



# HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



## BÀI GIẢNG MÔN KỸ THUẬT SỐ

**Giảng viên:**

**ThS. Lương Công Duẩn**

**Điện thoại/E-mail:**

**0916566268; duanlc@ptit.edu.vn**

**Bộ môn:**

**Điện tử máy tính - Khoa KTĐT1**

**Học kỳ/Năm biên soạn: Học kỳ 2/2010-2011**

- ***Bài giảng Điện tử số* - Nguyễn Trung Hiếu & Trần Thị Thúy Hà, Học viện CNBCVT**
- ***Giáo trình Điện tử số* - Trần Thị Thúy Hà & Đỗ Mạnh Hà, NXB Thông tin và truyền thông 2009.**
- ***Giáo trình Kỹ thuật số* - Trần Văn Minh, NXB Bưu điện 2001.**
- *Cơ sở kỹ thuật điện tử số*, Đại học Thanh Hoa, Bắc Kinh, NXB Giáo dục 1996.
- *Kỹ thuật số*, Nguyễn Thúy Vân, NXB Khoa học và kỹ thuật 1994.
- *Lý thuyết mạch logic và Kỹ thuật số*, Nguyễn Xuân Quỳnh, NXB Bưu điện 1984.
- *Fundamentals of logic design*, fourth edition, Charles H. Roth, Prentice Hall 1991.
- *Digital engineering design*, Richard F.Tinder, Prentice Hall 1991.
- *Digital design principles and practices*, John F.Wakerly, Prentice Hall 1990.



Chương 1: Hệ đếm

Chương 2: Đại số Boole

Chương 3: Cổng logic

Chương 4: Mạch logic tổ hợp

Chương 5: Mạch logic tuần tự

# HỆ ĐẾM

## ➤ Biểu diễn số

Chuyển đổi giữa các hệ đếm

Số nhị phân có dấu

Dấu phẩy động

Một số loại mã nhị phân thông dụng

# Biểu diễn số (1)

## ➤ Nguyên tắc chung

- Hệ đếm (hay hệ thống số) là một hệ gồm các ký hiệu ghép với nhau theo qui ước về vị trí.
  - + Các ký hiệu thường được gọi là chữ 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 chữ số ở vị trí thứ  $i$  trong một hệ đếm cơ số  $r$  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 ( $r$ )
<b>Hệ nhị phân (Binary)</b>	<b>0, 1</b>	<b>2</b>
<b>Hệ bát phân (Octal)</b>	<b>0, 1, 2, 3, 4, 5, 6, 7</b>	<b>8</b>
<b>Hệ thập phân (Decimal)</b>	<b>0, 1, 2, 3, 4, 5, 6, 7, 8, 9</b>	<b>10</b>
<b>Hệ thập lục phân (Hexadecimal)</b>	<b>0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F</b>	<b>16</b>

Chú ý: Cũng có thể gọi hệ đếm theo cơ số của chúng. VD: Hệ nhị phân = Hệ cơ số 2, Hệ thập phân = Hệ cơ số 10...

➤ Biểu diễn số tổng quát:

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

➤ 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:

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

Trong đó:

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

➤ Biểu diễn tổng quát:

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

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 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.

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

# Hệ nhị phân (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:

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

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.

# Hệ bát phân (2)

## ➤ 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.

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

## ➤ 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:

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

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.

Biểu diễn số

➤ Chuyển đổi cơ số giữa các hệ đếm

Số nhị phân có dấu

Dấu phẩy động

Một số loại mã nhị phân thông dụng



# 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

**10110**

Đổi với phần phân số:

Bước	Nhân	Kết quả	Phần nguyên
1	$0.125 \times 2$	0.25	0
2	$0.25 \times 2$	0.5	0
3	$0.5 \times 2$	1	1
4	$0 \times 2$	0	0

**0.001**

- **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	↑
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	

**1010011**

■ Đổi với phần phân số:

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

**0.11011110**

**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:  $1101110.10_2 = 6E.8$

Biểu diễn số

Chuyển đổi cơ số giữa các hệ đếm

➤ Số nhị phân có dấu

Dấu phẩy động

Một số loại mã nhị phân thông dụng

# 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: **11111011**.

## ➤ 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: **11111100**.

# 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:**
  - + **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.**
  - + **Số dương nhỏ 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 cùng dấu:

- + **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:

- + **Số dương lớn hơn:** cộng số dương với bù 1 của số âm. **Bit tràn** được cộng vào kết quả.
- + **Số dương nhỏ 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 theo bù 1: Hai số cùng dấu

- *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 theo bù 1: Hai số khác dấu

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

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

- *Số dương nhỏ 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 cùng dấu:

- + **Hai số dương:** cộng như cộng nhị phân thông thường. Kết quả là dương.
- + **Hai số âm:** cộng bù 2 của hai số hạng, kết quả xuất hiện một bit tràn, **bỏ bit tràn** đi được kết quả ở dạng bù 2.

### – Hai số khác dấu

- + **Số dương lớn hơn:** lấy số dương cộng với bù 2 của số âm. Kết quả xuất hiện một bit tràn, **bỏ bit tràn** đi được kết quả ở dạng bù 2.
- + **Số dương nhỏ hơn:** lấy số dương cộng với bù 2 của số âm. Kết quả không xuất hiện bit tràn và ở dạng bù 2.

## ➤ 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 theo bù 2: Hai số cùng dấu

- *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*: cộng bù 2 của hai số hạng, kết quả xuất hiện một bit tràn, **bỏ bit tràn** đi được kết quả ở dạng bù 2.

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

# Cộng theo bù 2: Hai số khác dấu

- *Số dương lớn hơn:* lấy số dương cộng với bù 2 của số âm. Kết quả xuất hiện một bit tràn, **bỏ bit tràn** đi được kết quả ở dạng bù 2.

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

- *Số dương nhỏ hơn:* lấy số dương cộng với bù 2 của số âm. Kết quả không xuất hiện bit tràn và ở dạng bù 2.

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

Biểu diễn số

Chuyển đổi cơ số giữa các hệ đếm

Số nhị phân có dấu

Dấu phẩy động

➤ Một số loại mã nhị phân thông dụng

# Một số loại mã nhị phân thông dụng

Số thập phân	Trọng số của mã BCD				
	8421	7421	5121	2421	4221
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0	0 0 1 0	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1
4	0 1 0 0	0 1 0 0	0 1 1 1	0 1 0 0	1 0 0 0
5	0 1 0 1	0 1 0 1	1 0 0 0	1 0 1 1	0 1 1 1
6	0 1 1 0	0 1 1 0	1 0 0 1	1 1 0 0	1 1 0 0
7	0 1 1 1	1 0 0 0	1 0 1 0	1 1 0 1	1 1 0 1
8	1 0 0 0	1 0 0 1	1 0 1 1	1 1 1 0	1 1 1 0
9	1 0 0 1	1 0 1 0	1 1 1 1	1 1 1 1	1 1 1 1

*Cấu tạo của mã BCD với các trọng số khác nhau*



# Một số loại mã nhị phân thông dụng

Số thập phân	Số nhị phân	Mã dư 3	Mã Gray	Mã Gray Dư 3	Mã Johnson	Mã vòng
0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 1 0	0 0 0 0 0	0 0 0 0 0 0 0 0 0 1
1	0 0 0 1	0 1 0 0	0 0 0 1	0 1 1 0	1 0 0 0 0	0 0 0 0 0 0 0 0 0 1 0
2	0 0 1 0	0 1 0 1	0 0 1 1	0 1 1 1	1 1 0 0 0	0 0 0 0 0 0 0 0 1 0 0
3	0 0 1 1	0 1 1 0	0 0 1 0	0 1 0 1	1 1 1 0 0	0 0 0 0 0 0 0 1 0 0 0
4	0 1 0 0	0 1 1 1	0 1 1 0	0 1 0 0	1 1 1 1 0	0 0 0 0 0 1 0 0 0 0 0
5	0 1 0 1	1 0 0 0	0 1 1 1	1 1 0 0	1 1 1 1 1	0 0 0 0 1 0 0 0 0 0 0
6	0 1 1 0	1 0 0 1	0 1 0 1	1 1 0 1	0 1 1 1 1	0 0 0 1 0 0 0 0 0 0 0
7	0 1 1 1	1 0 1 0	0 1 0 0	1 1 1 1	0 0 1 1 1	0 0 1 0 0 0 0 0 0 0 0
8	1 0 0 0	1 0 1 1	1 1 0 0	1 1 1 0	0 0 0 1 1	0 1 0 0 0 0 0 0 0 0 0
9	1 0 0 1	1 1 0 0	1 1 0 1	1 0 1 0	0 0 0 0 1	1 0 0 0 0 0 0 0 0 0 0

*Cấu tạo của một số mã nhị phân thông dụng*

Chọn phương án đúng (A hoặc B, C, D)

- 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

- 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

- 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

Chương 1: Hệ đếm

➤ **Chương 2: Đại số Boole**

Chương 3: Cổng logic

Chương 4: Mạch logic tổ hợp

Chương 5: Mạch logic tuần tự

# Đại số Boole

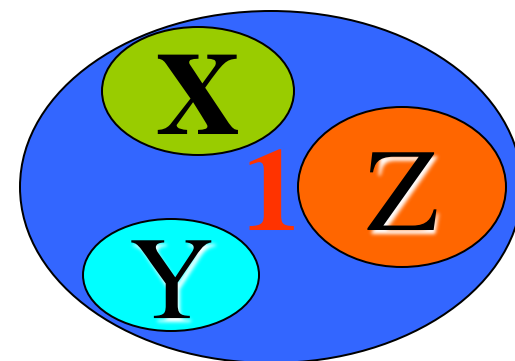
## ➤ Đại số Boole

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 đúp	$\bar{\bar{X}} = X$	
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 tính chất:

- 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 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

Ví dụ:  $f = A.B.C$



# Phương pháp Bảng Các nô (Karnaugh)

- Tổ chức của bảng Các nô:
  - Các tổ hợp biến được viết theo một dòng (thường là phía trên) và một cột (thường là bên trái).
  - Một hàm logic có  $n$  biến sẽ có  $2^n$  ô.
  - Mỗi ô thể hiện một hạng tích hay một hạng tổng, các hạng tích trong hai *ô kế cận* chỉ khác nhau một biến.
- Tính tuần hoàn của bảng Các nô:
  - Không những các *ô kế cận khác nhau một biến* mà *các ô đầu dòng và cuối dòng, đầu cột và cuối cột* cũng chỉ *khác nhau một biến* (kể cả 4 góc vuông của bảng). Bởi vậy *các ô này cũng gọi là kế cận*.
- Thiết lập bảng Các nô 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.

A \ B	0	1
0		
1		

A \ BC	00	01	11	10
0				
1				

AB \ CD	00	01	11	10
00				
01				
11				
10				

- Có 2 dạng biểu diễn là *tuyến* (tổng các tích) và *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

- Rút gọn mạch logic tổ hợp có một vai trò quan trọng trong việc tối giản các thiết kế mạch logic tổ hợp.
- Có 3 phương pháp phổ biến được sử dụng để tối giản mạch logic tổ hợp:
  - Phương pháp đại số,
  - Phương pháp bảng Karnaugh,
  - Phương pháp Quine Mc. Cluskey.
- Phương pháp đại số và bảng Karnaugh: rút gọn mạch logic tổ hợp với số lượng biến không lớn (thường  $< 6$ ), thực hiện bằng tay là chủ yếu.
- Phương pháp Quine Mc. Cluskey: rút gọn được các hàm (mạch) nhiều biến và có thể tiến hành công việc nhờ máy tính.

- Dựa vào các định lý đã học để đưa biểu thức về dạng tối giản.
- Một số cách rút gọn bằng phương pháp đại số
  1. Loại bỏ tổ hợp thừa
  2. Áp dụng định lý De Morgan
  3. Triển khai từ thành phần nhiều biến
  4. Triển khai từ thành phần ít biến, đặt nhân tử chung

## 1. Loại bỏ tổ hợp thừa

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:

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

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

## 2. Áp dụng định lí De Morgan

Ví dụ: Rút gọn biểu thức:  $f = \overline{CD} + \overline{C\bar{D}} \cdot \overline{AC} + \bar{D}$

Áp dụng định lí De Morgan, ta có:

$$\begin{aligned} f &= \overline{CD} + \overline{C\bar{D}} \cdot \overline{AC} + \bar{D} \\ &= \overline{CD + C\bar{D} + AC + D} \\ &= \overline{\bar{D} + CD + AC} \\ &= \overline{(D + \bar{D})(C + \bar{D}) + AC} \\ &= \overline{C + \bar{D} + AC} \\ &= \overline{C + \bar{D}} \\ &= \bar{C}D \end{aligned}$$

## 3. Triển khai từ thành phần nhiều biến

*Trong biểu thức dạng tổng các tích, số hạng nào có chứa nhiều biến nhất (nhưng không chứa đầy đủ các biến), thì ta áp dụng định lý bù bổ sung các biến còn thiếu để số hạng đó trở thành chứa đầy đủ thành phần các biến, đặt thừa số chung (nếu có) với các số hạng khác để triệt tiêu và tiếp tục áp dụng các định lý khác để rút gọn.*

Ví dụ: Hãy đưa hàm logic về dạng tối giản:  $f = AB + BCD + \bar{A}C + \bar{B}C$

Lời giải: Ví dụ:  $f_1 = A\bar{D} + \bar{B}D + B\bar{C}D + \bar{A}CD + ABC$

Nhận thấy trong biểu thức tổng các tích có số hạng BCD có chứa nhiều biến nhất nhưng không chứa biến A, Áp dụng định lý,  $A + \bar{A} = 1$ , nên ta lấy BCD  $(A + \bar{A})$

Áp dụng định lý,  $A + A.B = A$  ta có:

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

## 4. Triển khai từ thành phần ít biến, đặt nhân tử chung

*Khi trong biểu thức, hai hay một vài số hạng có chứa một biến thành phần nào đó giống nhau, mà sau khi đặt thành phần biến giống nhau đó làm thừa số chung thì trong ngoặc sẽ xuất hiện một tổ hợp có chứa các thành phần mà có chứa biến giống với số hạng khác trong biểu thức, thì ta sẽ làm theo phương pháp đặt nhân tử chung đó rồi áp dụng các định lý vào rút gọn*

Ví dụ: Rút gọn biểu thức sau:  $f = AB + BCD + \bar{A}C + \bar{B}C$

Nhận thấy trong biểu thức có  $\bar{A}C$ ,  $\bar{B}C$  có C chung, nếu sau khi đặt C ra làm thừa số chung thì trong ngoặc có  $(\bar{A} + \bar{B}) = \overline{AB}$ , có chứa thành phần biến giống AB. Do đó đặt C làm thừa số chung, áp dụng thêm định lý  $A + AB = A$ ,  $A + \bar{A}B = A + B$ , ta có

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

Ví dụ:  $f_1 = AB + BD + CDE + DA$



# 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.

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

$$f_1 = AB$$

$$f_2 = C$$

➤ Ví dụ: Dùng bảng Các nô để giản ước hàm:

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

Kết quả:  $f = AB + C$

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

- Chuyển các hàm sang dạng chuẩn của minterm và maxterm:

a)  $A.(B + \overline{A.C}).(\overline{A + B.C})$       b)  $\overline{A.B.A.C.A.D} + \overline{B.C.B.C} + \overline{C.D}$

- Rút gọn hàm sau theo phương pháp bảng Karnaugh:

a)  $F(A, B, C, D) = \Sigma(3, 7, 8, 9, 10, 12).$

b)  $F(A, B, C, D) = \Sigma(0, 1, 4, 9, 12, 13) + \Sigma d(2, 3, 6, 10, 11, 14).$

- Rút gọn hàm sau theo phương pháp đại số:

a)  $\overline{CD} + \overline{CD} . \overline{AC} + \overline{D}$       b)  $\overline{A}\overline{B}\overline{C} . \overline{A}\overline{B} + \overline{B}\overline{C} + \overline{C}\overline{A}$

- Tối thiểu hóa bằng phương pháp Quine Mc. Cluskey:

a)  $F(A, B, C, D) = \Sigma(0, 2, 5, 6, 7, 10, 13, 14, 15)$

Chương 1: Hệ đếm

Chương 2: Đại số Boole

➤ **Chương 3: Cổng logic**

Chương 4: Mạch logic tổ hợp

Chương 5: Mạch logic tuần tự

# Cổng logic

# Cổng logic cơ bản: AND, OR, NOT

- Cổng AND
- Cổng OR
- Cổng NOT

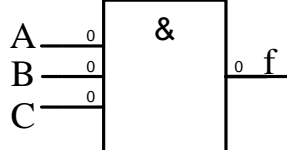
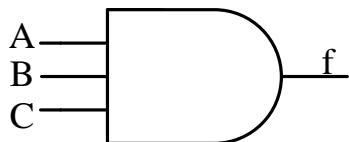
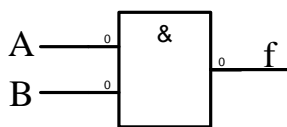
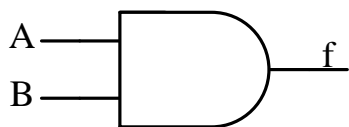
# Cổng AND

➤ 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**

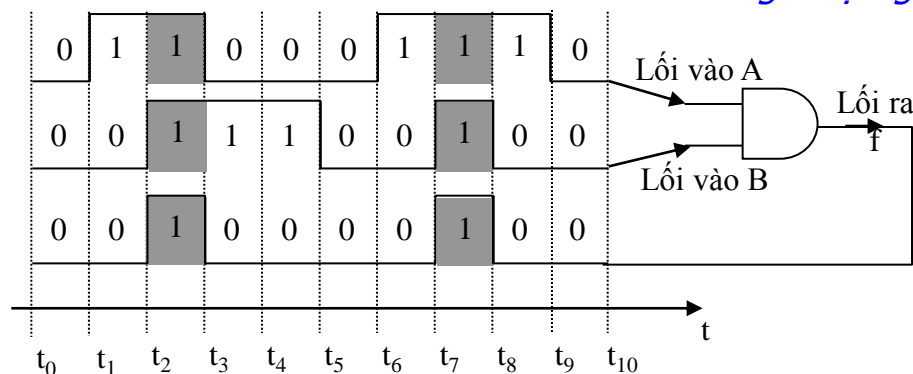
**Bảng trạng thái cổng AND 2 lối vào**

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**

**GIẢNG VIÊN: ThS. LƯƠNG CÔNG DUẤN**

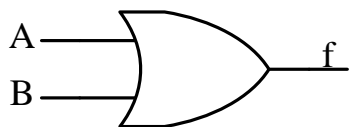
**BỘ MÔN: KTĐT - KHOA KTĐT1**

# Cổng OR

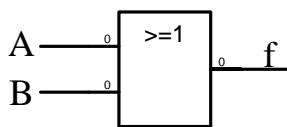
➤ Hàm ra của cổng OR 2 và nhiều biến vào như sau:

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

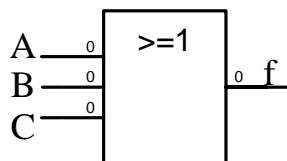
**Ký hiệu cổng OR**



Chuẩn ANSI



Chuẩn IEEE



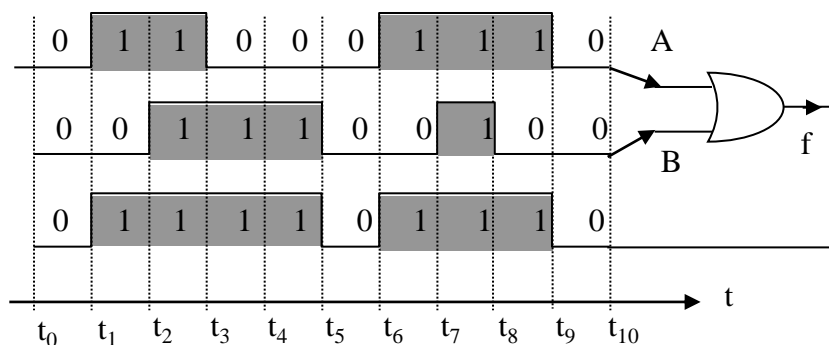
**Bảng trạng thái cổng OR 2 lối vào**

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.**

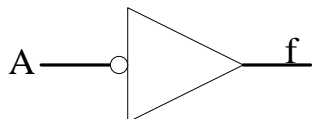
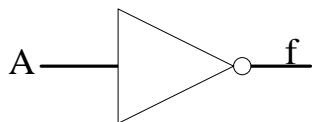
**GIẢNG VIÊN: ThS. LƯƠNG CÔNG DUÂN**

**BỘ MÔN: KTĐT - KHOA KTĐT1**

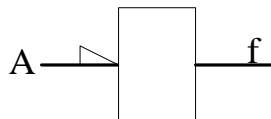
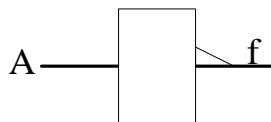
➤ Hàm ra của cổng NOT:

$$f = \bar{A}$$

Ký hiệu cổng NOT



Chuẩn ANSI



Chuẩn IEEE

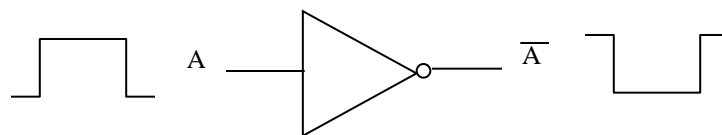
Bảng trạng thái cổng NOT

A	f
0	1
1	0

*Theo giá trị logic*

A	f
L	H
H	L

*Theo mức logic*



Dạng xung ra

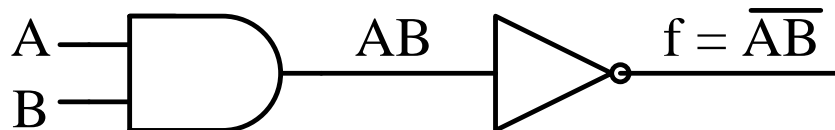


# Một số cổng ghép thông dụng

- 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.

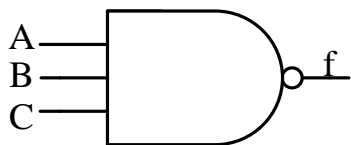
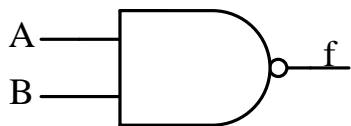


- Hàm ra của cổng NAND 2 và nhiều biến vào như sau:

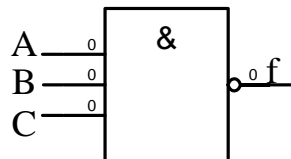
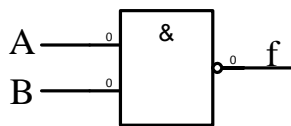
$$f = \overline{AB}$$

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

## 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

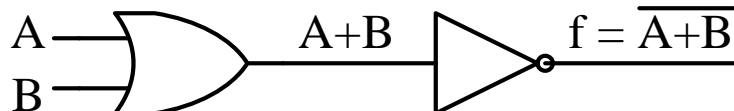
Theo giá trị logic

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.

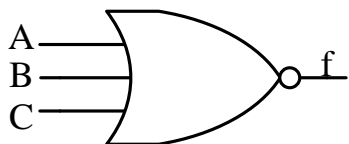
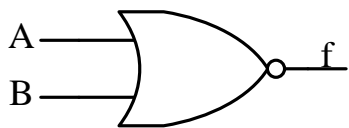


- Hàm ra của cổng NOR 2 và nhiều biến vào như sau:

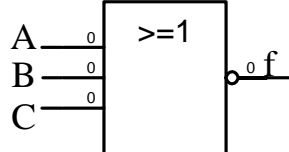
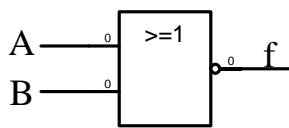
$$f = \overline{A + B}$$

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

## 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

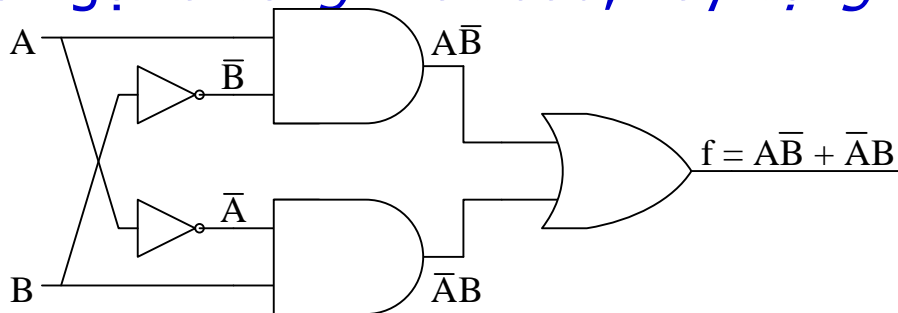
Theo giá trị logic

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

Theo mức logic

# Cổng XOR - cổng khác dấu

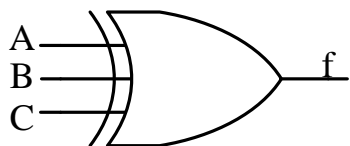
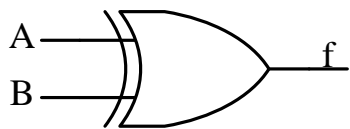
- Cổng XOR còn gọi là *cổng khác dấu*, hay *cộng modul 2*.



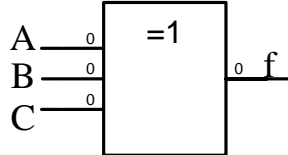
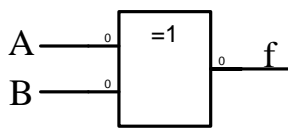
- Hàm ra của cổng XOR 2 biến vào như sau:

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

## 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

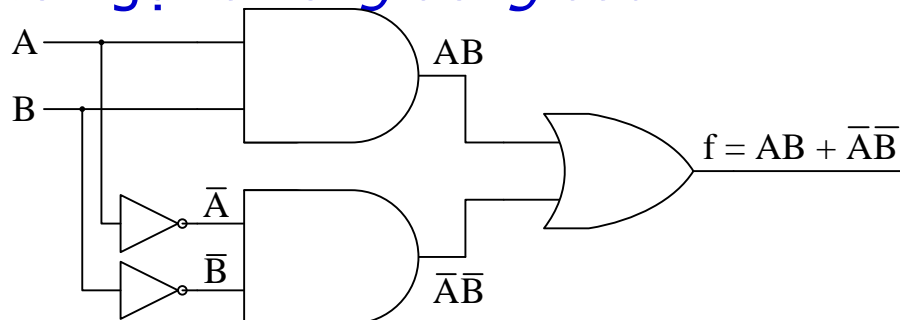
Theo giá trị logic

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

Theo mức logic

# Cổng XNOR - cổng đồng dấu

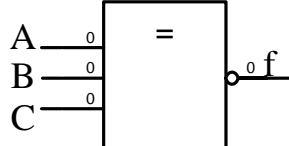
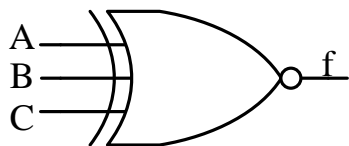
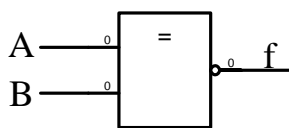
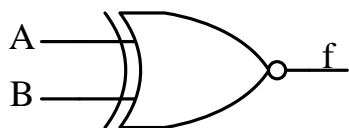
➤ Cổng XNOR còn gọi là *cổng đồng dấu*.



➤ 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**



**Bảng trạng thái cổng XNOR 2 lỗi vào**

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

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

*Theo giá trị logic*

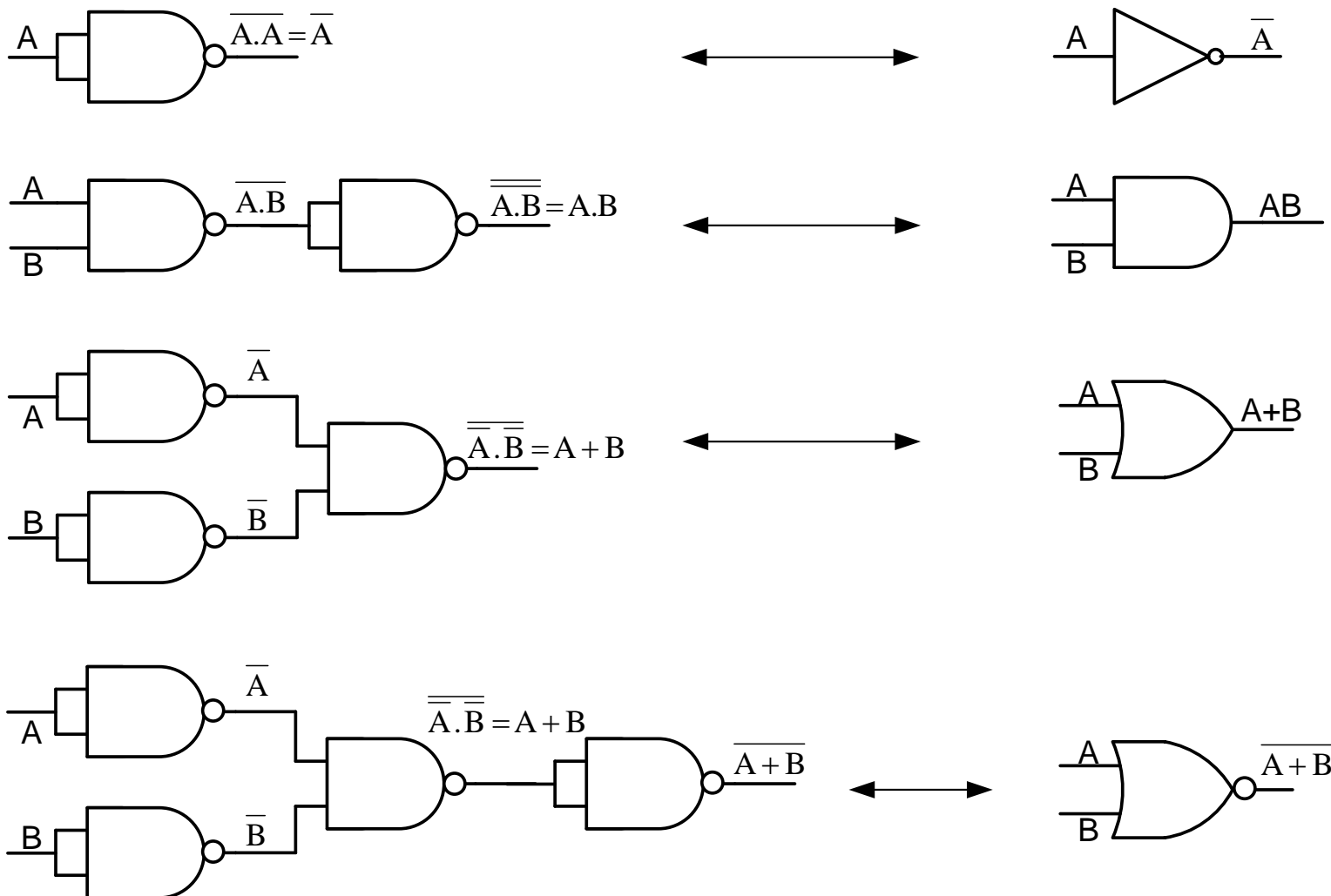
*Theo mức logic*

Chuẩn ANSI

Chuẩn IEEE

# Tính đa năng của cổng NAND

➤ Từ cổng NAND có thể tạo ra các cổng NOT, AND, OR và NOR.



# Tính đa năng của cổng NAND (2)

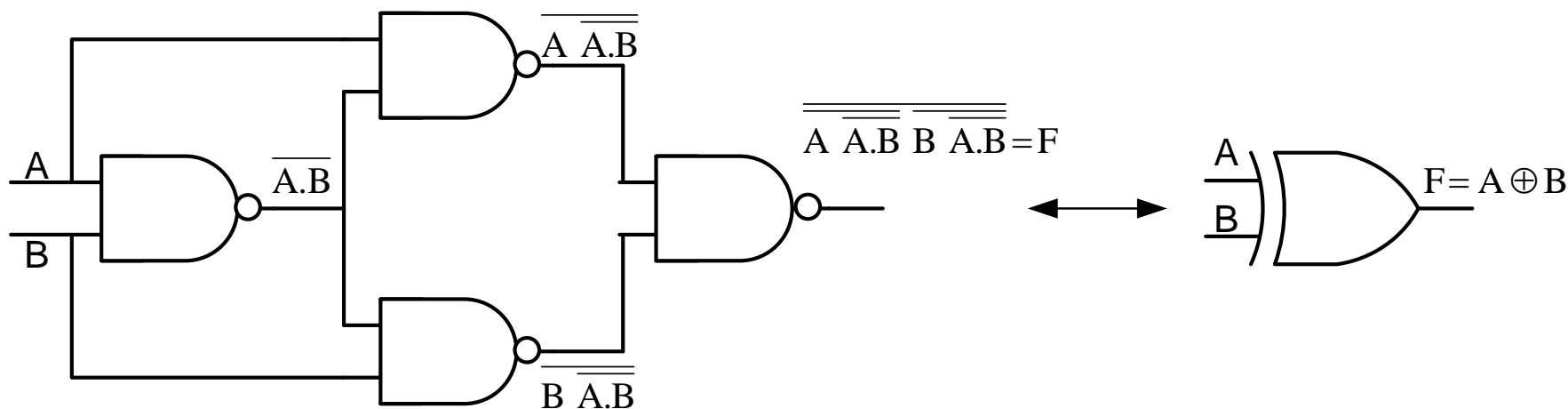
➤ Cổng XOR dùng toàn cổng NAND.

Hàm cổng XOR:  $F = A\bar{B} + \bar{A}B$

Biến đổi:

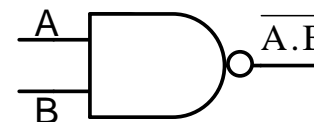
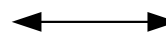
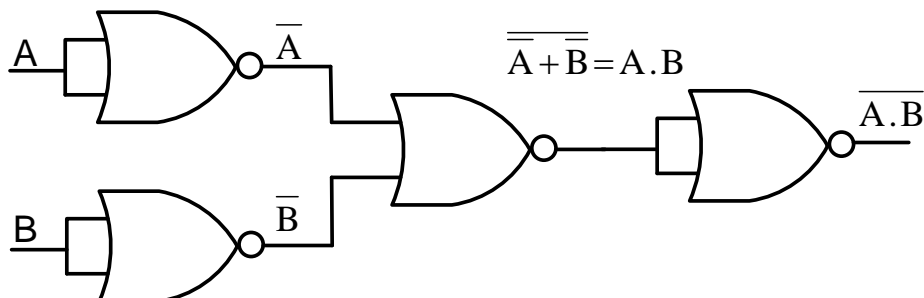
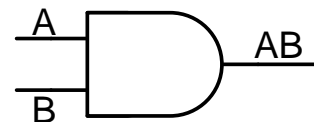
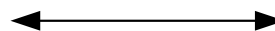
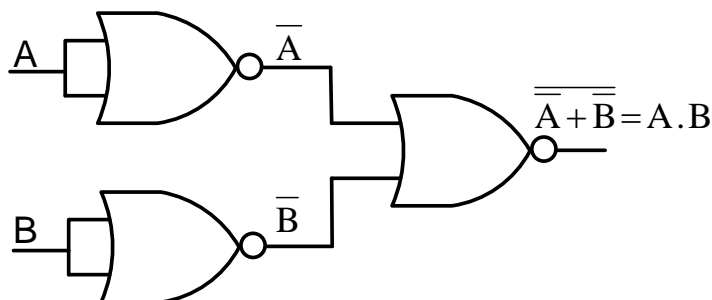
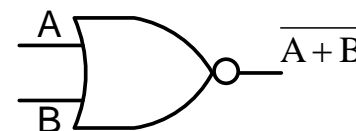
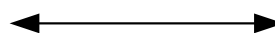
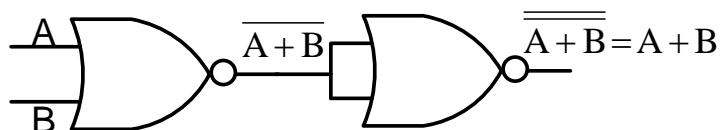
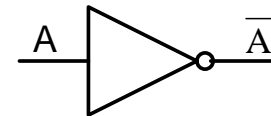
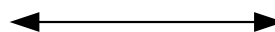
$$\begin{aligned} F &= A\bar{B} + \bar{A}B = A\bar{B} + A\bar{A} + \bar{A}B + B\bar{B} = A(\bar{A} + \bar{B}) + B(\bar{A} + \bar{B}) = \\ &= A\bar{A}\bar{B} + B\bar{A}\bar{B} = \overline{\overline{A\bar{A}\bar{B} + B\bar{A}\bar{B}}} = \overline{\overline{A\bar{A}\bar{B}} + \overline{B\bar{A}\bar{B}}} = \overline{\overline{A\bar{A}\bar{B}} \cdot \overline{B\bar{A}\bar{B}}} \end{aligned}$$

Mạch kết quả:



# Tính đa năng của cổng NOR

- Từ cổng NOR có thể tạo ra các cổng NOT, AND, OR và NAND.





## Tính đa năng của cổng NOR (2)

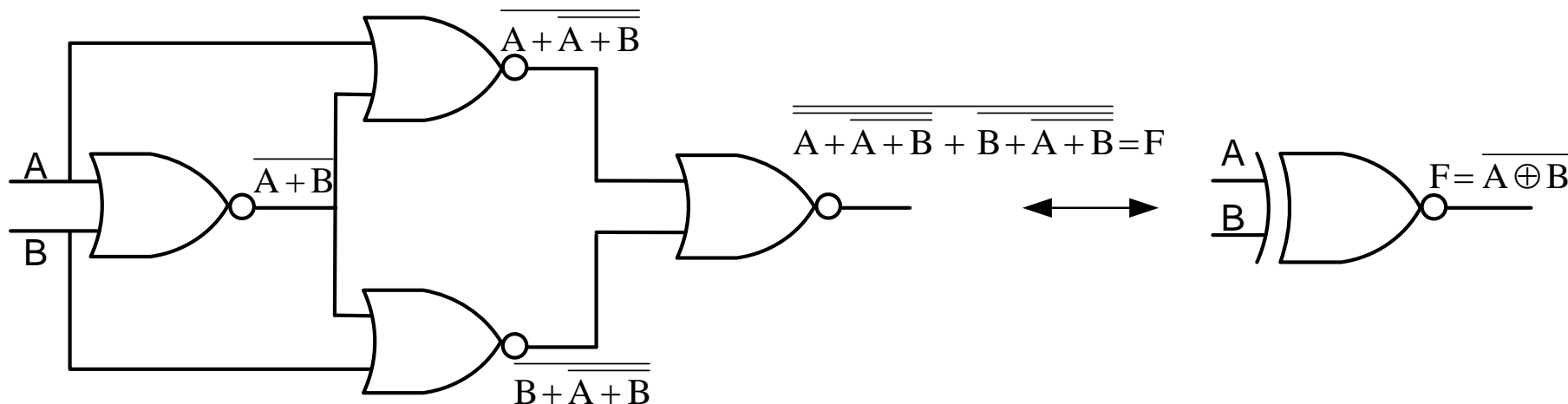
➤ Cổng XNOR dùng toàn cổng NOR.

Hàm cổng XNOR:  $F = A B + \overline{A} \overline{B}$

## Biến đổi:

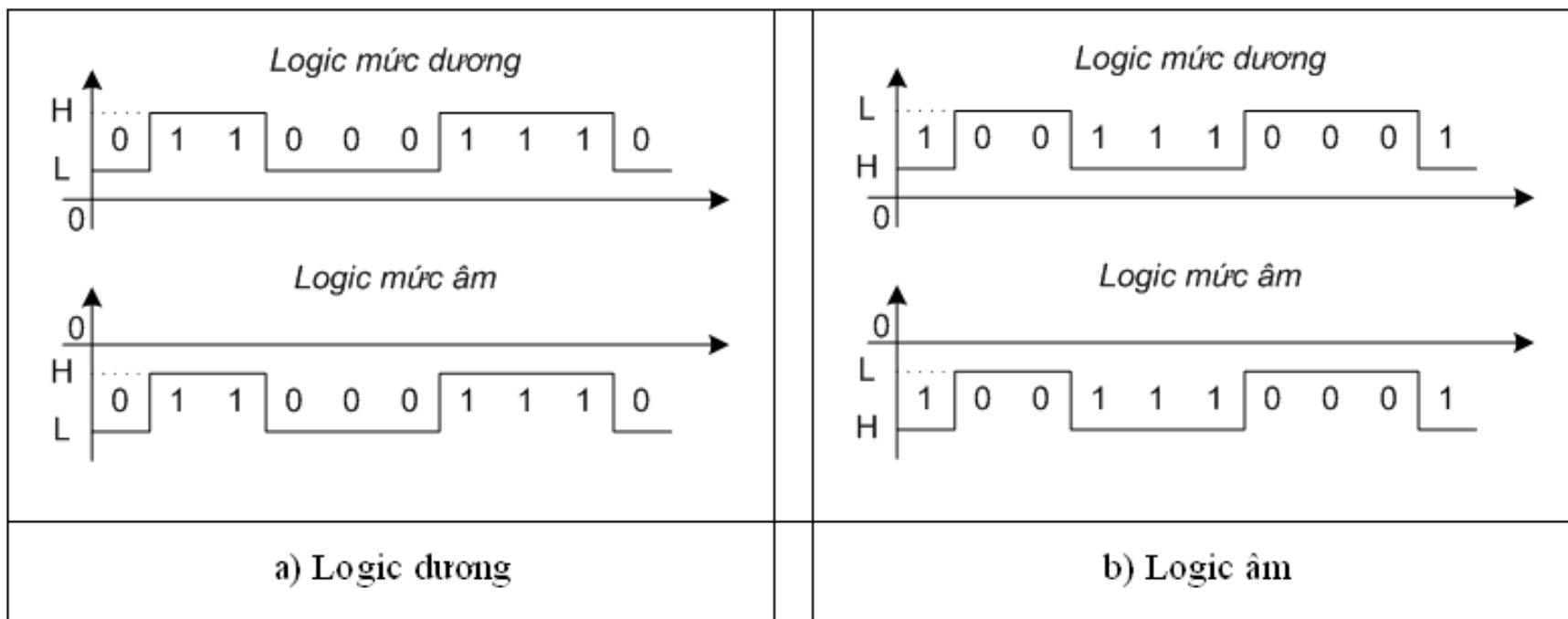
[illegible]

## Mạch kết quả:



# 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.



## Chương 3 – MẠCH LOGIC TỔ HỢP

### 3.1. Khái niệm chung

3.2. Phân tích mạch logic tổ hợp

3.3. Thiết kế mạch logic tổ hợp

3.4. Mạch mã hóa – giải mã

3.5. Mạch hợp kênh – phân kênh

3.6. Mạch số học

3.7. Mã phát hiện sai

## ***Mạch tổ hợp (1)***

### ➤ *Đặc điểm:*

- Trị số của tín hiệu đầu ra ở thời điểm đang xét chỉ phụ thuộc vào tổ hợp các giá trị tín hiệu đầu vào.
- Được cấu trúc nên từ các cổng logic.

### ➤ *Phương pháp biểu diễn chức năng logic của mạch tổ hợp:*

- Hàm logic (thường áp dụng cho vi mạch cỡ nhỏ - SSI)
- Bảng trạng thái (vi mạch cỡ vừa – MSI)
- Bảng Karnaugh
- Đồ thị dạng xung ...

## Mạch tổ hợp (2) – Sơ đồ khối tổng quát

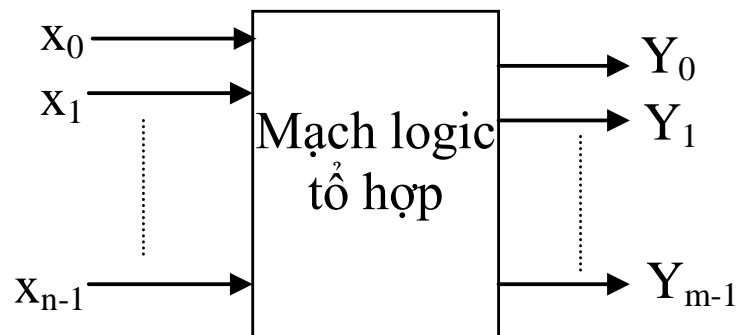
Hệ phương trình tổng quát:

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

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

...

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



## Chương 3 – MẠCH LOGIC TỔ HỢP

3.1. Khái niệm chung

**3.2. Phân tích mạch logic tổ hợp**

3.3. Thiết kế mạch logic tổ hợp

3.4. Mạch mã hóa – giải mã

3.5. Mạch hợp kênh – phân kênh

3.6. Mạch số học

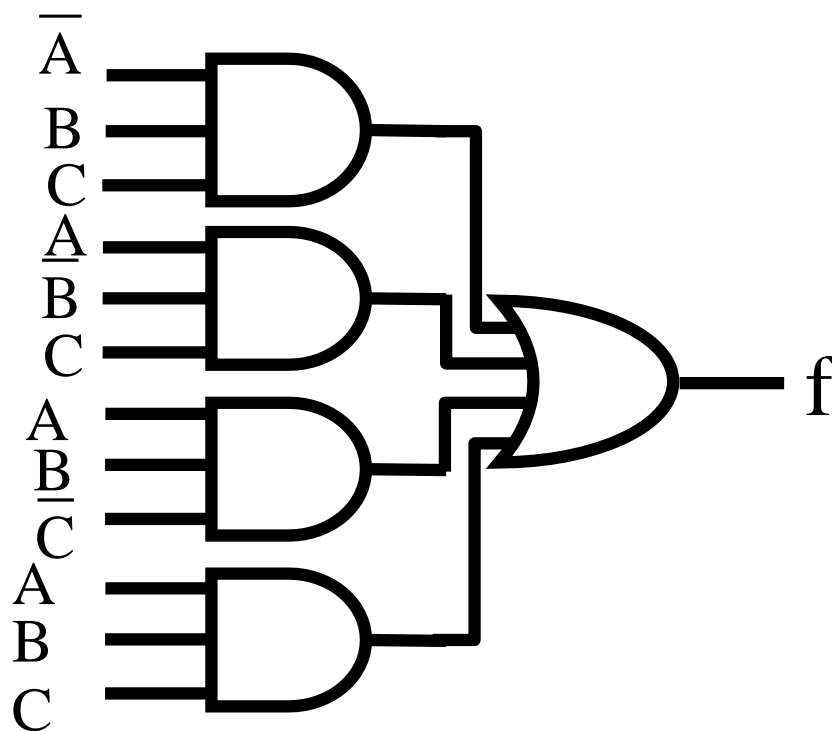
3.7. Mã phát hiện sai

## ***Phân tích mạch logic tổ hợp (1)***

- Phân tích mạch logic tổ hợp là từ sơ đồ cho trước xác định chức năng, dạng sóng, tính năng kỹ thuật,... của mạch. Từ đó có thể rút gọn, chuyển đổi dạng thực hiện của mạch điện để có được lời giải tối ưu.
- Phương pháp:
  - Viết biểu thức
  - Rút gọn, tối ưu (nếu cần)
  - Vẽ lại mạch điện.

## Phân tích mạch logic tổ hợp (2) – Ví dụ

➤ Phân tích mạch điện sau, nêu chức năng của mạch:

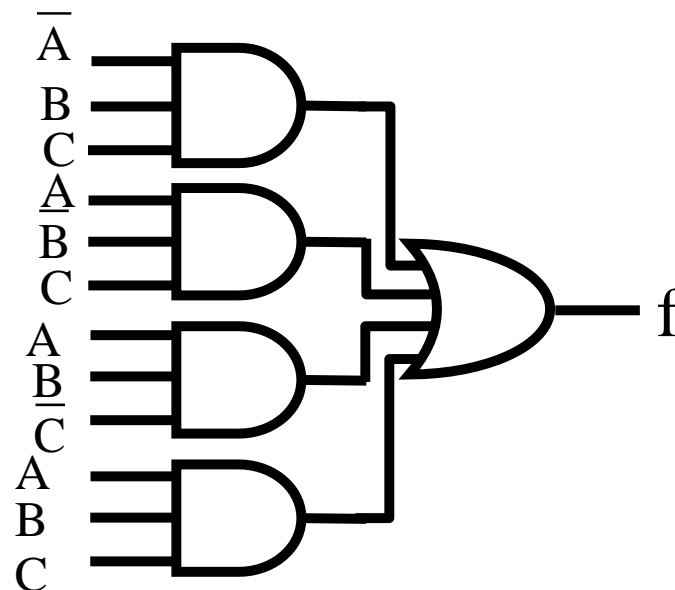




## Phân tích mạch logic tổ hợp (3) - VD

**Bước 1: Lập Bảng trạng thái:**

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



**- Viết biểu thức:**

$$f(A, B, C) = \sum(3, 5, 6, 7)$$

**Phân tích mạch logic tổ hợp (4) - VD**

Bước 2: Rút gọn:  $f(A, B, C) = \sum(3, 5, 6, 7)$

BC A	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$f_1 = AC$  (blue box around 01, 11 in row 1)  
 $f_2 = AB$  (red oval around 11, 10 in row 1)  
 $f_3 = BC$  (dashed red box around 11, 10 in row 1)

$$f = AB + BC + AC$$

- Tối ưu về dạng toàn NAND:

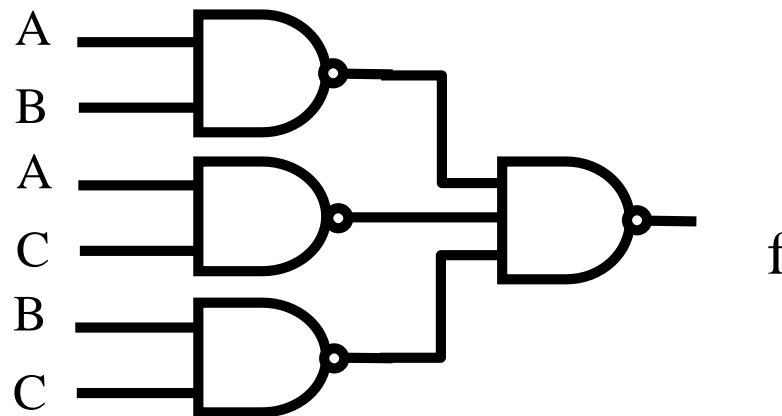
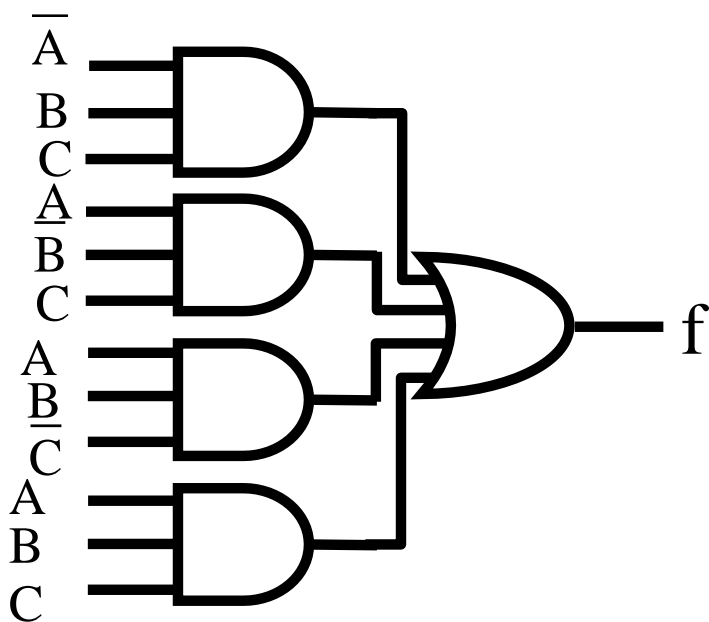
$$f = AB + BC + AC$$

$$\overline{\overline{f}} = \overline{\overline{AB + BC + AC}}$$

$$f = \overline{\overline{AB} \cdot \overline{BC} \cdot \overline{AC}}$$

## Phân tích mạch logic tổ hợp (5) - VD

Bước 3: Vẽ mạch điện dạng tối ưu:



## Chương 3 – MẠCH LOGIC TỔ HỢP

3.1. Khái niệm chung

3.2. Phân tích mạch logic tổ hợp

**3.3. Thiết kế mạch logic tổ hợp**

3.4. Mạch mã hóa – giải mã

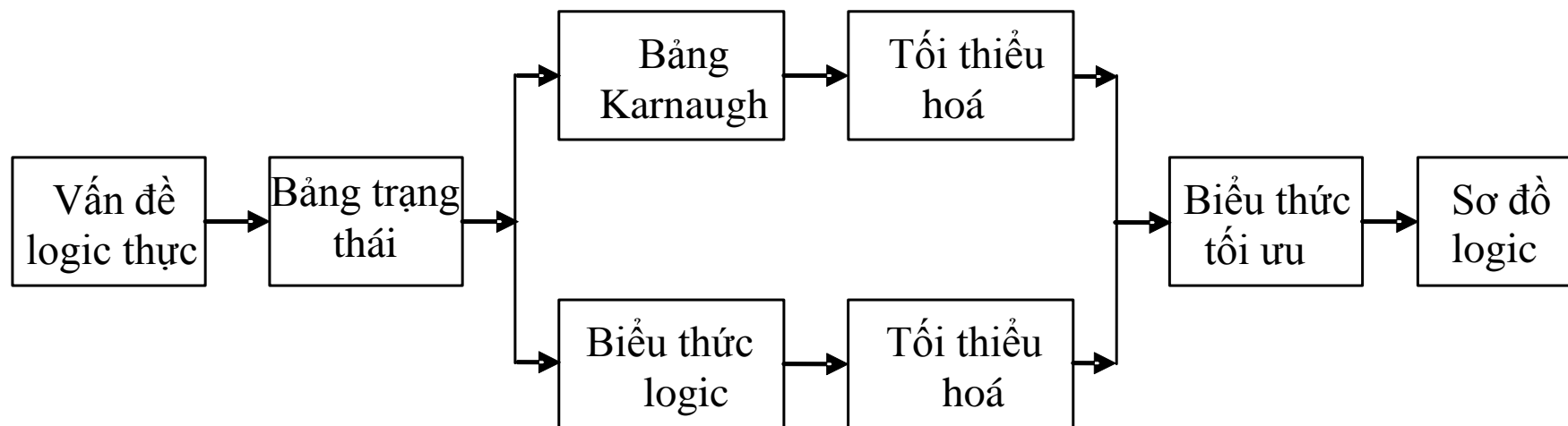
3.5. Mạch hợp kênh – phân kênh

3.6. Mạch số học

3.7. Mã phát hiện sai

## Thiết kế mạch logic tổ hợp

- Thiết kế mạch logic tổ hợp là bài toán từ yêu cầu (chức năng, dạng sóng, tính năng kỹ thuật, ...) xây dựng sơ đồ mạch thực hiện (ngược với bài toán phân tích).
- Phương pháp:



## Chương 3 – MẠCH LOGIC TỔ HỢP

- 3.1. Khái niệm chung
- 3.2. Phân tích mạch logic tổ hợp
- 3.3. Thiết kế mạch logic tổ hợp
- 3.4. Mạch mã hóa – giải mã**
- 3.5. Mạch hợp kênh – phân kênh
- 3.6. Mạch số học
- 3.7. Mã phát hiện sai

### 3.4.1. Một số loại mã thông dụng: Mã nhị - thập phân BCD

- Mã hóa 10 ký hiệu thập phân bằng 4 bit nhị phân (1 đecat).
- Từ số 10 trở lên: mỗi ký hiệu số được biểu thị ít nhất bằng 2 đecat nhị phân.

\* Một số mã BCD thông dụng:

- Mã BCD tự nhiên (N-BCD: Nature BCD hay BCD 8421):

Các chữ số thập phân được nhị phân hoá theo trọng số:  $2^3$ ,  $2^2$ ,  $2^1$ ,  $2^0$

Có 6 tổ hợp dư, ứng với các số thập phân 10, 11, 12, 13, 14 và 15.

- Mã BCD 7421
- Mã BCD 5121
- Mã BCD 2421 (mã Aiken)
- Mã BCD 4221

### 3.4.1. Một số loại mã thông dụng: Mã nhị - thập phân BCD

Ưu điểm:

- Mã BCD có trọng số không thay đổi
- Được sắp xếp theo qui luật
- Dễ nhớ

Số thập phân	Trọng số của mã BCD				
	8421	7421	5121	2421	4221
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0000
1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0001
2	0 0 1 0	0 0 1 0	0 0 1 0	0 0 1 0	0010
3	0 0 1 1	0 0 1 1	0 0 1 1	0 0 1 1	0011
4	0 1 0 0	0 1 0 0	0 1 1 1	0 1 0 0	1000
5	0 1 0 1	0 1 0 1	1 0 0 0	1 0 1 1	0111
6	0 1 1 0	0 1 1 0	1 0 0 1	1 1 0 0	1100
7	0 1 1 1	1 0 0 0	1 0 1 0	1 1 0 1	1101
8	1 0 0 0	1 0 0 1	1 0 1 1	1 1 1 0	1110
9	1 0 0 1	1 0 1 0	1 1 1 1	1 1 1 1	1111



## 3.4.1. Một số loại mã thông dụng: Mã Gray (Mã cách 1)

\* Mã Gray (mã cách 1):

- Các tổ hợp mã kề nhau chỉ khác nhau duy nhất 1 bit.
- Không có tính trọng số.
- Chỉ có thể giải mã thông qua bảng mã.

\* Cách xây dựng mã Gray:

- Phương pháp xen kẽ: đảo bit liền kề bên phải của bit 1
- Phương pháp soi gương:

## 3.4.1. Một số loại mã thông dụng: Một số mã khác

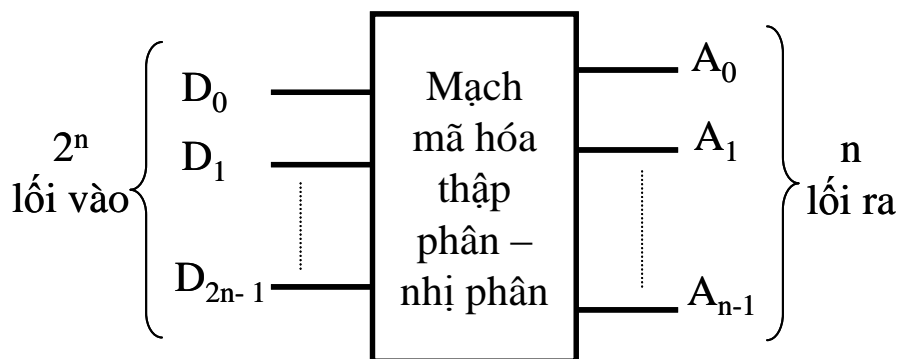
Số thập phân	Số nhị phân	Mã dư 3	Mã Gray	Mã Gray Dư 3	Mã Johnson	Mã vòng
0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 1 0	0 0 0 0 0	0 0 0 0 0 0 0 0 0 1
1	0 0 0 1	0 1 0 0	0 0 0 1	0 1 1 0	1 0 0 0 0	0 0 0 0 0 0 0 0 1 0
2	0 0 1 0	0 1 0 1	0 0 1 1	0 1 1 1	1 1 0 0 0	0 0 0 0 0 0 0 0 1 0 0
3	0 0 1 1	0 1 1 0	0 0 1 0	0 1 0 1	1 1 1 0 0	0 0 0 0 0 0 0 1 0 0 0
4	0 1 0 0	0 1 1 1	0 1 1 0	0 1 0 0	1 1 1 1 0	0 0 0 0 0 1 0 0 0 0 0
5	0 1 0 1	1 0 0 0	0 1 1 1	1 1 0 0	1 1 1 1 1	0 0 0 0 1 0 0 0 0 0 0
6	0 1 1 0	1 0 0 1	0 1 0 1	1 1 0 1	0 1 1 1 1	0 0 0 1 0 0 0 0 0 0 0
7	0 1 1 1	1 0 1 0	0 1 0 0	1 1 1 1	0 0 1 1 1	0 0 1 0 0 0 0 0 0 0 0
8	1 0 0 0	1 0 1 1	1 1 0 0	1 1 1 0	0 0 0 1 1	0 1 0 0 0 0 0 0 0 0 0
9	1 0 0 1	1 1 0 0	1 1 0 1	1 0 1 0	0 0 0 0 1	1 0 0 0 0 0 0 0 0 0 0

## 3.4.2. Mạch mã hóa – Khái niệm

- Mã hóa: là dùng văn tự, ký hiệu (hay mã) để biểu thị một đối tượng nào đó (tin tức).
- Mục đích: để tin tức truyền đi tốt hơn, cải thiện một số đặc tính như: độ tin cậy, tốc độ truyền tin, dung lượng kênh, tính bảo mật, ...
- Bộ mã hóa: thực hiện nhiệm vụ mã hóa. Một số bộ mã hóa:
  - Bộ mã hóa nhị phân
  - Bộ mã hóa BCD
  - Bộ mã hóa ưu tiên
  - ...

## 3.4.2. Mạch mã hóa: Mạch mã hóa thập phân – nhị phân (1)

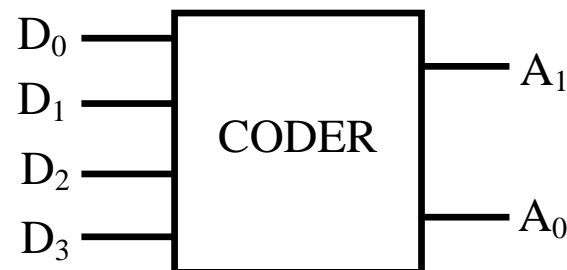
- Chức năng: mã hóa từ dạng thập phân đầu vào thành dạng nhị phân đầu ra.
- Bộ mã hóa N tín hiệu thập phân đầu vào có số bit đầu ra n thỏa mãn:  $N \leq 2^n$
- Sơ đồ khối tổng quát:



## 3.4.2. Mạch mã hóa: Mạch mã hóa thập phân – nhị phân (2)

➤ Bài toán: Thiết kế mạch mã hóa thập phân – nhị phân 2 bit.

➤ Sơ đồ khối:



➤ Bảng trạng thái:

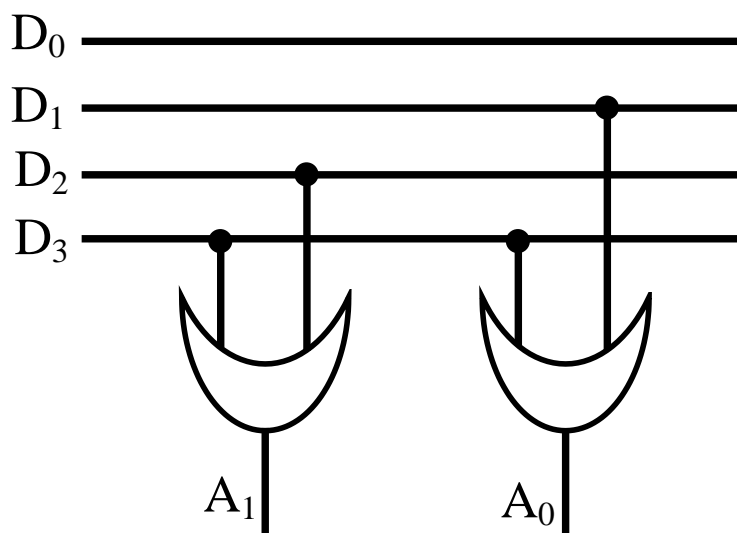
D0	D1	D2	D3	A1	A0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

## 3.4.2. Mạch mã hóa: Mạch mã hóa thập phân – nhị phân (3)

➤ Biểu thức hàm ra:

$$\begin{aligned} A_0 &= \overline{D_0} \cdot D_1 \cdot \overline{D_2} \cdot \overline{D_3} + \overline{D_0} \cdot \overline{D_1} \cdot \overline{D_2} \cdot D_3 \\ A_1 &= \overline{D_0} \cdot \overline{D_1} \cdot D_2 \cdot \overline{D_3} + \overline{D_0} \cdot \overline{D_1} \cdot \overline{D_2} \cdot D_3 \end{aligned} \Rightarrow \begin{aligned} A_0 &= D_1 + D_3 \\ A_1 &= D_2 + D_3 \end{aligned}$$

➤ Mạch điện:



## 3.4.2. Mạch mã hóa: Mạch mã hóa thập phân – NBCD (1)

➤ Chức năng: Mã hóa 10 ký hiệu thập phân bằng 4 bit nhị phân.

➤ Sơ đồ khối:



➤ Bảng trạng thái:

➤ Biểu thức hàm ra:

$$A = 8 + 9$$

$$B = 4 + 5 + 6 + 7$$

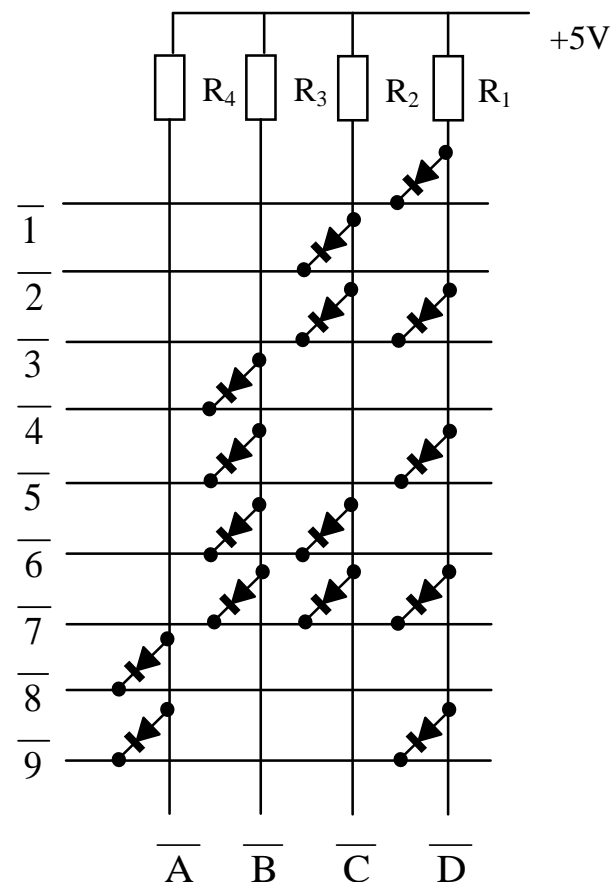
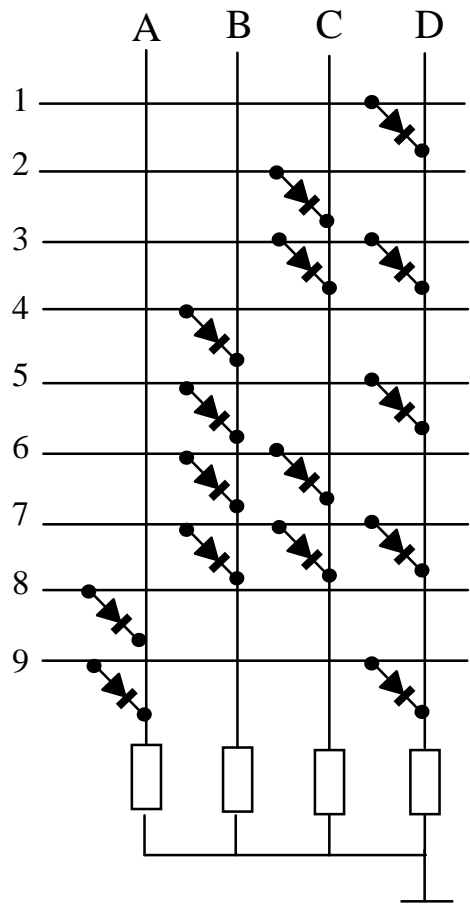
$$C = 2 + 3 + 6 + 7$$

$$D = 1 + 3 + 5 + 7 + 9$$

Vào thập phân	Ra BCD 8 4 2 1 A B C D
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

## 3.4.2. Mạch mã hóa: Mạch mã hóa thập phân – NBCD (2)

➤ Mạch điện (dạng ma trận OR): ➤ Mạch điện (dạng ma trận AND):

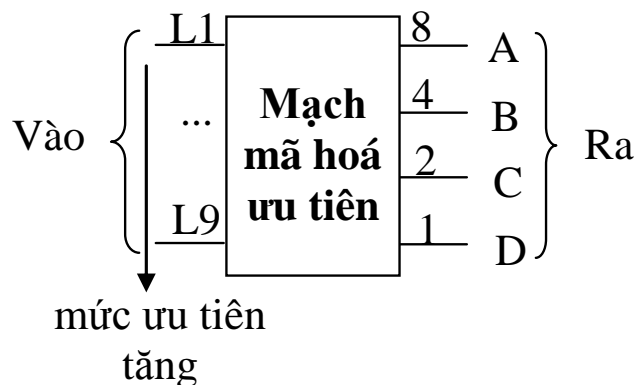




## 3.4.2. Mạch mã hóa: Mạch mã hóa ưu tiên thập phân – NBCD (1)

➤ Khi có nhiều đầu vào tác động đồng thời, chỉ mã hoá tín hiệu vào có mức ưu tiên cao nhất. Mức ưu tiên do người thiết kế mạch xác định.

➤ Sơ đồ khối:



➤ Bảng trạng thái:

Vào thập phân									Ra BCD			
1	2	3	4	5	6	7	8	9	A	B	C	D
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1
x	1	0	0	0	0	0	0	0	0	0	1	0
x	x	1	0						0	0	1	1
			1						0	1	0	0
				1					0	1	0	1
					1				0	1	1	0
						1			0	1	1	1
							1		1	0	0	0
								1	1	0	0	1

## 3.4.2. Mạch mã hóa: Mạch mã hóa ưu tiên thập phân – NBCD (2)

➤ *Biểu thức:*

$$D = \begin{cases} L1 = "1" \text{ và } L2,4,6,8 \text{ bằng } "0" \\ L3 = "1" \text{ và } L4,6,8 \text{ bằng } "0" \\ L5 = "1" \text{ và } L6,8 \text{ bằng } "0" \\ L7 = "1" \text{ và } L8 \text{ bằng } "0" \\ L9 = "1" \end{cases}$$

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

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

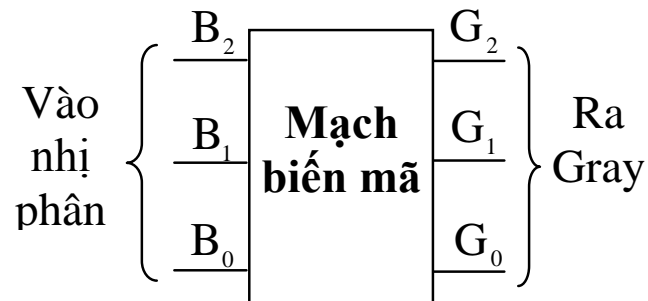
$$B = 4.\bar{8}.\bar{9} + 5.\bar{8}.\bar{9} + 6.\bar{8}.\bar{9} + 7.\bar{8}.\bar{9}$$

$$A = 8 + 9$$

### 3.4.3. Mạch biến mã (1)

➤ *Bài toán:* Thiết kế mạch biến mã từ nhị phân 3 bit sang Gray.

➤ *Sơ đồ khối:*



➤ *Bảng trạng thái:*

➤ *Biểu thức:*

$$G_0 = \sum(1, 2, 5, 6)$$

$$G_1 = \sum(2, 3, 4, 5)$$

$$G_2 = \sum(4, 5, 6, 7)$$

B2	B1	B0	G2	G1	G0
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	0
1	0	0	1	1	0
1	0	1	1	1	1
1	1	0	1	0	1
1	1	1	1	0	0

## 3.4.3. Mạch biến mã (2)

➤ Rút gọn:  $G_0 = \sum(1,2,5,6)$ ;  $G_1 = \sum(2,3,4,5)$ ;  $G_2 = \sum(4,5,6,7)$

**G0**

B1B0	00	01	11	10
B2				
0	0	1	0	1
1	0	1	0	1

$$G_0 = \overline{B_1} \cdot B_0 + B_1 \cdot \overline{B_0}$$

**G1**

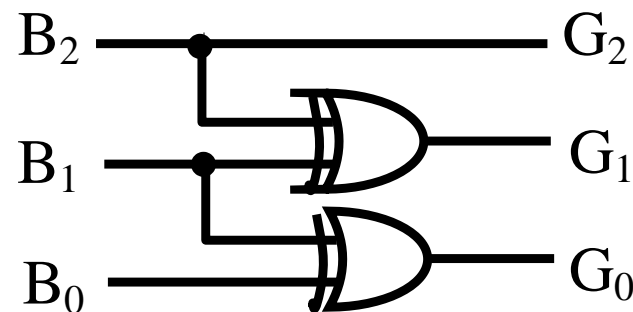
B1B0	00	01	11	10
B2				
0	0	0	1	1
1	1	1	0	0

$$G_1 = B_2 \cdot \overline{B_1} + \overline{B_2} \cdot B_1$$

**G2**

B1B0	00	01	11	10
B2				
0	0	0	0	0
1	1	1	1	1

$$G_2 = B_2$$



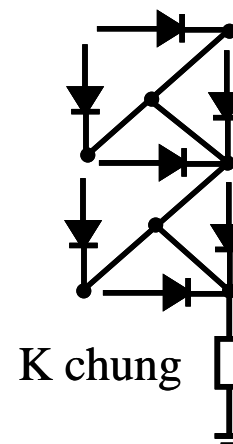
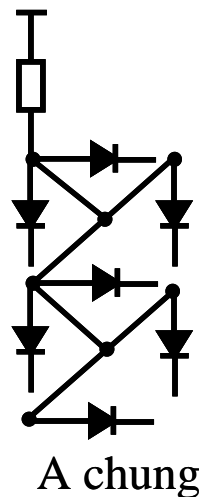
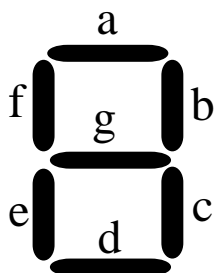
## 3.4.4. Mạch giải mã – Khái niệm

- Giải mã: là quá trình ngược lại của mã hóa, chuyển từ dạng mã sang tin tức.
- Bộ giải mã: thực hiện nhiệm vụ giải mã.
- Một số bộ giải mã:
  - Mạch giải mã BCD - 7 đoạn
  - Mạch giải mã nhị phân
  - ...

### 3.4.4. Mạch giải mã: Mạch giải mã BCD - 7 đoạn (1)

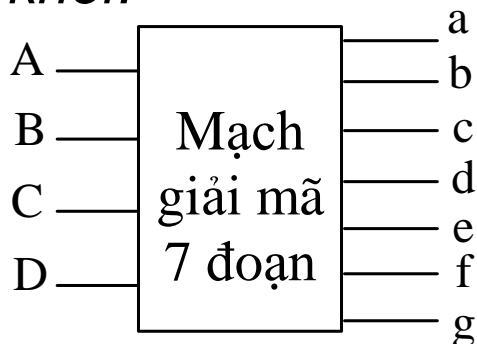
➤ *Dụng cụ 7 đoạn:*

- Dùng để hiển thị chữ số của một hệ đếm bất kỳ.
- Cấu tạo: gồm 7 đoạn làm bằng vật liệu có khả năng phát sáng (LED, LCD,...).
- Có hai loại LED 7 đoạn: Anode chung và Kathode chung.



## 3.4.4. Mạch giải mã: Mạch giải mã BCD - 7 đoạn (2)

➤ Sơ đồ khối:



➤ Bảng trạng thái: (A chung)

➤ Biểu thức:

$$a = \sum (1,4)$$

$$b = \sum (5,6)$$

$$c = \sum (2)$$

$$d = \sum (1,4,7)$$

$$e = \sum (1,3,4,5,7,9)$$

$$f = \sum (1,2,3,7)$$

$$g = \sum (0,1,7)$$

	A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	1	1	0	0	1	1	1	1
2	0	0	1	0	0	0	1	0	0	1	0
3	0	0	1	1	0	0	0	0	1	1	0
4	0	1	0	0	1	0	0	1	1	0	0
5	0	1	0	1	0	1	0	0	1	0	0
6	0	1	1	0	0	1	0	0	0	0	0
7	0	1	1	1	0	0	0	1	1	1	1
8	1	0	0	0	0	0	0	0	0	0	0
9	1	0	0	1	0	0	0	0	1	0	0

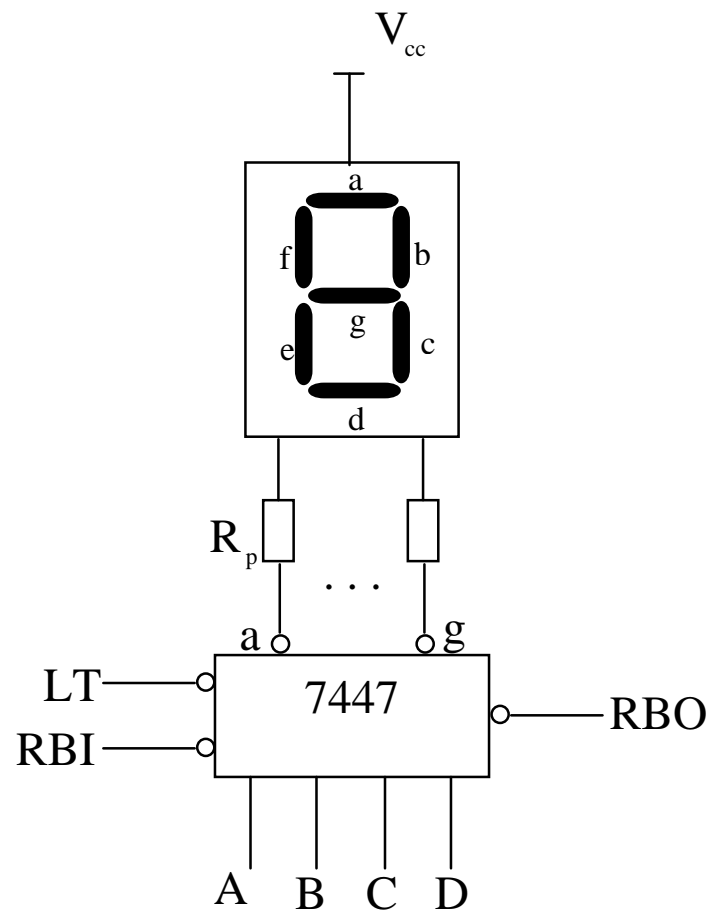
### 3.4.4. Mạch giải mã: Mạch giải mã BCD - 7 đoạn (3)

➤ IC giải mã 7 đoạn:

TTL: A chung: 7447, 74247 (đầu ra ở mức tích cực thấp)

K chung: 7448 (đầu ra ở mức tích cực cao)

CMOS: 4511





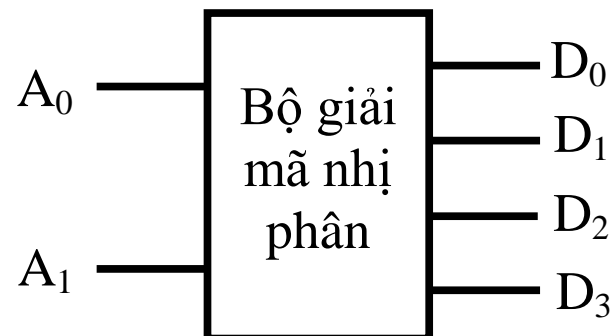
## 3.4.4. Mạch giải mã: Mạch giải mã nhị phân (1)

- Chức năng: giải mã từ dạng nhị phân đầu vào thành dạng thập phân đầu ra.
- Lựa chọn duy nhất một đầu ra ứng với một tổ hợp nhị phân đầu vào.
  - Mức tích cực cao: đầu ra được chọn bằng '1', các đầu ra còn lại bằng '0'.
  - Mức tích cực thấp: đầu ra được chọn bằng '0', các đầu ra còn lại bằng '1'.
- Bộ giải mã nhị phân  $n$  đầu vào có số đầu ra:  $N = 2^n$

### 3.4.4. Mạch giải mã: Mạch giải mã nhị phân (2)

➤ Bài toán 1: Thiết kế mạch giải mã nhị phân 2 vào – 4 ra.

➤ Sơ đồ khối:



➤ Bảng trạng thái:

➤ Biểu thức:

$$D_0 = \overline{A_1} \cdot \overline{A_0}$$

$$D_1 = \overline{A_1} \cdot A_0$$

$$D_2 = A_1 \cdot \overline{A_0}$$

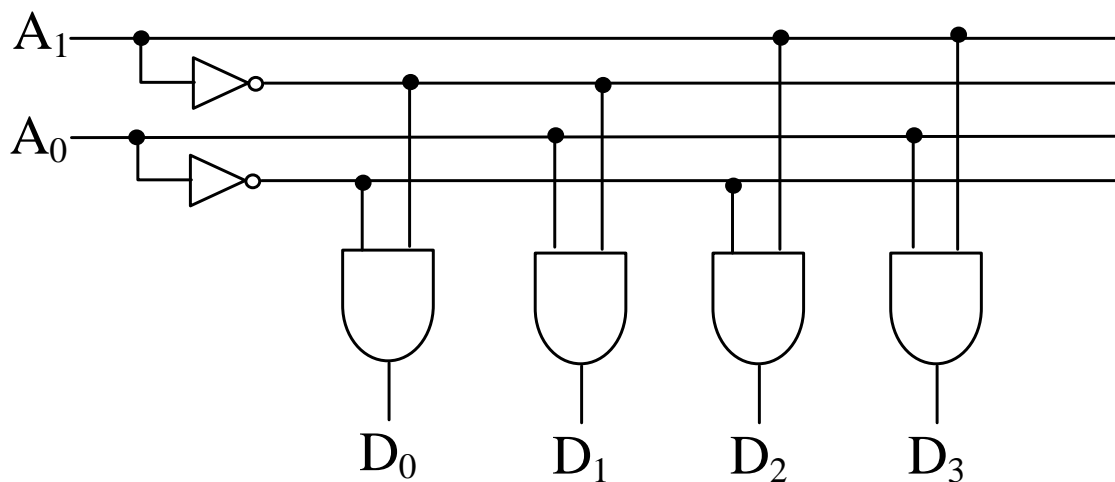
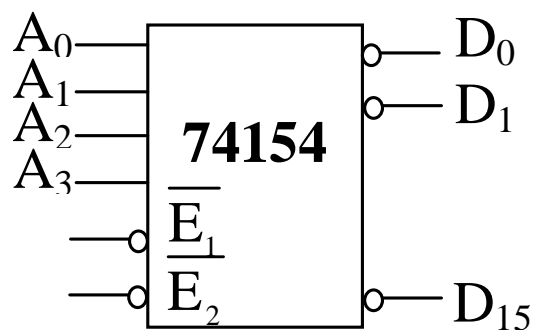
$$D_3 = A_1 \cdot A_0$$

A1	A0	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

### 3.4.4. Mạch giải mã: Mạch giải mã nhị phân (3)

➤ Mạch điện:

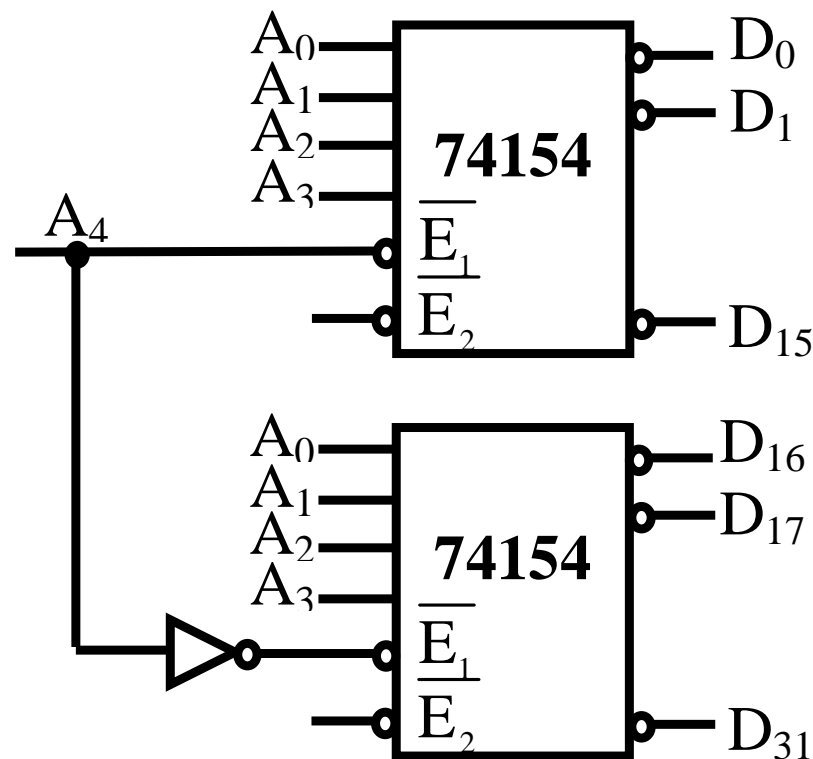
➤ IC thực tế: 74154



### 3.4.4. Mạch giải mã: Mạch giải mã nhị phân (4)

- Mở rộng dung lượng mạch giải mã nhị phân: Sử dụng IC 74154 thực hiện bộ giải mã nhị phân 5 lối vào địa chỉ.

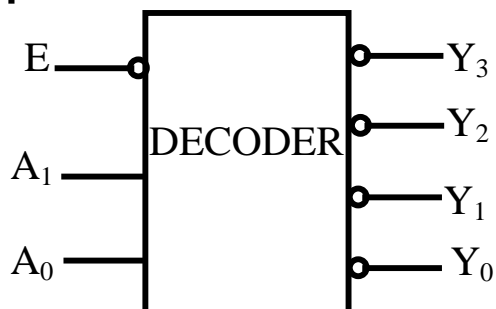
Sơ đồ khối:



### 3.4.4. Mạch giải mã: Mạch giải mã nhị phân (5)

➤ Bài toán 2: Cho mạch điện có BTT sau, nêu chức năng của mạch:

- Sơ đồ khối:



E	A1	A0	Y0	Y1	Y2	Y3
1	x	x	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

- Biểu thức hàm ra:

$$Y_0 = \overline{\overline{E}} \cdot \overline{\overline{A_1}} \cdot \overline{\overline{A_0}}$$

$$Y_1 = \overline{\overline{E}} \cdot \overline{\overline{A_1}} \cdot A_0$$

$$Y_2 = \overline{\overline{E}} \cdot A_1 \cdot \overline{\overline{A_0}}$$

$$Y_3 = \overline{\overline{E}} \cdot A_1 \cdot A_0$$

- Chức năng:

## Chương 3 – MẠCH LOGIC TỔ HỢP

3.1. Khái niệm chung

3.2. Phân tích mạch logic tổ hợp

3.3. Thiết kế mạch logic tổ hợp

3.4. Mạch mã hóa – giải mã

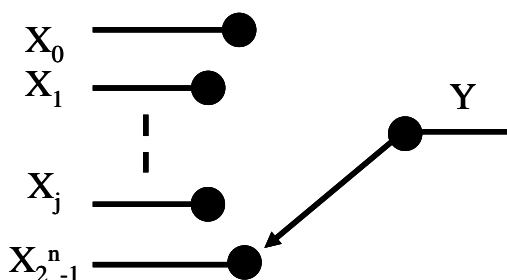
**3.5. Mạch hợp kênh – phân kênh**

3.6. Mạch số học

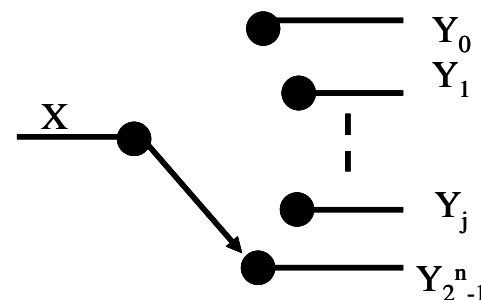
3.7. Mã phát hiện sai

## Mạch hợp kênh – phân kênh – Khái niệm

- Hợp kênh (ghép kênh): ghép nhiều kênh dữ liệu ở đầu vào thành một kênh duy nhất ở đầu ra.
- Mỗi kênh dữ liệu đầu vào có một địa chỉ xác định.
- Việc lựa chọn ghép kênh dữ liệu nào tới đầu ra phụ thuộc vào địa chỉ được chọn.
- Phân kênh (tách kênh): tách từ một kênh dữ liệu đầu vào thành nhiều kênh ở đầu ra (ngược của quá trình hợp kênh).



(a). Bộ hợp kênh (MUX)



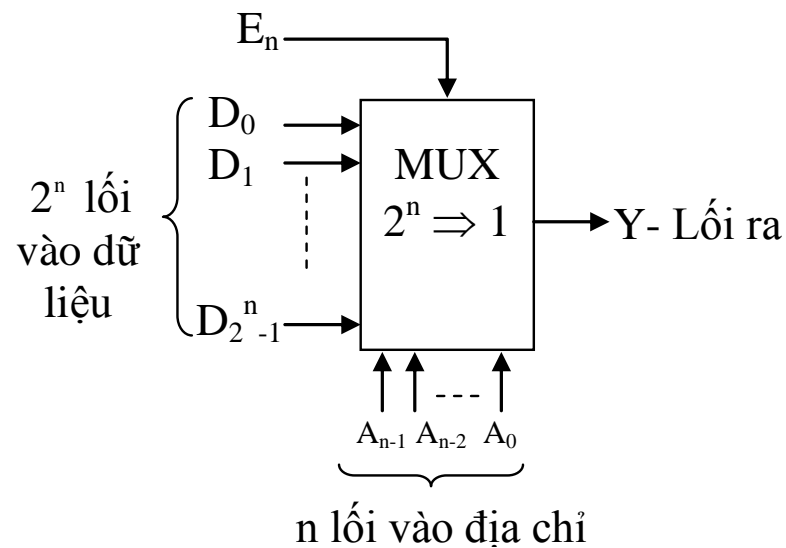
(b). Bộ phân kênh (DEMUX)

### 3.5.1. Mạch hợp kênh – MUX (1)

➤ *Chức năng:* lựa chọn nối một đầu vào dữ liệu tới đầu ra.

➤ *Sơ đồ khối:*

- Gồm:  $2^n$  lối vào dữ liệu,  $n$  lối vào địa chỉ, 1 lối vào chọn mạch  $E$  và 1 lối ra.
- Tùy theo giá trị của  $n$  lối vào địa chỉ mà lối ra sẽ bằng một trong những giá trị ở lối vào ( $D_j$ ).
- Nếu giá trị thập phân của  $n$  lối vào địa chỉ bằng  $j$  thì  $Y = D_j$ .

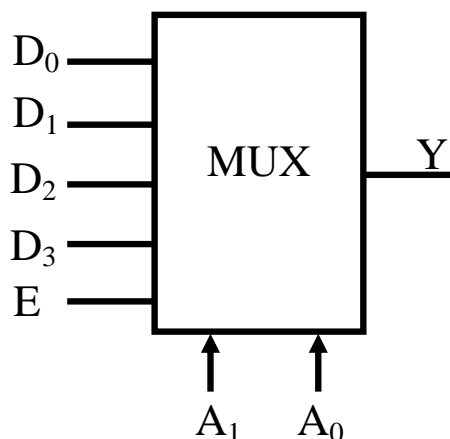




### 3.5.1. Mạch hợp kênh – MUX (2)

➤ *Bài toán:* Thiết kế MUX 2 lối vào địa chỉ, một lối vào điều khiển mức tích cực cao.

➤ *Sơ đồ khối:*



E	A1	A0	Y
0	x	x	0
1	0	0	D0
1	0	1	D1
1	1	0	D2
1	1	1	D3

➤ *Bảng trạng thái:*

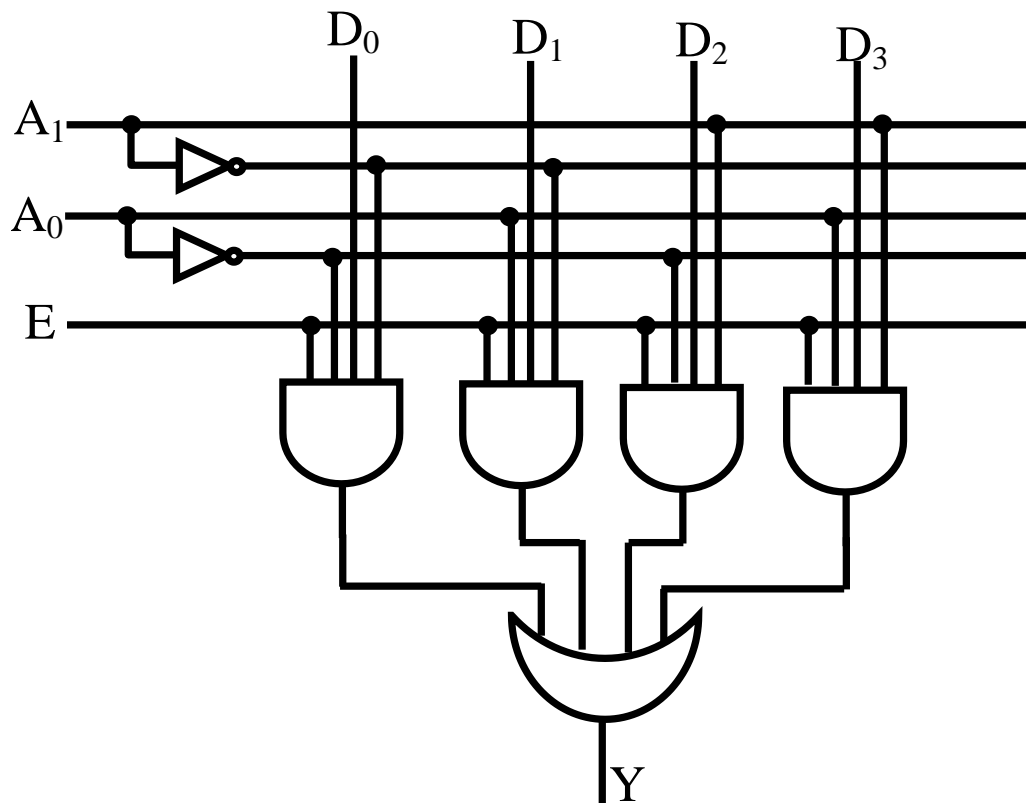
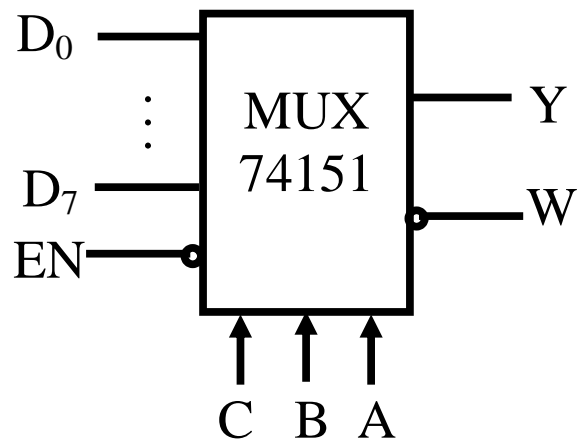
➤ *Biểu thức:*

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

### 3.5.1. Mạch hợp kênh – MUX (3)

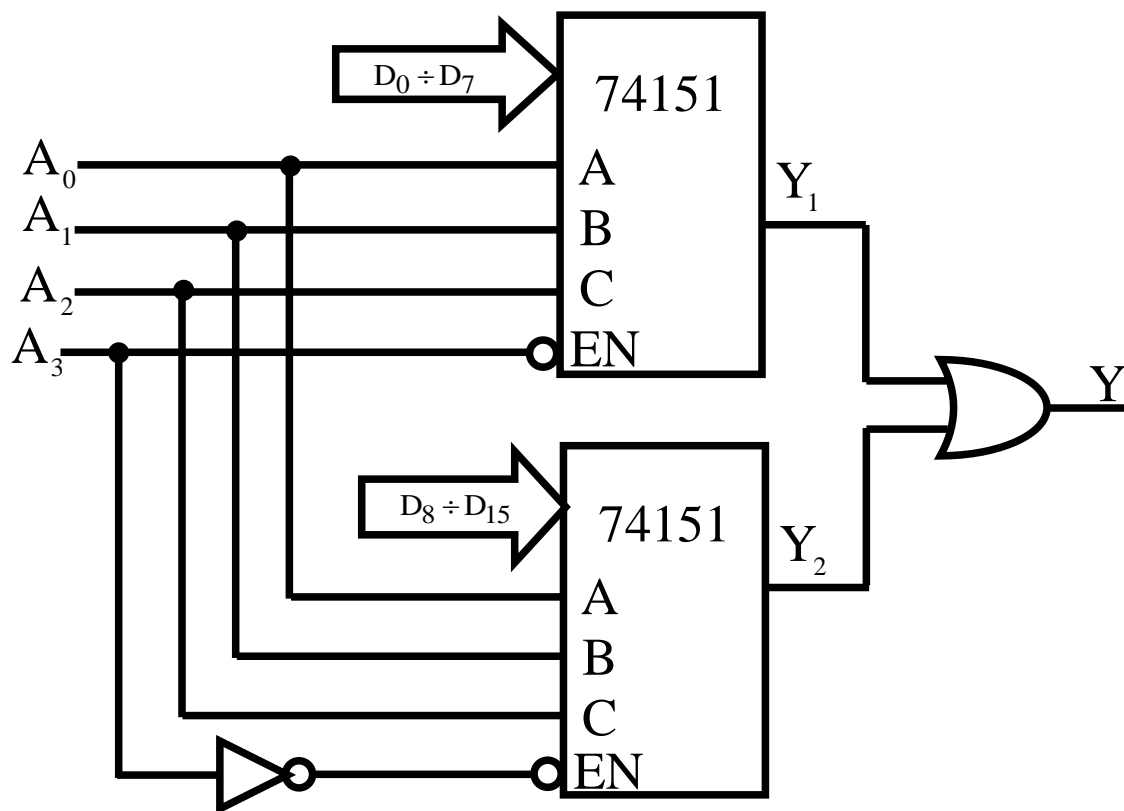
➤ Mạch điện:

➤ IC thực tế: 74151



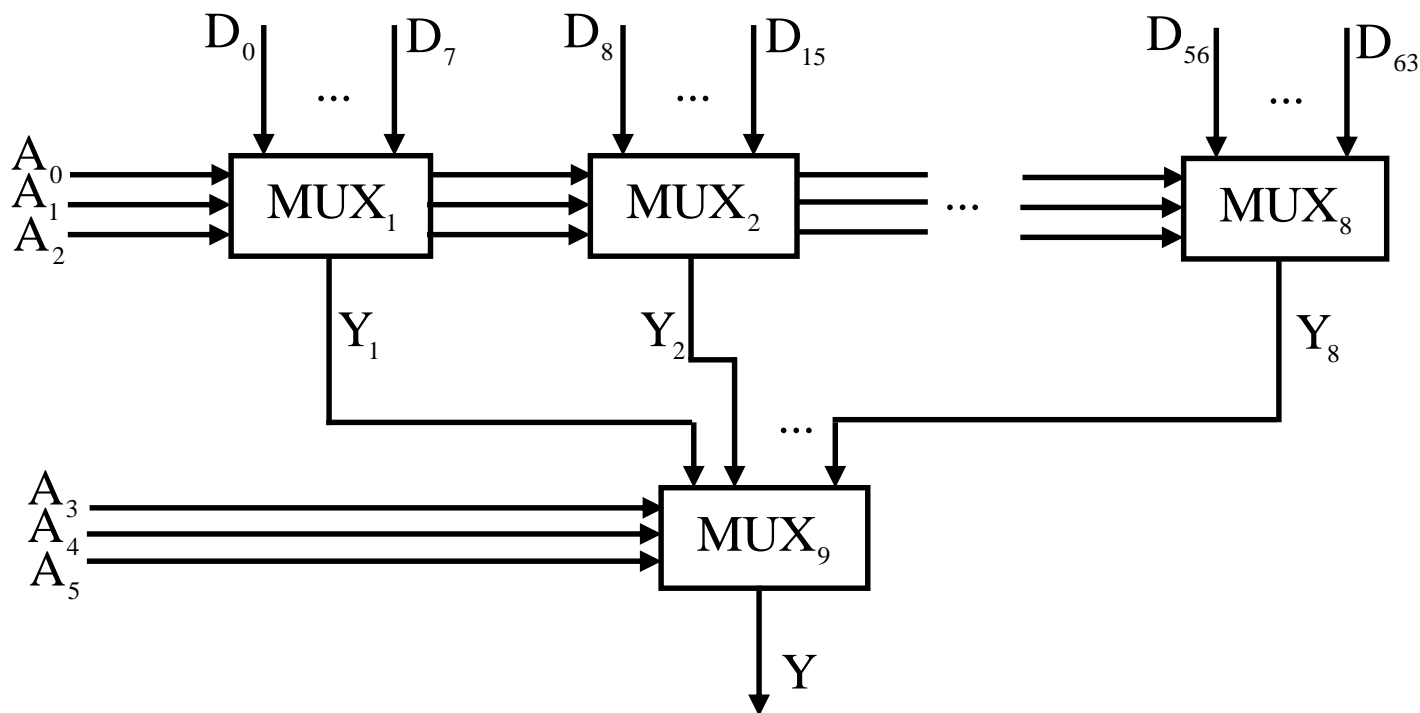
### 3.5.1. Mạch hợp kênh – MUX (4)

- Mở rộng dung lượng bộ hợp kênh: Sử dụng IC 74151 thực hiện MUX 16 vào – 1 ra.



### 3.5.1. Mạch hợp kênh – MUX (5)

- Mở rộng dung lượng bộ hợp kênh: Sử dụng IC 74151 thực hiện MUX 64 vào – 1 ra.

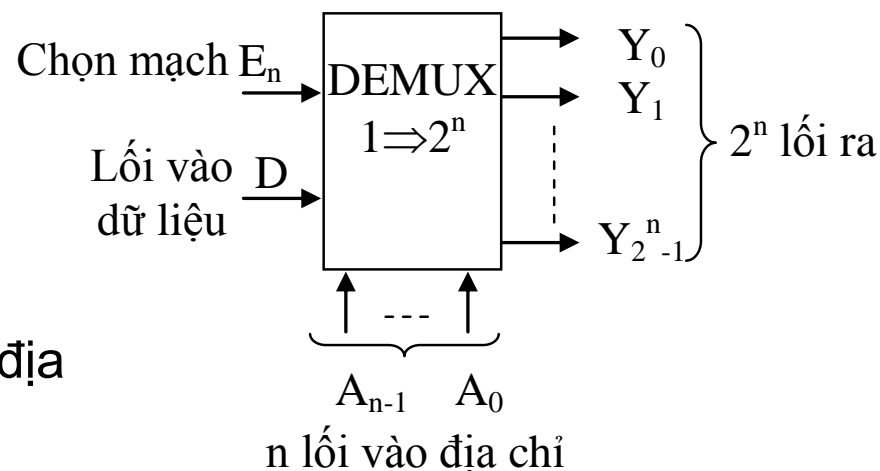


### 3.5.2. Mạch phân kênh – DEMUX (1)

➤ *Chức năng:* lựa chọn nối một đầu ra với đầu vào dữ liệu.

➤ *Sơ đồ khối:*

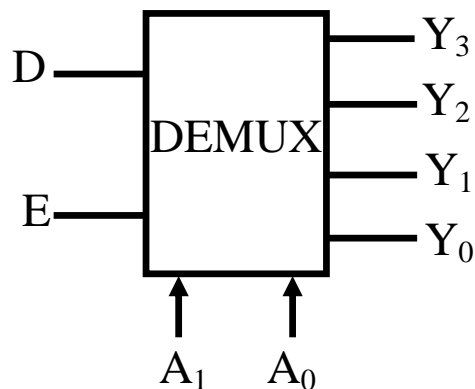
- Gồm: 1 lối vào dữ liệu  $D$ ,  $n$  lối vào địa chỉ, 1 lối vào chọn mạch  $E$  và  $2^n$  lối ra.
- Tùy theo giá trị của  $n$  lối vào địa chỉ mà một trong số các lối ra sẽ bằng giá trị ở lối vào ( $D$ ).
- Nếu giá trị thập phân của  $n$  lối vào địa chỉ bằng  $j$  thì  $Y_j = D$ .



### 3.5.2. Mạch phân kênh – DEMUX (2)

➤ *Bài toán:* Thiết kế DEMUX hai lối vào địa chỉ, một lối vào điều khiển mức tích cực cao.

➤ *Sơ đồ khối:*



*Bảng trạng thái:*

E	A1	A0	Y0	Y1	Y2	Y3
0	x	x	0	0	0	0
1	0	0	D	0	0	0
1	0	1	0	D	0	0
1	1	0	0	0	D	0
1	1	1	0	0	0	D

➤ *Biểu thức:*

$$Y_0 = E \cdot \overline{A_1} \cdot \overline{A_0} \cdot D$$

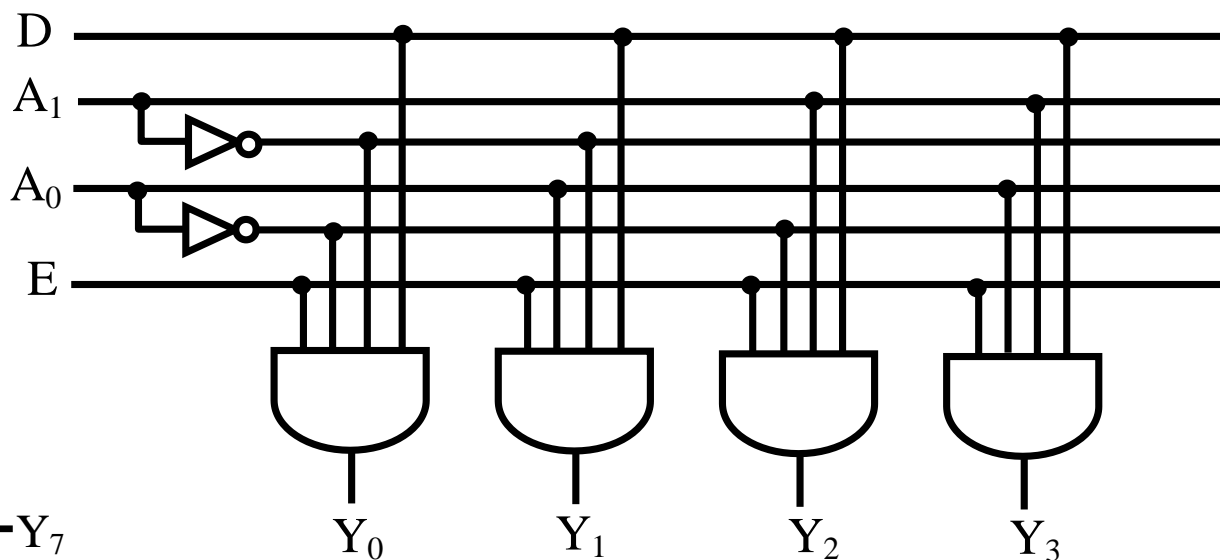
$$Y_1 = E \cdot \overline{A_1} \cdot A_0 \cdot D$$

$$Y_2 = E \cdot A_1 \cdot \overline{A_0} \cdot D$$

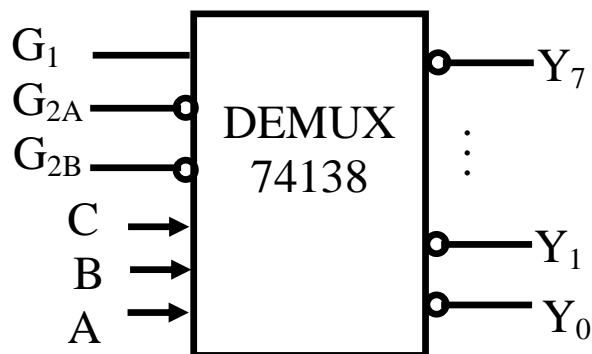
$$Y_3 = E \cdot A_1 \cdot A_0 \cdot D$$

### 3.5.2. Mạch phân kênh – DEMUX (3)

➤ Mạch điện:



➤ IC thực tế: 74138



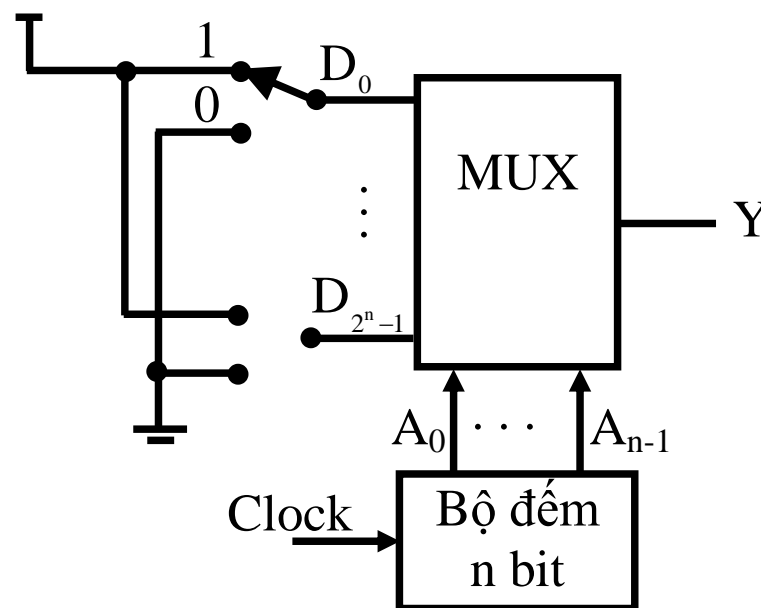
## 3.5.3. Một số ứng dụng của bộ hợp kênh – phân kênh (1)

- Chuyển đổi luồng dữ liệu từ nối tiếp sang song song và ngược lại.
  - Từ nối tiếp sang song song: 1 đầu vào – nhiều đầu ra, sử dụng IC phân kênh.
  - Từ song song sang nối tiếp: nhiều đầu vào – 1 đầu ra, sử dụng IC hợp kênh.



### 3.5.3. Một số ứng dụng của bộ hợp kênh – phân kênh (2)

- Tạo dãy nhị phân tuần hoàn: sử dụng MUX để tạo dãy nhị phân tuần hoàn.
- Thay đổi tính chất của dãy:
  - Thay đổi giá trị đầu vào dữ liệu MUX.
  - Thay đổi tần số bộ đếm.



### 3.5.3. Một số ứng dụng của bộ hợp kênh – phân kênh (3)

- Tạo hàm logic: sử dụng bộ hợp kênh hoặc bộ giải mã địa chỉ chỉ có thể thực hiện được các hàm logic.
- Ví dụ: Thực hiện hàm logic sau  $f(A,B,C) = \sum(2,3,5,6)$  sử dụng:
  - a. Bộ giải mã địa chỉ (Decoder)
  - b. Bộ hợp kênh 3 lối vào địa chỉ (Mux 8 – 1)
  - c. Bộ hợp kênh 2 lối vào địa chỉ (Mux 4 – 1)

### 3.5.3. (4) – Ví dụ: Tạo hàm logic

➤ Bảng trạng thái hàm  $f$ :  $f(A,B,C) = \sum(2,3,5,6)$

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

### 3.5.3. (5) – Ví dụ: a. Sử dụng Decoder

➤ A, B, C là 3 đầu vào của Decoder.

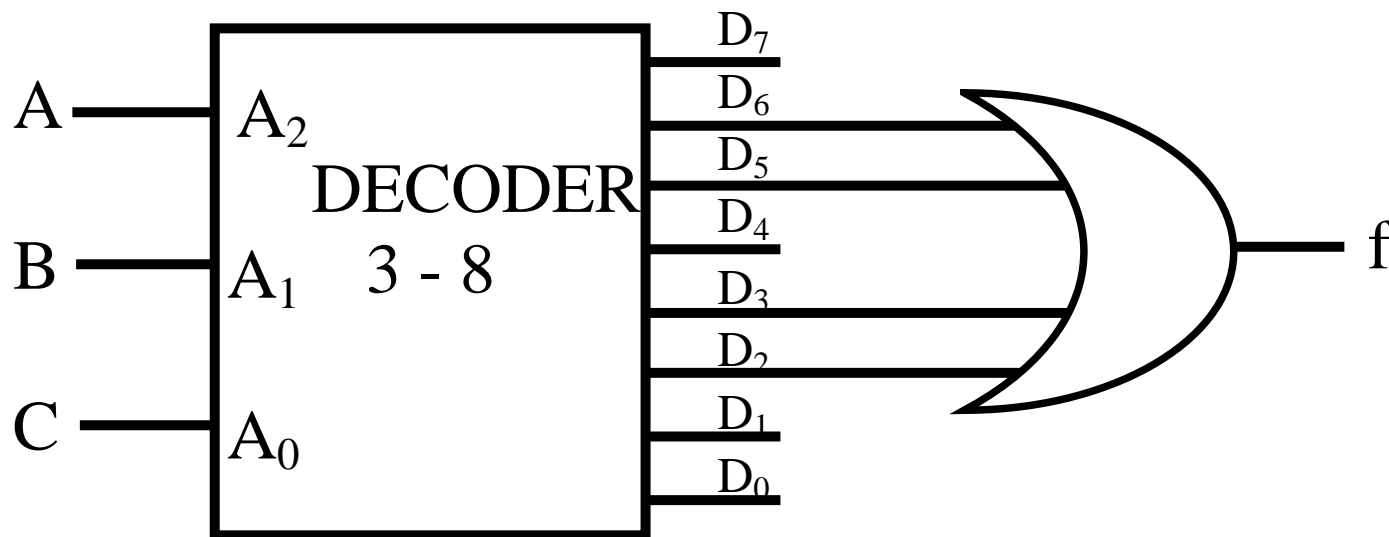
➤ Bảng trạng thái:

A	B	C	f	Decoder 3 - 8
0	0	0	0	D0 = 1
0	0	1	0	D1 = 1
0	1	0	1	D2 = 1
0	1	1	1	D3 = 1
1	0	0	0	D4 = 1
1	0	1	1	D5 = 1
1	1	0	1	D6 = 1
1	1	1	0	D7 = 1

$$\Rightarrow f(A, B, C) = D_2 + D_3 + D_5 + D_6$$

### 3.5.3. (6) – Ví dụ: a. Sử dụng Decoder

➤ Sơ đồ khối:  $f(A, B, C) = D_2 + D_3 + D_5 + D_6$



### 3.5.3. (7) – Ví dụ: b. Sử dụng Mux 8 - 1

➤ A, B, C là 3 đầu vào địa chỉ của Mux.

➤ Bảng trạng thái:

A	B	C	f = Y	Mux 8 - 1
0	0	0	0	$Y = D_0$
0	0	1	0	$Y = D_1$
0	1	0	1	$Y = D_2$
0	1	1	1	$Y = D_3$
1	0	0	0	$Y = D_4$
1	0	1	1	$Y = D_5$
1	1	0	1	$Y = D_6$
1	1	1	0	$Y = D_7$

### 3.5.3. (8) – Ví dụ: b. Sử dụng Mux 8 - 1

➤ Sơ đồ khối:

$D_0 = 0$

$D_1 = 0$

$D_2 = 1$

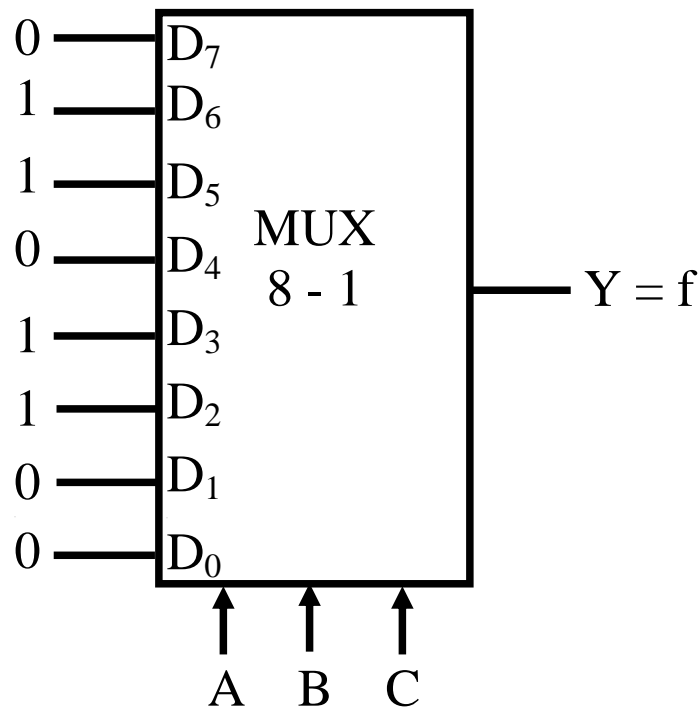
$D_3 = 1$

$D_4 = 0$

$D_5 = 1$

$D_6 = 1$

$D_7 = 0$



## 3.5.3. (9) – Ví dụ: c. Sử dụng Mux 4 - 1

- Chọn A, B là 2 đầu vào địa chỉ của Mux.
- C là đầu vào dữ liệu.
- Bảng trạng thái:

A	B	C	f = Y	Mux 4 - 1
0	0	0	0	Y = D0
0	0	1	0	
0	1	0	1	Y = D1
0	1	1	1	
1	0	0	0	Y = D2
1	0	1	1	
1	1	0	1	Y = D3
1	1	1	0	



### 3.5.3. (10) – Ví dụ: c. Sử dụng Mux 4 - 1

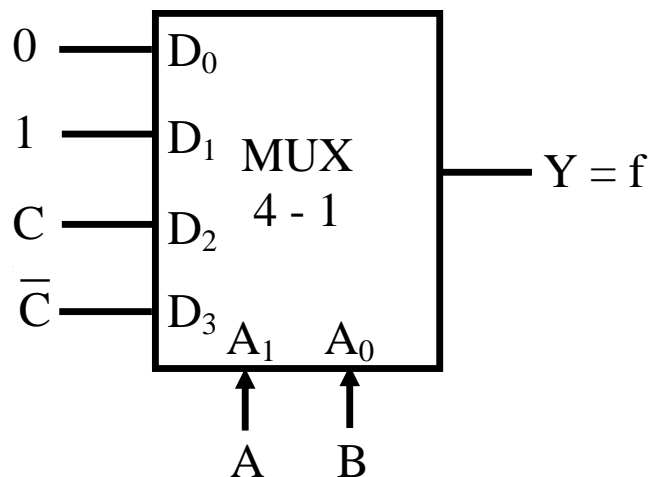
➤ Sơ đồ khối:

$$D_0 = 0$$

$$D_1 = 1$$

$$D_2 = C$$

$$D_3 = \bar{C}$$

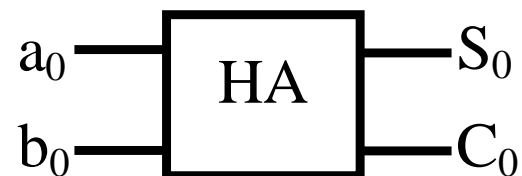


## Chương 3 – MẠCH LOGIC TỔ HỢP

- 3.1. Khái niệm chung
- 3.2. Phân tích mạch logic tổ hợp
- 3.3. Thiết kế mạch logic tổ hợp
- 3.4. Mạch mã hóa – giải mã
- 3.5. Mạch hợp kênh – phân kênh
- 3.6. Mạch số học**
- 3.7. Mã phát hiện sai

### 3.6.1. Mạch tổng (1) – Mạch bán tổng

➤ Sơ đồ khối:



➤ Bảng trạng thái:

a0	b0	S0	C0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

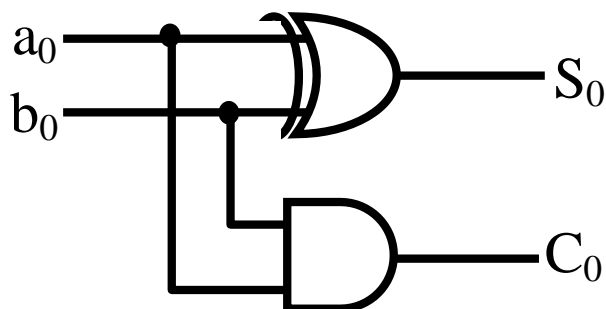
### 3.6.1. Mạch tổng (2) – Mạch bán tổng

➤ Biểu thức:

$$S_0 = a_0 \oplus b_0$$

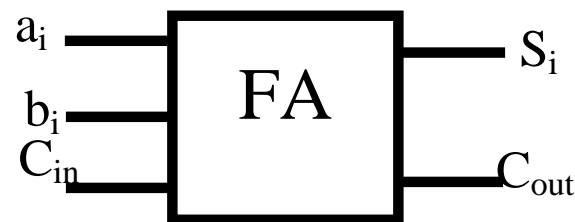
$$C_0 = a_0 \cdot b_0$$

➤ Mạch điện:



### 3.6.1. Mạch tổng (3) – Mạch tổng toàn phần

➤ Sơ đồ khối:



➤ Bảng trạng thái:

$C_{in}$	$a_i$	$b_i$	$S_i$	$C_{out}$
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

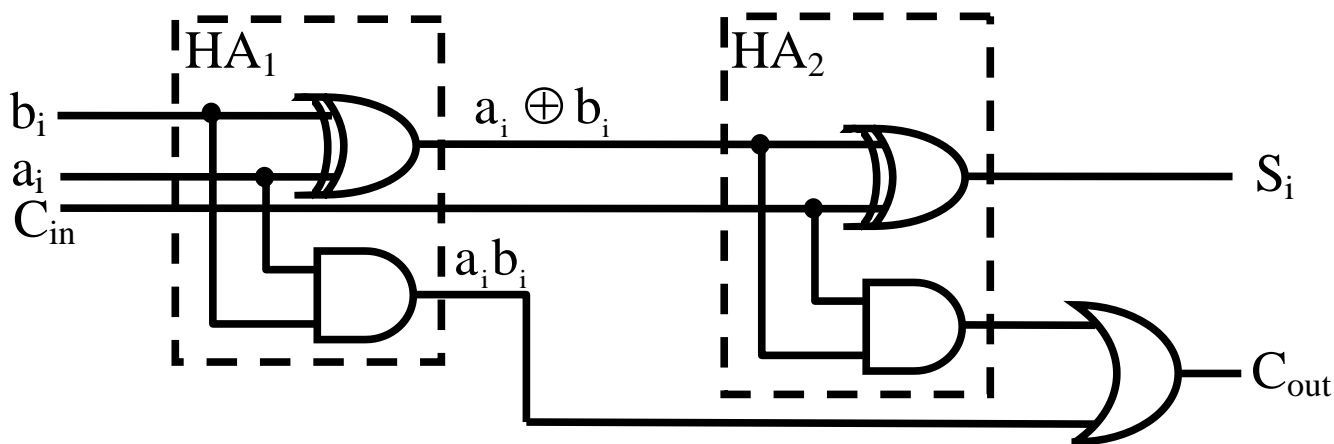
### 3.6.1. Mạch tổng (4) – Mạch tổng toàn phần

➤ *Biểu thức:*

$$S_i = \overline{a_i} \overline{b_i} \overline{C_{in}} + \overline{a_i} \overline{b_i} C_{in} + \overline{a_i} b_i \overline{C_{in}} + a_i \overline{b_i} \overline{C_{in}} + a_i b_i \overline{C_{in}} + \overline{a_i} b_i C_{in} + a_i \overline{b_i} C_{in} + a_i b_i C_{in} = a_i \oplus b_i \oplus C_{in}$$

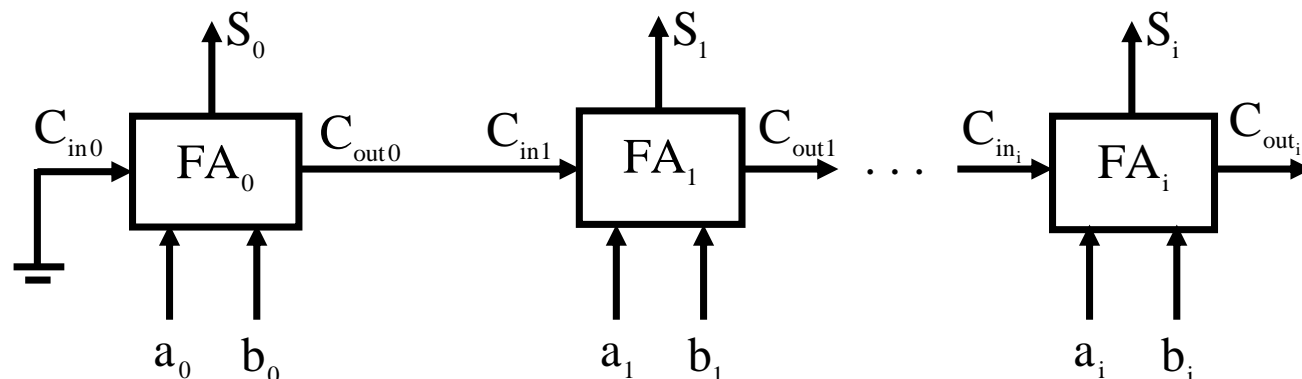
$$C_{out} = \overline{a_i} b_i \overline{C_{in}} + \overline{a_i} b_i C_{in} + a_i \overline{b_i} \overline{C_{in}} + a_i \overline{b_i} C_{in} + a_i b_i \overline{C_{in}} + a_i b_i C_{in} = a_i b_i + (a_i \oplus b_i) C_{in}$$

➤ *Mạch điện:*

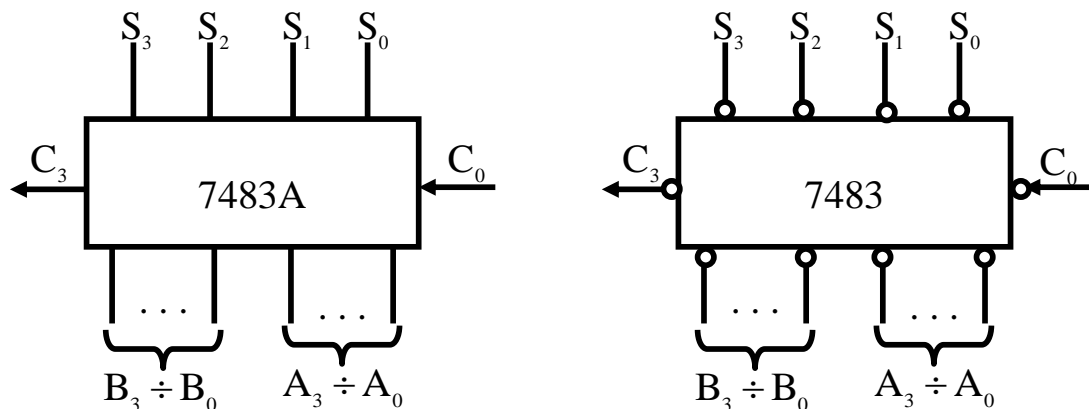


### 3.6.1. Mạch tổng (5) – Mạch cộng nhị phân song song

➤ Sơ đồ khối:



➤ IC thực tế: 7483, 7483A



### 3.6.2. Mạch hiệu (1) – Mạch bán hiệu

➤ Sơ đồ khối:



➤ Bảng trạng thái:

a0	b0	D0	B0
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

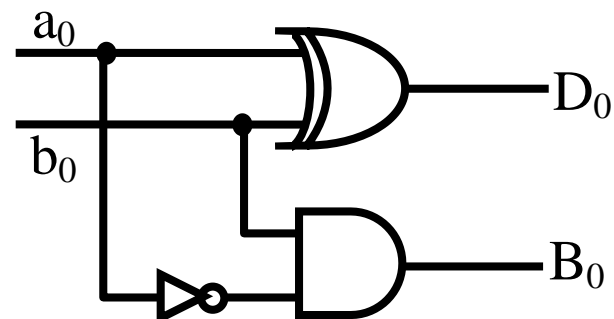


### 3.6.2. Mạch hiệu (2) – Mạch bán hiệu

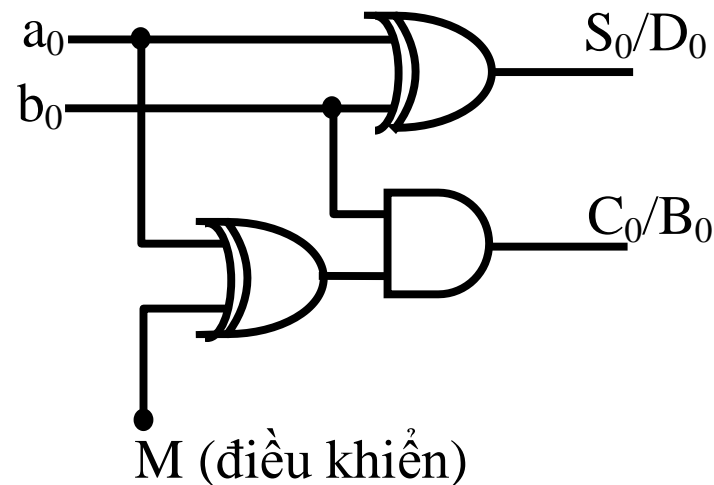
➤ Biểu thức:

$$D_0 = a_0 \oplus b_0$$

$$B_0 = \overline{a_0} b_0$$

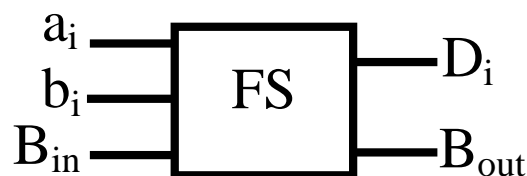


➤ Mạch điện:



### 3.6.2. Mạch hiệu (3) – Mạch hiệu toàn phần

➤ Sơ đồ khối:



➤ Bảng trạng thái:

$a_i$	$b_i$	$B_{in}$	$D_i$	$B_{out}$
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

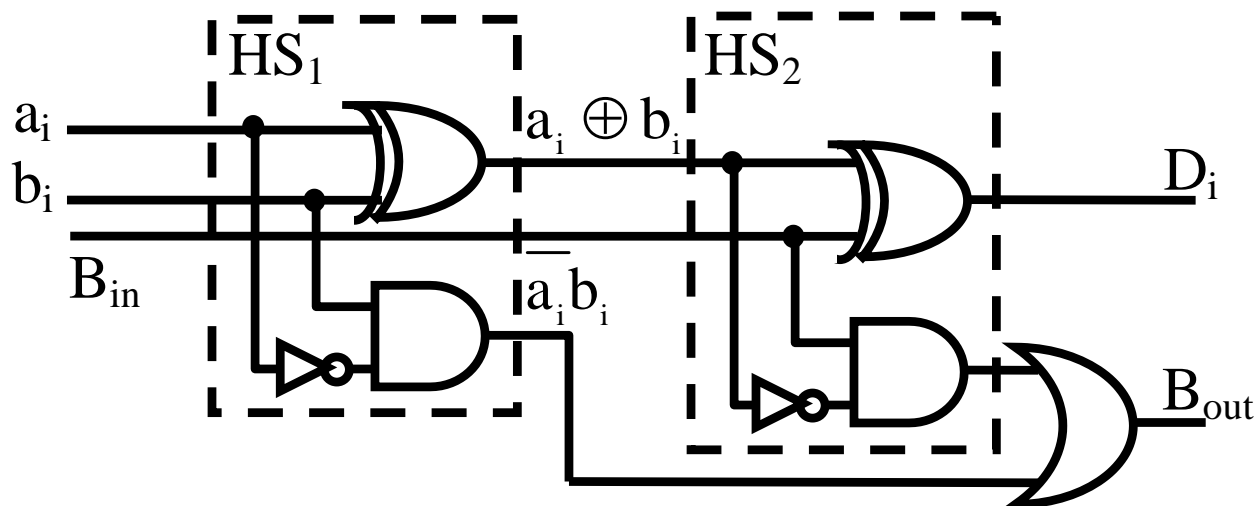
### 3.6.2. Mạch hiệu (4) – Mạch hiệu toàn phần

➤ *Biểu thức:*

$$D_i = a_i \oplus b_i \oplus B_{in}$$

$$B_{out} = \overline{a_i} b_i + \overline{(a_i \oplus b_i)} B_{in}$$

➤ *Mạch điện:*



### 3.6.3. Mạch so sánh (1) - Bộ so sánh bằng nhau

➤ Bộ so sánh bằng nhau 1 bit:

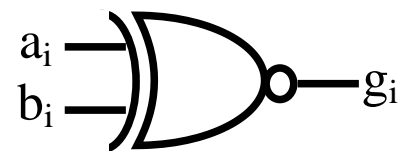
- Bảng trạng thái:

- Biểu thức:

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

- Mạch điện:

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

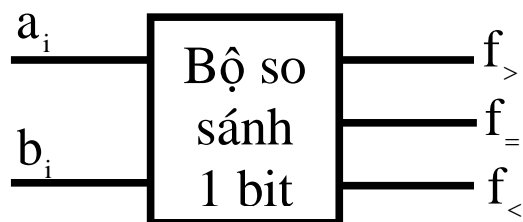


➤ Bộ so sánh bằng nhau 4 bit:

$$A = a_3 a_2 a_1 a_0 \text{ với } B = b_3 b_2 b_1 b_0 \quad A = B \Leftrightarrow \begin{cases} a_3 = b_3 \\ a_2 = b_2 \\ a_1 = b_1 \\ a_0 = b_0 \end{cases}$$

### 3.6.3. Mạch so sánh (2) - Bộ so sánh 1 bit

➤ Sơ đồ khối:



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

➤ Bảng trạng thái:

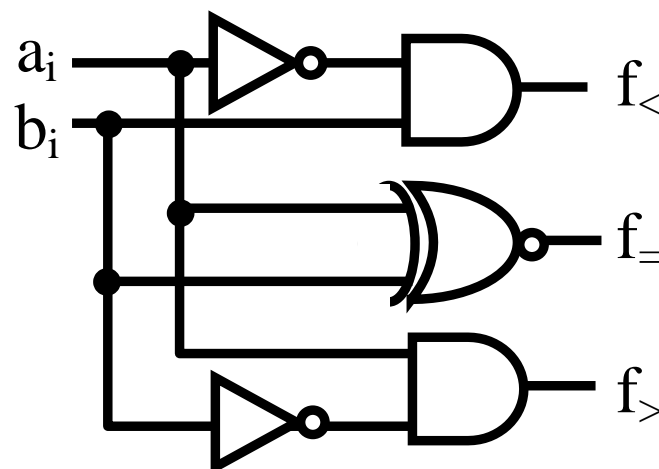
➤ Biểu thức:

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

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

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

➤ Mạch điện:



### 3.6.3. Mạch so sánh (3) - Bộ so sánh 4 bit

➤ So sánh hai số nhị phân 4 bit  $A = a_3a_2a_1a_0$  với  $B = b_3b_2b_1b_0$ .

$A > B$  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  $a_0 > b_0$ .

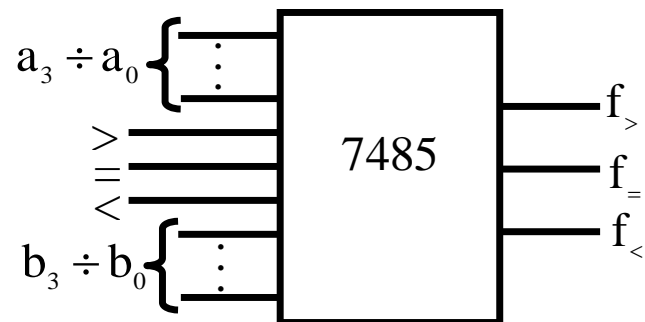
$$f_{>} = \overline{a_3 \cdot b_3} + \overline{a_3 \oplus b_3} \cdot \overline{a_2 \cdot b_2} +$$

$$\overline{a_3 \oplus b_3} \cdot \overline{a_2 \oplus b_2} \cdot \overline{a_1 \cdot b_1} +$$

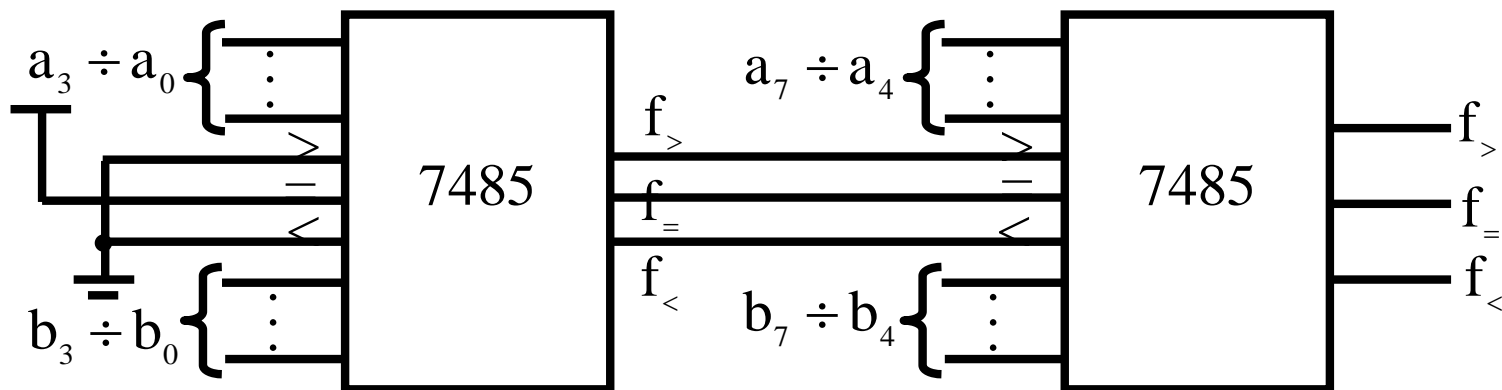
$$\overline{a_3 \oplus b_3} \cdot \overline{a_2 \oplus b_2} \cdot \overline{a_1 \oplus b_1} \cdot \overline{a_0 \cdot b_0}$$

### 3.6.3. Mạch so sánh (4) - Bộ so sánh 4 bit

➤ IC so sánh: 7485



➤ Mở rộng phạm vi so sánh:



## Chương 3 – MẠCH LOGIC TỔ HỢP

- 3.1. Khái niệm chung
- 3.2. Phân tích mạch logic tổ hợp
- 3.3. Thiết kế mạch logic tổ hợp
- 3.4. Mạch mã hóa – giải mã
- 3.5. Mạch hợp kênh – phân kênh
- 3.6. Mạch số học
- 3.7. Mã phát hiện sai**



### 3.7.1. Mã chẵn lẻ (1)

➤ Mã chẵn lẻ: là loại mã phát hiện lỗi thông dụng nhất.

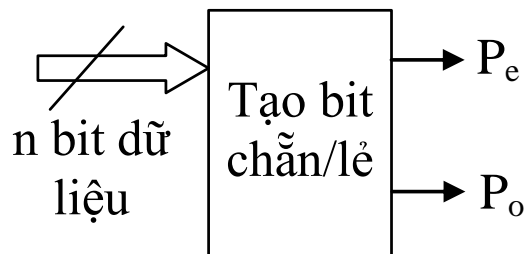
➤ Thiết lập mã chẵn lẻ: thêm một bit chẵn/lẻ (parity bit) vào thông tin, sao cho:

- Tổng số bit '1' là chẵn (mã chẵn hay tính chẵn)
- Tổng số bit '1' là lẻ (mã lẻ hay tính lẻ)

BCD 8421	BCD 8421chẵn $P_C$		BCD 8421 lẻ $P_L$	
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

### 3.7.1. Mã chẵn lẻ (2) - Mạch tạo bit chẵn lẻ

➤ Sơ đồ khối:



➤ Bảng trạng thái:

➤ Biểu thức:

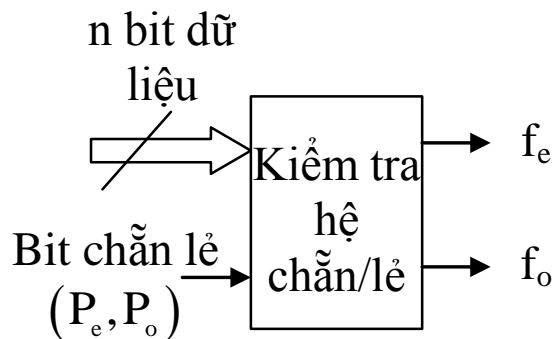
$$p_e = d_1 \oplus d_2 \oplus d_3$$

$$p_o = \overline{p_e} = \overline{d_1 \oplus d_2 \oplus d_3}$$

Vào			Ra	
$d_3$	$d_2$	$d_1$	$p_e$	$p_o$
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

## 3.7.1. Mã chẵn lẻ (3) - Mạch kiểm tra chẵn lẻ

➤ Sơ đồ khối:



➤ Bảng trạng thái:

➤ Biểu thức:

$$f_o = d_3 \oplus d_2 \oplus d_1 \oplus p$$

$$f_e = \overline{f_o} = \overline{d_3 \oplus d_2 \oplus d_1 \oplus p}$$

$d_3$	$d_2$	$d_1$	$P$	$f_e$	$f_o$
0	0	0	0	1	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	1	0
0	1	0	0	0	1
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	0	1
1	0	0	1	1	0
1	0	1	0	1	0
1	0	1	1	0	1
1	1	0	0	1	0
1	1	0	1	0	1
1	1	1	0	0	1
1	1	1	1	1	0

### 3.7.2. Mã Hamming (1)

- Mã Hamming: ghép thêm một số bit kiểm tra P vào thông tin để đảm bảo tính chẵn/lẻ của hệ thống.
- Mã Hamming có khả năng sửa sai một lỗi, có sơ đồ tạo mã và giải mã đơn giản.
- Số bit kiểm tra P và số bit tin tức D phải thỏa mãn biểu thức:  

$$D + P + 1 \leq 2^P$$
- Vị trí các bit P: là các bit có thứ tự  $2^i$  tính từ bit có trọng số nhỏ nhất.
- Ví dụ: với số bit tin tức  $D = 5$ , số bit kiểm tra chẵn/lẻ  $P = 4$ . Cách sắp xếp vị trí các bit như sau:

9	8	7	6	5	4	3	2	1
D <sub>9</sub>	P <sub>3</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	P <sub>2</sub>	D <sub>3</sub>	P <sub>1</sub>	P <sub>0</sub>

(\*)

- Bit chẵn lẻ P có nhiệm vụ đảm bảo tính chẵn lẻ của các bit ở các vị trí mà nó chiếm giữ có giá trị 1 tương ứng theo bảng nhị phân (kể cả nó).

### 3.7.2. Mã Hamming (2) - Mạch tạo mã Hamming

Bit  $P_0$  đảm bảo tính chẵn/lẻ tại các vị trí :

1, 3, 5, 7, 9, 11, 13, 15

Bit  $P_1$  đảm bảo tính chẵn/lẻ tại các vị trí :

2, 3, 6, 7, 10, 11, 14, 15

Bit  $P_2$  đảm bảo tính chẵn/lẻ tại các vị trí :

4, 5, 6, 7, 12, 13, 14, 15

Bit  $P_3$  đảm bảo tính chẵn/lẻ tại các vị trí :

8, 9, 10, 11, 12, 13, 14, 15

Vị trí	Hệ nhị phân $P_3P_2P_1P_0$
1	0 0 0 1
2	0 0 1 0
3	0 0 1 1
4	0 1 0 0
5	0 1 0 1
6	0 1 1 0
7	0 1 1 1
8	1 0 0 0
9	1 0 0 1
10	1 0 1 0
11	1 0 1 1
12	1 1 0 0
13	1 1 0 1
14	1 1 1 0
15	1 1 1 1

## 3.7.2. Mã Hamming (3) - Mạch tạo mã Hamming

9	8	7	6	5	4	3	2	1	
D <sub>9</sub>	P <sub>3</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	P <sub>2</sub>	D <sub>3</sub>	P <sub>1</sub>	P <sub>0</sub>	(*)

\* Tạo mã Hamming chẵn:

$$P_0 \oplus \text{vị trí 3} \oplus \text{vị trí 5} \oplus \text{vị trí 7} \oplus \text{vị trí 9} = 0$$

$$P_0 \oplus D_3 \oplus D_5 \oplus D_7 \oplus D_9 = 0$$

$$P_0 = D_3 \oplus D_5 \oplus D_7 \oplus D_9$$

$$P_1 \oplus D_3 \oplus D_6 \oplus D_7 = 0$$

$\Rightarrow$

$$P_1 = D_3 \oplus D_6 \oplus D_7$$

$$P_2 \oplus D_5 \oplus D_6 \oplus D_7 = 0$$

$$P_2 = D_5 \oplus D_6 \oplus D_7$$

$$P_3 \oplus D_9 = 0$$

$$P_3 = D_9$$

## 3.7.2. Mã Hamming (4) - Mạch tạo mã Hamming

9	8	7	6	5	4	3	2	1	
D <sub>5</sub>	P <sub>3</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	P <sub>2</sub>	D <sub>1</sub>	P <sub>1</sub>	P <sub>0</sub>	(*)

\* Tạo mã Hamming lẻ:

$$P_0 \oplus D_3 \oplus D_5 \oplus D_7 \oplus D_9 = 1$$

$$P_1 \oplus D_3 \oplus D_6 \oplus D_7 = 1$$

$$P_2 \oplus D_5 \oplus D_6 \oplus D_7 = 1$$

$$P_3 \oplus D_9 = 1$$

$\Rightarrow$

$$P_0 = \overline{D_3 \oplus D_5 \oplus D_7 \oplus D_9}$$

$$P_1 = \overline{D_3 \oplus D_6 \oplus D_7}$$

$$P_2 = \overline{D_5 \oplus D_6 \oplus D_7}$$

$$P_3 = \overline{D_9}$$

## 3.7.2. Mã Hamming (5) - Mạch kiểm tra mã Hamming

Khi nhận được từ mã, tiến hành kiểm tra tính chẵn lẻ trong nhóm bit tại các vị trí:

$$S_0: 1, 3, 5, 7, 9, 11, 13, 15$$

$$S_1: 2, 3, 6, 7, 10, 11, 14, 15$$

$$S_2: 4, 5, 6, 7, 12, 13, 14, 15$$

$$S_3: 8, 9, 10, 11, 12, 13, 14, 15$$

\* Kiểm tra mã Hamming: (ví dụ số bit thông tin  $D = 5$ , số bit kiểm tra  $P = 4$ )

$$S_0 = b_1 \oplus b_3 \oplus b_5 \oplus b_7 \oplus b_9$$

Hamming chẵn

Hamming lẻ

$$S_1 = b_2 \oplus b_3 \oplus b_6 \oplus b_7$$

$$S_3 S_2 S_1 S_0$$

= 0000: Không lỗi

$$= 1111$$

$$S_2 = b_4 \oplus b_5 \oplus b_6 \oplus b_7$$

$$S_3 S_2 S_1 S_0$$

≠ 0000: Có lỗi

$$\neq 1111$$

$$S_3 = b_8 \oplus b_9$$

Vị trí bit lỗi:  $S_3 S_2 S_1 S_0$

$$\overline{S_3} \overline{S_2} \overline{S_1} \overline{S_0}$$



### 3.7.2. Mã Hamming (6) - Ví dụ

Cho từ mã thông tin là 10110, hãy xây dựng từ mã Hamming chẵn.

Giả sử phía thu nhận được chuỗi thông tin là: 110010010.

Kiểm tra lỗi và sửa lỗi (nếu có).

*Giải:*

-Số bit thông tin:  $D = 5$ , suy ra số bit kiểm tra:  $P = 4$

- Vị trí các bit P:

9	8	7	6	5	4	3	2	1	
$D_9$	$P_3$	$D_7$	$D_6$	$D_5$	$P_2$	$D_3$	$P_1$	$P_0$	(*)

## 3.7.2. Mã Hamming (7) - Ví dụ

➤ Tìm P:

9	8	7	6	5	4	3	2	1
D <sub>9</sub>	P <sub>3</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	P <sub>2</sub>	D <sub>3</sub>	P <sub>1</sub>	P <sub>0</sub>
1	P <sub>3</sub>	0	1	1	P <sub>2</sub>	0	P <sub>1</sub>	P <sub>0</sub>

$$P_0 \oplus D_3 \oplus D_5 \oplus D_7 \oplus D_9 = 0$$

$$P_0 = D_3 \oplus D_5 \oplus D_7 \oplus D_9 = 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

$$P_1 \oplus D_3 \oplus D_6 \oplus D_7 = 0$$

$$P_1 = D_3 \oplus D_6 \oplus D_7 = 0 \oplus 1 \oplus 0 = 1$$

$$P_2 \oplus D_5 \oplus D_6 \oplus D_7 = 0$$

$$P_2 = D_5 \oplus D_6 \oplus D_7 = 1 \oplus 1 \oplus 0 = 0$$

$$P_3 \oplus D_9 = 0$$

$$P_3 = D_9 = 1$$

➤ Từ mã Hamming chẩn phát đi:

D <sub>9</sub>	P <sub>3</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	P <sub>2</sub>	D <sub>3</sub>	P <sub>1</sub>	P <sub>0</sub>
1	1	0	1	1	0	0	1	0

### 3.7.2. Mã Hamming (8) - Ví dụ

➤ Từ mã nhận được:

b9	b8	b7	b6	b5	b4	b3	b2	b1
1	1	0	0	1	0	0	1	0

➤ Kiểm tra các S:

$$S_0 = b_1 \oplus b_3 \oplus b_5 \oplus b_7 \oplus b_9 = 0 \oplus 0 \oplus 1 \oplus 0 \oplus 1 = 0$$

$$S_1 = b_2 \oplus b_3 \oplus b_6 \oplus b_7 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$S_2 = b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

$$S_3 = b_8 \oplus b_9 = 1 \oplus 1 = 0$$

➤ Lỗi ở vị trí 0110 (vị trí 6). Sửa lỗi:

b9	b8	b7	b6	b5	b4	b3	b2	b1
1	1	0	1	1	0	0	1	0

## Chương 4 – MẠCH LOGIC TUẦN TỰ

### 4.1. Khái niệm chung

### 4.2. Phương pháp mô tả mạch tuần tự

### 4.3. Phần tử nhớ của mạch tuần tự

### 4.4. Mạch tuần tự thông dụng

## Mạch tuần tự

➤ **Đặc điểm:** Trạng thái hoạt động của mạch điện không chỉ phụ thuộc vào các lối vào mà còn phụ thuộc vào trạng thái bên trong trước đó của mạch.

➤ **Mô hình toán học:**

- Tín hiệu ra:

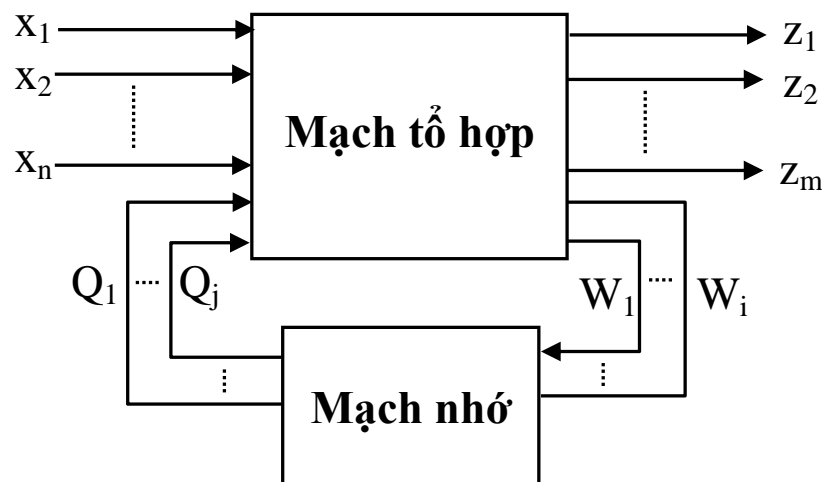
$$Z = f(Q(n), X)$$

- Trạng thái kế tiếp:

$$Q(n+1) = f(Q(n), X)$$

- Hàm kích:

$$W = f(Q(n), X)$$



## Chương 4 – MẠCH LOGIC TUẦN TỰ

4.1. Khái niệm chung

**4.2. Phương pháp mô tả mạch tuần tự**

4.3. Phần tử nhớ của mạch tuần tự

4.4. Mạch tuần tự thông dụng

### 4.2.1. Bảng (1) – Bảng chuyển đổi trạng thái

- Các hàng: ghi các trạng thái trong Q
- Các cột: ghi giá trị tín hiệu vào X
- Các ô: ghi trạng thái kế tiếp ứng với giá trị ở hàng và cột.

		Tín hiệu vào				
Trạng thái trong	<b>X</b> <b>Q</b>	<b>X<sub>1</sub></b>	<b>X<sub>2</sub></b>	.....	<b>X<sub>n</sub></b>	
	<b>Q<sub>1</sub></b>				→	Trạng thái kế tiếp Q <sub>k</sub>
	<b>Q<sub>2</sub></b>					
	.					
	.					
	<b>Q<sub>n</sub></b>					

## 4.2.1. Bảng (2) – Bảng tín hiệu ra

- Các hàng: ghi các trạng thái trong Q
- Các cột: ghi giá trị tín hiệu vào X
- Các ô: ghi tín hiệu ra tương ứng với giá trị ở hàng và cột.

		Tín hiệu vào					
Trạng thái trong	<b>X</b>	<b>X<sub>1</sub></b>	<b>X<sub>2</sub></b>	.....	<b>X<sub>n</sub></b>		
	<b>Q</b>						
	<b>Q<sub>1</sub></b>				→	Tín hiệu ra Z	
	<b>Q<sub>2</sub></b>						
	<b>·</b>						
	<b>Q<sub>n</sub></b>						



## 4.2.1. Bảng (3) – Bảng chuyển đổi trạng thái/tín hiệu ra

- Các hàng: ghi các trạng thái trong Q
- Các cột: ghi giá trị tín hiệu vào X
- Các ô: ghi trạng thái kế tiếp/tín hiệu ra.

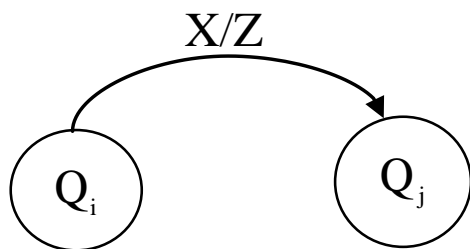
		Tín hiệu vào			
Trạng thái trong	<b>X</b> <b>Q</b>	<b>X<sub>1</sub></b>	<b>X<sub>2</sub></b>	.....	<b>X<sub>n</sub></b>
	<b>Q<sub>1</sub></b>				→
	<b>Q<sub>2</sub></b>				
	.				
	.				
	<b>Q<sub>n</sub></b>				

Trạng thái kế tiếp Q<sub>k</sub>/tín hiệu ra Z

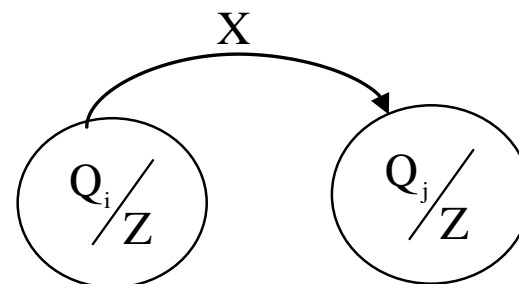
### 4.2.2. Đồ hình trạng thái

- 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
  - K - Tập các cung có hướng

- Mô hình Mealy:



- Mô hình Moore:



## Chương 4 – MẠCH LOGIC TUẦN TỰ

4.1. Khái niệm chung

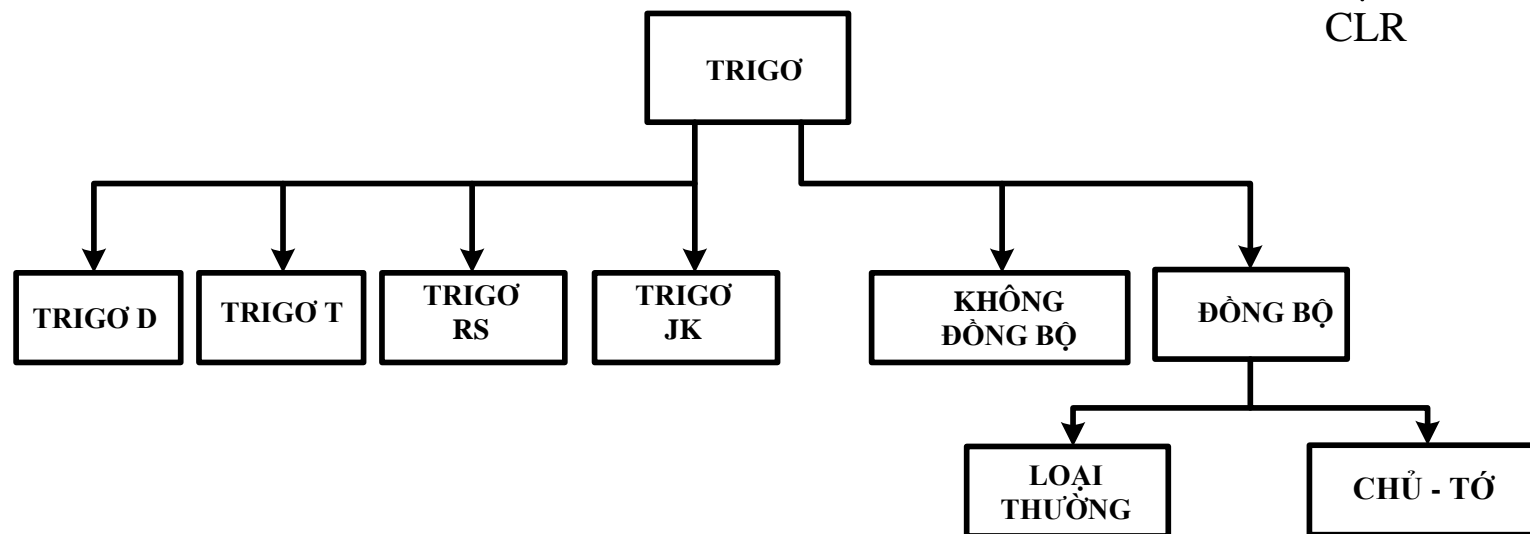
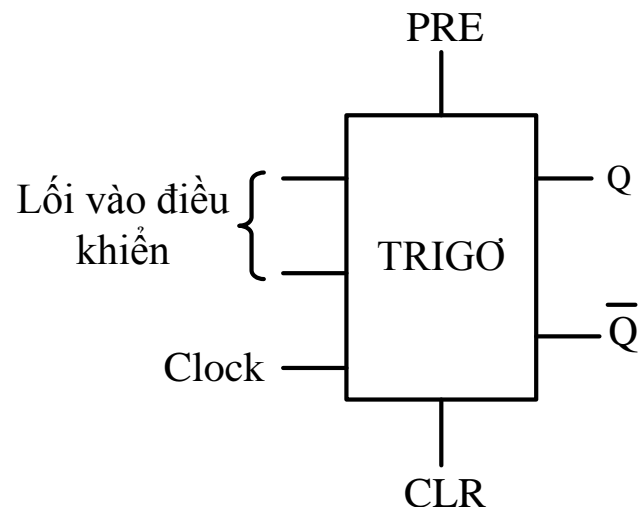
4.2. Phương pháp mô tả mạch tuần tự

**4.3. Phần tử nhớ của mạch tuần tự**

4.4. Mạch tuần tự thông dụng

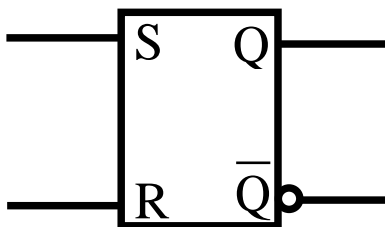
## 4.3.1. Các loại Trigo (1)

- **Định nghĩa:** Trigo là phần tử có khả năng lưu trữ (nhớ) một trong hai trạng thái 0 và 1.
- **Sơ đồ khối:**
- **Phân loại:**

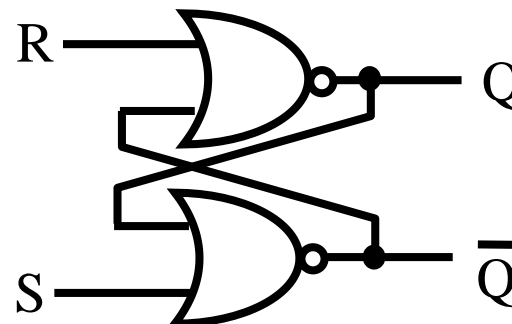
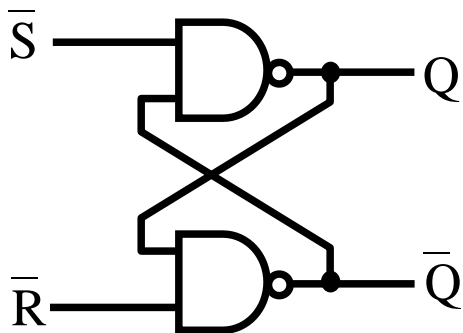


### 4.3.1.1. Trùng RS – RS không đồng bộ (1)

➤ Sơ đồ khối:



➤ Mạch điện:



### 4.3.1.1. Trùng RS – RS không đồng bộ (2)

➤ Hoạt động (sơ đồ dùng cổng NAND):

- Khi  $SR = 00$ :  $S/R/ = 11$ :

+ TH1: giả sử ban đầu  $Q = 0$ ;  $Q/ = 1$ :

$Q = 0$  nên  $Qk/ = 1$

$Q/ = 1$  nên  $Qk = 0$

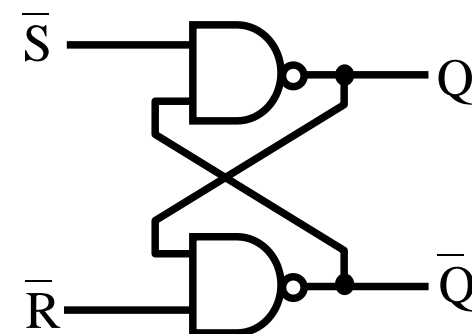
+ TH2: giả sử ban đầu  $Q = 1$ ;  $Q/ = 0$ :

$Q/ = 0$  nên  $Qk = 1$

$Q = 1$  nên  $Qk/ = 0$

Như vậy:  $Qk = Q$

$Qk/ = Q/$



### 4.3.1.1. Trạng RS – RS không đồng bộ (3)

➤ Hoạt động (sơ đồ dùng cổng NAND):

- Khi  $SR = 01$ :  $S/R/ = 10$ :

$R/ = 0$  nên  $Q_k/ = 1$

$Q_k = 0$

Như vậy:  $Q_k = 0$

$Q_k/ = 1$

- Khi  $SR = 10$ :  $S/R/ = 01$ :

$S/ = 0$  nên  $Q_k = 1$

$Q_k/ = 0$

Như vậy:  $Q_k = 1$

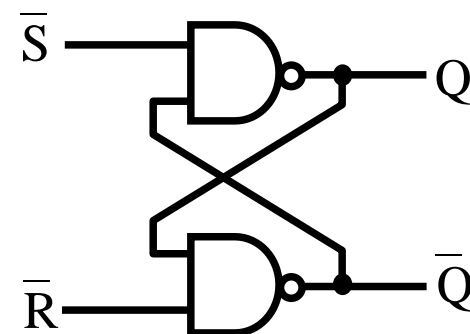
$Q_k/ = 0$

- Khi  $SR = 11$ :  $S/R/ = 00$ :

$S/ = 0$  nên  $Q_k = 1$

$R/ = 0$  nên  $Q_k/ = 1$

Như vậy:  $Q_k = Q_k/ = 1$



### 4.3.1.1. Trùng RS – RS không đồng bộ (4)

➤ Hoạt động (sơ đồ dùng cổng NOR):

- Khi  $SR = 00$ :

+ TH1: giả sử ban đầu  $Q = 0$ ;  $Q' = 1$ :

$Q = 0$  nên  $Q_k' = 1$

$Q' = 1$  nên  $Q_k = 0$

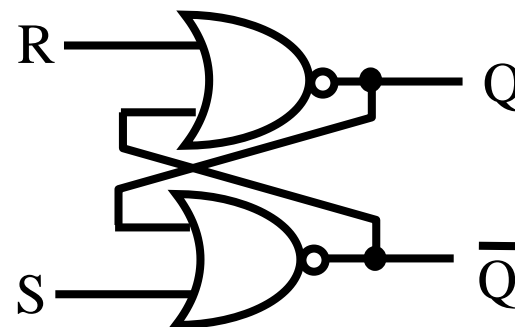
+ TH2: giả sử ban đầu  $Q = 1$ ;  $Q' = 0$ :

$Q' = 0$  nên  $Q_k = 1$

$Q = 1$  nên  $Q_k' = 0$

Như vậy:  $Q_k = Q$

$Q_k' = Q'$





### 4.3.1.1. Trùng RS – RS không đồng bộ (5)

➤ Hoạt động (sơ đồ dùng cổng NOR):

- Khi  $SR = 01$ :

$R = 1$  nên  $Q_k = 0$

$Q_{k+1} = 1$

Như vậy:  $Q_k = 0$

$Q_{k+1} = 1$

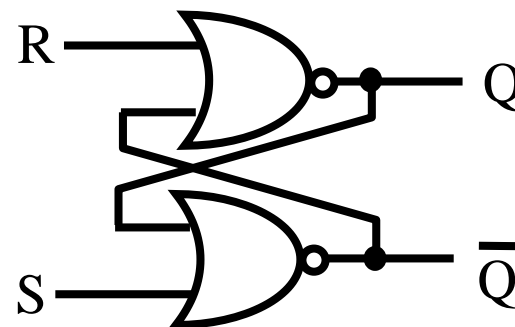
- Khi  $SR = 10$ :

$S = 1$  nên  $Q_{k+1} = 0$

$Q_k = 1$

Như vậy:  $Q_k = 1$

$Q_{k+1} = 0$



- Khi  $SR = 11$ :

$R = 1$  nên  $Q_k = 0$

$S = 1$  nên  $Q_{k+1} = 0$

Như vậy:  $Q_k = Q_{k+1} = 0$

## 4.3.1.1. Trùng RS – RS không đồng bộ (6)

➤ Bảng trạng thái:

S	R	$Q^k$	Chế độ
0	0	Q	Nhớ
0	1	0	Xóa
1	0	1	Lập
1	1	X	Cấm

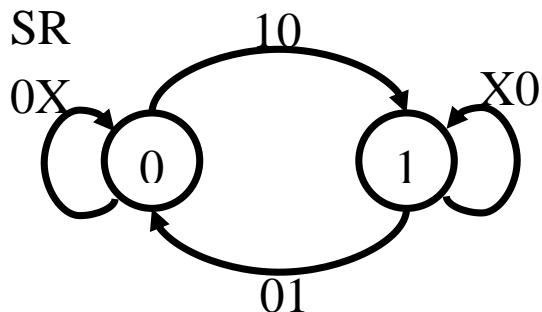
➤ Phương trình đặc trưng:

	SR			
	00	01	11	10
Q				
0	0	0	x	1
1	1	0	x	1

$$\begin{cases} Q^k = S + \bar{R} \cdot Q \\ S \cdot R = 0 \end{cases}$$

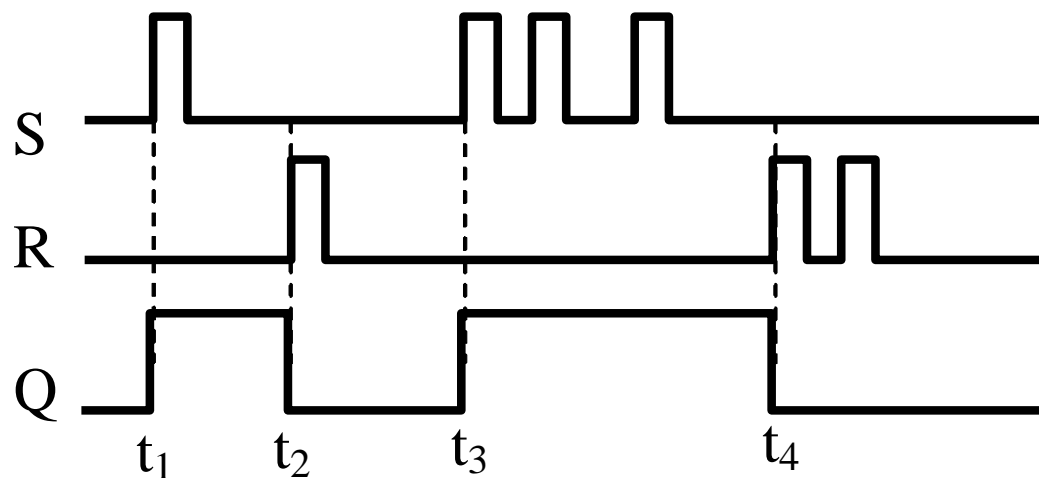
## 4.3.1.1. Trùng RS – RS không đồng bộ (7)

➤ Đồ hình trạng thái:

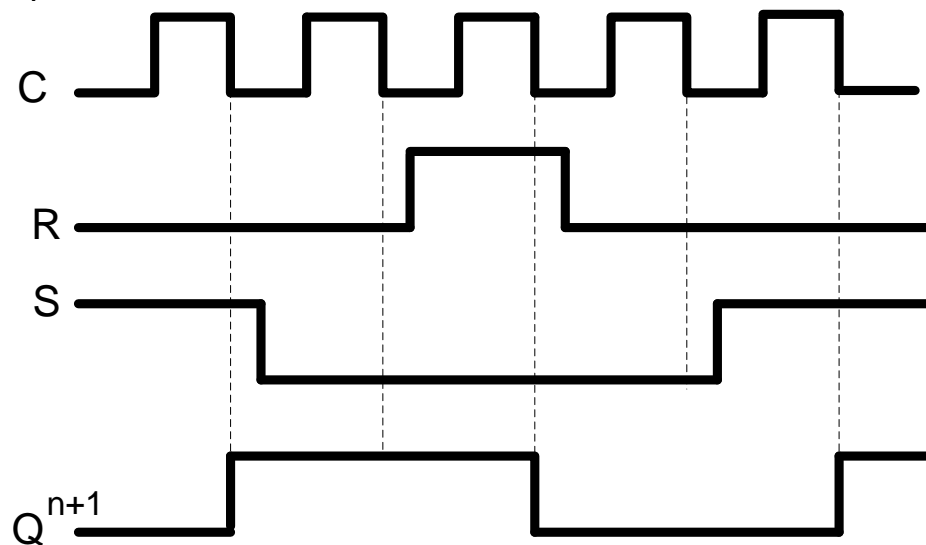
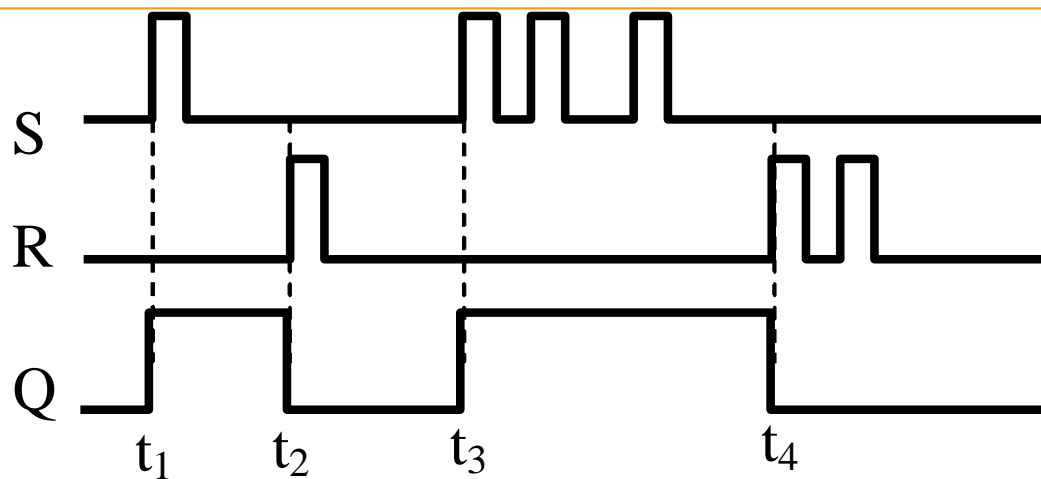


SR	00	01	11	10
Q				
0	0	0	x	1
1	1	0	x	1

➤ Giải đồ xung:

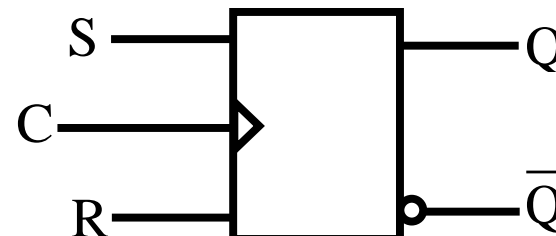
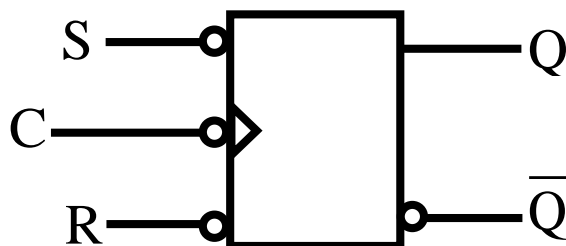


## Giải đồ xung

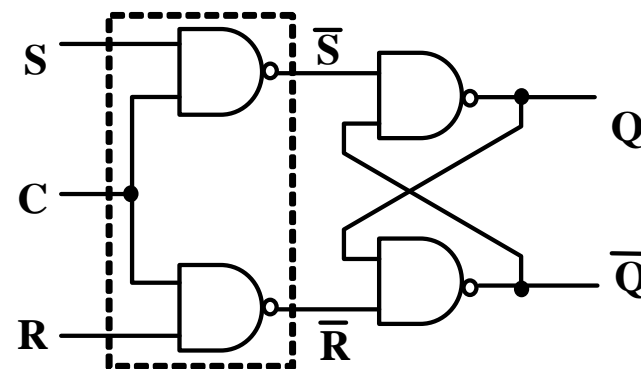
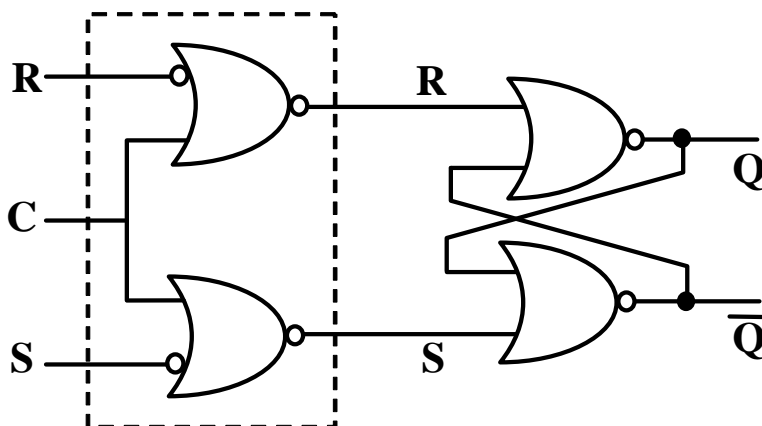


### 4.3.1.1. Trùng RS – RS đồng bộ (1)

➤ Sơ đồ khối:



➤ Mạch điện:



### 4.3.1.1. Triggers RS – RS đồng bộ (2)

➤ Hoạt động (sơ đồ dùng cổng NAND):

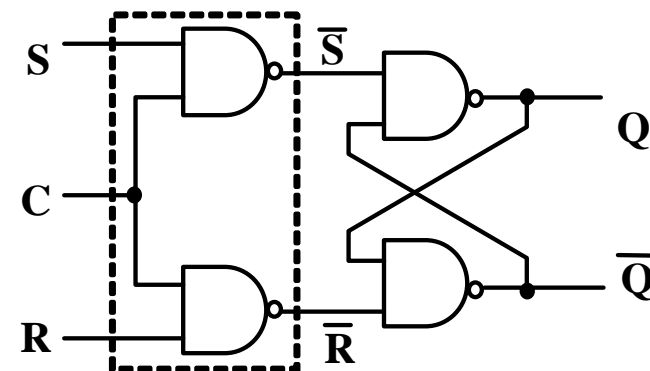
- Khi  $C = 0$ :

$S/R = 11$  không phụ thuộc vào đầu vào của RS đồng bộ.

Mạch ở chế độ nhớ.

- Khi  $C = 1$ :

Hoạt động của RS đồng bộ giống hoàn toàn hoạt động của RS không đồng bộ.



### 4.3.1.1. Triggers RS – RS đồng bộ (3)

➤ Hoạt động (sơ đồ dùng cổng NOR):

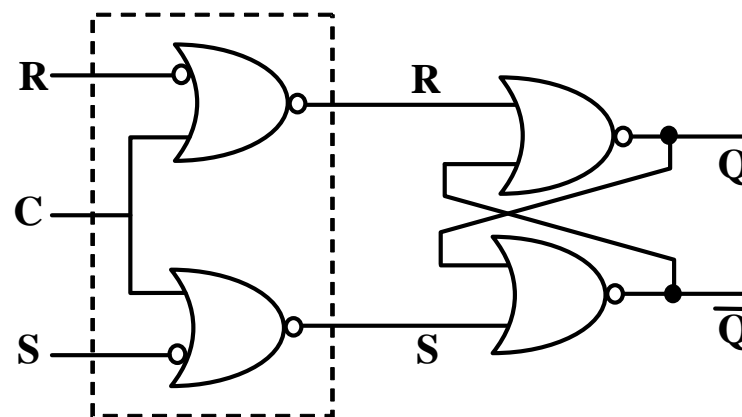
- Khi  $C = 1$ :

$SR = 00$  không phụ thuộc vào đầu vào của RS đồng bộ.

Mạch ở chế độ nhớ.

- Khi  $C = 0$ :

Hoạt động của RS đồng bộ giống hoàn toàn hoạt động của RS không đồng bộ.



## 4.3.1.1. Trùng RS – RS đồng bộ (4)

➤ Bảng trạng thái:

RS NAND

C	S	R	$Q^k$	Chế độ
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

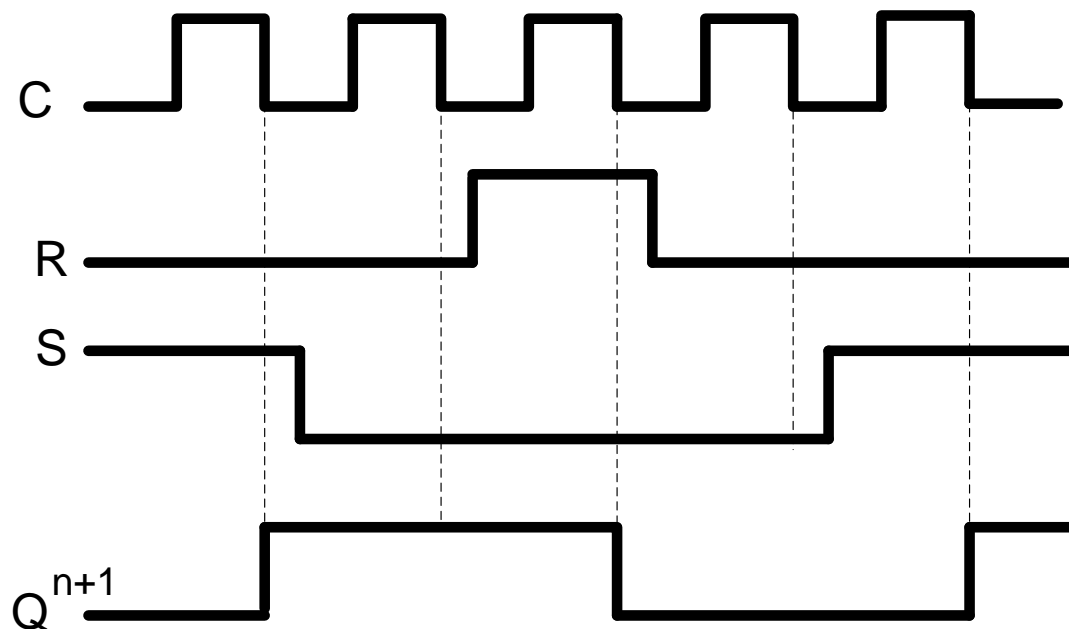
RS NOR

C	S	R	$Q^k$	Chế độ
1	X	X	Q	Nhớ
0	0	0	Q	Nhớ
0	0	1	0	Xóa
0	1	0	1	Lập
0	1	1	X	Cấm



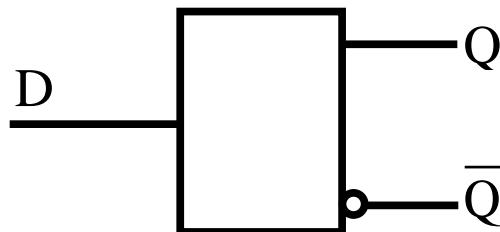
### 4.3.1.1. Trạng RS – RS đồng bộ (5)

➤ Giải đồ xung:

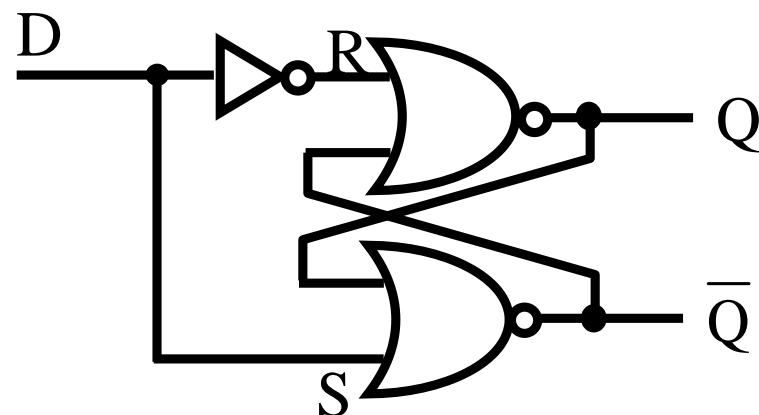
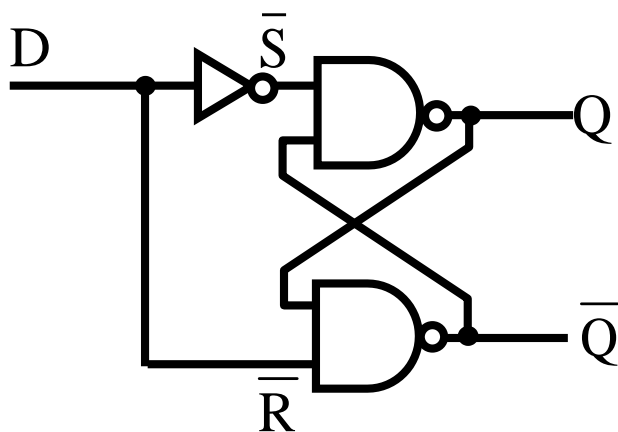


### 4.3.1.2. Trùng D – D không đồng bộ (1)

➤ Sơ đồ khối:



➤ Mạch điện:



### 4.3.1.2. Trùng D – D không đồng bộ (2)

➤ Hoạt động (D dùng cổng NAND):

- Khi  $D = 0$ :

$S/R = 10$ ,  $SR = 01$ : xóa  $Q_k = 0$

- Khi  $D = 1$ :

$S/R = 01$ ,  $SR = 10$ : lập  $Q_k = 1$

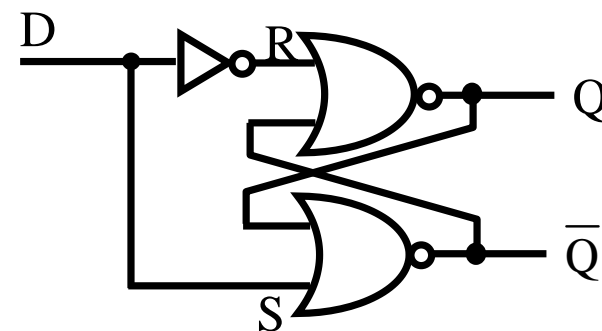
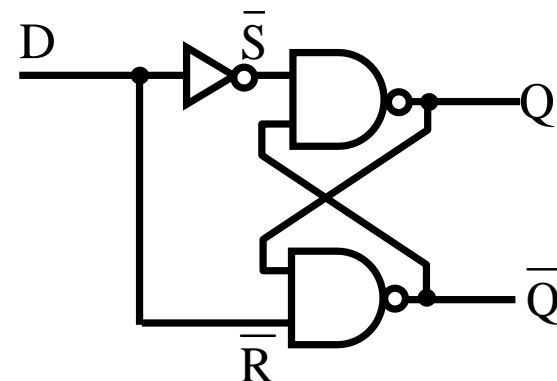
➤ Hoạt động (D dùng cổng NOR):

- Khi  $D = 0$ :

$SR = 01$ : xóa  $Q_k = 0$

- Khi  $D = 1$ :

$SR = 10$ : lập  $Q_k = 1$



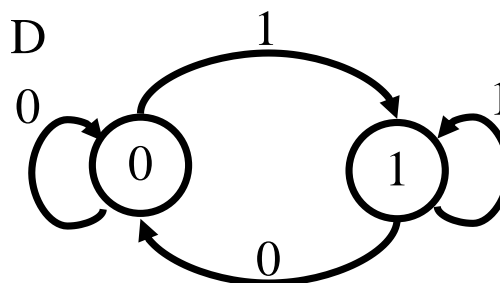
### 4.3.1.2. Trùng D – D không đồng bộ (3)

➤ Bảng trạng thái:

Q	D	Q <sup>k</sup>
0	0	0
0	1	1
1	0	0
1	1	1

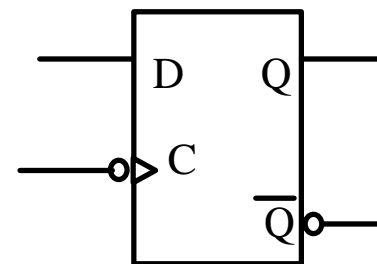
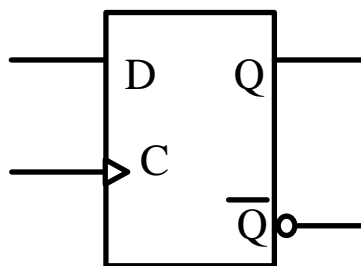
➤ Phương trình đặc trưng:  $Q^k = D$

➤ Đồ hình trạng thái:

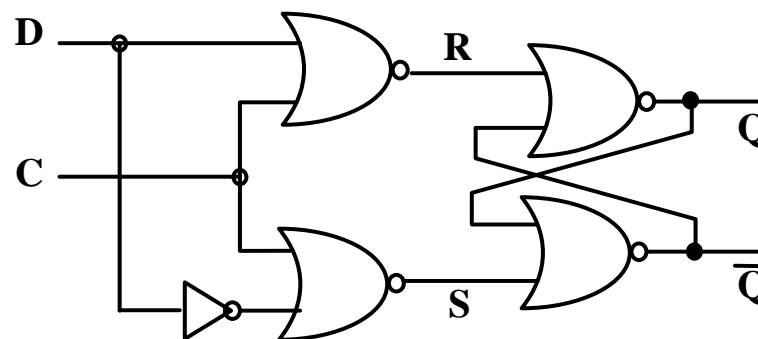
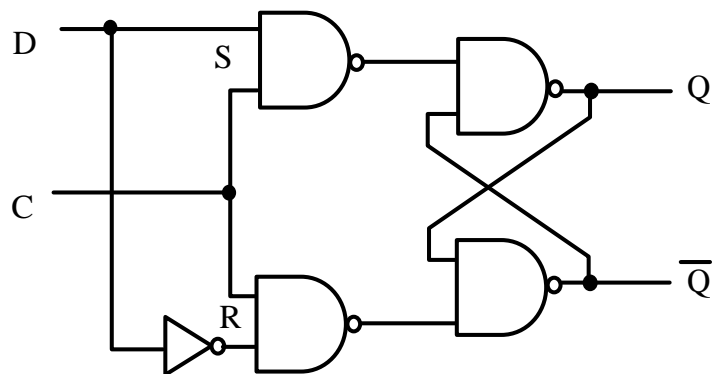


### 4.3.1.2. Trùng D – D đồng bộ (1)

➤ Sơ đồ khối:



➤ Mạch điện:



### 4.3.1.2. Trùng D – D đồng bộ (2)

➤ Hoạt động (D dùng cổng NAND):

- Khi  $C = 0$ :

$S/R = 11$ ,  $SR = 00$ :  $Q_k = Q$

- Khi  $C = 1$ :

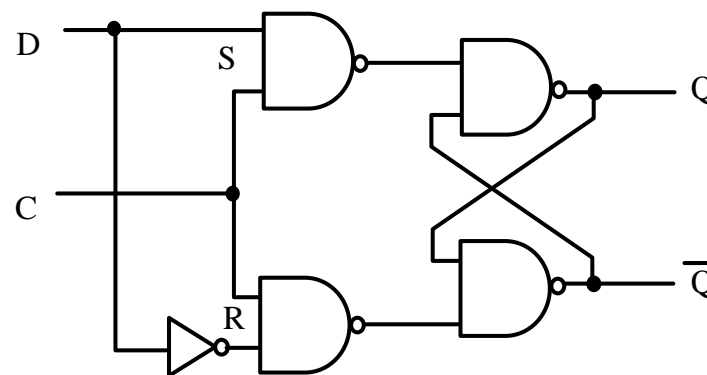
+ Nếu  $D = 0$ :

$S/R = 10$ ,  $SR = 01$ : xóa  $Q_k = 0$

+ Nếu  $D = 1$ :

$S/R = 01$ ,  $SR = 10$ : lập  $Q_k = 1$

D đồng bộ hoạt động giống D không đồng bộ.



### 4.3.1.2. Trùng D – D đồng bộ (3)

➤ Hoạt động (D dùng cổng NOR):

- Khi  $C = 1$ :

$SR = 00$ :  $Q_k = Q$

- Khi  $C = 0$ :

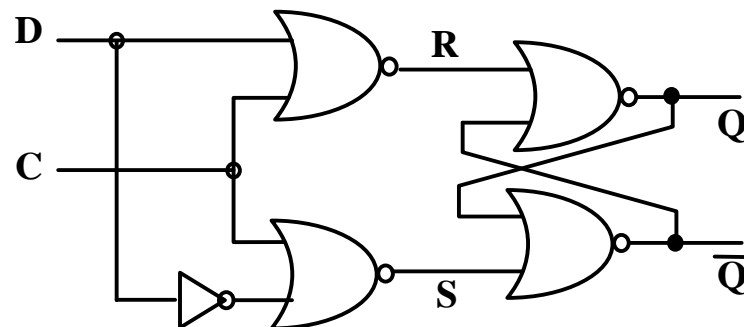
+ Nếu  $D = 0$ :

$SR = 01$ : xóa  $Q_k = 0$

+ Nếu  $D = 1$ :

$SR = 10$ : lập  $Q_k = 1$

D đồng bộ hoạt động giống D không đồng bộ.



### 4.3.1.2. Trùng D – D đồng bộ (4)

➤ *Bảng trạng thái:*

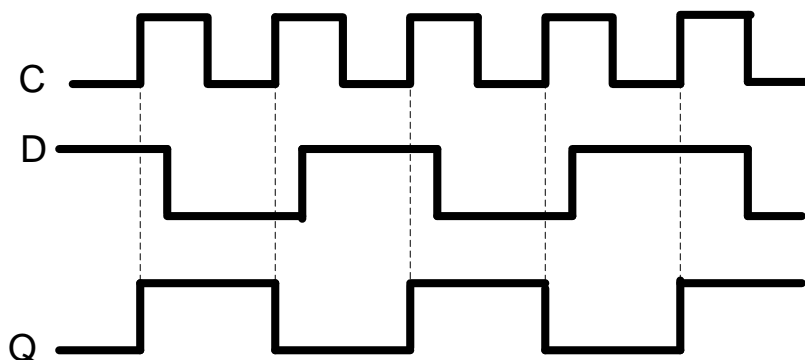
D đồng bộ dùng NAND

C	D	$Q^k$
0	x	Q
1	0	0
1	1	1

D đồng bộ dùng NOR

C	D	$Q^k$
1	x	Q
0	0	0
0	1	1

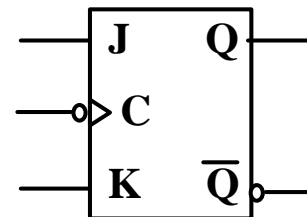
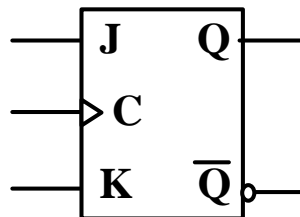
➤ *Giải đồ xung:*



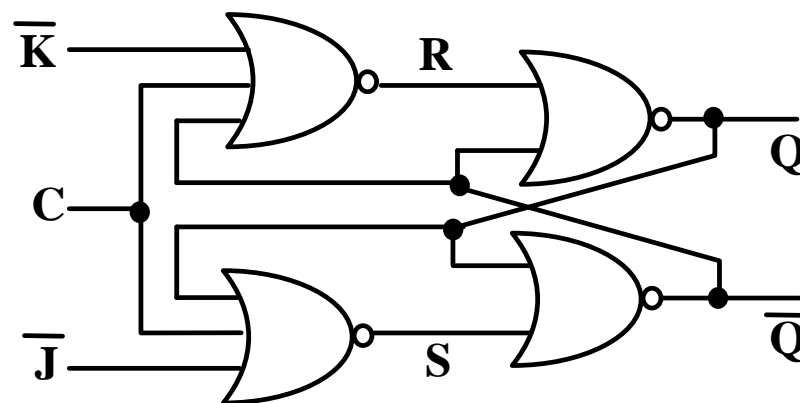
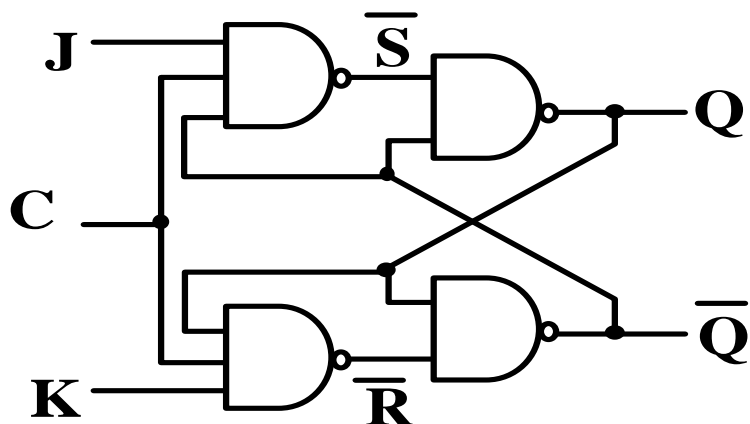


## 4.3.1.3. Trgger JK đồng bộ (1)

➤ Sơ đồ khối:



➤ Mạch điện:



### 4.3.1.3. Trùng JK đồng bộ (2)

➤ Hoạt động (JK dùng cổng NAND):

- Khi  $C = 0$ :

$S/R/ = 11$ ,  $SR = 00$ :  $Q_k = Q$

- Khi  $C = 1$ :

+ Nếu  $JK = 00$ :

$S/R/ = 11$ ,  $SR = 00$ :  $Q_k = Q$

+ Nếu  $JK = 01$ :

TH1: Giả sử ban đầu  $Q = 0$ ,  $Q/ = 1$ :

$J = 0$  nên  $S/ = 1$ ;  $Q = 0$  nên  $R/ = 1$

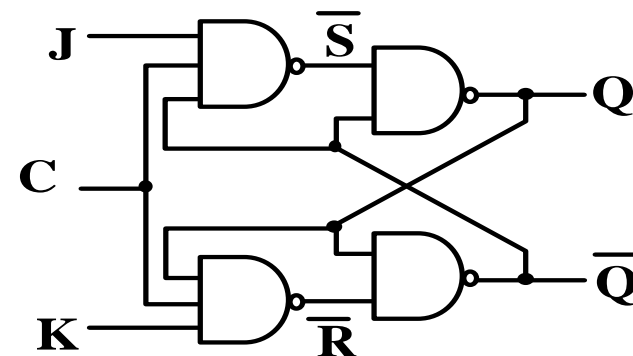
$S/R/ = 11$ ,  $SR = 00$ :  $Q_k = Q = 0$ ;  $Q_k/ = 1$

TH2: Giả sử ban đầu  $Q = 1$ ,  $Q/ = 0$ :

$J = 0$  nên  $S/ = 1$ ;  $Q = 1$  nên  $R/ = 0$

$S/R/ = 10$ ,  $SR = 01$ : Xóa  $Q_k = 0$ ;  $Q_k/ = 1$

Như vậy,  $Q_k = 0$ ,  $Q_k/ = 1$



### 4.3.1.3. Trgger JK đồng bộ (3)

➤ Hoạt động (JK dùng cổng NAND):

- Khi  $C = 1$ :

+ Nếu  $JK = 10$ :

TH1: Giả sử ban đầu  $Q = 0$ ,  $\bar{Q} = 1$ :

$K = 0$  nên  $\bar{R} = 1$ ;  $Q = 0$  nên  $S = 0$

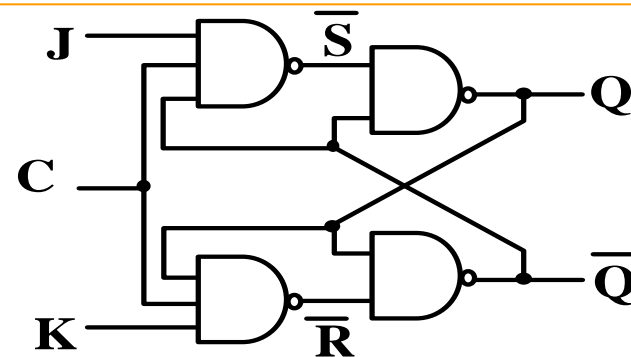
$S/\bar{R} = 01$ ,  $SR = 10$ : Lập  $Q_k = 1$ ;  $Q_k = 0$

TH2: Giả sử ban đầu  $Q = 1$ ,  $\bar{Q} = 0$ :

$K = 0$  nên  $\bar{R} = 1$ ;  $Q = 0$  nên  $S = 1$

$S/\bar{R} = 11$ ,  $SR = 00$ :  $Q_k = Q = 1$ ;  $Q_k = 0$

Như vậy,  $Q_k = 1$ ,  $Q_k = 0$



### 4.3.1.3. Trgger JK đồng bộ (4)

➤ Hoạt động (JK dùng cổng NAND):

- Khi  $C = 1$ :

+ Nếu  $JK = 11$ :

TH1: Giả sử ban đầu  $Q = 0$ ,  $\bar{Q} = 1$ :

$Q = 0$  nên  $R = 1$ ;  $\bar{Q} = 1$  nên  $S = 0$

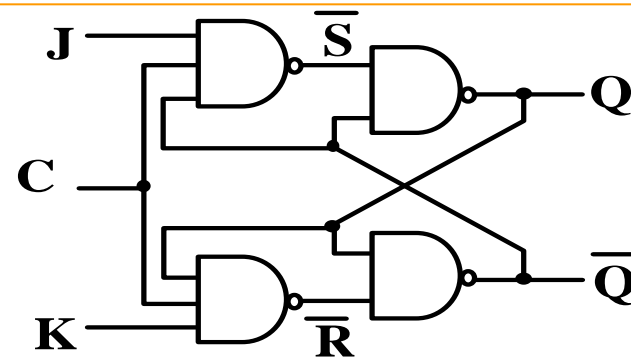
$S/R = 01$ ,  $SR = 10$ : Lập  $Q_k = 1$ ;  $\bar{Q}_k = 0$

TH2: Giả sử ban đầu  $Q = 1$ ,  $\bar{Q} = 0$ :

$\bar{Q} = 0$  nên  $S = 1$ ;  $Q = 1$  nên  $R = 0$

$S/R = 10$ ,  $SR = 01$ : Xóa  $Q_k = 0$ ;  $\bar{Q}_k = 1$

Như vậy,  $Q_k = \bar{Q}$ ,  $\bar{Q}_k = Q$



### 4.3.1.3. Trgger JK đồng bộ (5)

➤ Hoạt động (JK dùng cổng NOR):

- Khi  $C = 1$ :

$SR = 00$ :  $Q_k = Q$

- Khi  $C = 0$ :

+ Nếu  $JK = 00$ ,  $J/K/ = 11$ :

$SR = 00$ :  $Q_k = Q$

+ Nếu  $JK = 01$ ,  $J/K/ = 10$ :

TH1: Giả sử ban đầu  $Q = 0$ ,  $Q/ = 1$ :

$J/ = 1$  nên  $S = 0$ ;  $Q/ = 1$  nên  $R = 0$

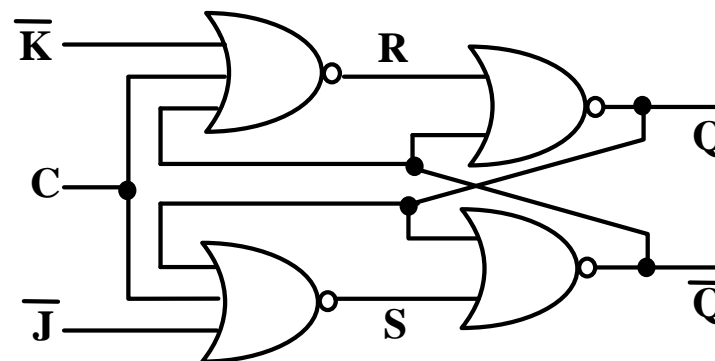
$SR = 00$ :  $Q_k = Q = 0$ ;  $Q_k/ = 1$

TH2: Giả sử ban đầu  $Q = 1$ ,  $Q/ = 0$ :

$J/ = 1$  nên  $S = 0$ ;  $Q/ = 0$  nên  $R = 1$

$SR = 01$ : Xóa  $Q_k = 0$ ;  $Q_k/ = 1$

Như vậy,  $Q_k = 0$ ,  $Q_k/ = 1$



### 4.3.1.3. Trùng JK đồng bộ (6)

➤ Hoạt động (JK dùng cổng NOR):

- Khi  $C = 0$ :

+ Nếu  $JK = 10$ ,  $J/K = 01$ :

TH1: Giả sử ban đầu  $Q = 0$ ,  $\overline{Q} = 1$ :

$K/ = 1$  nên  $R = 0$ ;  $Q = 0$  nên  $S = 1$

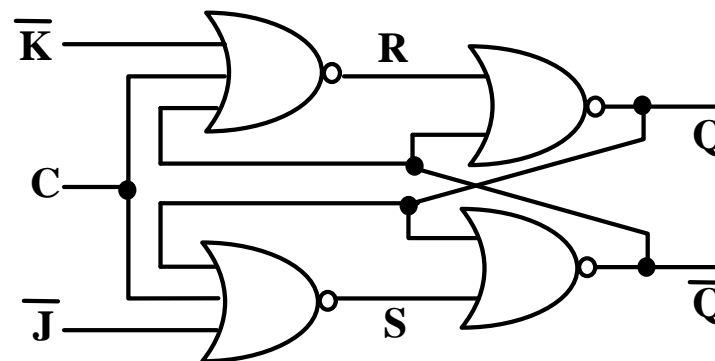
$SR = 10$ : Lập  $Q_k = 1$ ;  $Q_k/ = 0$

TH2: Giả sử ban đầu  $Q = 1$ ,  $\overline{Q} = 0$ :

$K/ = 1$  nên  $R = 0$ ;  $Q = 1$  nên  $S = 0$

$SR = 00$ :  $Q_k = Q = 1$ ;  $Q_k/ = 0$

Như vậy,  $Q_k = 1$ ,  $Q_k/ = 0$



### 4.3.1.3. Trùng JK đồng bộ (7)

➤ Hoạt động (JK dùng cổng NOR):

- Khi  $C = 0$ :

+ Nếu  $JK = 11$ ,  $J/K/ = 00$ :

TH1: Giả sử ban đầu  $Q = 0$ ,  $Q/ = 1$ :

$Q/ = 1$  nên  $R = 0$ ;  $Q = 0$  nên  $S = 1$

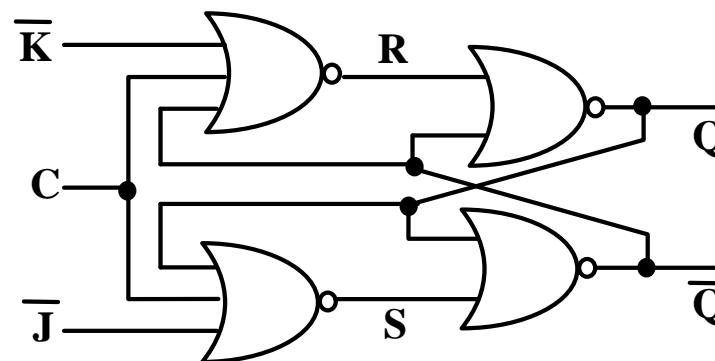
$SR = 10$ : Lập  $Q_k = 1$ ;  $Q_k/ = 0$

TH2: Giả sử ban đầu  $Q = 1$ ,  $Q/ = 0$ :

$Q/ = 0$  nên  $R = 1$ ;  $Q = 1$  nên  $S = 0$

$SR = 01$ : Xóa  $Q_k = 0$ ;  $Q_k/ = 1$

Như vậy,  $Q_k = Q/$ ,  $Q_k/ = Q$



## 4.3.1.3. Trigo JK đồng bộ (8)

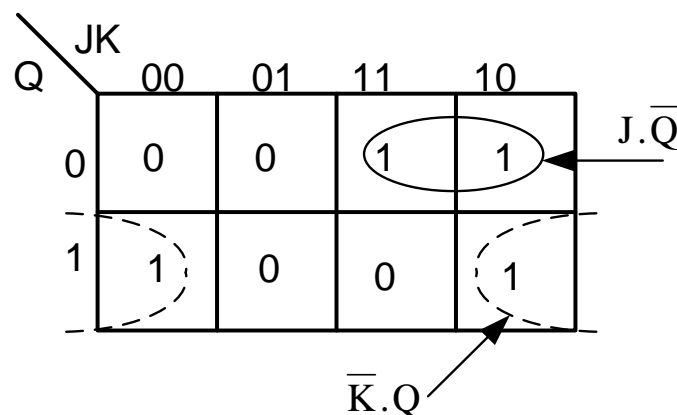
➤ Bảng trạng thái:

C	J	K	$Q^k$
0	X	X	Q
1	0	0	Q
	0	1	0
	1	0	1
	1	1	Q/

C	J	K	$Q^k$
1	X	X	Q
0	0	0	Q
	0	1	0
	1	0	1
	1	1	Q/

➤ Phương trình đặc trưng:

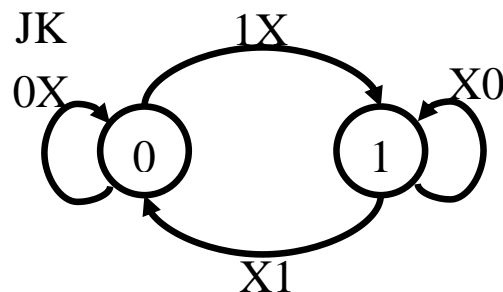
$$Q^k = J.\bar{Q} + \bar{K}.Q$$





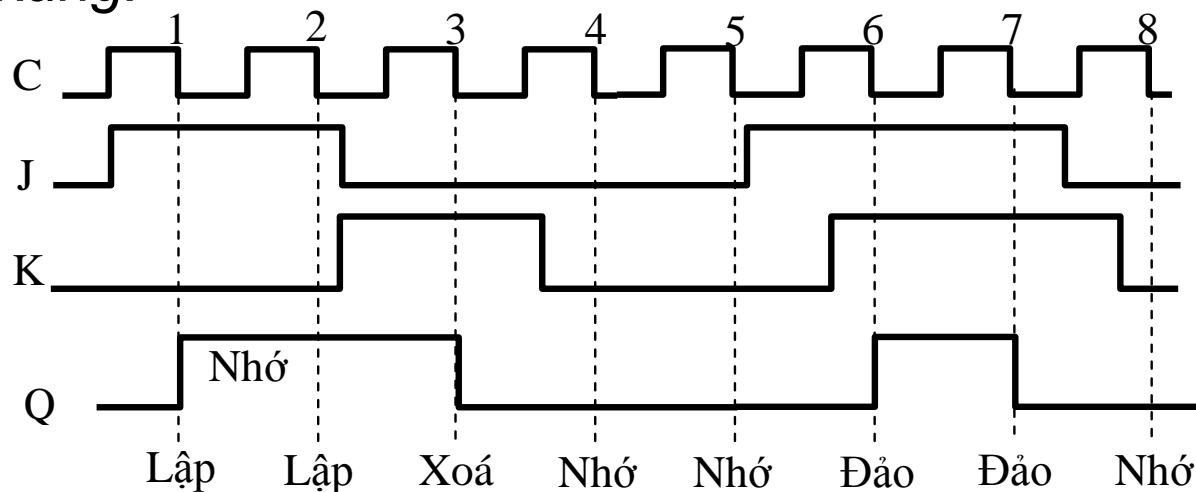
## 4.3.1.3. Trạng thái đồng bộ JK (9)

➤ Đồ hình trạng thái:



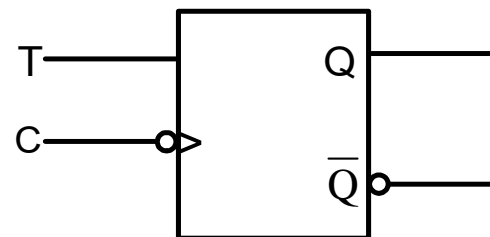
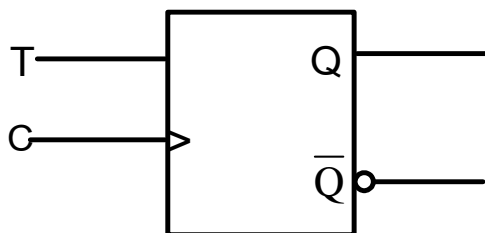
C	J	K	Q <sup>k</sup>
0	X	X	Q
1	0	0	Q
	0	1	0
	1	0	1
	1	1	Q/

➤ Giải đồ xung:

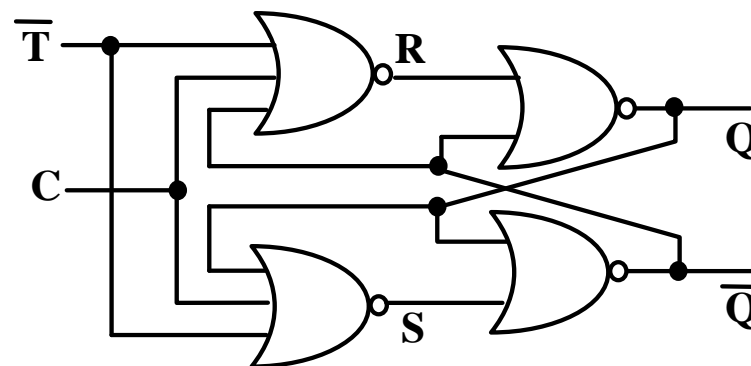
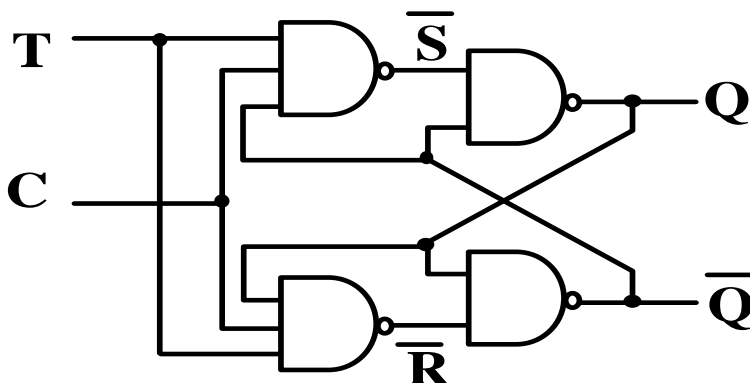


#### 4.3.1.4. Trùng T đồng bộ (1)

➤ Sơ đồ khối:



➤ Mạch điện:



### 4.3.1.4. Trùng T đồng bộ (2)

➤ Hoạt động (T dùng cổng NAND):

- Khi  $C = 0$ :

$S/R = 11$ ,  $SR = 00$ :  $Q_k = Q$

- Khi  $C = 1$ :

+ Nếu  $T = 0$ ,  $JK = 00$ :  $Q_k = Q$

+ Nếu  $T = 1$ ,  $JK = 11$ :  $Q_k = Q/$

➤ Hoạt động (T dùng cổng NOR):

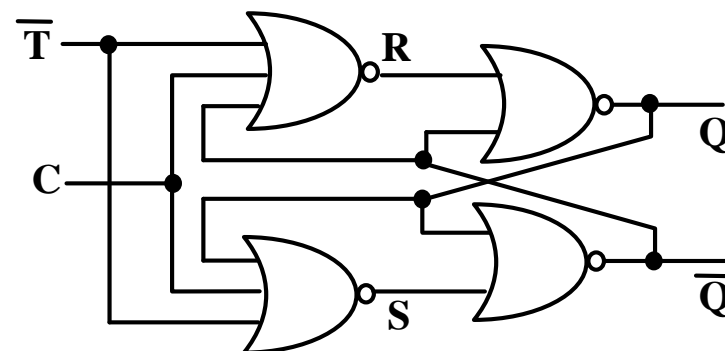
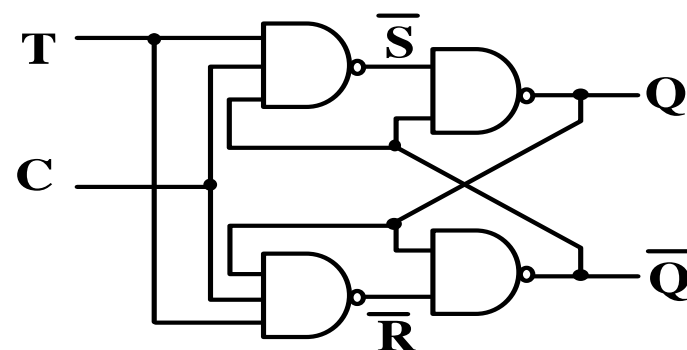
- Khi  $C = 1$ :

$SR = 00$ :  $Q_k = Q$

- Khi  $C = 0$ :

+ Nếu  $T = 0$ ,  $JK = 00$ :  $Q_k = Q$

+ Nếu  $T = 1$ ,  $JK = 11$ :  $Q_k = Q/$



4.3.1.4. Trigo  $T$  đồng bộ (3)

➤ Bảng trạng thái:

C	T	$Q^k$
0	x	Q
1	0	Q
1	1	Q/

C	T	$Q^k$
1	x	Q
0	0	Q
0	1	Q/

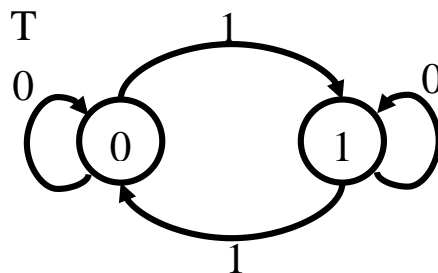
➤ Phương trình đặc trưng:

$$Q^k = T \oplus Q$$

		T	
		0	1
Q	0	0	1
	1	1	0

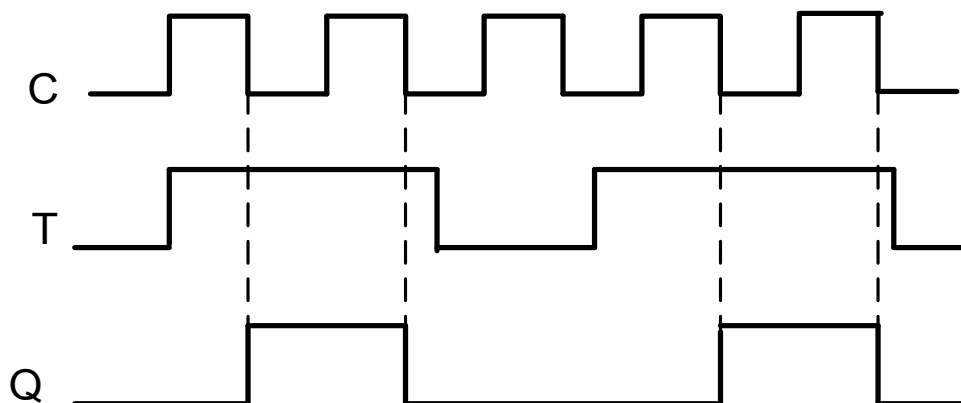
## 4.3.1.4. Triger T đồng bộ (4)

➤ Đồ hình trạng thái:



T \ Q	0	1
0	0	1
1	1	0

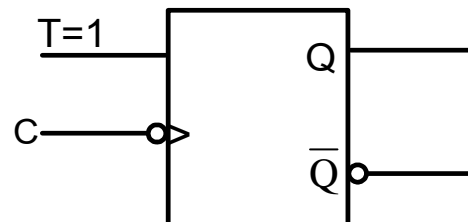
➤ Giải đồ xung:



### 4.3.1.5. Trigo $T'$

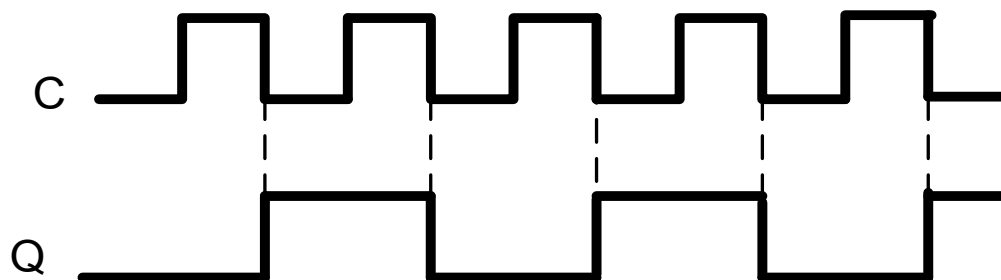
➤ Sơ đồ khối:

Là Trigo  $T$  với  $T = 1$



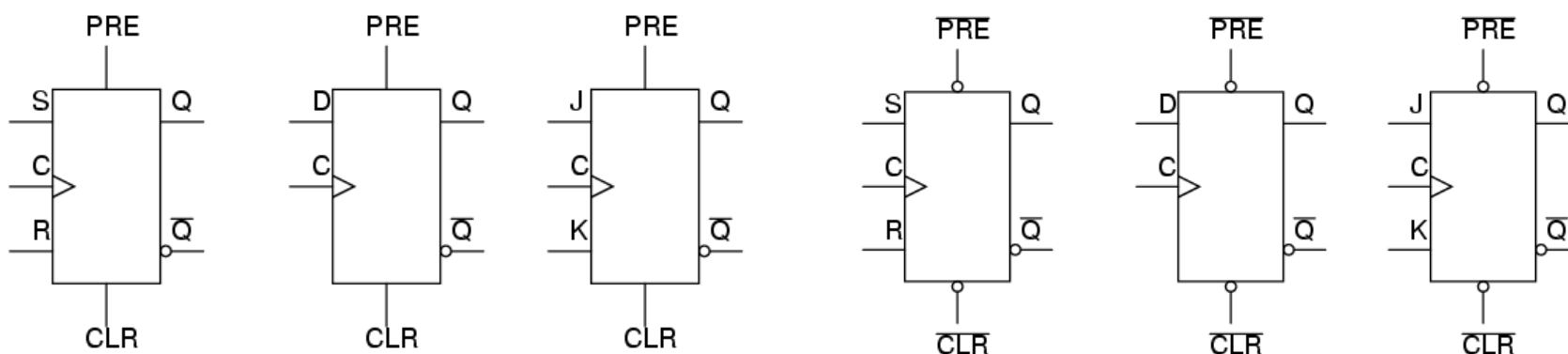
➤ Phương trình đặc trưng:  $Q^k = \bar{Q}$

➤ Giải đồ xung:

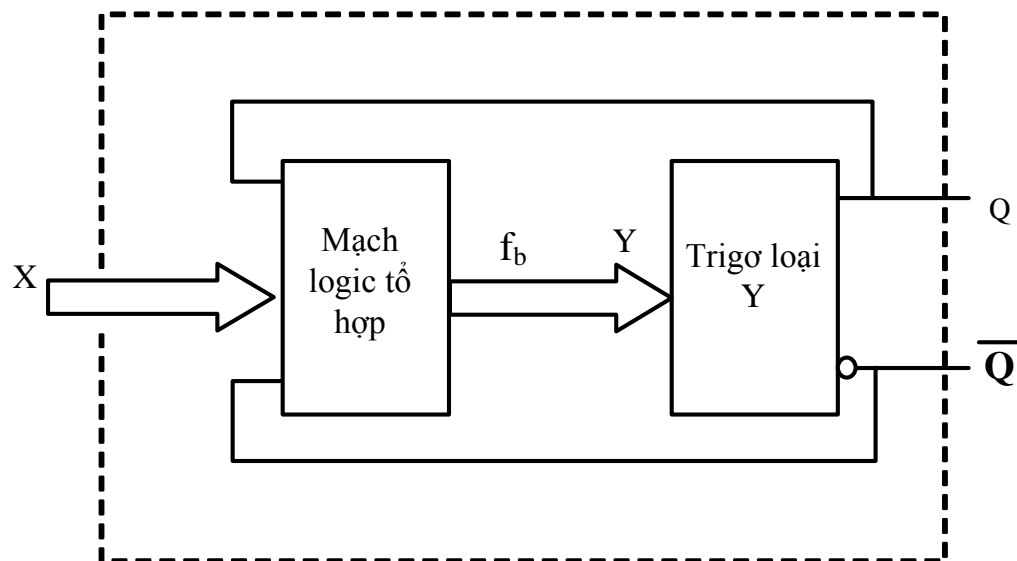
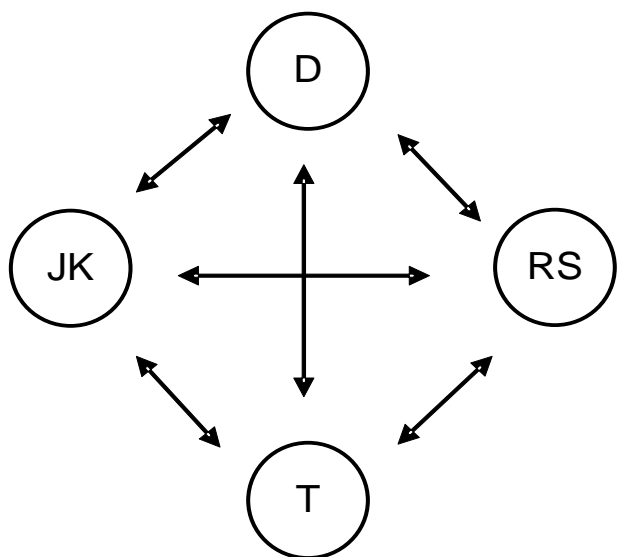


### 4.3.2. Đầu vào không đồng bộ của Trigo

- Các đầu vào D, S, R, J, K, T là những đầu vào đồng bộ vì chúng có ảnh hưởng lên các đầu ra khi có tác động của xung Clock.
- Trong thực tế các trigơ còn có thêm 2 đầu vào không đồng bộ, các lỗi này tác động trực tiếp lên các đầu ra mà không phụ thuộc vào xung Clock.
- Các đầu vào này thường được ký hiệu là: PRE (lập) và CLR (R - xóa) hoặc  $\overline{\text{PRE}}$  và  $\overline{\text{CLR}}$  ( $\overline{\text{R}}$ )



### 4.3.3. Chuyển đổi giữa các loại Trơ (1) – Sơ đồ



- Biến một trơ đã cho thành trơ cần tìm khác loại.
- Bài toán chuyển đổi từ trơ Y cho trước thành trơ X cần tìm là bài toán xây dựng các mạch logic tổ hợp, có:
  - đầu vào: X, Q
  - đầu ra:  $Y = f(X, Q)$



## 4.3.3. Chuyển đổi giữa các loại Trigon (2) – Phương pháp

➤ Dùng bảng hàm kích:

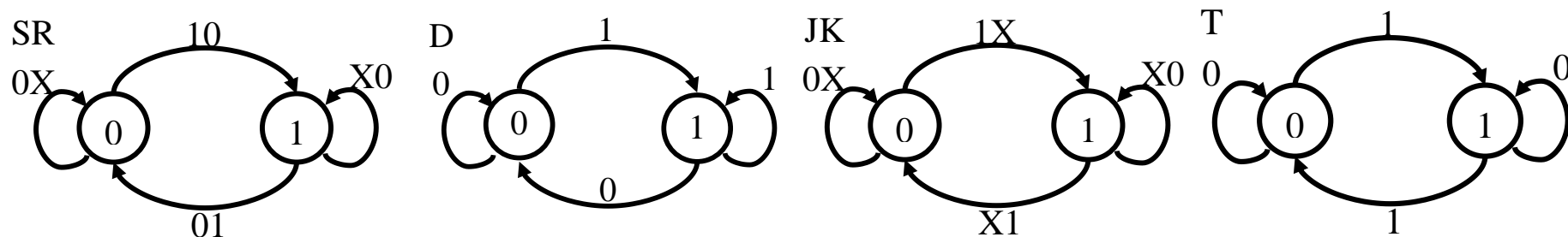
Bảng hàm kích: bảng trạng thái mô tả sự phụ thuộc của các đầu vào kích thích với đầu ra Q.

B1: Xác định hệ hàm theo bảng hàm kích.

B2: Tối thiểu hoá các hàm Y.

B3: Vẽ sơ đồ tương ứng.

## 4.3.3. Chuyển đổi giữa các loại Trigo (3) – Bảng hàm kích



Q	Q <sup>k</sup>	S	R	J	K	D	T
0	0	0	X	0	X	0	0
0	1	1	0	1	X	1	1
1	0	0	1	X	1	0	1
1	1	X	0	X	0	1	0

## 4.3.3. Chuyển đổi giữa các loại Trilog (4) – Ví dụ

**VD:** Chuyển đổi từ trilog RS sang JK.

Để chuyển đổi cần tìm mối quan hệ:

$$S = f_1(Q, J, K) ;$$

$$R = f_2(Q, J, K)$$

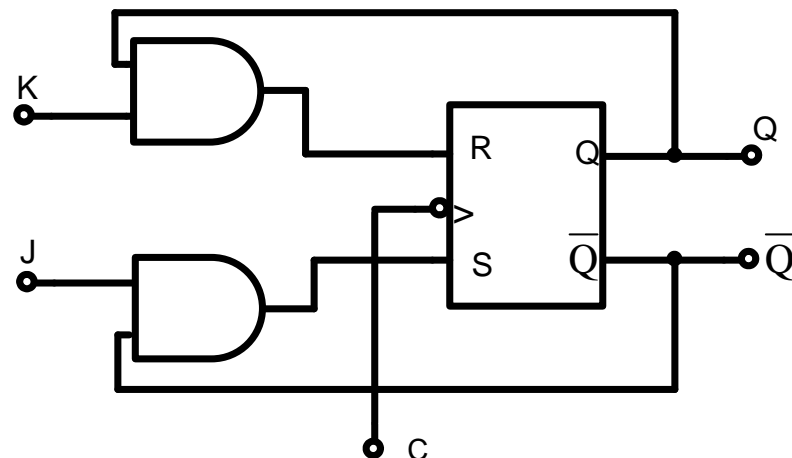
## 4.3.3. Chuyển đổi giữa các loại Trigo (5) – Ví dụ (tiếp)

Dùng bảng hàm kích:

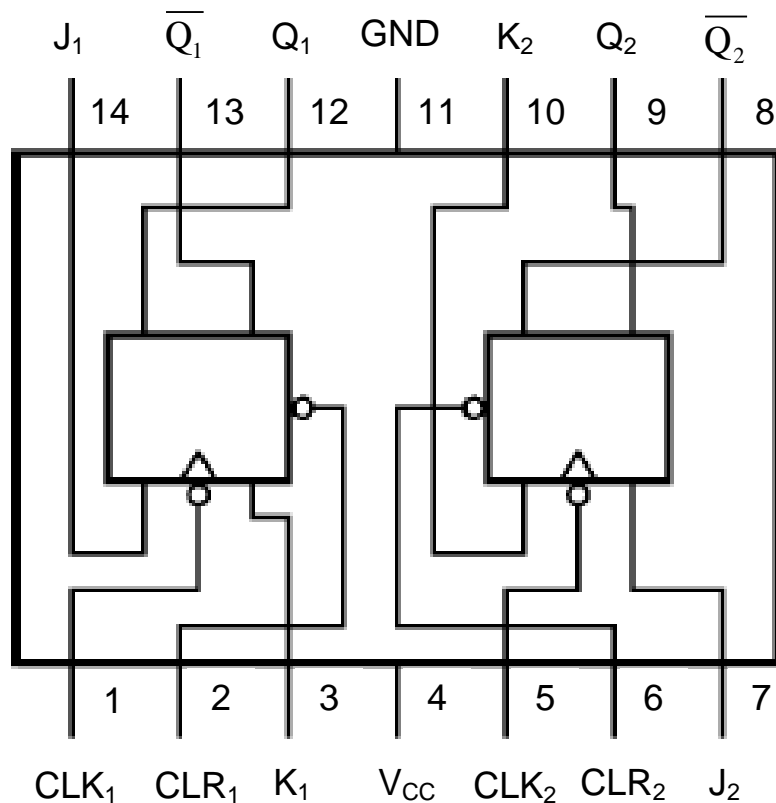
S	JK					
		00	01	11	10	
Q	0	0	0	1	1	$S = J \cdot \bar{Q}$
	1	x	0	0	x	

R	JK					
		00	01	11	10	
Q	0	x	x	0	0	$R = K \cdot Q$
	1	0	1	1	0	

Q	Q <sup>k</sup>	S	R	J	K	D	T
0	0	0	X	0	X	0	0
0	1	1	0	1	X	1	1
1	0	0	1	X	1	0	1
1	1	X	0	X	0	1	0



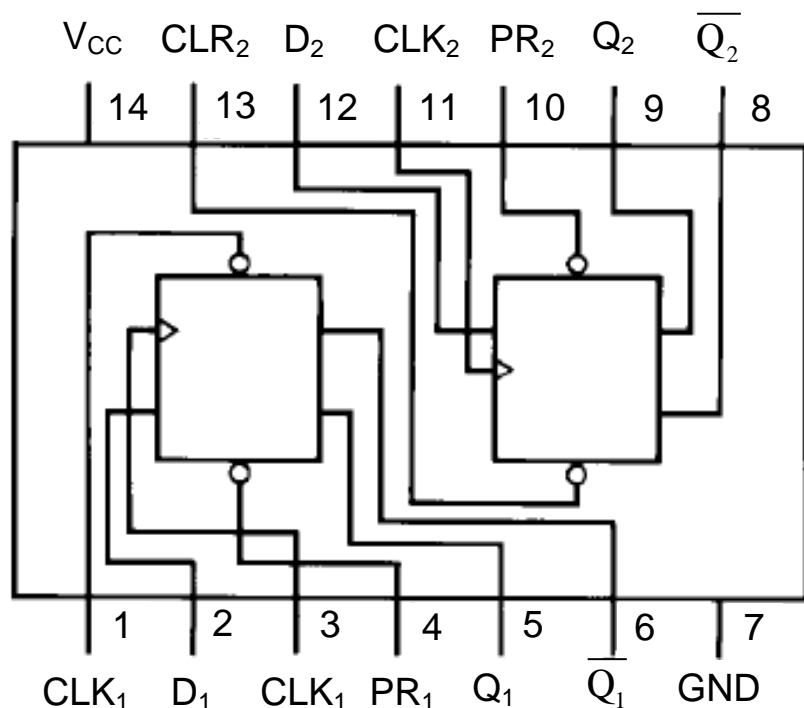
## 4.3.4. Giới thiệu một số loại IC trigơ thông dụng (1)



7473

CLR	CLK	J	K	Q <sup>k</sup>	Q <sup>k/</sup>
L	X	X	X	L	H
H	↓	L	L	Q	Q/
H	↓	L	H	L	H
H	↓	H	L	H	L
H	↓	H	H	Q/	Q

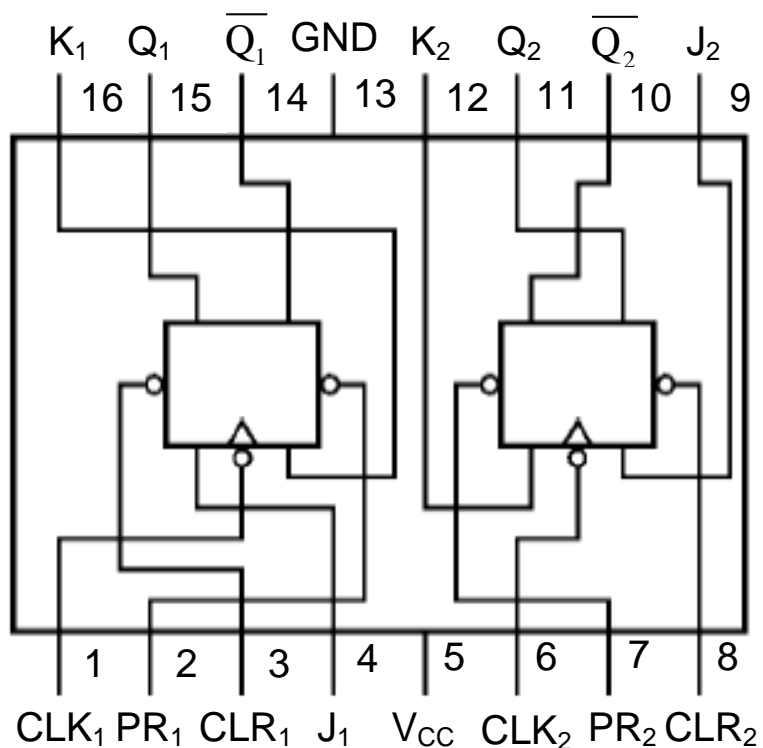
## 4.3.4. Giới thiệu một số loại IC trigơ thông dụng (2)



7474

PR	CLR	CLK	D	$Q^k$	$Q^{k/}$
L	L	X	X	H	H
L	H	X	X	H	L
H	L	X	X	L	H
H	H	↑	L	L	H
H	H	↑	H	H	L

## 4.3.4. Giới thiệu một số loại IC trigơ thông dụng (3)



7476

PR	CLR	CLK	J	K	$Q^k$	$Q^{k/}$
L	L	X	X	X	H	H
L	H	X	X	X	H	L
H	L	X	X	X	L	H
H	H	↓	L	L	Q	Q/
H	H	↓	L	H	L	H
H	H	↓	H	L	H	L
H	H	↓	H	H	Q/	Q

## Chương 4 – MẠCH LOGIC TUẦN TỰ

4.1. Khái niệm chung

4.2. Phương pháp mô tả mạch tuần tự

4.3. Phần tử nhớ của mạch tuần tự

**4.4. Mạch tuần tự thông dụng**



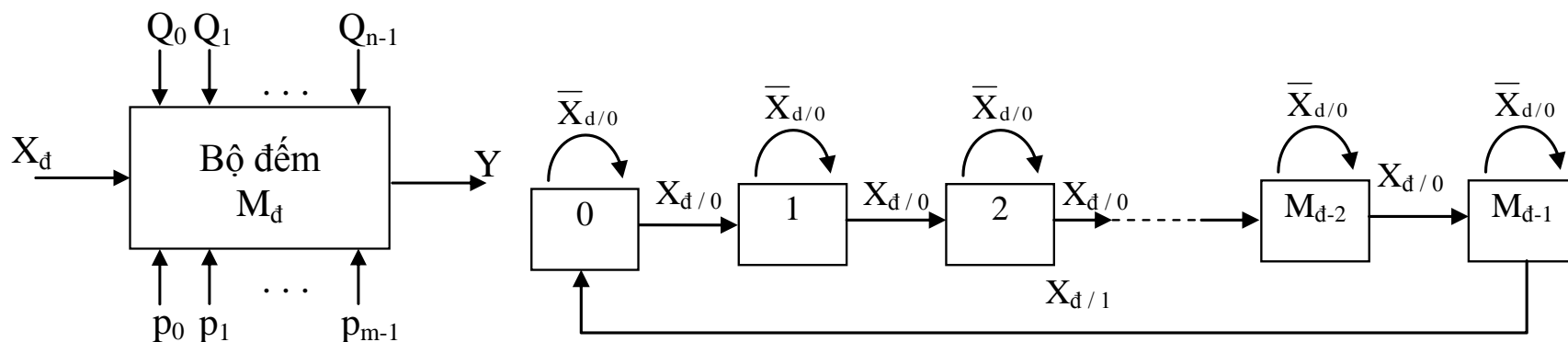
## 4.4. Mạch tuần tự thông dụng

### 4.4.1. Bộ đếm

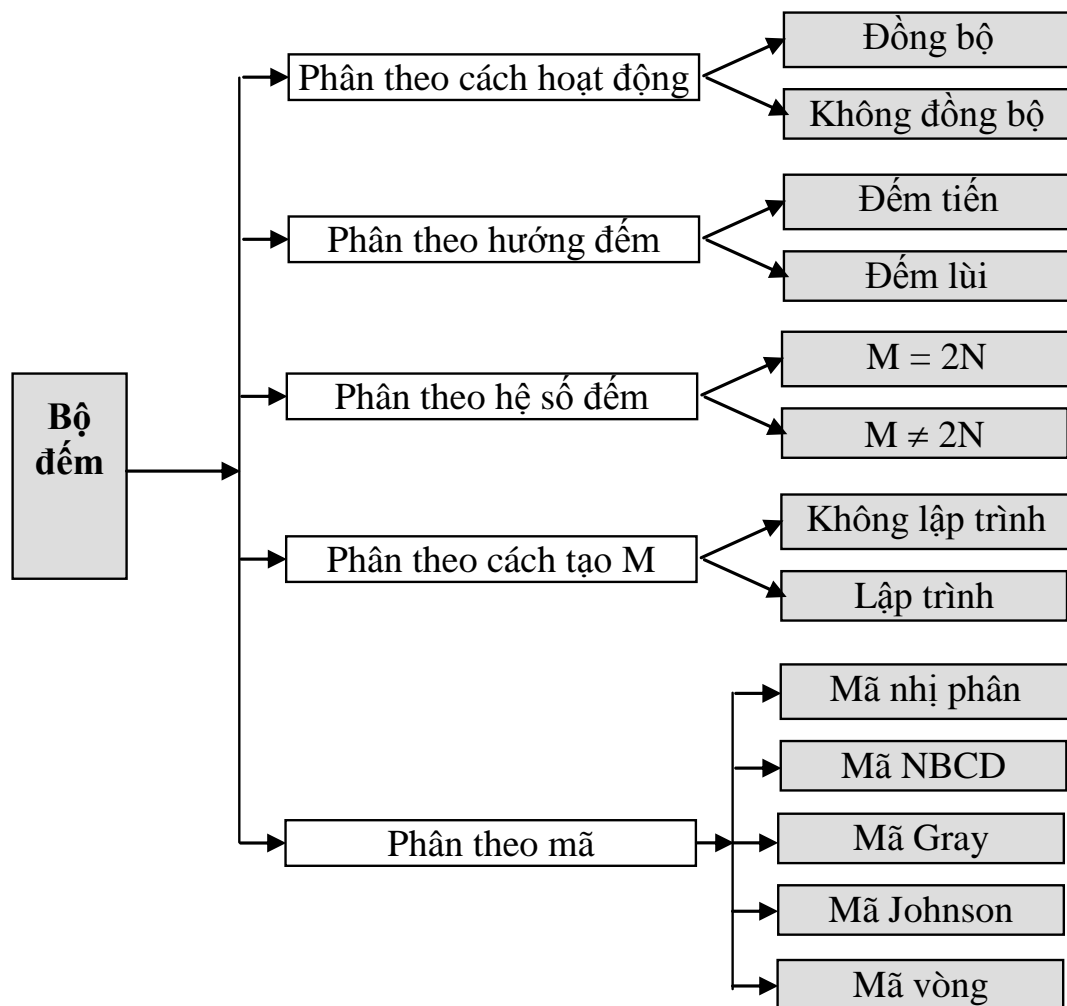
### 4.4.2. Bộ ghi dịch

### 4.4.1.1. Định nghĩa và phân loại (1)

- 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.

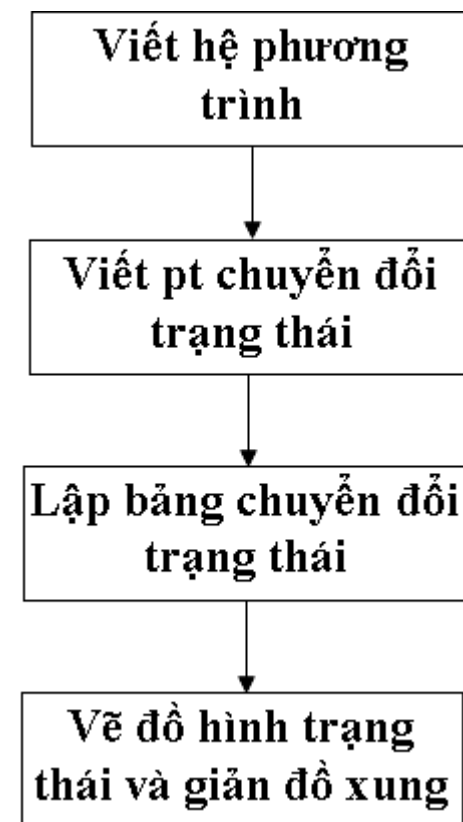


### 4.4.1.1. Định nghĩa và phân loại (2)

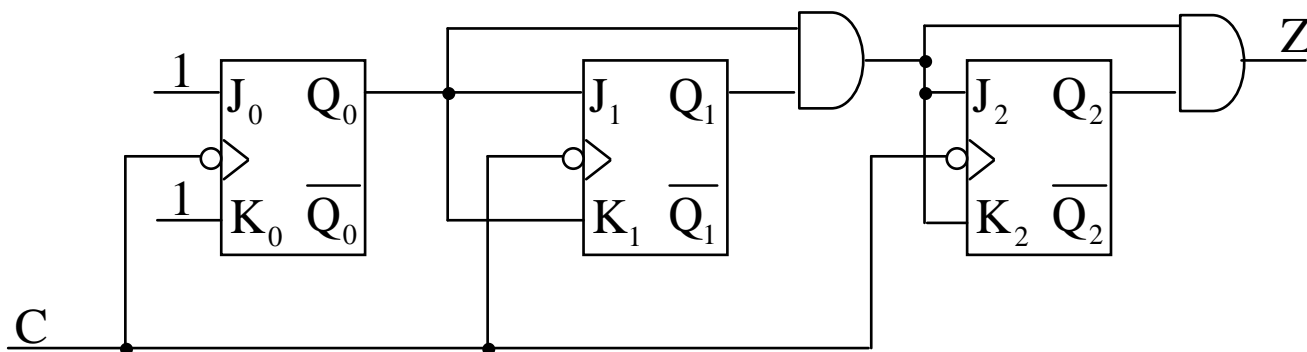


## 4.4.1.2. Phân tích bộ đếm (1) – Phương pháp chung

- B1: Viết phương trình định thời, phương trình hàm ra, phương trình hàm kích
- B2: Thay phương trình hàm kích vào phương trình đặc trưng để tìm phương trình chuyển đổi trạng thái
- B3: Thay các tổ hợp có thể có của trạng thái hiện tại và tín hiệu vào vào pt chuyển đổi trạng thái để tìm trạng thái kế tiếp và tín hiệu ra
- B4: Vẽ đồ hình trạng thái, giản đồ xung để tìm chức năng của bộ đếm



### 4.4.1.2. Phân tích bộ đếm (2) – Ví dụ 1:



\* Nhận xét: - Bộ đếm sử dụng 3 trigơ

- Clock C điều khiển cả 3 trigơ  $\Leftrightarrow$  bộ đếm đồng bộ

### 4.4.1.2. Phân tích bộ đếm (3) – Ví dụ 1:

B1: Phương trình định thời:  $C_0 = C_1 = C_2 = \downarrow C$

B2: Phương trình hàm ra:  $Z = Q_0 \cdot Q_1 \cdot Q_2$

B3: Phương trình hàm kích:  $J_0 = K_0 = 1$

$$J_1 = K_1 = Q_0$$

$$J_2 = K_2 = Q_0 \cdot Q_1$$

B4: Phương trình chuyển đổi trạng thái:  $Q^k = J \cdot \overline{Q} + \overline{K} \cdot Q$

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

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

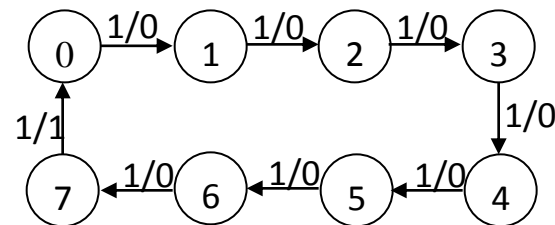
$$Q_2^k = J_2 \cdot \overline{Q_2} + \overline{K_2} \cdot Q_2 = Q_0 \cdot Q_1 \cdot \overline{Q_2} + \overline{Q_0 \cdot Q_1} \cdot Q_2 = Q_2 \oplus Q_1 \cdot Q_0$$

## 4.4.1.2. Phân tích bộ đếm (4) – Ví dụ 1:

B5: Bảng chuyển đổi trạng thái:

C	$Q_2$	$Q_1$	$Q_0$	$Q_2^k$	$Q_1^k$	$Q_0^k$	Z
1	0	0	0	0	0	1	0
2	0	0	1	0	1	0	0
3	0	1	0	0	1	1	0
4	0	1	1	1	0	0	0
5	1	0	0	1	0	1	0
6	1	0	1	1	1	0	0
7	1	1	0	1	1	1	0
8	1	1	1	0	0	0	1

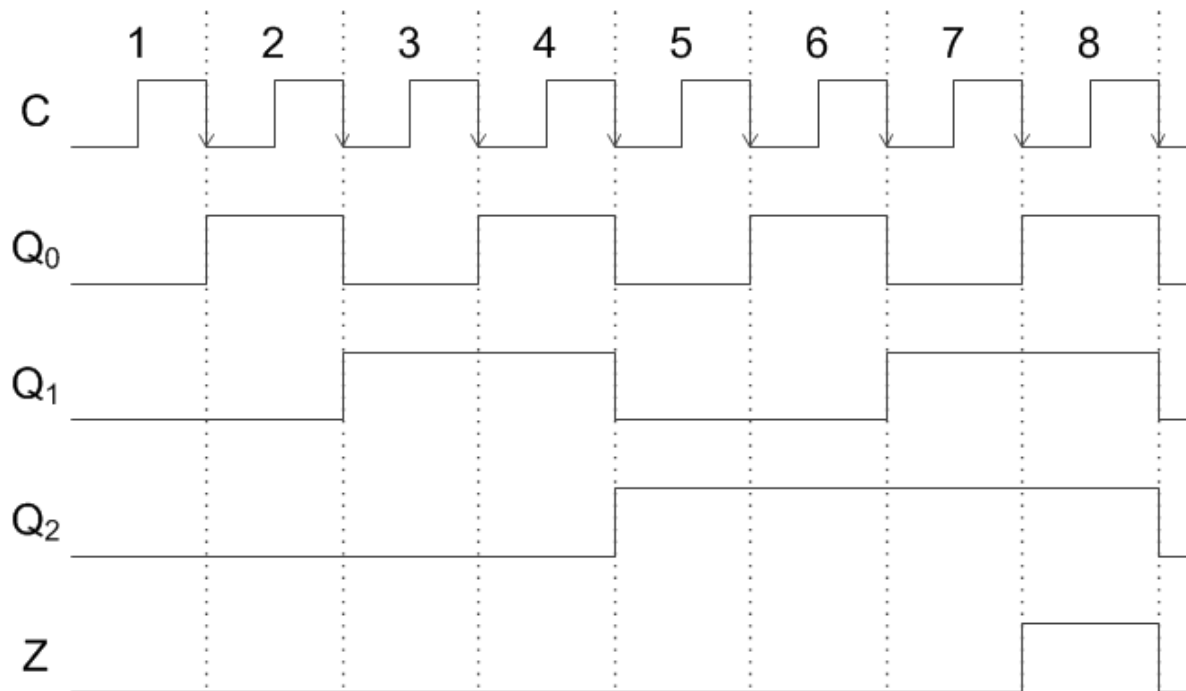
B6: Đồ hình trạng thái:



$$Z = Q_0 Q_1 Q_2, \quad Q_0^k = \overline{Q_0}, \quad Q_1^k = Q_0 \oplus Q_1, \quad Q_2^k = Q_2 \oplus Q_1 \cdot Q_0$$

### 4.4.1.2. Phân tích bộ đếm (5) – Ví dụ 1:

B7: Giải đồ xung:



Bộ đếm là bộ chia tần

- Qua trigơ thứ nhất:  $f_{Q_0} = \frac{f_C}{2}$

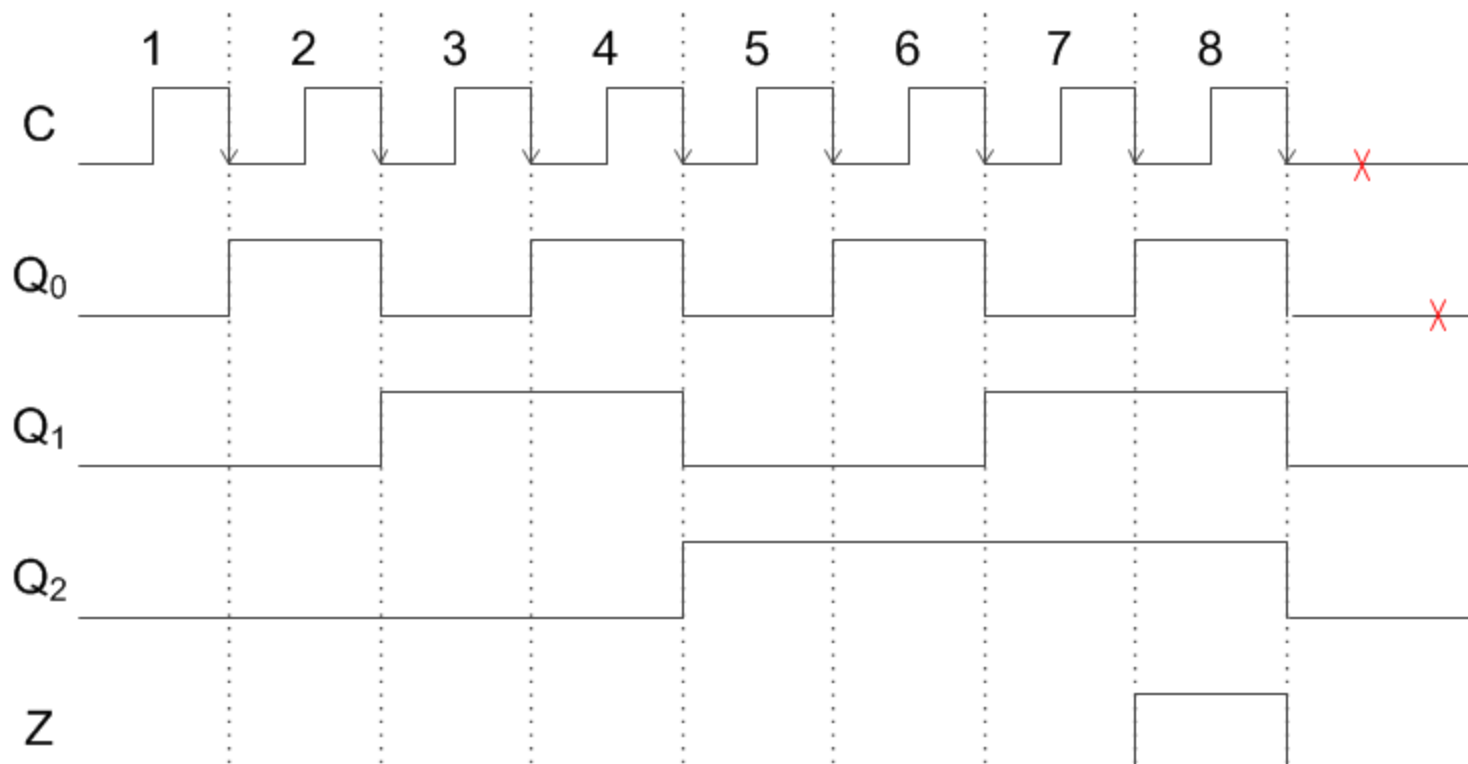
- Qua trigơ thứ hai:  $f_{Q_1} = \frac{f_{Q_0}}{2} = \frac{f_C}{4}$

- Qua trigơ thứ ba:  $f_{Q_2} = \frac{f_{Q_1}}{2} = \frac{f_{Q_0}}{4} = \frac{f_C}{8}$



### 4.4.1.2. Phân tích bộ đếm (5) – Ví dụ 1:

B7: Giải đồ xung: **Vẽ sai**

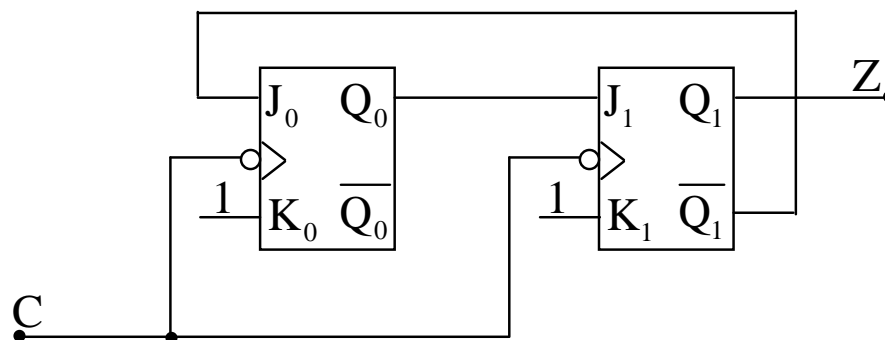


### 4.4.1.2. Phân tích bộ đếm (5) – Ví dụ 2:

$$B1: C_0 = C_1 = \downarrow C$$

$$Z = Q_1$$

$$\begin{cases} J_0 = \overline{Q_1} \\ K_0 = 1 \end{cases}; \begin{cases} J_1 = Q_0 \\ K_1 = 1 \end{cases}$$



$$B2: Q_0^k = J_0 \cdot \overline{Q_0} + \overline{K_0} \cdot Q_0 = \overline{Q_1} \cdot \overline{Q_0}$$

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

## 4.4.1.2. Phân tích bộ đếm (6) – Ví dụ 2:

B3:

C	$Q_1$	$Q_0$	$Q_1^k$	$Q_0^k$	Z
0	0	0	0	1	0
1	0	1	1	0	0
2	1	0	0	0	1
	1	1	0	0	1

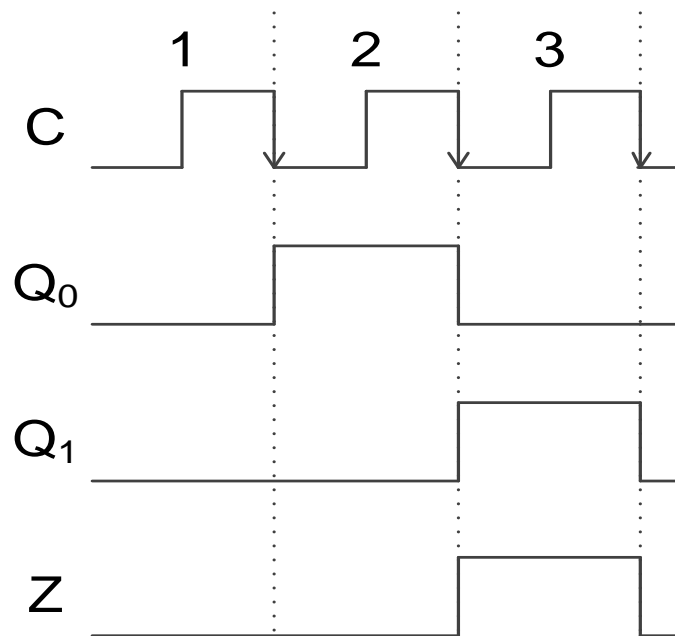
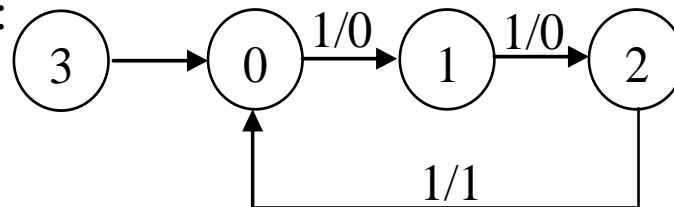
$$Q_0^k = J_0 \cdot \overline{Q_0} + \overline{K_0} \cdot Q_0 = \overline{Q_1} \cdot \overline{Q_0}$$

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

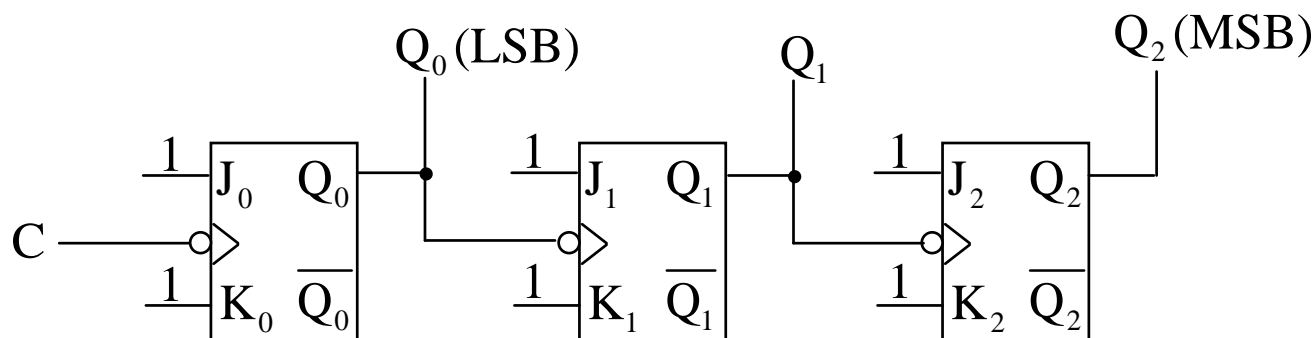
NX:

- Cứ 3 xung vào thì đầu ra có một xung.
- Đây là bộ đếm mod 3 tự khởi động, nghĩa là các trạng thái dư sau một số xung nhịp lại quay trở lại vòng đếm.

B4:



## 4.4.1.2. Phân tích bộ đếm (7) – Ví dụ 3:



$$B1: C_0 = \downarrow C, C_1 = \downarrow Q_0, C_2 = \downarrow Q_1$$

$$B2: J_0 = K_0 = 1, J_1 = K_1 = 1, J_2 = K_2 = 1$$

$$B3: Q_0^k = J_0 \cdot \overline{Q_0} + \overline{K_0} \cdot Q_0 = \overline{Q_0} \quad (\text{điều kiện: } \downarrow C)$$

$$Q_1^k = J_1 \cdot \overline{Q_1} + \overline{K_1} \cdot Q_1 = \overline{Q_1} \quad (\text{điều kiện: } \downarrow Q_0)$$

$$Q_2^k = J_2 \cdot \overline{Q_2} + \overline{K_2} \cdot Q_2 = \overline{Q_2} \quad (\text{điều kiện: } \downarrow Q_1)$$

## 4.4.1.2. Phân tích bộ đếm (8) – Ví dụ 3:

B4:

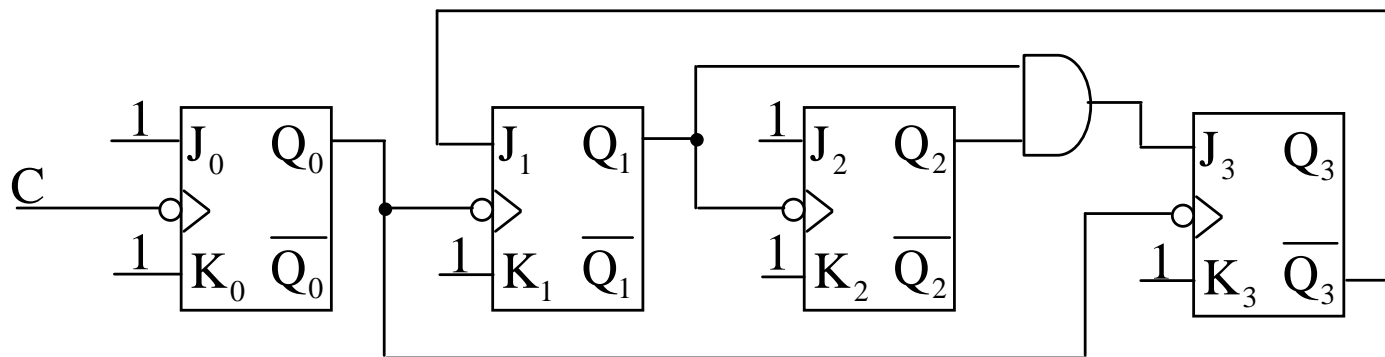
C	$Q_2$	$Q_1$	$Q_0$	$Q_2^k$	$Q_1^k$	$Q_0^k$	Điều kiện định thời
1	0	0	0	0	0	1	không có $\downarrow Q_0, \downarrow Q_1$
2	0	0	1	0	1	0	có $\downarrow Q_0$
3	0	1	0	0	1	1	không có $\downarrow Q_0, \downarrow Q_1$
4	0	1	1	1	0	0	có $\downarrow Q_0, \downarrow Q_1$
5	1	0	0	1	0	1	không có $\downarrow Q_0, \downarrow Q_1$
6	1	0	1	1	1	0	có $\downarrow Q_0$
7	1	1	0	1	1	1	không có $\downarrow Q_0, \downarrow Q_1$
8	1	1	1	0	0	0	có $\downarrow Q_0, \downarrow Q_1$

NX: Bộ đếm không đồng bộ có mạch điện đơn giản hơn bộ đếm đồng bộ, nhưng tốc độ làm việc thấp hơn (nghĩa là tần số làm việc nhỏ hơn)

Đặc điểm: - Chỉ dùng trigơ T hoặc JK ( $T=1$  hoặc  $J=K=1$ )

- Lỗi ra của trigơ tầng trước là lỗi vào C của trigơ tầng sau
- Tín hiệu vào đếm được đưa tới lỗi vào C của tầng thấp nhất (LSB)

## 4.4.1.2. Phân tích bộ đếm (9) – Ví dụ 4:



$$B1: C_0 = \downarrow C, C_1 = \downarrow Q_0, C_2 = \downarrow Q_1, C_3 = \downarrow Q_0$$

$$B2: J_0 = K_0 = 1 \quad J_1 = \overline{Q_3}, K_1 = 1$$

$$J_2 = K_2 = 1 \quad J_3 = Q_2 \cdot Q_1, K_3 = 1$$

$$B3: Q_0^k = J_0 \cdot \overline{Q_0} + \overline{K_0} \cdot Q_0 = \overline{Q_0} \quad (\downarrow C)$$

$$Q_1^k = J_1 \cdot \overline{Q_1} + \overline{K_1} \cdot Q_1 = \overline{Q_3} \cdot \overline{Q_1} \quad (\downarrow Q_0)$$

$$Q_2^k = J_2 \cdot \overline{Q_2} + \overline{K_2} \cdot Q_2 = \overline{Q_2} \quad (\downarrow Q_1)$$

$$Q_3^k = J_3 \cdot \overline{Q_3} + \overline{K_3} \cdot Q_3 = \overline{Q_3} \cdot Q_2 \cdot Q_1 \quad (\downarrow Q_0)$$

## 4.4.1.2. Phân tích bộ đếm (10) – Ví dụ 4:

B4:

C	$Q_3$	$Q_2$	$Q_1$	$Q_0$	$Q_3^k$	$Q_2^k$	$Q_1^k$	$Q_0^k$	Điều kiện định thời
1	0	0	0	0	0	0	0	1	$\downarrow C$
2	0	0	0	1	0	0	1	0	$\downarrow C, \downarrow Q_0$
3	0	0	1	0	0	0	1	1	$\downarrow C$
4	0	0	1	1	0	1	0	0	$\downarrow C, \downarrow Q_0, \downarrow Q_1$
5	0	1	0	0	0	1	0	1	$\downarrow C$
6	0	1	0	1	0	1	1	0	$\downarrow C, \downarrow Q_0$
7	0	1	1	0	0	1	1	1	$\downarrow C$
8	0	1	1	1	1	0	0	0	$\downarrow C, \downarrow Q_0, \downarrow Q_1$
9	1	0	0	0	1	0	0	1	$\downarrow C$
10	1	0	0	1	0	0	0	0	$\downarrow C, \downarrow Q_0$
11	1	0	1	0	1	0	1	1	$\downarrow C$
12	1	0	1	1	0	1	0	0	$\downarrow C, \downarrow Q_0$
13	1	1	0	0	1	1	0	1	$\downarrow C$
14	1	1	0	1	0	1	0	0	$\downarrow C, \downarrow Q_0$
15	1	1	1	0	1	1	1	1	$\downarrow C$
16	1	1	1	1	0	0	0	0	$\downarrow C, \downarrow Q_0, \downarrow Q_1$

$$B3: Q_0^k = \overline{Q_0} \quad (\downarrow C)$$

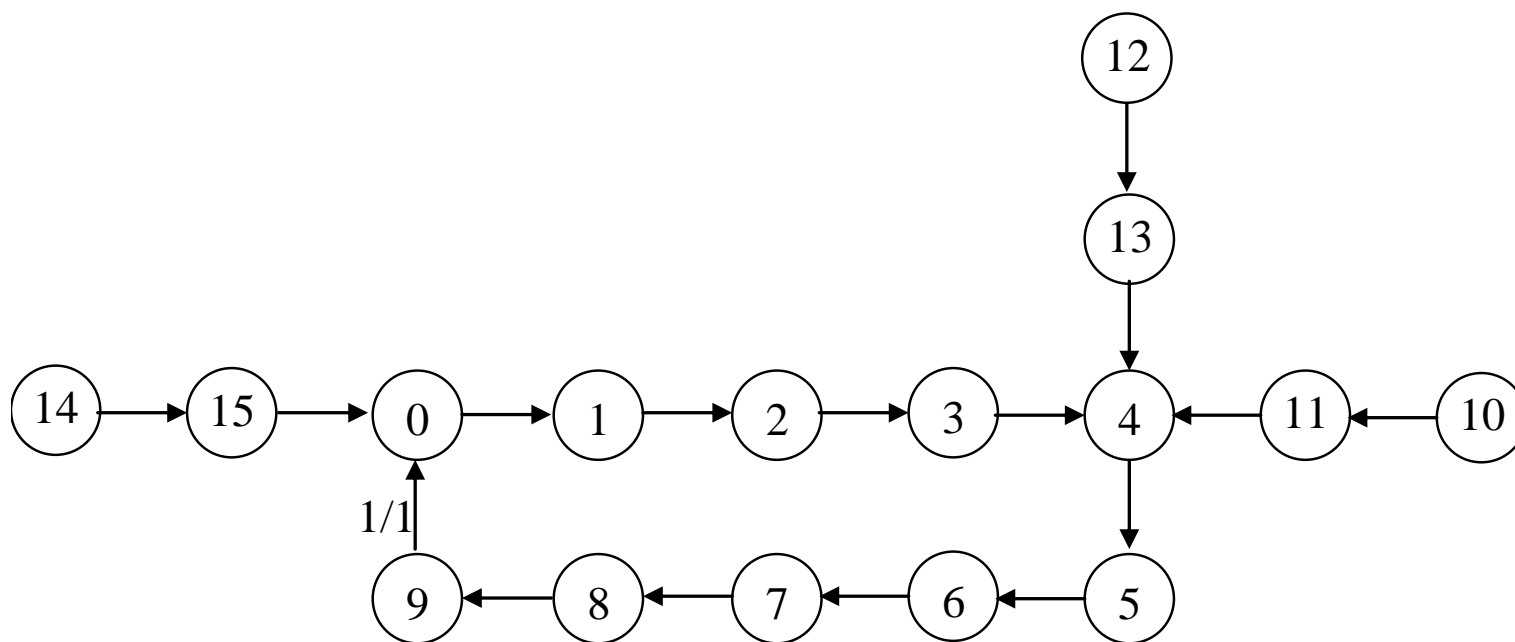
$$Q_1^k = \overline{Q_3} \cdot \overline{Q_1} \quad (\downarrow Q_0)$$

$$Q_2^k = \overline{Q_2} \quad (\downarrow Q_1)$$

$$Q_3^k = \overline{Q_3} \cdot Q_2 \cdot Q_1 \quad (\downarrow Q_0)$$

### 4.4.1.2. Phân tích bộ đếm (11) – Ví dụ 4:

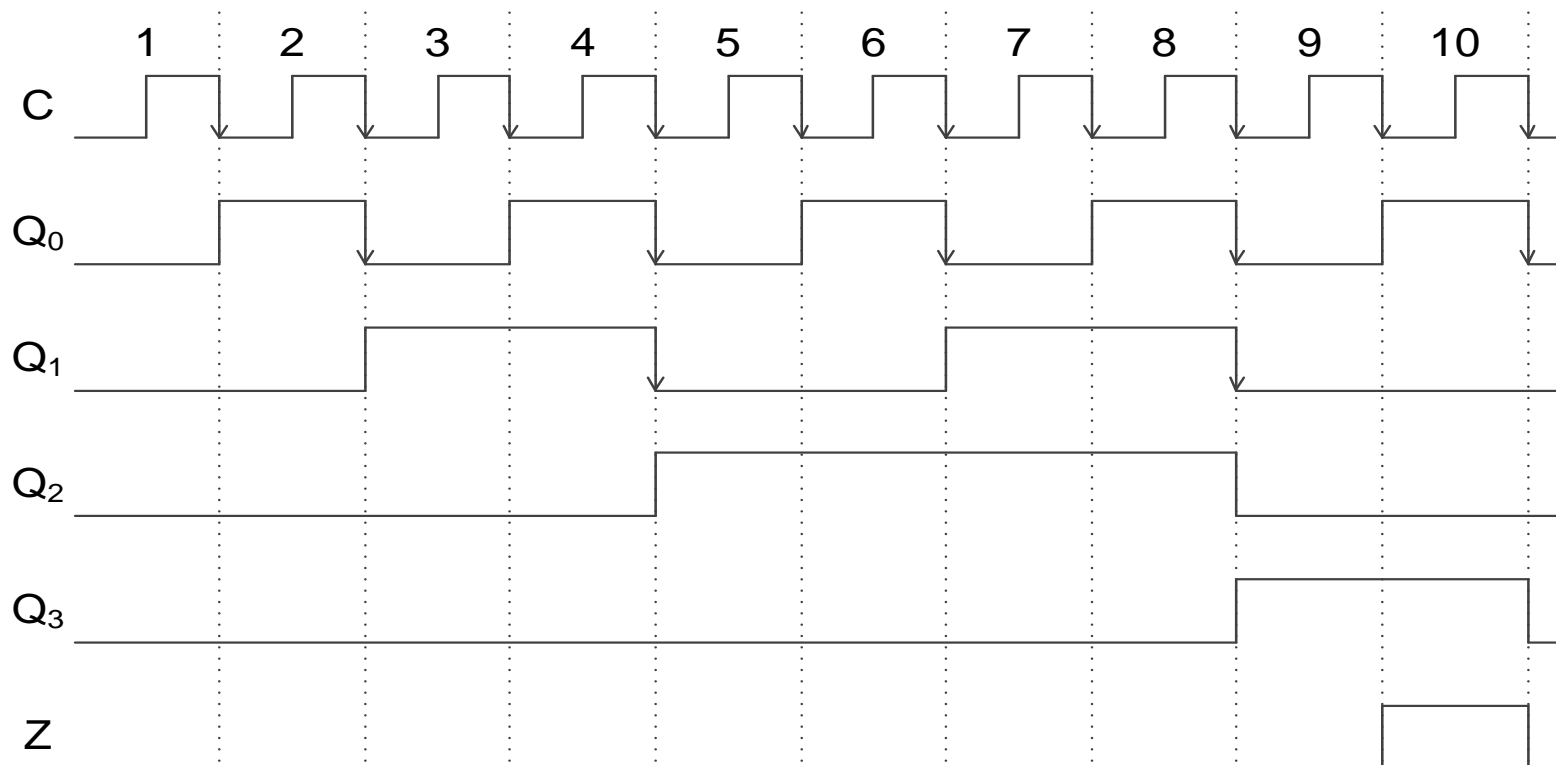
B5: Đồ hình trạng thái:





### 4.4.1.2. Phân tích bộ đếm (12) – Ví dụ 4:

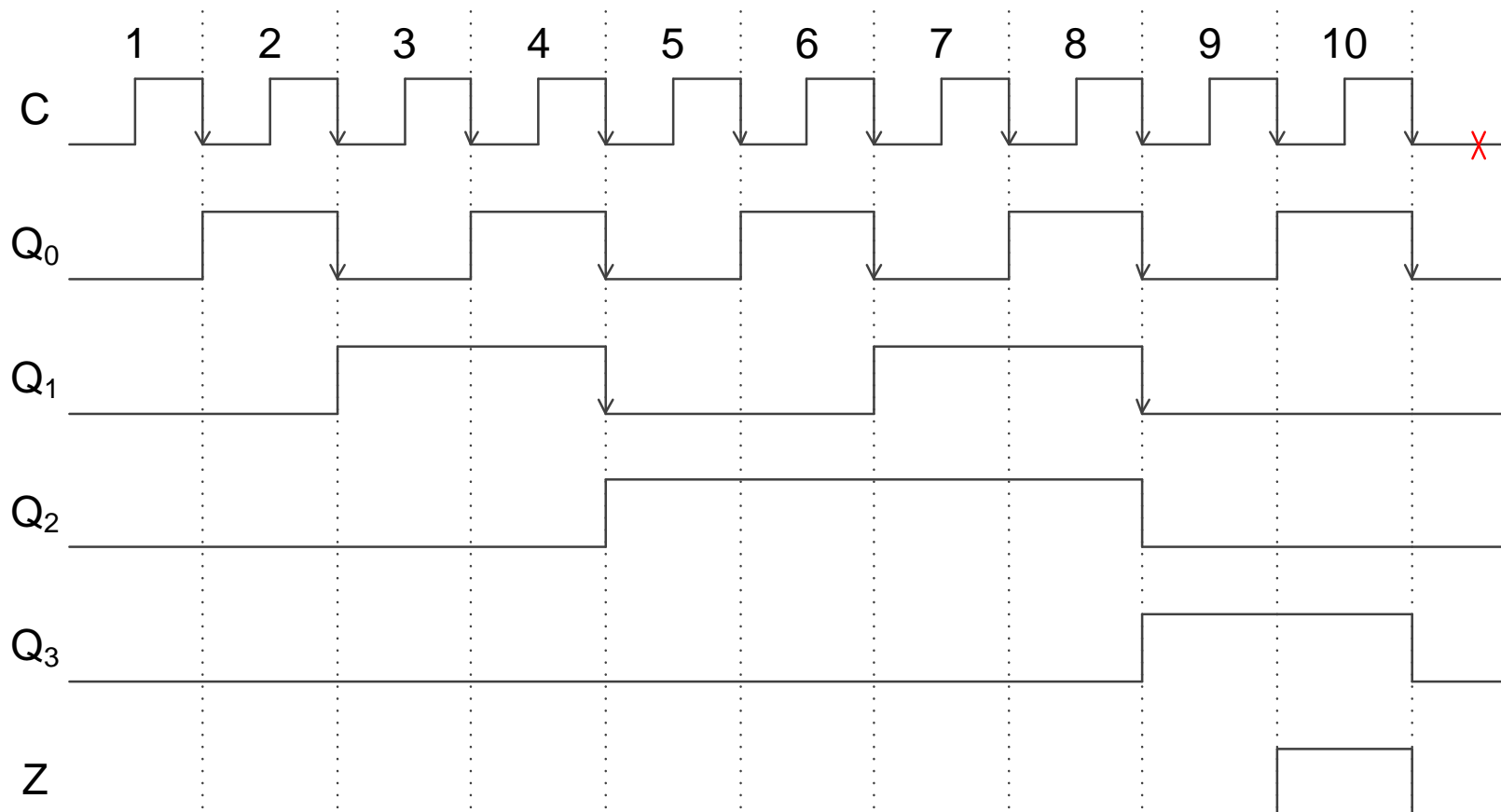
B6: Giải đồ xung:



NX: Bộ đếm mod 10, thuận, không đồng bộ, sử dụng tri gơ JK, có khả năng tự khởi động sau tối đa 02 chu kỳ clock.

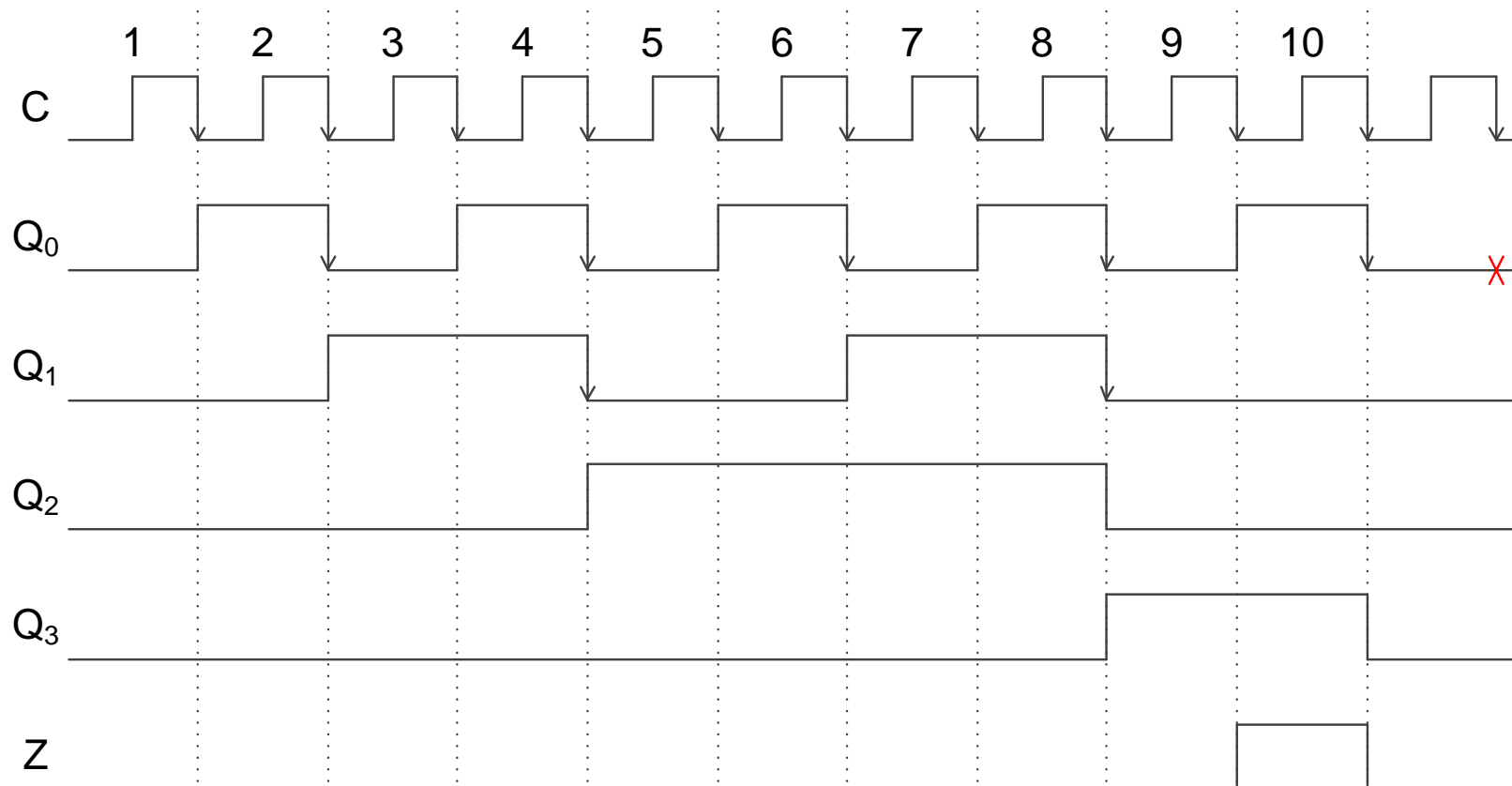
### 4.4.1.2. Phân tích bộ đếm (12) – Ví dụ 4:

B6: Giảm đồ xung: **Vẽ sai (1)**



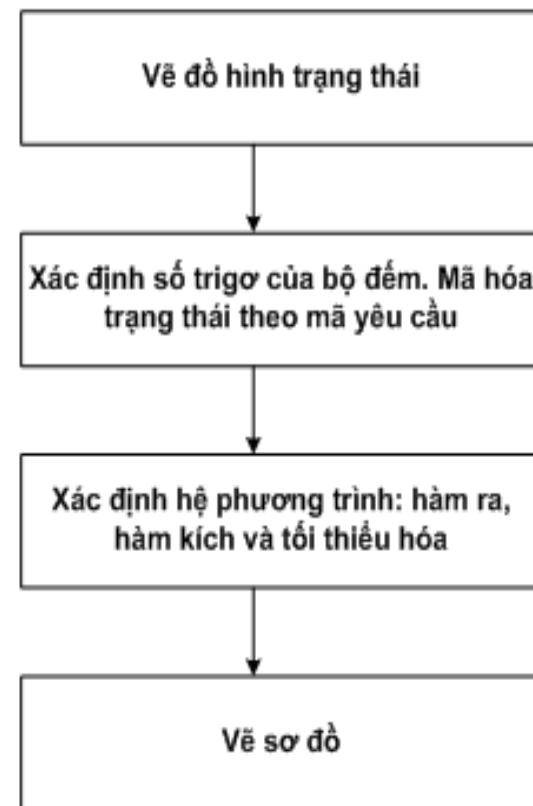
### 4.4.1.2. Phân tích bộ đếm (12) – Ví dụ 4:

B6: Giải đồ xung: **Vẽ sai (2)**



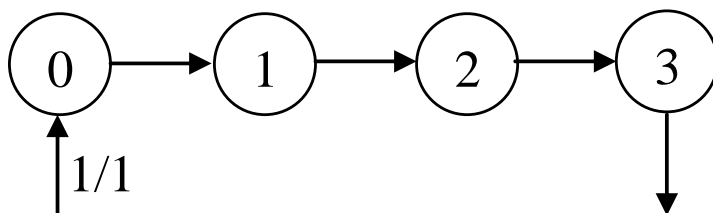
### 4.4.1.3. Thiết kế bộ đếm (1) – Bộ đếm đồng bộ

- B1: Phân tích bài toán để xây dựng đồ hình trạng thái.
- B2: Xác định số lượng và loại trigger sử dụng, thực hiện mã hoá trạng thái.
- B3: - Xác định hệ phương trình:  
phương trình định thời  
phương trình hàm ra  
phương trình hàm kích (từ phương trình đặc trưng hoặc từ bảng hàm kích)  
- Kiểm tra khả năng tự khởi động (nếu cần).
- B4: Vẽ mạch điện thực hiện.



### 4.4.1.3. Thiết kế bộ đếm (2) – Ví dụ 1: $M_d = 4$ , đồng bộ

B1:  $M_d = 4$ : có đồ hình trạng thái:

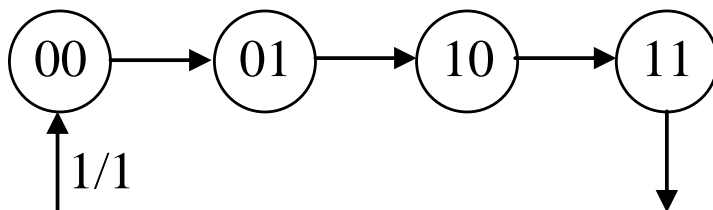


B3: - Phương trình hàm ra:

Z	Q <sub>0</sub>	
	0	1
Q <sub>1</sub>	0	0
	1	1

B2:  $M_d = 4$ : cần dùng 2 trigơ, sử dụng loại JK.

Mã hoá trạng thái:



$$Z = Q_1 \cdot Q_0$$

## 4.4.1.3. Thiết kế bộ đếm (3) – Ví dụ 1: $Md = 4$ , đồng bộ

$Q_1 \backslash Q_0$	0	1
0	01	10
1	11	00

$Q_1^k \backslash Q_0$	0	1
0	0	1
1	1	0

$Q_0^k \backslash Q_0$	0	1
0	1	0
1	1	0

- Bảng chuyển đổi trạng thái:

$$\Rightarrow Q_1^k = Q_0 \cdot \overline{Q_1} + \overline{Q_0} \cdot Q_1$$

$$\Rightarrow Q_0^k = \overline{Q_0}$$

### 4.4.1.3. Thiết kế bộ đếm (4) – Ví dụ 1: $Md = 4$ , đồng bộ

B4: Tìm phương trình hàm kích:

Cách 1: Tìm phương trình hàm kích từ phương trình đặc trưng:

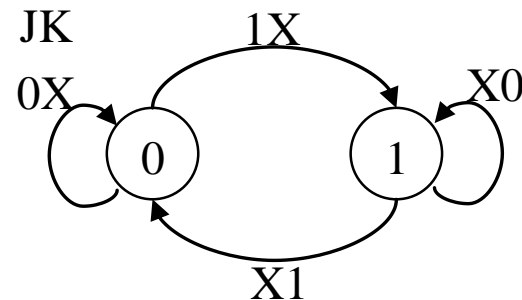
$$\begin{aligned} Q_0^k &= \overline{Q_0} = 1 \cdot \overline{Q_0} + 0 \cdot Q_0 \\ &= J_0 \cdot \overline{Q_0} + \overline{K_0} \cdot Q_0 \end{aligned} \Rightarrow J_0 = K_0 = 1$$

$$\begin{aligned} Q_1^k &= Q_0 \cdot \overline{Q_1} + \overline{Q_0} \cdot Q_1 \\ &= J_1 \cdot \overline{Q_1} + \overline{K_1} \cdot Q_1 \end{aligned} \Rightarrow J_1 = K_1 = Q_0$$

### 4.4.1.3. Thiết kế bộ đếm (5) – Ví dụ 1: $Md = 4$ , đồng bộ

Cách 2: Tìm phương trình hàm kích từ từ bảng hàm kích:

$Q_1$	$Q_0$	$Q_1^k$	$Q_0^k$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	1	0	X	1	X
0	1	1	0	1	X	X	1
1	0	1	1	X	0	1	X
1	1	0	0	X	1	X	1



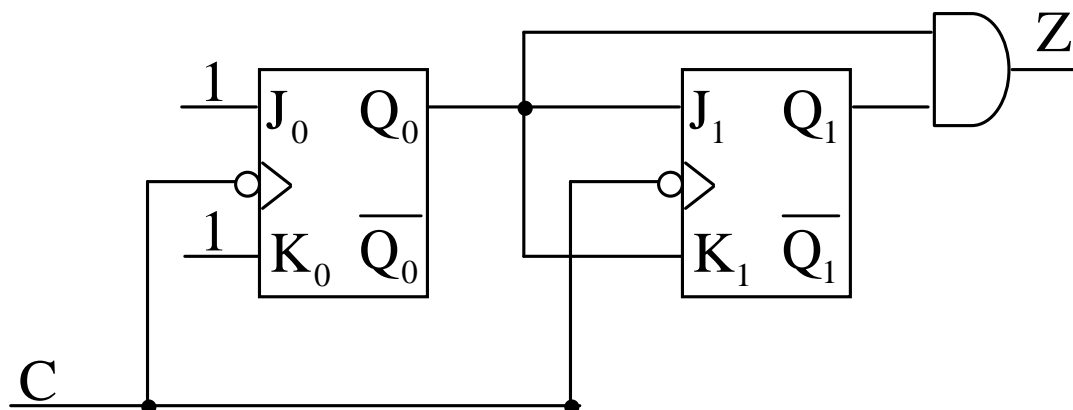
- Lập bảng Karnaugh của J,K:

$$\Rightarrow \begin{cases} J_1 = K_1 = Q_0 \\ J_0 = K_0 = 1 \end{cases}$$



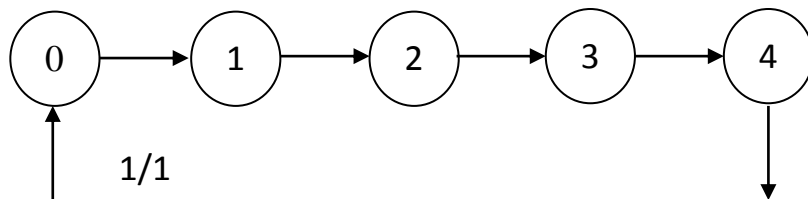
### 4.4.1.3. Thiết kế bộ đếm (6) – Ví dụ 1: $Md = 4$ , đồng bộ

B6: Vẽ mạch điện:

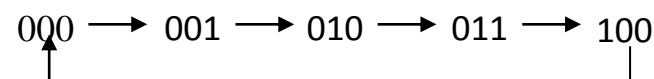


## 4.4.1.3. Thiết kế bộ đếm (7) – Ví dụ 2: $Md = 5$ , đồng bộ

B1: Đồ hình trạng thái:



B2: Mã hóa trạng thái:



B3: Tìm hệ phương trình:

$Q_2$	$Q_1$	$Q_0$	$Q_2^k$	$Q_1^k$	$Q_0^k$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
0	0	0	0	0	1	0	X	0	X	1	X
0	0	1	0	1	0	0	X	1	X	X	1
0	1	0	0	1	1	0	X	X	0	1	X
0	1	1	1	0	0	1	X	X	1	X	1
1	0	0	0	0	0	X	1	0	X	0	X

### 4.4.1.3. Thiết kế bộ đếm (8) – Ví dụ 2: $Md = 5$ , đồng bộ

Lập bảng Karnaugh của J,K, từ đó viết các phương trình chuyển đổi trạng thái:

$$\begin{cases} J_0 = \overline{Q_2} \\ K_0 = 1 \end{cases}, \begin{cases} J_1 = Q_0 \\ K_1 = Q_0 \end{cases}, \begin{cases} J_2 = Q_1 \cdot Q_0 \\ K_2 = 1 \end{cases}$$

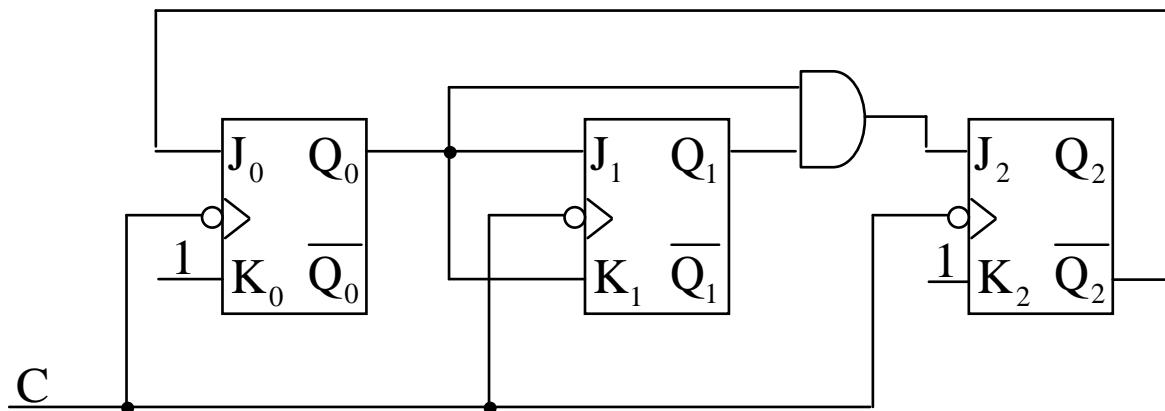
$$\begin{aligned} Q_0^k &= \overline{Q_2} \cdot \overline{Q_0} \\ Q_1^k &= Q_0 \cdot \overline{Q_1} + \overline{Q_0} \cdot Q_1 = Q_1 \oplus Q_0 \\ Q_2^k &= \overline{Q_2} \cdot Q_1 \cdot Q_0 \end{aligned}$$

Kiểm tra khả năng tự khởi động:

$Q_2$	$Q_1$	$Q_0$	$Q_2^k$	$Q_1^k$	$Q_0^k$
1	0	1	0	1	0
1	1	0	0	1	0
1	1	1	0	0	0

### 4.4.1.3. Thiết kế bộ đếm (9) – Ví dụ 2: $Md = 5$ , đồng bộ

B6: Vẽ mạch điện:



## 4.4.1.3. Thiết kế bộ đếm (10) – Bộ đếm không đồng bộ

B1: Phân tích bài toán để xây dựng đồ hình trạng thái.

B2: Xác định số lượng và loại trigơ sử dụng, thực hiện mã hoá trạng thái.

B3: Tìm hệ phương trình:

- Vẽ giản đồ xung để tìm phương trình định thời.
- Xác định phương trình hàm ra.
- Xác định phương trình hàm kích:

Cách 1: Tìm phương trình hàm kích từ pt chuyển đổi trạng thái:

- + Lập bảng chuyển đổi trạng thái
- + Nếu đúng sườn xung nhịp,  $Q^k$  có giá trị như trong bảng chuyển đổi trạng thái
- + Nếu không đúng sườn xung nhịp, giá trị  $Q^k$  là tùy chọn (x)
- + Chú ý đưa  $Q^k$  về dạng phương trình đặc trưng để tìm các đầu vào kích.

## 4.4.1.3. Thiết kế bộ đếm (11) – Bộ đếm không đồng bộ

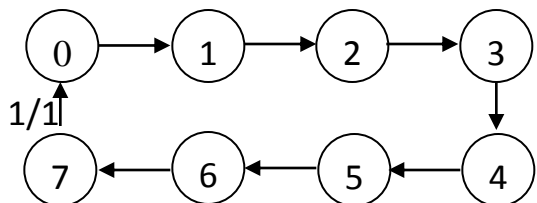
Cách 2: Tìm phương trình hàm kích từ bảng hàm kích:

- + Lập bảng hàm kích
- + Nếu đúng sườn xung nhịp, các đầu vào kích có giá trị như trong bảng hàm kích.
- + Nếu không đúng sườn xung nhịp, giá trị các đầu vào kích là tùy chọn.
- Kiểm tra khả năng tự khởi động (nếu cần).

B4: Vẽ mạch điện thực hiện.

### 4.4.1.3. Thiết kế bộ đếm (12) – VD1: $Md=8$ , không đồng bộ

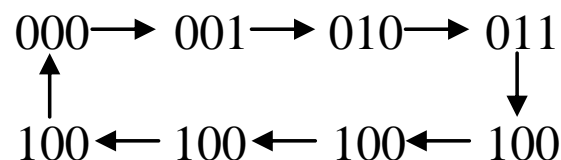
B1: Đồ hình trạng thái:



B2: Mã hóa trạng thái:

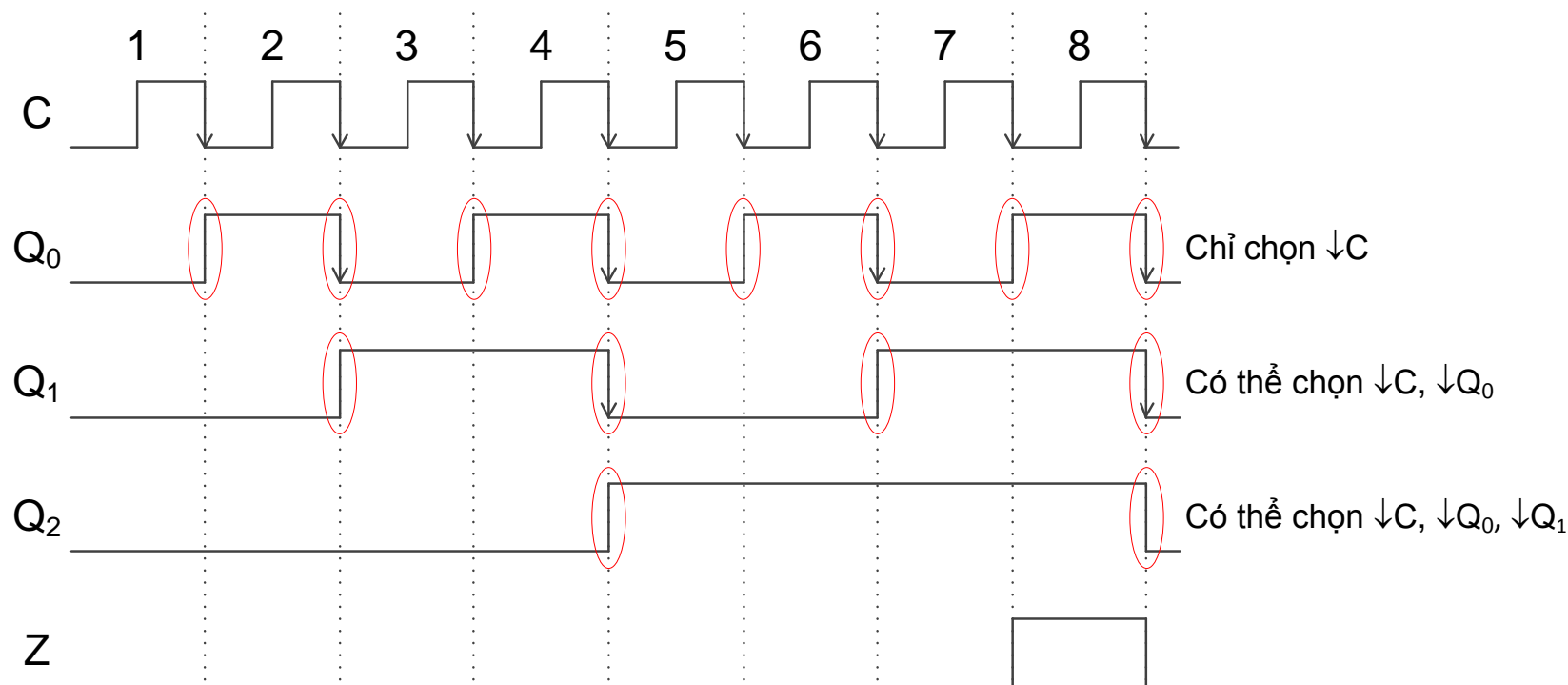
Sử dụng 3 trigơ, loại JK, kích khởi sườn âm.

Mã hoá trạng thái:



### 4.4.1.3. Thiết kế bộ đếm (13) – VD1: $Md=8$ , không đồng bộ

B3: Vẽ giản đồ xung: (dựa vào đồ thị mã hoá trạng thái)

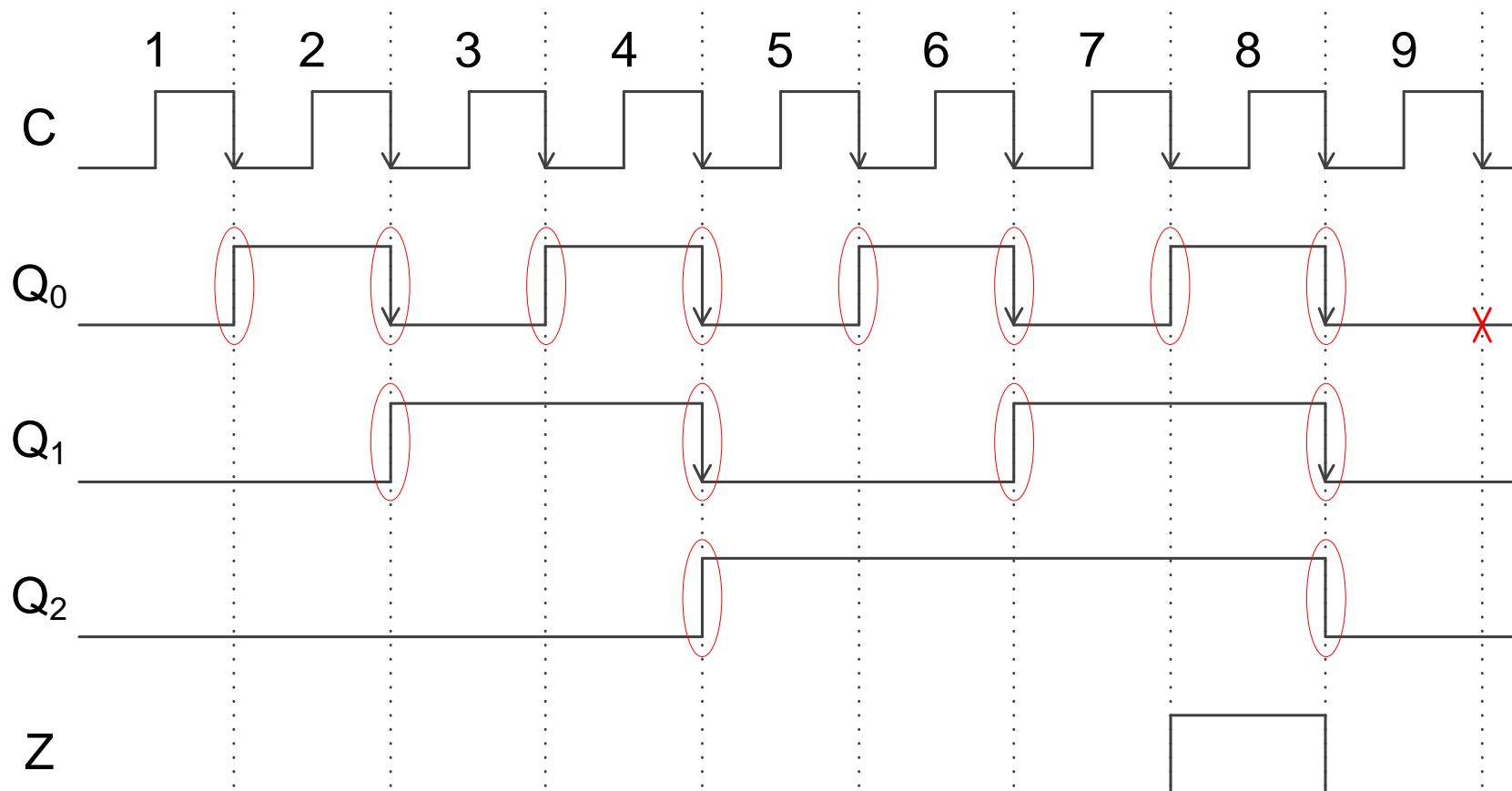


Từ giản đồ xung, chọn:  $C_0 = \downarrow C$ ,  $C_1 = \downarrow Q_0$ ,  $C_2 = \downarrow Q_1$



### 4.4.1.3. Thiết kế bộ đếm (13) – VD1: $Md=8$ , không đồng bộ

B3: Vẽ giản đồ xung: *(Vẽ sai)*



### 4.4.1.3. Thiết kế bộ đếm (14) – VD1: $Md=8$ , không đồng bộ

Lập bảng chuyển đổi trạng thái, viết phương trình chuyển đổi trạng thái:

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	001	010	100	011
1	101	110	000	111

$Q_2^k \backslash Q_1^k Q_0^k$	00	01	11	10
0	1	0	0	1
1	1	0	0	1

$Q_2^k \backslash Q_1^k Q_0^k$	00	01	11	10
0	X	1	0	X
1	X	1	0	X

$Q_2^k \backslash Q_1^k Q_0^k$	00	01	11	10
0	X	X	1	X
1	X	X	0	X

Rút gọn bằng bảng Karnaugh, ta có:

$$Q_0^k = \overline{Q_0} \quad , \quad Q_1^k = \overline{Q_1} \quad , \quad Q_2^k = \overline{Q_2}$$

### 4.4.1.3. Thiết kế bộ đếm (15) – VD1: $M_d=8$ , không đồng bộ

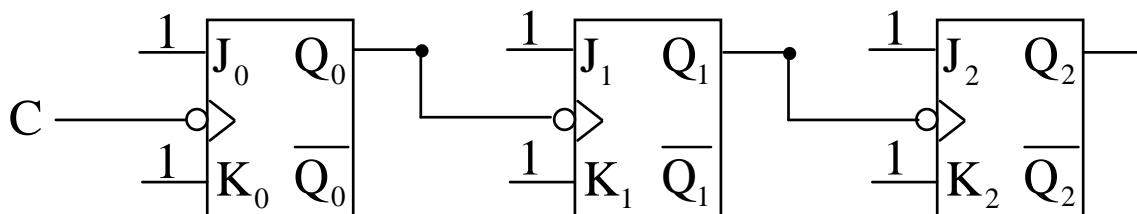
Phương trình hàm kích: (tìm pt hàm kích từ pt chuyển đổi trạng thái)

$$Q_0^k = \overline{Q_0} = 1.\overline{Q_0} + 0.Q_0 \Rightarrow J_0 = K_0 = 1$$

$$Q_1^k = \overline{Q_1} = 1.\overline{Q_1} + 0.Q_1 \Rightarrow J_1 = K_1 = 1$$

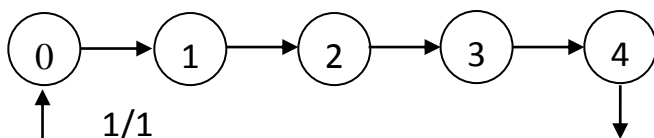
$$Q_2^k = \overline{Q_2} = 1.\overline{Q_2} + 0.Q_2 \Rightarrow J_2 = K_2 = 1$$

B4: Vẽ mạch điện:

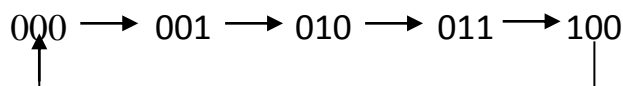


## 4.4.1.3. Thiết kế bộ đếm (16) – VD2: $Md=5$ , không đồng bộ

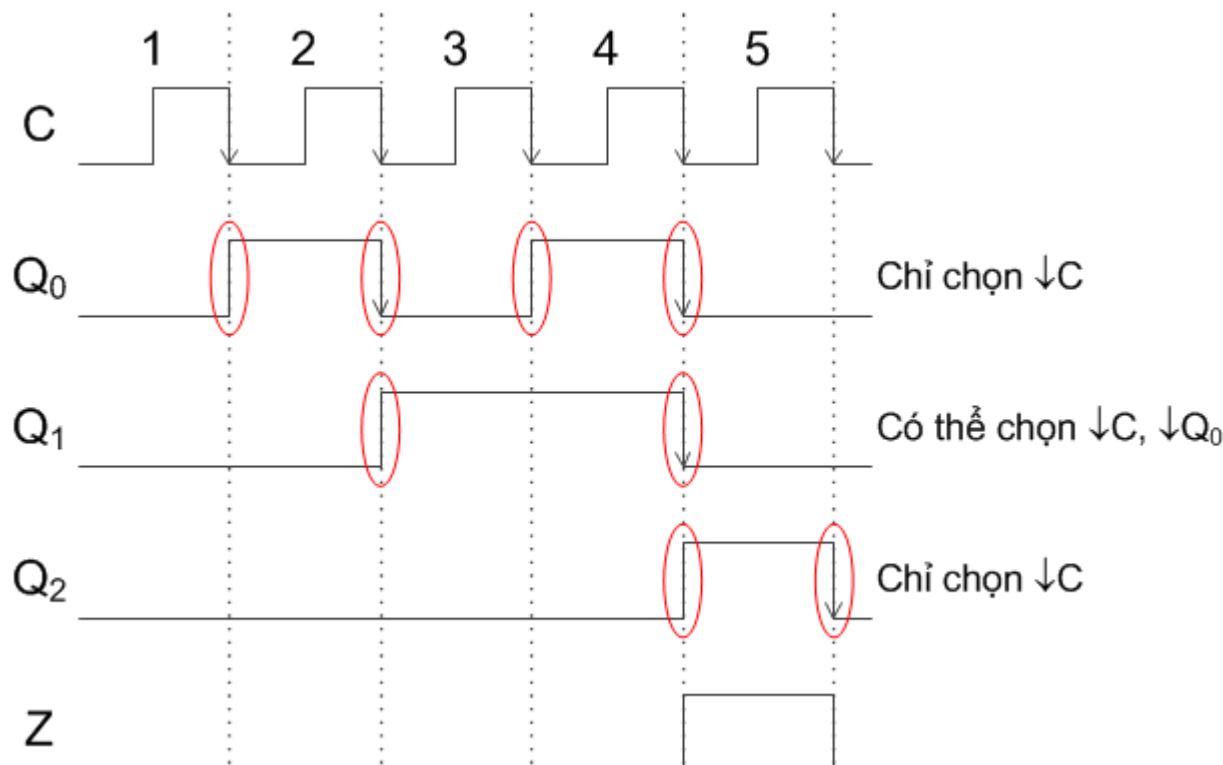
B1: Đồ hình trạng thái:



B2: Mã hóa trạng thái:



B3: - Giải đồ xung:



### 4.4.1.3. Thiết kế bộ đếm (17) – VD2: $Md=5$ , không đồng bộ

Lập bảng chuyển đổi trạng thái, viết phương trình chuyển đổi trạng thái:

$Q_2 \backslash Q_1 Q_0$	00	01	11	10
0	001	010	100	011
1	000	X	X	X

$Q_0^k \backslash Q_1 Q_0$	00	01	11	10
0	1	0	0	1
1	0	X	X	X

$Q_1^k \backslash Q_1 Q_0$	00	01	11	10
0	X	1	0	X
1	X	X	X	X

$Q_2^k \backslash Q_1 Q_0$	00	01	11	10
0	0	0	1	0
1	0	X	X	X

Rút gọn bằng bảng Karnaugh, ta có:

$$Q_0^k = \overline{Q_2} \cdot \overline{Q_0}, \quad Q_1^k = \overline{Q_1}, \quad Q_2^k = \overline{Q_2} \cdot Q_1 \cdot Q_0$$

### 4.4.1.3. Thiết kế bộ đếm (18) – VD2: $Md=5$ , không đồng bộ

Phương trình hàm kích (xác định từ phương trình chuyển đổi trạng thái):

$$Q_0^k = \overline{Q_2} \cdot \overline{Q_0} = \overline{Q_2} \cdot \overline{Q_0} + 0 \cdot Q_0 \Rightarrow J_0 = \overline{Q_2}, K_0 = 1$$

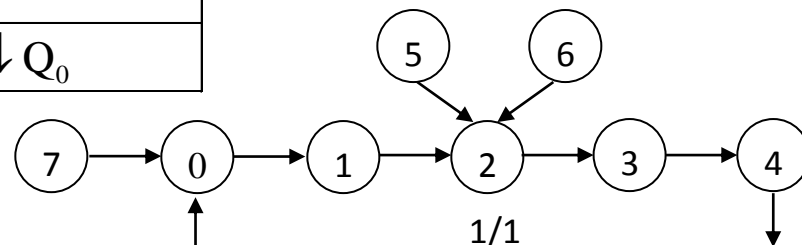
$$Q_1^k = \overline{Q_1} = 1 \cdot \overline{Q_1} + 0 \cdot Q_1 \Rightarrow J_1 = K_1 = 1$$

$$Q_2^k = \overline{Q_2} \cdot Q_1 \cdot Q_0 = Q_1 \cdot Q_0 \cdot \overline{Q_2} + 0 \cdot Q_2 \Rightarrow J_2 = Q_1 \cdot Q_0, K_2 = 1$$

Kiểm tra khả năng tự khởi động:

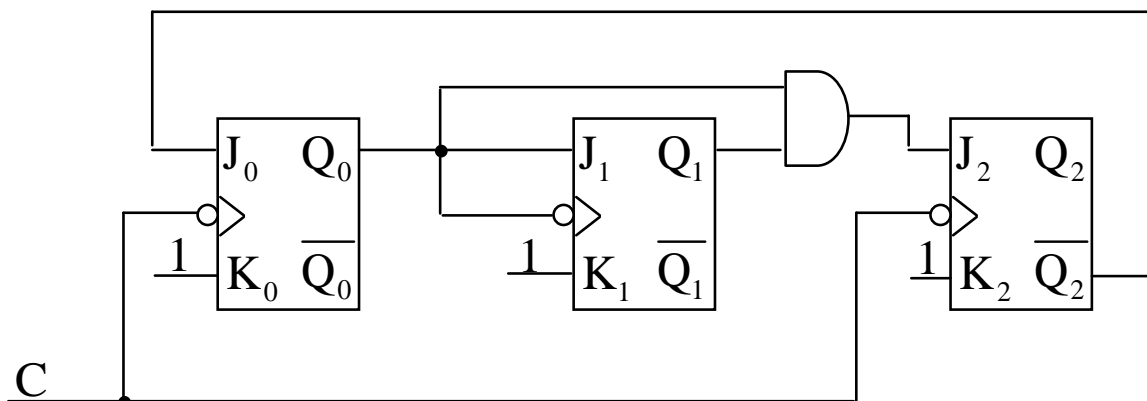
$Q_2$	$Q_1$	$Q_0$	$Q_2^k$	$Q_1^k$	$Q_0^k$	Định thời
1	0	1	0	1	0	$\downarrow C, \downarrow Q_0$
1	1	0	0	1	0	$\downarrow C$
1	1	1	0	0	0	$\downarrow C, \downarrow Q_0$

Bộ đếm có khả năng tự khởi động.



### 4.4.1.3. Thiết kế bộ đếm (19) – VD2: $M_d=5$ , không đồng bộ

B4: Vẽ mạch điện:



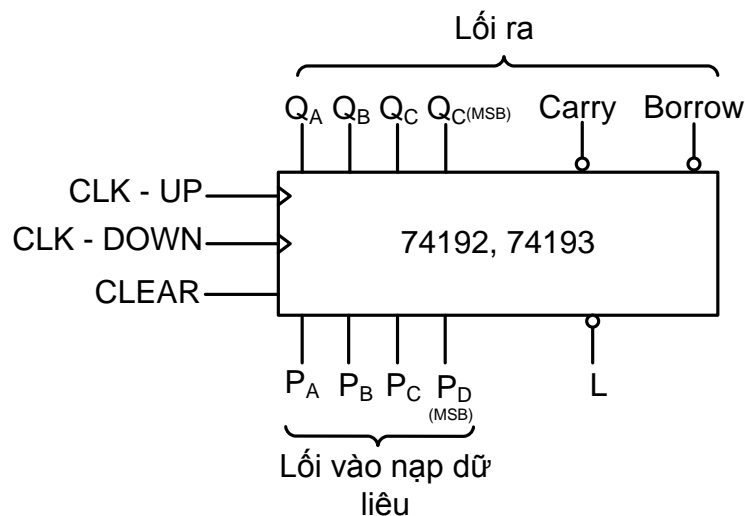
## 4.4.1.4. Giới thiệu một số IC đếm (1)

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	



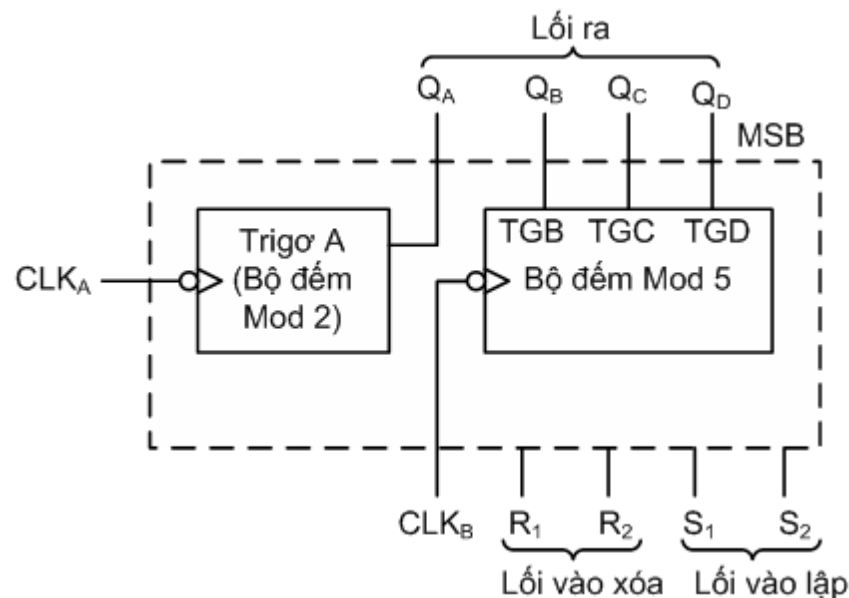
## 4.4.1.4. Giới thiệu một số IC đếm (2) – IC đếm 74192, 74193

- Trong các bộ đếm này, khi thực 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.



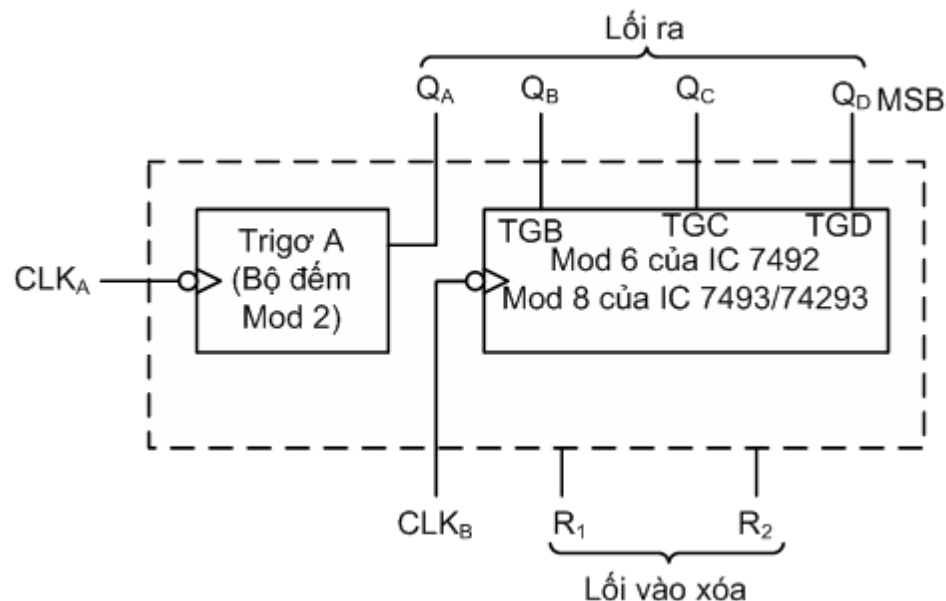
## 4.4.1.4. Giới thiệu một số IC đếm (3)– IC đếm 7490, 74390

- 7490 bao gồm 4 trigơ cung cấp bộ đếm gồm hai Mod đếm độc lập: Mod 2 và Mod 5.
- Trigơ A thực hiện đếm Mod 2, Trigơ B, C, D thực hiện đếm Mod 5.
- IC 74390 là bản kép (dual) của 7490



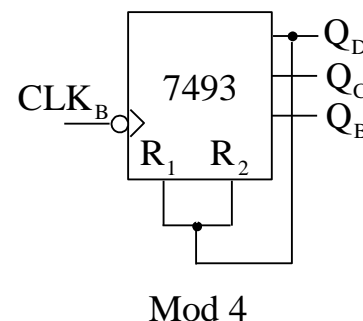
## 4.4.1.4. Giới thiệu một số IC đếm (3)– IC đếm 7492, 7493

- Bao gồm 4 trigơ cung cấp bộ đếm gồm hai Mod đếm: Mod 2 và Mod 6 (7492) hoặc mod 8 (7493).
- Trigơ A thực hiện đếm Mod 2, Trigơ B, C, D thực hiện đếm Mod 6 hoặc mod 8.
- 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.



### 4.4.1.5. Thiết kế bộ đếm bất kỳ dùng IC đếm

- Xây dựng mod đếm bất kỳ sử dụng phương pháp hồi tiếp đầu ra về đầu vào xóa:
  - Biểu diễn mod đếm M dưới dạng nhị phân
  - Xác định số bit '1' cần xóa về '0': bằng cách đưa đầu ra Q tương ứng hồi tiếp về chân Reset.
  - Nếu có nhiều hơn hai bit '1' cần xóa, sử dụng cổng AND hoặc NAND trước khi đưa về chân Reset.
- Ví dụ: Sử dụng IC 7493 thực hiện bộ đếm có  $M = 4$ .
  - $M = 4 = 100$  nên cần hồi tiếp QD về chân Reset.
  - Sơ đồ:



## 4.4. Mạch tuần tự thông dụng

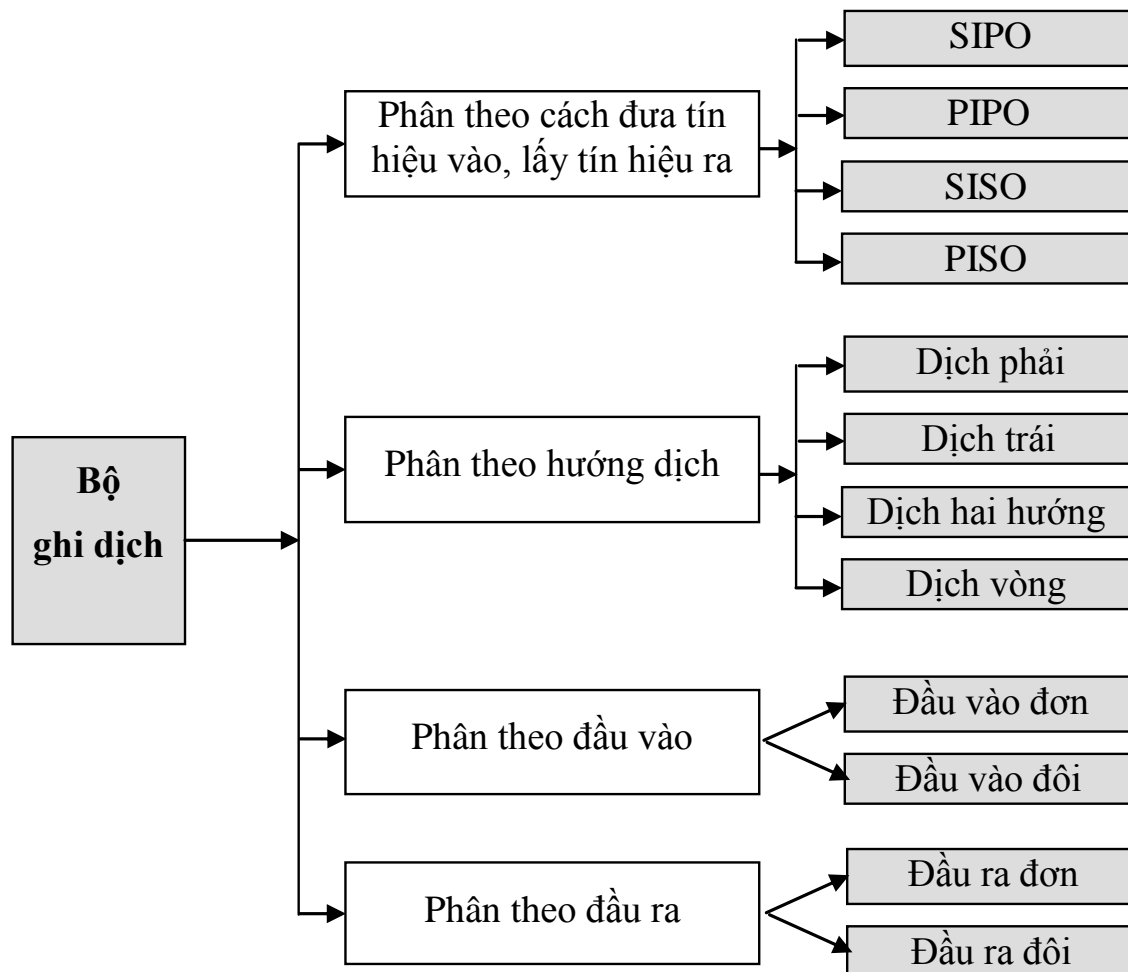
### 4.4.1. Bộ đếm

### 4.4.2. Bộ ghi dịch

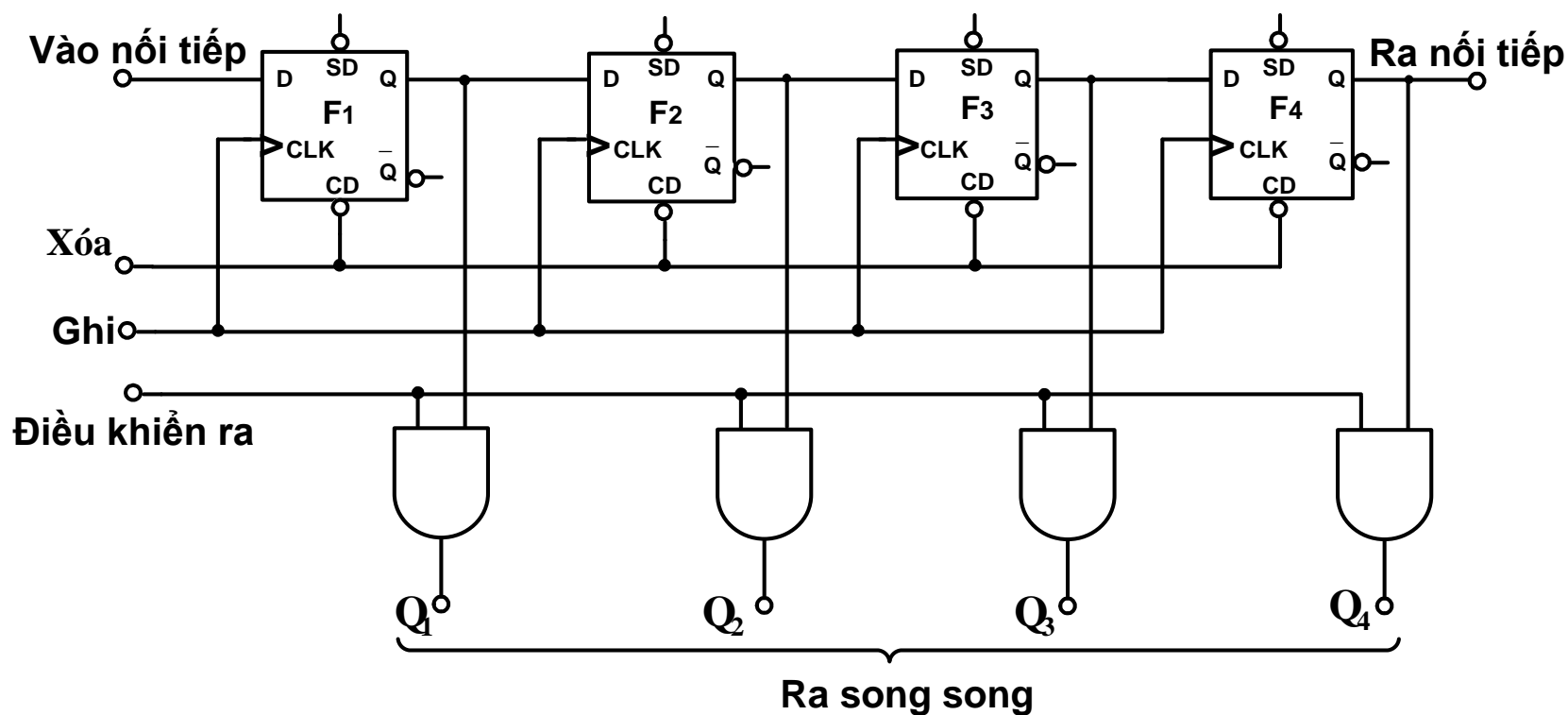
## 4.4.2.1. Định nghĩa và phân loại (1)

- Bộ ghi dịch (thanh ghi dịch) là phần tử không thể thiếu được trong CPU, trong các hệ vi xử lí.
- Nó có khả năng ghi (nhớ) số liệu và dịch thông tin (sang phải hoặc sang trái).
- Bộ ghi dịch 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 ( $n$  trigơ). Trong các bộ ghi dịch thường dùng các trigơ đồng bộ như trigơ RS, T, JK, D.
- Thông thường người ta hay dùng trigơ D hoặc các trigơ khác nhưng mắc theo kiểu trigơ D để tạo thành các bộ ghi.
- Hoạt động dịch dữ liệu được thực hiện từng bit theo điều khiển của xung nhịp.

### 4.4.2.1. Định nghĩa và phân loại (2)



### 4.4.2.2. Bộ ghi dịch 4 bit (1) – Sơ đồ khối:





### 4.4.2.2. Bộ ghi dịch 4 bit (2)– Hoạt động:

\* Cách ghi dữ liệu vào:

- Số liệu:  $D_1, D_2, D_3, D_4$
- Dịch phải, nhập  $D_4$  trước.
- Sau 4 xung nhịp: ghi xong.

\* Cách lấy số liệu ra:

- Ra song song: Sau 4 xung nhịp:

$$Q_1 Q_2 Q_3 Q_4 = D_1 D_2 D_3 D_4$$

Đặt “Điều khiển ra” = 1.

- Ra nối tiếp: Lấy ra ở  $Q_4$ .

Sau 4 xung nhịp:  $D_4$  xuất hiện ở  $Q_4$

Cần 3 xung nữa để

$D_1 D_2 D_3$  được đưa ra  $Q_4$ .

CLK	$Q_1$	$Q_2$	$Q_3$	$Q_4$
0	0	0	0	0
1	$D_4$	0	0	0
2	$D_3$	$D_4$	0	0
3	$D_2$	$D_3$	$D_4$	0
4	$D_1$	$D_2$	$D_3$	$D_4$
5	0	$D_1$	$D_2$	$D_3$
6	0	0	$D_1$	$D_2$
7	0	0	0	$D_1$

## 4.4.2.3. Ứng dụng của bộ ghi dịch (1)

- **Bộ chuyển dữ liệu từ nối tiếp sang song song**

Dùng bộ ghi dịch SIPO.

- **Bộ chuyển dữ liệu từ song song sang nối tiếp**

Dùng bộ ghi dịch PISO.

- **Bộ đếm vòng**

- **Bộ đếm vòng xoắn**

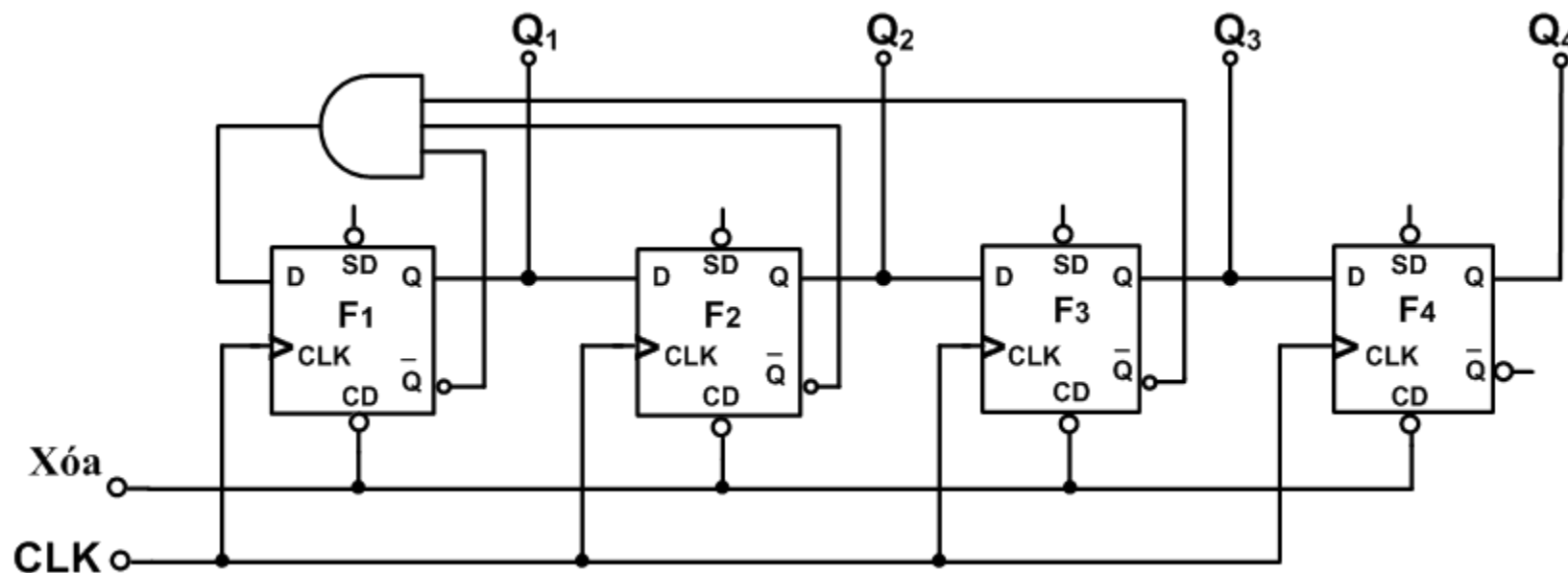
- **Bộ phát xung tuần tự**

Dùng làm bộ đếm, bộ phát xung giả ngẫu nhiên, ...

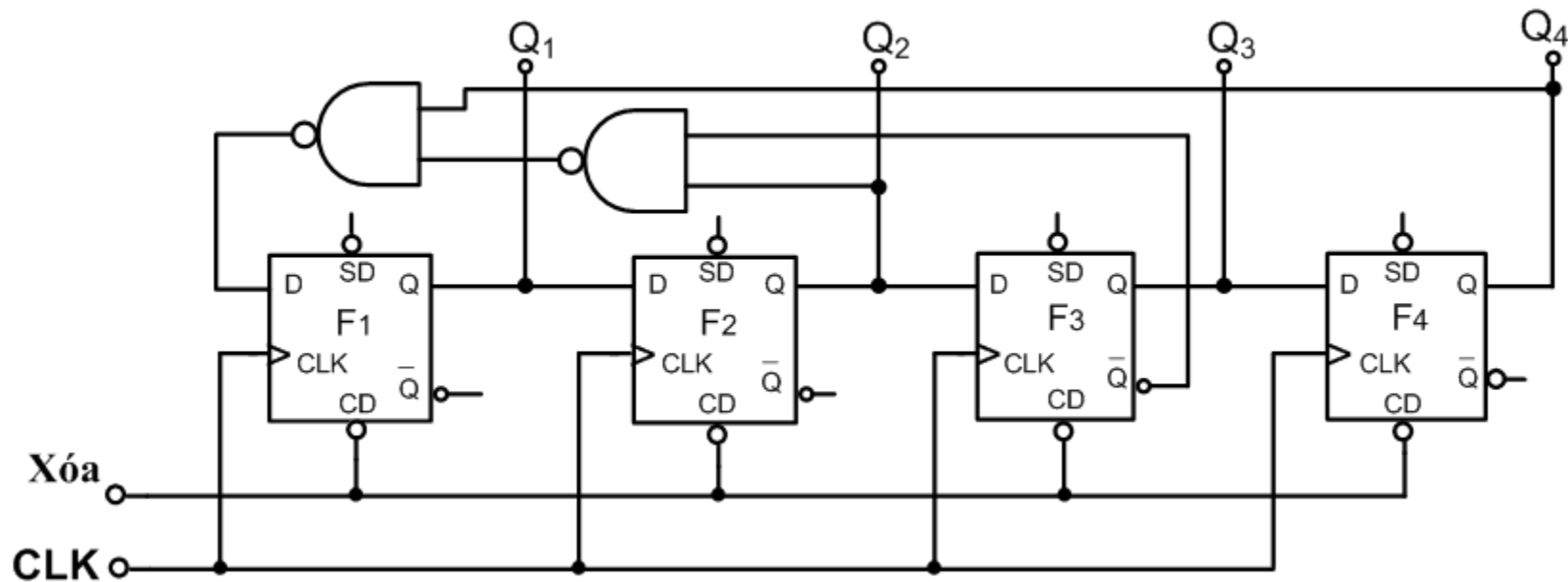
- **Thanh chốt dữ liệu (Latch)**

Sử dụng trong các mạch giao tiếp Bus dữ liệu, các bộ phân kênh, hợp kênh, và trong các mạch điều khiển...

## 4.4.2.3. Ứng dụng của bộ ghi dịch (2) – Bộ đếm vòng 4 bit



#### 4.4.2.3. Ứng dụng của bộ ghi dịch (3) – Bộ đếm vòng xoắn 4 bit



## 4.4.2.3. Ứng dụng của bộ ghi dịch (4) – Bộ tạo tín hiệu giả ngẫu nhiên

