



HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÀI GIẢNG MÔN

KỸ THUẬT SỐ

Giảng viên:

Vũ Anh Đào

Điện thoại/E-mail:

daova@ptit.edu.vn

Bộ môn:

Kỹ thuật điện tử

Học kỳ/Năm biên soạn: 2020

Giới thiệu môn học

Mục đích:

- Trang bị cho sinh viên phương pháp phân tích, thiết kế, chế tạo một hệ thống số; các kiến thức phần cứng, phần mềm, mối liên hệ giữa phần cứng, phần mềm.

* **Đối tượng:** Đại học Công nghệ thông tin

* **Thời lượng:** 2 tín chỉ

- Lý thuyết : 26 tiết
- Kiểm tra : 2 tiết
- Thí nghiệm: 2 tiết

* **Điểm thành phần:**

- Chuyên cần : 10%
- Kiểm tra : 10%
- Thí nghiệm : 10%
- Thi kết thúc học phần : 70%

Nội dung

Chương 1: Hệ đếm

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

Chương 3: Cổng logic

Chương 4: Mạch logic tổ hợp

Chương 5: Mạch logic tuần tự

HỆ ĐẾM

Hệ đếm (1)

- ❖ Khái niệm chung
- ❖ Biểu diễn số
- ❖ Chuyển đổi giữa các hệ đếm
- ❖ Số nhị phân có dấu
- ❖ Dấu phẩy động

Hệ đếm (2)

❖ Khái niệm chung

- ✓ Dùng một số hữu hạn các ký hiệu ghép với nhau theo qui ước về vị trí, số ký hiệu (r) là cơ số.
 - ✓ Giá trị biểu diễn của các ký hiệu được phân biệt thông qua trọng số r^i , với i là số nguyên dương hoặc âm.
- ❖ Tên gọi, ký hiệu và cơ số của một vài hệ đếm thông dụng

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

Chú ý: Gọi hệ đếm theo cơ số. VD: hệ nhị phân = Hệ cơ số 2...

Hệ đếm (3)

❖ Biểu diễn số tổng quát:

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

Trong đó N là giá trị, a là hệ số nhân; n là số chữ số phần nguyên; m là số chữ số phần phân số.

❖ Thêm chỉ số để tránh nhầm lẫn giữa các hệ, VD: 36_{10} , 36_8 ...

❖ Hệ thập phân (Decimal): $r = 10$. VD:

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

✓ **Ưu:** dễ nhận biết, biểu diễn gọn, ít thời gian viết và đọc.

✓ **Nhược:** Khó thể hiện bằng thiết bị kỹ thuật

Hệ đếm (4)

❖ Hệ nhị phân (Binary): $r = 2$. VD:

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

❖ Ưu: Dễ thể hiện bằng các thiết bị cơ, điện, là ngôn ngữ của mạch logic, thiết bị tính toán hiện đại - ngôn ngữ máy.

✓ Nhược: Biểu diễn dài, mất nhiều thời gian viết, đọc.

✓ Các phép tính:

➤ Cộng: $0 + 0 = 0$, $1 + 0 = 1$, $1 + 1 = 10$

➤ Trừ: $0 - 0 = 0$; $1 - 1 = 0$; $1 - 0 = 1$; $10 - 1 = 1$ (mượn 1)

➤ Nhân: $0 \times 0 = 0$, $0 \times 1 = 0$, $1 \times 0 = 0$, $1 \times 1 = 1$

➤ Chia: Tương tự phép chia 2 số thập phân

❖ VD: $1011101_2 + 10010111$ $1110101_2 + 100011$

Hệ đếm (5)

- ❖ **Hệ bát phân (Octal):** $r = 8 = 2^3 \rightarrow$ thay bằng 3 bit nhị phân:

$$1234.56_8 = 1 \times 8^3 + 2 \times 8^2 + 3 \times 8^1 + 4 \times 8^0 + 5 \times 8^{-1} + 6 \times 8^{-2}$$

$$= 001\ 010\ 011\ 100.101\ 110$$

- ❖ **Phép cộng:** cộng hai hoặc nhiều chữ số cùng trọng số lớn hơn hoặc bằng 8 phải nhớ lên chữ số có trọng số lớn hơn liền kề.
- ❖ **Phép trừ:** mượn 1 ở chữ số có trọng số lớn hơn thì cộng thêm 8.

$3\ 7\ 6_8$	$3\ 5\ 7_8$	$2\ 4\ 5_8$	$5\ 2\ 1_8$	$5\ 2\ 3_8$	$6\ 1\ 1_8$
+	+	-	-	+	-
$\underline{5\ 3\ 4_8}$	$\underline{4\ 3\ 6_8}$	$\underline{1\ 7\ 6_8}$	$\underline{3\ 5\ 2_8}$	$\underline{7\ 6\ 7_8}$	$\underline{5\ 2\ 7_8}$
$1\ 1\ 3\ 2_8$	$1\ 0\ 1\ 5_8$	$0\ 4\ 7_8$	$1\ 4\ 7_8$		

Hệ đếm (6)

- ❖ **Hệ thập lục phân (HexaDecimal):** $r = 16 = 2^4$

$$4ABF = 4 \times 16^3 + 10 \times 16^2 + 11 \times 16^1 + 15 \times 16^0$$

$$= 0100 \ 1010 \ 1011 \ 1111$$

- ❖ **Phép cộng:** cộng hai hoặc nhiều chữ số cùng trọng số lớn hơn hoặc bằng 16 phải nhớ lên chữ số có trọng số lớn hơn liền kề.

8 A C	9 9 D
+	+
<u>9 8 F</u>	<u>7 5 E</u>
1 2 3 B	

- ❖ **Phép trừ:** mượn 1 ở chữ số có trọng số lớn hơn thì cộng thêm 16.

3 5 8	4 2 5
-	-
<u>1 A D</u>	<u>3 C F</u>
1 A B	

- ❖ **Phép nhân:** đổi về số thập phân rồi thực hiện

Hệ đếm (7)

❖ Chuyển đổi cơ số giữa các hệ đếm

- Chuyển từ hệ cơ số 10 sang các hệ khác

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

- 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.

- 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.

Hệ đếm (8)

❖ **Đổi số 22.125_{10} sang số nhị phân**

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

Phần phân số

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

Kết quả biểu diễn nhị phân: 10110.001

Bài tập: chuyển số 83.87_{10} sang số nhị phân

Hệ đếm (9)

❖ 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

- Nhóm các cặp 3(hoặc 4 bit) từ bit LSB lại thành từng nhóm, chuyển nhóm đó sang Octal (hoặc hex). Nếu nhóm cuối thiếu bit thì thêm 0 vào cho đủ nhóm.

Hệ đếm (10)

❖ Số nhị phân có dấu

- Sử dụng một bit dấu: '0' là dương (+), '1' là âm (-). VD: số 6: **00000110**, số -6: **10000110**.
- Sử dụng phép bù 1: Lấy bù 1 các bit trị số (đảo của các bit). VD: số 4: **00000100**, số -4: **111111011**.
- Sử dụng phép bù 2: Số dương là số nhị phân không bù, số âm được biểu diễn qua bù 2 (bù 1 cộng 1).
- ✓ Bù 2 theo phương pháp xen kẽ: 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. VD: số 4: **00000100**, số -4: **111111100**.

❖ VD. Tìm bù 1 và bù 2 của các số sau:

10010101; 01101011; 10110111

Hệ đếm (11)

❖ Phép cộng theo bù 1

- **Hai số cùng dấu:** cộng trị số, dấu chung.
- **Số dương > số âm:** cộng trị số của số dương với bù 1 của số âm. Bit tràn vào kết quả trung gian. Dấu dương.
- **Số dương < số âm:** cộng trị số của số dương với bù 1 của số âm. Lấy bù 1 của tổng trung gian. Dấu âm.

❖ VD:

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

Hệ đếm (12)

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

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

❖ Phép cộng theo bù 2

- **Hai số dương:** cộng bình thường, dấu dương.
- **Hai số âm:** lấy bù 2 cả hai số và cộng, kết quả ở dạng bù 2.
- **Số dương > số âm:** số dương cộng với bù 2 của số âm. Kết quả bao gồm cả bit dấu, bit tràn bỏ đi.
- **Số dương < số âm:** số dương cộng với bù 2 của số âm. Kết quả ở dạng bù 2 của số dương tương ứng. Bit dấu là 1.

Hệ đếm (13)

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

$$\begin{array}{r}
 00001011_2 \quad (+11_{10}) \\
 + 11111001_2 \quad (-7_{10}) \\
 \hline
 100000100_2
 \end{array}$$

↓

Bít tràn → bỏ đi

$$\begin{array}{r}
 00000100_2 \quad (+4_{10})
 \end{array}$$

$$\begin{array}{r}
 11110101_2 \quad (-11_{10}) \\
 + 11111001_2 \quad (-7_{10}) \\
 \hline
 111101110_2
 \end{array}$$

↓

Bít tràn → bỏ đi

$$\begin{array}{r}
 11101110_2 \quad (-18_{10})
 \end{array}$$

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

Hệ đếm (14)

❖ Dấu phẩy động

- VD: $197,627_{10} = 197627 \times 10^{-3}$; $97,627_{10} = 0,197627 \times 10^{+3}$
- Gồm hai phần: số mũ E (phần đặc tính) và phần định trị M (trường phân số). E từ 5 đến 20 bit, M từ 8 đến 200 bit và:

$$X = 2^{E_x} (M_x) \quad 1/2 \leq |M| \leq 1$$

- Giả sử $X = 2^{E_x} (M_x)$ và $Y = 2^{E_y} (M_y)$ thì:
- Nhân: $Z = X.Y = 2^{E_x + E_y} (M_x.M_y) = 2^{E_z} M_z$
- Chia: $W = X/Y = 2^{E_x - E_y} (M_x / M_y) = 2^{E_w} M_w$
- Tổng(hiệu): đưa các số hạng về cùng số mũ, số mũ của tổng(hiệu) là số mũ chung, định trị của tổng(hiệu) là tổng(hiệu) các định trị.

Hệ đếm (15)

- ❖ Đổi số Binary sau sang dạng Octal: 0101111101001110
A) 57514 B) 57515 C) 57516 D) 57517
- ❖ Thực hiện phép tính: $132,44_{16} + 215,02_{16}$.
 - A) 347,46 B) 357,46 C) 347,56 D) 357,67
- ❖ Thực hiện phép cộng hai số có dấu sau theo bù 1:
 $0000\ 1101_2 + 1000\ 1011_2$
 - A) 0000 0101 B) 0000 0100 C) 0000 0011 D) 0000 0010
- ❖ Thực hiện phép cộng hai số có dấu sau theo bù 2:
 $0000\ 1101_2 - 1001\ 1000_2$
 - A) 1000 1110 B) 1000 1011 C) 1000 1100 D) 1000 1110

ĐẠI SỐ BOOLE VÀ CÁC PHƯƠNG PHÁP BIỂU DIỄN HÀM

Đại số Boole và các phương pháp biểu diễn hàm(1)

Nội dung

- ❖ Đại số Boole
- ❖ Các phương pháp biểu diễn hàm Boole
- ❖ Các phương pháp rút gọn hàm

Đại số Boole và các phương pháp biểu diễn hàm(2)

❖ Đại số Boole

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 định luật cơ bản:

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

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

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

Đại số Boole và các phương pháp biểu diễn hàm(3)

❖ Các phương pháp biểu diễn hàm Boole

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

❖ Bảng trạng thái

- ✓ Liệt kê giá trị mỗi biến và hàm theo từng cột riêng.
- ✓ Hàm n biến có 2^n tổ hợp độc lập gọi là các *hạng tích (minterm)*.
- ✓ Ưu: Rõ ràng, trực quan.
- ✓ Nhược: Phức tạp nếu nhiều biến

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

Đại số Boole và các phương pháp biểu diễn hàm(4)

❖ Bảng Karnaugh

- Tổ chức của bảng Các nô:
 - ✓ Biến được viết theo một dòng và một cột
 - ✓ 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:
 - ✓ Các ô *kế cận khác nhau một biến*
- Thiết lập bảng Các nô của một hàm:
 - ✓ Dạng **tổng các tích**, ghi 1 vào các ô ứng với hạng tích có mặt trong biểu diễn
 - ✓ Dạng **tích các tổng**, ghi 0 vào các ô ứng với hạng tổng

A \ BC	00	01	11	10
0				
1				

Đại số Boole và các phương pháp biểu diễn hàm(5)

❖ Phương pháp đại số

- 2 dạng biểu diễn: *tuyển* (tổng các tích) & *hội* (tích các tổng).
- ✓ Dạng tuyển: Mỗi số hạng là một *hạng tích* hay *mintex*, m_i .
- ✓ Dạng hội: Mỗi thừa số là *hạng tổng* hay *maxtex*, M_i .
- Dạng *chuẩn*: mỗi số hạng có đủ mặt các biến, là duy nhất.
- Tổng quát, hàm logic n biến dạng tổng các tích:

hoặc tích các tổng:

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

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

$a_i = '0'$ hoặc $'1'$. Đối với một hàm: *mintex* là bù của *maxtex*.

Đại số Boole và các phương pháp biểu diễn hàm(6)

Có 3 phương pháp rút gọn hàm:

- ✓ Phương pháp đại số
- ✓ Phương pháp bảng Karnaugh
- ✓ Phương pháp Quine Mc. Cluskey

❖ Phương pháp đại số

- Dựa vào các định lý để đưa biểu thức về dạng tối giản.
- Ví dụ: Biến đổi hàm logic sau về dạng tối giản:

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

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

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

Đại số Boole và các phương pháp biểu diễn hàm(7)

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

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

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

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

Bài tập: Tối giản hàm sau theo phương pháp đại số:

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

Đại số Boole và các phương pháp biểu diễn hàm(8)

❖ Phương pháp bảng Karnaugh

- Rút gọn các hàm có số biến không vượt quá 5.
- Các bước tối thiểu hóa:
 - ✓ Gộp 2^i ô kề cận có giá trị '1' (hoặc '0') thành từng nhóm. Gộp các ô '0' được biểu thức hàm bù.
 - ✓ Thay mỗi nhóm bằng một hạng tích mới.
 - ✓ Cộng các hạng tích mới.
- Ví dụ: Tối giản hàm:

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

Kết quả: $f = AB + C$

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

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

$f_1 = AB$

$f_2 = C$

Đại số Boole và các phương pháp biểu diễn hàm(9)

❖ Phương pháp Quine Mc. Cluskey

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

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

Đại số Boole và các phương pháp biểu diễn hàm(10)

- Ví dụ: $f(A, B, C, D) = \sum(10, 11, 12, 13, 14, 15)$
- Bước 1: Lập bảng

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

- Bước 2: Nhóm hạng tích:

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

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

CÔNG LOGIC

Cổng logic(1)

- ❖ **Nội dung**
 - ✓ Các cổng logic và các tham số chính
 - ✓ Các họ cổng logic
 - ✓ Giao tiếp giữa các cổng logic cơ bản
- ❖ **Các cổng logic và các tham số chính**
 - ✓ Cổng logic cơ bản
 - ✓ Một số cổng ghép thông dụng
 - ✓ Logic dương và logic âm
 - ✓ Các tham số chính

Cổng logic(2)

❖ Cổng logic cơ bản: AND, OR, NOT

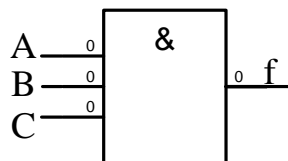
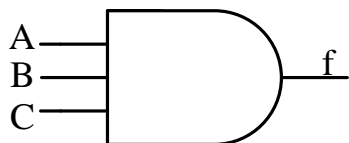
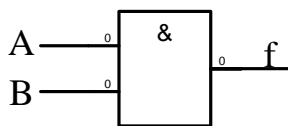
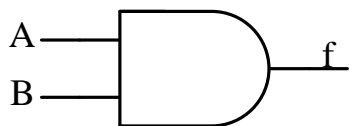
❖ Cổng AND

- 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

BTT 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

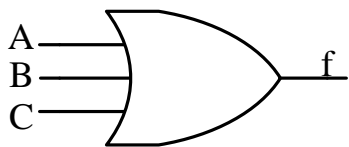
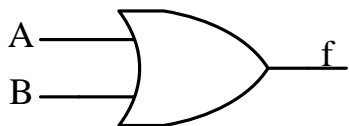
Cổng logic(4)

❖ Cổng OR

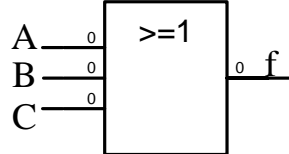
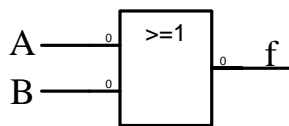
- Hàm ra của cổng OR 2 và nhiều biến vào như sau:

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

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

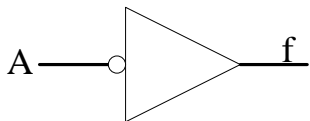
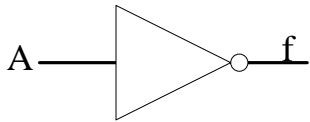
Theo mức logic

Cổng logic(6)

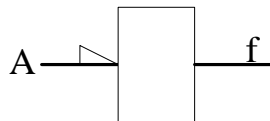
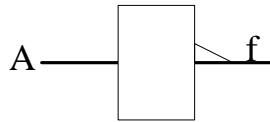
❖ Cổng NOT

- Hàm ra của cổng NOT: $f = \bar{A}$

Ký hiệu cổng NOT



Chuẩn ANSI



Chuẩn IEEE

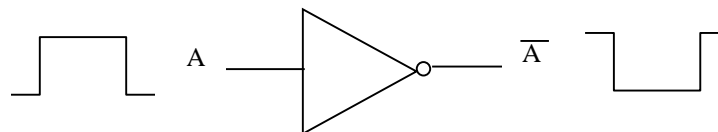
Bảng trạng thái cổng NOT

A	f
0	1
1	0

Theo giá trị logic

A	f
L	H
H	L

Theo mức logic



Dạng xung ra

Cổng logic(7)

❖ Một số cổng ghép thông dụng: NAND, NOR, XOR, XNOR

▪ **Cổng NAND**

✓ **NAND= AND + NOT**

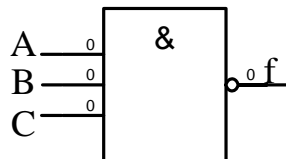
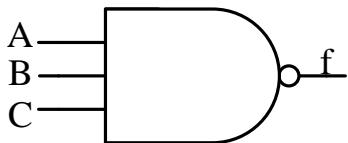
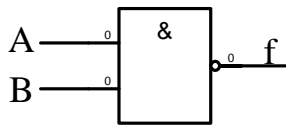
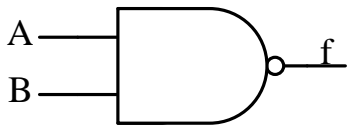
✓ **Hàm ra của cổng NAND:** $f = \overline{AB}$



$$f = \overline{AB}$$

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

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

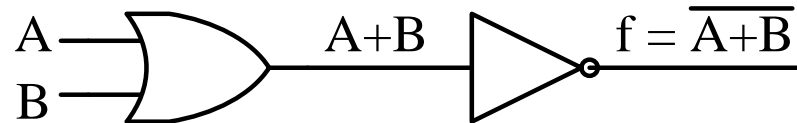
Theo giá trị logic

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

Theo mức logic

Cổng logic(8)

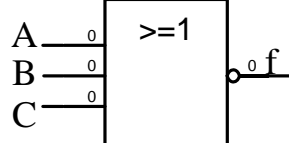
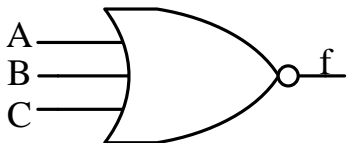
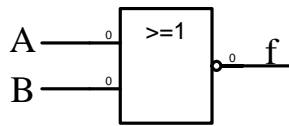
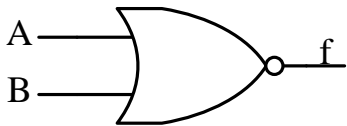
- **Cổng NOR:** NOR= OR+ NOT



- ✓ Hàm ra cổng NOR: $f = \overline{A + B}$

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

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

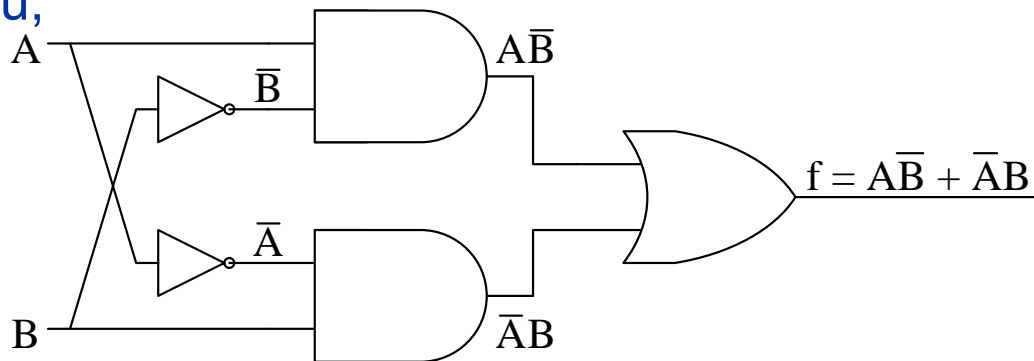
Cổng logic(9)

- **Cổng XOR:** (cổng khác dấu, cổng cộng modul 2).

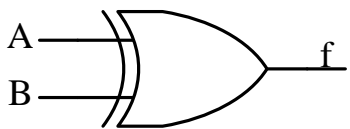
✓ Hàm ra của cổng XOR:

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

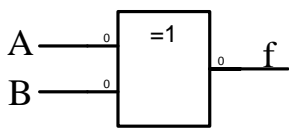
$$f = A \oplus B$$



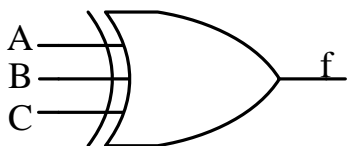
Ký hiệu cổng XOR



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

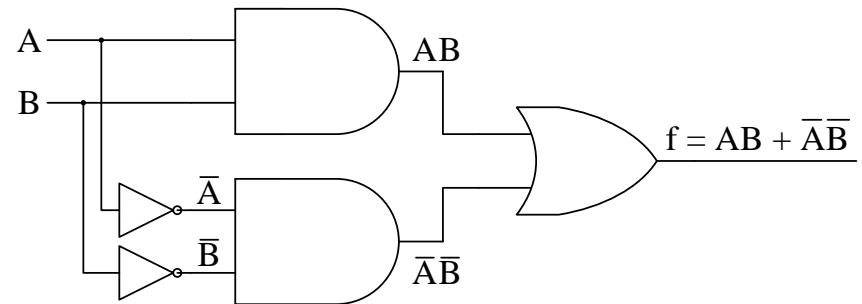
Theo mức logic

Cổng logic(10)

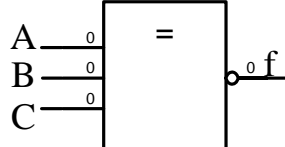
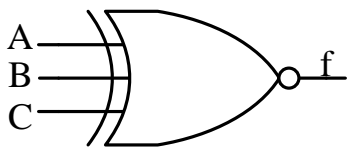
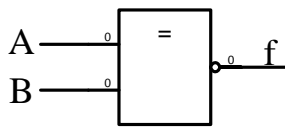
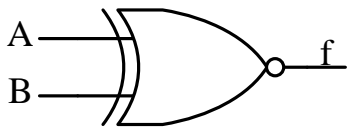
- ❖ **Cổng XNOR:** (cổng cùng dấu, cổng cộng modul 2).
- ❖ Hàm ra của cổng XNOR:

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

$$f = \overline{A \oplus B} = A \sim B$$



Ký hiệu cổng XNOR



Chuẩn ANSI

Chuẩn IEEE

Bảng trạng thái cổng XNOR 2 lối vào

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

Theo giá trị logic

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

Theo mức logic

MẠCH LOGIC TỔ HỢP

Mạch logic tổ hợp(1)

❖ Nội dung:

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

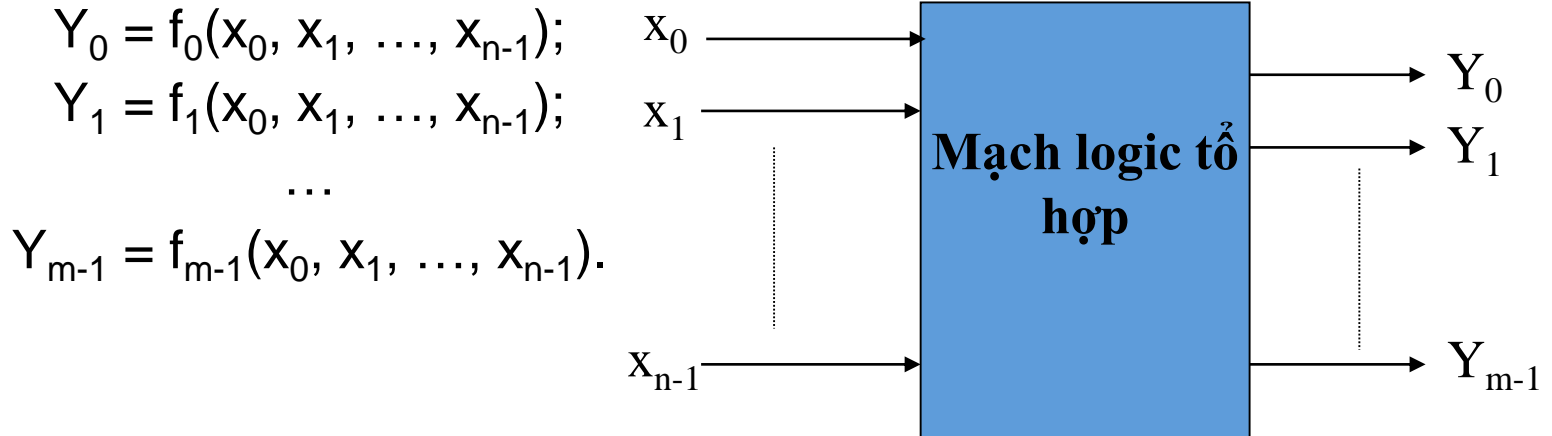
Mạch logic tổ hợp(2)

❖ Khái niệm chung

- Đặc điểm:
 - ✓ Tín hiệu đầu ra chỉ phụ thuộc các tín hiệu đầu vào nên trạng thái ra chỉ tồn tại trong thời gian có tác động vào
 - ✓ Được tạo ra từ các cổng logic.
- Phương pháp biểu diễn chức năng logic
 - ✓ Hàm số logic, bảng trạng thái, bảng Cacar nô (Karnaugh), cũng có khi biểu thị bằng đồ thị thời gian dạng xung.
 - ✓ Đối với vi mạch cỡ nhỏ (SSI) thường biểu diễn bằng hàm logic.
 - ✓ Đối với vi mạch cỡ vừa (MSI) thường biểu diễn bằng bảng trạng thái.

Mạch logic tổ hợp(3)

- Mạch logic tổ hợp có thể có n lối vào và m lối ra. Mỗi lối ra là một hàm của các biến vào.



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

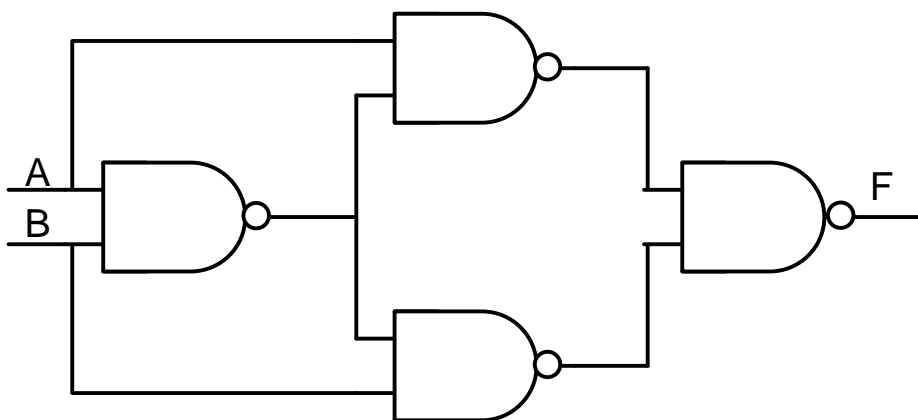
Mạch logic tổ hợp(4)

❖ Phân tích mạch logic tổ hợp

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

Mạch logic tổ hợp(5)

• Ví dụ: Cho mạch logic tổ hợp như hình vẽ:



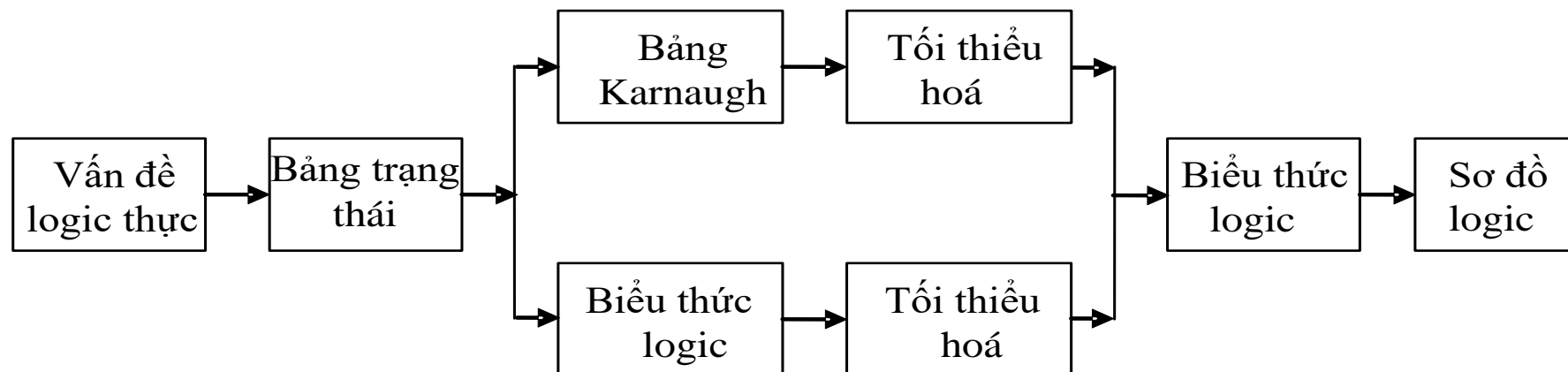
A	B	$F=A\oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

- Từ hình vẽ, ta lập được bảng TT.
- Hàm F tương đương với đầu ra cổng cộng khác dấu

Mạch logic tổ hợp(6)

❖ Thiết kế mạch logic tổ hợp:

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



Mạch logic tổ hợp(7)

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

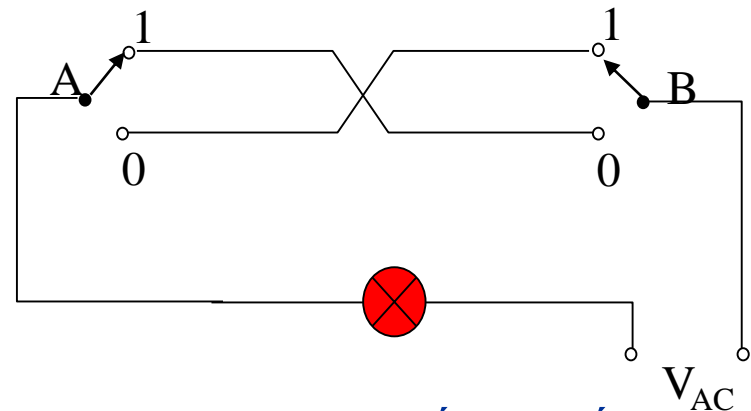
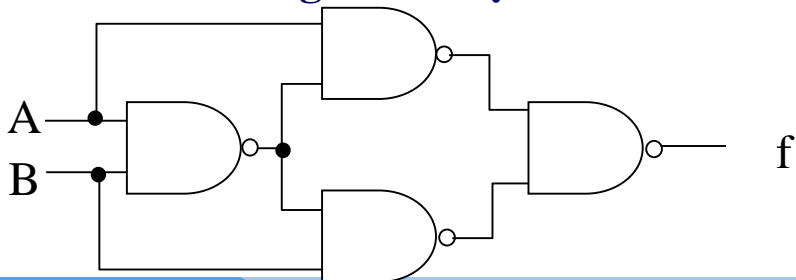
- ❖ **Lời giải:**

Hệ thống chiếu sáng như sơ đồ

Biểu thức của hàm là:

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

Sơ đồ logic thể hiện hàm f



Mạch điện của hệ thống chiếu sáng

Bảng trạng thái

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

Mạch logic tổ hợp(8)

❖ Mạch mã hóa:

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

Mạch logic tổ hợp(9)

❖ **Mạch giải mã:**

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

Mạch logic tổ hợp(10)

❖ MÃ BCD (Binary Coded Decimal):

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

❖ Mã Dư-3

- Cấu tạo: = BCD + 3 vào mỗi tổ hợp mã. Như vậy, mã không bao gồm tổ hợp toàn Zero.
- Ứng dụng: để truyền dẫn tín hiệu, không dùng tính toán trực tiếp.

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

Mạch logic tổ hợp(11)

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

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

Mạch logic tổ hợp(12)

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

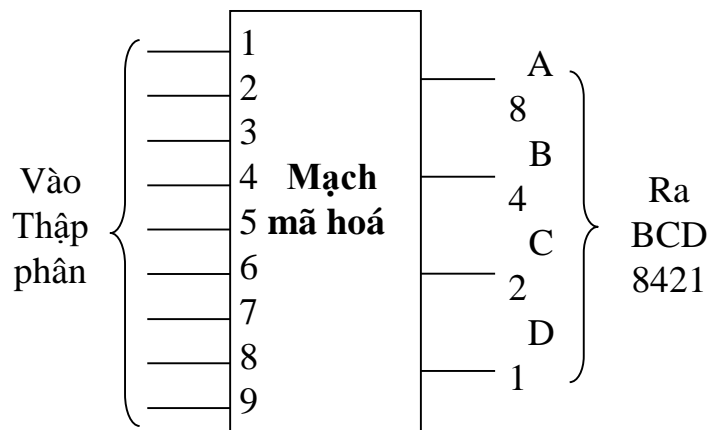
BCD 8421	BCD 8421chẵn P_C		BCD 8421 lẻ P_L	
0000	0000	0	0000	1
0001	0001	1	0001	0
0010	0010	1	0010	0
0011	0011	0	0011	1
0100	0100	1	0100	0
0101	0101	0	0101	1
0110	0110	0	0110	1
0111	0111	1	0111	0
1000	1000	1	1000	0
1001	1001	0	1001	1

Mạch logic tổ hợp(13)

❖ Mạch mã hoá từ thập phân sang BCD 8421

- 9 lối vào (biến), 4 lối ra A, B, C, D (hàm) thể hiện tổ hợp mã tương ứng với mỗi chữ số thập phân trên lối vào theo trọng số 8421.

Sơ đồ khối của mạch mã hóa



Bảng trạng thái

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

■ Từ bảng trạng thái ta viết được các hàm ra:

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

Mạch logic tổ hợp(14)

❖ Mạch mã hoá từ thập phân sang BCD 8421:

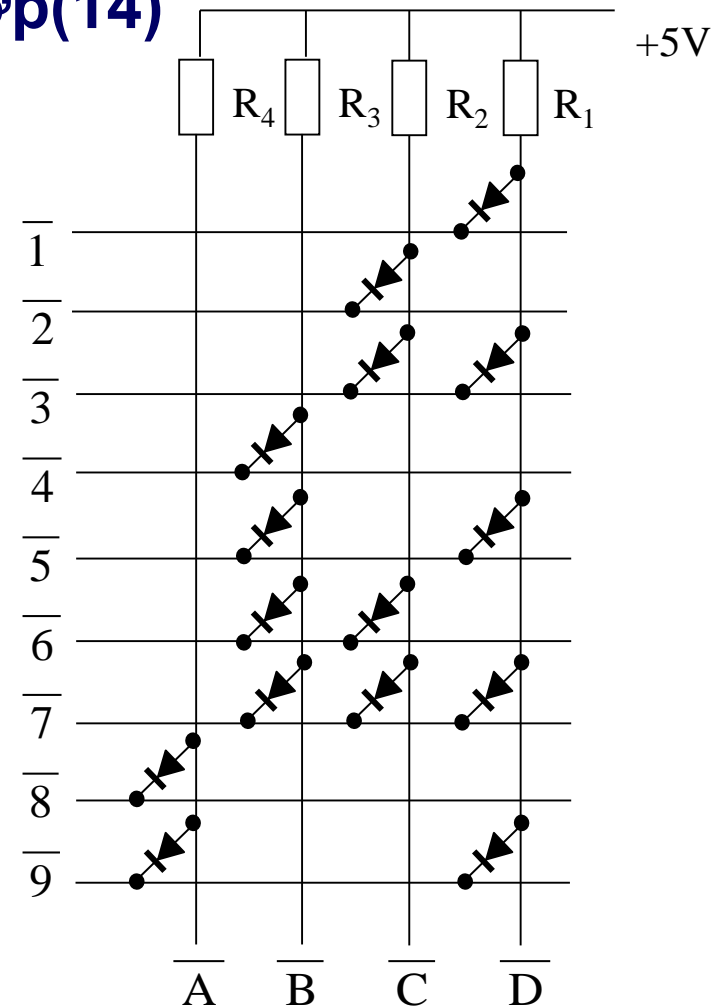
- Hoặc dùng ma trận diode (cổng OR)
- Hoặc có thể được viết lại như sau (dùng định lý DeMorgan) và dùng ma trận diode (cổng AND)

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

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

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

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



Mạch điện của bộ mã hoá dùng diode

Mạch logic tổ hợp(15)

❖ Mạch mã hóa ưu tiên:

- Trong bộ mã hoá vừa xét trên, tại một thời điểm chỉ có 1 tín hiệu đầu vào tác động.
- Để giải quyết trường hợp có nhiều đầu vào tác động đồng thời ta có bộ mã hoá ưu tiên. Trong các trường hợp này thì bộ mã hoá ưu tiên chỉ tiến hành mã hoá tín hiệu vào nào có cấp ưu tiên cao nhất ở thời điểm xét. Việc xác định cấp ưu tiên cho mỗi tín hiệu vào là do người thiết kế mạch.
- Xét nguyên tắc hoạt động và quá trình thiết kế của bộ mã hoá ưu tiên thập phân – nhị phân 9 lối vào, 4 lối ra.

Mạch logic tổ hợp(16)

- D sẽ lấy logic 1 ứng với đầu vào là 1, 3, 5, 7, 9. Tuy nhiên, lỗi vào 1 chỉ hiệu lực khi tất cả các lỗi vào cao hơn đều bằng 0...

D="1" nếu

- 1="1" và 2,4,6,8 bằng "0"
- 3="1" và 4,6,8 bằng "0"
- 5="1" và 6,8 bằng "0"
- 7="1" và 8 bằng "0"
- 9="1"

- Tương tự:

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

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

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

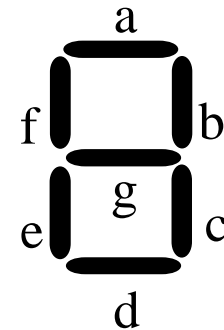
$$A = 8 + 9$$

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

Mạch logic tổ hợp(17)

❖ Mạch giải mã 7 đoạn

- Dụng cụ 7 đoạn
- ✓ Để hiển thị chữ số của một hệ đếm phân bất kỳ.
- ✓ Các đoạn phải có khả năng hiển thị trong các điều kiện ánh sáng khác nhau và tốc độ chuyển mạch phải đủ lớn. Trong kỹ thuật số, các đoạn thường được dùng là LED hoặc LCD.
- ✓ Đối với LED, mỗi đoạn là một Diode phát quang và khi có dòng điện đi qua đủ lớn (5 đến 30 mA) thì đoạn tương ứng sẽ sáng.
- ✓ Ngoài 7 đoạn sáng chính, mỗi LED cũng có thêm Diode để hiển thị dấu phân số khi cần thiết. LED có hai loại chính: LED Anốt chung và Ktốt chung, logic của tín hiệu điều khiển hai loại này là ngược nhau.

Cấu tạo dụng cụ 7
đoạn sáng

Mạch logic tổ hợp(18)

❖ Mạch giải mã 7 đoạn

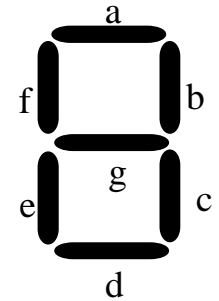
- 4 lối vào và 7 lối ra.
- đoạn a sẽ sáng khi hiển thị chữ số : 0 hoặc 2, hoặc 3, hoặc 5, hoặc 7, hoặc 8, hoặc 9.

$$a = \sum (0,2,3,5,6,7,8,9).$$
- Tương tự:

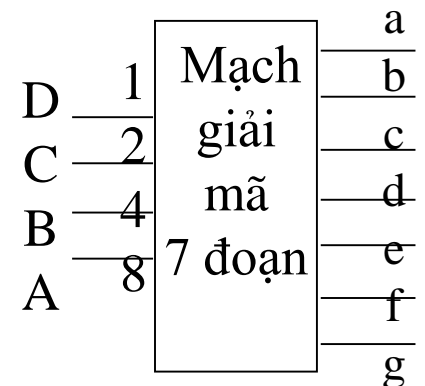
$$b = \sum (0,1,2,3,4,7,8,9), \quad c = \sum (0,1,3,4,5,6,7,8,9),$$

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

$$f = \sum (0,4,5,6,8,9), \quad g = \sum (2,3,4,5,6,8,9).$$
- IC 7447, 74247 (Anốt chung), 7448 (K chung), 4511 (CMOS) là các IC giải mã từ NBCD sang thập phân theo phương pháp hiển thị 7 đoạn.



Dụng cụ 7 đoạn
sáng



Sơ đồ khối

Mạch logic tổ hợp(19)

❖ Mạch giải mã nhị phân:

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



Sơ đồ khối của bộ giải mã nhị phân

- Số nhị phân là n bit (n lối vào) sẽ nhận diện được 2^n địa chỉ khác nhau (trên 2^n lối ra). Nói khác đi, mạch chọn địa chỉ nhị phân là một mạch logic tổ hợp có n lối vào và 2^n lối ra, nếu tác động tới đầu vào một số nhị phân thì chỉ duy nhất một lối ra được lựa chọn, lấy giá trị 1 (tích cực cao) hoặc 0 (tích cực thấp), các lối ra còn lại đều không được lựa chọn, lấy giá trị 0 hoặc 1.

Mạch logic tổ hợp(20)

❖ Bộ hợp kênh và phân kênh:

■ Bộ hợp kênh (MUX-Multiplexer)

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

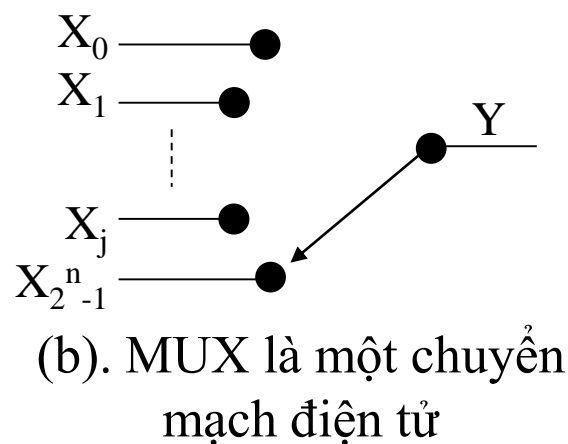
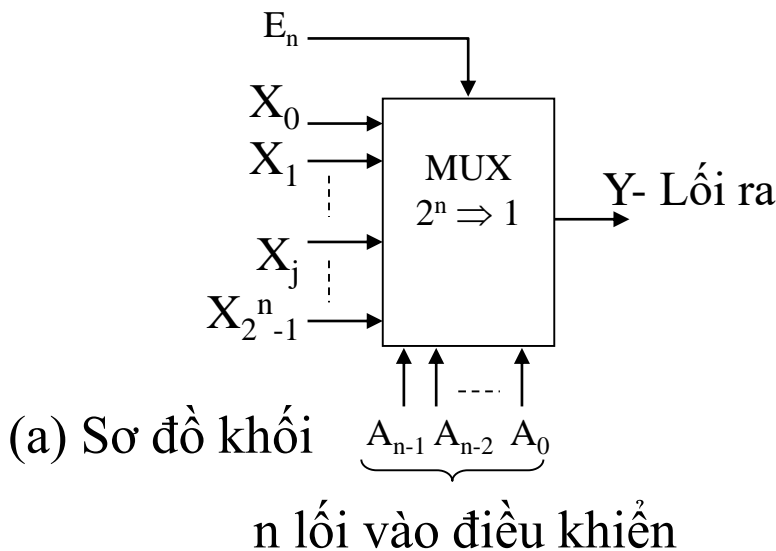
■ Bộ phân kênh (DEMUX-DeMultiplexer)

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

Mạch logic tổ hợp(21)

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

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



- MUX là chuyển mạch điện tử dùng các tín hiệu điều khiển để điều khiển sự nối mạch của lối ra với 1 trong số 2^n lối vào.
- MUX được dùng như 1 phần tử vạn năng để xây dựng những mạch tổ hợp khác.

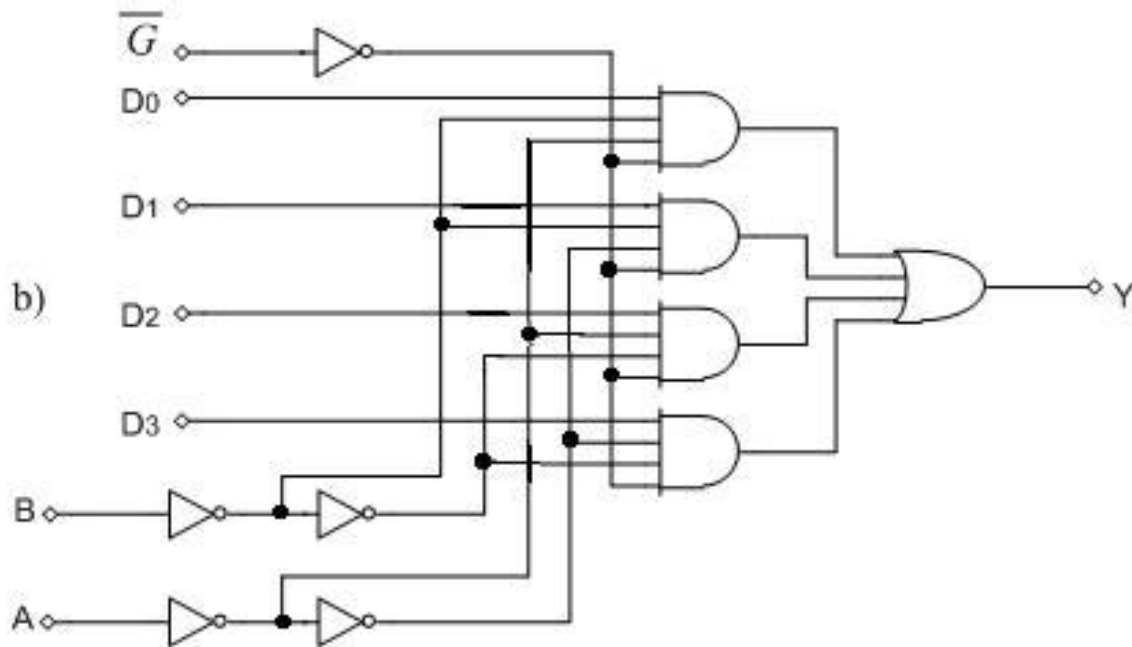
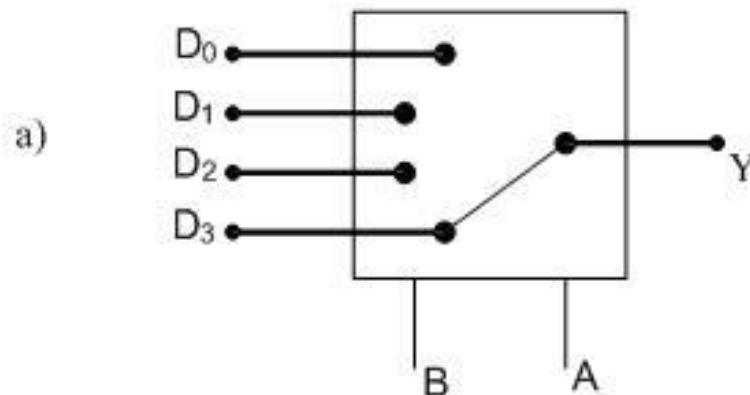
- ❖ MUX 2 lối vào điều khiển, 4 lối vào dữ liệu:

\overline{G} : CS

- $\overline{G} = 1$: Y cấm làm việc ($Y=0$)
- $\overline{G} = 0$:

$$Y = \overline{B}.\overline{A}.D_0 + \overline{B}.A.D_1 + B.\overline{A}.D_2 + B.A.D_3$$

- Kênh được chọn phụ thuộc tín hiệu điều khiển

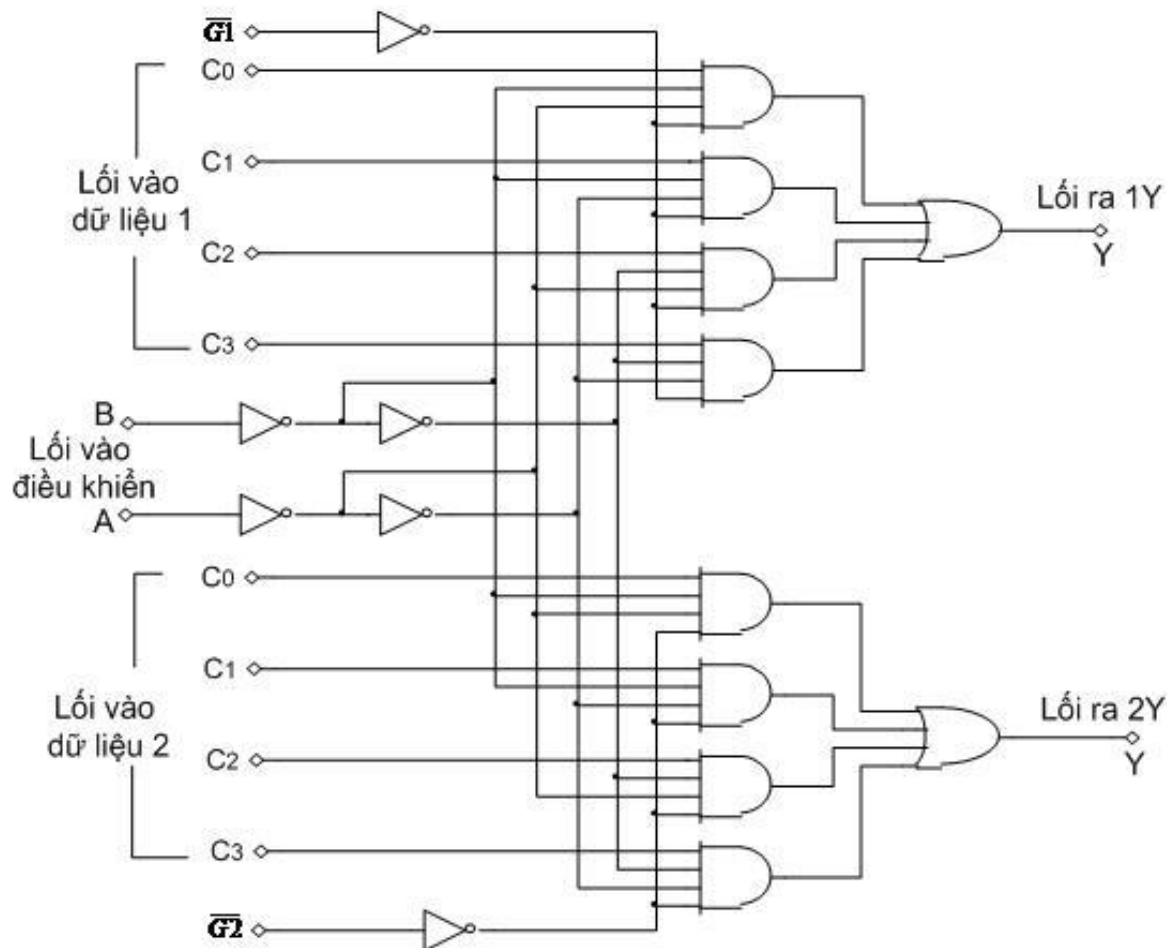


- ❖ Nếu $BA = 00$ thì $Y = D_0$.
- ❖ Nếu $BA = 01$ thì $Y = D_1$.
- ❖ Nếu $BA = 10$ thì $Y = D_2$.
- ❖ Nếu $BA = 11$ thì $Y = D_3$.

- ❖ Đây là bảng chức năng của bộ ghép kênh

B	A	\overline{G}	Y
X	X	H	L
L	L	L	D_0
L	H	L	D_1
H	L	L	D_2
H	H	L	D_3

- ❖ **IC 74LS153**: gồm 2 bộ ghép kênh 4 đường dữ liệu (C_0, C_1, C_2, C_3), 1 đường ra Y . Hai bộ ghép kênh có chung 2 đầu vào điều khiển A, B , mỗi bộ ghép kênh đều có đầu vào cho phép riêng. Mạch thuộc họ logic TTL, chân 16 là nguồn nuôi VCC : +5V, chân 8 là đất (GND): 0V



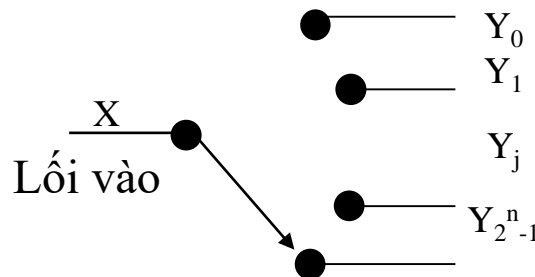
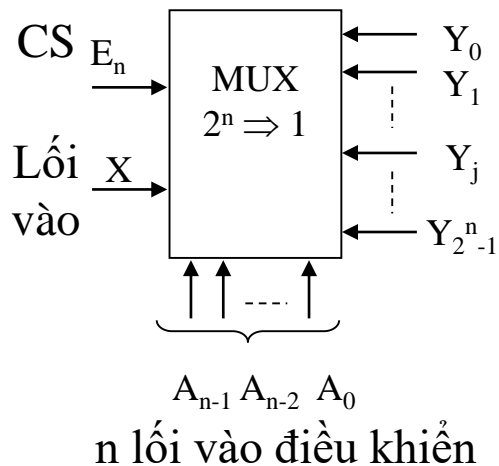
❖ **IC 74LS151:**
MUX 8 vào -
1 ra.

Đây là bảng
trạng thái của
IC 74LS151

Đầu vào điều khiển			Cho phép	Đầu ra
C	B	A		Y
L	L	L	L	D ₀
L	L	H	L	D ₁
L	H	L	L	D ₂
L	H	H	L	D ₃
H	L	L	L	D ₄
H	L	H	L	D ₅
H	H	L	L	D ₆
H	H	H	L	D ₇
x	x	x	H	L

Mạch logic tổ hợp(22)

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



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

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

.....

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

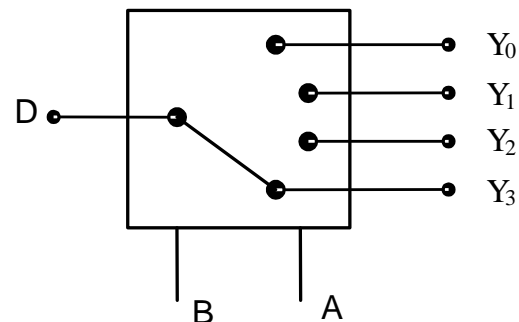
(a) Sơ đồ khối (b). DEMUX là một chuyển mạch điện tử

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

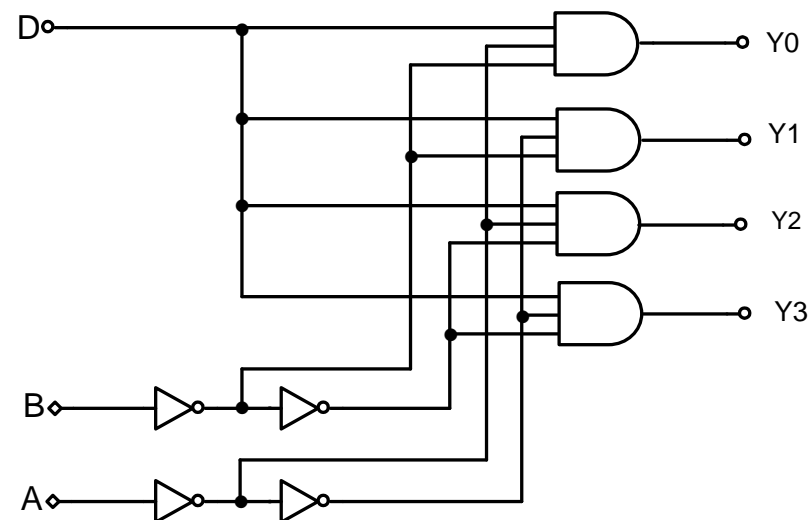
❖ DEMUX 2 lối vào điều khiển, 1 vào và 4 ra dữ liệu:

B	A	Y_0	Y_1	Y_2	Y_3
0	0	D	0	0	0
0	1	0	D	0	0
1	0	0	0	D	0
1	1	0	0	0	D

a)



b)



$$Y_0 = \bar{A} \cdot \bar{B} \cdot D, Y_1 = A \cdot \bar{B} \cdot D$$

$$Y_2 = \bar{A} \cdot B \cdot D, Y_3 = A \cdot B \cdot D$$

Mạch logic tổ hợp(23)

- ❖ **Mạch bán tổng (Half Adder - HA)**
- ❖ Mạch cộng bán phần có 2 đầu vào:
 - A_i là chữ số cột thứ i của số A .
 - B_i là chữ số cột thứ i của số B .
- ❖ Mạch có 2 đầu ra:
 - S_i là kết quả phép cộng ở cột thứ i .
 - C_i là giá trị nhớ sang cột có trọng số cao hơn kế tiếp.

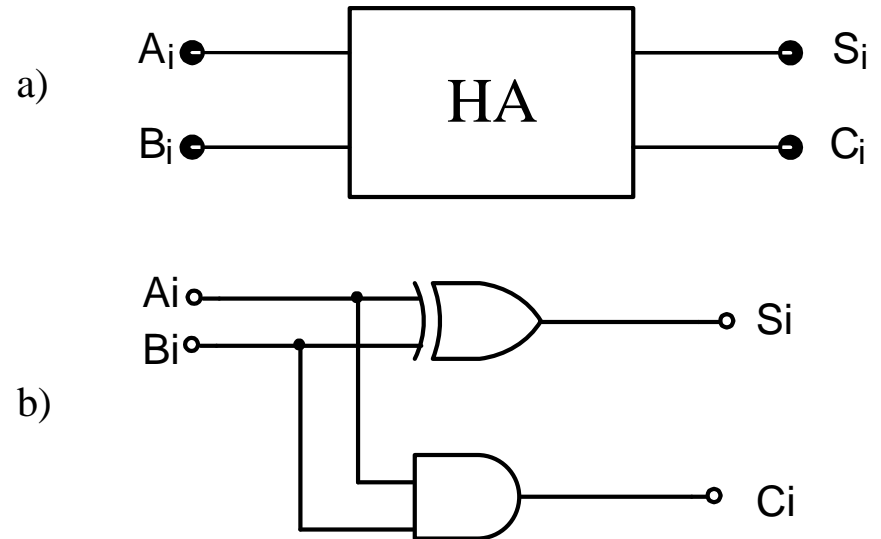
A_i	B_i	S_i	C_i
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Mạch logic tổ hợp(24)

- ❖ Hình vẽ là sơ đồ khối và sơ đồ logic của HA.
- ❖ Ta có:

$$S_i = A_i \oplus B_i$$

$$C_i = A_i \cdot B_i$$



Mạch logic tổ hợp(25)

- ❖ **Mạch toàn tổng (Full Adder - FA):** Mạch logic thực hiện phép cộng hai số nhị phân 1 bit có lối nhớ đầu vào
- ❖ Mạch có 3 đầu vào:
 - Ai là chữ số cột thứ i của số A.
 - Bi là chữ số cột thứ i của số B.
 - Ci-1 là bit nhớ của trọng số nhỏ hơn liền kề chuyển đến
- ❖ Mạch có 2 đầu ra:
 - Si là kết quả phép cộng ở cột thứ i.
 - Ci là bit nhớ sang trọng số lớn hơn kế tiếp.
- ❖ Bảng trạng thái của FA được trình bày ở bảng sau:

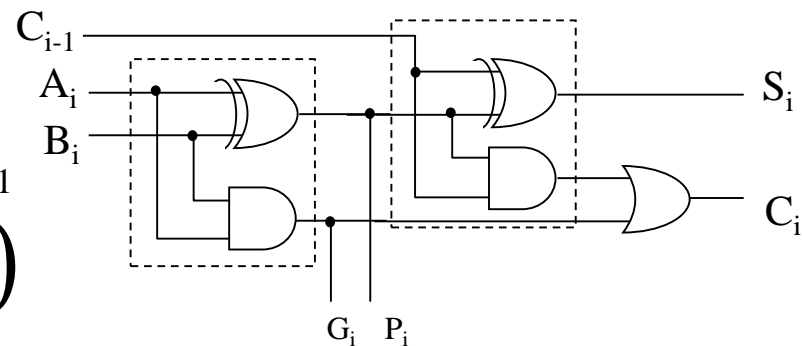
Mạch logic tổ hợp(26)

Bảng trạng thái				
C_{i-1}	A_i	B_i	S_i	C_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Mạch logic tổ hợp(27)

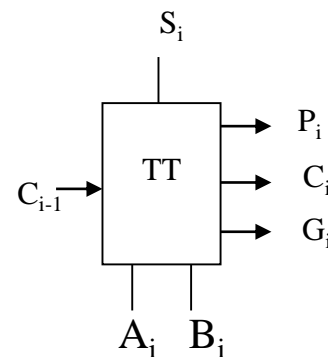
❖ Từ bảng trạng thái suy ra hàm logic của FA:

$$\begin{aligned}
 S_i &= \bar{A}_i \cdot \bar{B}_i C_{i-1} + \bar{A}_i B_i \bar{C}_{i-1} + A_i \bar{B}_i \bar{C}_{i-1} + A_i B_i C_{i-1} \\
 &= \bar{A}_i (\bar{B}_i C_{i-1} + B_i \bar{C}_{i-1}) + A_i (\bar{B}_i \bar{C}_{i-1} + B_i C_{i-1}) \\
 &= A_i \oplus B_i \oplus C_{i-1}
 \end{aligned}$$



a) Mạch điện

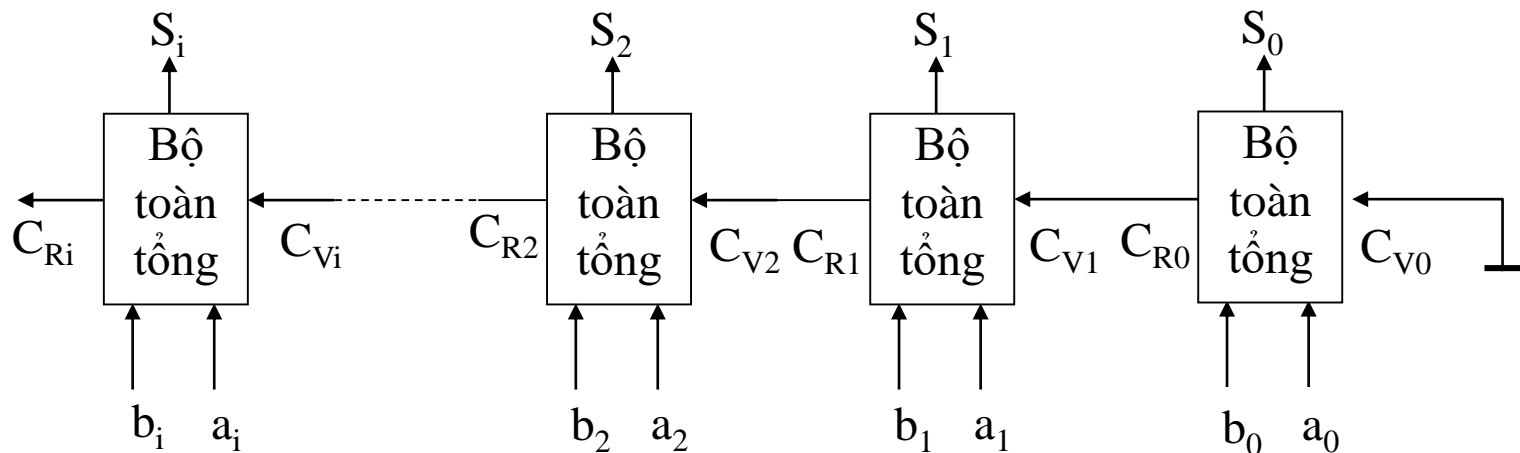
$$C_i = \bar{A}_i \cdot B_i C_{i-1} + A_i \bar{B}_i C_{i-1} + A_i B_i \bar{C}_{i-1} + A_i B_i C_{i-1}$$



b) Ký hiệu

Mạch logic tổ hợp(28)

- ❖ **Mạch cộng nhị phân song song:** ghép nhiều bộ cộng hai số nhị một bit lại với nhau để thực hiện phép cộng hai số nhị phân nhiều bit.



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

Mạch logic tổ hợp(29)

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

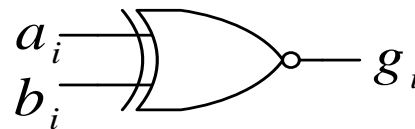
Mạch logic tổ hợp(30)

❖ Bộ so sánh bằng nhau 1 bit

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

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

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



Bảng trạng thái của bộ
so sánh bằng 1 bit

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

❖ Bộ so sánh bằng nhau 4 bit

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

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

Mạch logic tổ hợp(31)

❖ Bộ so sánh 1 bit

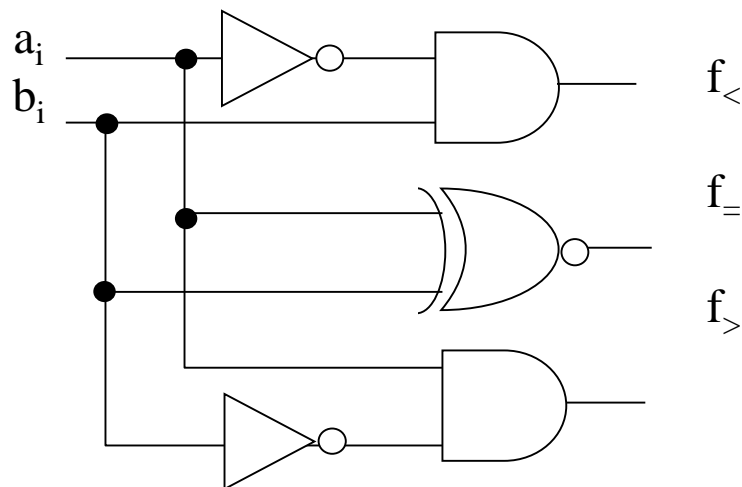
Bảng trạng thái của mạch so sánh				
a_i	b_i	$f_{<}$	$f_{=}$	$f_{>}$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0

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

❖ Biểu thức đầu ra: $f_{=} = \overline{a_i \oplus b_i}$

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

Mạch điện của bộ so sánh 1 bit



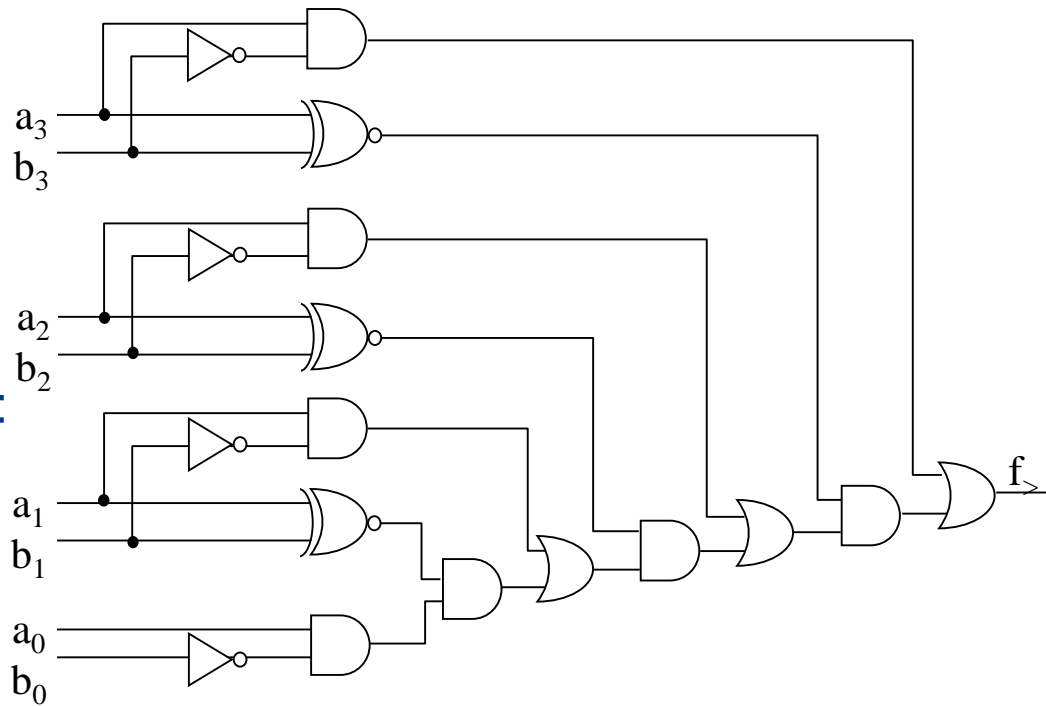
Mạch logic tổ hợp(32)

❖ **Bộ so sánh 4 bit:** $A = a_3a_2a_1a_0, B = b_3b_2b_1b_0$. $A > B$ 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 $a_0 > b_0$.

Từ đó ta có biểu thức hàm ra là:

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

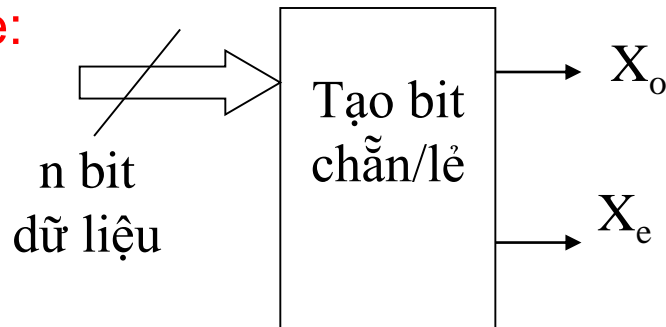


Mạch logic tổ hợp(33)

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

Mạch logic tổ hợp(34)

❖ Mạch tạo bit chẵn/lẻ:



❖ Xét trường hợp 3 bit dữ liệu d_1, d_2, d_3

❖ Gọi X_e, X_o là 2 bit chẵn, lẻ thêm vào dữ liệu.

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

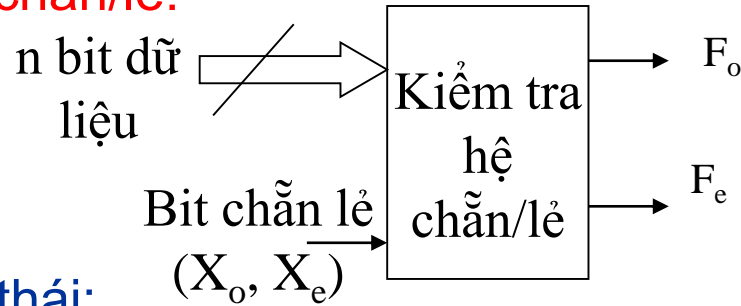
❖ Và biểu thức dựa vào d_1, d_2, d_3 là

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

Bảng trạng thái của mạch tạo bit chẵn lẻ				
Vào			Ra	
d_1	d_2	d_3	X_e	X_o
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	0

Mạch logic tổ hợp(35)

❖ Mạch kiểm tra chẵn/lẻ:



❖ Từ bảng trạng thái:

- $F_e = 1$ nếu hệ là chẵn (F_e chỉ ra tính chẵn của hệ).
 - $F_o = 1$ nếu hệ là lẻ (F_o chỉ ra tính lẻ của hệ).
- ❖ Hai hàm kiểm tra chẵn/lẻ luôn là phủ định của nhau. Do tính chất của hàm cộng XOR, ta có:
- $F_o = d_1 \oplus d_2 \oplus d_3 \oplus X$
 - $F_e = \overline{F_o}$

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

MẠCH LOGIC TUẦN TỰ

Mạch logic tuần tự(1)

❖ **Nội dung:**

- Khái niệm chung
- Phần tử nhớ trong mạch tuần tự
- Phương pháp mô tả mạch tuần tự
- Phân tích và thiết kế mạch tuần tự
- Mạch tuần tự đồng bộ
- Mạch tuần tự không đồng bộ

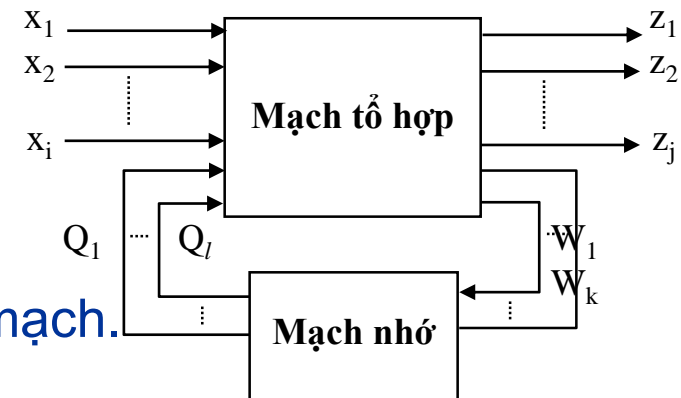
Mạch logic tuần tự(2)

❖ Khái niệm chung

- Còn gọi là mạch dãy - Sequential Circuit.
- Trạng thái của mạch phụ thuộc đầu vào và trạng thái bên trong trước đó của mạch. Nói cách khác các hệ thống này làm việc theo nguyên tắc có nhớ.

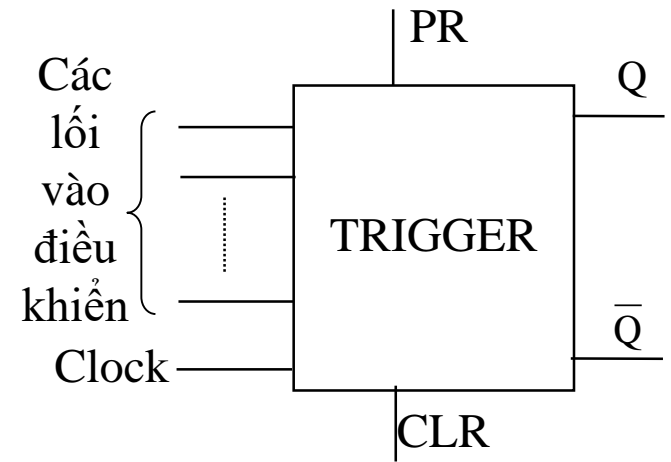
❖ Mô hình toán học

- $Z = f(Q, X)$
 - X - tập tín hiệu vào.
 - Q - tập trạng thái trong trước đó của mạch.
 - W - hàm kích và Z - các hàm ra
- Biểu diễn khác: $Z = f(Q(n), X)$; $Q(n+1) = f(Q(n), X)$
 - $Q(n+1)$: là trạng thái tiếp theo của mạch.
 - $Q(n)$: là trạng thái bên trong trước đó.



Mạch logic tuần tự(3)

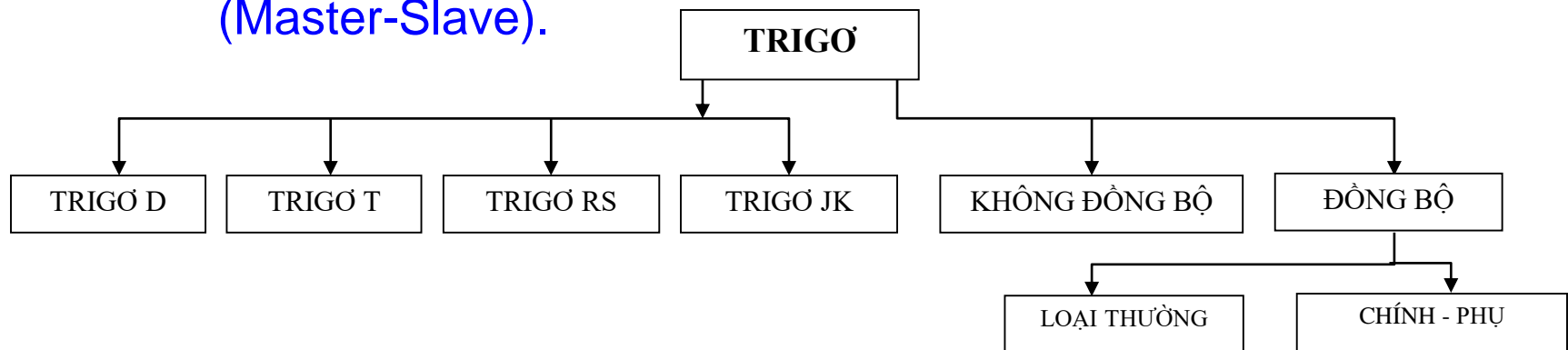
- ❖ **Trigger:** là phần tử nhớ một trong hai trạng thái 0 và 1.
- Trigo^o có từ 1 đến một vài lối điều khiển, có hai lối ra luôn luôn ngược nhau là Q và \bar{Q} . Tùy từng loại trigo^o có thể có thêm các lối vào lập (PRESET) và lối vào xóa (CLEAR). Ngoài ra, trigo^o còn có lối vào đồng bộ (CLOCK). Hình bên là sơ đồ khối tổng quát của trigo^o.



Mạch logic tuần tự(4)

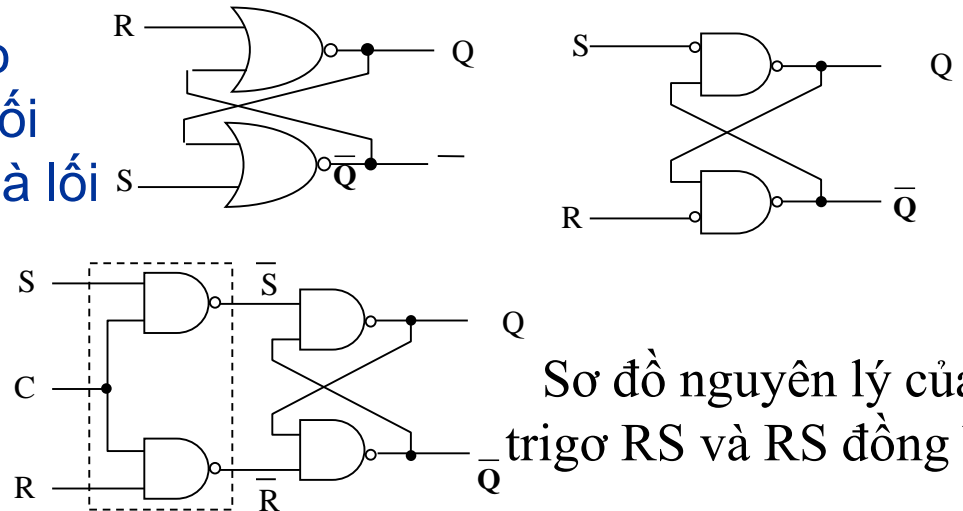
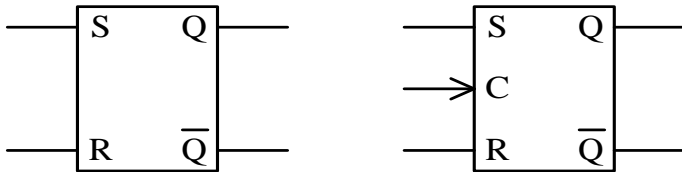
❖ Phân loại:

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



Mạch logic tuần tự(5)

- ❖ **Trigger RS**: là loại có hai lối vào điều khiển S, R. Chân S gọi là lối vào "lập" (SET) và R được gọi là lối vào "xóa" (RESET).



Sơ đồ nguyên lý của
trigơ RS và RS đồng bộ

Bảng TT của trigơ RS

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

Bảng TT của trigơ RS đồng bộ cổng
NAND

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

Mạch logic tuần tự(6)

❖ Tri gơ RS không đồng bộ:

Bảng trạng thái

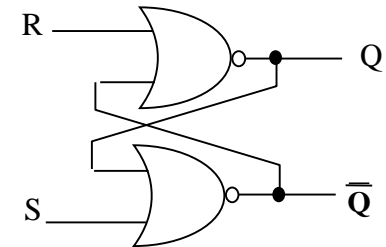
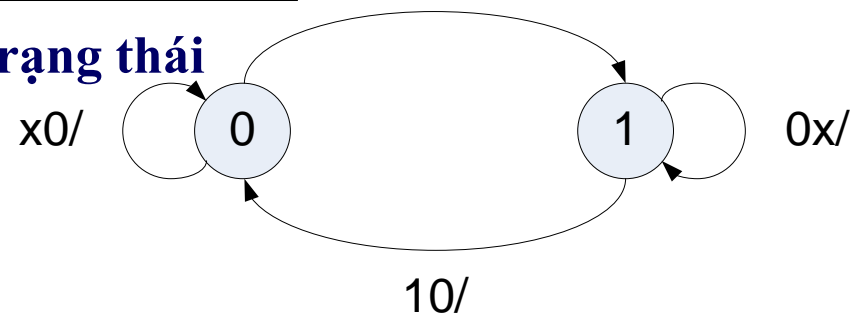
Q	R	S	Q ^k
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	X
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	X

Biểu thức:
$$\begin{cases} Q^K = S + \bar{R} \cdot Q \\ RS = 0 \text{ (Điều kiện để tránh tổ hợp cấm)} \end{cases}$$

Bảng Các nô

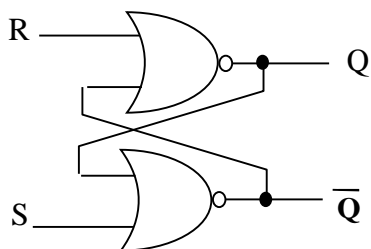
RS \ Q	00	01	11	10
0	0	1	X	0
1	1	1	X	0

Đồ hình trạng thái



Mạch logic tuần tự(7)

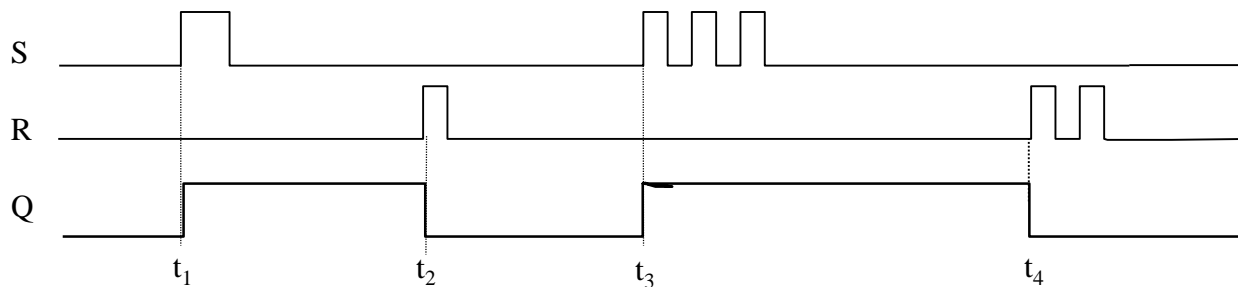
❖ Tri gơ RS không đồng bộ:



Bảng trạng thái

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

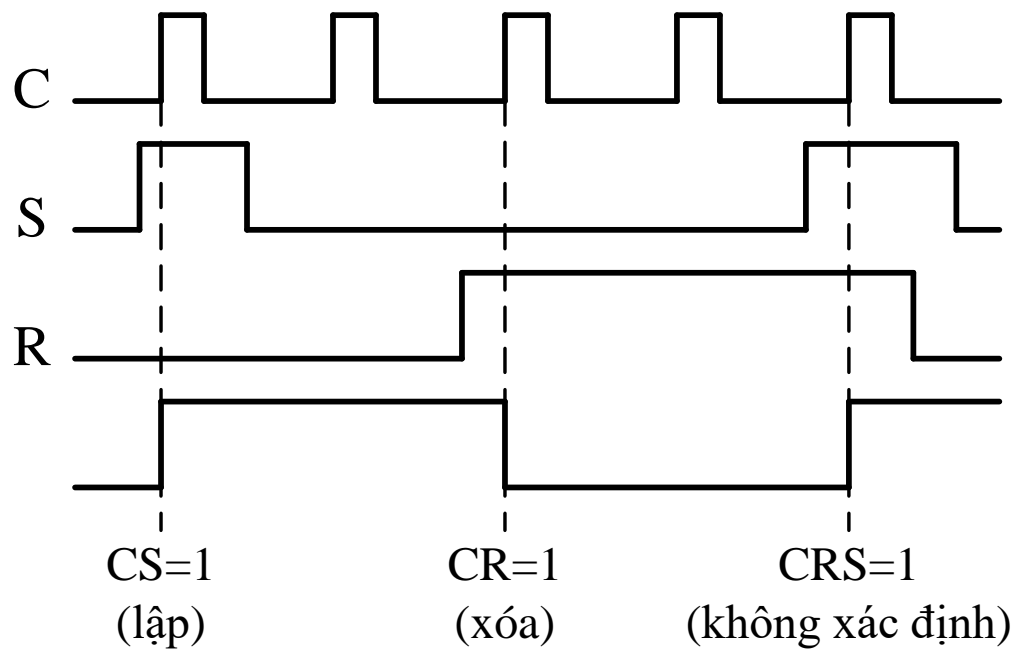
Đồ thị dạng xung:



Mạch logic tuần tự(8)

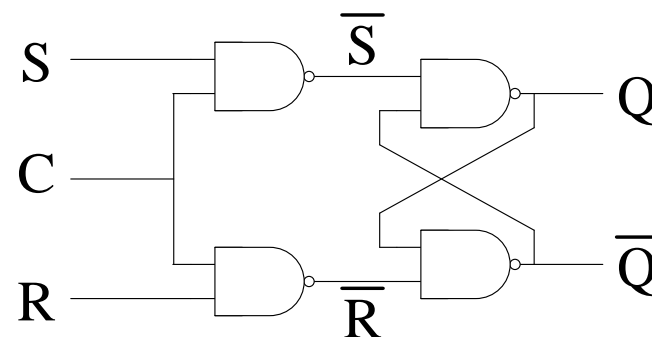
❖ Trơ RS đồng bộ

Đồ thị dạng xung



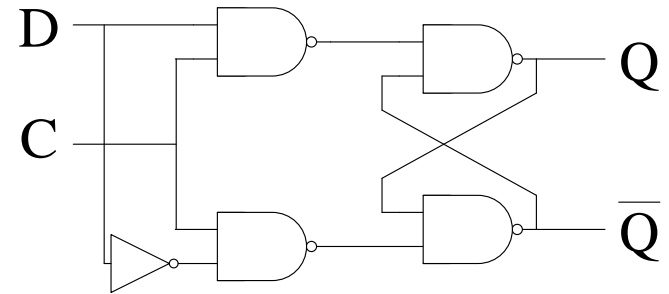
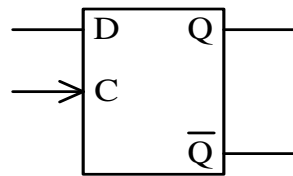
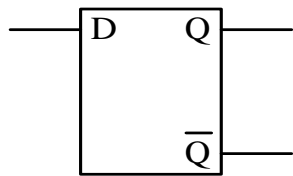
Bảng TT của trơ RS đồng bộ cổng NAND

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



Mạch logic tuần tự(9)

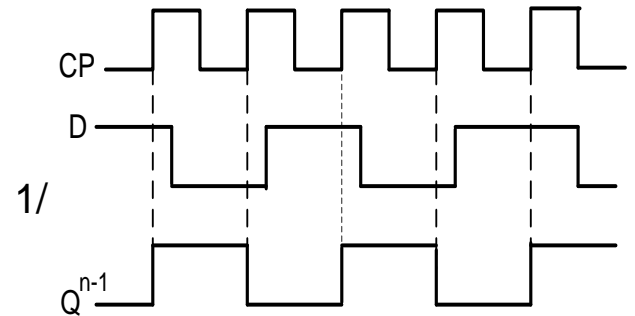
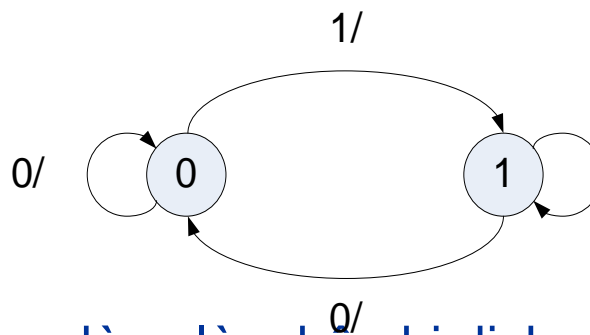
- ❖ **Trigơ D:** là loại trigơ có một lối vào điều khiển D.
- ❖ Biểu thức: $Q^k = D$, khi có sườn Clock.



Bảng trạng thái

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

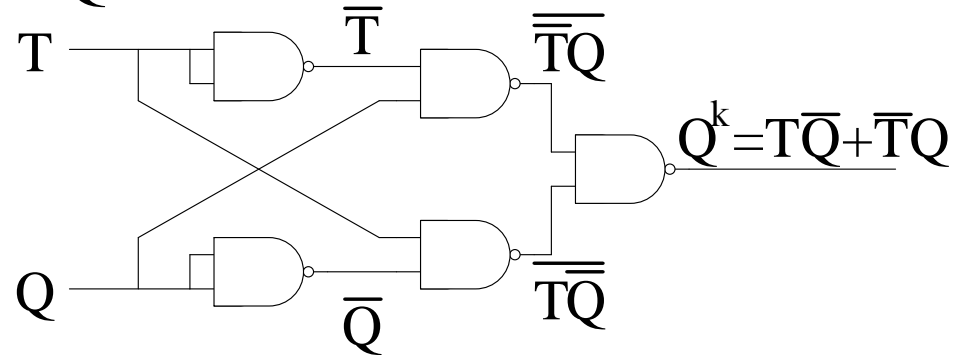
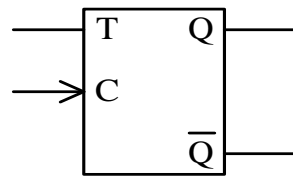
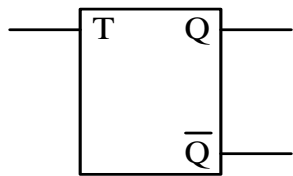
Đồ hình trạng thái



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

Mạch logic tuần tự(10)

- ❖ **Trigơ T:** là loại trigơ có một lối vào điều khiển T. Mỗi khi có xung tới lối vào T thì lối ra Q sẽ thay đổi trạng thái.
- ❖ Biểu thức: $Q^k = \bar{T}Q + T\bar{Q} = T \oplus Q$
- ❖ Sơ đồ khối:



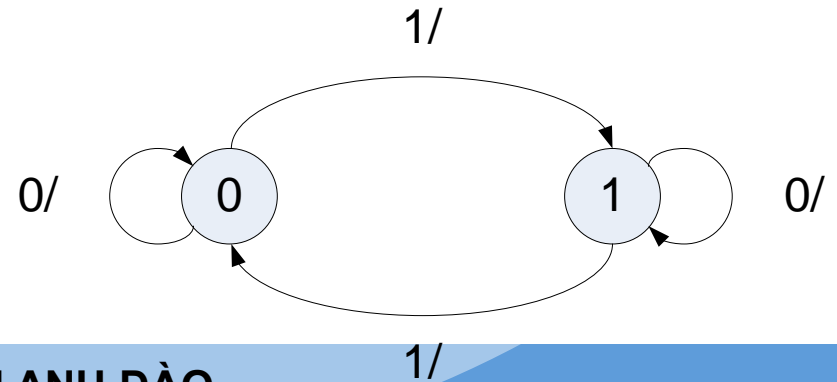
Bảng trạng thái

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

Bảng TT rút gọn

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

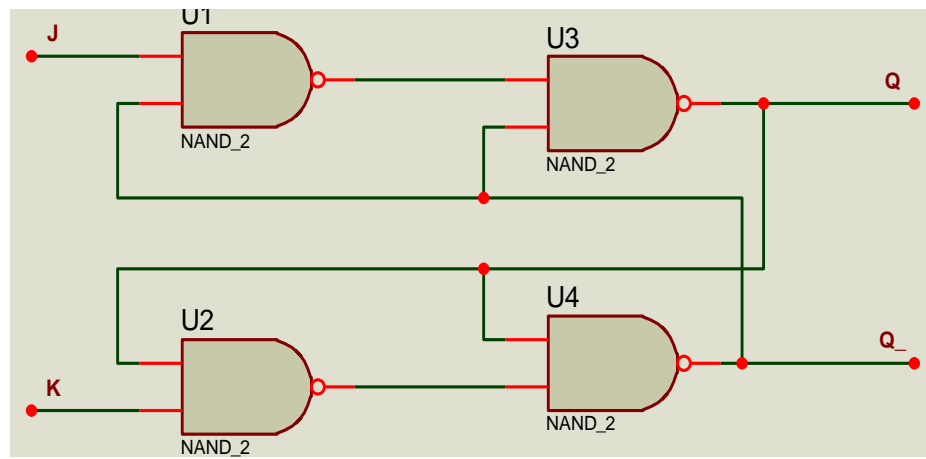
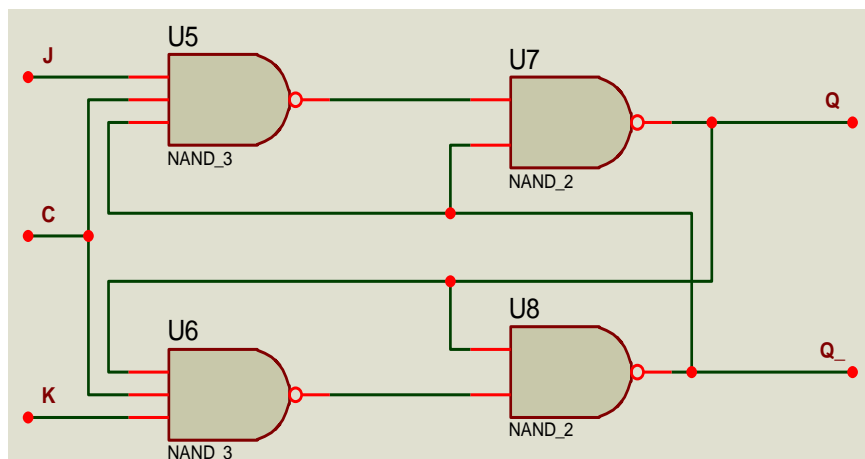
Đồ hình trạng thái



Mạch logic tuần tự(11)

❖ **Trigơ JK:** là loại trigơ có hai lối vào điều khiển J, K.

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



Mạch logic tuần tự(12)

Bảng TT đầy đủ			
J	K	Q	Q^k
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Bảng TT rút gọn		
J	K	Q^k
0	0	Q
0	1	0
1	0	1
1	1	Q'

Bảng TT của trigơ JK đồng bộ			
C	J	K	Q^k
0	X	X	Q
1	0	0	Q
	0	1	0
	1	0	1
	1	1	Q'

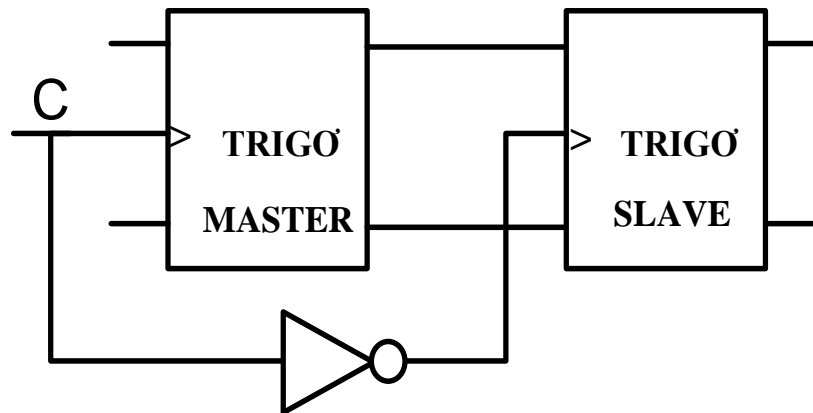
Mạch logic tuần tự(13)

Bảng hàm kích thích của các loại Trigo

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

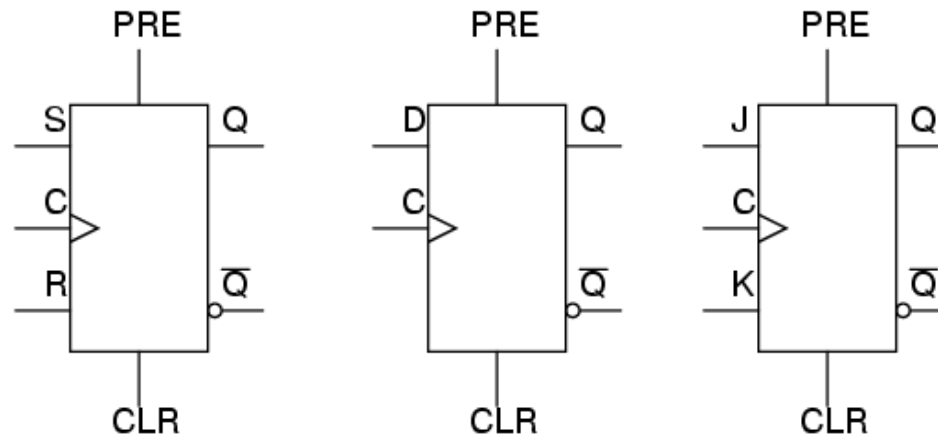
Mạch logic tuần tự(14)

- ❖ **Trigơ Chính-Phụ (Master-Slave):** Do các loại trigơ đồng bộ trên đều hoạt động tại sườn dương hay sườn âm của xung nhịp nên khi làm việc ở tần số cao thì lối ra Q không đáp ứng kịp với sự thay đổi của xung nhịp, nên mạch hoạt động ở tình trạng không được tin cậy.
- ❖ Lối ra của trigơ MS thay đổi tại sườn dương và sườn âm của xung nhịp. Cấu trúc của 2 trigơ giống nhau xung Clock ngược nhau để đảm bảo sao cho tại mỗi sườn của xung sẽ có một trigơ hoạt động.



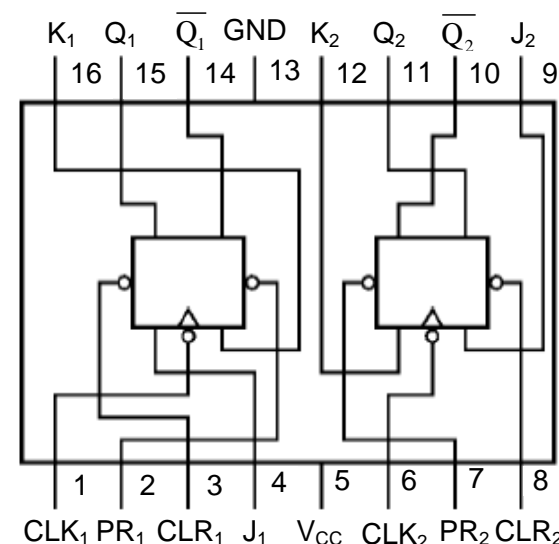
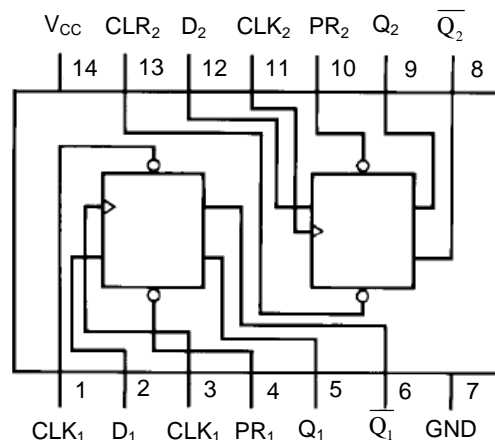
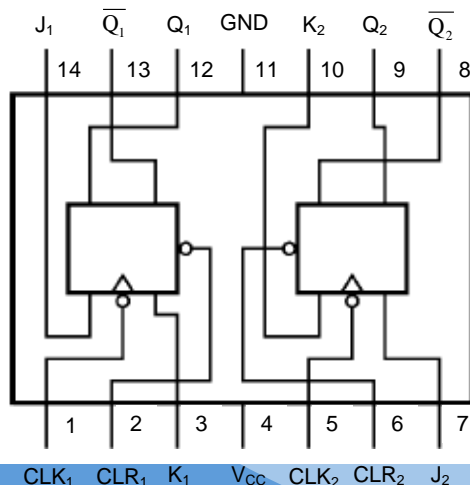
Mạch logic tuần tự(15)

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



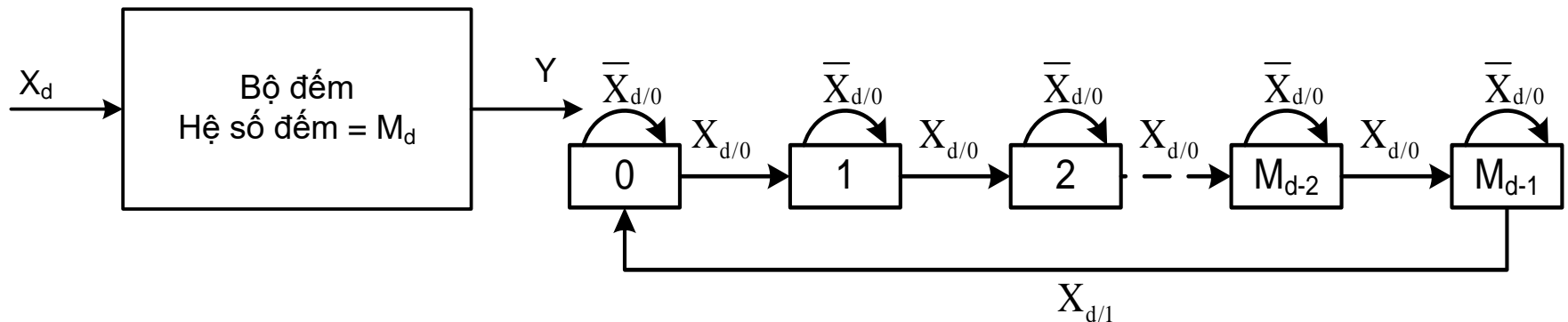
Mạch logic tuần tự(16)

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



Mạch logic tuần tự(38)

- ❖ **Bộ đếm:** là mạch tuần tự tuần hoàn có một lối vào đếm và một lối ra, mạch có số trạng thái trong bảng chính hệ số đếm (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.
- ❖ Sau M_d tín hiệu vào đếm mạch lại trở về trạng thái xuất phát ban đầu.
- ❖ Được dùng nhiều trong các dụng cụ đo lường chỉ thị số, các máy tính điện tử. Bất kỳ hệ thống số hiện đại nào đều sử dụng các bộ đếm.



Mạch logic tuần tự(39)

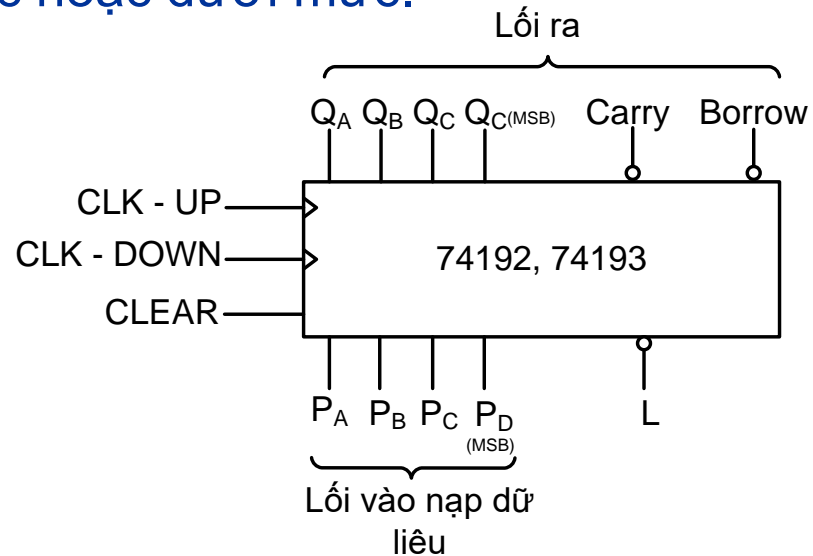
- ❖ **Phân loại bộ đếm**
- ❖ Theo sự chuyển đổi trạng thái:
 - Bộ đếm đồng bộ (Synchronous): Các trigơ đều chịu tác dụng điều khiển của một xung đồng hồ duy nhất
 - Bộ đếm không đồng bộ (Asynchronous): có trigơ chịu tác dụng điều khiển trực tiếp của xung đếm đầu vào, nhưng cũng có trigơ chịu tác dụng điều khiển của xung ở đầu ra của trigơ khác .
- ❖ Theo hệ số đếm
 - Bộ đếm nhị phân
 - Bộ đếm thập phân
 - Bộ đếm N phân
- ❖ Theo xung đếm
 - Bộ đếm thuận (Up counter) hay còn gọi là bộ đếm tiến
 - Bộ đếm nghịch (Down counter) hay còn gọi là bộ đếm lùi
 - Bộ đếm thuận nghịch

Mạch logic tuần tự(40)

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

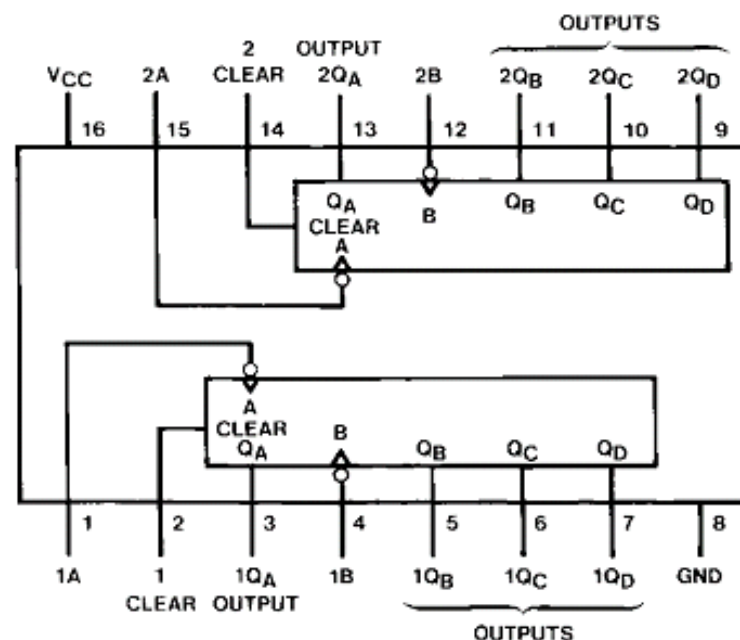
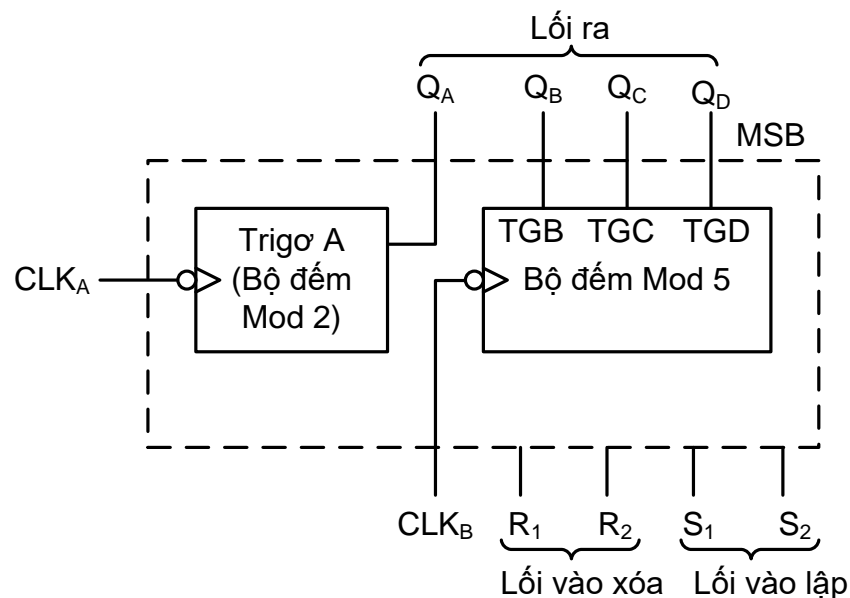
Mạch logic tuần tự(41)

- ❖ IC 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.



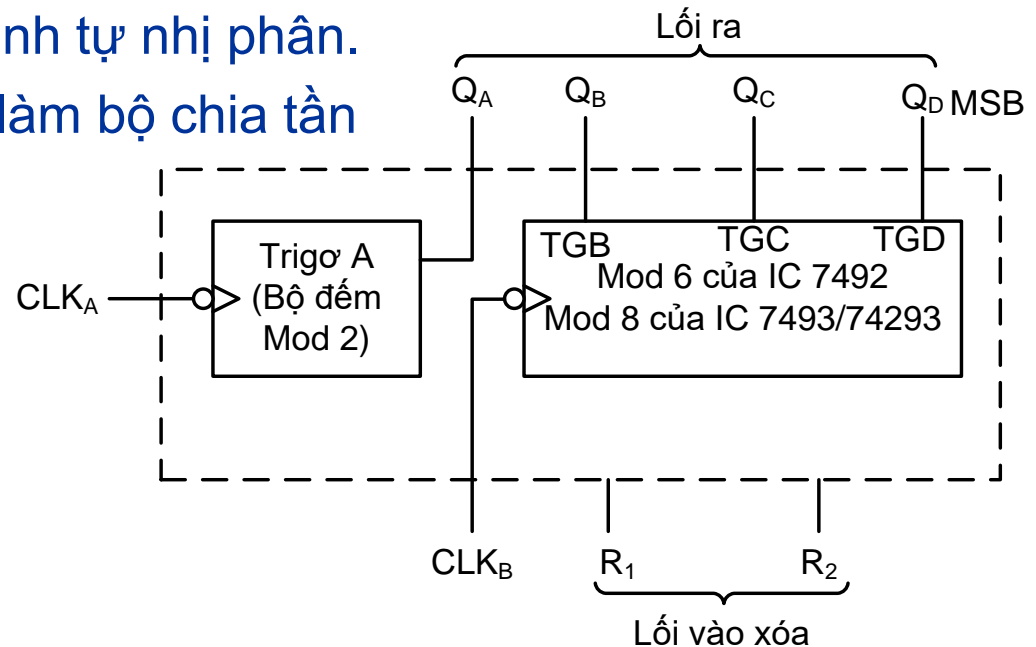
Mạch logic tuần tự(42)

- ❖ **IC 7490, 74390:** gồm 4 trigơ cung cấp bộ đếm gồm hai Mod đếm: Mod 2 và Mod 5.
- ❖ Các bộ đếm Mod 2 và Mod 5 có thể được sử dụng độc lập.
- ❖ Trigơ A thực hiện đếm Mod 2, Trigơ B, C, D thực hiện đếm Mod 5.
- ❖ IC 74390 là bản kép (dual) của 7490



Mạch logic tuần tự(43)

- ❖ **IC 7492, 7493, IC 74293, 74393**: gồm 4 trigơ cung cấp bộ đếm gồm hai Mod đếm: Mod 2 và Mod 6 hoặc mod 8.
- ❖ Trigơ A thực hiện đếm Mod 2, Trigơ B, C, D thực hiện đếm Mod 5.
- ❖ Hoạt động giống IC 7490, chỉ khác là không có các lối vào lập và Mod 6 không đếm theo trình tự nhị phân.
- ❖ Các IC này thường dùng làm bộ chia tần



Mạch logic tuần tự(44)

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