt  $\Delta/2$  trước khi đem so sánh với  $U_i$ .

Trong ví dụ này ta thấy một lần chuyển đổi AD cần 5 chu kỳ đồng hồ CP. Nếu số bit tăng thêm thì thời gian chuyển đổi cũng tăng lên theo. Hiện nay ADC xấp xỉ tiệm cận được ứng dụng rộng rãi.

### 7.3.4. Các chỉ tiêu kỹ thuật của ADC

#### 1. Độ phân giải

Độ phân giải của ADC biểu thị bằng số bit của tín hiệu số đầu ra. Số bit càng nhiều thì sai số lượng tử càng nhỏ, độ chính xác càng cao.

#### 2. Dải động, điện trở đầu vào

#### 3. Mức logic của tín hiệu số đầu ra và khả năng chịu tải

#### 4. Độ chính xác tương đối

Nếu lí tưởng hoá thì tất cả các điểm chuyển đổi phải nằm trên một đường thẳng. Độ chính xác tương đối là sai số của các điểm chuyển đổi thực tế so

với đặc tuyến chuyển đổi lí tưởng. Ngoài ra, nói chung yêu cầu ADC không bị mất bit trong toàn bộ phạm vi công tác.

### 5. Tốc độ chuyển đổi

Tốc độ chuyển đổi xác định bởi thời gian cần thiết hoàn thành 1 lần chuyển đổi AD. Thời gian này tính từ khi xuất hiện tín hiệu điều khiển chuyển đổi đến khi tín hiệu số đầu ra đã ổn định.

### 6. Hệ số nhiệt độ

Hệ số nhiệt độ là biến thiên tương đối tín hiệu số đầu ra khi nhiệt độ biến đổi  $1^{\circ}\text{C}$  trong phạm vi nhiệt độ cho phép với điều kiện mức tương tự đầu vào cũng không thay đổi (cũng có thể dùng biến thiên tuyệt đối)

### 7. Tỉ số phụ thuộc nguồn

Giả sử điện áp tương tự đầu vào không đổi, nếu nguồn cung cấp cho ADC biến thiên mà ảnh hưởng tới tín hiệu số đầu ra càng lớn thì tỉ số phụ thuộc nguồn càng lớn. Vậy dùng biến thiên tuyệt đối đầu ra biểu thị.

### 8. Công suất tiêu hao

#### 7.3.5 Bộ biến đổi ADC 8 bit 0809

Bộ biến đổi ADC 0809 là ADC thông dụng nhất được sử dụng rất rộng rãi, có 8 đầu vào tương tự và đầu ra 8 bit số, chuyển đổi theo phương pháp xấp xỉ liên tiếp. Các đặc trưng cơ bản của ADC 0809 là:

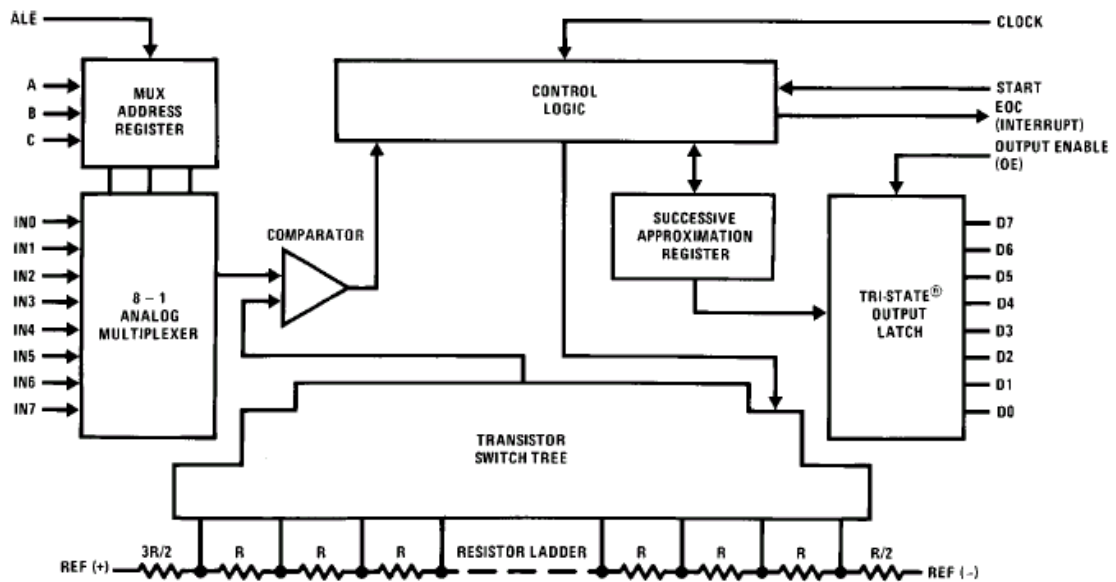
Đầu ra có bộ đệm ba trạng thái để ghép trực tiếp vào kênh dữ liệu của hệ vi xử lí.

Giải tín hiệu lỗi vào tương tự 5V khi nguồn nuôi là +5V. Có thể mở rộng thang đo bằng các giải pháp kĩ thuật cho từng mạch cụ thể.

- Không đòi hỏi điều chỉnh “0”
- Thời gian biến đổi  $100\mu\text{s}$
- Sai số tổng cộng  $\pm 1/2 \text{ LSB}$
- Sử dụng nguồn nuôi đơn +5V, hiệu suất cao.
- Đảm bảo sai số tuyến tính trong dải nhiệt độ từ  $-40$  đến  $+85^{\circ}\text{C}$ .

## 1. Sơ đồ chức năng của ADC 0809

Sơ đồ khối bộ chuyển đổi ADC 0809 trên hình 7-3-6.



Hình 7-3-6: Sơ đồ khối của bộ chuyển đổi ADC 0809

## 2. Ghép tín hiệu vào ADC 0809

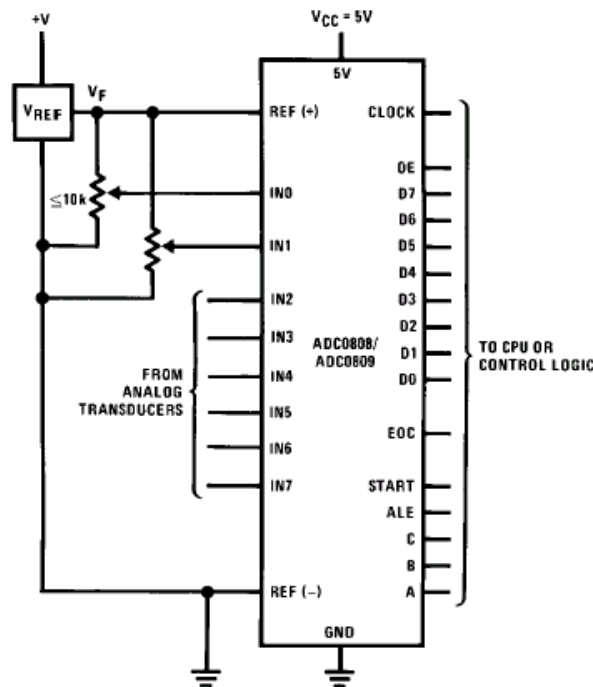
Trong chu trình chuyển đổi dòng đầu vào, tín hiệu giữ nhịp 500 KHz dùng cho bộ ADC 0809 được tạo ra ở bên ngoài và được đưa đến chân Clock (bộ chuyển đổi ADC 0809 sử dụng tần số Clock có thể trong dải 200KHz)1MHz).

### ▪ Tín hiệu đơn cực đến đầu vào của ADC 0809

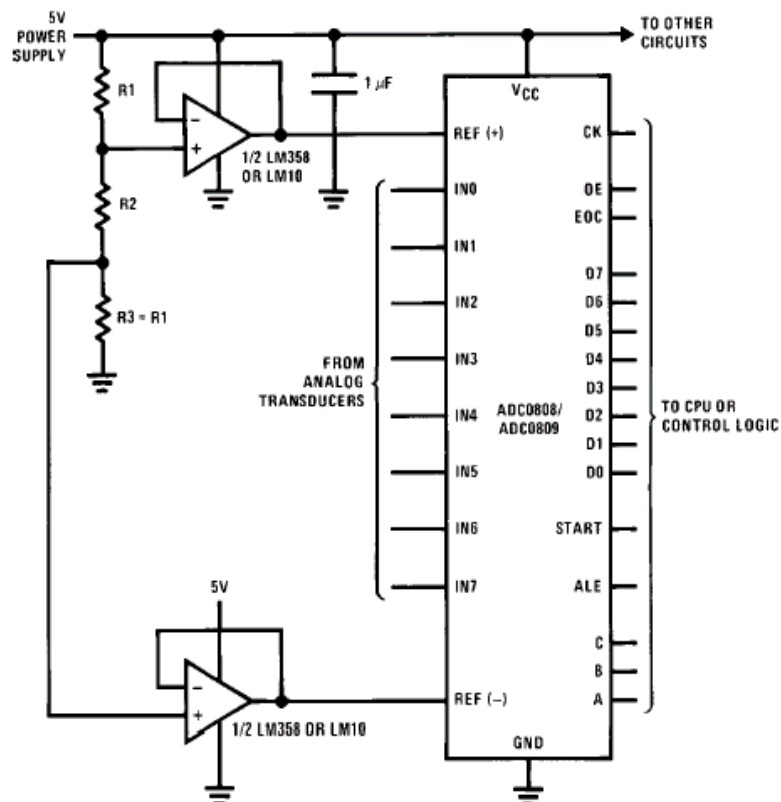
Khi tín hiệu đầu vào của bộ chuyển đổi ADC là đơn cực (một cực tính) sơ đồ nguyên lí khi đó được mắc như hình 7-3-7. Trong trường hợp này thang đo tương ứng từ 5V cho các đầu vào (tức là các đại lượng cần đo qua các bộ cảm biến phải đưa về dải từ 0 đến 5V). Tương ứng với mức 0V ở đầu vào là tổ hợp nhị phân 0000 0000 ở đầu ra. Còn trong tương ứng với mức 5V có tổ hợp 1111 1111 ở đầu ra.

### ▪ Ghép tín hiệu lưỡng cực đến đầu vào của ADC0809

Trong thực tế khảo sát các đại cần biến đổi thường là các tín hiệu xoay chiều (tín hiệu vào có hai cực tính). Với trường hợp này cần mở rộng thang đo thích hợp với tín hiệu. Sơ đồ nguyên lí của mạch thể hiện dạng này như hình 7-3-8.



Hình 7-3-7: Sơ đồ hoạt động của ADC 0809 khi đầu vào là đơn cực



Hình 7-3-8: Sơ đồ hoạt động của ADC 0809 khi đầu vào là lưỡng cực

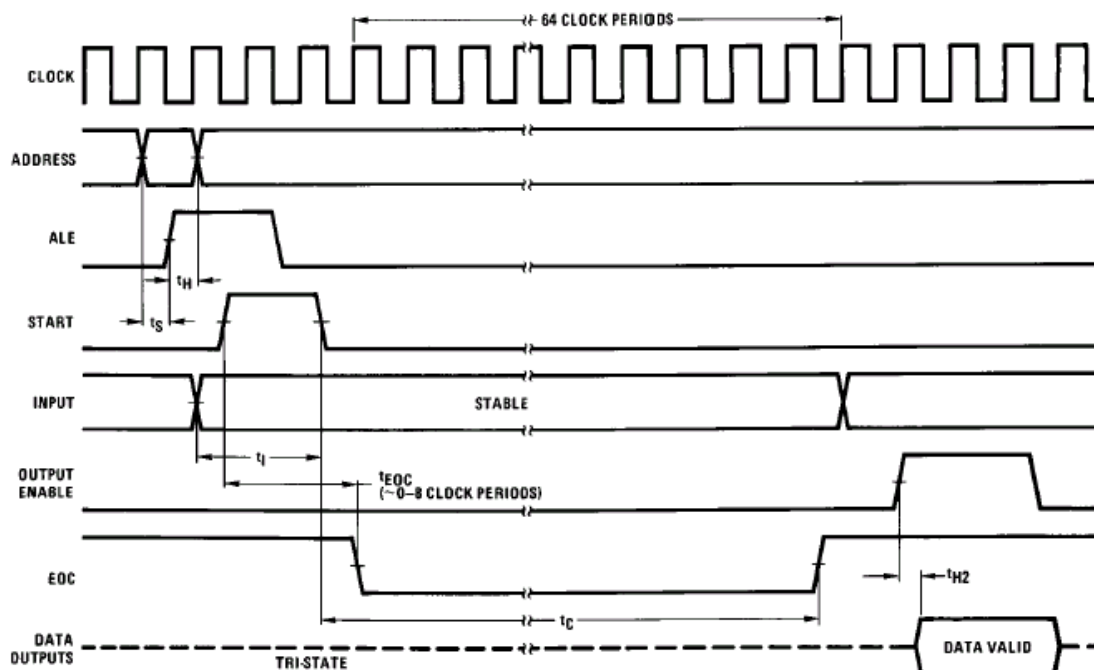
Để mở rộng thang đo, điện áp so sánh được đưa ra qua tầng lặp lại điện áp để đưa đến chân 12 (chân ref(+)) của ADC 0809. Các kênh lối vào tương tự được dẫn đến các chân  $V_{IN0}$  đến  $V_{IN7}$ .

Nếu chọn  $V_{\text{ref}(+)} = +2.5\text{V}$  thì từ đầu ra của bộ cảm biến phải đưa về dải 0 đến 2,5V (cho phép đo trong dải  $\pm 2,5\text{V}$ ). Khi đó cận trên thang đo (+2,49V) tương ứng với tổ hợp mã nhị phân ở đầu ra là 1111 1111. Cận dưới thang đo (-2,49V) ứng với tổ hợp mã nhị phân ở đầu ra là 0000 0000. Còn mức “0” của thang đo ứng với tổ hợp: 0111 1111.

#### ▪ Đồ thị của thời gian của ADC 0809

Các bit địa chỉ ở lối vào A,B,C từ bộ giải mã địa chỉ sẽ chốt và xác định kênh đầu vào nào được lựa chọn. Tương ứng với kênh đầu vào xác định ở trên khi có một xung dương đặt vào chân số 6 (Start) với độ rộng tối thiểu ( $t = 200\text{ns}$ ). Sau thời gian  $t_{\text{EOC}}$  (tính sườn trước xung start) bộ chuyển đổi bắt đầu thực hiện việc chuyển đổi và trong suốt thời gian này chân tín hiệu ra EOC luôn ở mức thấp. Đồng thời đầu ra ba trạng thái của ADC 0809 bị thả nổi (ở trạng thái trở kháng cao).

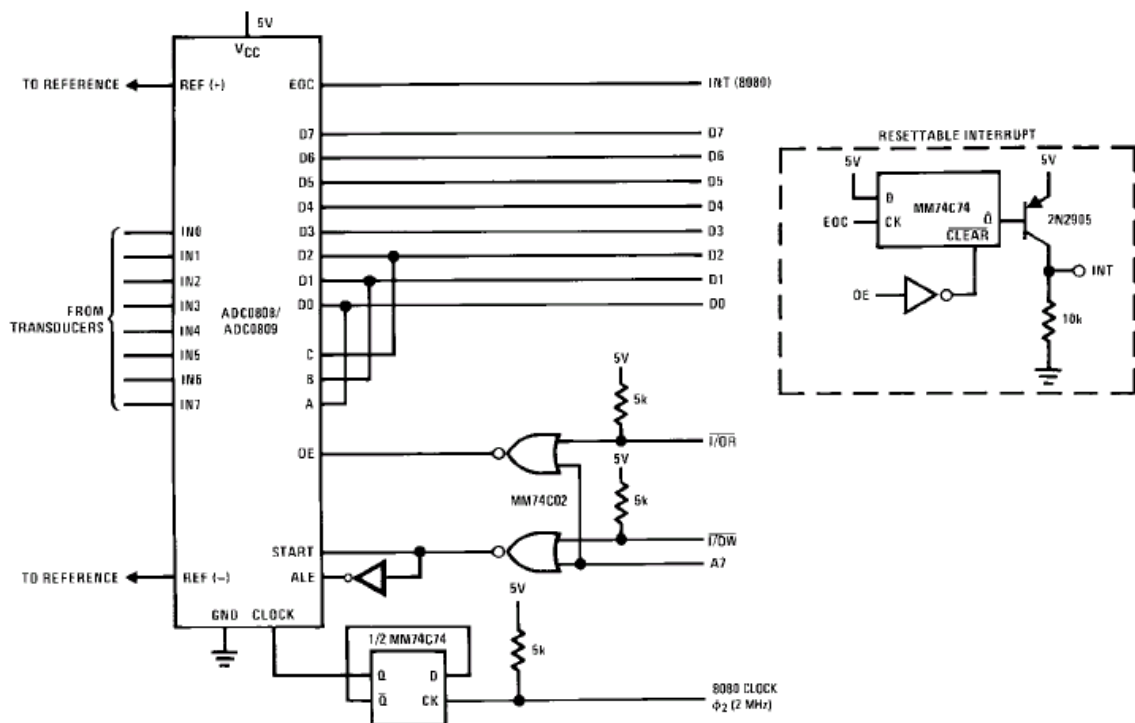
Sau thời gian  $t_{\text{C}} = 100\mu\text{s}$  ADC 0809 thực hiện chuyển đổi xong. Dữ liệu đầu vào được đưa đến bộ đếm đầu ra ba trạng thái, đồng thời chân tín hiệu EOC chuyển lên mức cao báo hiệu cho hệ vi xử lý biết để đọc kết quả vào. Hoạt động của mạch được thể hiện trên đồ thị hình 7-3-9.



Hình 7-3-9: Đồ thị thời gian của ADC 0809

### 3. Ghép ADC 0809 với hệ vi xử lý.

Ghép ADC 0809 với hệ vi xử lý được thể hiện trên hình 7-3-10. Các kênh vào Analog được nối với đầu vào tương tự của ADC (có 8 kênh vào Analog). Mỗi kênh có một địa chỉ riêng do tổ hợp 3 bit địa chỉ ABC quy định. Các tín hiệu địa chỉ chọn kênh này thường được nối với kênh địa chỉ của vi xử lý hoặc cũng có thể nối với kênh dữ liệu. Như trên hình 7-3-10 các dây tín hiệu địa chỉ ABC được nối với 3 dây kênh dữ liệu D0, D1, D2 của hệ vi xử lý. Bit địa chỉ A7 của hệ vi xử lý dùng để chọn chip cho ADC (cho phép ADC làm giao tiếp với hệ vi xử lý), tín hiệu A7 kết hợp với tín hiệu I/O $\overline{W}$  thông qua cổng logic NOR để khởi động ADC (Start) và đồng thời chốt địa chỉ (ALE) của kênh được chọn. Tín hiệu A7 cũng đưa vào đầu vào của cổng logic NOR thứ 2, cùng với tín hiệu I/OR để tạo tín hiệu cho phép dữ liệu ra (chân OE) khi ADC đã biến đổi xong.



Hình 7-3-10: Sơ đồ ghép ADC 0809 với hệ vi xử lý

A7	A6	A5	A4	A3	A2	A1	A0	D2	D1	D0	Output Port Description
1	1	1	1	1	1	1	0	X	X	X	Spare Port
1	1	1	1	1	1	0	1	X	X	X	Spare Port
1	1	1	1	1	0	1	1	X	X	X	Spare Port
1	1	1	1	0	1	1	1	X	X	X	Spare Port
1	1	1	0	1	1	1	1	X	X	X	Spare Port
1	1	0	1	1	1	1	1	X	X	X	Spare Port
1	0	1	1	1	1	1	1	X	X	X	Spare Port
0	1	1	1	1	1	1	1	0	0	0	Channel 0 Port
0	1	1	1	1	1	1	1	0	0	1	Channel 1 Port
0	1	1	1	1	1	1	1	0	1	0	Channel 2 Port
0	1	1	1	1	1	1	1	0	1	1	Channel 3 Port
0	1	1	1	1	1	1	1	1	0	0	Channel 4 Port
0	1	1	1	1	1	1	1	1	0	1	Channel 5 Port
0	1	1	1	1	1	1	1	1	1	0	Channel 6 Port
0	1	1	1	1	1	1	1	1	1	1	Channel 7 Port

Hình 7-3-11: Bảng trạng thái chọn kênh của ADC 0809 với hệ vi xử lý