



HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÀI GIẢNG MÔN ĐIỆN TỬ SỐ

Giảng viên:

KS. Nguyễn Trung Hiếu

Điện thoại/E-mail:

0916566268; trunghieutq@gmail.com

Bộ môn:

Kỹ thuật điện tử - Khoa KTĐT1

Học kỳ/Năm biên soạn: Học kỳ 1/2009-2010



- www.ptit.edu.vn **Bài giảng Điện tử số**
KS. Nguyễn Trung Hiếu, Bộ môn KTĐT 2



Nội dung

- **Chương 1: Hệ đếm**

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

Chương 3: Cổng logic

Chương 4: Mạch logic tổ hợp

Chương 5: Mạch logic tuần tự

Chương 6: Mạch phát xung và tạo dạng xung

Chương 7: Bộ nhớ bán dẫn

Hệ đếm



Nội dung

- Khái niệm chung
- Biểu diễn số

Chuyển đổi giữa các hệ đếm

Số nhị phân có dấu

Dấu phẩy động

Biểu diễn số (1)

■ Nguyên tắc chung

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

■ Tên gọi, số ký hiệu và cơ số của một vài hệ đếm thông dụng

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

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

Biểu diễn số (2)

- **Biểu diễn số tổng quát:**

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

- **Trong một số trường hợp, ta phải thêm chỉ số để tránh nhầm lẫn giữa biểu diễn của các hệ.**

Ví dụ: 36_{10} , 36_8 , 36_{16}

Hệ thập phân (1)

- **Biểu diễn tổng quát:**

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

Trong đó:

- N_{10} : biểu diễn bất kì theo hệ 10,
 - d : các hệ số nhân (ký hiệu bất kì của hệ),
 - n : số chữ số ở phần nguyên,
 - m : số chữ số ở phần phân số.
- *Giá trị biểu diễn của một số trong hệ thập phân sẽ bằng tổng các tích của ký hiệu (có trong biểu diễn) với trọng số tương ứng*
 - **Ví dụ: 1265.34 là biểu diễn số trong hệ thập phân:**

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

Hệ thập phân (2)

- **Ưu điểm của hệ thập phân:**

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

- **Nhược điểm:**

- Do có nhiều ký hiệu nên việc thể hiện bằng thiết bị kỹ thuật sẽ khó khăn và phức tạp.

Hệ nhị phân (1)

- **Biểu diễn tổng quát:**

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

Trong đó:

- N_2 : biểu diễn bất kì theo hệ 2,
 - b : là hệ số nhân lấy các giá trị 0 hoặc 1,
 - n : số chữ số ở phần nguyên,
 - m : số chữ số ở phần phân số.
- **Hệ nhị phân (Binary number system) còn gọi là hệ cơ số hai, gồm chỉ hai ký hiệu 0 và 1, cơ số của hệ là 2, trọng số của hệ là 2^n .**
 - **Ví dụ: 1010.01_2 là biểu diễn số trong hệ nhị phân.**

$$1010.01_2 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

■ *Ưu điểm:*

- Chỉ có hai ký hiệu nên rất dễ thể hiện bằng các thiết bị cơ, điện.
- Hệ nhị phân được xem là ngôn ngữ của các mạch logic, các thiết bị tính toán hiện đại - ngôn ngữ máy.

■ *Nhược điểm:*

- Biểu diễn dài, mất nhiều thời gian viết, đọc.

■ *Các phép tính:*

- Phép cộng:

$$0 + 0 = 0, 1 + 0 = 1, 1 + 1 = 10$$

- Phép trừ:

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

- Phép nhân: (thực hiện giống hệ thập phân)

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

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

- Phép chia: Tương tự phép chia 2 số thập phân

Hệ bát phân (1)

- **Biểu diễn tổng quát:**

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

Trong đó:

- N_8 : biểu diễn bất kì theo hệ 8,
 - O : các hệ số nhân (ký hiệu bất kì của hệ),
 - n : số chữ số ở phần nguyên,
 - m : số chữ số ở phần phân số.
-
- *Hệ này gồm 8 ký hiệu : 0, 1, 2, 3, 4, 5, 6 và 7. Cơ số của hệ là 8. Việc lựa chọn cơ số 8 là xuất phát từ chỗ $8 = 2^3$. Do đó, mỗi chữ số bát phân có thể thay thế cho 3 bit nhị phân.*
 - Ví dụ: 1265.34_8 là biểu diễn số trong bát phân.

■ Phép cộng

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

$$\begin{array}{r}
 253 \\
 + 126 \\
 \hline
 401
 \end{array}
 \begin{array}{l}
 \text{don vi : } 3 + 6 = 9 = 1 + 8 \text{ (viet 1 nho 1 len hang chuc)} \\
 \text{chuc : } 5 + 1 + 2 = 8 = 0 + 8 \text{ (viet 0 nho 1 len hang tram)} \\
 \text{tram : } 2 + 1 + 1 = 4 \text{ (1 la nho tu hang chuc)}
 \end{array}$$

■ Phép trừ

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

$$\begin{array}{r}
 253 \\
 - 126 \\
 \hline
 125
 \end{array}
 \begin{array}{l}
 \text{don vi : } 3 < 6 \rightarrow 8 + 3 - 6 = 5 \text{ (no 1 hang chuc)} \\
 \text{chuc : } 5 - 1 - 2 = 2 \text{ (1 la cho hang don vi vay)}
 \end{array}$$

- **Chú ý:** Các phép tính trong hệ bát phân ít được sử dụng.

Hệ thập lục phân (1)

- **Biểu diễn tổng quát:**

$$\begin{aligned} N_{16} &= H_{n-1} \times 16^{n-1} + \dots + H_0 \times 16^0 + H_{-1} \times 16^{-1} + \dots + H_{-m} \times 16^{-m} \\ &= \sum_{i=n-1}^{-m} H_i \times 16^i \end{aligned}$$

Trong đó:

- N_{16} : biểu diễn bất kì theo hệ 16,
 - d : các hệ số nhân (ký hiệu bất kì của hệ),
 - n : số chữ số ở phần nguyên,
 - m : số chữ số ở phần phân số.
- **Hệ thập lục phân (hay hệ Hexadecimal, hệ cơ số 16).**
 - Hệ gồm 16 ký hiệu là 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
 - Trong đó, $A = 10_{10}$, $B = 11_{10}$, $C = 12_{10}$, $D = 13_{10}$, $E = 14_{10}$, $F = 15_{10}$.
 - **Ví dụ: 1FFA là biểu diễn số trong hệ thập lục phân**

Hệ thập lục phân (2)

■ Phép cộng

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

$$\begin{array}{r} 1 \quad 6 \quad 9 \\ + \quad 2 \quad 5 \quad 8 \\ \hline 3 \quad C \quad 1 \end{array}$$

■ Phép trừ

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

$$\begin{array}{r} 2 \quad 5 \quad 8 \\ - \quad 1 \quad 6 \quad 9 \\ \hline 0 \quad E \quad F \end{array}$$

■ Phép nhân

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



Nội dung

Biểu diễn số

- **Chuyển đổi cơ số giữa các hệ đếm**

Số nhị phân có dấu

Dấu phẩy động



Chuyển đổi từ hệ cơ số 10 sang các hệ khác

Ví dụ: Đổi số 22.125_{10} , 83.87_{10} sang số nhị phân

- **Đổi với phần nguyên:**

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

- **Đổi với phần phân số:**

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

Đổi số 22.125_{10} sang số nhị phân

■ Đổi với phần nguyên:

Bước	Chia	Được	Dư	
1	$22/2$	11	0	LSB
2	$11/2$	5	1	
3	$5/2$	2	1	
4	$2/2$	1	0	
5	$1/2$	0	1	MSB

■ Đổi với phần phân số:

Bước	Nhân	Kết quả	Phần nguyên
1	0.125×2	0.25	0
2	0.25×2	0.5	0
3	0.5×2	1	1
4	0×2	0	0

■ Kết quả biểu diễn nhị phân: 10110.001

Đổi số 83.87_{10} sang số nhị phân

■ Đổi với phần nguyên:

Bước	Chia	Được	Dư	
1	$83/2$	41	1	LSB
2	$41/2$	20	1	
3	$20/2$	10	0	
4	$10/2$	5	0	
5	$5/2$	2	1	
6	$2/2$	1	0	
7	$1/2$	0	1	MSB

■ Đổi với phần phân số:

Bước	Nhân	Kết quả	Phần nguyên
1	0.87×2	1.74	1
2	0.74×2	1.48	1
3	0.48×2	0.96	0
4	0.96×2	1.92	1
5	0.92×2	1.84	1
6	0.84×2	1.68	1
7	0.68×2	1.36	1
8	0.36×2	0.72	0

■ Kết quả biểu diễn nhị phân: 1010011.11011110

Đổi một biểu diễn trong hệ bất kì sang hệ 10

- **Công thức chuyển đổi:**

$$N_{10} = a_{n-1} \times r^{n-1} + a_{n-2} \times r^{n-2} + \dots + a_0 \times r^0 + a_{-1} \times r^{-1} + \dots + a_{-m} \times r^{-m}$$

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

- **Ví dụ: Chuyển 1101110.10_2 sang hệ thập phân**

$$\begin{aligned} N_{10} &= 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} \\ &= 64 + 32 + 0 + 8 + 4 + 2 + 0 + 0.5 + 0 = 110.5 \end{aligned}$$

Đổi các số từ hệ nhị phân sang hệ cơ số 8, 16

■ Quy tắc:

- Vì $8 = 2^3$ và $16 = 2^4$ nên ta chỉ cần dùng một số nhị phân 3 bit là đủ ghi 8 ký hiệu của hệ cơ số 8 và từ nhị phân 4 bit cho hệ cơ số 16.
- Do đó, muốn đổi một số nhị phân sang hệ cơ số 8 và 16 ta chia số nhị phân cần đổi, kể từ dấu phân số sang trái và phải thành từng nhóm 3 bit hoặc 4 bit. Sau đó thay các nhóm bit đã phân bằng ký hiệu tương ứng của hệ cần đổi tới.

■ Ví dụ: Chuyển 1101110.10_2 sang hệ cơ số 8 và 16

Tính từ dấu phân số, chia số đã cho thành các nhóm 3 bit

001 101 110 . 100

↓ ↓ ↓ ↓

1 5 6 4

Kết quả: $1101110.10_2 = 156.4$

Tính từ dấu phân số, chia số đã cho thành các nhóm 4 bit

0110 1110 . 1000

↓ ↓ ↓

6 E 8

Kết quả: $1101110.10_2 = 6E.8$



Nội dung

Biểu diễn số

Chuyển đổi cơ số giữa các hệ đếm

- **Số nhị phân có dấu**

Dấu phẩy động

3 phương pháp biểu diễn số nhị phân có dấu

■ Sử dụng một bit dấu.

- Trong phương pháp này ta dùng một bit phụ, đứng trước các bit trị số để biểu diễn dấu, '0' chỉ dấu dương (+), '1' chỉ dấu âm (-).
- Ví dụ: số 6: **00000110**, số -6: **10000110**.

■ Sử dụng phép bù 1.

- Giữ nguyên bit dấu và lấy bù 1 các bit trị số (bù 1 bằng đảo của các bit cần được lấy bù).
- Ví dụ: số 4: **00000100**, số -4: **111111011**.

■ Sử dụng phép bù 2

- Là phương pháp phổ biến nhất. Số dương thể hiện bằng số nhị phân không bù (bit dấu bằng 0), còn số âm được biểu diễn qua bù 2 (bit dấu bằng 1). Bù 2 bằng bù 1 cộng 1.
- Có thể biểu diễn số âm theo phương pháp bù 2 xen kẽ: bắt đầu từ bit LSB, dịch về bên trái, giữ nguyên các bit cho đến gặp bit 1 đầu tiên và lấy bù các bit còn lại. Bit dấu giữ nguyên.
- Ví dụ: số 4: **00000100**, số -4: **111111100**.

Cộng và trừ các số theo biểu diễn bit dấu

■ Phép cộng

- **Hai số cùng dấu:** cộng hai phần trị số với nhau, còn dấu là dấu chung.
- **Hai số khác dấu và số dương lớn hơn:** cộng trị số của số dương với bù 1 của số âm. Bit tràn được cộng thêm vào kết quả trung gian. Dấu là dấu dương.
- **Hai số khác dấu và số dương lớn hơn:** cộng trị số của số dương với bù 1 của số âm. Lấy bù 1 của tổng trung gian. Dấu là dấu âm.

■ Phép trừ.

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

■ Ví dụ:

Cộng và trừ các số theo biểu diễn bù 1

■ Phép cộng

- **Hai số dương:** cộng như cộng nhị phân thông thường, kể cả bit dấu.
- **Hai số âm:** biểu diễn chúng ở dạng bù 1 và cộng như cộng nhị phân, kể cả bit dấu. Bit tràn cộng vào kết quả. Chú ý, kết quả được viết dưới dạng bù 1.
- **Hai số khác dấu và số dương lớn hơn:** cộng số dương với bù 1 của số âm. Bit tràn được cộng vào kết quả.
- **Hai số khác dấu và số âm lớn hơn:** cộng số dương với bù 1 của số âm. Kết quả không có bit tràn và ở dạng bù 1.

■ Phép trừ

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

■ Ví dụ:



Cộng các số theo biểu diễn bù 1: Ví dụ

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

$$\begin{array}{r} 00000101_2 \quad (5_{10}) \\ + 00000111_2 \quad (7_{10}) \\ \hline 00001100_2 \quad (12_{10}) \end{array}$$

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

$$\begin{array}{r} 11111010_2 \quad (-5_{10}) \\ + 11111000_2 \quad (-7_{10}) \\ \hline 111110010_2 \\ \downarrow \quad + \\ \text{Bit tràn} \rightarrow 1 \\ \hline 11110011_2 \quad (-12) \end{array}$$

Cộng các số theo biểu diễn bù 1: Ví dụ

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

$$\begin{array}{r}
 00001010_2 \quad (+10_{10}) \\
 + 11111010_2 \quad (-5_{10}) \\
 \hline
 100000100_2 \\
 \downarrow \quad + \\
 \text{Bit tràn} \rightarrow 1 \\
 \hline
 00000101_2 \quad (+5_{10})
 \end{array}$$

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

$$\begin{array}{r}
 11110101_2 \quad (-10_{10}) \\
 + 00000101_2 \quad (+5_{10}) \\
 \hline
 11111010_2 \quad (-5_{10})
 \end{array}$$

Cộng và trừ các số theo biểu diễn bù 2

■ Phép cộng

- **Hai số dương:** cộng như cộng nhị phân thông thường. Kết quả là dương.
- **Hai số âm:** lấy bù 2 cả hai số hạng và cộng, kết quả ở dạng bù 2.
- **Hai số khác dấu và số dương lớn hơn:** lấy số dương cộng với bù 2 của số âm. Kết quả bao gồm cả bit dấu, bit tràn bỏ đi.
- **Hai số khác dấu và số âm lớn hơn:** số dương được cộng với bù 2 của số âm, kết quả ở dạng bù 2 của số dương tương ứng. Bit dấu là 1.

■ Phép trừ

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

■ Ví dụ:



Cộng các số theo biểu diễn bù 2: Ví dụ

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

$$\begin{array}{r}
 00001011_2 \quad (11_{10}) \\
 + 00000111_2 \quad (7_{10}) \\
 \hline
 00010010_2 \quad (18_{10})
 \end{array}$$

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

$$\begin{array}{r}
 11110101_2 \quad (-11_{10}) \\
 + 11111001_2 \quad (-7_{10}) \\
 \hline
 111101110_2 \\
 \downarrow \quad + \\
 \text{Bít tràn} \rightarrow \text{bỏ đi} \\
 \hline
 11101110_2 \quad (-18_{10})
 \end{array}$$

Cộng các số theo biểu diễn bù 2: Ví dụ

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

$$\begin{array}{r}
 00001011_2 \quad (+11_{10}) \\
 + 11111001_2 \quad (-7_{10}) \\
 \hline
 100000100_2 \\
 \downarrow \quad + \\
 \text{Bít tràn} \rightarrow \text{bỏ đi} \\
 \hline
 00000100_2 \quad (+4_{10})
 \end{array}$$

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

$$\begin{array}{r}
 11110101_2 \quad (-11_{10}) \\
 + 00000111_2 \quad (+7_{10}) \\
 \hline
 11111100_2 \quad (-4_{10})
 \end{array}$$



Nội dung

Biểu diễn số

Chuyển đổi cơ số giữa các hệ đếm

Số nhị phân có dấu

- **Dấu phẩy động**

Biểu diễn theo dấu phẩy động

- Ví dụ: $197,627_{10} = 197627 \times 10^{-3}$

$$197,627_{10} = 0,197627 \times 10^{+3}$$

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

$$X = 2^{E_x} (M_x)$$

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

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

Các phép tính với biểu diễn dấu phẩy động

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

thì:

- **Nhân:** $Z = X.Y = 2^{E_x + E_y} (M_x.M_y) = 2^{E_z} M_z$
- **Chia:** $W = X / Y = 2^{E_x - E_y} (M_x / M_y) = 2^{E_w} M_w$
- **Tích: Thương:** Muốn lấy tổng và hiệu, cần đưa các số hạng về cùng số mũ, sau đó số mũ của tổng và hiệu sẽ lấy số mũ chung, còn định trị của tổng và hiệu sẽ bằng tổng và hiệu các định trị.

- **Đổi số nhị phân sau sang dạng bát phân: 0101 1111 0100 1110**

- A) 57514 B) 57515 C) 57516 D) 57517

- **Thực hiện phép tính hai số thập lục phân sau: $132,44_{16} + 215,02_{16}$.**

- A) 347,46 B) 357,46 C) 347,56 D) 357,67

- **Thực hiện phép cộng hai số có dấu sau theo phương pháp bù 1:**

$$0000\ 1101_2 + 1000\ 1011_2$$

- A) 0000 0101 B) 0000 0100 C) 0000 0011 D) 0000 0010

- **Thực hiện phép cộng hai số có dấu sau theo phương pháp bù 2:**

$$0000\ 1101_2 - 1001\ 1000_2$$

- A) 1000 1110 B) 1000 1011 C) 1000 1100 D) 1000 1110



Nội dung

Chương 1: Hệ đếm

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

Chương 3: Cổng logic

Chương 4: Mạch logic tổ hợp

Chương 5: Mạch logic tuần tự

Chương 6: Mạch phát xung và tạo dạng xung

Chương 7: Bộ nhớ bán dẫn

Đại số Boole và các phương pháp biểu diễn hàm

Nội dung

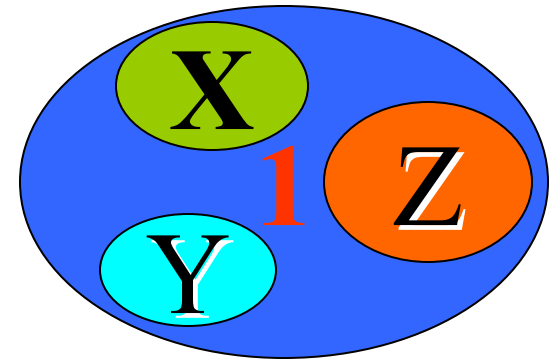
- Đại số Boole

Các phương pháp biểu diễn hàm Boole

Các phương pháp rút gọn hàm

■ Các định lý cơ bản:

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



■ Các định luật cơ bản:

- Hoán vị: $X.Y = Y.X$, $X + Y = Y + X$
- Kết hợp: $X.(Y.Z) = (X.Y).Z$, $X + (Y + Z) = (X + Y) + Z$
- Phân phối: $X.(Y + Z) = X.Y + X.Z$, $(X + Y).(X + Z) = X + Y.Z$



Các phương pháp biểu diễn hàm Boole

Có 3 phương pháp biểu diễn:

- **Bảng trạng thái**
- **Bảng các nô (Karnaugh)**
- **Phương pháp đại số**

Phương pháp Bảng trạng thái

- Liệt kê giá trị (trạng thái) mỗi biến theo từng cột và giá trị hàm theo một cột riêng (thường là bên phải bảng). Bảng trạng thái còn được gọi là *bảng sự thật* hay *bảng chân lý*.
- Đối với hàm n biến sẽ có 2^n tổ hợp độc lập. Các tổ hợp này được kí hiệu bằng chữ m_i , với $i = 0 \div 2^n - 1$ và có tên gọi là các *hạng tích* hay còn gọi là *mintex*.
- **Ưu điểm:** Rõ ràng, trực quan. Sau khi xác định các giá trị biến vào thì ta có thể tìm được giá trị đầu ra nhờ bảng trạng thái.
- **Nhược điểm:** Sẽ phức tạp nếu số biến quá nhiều, không thể dùng các công thức và định lý để tính toán

m	A	B	C	f
m_0	0	0	0	0
m_1	0	0	1	0
m_2	0	1	0	0
m_3	0	1	1	0
m_4	1	0	0	0
m_5	1	0	1	0
m_6	1	1	0	0
m_7	1	1	1	1

Phương pháp Bảng Các ô (Karnaugh)

■ Tổ chức của bảng Các ô:

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

A \ B	0	1
0		
1		

■ Tính tuần hoàn của bảng Các ô:

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

A \ BC	00	01	11	10
0				
1				

■ Thiết lập bảng Các ô của một hàm:

- Dưới dạng chuẩn **tổng các tích**, ta chỉ việc ghi giá trị 1 vào các ô ứng với hạng tích có mặt trong biểu diễn, các ô còn lại sẽ lấy giá trị 0 (theo định lý DeMorgan).
- Dưới dạng **tích các tổng**, cách làm cũng tương tự, nhưng các ô ứng với hạng tổng có trong biểu diễn lại lấy giá trị 0 và các ô khác lấy giá trị 1.

AB \ CD	00	01	11	10
00				
01				
11				
10				

- Có 2 dạng biểu diễn là dạng *tuyến (tổng các tích)* và dạng *hội (tích các tổng)*.
 - Dạng tuyến: Mỗi số hạng là một *hạng tích* hay *mintex*, thường kí hiệu bằng chữ " m_i ".
 - Dạng hội: Mỗi thừa số là *hạng tổng* hay *maxtex*, thường được kí hiệu bằng chữ " M_i ".
- Nếu trong tất cả mỗi hạng tích hay hạng tổng có đủ mặt các biến, thì dạng tổng các tích hay tích các tổng tương ứng được gọi là dạng *chuẩn*. Dạng chuẩn là duy nhất.
- Tổng quát, hàm logic n biến có thể biểu diễn chỉ bằng một dạng tổng các tích:

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

hoặc bằng chỉ một dạng tích các tổng:

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

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



Các phương pháp rút gọn hàm

Có 3 phương pháp rút gọn hàm:

- Phương pháp đại số
- Phương pháp bảng Karnough
- Phương pháp Quine Mc. Cluskey

- Dựa vào các định lý đã học để đưa biểu thức về dạng tối giản.
- Ví dụ: Hãy đưa hàm logic về dạng tối giản:

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

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

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

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

Phương pháp đại số (tiếp)

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

$$f = AB + BCD + \bar{A}C + \bar{B}C$$

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

$$\begin{aligned} f &= AB + BCD(A + \bar{A}) + \bar{A}C + \bar{B}C \\ &= (AB + ABCD) + (\bar{A}BCD + \bar{A}C) + \bar{B}C \\ &= AB + \bar{A}C + \bar{B}C = AB + \overline{AB}.C \\ &= AB(1 + C) + \overline{AB}.C \\ &= AB + C \end{aligned}$$

$$f_1 = A\bar{D} + \bar{B}D + B\bar{C}D + \bar{A}CD + ABC$$

Phương pháp Bảng Các nô (Karnaugh)

- Phương pháp này thường được dùng để rút gọn các hàm có số biến không vượt quá 5.
- Các bước tối thiểu hóa:
 - 1. Gộp các ô kế cận có giá trị '1' (hoặc '0') lại thành từng nhóm 2, 4, ..., 2^i ô. Số ô trong mỗi nhóm càng lớn kết quả thu được càng tối giản. Một ô có thể được gộp nhiều lần trong các nhóm khác nhau. Nếu gộp theo các ô có giá trị '0' ta sẽ thu được biểu thức bù của hàm.
 - 2. Thay mỗi nhóm bằng một hạng tích mới, trong đó giữ lại các biến giống nhau theo dòng và cột.
 - 3. Cộng các hạng tích mới lại, ta có hàm đã tối giản.

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

$$f = AB + BCD + \bar{A}C + \bar{B}C$$

CD \ AB	00	01	11	10
00			1	1
01			1	1
11	1	1	1	1
10			1	1

$$f_1 = AB$$

$$f_2 = C$$

Kết quả $f = AB + C$

$$f_3(A, B, C, D) = \sum (0, 1, 2, 3, 5, 7, 8, 9, 10, 13)$$

Phương pháp Quine Mc. Cluskey

- Phương pháp này có thể tối thiểu hóa được hàm nhiều biến và có thể tiến hành công việc nhờ máy tính.
- Các bước tối thiểu hóa:
 1. Lập bảng liệt kê các hạng tích dưới dạng nhị phân theo từng nhóm với số bit 1 giống nhau và xếp chúng theo số bit 1 tăng dần.
 2. Gộp 2 hạng tích của mỗi cặp nhóm chỉ khác nhau 1 bit để tạo các nhóm mới. Trong mỗi nhóm mới, giữ lại các biến giống nhau, biến bỏ đi thay bằng một dấu ngang (-).

Lặp lại cho đến khi trong các nhóm tạo thành không còn khả năng gộp nữa. Mỗi lần rút gọn, ta đánh dấu # vào các hạng ghép cặp được. Các hạng không đánh dấu trong mỗi lần rút gọn sẽ được tập hợp lại để lựa chọn biểu thức tối giản.
- Ví dụ: $f(A, B, C, D) = \sum(10, 11, 12, 13, 14, 15)$

Phương pháp Quine Mc. Cluskey (tiếp)

■ Bước 1: Lập bảng

Bảng a		Bảng b	
Hạng tích sắp xếp	Nhị phân (ABCD)	Rút gọn lần 1 (ABCD)	Rút gọn lần thứ 2 (ABCD)
10	1 0 1 0	1 0 1 - # (10,11)	1 1 - - (12,13,14,15)
<u>12</u>	<u>1 1 0 0</u>	1 - 1 0 # (10,14)	1 - 1 - (10,11,14,15)
11	1 0 1 1	1 1 0 - # (12,13)	
13	1 1 0 1	<u>1 1 - 0</u> # (12,14)	
<u>14</u>	<u>1 1 1 0</u>	1 - 1 1 # (11,15)	
15	1 1 1 1	1 1 - 1 # (13,15)	
		1 1 1 - # (14,15)	

■ Bước 2: Thực hiện nhóm các hạng tích

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

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

A BCD	10	11	12	13	14	15
1 1 - -			x	x	x	x
1 - 1 -	x	x			x	x



Nội dung

Chương 1: Hệ đếm

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

■ **Chương 3: Cổng logic**

Chương 4: Mạch logic tổ hợp

Chương 5: Mạch logic tuần tự

Chương 6: Mạch phát xung và tạo dạng xung

Chương 7: Bộ nhớ bán dẫn

Cổng logic



Nội dung

- Các cổng logic và các tham số chính
- Các họ cổng logic
- Giao tiếp giữa các cổng logic cơ bản



Cổng logic và các tham số chính

- Cổng logic cơ bản
- Một số cổng ghép thông dụng
- Logic dương và logic âm
- Các tham số chính



Cổng logic cơ bản: AND, OR, NOT

- Cổng AND
- Cổng OR
- Cổng NOT

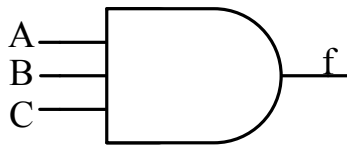
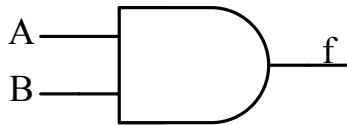
Cổng AND

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

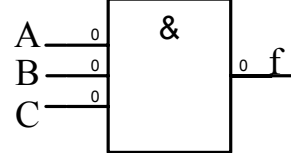
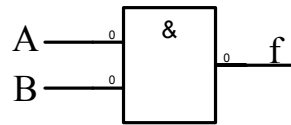
$$f = f(A, B) = AB;$$

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

Ký hiệu cổng AND



Chuẩn ANSI



Chuẩn IEEE

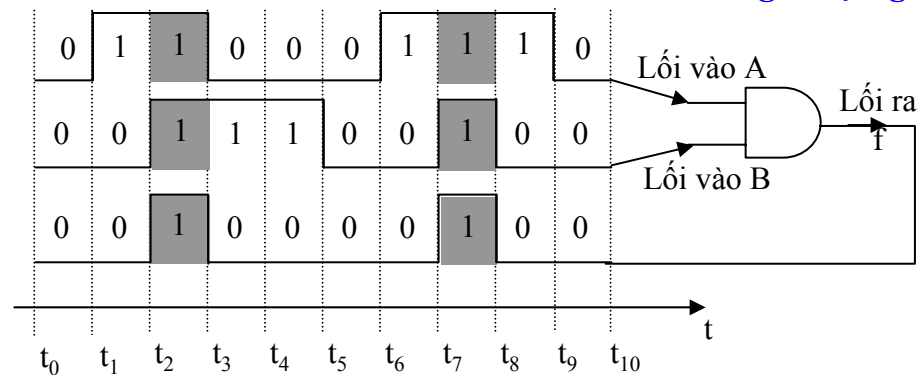
Bảng trạng thái cổng AND 2 lối vào

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

Theo giá trị logic

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

Theo mức logic



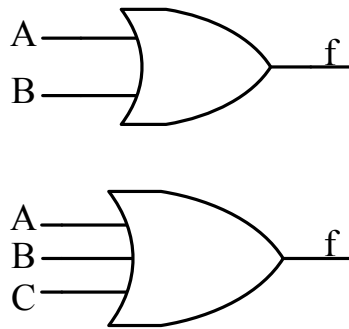
Đồ thị dạng xung vào, ra của cổng AND

Cổng OR

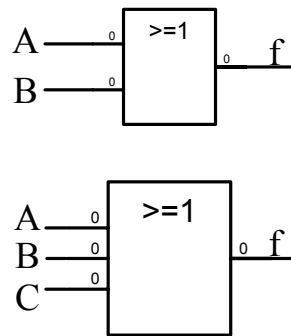
- Hàm ra của cổng OR 2 và nhiều biến vào như sau:

$$f = f(A, B) = A + B; \quad f = f(A, B, C, D, \dots) = A + B + C + D + \dots$$

Ký hiệu cổng OR



Chuẩn ANSI



Chuẩn IEEE

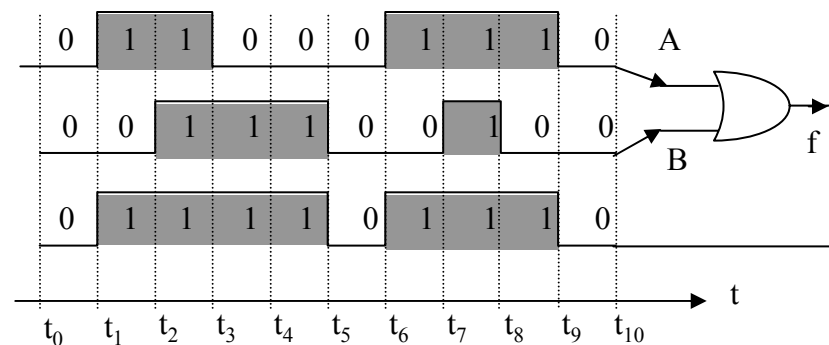
Bảng trạng thái cổng OR 2 lối vào

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

Theo giá trị logic

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

Theo mức logic

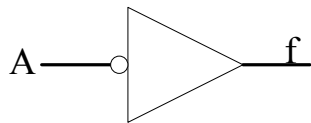
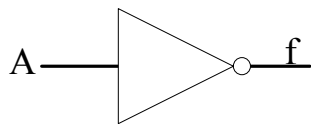


Đồ thị dạng xung của cổng OR.

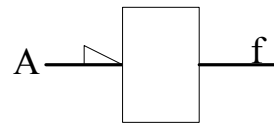
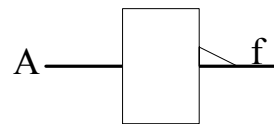
■ Hàm ra của cổng NOT:

$$f = \bar{A}$$

Ký hiệu cổng NOT



Chuẩn ANSI



Chuẩn IEEE

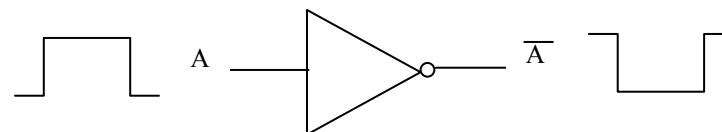
Bảng trạng thái cổng NOT

A	f
0	1
1	0

Theo giá trị logic

A	f
L	H
H	L

Theo mức logic



Dạng xung ra

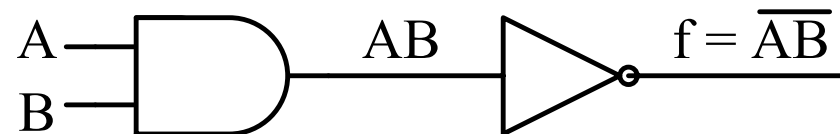


Một số cổng ghép thông dụng

- Cổng NAND
- Cổng NOR
- Cổng khác dấu (XOR)
- Cổng đồng dấu (XNOR)

Cổng NAND

- Ghép nối tiếp một cổng AND với một cổng NOT ta được cổng NAND.

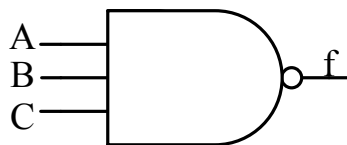
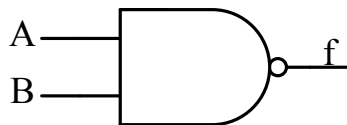


- Hàm ra của cổng NAND 2 và nhiều biến vào như sau:

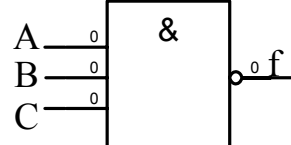
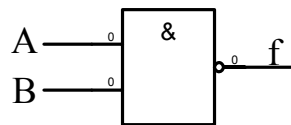
$$f = \overline{AB}$$

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

Ký hiệu cổng NAND



Chuẩn ANSI



Chuẩn IEEE

Bảng trạng thái cổng NAND 2 lối vào

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

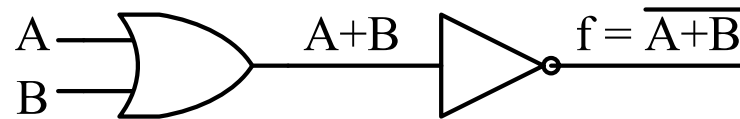
Theo giá trị logic

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

Theo mức logic

Cổng NOR

- Ghép nối tiếp một cổng OR với một cổng NOT ta được cổng NOR.

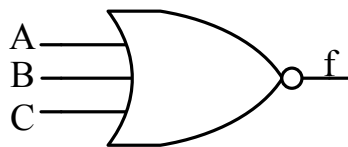
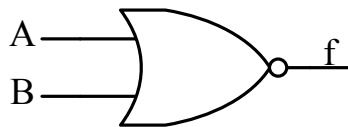


- Hàm ra của cổng NOR 2 và nhiều biến vào như sau:

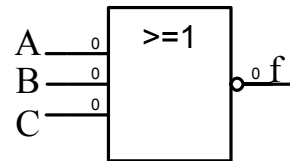
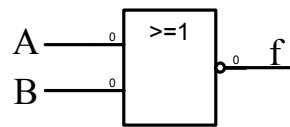
$$f = \overline{A + B}$$

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

Ký hiệu cổng NOR



Chuẩn ANSI



Chuẩn IEEE

Bảng trạng thái cổng NOR 2 lối vào

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

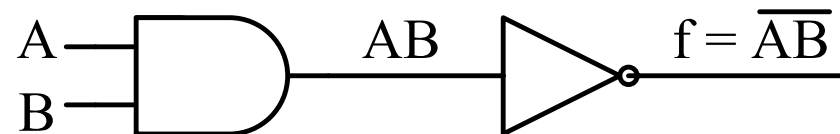
Theo giá trị logic

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

Theo mức logic

Cổng NAND

- Ghép nối tiếp một cổng AND với một cổng NOT ta được cổng NAND.

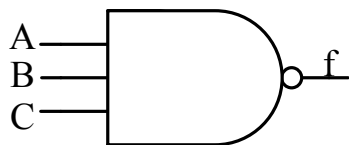
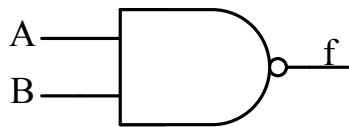


- Hàm ra của cổng NAND 2 và nhiều biến vào như sau:

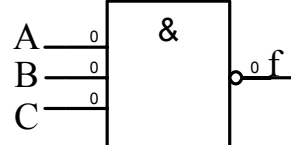
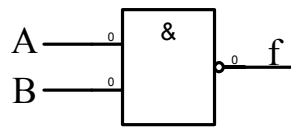
$$f = \overline{AB}$$

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

Ký hiệu cổng NAND



Chuẩn ANSI



Chuẩn IEEE

Bảng trạng thái cổng NAND 2 lối vào

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

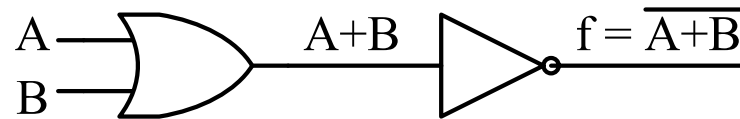
Theo giá trị logic

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

Theo mức logic

Cổng NOR

- Ghép nối tiếp một cổng OR với một cổng NOT ta được cổng NOR.

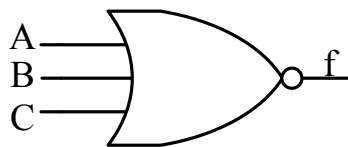
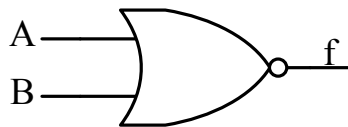


- Hàm ra của cổng NOR 2 và nhiều biến vào như sau:

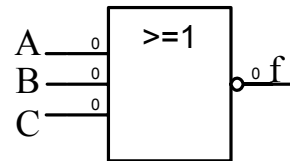
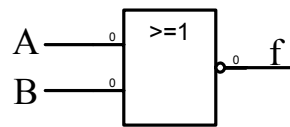
$$f = \overline{A + B}$$

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

Ký hiệu cổng NOR



Chuẩn ANSI



Chuẩn IEEE

Bảng trạng thái cổng NOR 2 lối vào

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

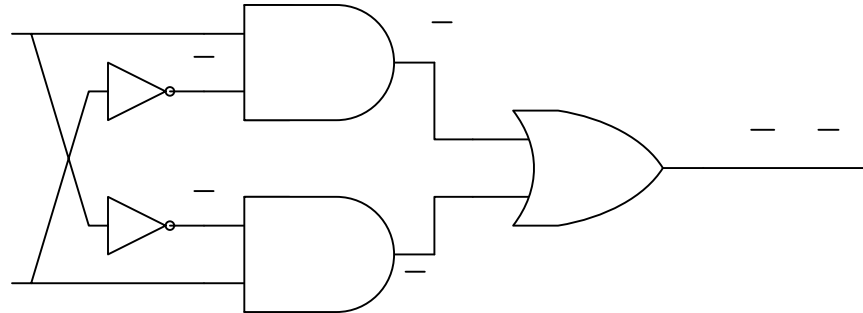
Theo giá trị logic

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

Theo mức logic

Cổng XOR - cổng khác dấu

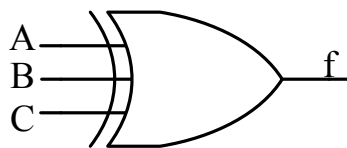
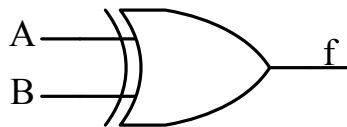
- Cổng XOR còn gọi là *cổng khác dấu*, hay *cộng modul 2*.



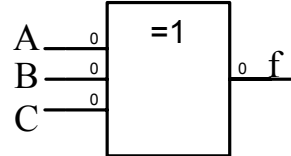
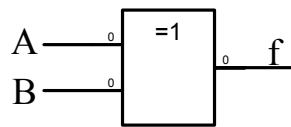
- Hàm ra của cổng XOR 2 biến vào như sau:

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

Ký hiệu cổng XOR



Chuẩn ANSI



Chuẩn IEEE

Bảng trạng thái cổng XOR 2 lối vào

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

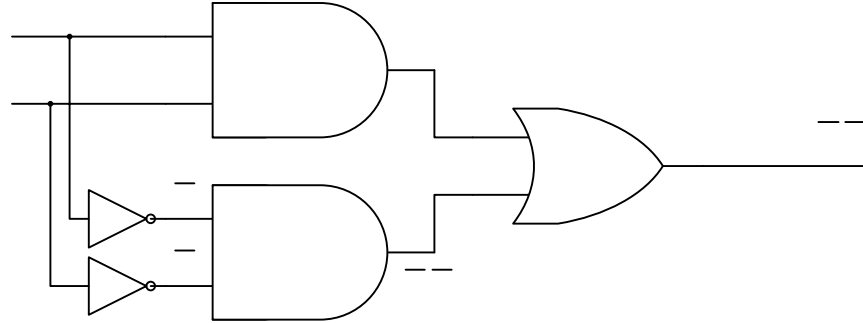
Theo giá trị logic

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

Theo mức logic

Cổng XNOR - cổng đồng dấu

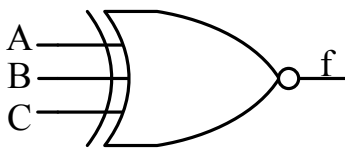
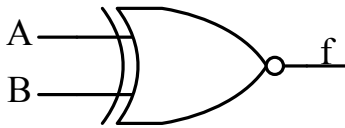
- Cổng XNOR còn gọi là *cổng đồng dấu*.



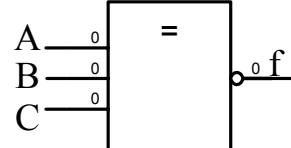
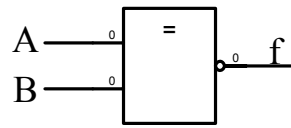
- Hàm ra của cổng XNOR 2 biến vào như sau:

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

Ký hiệu cổng XNOR



Chuẩn ANSI



Chuẩn IEEE

Bảng trạng thái cổng XNOR 2 lối vào

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

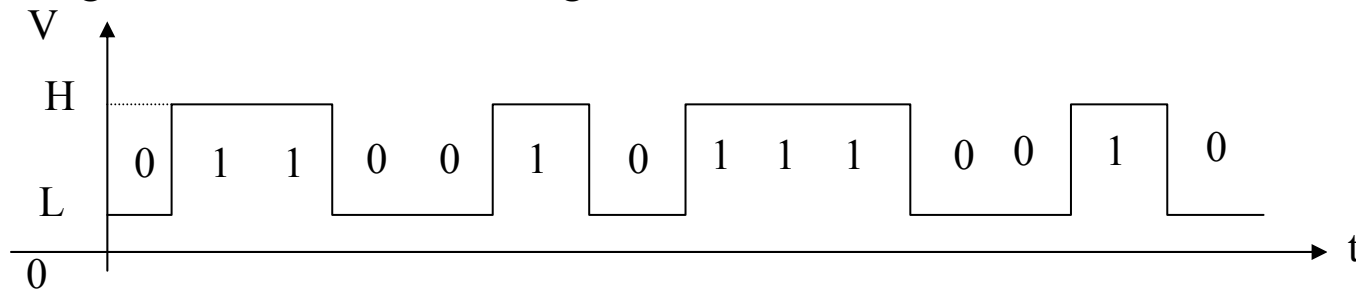
Theo giá trị logic

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

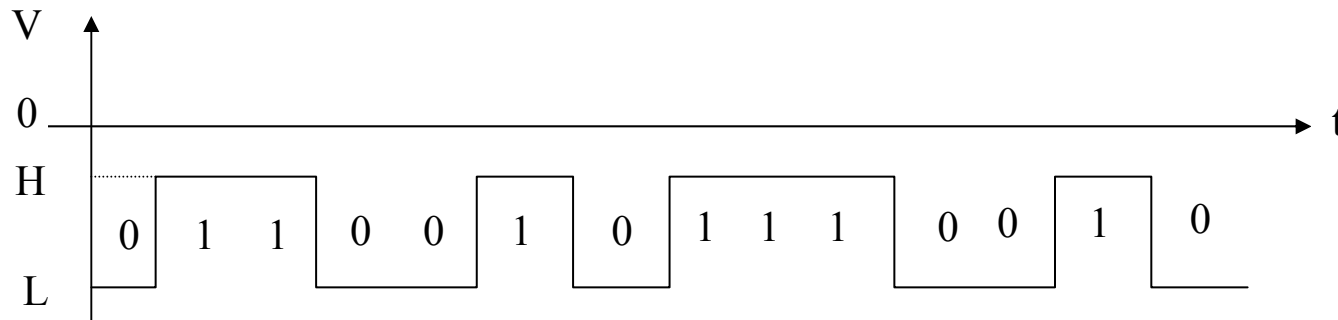
Theo mức logic

Logic dương và logic âm

- **Logic dương** là logic có điện thế mức cao H luôn lớn hơn điện thế mức thấp L ($V_H > V_L$).
- **Logic âm** là đảo của logic dương ($V_H < V_L$).
 - Khái niệm logic âm thường được dùng để biểu diễn trị các biến.
 - Logic âm và mức âm của logic là hoàn toàn khác nhau.



a) Logic dương với mức dương.

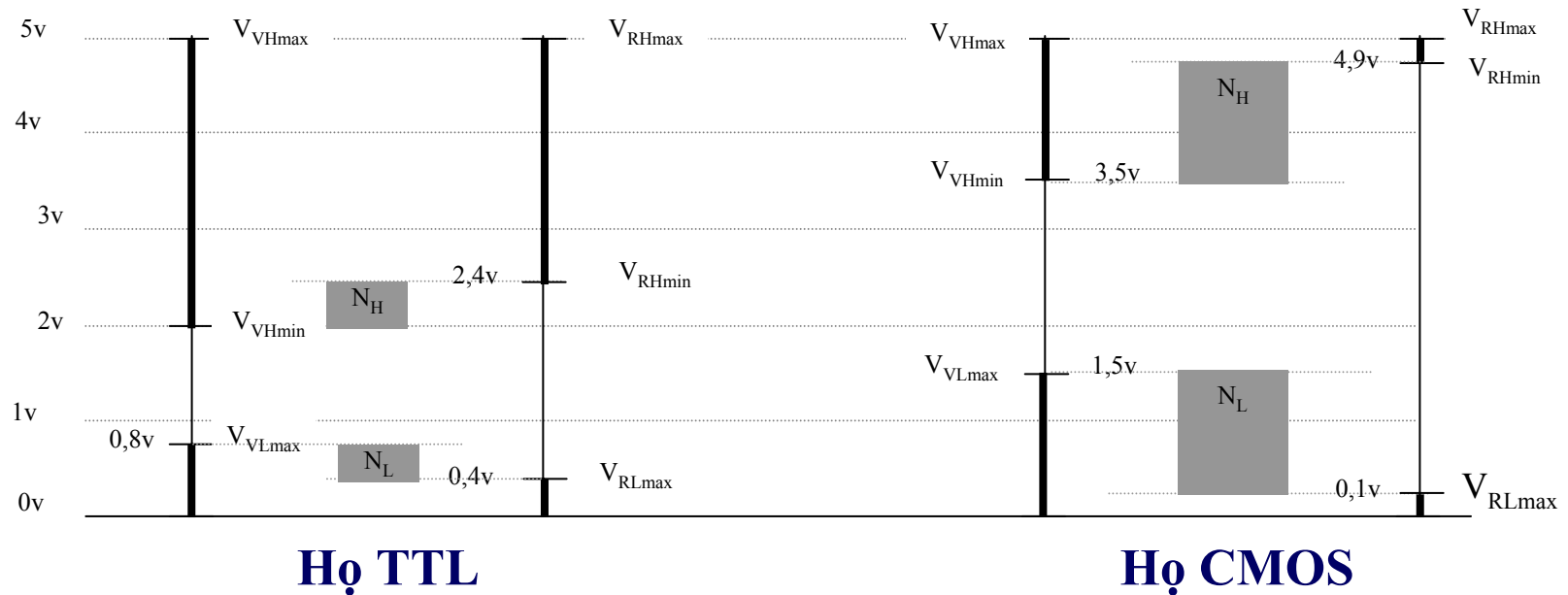


b) Logic dương với mức âm.



Các tham số chính

- **Mức logic**
- **Độ chống nhiễu**
- **Hệ số ghép tải K**
- **Công suất tiêu thụ**
- **Trễ truyền lan**

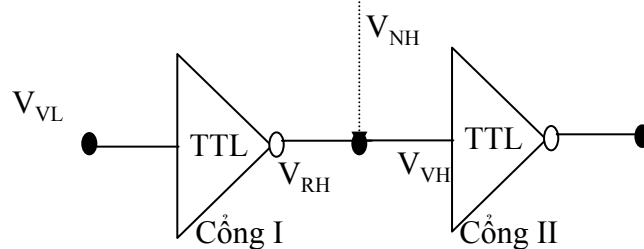


Mức logic là mức điện thế trên đầu vào và đầu ra của cổng tương ứng với logic "1" và logic "0", nó phụ thuộc điện thế nguồn nuôi của cổng (V_{CC} đối với họ TTL (Transistor Transistor Logic) và V_{DD} đối với họ MOS (Metal Oxide Semiconductor)).

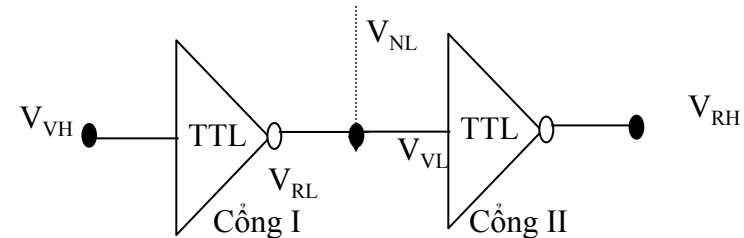
Lưu ý: mức logic vào vượt quá điện thế nguồn nuôi có thể gây hư hỏng cho cổng.

Độ chống nhiễu

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



a) Tác động nhiễu khi mức ra cao



b) Tác động nhiễu khi mức ra thấp

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

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

$$V_{RHmin} + V_{NH} \geq V_{VHmin} \Leftrightarrow V_{NH} \geq V_{VHmin} - V_{RHmin}$$

Với cổng TTL:

$$V_{NL} \geq 2V - 2,4V = -0,4V$$

Với cổng CMOS:

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

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

$$V_{RLmax} + V_{NL} \leq V_{VLmax} \Leftrightarrow V_{NL} \leq V_{VLmax} - V_{RLmax}$$

Với cổng TTL:

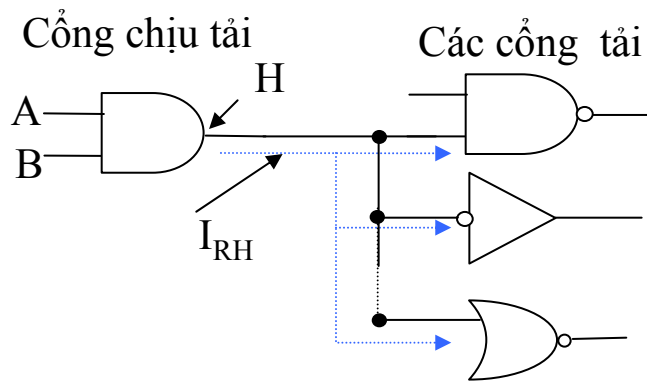
$$V_{NL} \leq 0,8V - 0,4V = 0,4V$$

Với cổng CMOS:

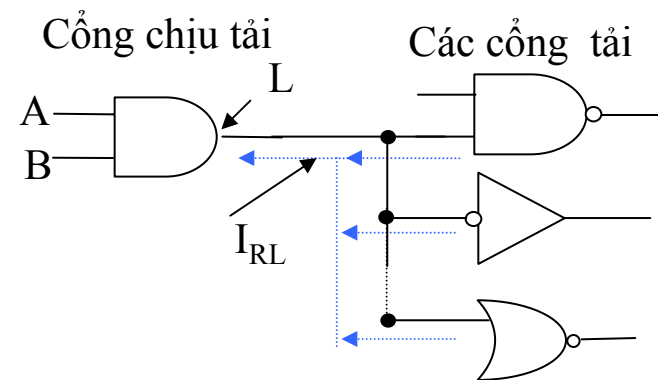
$$V_{NL} \leq 1,5V - 0,1V = 1,4V$$

Hệ số ghép tải K

- Cho biết khả năng nối được bao nhiêu lối vào tới đầu ra của 1 cổng đã cho.
- Hệ số ghép tải phụ thuộc dòng ra (hay dòng phun) của cổng chịu tải và dòng vào (hay dòng hút) của các cổng tải ở cả hai trạng thái H, L.



a) Mức ra của cổng chịu tải là H



b) Mức ra của cổng chịu tải là L

- Công thức tính hệ số ghép tải:

$$K_t = \frac{I_{RL\max}}{I_{RL}} ;$$

$I_{RL}=1,6\text{mA}$ gọi là đơn vị ghép tải (D_t)



Hai trạng thái tiêu thụ dòng của cổng logic

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

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

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

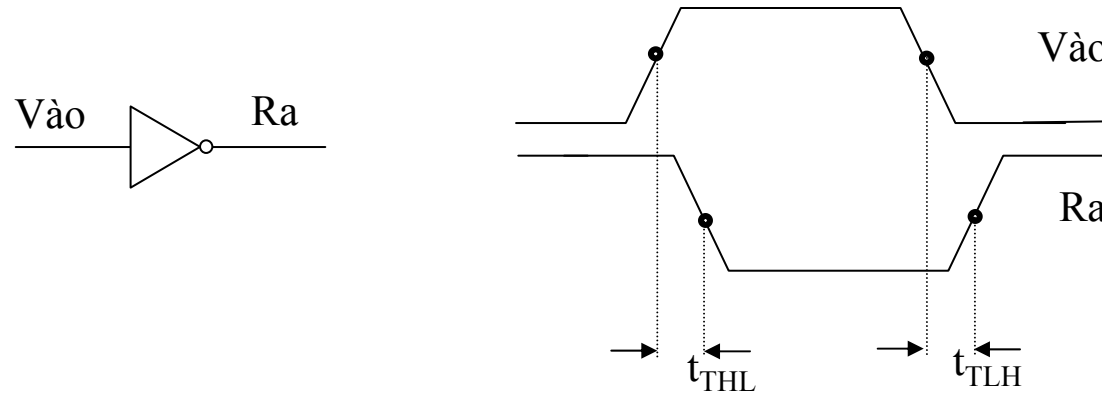
$$I_{CC} = (I_{CCH} + I_{CCL}) / 2$$

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

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

Trễ truyền lan

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



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

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

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



Các họ cổng logic

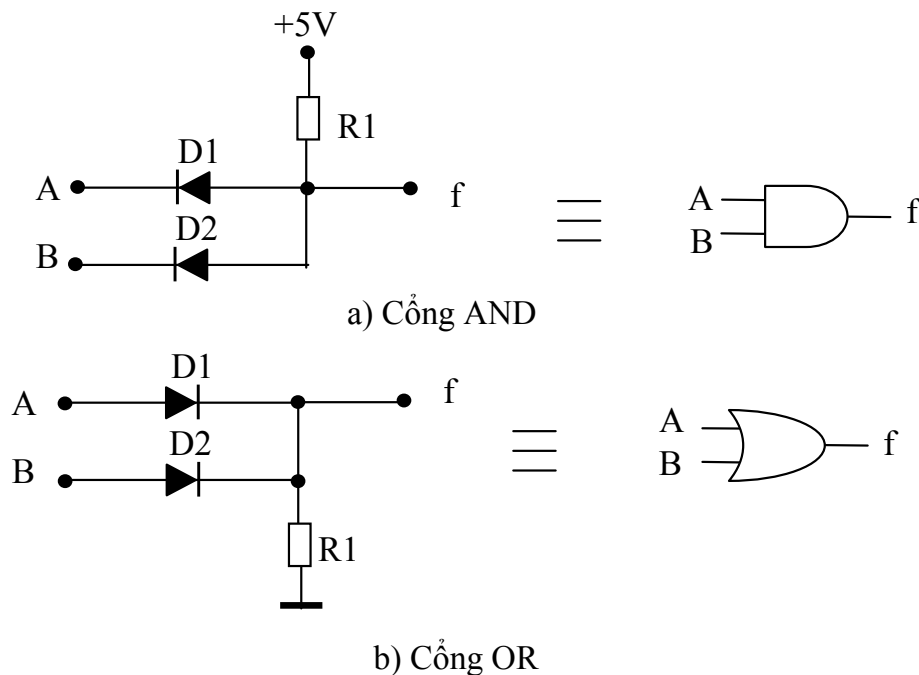
- **H₀ DDL**
- **H₀ DTL**
- **H₀ RTL**
- **H₀ TTL**
- **H₀ MOS FET**

- DDL (Diode Diode Logic) là họ cổng logic do các diode bán dẫn tạo thành.

Cổng AND, OR 2 lối vào họ DDL:

Bảng trạng thái

thể hiện nguyên lý hoạt động của các cổng AND, OR họ DDL



AND		
A(v)	B(v)	f(v)
0	0	0,7
0	3	0,7
3	0	0,7
3	3	4,7

OR		
A(v)	B(v)	f(v)
0	0	0
0	5	4,3
5	0	4,3
5	5	4,3

Theo mức điện áp vào/ra

■ *Ưu điểm của họ DDL:*

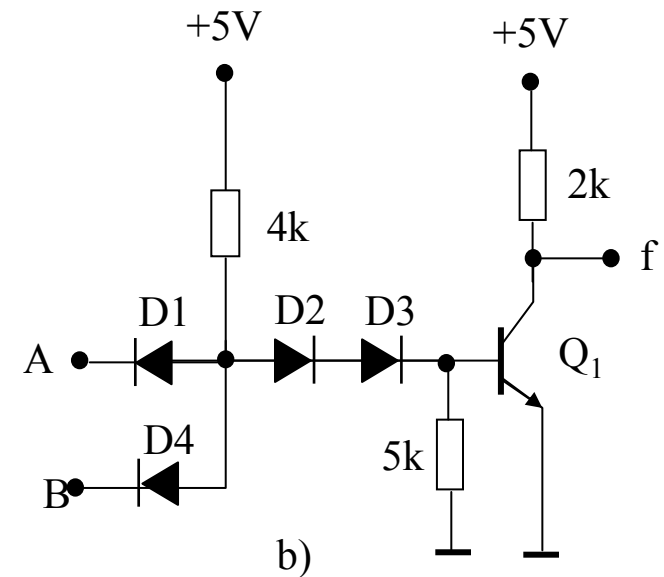
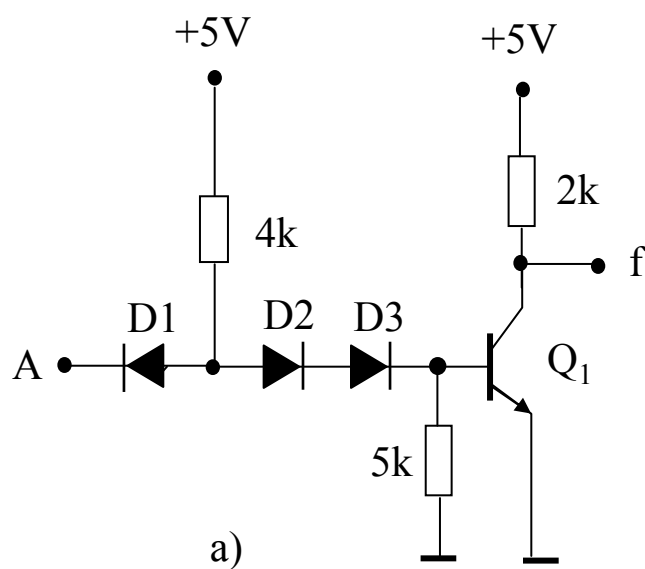
- Mạch điện đơn giản, dễ tạo ra các cổng AND, OR nhiều lối vào. Ưu điểm này cho phép xây dựng các ma trận diode với nhiều ứng dụng khác nhau;
- Tần số công tác có thể đạt cao bằng cách chọn các diode chuyển mạch nhanh;
- Công suất tiêu thụ nhỏ.

■ *Nhược điểm của họ DDL:*

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

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

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



- Bằng cách tương tự, ta có thể thiết lập cổng NOR hoặc các cổng liên hợp phức tạp hơn.

■ *Ưu điểm của họ DTL:*

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

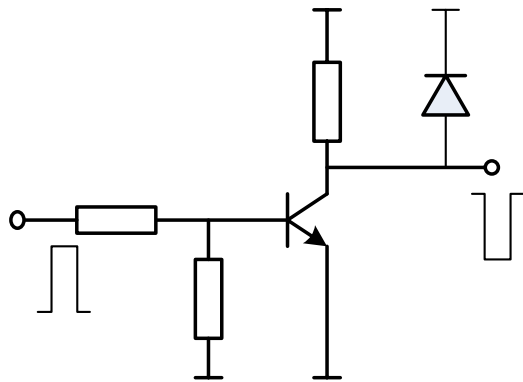
■ *Nhược điểm của họ DTL:*

- Vì tải của các cổng là điện trở nên hệ số ghép tải (đặc biệt đối với NH) còn bị hạn chế,
- Trễ truyền lan của họ cổng này còn lớn.

Những tồn tại trên sẽ được khắc phục từng phần ở các họ cổng sau.

- Họ RTL (Resistor Transistor Logic) là các cổng logic được cấu tạo bởi các điện trở và transistor.

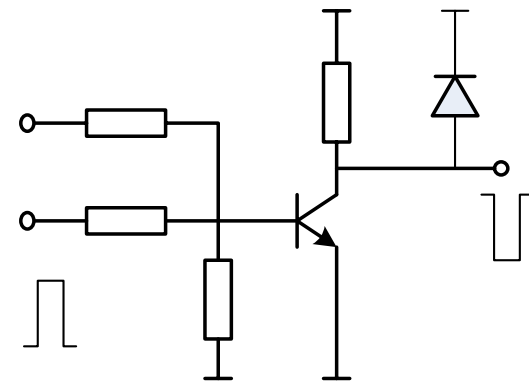
Cổng NOT họ RTL



Bảng trạng thái

A(V)	f(V)
0	5,7
5	0

Cổng NOR 2 lối vào họ RTL



Bảng trạng thái

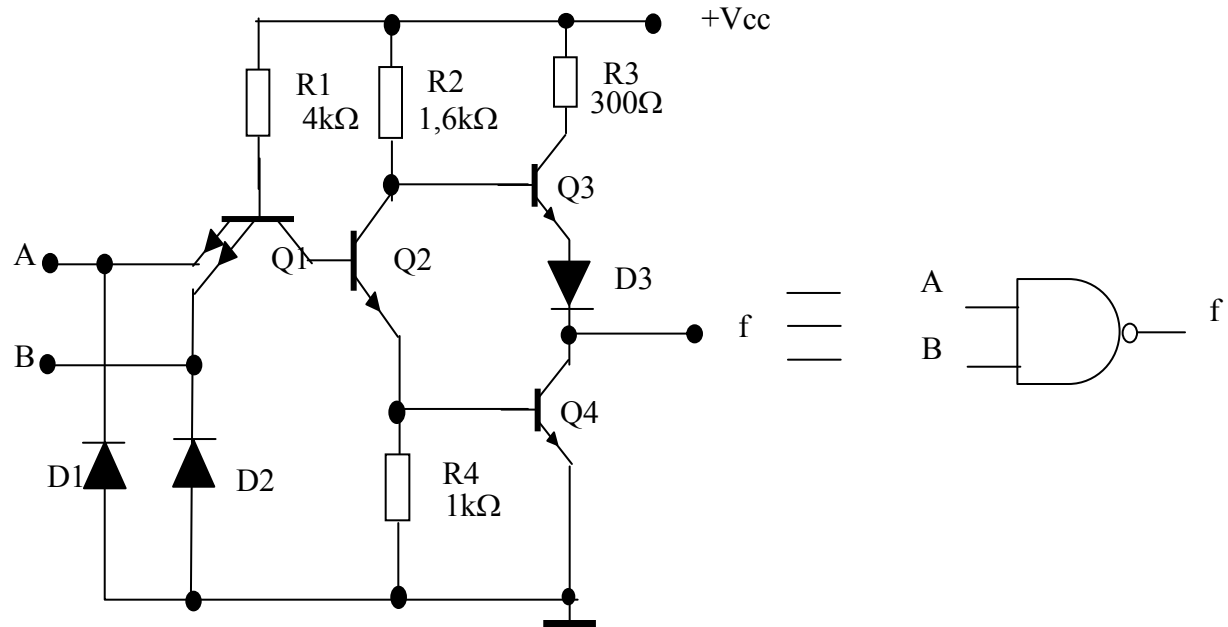
A(V)	B(V)	f(V)
0	0	5,7
0	5	0
5	0	0
5	5	0

- Thay các điốt đầu vào họ DTL thành transistor đa lớp tiếp giáp BE ta được họ TTL (Transistor Transistor Logic).
- Một số mạch TTL
 - Mạch cổng NAND
 - Mạch cổng OR
 - Mạch cổng collector để hở
 - Mạch cổng TTL 3 trạng thái
 - Họ TTL có diode Schottky (TTL + S)

Mạch cổng NAND TTL

Sơ đồ nguyên lý của mạch NAND TTL có thể được chia ra thành 3 phần.

- **Mạch đầu vào:** gồm Transistor Q1, trở R1 và các diode D1, D2. Mạch này thực hiện chức năng NAND.
- **Mạch giữa:** gồm Transistor Q2, các trở R2, R4.
- **Mạch đầu ra:** gồm Q3, Q4, R3 và diode D3.

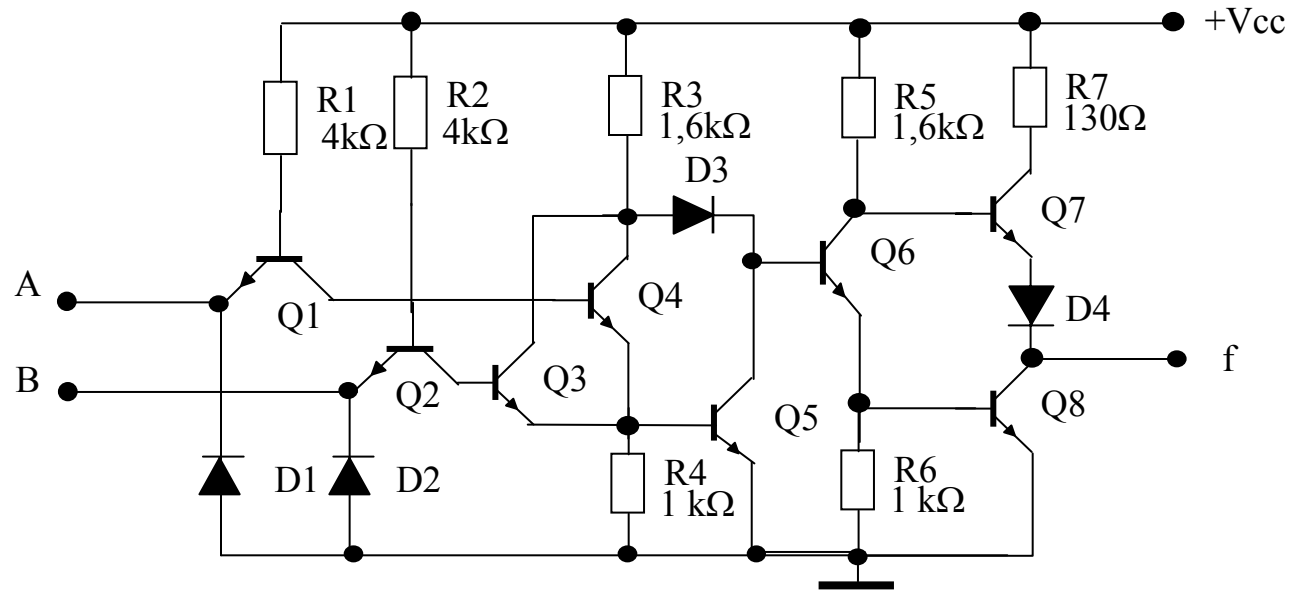


- Khi bất kỳ một lối vào ở mức thấp thì Q1 đều trở thành thông bão hoà, do đó Q2 và Q4 đóng, còn Q3 thông nên đầu ra của mạch sẽ ở mức cao. Lối ra sẽ chỉ xuống mức thấp khi tất cả các lối vào đều ở mức logic cao và làm transistor Q1 cấm. Diode D3 được sử dụng như mạch dịch mức điện áp, nó có tác dụng làm cho Q3 cấm hoàn toàn khi Q2 và Q4 thông. Diode này nhiều khi còn được mắc vào mạch giữa collector Q2 và base của Q3.

Mạch cổng OR TTL

Sơ đồ nguyên lý của mạch NAND TTL có thể được chia ra thành 3 phần.

- **Mạch đầu vào:** gồm Transistor Q1, Q2, Q3, trở R1, R2 và các diode D1, D2. Mạch này thực hiện chức năng OR.
- **Mạch giữa:** gồm Transistor Q4, Q5, các trở R3, R4, và diode D3.
- **Mạch đầu ra:** gồm Q6, Q7, Q8, các trở R5, R6, R7 và diode D4.

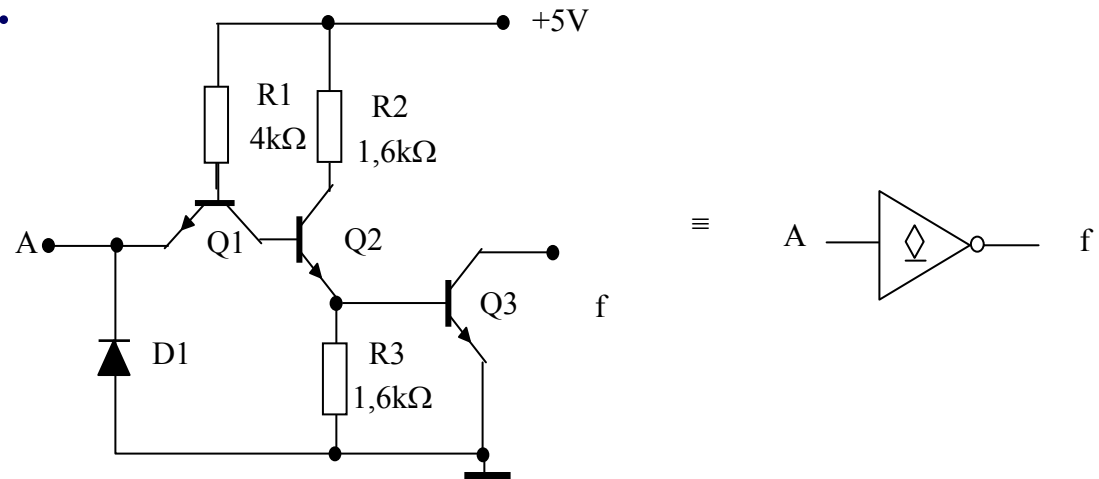


Sơ đồ mạch điện của một cổng OR TTL 2 lối vào.

- **Nguyên lý hoạt động của mạch vào này cũng giống với cổng NAND**

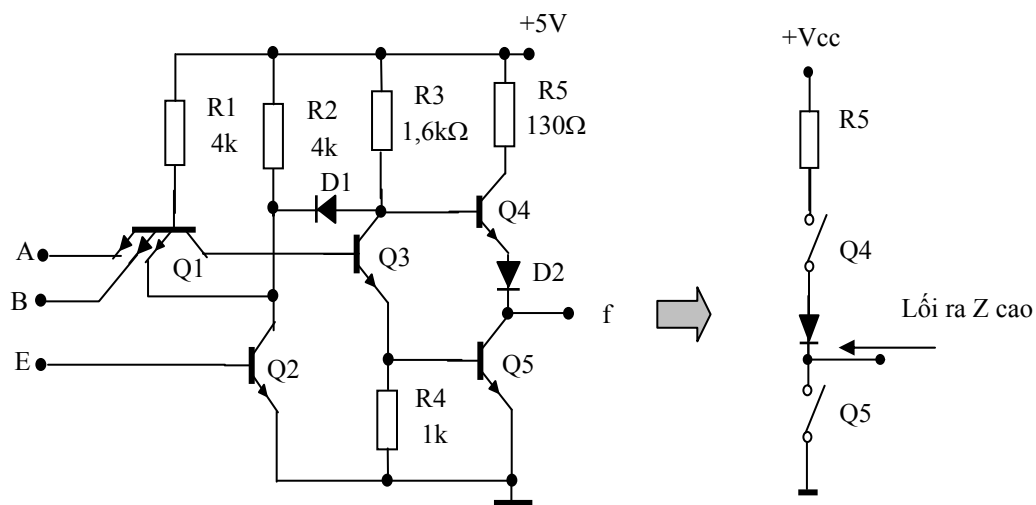
Mạch cổng collector để hở

- Nhược điểm của họ cổng TTL có mạch ra khép kín là hệ số tải đầu ra không thể thay đổi, nên nhiều khi gây khó khăn trong việc kết nối với đầu vào của các mạch điện tử tầng sau. Cổng logic collector để hở khắc phục được nhược điểm này.



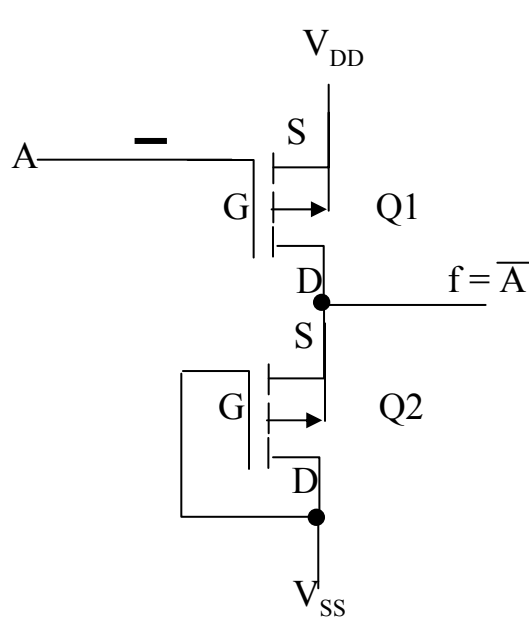
- Hình trên là sơ đồ của một cổng TTL đảo collector hở tiêu chuẩn. Muốn đưa cổng vào hoạt động, cần đấu thêm trở gánh ngoài, từ cực collector đến +Vcc.
- Một nhược điểm của cổng logic collector hở là tần số hoạt động của mạch sẽ giảm xuống do phải sử dụng điện trở gánh ngoài.

Mạch cổng TTL 3 trạng thái

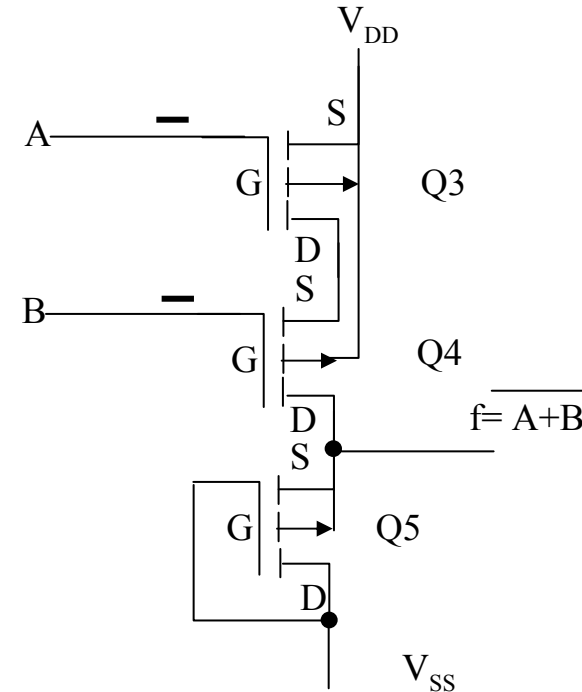


- **Bán dẫn trường (MOS FET) cũng được dùng rất phổ biến để xây dựng mạch điện các loại cổng logic. Đặc điểm chung và nổi bật của họ này là:**
 - Mạch điện chỉ bao gồm các MOS FET mà không có điện trở
 - Dải điện thế công tác rộng, có thể từ +3 đến +15 V
 - Độ trễ thời gian lớn, nhưng công suất tiêu thụ rất bé
- **Tuỳ theo loại MOS FET được sử dụng, họ này được chia ra các tiểu họ:**
 - PMOS
 - NMOS
 - CMOS
 - Cổng truyền dẫn

- Mạch điện của họ cổng này chỉ dùng MOSFET có kênh dẫn loại P. Công nghệ PMOS cho phép sản xuất các mạch tích hợp với mật độ cao nhất.
- Hình dưới là sơ đồ cổng NOT và cổng NOR loại PMOS. Ở đây MOSFET Q2, Q5 đóng chức năng các điện trở.

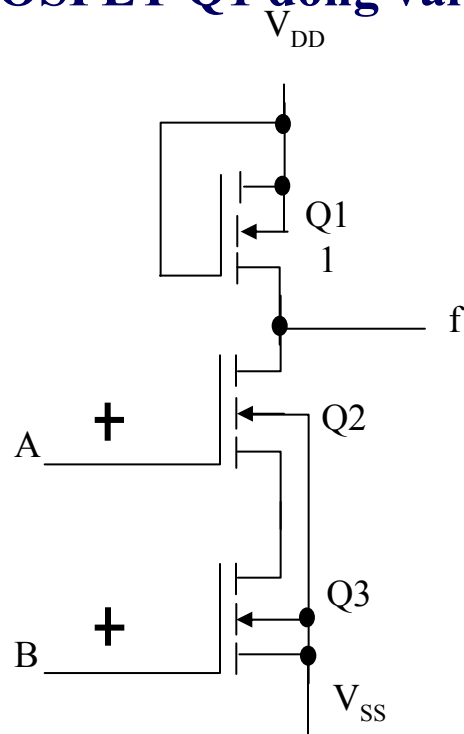


a) Cổng NOT

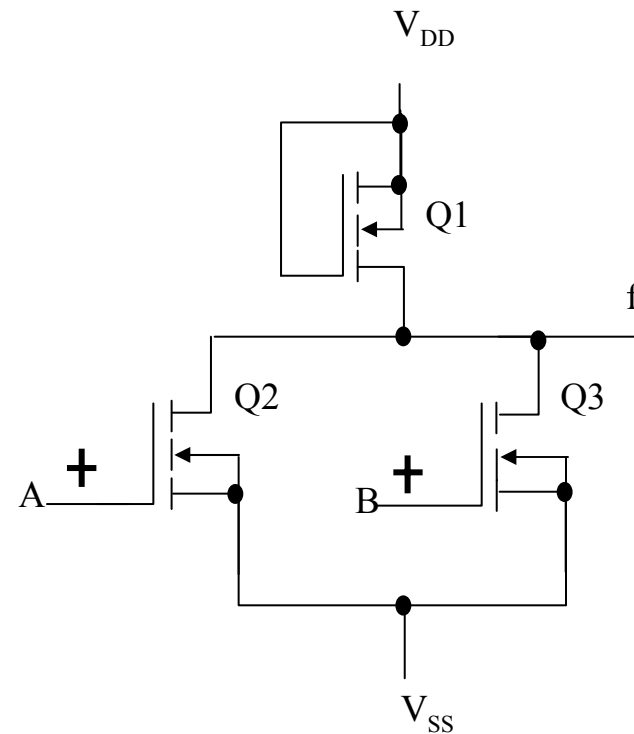


b) Cổng NOR

- Mạch điện của họ cổng này chỉ dùng MOSFET có kênh dẫn loại N.
- Hình dưới là sơ đồ cổng NAND và cổng NOR loại NMOS. Ở đây MOSFET Q1 đóng vai trò điện trở.

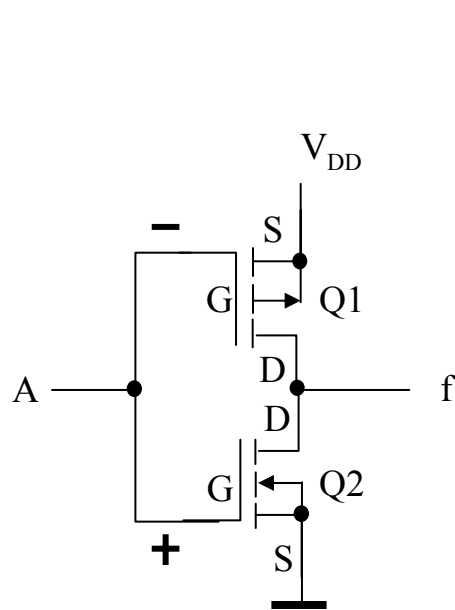


a) Cổng NAND

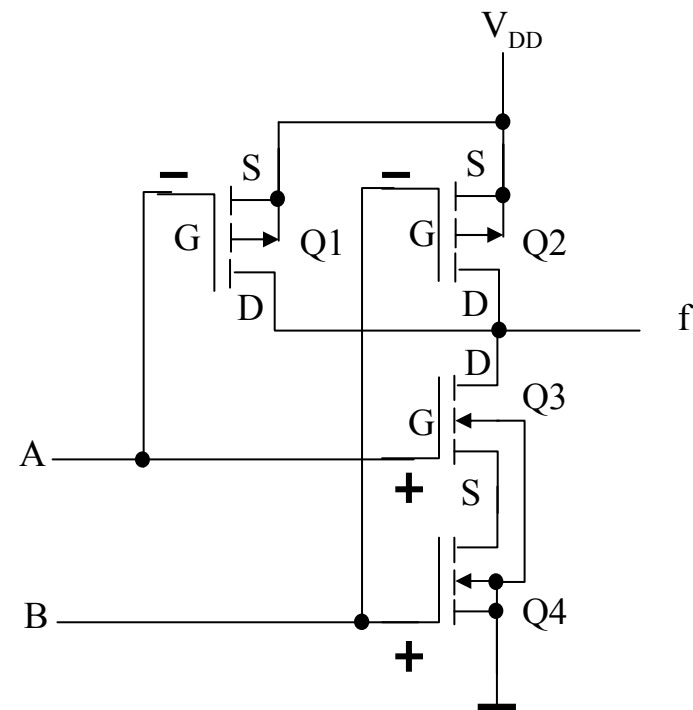


b) Cổng NOR

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



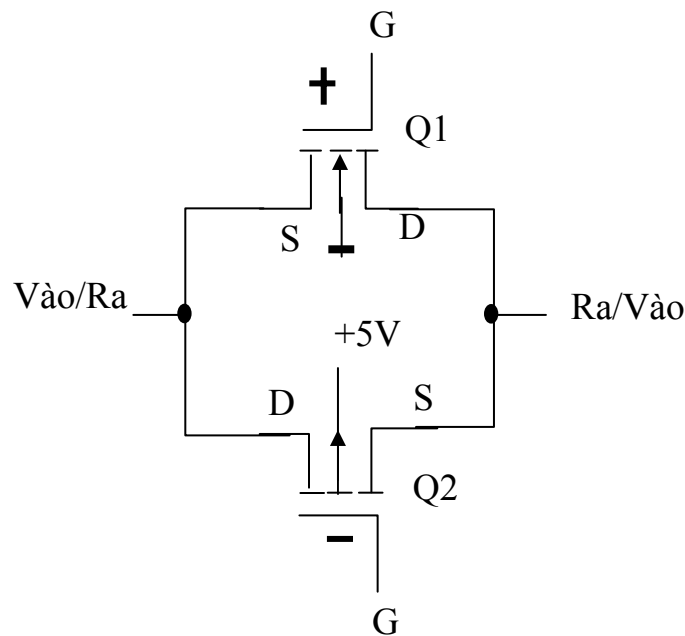
a) Cổng NOT



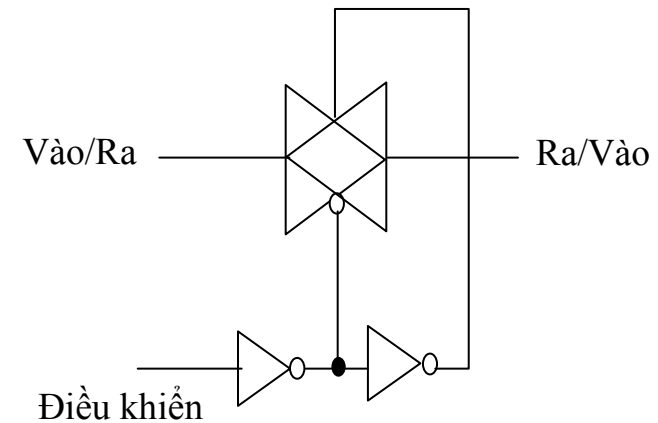
b) Cổng NAND

Cổng truyền dẫn

- Dựa trên công nghệ CMOS, người ta sản xuất loại cổng có thể cho qua cả tín hiệu số lẫn tín hiệu tương tự. Bởi vậy cổng được gọi là cổng truyền dẫn

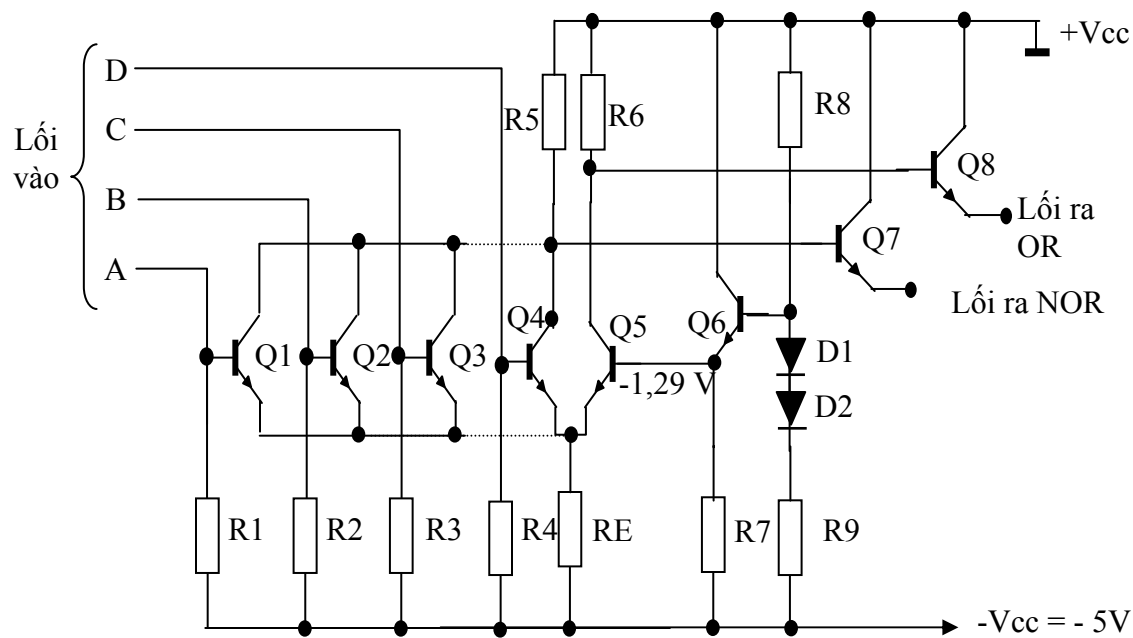


a) Mạch điện

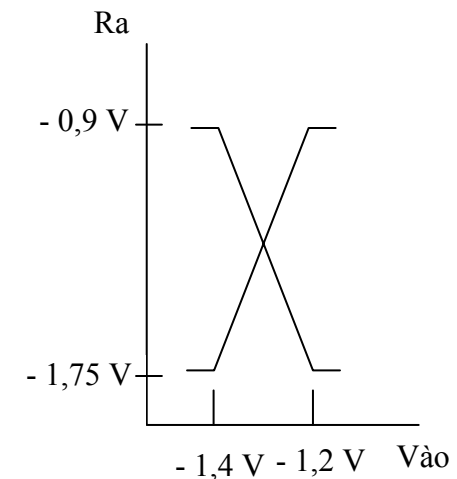


b) Ký hiệu

- ECL (Emitter Coupled Logic) là họ cổng logic có cực E của một số bán dẫn nối chung với nhau. Họ mạch này cũng sử dụng công nghệ TTL, nhưng cấu trúc mạch có những điểm khác hẳn với họ TTL.



a) Mạch điện nguyên lý



b) Đồ thị mức vào/ra



Giao tiếp giữa các cổng logic cơ bản

- **Giao tiếp giữa TTL và CMOS**
- **Giao tiếp giữa CMOS và TTL**



Câu hỏi



Nội dung

Chương 1: Hệ đếm

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

Chương 3: Cổng logic TTL và CMOS

■ **Chương 4: Mạch logic tổ hợp**

Chương 5: Mạch logic tuần tự

Chương 6: Mạch phát xung và tạo dạng xung

Chương 7: Bộ nhớ bán dẫn

Mạch logic tổ hợp



Nội dung

- **Khái niệm chung**
- **Phân tích mạch logic tổ hợp**
- **Thiết kế mạch logic tổ hợp**
- **Mạch mã hóa và giải mã**
- **Bộ hợp kênh và phân kênh**
- **Mạch số học**
- **Mạch tạo và kiểm tra chẵn lẻ**
- **Đơn vị số học và logic (ALU)**
- **Hazzards**

■ Đặc điểm cơ bản của mạch tổ hợp

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

■ Phương pháp biểu diễn chức năng logic

- Các phương pháp thường dùng để biểu diễn chức năng logic của mạch tổ hợp là hàm số logic, bảng trạng thái, bảng Cac nô (Karnaugh), cũng có khi biểu thị bằng đồ thị thời gian dạng xung.
- Đối với vi mạch cỡ nhỏ (SSI) thường biểu diễn bằng hàm logic. Đối với vi mạch cỡ vừa (MSI) thường biểu diễn bằng bảng trạng thái.

■ Phương pháp biểu diễn chức năng logic (tiếp)

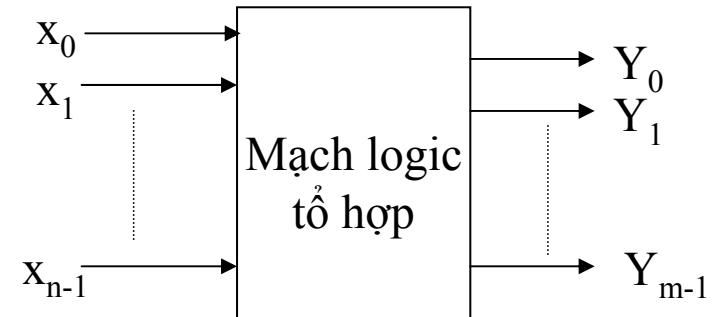
- Sơ đồ khối tổng quát của mạch logic tổ hợp được trình bày ở hình vẽ.
- Mạch logic tổ hợp có thể có n lối vào và m lối ra. Mỗi lối ra là một hàm của các biến vào. Quan hệ vào, ra này được thể hiện bằng hệ phương trình tổng quát sau:

$$Y_0 = f_0(x_0, x_1, \dots, x_{n-1});$$

$$Y_1 = f_1(x_0, x_1, \dots, x_{n-1});$$

...

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



- Đặc điểm nổi bật của mạch logic tổ hợp là hàm ra chỉ phụ thuộc các biến vào mà không phụ thuộc vào trạng thái của mạch. Cũng chính vì thế, trạng thái ra chỉ tồn tại trong thời gian có tác động vào.
- Thể loại của mạch logic tổ hợp rất phong phú. Phạm vi ứng dụng của chúng cũng rất rộng.

Phân tích mạch logic tổ hợp

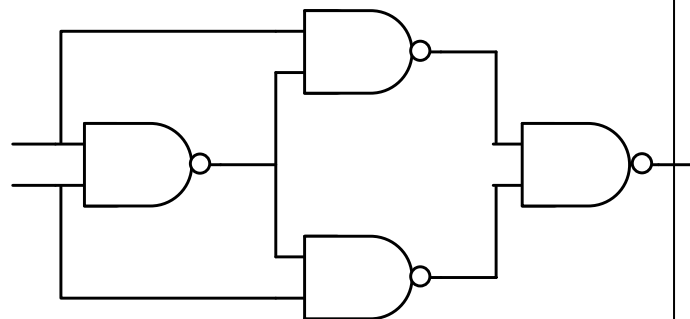
- **Định nghĩa:** là đánh giá, phê phán một mạch. Trên cơ sở đó, có thể rút gọn, chuyển đổi dạng thực hiện của mạch điện để có được lời giải tối ưu theo một nghĩa nào đấy.
- **Mạch tổ hợp có thể bao gồm hai hay nhiều tầng, mức độ phức tạp của của mạch cũng rất khác nhau. Thực hiện:**
 - Nếu mạch đơn giản thì ta tiến hành lập bảng trạng thái, viết biểu thức, rút gọn, tối ưu (nếu cần) và cuối cùng vẽ lại mạch điện.
 - Nếu mạch phức tạp thì ta tiến hành phân đoạn mạch để viết biểu thức, sau đó rút gọn, tối ưu (nếu cần) và cuối cùng vẽ lại mạch điện.

Ví dụ

A	B	$F = A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

Bảng 4-3. Bảng trạng thái mô tả hoạt động của hệ chiếu sáng

Hình 4-4. Sơ đồ logic thể hiện hàm f

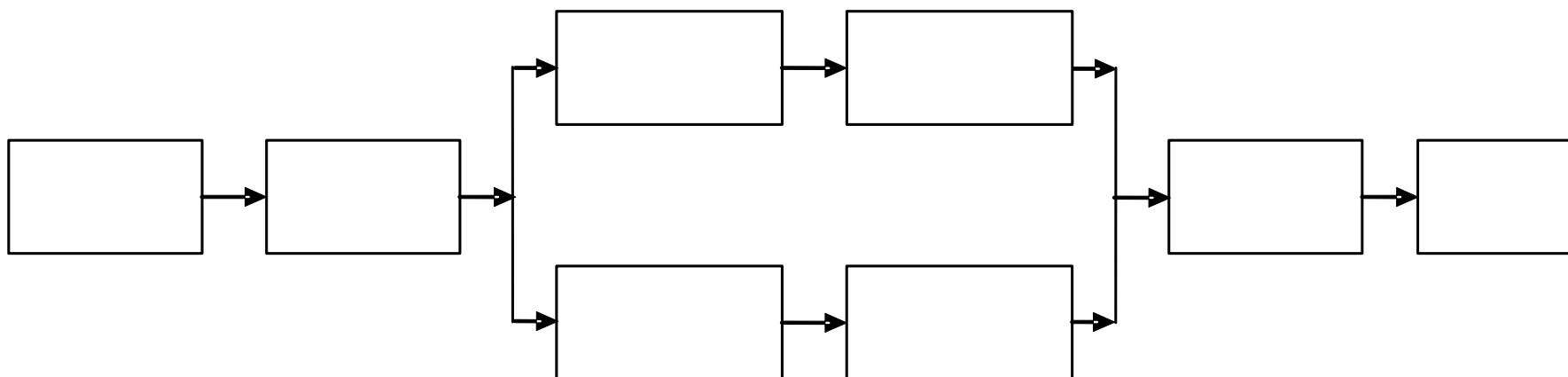


Thiết kế mạch logic tổ hợp

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

- 1. Phân tích bài toán đã cho để gán hàm và biến, xác lập mối quan hệ logic giữa hàm và các biến đó;**
- 2. Lập bảng trạng thái tương ứng;**
- 3. Từ bảng trạng thái có thể viết trực tiếp biểu thức đầu ra hoặc thiết lập bảng Các nô tương ứng;**
- 4. Dùng phương pháp thích hợp để rút gọn, đưa hàm về dạng tối giản hoặc tối ưu theo mong muốn;**
- 5. Vẽ mạch điện thể hiện.**

Các bước thiết kế mạch tổ hợp



Bảng
Karnaugh

Vấn đề logic thực
Bảng trạng thái

Thiết kế mạch logic tổ hợp

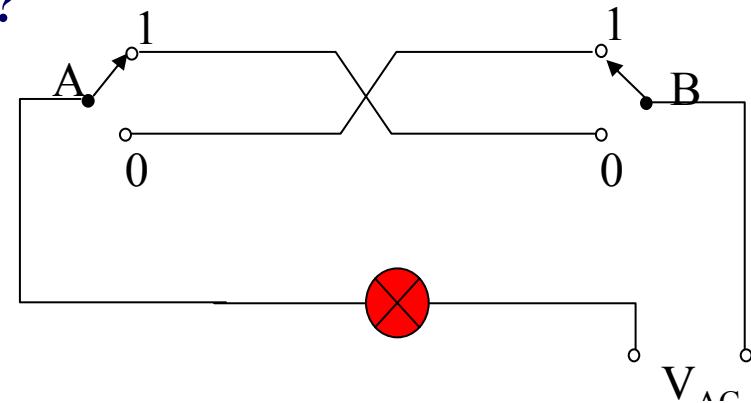
- Ví dụ:** Một ngôi nhà hai tầng. Người ta lắp hai chuyển mạch hai chiều tại hai tầng, sao cho ở tầng nào cũng có thể bật hoặc tắt đèn. Hãy thiết kế một mạch logic mô phỏng hệ thống đó?

- Lời giải:**

Hệ thống chiếu sáng như sơ đồ

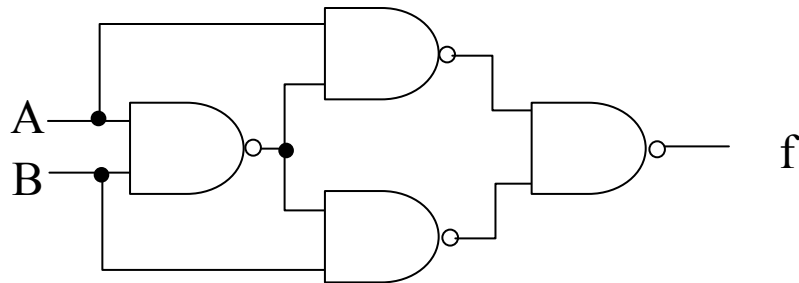
Biểu thức của hàm là:

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



Mạch điện của hệ thống chiếu sáng

Sơ đồ logic thể hiện hàm f



Bảng trạng thái

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

- Mã hoá là dùng văn tự, ký hiệu hay mã để biểu thị một đối tượng xác định.
- Bộ mã hoá là mạch điện thao tác mã hoá, có nhiều bộ mã hoá khác nhau, bộ mã hoá nhị phân, bộ mã hoá nhị - thập phân, bộ mã hoá ưu tiên v.v.
- Mã nhị phân n bit có 2^n trạng thái, có thể biểu thị 2^n tín hiệu. Vậy để tiến hành mã hoá N tín hiệu, cần sử dụng n bit sao cho $2^n \geq N$.
- Một số loại mã thông dụng
 - Mã BCD và mã dư 3
 - Mã Gray
 - Mã chẵn, lẻ
- Mạch mã hoá
 - Mạch mã hoá từ thập phân sang BCD 8421
 - Mạch mã hoá ưu tiên



Mạch giải mã

- Giải mã là một quá trình phiên dịch hàm đã được gán bằng một từ mã.
- Mạch điện thực hiện giải mã gọi là bộ giải mã.
- Bộ giải mã biến đổi từ mã thành tín hiệu ở đầu ra.
- Mạch giải mã
 - Mạch giải mã 7 đoạn
 - Mạch giải mã nhị phân

■ MÃ BCD (Binary Coded Decimal)

- Cấu tạo: dùng từ nhị phân 4 bit để mã hóa 10 kí hiệu thập phân, nhưng cách biểu diễn vẫn theo thập phân.
- Ví dụ đối với mã NBCD, các chữ số thập phân được nhị phân hoá theo trọng số như nhau $2^3, 2^2, 2^1, 2^0$ nên có 6 tổ hợp dư, ứng với các số thập phân 10, 11, 12, 13, 14 và 15.
- Ứng dụng: Do trọng số nhị phân của mỗi vị trí biểu diễn thập phân là tự nhiên nên máy có thể thực hiện trực tiếp các phép tính cộng, trừ, nhân, chia theo mã NBCD.
- Nhược điểm chính của mã là tồn tại tổ hợp toàn Zero, gây khó khăn trong việc đồng bộ khi truyền dẫn tín hiệu.

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

■ Mã Dư-3

- Cấu tạo: được hình thành từ mã NBCD bằng cách cộng thêm 3 vào mỗi tổ hợp mã. Như vậy, mã không bao gồm tổ hợp toàn Zero.
- Ứng dụng để truyền dẫn tín hiệu mà không dùng cho việc tính toán trực tiếp.

- Mã Gray còn được gọi là mã cách 1, là loại mã mà các tổ hợp mã kế nhau chỉ khác nhau duy nhất 1 bit. Loại mã này không có tính trọng số. Do đó, giá trị thập phân đã được mã hóa chỉ được giải mã thông qua bảng mã mà không thể tính theo tổng trọng số như đối với mã BCD.
- Mã Gray có thể được tổ chức theo nhiều bit. Bởi vậy, có thể đếm theo mã Gray.
- Cũng tương tự như mã BCD, ngoài mã Gray chính còn có mã Gray dư-3.

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

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

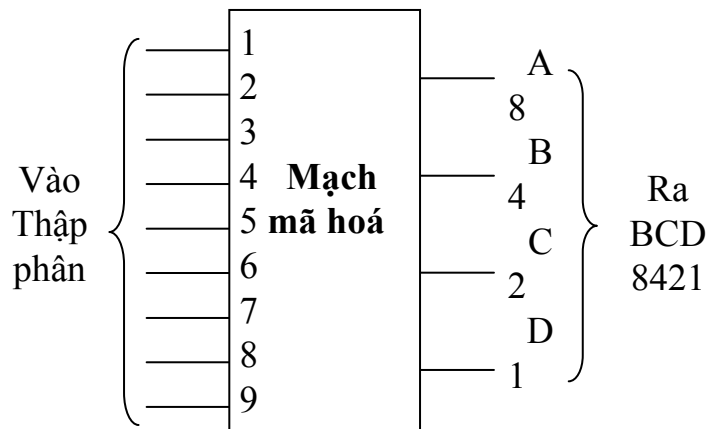
BCD 8421	BCD 8421chẵn		BCD 8421 lẻ	
		P_C		P_L
0000	0000	0	0000	1
0001	0001	1	0001	0
0010	0010	1	0010	0
0011	0011	0	0011	1
0100	0100	1	0100	0
0101	0101	0	0101	1
0110	0110	0	0110	1
0111	0111	1	0111	0
1000	1000	1	1000	0
1001	1001	0	1001	1

Mạch mã hoá từ thập phân sang BCD 8421

Sơ đồ khối tổng quát của mạch mã hoá

- Gồm 9 lối vào (biến) ứng Với các chữ số thập phân từ 1 đến 9. Lối vào zero là không cần thiết, vì khi tất cả các lối vào khác bằng 0 thì lối ra cũng bằng 0.
- Bốn lối ra A, B, C, D (hàm) thể hiện tổ hợp mã tương ứng với mỗi chữ số thập phân trên lối vào theo trọng số 8421.

Sơ đồ khối của mạch mã hóa



Bảng trạng thái

Vào thập phân	Ra BCD 8 4 2 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

Từ bảng trạng thái ta viết được các hàm ra:

$$\begin{aligned}
 A &= 8 + 9 &&= \Sigma (8,9) \\
 B &= 4 + 5 + 6 + 7 &&= \Sigma (4,5,6,7) \\
 C &= 2 + 3 + 6 + 7 &&= \Sigma (2,3,6,7) \\
 D &= 1 + 3 + 5 + 7 + 9 &&= \Sigma (1,3,5,7,9)
 \end{aligned}$$

Bảng mã hoá BCD – 8421

Số thập phân	A	B	C	D
0 (Y_0)	0	0	0	0
1 (Y_1)	0	0	0	1
2 (Y_2)	0	0	1	0
3 (Y_3)	0	0	1	1
4 (Y_4)	0	1	0	0
5 (Y_5)	0	1	0	1
6 (Y_6)	0	1	1	0
7 (Y_7)	0	1	1	1
8 (Y_8)	1	0	0	0
9 (Y_9)	1	0	0	1

Bảng 4-4. Bảng mã hoá BCD – 8421:

Mạch mã hoá từ thập phân sang BCD 8421

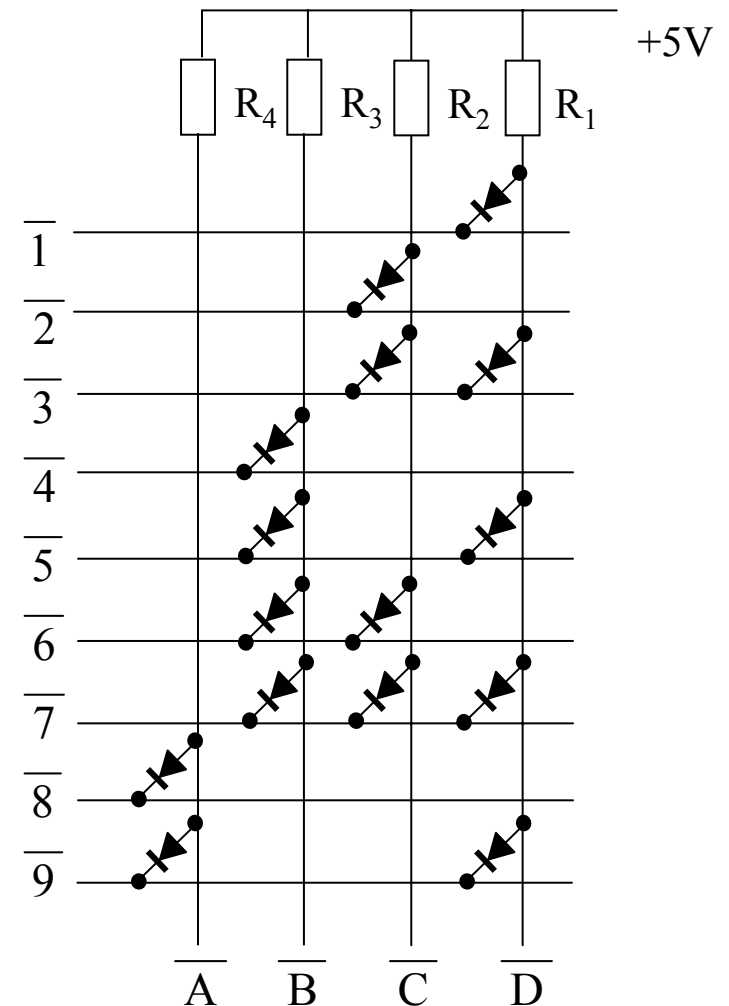
- Căn cứ hệ phương trình, ta xây dựng được mạch điện của bộ mã hoá. Hoặc dùng ma trận diode (công OR) để xây dựng
- Hoặc có thể được viết lại như sau (dùng định lý DeMorgan) và dùng ma trận diode (công AND) để xây dựng mạch:

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

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

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

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



Mạch điện của bộ mã hoá dùng diode

Mạch mã hóa ưu tiên

- Trong bộ mã hoá vừa xét trên, tín hiệu vào tồn tại độc lập, (không có trường hợp có 2 tổ hợp trở lên đồng thời tác động).
- Để giải quyết trường hợp có nhiều đầu vào tác động đồng thời ta có Bộ mã hoá ưu tiên. Trong các trường hợp này thì bộ mã hoá ưu tiên chỉ tiến hành mã hoá tín hiệu vào nào có cấp ưu tiên cao nhất ở thời điểm xét. Việc xác định cấp ưu tiên cho mỗi tín hiệu vào là do người thiết kế mạch.
- Xét nguyên tắc hoạt động và quá trình thiết kế của bộ mã hoá ưu tiên 9 lỗi vào, 4 lỗi ra.

Mạch mã hóa ưu tiên (tiếp)

- D sẽ lấy logic 1 ứng với đầu vào là 1, 3, 5, 7, 9. Tuy nhiên, lỗi vào 1 chỉ hiệu lực khi tất cả các lỗi vào cao hơn đều bằng 0; lỗi vào 3 chỉ có hiệu lực khi 4, 6, 8 đều bằng 0 và tương tự đối với 5, 7, 9, nghĩa là:

D="1" nếu {
 1="1" và 2,4,6,8 bằng "0"
 3="1" và 4,6,8 bằng "0"
 5="1" và 6,8 bằng "0"
 7="1" và 8 bằng "0"
 9="1"

$$\Rightarrow D = 1.\bar{2}.\bar{4}.\bar{6}.\bar{8} + 3.\bar{4}.\bar{6}.\bar{8} + 5.\bar{6}.\bar{8} + 7.\bar{8} + 9$$

- Lý luận tương tự ta có:

$$C = 2.\bar{4}.\bar{5}.\bar{8}.\bar{9} + 3.\bar{4}.\bar{5}.\bar{8}.\bar{9} + 6.\bar{8}.\bar{9} + 7.\bar{8}.\bar{9}$$

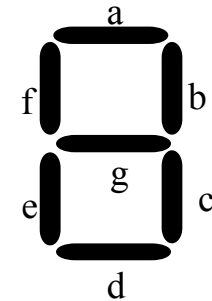
$$B = 4.\bar{8}.\bar{9} + 5.\bar{8}.\bar{9} + 6.\bar{8}.\bar{9} + 7.\bar{8}.\bar{9}$$

$$A = 8 + 9$$

Vào thập phân									Ra BCD			
1	2	3	4	5	6	7	8	9	8	4	2	1
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1
X	1	0	0	0	0	0	0	0	0	0	1	0
X	X	1	0	0	0	0	0	0	0	0	1	1
X	X	X	1	0	0	0	0	0	0	1	0	0
X	X	X	X	1	0	0	0	0	0	1	0	1
X	X	X	X	X	1	0	0	0	0	1	1	0
X	X	X	X	X	X	1	0	0	0	1	1	1
X	X	X	X	X	X	X	1	0	1	0	0	0
X	X	X	X	X	X	X	X	1	1	0	0	1

■ Dụng cụ 7 đoạn

- Để hiển thị chữ số của một hệ đếm phân bất kỳ, ta có thể dùng dụng cụ 7 đoạn. Cấu tạo của nó như chỉ ở hình 4-15.
- Các đoạn được hình thành bằng nhiều loại vật liệu khác nhau, nhưng phải có khả năng hiển thị được trong các điều kiện ánh sáng khác nhau và tốc độ chuyển mạch phải đủ lớn. Trong kỹ thuật số, các đoạn thường được dùng là LED hoặc tinh thể lỏng (LCD).
- Đối với LED, mỗi đoạn là một Diode phát quang và khi có dòng điện đi qua đủ lớn (5 đến 30 mA) thì đoạn tương ứng sẽ sáng.
- Ngoài 7 đoạn sáng chính, mỗi LED cũng có thêm Diode để hiển thị dấu phân số khi cần thiết. LED có hai loại chính: LED Anôt chung và Ktôt chung. Do đó, logic của tín hiệu điều khiển hai loại này là ngược nhau.



Cấu tạo dụng cụ 7 đoạn sáng

Mạch giải mã 7 đoạn

■ Mạch giải mã 7 đoạn

- Nhiệm vụ của ta là phải thiết kế một mạch logic liên hợp với 4 lối vào và 7 lối ra để chuyển mã NBCD thành mã 7 đoạn.
- Sơ đồ khối tổng quát của bộ giải mã như hình b).
- Từ hình a) dễ nhận thấy rằng, đoạn a sẽ sáng khi hiển thị chữ số : 0 hoặc 2, hoặc 3, hoặc 5, hoặc 7, hoặc 8, hoặc 9. Do đó, ta có thể viết:

$$\blacklozenge a = \sum (0,2,3,5,6,7,8,9).$$

- Tương tự, ta có:

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

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

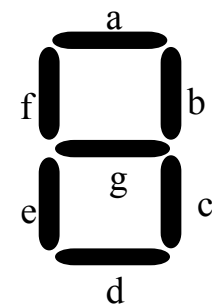
$$\blacklozenge d = \sum (0,2,3,5,6,8,9),$$

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

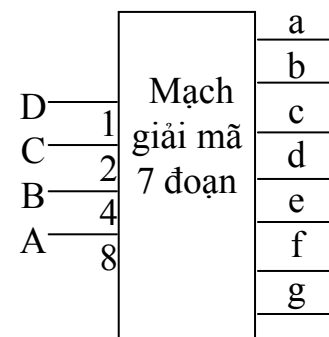
$$\blacklozenge f = \sum (0,4,5,6,8,9),$$

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

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



a) Cấu tạo dụng cụ 7 đoạn sáng



b) Sơ đồ khối của mạch giải mã 7 đoạn sáng

Mạch giải mã nhị phân

- Bộ giải mã nhị phân còn có tên là bộ giải mã "*1* từ *n*", bộ giải mã địa chỉ hoặc bộ chọn địa chỉ nhị phân. Chức năng của nó là lựa chọn duy nhất một lối ra (lấy giá trị 1 hoặc 0), khi tác động tới đầu vào một số nhị phân.
- Như vậy, nếu số nhị phân là *n* bit (*n* lối vào) sẽ nhận diện được 2^n địa chỉ khác nhau (trên 2^n lối ra). Nói khác đi, mạch chọn địa chỉ nhị phân là một mạch logic tổ hợp có *n* lối vào và 2^n lối ra, nếu tác động tới đầu vào một số nhị phân thì chỉ duy nhất một lối ra được lựa chọn, lấy giá trị 1 (tích cực cao) hoặc 0 (tích cực thấp), các lối ra còn lại đều không được lựa chọn, lấy giá trị 0 hoặc 1.



Sơ đồ khối của bộ giải mã nhị phân

■ Bộ hợp kênh (MUX-Multiplexer)

- **Định nghĩa:** Bộ hợp kênh là mạch có 2^n lối vào dữ liệu, n lối vào điều khiển, 1 lối vào chọn mạch và 1 lối ra.
- Tùy theo giá trị của n lối vào điều khiển mà lối ra sẽ bằng một trong những giá trị ở lối vào (X_j). Nếu giá trị thập phân của n lối vào điều khiển bằng j thì $Y = X_j$.

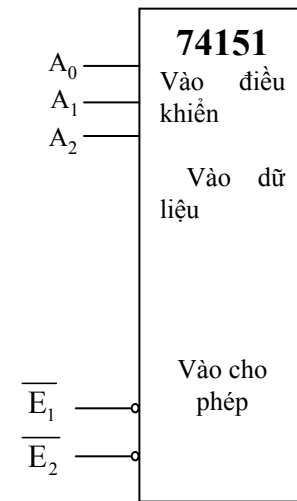
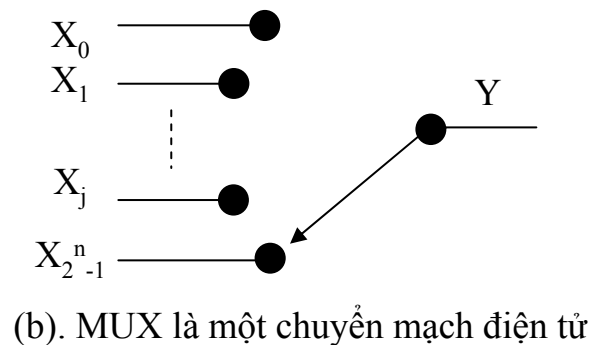
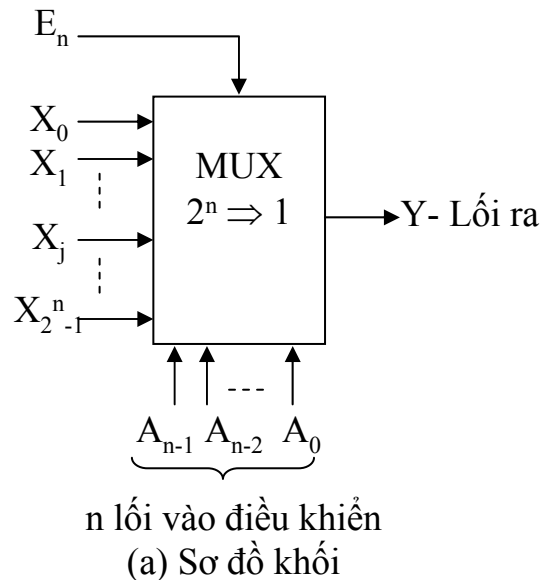
■ Bộ phân kênh (DEMUX-DeMultiplexer)

- **Định nghĩa:** Bộ phân kênh là mạch có 1 lối vào dữ liệu, n lối vào điều khiển, 1 lối vào chọn mạch và 2^n lối ra.
- Tùy theo giá trị của n lối vào điều khiển mà lối ra thứ i (Y_i) sẽ bằng giá trị của lối vào. Cụ thể nếu gọi n lối vào điều khiển là $A_{n-1}A_{n-2}\dots A_0$ thì $Y_i = X$ khi $(A_{n-1}A_{n-2}\dots A_0)_2 = (i)_{10}$.

Bộ hợp kênh (MUX-Multiplexer)

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

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



Bộ hợp kênh MUX $2^n \Rightarrow 1$

- Thực chất, MUX là chuyển mạch điện tử dùng các tín hiệu điều khiển ($A_{n-1} A_{n-2} \dots A_0$) để điều khiển sự nối mạch của lối ra với 1 trong số 2^n lối vào.
- MUX được dùng như 1 phần tử vạn năng để xây dựng những mạch tổ hợp khác.
- IC 74151 là bộ MUX 8 lối vào dữ liệu - 1 lối ra.

Bộ phân kênh (DEMUX-DeMultiplexer)

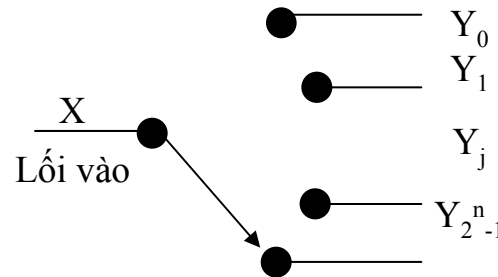
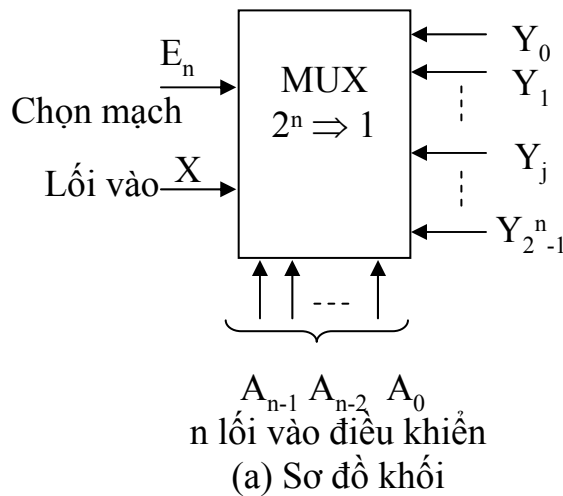
- Phương trình tín hiệu ra của DEMUX $1 \Rightarrow 2^n$:

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

$$Y_1 = X \cdot \bar{A}_{n-1} \bar{A}_{n-2} \dots \bar{A}_i \dots A_1 A_0$$

.....

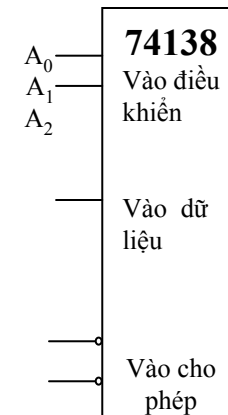
$$Y_{2^n-1} = X \cdot A_{n-1} \cdot A_{n-2} \dots A_i \dots A_0$$



(b). DEMUX là một chuyển mạch điện tử

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

- Bộ phân kênh còn được gọi là bộ giải mã 1 trong 2^n . Tại một thời điểm chỉ có 1 trong số 2^n lối ra ở mức tích cực.
- IC 74138 là bộ DEMUX 1 lối vào dữ liệu - 8 lối ra.



Mạch cộng: Mạch toàn tổng

- **Định nghĩa:** Mạch logic thực hiện phép cộng hai số nhị phân 1 bit có lỗi nhớ đầu vào được gọi là mạch toàn tổng.
- Theo sơ đồ khối tổng quát của mạch toàn tổng và nguyên lý cộng hai số nhị phân một bit có trọng số bất kỳ, ta có thể lập bảng trạng thái và các hàm ra S_i , C_i .

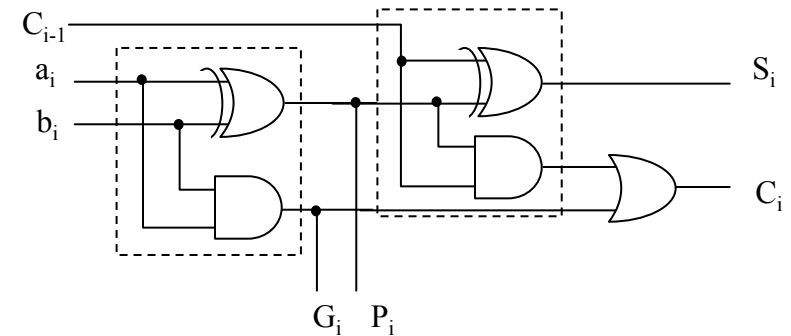
$$S_i = a_i \oplus b_i \oplus C_{i-1}$$

$$C_i = a_i b_i C_{i-1} + \bar{a}_i b_i C_{i-1} + a_i \bar{b}_i C_{i-1}$$

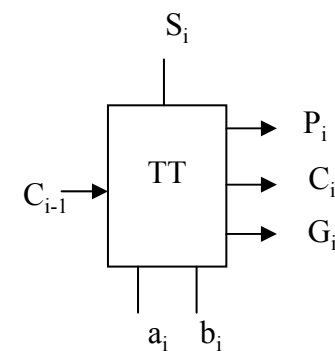
$$C_i = a_i b_i + (a_i \oplus b_i) C_{i-1}$$

Bảng trạng thái

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



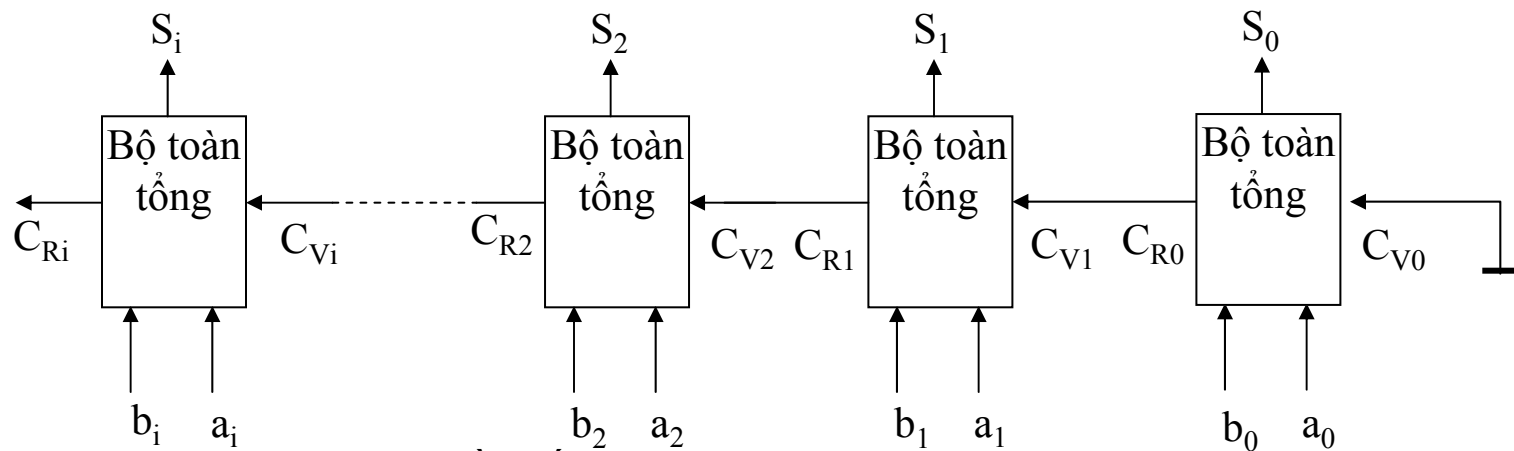
a) Mạch điện



b) Ký hiệu

Mạch cộng: Mạch cộng nhị phân song song

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



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

- Để giảm bớt mức độ phức tạp của mạch, trong thực tế người ta thường sản xuất bộ tổng 4 bit. Muốn cộng nhiều bit, có thể hợp nối tiếp một vài bộ tổng một bit theo phương pháp nêu trên.
- Một trong những bộ cộng thông dụng hiện nay là 7483. IC này được sản xuất theo hai loại: 7483 và 7483A với logic vào, ra khác nhau.

- Trong các hệ thống số, đặc biệt là trong máy tính, thường thực hiện việc so sánh hai số.
- Hai số cần so sánh có thể là các số nhị phân, có thể là các ký tự đã mã hoá nhị phân.
- Mạch so sánh có thể hoạt động theo kiểu nối tiếp hoặc theo kiểu song song. Trong phần này ta sẽ nghiên cứu bộ so sánh theo kiểu song song.
 - Bộ so sánh bằng nhau
 - ◆ Bộ so sánh bằng nhau 1 bit
 - ◆ Bộ so sánh bằng nhau 4 bit
 - Bộ so sánh
 - ◆ Bộ so sánh 1 bit
 - ◆ Bộ so sánh 4 bit (So sánh lớn hơn)

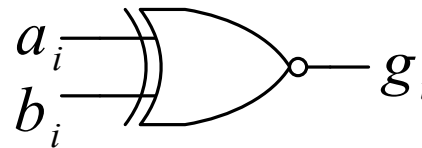
Bộ so sánh bằng nhau

■ Bộ so sánh bằng nhau 1 bit

- Xét 2 bit a_i và b_i , gọi g_i là kết quả so sánh.
- Ta có:

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

Sơ đồ logic của hàm ra
bộ so sánh bằng 1 bit



Bảng trạng thái của bộ
so sánh bằng 1 bit

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

■ Bộ so sánh bằng nhau 4 bit

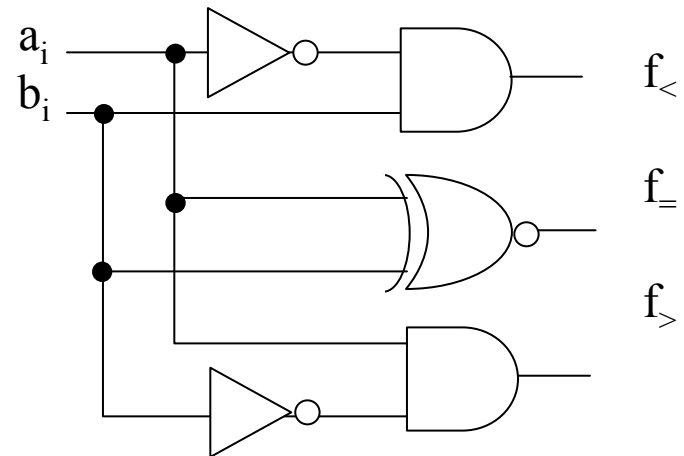
- So sánh hai số nhị phân 4 bit $A = a_3a_2a_1a_0$ với $B = b_3b_2b_1b_0$.
Có $A = B \Leftrightarrow a_3 = b_3, a_2 = b_2, a_1 = b_1, a_0 = b_0$.
- Biểu thức đầu ra tương ứng là: $G = g_3g_2g_1g_0$ với:

$$g_3 = \overline{a_3 \oplus b_3}, \quad g_2 = \overline{a_2 \oplus b_2}, \quad g_1 = \overline{a_1 \oplus b_1}, \quad g_0 = \overline{a_0 \oplus b_0}$$

Bảng trạng thái của mạch so sánh

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

Mạch điện của bộ so sánh 1 bit



- Biểu thức đầu ra:**

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

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

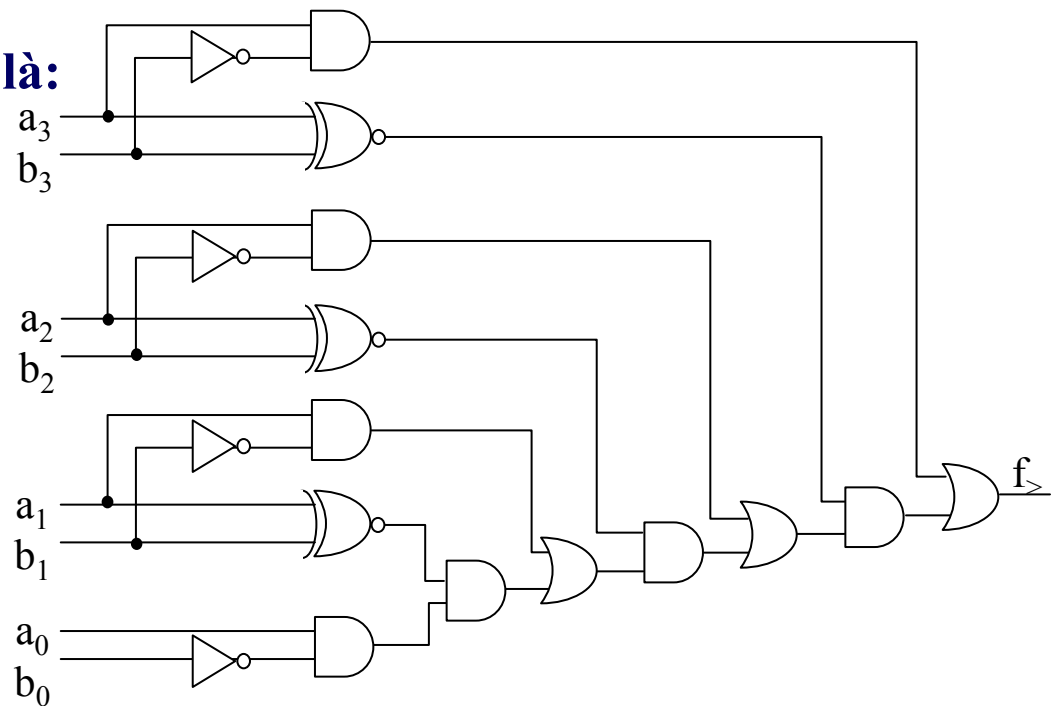
$$f_{>} = a_i \cdot \overline{b_i}$$

Bộ so sánh 4 bit (So sánh lớn hơn)

- So sánh hai số nhị phân 4 bit $A = a_3a_2a_1a_0$ với $B = b_3b_2b_1b_0$. Có $A > B$ khi:
 - hoặc $a_3 > b_3$,
 - hoặc $a_3 = b_3$, và $a_2 > b_2$,
 - hoặc $a_3 = b_3$, và $a_2 = b_2$, và $a_1 > b_1$,
 - hoặc $a_3 = b_3$, và $a_2 = b_2$, và $a_1 = b_1$, và $a_0 > b_0$.

- Từ đó ta có biểu thức hàm ra là:

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

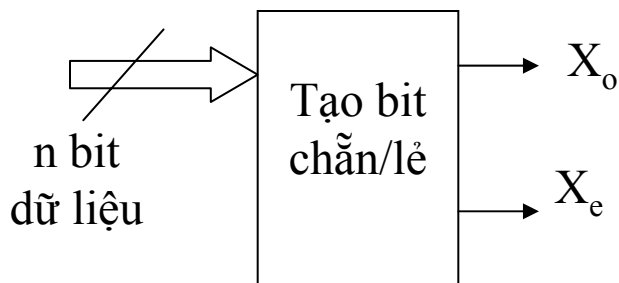


Mạch điện của bộ so sánh lớn hơn 4 bit

Mạch tạo và kiểm tra chẵn lẻ

- Có nhiều phương pháp mã hoá dữ liệu để phát hiện lỗi và sửa lỗi khi truyền dữ liệu từ nơi này sang nơi khác. Phương pháp đơn giản nhất là thêm một bit vào dữ liệu được truyền đi sao cho số chữ số 1 trong dữ liệu luôn là chẵn hoặc lẻ. Bit thêm vào đó được gọi là bit chẵn/lẻ.
- Để thực hiện được việc truyền dữ liệu theo kiểu đưa thêm bit chẵn, lẻ vào dữ liệu chúng ta phải:
 - Xây dựng sơ đồ tạo được bit chẵn, lẻ để thêm vào n bit dữ liệu.
 - Xây dựng sơ đồ kiểm tra hệ xem đó là hệ chẵn hay lẻ với $(n + 1)$ bit ở đầu vào (n bit dữ liệu, 1 bit chẵn/lẻ).

Mạch tạo bit chẵn/lẻ



Sơ đồ khối tổng quát của mạch tạo bit chẵn/lẻ

- Xét trường hợp 3 bit dữ liệu d_1, d_2, d_3
- Gọi X_e, X_o là 2 bit chẵn, lẻ thêm vào dữ liệu.

- Từ bảng trạng thái ta thấy

$$X_o = \overline{X_e} \text{ hay } X_e = \overline{X_o}$$

- Và biểu thức của X_o và X_e là

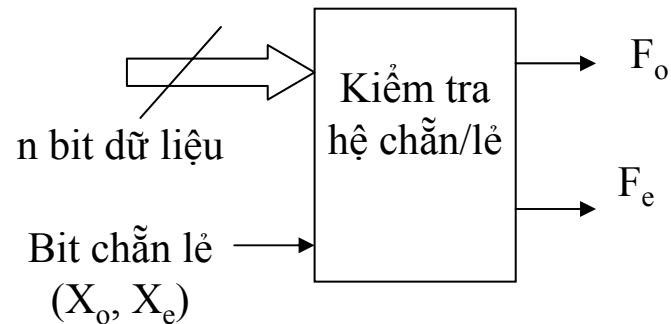
$$X_e = d_1 \oplus d_2 \oplus d_3$$

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

Bảng trạng thái của mạch tạo bit chẵn lẻ

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

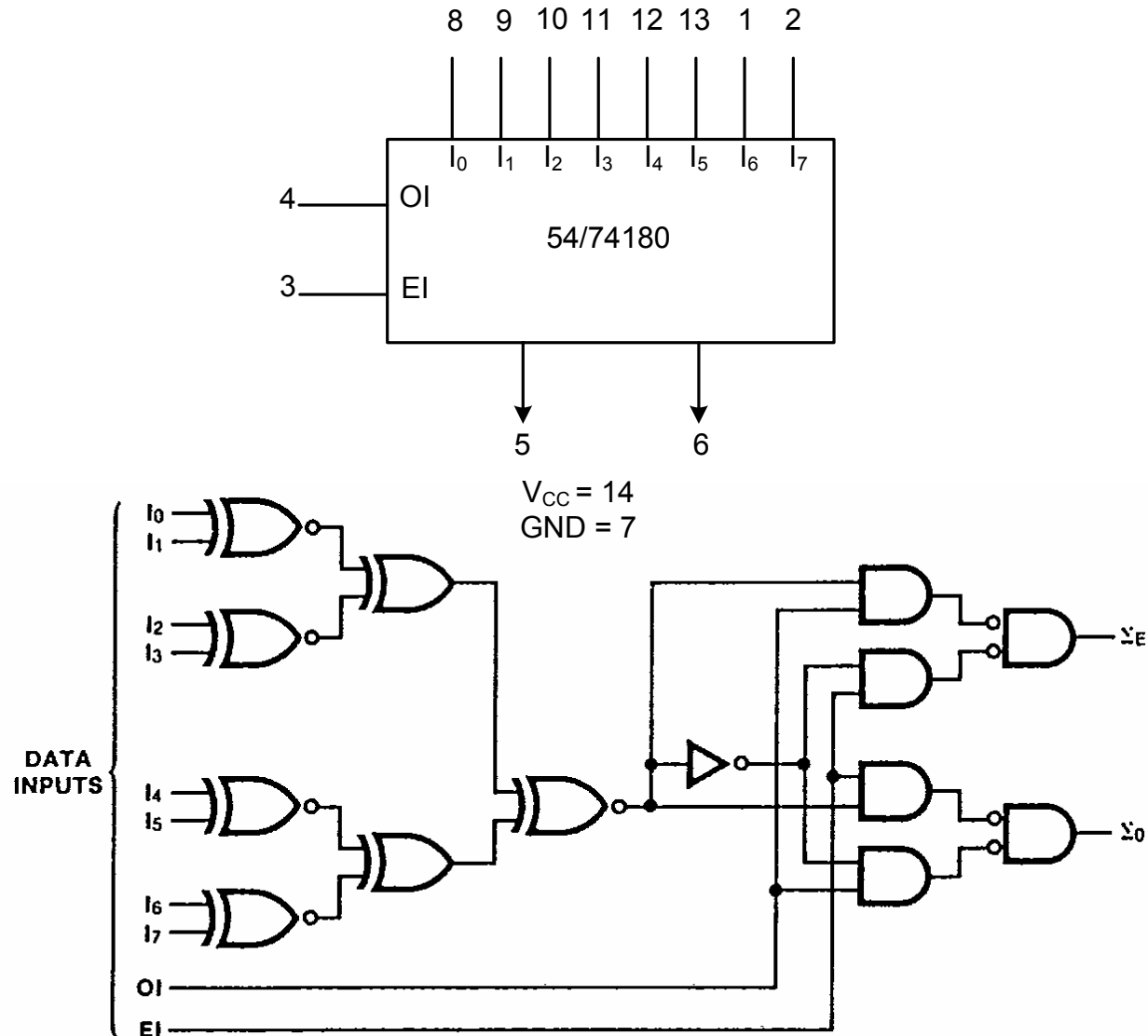
Mạch kiểm tra chẵn/lẻ



Sơ đồ khối của mạch kiểm tra chẵn/lẻ

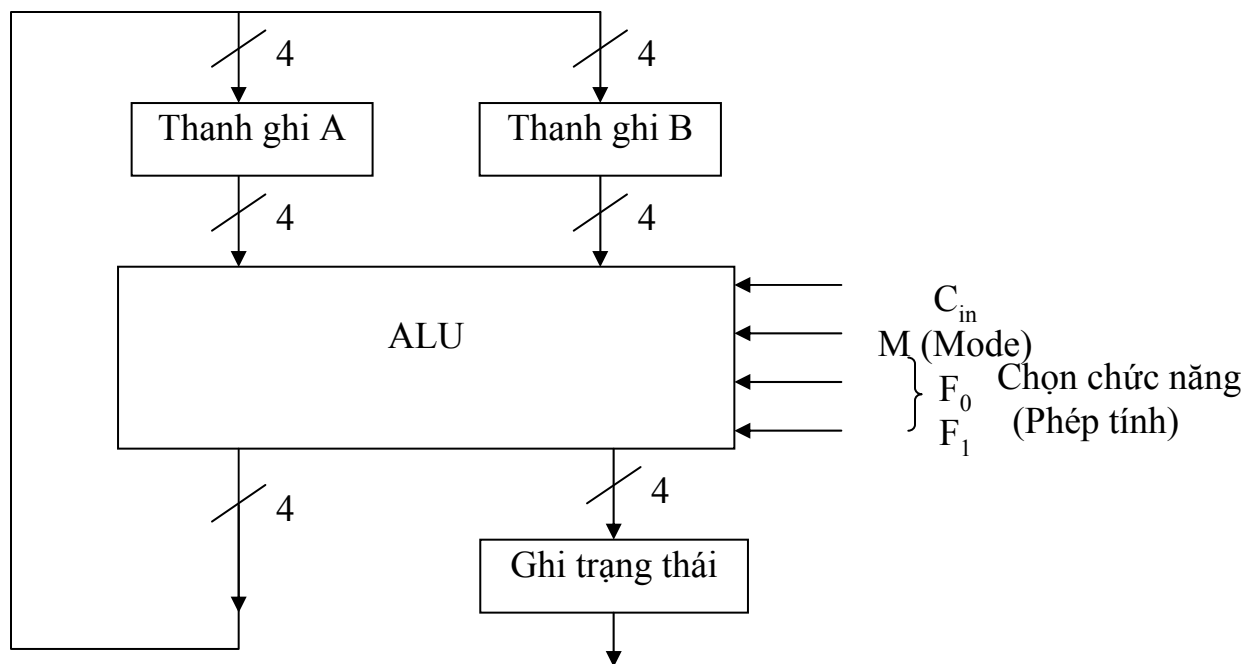
- **Từ bảng trạng thái của mạch kiểm tra tính chẵn/lẻ ta thấy:**
 - $F_e = 1$ nếu hệ là chẵn (F_e chỉ ra tính chẵn của hệ).
 - $F_o = 1$ nếu hệ là lẻ (F_o chỉ ra tính lẻ của hệ).
- **Hai hàm kiểm tra chẵn/lẻ luôn là phủ định của nhau. Mặt khác do tính chất của hàm cộng XOR, ta có:**
 - $F_o = d_1 \oplus d_2 \oplus d_3 \oplus X$
 - $F_e = \overline{F_o}$

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



Σ_E

Đơn vị số học và logic (ALU)



Sơ đồ khối của ALU 4 bit

- **Đơn vị số học và logic (Arithmetic – Logic Unit) là một thành phần cơ bản không thể thiếu được trong các máy tính. Nó bao gồm 2 khối chính là khối logic và khối số học và một khối ghép kênh.**
 - Khối logic: Thực hiện các phép tính logic như là AND, OR, NOT, XOR.
 - Khối số học: Thực hiện các phép tính số học như là: cộng, trừ, tăng 1, giảm 1.

- Hazard còn được gọi là sự "sai nhầm", hoạt động lúc được lúc không của mạch logic.
- Sự "sai nhầm" này có thể xảy ra trong một mạch điện hoàn toàn không bị hỏng linh kiện làm cho mạch hoạt động không có sự tin cậy.
- **Hiện tượng của Hazard trong mạch tổ hợp có thể gặp là:**
 - Hazard chỉ xuất hiện một lần và không bao giờ gặp lại nữa.
 - Hazard có thể xuất hiện nhiều lần (theo một chu kỳ nào đó hoặc không theo một chu kỳ nào).
 - Hazard có thể do chính chức năng của mạch điện gây ra. Đây là trường hợp khó giải quyết nhất khi thiết kế.

- **Bản chất của hazzards**

- Do sự chạy đua giữa các tín hiệu
- VD: demo trên Logicworks

- **Phân loại hazzard**

- Hazzard tĩnh : Đầu ra chỉ xuống 0 hoặc 1 một lần
- Hazzard động : Đầu ra có thể thay đổi nhiều hơn 1 lần



Nội dung

Chương 1: Hệ đếm

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

Chương 3: Cổng logic TTL và CMOS

Chương 4: Mạch logic tổ hợp

▪ **Chương 5: Mạch logic tuần tự**

Chương 6: Mạch phát xung và tạo dạng xung

Chương 7: Bộ nhớ bán dẫn

Mạch logic tuần tự

- **Khái niệm chung**
- **Phần tử nhớ trong mạch tuần tự**
- **Phương pháp mô tả mạch tuần tự**
- **Phân tích và thiết kế mạch tuần tự**
- **Mạch tuần tự đồng bộ**
- **Mạch tuần tự không đồng bộ**
- **Hiện tượng chu kỳ và chạy đua trong mạch không đồng bộ**
- **Một số mạch tuần tự thông dụng**

Khái niệm chung và mô hình toán học

■ Khái niệm chung

- Mạch logic tuần tự hay còn gọi là mạch dãy - Sequential Circuit.
- Hoạt động của hệ này có tính chất kế tiếp nhau, tức là trạng thái hoạt động của mạch điện không những phụ thuộc trực tiếp lối vào mà còn phụ thuộc vào trạng thái bên trong trước đó của chính nó. Nói cách khác các hệ thống này làm việc theo nguyên tắc có nhớ.

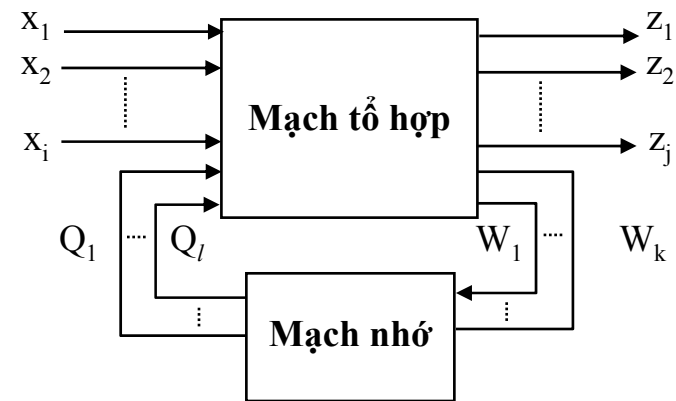
■ Mô hình toán học

- $Z = f(Q, X)$

- ◆ X - tập tín hiệu vào.
- ◆ Q - tập trạng thái trong trước đó của mạch.
- ◆ W - hàm kích.
- ◆ Z - các hàm ra

- Biểu diễn khác: $Z = f(Q(n), X); Q(n+1) = f(Q(n), X)$

- ◆ $Q(n+1)$: là trạng thái tiếp theo của mạch.
- ◆ $Q(n)$: là trạng thái bên trong trước đó.



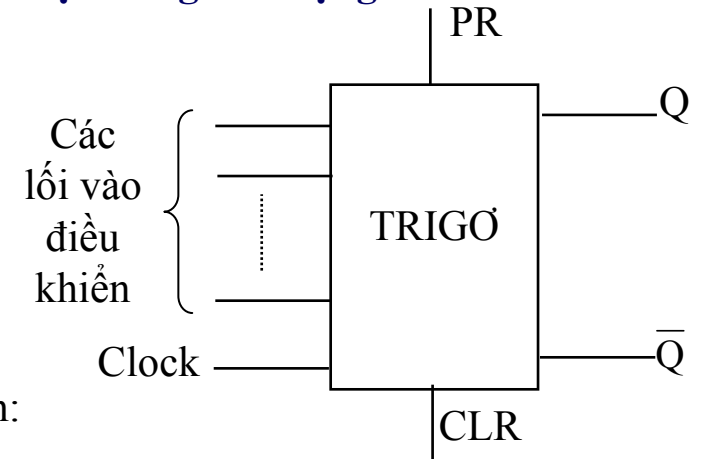
Sơ đồ khối của mạch tuần tự.

Trigơ – Phần tử nhớ của mạch tuần tự

- **Định nghĩa:** Trigơ là phần tử có khả năng lưu trữ (nhớ) một trong hai trạng thái 0 và 1.

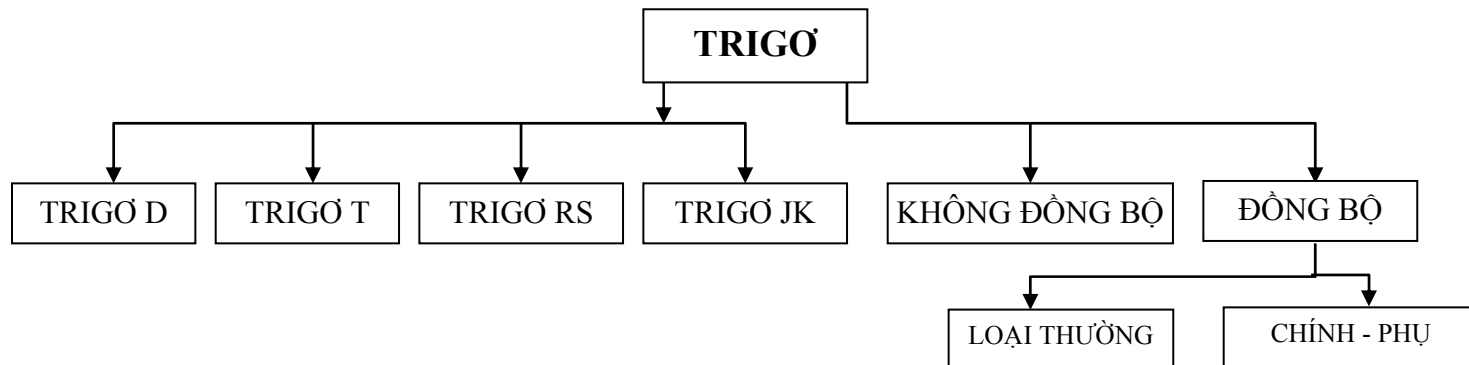
- **Cấu trúc**

- Trigơ có từ 1 đến một vài lối điều khiển, có hai lối ra luôn luôn ngược nhau là Q và \bar{Q} . Tùy từng loại trigơ có thể có thêm các lối vào lập (PRESET) và lối vào xoá (CLEAR). Ngoài ra, trigơ còn có lối vào đồng bộ (CLOCK). Hình bên là sơ đồ khối tổng quát của trigơ.



- **Phân loại:**

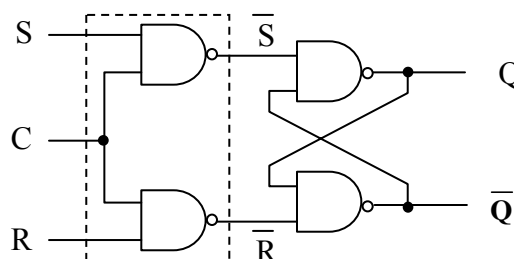
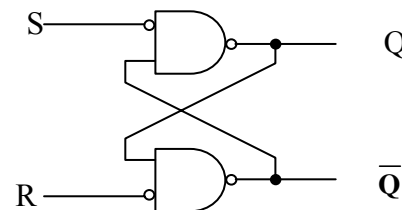
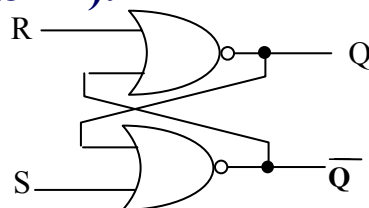
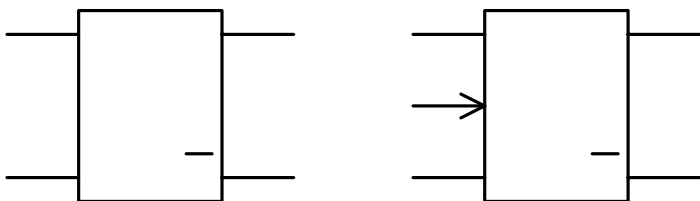
- Theo chức năng làm việc của của các lối vào điều khiển:
 - ◆ Trigơ 1 lối vào như trigơ D, T;
 - ◆ Trigơ 2 lối vào như trigơ RS, trigơ JK.
- Theo phương thức hoạt động thì ta có hai loại:
 - ◆ Trigơ không đồng bộ
 - ◆ Trigơ đồng bộ, có hai loại: trigơ thường và trigơ chính-phụ (Master-Slave).



Trigơ RS (1)

- Trigơ RS là loại có hai lối vào điều khiển S, R. Chân S gọi là lối vào "lập" (SET) và R được gọi là lối vào "xoá" (RESET).

Sơ đồ khối:



Sơ đồ nguyên lý của trigơ RS và RS đồng bộ

Bảng TT của trigơ RS

S	R	Q^k	Mod hoạt động
0	0	Q	Nhớ
0	1	0	Xóa
1	0	1	Lập
1	1	X	Cấm

Bảng TT của trigơ RS đồng bộ cổng NAND

C	S	R	Q^k	Mod hoạt động
0	X	X	Q	Nhớ
1	0	0	Q	Nhớ
1	0	1	0	Xóa
1	1	0	1	Lập
1	1	1	X	Cấm

Trigơ RS (2)

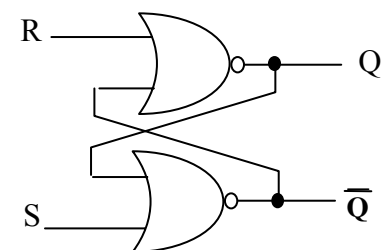
Trigơ RS không đồng bộ

Bảng trạng thái

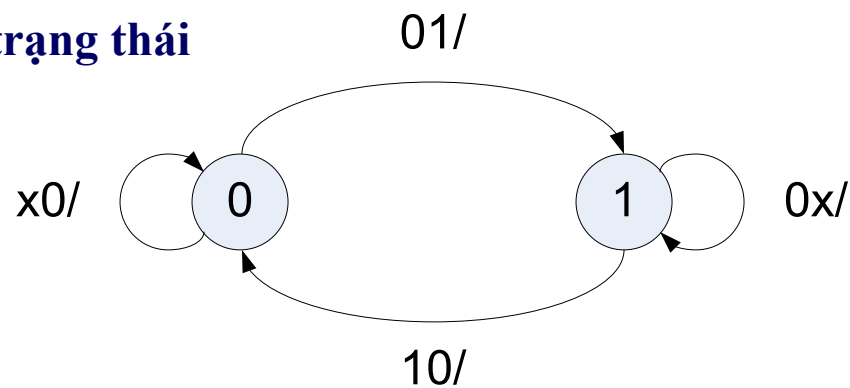
Q	R	S	Q ^k
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	X
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	X

Bảng Các nô

RS	00	01	11	10
Q				
0	0	1	X	0
1	1	1	X	0



Đồ hình trạng thái



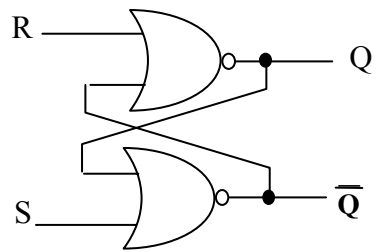
Biểu thức

$$Q^k = S + \bar{R} \cdot Q$$

RS = 0 (điều kiện để tránh tó hợp cam)

Trigơ RS (3)

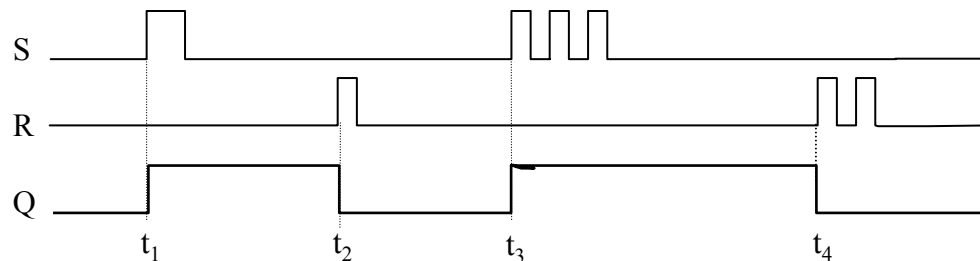
Trigơ RS không đồng bộ



Bảng trạng thái

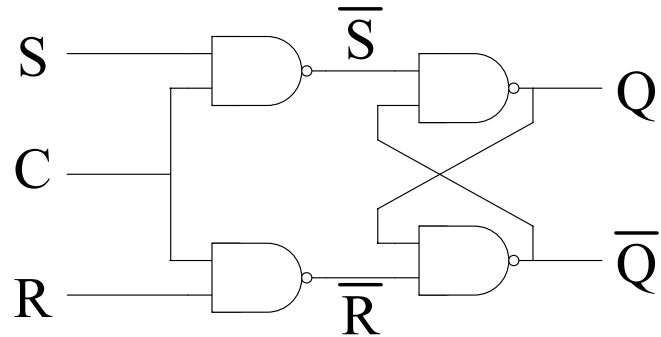
Q	R	S	Q ^k
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	X
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	X

Đồ thị dạng xung

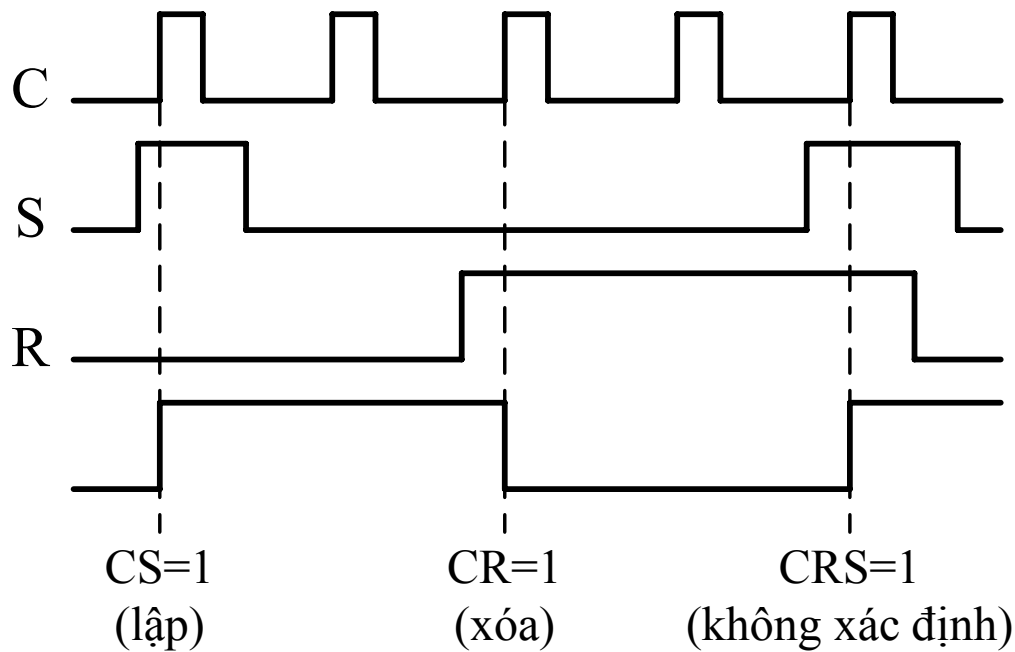


Trigơ RS (4)

Tri gơ RS đồng bộ



Đồ thị dạng xung

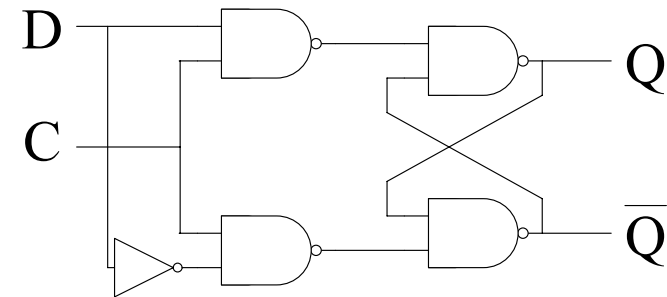
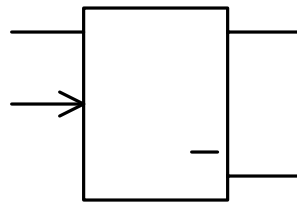
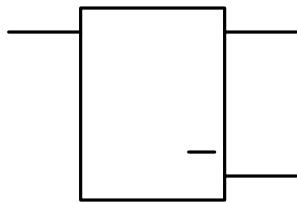


Bảng trạng thái

Bảng TT của trigơ RS đồng bộ cổng NAND				
C	S	R	Q ^k	Mod h.động
0	X	X	Q	Nhớ
1	0	0	Q	Nhớ
1	0	1	0	Xóa
1	1	0	1	Lập
1	1	1	X	Cắm

Trigơ D

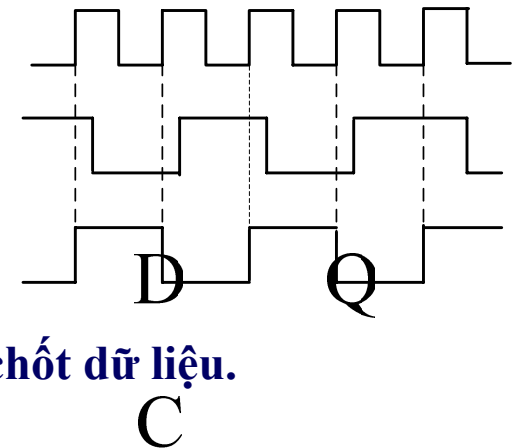
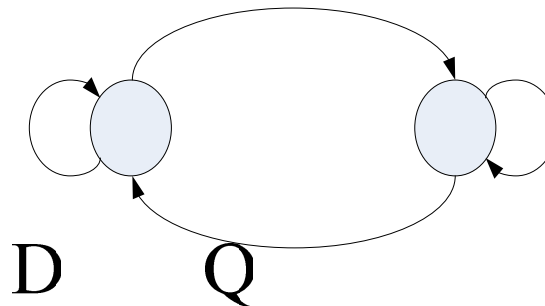
- Trigơ D là loại trigơ có một lối vào điều khiển D.
- Biểu thức: $Q^k = D$, mỗi khi xuất hiện xung nhịp C.
- Sơ đồ khối:



Bảng trạng thái

Q	D	Q^k
0	0	0
0	1	1
1	0	0
1	1	1

Đồ hình trạng thái



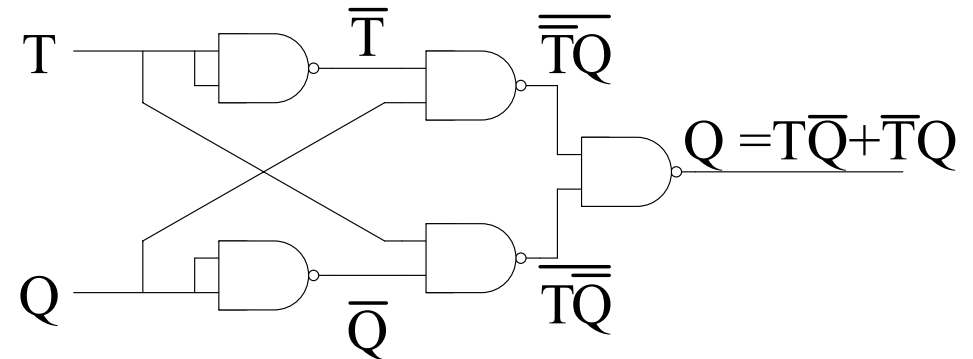
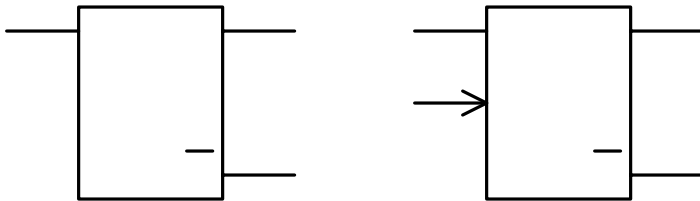
- Ứng dụng: thường dùng làm bộ ghi dịch dữ liệu hay bộ chốt dữ liệu.

Trigơ T

- Trigơ T là loại trigơ có một lối vào điều khiển T. Mỗi khi có xung tới lối vào T thì lối ra Q sẽ thay đổi trạng thái.

- Biểu thức: $Q^k = \bar{T}Q + T\bar{Q} = T \oplus Q$

- Sơ đồ khối:



Bảng trạng thái

T	Q	Q^k
0	0	0
0	1	1
1	0	1
1	1	0

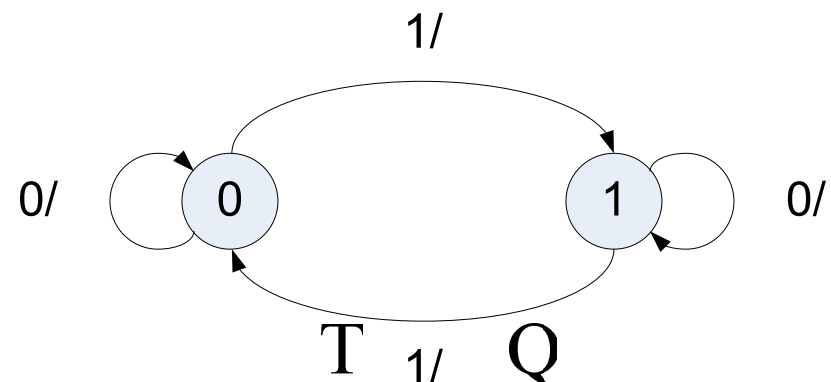
T

Bảng trạng thái
rút gọn

T	Q^k
0	Q
1	\bar{Q}

Q

Đồ hình trạng thái





- Trigrơ JK là loại trigrơ có hai lối vào điều khiển J, K.
- Ưu điểm hơn trigrơ RS là không còn tồn tại tổ hợp cấm bằng các đường hồi tiếp từ Q về chân R và từ \overline{Q} về S.
- Trigrơ JK còn có thêm đầu vào đồng bộ C. Trigrơ có thể lập hay xoá trong khoảng thời gian ứng với sườn âm hoặc sườn dương của xung đồng bộ C. Ta nói, trigrơ JK thuộc loại *đồng bộ*.

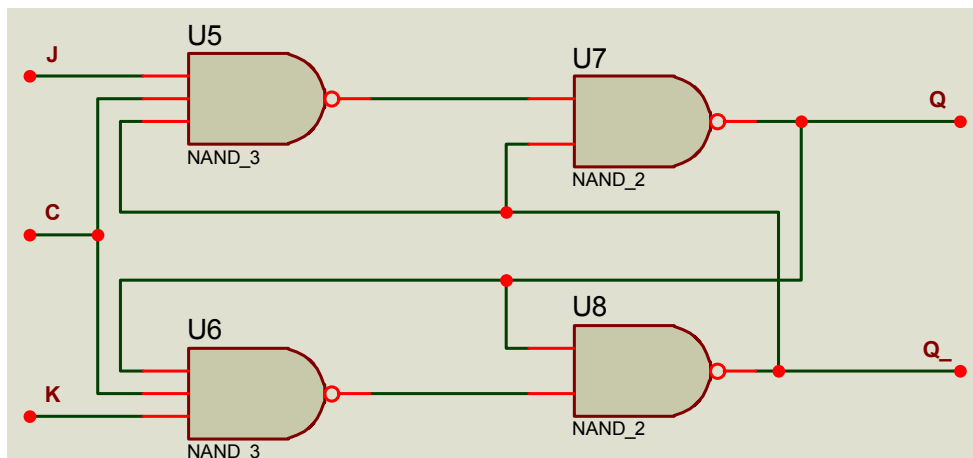
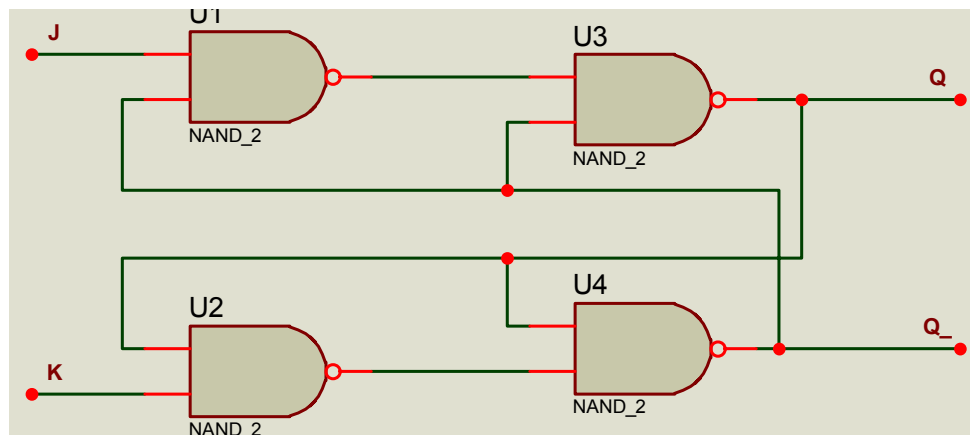
Trigơ JK (2)

Bảng TT đầy đủ

J	K	Q	Q ^k
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Bảng TT rút gọn

J	K	Q ^k
0	0	Q
0	1	0
1	0	1
1	1	Q'



Bảng TT của trigơ JK đồng bộ

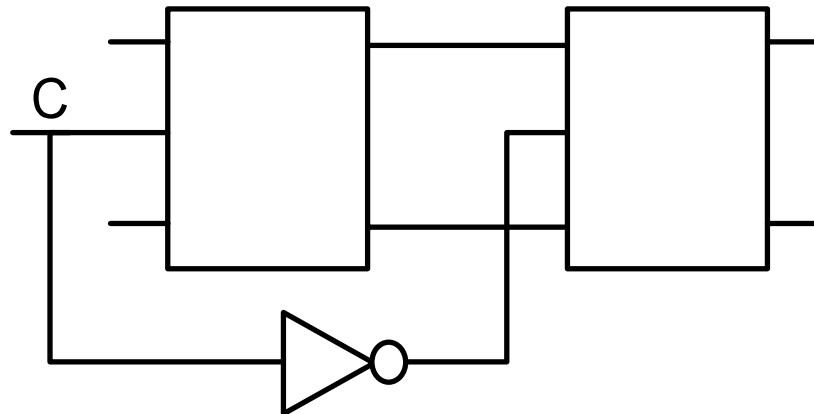
C	J	K	Q ^k
0	X	X	Q
1	0	0	Q
	0	1	0
	1	0	1
	1	1	Q'

Bảng hàm kích thích của các loại Trigo

Q	Q^k	S	R	J	K	T	D
0	0	0	X	0	X	0	0
0	1	1	0	1	X	1	1
1	0	0	1	X	1	1	0
1	1	X	0	X	0	0	1

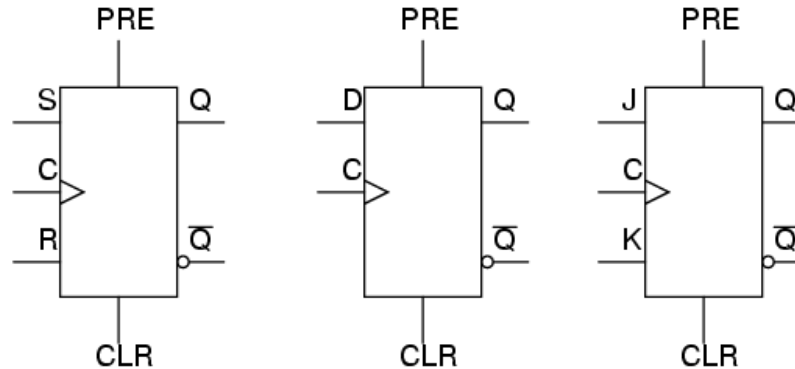
Trigơ Chính-Phụ (Master-Slave)

- Do các loại trigơ đồng bộ trên đều hoạt động tại sườn dương hay sườn âm của xung nhịp nên khi làm việc ở tần số cao thì lỗi ra Q không đáp ứng kịp với sự thay đổi của xung nhịp, dẫn đến mạch hoạt động ở tình trạng không được tin cậy.
- Lỗi ra của trigơ MS thay đổi tại sườn dương và sườn âm của xung nhịp, nên cấu trúc của nó gồm 2 trigơ giống nhau nhưng cực tính điều khiển của xung Clock thì ngược nhau để đảm bảo sao cho tại mỗi sườn của xung sẽ có một trigơ hoạt động.



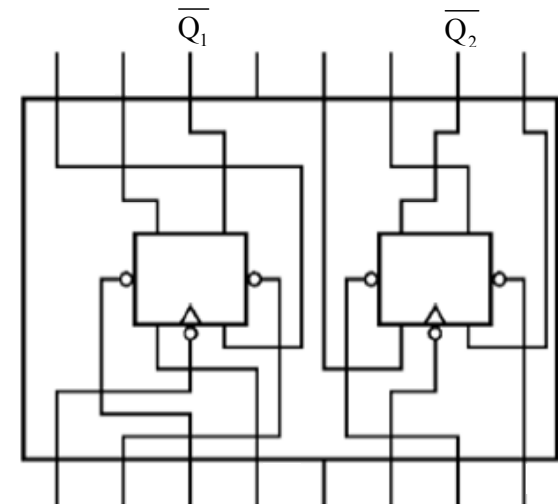
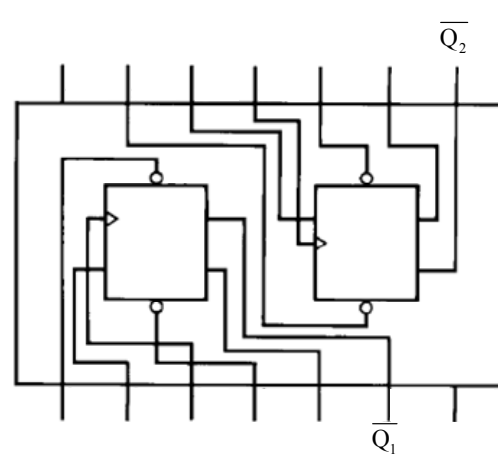
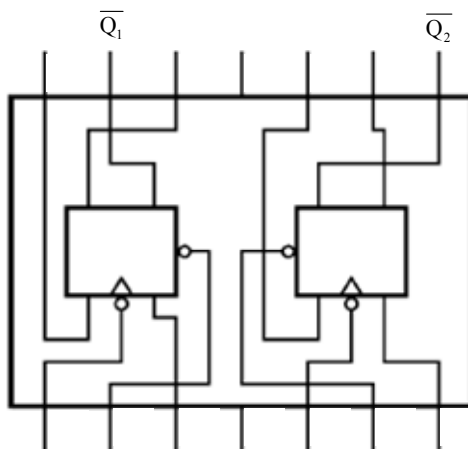
Lỗi vào không đồng bộ của Trigo

- Các lỗi vào dữ liệu thông thường của trigơ như D, S, R, J hoặc K là những lỗi vào đồng bộ
- Các trigơ còn có thêm 2 đầu vào không đồng bộ, các lỗi này tác động trực tiếp lên các lỗi ra mà không phụ thuộc vào xung Clock
- Các lỗi vào này thường được ký hiệu là: PRE (lập) và CLR (R -xóa) hoặc $\overline{\text{PRE}}$ và $\overline{\text{CLR}}$ ($\overline{\text{R}}$)



Một số IC Trơ thông dụng

- **Trơ JK: IC 54/7473-** IC này gồm hai trơ JK có lỗi vào xóa và không có lỗi vào lập hoạt động tại sườn âm của xung Clock
- **Trơ D: IC 54/7474-** IC này gồm hai trơ D có lỗi vào xóa và lỗi vào lập, hoạt động tại sườn dương của xung Clock
- **Trơ JK: IC 54/7476-** IC này gồm hai trơ JK có lỗi vào xóa và lỗi vào lập, hoạt động tại sườn âm của xung Clock.



Phương pháp mô tả mạch tuần tự

- **Phương trình logic (hay phương pháp đại số)**
 - Dùng các phương trình logic để mô tả trạng thái và đầu ra.
- **Bảng trạng thái**
 - Bảng chuyển đổi trạng thái
 - Bảng tín hiệu ra
- **Đồ hình trạng thái**
 - Mô hình Mealy thực hiện ánh xạ
 - Mô hình Moore
- **Đồ thị dạng xung**



Bảng trạng thái (1)

■ Bảng chuyển đổi trạng thái

- Bao gồm các hàng và các cột
- Các hàng ghi các trạng thái trong
- các cột ghi các giá trị của tín hiệu vào.
- Các ô ghi giá trị các trạng thái trong kế tiếp mà mạch sẽ chuyển đến ứng với các giá trị ở hàng và cột

Trạng thái trong	Tín hiệu vào					Trạng thái kế tiếp Qk
	V S	V1	V2	Vn	
	S ₁				→	
	S ₂					
	.					
	S _n					



Bảng trạng thái (2)

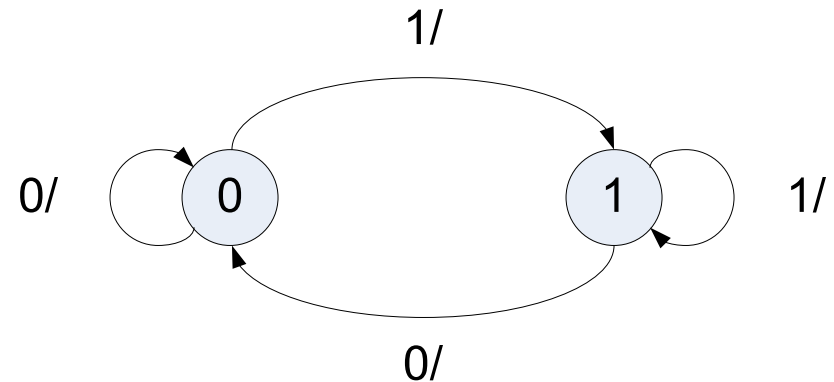
■ Bảng tín hiệu ra

- Các hàng của bảng ghi các trạng thái trong
- Các cột ghi các tín hiệu vào.
- Các ô ghi giá trị của tín hiệu ra tương ứng.

Trạng thái trong	Tín hiệu vào					Tín hiệu ra
	V S	V ₁	V ₂	V _n	
	S ₁				→	
	S ₂					
	⋮					
	S _n					

Đồ hình trạng thái

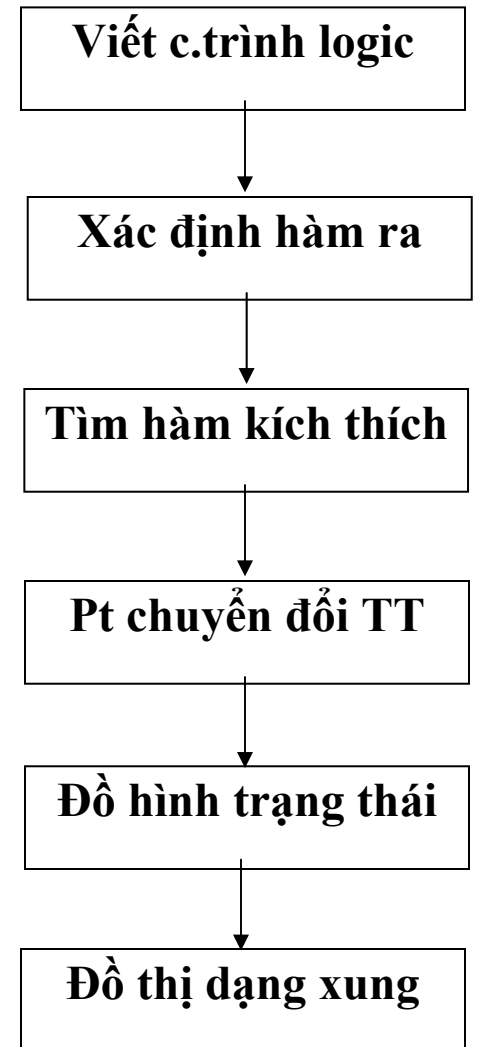
- Đồ hình trạng thái là hình vẽ phản ánh quy luật chuyển đổi trạng thái và tình trạng các giá trị ở lối vào và lối ra tương ứng của mạch tuần tự.
- Đồ hình trạng thái là một đồ hình có hướng gồm hai tập:
- M - Tập các đỉnh và K - Tập các cung có hướng.
- Mô hình Mealy
- Mô hình Moore



Q	D	Q ^k
0	0	0
0	1	1
1	0	0
1	1	1

Phân tích mạch tuần tự - Lý thuyết

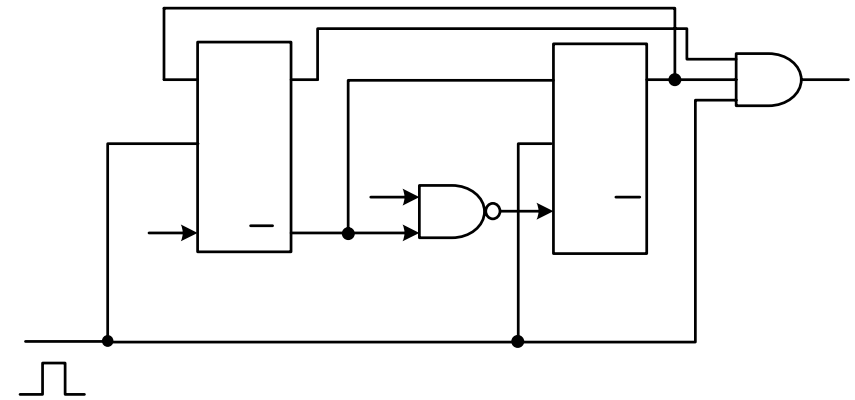
- **Viết chương trình logic:**
 - Viết chương trình logic cho lối vào đồng bộ, chỉ ra điều kiện chuyển trạng thái của các phần tử nhớ.
- **Xác định hàm ra:**
- **Tìm hàm kích thích:**
 - Căn cứ loại TG để tìm kích thích, phương trình chuyển đổi trạng thái (chính là phương trình đặc trưng của TG đã cho).
- **Phương trình chuyển đổi trạng thái:**
 - Xác định số tổ hợp trạng thái và thay các tổ hợp này vào các phương trình kích thích, phương trình chuyển đổi trạng thái để tính bảng chuyển đổi trạng thái.
- **Vẽ đồ hình trạng thái dưới dạng nhị phân hoặc dạng rút gọn**
- **Vẽ đồ thị dạng xung gồm:**
 - Xung đồng hồ,
 - Xung của mỗi biến trạng thái,
 - Xung ra.



Các bước phân tích mạch tuần tự

Phân tích mạch tuần tự - Ví dụ

- **Bước 1:** Sơ đồ có hai đầu vào là tín hiệu X và xung nhịp Clock. Có một tín hiệu Z ra, mạch sử dụng hai phần tử nhớ là hai trigơ JK (Q_0 và Q_1).
- **Bước 2:** Xác định đầu vào, đầu ra và số trạng thái trong của mạch.
- Mạch này có thể được biểu diễn bằng một “hộp đen” có hai đầu vào và một đầu ra. Do mạch được cấu tạo bằng hai trigơ nên số trạng thái có thể có của mạch là 4. Cụ thể là: $Q_1Q_0 = 00, 01, 10$ và 11.



Phân tích mạch tuần tự - Ví dụ

■ **Bước 3: Xác định phương trình hàm ra và hàm kích cho trigơ.**

- Từ sơ đồ trên ta tìm được:
- Phương trình hàm ra: $Z = C Q_1 Q_0$
- Phương trình hàm kích
 - ◆ $J_0 = Q_1; K_0 = 1$
 - ◆ $J_1 = \overline{Q_0}; K_1 = \overline{X \overline{Q_0}} = \overline{X} + Q_0$

■ **Bước 4. Bảng chuyển đổi trạng thái**

- Phương trình đặc trưng của trigơ JK là $Q^k = J\overline{Q} + \overline{K}Q$
- Phương trình chuyển đổi trạng thái:

$$Q_0^k = J_0 \overline{Q_0} + \overline{K_0} Q_0 = Q_1 \overline{Q_0}$$

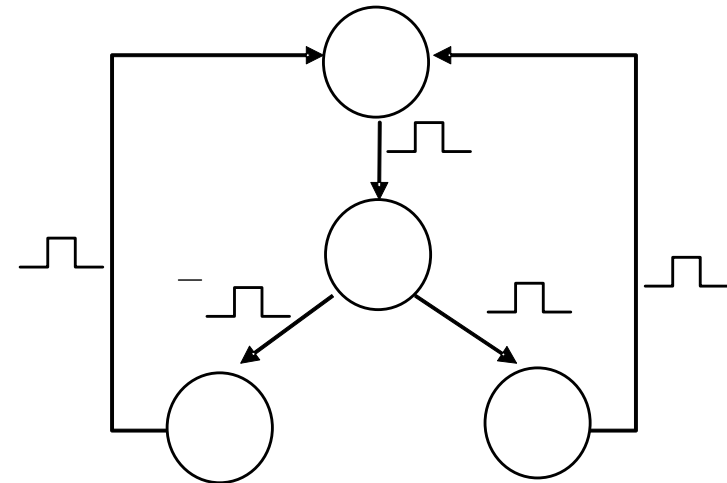
$$Q_1^k = J_1 \overline{Q_1} + \overline{K_1} Q_1 = \overline{Q_0} \overline{Q_1} + \overline{\overline{X} + Q_0} Q_1 = \overline{Q_0} \overline{Q_1} + X \overline{Q_0} Q_1$$

Phân tích mạch tuần tự - Ví dụ

	Trạng thái hiện tại	Trạng thái kế tiếp		Tín hiệu ra	
		X = 0 Q_0Q_1	X = 1 Q_0Q_1	X = 0 Z	X = 1 Z
S ₀	00	01	01	0	0
S ₁	01	10	11	0	0
S ₂	11	00	00	1	1
S ₃	10	00	00	0	0

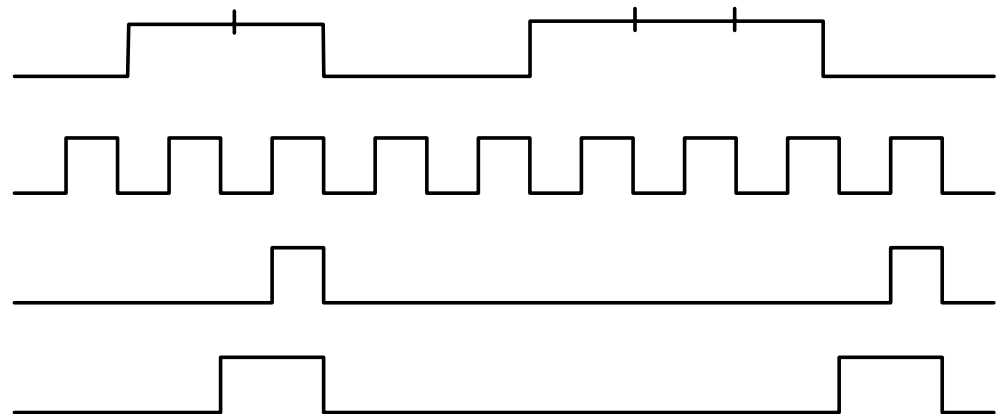
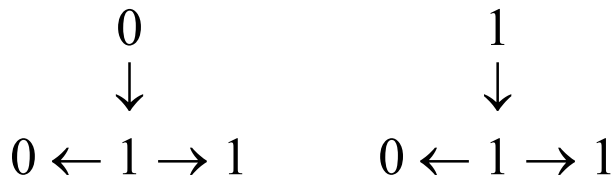
. Bảng chuyển đổi trạng thái

■ Bước 5: Đồ hình trạng thái.



Phân tích mạch tuần tự - Ví dụ

- **Bước 6: Chức năng của mạch:**
- Trên đồ hình trạng thái ta thấy có hai đường chuyển đổi trạng thái là $S_0 \rightarrow S1 \rightarrow S2 \rightarrow S--0$ và $S0 \rightarrow S1 \rightarrow S3 \rightarrow S--0$.
- Theo đường $S0 \rightarrow S1 \rightarrow S2 \rightarrow S--0$ thì tín hiệu ra $Z = 1$ sẽ được đưa ra cùng thời điểm có xung nhịp thứ 3.
- Theo đường $S0 \rightarrow S1 \rightarrow S3 \rightarrow S--0$ thì không có tín hiệu ra ($Z = 0$). Do vậy ta sẽ phân tích theo con đường thứ nhất $S0 \rightarrow S1 \rightarrow S2 \rightarrow S--0$: Sự chuyển đổi trạng thái đầu tiên từ $S0 \rightarrow S1$ chỉ nhờ tác động của xung nhịp mà không phụ thuộc vào trạng thái của X .
- Chuyển đổi trạng thái thứ hai từ $S1 \rightarrow S2$ nhờ tác động của xung nhịp và sự tác động của tín hiệu vào $X = 1$.
- Còn sự chuyển đổi trạng thái thứ ba từ $S2 \rightarrow S0$ chỉ nhờ tác động của xung nhịp mà không phụ thuộc vào tín hiệu vào.



Thiết kế mạch tuần tự - Lý thuyết

■ Bài toán ban đầu:

- Nhiệm vụ thiết kế được mô tả bằng ngôn ngữ hoặc bằng lưu đồ thuật toán.

■ Hình thức hoá:

- Từ các dữ kiện đề bài cho mà ta mô tả hoạt động của mạch bằng cách hình thức hoá dữ kiện ban đầu ở dạng bảng trạng thái, bảng ra hay đồ hình trạng thái. Sau đó rút gọn các trạng thái của mạch để có được số trạng thái trong ít nhất.

■ Mã hoá trạng thái:

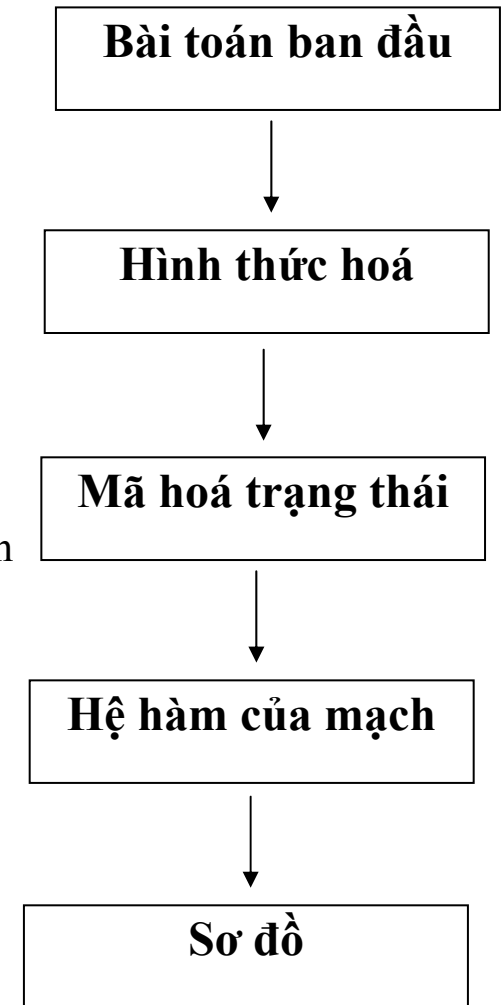
- Mã hoá tín hiệu vào ra, trạng thái trong để nhận được mã nhị phân (hoặc có thể là các loại mã khác) có tập tín hiệu vào là X, tập tín hiệu ra là Y, tập các trạng thái trong là Q.

■ Hệ hàm của mạch:

- Xác định hệ phương trình logic của mạch và tối thiểu hoá các phương trình này. Nếu mạch tuần tự khi thiết kế cần dùng các trigger và mạch tổ hợp thì tùy theo yêu cầu mà ta viết hệ phương trình cho các lối vào kích cho từng loại trigger đó.

■ Xây dựng sơ đồ:

- Từ hệ phương trình của mạch đã viết được ta xây dựng mạch điện thực hiện.



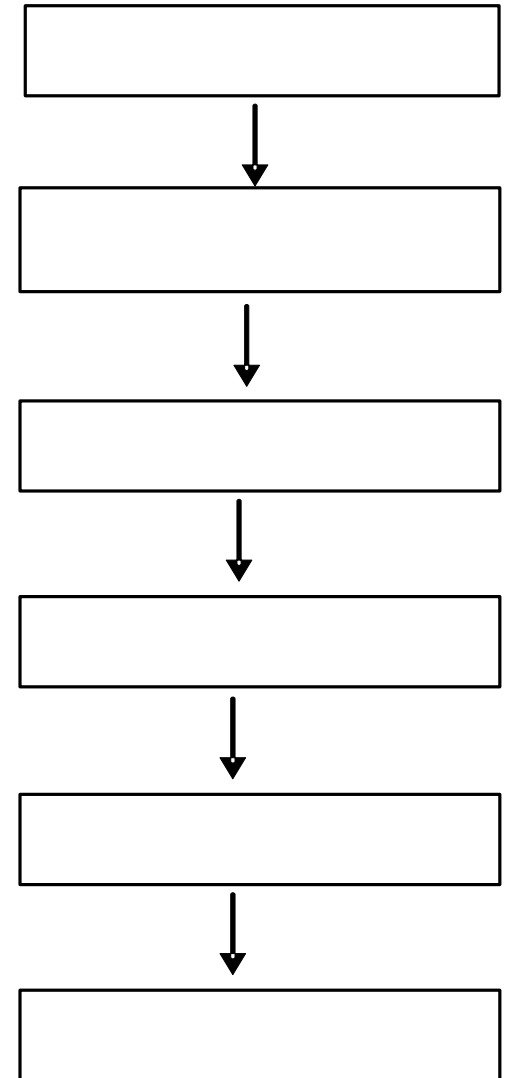
Các bước thiết kế mạch tuần tự

Thiết kế mạch tuần tự - Ví dụ

- **Bài toán :Thiết kế mạch điều khiển đèn đường**
- **Hình thức hóa và mã hóa**
 - Ký hiệu trạng thái các đèn (sáng: 1, tắt 0)
 - Tính toán số trạng thái
 - Vẽ sơ đồ trạng thái
 - Mã hóa trạng thái
 - Xây dựng bảng sự thật
- **Xây dựng hàm**
 - Từ bảng sự thật, rút gọn và xây dựng hàm
- **Xây dựng sơ đồ mạch**
 - Xây dựng sơ đồ mạch từ các phương trình đại số logic.

Mạch tuần tự đồng bộ

- **Bước 1:** Xác định bài toán, gán hàm và biến, tìm hiểu mối quan hệ giữa chúng.
- **Bước 2:** Xây dựng đồ hình trạng thái, bảng chuyển đổi trạng thái và hàm ra.
- **Bước 3:** Rút gọn trạng thái (tối thiểu hoá trạng thái).
- **Bước 4:** Mã hoá trạng thái.
 - Nếu số lượng trạng thái trong là N , số biến nhị phân cần dùng là n thì n phải thoả mãn điều kiện: $n \geq \log_2 N$.
- **Bước 5:** Xác định hệ phương trình của mạch. Có hai cách xác định:
 - + Lập bảng chuyển đổi trạng thái và tín hiệu ra, từ đó xác định các phương trình kích cho các trigơ.
 - + Dựa trực tiếp vào đồ hình trạng thái, viết hệ phương trình Ton, Toff của các trigơ và phương trình hàm ra.
- **Bước 6:** Vẽ sơ đồ thực hiện.



Mạch tuần tự không đồng bộ

- **Bước 1: Xác định bài toán, gán hàm và biến, tìm hiểu mối quan hệ giữa chúng.**
- **Bước 2: Xây dựng đồ hình trạng thái, bảng chuyển đổi trạng thái và hàm ra.**
- **Bước 3: Rút gọn trạng thái (tối thiểu hoá trạng thái).**
- **Bước 4: Mã hoá trạng thái.**
 - Nếu số lượng trạng thái trong là N, số biến nhị phân cần dùng là n thì n phải thoả mãn điều kiện: $n \geq \log_2 N$.
 - Do mạch không đồng bộ hoạt động không có sự tác động của xung nhịp cho nên trong mạch thường có các hiện tượng chạy đua làm cho hoạt động của mạch bị sai, vì vậy khi mã hoá trạng thái phải tránh hiện tượng này.
- **Bước 5: Xác định hệ phương trình của mạch. Có hai cách xác định:**
 - + Lập bảng chuyển đổi trạng thái và tín hiệu ra, từ đó xác định các phương trình kích cho các trigơ.
 - + Dựa trực tiếp vào đồ hình trạng thái, viết hệ phương trình Ton, Toff của các trigơ và phương trình hàm ra.
 - Cả hai cách này đều có dạng phương trình:
 - Phương trình của mạch chỉ dùng NAND.
 - Phương trình của mạch dùng trigơ RS không đồng bộ và các mạch NAND.
 - Phương trình của mạch dùng các loại trigơ khác.
- **Bước 6: Vẽ sơ đồ thực hiện.**



Các cách thiết kế mạch tuần tự

- *Cách 1: Dựa vào bảng chuyển đổi trạng thái.*
- *Ký hiệu : A, B, \dots, N là các biến nhị phân dùng để mã hoá các trạng thái trong của mạch.*

$X1, X2 \dots X_m$ là các tín hiệu vào đã được mã hoá nhị phân.

$Z1, Z2 \dots Z_m$ là các tín hiệu ra đã được mã hoá nhị phân.

Dựa vào bảng chuyển đổi trạng thái xác định hệ phương trình:

$$A_k = f_A(A, B, \dots, N, X1, X2 \dots X_m)$$

$$B_k = f_B(A, B, \dots, N, X1, X2 \dots X_m)$$

.....

$$N_k = f_N(A, B, \dots, N, X1, X2 \dots X_m)$$

$$Z1 = g1(A, B, \dots, N, X1, X2 \dots X_m)$$

$$Z2 = g2(A, B, \dots, N, X1, X2 \dots X_m)$$

.....

$$Zn = g_n(A, B, \dots, N, X1, X2 \dots X_m)$$

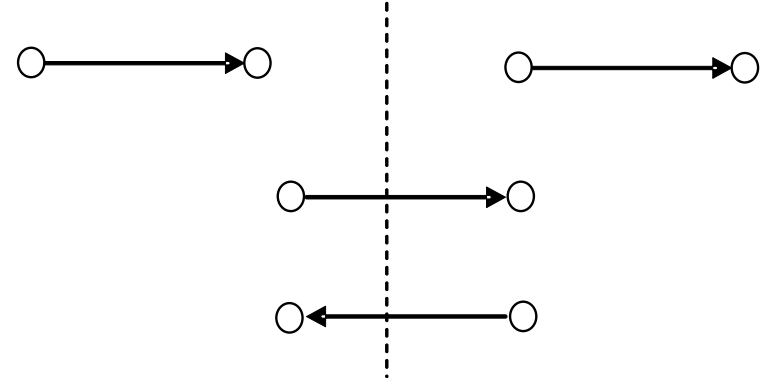
Tối thiểu hoá hệ hàm và viết phương trình ở dạng chỉ dùng NAND.

Các cách thiết kế mạch tuần tự

- ***Cách 2: Dựa trực tiếp vào đồ hình trạng thái***
 - Cho đồ hình trạng thái của mạch có tập tín hiệu vào V, tập tín hiệu ra R, tập trạng thái trong S (chưa mã hoá nhị phân).
- **Các bước thiết kế**
 - Mã hoá tín hiệu vào V, tín hiệu ra R, trạng thái trong S để chuyển thành mạch dạng nhị phân có các tập tín hiệu vào X, tín hiệu ra Y, trạng thái trong Q.
 - Xác định hệ phương trình tín hiệu ra: $Y_i = f_i(X, Q)$. Phương trình này được xác định trên các cung với mô hình kiểu Mealy, trên các đỉnh với mô hình kiểu Moore. Tối thiểu các hàm này.
 - Xác định hệ phương trình hàm kích cho các trigơ và tối thiểu hoá nó.
 - Sau đây giới thiệu thuật toán xác định phương trình lỗi vào kích cho các trigơ từ đồ hình trạng thái.
 - Đối với trigơ Q_i bất kỳ sự thay đổi trạng thái từ Q_i đến Q_{ki} chỉ có thể có 4 khả năng.

- các cung biểu diễn sự thay đổi từ đến được ký hiệu như sau:

- $0 \rightarrow 0$ là (0)
- $1 \rightarrow 1$ (là 1)
- $0 \rightarrow 1$ là (2)
- $1 \rightarrow 0$ là (3).

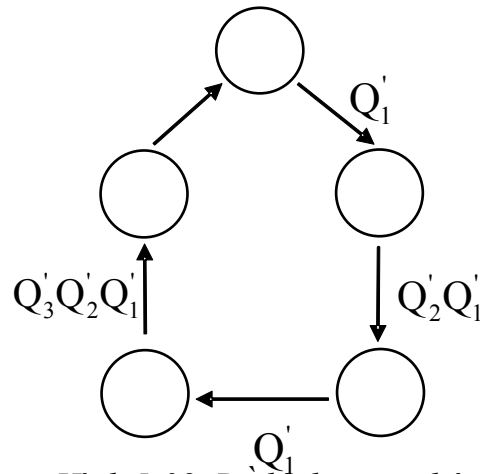


- Thuật toán xác định phương trình lỗi vào kích cho trigơ Q_i loại D. $Q_i^{n+1} = D_i$

$$D_i = Q_i^{n+1}$$

- = tuyển tất cả các cung đi tới đỉnh có $Q_i = 1$.
- = \sum các cung loại (2), kể cả khuyên tại đỉnh đó tức là cung loại 1
- = \sum (1) và (2)

- Thiết kế bộ đếm đồng bộ có $M_d = 5$



Hình 5-38. Đồ hình trạng thái

				100

Bảng 5-17. Bảng chuyển đổi trạng thái

Bảng 5-16. Bảng mã hóa trạng thái

0 000

			001

Bảng 5-18. Bảng Các nô tìm hàm ra



Ví dụ dùng trigơ D

- Nhìn vào đồ hình trạng thái ta thấy: $Q_3 = 1$ tại đỉnh (4), $Q_2 = 1$ tại đỉnh (2), (3), $Q_1 = 1$ tại đỉnh (1), (3).
- $D_3 = \sum \text{Các cung đi đến đỉnh (4)} = (3) = \overline{Q_3} Q_2 Q_1$
- $D_2 = \sum \text{Các cung đi đến đỉnh (2), (3)} = (1) + (2) = \overline{Q_3} \overline{Q_2} Q_1 + \overline{Q_3} Q_2 \overline{Q_1}$
- $D_1 = \sum \text{Các cung đi đến đỉnh (1), (3)} = (0) + (2) = \overline{Q_3} \overline{Q_2} \overline{Q_1} + \overline{Q_3} Q_2 \overline{Q_1}$
- Từ đó ta lập bảng Các nô để tối thiểu hóa hàm Di

Ví dụ trigơ D

Q_2Q_1		Q_3			
		00	01	11	10
Q_3	0	0	0	1	0
	1	0	x	x	x

$$D_3 = Q_2Q_1$$

Q_2Q_1		00	01	11	10
Q_3					
0		0	1	0	1
1		0	x	x	x

$$D_2 = \overline{Q_2} \cdot Q_1 + Q_2 \cdot \overline{Q_1} = Q_1 \oplus Q_2$$

Q_2Q_1		00	01	11	10
Q_3					
0		1	0	0	1
1		0	x	x	x

$$D_1 = \overline{Q_1} \cdot \overline{Q_3}$$

Bảng 5-20. Bảng Các nô tìm hàm kích

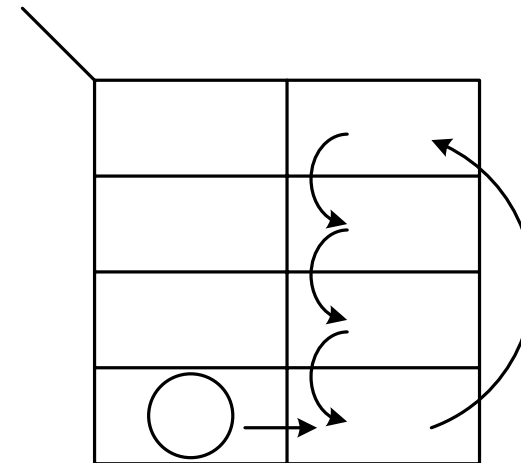
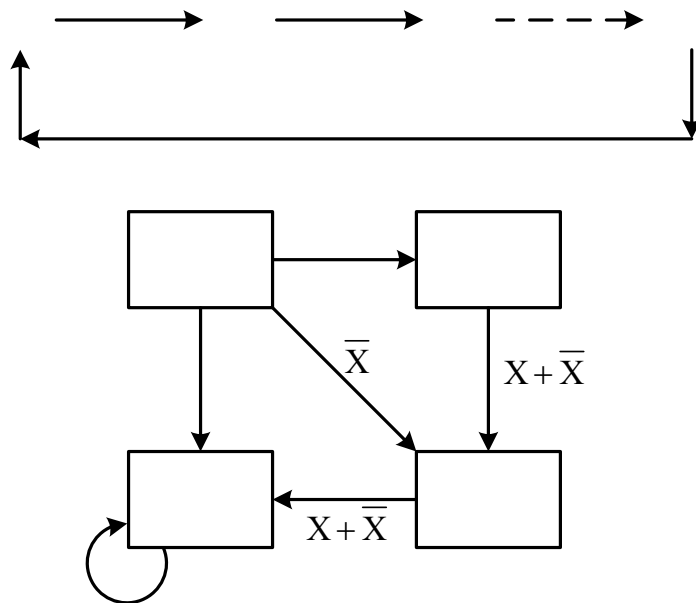
$$D_3 = Q_2Q_3$$

$$D_2 = \overline{Q_2} Q_3 + Q_2 \overline{Q_3} = Q_2 \oplus Q_3$$

$$D_1 = \overline{Q_1} \overline{Q_3}$$

■ Định nghĩa hiện tượng chu kỳ:

- Hiện tượng chu kỳ là hiện tượng tại một tổ hợp tín hiệu vào nào đó, mạch liên tục chuyển từ trạng thái này sang trạng thái khác theo một chu kỳ kín.
- Nghĩa là trong quá trình đó không có trạng thái nào ổn định.
- Do vậy, khi thay đổi tín hiệu vào không xác định được mạch đang ở trạng thái nào trong dãy trạng thái nói trên.



Hiện tượng chạy đua trong mạch không DB

■ Định nghĩa:

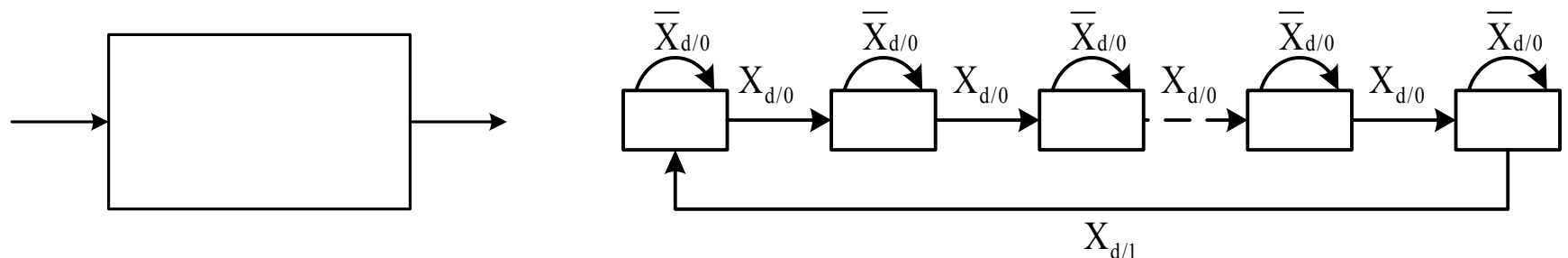
- Hiện tượng chạy đua trong mạch không đồng bộ là hiện tượng: do tính không đồng nhất của các phần tử nhị phân dùng để mã hoá trạng thái, vì mạch hoạt động không đồng bộ, khi mạch chuyển trạng thái từ $S_i \rightarrow S_j$ mạch có thể chuyển biến trạng thái theo những con đường khác nhau.
- Nếu trạng thái cuối cùng của những con đường đó là ổn định và duy nhất thì chạy đua không nguy hiểm.
- Ngược lại, chạy đua nguy hiểm là những cách chuyển biến trạng thái khác nhau đó cuối cùng dẫn đến các trạng thái ổn định khác nhau, có thể tới trạng thái khoá và không thoát ra được.



Một số mạch tuần tự thông dụng

- Bộ đếm
- Bộ ghi dịch
- ...

- **Định nghĩa :** Bộ đếm là một mạch tuần tự tuần hoàn có một lối vào đếm và một lối ra, mạch có số trạng thái trong bằng chính hệ số đếm (ký hiệu là M_d).
- Dưới tác dụng của tín hiệu vào đếm, mạch sẽ chuyển từ trạng thái trong này đến một trạng thái trong khác theo một thứ tự nhất định.
- Cứ sau M_d tín hiệu vào đếm mạch lại trở về trạng thái xuất phát ban đầu.
- Bộ đếm được dùng rất nhiều trong các dụng cụ đo lường chỉ thị số, các máy tính điện tử.
- Bất kỳ hệ thống số hiện đại nào đều sử dụng các bộ đếm.



Phân loại bộ đếm

■ Theo sự chuyển đổi trạng thái:

- Bộ đếm đồng bộ (Synchronous): Các trigơ đều chịu tác dụng điều khiển của một xung đồng hồ duy nhất
- Bộ đếm không đồng bộ (Asynchronous): có trigơ chịu tác dụng điều khiển trực tiếp của xung đếm đầu vào, nhưng cũng có trigơ chịu tác dụng điều khiển của xung ở đầu ra của trigơ khác .

■ Theo hệ số đếm

- Bộ đếm nhị phân
- Bộ đếm thập phân
- Bộ đếm N phân

■ Theo xung đếm

- Bộ đếm thuận (Up counter) hay còn gọi là bộ đếm tiến
- Bộ đếm nghịch (Down counter) hay còn gọi là bộ đếm lùi
- Bộ đếm thuận nghịch

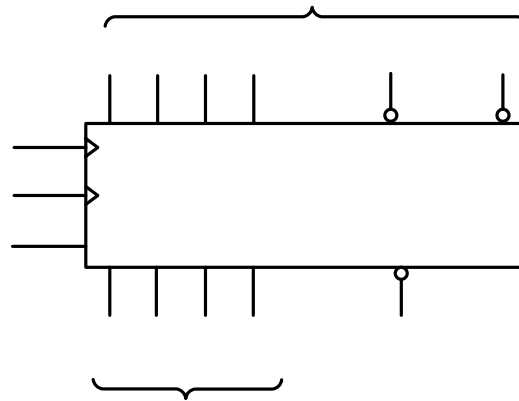


Một số IC đếm

Tên IC	Mô tả	Đặc tính
7492	Gồm 4 trigơ JK mắc thành hai bộ đếm không đồng bộ mod 2 và mod 6 độc lập.	
7493	Gồm 4 trigơ JK mắc thành hai bộ đếm không đồng bộ mod 2 và mod 8 độc lập.	
74190	Bộ đếm thuận nghịch (UP/DOWN) thập phân	Preset đồng bộ và không Clear
74191	Bộ đếm thuận nghịch (UP/DOWN) nhị phân 4 bit	Preset đồng bộ và không Clear
74192	Bộ đếm thuận nghịch (UP/DOWN) thập phân	Preset đồng bộ và Clear
74193	Bộ đếm thuận nghịch (UP/DOWN) nhị phân 4 bit	Preset đồng bộ và Clear
74390	Gồm hai khối giống hệt nhau, mỗi khối gồm 4 trigơ JK mắc thành hai bộ đếm không đồng bộ mod 2 và mod 5 độc lập	

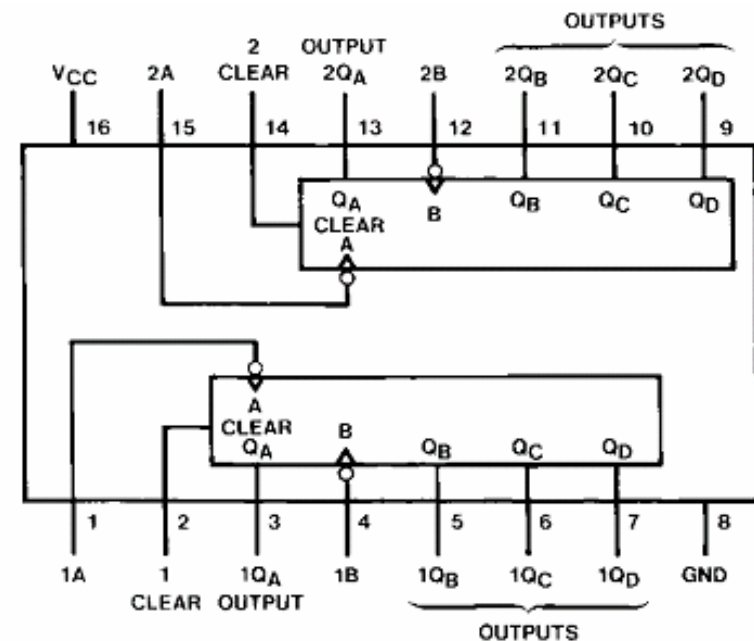
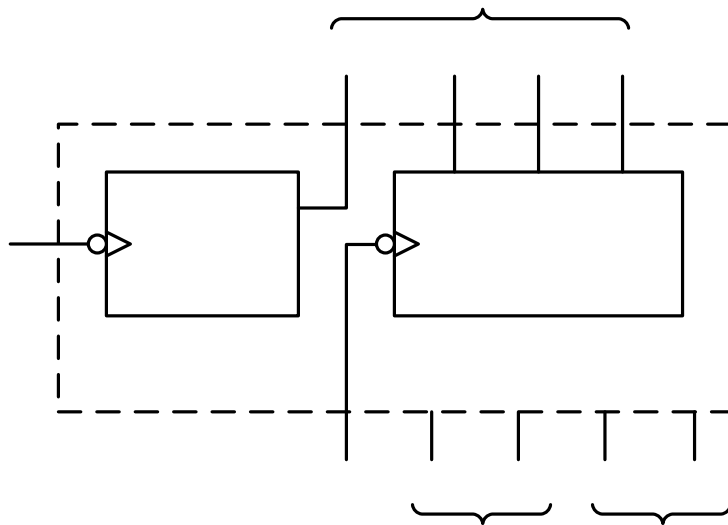
IC 74192, 74193

- Trong các bộ đếm này, khi thức hiện đếm thuận thì xung Clock được nối với CLK-UP, còn chân CLK-DOWN được nối với logic 1; khi đếm nghịch thì ngược lại.
- Các chân CARRY (nhớ) và BORROW (mượn) có logic 1 và nó sẽ chuyển mức thấp khi tràn mức hoặc dưới mức.
- Chân $LOAD = 0$ có thể nạp dữ liệu vào bộ đếm.



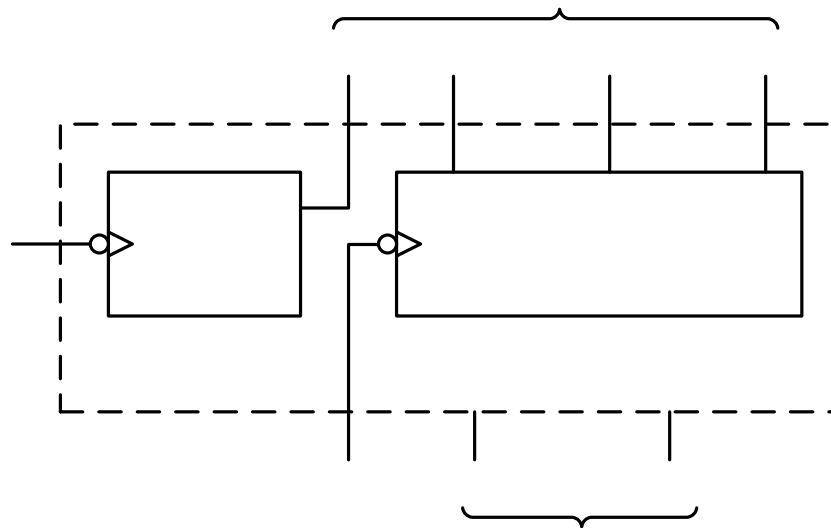
IC 7490, 74390

- Nó bao gồm 4 trigger cung cấp bộ đếm gồm hai Mod đếm: Mod 2 và Mod 5.
- Các bộ đếm Mod 2 và Mod 5 có thể được sử dụng một cách độc lập.
- Trigger A thực hiện đếm Mod 2, Trigger B, C, D thực hiện đếm Mod 5.
- IC 74390 là bản kép (dual) của 7490



IC 7492, 7493, IC 74293, 74393

- Nó bao gồm 4 trigger cung cấp bộ đếm gồm hai Mod đếm: Mod 2 và Mod 6 hoặc mod 8.
- Trigger A thực hiện đếm Mod 2, Trigger B, C, D thực hiện đếm Mod 5.
- Hoạt động của những bộ đếm này giống như IC 7490, chỉ khác là không có các lối vào lập và Mod 6 không đếm theo trình tự nhị phân.
- Các IC này thường không dùng làm các bộ đếm mà dùng làm bộ chia tần





Thiết kế bộ đếm bất kỳ dùng bộ đếm vạn năng

- Một số bộ đếm có các chân xóa (CLR), lối nạp dữ liệu, chân RC (ripple carry) ra có thể lập trình được
- VD IC 74192, 74193
- Để tìm một bộ đếm chia hết cho m thì đầu vào nạp P được cho bởi công thức: $P=(16-m)$ (nếu dùng bộ đếm hex) hoặc $=10-m$ nếu dùng bộ đếm thập phân
- Khi bộ đếm đếm tới giá trị m thì dùng giá trị này để nối vào chân CLR. Nhiệm vụ của chân Clear là gập bit 1 thì xóa về 0. Nếu số bit 1 nhiều hơn số chân Clear thì ta phải dùng thêm cổng NAND (hoặc cổng AND) tùy mức tích cực của chân Clear
- Nếu bộ đếm không bắt đầu từ 0 (VD đếm từ n đến m) thì phải nạp giá trị n khi bắt đầu đếm lại)

- Có khả năng ghi (nhớ) số liệu và dịch thông tin (sang phải hoặc sang trái).
- Được cấu tạo từ một dãy phần tử nhớ được mắc liên tiếp với nhau và một số các cổng logic cơ bản hỗ trợ.
- Muốn ghi và truyền một từ nhị phân n bit cần n phần tử nhớ (n trigger)

Phân loại:

- **Phân theo cách đưa tín hiệu vào và lấy tín hiệu ra:**
 - Vào nối tiếp, ra song song – SIPO (Serial Input, Parallel Output)
 - Vào song song, ra song song – PIPO (Parallel Input, Parallel Output)
 - Vào nối tiếp, ra nối tiếp – SISO (Serial Input, Serial Output)
 - Vào song song, ra nối tiếp – PISO (Parallel Input, Serial Output):
- **Phân theo hướng dịch:**
 - Dịch phải, dịch trái, dịch hai hướng, dịch vòng
- **Phân theo đầu vào:**
 - Đầu vào đơn:
 - Đầu vào đôi:
- **Phân theo đầu ra:**
 - Đầu ra đơn:
 - Đầu ra đôi:

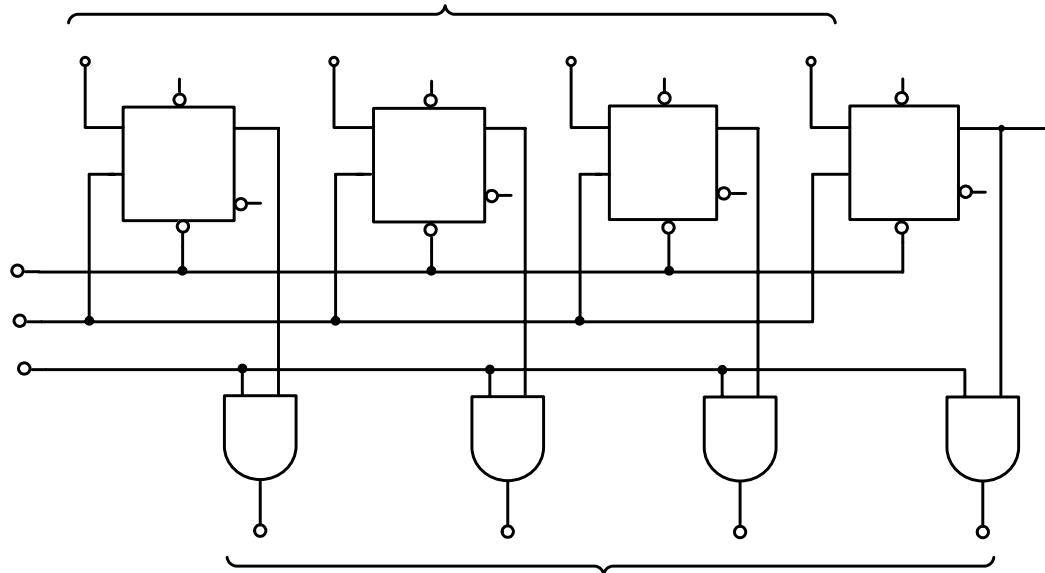


Ứng dụng của bộ ghi dịch

- nhớ dữ liệu
- chuyển dữ liệu từ song song thành nối tiếp và ngược lại.
- để thiết kế bộ đếm
- tạo dãy tín hiệu nhị phân tuần hoàn
- Một số IC ghi dịch (giáo trình DTS mục 5.9.4)

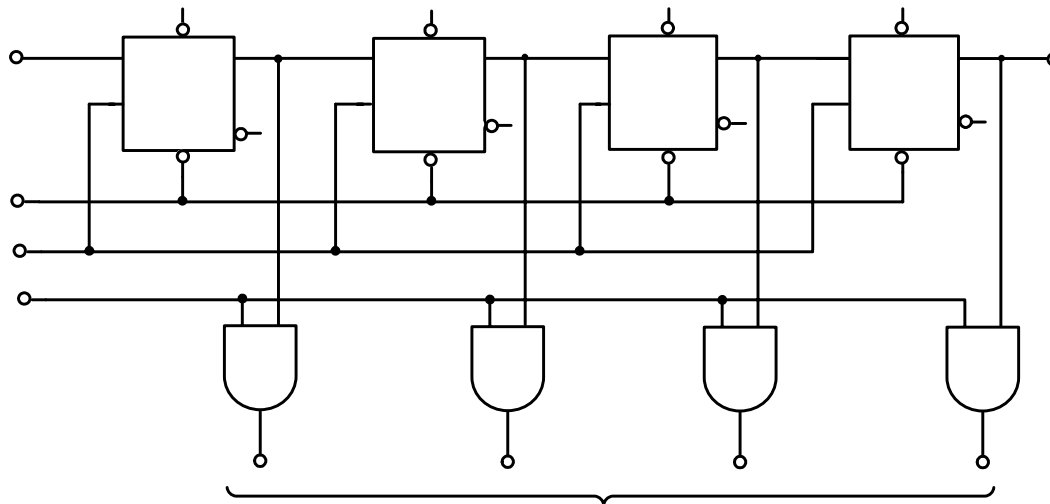
Bộ ghi dịch song song

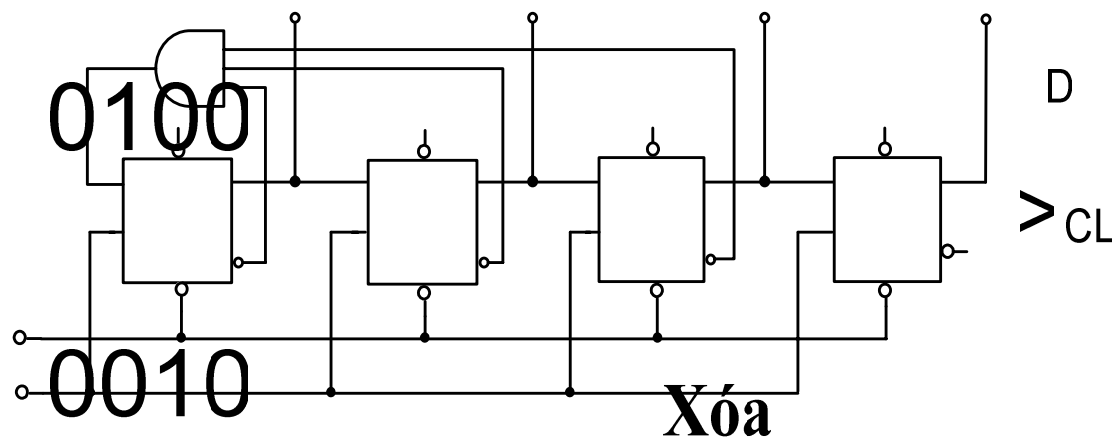
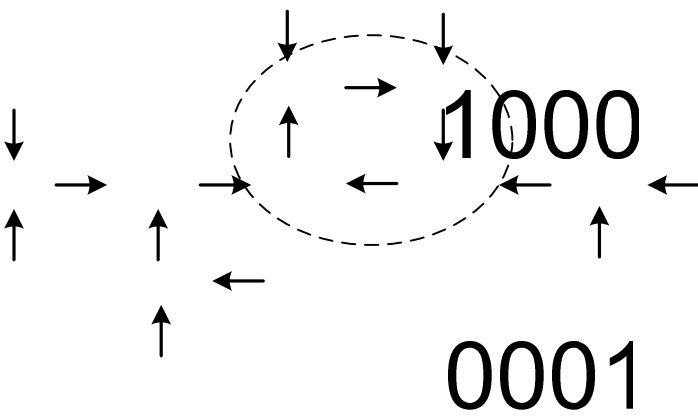
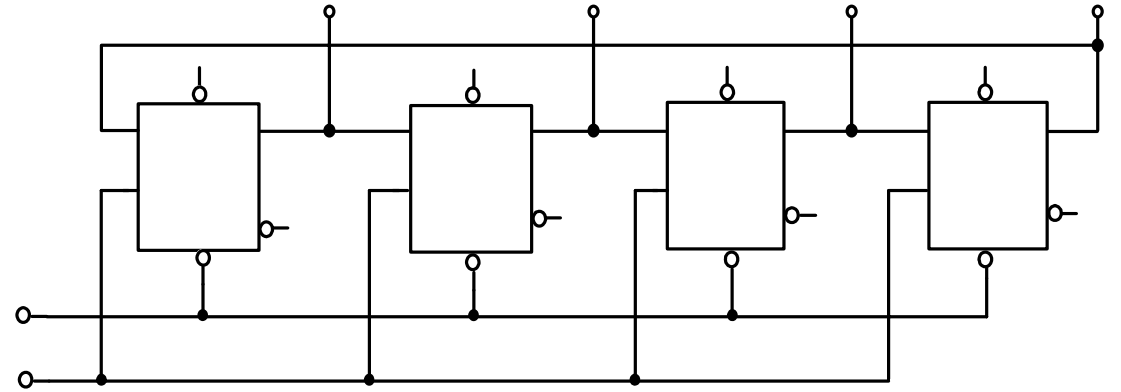
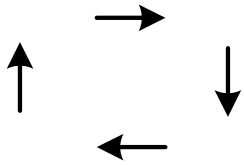
- Các số liệu cần ghi đưa vào D1, D2, D3, D4
- Khi có một xung điều khiển ghi đưa tới lối vào CLK, dữ liệu được nạp vào bộ nhớ song song và cho lối ra song song Q1 Q2 Q3 Q4 = D1 D2 D3 D4.
- Muốn cho dữ liệu tới các lối ra, lối vào “điều khiển ra” phải bằng 1.



Bộ ghi dịch nối tiếp

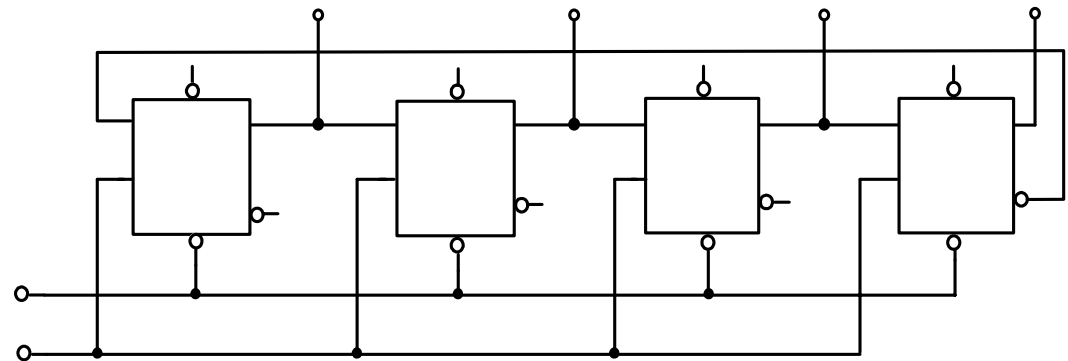
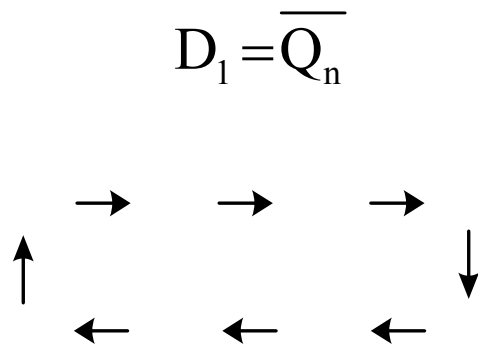
- có thể dịch phải, dịch trái và cho ra song song hoặc ra nối tiếp
- muốn ghi nối tiếp 4 bit cần 4 xung CLK và cho ra ở lối ra song song.
- Còn để lấy số liệu ra nối tiếp cần thêm 3 xung nhịp nữa





Bộ đếm vòng xoắn (mã Johnson)

- là bộ đếm có số bit 1 trong từ mã tăng dần, sau đó lại giảm dần.
- Tương tự có bộ đếm vòng xoắn tự khởi động.



Thanh chốt dữ liệu (Latch)

- là mạch logic số được dùng để lưu trữ trạng thái số (1 hoặc 0) trong bộ lưu trữ dữ liệu.
- thường được sử dụng trong các mạch giao tiếp Bus dữ liệu, các bộ phân kênh, hợp kênh, và trong các mạch điều khiển

D_n	LE	\overline{OE}	O_n
H	\uparrow	L	H
L		L	L
X	X	H	Z

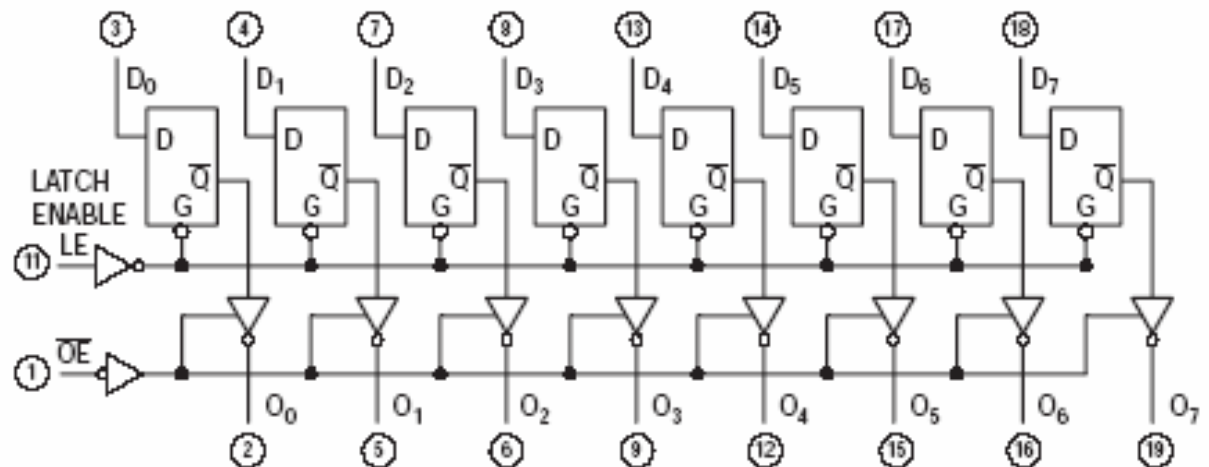
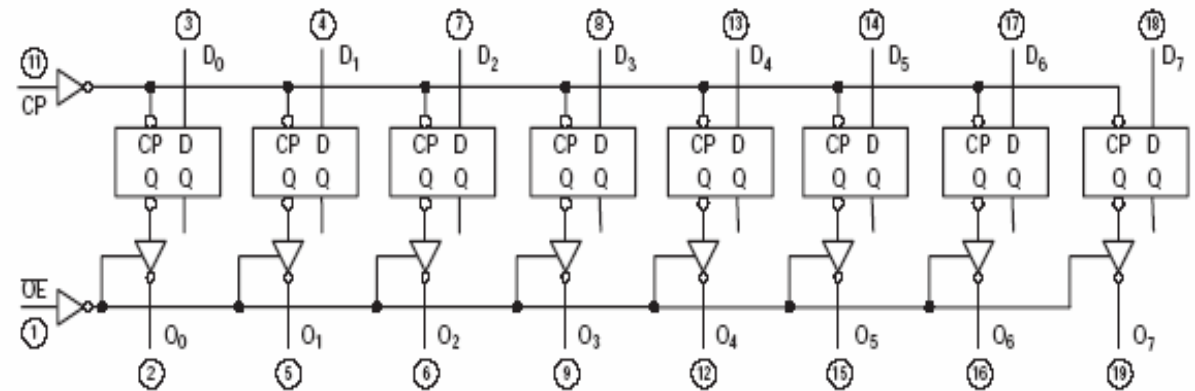
Bảng 5-64b. Bảng chức năng của IC 74374

SN74LS373

D_n	LE	\overline{OE}	O_n
H	H	L	H
L	H	L	L
X	L	L	Q_0
X	X	H	Z

Bảng 5-64a. Bảng chức năng của IC 74373

SN74LS374





Nội dung

Chương 1: Hệ đếm

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

Chương 3: Cổng logic TTL và CMOS

Chương 4: Mạch logic tổ hợp

Chương 5: Mạch logic tuần tự

▪ **Chương 6: Mạch phát xung và tạo dạng xung**

Chương 7: Bộ nhớ bán dẫn

Mạch phát xung và tạo dạng xung

- **Mạch phát xung**

- Mạch dao động đa hài cơ bản cổng NAND TTL
- Mạch dao động đa hài vòng RC
- Mạch dao động đa hài thạch anh
- Mạch dao động đa hài CMOS

- **Trigơ Schmit**

- **Mạch đa hài đợi**

- Mạch đa hài đợi CMOS
- Mạch đa hài đợi TTL

- **IC định thời**

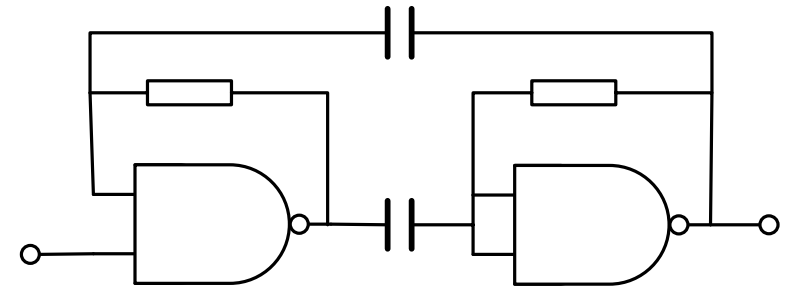


Mạch phát xung

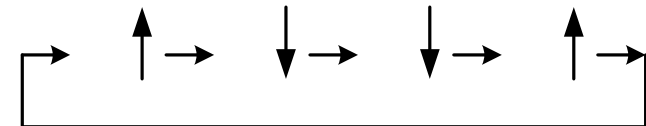
- Mạch dao động đa hài cơ bản cổng NAND TTL
- Mạch dao động đa hài vòng RC
- Mạch dao động đa hài thạch anh
- Mạch dao động đa hài CMOS

Mạch dao động đa hài cơ bản cổng NAND TTL (1)

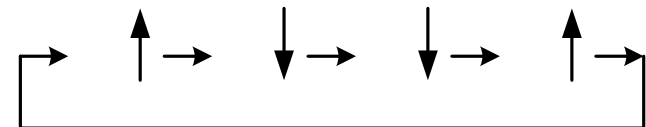
- Cổng NAND khi làm việc trong vùng chuyển tiếp có thể k.đại mạnh tín hiệu đầu vào. 2 cổng NAND được ghép điện dung thành mạch vòng thì có bộ dao động đa hài. V_K là đầu vào điều khiển, khi ở mức cao mạch phát xung, và khi ở mức thấp mạch ngừng phát.
- Nếu các cổng I và II thiết lập điểm công tác tĩnh trong vùng chuyển tiếp và $V_K = 1$, thì mạch sẽ phát xung khi được nối nguồn.
- Nguyên tắc làm việc của mạch:
 - Giả sử do tác động của nhiễu làm cho V_{i1} tăng một chút, lập tức xuất hiện quá trình phản hồi dương (hình 6.2a). Cổng I nhanh chóng trở thành thông bão hoà, cổng II nhanh chóng ngắt, mạch bước vào trạng thái tạm ổn định. Lúc này, C1 nạp điện và C2 phóng điện.
 - C1 nạp đến khi V_{i2} tăng đến ngưỡng thông V_T , trong mạch xuất hiện quá trình phản hồi dương (hình 6.2b). Cổng I nhanh chóng ngắt còn cổng II thông bão hoà, mạch điện bước vào trạng thái tạm ổn định mới. Lúc này C2 nạp điện còn C1 phóng cho đến khi V_{i1} bằng ngưỡng thông VT làm xuất hiện quá trình phản hồi dương đưa mạch về trạng thái ổn định ban đầu.
 - Mạch không ngừng dao động.



Hình 6.1



Hình 6.2a

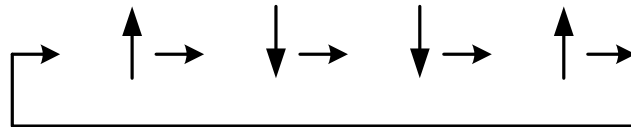


Hình 6.2b

Mạch dao động đa hài cơ bản cổng NAND TTL (2)

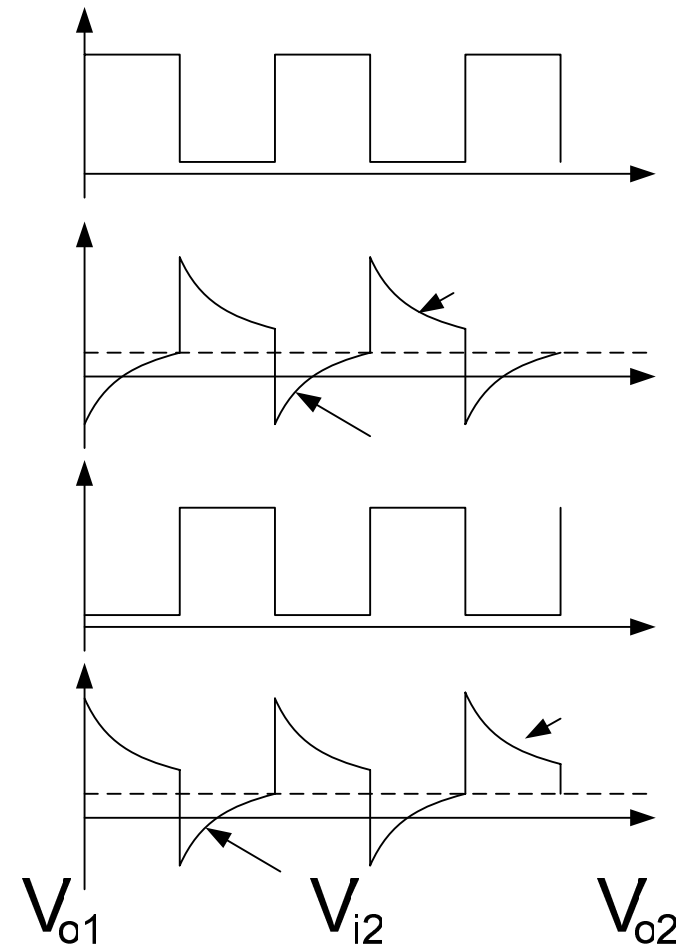
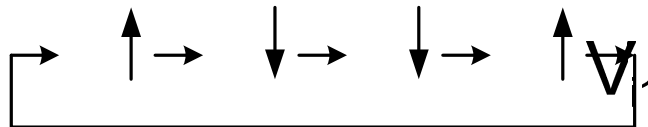
- Giả sử do tác động của nhiễu làm cho V_{i1} tăng một chút, lập tức xuất hiện quá trình phản hồi dương (hình 6.2a). Cổng I nhanh chóng trở thành thông bão hoà, cổng II nhanh chóng ngắt, mạch bước vào trạng thái tạm ổn định. Lúc này, C1 nạp điện và C2 phóng điện.

Hình 6.2a



- C1 nạp đến khi V_{i2} tăng đến ngưỡng thông V_T , trong mạch xuất hiện quá trình phản hồi dương (hình 6.2b). Cổng I nhanh chóng ngắt còn cổng II thông bão hoà, mạch điện bước vào trạng thái tạm ổn định mới. Lúc này C2 nạp điện còn C1 phóng cho đến khi V_{i1} bằng ngưỡng thông V_T làm xuất hiện quá trình phản hồi dương đưa mạch về trạng thái ổn định ban đầu.

Hình 6.2b

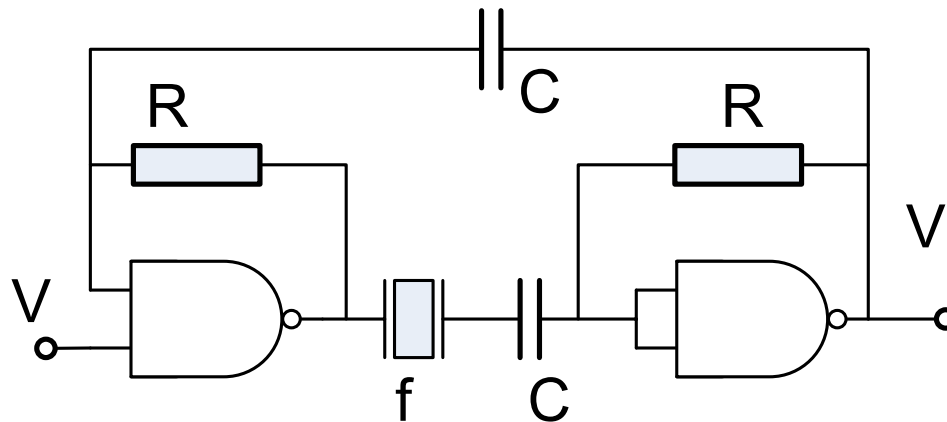


Hình 6.3

- Mạch không ngừng dao động.

Mạch dao động đa hài thạch anh

- Để có các tín hiệu đồng hồ có tần số chính xác và có độ ổn định cao, các mạch đa hài trình bày trên đây không đáp ứng được. Tinh thể thạch anh thường được sử dụng trong các trường hợp này. Thạch anh có tính ổn định tần số tốt, hệ số phẩm chất rất cao dẫn đến tính chọn lọc tần số rất cao.
- Hình dưới là một mạch dao động đa hài điển hình sử dụng tinh thể thạch anh. Tần số của mạch dao động chỉ phụ thuộc vào tinh thể thạch anh mà không phụ thuộc vào giá trị các tụ điện và điện trở trong mạch





- Xem giáo trình



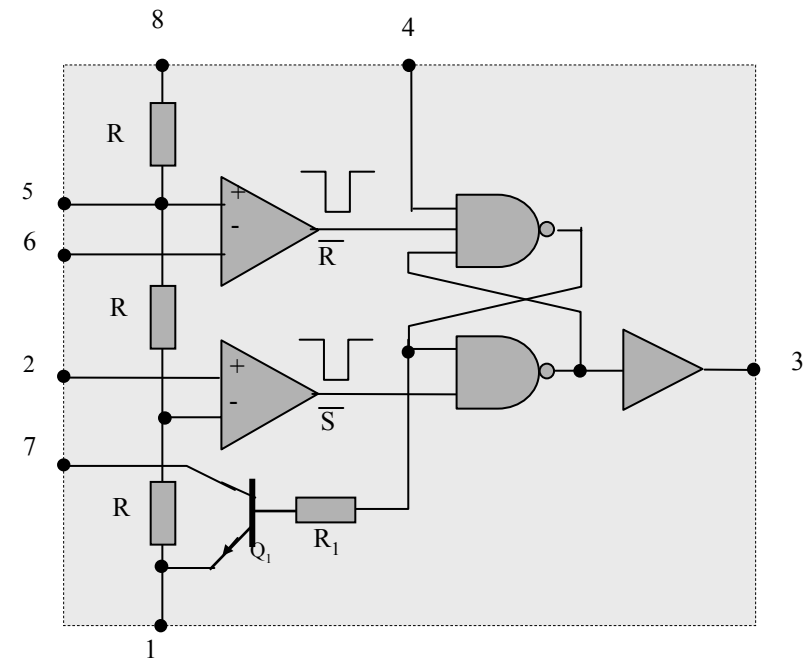
Mạch đa hài đợi

- Xem giáo trình

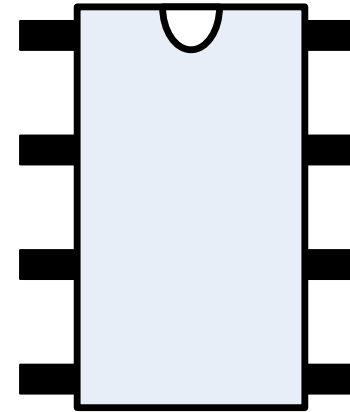
IC định thời (1)

Chân	Chức năng	Chân	Chức năng
1	Đất - GND	5	Điện áp điều khiển
2	Chân kích thích	6	Chân ngưỡng
3	Đầu ra	7	Đầu phóng điện
4	Xoá - Reset	8	Nguồn – Vcc

Bảng 6-1. Bảng mô tả chức năng của các chân trong IC

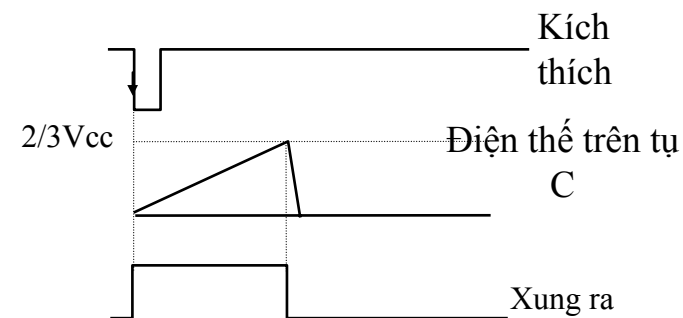
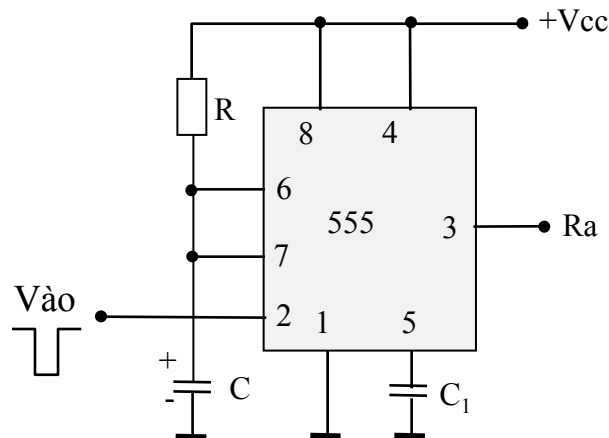
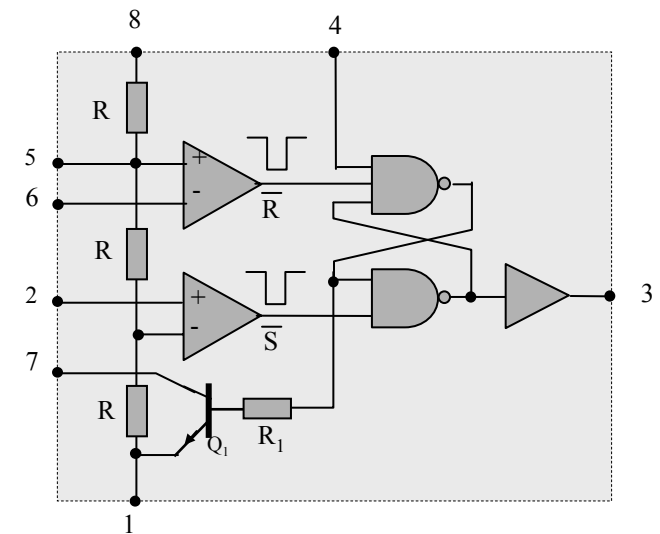


Mạch điện IC 555.



Tạo mạch đơn ổn

- Khi chân 2 nhận kích thích (nổi đất), ta thấy $S\sim$ sẽ lập Q lên 1 và xung sẽ xuất hiện ở lối ra 3. Lúc này, $Q\sim = 0$ nên Q1 khóa. Tụ C nạp điện. Khi điện thế trên tụ (chân 6) vượt quá $2/3V_{cc}$ thì $R\sim = 0$, do đó $Q\sim = 1$. Xung lối ra kết thúc, Q1 thông và tụ C phóng rất nhanh qua Q1. Trạng thái này giữ nguyên cho tới xung kích thích sau (nên chọn R1 lớn để không nóng transistor Q1)
- Độ rộng xung ra được tính theo công thức: **$T = 1,1RC$**
- Tụ C1 thường chọn bằng 0,1uF và có chức năng là tụ lọc để hạn chế nhiễu do nguồn nuôi gây ra.



Tạo mạch dao động đa hài

- Chân 2, 6 và tụ C được nối với nhau, nên điện thế trên tụ sẽ điều khiển đồng thời cả hai bộ so áp. Nếu điện thế này vượt quá mức ngưỡng $2/3 V_{CC}$, thì xung trên đầu ra của TG sẽ bị xoá. Ngược lại, khi tụ phóng xuống dưới mức $1/3 V_{CC}$ thì xung ra lại được lập. Quá trình này sẽ tiếp diễn và cho một chuỗi xung ở lối ra.
- Chu kì của dao động sẽ là:

$$T = T_N + T_P$$

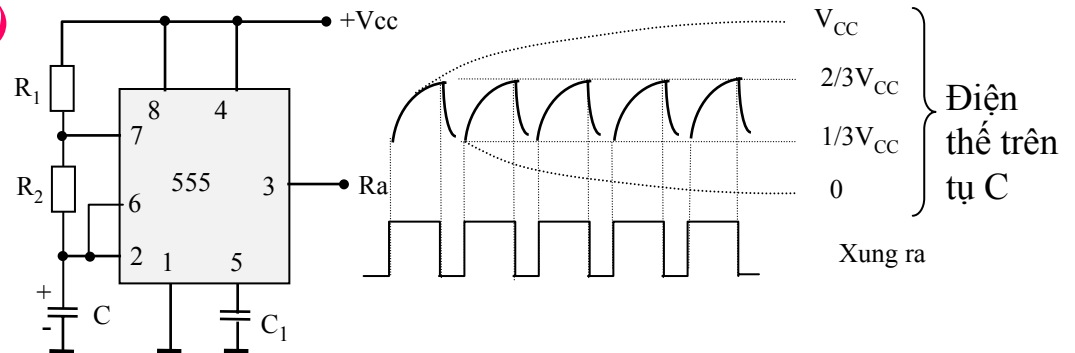
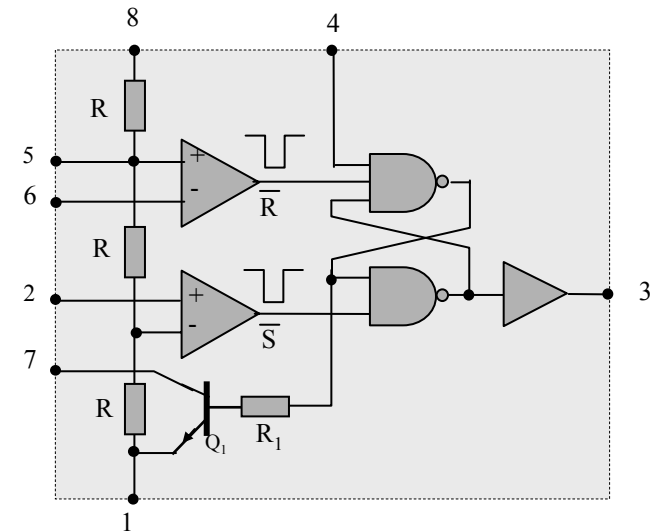
- T_N là thời gian nạp và được tính theo công thức:

$$T_N = 0,7C (R_1 + R_2)$$

- T_P thời gian phóng và bằng:

$$T_P = 0,7.C.R_2$$

- Như vậy: $T = 0,7C (R_1 + 2R_2)$



Tạo mạch dao động – xung vuông

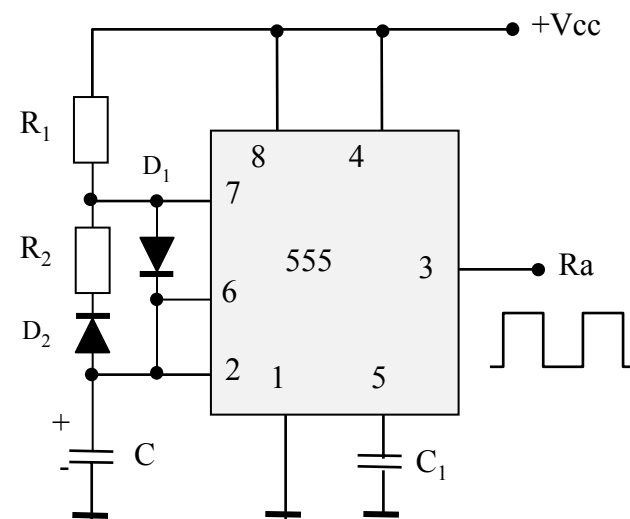
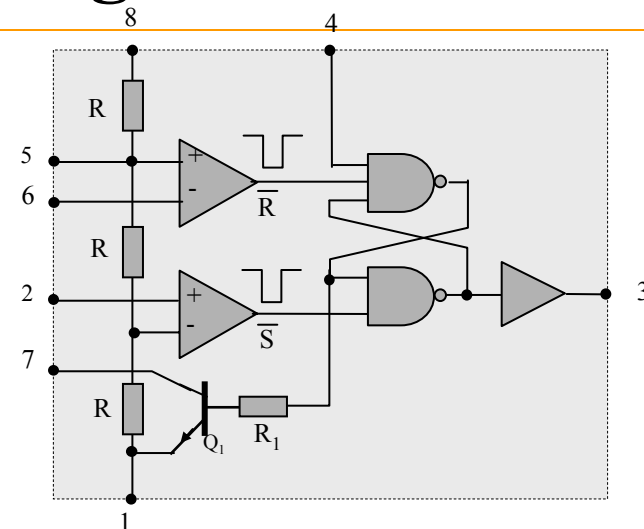
- Các biểu thức trên chỉ ra rằng dãy xung ra chỉ vuông đều khi T_N và T_P bằng nhau, nghĩa là $R_1 = 0$. Điều này không thực tế, vì lúc đó cực C của Q_1 nối trực tiếp với V_{cc} . Khi Q_1 dẫn điện xem như nguồn V_{cc} bị ngắn mạch. Có thể cân bằng T_N và T_P bằng các diode phụ như chỉ ở hình bên.

- Tần số dao động của chuỗi xung ra là:

$$f = \frac{1,4}{C(R_1 + 2R_2)}$$

- Với $R_1 = R_2 = R$ thì (có Diod):

$$f = \frac{0,7}{CR}$$



Hình 6.



Câu hỏi

- **Thiết kế mạch tạo xung dùng 555 cho các tần số 1Hz, 10Hz, 100Hz, 1KHz**



Nội dung

Chương 1: Hệ đếm

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

Chương 3: Cổng logic TTL và CMOS

Chương 4: Mạch logic tổ hợp

Chương 5: Mạch logic tuần tự

Chương 6: Mạch phát xung và tạo dạng xung

■ **Chương 7: Bộ nhớ bán dẫn**

Bộ nhớ bán dẫn



Nội dung

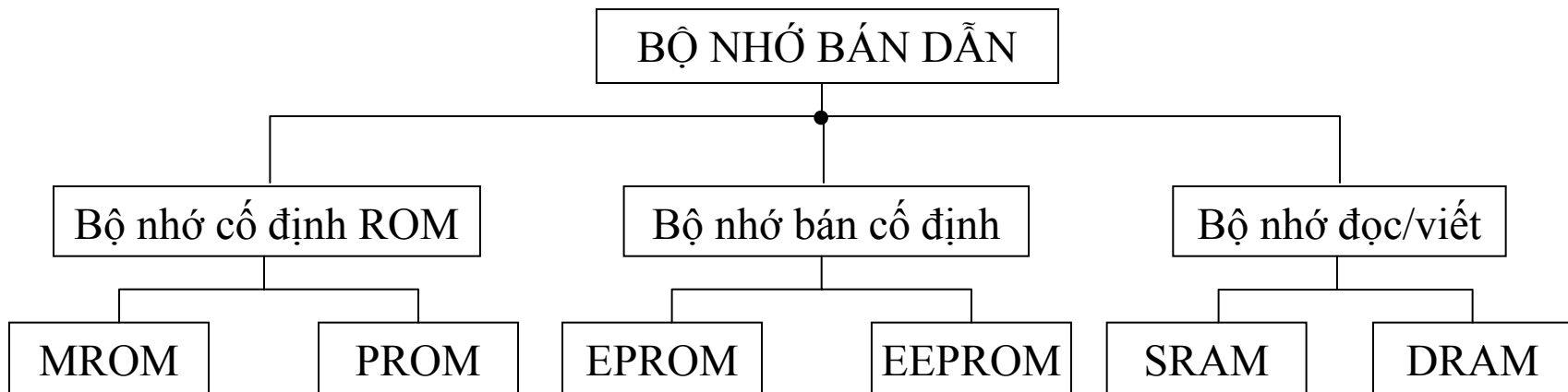
- **Khái niệm chung**
- **DRAM**
- **SRAM**
- **Bộ nhớ cố định – ROM**
- **Bộ nhớ bán cố định**
- **Mở rộng dung lượng bộ nhớ**

■ Khái niệm:

- Bộ nhớ là một thiết bị có khả năng lưu trữ thông tin (nhị phân). Muốn sử dụng bộ nhớ, trước tiên ta phải ghi dữ liệu và các thông tin cần thiết vào nó, sau đó lúc cần thiết phải lấy dữ liệu đã ghi trước đó để sử dụng. Thủ tục ghi vào và đọc ra phải được kiểm soát chặt chẽ, tránh nhầm lẫn nhờ định vị chính xác từng vị trí ô nhớ và nội dung của nó theo một mã địa chỉ duy nhất.

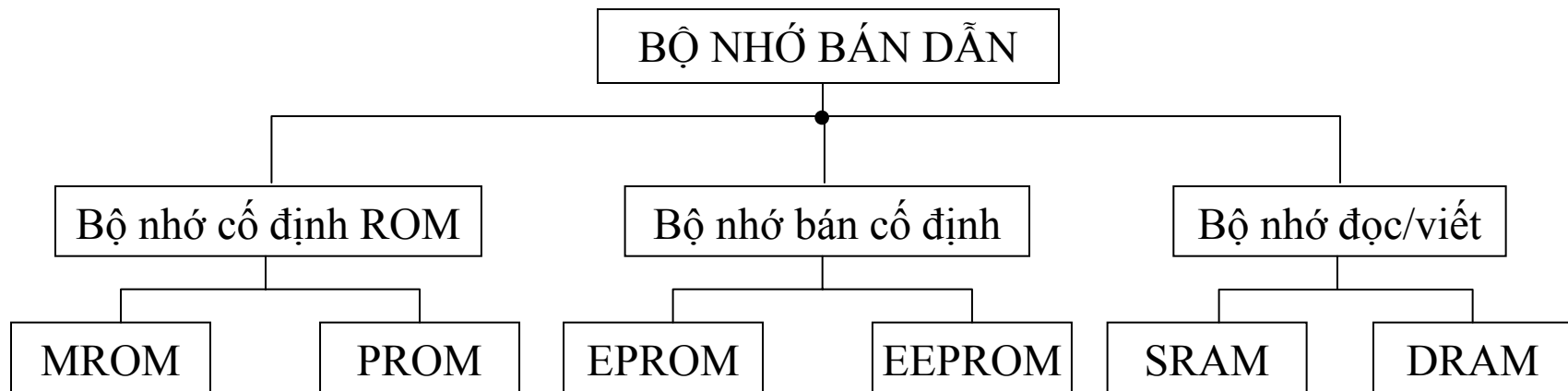
- **Những đặc trưng chính của bộ nhớ**
 - Dung lượng của bộ nhớ.
 - ◆ Dung lượng bộ nhớ là số bit thông tin tối đa có thể lưu giữ trong nó. Dung lượng cũng có thể biểu thị bằng số từ nhớ n bit. **Từ nhớ n bit** là số bit (n) thông tin mà ta có thể đọc hoặc ghi đồng thời vào bộ nhớ. Ví dụ: Một bộ nhớ có dung lượng là 256 bit; nếu nó có cấu trúc để có thể truy cập cùng một lúc 8 bit thông tin, thì ta cũng có thể biểu thị dung lượng bộ nhớ là 32 từ nhớ \times 8 bit = 32 byte.
 - Cách truy cập thông tin: Có 2 cách là trực tiếp và gián tiếp
 - ◆ **Truy cập trực tiếp**, hay còn gọi là truy cập ngẫu nhiên (random access). Ở cách này, không gian bộ nhớ được chia thành nhiều ô nhớ. Mỗi ô nhớ chứa được 1 từ nhớ n bit và có một địa chỉ xác định, mã hoá bằng số nhị phân k bit. Như vậy, người sử dụng có thể truy cập trực tiếp thông tin ở ô nhớ có địa chỉ nào đó trong bộ nhớ. Mỗi bộ nhớ có k bit địa chỉ sẽ có 2^k ô nhớ và có thể ghi được 2^k từ nhớ n bit.
 - ◆ **Truy cập tuần tự** (serial access) hay còn gọi là kiểu truy cập tuần tự. Các đĩa từ, băng từ, trống từ, thanh ghi dịch... có kiểu truy cập này. Các bit thông tin được đưa vào và lấy ra một cách tuần tự.
 - Tốc độ truy cập thông tin.
 - ◆ Đây là thông số rất quan trọng của bộ nhớ. Nó được đặc trưng bởi thời gian cần thiết để truy cập thông tin.

Khái niệm chung – Phân loại bộ nhớ



- **Dựa trên thời gian viết và cách viết, có thể chia thành: bộ nhớ cố định, bộ nhớ bán cố định và bộ nhớ đọc/viết được.**
- **Bộ nhớ cố định ROM (Read Only Memory): có nội dung được viết sẵn một lần.**
 - MROM: là loại ROM sau khi đã được viết (bằng mặt nạ-mask) từ nhà máy thì không viết lại được nữa.
 - PROM là một dạng khác, các bit có thể được viết bằng thiết bị ghi của người sử dụng trong một lần (Programmable ROM).
- **Bộ nhớ có thể đọc/ viết nhiều lần RAM (Random Access Memory) gồm hai loại:**
 - RAM tĩnh-SRAM (Static RAM) thường được xây dựng trên các mạch điện tử trigơ.
 - RAM động-DRAM (Dynamic RAM) được xây dựng trên cơ sở nhớ các điện tích ở tụ điện; bộ nhớ này phải được hồi phục nội dung đều đặn, nếu không nội dung sẽ mất đi theo sự rò điện tích trên tụ.

Khái niệm chung – Phân loại bộ nhớ



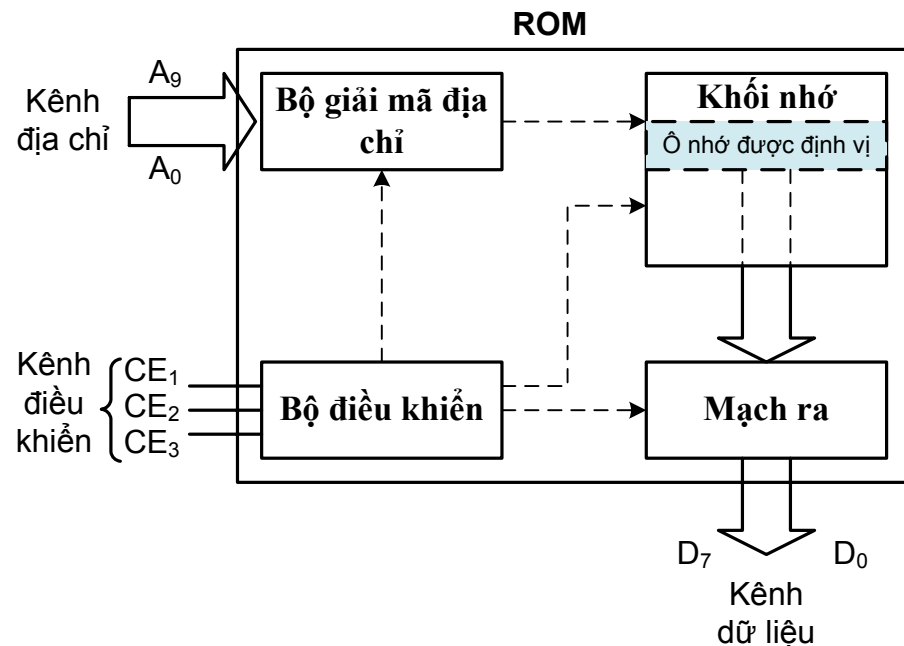
- Giữa ROM và RAM có một lớp các bộ nhớ được gọi là EPROM (Erasable PROM), dữ liệu trong đó có thể xoá được bằng tia cực tím và ghi lại được, EEPROM (Electric EPROM) có thể xoá được bằng dòng điện. Các loại này còn được gọi là bộ nhớ bán cố định.
- Các bộ nhớ DRAM thường thoả mãn những yêu cầu khi cần bộ nhớ có dung lượng lớn; trong khi đó khi cần có tốc độ truy xuất lớn thì phải dùng các bộ nhớ SRAM có giá thành đắt hơn. Nhưng cả hai loại này đều có nhược điểm là thuộc loại “bay hơi” (volatile), thông tin sẽ bị mất đi khi nguồn nuôi bị ngắt. Do vậy các chương trình dùng cho việc khởi động PC như BIOS thường phải nạp trên các bộ nhớ ROM.

Khái niệm chung – Tổ chức của bộ nhớ

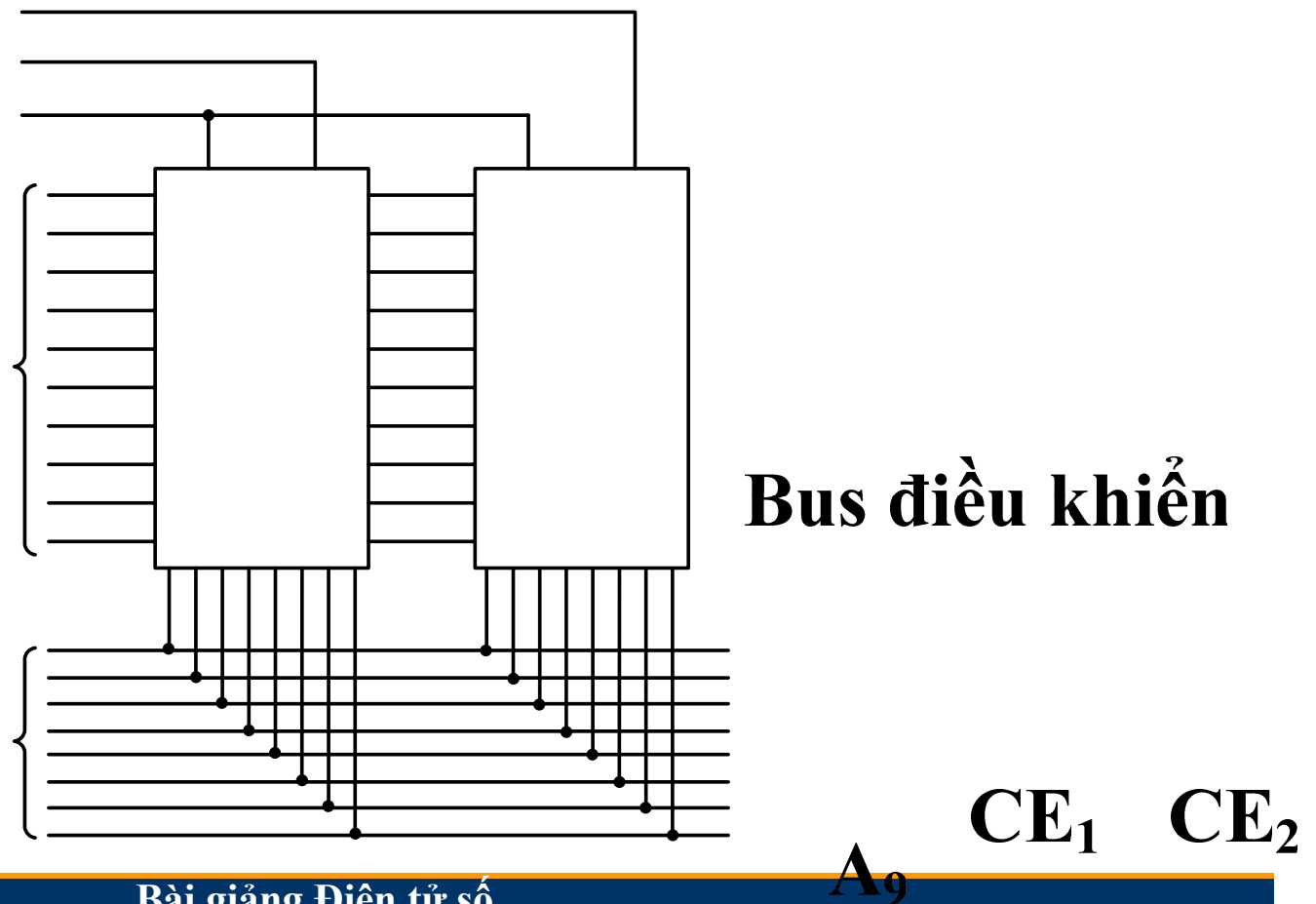
- Bộ nhớ thường được tổ chức gồm nhiều vi mạch nhớ được ghép lại để có độ dài từ và tổng số từ cần thiết. Những chip nhớ được thiết kế sao cho có đầy đủ một số chức năng của bộ nhớ như:
 - Một ma trận nhớ gồm các ô nhớ, mỗi ô nhớ ứng với một bit nhớ.
 - Mạch logic giải mã địa chỉ ô nhớ.
 - Mạch logic cho phép đọc nội dung ô nhớ.
 - Mạch logic cho phép viết nội dung ô nhớ.
 - Các bộ đếm vào, bộ đếm ra và bộ mở rộng địa chỉ.

Cấu trúc cơ bản của bộ nhớ ROM

- ROM bao gồm 4 khối cơ bản:
 - + Bộ nhớ chứa các ô nhớ và trong các ô nhớ là các từ nhớ.
 - + Mạch điều khiển tiếp nhận các tín hiệu vào từ kênh điều khiển.
 - + Bộ giải mã địa chỉ dùng để định vị ô nhớ.
 - + Mạch ra dùng để đưa nội dung ô nhớ tới các thiết bị có liên quan cần tiếp nhận nội dung này.

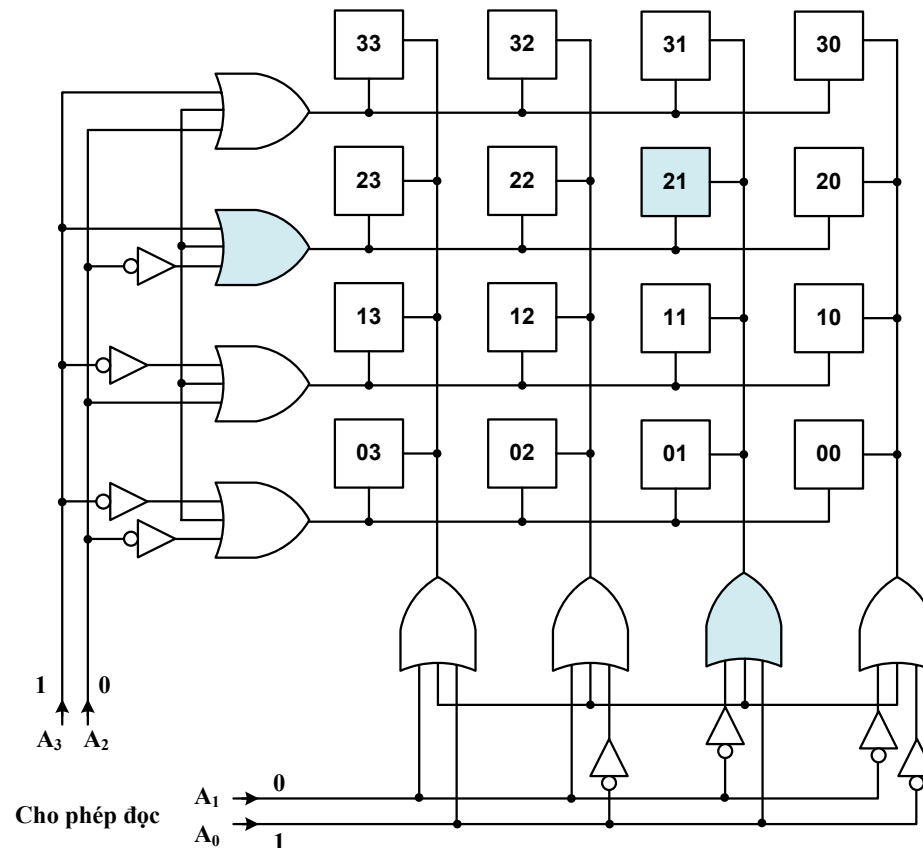


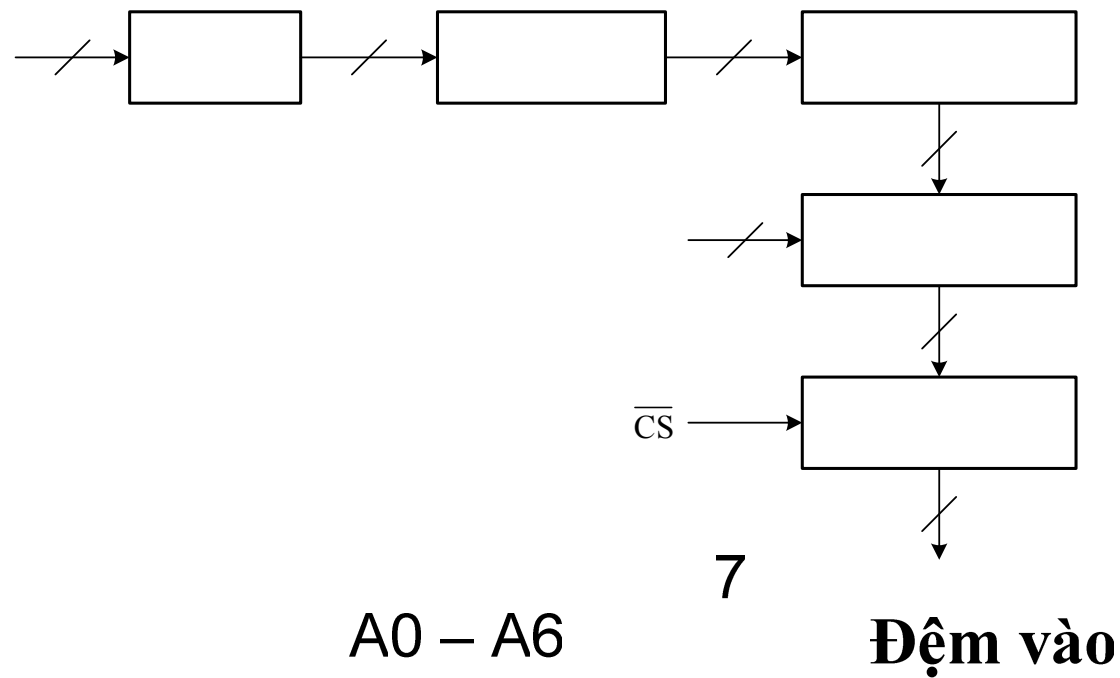
- Mỗi ô nhớ nhị phân có chức năng lưu giữ một trong hai trạng thái 0 hoặc 1.
- Mở rộng bộ nhớ



Bộ giải mã địa chỉ

- Bộ giải mã địa chỉ là giao diện giữa kênh địa chỉ và khối nhớ.
- Nó có khả năng truyền rất nhiều địa chỉ trên một số ít đường truyền.
- Địa chỉ nhị phân phải được giải mã trước khi tác động tới mảng ô nhớ.

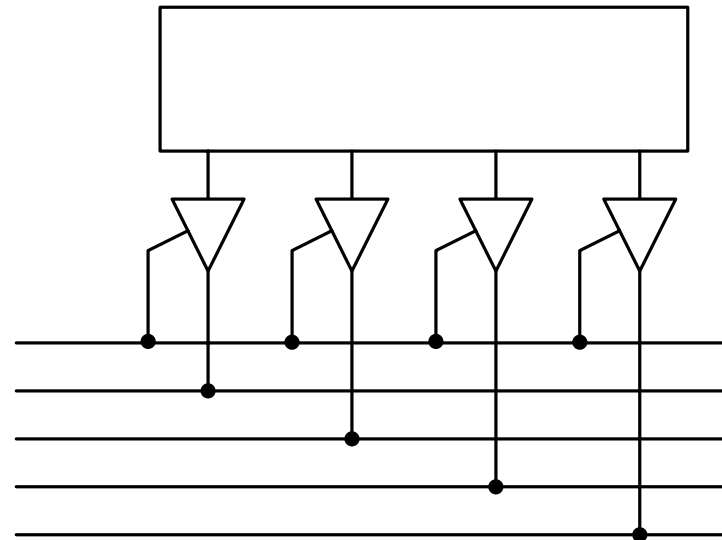




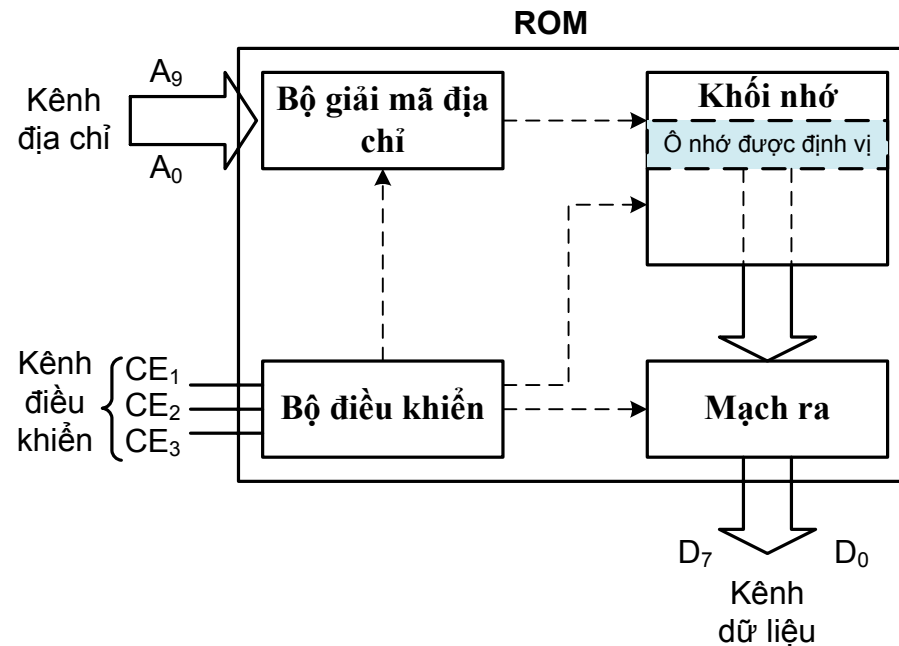
Hình 7-2. Ví dụ về bộ giải mã cho ma trận ROM 128 x 128

Mạch ra của bộ nhớ

- Mạch ra có nhiệm vụ kết nối dữ liệu đã chọn với kênh dữ liệu vào lúc thích hợp.

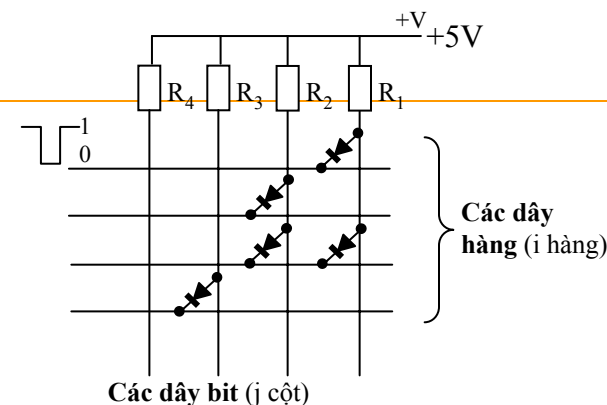


- Mạch điều khiển trong ROM có chức năng khá đơn giản.



Bộ nhớ cố định – MROM

Các chip RAM không thích hợp cho các chương trình khởi động do các thông tin trên đó bị mất khi tắt nguồn. Do vậy phải dùng đến ROM, trong đó các số liệu cần lưu trữ được viết một lần theo cách không bay hơi để nhằm giữ được mãi.



Hình 7-6. MROM diode đơn giản

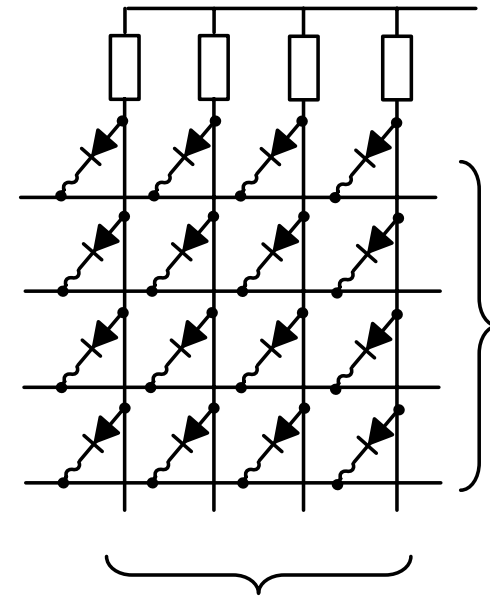
■ MROM – ROM lập trình theo kiểu mặt nạ

- Được chế tạo trên một phiến silic theo một số bước xử lý như quang khắc và khắc tán để tạo ra những tiếp giáp bán dẫn có tính dẫn điện theo một chiều (như diode, transistor trường). Người thiết kế định rõ chương trình muốn ghi vào ROM, thông tin này được sử dụng để điều khiển quá trình làm mặt nạ. Hình 7-6 là một ví dụ đơn giản về sơ đồ MROM dùng diode.
- Chỗ giao nhau giữa các dây từ (hàng) và các dây bit (cột) tạo nên một phần tử nhớ (ô nhớ). Một diode được đặt tại đó (hình vẽ) sẽ cho phép lưu trữ số liệu “0”. Ngược lại những vị trí không có diode thì sẽ cho phép lưu trữ số liệu “1”. Khi đọc một từ số liệu thứ i của ROM, bộ giải mã sẽ đặt dây từ đó xuống mức logic thấp, các dây còn lại ở mức cao. Do vậy chỉ những diode nối với dây này được phân cực thuận, do đó nó sẽ dẫn làm cho điện thế lồi ra trên các dây bit tương ứng ở mức logic thấp, các dây bit còn lại sẽ giữ ở mức cao.

Bộ nhớ cố định – PROM

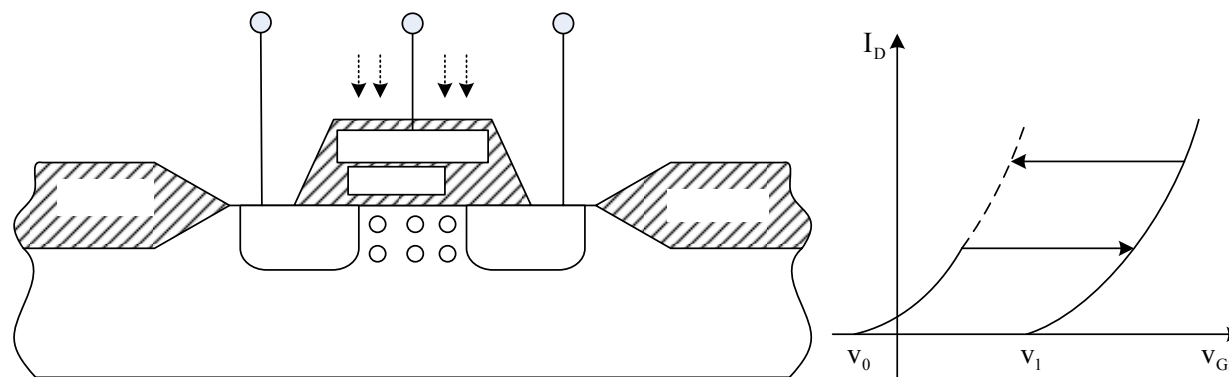
- PROM cũng gồm có các diode như ở MROM nhưng chúng có mặt đầy đủ tạo các vị trí giao nhau giữa dây từ và dây bit. Mỗi diode được nối với một cầu chì.
- Bình thường khi chưa lập trình, các cầu chì còn nguyên vẹn, nội dung của PROM sẽ toàn là 0. Khi định vị đến một bit bằng cách đặt một xung điện ở lối ra tương ứng, cầu chì sẽ bị đứt và bit này sẽ bằng 1. Bằng cách đó ta có thể lập trình toàn bộ các bit trong PROM.
- Như vậy, việc lập trình đó có thể được thực hiện bởi người sử dụng chỉ một lần duy nhất, không thể sửa đổi được.

■ Hình 7-11. PROM dùng diode



Bộ nhớ bán cố định - EPROM (Erasable PROM)

- Số liệu vào có thể được viết vào bằng xung điện nhưng được lưu giữ theo kiểu không bay hơi. Đó là loại ROM có thể lập trình được và xóa được. Hình 7-7 chỉ ra cấu trúc của một transistor dùng để làm một ô nhớ gọi là FAMOST (Floating gate avalanche injection MOS transistor).
- Trong ô nhớ dùng transistor này, cực cửa được nối với đường từ, cực máng được nối với đường bit và cực nguồn được nối với nguồn chuẩn được coi là nguồn cho mức logic 1. Khác với transistor MOS bình thường, transistor loại này còn có thêm một cửa gọi là *cửa nổi* (floating gate); đó là một vùng vật liệu được thêm vào vào giữa lớp cách điện cao như ở hình 7-7. Nếu cửa nổi không có điện tích thì nó không ảnh hưởng gì đến cực cửa điều khiển và transistor hoạt động như bình thường. Tức là khi dây từ được kích hoạt (cực cửa có điện thế dương) thì transistor dẫn, cực máng và nguồn được nối với nhau qua kênh dẫn và dây bit có mức logic 1. Nếu cửa nổi có các điện tử trong đó với điện tích âm thì chúng sẽ ngăn trường điều khiển của cực cửa và dù dây từ được kích hoạt thì cũng không thể phát ra trường đủ mạnh với cực cửa điều khiển để làm thông transistor. Lúc này đường bit không được nối với nguồn chuẩn và ô nhớ coi như được giữ giá trị 0.



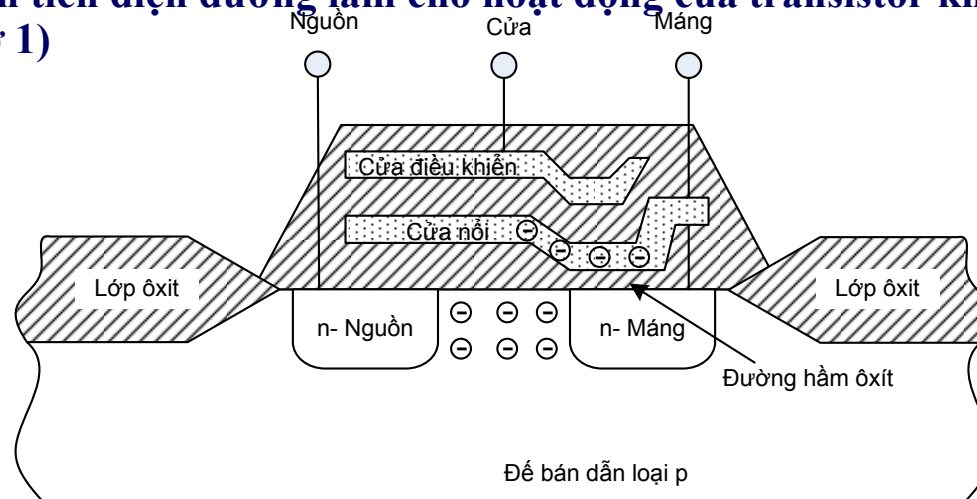
Hình 7-7. Cấu trúc của một EPROM

Bộ nhớ bán cố định - EPROM (Erasable PROM)

- Việc nạp các điện tử vào vùng cửa nổi, tức là tạo ra các ô nhớ mang giá trị 0 được thực hiện bởi xung điện có độ dài cỡ 50 ms và độ lớn + 20 V đặt giữa cực cửa và cực máng. Lúc đó những điện tích mang năng lượng lớn sẽ đi qua lớp cách điện giữa đế và cửa nổi. Chúng tích tụ trong vùng cửa nổi và được giữ ở đây sau khi xung lập trình tắt. Đó là do cửa nổi được cách điện cao với xung quanh và các điện tử không còn đủ năng lượng sau khi lạnh đi, để có thể vượt ra ngoài lớp cách điện đó nữa. Chúng sẽ được giữ ở đây trong một thời gian rất dài (ít nhất là 10 năm).
- Để xoá các thông tin, tức là làm mất các điện tích điện tử trong vùng cửa nổi, phải chiếu ánh sáng tử ngoại UV vào chip nhớ. Lúc này, những điện tử hấp thụ được năng lượng và sẽ nhảy lên các mức năng lượng cao và rời khỏi cửa nổi giống như cách mà chúng đã thâm nhập vào. Trong chip EPROM có một cửa sổ làm bằng thuỷ tinh thạch anh chỉ để cho ánh sáng tử ngoại đi qua khi cần xoá số liệu trong bộ nhớ.

Bộ nhớ bán cố định - EEPROM (Electrically Erasable PROM)

- Cửa sổ thạch anh có giá thành khá đắt và không tiện lợi nên những năm gần đây xuất hiện các chip PROM có thể xóa số liệu bằng phương pháp điện. Cấu trúc của ô nhớ giống như hình 7-8.
- Việc nạp các điện tử cho cửa nổi được thực hiện như cách ở EPROM. Bằng một xung điện tương đối dài, các điện tích mang năng lượng cao được phát ra trong đế sẽ thấm qua lớp cửa ôxít và tích tụ trong cửa nổi. Để xóa EEPROM, một lớp kênh màng mỏng ôxít giữa vùng cửa nổi trải xuống dưới đế và cực máng giữ vai trò quan trọng. Các lớp cách điện không thể là lý tưởng được, các điện tích có thể thấm qua lớp phân cách với một xác suất thấp. Xác suất này tăng lên khi bề dày của lớp giảm đi và điện thế giữa hai điện cực ở hai mặt lớp cách điện tăng lên. Muốn phóng các điện tích trong vùng cửa nổi một điện thế (-20 V) được đặt vào cực cửa điều khiển và cực máng. Lúc này các điện tử âm trong cửa nổi được chảy về cực máng qua kênh màng mỏng ôxít và số liệu lưu giữ được xóa đi. Điều lưu ý là phải làm sao cho dòng điện tích này chảy không quá lâu vì nếu không vùng cửa nổi này lại trở nên tích điện dương làm cho hoạt động của transistor không được trạng thái bình thường (mức nhớ 1)



Hình 7-8. Cấu trúc của một EEPROM



RAM

- RAM có khả năng cho phép ghi lưu trữ dữ liệu thông tin tạm thời trong một thời gian, sau đó lại đọc thông tin đó để tiếp tục xử lý khi cần thiết nên nó có tên là bộ nhớ đọc/viết.
- Một đặc tính quan trọng khác của RAM là các dữ liệu trong RAM chỉ có tính chất tạm thời, dễ bị xóa khi mất nguồn năng lượng cấp

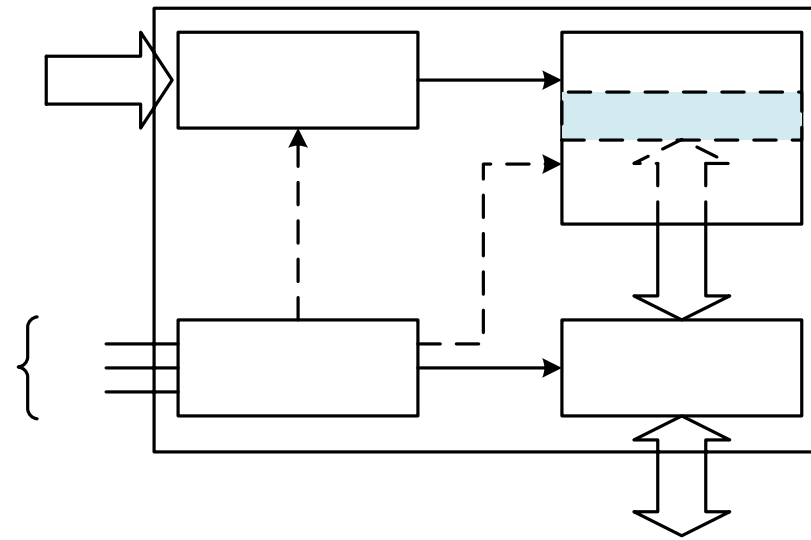


Cấu trúc khối của RAM

- **RAM cũng có 4 phần chính như mô tả trên hình 7-17. Điểm khác biệt là:**
 - + Mạch điều khiển của RAM phải có thêm đầu vào R/W điều khiển hai quá trình cơ bản trong thao tác của RAM: ghi dữ liệu thông tin vào nó và quá trình xuất (đọc) thông tin đã ghi.
 - + Mạch đầu ra có khả năng kiểm soát hai chiều trước khi cho phép giao tiếp với kênh dữ liệu. Quá trình này tuân theo nguyên tắc: (đồng bộ với việc điều khiển R/W) khi bộ nhớ đang đọc thì không được ghi và ngược lại; trạng thái thứ ba có thể chờ quyết định.

Cấu trúc RAM

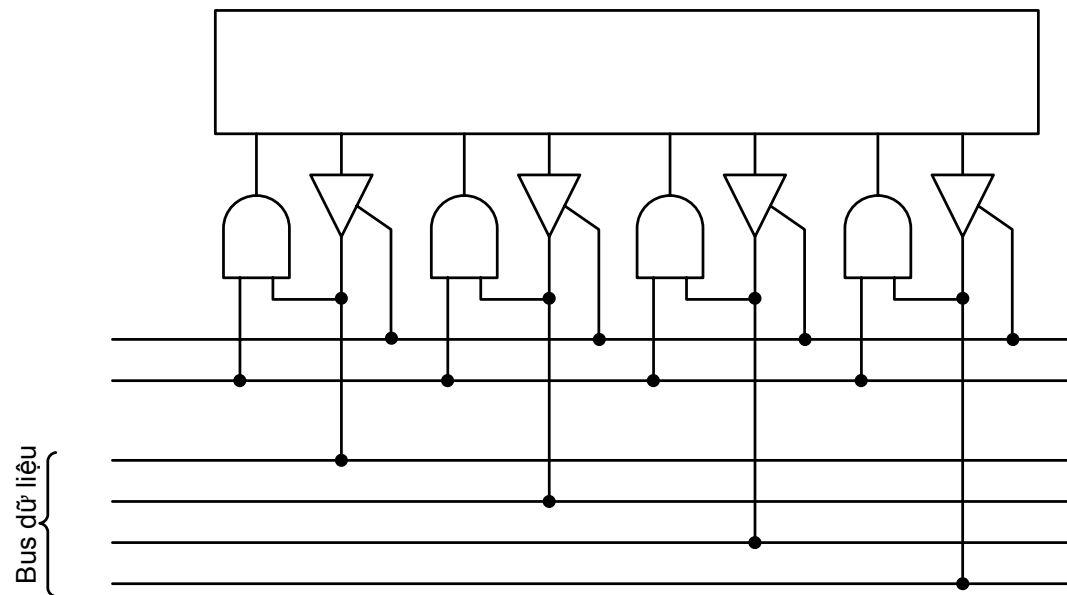
- Cấu trúc 4 khối của một RAM có 8 bit dữ liệu và 8 bit địa chỉ



Kênh
địa chỉ

A_9
 A_0

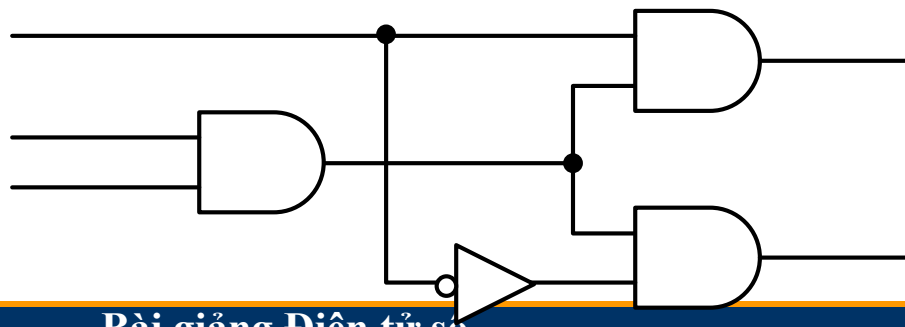
Bộ giải mã địa
chỉ

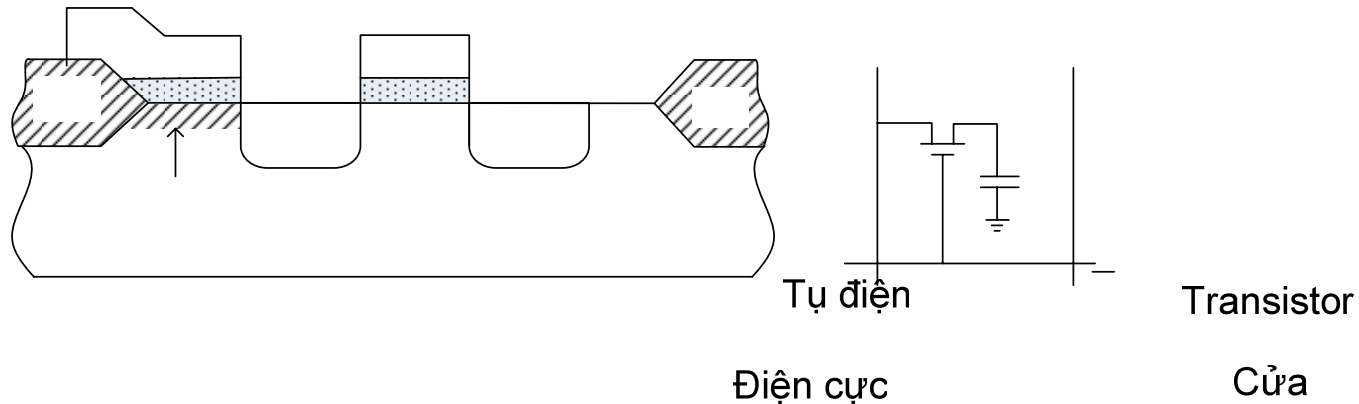


G'_0

Mạch điều khiển

- + Khi ở chế độ đọc, xung R/W ở mức logic 1. Đồng thời các tín hiệu cho phép chọn CE1, CE2 được kích hoạt ở mức 1 nên lúc này $RE = 1$, tức là chế độ đọc được thiết lập. Khi đó tín hiệu $WE = 0$ nên tín hiệu cho phép ghi $WE = 0$ (cấm ghi).
- + Khi ở chế độ ghi, xung R/W ở mức logic 0, $WE = 1$, đồng thời các tín hiệu cho phép chọn CE1, CE2 được kích hoạt ở mức 1 nên lúc này $WE = 1$, tức là nó ở chế độ ghi. Khi đó tín hiệu R/W = 0 nên tín hiệu cho phép đọc $RE = 0$ (cấm đọc).
- + Tín hiệu tích cực đồng thời $CE1 = CE2 = 1$ ở cả hai chế độ đọc và ghi phải được chuyển cùng lúc tới mảng ô nhớ nhằm thông báo việc xuất (khi đọc) hay việc nhập (khi ghi) dữ liệu tới địa chỉ ô nhớ đã được mạch giải mã chọn.
- + Khi tín hiệu $CE1 \cdot CE2 = 0$ (có ít nhất một tín hiệu CE ở trạng thái không tích cực) thì mạch điều khiển ở hình 7-19 sẽ chuyển bộ nhớ sang chế độ chờ (Standby) bất chấp tín hiệu R/W có tích cực hay không, lúc này $RE = 0$ và $WE = 0$.

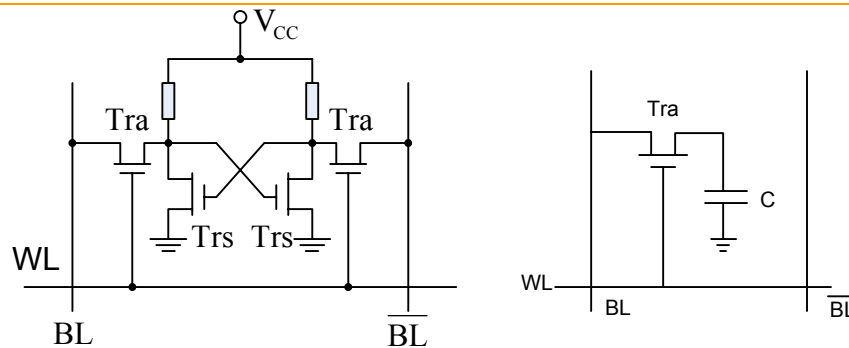




- Các ô nhớ được sắp xếp theo hàng và cột trong một ma trận nhớ. Địa chỉ ô nhớ được chia thành hai phần: địa chỉ hàng và cột. Hai địa chỉ này được đọc vào bộ đệm một cách lần lượt. Xử lý kiểu này được gọi là hợp kênh, lý do là để giảm kích thước bộ giải mã, tức là giảm kích thước và giá thành của bộ giải mã. Quá trình dồn kênh địa chỉ này được điều khiển bởi các tín hiệu RAS (Row Access Strobe) và CAS (Column Access Strobe).

- Nếu \overline{RAS} ở mức tích cực thấp thì DRAM nhận được địa chỉ đặt vào nó và sử dụng như địa chỉ hàng.
- Nếu \overline{CAS} ở mức tích cực thấp thì DRAM nhận được địa chỉ đặt vào nó và sử dụng như địa chỉ cột.

- Một ô nhớ của DRAM gồm có một transistor trường MOS có trở lồi vào rất lớn và một tụ điện C là linh kiện lưu trữ một bit thông tin tương ứng với hai trạng thái có hoặc không có điện tích trên tụ.
- Transistor hoạt động như một công tắc, cho phép nạp hay phóng điện tích của tụ khi thực hiện phép đọc hay viết. Cực cửa (Gate) của transistor được nối với dây hàng (còn gọi là dây từ-WL-Word Line) và cực máng (Drain) được nối với dây cột (còn được gọi là dây bit BL-Bit Line), cực nguồn (Source) được nối với tụ điện. Điện áp nạp trên tụ tương đối nhỏ, vì thế cần sử dụng khuếch đại nhảy trong mạch nhớ.
- Do dòng rò của transistor nên ô nhớ cần được nạp lại trước khi điện áp trên tụ thấp hơn một ngưỡng nào đó. Quá trình này được thực hiện nhờ một chu kỳ “làm tươi” (refresh), khi đó điện áp trên tụ được xác định (ở trạng thái 0 hay 1) và mức điện áp logic này được viết lại vào ô nhớ.



- Một ô nhớ của SRAM giữ thông tin bởi trạng thái của mạch trigrơ. Thuật ngữ “tĩnh” chỉ ra rằng khi nguồn nuôi chưa bị cắt thì thông tin của ô nhớ vẫn được giữ nguyên. Khác với ô nhớ DRAM, ở đây ô nhớ trigrơ cung cấp một tín hiệu số mạnh hơn nhiều vì đã có các transistor trong các ô nhớ, chúng có khả năng khuếch đại tín hiệu và do đó có thể cấp trực tiếp cho các đường bit. Trong DRAM, sự khuếch đại tín hiệu trong các bộ khuếch đại cần nhiều thời gian và do đó thời gian truy nhập dài hơn. Khi định địa chỉ trong các trigrơ ở SRAM, các transistor bổ sung cho các trigrơ, các bộ giải mã địa chỉ...cũng được đòi hỏi như ở DRAM.**

- Như trong DRAM, cực cửa của transistor được nối với đường từ và cực máng nối với cặp đường bit. Nếu số liệu được đọc từ ô nhớ, khi đó bộ giải mã hàng kích hoạt đường dây từ WL tương ứng. Hai transistor T dẫn và nối trigơ nhớ với cặp dây bit. Như vậy hai lối ra Q và $Q_{\bar{}}$ được nối với các đường bit và các tín hiệu được truyền tới bộ khuếch đại ở cuối đường dây này. Vì điện thế chênh lệch lớn nên xử lý khuếch đại như vậy sẽ nhanh hơn trong DRAM (cỡ 10 ns hoặc ngắn hơn), do đó chip SRAM cần địa chỉ cột sớm hơn nếu thời gian truy nhập không được giảm. Như vậy SRAM không cần thực hiện phân kênh các địa chỉ hàng và cột. Sau khi số liệu ổn định, bộ giải mã cột chọn cột phù hợp và cho ra tín hiệu số liệu tới bộ đệm số liệu ra và tới mạch ra.

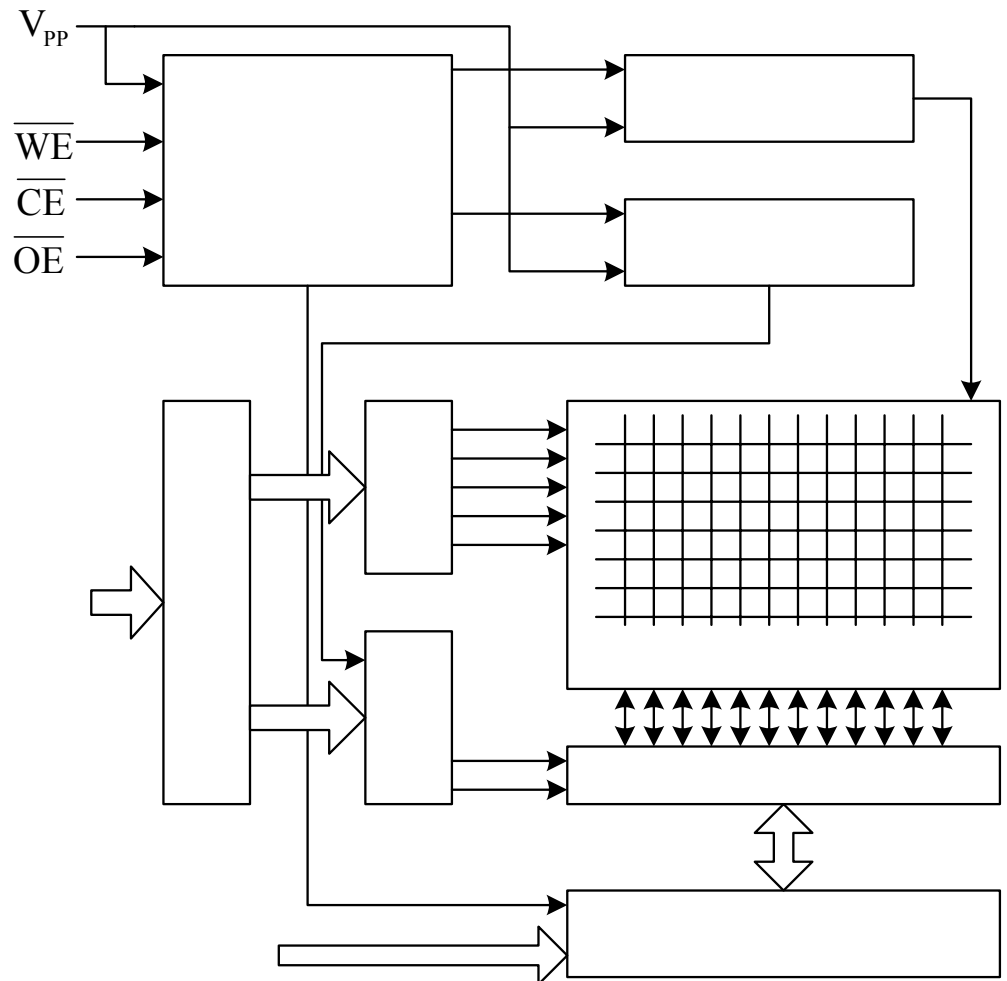
- **Viết số liệu được thực hiện theo cách ngược lại. Qua bộ đệm vào và bộ giải mã cột, số liệu viết được đặt vào bộ khuếch đại phù hợp. Cùng lúc đó bộ giải mã hàng kích hoạt đường dây từ và làm transistor T dẫn. Trigo đưa số liệu được lưu trữ vào cặp dây bit. Tuy vậy, bộ khuếch đại nhạy hơn các transistor nên nó sẽ cấp cho các đường bit một tín hiệu phù hợp với số liệu viết. Do đó, trigo sẽ chuyển trạng thái phù hợp với số liệu mới hoặc giữ giá trị đã được lưu trữ phụ thuộc vào việc số liệu viết trùng với số liệu đã lưu trữ hay không.**

Bộ nhớ bán cố định - Bộ nhớ FLASH

- Trong những năm gần đây, một loại bộ nhớ không bay hơi mới đã xuất hiện trên thị trường, thường được sử dụng thay thế cho các ổ đĩa mềm và cứng trong những máy tính. Đó là bộ nhớ flash. Cấu trúc của chúng cơ bản như EEPROM, chỉ có lớp kênh ôxit ở các ô nhớ mỏng hơn. Do vậy chỉ cần điện thế cỡ 12 V là có thể cho phép thực hiện 10 000 chu trình xoá và lập trình. Bộ nhớ flash có thể hoạt động gần mềm dẻo như DRAM và SRAM nhưng lại không bị mất số liệu khi bị cắt điện. Hình 7- 9 chỉ ra sơ đồ khối của nó.
- Phần chính là mạng nhớ bao gồm các ô nhớ FAMOST như được mô tả ở mục trên. Giống như SRAM, bộ nhớ flash không dùng phân kênh địa chỉ. Các bộ giải mã hàng và cột chọn một đường từ và một hoặc nhiều cặp đường bit. Số liệu đọc được đưa ra ngoài bộ đệm số liệu I/O hoặc được viết vào ô nhớ đã được định địa chỉ bởi bộ đệm này qua cổng I/O. Xử lý đọc được thực hiện với điện thế MOS thông thường là 5V. Để lập trình một ô nhớ, đơn vị điều khiển flash đặt một xung điện thế ngắn cỡ 10 μ s và 12 V gây nên một sự chọc thủng thác lũ vào transistor nhớ để nạp vào cửa nổi. Một chip nhớ flash 1 Mb có thể được lập trình trong khoảng 2 sec, nhưng khác với EEPROM việc xoá được thực hiện từng chip một. Thời gian xoá cho toàn bộ bộ nhớ flash khoảng 1 sec. Xử lý đọc, lập trình và xoá được điều khiển bởi các lệnh có độ dài 2 byte được bộ xử lý viết vào các thanh ghi lệnh của mạch điều khiển flash.

Bộ nhớ bán cố định - Bộ nhớ FLASH

- Mục đích sử dụng chính của bộ nhớ flash là để thay thế cho các ổ đĩa mềm và ổ đĩa cứng dung lượng nhỏ. Do nó là mạch tích hợp nên có ưu điểm là kích thước nhỏ và tiêu thụ năng lượng thấp, không bị ảnh hưởng của va đập. Các đĩa cứng chất rắn dựa trên cơ sở các bộ nhớ flash có lợi thế về công suất tiêu thụ cũng như giá thành có dung lượng tới vài Mbyte. Các card nhớ loại này có ưu điểm là không gặp phải vấn đề mất thông tin như trường hợp RAM CMOS khi pin Ni-Cd bị hỏng. Thời gian lưu trữ thông tin trong bộ nhớ flash ít nhất là 10 năm, thông thường là 100 năm, với khoảng thời gian này thì các đĩa mềm và cứng đã bị hỏng rồi.
- Nhược điểm của bộ nhớ flash là chỉ có thể xóa theo kiểu lần lượt từng chip hoặc lần lượt từng trang.



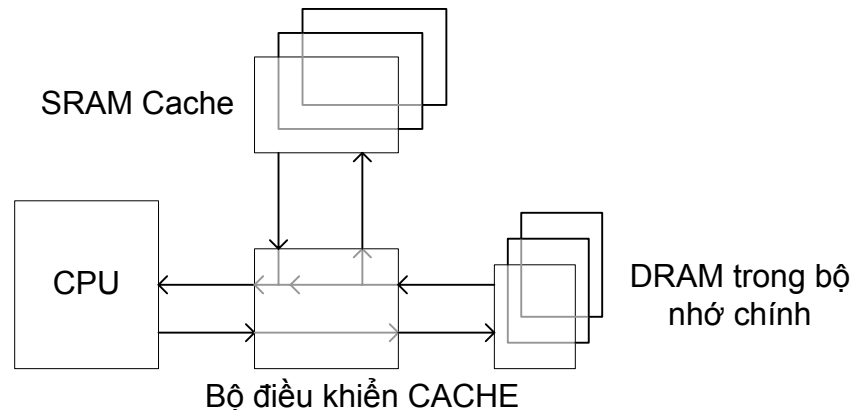
Hình 7-9. Sơ đồ bộ nhớ FLASH

Bộ nhớ bán cố định - EPROM (Erasable PROM)

- Việc nạp các điện tử vào vùng cửa nổi, tức là tạo ra các ô nhớ mang giá trị 0 được thực hiện bởi xung điện có độ dài cỡ 50 ms và độ lớn + 20 V đặt giữa cực cửa và cực máng. Lúc đó những điện tích mang năng lượng lớn sẽ đi qua lớp cách điện giữa đế và cửa nổi. Chúng tích tụ trong vùng cửa nổi và được giữ ở đây sau khi xung lập trình tắt. Đó là do cửa nổi được cách điện cao với xung quanh và các điện tử không còn đủ năng lượng sau khi lạnh đi, để có thể vượt ra ngoài lớp cách điện đó nữa. Chúng sẽ được giữ ở đây trong một thời gian rất dài (ít nhất là 10 năm).
- Để xoá các thông tin, tức là làm mất các điện tích điện tử trong vùng cửa nổi, phải chiếu ánh sáng tử ngoại UV vào chip nhớ. Lúc này, những điện tử hấp thụ được năng lượng và sẽ nhảy lên các mức năng lượng cao và rời khỏi cửa nổi giống như cách mà chúng đã thâm nhập vào. Trong chip EPROM có một cửa sổ làm bằng thuỷ tinh thạch anh chỉ để cho ánh sáng tử ngoại đi qua khi cần xoá số liệu trong bộ nhớ.

Bộ nhớ bán cố định - Bộ nhớ CACHE

- Giữa CPU và bộ nhớ chính bằng DRAM, người ta xen vào một bộ nhớ SRAM nhanh có dung lượng nhỏ bằng 1/10 hoặc 1/100 lần bộ nhớ chính gọi là cache; dưới sự điều khiển của mạch điều khiển cache, bộ nhớ này sẽ lưu trữ tạm thời các số liệu thường được gọi và cung cấp nó cho CPU trong thời gian ngắn.
- Cache chứa các thông tin mới vừa được CPU sử dụng gần đây nhất. Khi CPU đọc số liệu nó sẽ đưa ra một địa chỉ tới bộ điều khiển cache. Sau đó một trong hai quá trình sau sẽ xảy ra:
- Cache hit: nếu địa chỉ đó đã có sẵn trong RAM cache.
- Cache miss: ngược lại, nếu địa chỉ đó không có sẵn trong RAM cache.

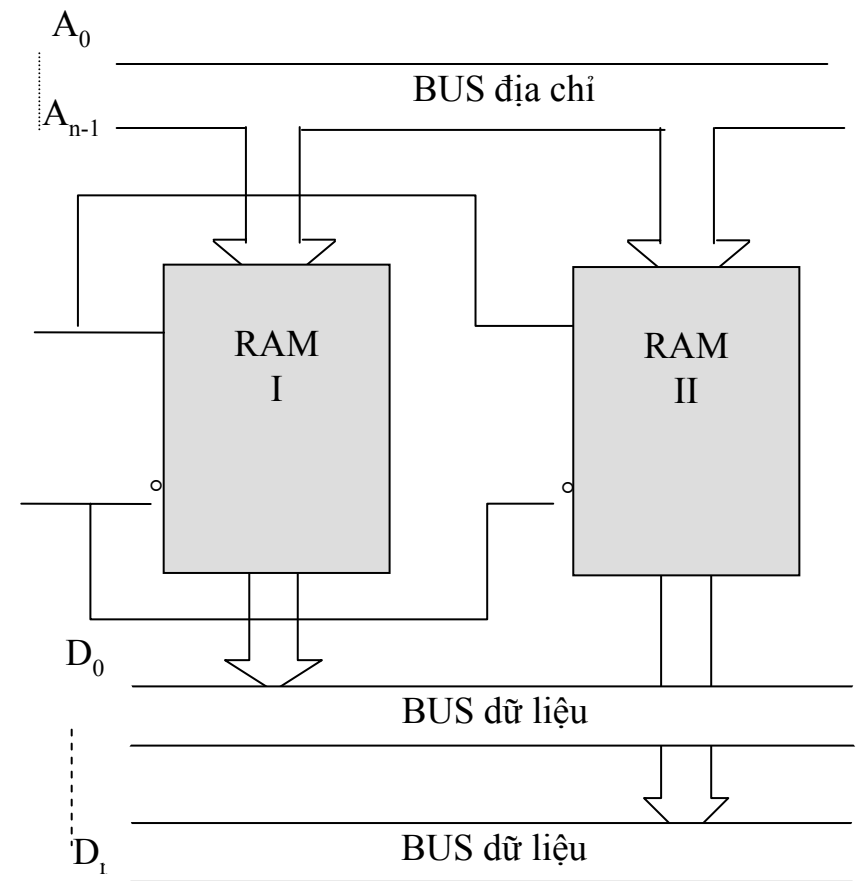


Mở rộng dung lượng bộ nhớ

- **Các vi mạch nhớ bán dẫn chỉ có dung lượng xác định. Muốn có bộ nhớ có dung lượng lớn hơn, ta tìm cách ghép nhiều vi mạch nhớ nhằm một trong ba mục đích sau:**
 - Tăng độ dài nhớ, nhưng không làm tăng số lượng từ nhớ.
 - Tăng số lượng từ nhớ nhưng không làm tăng độ dài từ nhớ.
 - Tăng cả số lượng và độ dài từ nhớ.

Mở rộng độ dài từ

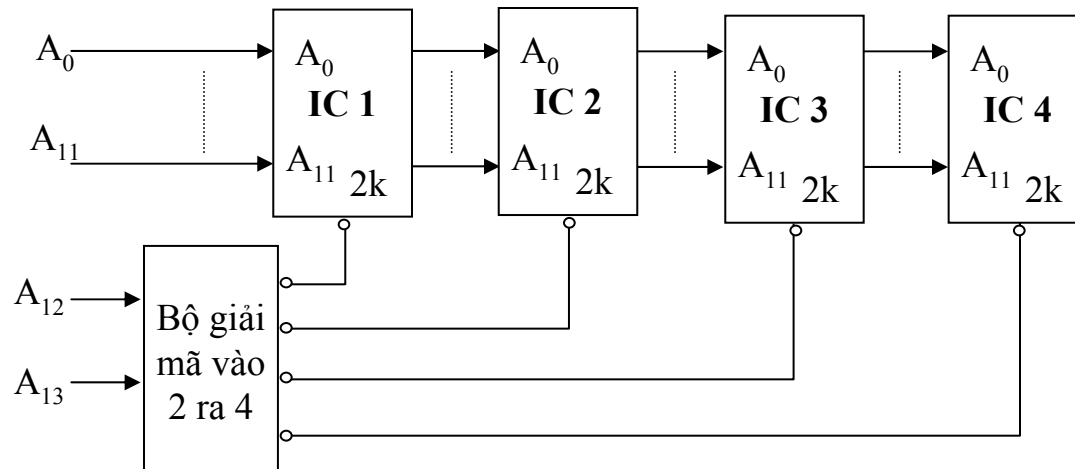
- Trên một chip nhớ, có thể có được 1 đến một số hữu hạn lối ra, thường là 4 hoặc 8 bit. Muốn có độ dài từ lớn hơn, chẳng hạn từ 4 lên 8 hoặc 16 bit, ta tiến hành ghép nhiều chip nhớ như chỉ ở hình 7-10 đối với RAM. Đối với ROM cách làm cũng tương tự, chỉ khác trong trường hợp này, có thể không có lối vào R/ \bar{W} .



Hình 7-10. Sơ đồ mở rộng độ dài từ.

Mở rộng dung lượng (1)

- Muốn mở rộng dung lượng, ta cũng ghép nhiều chip lại với nhau. Như đã biết, dung lượng có liên quan đến số lối vào địa chỉ ($C = 2N \times \text{độ dài từ}$, với N là số lối vào địa chỉ). Cứ tăng 1 chip thì cần có thêm một lối vào địa chỉ.
- Khác với trường hợp mở rộng độ dài từ, khi mở rộng dung lượng các lối vào/ra dữ liệu D và R được nối song song. Một phần dung lượng được trữ vào mỗi chip. Sự phân chia này dựa trên cơ sở tổ hợp địa chỉ vào và lối vào điều khiển. Hình 7-11 là một sơ đồ ví dụ.

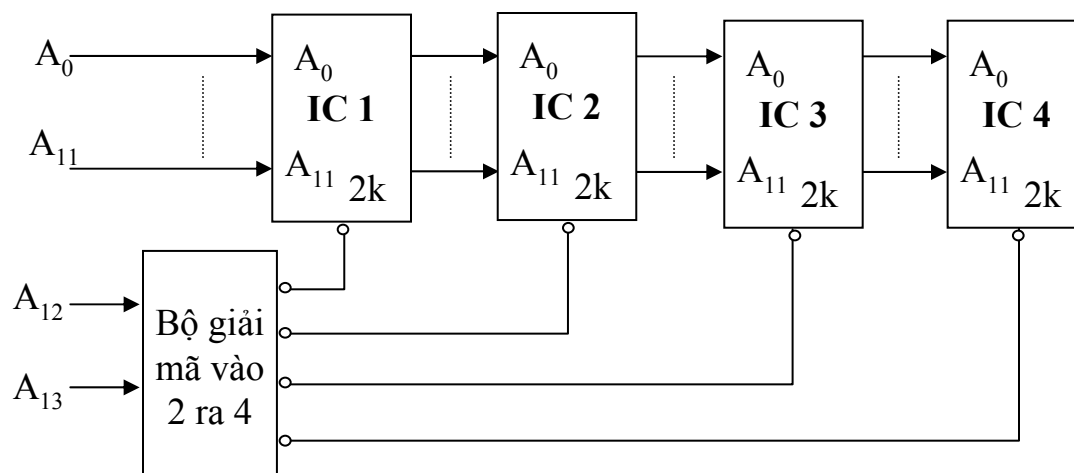


Hình 7-11. Phương pháp mở rộng dung lượng.

Mở rộng dung lượng (2)

- Để thực hiện phép mở rộng ta phải sử dụng một số lối vào địa chỉ dành riêng cho bộ giải mã (thường là các địa chỉ có trọng số cao). Ở sơ đồ trên ta chọn 2 địa chỉ A12 và A13 để giải mã. Do đó ta có thể nhận được 4 giá trị ra tương ứng. Các giá trị này tác động lên các lối vào CS để mở tuần tự các IC nhớ. Các IC nhớ này có thể làm ROM hoặc RAM hoặc cả hai là tùy chọn. Tuần tự mở các IC theo A12, A13 như chỉ ra ở bảng hoạt động sau.

A_{13}	A_{12}	CS	IC mở	Khoảng địa chỉ
0	0	CS_1	IC I	$0000_{16} - 0FFF_{16}$
0	1	CS_2	IC II	$1000_{16} - 1FFF_{16}$
1	0	CS_3	IC III	$2000_{16} - 2FFF_{16}$
1	1	CS_4	IC IV	$3000_{16} - 3FFF_{16}$



Hình 7-11. Phương pháp mở rộng dung lượng.