



Ngân hàng Sách (VTC)



0000000000  
0000000000  
0000000000  
0000000000  
0000000000  
0000000000  
0000000000  
0000000000  
0000000000  
0000000000

# KỸ THUẬT TRUYỀN SỐ LIỆU



- Giới thiệu về kỹ thuật truyền số liệu
- Các phương pháp truyền số liệu
- Các thiết bị truyền số liệu
- Các giao thức truyền số liệu
- Các ứng dụng truyền số liệu
- Tổng kết và bài tập

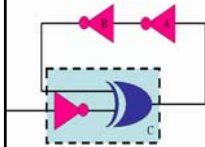
0000000000 0000000000

WIKILOOK.COM



# KỸ THUẬT SỐ

## (Digital Electronics)

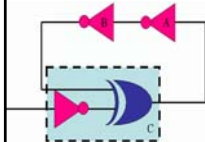


Th.S Đặng Ngọc Khoa  
Khoa Điện - Điện Tử

1

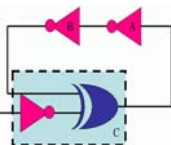


## Nội dung môn học



- Chương 1: Một số khái niệm mở đầu
- Chương 2: Hệ thống số
- Chương 3: Các cổng logic và đại số Boolean
- Chương 4: Mạch logic
- Chương 5: Flip-Flop
- Chương 6: Mạch số học
- Chương 7: Bộ đếm và thanh ghi
- Chương 8: Đặc điểm của các IC số
- Chương 9: Các mạch số thường gặp
- Chương 10: Kết nối với mạch tương tự
- Chương 11: Thiết bị nhớ

2

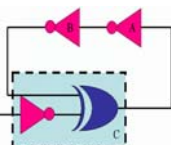


## Giáo trình và tài liệu tham khảo



- **Bài giảng** – Th.S Đặng Ngọc Khoa
- **Kỹ thuật số** - Nguyễn Thúy Vân, NXB.KHKT
- **Kỹ thuật số 1** - Nguyễn Như Anh, NXB. ĐHQG
- **Digital Systems: Principles and Applications** – Ronald J.Tocci, Prentice-Hall

3

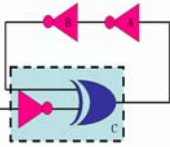


## Chương 1 Một số khái niệm mở đầu



Th.S Đặng Ngọc Khoa  
Khoa Điện - Điện Tử

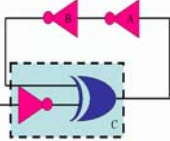
4



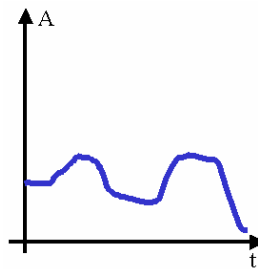
## Mô tả số học

- Tín hiệu analog (tương tự) là tín hiệu có giá trị thay đổi một cách liên tục
- Tín hiệu digital (số) là tín hiệu có giá trị thay đổi theo những bước rời rạc.
- Analog == Tương tự.
- Digital == Rời rạc (step by step)

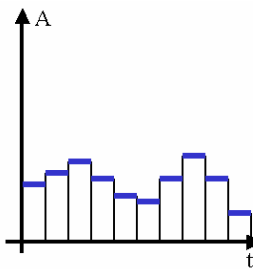
5



## Tín hiệu analog và digital

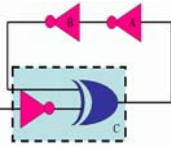


Tín hiệu analog



Tín hiệu digital

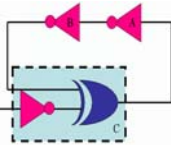
6



## Ví dụ 1.1

- Những đại lượng sau đây là analog hay digital?
  - Công tắc 10 trạng thái
  - Dòng ngõ ra của một thiết bị điện.
  - Nhiệt độ phòng.
  - Tốc độ của một mô tơ điện.
  - Nút điều chỉnh âm thanh của radio.

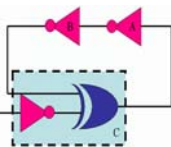
7



## Hệ thống số

- Hệ thống số là một kết hợp của các thiết bị được thiết kế để làm việc với các đại lượng vật lý được miêu tả dưới dạng số.
- Ví dụ: máy vi tính, máy tính tay, các thiết bị audio/video số, điện thoại số, truyền hình kỹ thuật số...

8

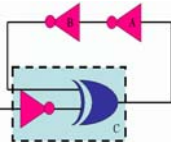


## Ưu điểm của kỹ thuật số



- Nhìn chung, hệ thống số dễ thiết kế.
- Các thông tin được lưu trữ dễ dàng.
- Độ chính xác cao.
- Có thể lập trình hoạt động của hệ thống.
- Các mạch số ít bị ảnh hưởng bởi nhiễu.
- Nhiều mạch số có thể được tích hợp vào trong một IC.

9



## Hạn chế của kỹ thuật số

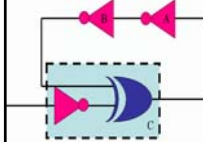


- Trong thực tế phần lớn các đại lượng là analog.
- Để xử lý tín hiệu analog, hệ thống cần thực hiện theo ba bước sau:
  - Biến đổi tín hiệu analog ngõ vào thành tín hiệu số (analog-to-digital converter, ADC)
  - Xử lý thông tin số
  - Biến đổi tín hiệu digital ở ngõ ra thành tín hiệu analog (digital-to-analog converter, DAC)

10



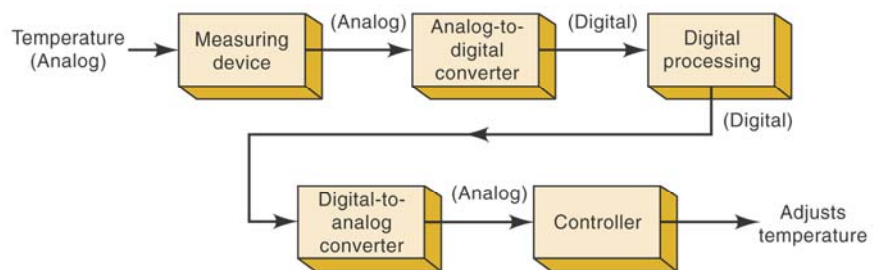
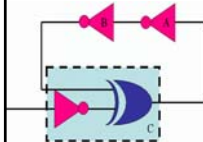
## Một số ví dụ về hệ thống số



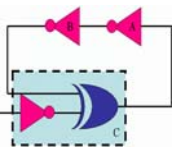
11



## Hệ thống điều khiển nhiệt độ



12

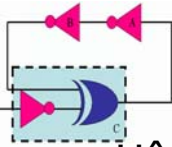


## Đĩa CD (Compact Disk)



- Âm thanh của các nhạc cụ và tiếng hát sẽ tạo ra một tín hiệu điện áp analog trong microphone.
- Tín hiệu analog này sẽ được biến đổi thành dạng số.
- Thông tin số sẽ được lưu trữ trong đĩa CD
- Trong quá trình playback, máy CD nhận thông tin số từ đĩa CD và biến đổi thành tín hiệu analog, sau đó khuếch đại và đưa ra loa.

13



## Lựa chọn giữa digital & analog

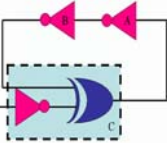


- Hệ thống số phải thêm vào 2 bộ ADC và DAC (phức tạp, tốn kém)
- Hệ thống số yêu cầu thêm thời gian cho các quá trình biến đổi (hạn chế tốc độ)
- Trong phần lớn các ứng dụng, hệ thống số thường được ưu tiên ứng dụng do các ưu điểm của nó.
- Mạch analog được sử dụng dễ dàng cho quá trình khuếch đại tín hiệu.

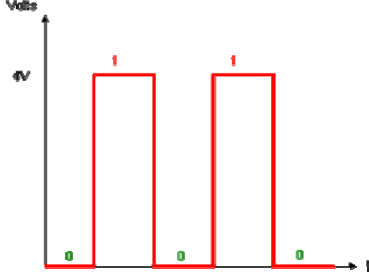
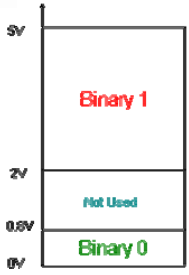
➔ **Kết hợp giữa analog và digital**

14



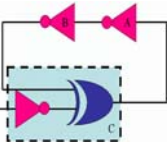


## Giá trị điện áp trong Digital



- **Binary 1**: Điện áp từ 2V đến 5V
- **Binary 0**: Điện áp từ 0V đến 0.8V
- **Not used**: Điện áp từ 0.8V đến 2V, vùng này có thể gây ra lỗi trong mạch số.

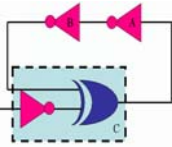
15



## Mạch số

- Mạch số phải được thiết kế để điện áp ngõ ra nằm trong khoảng logic 0 hoặc logic 1
- Một mạch số làm việc với các giá trị ngõ vào là logic 0 hoặc 1 mà không quan tâm đến giá trị điện áp thực tế.
- Mỗi một mạch số tuân theo một tập hợp các quy luật logic nhất định.

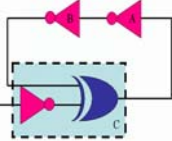
16



## Mạch số tích hợp

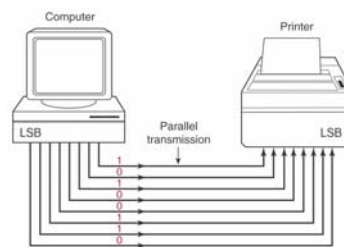
- Phần lớn các mạch số được tích hợp trong IC.
- Một số kỹ thuật chế tạo IC
  - TTL
  - CMOS
- Những họ IC này sẽ được đề cập trong chương 8.

17

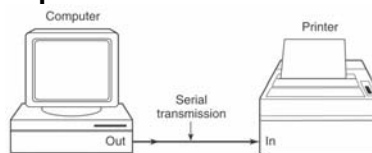


## Truyền song song & nối tiếp

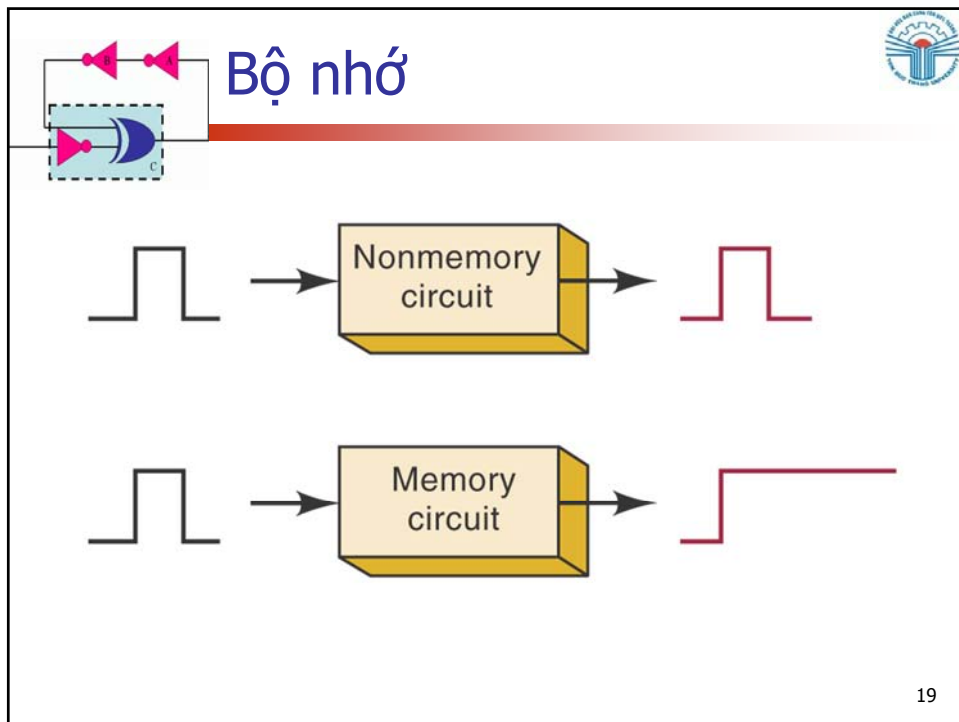
- Truyền song song




- Truyền nối tiếp



18

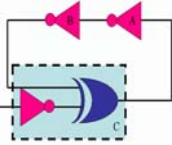




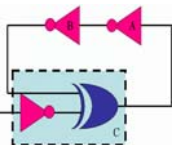
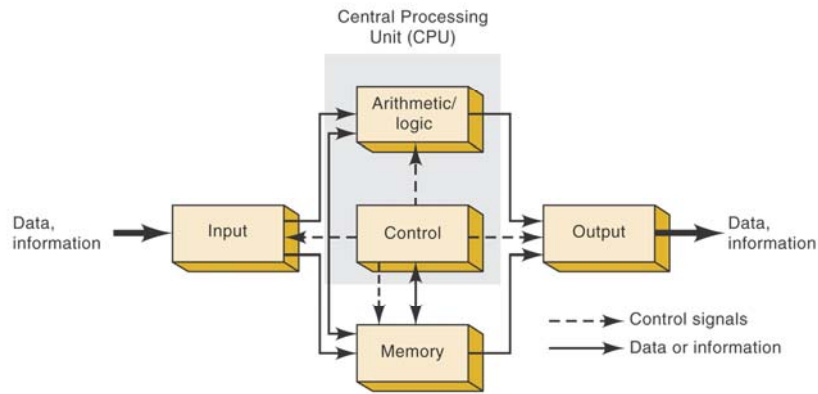
## Bộ nhớ

- Trạng thái của mạch có thể được lưu trữ sau khi chấm dứt tín hiệu ngõ vào.
- Thuộc tính lưu giá trị của nó tương ứng với thiết bị nhớ nên được gọi là bộ nhớ (memory)
- Bộ nhớ thường được làm từ các mạch Latches (chốt) hoặc Flip-Flop.

20



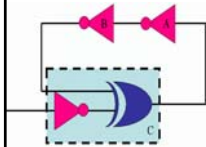
# Các phần chính của máy tính



# Câu hỏi?



## Chương 2 Hệ thống số

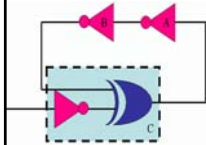


Th.S Đặng Ngọc Khoa  
Khoa Điện - Điện Tử

1

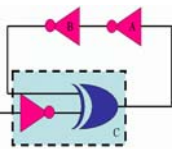


## Định nghĩa



- Một hệ thống số bao gồm các ký tự trong đó định nghĩa các phép toán cộng, trừ, nhân, chia.
- Hệ cơ số của một hệ thống số là tổng ký tự có trong hệ thống số đó.
- Trong kỹ thuật số có các hệ thống số sau đây: **Binary, Octal, Decimal, Hexadecimal**.

2

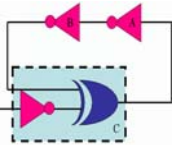


## Định nghĩa (tt)



Hệ thống số	Cơ số	Các ký tự có trong hệ thống
Decimal	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Binary	2	0, 1
Octal	8	0, 1, 2, 3, 4, 5, 6, 7
Hexa-decimal	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

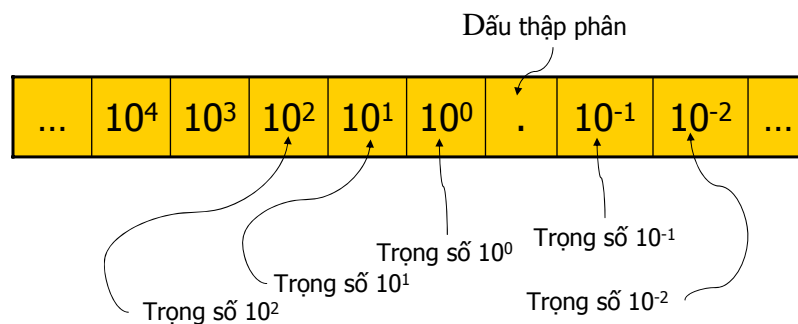
3



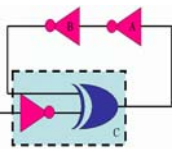
## Hệ thống số thập phân



- Hệ thống số thập phân có phân bố các trọng số như sau:



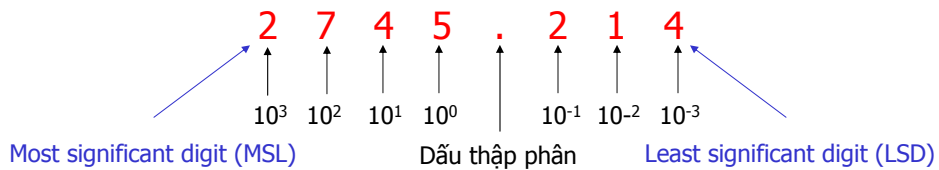
4



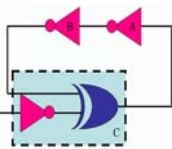
# Hệ thống số thập phân (tt)



- Ví dụ: phân tích số thập phân  $2745.214_{10}$



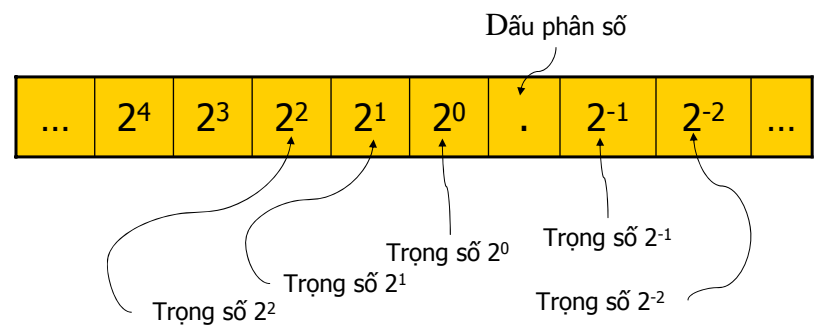
- $2745.214_{10} =$   
 $(2 \times 10^3) + (7 \times 10^2) + (4 \times 10^1) +$   
 $(5 \times 10^0) + (2 \times 10^{-1}) + (1 \times 10^{-2}) +$   
 $(4 \times 10^{-3})$

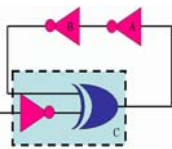


# Hệ thống số nhị phân



- Hệ thống số nhị phân có phân bố các trọng số như sau:

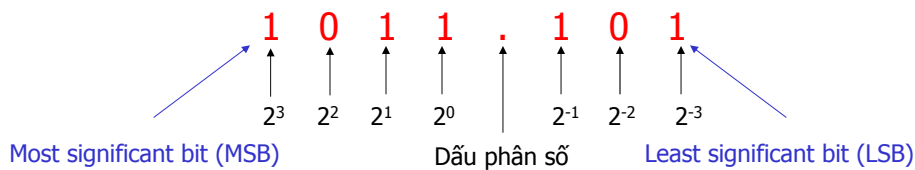




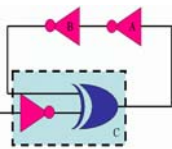
## Hệ thống số nhị phân (tt)



- Ví dụ: phân tích số nhị phân  $1011.101_2$



- $1011.101_2 =$   
 $(1 \times 2^3) + (0 \times 2^2) + (1 \times 2^1) +$   
 $(1 \times 2^0) + (1 \times 2^{-1}) + (0 \times 2^{-2}) +$   
 $(1 \times 2^{-3}) = 11.625_{10}$



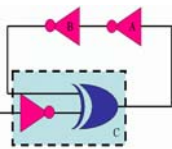
## Phép cộng nhị phân



- Cộng hai bit nhị phân

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	10



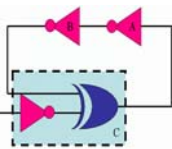


## Phép cộng nhị phân (tt)



- Cộng hai số nhị phân không dấu

$$\begin{array}{r} \text{a) } \quad 11 \quad (3) \\ \quad +110 \quad (6) \\ \hline \quad 1001 \quad (9) \end{array} \quad \Bigg| \quad \begin{array}{r} \text{b) } \quad 11.011 \quad (3.375) \\ \quad +10.110 \quad (2.750) \\ \hline \quad 110.001 \quad (6.125) \end{array}$$

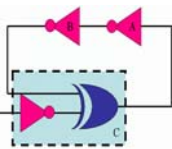


## Phép nhân nhị phân



- Nhân 2 bit nhị phân

A	B	A x B
0	0	0
0	1	0
1	0	0
1	1	1



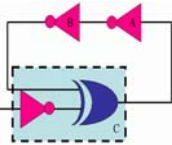
## Phép nhân nhị phân



- Nhân 2 số nhị phân

$$\begin{array}{r} 1110 \\ \times 1011 \\ \hline 1110 \\ 1110 \\ 0000 \\ 1110 \\ \hline 10011010 \end{array}$$

11

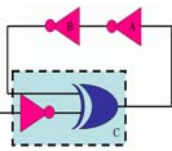


## Số nhị phân có dấu



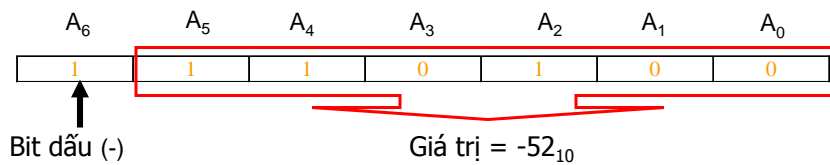
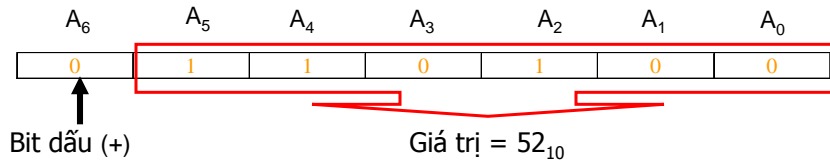
- Trong trường hợp cần thể hiện dấu, số nhị phân sử dụng 1 bit để xác định dấu.
- Bit này thường ở vị trí đầu tiên
- Bit dấu bằng 0 xác định số dương.
- Bit dấu bằng 1 xác định số âm.

12

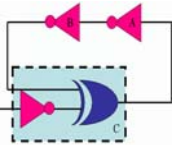


## Số nhị phân có dấu

### ■ Số nhị phân 6 bit có dấu



13

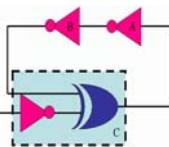


## Bội trong hệ nhị phân

- Để đo lường dung lượng của bộ nhớ, đơn vị Kilo, Mega, Giga được sử dụng

Bội	Đơn vị	Ký hiệu	Giá trị
2 <sup>10</sup>	Kilo	K	1024
2 <sup>20</sup>	Mega	M	1048576
2 <sup>30</sup>	Giga	G	1073741824

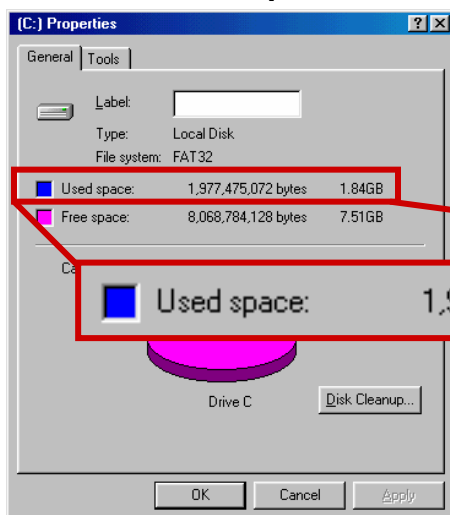
14



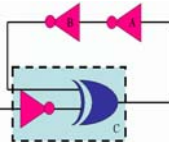
## Bội trong hệ nhị phân



### ■ Ví dụ



15



## Hệ thống số bát phân

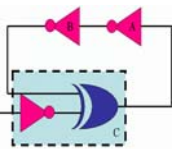


- Hệ thống số bát phân có phân bố các trọng số như sau:

...	$8^4$	$8^3$	$8^2$	$8^1$	$8^0$	.	$8^{-1}$	$8^{-2}$	...
-----	-------	-------	-------	-------	-------	---	----------	----------	-----

- Ví dụ: phân tích số bát phân  $372_8$   
$$372_8 = (3 \times 8^2) + (7 \times 8^1) + (2 \times 8^0)$$
$$= (3 \times 64) + (7 \times 8) + (2 \times 1)$$
$$= \boxed{250}_{10}$$

16



## Hệ thống số thập lục phân

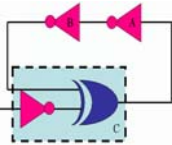


- Hệ thống số thập lục phân có phân bố các trọng số như sau:

...	$16^4$	$16^3$	$16^2$	$16^1$	$16^0$	.	$16^{-1}$	$16^{-2}$	...
-----	--------	--------	--------	--------	--------	---	-----------	-----------	-----

- Ví dụ: phân tích số thập lục phân  $3BA_{16}$   
 $3BA_{16} = (3 \times 16^2) + (11 \times 16^1) + (10 \times 16^0)$   
 $= (3 \times 256) + (11 \times 16) + (10 \times 1)$   
 $= 954_{10}$

17



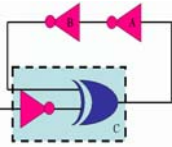
## Mã BCD (Binary coded decimal)



- Mỗi chữ số trong một số thập phân được miêu tả bằng giá trị nhị phân tương ứng.
- Mỗi chữ số thập phân sẽ được miêu tả bằng 4 bit nhị phân.

0	1	2	3	4	5	6	7	8	9
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
0000	0001	0010	0011	0100	0101	0110	0111	1000	1001

18

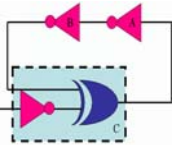


## Mã BCD

- Ví dụ hai số thập phân 847 và 943 được miêu tả bởi mã BCD như sau:

8	4	7	9	4	3
↓	↓	↓	↓	↓	↓
1000	0100	0111	1001	0100	0011

19



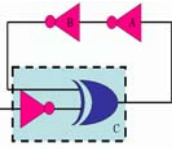
## So sánh BCD và Binary

$137_{10} = 10001001_2$  (Binary)

$137_{10} = 0001\ 0011\ 0111$  (BCD)

- Mã BCD sử dụng nhiều bit hơn nhưng quá trình biến đổi đơn giản hơn

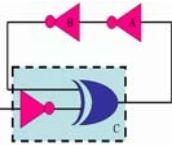
20



## Bảng chuyển đổi

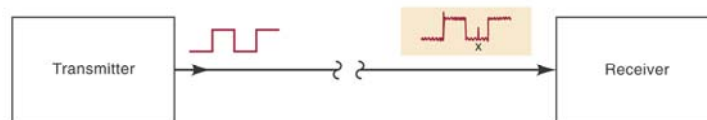
Decimal	Binary	Octal	Hexadecimal	BCD
0	0	0	0	0000
1	01	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	10	8	1000
9	1001	11	9	1001
10	1010	12	A	1000 0000
11	1011	13	B	1000 0001
12	1100	14	C	1000 0010
13	1101	15	D	1000 0011
14	1110	16	E	1000 0100
15	1111	17	F	1000 0101

21



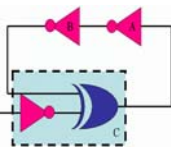
## Sử dụng bit Parity để phát hiện lỗi

- Trong quá trình truyền dữ liệu nhị phân, nhiễu có thể gây nên những lỗi trên đường truyền.



- Phương pháp đơn giản để phát hiện lỗi là sử dụng bit Parity

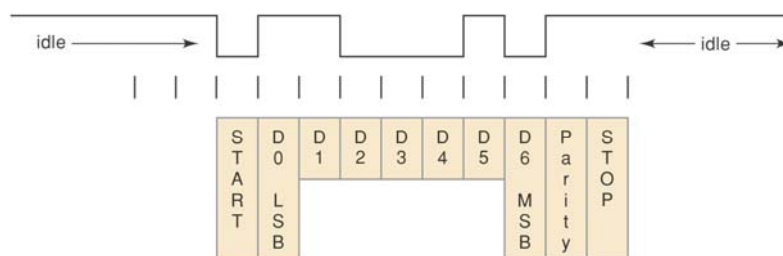
22



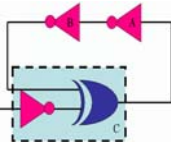
## Sử dụng bit Parity để phát hiện lỗi



- Trong phương pháp này, một bit mở rộng sẽ được thêm vào, bit mở rộng được gọi là bit Parity



23



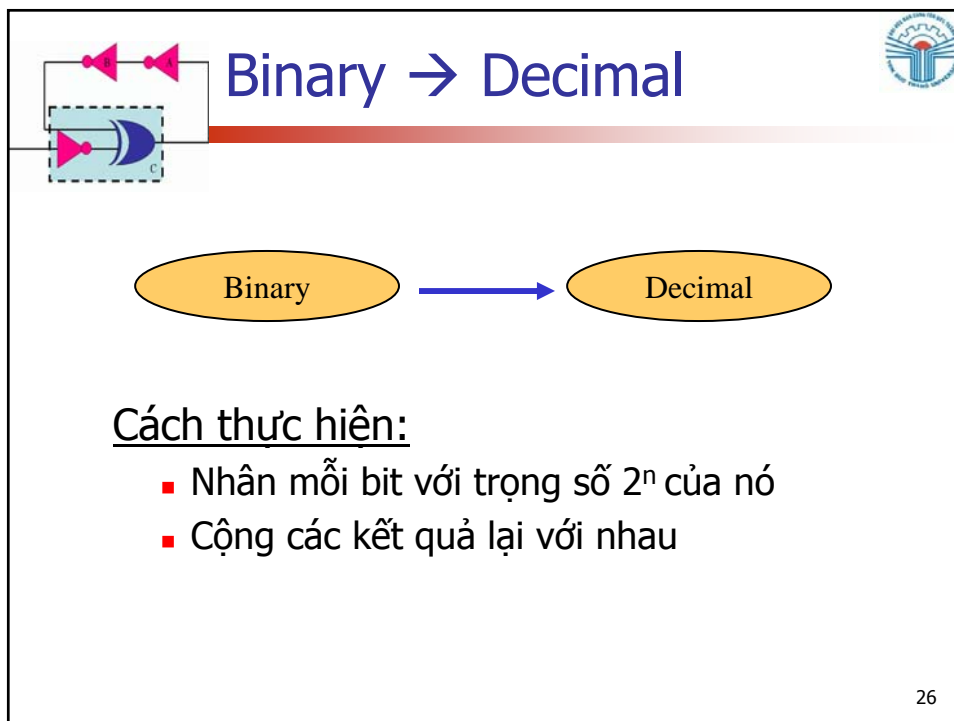
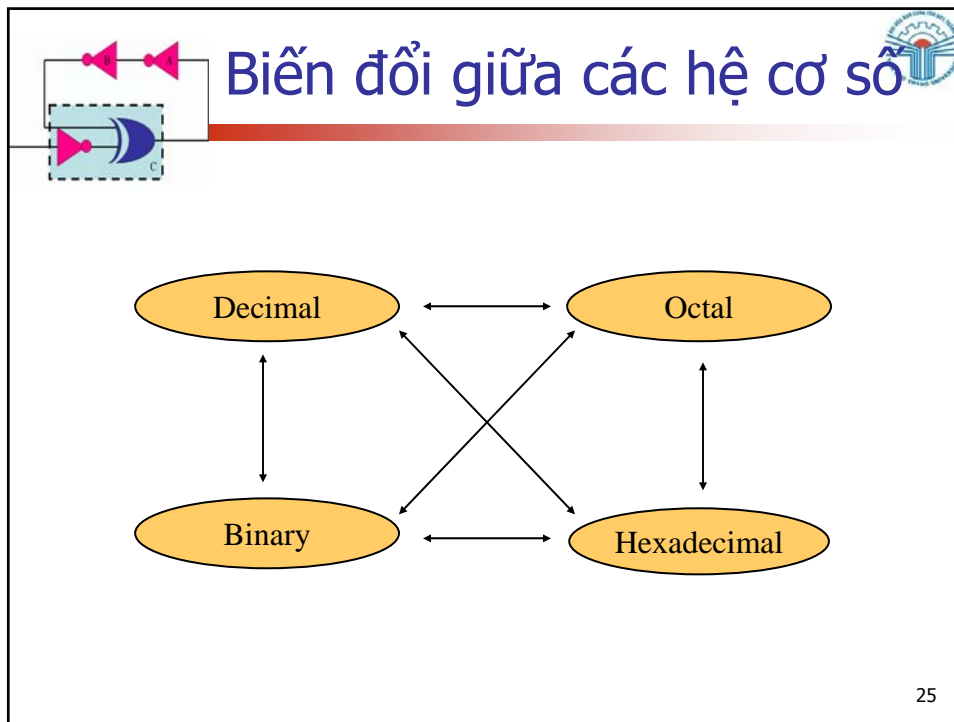
## Sử dụng bit Parity để phát hiện lỗi

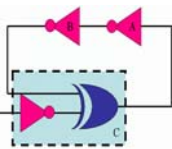


- Giá trị của bit Parity phụ thuộc vào phương pháp sử dụng và số bit 1 trong khung dữ liệu.
  - Phương pháp Parity chẵn: tổng số bit 1 trong khung dữ liệu (kể cả bit parity) phải là số chẵn.
    - Dữ liệu 1 0 1 1, bit parity thêm vào **1** 1 0 1 1
  - Phương pháp Parity lẻ: tổng số bit 1 trong khung dữ liệu (kể cả bit parity) phải là số lẻ.
    - Dữ liệu 1 1 1 1, bit parity thêm vào **1** 1 1 1 1

24







## Binary → Decimal (tt)

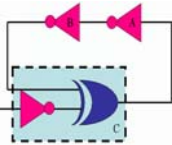


- Ví dụ: biến đổi  $(10101101)_2$  sang thập phân

Binary	→	1	0	1	0	1	1	0	1
		X	X	X	X	X	X	X	X
Giá trị	→	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
Kết quả	→	$128 + 32 + 8 + 4 + 1$							

**173<sub>10</sub>**

27




## Decimal → Binary



### Cách thực hiện:

- Chia 2 lấy phần dư
- Số dư đầu tiên là bit LSB (least significant bit)
- Số dư cuối cùng là bit MLB (most significant bit)

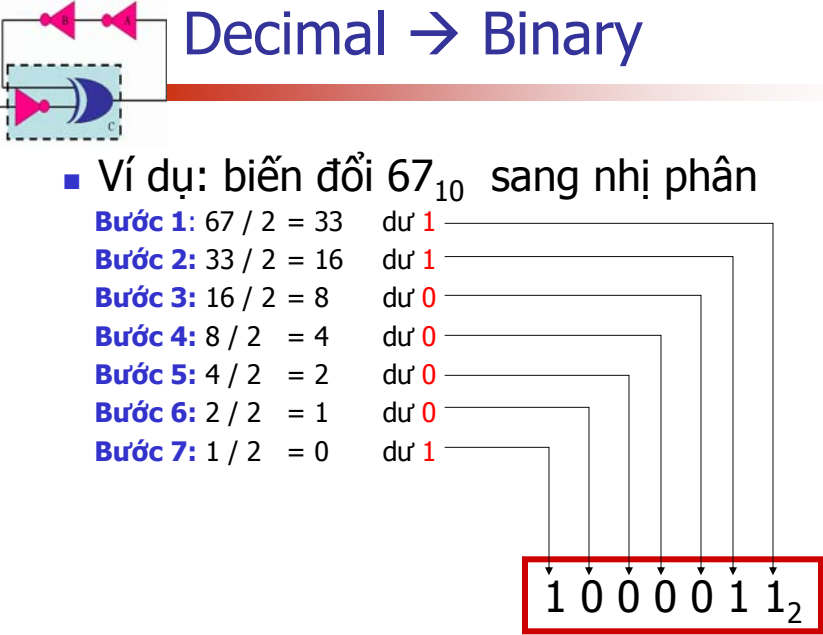
28




## Decimal → Binary

Ví dụ: biến đổi  $67_{10}$  sang nhị phân


**Bước 1:**  $67 / 2 = 33$  dư **1**  
**Bước 2:**  $33 / 2 = 16$  dư **1**  
**Bước 3:**  $16 / 2 = 8$  dư **0**  
**Bước 4:**  $8 / 2 = 4$  dư **0**  
**Bước 5:**  $4 / 2 = 2$  dư **0**  
**Bước 6:**  $2 / 2 = 1$  dư **0**  
**Bước 7:**  $1 / 2 = 0$  dư **1**



29



## Octal → Binary

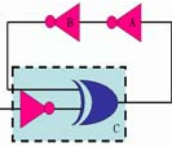


Cách thực hiện:

- Biến mỗi ký tự số trong Octal thành 3 bit nhị phân tương ứng.

Octal	0	1	2	3	4	5	6	7
	↓	↓	↓	↓	↓	↓	↓	↓
Binary	000	001	010	011	100	101	110	111

30



## Octal → Binary (tt)

- Biến đổi  $472_8$  sang hệ nhị phân

4    7    2  
↓    ↓    ↓

**100111010<sub>2</sub>**

100   111   010

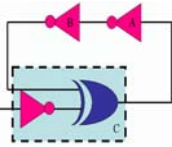
- Biến đổi  $5431_8$  sang hệ nhị phân

5    4    3    1  
↓    ↓    ↓    ↓

**101100011001<sub>2</sub>**

101   100   011   001

31



## Hexa → Binary

Hexa



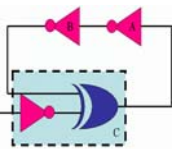
Binary

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

### Cách thực hiện:

- Biến mỗi ký tự số trong Hexa thành 4 bit nhị phân tương ứng.

32



## Hexa → Binary (tt)



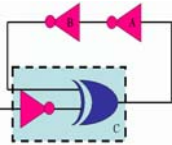
- Biến đổi  $47C_{16}$  sang hệ nhị phân

4	7	C	10001111100 <sub>2</sub>
↓	↓	↓	
0100	0111	1100	

- Biến đổi  $10AF_{16}$  sang hệ nhị phân

1	0	A	F	1000010101111 <sub>2</sub>
↓	↓	↓	↓	
0001	0000	1010	1111	

33




## Decimal → Octal



### Cách thực hiện:

- Chia 8 lấy phần dư
- Số dư đầu tiên là LSD (least significant digit)
- Số dư cuối cùng là MLD (most significant digit)

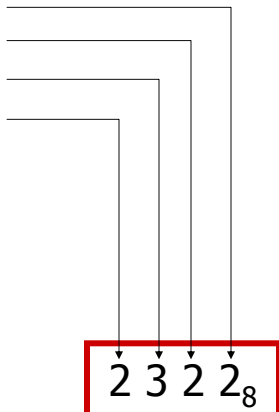
34



## Decimal → Octal (tt)


Ví dụ: biến đổi  $1234_{10}$  sang bát phân

<b>Bước 1:</b>	$1234 / 8$	$= 154$	dư <b>2</b>	—
<b>Bước 2:</b>	$154 / 8$	$= 19$	dư <b>2</b>	—
<b>Bước 3:</b>	$19 / 8$	$= 2$	dư <b>3</b>	—
<b>Bước 4:</b>	$2 / 8$	$= 0$	dư <b>2</b>	—




$2\ 3\ 2\ 2_8$

35



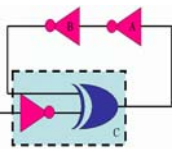
## Decimal → Hexa



**Cách thực hiện:**

- Chia 16 lấy phần dư
- Số dư đầu tiên là LSD (least significant digit)
- Số dư cuối cùng là MLD (most significant digit)

36



## Decimal → Hexa (tt)



- Ví dụ: biến đổi  $4660_{10}$  sang thập lục phân

**Bước 1:**  $4660 / 16 = 291$  dư **4**

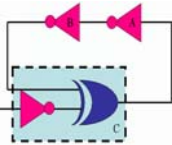
**Bước 2:**  $291 / 16 = 18$  dư **3**

**Bước 3:**  $18 / 16 = 1$  dư **2**

**Bước 4:**  $1 / 16 = 0$  dư **1**

1 2 3 4<sub>16</sub>

37



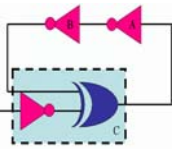
## Binary → Octal



### Cách thực hiện:

- Bắt đầu từ bên trái, nhóm số nhị phân thành các nhóm 3 bit
- Biến đổi mỗi nhóm 3 bit thành một số Octal

38



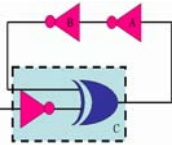
## Binary → Octal (tt)

- Ví dụ: biến đổi  $1011010111_2$  sang Octal

1 3 2 7  
1 011 010 111

$$1011010111_2 = 1327_8$$

39



## Binary → Hexa

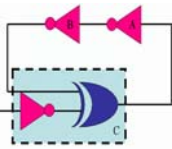


### Cách thực hiện:

- Bắt đầu từ bên trái, nhóm số nhị phân thành các nhóm 4 bit
- Biến đổi mỗi nhóm 4 bit thành một số Hexa

40





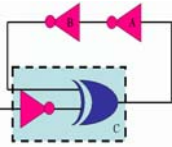
## Binary → Hexa (tt)

- Ví dụ: biến đổi  $10101101010111001101010_2$  sang Hexa

5 6 A E 6 A  
101 0110 1010 1110 0110 1010

$$10101101010111001101010_2 = \boxed{56AE6A}_{16}$$

41



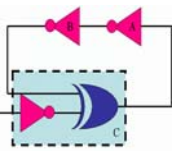
## Octal → Hexa



### Cách thực hiện:

- Biến đổi số Octal thành số Binary
- Biến đổi số Binary thành số Hexa

42



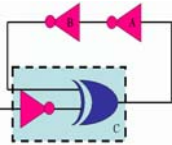
## Octal → Hexa (tt)

- Ví dụ: biến đổi  $1076_8$  sang Hexa

1	0	7	6
↓	↓	↓	↓
001	000	111	110
└───┘	└───┘	└───┘	└───┘
2	3	E	

$1076_8 = \boxed{23E_{16}}$

43



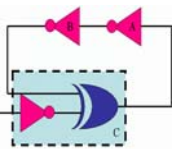
## Hexa → Octal



### Cách thực hiện:

- Biến đổi số Hexa thành số Binary
- Biến đổi số Binary thành số Octal

44



## Hexa → Octal (tt)

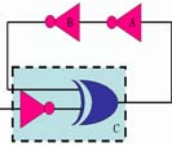


- Ví dụ: biến đổi  $1F0C_{16}$  sang Octal

1	F	0	C	
↓	↓	↓	↓	
0001	1111	0000	1100	
1	7	4	1	4

$1F0C_{16} =$   $17414_8$

45



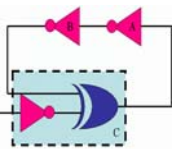
## Bài tập - Biến đổi 1



- Thực hiện các phép biến đổi sau:

Decimal	Binary	Octal	Hexa
33			
	1110101		
		703	
			1AF

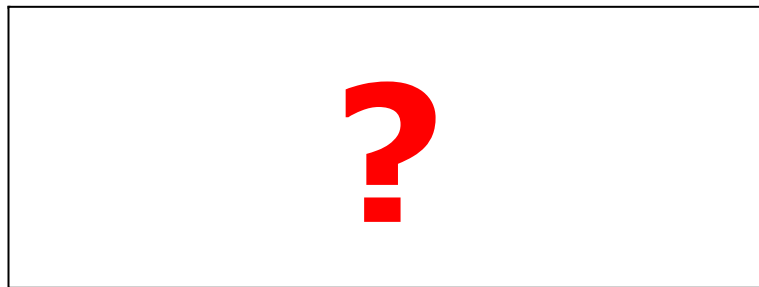
46



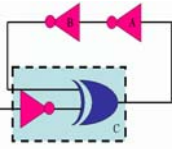
## Bài tập - Biến đổi 1 (tt)



- Kết quả:



47



## Phân số

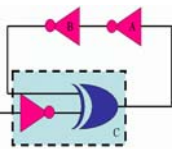


- Binary  $\rightarrow$  Decimal

$$\begin{array}{r} 1 \ 0.1 \ 0 \ 1 \ 1 \\ \begin{array}{l} \rightarrow 1 \times 2^{-4} = 0.0625 \\ \rightarrow 1 \times 2^{-3} = 0.125 \\ \rightarrow 0 \times 2^{-2} = 0.0 \\ \rightarrow 1 \times 2^{-1} = 0.5 \\ \rightarrow 0 \times 2^0 = 0.0 \\ \rightarrow 1 \times 2^1 = 2.0 \end{array} \end{array}$$

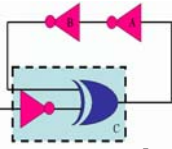
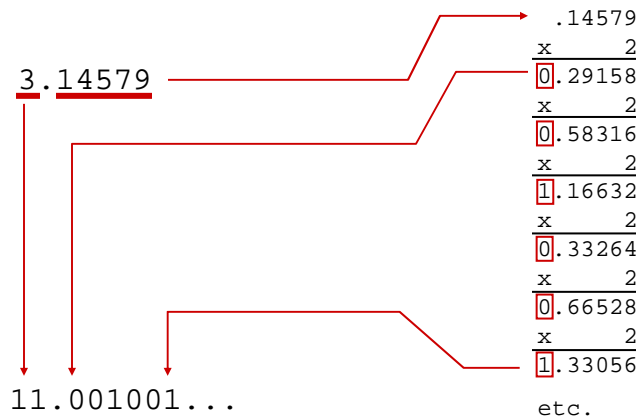
$$10.1011_2 = \boxed{2.6875}$$

48



# Phân số

## Deciaml → Binary

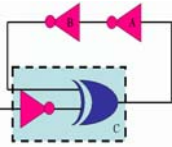


# Phân số

## Ví dụ: chuyển $189.023_{10}$ thành số binary

$189/2 = 94$	dư 1	$0.023 \times 2 = 0.046$	dư 0
$94/2 = 47$	dư 0	$0.046 \times 2 = 0.092$	dư 0
$47/2 = 23$	dư 1	$0.092 \times 2 = 0.184$	dư 0
$23/2 = 11$	dư 1	$0.184 \times 2 = 0.368$	dư 0
$11/2 = 5$	dư 1	$0.368 \times 2 = 0.736$	dư 0
$5/2 = 2$	dư 1	$0.736 \times 2 = 1.472$	dư 1
$2/2 = 1$	dư 0	$0.472 \times 2 = 0.944$	dư 0
$1/2 = 0$	dư 1	...	

$$189.023 = 10111101.0000010_2$$

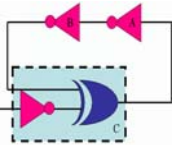


## Bài tập - Biến đổi 2

- Thực hiện các phép biến đổi sau:

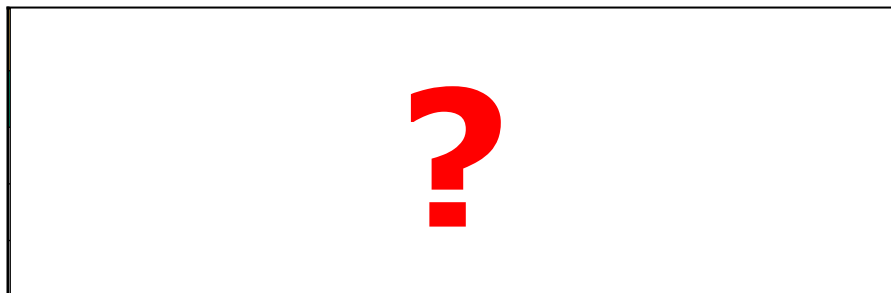
Decimal	Binary	Octal	Hexa
29.8			
	101.1101		
		3.07	
			C.82

51

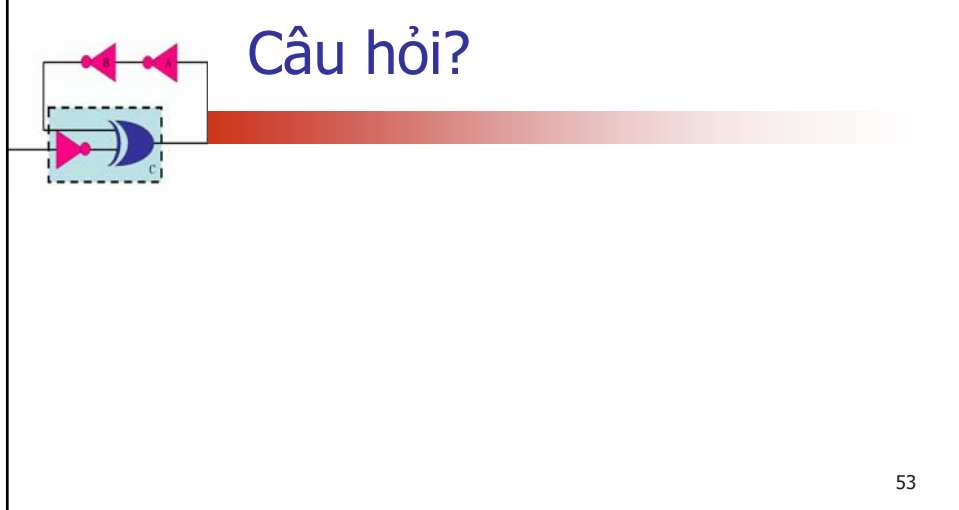


## Bài tập - Biến đổi 2 (tt)

- Kết quả:



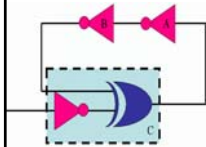
52





## Chương 3

### Các cổng logic & Đại số Boolean

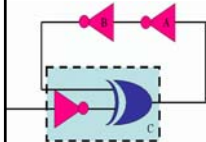


Th.S Đặng Ngọc Khoa  
Khoa Điện - Điện Tử

1



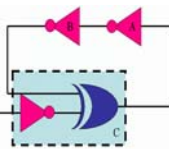
### Hằng số Boolean và biến



- Khác với các đại số khác, các hằng và biến trong đại số Boolean chỉ có hai giá trị: 0 và 1
- Trong đại số Boolean không có: phân số, số âm, lũy thừa, căn số, ...
- Đại số Boolean chỉ có 3 toán tử:
  - Cộng logic, hay còn gọi toán tử **OR**
  - Nhân logic, hay còn gọi toán tử **AND**
  - Bù logic, hay còn gọi toán tử **NOT**

2





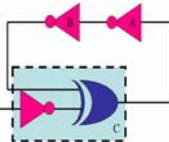
## Hàng số Boolean và biến (tt)



- Giá trị 0 và 1 trong đại số Boolean mang ý nghĩa miêu tả các trạng thái hay mức logic

Logic 0	Logic 1
False	True
Off	On
Low	High
No	Yes
Open switch	Closed switch

3



## Bảng chân trị



- Bảng chân trị miêu tả mối quan hệ giữa giá trị các ngõ vào và ngõ ra. Ví dụ:

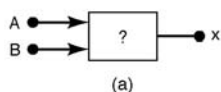
Inputs		Output
A	B	x
0	0	1
0	1	0
1	0	1
1	1	0

A	B	C	x
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

(b)

A	B	C	D	x
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

(c)



(a)

4



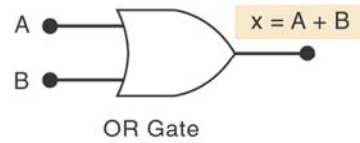
# Cổng OR

- Biểu thức Boolean của cổng OR

$$x = A + B$$

OR		
A	B	$x = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

(a)

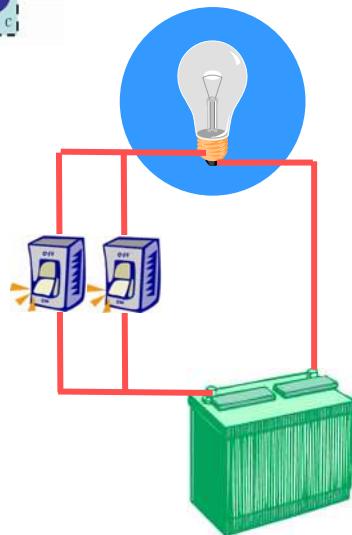


OR Gate

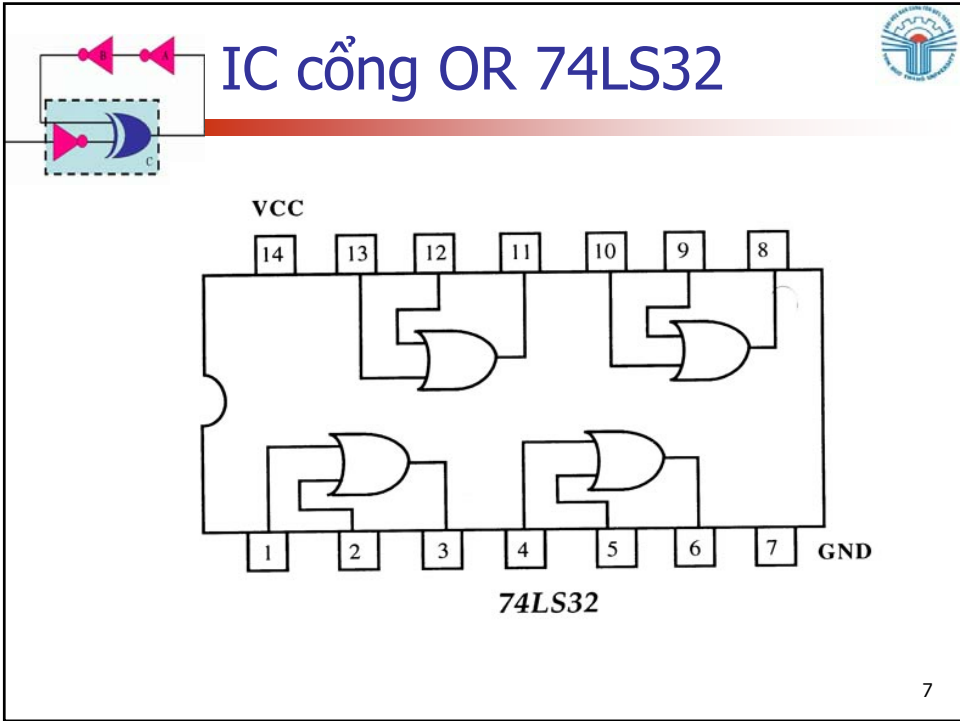
(b)

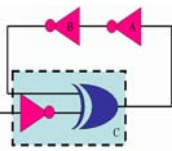


# Cổng OR (tt)



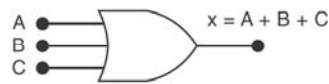
**Ngõ ra ở trạng thái tích cực khi ít nhất một ngõ vào ở trạng thái tích cực.**





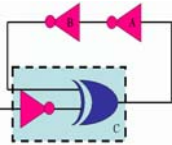
## Cổng OR (tt)

- Cổng OR có thể có nhiều hơn 2 ngõ vào.



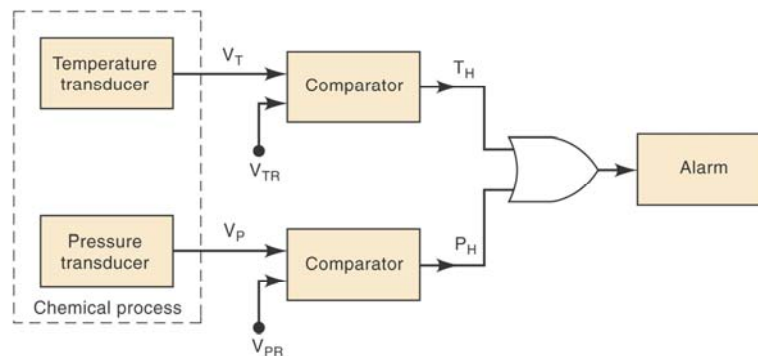
A	B	C	$x = A + B + C$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

9

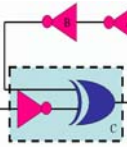


## Ví dụ 3-1

- Cổng OR được sử dụng trong một hệ thống báo động.

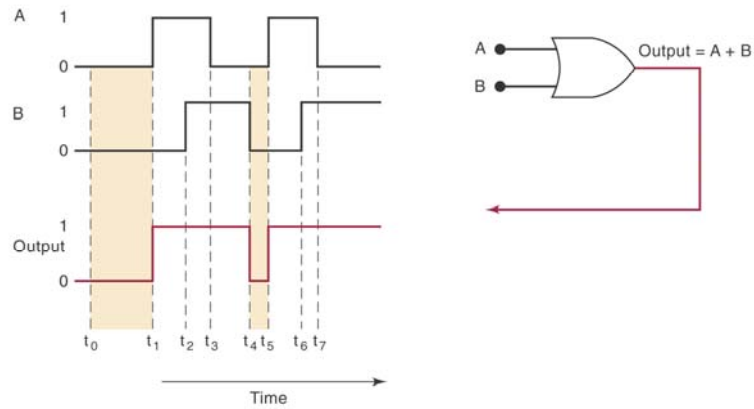


10

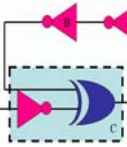


### Ví dụ 3-2

#### ■ Biểu đồ thời gian cho cổng OR.

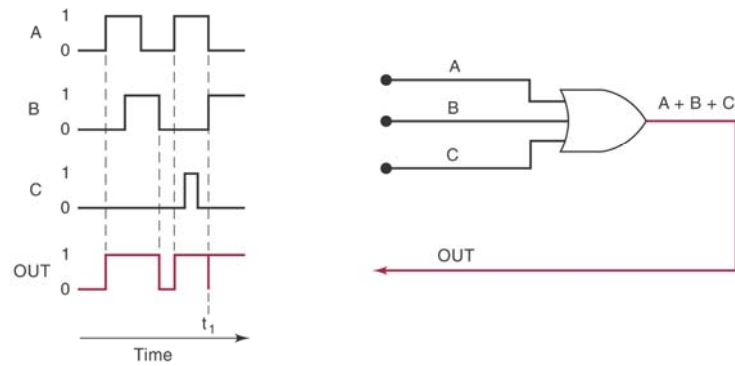


11

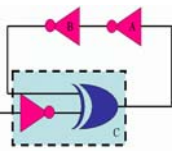


### Ví dụ 3-3

#### ■ Biểu đồ thời gian cho cổng OR.



12



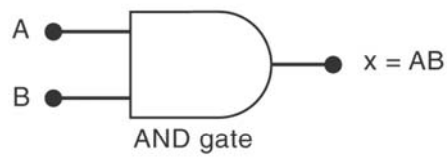
## Cổng AND

- Biểu thức Boolean của cổng AND

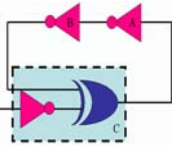
$$x = A * B$$

AND		
A	B	$x = A \cdot B$
0	0	0
0	1	0
1	0	0
1	1	1

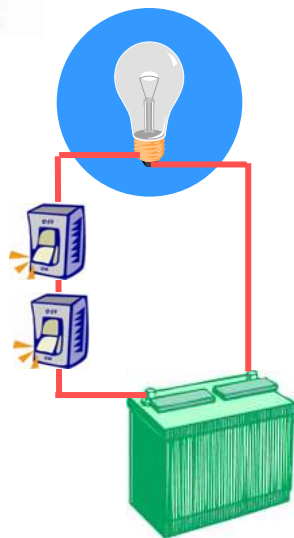
(a)



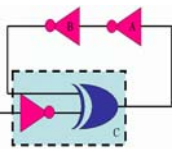
(b)



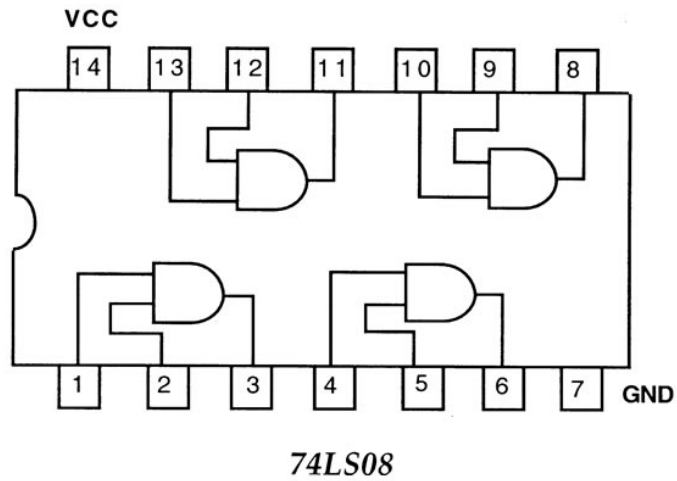
## Cổng AND (tt)



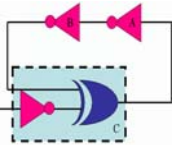
**Ngõ ra ở trạng thái tích cực khi tất cả các ngõ vào ở trạng thái tích cực.**



## IC cổng AND 74LS08



15

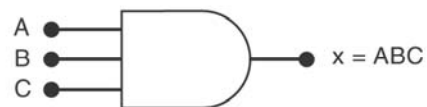


## Cổng AND (tt)

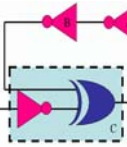


- Cổng AND có thể có nhiều hơn 2 ngõ vào.

A	B	C	x = ABC
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

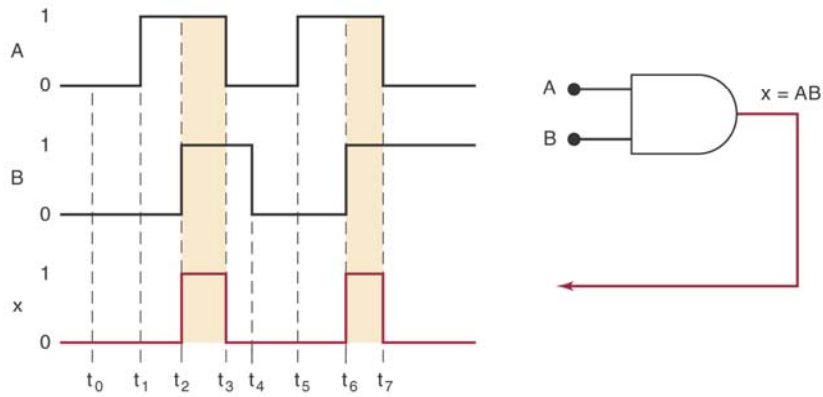


16

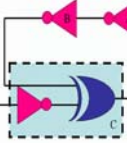


## Ví dụ 3-4

- Biểu đồ thời gian cho cổng AND.

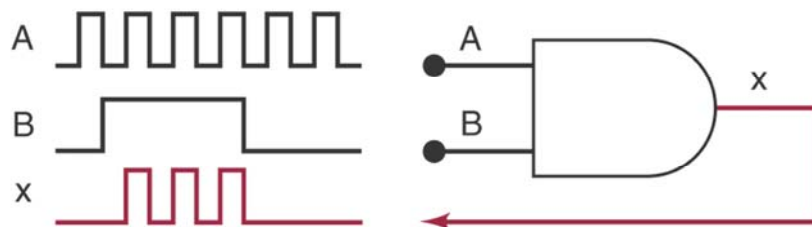


--



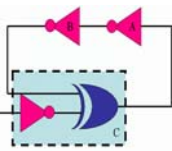
## Mạch Enable/Disable

- Cổng AND được sử dụng làm một mạch khóa đơn giản



18





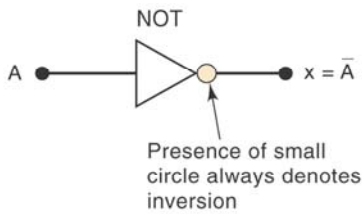
# Cổng NOT

- Cổng NOT luôn luôn chỉ có một ngõ vào
- Biểu thức Boolean của cổng NOT

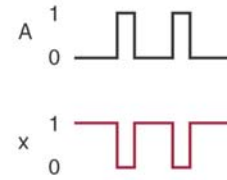
$$x = \bar{A}$$

NOT	
A	x = $\bar{A}$
0	1
1	0

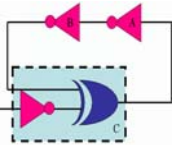
(a)



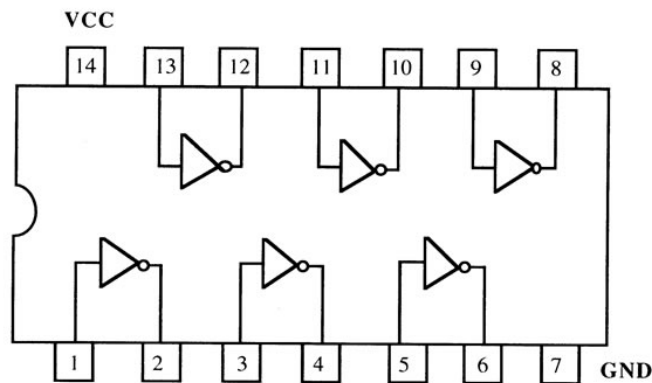
(b)



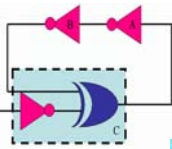
(c)



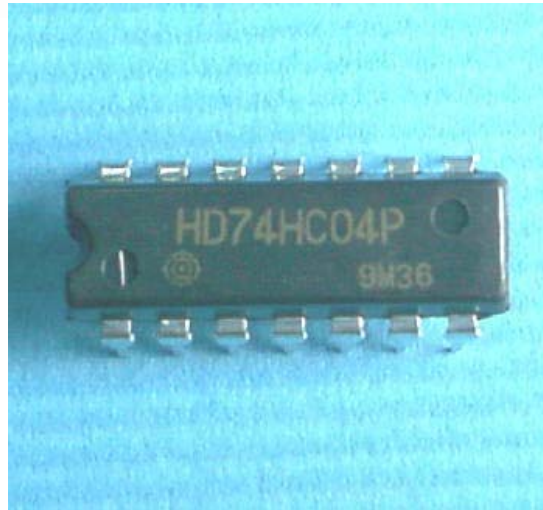
# IC cổng NOT 74LS04



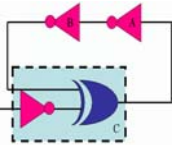
74LS04



## IC cổng NOT 74LS04

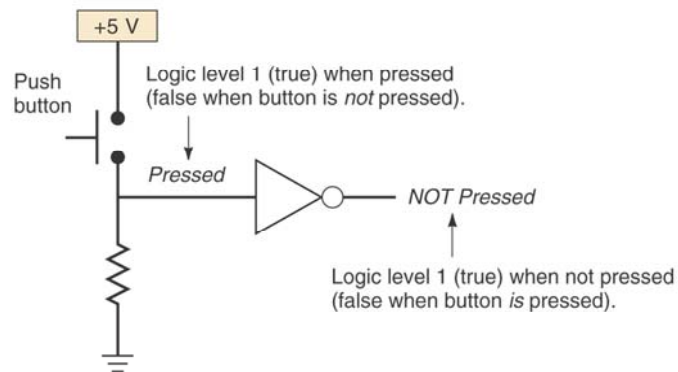


21

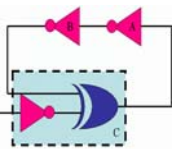


## Ví dụ 3-5

- Ngõ ra của cổng NOT xác định trạng thái của nút nhấn.



22

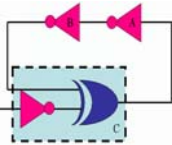


## Miêu tả đại số mạch logic

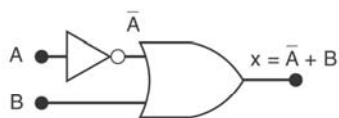


- Bất kỳ mạch logic nào cũng có thể được xây dựng từ 3 cổng logic cơ bản: AND, OR và NOT.
- Ví dụ:
  - $x = AB + C$
  - $x = (A+B)C$
  - $x = (A+B)$
  - $x = ABC(A+D)$

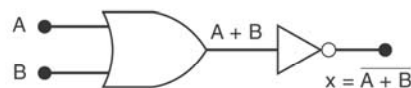
23



## Ví dụ 3-6



(a)

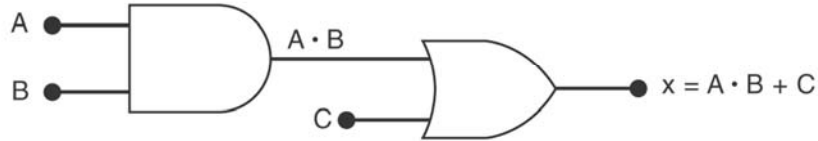
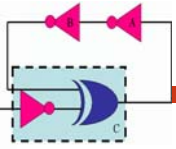


(b)

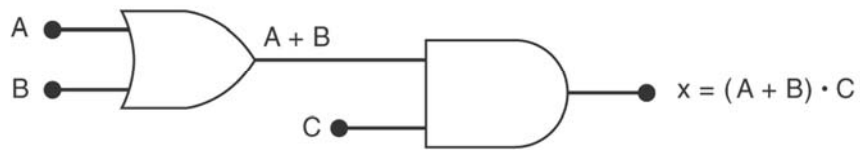
24



### Ví dụ 3-7



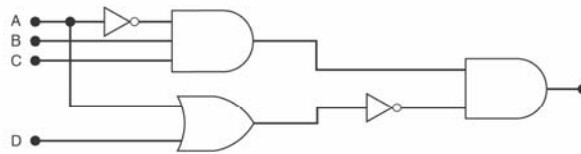
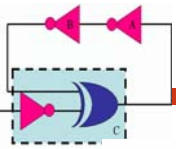
(a)



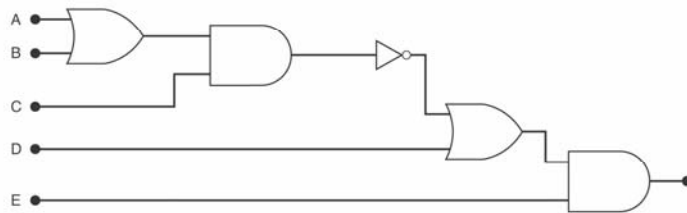
(b)



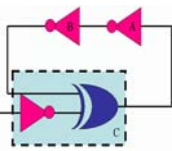
### Ví dụ 3-8



(a)

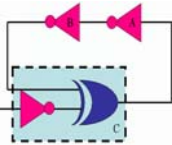
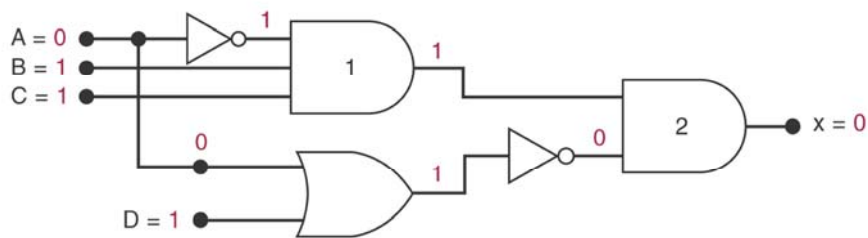


(b)



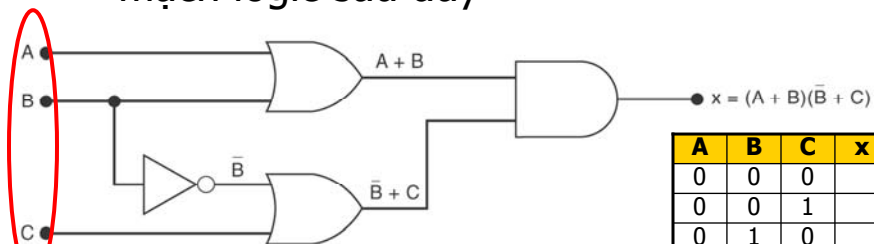
## Xác định giá trị ngõ ra

- Cho mạch có biểu thức  $x = \overline{ABC}(A+D)$
- Xác định giá trị ngõ ra x khi A=0, B=1, C=1, D=1
- Giá trị ngõ ra có thể được xác định



## Thiết lập bảng chân trị

- Ví dụ hãy thiết lập bảng chân trị từ sơ đồ mạch logic sau đây



$2^{\text{INPUTS}}$  = Số trạng thái ngõ vào  
 $2^3$  = 8 trạng thái

A	B	C	x
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

28

## Thiết lập bảng chân trị

A	B	C	x
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

29

## Thiết lập mạch từ biểu thức

- Hãy thiết kế một mạch logic được xác định bởi biểu thức:  **$y = AC + BC + \bar{A}BC$**
- Khi một mạch được định nghĩa bởi biểu thức logic, ta có thể thiết kế mạch logic trực tiếp từ biểu thức đó.
- Biểu thức gồm 3 thành phần OR với nhau.
- Ngõ vào của cổng OR là ngõ ra của các cổng AND

30

## Thiết lập mạch từ biểu thức

(a)  $y = AC + BC + \bar{A}BC$

(b)  $y = AC + BC + \bar{A}BC$

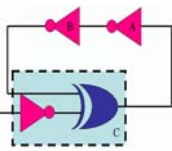
31

## Thiết lập mạch từ biểu thức

- Ví dụ hãy thiết lập mạch logic cho biểu thức  $x = (A + B)(\bar{B} + C)$

$x = (A + B)(\bar{B} + C)$

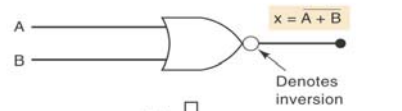
32



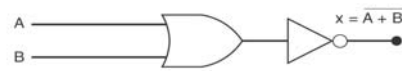
# Cổng NOR

- Biểu thức Boolean của cổng NOR

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



(a) ↓

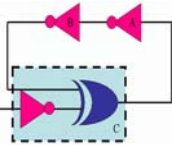


(b)

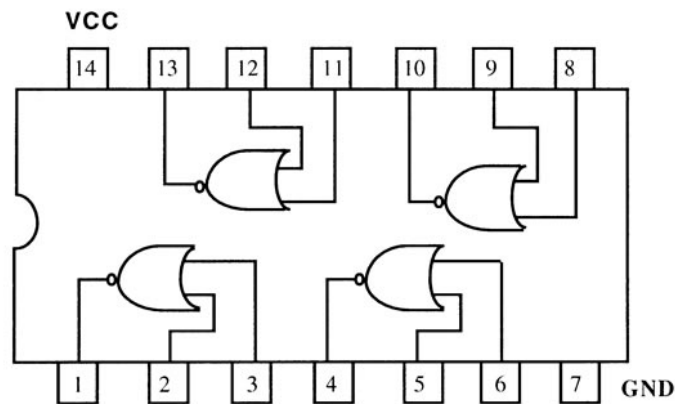
A	B	OR	
		A + B	$\overline{A + B}$
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

(c)

33



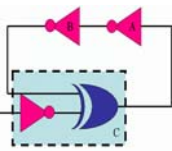
# IC cổng NOR 74LS02



74LS02

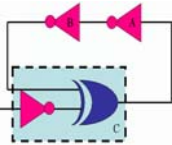
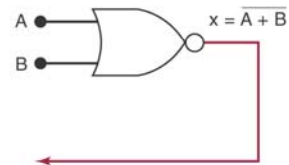
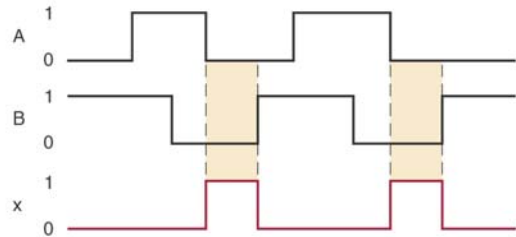
34





### Ví dụ 3-9

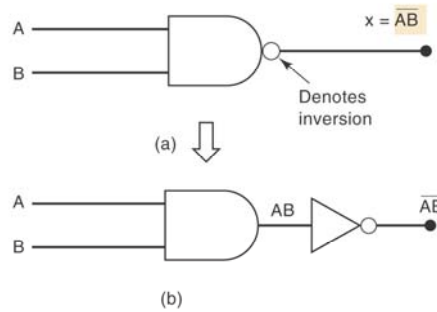
- Biểu đồ thời gian cho cổng NOR.



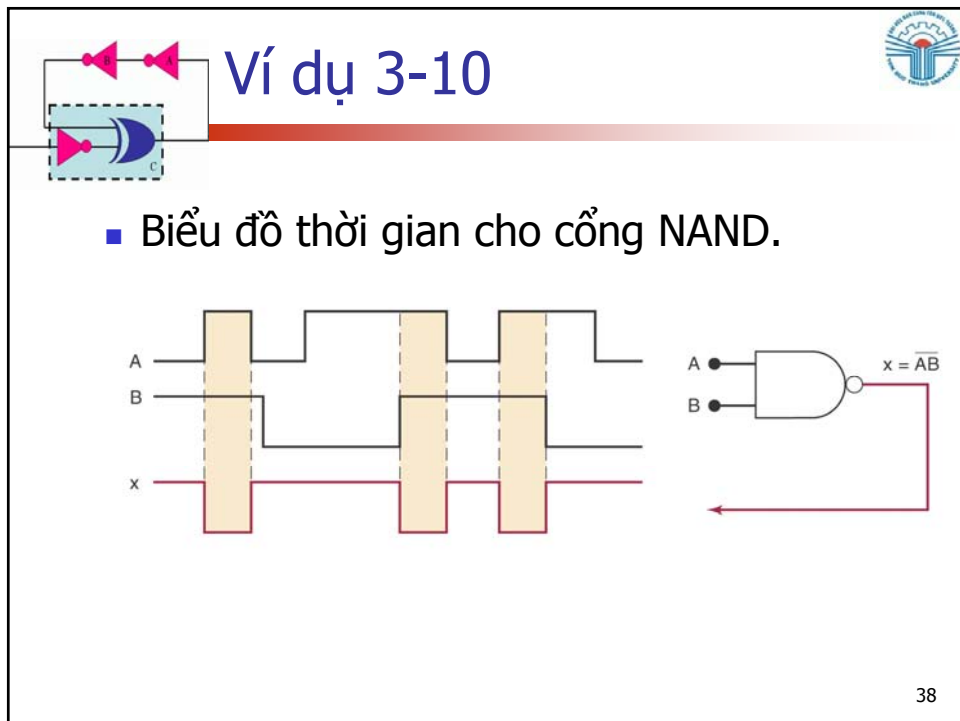
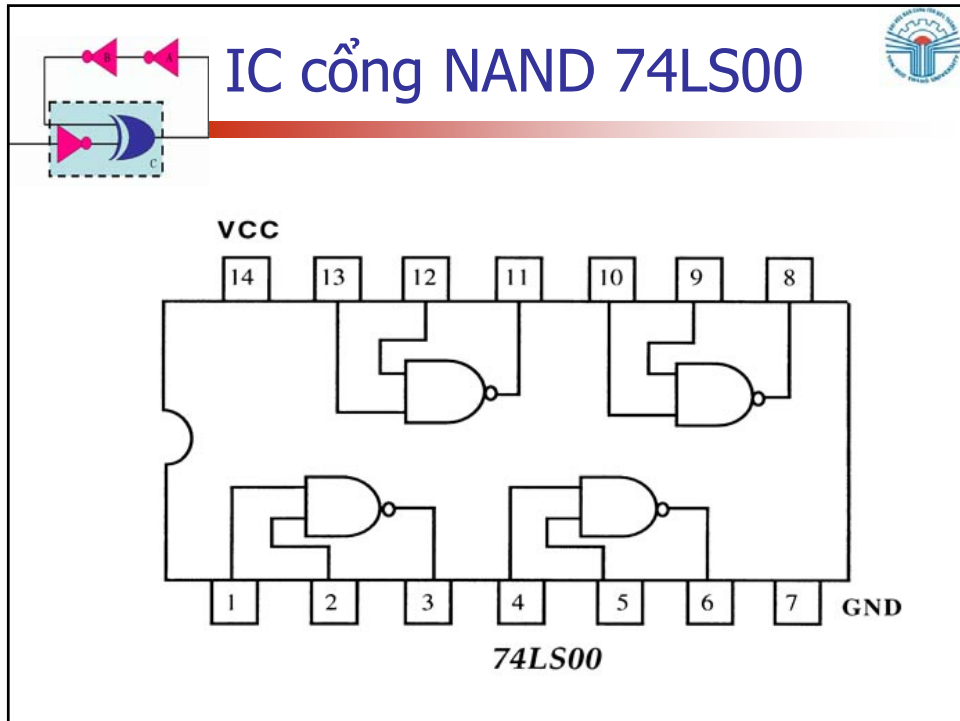
### Cổng NAND

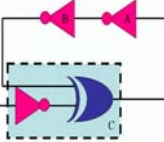
- Biểu thức Boolean của cổng NAND

$$x = \overline{A * B}$$



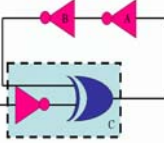
A	B	AND		NAND	
		AB	AB		
0	0	0	1		
0	1	0	1		
1	0	0	1		
1	1	1	0		




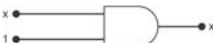









## Các định lý cơ bản trong đại số Boolean

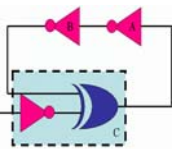
39

## Các định lý đơn biến

$x * 0 = 0$	(1) $x * 0 = 0$	
$x * 1 = x$	(2) $x * 1 = x$	
$x * x = x$	(3) $x * x = x$	
$x * \bar{x} = 0$	(4) $x * \bar{x} = 0$	
$x + 0 = x$	(5) $x + 0 = x$	
$x + 1 = 1$	(6) $x + 1 = 1$	
$x + x = x$	(7) $x + x = x$	
$x + \bar{x} = 1$	(8) $x + \bar{x} = 1$	

40



## Các định lý nhiều biến



- Luật giao hoán

$$x * y = y * x$$

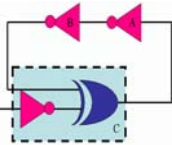
$$x + y = y + x$$

- Luật kết hợp

$$(x * y) * z = x * (y * z)$$

$$(x + y) + z = x + (y + z)$$

41



## Các định lý nhiều biến (tt)



- Luật phân phối

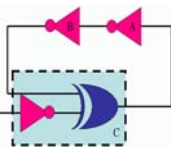
$$x * (y + z) = xy + xz$$

$$(x + y)(w + z) = xw + xz + yw + yz$$

- Luật hoàn nguyên

$$\overline{\overline{x}} = x$$

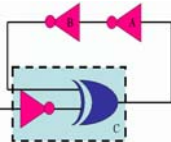
42



## Một số công thức thường dùng

- a)  $x.y + x.\bar{y} = x$
- b)  $x + x.y = x$
- c)  $x + \bar{x}.y = x + y$

43



## Định lý DeMORGAN

Định lý DeMORGAN 2 biến

$$\overline{x.y} = \bar{x} + \bar{y}$$


$$\overline{x + y} = \bar{x}.\bar{y}$$

Định lý DeMorGAN nhiều biến

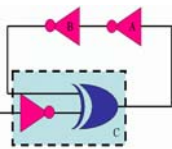
$$\overline{x.y.z.w \dots} = \bar{x} + \bar{y} + \bar{z} + \bar{w} \dots$$

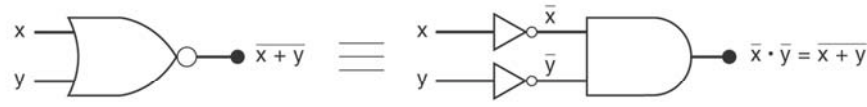
$$\overline{\bar{x} + \bar{y} + \bar{z} + \dots} = x.y.z \dots$$

44

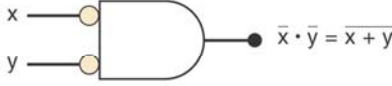


## Áp dụng định lý DeMORGAN






(a)

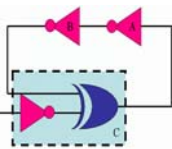


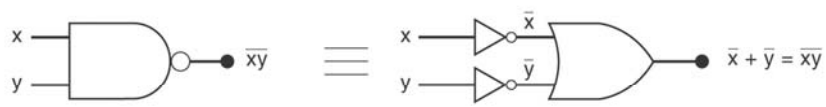
(b)

45

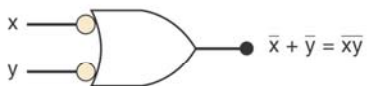


## Áp dụng định lý DeMORGAN





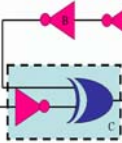
(a)



(b)

46

## Sự đa nhiệm của cổng NAND



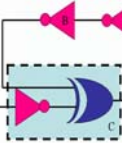
(a)  $x = \overline{A \cdot A} = \overline{A}$  → INVERTER

(b)  $x = \overline{\overline{A} \cdot \overline{B}} = AB$  → AND

(c)  $x = \overline{\overline{\overline{A} \cdot \overline{B}}} = A + B$  → OR

47

## Sự đa nhiệm của cổng NOR

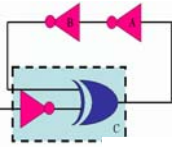


(a)  $x = \overline{A + A} = \overline{A}$  → INVERTER

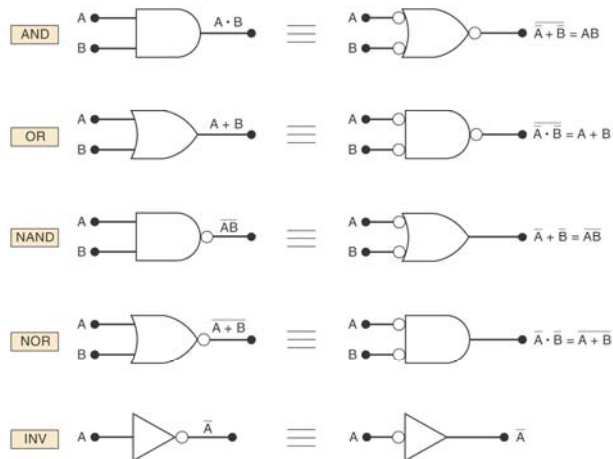
(b)  $x = \overline{\overline{A + B}} = A + B$  → OR

(c)  $x = \overline{\overline{\overline{A} + \overline{B}}} = \overline{\overline{A} + \overline{B}} = AB$  → AND

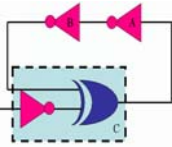
48



## Miêu tả cổng logic



49

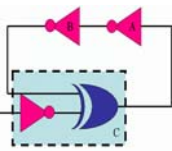


## Miêu tả cổng logic (tt)

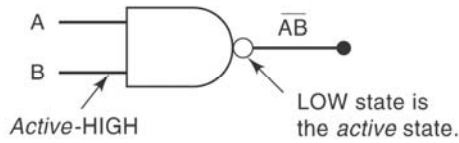
- Khi một ngõ vào hay ngõ ra trên cổng logic có ký hiệu vòng tròn thì ngõ vào hay ngõ ra đó được gọi là tích cực mức thấp.
- Trường hợp ngược lại, không có vòng tròn, thì gọi là tích cực mức cao.

50



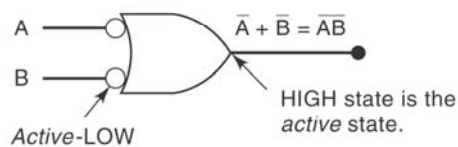


# Miêu tả cổng logic (tt)



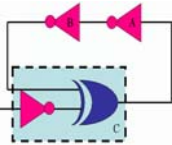
Output goes LOW only when *all* inputs are HIGH.

(a)

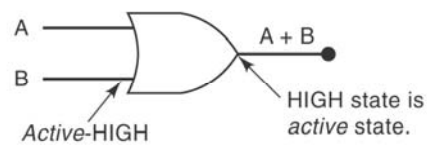


Output is HIGH when *any* input is LOW.

(b)

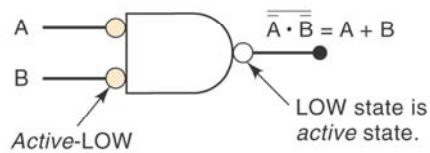


# Miêu tả cổng logic (tt)



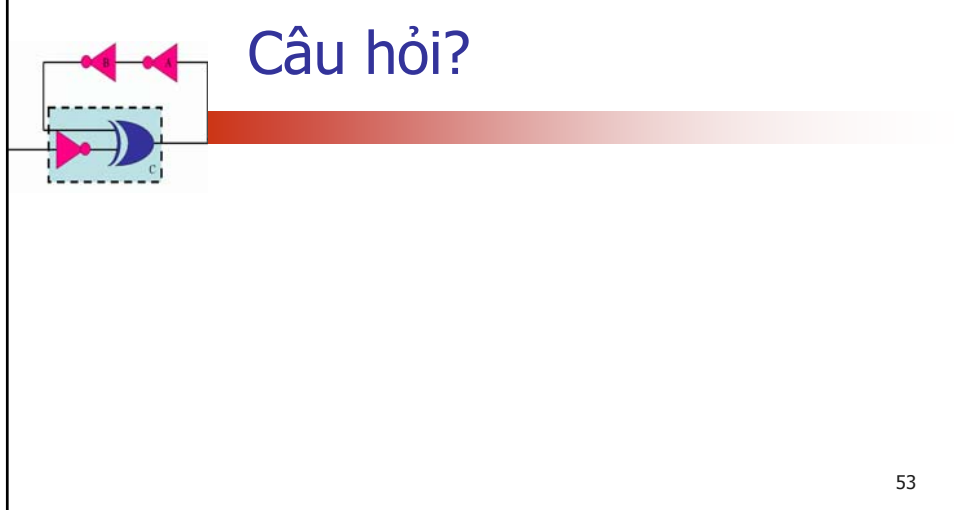
Output goes HIGH when *any* input is HIGH.

(a)



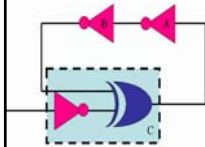
Output goes LOW only when *all* inputs are LOW.

(b)





## Chương 4 Mạch logic

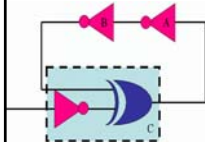


Th.S Đặng Ngọc Khoa  
Khoa Điện - Điện Tử

1

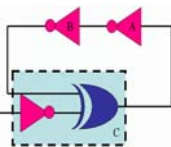


## Biểu diễn bằng biểu thức đại số



- Một hàm logic  $n$  biến bất kỳ luôn có thể biểu diễn dưới dạng:
  - Tổng của các tích (**Chuẩn tắc tuyển** - CTT): là dạng tổng của nhiều thành phần mà mỗi thành phần là tích của đầy đủ  $n$  biến.
  - Tích của các tổng (**Chuẩn tắc hội** - CTH): là dạng tích của nhiều thành phần mà mỗi thành phần là tổng của đầy đủ  $n$  biến.

2



## Biểu diễn bằng biểu thức đại số



Vị trí	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	0

### ■ Dạng chuẩn tắc tuyến

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

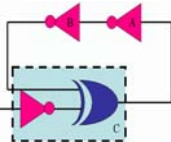
$$F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

### ■ Dạng chuẩn tắc hội

$$F = \prod (0, 3, 4, 7)$$

$$F = (A+B+C)(A+\bar{B}+\bar{C})(\bar{A}+B+C)(\bar{A}+\bar{B}+\bar{C})$$

3

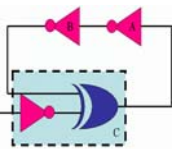


## Biểu diễn bằng biểu thức đại số



Chuẩn tắc tuyến	Chuẩn tắc hội
$\Sigma$	$\Pi$
Tổng của các tích	Tích của các tổng
Lưu ý các giá trị 1	Lưu ý các giá trị 0
X = 0 ghi $\bar{X}$	X = 0 ghi X
X = 1 ghi X	X = 1 ghi $\bar{X}$

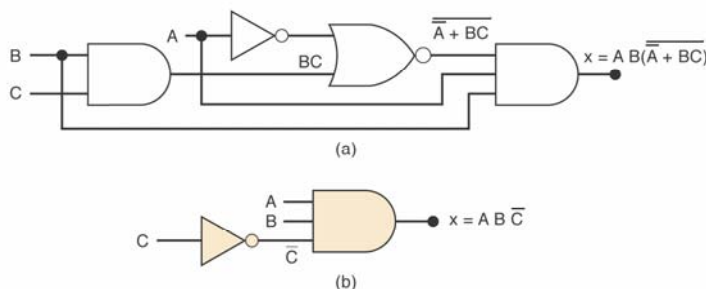
4



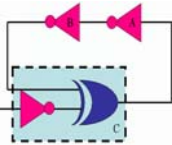
## Rút gọn mạch logic



- Làm cho biểu thức logic đơn giản nhất và do vậy mạch logic sử dụng ít cổng logic nhất.
- Hai mạch sau đây là tương đương nhau



5

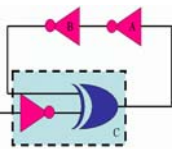


## Phương pháp rút gọn



- Có hai phương pháp chính để rút gọn một biểu thức logic.
  - **Phương pháp biến đổi đại số:** sử dụng các định lý và các phép biến đổi Boolean để rút gọn biểu thức.
  - **Phương pháp bìa Karnaugh:** sử dụng bìa Karnaugh để rút gọn biểu thức logic

6



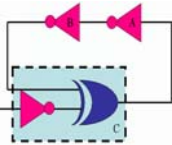
## Phương pháp biến đổi đại số



- Sử dụng các định lý và các phép biến đổi Boolean để rút gọn biểu thức.
- Ví dụ:

Biểu thức ban đầu	Rút gọn
$ABC + AB'(A'C)'$	
$ABC + ABC' + AB'C$	
$A'C(A'BD)'+A'BC'D'+AB'C$	
$(A'+B)(A+B+D)D'$	

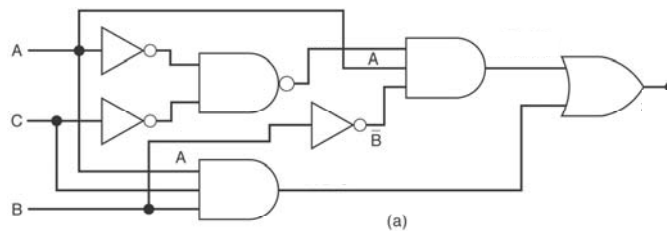
7



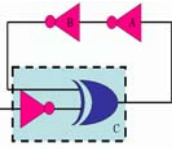
## Ví dụ 4-1



- Hãy rút gọn mạch logic sau



8

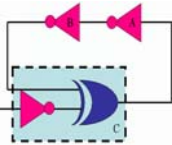


## Bài toán thiết kế

Hãy thiết kế một mạch logic có:

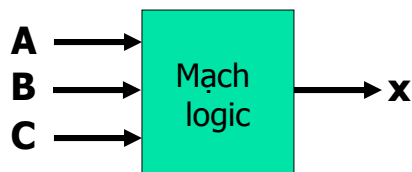
- Ba ngõ vào
- Một ngõ ra
- Ngõ ra ở mức cao chỉ khi đa số ngõ vào ở mức cao

9



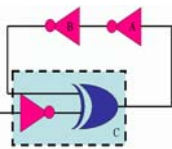
## Trình tự thiết kế

- Bước 1: Thiết lập bảng chân trị.



A	B	C	x
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

10



## Trình tự thiết kế



- Bước 2: Thiết lập phương trình từ bảng chân trị.

A	B	C	x
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

$$x = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

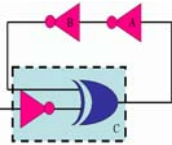
$$\bar{A}.B.C$$

$$A.\bar{B}.C$$

$$A.B.\bar{C}$$

$$A.B.C$$

11



## Trình tự thiết kế



- Bước 3: Rút gọn biểu thức logic

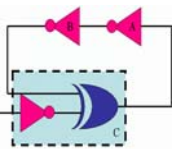
$$x = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$x = \bar{A}BC + ABC + A\bar{B}C + ABC + AB\bar{C} + ABC$$

$$x = BC + AC + AB$$

12



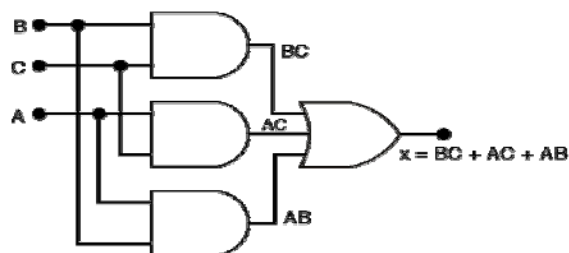


## Trình tự thiết kế

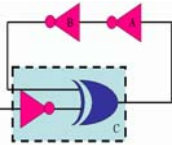


- Bước 4: Vẽ mạch logic ứng với biểu thức logic vừa rút gọn

$$x = BC + AC + AB$$



13

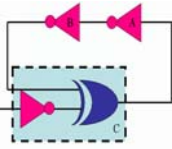


## Ví dụ 4-1

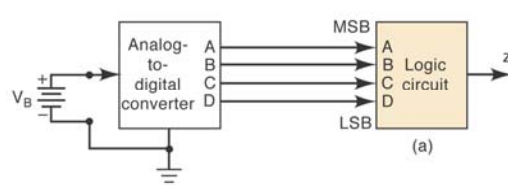


- Hãy thiết kế một mạch logic có 4 ngõ vào A, B, C, D và một ngõ ra. Ngõ ra chỉ ở mức cao khi điện áp (được miêu tả bởi 4 bit nhị phân ABCD) lớn hơn 6.

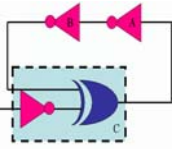
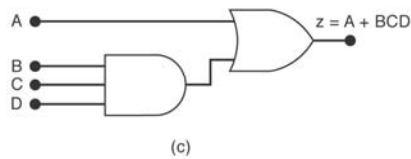
14



# Kết quả

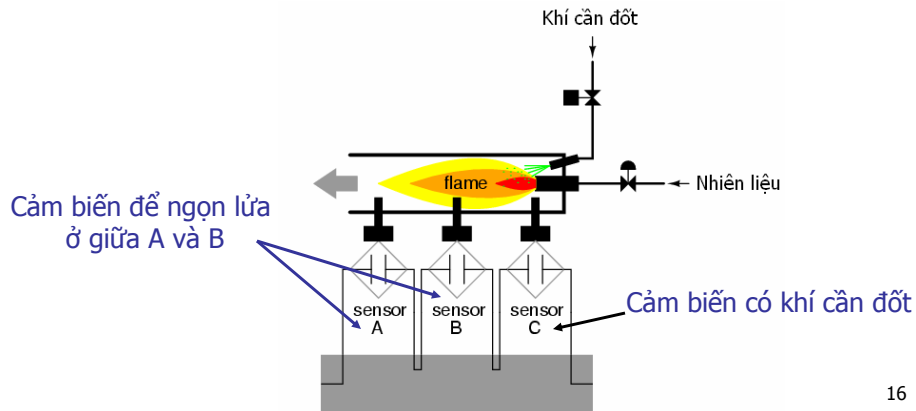


	A	B	C	D	z
(0)	0	0	0	0	0
(1)	0	0	0	1	0
(2)	0	0	1	0	0
(3)	0	0	1	1	0
(4)	0	1	0	0	0
(5)	0	1	0	1	0
(6)	0	1	1	0	0
(7)	0	1	1	1	1 → $\bar{A}BCD$
(8)	1	0	0	0	1 → $ABCD$
(9)	1	0	0	1	1 → $ABCD$
(10)	1	0	1	0	1 → $ABCD$
(11)	1	0	1	1	1 → $ABCD$
(12)	1	1	0	0	1 → $ABCD$
(13)	1	1	0	1	1 → $ABCD$
(14)	1	1	1	0	1 → $ABCD$
(15)	1	1	1	1	1 → $ABCD$



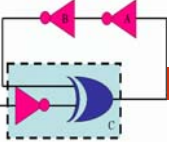
# Ví dụ 4-3

- Thiết kế mạch logic điều khiển mạch phun nhiên liệu trong mạch đốt như sau:





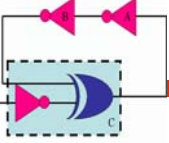
# Bìa Karnaugh



17

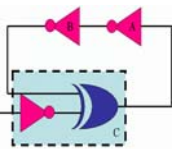


# Phương pháp bìa Karnaugh



- Giống như bảng chân trị, bìa Karnaugh là một cách để thể hiện mối quan hệ giữa các mức logic ngõ vào và ngõ ra.
- Bìa Karnaugh là một phương pháp được sử dụng để đơn giản biểu thức logic.
- Phương pháp này dễ thực hiện hơn phương pháp đại số.
- Bìa Karnaugh có thể thực hiện với bất kỳ số ngõ vào nào, nhưng trong chương trình chỉ khảo sát số ngõ vào nhỏ hơn 6.

18

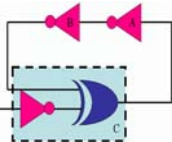


## Định dạng bìa Karnaugh



- Mỗi một trường hợp trong bảng chân trị tương ứng với 1 ô trong bìa Karnaugh
- Các ô trong bìa Karnaugh được đánh số sao cho 2 ô kề nhau chỉ khác nhau 1 giá trị.
- Do các ô kề nhau chỉ khác nhau 1 giá trị nên chúng ta có thể nhóm chúng lại để tạo một thành phần đơn giản hơn ở dạng tổng các tích.

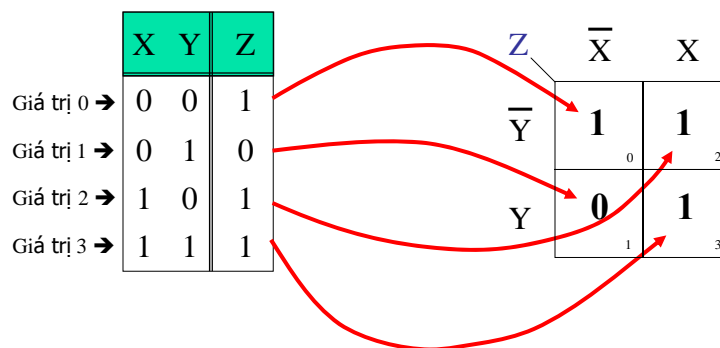
19



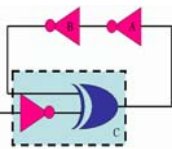
## Bảng chân trị $\Rightarrow$ K-map



- Một ví dụ tương ứng giữa bảng chân trị và bìa Karnaugh



20



## Xác định giá trị các ô



	$\bar{X}$	X
$\bar{Y}$	1	0
Y	0	0

 $\bar{X} \bar{Y}$ 

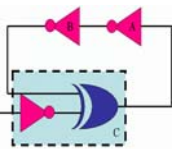
	$\bar{X}$	X
$\bar{Y}$	0	1
Y	0	0

 $X \bar{Y}$ 
  

	$\bar{X}$	X
$\bar{Y}$	0	0
Y	1	0

 $\bar{X} Y$ 

	$\bar{X}$	X
$\bar{Y}$	0	0
Y	0	1

 $X Y$ 


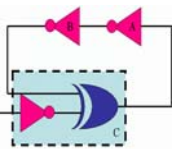
## Nhóm các ô kề nhau



	$\bar{X} \bar{Y}$	$\bar{X} X$	$X \bar{Y}$
$\bar{Y}$	1	1	
Y	0	0	

$$Z = \bar{X} \bar{Y} + X \bar{Y} = \bar{Y} (\bar{X} + X) = \bar{Y}$$

	$\bar{X}$	X
$\bar{Y}$	1	
Y	0	0

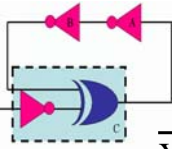


## Nhóm các ô lại với nhau



- Nhóm 2 ô "1" kề nhau, loại ra biến xuất hiện ở cả hai trạng thái bù và không bù.
- Nhóm 4 ô "1" kề nhau, loại ra 2 biến xuất hiện ở cả hai trạng thái bù và không bù.
- Nhóm 8 ô "1" kề nhau, loại ra 3 biến xuất hiện ở cả hai trạng thái bù và không bù.
- ...

23



## K-map 2 biến: nhóm 2



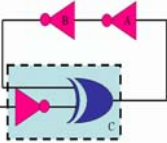
	$\bar{X}$	$X$	
$\bar{Y}$	1	1	$\bar{Y}$
$Y$	0	0	

	$\bar{X}$	$X$	
$\bar{Y}$	1	0	$\bar{X}$
$Y$	1	0	

	$\bar{X}$	$X$	
$\bar{Y}$	0	0	
$Y$	1	1	$Y$

	$\bar{X}$	$X$	
$\bar{Y}$	0	1	
$Y$	0	1	$X$

24

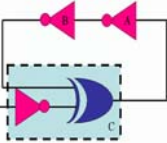


## K-map 2 biến: nhóm 4

	$\bar{X}$	$X$
$\bar{Y}$	1	1
$Y$	1	1

A red square highlights all four cells containing '1' in the K-map. A red arrow points from the top-right corner of this square to the number '1'.

25



## Ví dụ K-map 2 biến

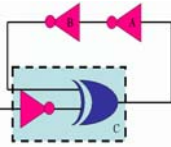
R	S	T
0	0	1
0	1	0
1	0	1
1	1	0

	$\bar{R}$	$R$
$\bar{S}$	1	1
$S$	0	0

A red square highlights the top two cells (1,1) and (1,2) of the K-map. A red arrow points from the top-left corner of this square to the number '1'.

$$T = F_{(R,S)} = \bar{S}$$

26

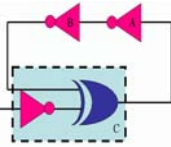


## K-map 3 biến

	A	B	C	Y
0 →	0	0	0	1
1 →	0	0	1	0
2 →	0	1	0	1
3 →	0	1	1	1
4 →	1	0	0	0
5 →	1	0	1	0
6 →	1	1	0	1
7 →	1	1	1	0

Y	$\bar{A} \bar{B}$	$\bar{A} B$	$A B$	$A \bar{B}$
$\bar{C}$	1 <sub>0</sub>	1 <sub>2</sub>	1 <sub>6</sub>	0 <sub>4</sub>
C	0 <sub>1</sub>	1 <sub>3</sub>	0 <sub>7</sub>	0 <sub>5</sub>

27

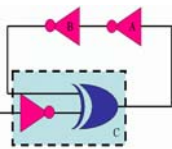


## K-map 3 biến: nhóm 2

	$\bar{A} \bar{B}$	$\bar{A} B$	$A B$	$A \bar{B}$
$\bar{C}$	0	0	0	0
C	0	0	0	0

28



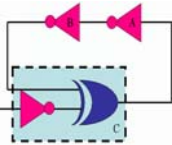


## K-map 3 biến: nhóm 4



	$\bar{A} \bar{B}$	$\bar{A} B$	$A B$	$A \bar{B}$
$\bar{C}$	0	0	0	0
$C$	0	0	0	0

29

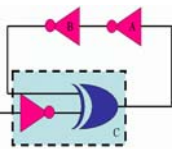


## K-map 3 biến: nhóm 8



	$\bar{A} \bar{B}$	$\bar{A} B$	$A B$	$A \bar{B}$
$\bar{C}$	1	1	1	1
$C$	1	1	1	1

30



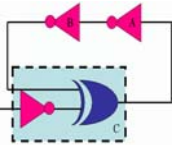
## Bìa Karnaugh 4 biến



A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

		AB			
		00	01	11	10
CD	00				
	01				
	11				
	10				

31



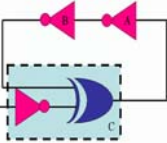
## Bìa Karnaugh 4 biến




Lưu ý các ký hiệu trong bìa Karnaugh

		AB			
		00	01	11	10
CD	00				
	01				
	11				
	10				

32



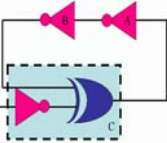
## Bìa Karnaugh 4 biến




A	B	C	D	F
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	0

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

33



## K-map 4 biến: nhóm 2

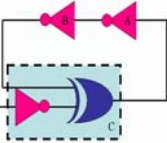


		AB			
		00	01	11	10
CD	00	0	0	1	1
	01	0	0	0	0
11	0	0	0	0	
10	1	0	0	1	

$\bar{A}\bar{C}\bar{D}$

$\bar{B}\bar{C}\bar{D}$

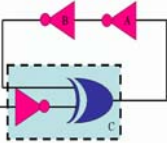
34



### K-map 4 biến: nhóm 4

		F				
		AB				
		00	01	11	10	
	CD					
	00	0	0	0	0	
	01	1	1	1	1	→ $\bar{C}D$
	11	0	0	0	0	
	10	0	0	0	0	

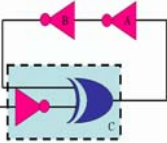
35



### K-map 4 biến: nhóm 4

		F				
		AB				
		00	01	11	10	
	CD					
	00	0	0	0	0	
	01	0	1	1	0	→ $BD$
	11	0	1	1	0	
	10	0	0	0	0	

36

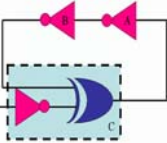


### K-map 4 biến: nhóm 4

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

Group 4:  $\bar{B}C$

37

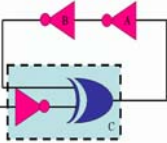


### K-map 4 biến: nhóm 4

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

Group 4:  $\bar{B}\bar{D}$

38

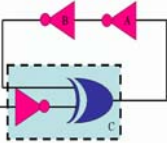


### K-map 4 biến: nhóm 8

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

A red circle highlights the four cells where F=1 (AB=01, 11, 10 for CD=00, 01, 11, 10). A red arrow points from this group to the label **B**.

39

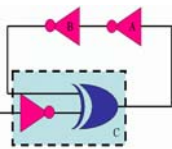


### K-map 4 biến: nhóm 8

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

A red circle highlights the four cells where F=1 (AB=00, 01 for CD=00, 01, 11, 10). A red arrow points from this group to the label  $\bar{A}$ .

40



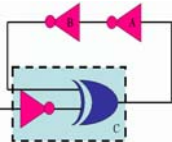
## K-map 4 biến: nhóm 8



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

A red arrow points from the group of 1s in the 10 column to the label  $\bar{B}$ .

41

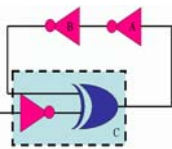


## Rút gọn bằng bìa Karnaugh



- Bước 1: Biểu diễn hàm đã cho trên bìa Karnaugh.
- Bước 2: Nhóm các ô có giá trị bằng 1 theo các quy tắc:
  - Tổng các ô là lớn nhất.
  - Tổng các ô phải là  $2^n$  (n nguyên).
  - Các ô này phải nằm kề nhau.

42

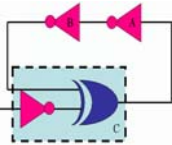


## Rút gọn bằng bìa Karnaugh



- Bước 3: Làm lại bước 2 cho đến khi tất cả các ô logic 1 đều được sử dụng.
- Bước 4: Xác định kết quả theo các quy tắc:
  - Mỗi nhóm sẽ là một tích của các biến.
  - Kết quả là tổng của các tích ở trên.

43



## Ví dụ 4-4



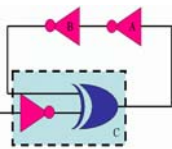
M	$\bar{J}\bar{K}$	$\bar{J}K$	$JK$	$J\bar{K}$
$\bar{L}$	1 <sub>0</sub>	1 <sub>2</sub>	0 <sub>6</sub>	0 <sub>4</sub>
L	0 <sub>1</sub>	1 <sub>3</sub>	0 <sub>7</sub>	1 <sub>5</sub>

$\bar{J}\bar{K}$                        $J\bar{K}L$

$$M = F_{(J,K,L)} = \bar{J}\bar{L} + \bar{J}K + J\bar{K}L$$

44





### Ví dụ 4-5

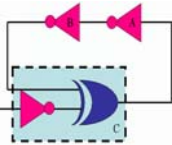


$\overline{B}\overline{C}$	$\overline{A}\overline{B}$	$\overline{A}B$	$AB$	$A\overline{B}$
$\overline{C}$	1	0	0	1
$C$	0	0	1	1

$AC$

$$Z = F_{(A,B,C)} = AC + \overline{B}\overline{C}$$

45



### Ví dụ 4-6



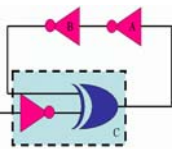
	$\overline{A}\overline{B}$	$\overline{A}B$	$AB$	$A\overline{B}$
$\overline{C}$	1	0	1	0
$C$	1	1	1	0

$\overline{A}\overline{B}$        $AB$

$$F_1 = F_{(A,B,C)} = \overline{A}\overline{B} + AB + \overline{A}C$$

$$F_2 = F_{(A,B,C)} = \overline{A}\overline{B} + AB + BC$$

46



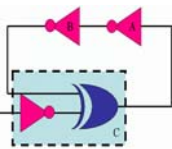
### Ví dụ 4-7



	$\bar{W}\bar{X}$	$\bar{W}X$	$WX$	$W\bar{X}$	
$\bar{Y}\bar{Z}$	1	1	1	0	$WX\bar{Y}$
$\bar{Y}Z$	0	0	1	0	
$YZ$	0	0	0	0	
$Y\bar{Z}$	1	1	0	1	$\bar{X}Y\bar{Z}$
$\bar{W}\bar{Z}$					

$$F1 = F_{(w,x,y,z)} = WX\bar{Y} + \bar{W}\bar{Z} + \bar{X}Y\bar{Z}$$

47



### Ví dụ 4-8

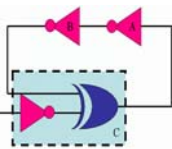


Rút gọn biểu thức sau đây:

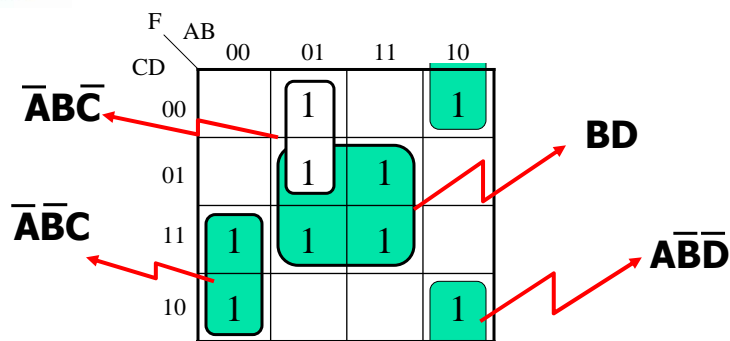
$$f(A,B,C,D) = \sum(2,3,4,5,7,8,10,13,15)$$

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

48

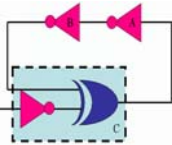


## Ví dụ 4-8



$$f(A,B,C,D) = \mathbf{BD} + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{D} + \bar{A}\bar{B}C$$

49



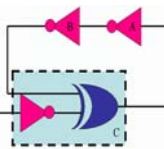
## Trạng thái Don't Care



- Một số mạch logic có đặc điểm: với một số giá trị ngõ vào xác định, giá trị ngõ ra không được xác định cụ thể.
- Trạng thái không xác định của ngõ ra được gọi là trạng thái Don't Care.
- Với trạng thái này, giá trị của nó có thể là 0 hoặc 1.
- Trạng thái Don't Care rất tiện lợi trong quá trình rút gọn bìa Karnaugh.

50

### Ví dụ trạng thái Don't Care



A	B	C	z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	x
1	0	0	x
1	0	1	1
1	1	0	1
1	1	1	1

(a)

	$\bar{A}\bar{B}$	$\bar{A}B$	$AB$	$A\bar{B}$
$\bar{C}$	0	0	1	x
C	0	x	1	1

(b)

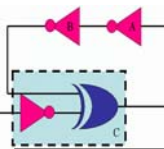
	$\bar{A}\bar{B}$	$\bar{A}B$	$AB$	$A\bar{B}$
$\bar{C}$	0	0	1	x
C	0	x	1	1

(c)

$z = A$

51

### Ví dụ 4-9



W	X	Y	Z	F2
0	0	0	0	1
0	0	0	1	x
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	x
0	1	1	0	0
0	1	1	1	x
1	0	0	0	x
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	x
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

	$\bar{W}\bar{X}$	$\bar{W}X$	$WX$	$W\bar{X}$
$\bar{Y}\bar{Z}$	1	X	0	1
$\bar{Y}Z$	0	X	X	0
$YZ$	X	1	1	1
$Y\bar{Z}$	X	1	1	0

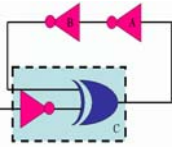
$\bar{X}\bar{Y}\bar{Z}$

$YZ$

$XY$

$F2 = F_{(w,x,y,z)} = \bar{X}\bar{Y}\bar{Z} + YZ + XY$

52



### Ví dụ 4-10

- Xác định biểu thức cho bảng chân trị sau đây

Dạng chuẩn tắc tuyến

$$f(A,B,C,D)$$

$$= \Sigma(1,3,4,7,11) + d(5,12,13,14,15)$$

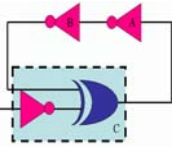
Dạng chuẩn tắc hội

$$f(A,B,C,D)$$

$$= \Pi(0,2,6,8,9,10) \cdot D(5,12,13,14,15)$$

A	B	C	D	F
0	0	0	0	0
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	x
0	1	1	0	0
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	x
1	1	0	1	x
1	1	1	0	x
1	1	1	1	x

53



### Ví dụ 4-10

$$f(A,B,C,D) = \Sigma(1,3,4,7,11) + d(5,12,13,14,15)$$

$$f(A,B,C,D) = (0,2,6,8,9,10) \cdot D(5,12,13,14,15)$$

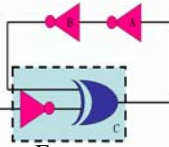
F	AB	00	01	11	10
CD	00		1	x	
01	1	x	x		
11	1	1	x	1	
10			x		

CTT

F	AB	00	01	11	10
CD	00	0		x	0
01		x	x		0
11				x	
10	0	0	x		0

CTH

54



### Ví dụ 4-10

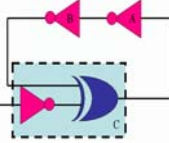
F	AB	00	01	11	10
CD	00		1	x	
	01	1	x	x	
	11	1	1	x	1
	10			x	

F	AB	00	01	11	10
CD	00	0		x	0
	01		x	x	0
	11			x	
	10	0	0	x	0

$$f(A,B,C,D) = CD + B\bar{C} + \bar{A}D$$

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

55



### K-map 5 biến

$$f(A,B,C,D,E) = \Sigma(0,2,4,7,10,12,13,18,23,26,28,29)$$

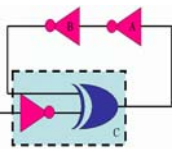
**A=0**

F	BC	00	01	11	10
DE	00	0	4	12	8
	01	1	5	13	9
	11	3	7	15	11
	10	2	6	14	10

**A=1**

F	BC	00	01	11	10
DE	00	16	20	28	24
	01	17	21	29	25
	11	19	23	31	27
	10	18	22	30	26

56



## K-map 5 biến



$$f(A,B,C,D,E) = \Sigma(0,2,4,7,10,12,13,18,23,26,28,29)$$

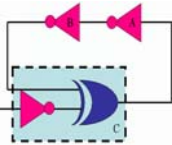
**A=0**

F	BC	00	01	11	10
DE	00	1	1	1	
	01			1	
	11		1		
	10	1			1

**A=1**

F	BC	00	01	11	10
DE	00			1	
	01			1	
	11		1		
	10	1			1

57



## K-map 5 biến



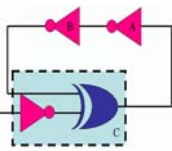
$$f(A,B,C,D,E) = \Sigma(0,2,4,7,10,12,13,18,23,26,28,29)$$

		<b>A=0</b>					<b>A=1</b>				
F	BC	00	01	11	10	F	BC	00	01	11	10
DE	00	1	1	1		DE	00			1	
	01			1			01			1	
	11		1				11		1		
	10	1			1		10	1			1

$\overline{ABDE}$  (points to cell 00,00 in A=0)  
 $BC\overline{D}$  (points to cells 11,00 and 11,01)  
 $\overline{BCDE}$  (points to cell 01,11)  
 $\overline{CDE}$  (points to cells 00,10 and 10,10)

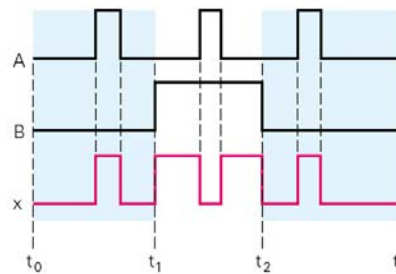
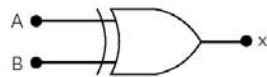
$$f(A,B,C,D) = \overline{ABDE} + BC\overline{D} + \overline{BCDE} + \overline{CDE}$$

58

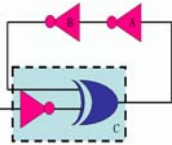


# Cổng EX-OR

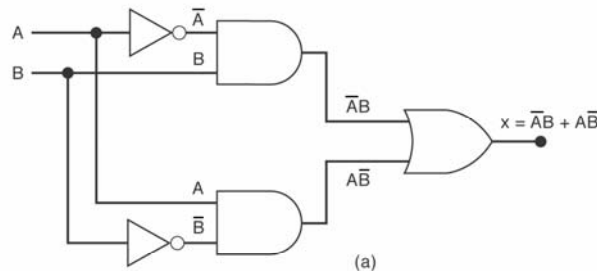
- Cổng EX-OR có hai ngõ vào.
- Ngõ ra của cổng EX-OR ở mức cao chỉ khi hai ngõ vào có giá trị khác nhau.



59

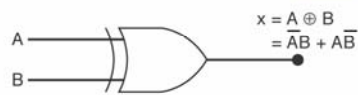


# Cổng EX-OR

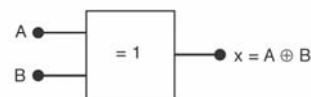


A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

XOR gate symbols



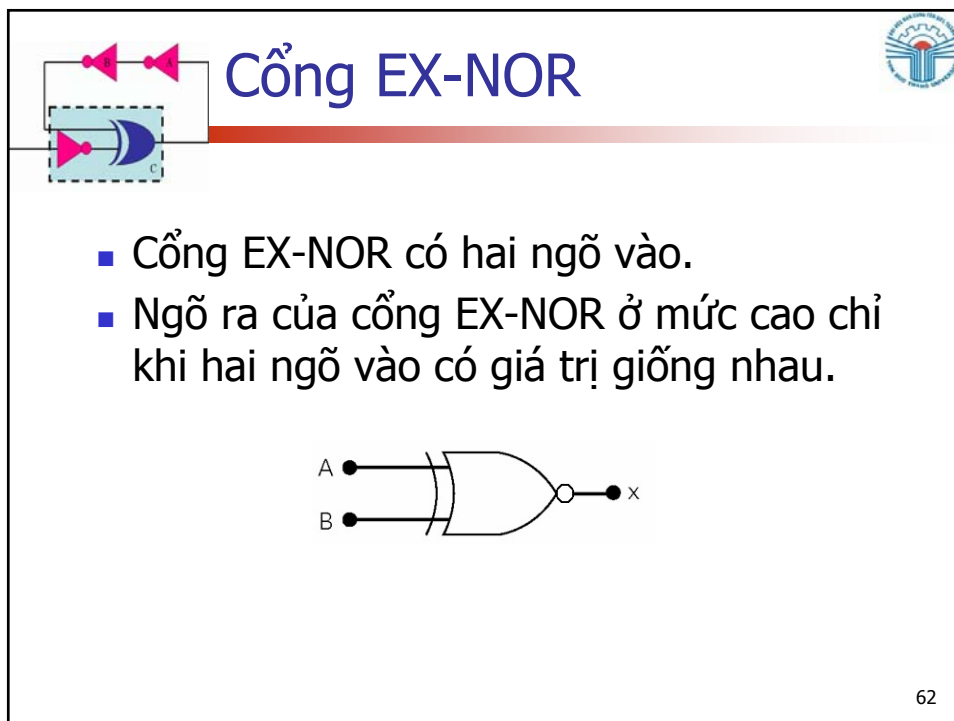
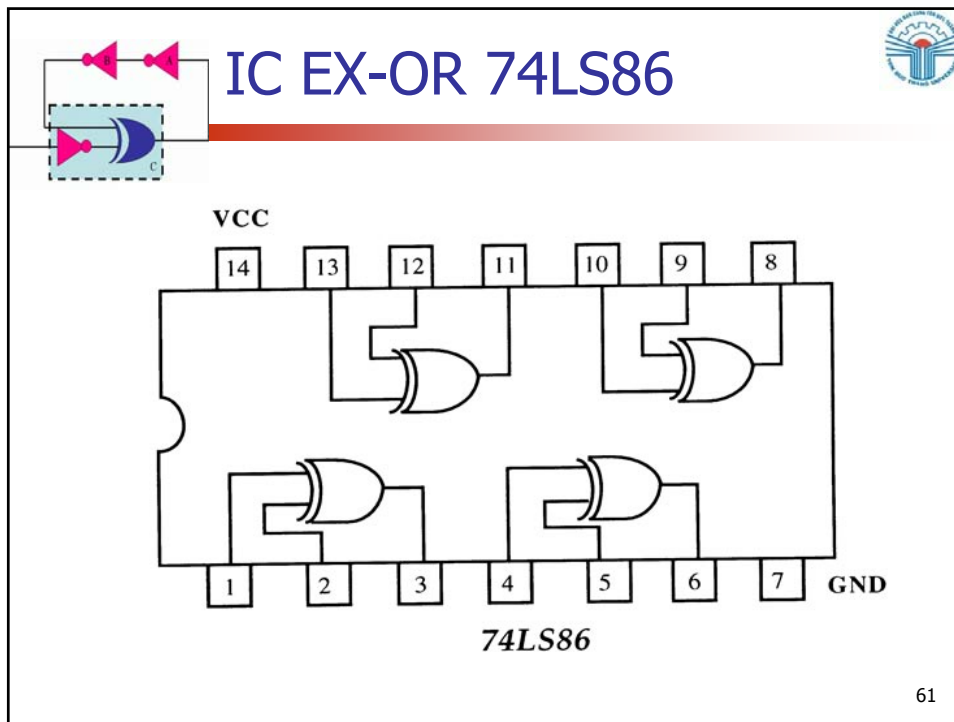
(b)

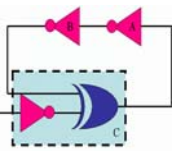


(c)

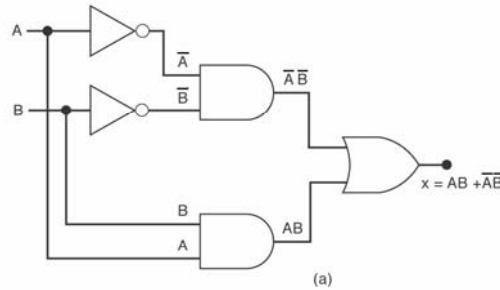
50





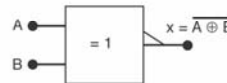
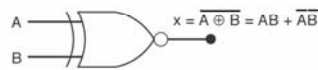


# Cổng EX-NOR



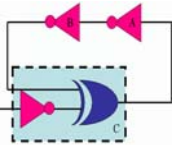
A	B	x
0	0	1
0	1	0
1	0	0
1	1	1

XNOR gate symbols



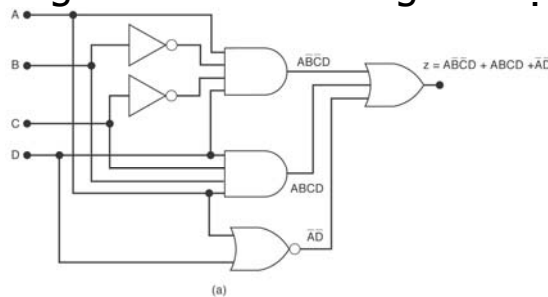
(b)

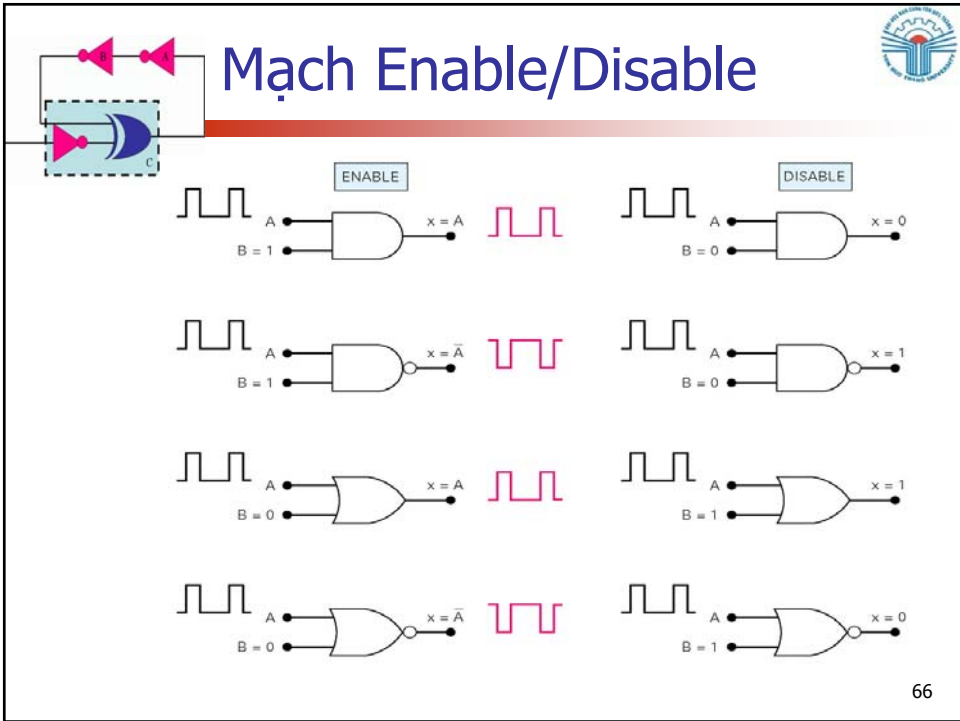
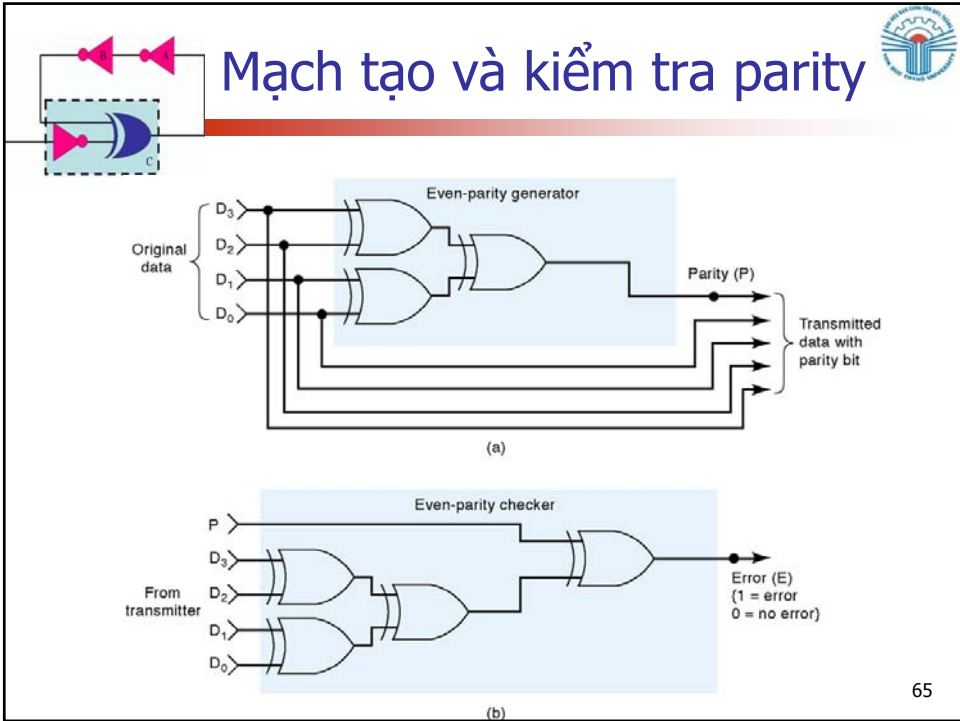
(c)

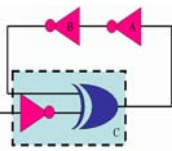


# Ví dụ 4-11

- Sử dụng cổng EX-NOR để đơn giản mạch logic sau



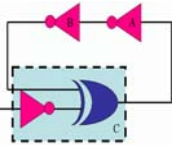




## Đặc điểm của IC số

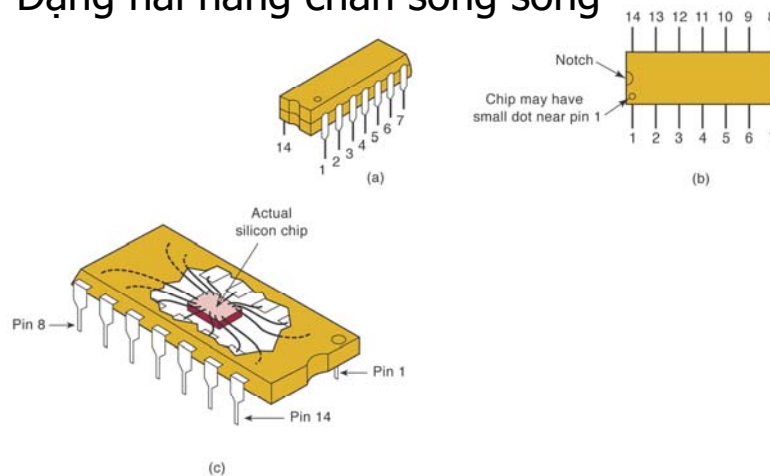
- IC được cấu tạo từ các điện trở, diode, transistor, các linh kiện này được đặt trên một lớp bán dẫn làm nền.
- Để tránh các tác động cơ học, hóa học, IC được đóng trong những vỏ silicon hoặc plastic.
- Chip thực tế nhỏ hơn hình dáng của nó rất nhiều

67

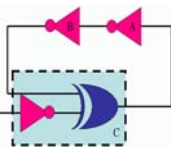


## Dạng cơ bản của IC số

- Dạng hai hàng chân song song



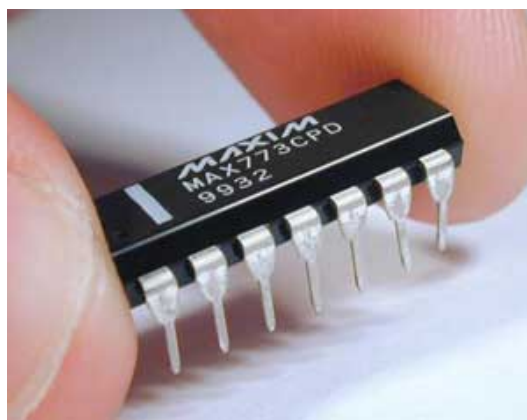
58



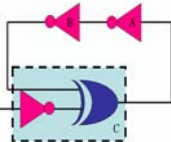
## Dạng cơ bản của IC số



- Dạng hai hàng chân song song



69



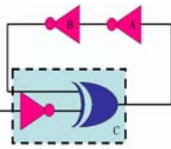
## Dạng cơ bản của IC số



- Dạng đóng vỏ hộp (flat pack)

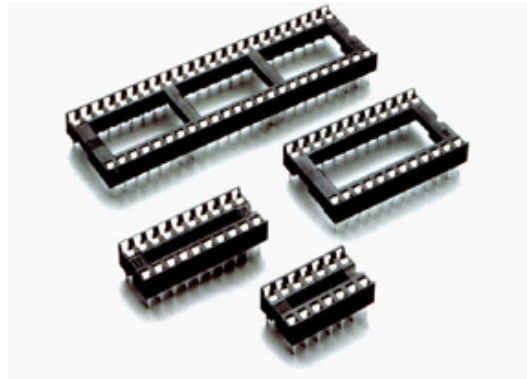


70

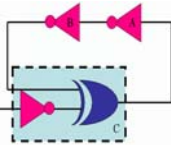


## Đế gắn IC

- Để thuận lợi trong quá trình lắp ráp và thay đổi, IC thường được gắn trên các đế.



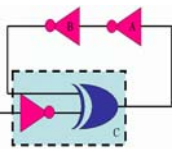
71



## Mạch số tích hợp (IC)

Độ tích hợp	Số cổng logic
Small-scale integration ( <b>SSI</b> )	<12
Medium-scale integration ( <b>MSI</b> )	12 to 99
Large-scale integration ( <b>LSI</b> )	100 to 9999
Very large-scale integration ( <b>VLSI</b> )	10,000 to 99,999
Ultra large-scale integration ( <b>ULSI</b> )	100,000 to 999,999
Giga-scale integration ( <b>GSI</b> )	1,000,000 or more

72

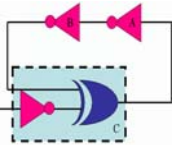


## IC số Bipolar và Unipolar

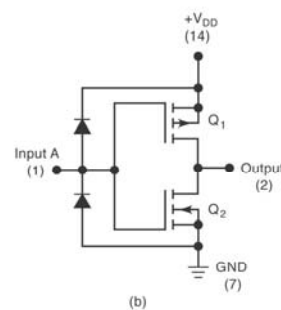
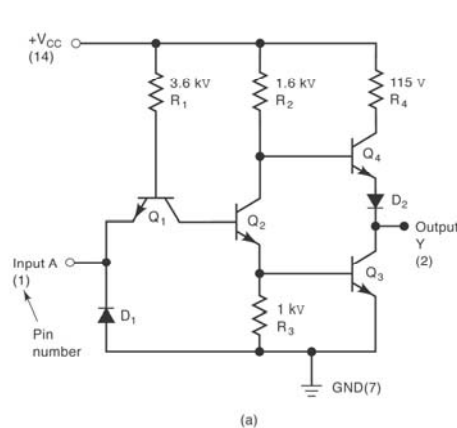


- IC số được phân thành IC bipolar và IC unipolar.
  - IC bipolar là những IC được tạo thành từ những transistor BJT (PNP hoặc NPN)
  - IC unipolar được tạo thành từ những transistor hiệu ứng trường (MOSFET)

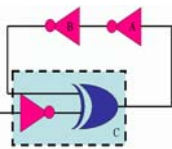
73



## Cổng NOT bipolar và unipolar



74

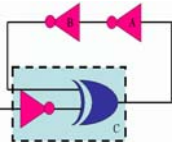


## Họ IC



- IC số được phân thành hai loại chính là TTL và CMOS.
  - Họ TTL là những IC bipolar (bảng 4-1)
  - Họ CMOS là những IC unipolar (bảng 4-2)

75



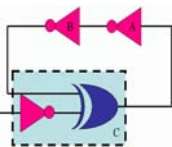
## Họ TTL (Bảng 4-1)



Phân loại TTL	Ký hiệu	Ví dụ IC
Standard TTL	74	7404 (NOT)
Schottky TTL	74S	74S04
Low-power Schottky TTL	74LS	74LS04
Advanced Schottky TTL	74AS	74AS04
Advanced low-power Schottky TTL	74ALS	74ALS04

76



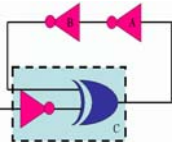


## Họ CMOS (Bảng 4-2)



Phân loại CMOS	Ký hiệu	Ví dụ IC
Metal-gate CMOS	40	4001 (NOR)
Metal-gate, pin-compatible with TTL	74C	74C02
Silicon-gate, pin-compatible with TTL, high-speed	74HC	74HC02
Silicon-gate, high-speed, pin-compatible and electrically compatible with TTL	74HCT	74HCT02
Advanced-performance CMOS, not pin or electrically compatible with TTL	74AC	74AC02
Advanced-performance CMOS, not pin but electrically compatible with TTL	74ACT	74ACT02

77

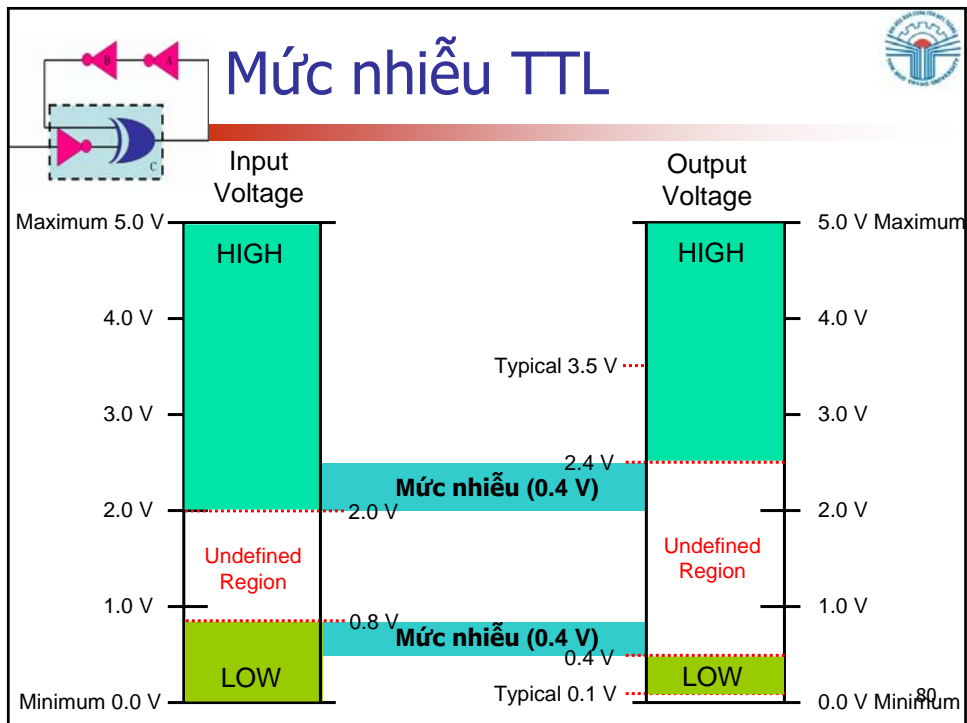
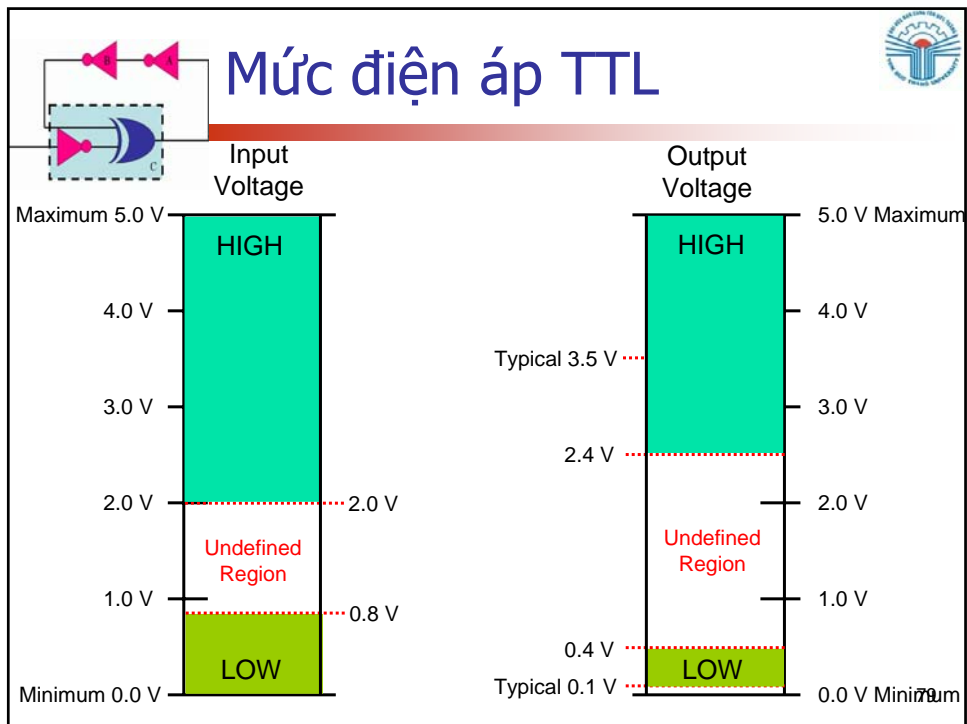


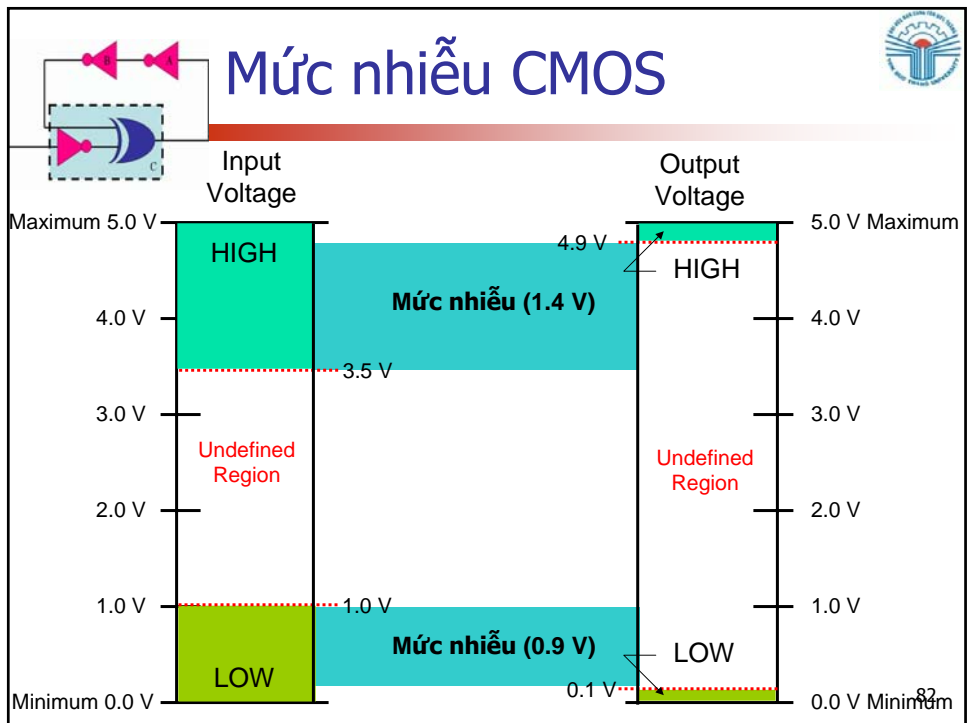
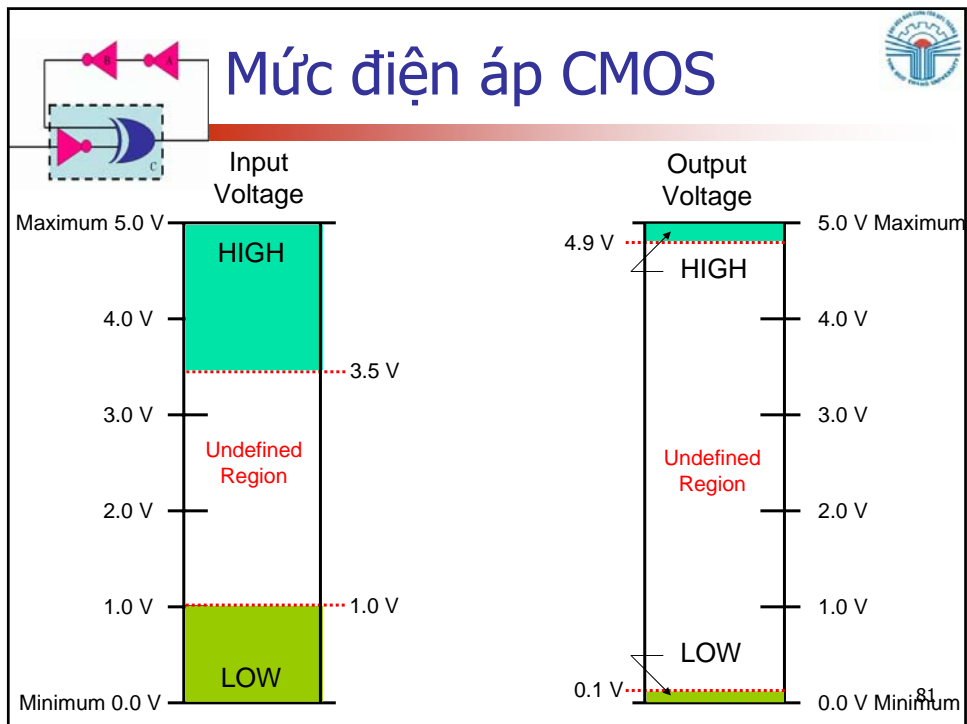
## Nguồn cung cấp và nối đất

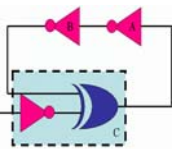


- Để có thể sử dụng được những IC số ta cần phải cung cấp nguồn cho nó.
  - Chân nguồn (power) ký hiệu là  $V_{CC}$  cho họ TTL và  $V_{DD}$  cho họ CMOS.
  - Chân đất (ground)

78



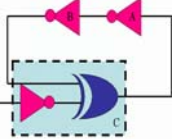




## Ngõ vào không kết nối



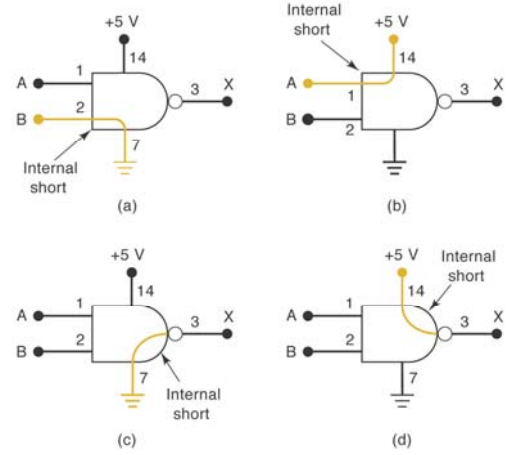
- Với họ TTL, ngõ vào không kết nối làm việc giống như mức logic 1, tuy nhiên khi đo thì điện áp DC tại chân đó nằm trong khoảng 1,4 – 1,8V.
- Với học CMOS tất cả các ngõ vào phải được kết nối.

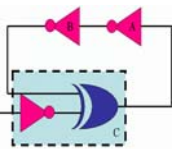


## Những lỗi bên trong IC



- Ngõ vào hoặc ra bị nối đến đất hoặc nguồn  $V_{CC}$

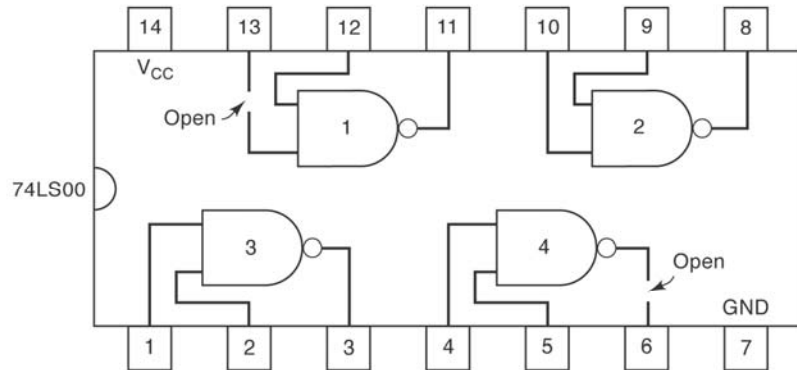




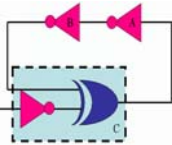
# Những lỗi bên trong IC



- Ngõ vào hoặc ra bị hở mạch



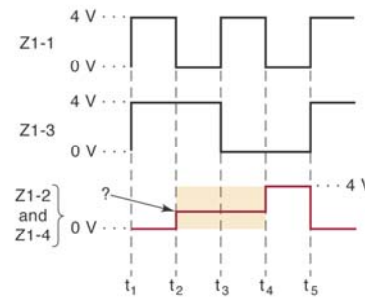
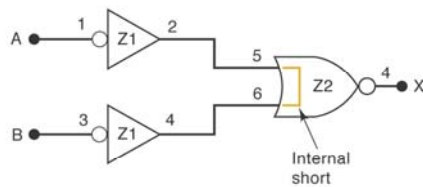
85



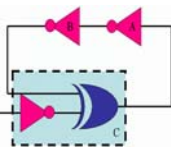
# Những lỗi bên trong IC



- Ngắn mạch giữa hai chân



86

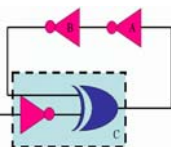


## Những lỗi bên ngoài IC



- **Đường dây tín hiệu bị hở mạch:** dây đứt, mối hàn không tốt, chân IC gãy, chân đế IC gãy.
- **Đường dây tín hiệu bị ngắn mạch:** do đường dây, mối hàn, board mạch bị đứt.
- **Nguồn cung cấp không đúng.**
- **Output loading:** khi ngõ ra kết nối với quá nhiều ngõ vào khác.

87



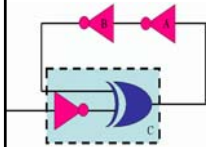
## Câu hỏi?



88



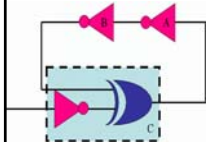
# Chương 5 Flip – Flops



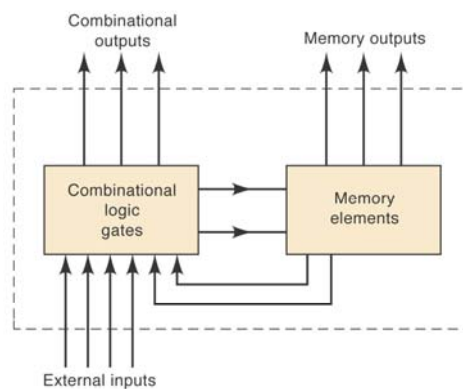
Th.S Đặng Ngọc Khoa  
Khoa Điện - Điện Tử

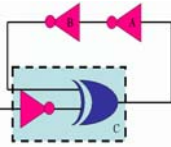


# Giới thiệu



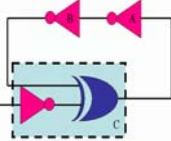
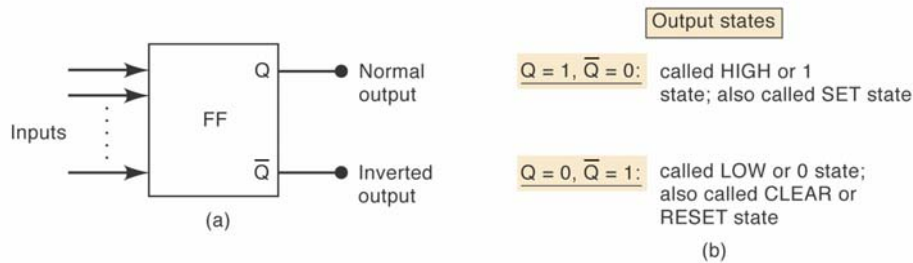
- Sơ đồ hệ thống số tổng quát bao gồm thành phần nhớ và các cổng logic





## Flip-Flops

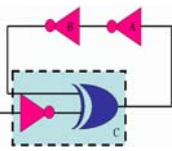
- Thành phần nhớ phổ biến nhất là các Flip-flop, flip-flop được cấu thành từ những cổng logic đơn giản.
- Ký hiệu tổng quát của một flip-flop



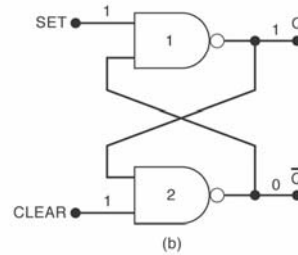
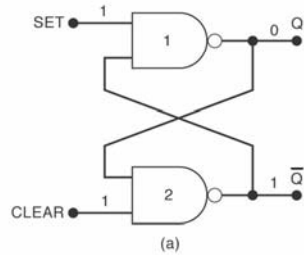
## Mạch chốt cổng NAND

- Mạch chốt cổng NAND là một flip-flop đơn giản.
- Mạch chốt có hai ngõ vào là set và clear (preset).
- Ngõ vào tích cực mức thấp, ngõ ra sẽ thay đổi trạng thái khi có xung thấp ở ngõ vào.
- Khi mạch ở trạng thái set
 
$$Q = 1 \text{ và } \bar{Q} = 0$$
- Khi mạch ở trạng thái clear (preset)
 
$$Q = 0 \text{ và } \bar{Q} = 1$$

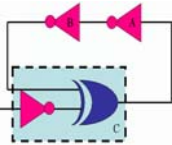




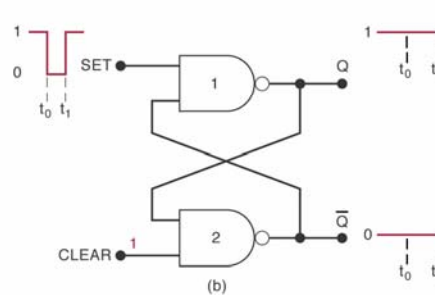
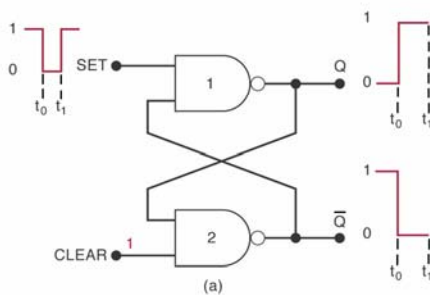
# Mạch chốt cổng NAND



Mạch chốt cổng NAND có hai trạng thái ổn định (trạng thái chốt) ứng với trường hợp  $SET = CLEAR = 1$ .



# Trạng thái SET mạch chốt



Khi ngõ vào SET chuyển từ trạng thái cao xuống trạng thái thấp, trong cả hai trường hợp ngõ ra Q sẽ ở trạng thái cao

## Trạng thái clear mạch chốt

(a)

(b)

Khi ngõ vào CLEAR chuyển từ trạng thái cao xuống trạng thái thấp, trong cả hai trường hợp ngõ ra Q sẽ ở trạng thái thấp

7

## Mạch chốt cổng NAND

(a)

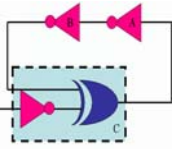
Set	Clear	Output
1	1	No change
0	1	Q = 1
1	0	Q = 0
0	0	Invalid*

\*Produces Q = Q-bar = 1.

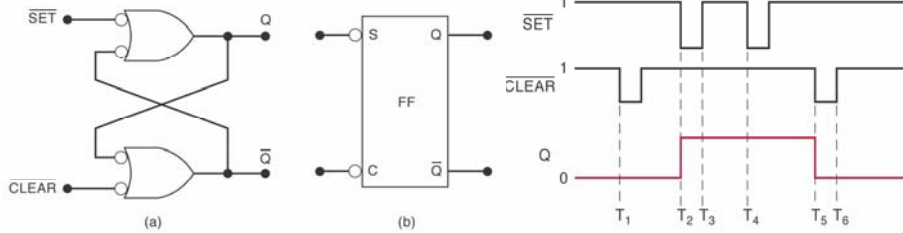
(b)

- SET = RESET = 1. Trạng thái ổn định, ngõ ra vẫn giữ trạng thái trước đó.
- SET = 0, RESET = 1. Q ở mức cao.
- SET = 1, RESET = 0. Q ở mức thấp.
- SET = RESET = 0. Ngõ ra không được xác định chính xác do cả hai trạng thái set và clear cùng tác động.

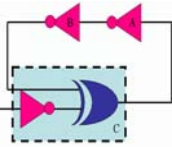
8



# Mô tả tương đương mạch chốt

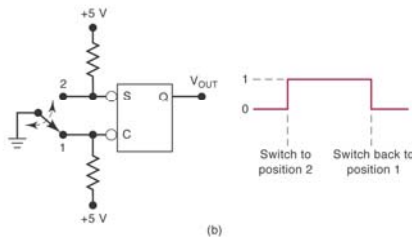
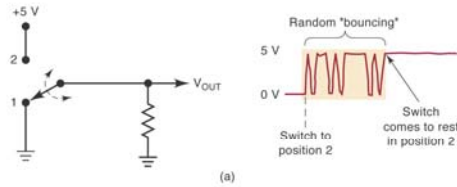


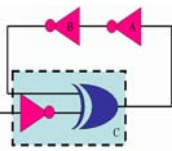
Ngõ ra mạch chốt nhớ trạng thái trước đó và ngõ ra chỉ có thể thay đổi khi một trong hai ngõ vào ở trạng thái tích cực



# Ví dụ 5-1

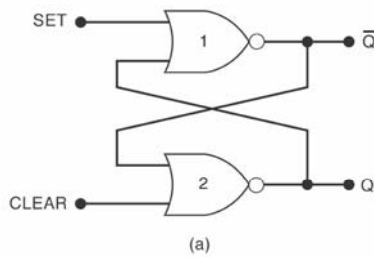
## ■ Khóa chống nảy





## Mạch chốt công NOR

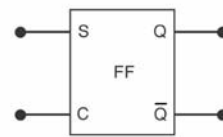
- Tương tự như mạch chốt công NAND chỉ khác vị trí hai ngõ ra  $Q$  và  $\bar{Q}$  được thay đổi cho nhau.
- Ngõ vào tích cực mức cao



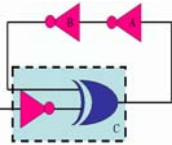
Set	Clear	Output
0	0	No change
1	0	$Q = 1$
0	1	$Q = 0$
1	1	Invalid*

\*Produces  $Q = \bar{Q} = 0$ .

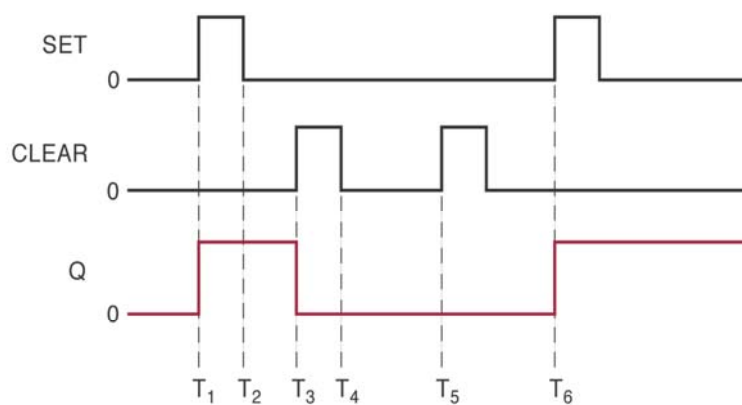
(b)

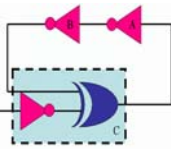


(c)



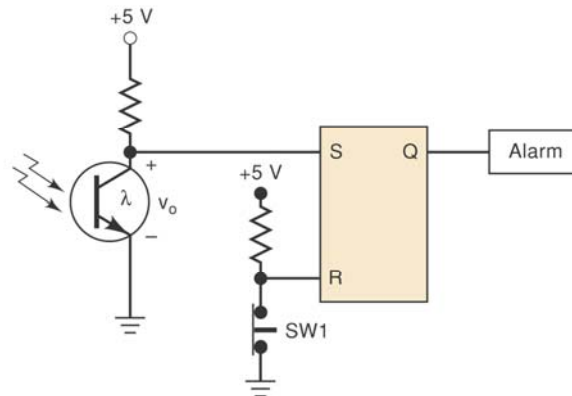
## Dạng sóng mạch chốt công NOR



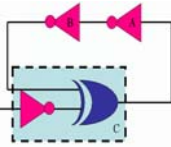


## Ví dụ 5-2

- Khi mất nguồn ánh sáng hệ thống sẽ báo động.
- Công tắc SW1 dùng để reset hệ thống



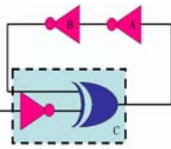
13



## Đồng bộ và bất đồng bộ

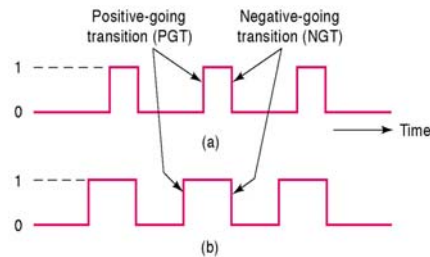
- Hệ thống số có thể hoạt động ở trạng thái:
  - Bất đồng bộ (Asynchronously): Trạng thái ngõ ra sẽ thay đổi khi có bất kỳ sự thay đổi nào ở ngõ vào.
  - Đồng bộ (Synchronously): Ngõ ra chỉ thay đổi tại những thời điểm có cạnh xung clock (đồng bộ với cạnh xung clock)

14

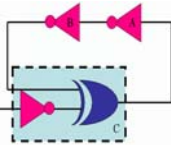


## Xung clock

- Với hệ thống đồng bộ, ngõ ra thay đổi trạng thái tại những thời điểm có cạnh xung clock.
  - Cạnh xung dương Positive-going transitions (PGT)
  - Cạnh xung âm: Negative-going transitions (NGT)

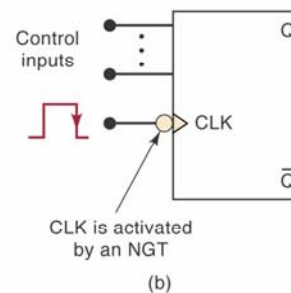
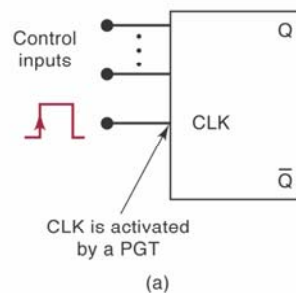


15

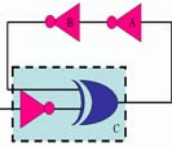


## Flip-Flops và xung clock

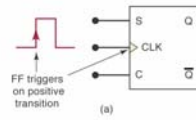
- Trong các FF có ngõ vào xung clock (CLK)
  - (a) Xung clock tích cực cạnh dương
  - (b) Xung clock tích cực cạnh âm



16

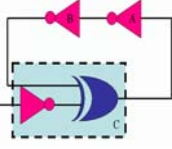
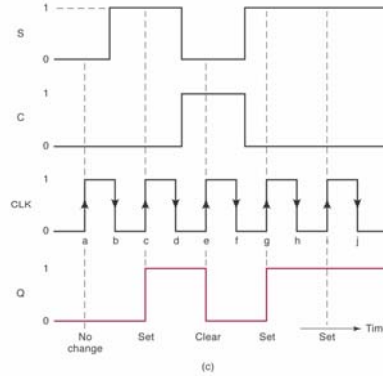


# Dạng sóng của SC-FF

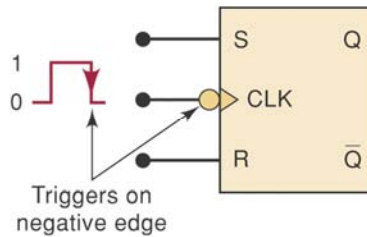


Inputs			Output
S	C	CLK	Q
0	0	↑	$Q_0$ (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambiguous

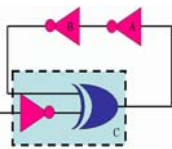
$Q_0$  is output level prior to ↑ of CLK.  
 ↓ of CLK produces no change in Q.



# SC-FF tích cực cạnh âm



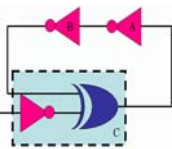
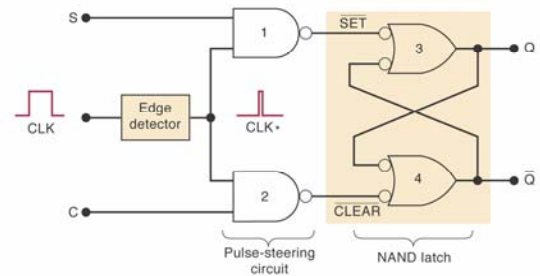
Inputs			Output
S	R	CLK	Q
0	0	↓	$Q_0$ (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	Ambiguous



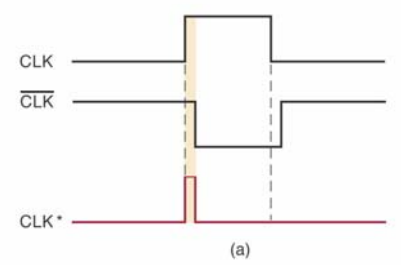
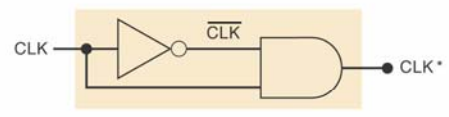
# Cấu trúc bên trong SC-FF



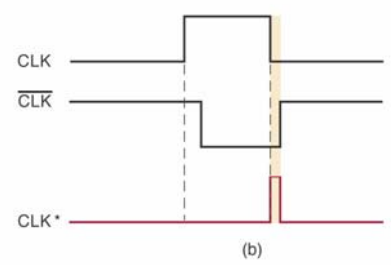
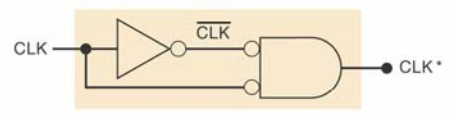
- Bao gồm:
  - Mạch phát hiện cạnh xung
  - Mạch thiết lập trạng thái
  - Mạch chốt cổng NAND



# Mạch phát hiện cạnh xung

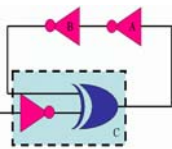


Phát hiện cạnh dương



Phát hiện cạnh âm

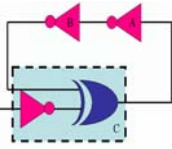




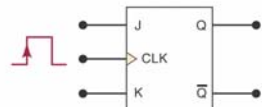
# JK-FF



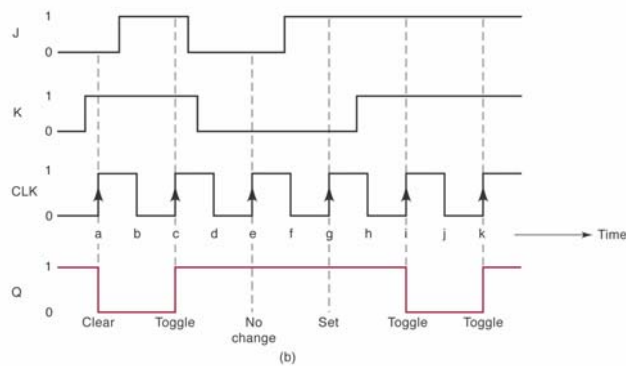
- Hoạt động giống SC-FF. J là ngõ set, K là ngõ clear
- Khi cả J và K đều ở mức cao, ngõ ra sẽ đảo trạng thái so với trạng thái trước đó.
- Có thể tích cực cạnh dương hay cạnh âm xung clock.



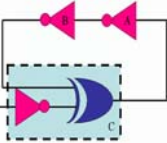
# JK-FF




J	K	CLK	Q
0	0	↑	Q <sub>0</sub> (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	Q <sub>0</sub> (toggles)



## JK-FF tích cực cạnh âm





J

Q

CLK

K

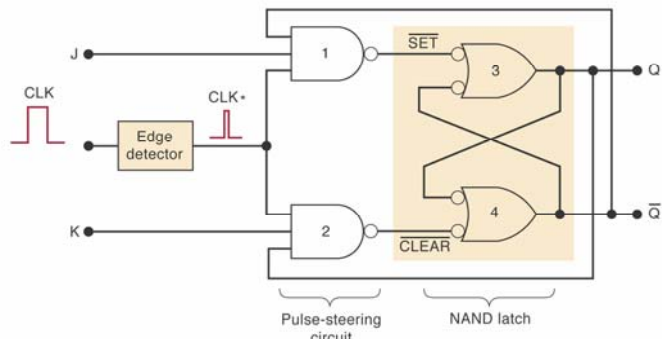
$\bar{Q}$

J	K	CLK	Q
0	0	↓	$Q_0$ (no change)
1	0	↓	1
0	1	↓	0
1	1	↓	$\bar{Q}_0$ (toggles)

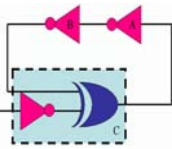
23

## Cấu trúc bên trong của JK-FF

- Khác nhau duy nhất giữa JK và SC-FF là JK có phần hồi tiếp tín hiệu.



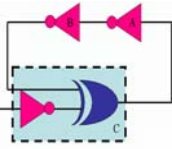
24



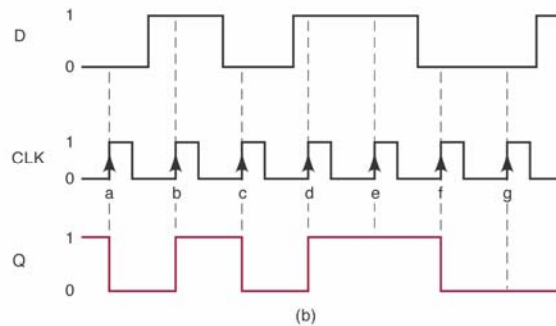
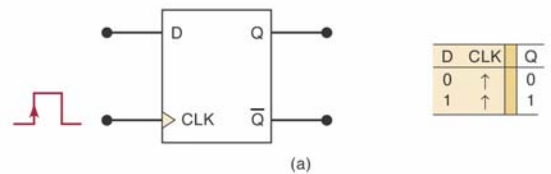
## D Flip-Flop

- Chỉ có một ngõ vào D, tương ứng với ngõ vào data.
- Ngõ ra Q sẽ có cùng giá trị với ngõ vào D khi có tác động của cạnh xung clock.
- Trong những thời điểm khác, D-FF sẽ lưu giá trị trước đó của nó.
- Được sử dụng trong ứng dụng truyền dữ liệu song song

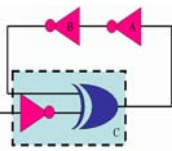
25



## D Flip-Flop

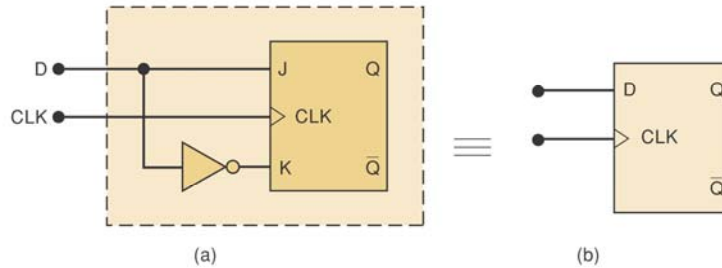


26

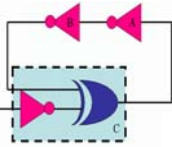


## D-FF và JK-KK

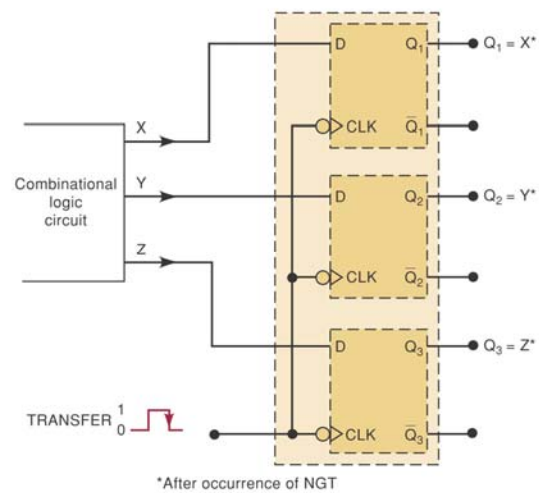
- Có thể tạo ra D-FF từ JK-FF



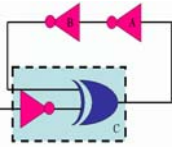
27



## Truyền dữ liệu song song

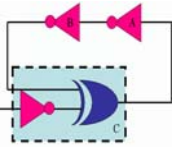


28

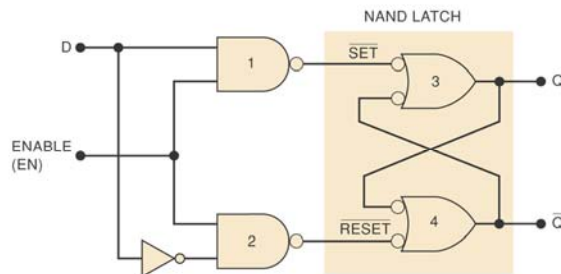


# Mạch chốt D

- Không có mạch phát hiện cạnh xung
- Ngõ vào xung clock được thay bằng ngõ vào enable
- Ngõ ra được xác định theo ngõ vào chỉ khi enable ở mức cao



# Mạch chốt D

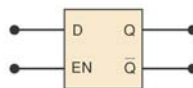


Inputs		Output
EN	D	Q
0	X	Q <sub>0</sub> (no change)
1	0	0
1	1	1

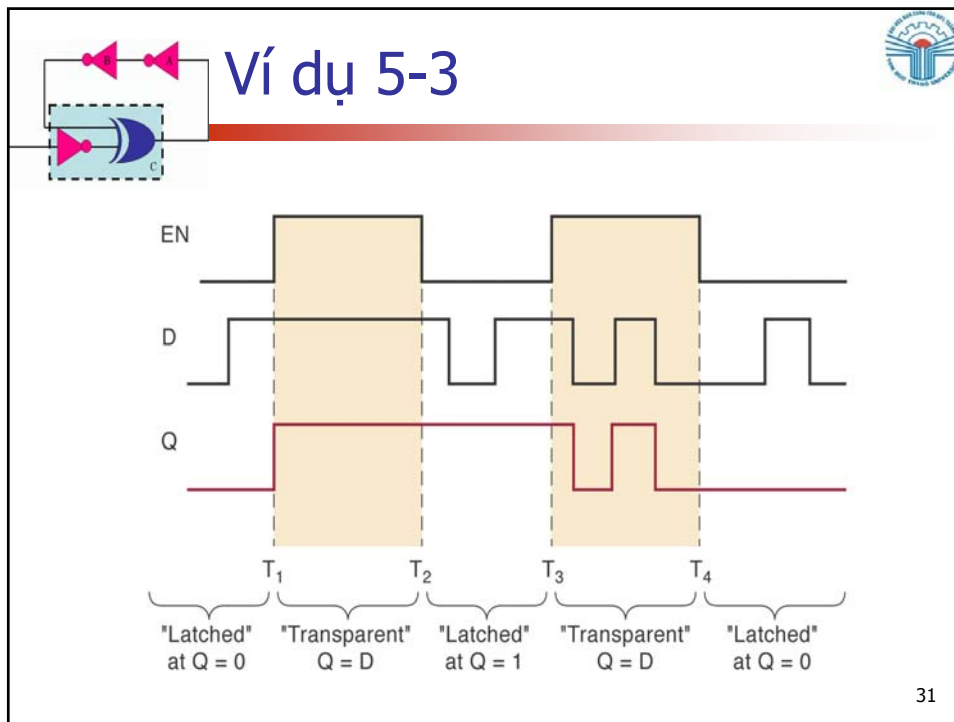
\*X indicates 'don't care.'  
Q<sub>0</sub> is state Q just prior to EN going LOW.

(a)

(b)



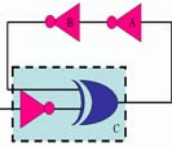
(c)



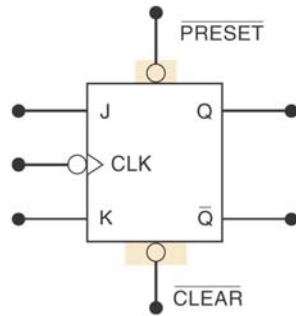
**Ngõ vào không đồng bộ**

- S, C, J, K và D được gọi là những ngõ vào đồng bộ bởi vì ảnh hưởng của chúng đồng bộ với xung clock.
- Ngõ vào không đồng bộ hoạt động độc lập với những ngõ vào đồng bộ, chúng có thể set (1) hoặc clear (0) Flip-Flop vào bất kỳ thời điểm nào.

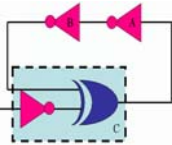
32



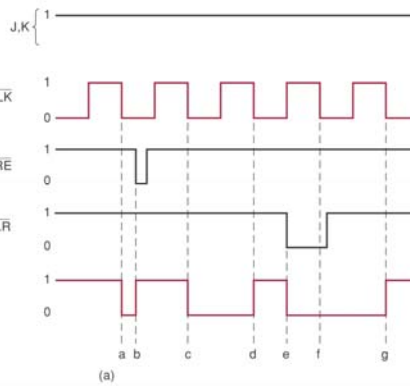
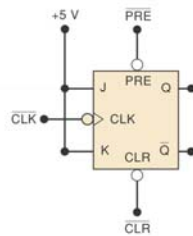
# JK-FF với ngõ vào không đồng bộ



J	K	Clk	PRE	CLR	Q
0	0	↓	1	1	Q (no change)
0	1	↓	1	1	0 (Synch reset)
1	0	↓	1	1	1 (Synch set)
1	1	↓	1	1	$\bar{Q}$ (Synch toggle)
x	x	x	1	1	Q (no change)
x	x	x	1	0	0 (asynch clear)
x	x	x	0	1	1 (asynch preset)
x	x	x	0	0	(Invalid)



# Ví dụ 5-4

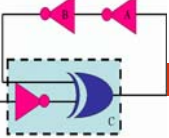


Point	Operation
a	Synchronous toggle on NGT of CLK
b	Asynchronous set on PRE = 0
c	Synchronous toggle
d	Synchronous toggle
e	Asynchronous clear on CLR = 0
f	CLR overrides the NGT of CLK
g	Synchronous toggle

(b)



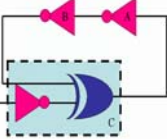
# Ứng dụng của Flip-Flop



35



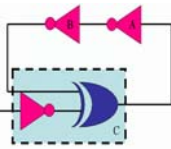
# Ứng dụng của FF



- Một số ứng dụng của flip-flop
  - Bộ đếm
  - Lưu dữ liệu nhị phân
  - Truyền dữ liệu nhị phân giữa các thiết bị

36

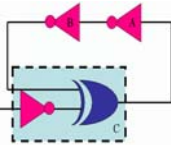




## Đồng bộ tín hiệu

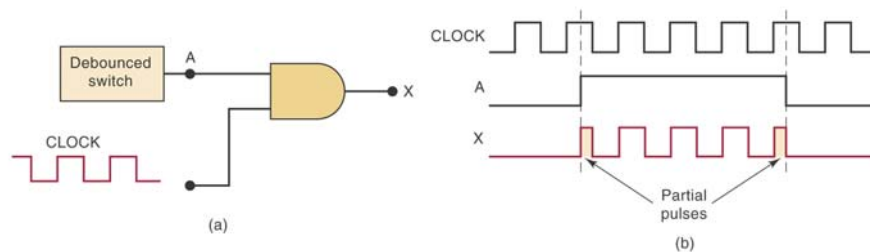
- Đa số hệ thống hoạt động ở chế độ đồng bộ.
- Các tín hiệu tự nhiên là những tín hiệu không đồng bộ.
- Chúng ta phải đồng bộ những tín hiệu này với xung clock.

37



## Đồng bộ tín hiệu

- Tín hiệu không đồng bộ A có thể tạo ra những mẫu xung không đúng.



38

## Đồng bộ tín hiệu

(a)

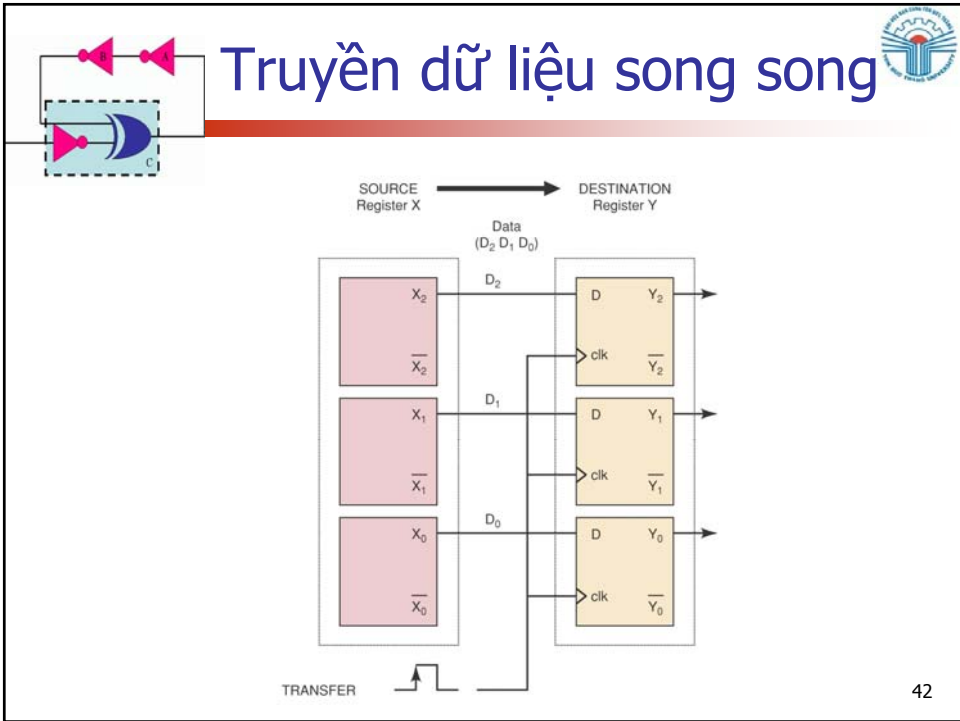
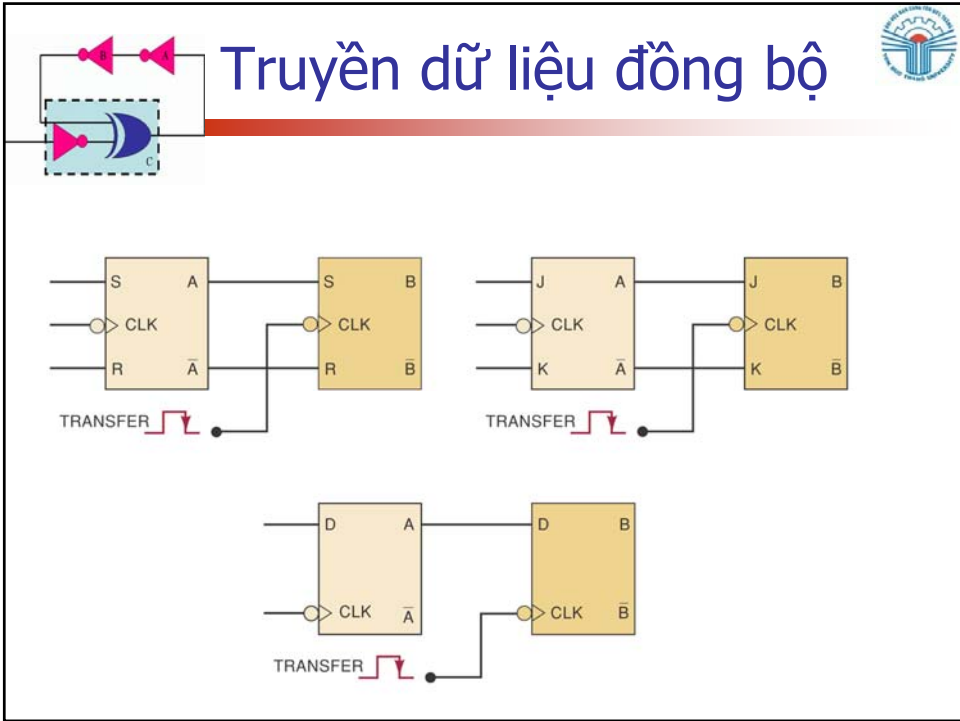
(b)

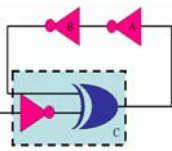
39

## Lưu và truyền dữ liệu

- FF thường được sử dụng để lưu và truyền dữ liệu dạng nhị phân.
- Nhóm FF sử dụng để lưu data là thanh ghi
- Dữ liệu được truyền khi data chuyển đổi giữa những FF hoặc thanh ghi.
- Trong trường hợp truyền đồng bộ, cần phải có xung đồng bộ

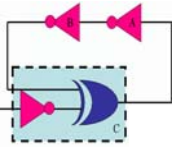
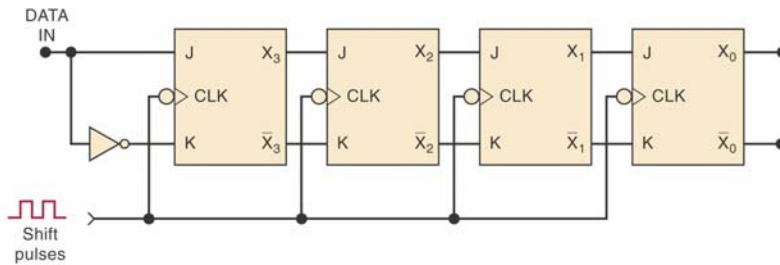
40



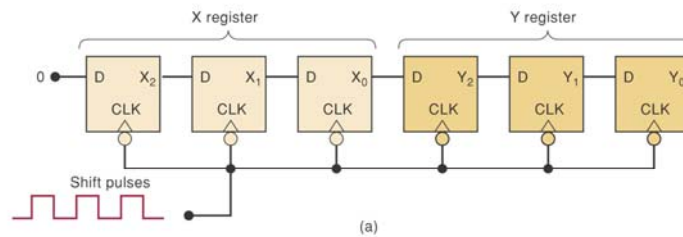


# Thanh ghi dịch

- Trong trường hợp này dữ liệu sẽ được truyền nối tiếp.



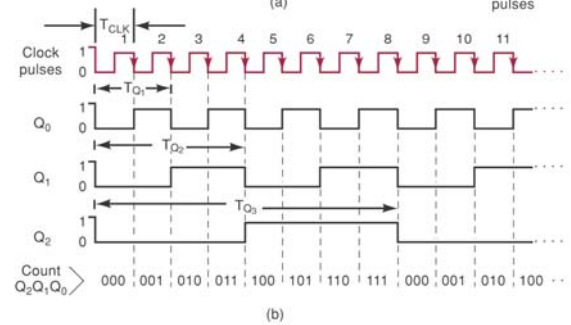
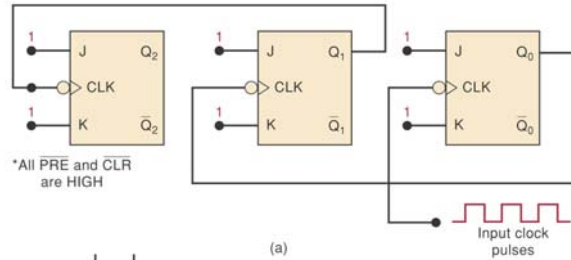
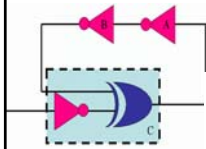
# Truyền data giữa hai thanh ghi



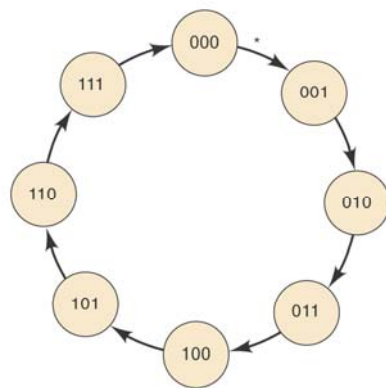
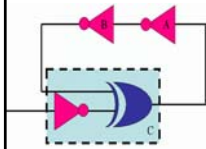
X <sub>2</sub>	X <sub>1</sub>	X <sub>0</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	
1	0	1	0	0	0	← Before pulses applied
0	1	0	1	0	0	← After first pulse
0	0	1	0	1	0	← After second pulse
0	0	0	1	0	1	← After third pulse



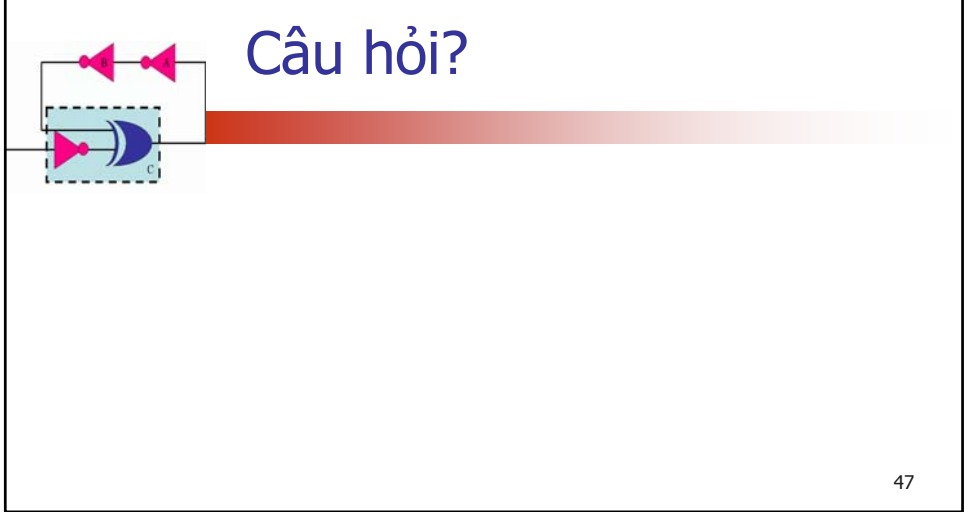
# Chia tần số



# Bộ đếm

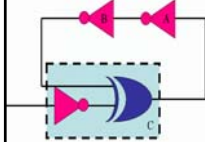


\* Note: each arrow represents the occurrence of a clock pulse





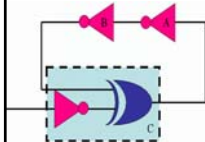
# Chương 6 Mạch số học



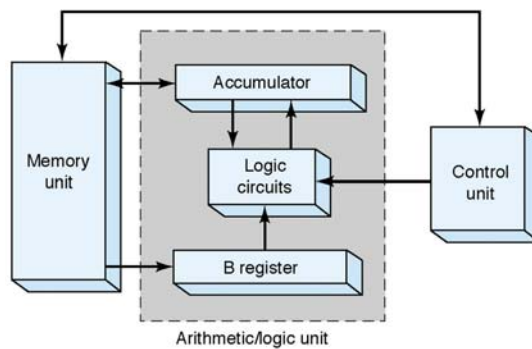
Th.S Đặng Ngọc Khoa  
Khoa Điện - Điện Tử

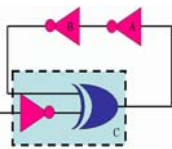


# Mạch số học



- ALU (arithmetic/logic unit) sẽ lấy data từ trong bộ nhớ để thực thi những lệnh theo control unit



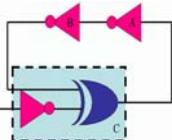


## Mạch số học



- Ví dụ quá trình một lệnh được thực thi:
  - Đơn vị điều khiển ra lệnh cộng một số được chỉ định trong bộ nhớ với số có trong thanh ghi *accumulator*.
  - Số cộng được truyền từ bộ nhớ đến thanh ghi B.
  - Dữ liệu trong thanh ghi B và thanh ghi *accumulator* sẽ được cộng lại với nhau.
  - Kết quả sẽ được lưu vào trong thanh ghi *accumulator*
  - Giá trị trong thanh ghi *accumulator* sẽ được giữ cho đến khi có lệnh mới.

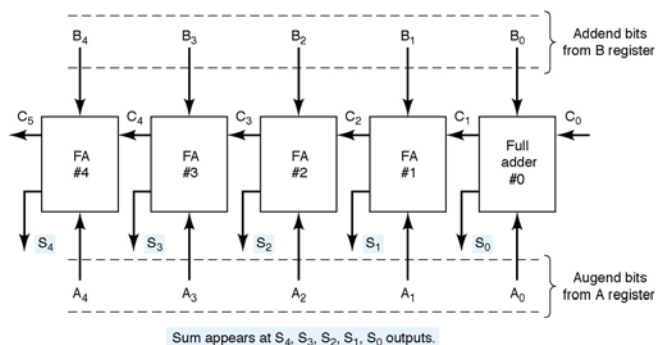
3



## Bộ cộng nhị phân song song

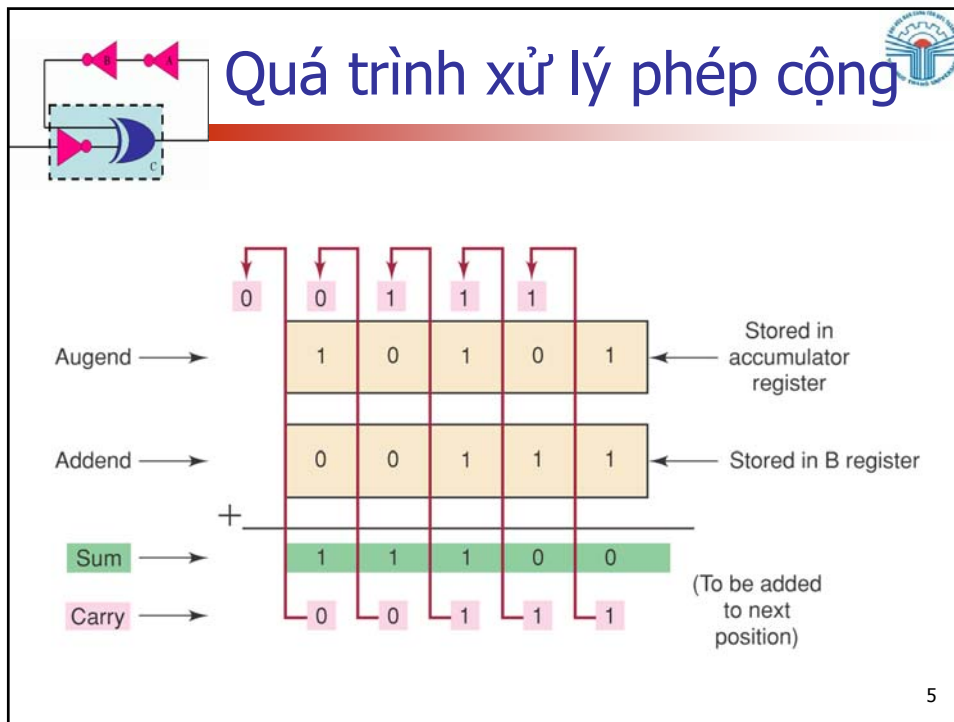


- A, B là giá trị cần cộng. C là giá trị nhớ. S là kết quả của phép cộng



4

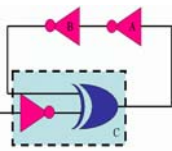




## Ví dụ 6-1

- Hãy thiết kế một bộ cộng đầy đủ:
  - Bộ cộng có 3 ngõ vào
    - 2 ngõ vào thể hiện số cần cộng
    - 1 ngõ vào chứa số nhớ ngõ vào
  - Có 2 ngõ ra
    - 1 ngõ ra là kết quả của phép cộng
    - 1 ngõ ra là số nhớ ngõ ra

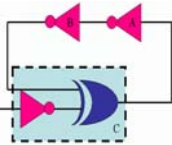
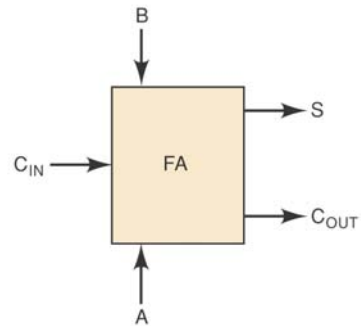
6



## Ví dụ 6-1 – Giải

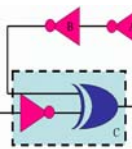
### ■ Bảng chân trị

Augend bit input	Addend bit input	Carry bit input	Sum bit output	Carry bit output
A	B	C <sub>IN</sub>	S	C <sub>OUT</sub>
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



## Ví dụ 6-1 – Giải

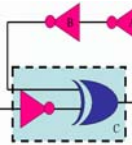
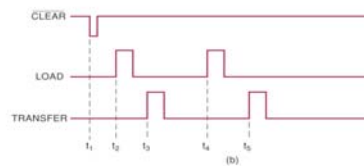
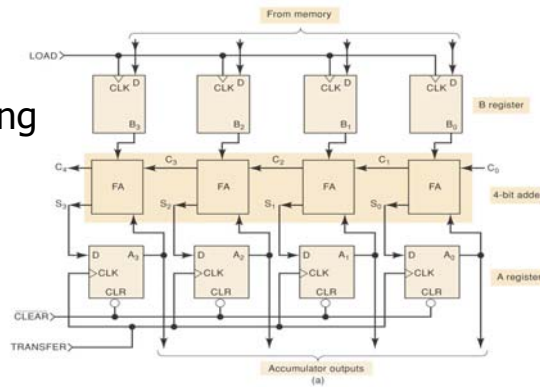
### ■ Sơ đồ mạch kết quả



## Ví dụ 6-2



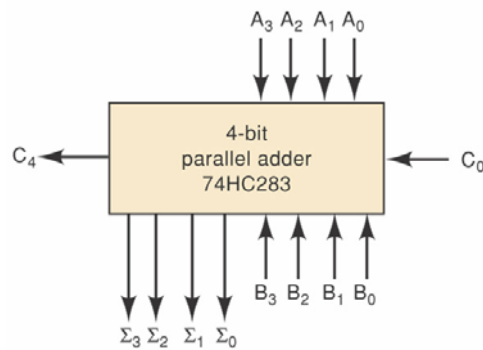
- Giải thích hoạt động của mạch sau

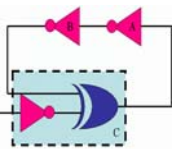


## IC bộ cộng



- IC 74HC283 là IC bộ cộng song song 4 bit
- A và B là hai số 4 bit
- $C_0$  là số nhớ ngõ vào,  $C_4$  là số nhớ ngõ ra

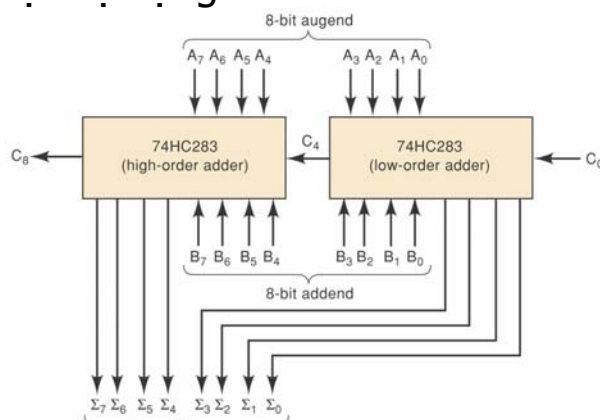




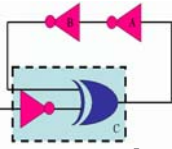
## IC bộ cộng



- Ta có thể nối tiếp hai bộ cộng 4 bit để tạo ra một bộ cộng 8 bit



11



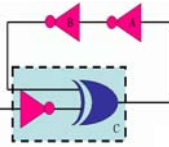
## Bộ cộng BCD



- Có thêm phần mạch để xử lý trường hợp tổng lớn hơn 9

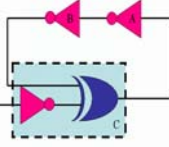
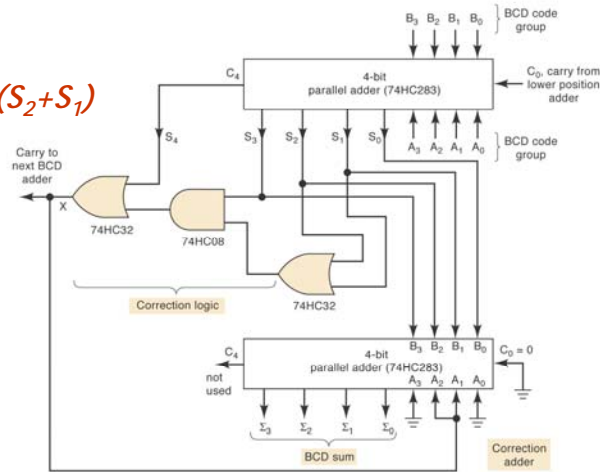
$S_4$	$S_3$	$S_2$	$S_1$	$S_0$	
0	1	0	1	0	(10)
0	1	0	1	1	(11)
0	1	1	0	0	(12)
0	1	1	0	1	(13)
0	1	1	1	0	(14)
0	1	1	1	1	(15)
1	0	0	0	0	(16)
1	0	0	0	1	(17)
1	0	0	1	0	(18)

12

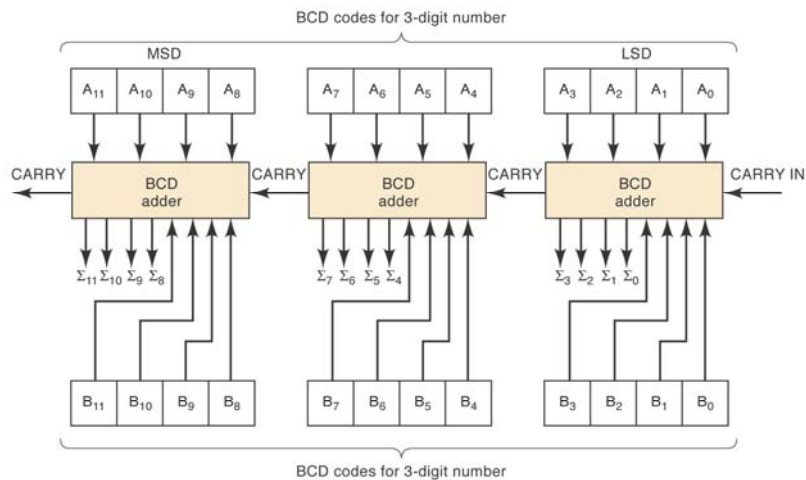


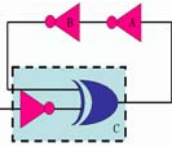
# Bộ cộng BCD

$$X = S_4 + S_3(S_2 + S_1)$$



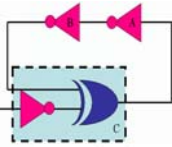
# Bộ cộng BCD nối tiếp



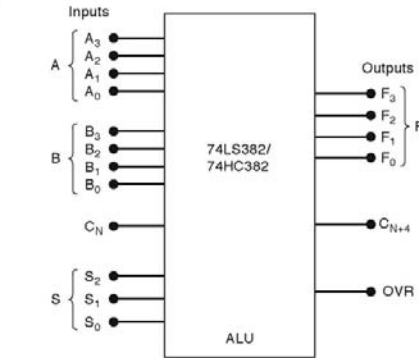


# IC ALU

- ALU có thể thực thi nhiều toán tử và hàm logic khác nhau, các toán tử và hàm này được xác định bởi một mã ngõ vào.
- 74LS382 (TTL) và HC382 (CMOS) là thiết bị ALU tiêu biểu có thể thực hiện 8 hàm khác nhau.



# IC ALU



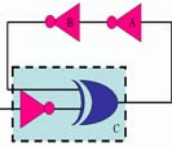
Function Table

S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	Operation	Comments
0	0	0	CLEAR	F <sub>3</sub> F <sub>2</sub> F <sub>1</sub> F <sub>0</sub> = 0000
0	0	1	B minus A	Needs C <sub>N</sub> = 1
0	1	0	A minus B	
0	1	1	A plus B	Needs C <sub>N</sub> = 0
1	0	0	A ⊕ B	Exclusive-OR
1	0	1	A + B	OR
1	1	0	AB	AND
1	1	1	PRESET	F <sub>3</sub> F <sub>2</sub> F <sub>1</sub> F <sub>0</sub> = 1111

Notes: S inputs select operation.  
OVR = 1 for signed-number overflow.

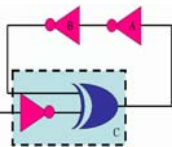
(b)

A = 4-bit input number      F = 4-bit output number  
 B = 4-bit input number      C<sub>N+4</sub> = carry out of MSB position  
 C<sub>N</sub> = carry into LSB position      OVR = overflow indicator  
 S = 3-bit operation select inputs



## Ví dụ 6-3

- Hãy sử dụng 2 IC 74LS382 để tạo thành bộ cộng 8 bit

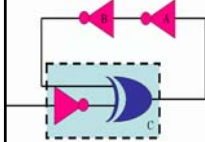


## Câu hỏi?



## Chương 7

### Bộ đếm và thanh ghi

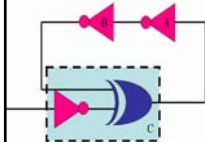


Th.S Đặng Ngọc Khoa  
Khoa Điện - Điện Tử

1



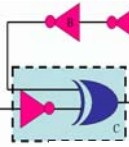
### Bộ đếm không đồng bộ



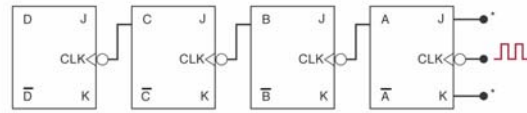
- Xét bộ đếm 4 bit ở hình 7-1
  - Xung clock chỉ được đưa đến FF A, ngõ vào J, K của tất cả các FF đều ở mức logic 1.
  - Ngõ ra của FF sau được nối đến ngõ vào CLK của FF trước nó.
  - Ngõ ra D, C, B, A là một số nhị phân 4 bit với D là bit có trọng số cao nhất.
- Đây là bộ đếm không đồng bộ vì trạng thái của các FF không thay đổi cùng với xung clock.

2

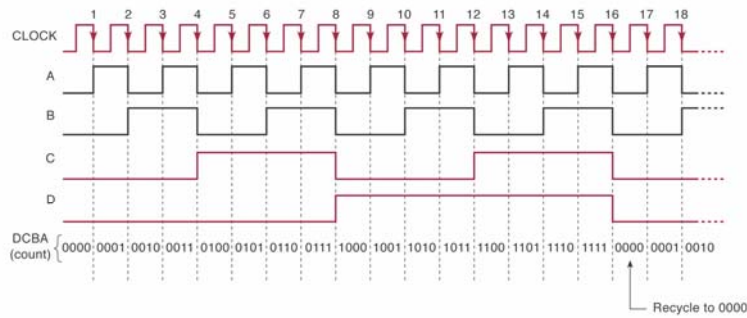




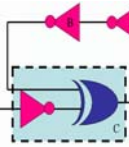
## Hình 7-1 Bộ đếm 4 bit



\*All J and K inputs assumed to be 1.



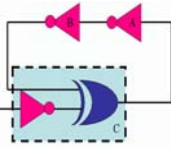
3



## Quy ước về trọng số

- Trong phần lớn các mạch, dòng tín hiệu thường chạy từ trái sang phải.
- Trong chương này, nhiều mạch điện có dòng tín hiệu chạy từ phải sang trái.
- Ví dụ, trong hình 7-1:
  - Flip-Flop A: LSB
  - Flip-Flop D: MSB

4

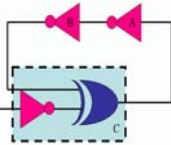


## SỐ MOD



- Số MOD là số trạng thái trong một chu kỳ của một bộ đếm.
- Bộ đếm trong hình 7-1 có 16 trạng thái khác nhau, do vậy nó là bộ đếm MOD-16
- Số MOD của một bộ đếm được thay đổi cùng với số Flip-Flop.
  - Số MOD  $\leq 2^N$

5

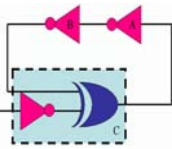


## SỐ MOD



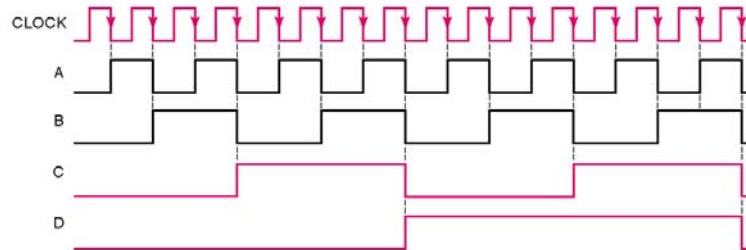
- Ví dụ
  - Một bộ đếm được sử dụng để đếm sản phẩm chạy qua một băng tải. Mỗi sản phẩm đi qua băng chuyền, bộ cảm biến sẽ tạo ra một xung. Bộ đếm có khả năng đếm được 1000 sản phẩm. Hỏi ít nhất phải có bao nhiêu Flip-Flop trong bộ đếm?
  - Trả lời:  $1000 \leq 2^{10} = 1024$ . Phải có 10 FF

6

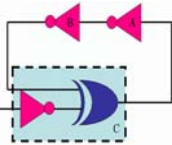


## Chia tần số

- Trong một bộ đếm, tín hiệu ngõ ra của FF cuối cùng (MSB) có tần số bằng tần số ngõ vào chia cho số MOD. Một bộ đếm MOD-N là bộ chia N.

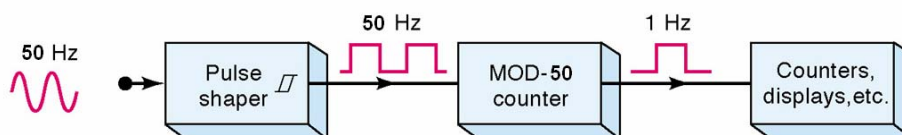


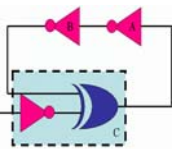
7



## Ví dụ 7-1

- Ví dụ mạch tạo ra dao động xung vuông có tần số 1Hz.
  - Tạo ra tín hiệu xung vuông 50Hz từ lưới điện.
  - Cho đi qua bộ đếm MOD-50 để chia tần số 50 lần.
  - Có được tín hiệu xung vuông tần số 1Hz

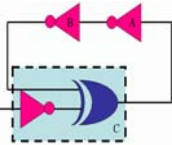




## Trễ trong bộ đếm không đồng bộ



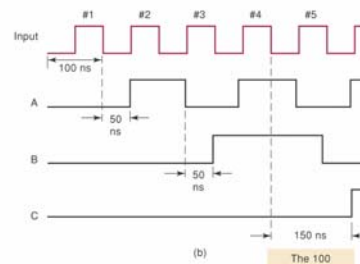
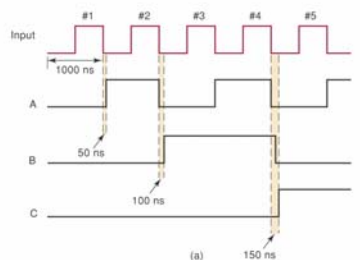
- Cấu trúc của bộ đếm không đồng bộ khá đơn giản nhưng vấn đề trễ khi truyền tín hiệu qua mỗi FF sẽ làm hạn chế tần số của bộ đếm.
- Với bộ đếm không đồng bộ ta phải có
  - $T_{\text{clock}} \geq N \times t_{pd}$
  - $F_{\text{max}} = 1 / (N \times t_{pd})$

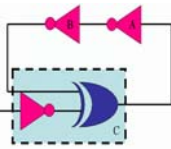


## Trễ trong bộ đếm không đồng bộ



- Bộ đếm 3 bit với những tần số xung clock khác nhau

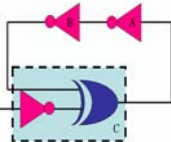




## Câu hỏi?

- Trong bộ đếm bất đồng bộ, tất cả các FF thay đổi trạng thái cùng một lúc?
- Giả sử bộ đếm trong hình 7-1 đang ở trạng thái 0101. Sau 27 xung clock, trạng thái của bộ đếm sẽ là bao nhiêu?
- Số MOD của bộ đếm có 5 Flip-Flop?

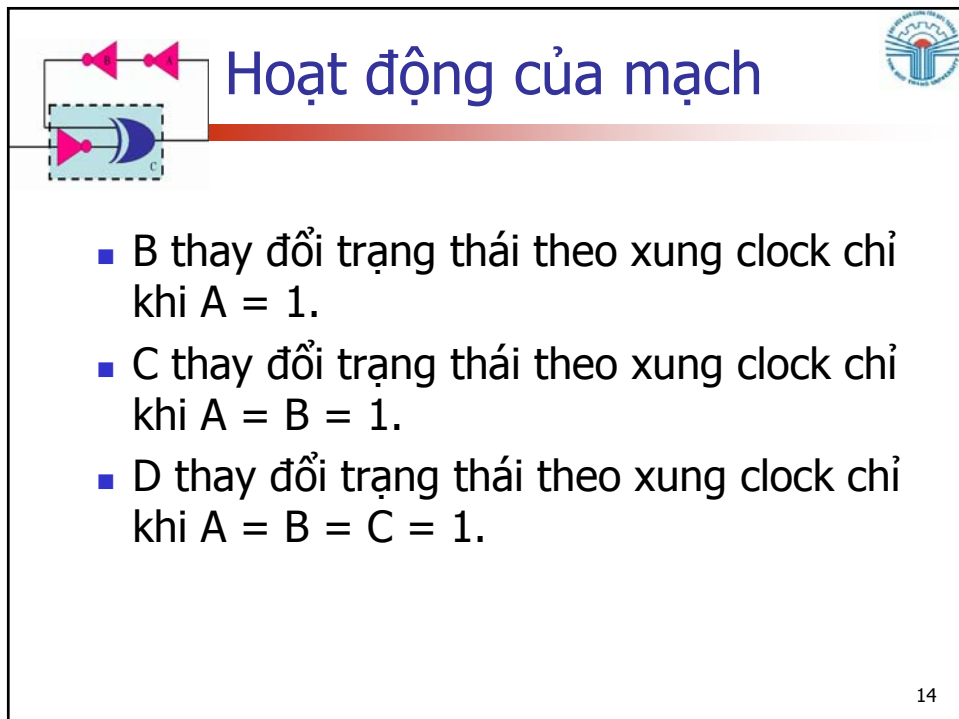
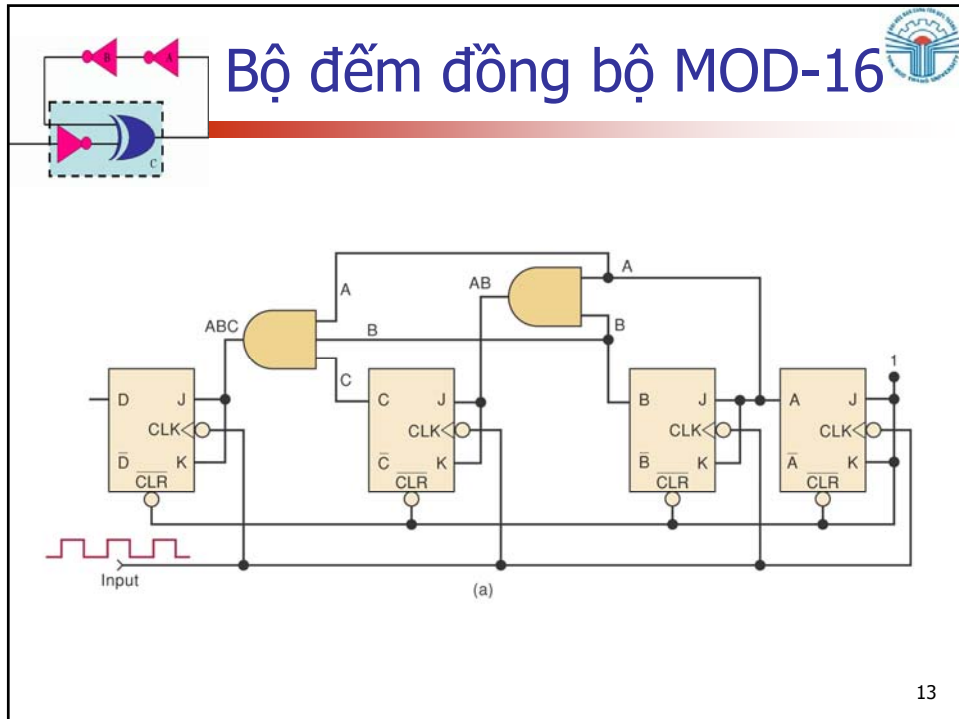
11

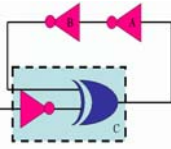


## Bộ đếm đồng bộ

- Trạng thái của tất cả các FF sẽ được thay đổi cùng một lúc với xung clock.
- Hình sau mô tả hoạt động của một bộ đếm đồng bộ
  - Mỗi FF có ngõ vào J, K được kết nối sao cho chúng ở trạng thái cao chỉ khi ngõ ra của tất cả các FF sau nó đều ở trạng thái cao.
  - Bộ đếm đồng bộ có thể hoạt động với tần số cao hơn bộ đếm không đồng bộ.

12





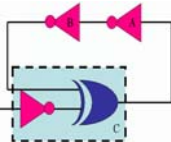
## Bảng chân trị



Count	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1
0	0	0	0	0
.	.	.	.	.
.	.	.	.	.
.	.	etc.	.	.

(b)

15

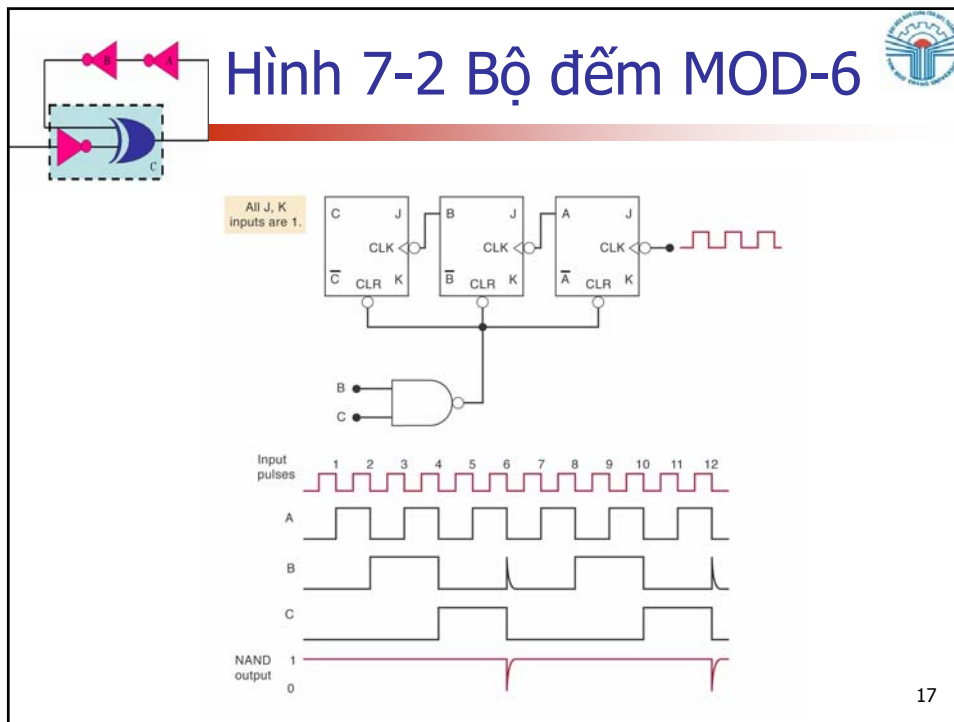


## Bộ đếm có số MOD <math>2^N</math>



- Sử dụng ngõ vào không đồng bộ (clear, set) để buộc bộ đếm bỏ đi một số trạng thái.
- Trong hình 7-2, ngõ ra cổng NAND được nối đến ngõ vào không đồng bộ CLEAR của mỗi Flip-Flop.
- Khi  $A=0, B=C=1$ , ( $CBA = 110_2 = 6_{10}$ ) ngõ ra cổng NAND sẽ tích cực và các FF sẽ bị CLEAR về trạng thái 0.

16

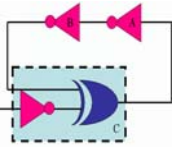


## Trạng thái tạm

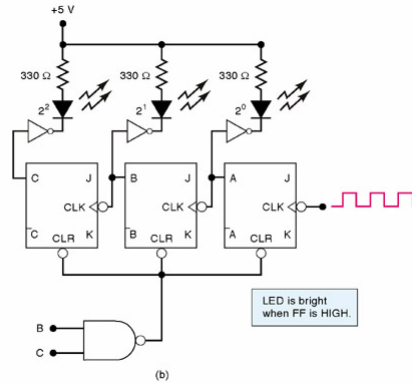
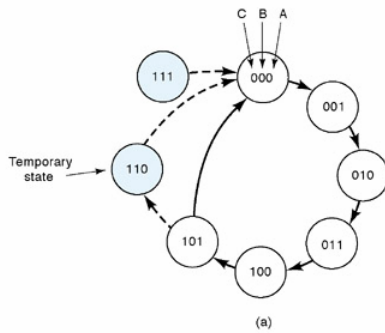
- Lưu ý rằng trong hình 7-2, 110 là một trạng thái tạm thời. Mạch chỉ tồn tại ở trạng thái này trong thời gian rất ngắn sau đó sẽ chuyển sang trạng thái 000.
- 000 → 001 → 010 → 011 → 100 → 101 → 000
- Ngõ ra của FF C có tần số bằng 1/6 tần số ngõ vào.

18

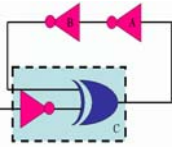




## Sơ đồ trạng thái



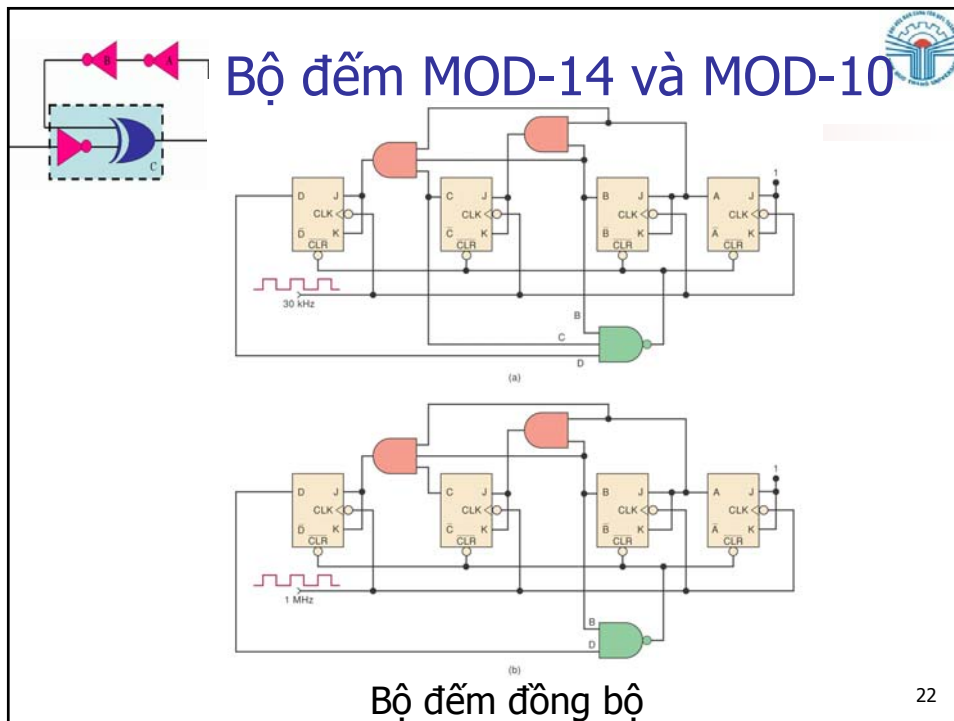
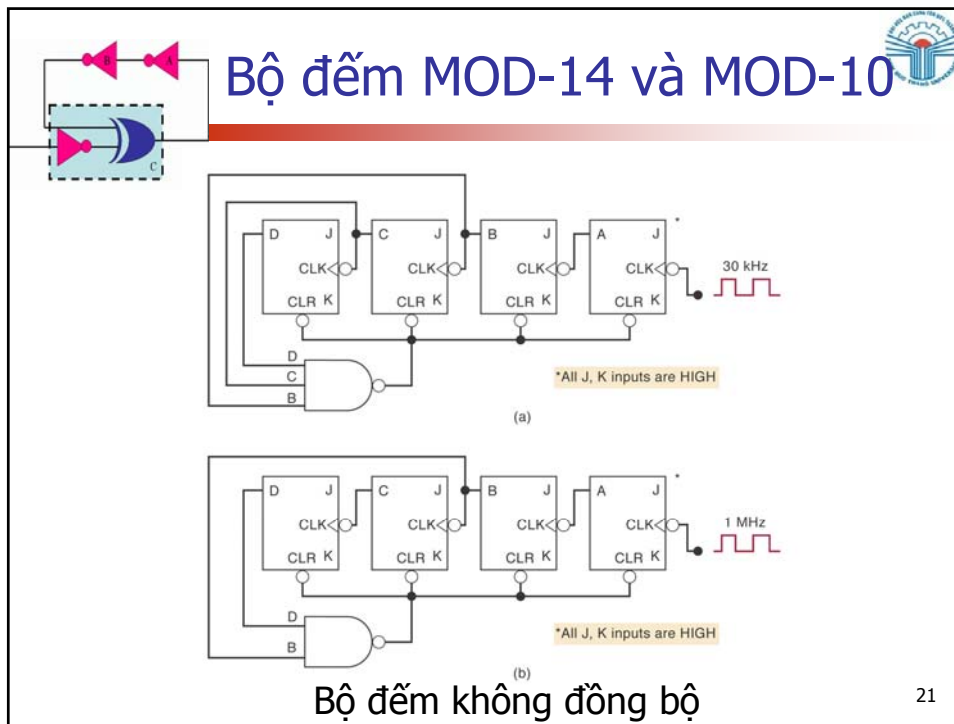
19

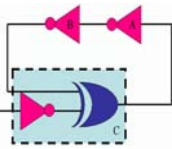


## Thiết kế bộ đếm MOD-X

- **Bước 1:** Tìm số FF nhỏ nhất sao cho  $2^N \geq X$ . Kết nối các FF lại với nhau. Nếu  $2^N = X$  thì không làm bước 2 và 3.
- **Bước 2:** Nối một cổng NAND đến ngõ vào CLEAR của tất cả các FF.
- **Bước 3:** Xác định FF sẽ ở mức cao ứng với trạng thái bộ đếm = X. Nối ngõ ra của các FF đến ngõ vào của cổng NAND.

20

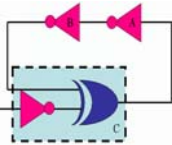




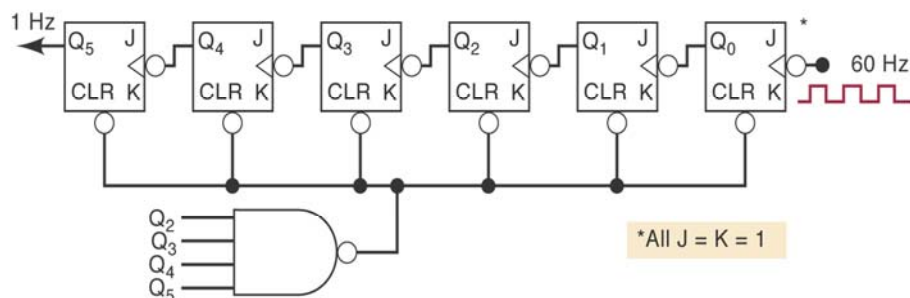
## Bộ đếm thập phân

- Bộ đếm thập phân
  - Là bất kỳ bộ đếm nào có mười trạng thái phân biệt.
- Bộ đếm BCD
  - Là một bộ đếm thập phân mà các trạng thái trong bộ đếm tương ứng từ 0000 (zero) đến 1001 (9)

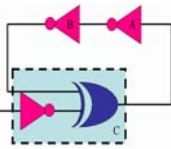
23



## Bộ đếm MOD-60 không đồng bộ



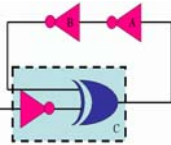
24



## Ví dụ 7-2

- Xác định mạch bộ đếm đồng bộ MOD-60

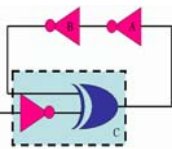
25



## Câu hỏi?

- Trong bộ đếm MOD-13, ngõ ra của FF nào được nối đến ngõ vào cổng NAND của mạch clear?
- Tất cả các bộ đếm BCD là bộ đếm thập phân?
- Cho một bộ đếm thập phân, tần số ngõ vào là 50KHz. Tần số ngõ ra là bao nhiêu?

26

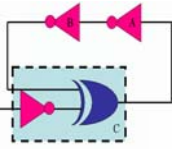


## Bộ đếm xuống không đồng bộ

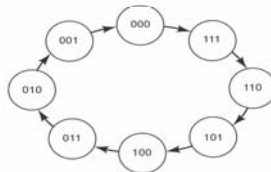
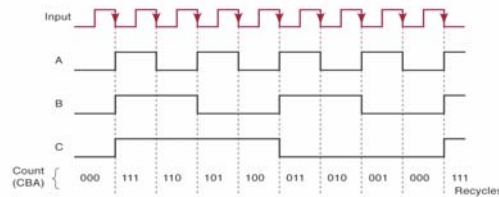
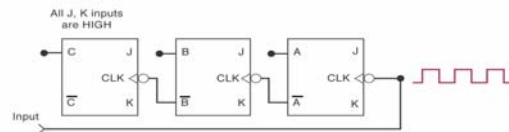


- 111 → 110 → 101 → 100 → 011 → 010 → 001 → 000
- Bộ đếm lên có thể chuyển thành bộ đếm xuống bằng cách sử dụng những ngõ ra đảo để lái các ngõ vào xung clock.

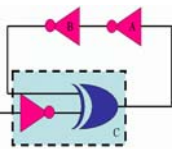
27



## Bộ đếm xuống MOD-8



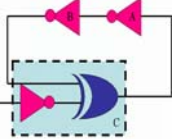
28



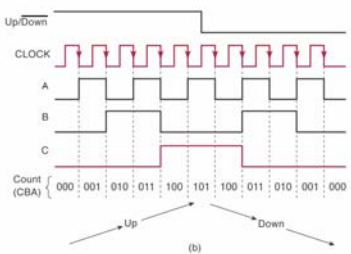
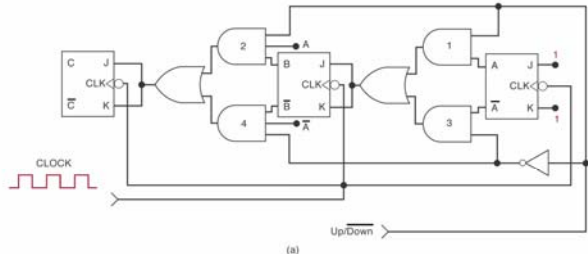
## Bộ đếm xuống đồng bộ

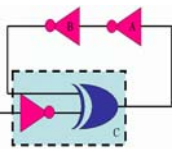


- Bộ đếm xuống đồng bộ có cấu tạo hoàn toàn tương tự như bộ đếm lên đồng bộ.
- Chỉ khác là sử dụng các ngõ ra đảo để điều khiển.



## Bộ đếm lên/xuống đồng bộ





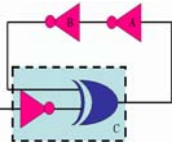
## IC bộ đếm không đồng bộ



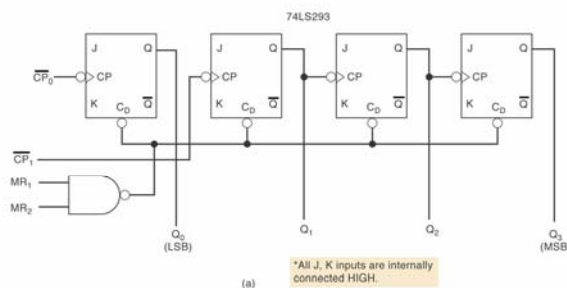
### ■ IC 74LS293 họ TTL

- Có 4 J-K Flip-Flop,  $Q_3Q_2Q_1Q_0$
- Mỗi FF có một ngõ vào CP (clock pulse) tương tự như ngõ vào CLK. Ngõ vào clock của  $Q_1$  và  $Q_0$  được nối đến chân 11 và chân 10.
- Ngõ vào clear của mỗi FF được nối đến ngõ ra của một cổng NAND hai ngõ vào  $MR_1$  và  $MR_2$ .
- $Q_3Q_2Q_1$  được nối với nhau hình thành nên một bộ đếm 3 bit.
- $Q_0$  không được nối đến các phần khác.

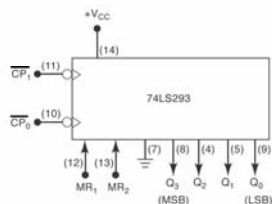
31



## IC bộ đếm không đồng bộ

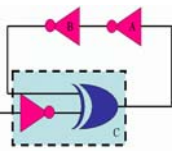


(a)



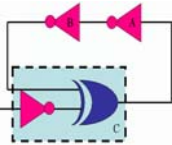
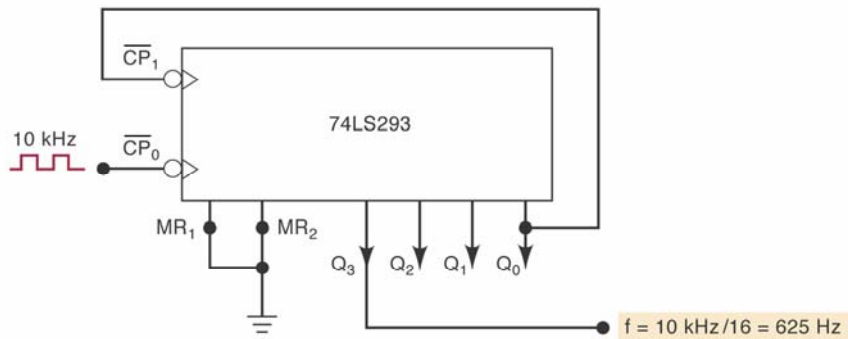
(b)

32



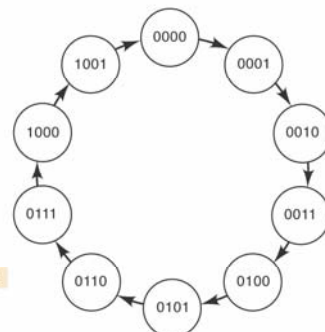
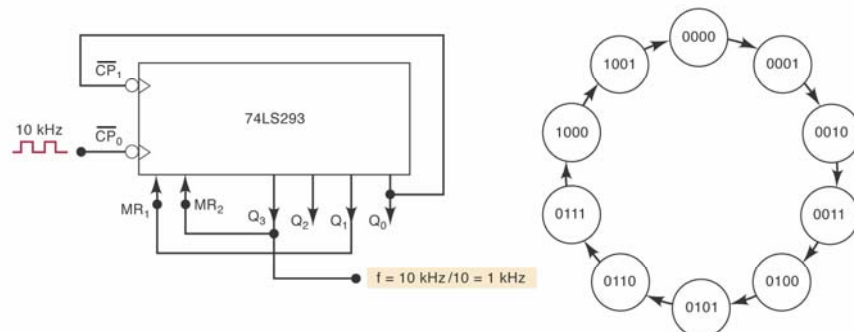
### Ví dụ 7-3

- Sử dụng IC 74LS293 thiết lập bộ đếm MOD-16

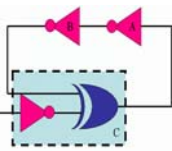


### Ví dụ 7-4

- Sử dụng IC 74LS293 thiết lập bộ đếm MOD-10

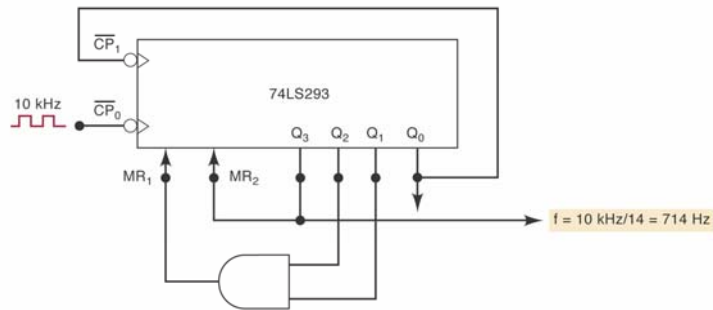




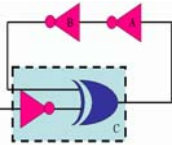


## Ví dụ 7-5

- Sử dụng IC 74LS293 thiết lập bộ đếm MOD-14
- Trường hợp này cần phải sử dụng thêm một cổng AND

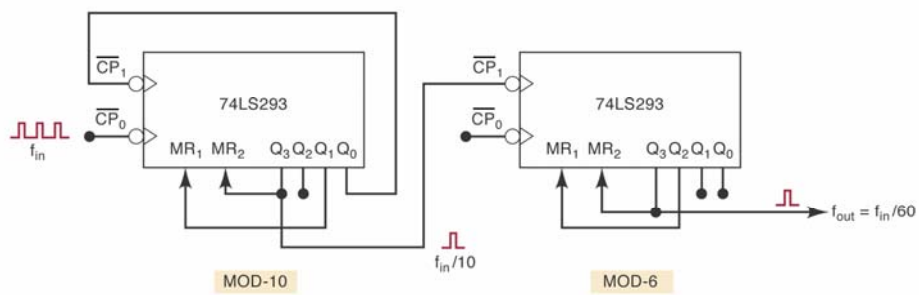


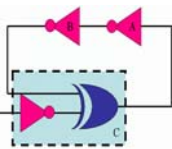
35



## Ví dụ 7-6

- Sử dụng IC 74LS293 thiết lập bộ đếm MOD-60
- Sử dụng 2 IC mắc nối tiếp nhau.

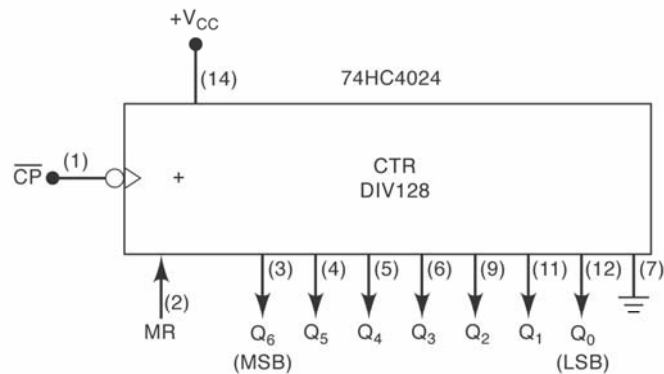




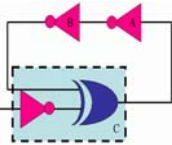
## IC bộ đếm



### ■ Giới thiệu IC bộ đếm 7 bit họ CMOS



37



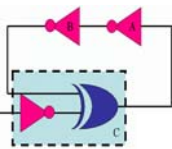
## IC bộ đếm đồng bộ



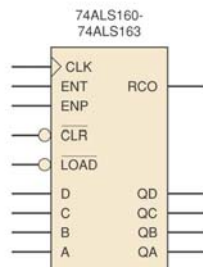
### ■ TTL 74ALS160

- Bộ đếm có 4 FF
- FF thay đổi trạng thái theo cạnh dương của xung clock
- IC có một ngõ vào CLEAR không đồng bộ.
- Bộ đếm có thể preset đến bất kỳ giá trị nào (theo các ngõ vào A, B, C, và D) bằng cách tích cực ngõ vào LOAD.
- Bộ đếm được điều khiển bởi các ngõ vào khác nhau, thể hiện trong bảng Function table.

38



# 74ALS160



Part Number	Modulus
74ALS160	10
74ALS161	16
74ALS162	10
74ALS163	16

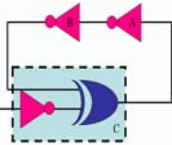
(b)

(a)

74ALS160-74ALS163 Function Table

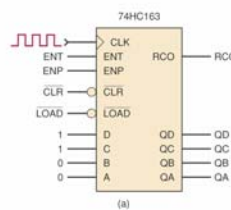
CLR	LOAD	ENP	ENT	CLK	Function	Part Numbers
L	X	X	X	X	Asynch. Clear	74ALS160 & 74ALS161
L	X	X	X	↑	Synchr. Clear	74ALS162 & 74ALS163
H	L	X	X	↑	Synchr. Load	All
H	H	H	H	↑	Count up	All
H	H	L	X	X	No change	All
H	H	X	L	X	No change	All

(c)

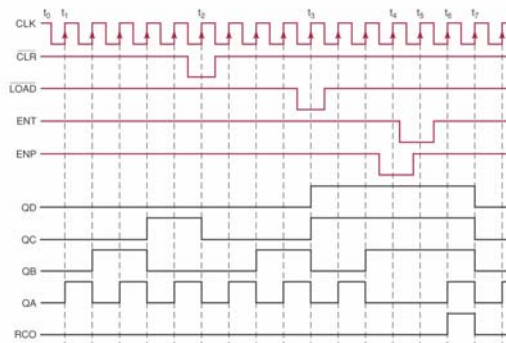


# 74ALS160

## ■ Ví dụ



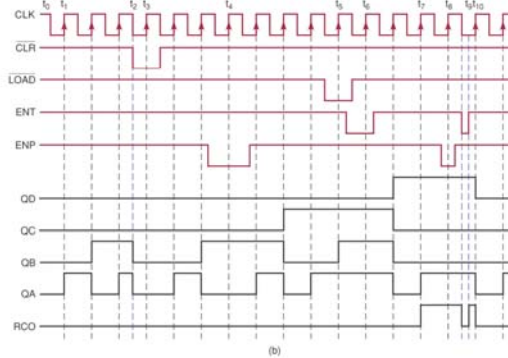
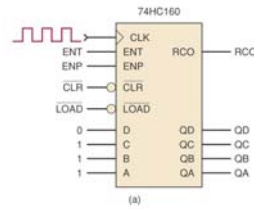
(a)



(b)



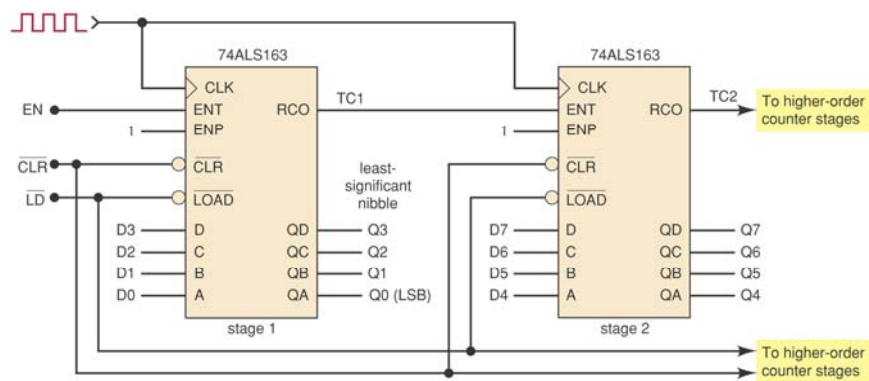
■ Ví dụ

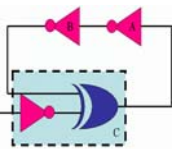


41



■ Có thể kết hợp nhiều IC để tạo ra một bộ đếm lớn hơn

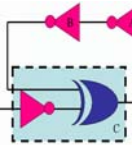




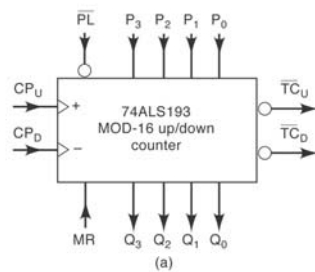
# IC bộ đếm đồng bộ



- 74ALS193/HC193
  - Bộ đếm đồng bộ
  - MOD-16
  - Presettable up/down
  - Preset không đồng bộ
  - Master reset không đồng bộ



# IC 74ALS193



Pin names	Description
CP <sub>U</sub>	Count-up clock input (active rising edge)
CP <sub>D</sub>	Count-down clock input (active rising edge)
MR	Asynchronous master reset input (active HIGH)
$\overline{PL}$	Asynchronous parallel load input (active LOW)
P <sub>0</sub> -P <sub>3</sub>	Parallel data inputs
Q <sub>0</sub> -Q <sub>3</sub>	Flip-flop outputs
$\overline{TC}_D$	Terminal count-down (borrow) output (active LOW)
$\overline{TC}_U$	Terminal count-up (carry) output (active LOW)

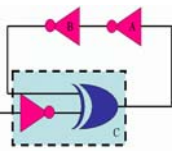
Mode Select

MR	$\overline{PL}$	CP <sub>U</sub>	CP <sub>D</sub>	Mode
H	X	X	X	Asynch. reset
L	L	X	X	Asynch. preset
L	H	H	H	No change
L	H	↑	H	Count up
L	H	H	↑	Count down

H = HIGH; L = LOW  
 X = don't care; ↑ = PGT

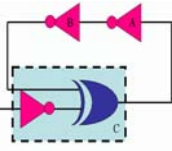
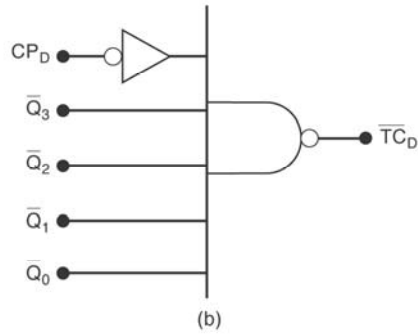
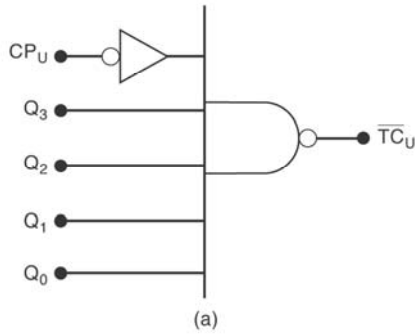
(c)

(b)

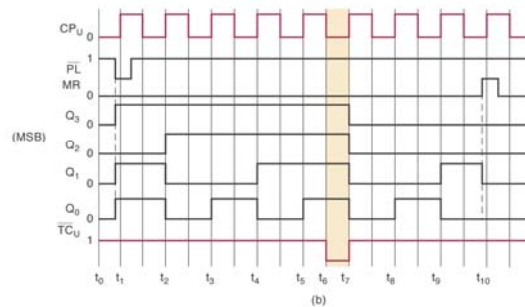
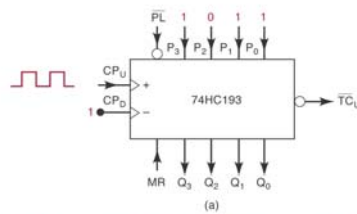


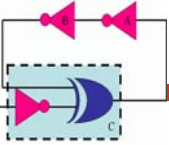
# IC 74ALS193

## ■ Mạch tạo ra TC<sub>U</sub> và TC<sub>D</sub>

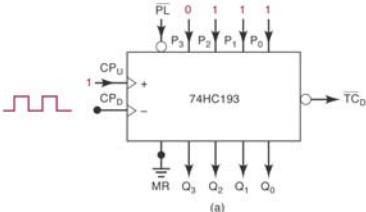


# 74ALS193 - bộ đếm lên

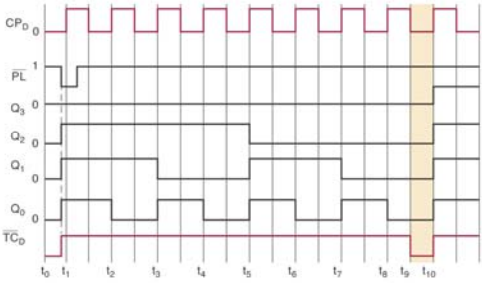




# 74ALS193 - bộ đếm xuống

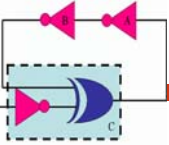


(a)



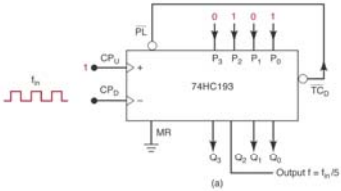
(b)

47

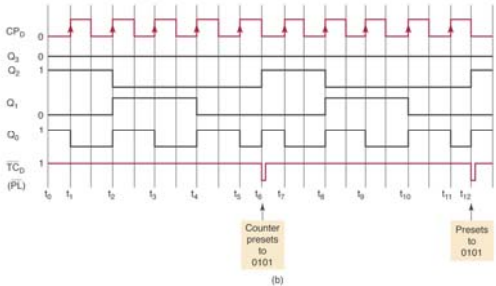


# Ví dụ 7-7

- Bộ đếm xuống MOD-5



(a)

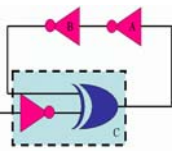


(b)

Counter presets to 0101

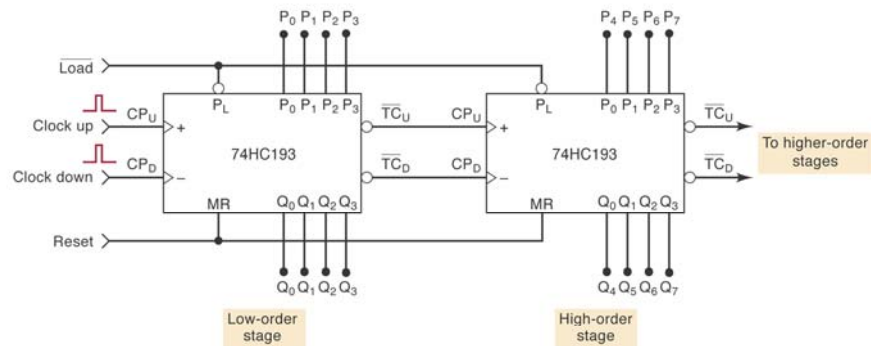
Presets to 0101

Output  $f = f_{CP} / 5$

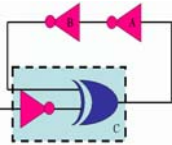


## IC 74ALS193

- Kết hợp nhiều IC để mở rộng bộ đếm



Note: Reset input has priority over Load and clock inputs.  
Load input has priority over clock inputs.



## Giải mã bộ đếm

- Giải mã là biết đổi giá trị nhị phân ở ngõ ra thành giá trị thập phân.
- Bộ giải mã sau tích cực mức cao, có thể sử dụng các đèn LED để thể hiện các số thập phân từ 0 – 7.



## Giải mã bộ đếm MOD-8

The diagram illustrates a 3-bit counter (MOD-8) implemented with three JK flip-flops (A, B, C) and a 3-to-8 decoder. The counter outputs are C, B, and A. The decoder outputs are labeled 0 through 7, corresponding to the binary values of CBA. The timing diagram shows the CLK signal and the resulting output waveforms for each decoder output.

CLK	CBA	0	1	2	3	4	5	6	7
0	000	1	0	0	0	0	0	0	0
1	001	0	1	0	0	0	0	0	0
2	010	0	0	1	0	0	0	0	0
3	011	0	0	0	1	0	0	0	0
4	100	0	0	0	0	1	0	0	0
5	101	0	0	0	0	0	1	0	0
6	110	0	0	0	0	0	0	1	0
7	111	0	0	0	0	0	0	0	1

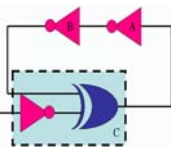
51

## Thiết kế bộ đếm đồng bộ

- Bảng chuyển đổi trạng thái của JK-FF

Chuyển đổi	Hiện tại	Kế tiếp	J	K
0→0	0	0	0	X
0→1	0	1	1	X
1→0	1	0	X	1
1→1	1	1	X	0

52

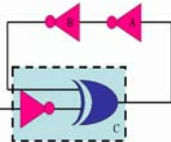


## Trình tự thiết kế



- Bước 1: Xác định số bit (số FF cần)
- Bước 2: vẽ sơ đồ chuyển đổi của tất cả các trạng thái, bao gồm cả những trạng thái không xuất hiện trong chu trình.
- Bước 3: dựa vào sơ đồ chuyển đổi trạng thái để thiết lập một bảng, trong đó liệt kê tất cả các trạng thái hiện tại và kế tiếp.

53

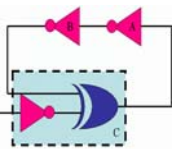


## Trình tự thiết kế



- Bước 4: trong bảng vừa tạo, thêm cột giá trị J, K của các FF. Với mỗi trạng thái hiện tại, xác định giá trị của J và K để bộ đếm chuyển đến trạng thái kế tiếp.
- Bước 5: thiết kế mạch logic để tạo ra các mức logic cho mỗi ngõ vào J và K.
- Bước 6: xác định sơ đồ mạch.

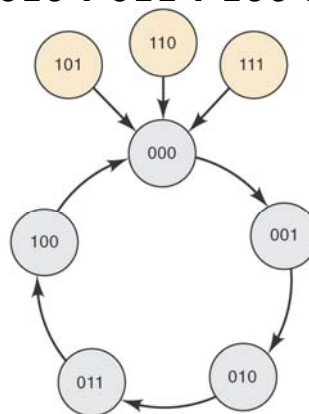
54



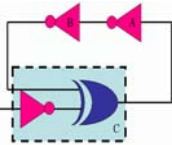
## Ví dụ, thiết kế bộ đếm MOD-5



- Sơ đồ chuyển đổi trạng thái
- 000→001→010→011→100→000→...



55

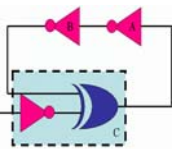


## Trạng thái hiện tại và kế tiếp



	PRESENT			NEXT		
	C	B	A	C	B	A
Line 1	0	0	0	0	0	1
2	0	0	1	0	1	0
3	0	1	0	0	1	1
4	0	1	1	1	0	0
5	1	0	0	0	0	0
6	1	0	1	0	0	0
7	1	1	0	0	0	0
8	1	1	1	0	0	0

56

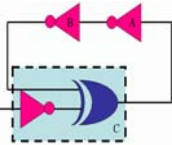


## Bảng trạng thái của mạch



	PRESENT			NEXT			$J_C$	$K_C$	$J_B$	$K_B$	$J_A$	$K_A$
	C	B	A	C	B	A						
Line 1	0	0	0	0	0	1	0	x	0	x	1	x
2	0	0	1	0	1	0	0	x	1	x	x	1
3	0	1	0	0	1	1	0	x	x	0	1	x
4	0	1	1	1	0	0	1	x	x	1	x	1
5	1	0	0	0	0	0	x	1	0	x	0	x
6	1	0	1	0	0	0	x	1	0	x	x	1
7	1	1	0	0	0	0	x	1	x	1	0	x
8	1	1	1	0	0	0	x	1	x	1	x	1

57



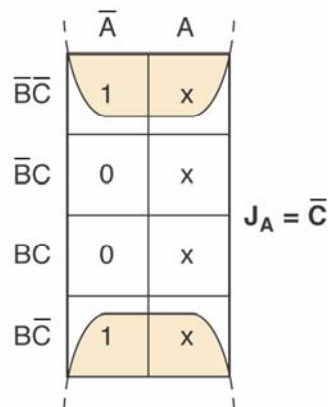
## Xác định các giá trị J và K



- Tính giá trị của  $J_A$

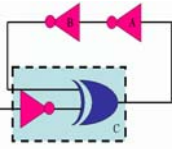
PRESENT			$J_A$
C	B	A	
0	0	0	1
0	0	1	x
0	1	0	1
0	1	1	x
1	0	0	0
1	0	1	x
1	1	0	0
1	1	1	x

(a)



(b)

58



# Tương tự ta có

	$\bar{A}$	A
$\bar{B}\bar{C}$	0	0
$\bar{B}C$	x	x
$BC$	x	x
$B\bar{C}$	0	1

$$J_C = AB$$

	$\bar{A}$	A
$\bar{B}\bar{C}$	x	x
$\bar{B}C$	1	1
$BC$	1	1
$B\bar{C}$	x	x

$$K_C = 1$$

	$\bar{A}$	A
$\bar{B}\bar{C}$	0	1
$\bar{B}C$	0	0
$BC$	x	x
$B\bar{C}$	x	x

$$J_B = A\bar{C}$$

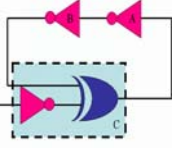
	$\bar{A}$	A
$\bar{B}\bar{C}$	x	x
$\bar{B}C$	x	x
$BC$	1	1
$B\bar{C}$	0	1

$$K_B = A + C$$

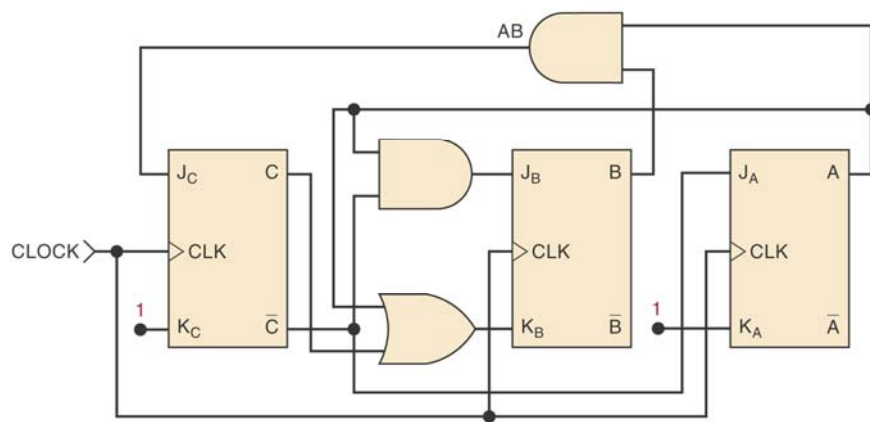
(a)

(b)

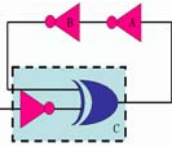
⇒



# Sơ đồ mạch



60

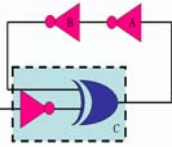


## Ví dụ 7-8

- Sử dụng D-FF để thiết kế bộ đếm MOD-5

PRESENT			NEXT			$D_C$	$D_B$	$D_A$
C	B	A	C	B	A			
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	x
0	1	0	0	1	1	0	x	1
0	1	1	1	0	0	1	x	x
1	0	0	0	0	0	x	0	0
1	0	1	0	0	0	x	0	x
1	1	0	0	0	0	x	x	0
1	1	1	0	0	0	x	x	x

61



## Ví dụ 7-8

- Xác định giá trị các ngõ vào D

	$\bar{A}$	A
$\bar{C}\bar{B}$	0	0
$\bar{C}B$	0	1
CB	0	0
$C\bar{B}$	0	0

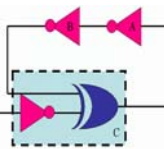
$$D_C = \bar{C}BA$$

	$\bar{A}$	A
$\bar{C}\bar{B}$	0	1
$\bar{C}B$	1	0
CB	0	0
$C\bar{B}$	0	0

$$D_B = \bar{C}\bar{B}A + \bar{C}B\bar{A}$$

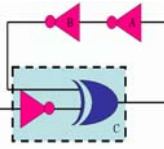
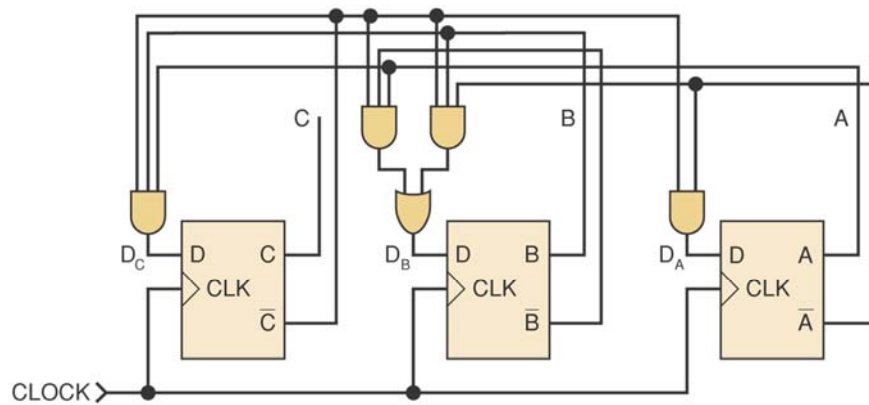
	$\bar{A}$	A
$\bar{C}\bar{B}$	1	0
$\bar{C}B$	1	0
CB	0	0
$C\bar{B}$	0	0

$$D_A = \bar{C}\bar{A}$$



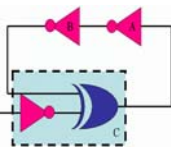
## Ví dụ 7-8

### ■ Sơ đồ mạch



## Mạch thanh ghi tích hợp

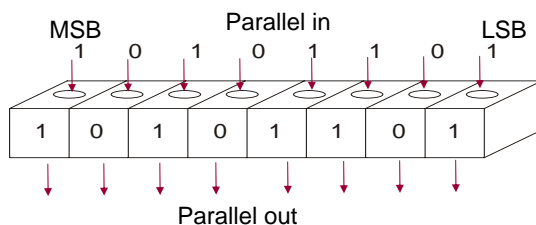
- Thanh ghi có thể được phân loại dựa vào cách dữ liệu được đưa vào và cách mà dữ liệu được lấy ra:
  - Parallel in/parallel out (PIPO)
  - Serial in/serial out (SISO)
  - Parallel in/serial out (PISO)
  - Serial in/parallel out (SIPO)



## Mạch thanh ghi tích hợp

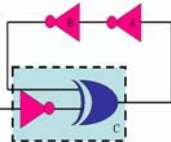


- Vào song song, ra song song :



- IC 74174 và 74178

65



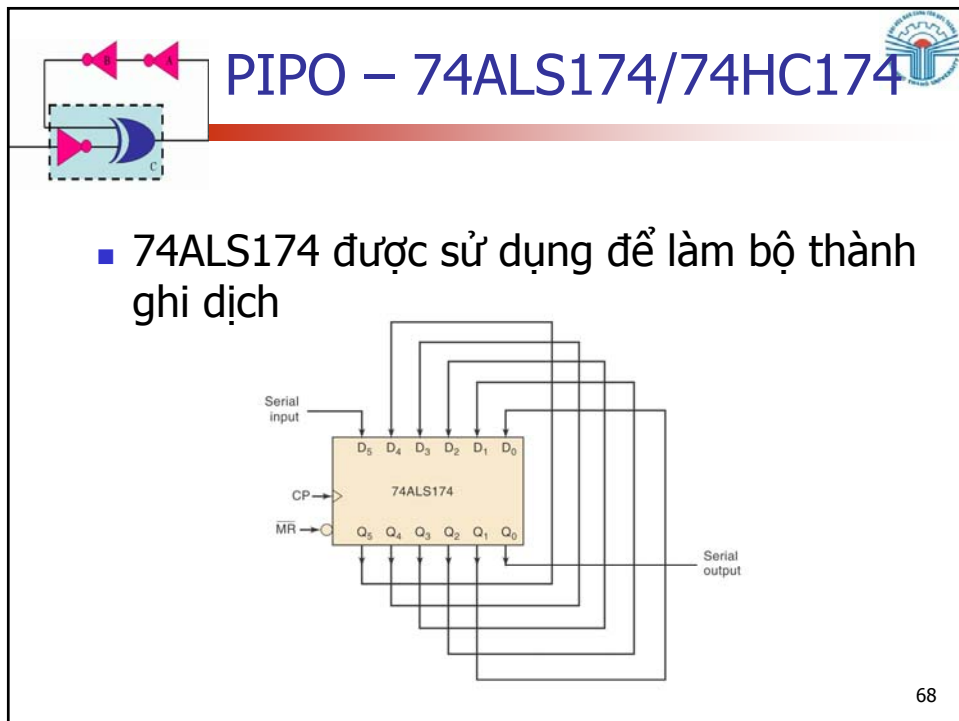
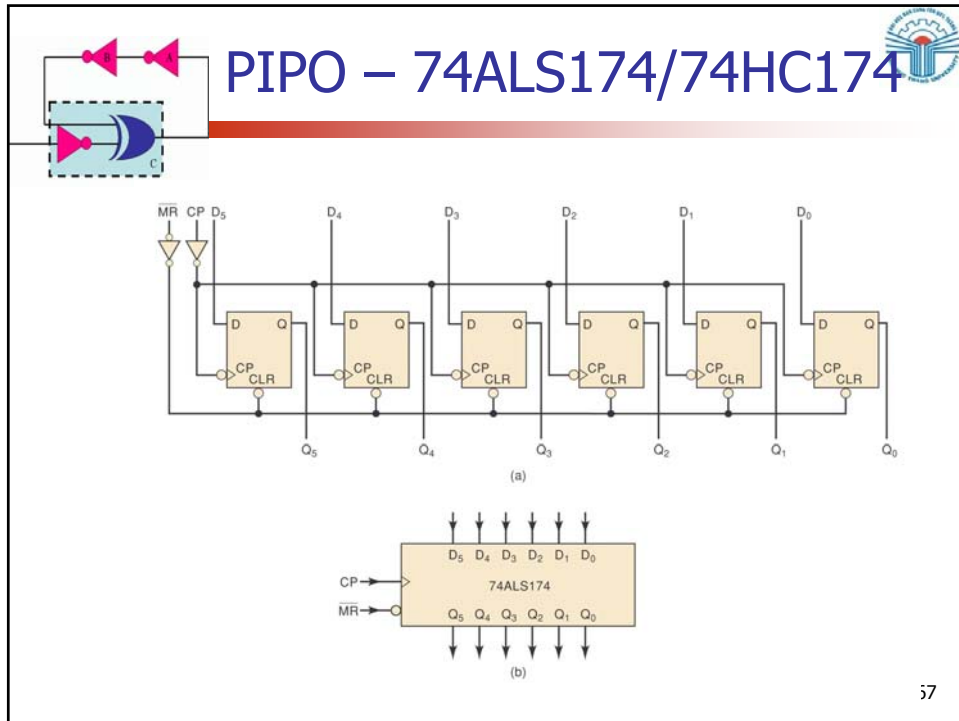
## PIPO – 74ALS174/74HC174

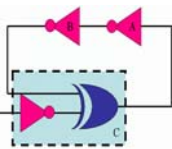


- 74ALS174/74HC174
  - Thanh ghi 6 bit
  - $D_5, D_0$ : ngõ vào song song
  - $Q_5, Q_0$ : ngõ ra song song
- Dữ liệu được đưa vào thanh ghi theo cạnh dương của xung clock
- Master reset có thể reset tất cả các FFs một cách độc lập với xung clock

66



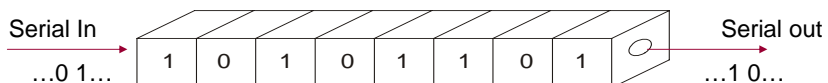




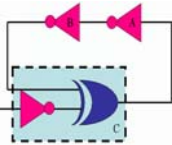
# Mạch thanh ghi tích hợp



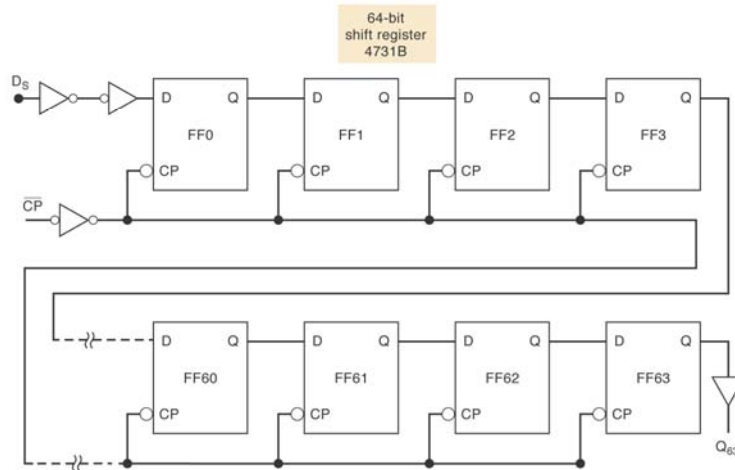
- Vào nối tiếp, ra nối tiếp:

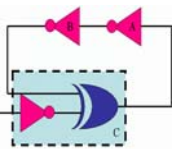


- IC 4731B



# SISO - 4731B

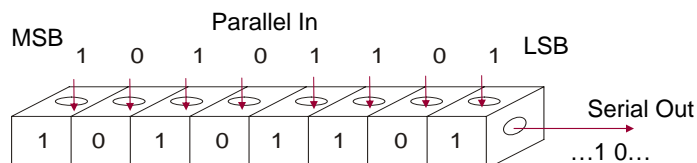




## Mạch thanh ghi tích hợp

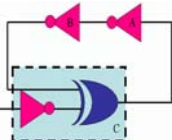


- Vào song song, ra nối tiếp:



- IC 74165, 74LS165, 74HC165

71

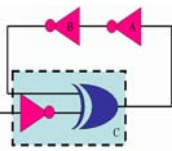


## PISO - 74HC165

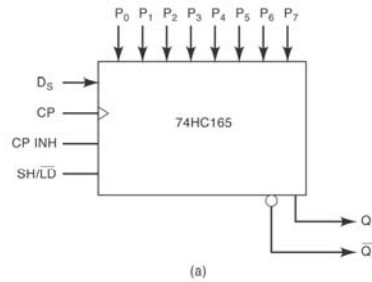


- 74HC165
  - Thanh ghi 8 bit
  - Dữ liệu nối tiếp được đưa vào  $D_5$
  - Dữ liệu song song không đồng bộ được đưa vào qua  $P_0 - P_7$
  - Chỉ có ngõ ra  $Q_7$  được sử dụng
- CP là ngõ vào xung clock
- CP INH ngõ vào ngăn xung clock
- $\overline{SH/LD}$  ngõ vào load dữ liệu

72



# PISO - 74HC165

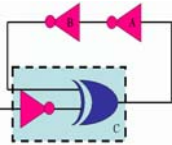


Function Table

Inputs			Operation
SH/LD	CP	CP INH	
L	X	X	Parallel load
H	H	X	No change
H	X	H	No change
H	F	L	Shifting
H	L	F	Shifting

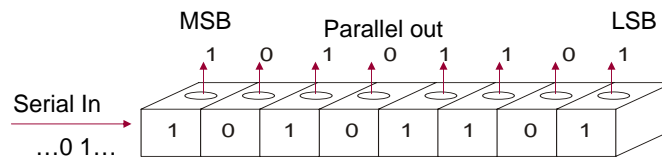
H = high level  
L = low level  
X = immaterial  
F = PGT

(b)

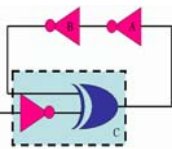


# Mạch thanh ghi tích hợp

- Vào nối tiếp, ra song song:



- IC 74164, 74LS164, 74HC164

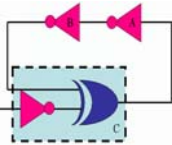


## SIPO – 74ALS164/74HC164

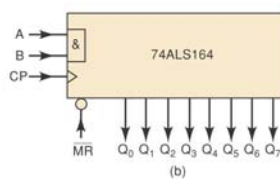
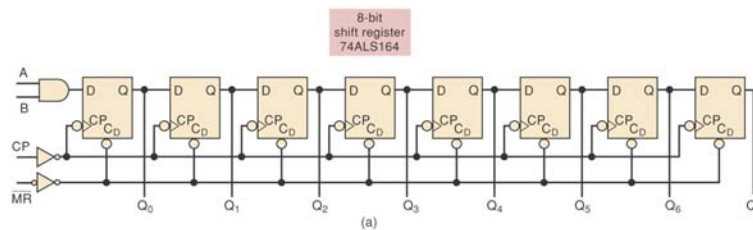


- 74ALS164
  - Chứa thanh ghi dịch 8 bit
  - A và B là hai ngõ vào của một cổng AND, ngõ ra của cổng AND là đầu vào nối tiếp.
- Quá trình dịch xảy ra khi có cạnh âm của xung clock

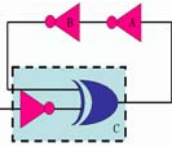
75



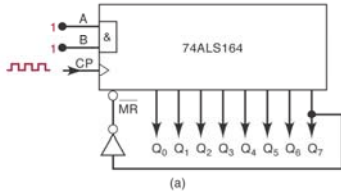
## IC 74ALS164



76



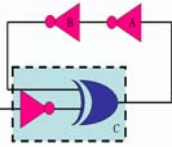
# Ví dụ IC 74ALS164



Input pulse number	Q <sub>0</sub>	Q <sub>1</sub>	Q <sub>2</sub>	Q <sub>3</sub>	Q <sub>4</sub>	Q <sub>5</sub>	Q <sub>6</sub>	Q <sub>7</sub>
0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0
3	1	1	1	0	0	0	0	0
4	1	1	1	1	0	0	0	0
5	1	1	1	1	1	0	0	0
6	1	1	1	1	1	1	0	0
7	1	1	1	1	1	1	1	0
8	1	1	1	1	1	1	1	1

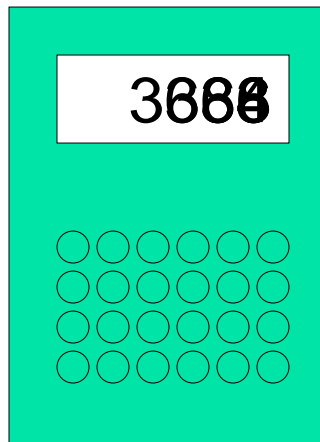
Temporary state

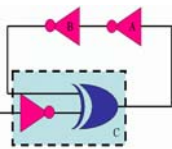
Recycles



# Thanh ghi dịch

Ví dụ về thanh ghi dịch trong máy tính

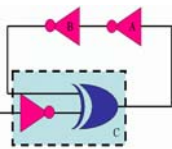




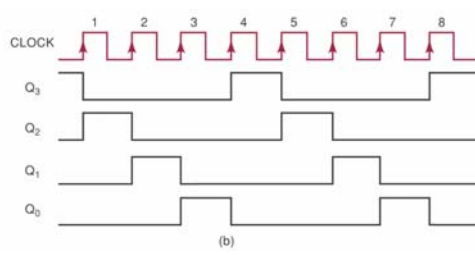
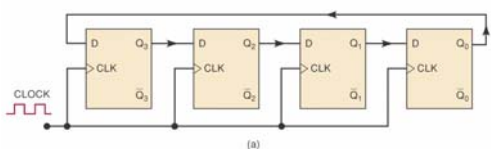
# Bộ đếm thanh ghi dịch



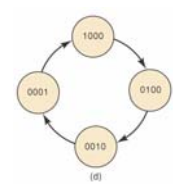
- Bộ đếm vòng (FF cuối nối đến FF đầu tiên)
  - FF cuối cùng sẽ dịch giá trị của nó đến FF đầu tiên
  - D-FF được sử dụng (JK-FF cũng có thể được sử dụng)
  - Phải bắt đầu với trạng thái chỉ có một FF có giá trị 1 và những cái còn lại ở trạng thái 0.

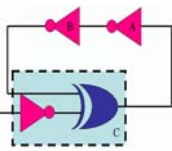


# Dịch vòng MOD-4



Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	CLOCK pulse
1	0	0	0	0
0	1	0	0	1
0	0	1	0	2
0	0	0	1	3
1	0	0	0	4
0	1	0	0	5
0	0	1	0	6
0	0	0	1	7
.	.	.	.	.
.	.	.	.	.



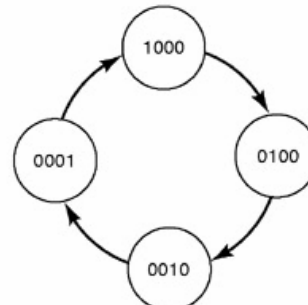


## Dịch vòng 4 bit

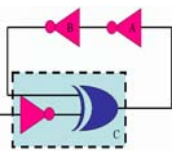
### ■ Bộ đếm MOD-4

Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>	Q <sub>0</sub>	CLOCK pulse
1	0	0	0	0
0	1	0	0	1
0	0	1	0	2
0	0	0	1	3
1	0	0	0	4
0	1	0	0	5
0	0	1	0	6
0	0	0	1	7
.	.	.	.	.
.	.	.	.	.

(c)



(d)

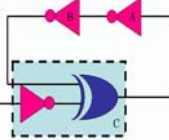


## Câu hỏi?





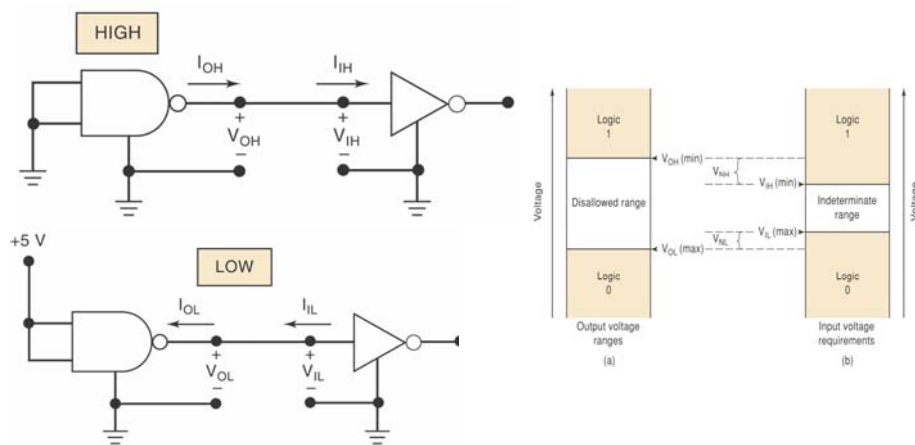
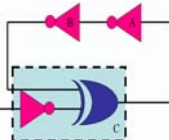
# Chương 8 Đặc điểm của IC số

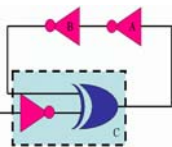


Th.S Đặng Ngọc Khoa  
Khoa Điện - Điện Tử



# Thông số dòng và áp





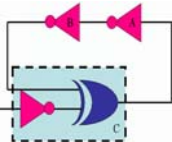
## Thông số dòng và áp



### ■ Mức điện áp

- $V_{IH}(\text{min})$ : Điện áp ngõ vào mức cao: giá trị điện áp thấp nhất cho mức logic 1 ở ngõ vào.
- $V_{IL}(\text{max})$ : Điện áp ngõ vào mức thấp: giá trị điện áp cao nhất cho mức logic 0 ở ngõ vào.
- $V_{OH}(\text{min})$ : Điện áp ngõ ra mức cao: giá trị điện áp thấp nhất cho mức logic 1 ở ngõ ra.
- $V_{OL}(\text{max})$ : Điện áp ngõ ra mức thấp: giá trị điện áp cao nhất cho mức logic 0 ở ngõ ra.

3



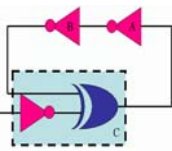
## Thông số dòng và áp



### ■ Mức dòng điện

- $I_{IH}$ : Dòng điện ngõ vào mức cao: dòng điện chảy vào ở mức logic 1
- $I_{IL}$ : Dòng điện ngõ vào mức thấp: dòng điện chảy vào ở mức logic 0
- $I_{OH}$ : Dòng điện ngõ ra mức cao: dòng điện chảy ra ở mức logic 1
- $I_{OL}$ : Dòng điện ngõ ra mức thấp: dòng điện chảy ra ở mức logic thấp

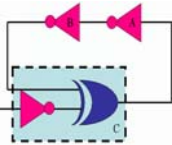
4



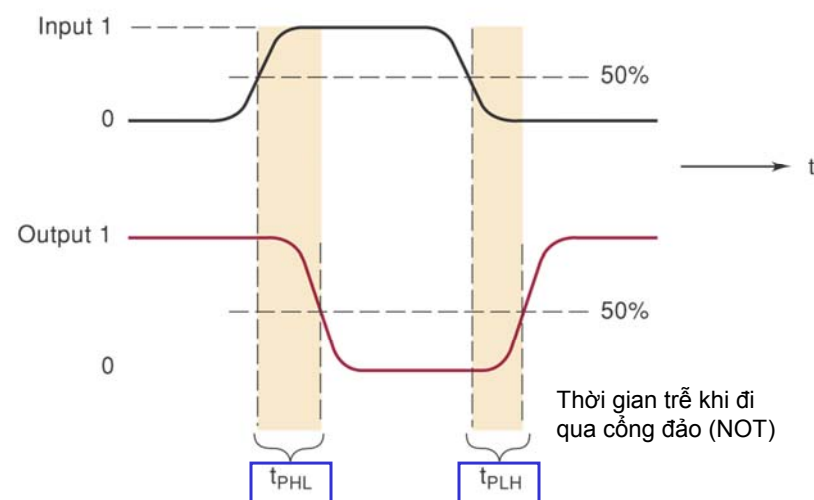
## Thời gian trễ

- Xét tín hiệu đi qua một cổng đảo:
  - $t_{PLH}$  thời gian trễ khi chuyển logic 0 sang logic 1
  - $t_{PHL}$  thời gian trễ khi chuyển logic 1 sang logic 0
- $t_{PLH}$  và  $t_{PHL}$  không nhất thiết phải bằng nhau
- Thời gian trễ liên quan đến tốc độ của mạch logic. Thời gian trễ càng nhỏ thì tốc độ của mạch càng cao.

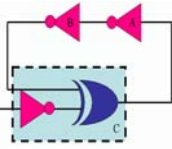
5



## Thời gian trễ



6



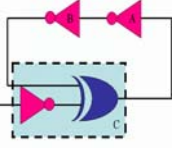
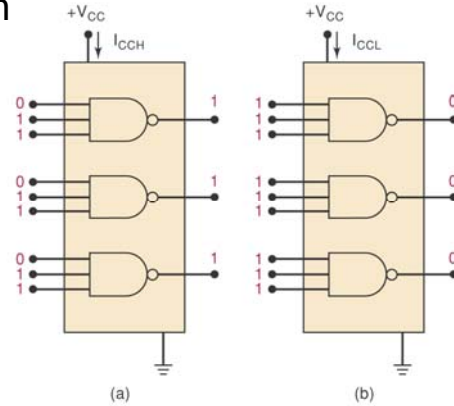
# Nguồn cung cấp

- Dòng điện trung bình

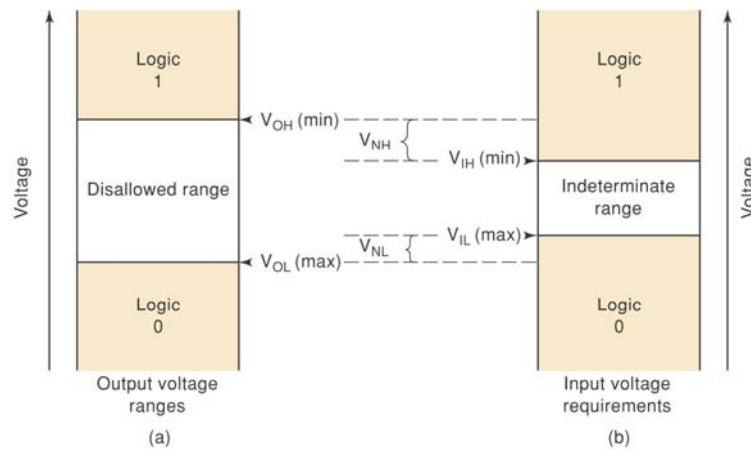
$$I_{CC(avg)} = \left( \frac{I_{CCH} + I_{CCL}}{2} \right)$$

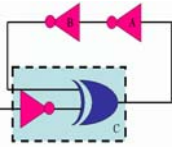
- Công suất

$$P_{(avg)} = I_{CC(avg)} \cdot V_{CC}$$



# Ảnh hưởng của nhiễu

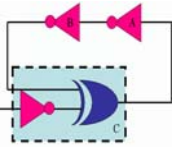




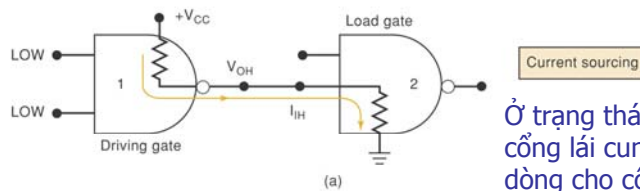
## Mức điện áp

- Mạch hoạt động đúng yêu cầu điện áp ngõ vào nằm trong khoảng xác định nhỏ hơn  $V_{IL}(\max)$  hoặc lớn hơn  $V_{IH}(\min)$
- Điện áp ngõ ra có thể nằm ngoài khoảng xác định phụ thuộc vào nhà sản xuất hoặc trong trường hợp quá tải.
- Nguồn cung cấp có mức điện áp không đúng có thể sẽ gây ra mức điện áp ngõ ra không đúng.

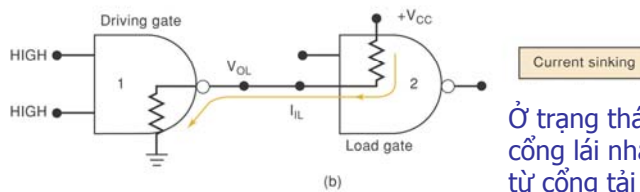
9



## Current-Sourcing và Current- Sinking

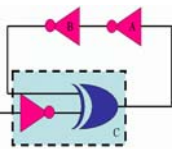


Ở trạng thái logic cao, cổng lái cung cấp dòng cho cổng tải



Ở trạng thái logic thấp, cổng lái nhận dòng từ cổng tải

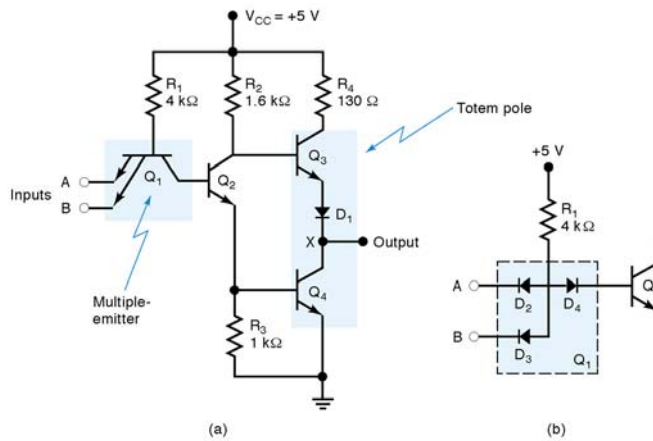
10



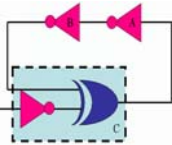
# Họ IC TTL



## ■ Sơ đồ mạch của cổng NAND TTL



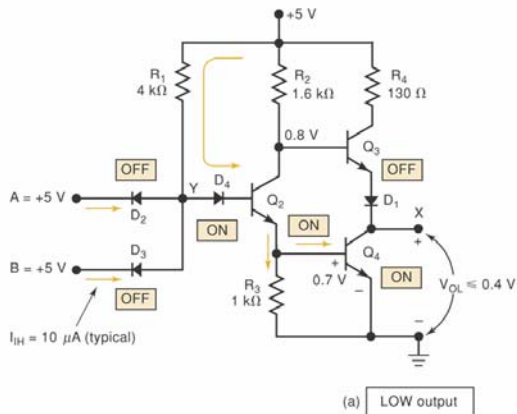
11



# IC TTL cổng NAND



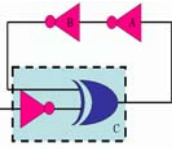
## ■ Ngõ ra ở trạng thái thấp



Input conditions	Output conditions
A and B are both HIGH ( $\geq 2V$ )	$Q_3$ OFF
Input currents are very low $I_{IH} = 10 \mu A$	$Q_4$ ON so that $V_X$ is LOW ( $\leq 0.4V$ )

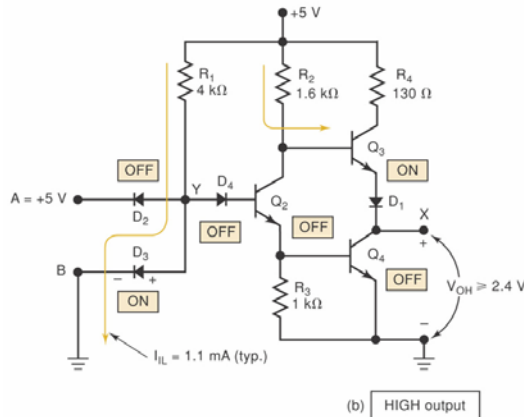
(a) LOW output

12



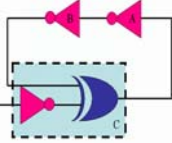
# IC TTL cổng NAND

## ■ Ngõ ra ở trạng thái cao

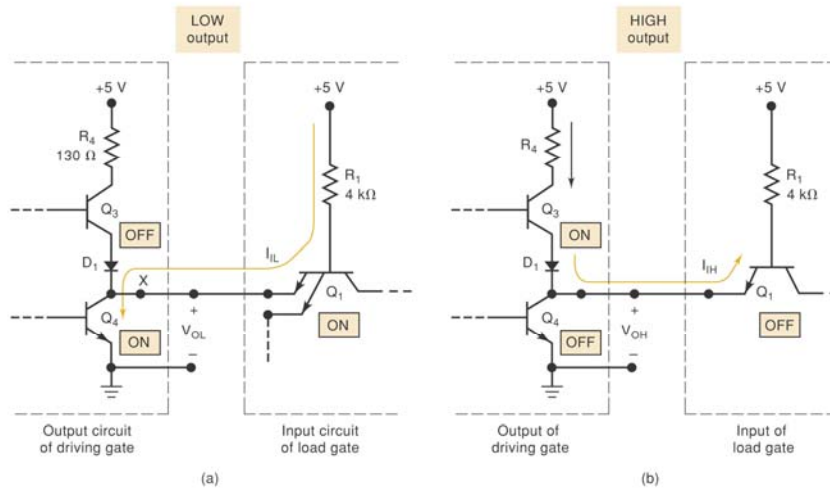


Input conditions	Output conditions
A or B or both are LOW ( $\leq 0.8\text{ V}$ )	Q <sub>4</sub> OFF
Current flows back to ground through LOW input terminal. $I_{IL} = 1.1\text{ mA}$	Q <sub>3</sub> acts as emitter-follower and $V_{OH} \approx 2.4\text{ V}$ , typically 3.6 V

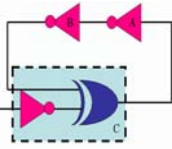
13



# Kết nối giữa các mạch logic



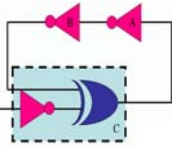
14



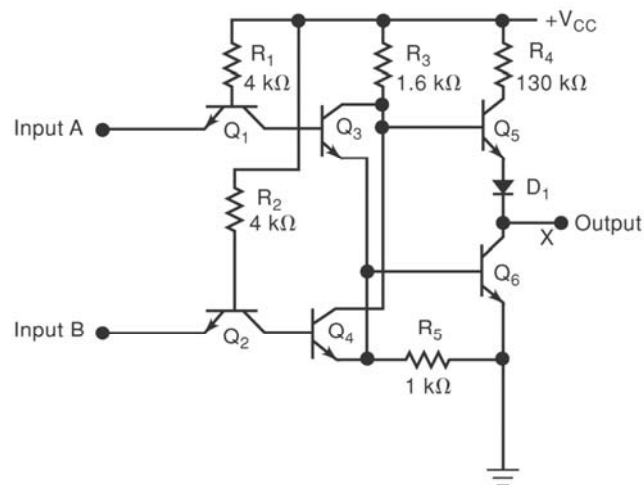
## Họ IC TTL

- Mạch TTL có cấu trúc tương tự như trên
- Ngõ vào là cathode của tiếp giáp PN
  - Ngõ vào ở mức cao sẽ turn off mỗi nối và chỉ có dòng rò rỉ chạy qua.
  - Ngõ vào mức thấp sẽ turns on mỗi nối và có dòng tương đối lớn chạy qua.
- Phần lớn mạch TTL có cùng cấu trúc ngõ ra, tương tự như trên.

15

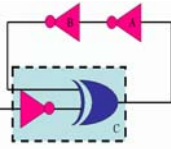


## Mạch TTL cổng NOR



16



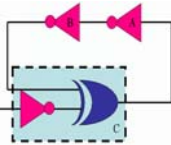


## Ký hiệu họ TTL



- Ký hiệu đầu tiên của IC TTL là số series 54/74
  - Series 54 hoạt động trong khoảng nhiệt độ rộng hơn.
- Ký hiệu chữ thể hiện hãng sản xuất
  - SN – Texas Instruments
  - DM – National Semiconductor
  - S – Signetics
  - DM7402, SN7402, S7402 có cùng một chức năng

17

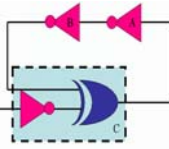


## Ký hiệu họ TTL



- Chuẩn 74 TTL có thể phân loại thành:
  - Standard TTL, 74 series
  - Schottky TTL, 74S series
  - Low power Schottky TTL, 74LS series (LS-TTL)
  - Advanced Schottky TTL, 74AS series (AS-TTL)
  - Advanced low power Schottky TTL, 74ALS series
  - 74F fast TTL
- Bảng 8-6 so sánh giữa các phân loại khác nhau

18

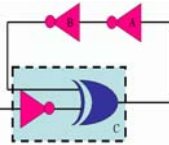


# Phân loại họ TTL



TABLE 8-6 Typical TTL series characteristics.

	74	74S	74LS	74AS	74ALS	74F
<b>Performance ratings</b>						
Propagation delay (ns)	9	3	9.5	1.7	4	3
Power dissipation (mW)	10	20	2	8	1.2	6
Speed-power product (pJ)	90	60	19	13.6	4.8	18
Max. clock rate (MHz)	35	125	45	200	70	100
Fan-out (same series)	10	20	20	40	20	33
<b>Voltage parameters</b>						
$V_{OH}(\text{min})$	2.4	2.7	2.7	2.5	2.5	2.5
$V_{OL}(\text{max})$	0.4	0.5	0.5	0.5	0.5	0.5
$V_{IH}(\text{min})$	2.0	2.0	2.0	2.0	2.0	2.0
$V_{IL}(\text{max})$	0.8	0.8	0.8	0.8	0.8	0.8



# DataSheet họ TTL



recommended operating conditions

	SN54ALS00A			SN74ALS00A			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
$V_{CC}$ Supply voltage	4.5	5	5.5	4.5	5	5.5	V
$V_{IH}$ High-level input voltage	2			2			V
$V_{IL}$ Low-level input voltage	0.8 <sup>†</sup>			0.8			V
$I_{OH}$ High-level output current	-0.4			-0.4			mA
$I_{OL}$ Low-level output current	8			8			mA
$T_A$ Operating free-air temperature	-55			125			°C

<sup>†</sup> Applies over temperature range -55°C to 70°C  
<sup>‡</sup> Applies over temperature range 70°C to 125°C

electrical characteristics over recommended operating free-air temperature range unless otherwise noted

PARAMETER	TEST CONDITIONS	SN54ALS00A			SN74ALS00A			UNIT
		MIN	TYP <sup>‡</sup>	MAX	MIN	TYP <sup>‡</sup>	MAX	
$V_{IK}$	$V_{CC} = 4.5\text{ V}$ , $I_I = -18\text{ mA}$	-1.2			-1.5			V
$V_{OH}$	$V_{CC} = 4.5\text{ V to }5.5\text{ V}$ , $I_{OH} = -0.4\text{ mA}$	$V_{CC} - 2$			$V_{CC} - 2$			V
$V_{OL}$	$V_{CC} = 4.5\text{ V}$ , $I_{OL} = 4\text{ mA}$ $I_{OL} = 8\text{ mA}$	0.25 0.4			0.25 0.4			V
$I_I$	$V_{CC} = 5.5\text{ V}$ , $V_I = 7\text{ V}$	0.1			0.1			mA
$I_{IH}$	$V_{CC} = 5.5\text{ V}$ , $V_I = 2.7\text{ V}$	20			20			µA
$I_{IL}$	$V_{CC} = 5.5\text{ V}$ , $V_I = 0.4\text{ V}$	-0.1			-0.1			mA
$I_{O1}^{\dagger}$	$V_{CC} = 5.5\text{ V}$ , $V_O = 2.25\text{ V}$	-20			-112			mA
$I_{OCH}^{\dagger}$	$V_{CC} = 5.5\text{ V}$ , $V_I = 0$	0.5 0.85			0.5 0.85			mA
$I_{OCL}^{\dagger}$	$V_{CC} = 5.5\text{ V}$ , $V_I = 4.5\text{ V}$	1.5 3			1.5 3			mA

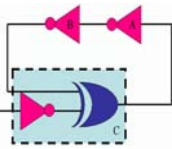
<sup>†</sup> All typical values are at  $V_{CC} = 5\text{ V}$ ,  $T_A = 25^\circ\text{C}$ .

<sup>‡</sup> The output conditions have been chosen to produce a current that closely approximates one half of the true short-circuit output current,  $I_{OS}$ .

switching characteristics (see Figure 1)

PARAMETER	FROM (INPUT)	TO (OUTPUT)	$V_{CC} = 4.5\text{ V to }5.5\text{ V}$ , $C_L = 50\text{ pF}$ , $R_L = 500\ \Omega$ , $T_A = \text{MIN to MAX}^{\S}$				UNIT
			SN54ALS00A		SN74ALS00A		
			MIN	MAX	MIN	MAX	
$t_{PLH}$	A or B	Y	3	15	3	11	ns
$t_{PHL}$			2	9	2	8	ns

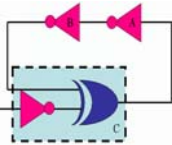
<sup>§</sup> For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.



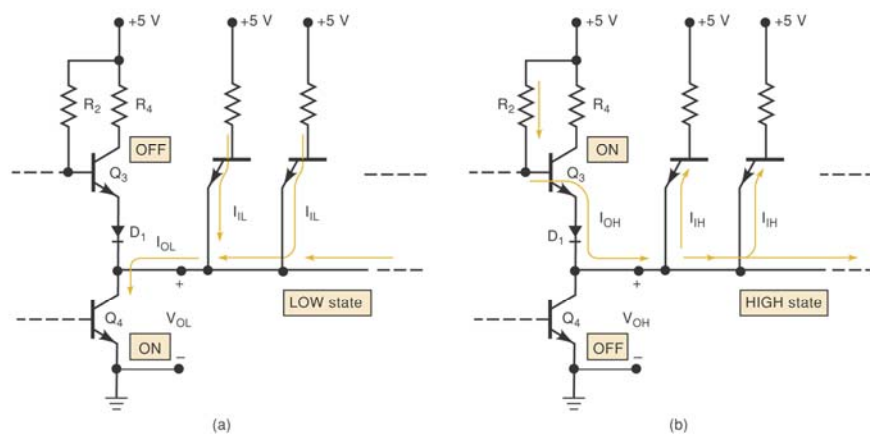
## Tải và Fan-Out

- Fan out thể hiện khả năng ngõ ra của một IC có thể lái được bao nhiêu ngõ vào của những IC khác.
  - Một ngõ ra TTL bị giới hạn dòng chảy vào nó khi ở trạng thái thấp.
  - Một ngõ ra TTL bị giới hạn dòng cung cấp (dòng chảy ra) khi nó ở trạng thái cao.
  - Nếu dòng điện vượt quá những giới hạn trên thì điện áp ngõ ra sẽ nằm ngoài khoảng cho phép.

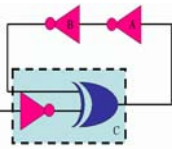
21



## Tải và Fan-Out



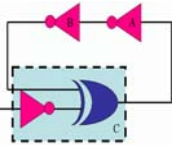
22



## Tải và Fan-Out

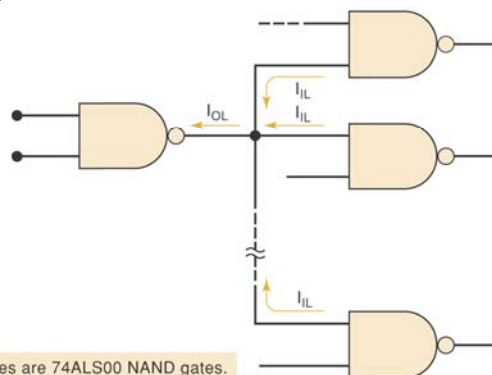
- Xác định fan out
  - Cộng  $I_{IH}$  của tất cả các ngõ vào có kết nối đến ngõ ra đang xét. Tổng phải nhỏ hơn  $I_{OH}$ .
  - Cộng  $I_{IL}$  của tất cả các ngõ vào có kết nối đến ngõ ra đang xét. Tổng phải nhỏ hơn  $I_{OL}$ .

23



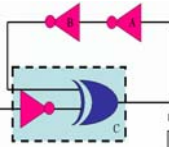
## Tải và Fan-Out

- Ví dụ: ngõ ra của 74ALS00 có thể lái bao nhiêu ngõ vào 74ALS00?



\*All gates are 74ALS00 NAND gates.

24



# DataSheet của 74ALS00

## recommended operating conditions

		SN54ALS00A			SN74ALS00A			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
V <sub>CC</sub>	Supply voltage	4.5	5	5.5	4.5	5	5.5	V
V <sub>IH</sub>	High-level input voltage	2			2			V
V <sub>IL</sub>	Low-level input voltage	0.8 <sup>‡</sup>			0.8			V
I <sub>OH</sub>	High-level output current	-0.4			-0.4			mA
I <sub>OL</sub>	Low-level output current	4			8			mA
T <sub>A</sub>	Operating free-air temperature	-55			125			°C

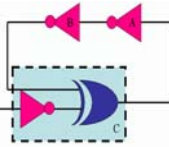
<sup>‡</sup> Applies over temperature range -55°C to 70°C  
<sup>§</sup> Applies over temperature range 70°C to 125°C

## electrical characteristics over recommended operating free-air temperature range unless otherwise noted

PARAMETER	TEST CONDITIONS	SN54ALS00A			SN74ALS00A			UNIT
		MIN	TYP <sup>†</sup>	MAX	MIN	TYP <sup>†</sup>	MAX	
V <sub>IK</sub>	V <sub>CC</sub> = 4.5 V, I <sub>I</sub> = -18 mA	-1.2			-1.5			V
V <sub>OH</sub>	V <sub>CC</sub> = 4.5 V to 5.5 V, I <sub>OH</sub> = -0.4 mA	V <sub>CC</sub> - 2			V <sub>CC</sub> - 2			V
V <sub>OL</sub>	V <sub>CC</sub> = 4.5 V, I <sub>OL</sub> = 4 mA	0.25			0.25			V
	I <sub>OL</sub> = 8 mA	0.4			0.5			V
I <sub>I</sub>	V <sub>CC</sub> = 5.5 V, V <sub>I</sub> = 7 V	0.1			0.1			mA
I <sub>IH</sub>	V <sub>CC</sub> = 5.5 V, V <sub>I</sub> = 2.7 V	20			20			μA
I <sub>IL</sub>	V <sub>CC</sub> = 5.5 V, V <sub>I</sub> = 0.4 V	-0.1			-0.1			mA
I <sub>O<sup>†</sup></sub>	V <sub>CC</sub> = 5.5 V, V <sub>O</sub> = 2.25 V	-20			-30			mA
I <sub>CCH</sub>	V <sub>CC</sub> = 5.5 V, V <sub>I</sub> = 0	0.5			0.5			mA
I <sub>CCL</sub>	V <sub>CC</sub> = 5.5 V, V <sub>I</sub> = 4.5 V	1.5			1.5			mA

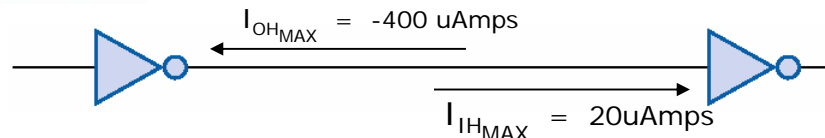
<sup>†</sup> All typical values are at V<sub>CC</sub> = 5 V, T<sub>A</sub> = 25°C.

<sup>‡</sup> The output conditions have been chosen to produce a current that closely approximates one half of the true short-circuit output current, I<sub>O<sup>§</sup></sub>.



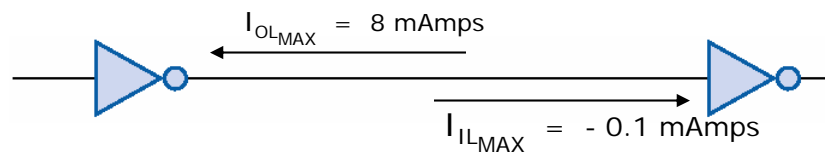
# Tính Fan - out

## High Output

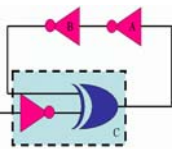


$$\text{Fanout}_{\text{HIGH}} = 400\text{uAmp} / 20 \text{ uAmp} = 20$$

## Low Output



$$\text{Fanout}_{\text{LOW}} = 8 \text{ mAmp} / 0.4 \text{ mAmp} = 80$$

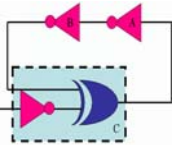


## Công nghệ MOS



- MOSFETs - Metal Oxide Semiconductor Field Effect Transistors
  - Sản xuất đơn giản và rẻ hơn
  - Tiêu tốn ít năng lượng
  - Có thể thực hiện được nhiều mạch hơn
  - Dễ bị tác động bởi tĩnh điện

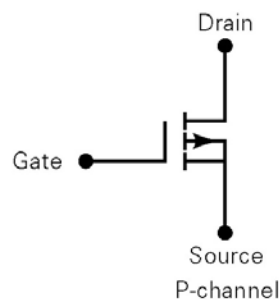
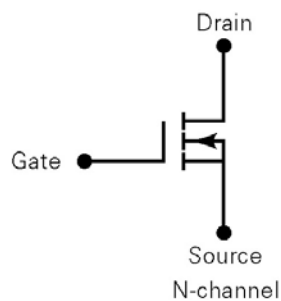
27



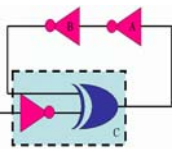
## Công nghệ MOS



- Ký hiệu MOSFET kênh N và kênh P



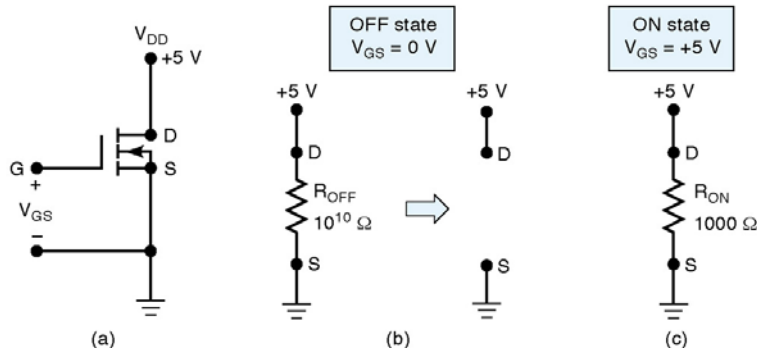
28



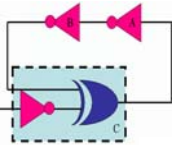
# Công nghệ MOS



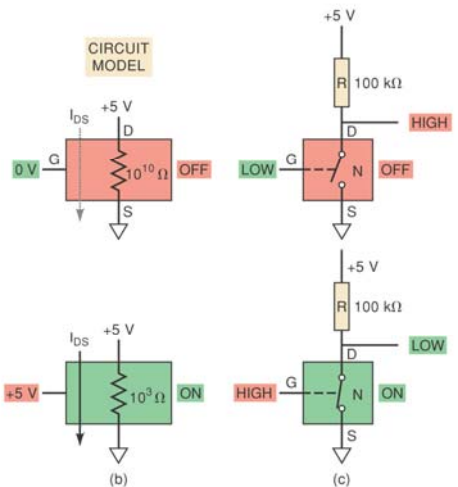
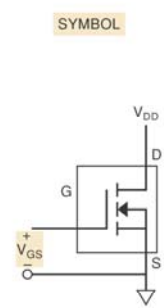
## Trạng thái của MOSFET



29



# Logic MOSFET kênh N



30

## Logic MOSFET kênh P

(a) (b) (c)

31

## Công đảo CMOS

$V_{IN}$	$Q_1$	$Q_2$	$V_{OUT}$
$+V_{DD}$ (logic 1)	OFF $R_{OFF} = 10^{10} \Omega$	ON $R_{ON} = 1 \text{ k}\Omega$	$= 0 \text{ V}$
$0 \text{ V}$ (logic 0)	ON $R_{ON} = 1 \text{ k}\Omega$	OFF $R_{OFF} = 10^{10} \Omega$	$= +V_{DD}$

$V_{OUT} = \overline{V_{IN}}$

32



## Cổng NAND CMOS

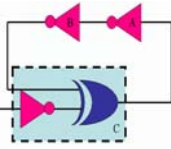
A	B	X
LOW	LOW	HIGH
LOW	HIGH	HIGH
HIGH	LOW	HIGH
HIGH	HIGH	LOW

33

## Cổng NOR CMOS

A	B	X
LOW	LOW	HIGH
LOW	HIGH	LOW
HIGH	LOW	LOW
HIGH	HIGH	LOW

34

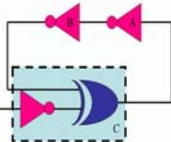


## Ký hiệu họ CMOS



- 4000/1400
- 74C
- 74HC/HCT (high-speed CMOS)
- 74AC/ACT (advanced CMOS)
- 74AHC/AHCT (advanced high-speed CMOS)

35

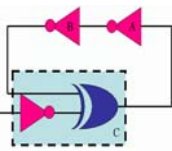


## IC điện áp thấp



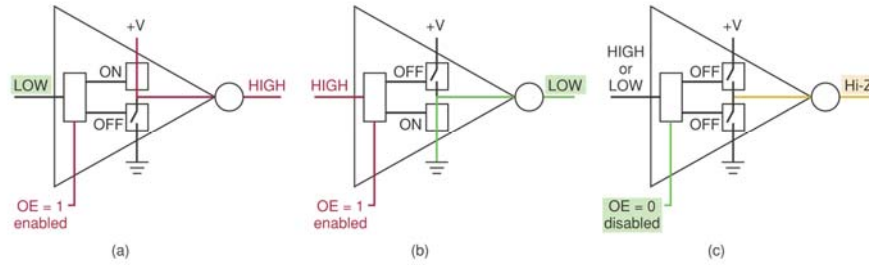
- Họ CMOS :
  - 74LVC (low voltage CMOS)
  - 74ALVC (advanced low voltage CMOS)
  - 74LV (low voltage)
  - 74AVC (advanced very low voltage CMOS)
  - 74AUC (advanced ultra-low voltage CMOS)
  - 74AUP (advanced ultra-low power)
  - 74CBT (cross bar technology)
  - 74CBTLV (cross bar technology low voltage)
  - 74GTLP (gunning transceiver logic plus)
  - 74SSTV (stub series terminated logic)
  - 74TVC (translation voltage clamp)

36

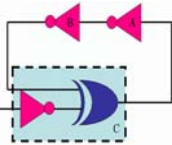


# Cổng Tristate

- Ba trạng thái đó là: HIGH, LOW và tổng trở cao.

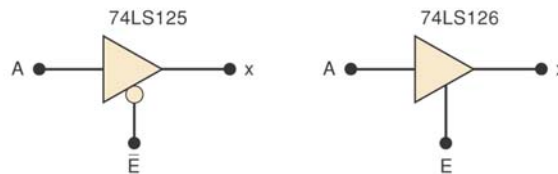


37



# Cổng Tristate

- Tristate được sử dụng làm bộ đệm



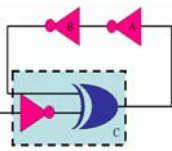
$\bar{E}$	x
0	A
1	Hi-Z

(a)

E	x
0	Hi-Z
1	A

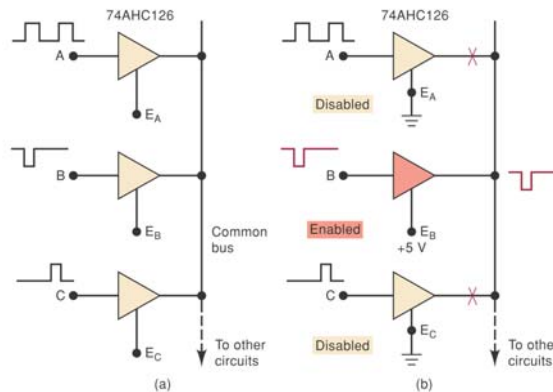
(b)

38

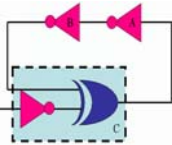


## Cổng Tristate

- Bộ đệm Tristate được sử dụng khi nhiều tín hiệu sử dụng chung bus



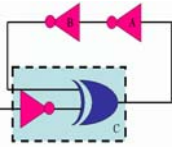
39



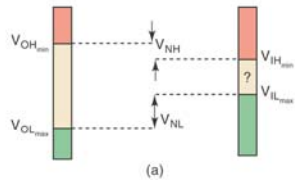
## Giao tiếp giữa các IC

- Lái (driver) – cung cấp tín hiệu ngõ ra.
- Tải (load) – nhận tín hiệu.
- Mạch giao tiếp (interface circuit) – kết nối giữa thiết bị lái và tải.
- Kết nối giữa những họ IC khác nhau trong cùng một mạch.

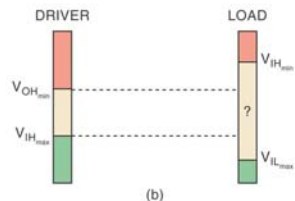
40



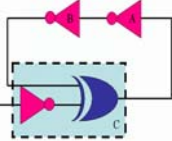
# Giao tiếp giữa các IC



HIGHS are high enough  
and  
LOWS are low enough  
NO INTERFACE NEEDED:  
DIRECT CONNECT  
(ASSUMING CURRENT LOADING  
ACCEPTABLE)

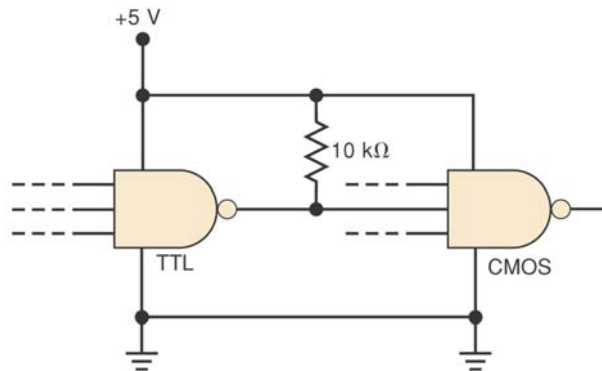


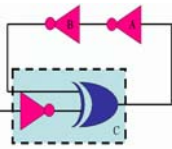
HIGHS not high enough  
or  
LOWS not low enough  
REQUIRES INTERFACE CIRCUIT



# TTL lái CMOS

- Về dòng điện, TTL hoàn toàn có thể lái được CMOS.
- Về điện áp, cần phải có điện trở kéo lên khi

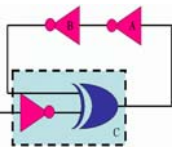




## CMOS lái TTL

- Trạng thái HIGH
  - Không có vấn đề gì xảy ra khi CMOS lái TTL
- Trạng thái LOW
  - Không có vấn đề với họ 74HC, 74HCT
  - Những họ khác tùy theo từng trường hợp mà có sự tương thích với nhau

43

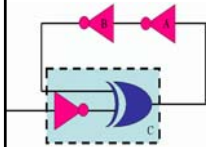


## Câu hỏi?

44



## Chương 9 Các mạch số thường gặp

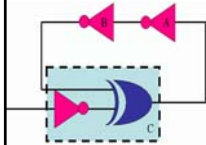


Th.S Đặng Ngọc Khoa  
Khoa Điện - Điện Tử

1

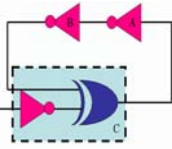


## Nội dung



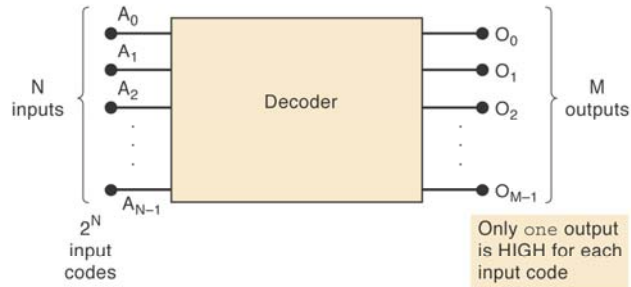
- Mạch giải mã/Mạch mã hóa
- Mạch ghép kênh
- Mạch phân kênh
- Mạch so sánh
- Chuyển mã
- Data Bus

2

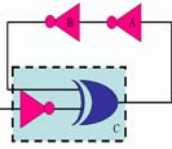


# Mạch giải mã

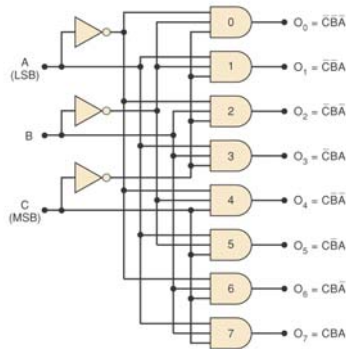
- Ứng với mỗi trạng thái của ngõ vào chỉ có một ngõ ra ở trạng thái tích cực.
- Mức tích cực có thể là mức thấp hoặc cao



3



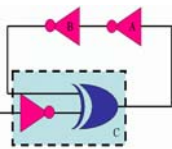
# Mạch giải mã 1 sang 8



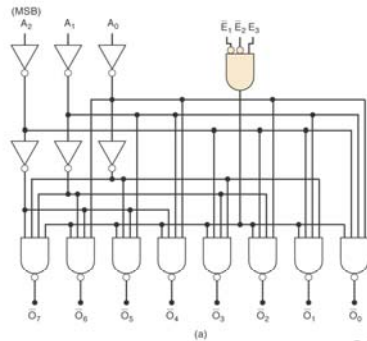
C	B	A	O <sub>7</sub>	O <sub>6</sub>	O <sub>5</sub>	O <sub>4</sub>	O <sub>3</sub>	O <sub>2</sub>	O <sub>1</sub>	O <sub>0</sub>
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	1	0	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

4



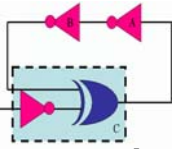
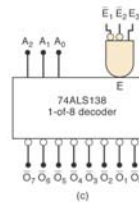


# IC giải mã 74LS138



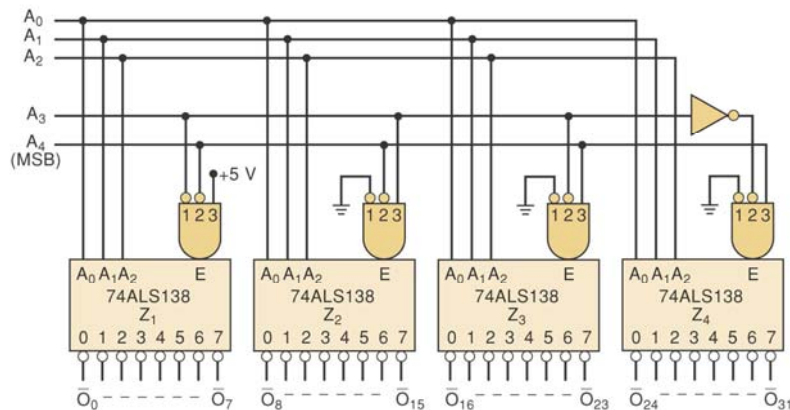
$E_1$	$E_2$	$E_3$	Outputs
0	0	1	Respond to input code $A_2A_1A_0$
1	X	X	Disabled - all HIGH
X	1	X	Disabled - all HIGH
X	X	0	Disabled - all HIGH

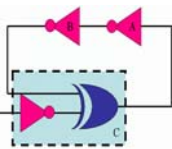
(b)



# Bộ giải mã 1 sang 32

- Có thể sử dụng 4 IC 74ALS138 để làm bộ giải mã 1 sang 32



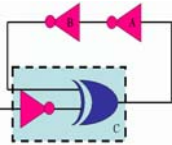


## Giải mã BCD - Decimal

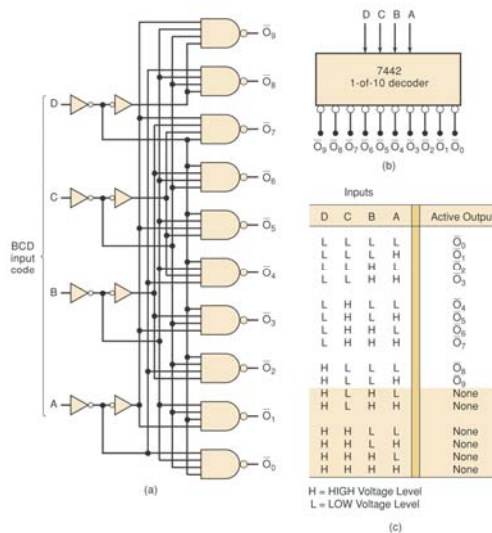


- Một số mạch giải mã không sử dụng tất cả  $2^N$  ngõ vào của nó
- Mạch giải mã BCD – Decimal có 4 ngõ vào và 10 ngõ ra
- Ngõ ra ở trạng thái tích cực (mức thấp) chỉ khi mã BCD tương ứng với nó được đưa đến ngõ vào
- Khi ngõ vào không phải là giá trị BCD thì không có ngõ ra nào tích cực
- IC 74LS42 giải mã BCD - Decimal

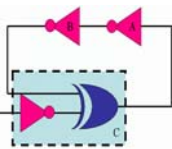
7



## Giải mã BCD - Decimal



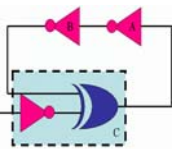
8



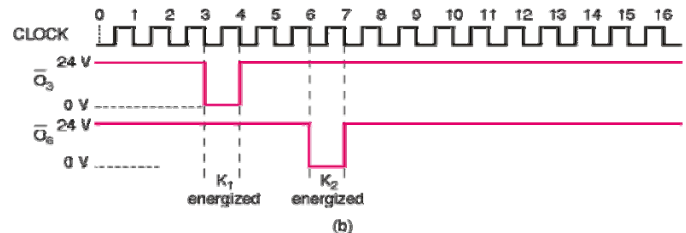
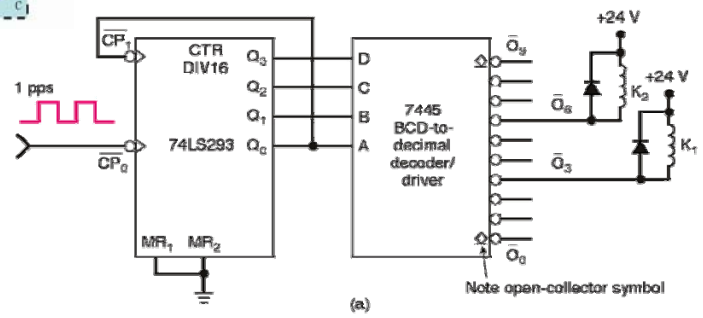
# Ứng dụng mạch giải mã

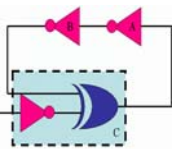


- Kết hợp bộ đếm và bộ giải mã để cung cấp tín hiệu theo trình tự thời gian cho các thiết bị
- Bộ đếm không đồng bộ 74LS239 hoạt động ở MOD-16
- Ngõ ra của bộ đếm được đưa đến ngõ vào của mạch giải mã



# Ứng dụng mạch giải mã

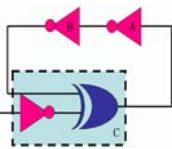
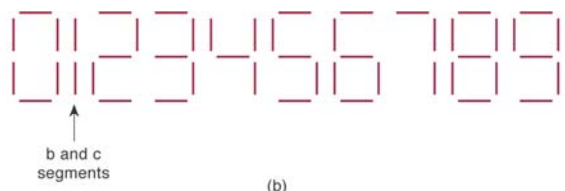
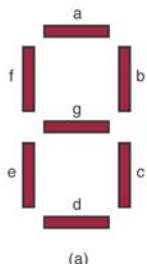




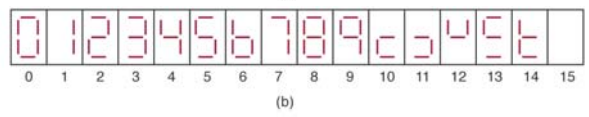
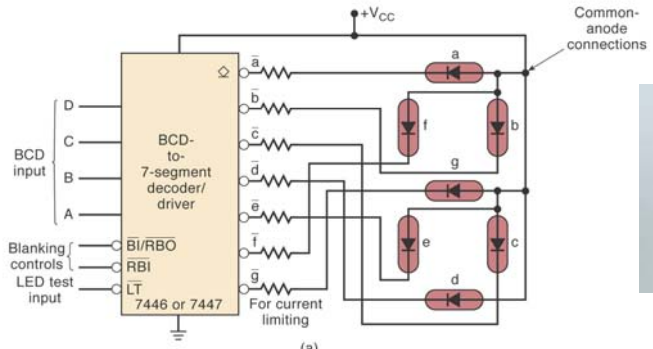
# Giải mã BCD – LED 7 đoạn

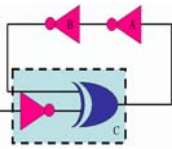


- LED 7 đoạn được chế tạo từ 7 LED thông thường



# IC hiển thị LED 7 đoạn

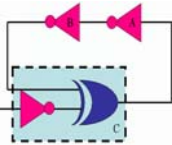




## LED và LCD

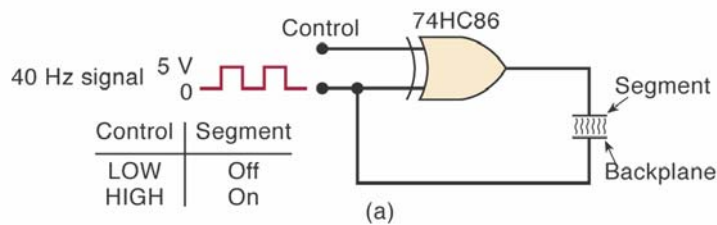
- Đèn LED (Light Emitting Diode) phát sáng khi có dòng điện chạy qua nó
- LCD (Liquid Crystal Display) hiển thị tinh thể lỏng
- LCD hoạt động với tín hiệu xoay chiều điện áp thấp, tần số thấp
- Đèn LED tạo ra ánh sáng mạnh hơn, LCD sử dụng ít công suất hơn

13

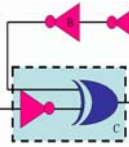


## Nguyên lý hoạt động của LCD

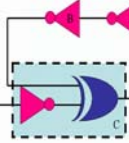
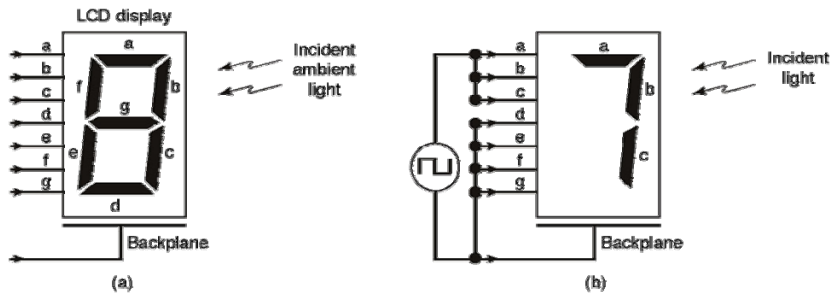
- Control = low, ngõ ra của EX-OR sẽ giống với ngõ vào. Điện áp trên LCD = 0, LCD = off
- Control = high, ngõ ra của EX-OR sẽ ngược với sóng ngõ vào. Điện áp trên LCD là sóng vuông 5 và -5V, LCD ở trạng thái on



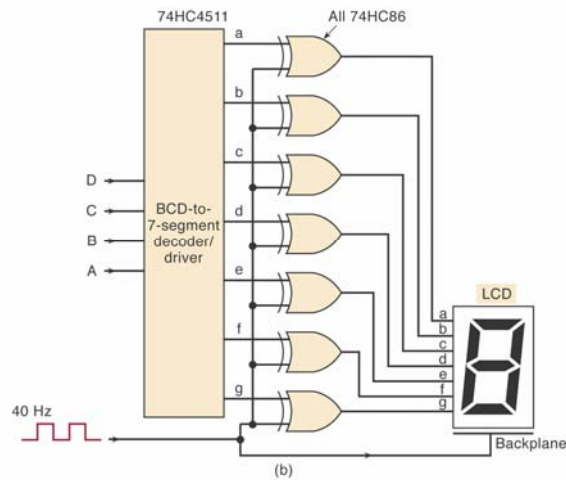
14

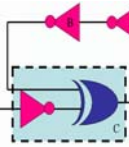


# LCD 7 đoạn



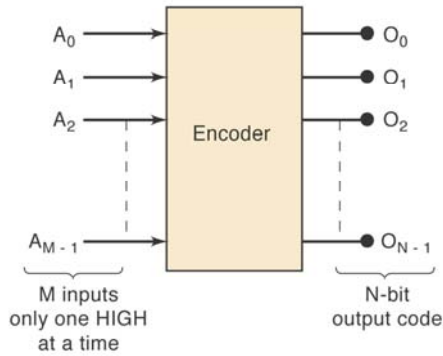
# Nguyên lý hoạt động của LCD 7 đoạn



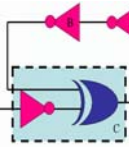


# Mạch mã hóa

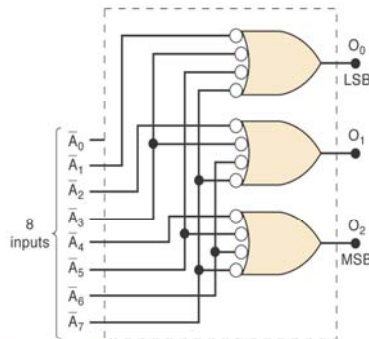
- Hoạt động ngược lại với mạch giải mã
- Mạch mã hóa có một số ngõ vào nhưng vào một thời điểm chỉ có một ngõ vào ở trạng thái tích cực



17

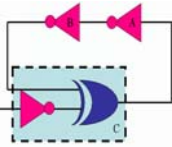


# Mạch mã hóa octal-binary



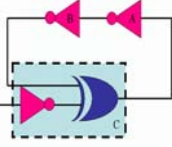
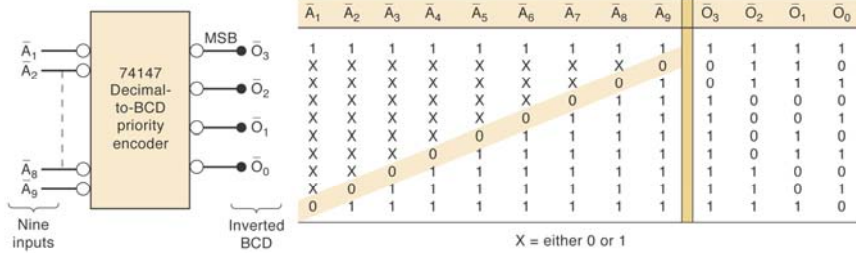
	Inputs								Outputs		
	$\bar{A}_0$	$\bar{A}_1$	$\bar{A}_2$	$\bar{A}_3$	$\bar{A}_4$	$\bar{A}_5$	$\bar{A}_6$	$\bar{A}_7$	$O_2$	$O_1$	$O_0$
X	1	1	1	1	1	1	1	1	0	0	0
X	0	1	1	1	1	1	1	1	0	0	1
X	1	0	1	1	1	1	1	1	0	1	0
X	1	1	0	1	1	1	1	1	0	1	1
X	1	1	1	0	1	1	1	1	1	0	0
X	1	1	1	1	0	1	1	1	1	0	1
X	1	1	1	1	1	0	1	1	1	1	0
X	1	1	1	1	1	1	0	1	1	1	1

\*Only one LOW input at a time

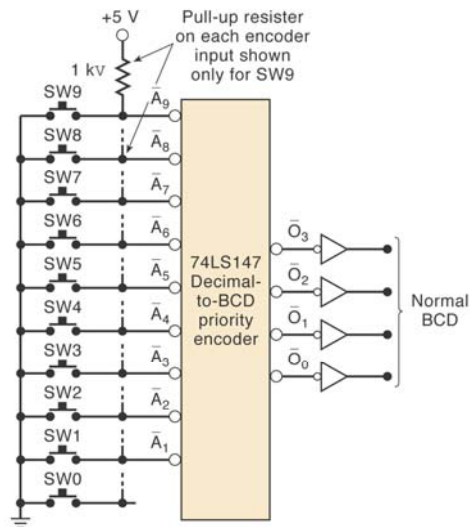


# Ưu tiên trong mã hóa

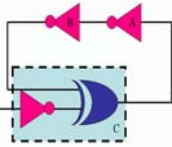
- Trong trường hợp có nhiều ngõ vào ở trạng thái tích cực thì ngõ ra sẽ tương ứng với ngõ vào có trọng số cao nhất



# Ví dụ mạch mã hóa



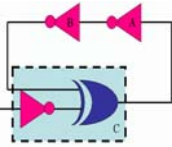




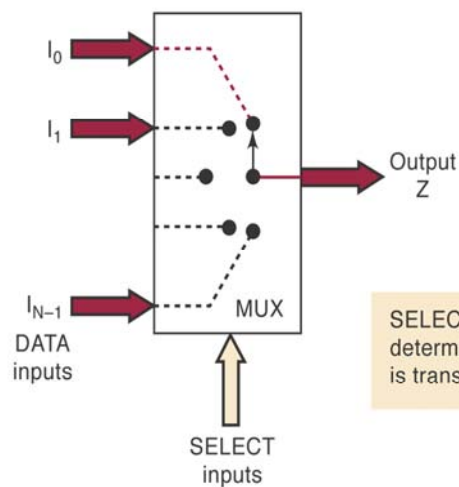
## Ví dụ mạch mã hóa

- Sử dụng IC 74LS147
- Các công tắc tương ứng với các nút nhất từ 0 đến 9
- Bình thường tất cả các công tắc mở, các ngõ vào ở trạng thái cao, BCD ngõ ra là 0000
- Khi có một phím nhấn, mạch sẽ tạo ra một mã BCD tương ứng

21

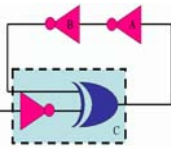


## Mạch ghép kênh



SELECT input code determines which input is transmitted to output Z.

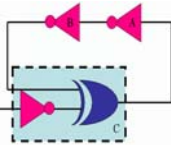
22



## Mạch ghép kênh

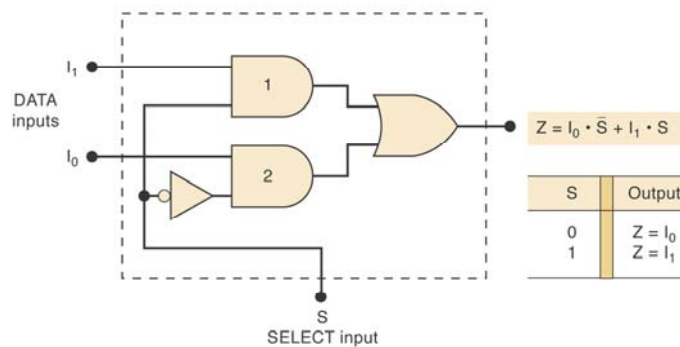
- Mạch ghép kênh còn được gọi là mạch chọn dữ liệu.
- Mạch có nhiều ngõ vào.
- Tại một thời điểm chỉ có một ngõ vào được đưa đến ngõ ra.
- Các đường select quyết định ngõ vào nào được chọn.

23



## Mạch ghép kênh 2 ngõ vào

- Mạch ghép kênh 2 ngõ vào  $Z = I_0S' + I_1S$



24

### Mạch ghép kênh 4 ngõ vào

S <sub>1</sub>	S <sub>0</sub>	Output
0	0	Z = I <sub>0</sub>
0	1	Z = I <sub>1</sub>
1	0	Z = I <sub>2</sub>
1	1	Z = I <sub>3</sub>

25

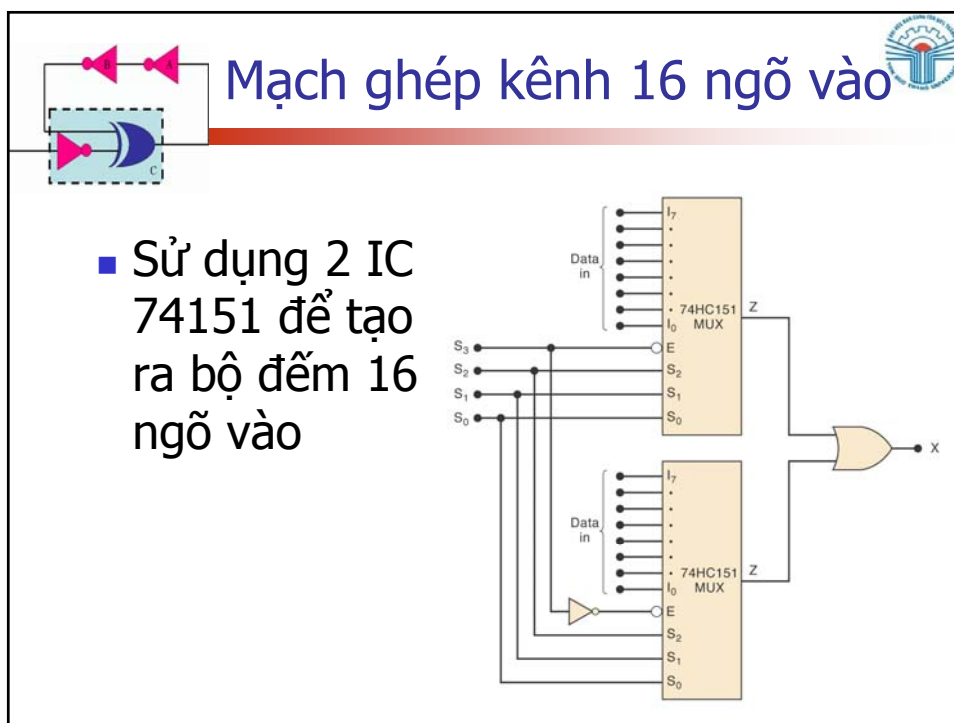
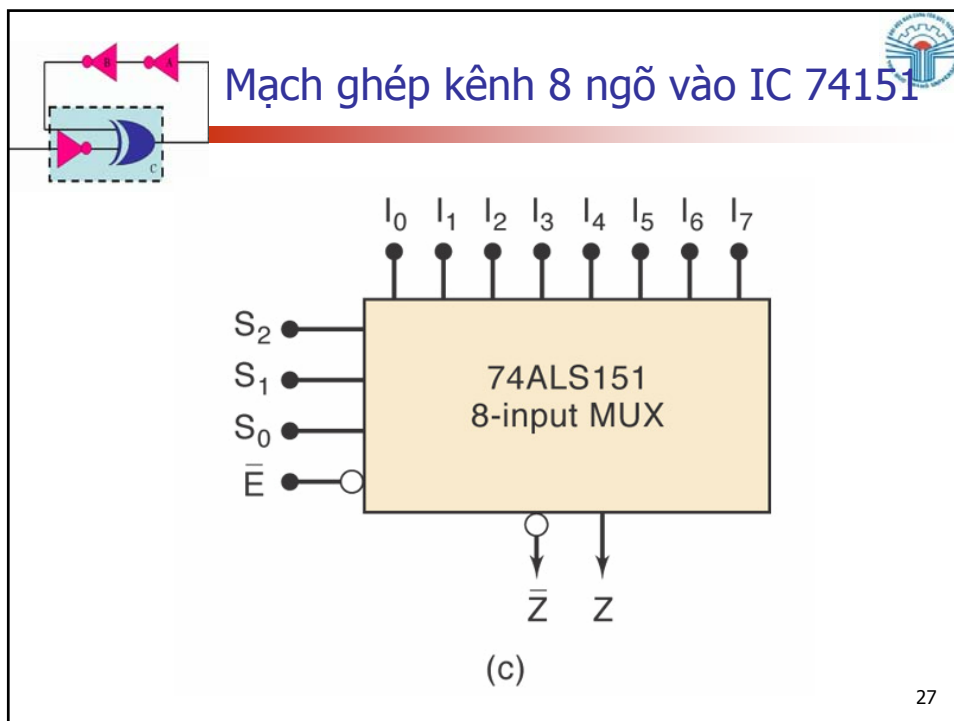
### Mạch ghép kênh 8 ngõ vào IC 74151

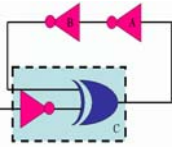
Inputs				Outputs	
$\bar{E}$	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	$\bar{Z}$	Z
H	X	X	X	H	L
L	L	L	L	I <sub>0</sub>	I <sub>0</sub>
L	L	L	H	I <sub>1</sub>	I <sub>1</sub>
L	L	H	L	I <sub>2</sub>	I <sub>2</sub>
L	L	H	H	I <sub>3</sub>	I <sub>3</sub>
L	H	L	L	I <sub>4</sub>	I <sub>4</sub>
L	H	L	H	I <sub>5</sub>	I <sub>5</sub>
L	H	H	L	I <sub>6</sub>	I <sub>6</sub>
L	H	H	H	I <sub>7</sub>	I <sub>7</sub>

(b)

(a)

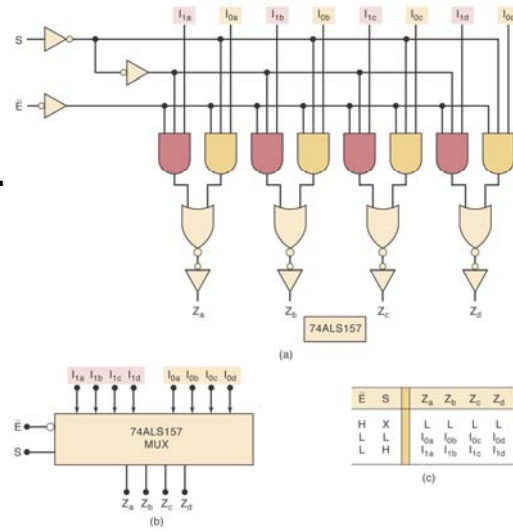
26



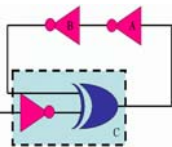


## Mạch ghép kênh 2 ngõ vào 4 bit

- Mạch có 2 nhóm ngõ vào, mỗi ngõ vào có 4 bit.
- Mạch có 1 ngõ select để chọn 1 trong 2 nhóm ngõ vào

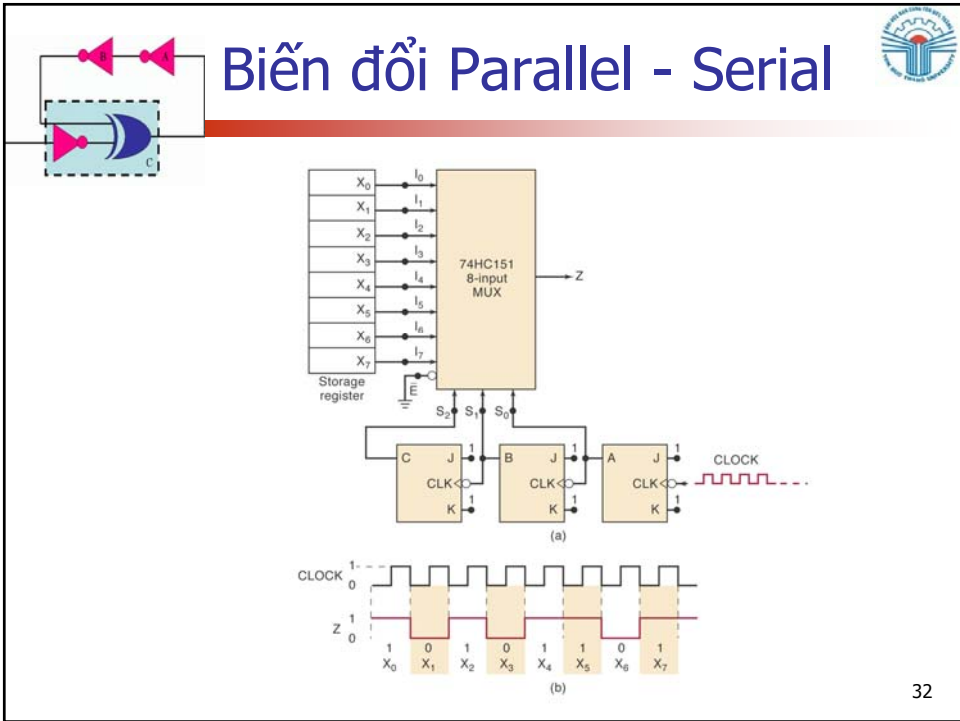
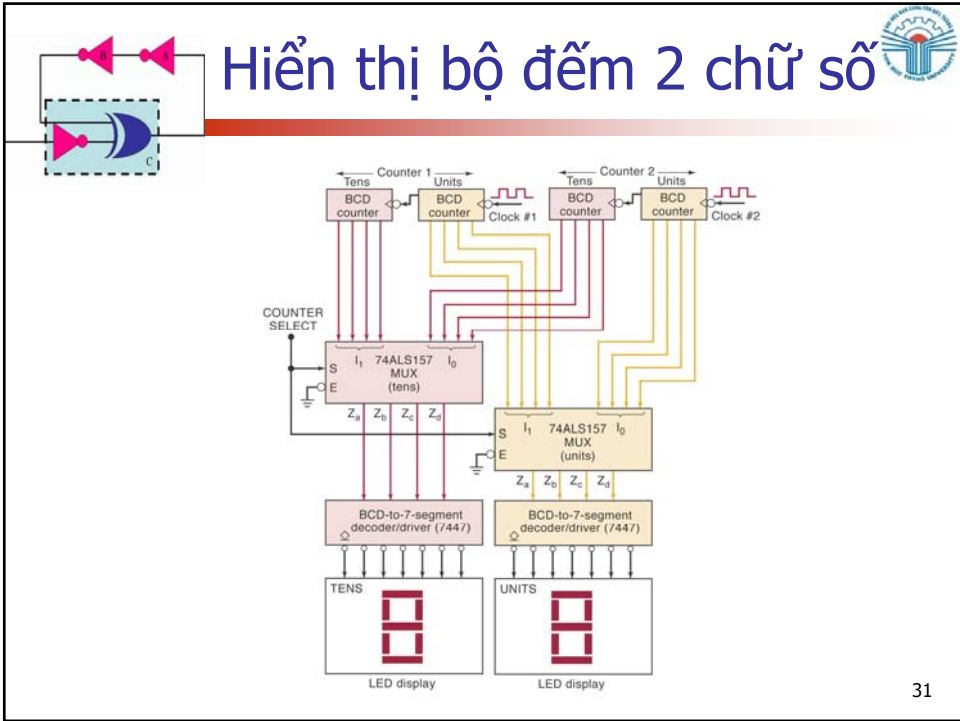


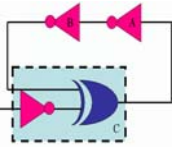
29



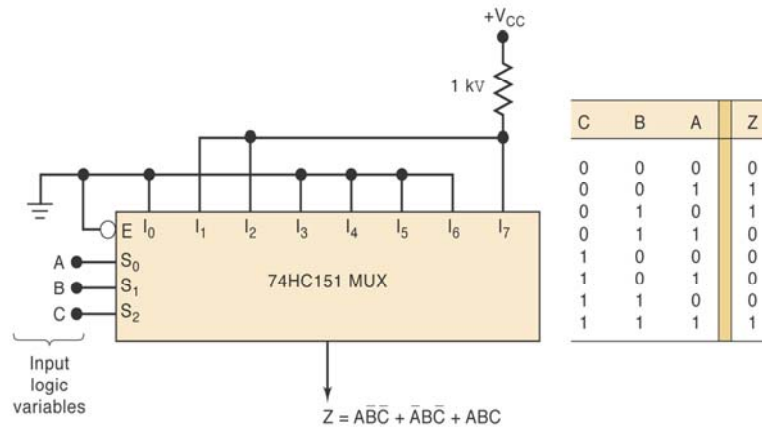
## Ứng dụng mạch ghép kênh

30



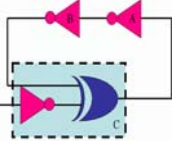


# Tạo hàm logic



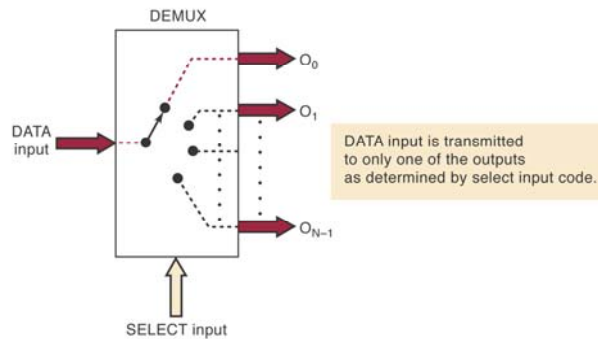
(a)

(b)



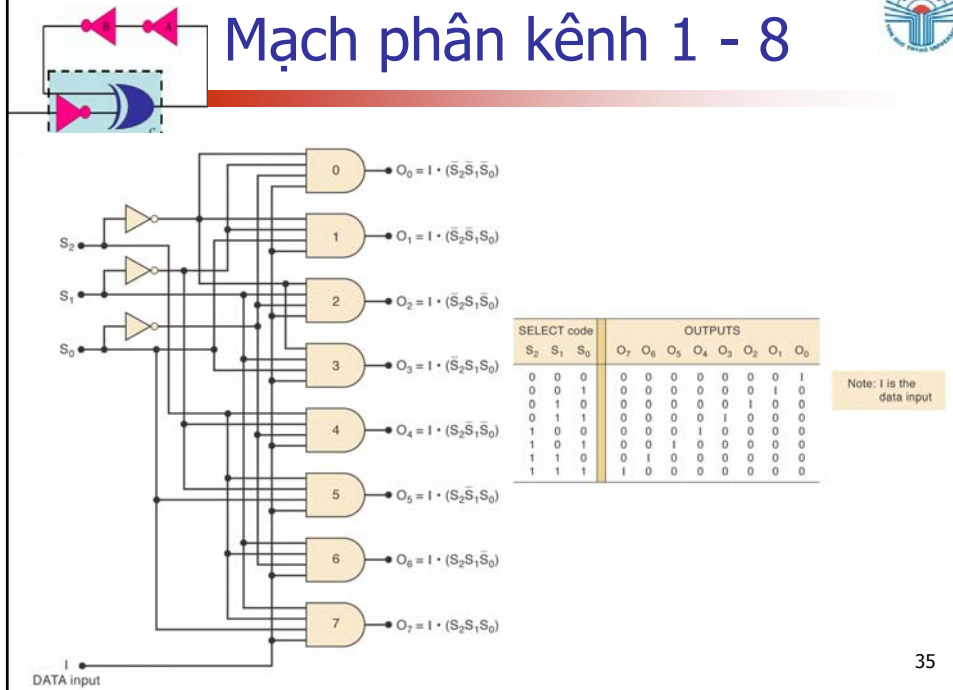
# Mạch phân kênh

- Mạch phân kênh (DEMUX) có một ngõ vào và ngõ vào này sẽ được phân đến một trong nhiều ngõ ra





# Mạch phân kênh 1 - 8

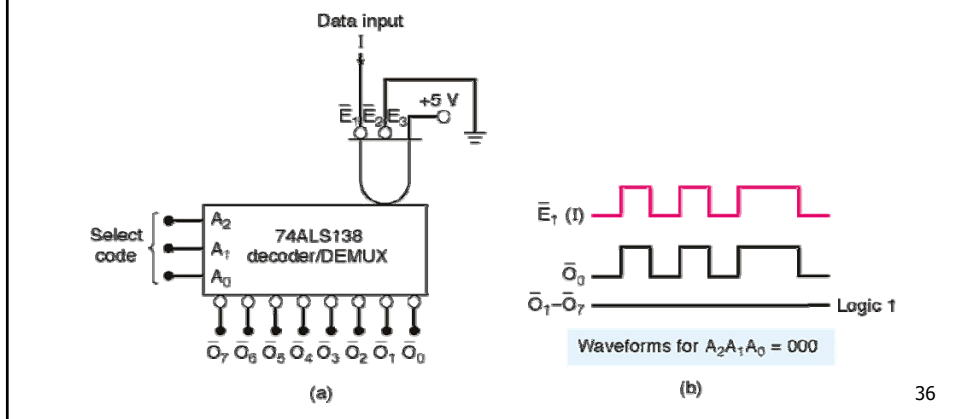


35



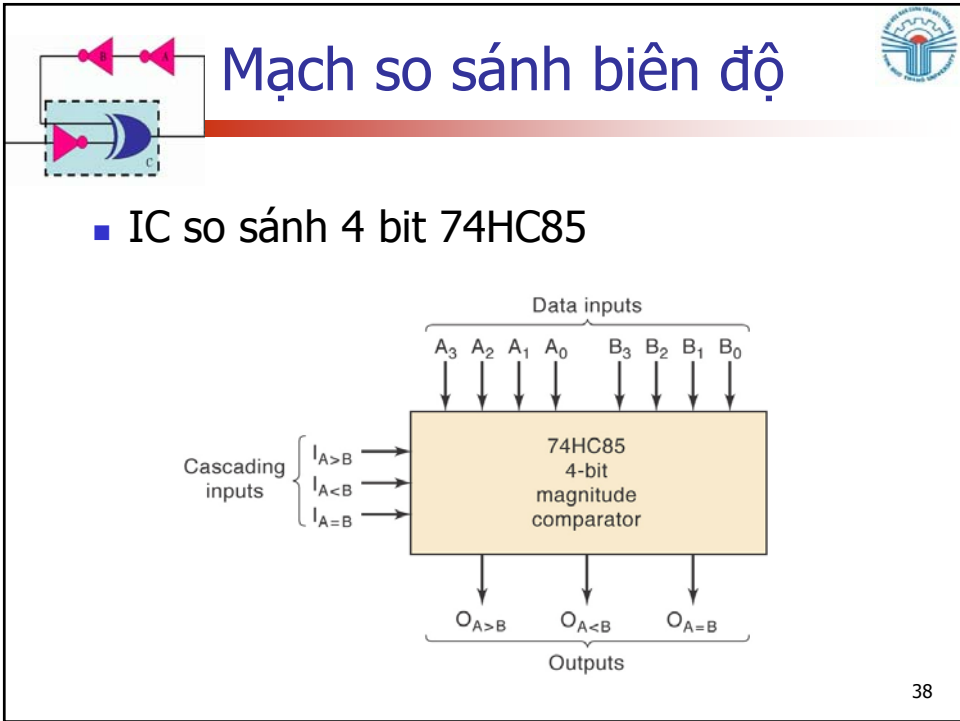
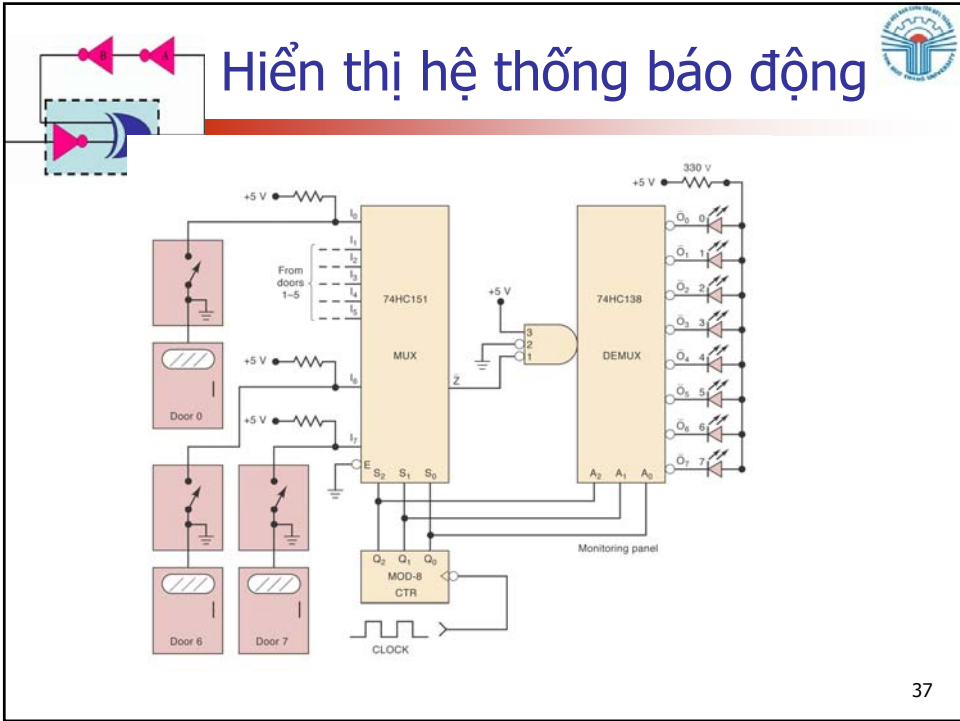
# Mạch phân kênh

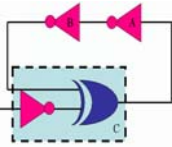
- IC giải mã 74LS138 có thể được sử dụng để làm bộ phân kênh với ngõ vào E<sub>1</sub> làm ngõ vào data



36



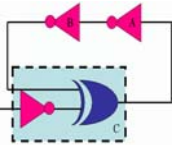




## IC 74HC85

- So sánh nhóm bit A và nhóm bit B
- IC có 3 ngõ ra tương ứng với  $A > B$ ,  $A < B$ ,  $A = B$ .
- Ngõ vào tầng được sử dụng trong trường hợp dùng nhiều IC 74HC85 để là bộ so sánh nhiều hơn 4 bit.
- Trong trường hợp so sánh 4 bit,  $I_{A < B}$ ,  $I_{A > B}$  được nối đất,  $I_{A = B}$  nối nguồn +5V

39



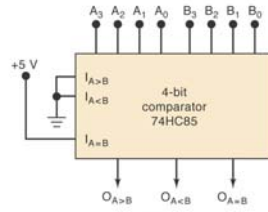
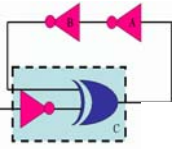
## Bảng chân trị IC 74HC85

COMPARING INPUTS				CASCADING INPUTS			OUTPUTS		
$A_3, B_3$	$A_2, B_2$	$A_1, B_1$	$A_0, B_0$	$I_{A > B}$	$I_{A < B}$	$I_{A = B}$	$O_{A > B}$	$O_{A < B}$	$O_{A = B}$
$A_3 > B_3$	X	X	X	X	X	X	H	L	L
$A_3 < B_3$	X	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 > B_2$	X	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 < B_2$	X	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 > B_1$	X	X	X	X	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 < B_1$	X	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	X	X	X	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	X	X	X	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	H	L	L	H	L	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	H	L	L	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	X	X	H	L	L	H
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	L	L	H	H	L
$A_3 = B_3$	$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	H	H	L	L	L	L

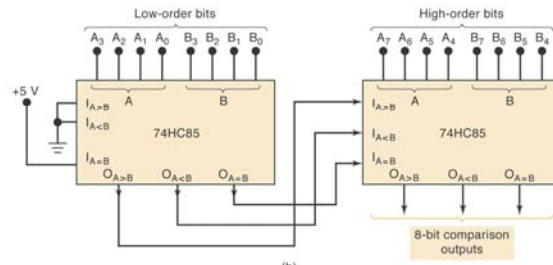
40



# So sánh nhiều hơn 4 bit



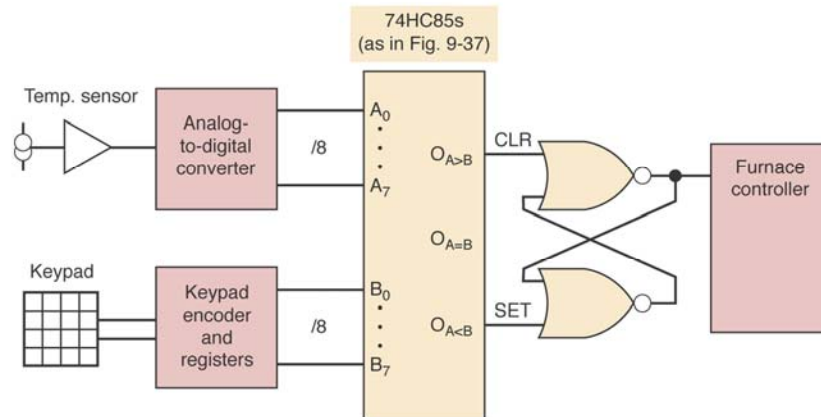
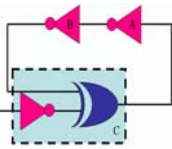
(a)

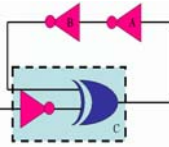


(b)



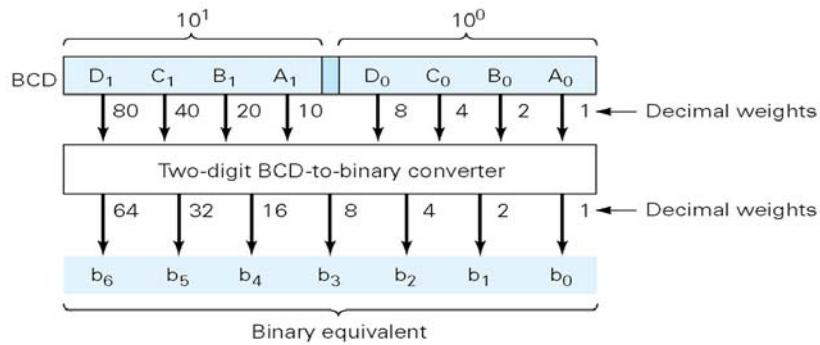
# Ứng dụng điều khiển nhiệt độ



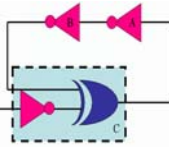


# Chuyển mã

- Mạch chuyển mã có chức năng biết đổi dữ liệu thành ra mã nhị phân hay ngược lại

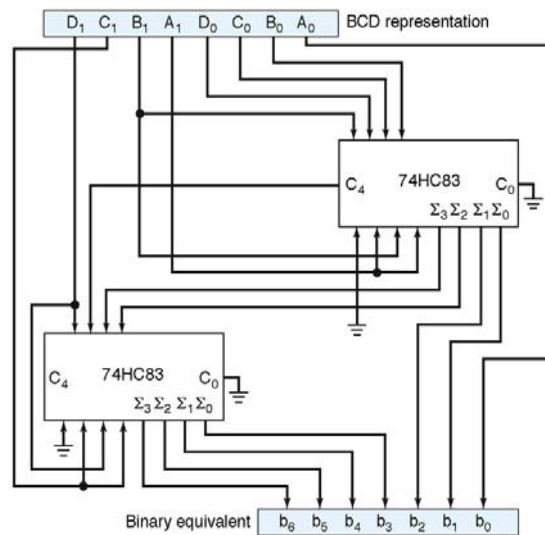


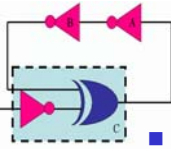
Biến đổi 2 số BCD sang nhị phân



# Chuyển mã

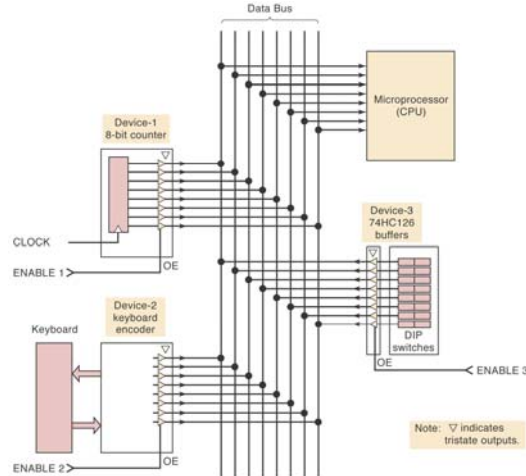
Sử dụng bộ cộng song song 4 bit 74HC83 để thực hiện bộ biến đổi BCD sang nhị phân



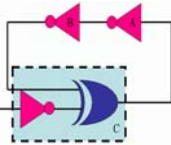


## Data Bus

- 3 thiết bị có thể chung một đường truyền để truyền tín hiệu đến CPU

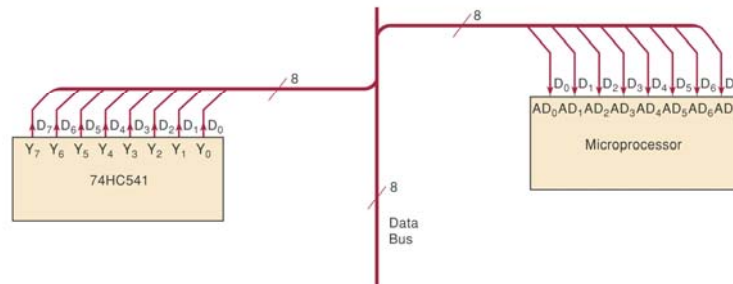


45

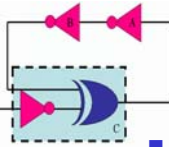


## Data Bus

- Phương pháp miêu tả kết nối data bus, "/8" ký hiệu data bus có 8 đường

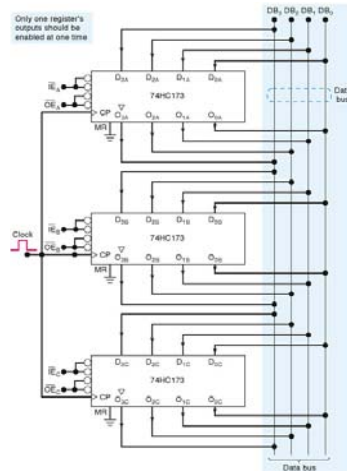


46

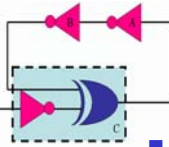


# Data Bus

- Thanh ghi 3 trạng thái được sử dụng để kết nối với data bus

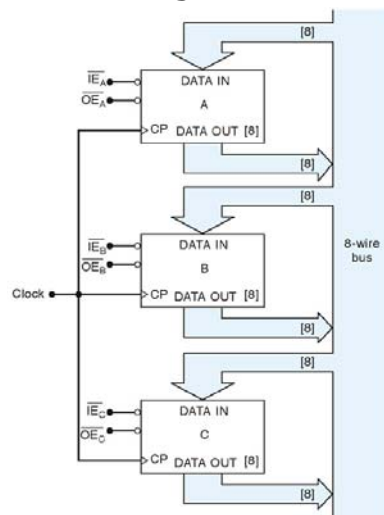


47

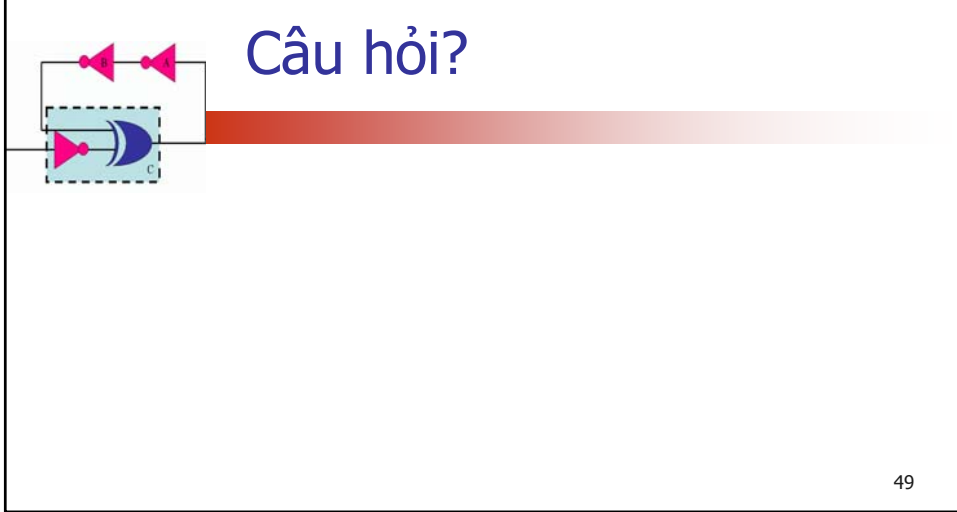


# Data Bus

- Miêu tả đơn giản tổ chức của BUS



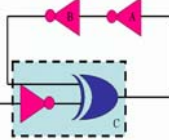
48





# Chương 10

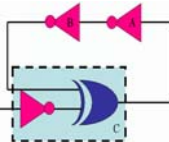
## Kết nối với mạch tương tự



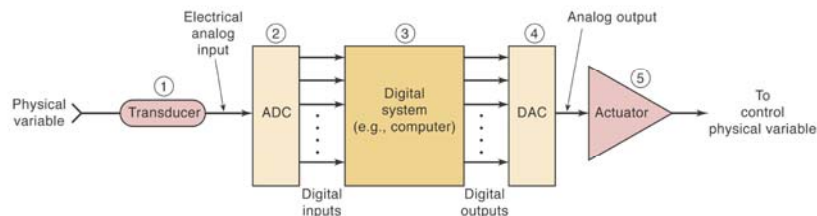
Th.S Đặng Ngọc Khoa  
Khoa Điện - Điện Tử



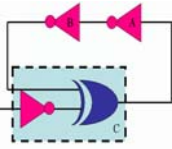
## Kết nối với mạch tương tự



- Transducer: biến đổi đại lượng vật lý thành tín hiệu điện
- Analog-to-digital converter (ADC)
- Digital system: xử lý tín hiệu
- Digital-to-analog converter (DAC)
- Thực thi kết quả

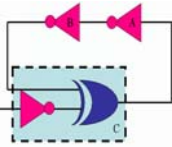






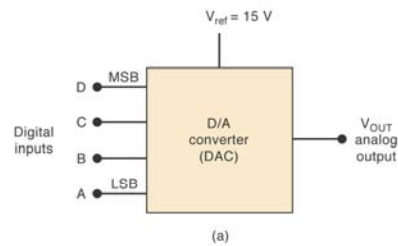
# Biến đổi D/A

- Nhiều phương pháp ADC sử dụng DAC
- $V_{ref}$  được sử dụng để xác định ngõ ra full-scale.
- Trong trường hợp tổng quát, ngõ ra analog =  $K \times$  giá trị số ngõ vào



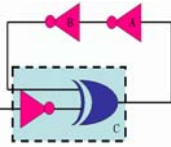
# Biến đổi D/A

- DAC 4 bit, ngõ ra điện áp tương tự



D	C	B	A	$V_{OUT}$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

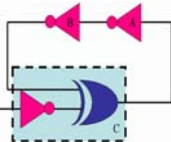
(b)



## Ngõ ra tương tự

- Ngõ ra của bộ biến đổi DAC không hoàn toàn là tín hiệu analog bởi vì nó chỉ xác định ở một số giá trị nhất định.
- Với mạch trên, ngõ ra chỉ có thể có những giá trị, 0, 1, 2, ..., 15 volt.
- Khi số ngõ vào tăng lên thì tín hiệu ngõ ra càng giống với tín hiệu tương tự.

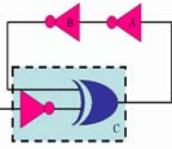
5



## Bước nhảy

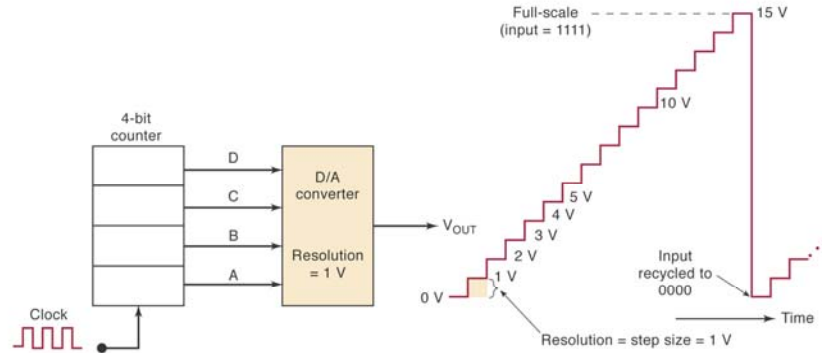
- Bước nhảy của bộ biến đổi D/A được định nghĩa là khoảng thay đổi nhỏ nhất của ngõ ra khi có sự thay đổi giá trị ngõ vào.
- Bộ biến đổi D/A N bit: số mức ngõ ra khác nhau =  $2^N$ , số bước nhảy =  $2^N - 1$
- Bước nhảy =  $K = V_{ref} / (2^N - 1)$

6

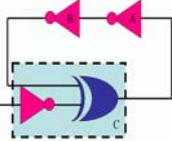


## Bước nhảy

- Bước nhảy = 1 volt

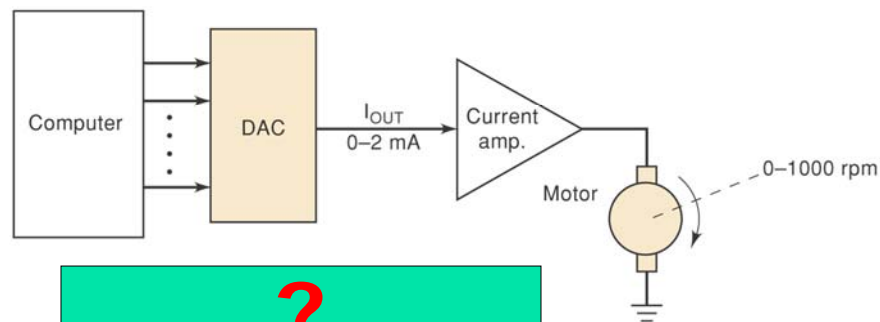


7

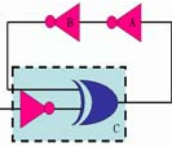


## Ví dụ 10-1

- Cần sử dụng bộ DAC bao nhiêu bit để có thể điều khiển motor thay đổi tốc độ mỗi 2 vòng.

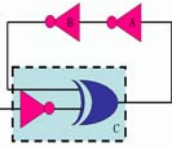
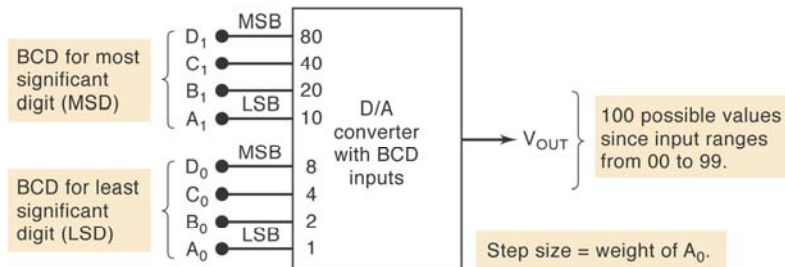


8



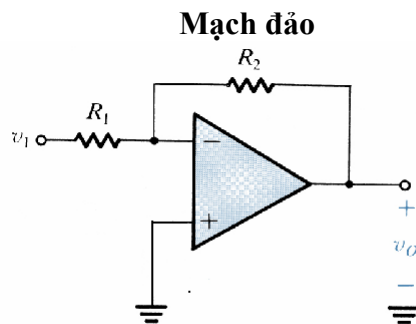
# Ngõ vào BCD

- Trọng số của những ngõ vào khác nhau
- Ngõ vào 2 số BCD

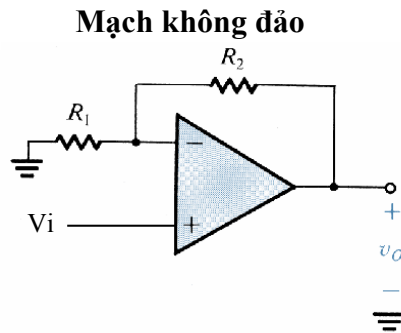


# Mạch biến đổi D/A

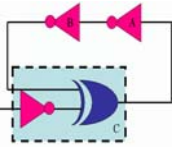
- Tính chất của Opamp



$$\begin{aligned} V_o/V_i &= -R_2/R_1 \\ R_{in} &= R_1 \end{aligned}$$

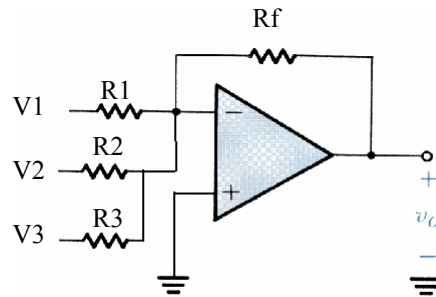


$$\begin{aligned} V_o/V_i &= 1+R_2/R_1 \\ R_{in} &= \text{infinity} \end{aligned}$$

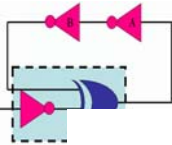


# Mạch biến đổi D/A

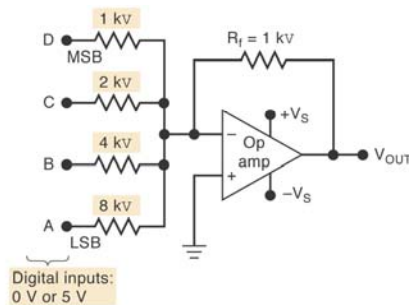
- Trọng số của những ngõ vào khác nhau



$$V_o = -R_f(V_1/R_1 + V_2/R_2 + V_3/R_3)$$



# Mạch biến đổi D/A



Digital inputs:  
0 V or 5 V

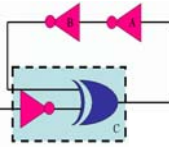
(a)

Input code				V <sub>OUT</sub> (volts)
D	C	B	A	
0	0	0	0	0
0	0	0	1	-0.625 ← LSB
0	0	1	0	-1.250
0	0	1	1	-1.875
0	1	0	0	-2.500
0	1	0	1	-3.125
0	1	1	0	-3.750
0	1	1	1	-4.375
1	0	0	0	-5.000
1	0	0	1	-5.625
1	0	1	0	-6.250
1	0	1	1	-6.875
1	1	0	0	-7.500
1	1	0	1	-8.125
1	1	1	0	-8.750
1	1	1	1	-9.375 ← Full-scale

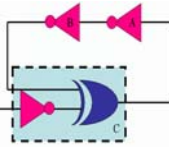
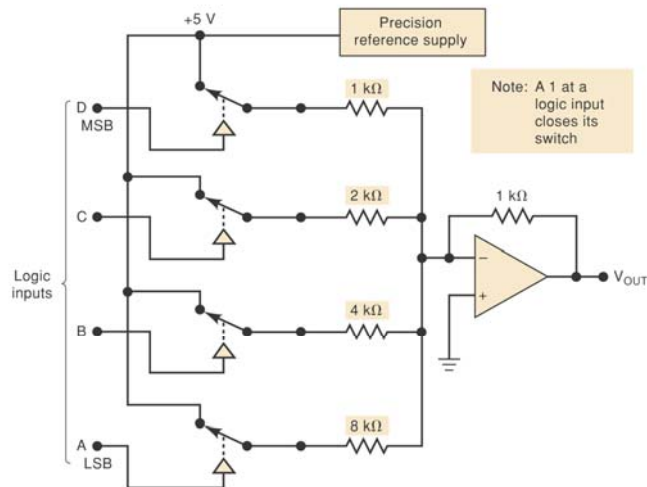
(b)

Bước nhảy =  $|5V(1K/8K)| = .625V$

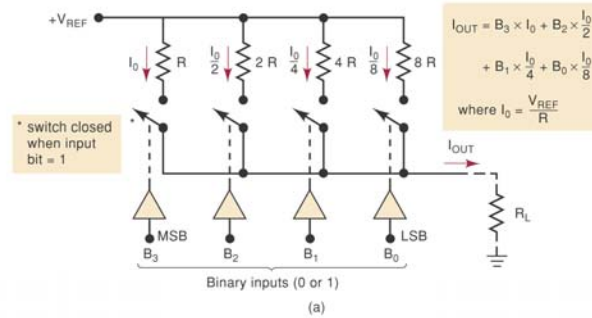
Max out =  $5V(1K/8K + 1K/4K + 1K/2K + 1K/1K) = -9.375V$



# BỘ DAC 4 bit

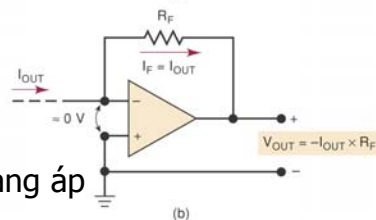


# DAC với ngõ ra dòng điện

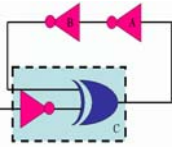


$$I_{OUT} = B_3 \times I_0 + B_2 \times \frac{I_0}{2} + B_1 \times \frac{I_0}{4} + B_0 \times \frac{I_0}{8}$$

where  $I_0 = \frac{V_{REF}}{R}$



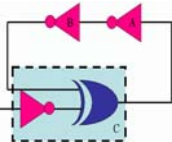
Biến đổi dòng sang áp



## Mạch biến đổi D/A

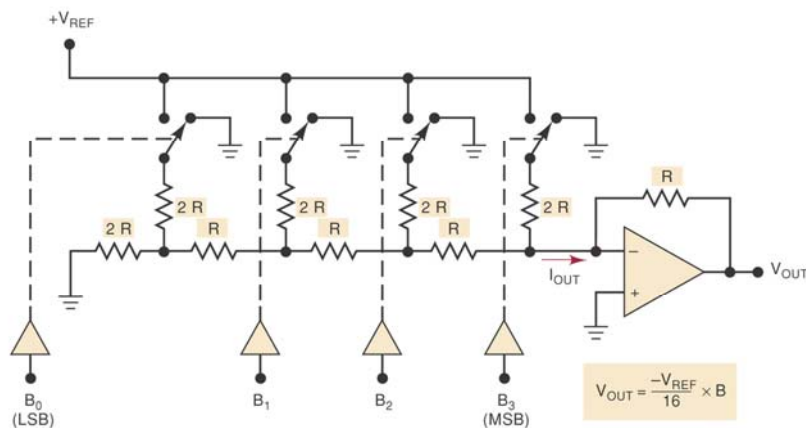
- Với những mạch biến đổi D/A ở trên, trọng số các bit được xác định dựa vào giá trị của các điện trở.
- Trong một mạch phải sử dụng nhiều điện trở với những giá trị khác nhau
  - Bộ DAC 12 bit
    - Điện trở MSB = 1K
    - Điện trở LSB =  $1 \times 2^{12} = 2M$
- Mạch sau chỉ sử dụng 2 giá trị điện trở

15

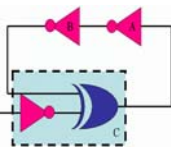


## Mạch biến đổi D/A

### ■ DAC R/2R



16

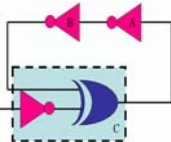


## DAC – Thông số kỹ thuật



- Nhiều bộ DAC được tích hợp vào trong những IC, một số thông số tiêu biểu của nó
  - Resolution: bước nhảy của bộ DAC
  - Accuracy: sai số sai số của bộ DAC
  - Offset error: ngõ ra của DAC khi tất cả ngõ vào bằng 0
  - Settling time: thời gian yêu cầu để DAC thực hiện biến đổi khi ngõ vào chuyển đổi từ trạng thái all 0 đến trạng thái all 1

17



## IC DAC



- AD7524 (Figure 11-9)
  - CMOS IC
  - 8 bit D/A
  - Sử dụng R/2R
  - Max settling time: 100 ns
  - Full range accuracy: +/- 0.2% F.S.

18



## IC DAC

(a) (b) (c)

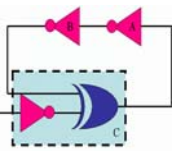
- Khi ngõ vào CS và WR ở mức thấp, OUT1 là ngõ ra analog.
- Khi cả 2 ở mức cao, OUT1 được chốt và giá trị nhị phân ngõ vào không được biến đổi ở ngõ ra.
- OUT2 thông thường được nối đất

19

## Ứng dụng DAC

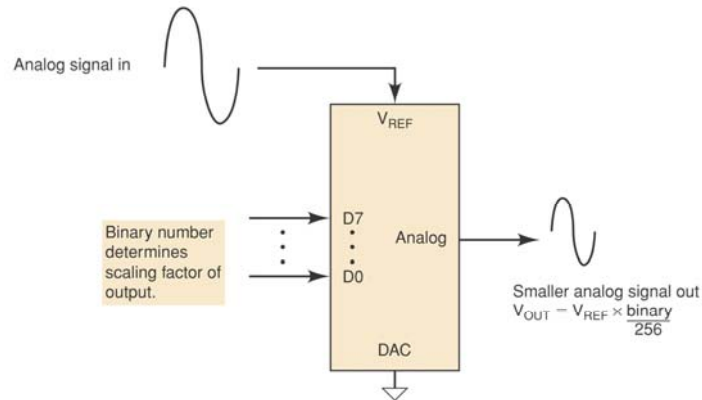
- Control
  - Sử dụng ngõ ra số của máy tính để điều chỉnh tốc độ của motor hay nhiệt độ.
- Automatic testing
  - Tạo tín hiệu từ máy tính để kiểm tra mạch analog
- Signal reconstruction
  - Tái tạo tín hiệu analog từ tín hiệu số. Ví dụ hệ thống audio CD
- A/D conversion

20

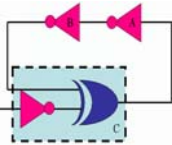


## Ví dụ 10-2

- Sử dụng DAC để điều chỉnh biên độ của tín hiệu analog



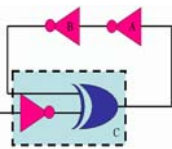
21



## Biến đổi A/D

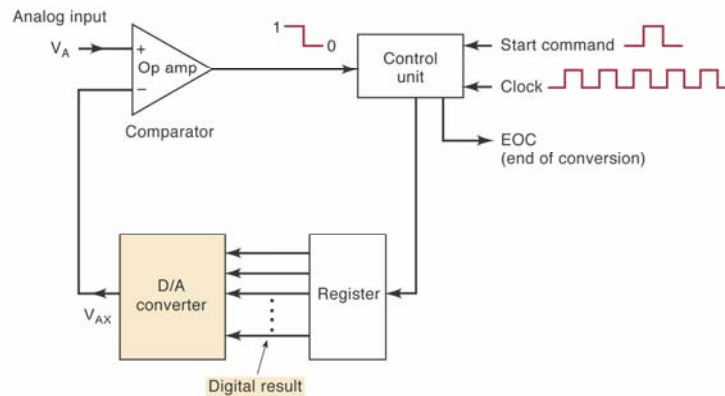
- ADC – miêu tả giá trị analog ngõ vào bằng giá trị số nhị phân.
- ADC phức tạp và tốn nhiều thời gian biến đổi hơn DAC
- Một số ADC sử dụng bộ DAC là một phần của nó
- Một opamp được sử dụng làm bộ so sánh trong ADC

22

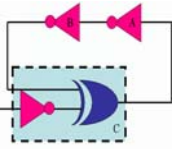


## Biến đổi A/D

- Bộ đếm nhị phân được sử dụng như là một thanh ghi và cho phép xung clock tăng giá trị bộ đếm cho đến khi  $V_{AX} \geq V_A$



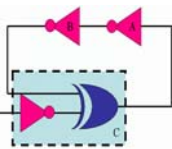
23



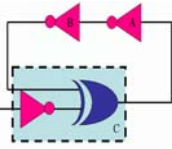
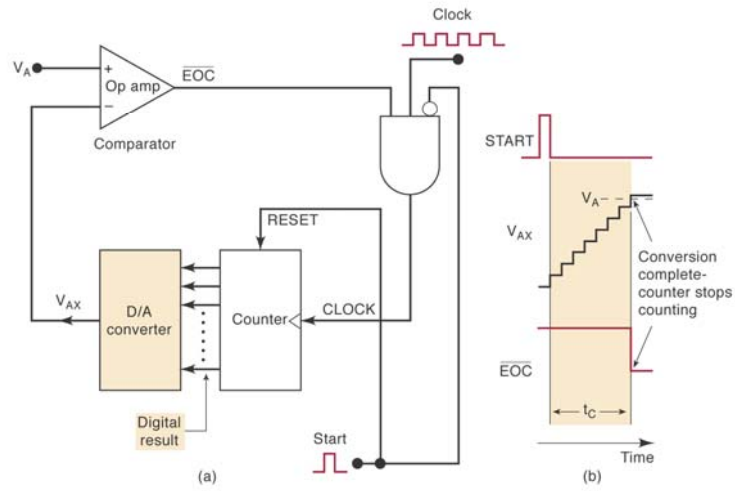
## Hoạt động của bộ ADC

- Lệnh START bắt đầu quá trình biến đổi
- Control unit thay đổi giá trị nhị phân trong thanh ghi
- Giá trị nhị phân trong thanh ghi được biến đổi thành giá trị nhị phân  $V_{AX}$
- Bộ so sánh so sánh  $V_{AX}$  với  $V_A$ . Khi  $V_{AX} < V_A$ , ngõ ra bộ so sánh ở mức cao. When  $V_{AX} > V_A$ , ngõ ra có mức thấp, quá trình biến đổi kết thúc, giá trị nhị phân nằm trong thanh ghi.
- Bộ phận điều khiển sẽ phát ra tín hiệu end-of-conversion signal, EOC.

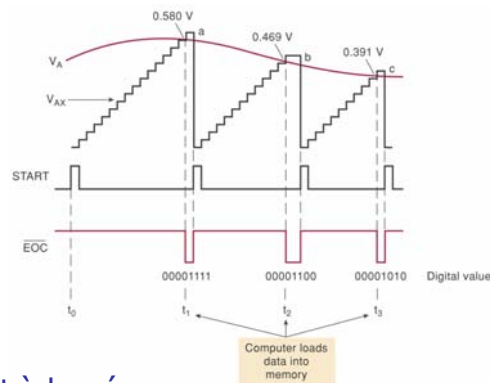
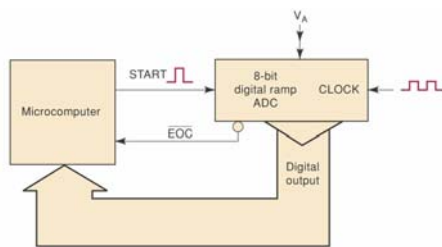
24



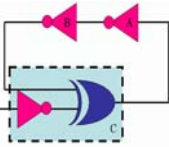
# Biến đổi A/D



# Biến đổi A/D

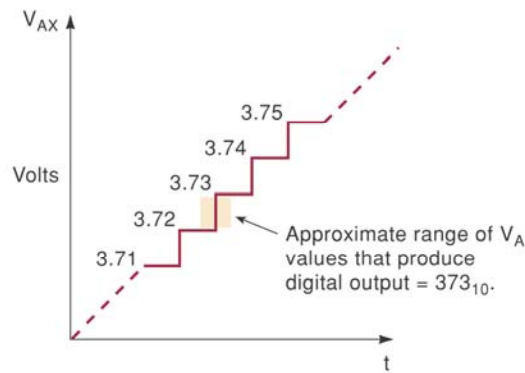


Dạng sóng thể hiện quá trình máy tính thiết lập một chu trình biến đổi là lưu giá trị nhị phân vào bộ nhớ.

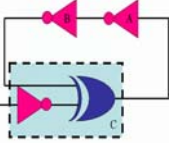


## Sai số lượng tử

- Có thể giảm sai số lượng tử bằng cách tăng số bit nhưng không thể loại bỏ hoàn toàn

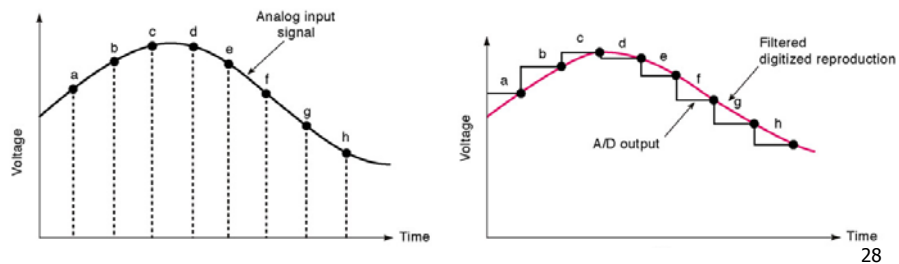


27

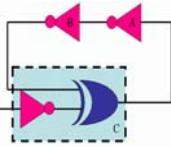


## Khôi phục tín hiệu

- Sau khi kết thúc một quá trình ADC ta sẽ có giá trị nhị phân của một mẫu.
- Quá trình khôi phục tín hiệu analog như sau



28

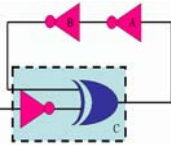


## Khôi phục tín hiệu

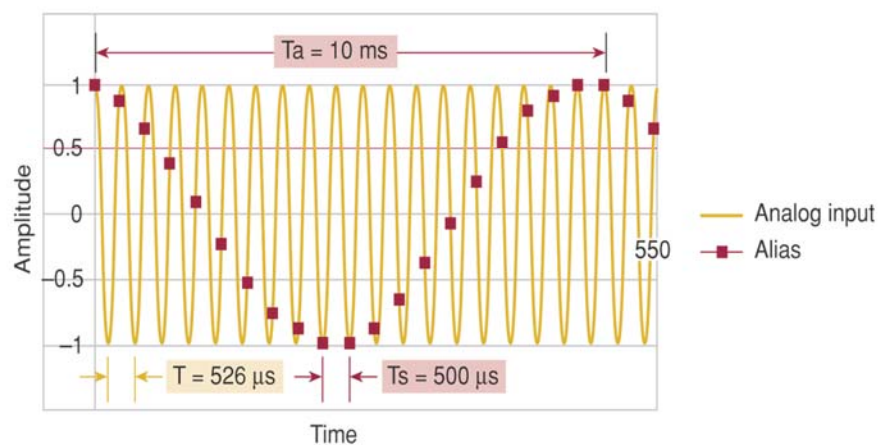
### ■ Aliasing

- Nguyên nhân là do tần số lấy mẫu không đúng
- Giới hạn Nyquist
  - Tần số lấy mẫu phải ít nhất lớn hơn 2 lần tần số cao nhất của tín hiệu ngõ vào.
  - Lấy mẫu ở tần số nhỏ hơn 2 lần tần số ngõ vào sẽ tạo nên kết quả sai khi khôi phục tín hiệu.

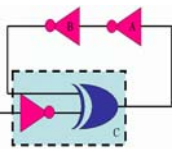
29



## Quá trình lấy mẫu không đúng



30

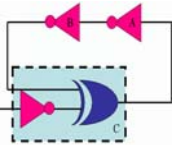


## ADC xấp xỉ liên tục (SDC)

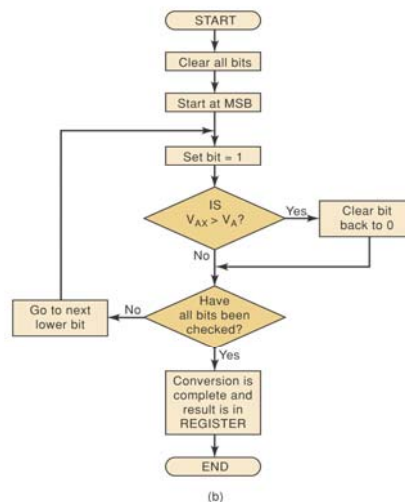
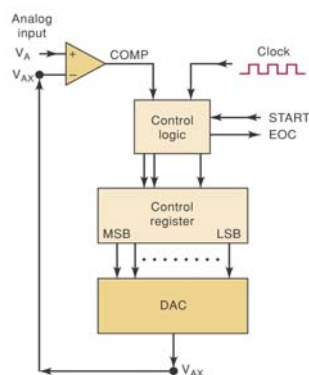


- Sử dụng rộng rãi hơn ADC
- Phức tạp hơn nhưng có thời gian biết đổi ngắn hơn
- Thời gian biến đổi cố định, không phụ thuộc vào giá trị analog ngõ vào
- Nhiều SAC được tích hợp trong những IC

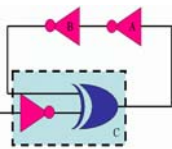
31



## Successive-approximation ADC

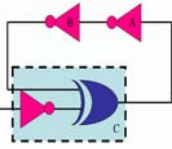
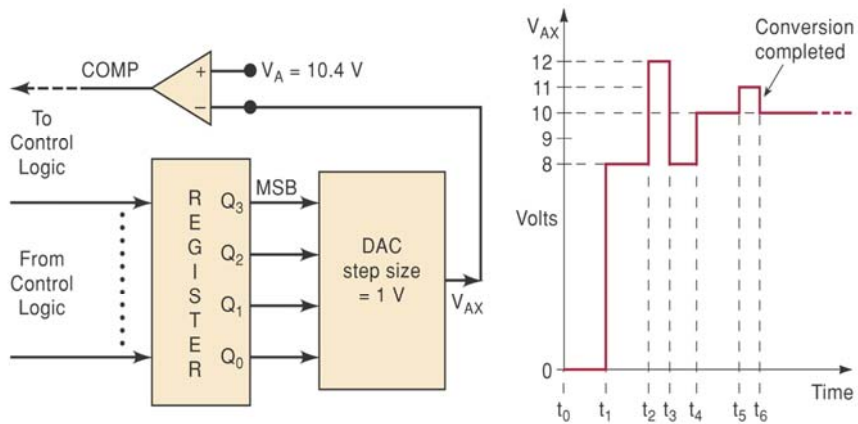


32

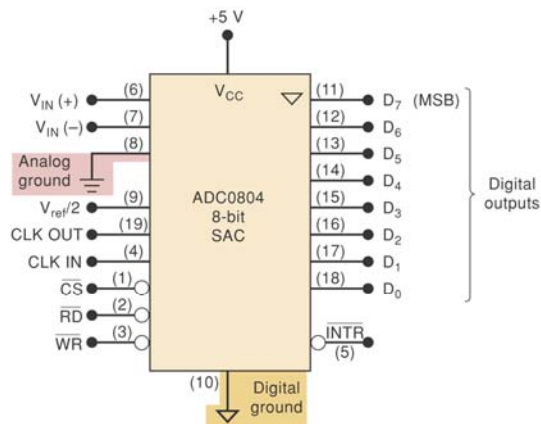


# Successive-approximation ADC

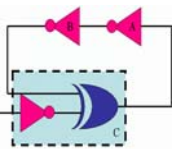
- SAC 4 bit sử dụng DAC có bước nhảy 1 V



# ADC0804 – SAC 8 bit





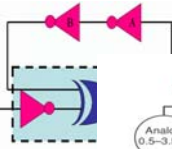


## ADC0804 – SAC 8 bit

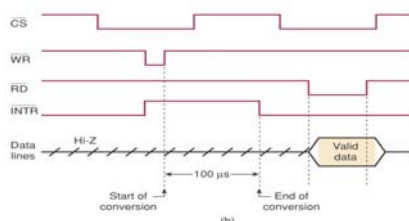
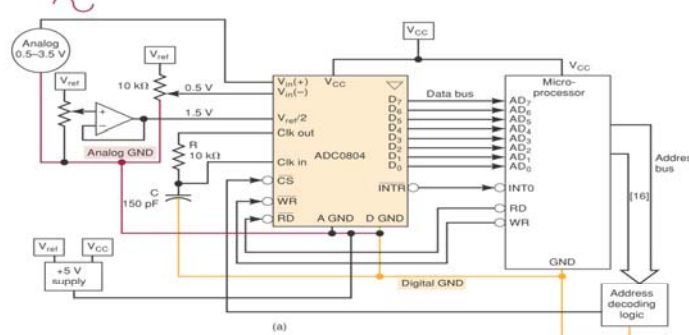


- Có hai ngõ vào analog cho phép hai ngõ vào vi sai.
- Ngưỡng xác định tại  $\pm 1/2\text{LSB}$ . Ví dụ, bước nhảy là 10mV, bit LSB sẽ ở trạng thái 1 tại 5mV.
- IC có thanh ghi xung clock bên trong tạo ra tần số  $f = 1/(1.1RC)$ . Hoặc có thể sử dụng xung clock bên ngoài.
- Nếu sử dụng xung clock có tần số 606kHz, thời gian biến đổi xấp xỉ 100us.
- Sử dụng nối đất riêng bởi vì đất của thiết bị số tồn tại tại nhiều do quá trình thay đổi dòng đột ngột khi thay đổi trạng thái.

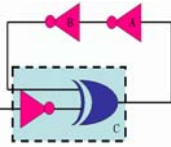
35



## Ứng dụng của IC ADC0804



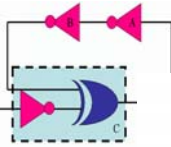
36



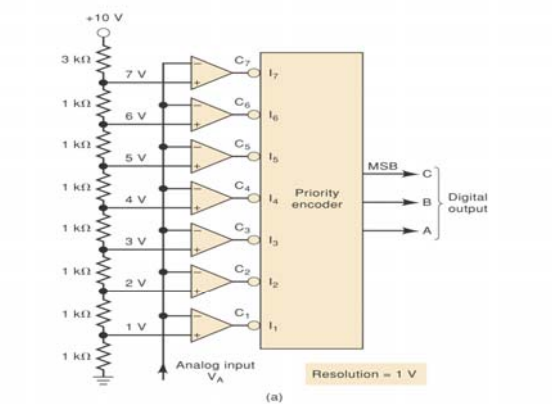
# Flash ADC

- Tốc độ biến đổi cao
- Mạch phức tạp hơn nhiều
  - Flash ADC 6 bit yêu cầu 63 bộ so sánh tương tự
  - Flash ADC 8 bit yêu cầu 255 bộ so sánh tương tự
  - Flash ADC 10 bit yêu cầu 1023 bộ so sánh tương tự
- Thời gian biến đổi – không sử dụng xung clock do vậy quá trình biến đổi là liên tục. Thời gian biến đổi rất ngắn chỉ khoảng 17 ns.
- Bộ biến đổi flash 3 bit được miêu tả như hình sau

37

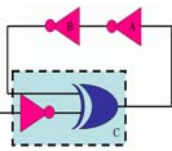


# Flash ADC 3 bit

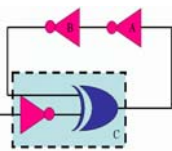
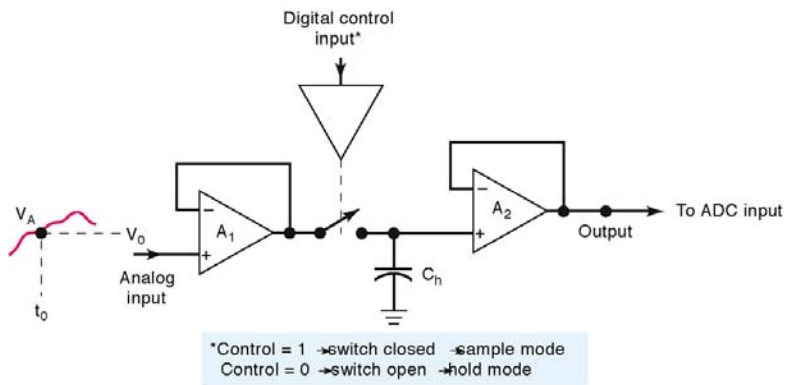


Analog in $V_A$	Comparator outputs							Digital outputs		
	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$	$C_6$	$C_7$	C	B	A
0-1 V	1	1	1	1	1	1	1	0	0	0
1-2 V	0	1	1	1	1	1	1	0	0	1
2-3 V	0	0	1	1	1	1	1	0	1	0
3-4 V	0	0	0	1	1	1	1	0	1	1
4-5 V	0	0	0	0	1	1	1	1	0	0
5-6 V	0	0	0	0	0	1	1	1	0	1
6-7 V	0	0	0	0	0	0	1	1	1	0
> 7 V	0	0	0	0	0	0	0	1	1	1

38



# Mạch lấy mẫu và giữ

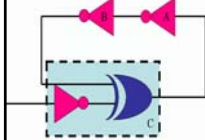


# Câu hỏi?



# Chương 11

## Thiết bị nhớ

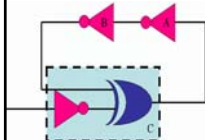


Th.S Đặng Ngọc Khoa  
Khoa Điện - Điện Tử

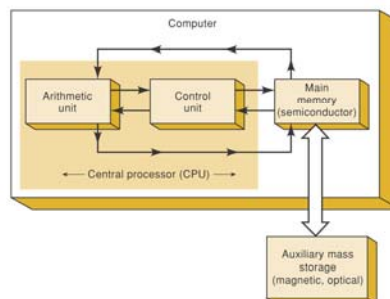
1



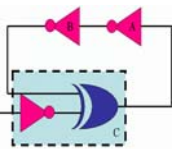
## Thiết bị nhớ



- Một hệ thống thường sử dụng
  - Bộ nhớ trong (làm việc) tốc độ cao
  - Bộ nhớ ngoài (lưu trữ) tốc độ thấp hơn



2

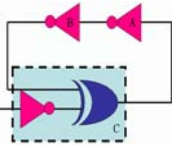


## Thuật ngữ thường sử dụng



- **Memory Cell:** một thiết bị hay một mạch có khả năng lưu trữ một bit dữ liệu
- **Memory Word:** một nhóm các bit, thông thường một từ có 8 – 64 bit.
- **Byte:** một nhóm 8 bit.
- **Dung lượng:** mô tả khả năng lưu trữ của bộ nhớ. Dung lượng mô tả số word có trong bộ nhớ.
  - $1K = 2^{10}$  word
  - $1M = 2^{20}$  word
  - $1G = 2^{30}$  word
  - $2K \times 8 = 2 \cdot 2^{10} \times 8 = 2.1024.8$  word

3



## Thuật ngữ thường sử dụng

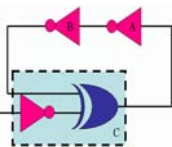


- **Address:** là số xác định vị trí của từ (word) trong bộ nhớ.
- **Lệnh đọc:** thực hiện việc đọc dữ liệu ra từ bộ nhớ.
- **Lệnh ghi:** thực hiện lệnh ghi dữ liệu vào bộ nhớ.

Addresses

000	Word 0
001	Word 1
010	Word 2
011	Word 3
100	Word 4
101	Word 5
110	Word 6
111	Word 7

4

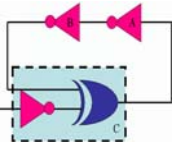


## Thuật ngữ thường sử dụng



- **RAM:** Random-Access Memory.
- **SAM:** Sequential-Access Memory
- **ROM:** Read Only Memory
- **RWM:** Read/Write Memory
- **Static Memory Devices:** dữ liệu được lưu mãi mãi khi còn nguồn cung cấp.
- **Dynamic Memory Devices:** dữ liệu không được lưu mãi mãi, để lưu dự liệu được lưu trữ ta cần rewritten dữ liệu.
- **Main Memory:** bộ nhớ làm việc
- **Auxiliary Memory:** bộ nhớ thứ cấp dùng để lưu trữ.

5



## Hoạt động của bộ nhớ

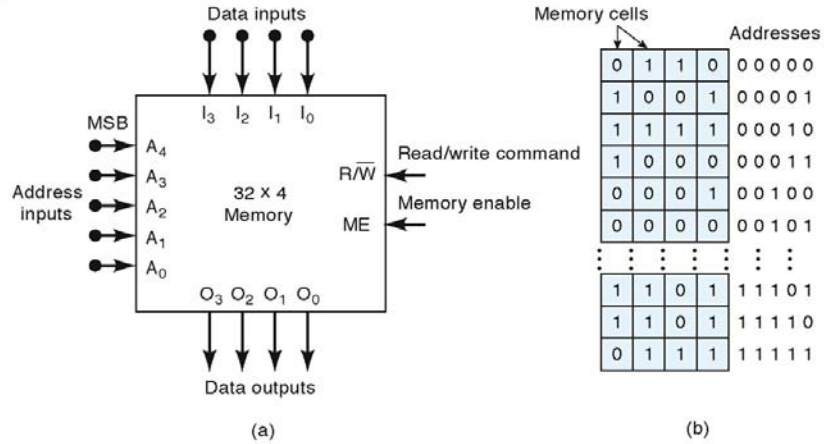
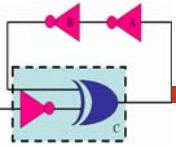


1. Xác định địa chỉ trong bộ nhớ được truy cập bởi lệnh ghi hoặc đọc.
2. Xác định lệnh (ghi hoặc đọc) cần thực hiện.
3. Cung cấp dữ liệu để lưu vào bộ nhớ trong quá trình ghi.
4. Nhận dữ liệu ở ngõ ra trong quá trình đọc.
5. Enable hay Disable sao cho bộ nhớ đáp ứng đến địa chỉ và lệnh thực thi.

6



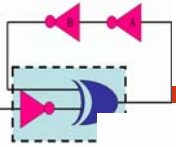
# Cấu trúc của bộ nhớ



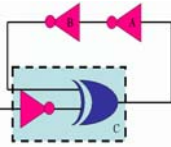
Cấu trúc bộ nhớ 32x4



# Cấu trúc của bộ nhớ



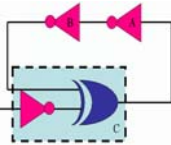
- a) Ghi dữ liệu 0100 vào bộ nhớ tại địa chỉ 00011.
- b) Đọc dữ liệu 1101 từ bộ nhớ ở địa chỉ 11110



## Bài tập 01

- Xác định giá trị của các ngõ vào và ngõ ra khi đọc dữ liệu từ địa chỉ 00100.
- Xác định giá trị của các ngõ vào và ra khi ghi dữ liệu 1110 vào 01101.

9

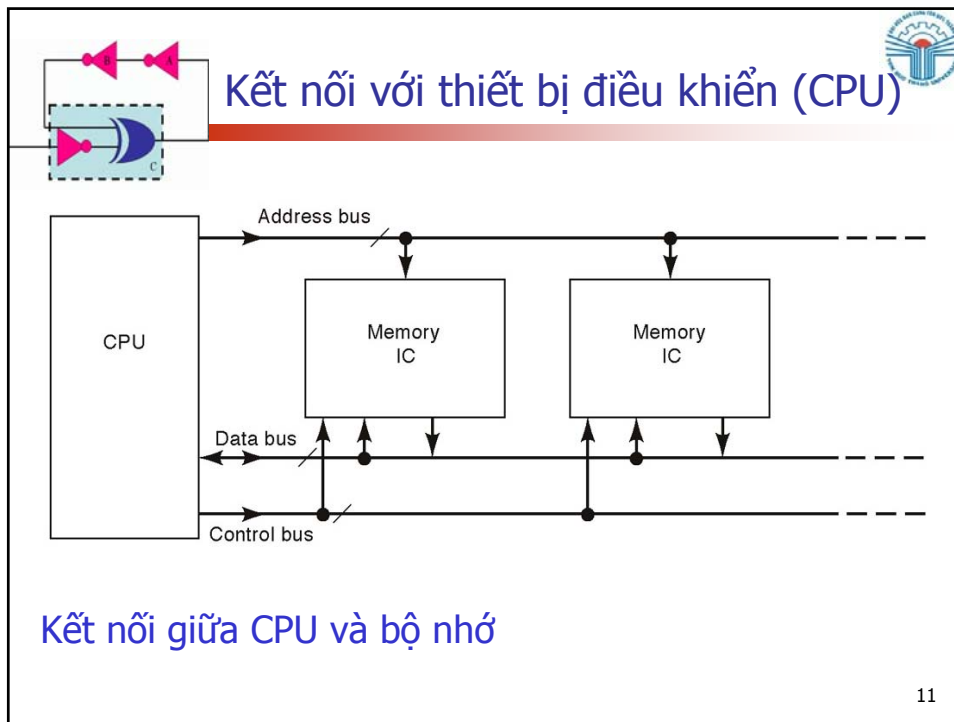


## Bài tập 02

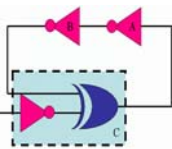
- Cho một bộ nhớ có dung lượng 4Kx8
  - Cần bao nhiêu đường dữ liệu ngõ vào và ngõ ra.
  - Cần bao nhiêu đường địa chỉ.
  - Tính tổng số byte có trong bộ nhớ.

10





- 
- The diagram illustrates the connection between a CPU and two Memory ICs. The CPU is connected to the Memory ICs through three buses: the Address bus, the Data bus, and the Control bus. The Address bus is a single line that carries the address from the CPU to the Memory ICs. The Data bus is a multi-line bus that carries data between the CPU and the Memory ICs. The Control bus is a single line that carries control signals from the CPU to the Memory ICs. The diagram shows the CPU on the left, and two Memory ICs on the right. The Address bus is at the top, the Data bus is in the middle, and the Control bus is at the bottom. Arrows indicate the direction of data flow: from the CPU to the Memory ICs on the Address bus, from the Memory ICs to the CPU on the Data bus, and from the CPU to the Memory ICs on the Control bus.
- Kết nối với thiết bị điều khiển (CPU)
- **Viết dữ liệu**
    1. CPU cung cấp địa chỉ nhị phân.
    2. CPU đưa dữ liệu vào data bus
    3. CPU kích hoạt tín hiệu điều khiển phù hợp.
    4. Bộ nhớ sẽ giải mã địa chỉ nhị phân
    5. Data được đưa đến địa chỉ được chọn.
  - **Đọc dữ liệu**
    1. CPU cung cấp địa chỉ nhị phân.
    2. CPU kích hoạt tín hiệu điều khiển phù hợp.
    3. Bộ nhớ sẽ giải mã địa chỉ nhị phân
    4. Bộ nhớ đưa dữ liệu phù hợp lên data bus
- 12

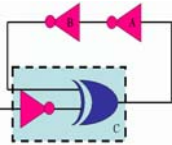


## ROM (Read Only Memories)

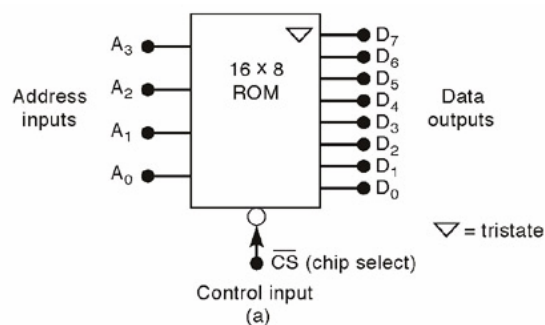


- ROM là bộ nhớ bán dẫn được thiết kế để lưu dữ liệu lâu dài.
- Trong quá trình hoạt động, dữ liệu không thể ghi vào ROM nhưng có thể đọc ra từ ROM.
- ROM có thể được nạp dữ liệu bởi nhà sản xuất hoặc người sử dụng.
- Dữ liệu trong ROM không bị mất bi khi hệ thống bị mất điện

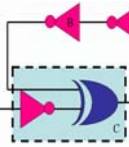
13



## ROM (Read Only Memories)



14



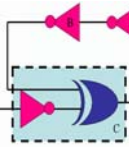
# ROM(tt)

Word	Address				Data							
	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	0	1	1	0	1	1	1	1	0
1	0	0	0	1	0	0	1	1	1	0	1	0
2	0	0	1	0	1	0	0	0	0	1	0	1
3	0	0	1	1	1	0	1	0	1	1	1	1
4	0	1	0	0	0	0	0	1	1	0	0	1
5	0	1	0	1	0	1	1	1	1	0	1	1
6	0	1	1	0	0	0	0	0	0	0	0	0
7	0	1	1	1	1	1	1	0	1	1	0	1
8	1	0	0	0	0	0	1	1	1	1	0	0
9	1	0	0	1	1	1	1	1	1	1	1	1
10	1	0	1	0	1	0	1	1	1	0	0	0
11	1	0	1	1	1	1	0	0	0	1	1	1
12	1	1	0	0	0	0	1	0	0	1	1	1
13	1	1	0	1	0	1	1	0	1	0	1	0
14	1	1	1	0	1	1	0	1	0	0	1	0
15	1	1	1	1	0	1	0	1	1	0	1	1

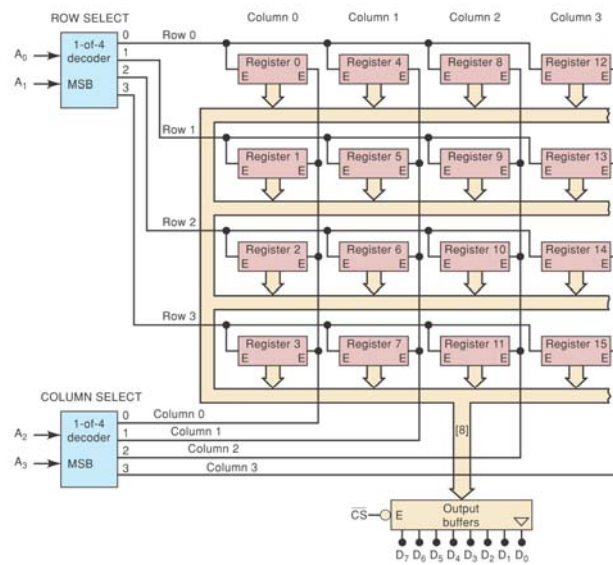
(b)

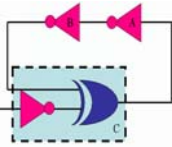
Word	Address				Data
	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	A <sub>0</sub>	D <sub>7</sub> -D <sub>0</sub>
0	0				DE
1	1				3A
2	2				85
3	3				AF
4	4				19
5	5				7B
6	6				00
7	7				ED
8	8				3C
9	9				FF
10	A				B8
11	B				C7
12	C				27
13	D				6A
14	E				D2
15	F				5B

(c)



# Cấu trúc của ROM

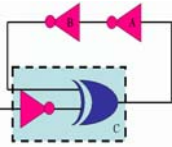




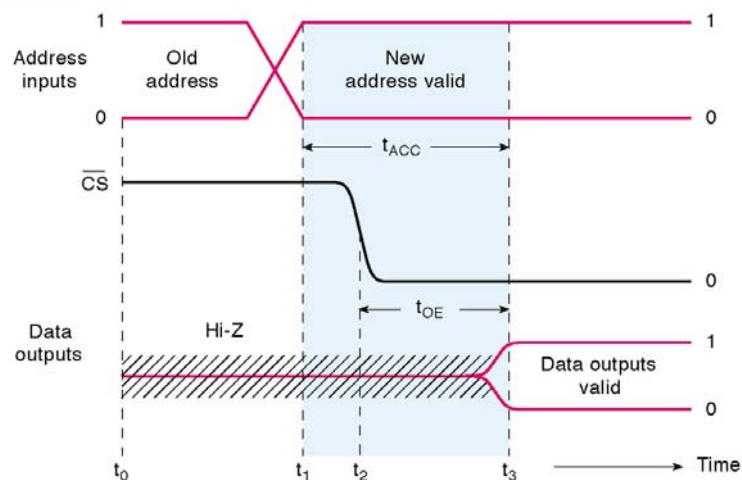
## Cấu trúc của ROM

- Cấu trúc bên trong của ROM rất phức tạp nhưng, bao gồm những phần chính sau:
  - **Ma trận thanh ghi:** gồm những thanh ghi lưu trữ dữ liệu trong ROM. Mỗi thanh ghi chứa được một từ và có một địa chỉ tương ứng.
  - **Giải mã địa chỉ:** bao gồm giải mã địa chỉ hàng và giải mã địa chỉ cột.
  - **Bộ đệm ngõ ra:** dữ liệu được chọn sẽ được đến bộ đệm ngõ ra khi CS ở mức thấp. Khi CS ở mức cao, các ngõ ra của bộ đệm sẽ ở trạng thái tổng trở cao.

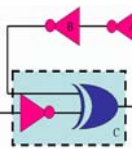
17



## Giải đồ thời gian



18

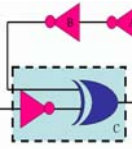


## MROM (Mask-programmed ROM)

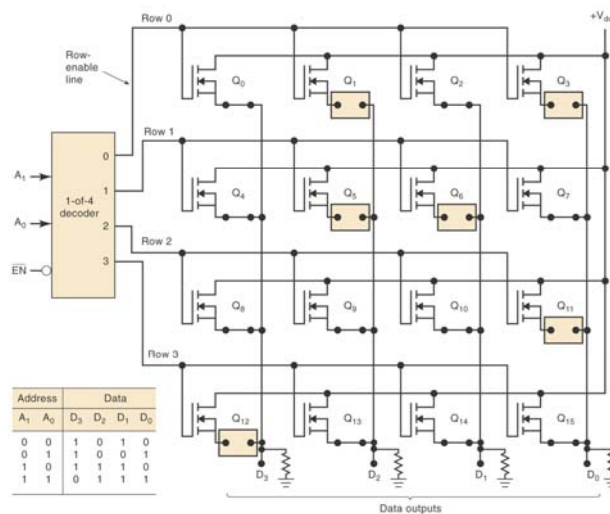


- MROM là ROM mà dữ liệu được nhập bởi nhà sản xuất theo yêu cầu của khách hàng.
- Phim âm bản (mask) được sử dụng để kết nối trong ROM.
- Có hiệu quả kinh tế khi sản xuất với số lượng lớn
- Cấu trúc của một MROM 16 bit nhớ như sau


19



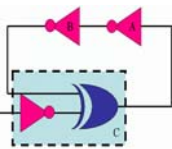
## MROM (Mask-programmed ROM)



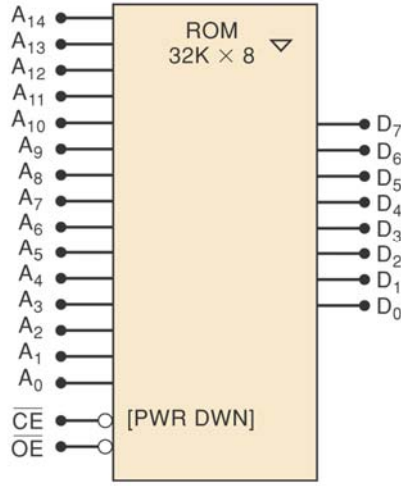
20




## MROM (Mask-programmed ROM)



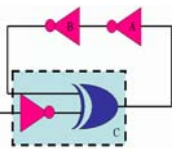
- Khi CE disabled, tất cả các chức năng của chip sẽ disabled.
- Khi OE disabled, chỉ những ngõ ra 3 trạng thái là disabled



21

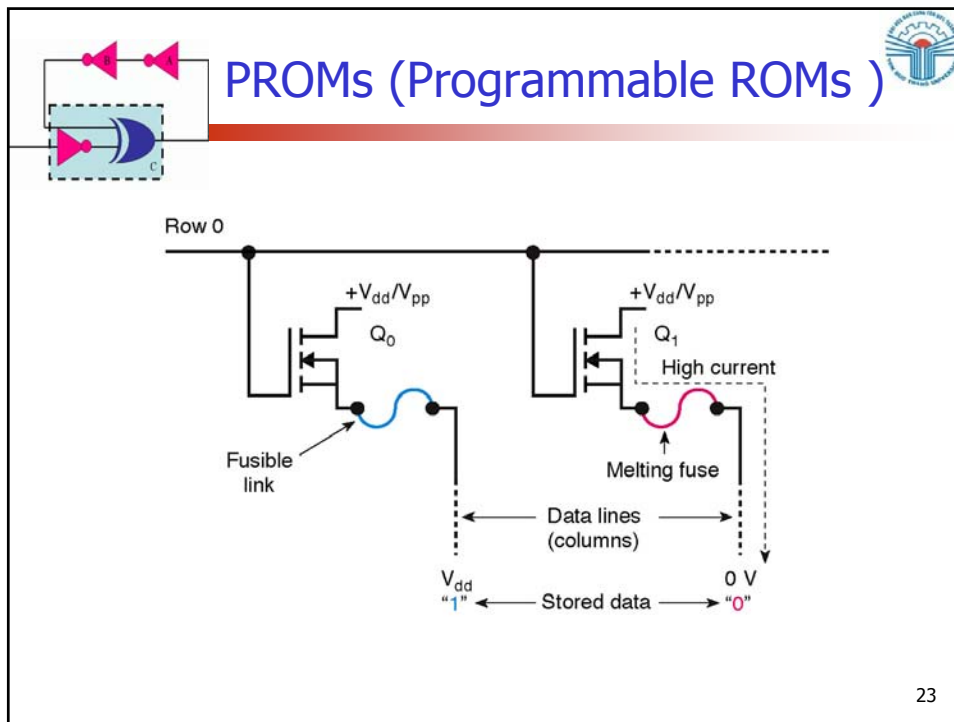


## PROMs (Programmable ROMs)



- PROM là các loại ROM có thể được lập trình ( nạp dữ liệu) bởi người sử dụng.
- PROM có cấu trúc dựa vào các *kết nối nấu chảy* (cầu chì).
- Khi nạp dữ liệu cho ROM thì chương trình sẽ *nấu chảy các kết nối* tương ứng.
- PROM là loại ROM sử dụng một lần.
- Kinh tế trong trường hợp sử dụng với số lượng nhỏ

22

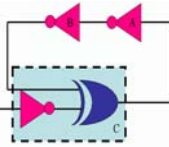


23

The diagram illustrates the internal structure of a PROM cell. It shows a horizontal line labeled "Row 0" at the top. Below this line, two transistors,  $Q_0$  and  $Q_1$ , are connected. The gates of these transistors are connected to the Row 0 line. The drains of  $Q_0$  and  $Q_1$  are connected to a common supply voltage  $+V_{dd}/V_{pp}$ . The sources of  $Q_0$  and  $Q_1$  are connected to data lines (columns). A blue wavy line labeled "Fusible link" is shown between the source of  $Q_0$  and the data line. A red wavy line labeled "Melting fuse" is shown between the source of  $Q_1$  and the data line. A dashed line indicates a "High current" path through  $Q_1$  to the data line. The data lines are labeled "Data lines (columns)". The supply voltage is labeled  $V_{dd}$  and the ground is labeled "0 V". The stored data is labeled "1" and "0".

- Bipolar PROM phổ biến là 74186, ROM này có cấu trúc gồm 64 từ 8 bit.
- TBP28S166 cũng là một bipolar PROM có dung lượng 2K x 8.
- MOS PROM có dung lượng lớn hơn bipolar PROM. TMS27PC256 là một MOS PROM có dung lượng 32K x 8.

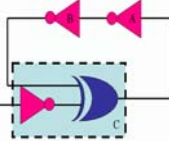
24



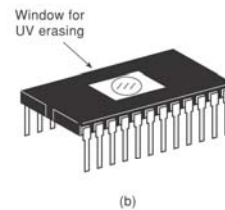
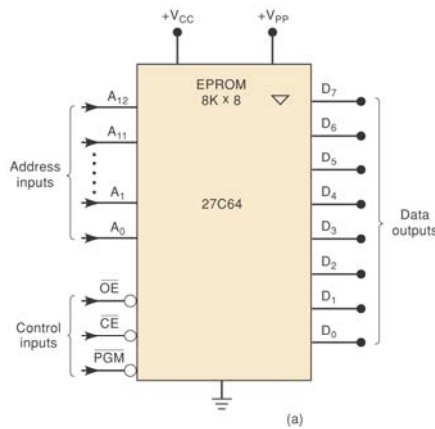
## EPROM (Erasable Programmable ROM)

- EPROM có thể được lập trình bởi người sử dụng và nó cũng có thể được xóa và lập trình lại.
- Phải có mạch nạp dữ liệu chuyên dụng dành riêng cho từng ROM.
- Sử dụng tia UV để xóa dữ liệu
- Tất cả dữ liệu trong EPROM sẽ được xóa
- Sơ đồ của một EPROM tiêu biểu (27C64) như sau:

25



## EPROM (Erasable Programmable ROM)

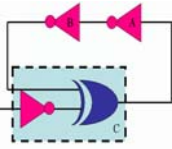


Mode	Inputs			VPP	Outputs
	CE	OE	PGM		
Read	0	0	1	0-5V	DATA <sub>out</sub>
Output Disable	0	1	1	0-5V	High Z
Standby	1	X	X	X	High Z
Program	0	1	0	12.75 V	DATA <sub>in</sub>
PGM Verify	0	0	1	12.75 V	DATA <sub>out</sub>

(c)

26

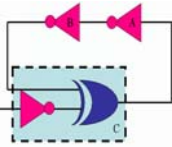




## EEPROM (Electrically Erasable PROM)

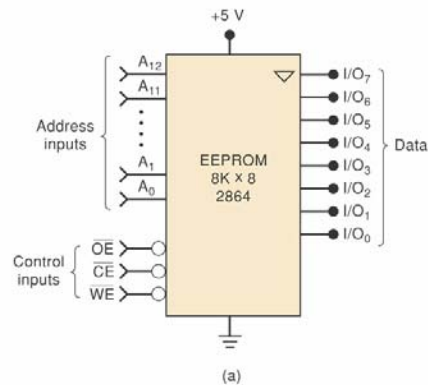
- EPROM có hai nhược điểm chính là:
  - Chúng ta phải tháo chúng ta khỏi mạch để xóa và lập trình lại.
  - Mỗi lần xóa và lập trình lại phải làm thực hiện cho toàn bộ ROM
  - Thời gian xóa lâu (khoảng 30 phút)
- EEPROM có thể khắc phục được những nhược điểm ở trên.
  - Sử dụng điện áp để xóa dữ liệu
  - Có thể xóa dữ liệu cho từng byte

27



## EEPROM (Electrically Erasable PROM)

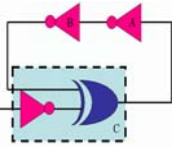
### ■ EEPROM 2864 8K x 8



Mode	Inputs			I/O pins
	$\overline{CE}$	OE	$\overline{WE}$	
READ	LOW	LOW	HIGH	DATA <sub>OUT</sub>
WRITE	LOW	HIGH	LOW	DATA <sub>IN</sub>
STANDBY	HIGH	X	X	High Z

(b)

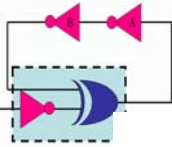
28



## CD ROM

- CD ROM là những đĩa được sản xuất với một bề mặt phản xạ.
- Dữ liệu số được lưu trên đĩa bằng cách đốt cháy hay không một khe trên bề mặt đĩa.
- Là phải phải pháp lưu dữ một cách kinh tế dung lượng lớn dữ liệu

29



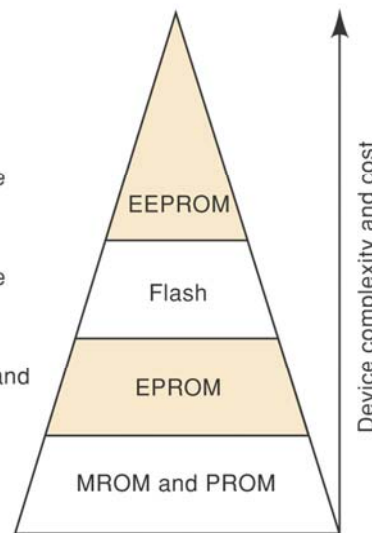
## Flash Memory

In-circuit, electrically erasable byte-by-byte

In-circuit, electrically erasable by sector or in bulk (all cells)

UV erasable in bulk; erased and reprogrammed out of circuit

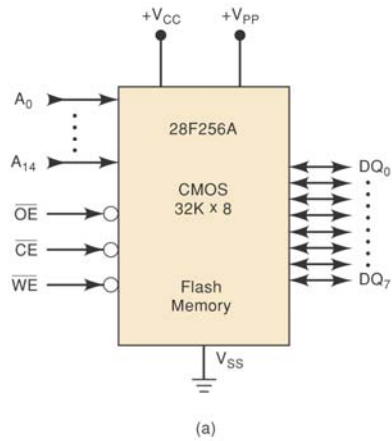
Cannot be erased and reprogrammed



30



## ■ 28F256A flash memory



Mode	Inputs			Data pins
	$\overline{\text{CE}}$	$\overline{\text{OE}}$	$\overline{\text{WE}}$	
READ	LOW	LOW	HIGH	DATA <sub>OUT</sub>
STANDBY	HIGH	X	X	High Z
WRITE*	LOW	HIGH	LOW	DATA <sub>IN</sub>

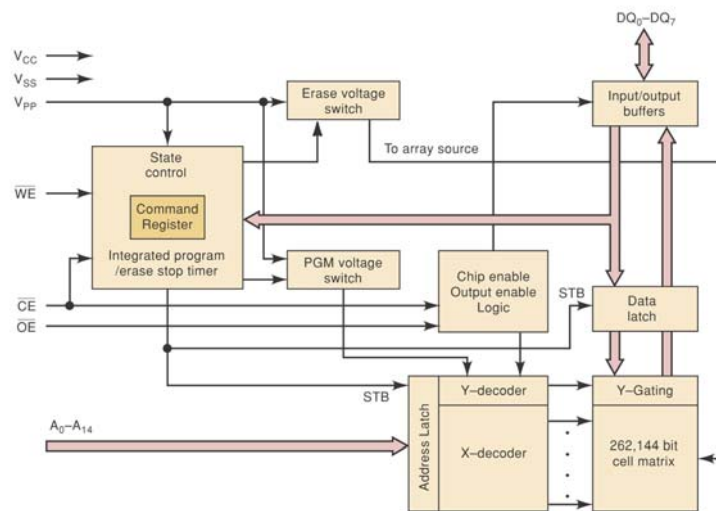
\*Note: If  $V_{PP} \leq 6.5$  V, a write operation cannot be performed

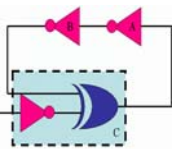
(a)

(b)



## Sơ đồ chức năng của 28F256A



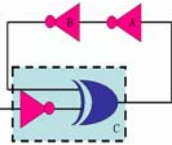


## Ứng dụng của ROM



- ROM có thể được sử dụng trong bất kỳ ứng dụng nào cần lưu trữ dữ liệu ít hay không thay đổi.
  - Firmware: dùng để lưu trữ dữ liệu firmware cho các hệ thống microcomputer.
  - Bảng dữ liệu: lưu trữ data cho những ứng dụng tra dữ liệu.
  - Chuyển đổi dữ liệu: lưu trữ data cho các phép biến đổi.
  - Tạo ra những hàm biến đổi dữ liệu.

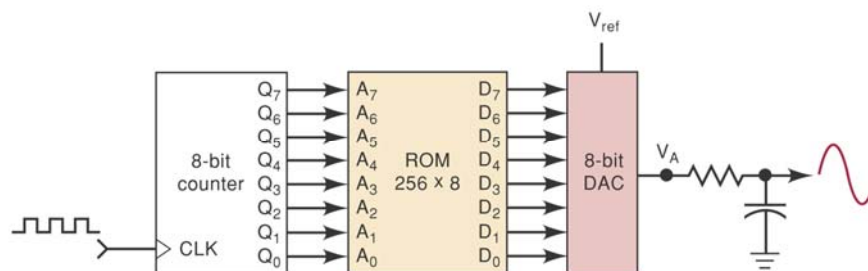
33



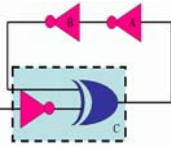
## Ứng dụng của ROM



- Ví dụ ứng dụng của ROM trong mạch tạo ra sóng dạng hình sin.



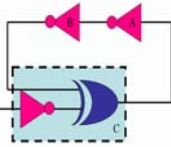
34



## Bài tập 03

- Sử dụng ROM để thiết kế một mạch có 3 bit ngõ vào và ngõ ra thể hiện bình phương giá trị ngõ vào.

35

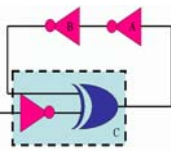


## Bài tập 04

- Sử dụng mạch cho ở trang 21 để thiết lập mạch thể hiện hàm

$$y = 3x + 5$$

36

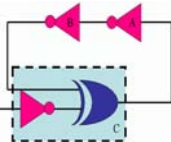


## RAM bán dẫn



- Ngược lại với ROM, RAM bán dẫn là bộ nhớ có thể ghi và đọc được.
- Nhược điểm chính: dữ liệu dễ bị thay đổi.
- Ưu điểm chính: có thể ghi và đọc một cách nhanh chóng và dễ dàng.

37

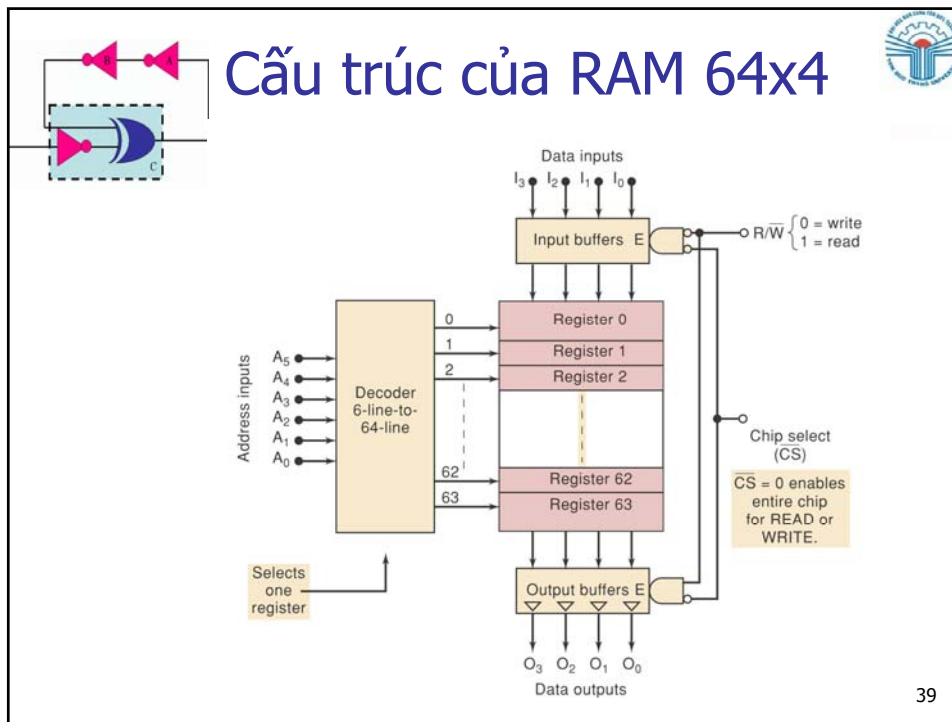


## Cấu trúc của RAM



- Tương tự như ROM, RAM cũng bao gồm một số thanh ghi. Mỗi thanh ghi chứa một từ và có một địa chỉ duy nhất.
- Thông thường dung lượng của RAM là 1K, 4K, 8K, 16K, 64K, 128K, 256K.
- Kích thước của một từ trong RAM có thể là 1, 4 hay 8 bit.

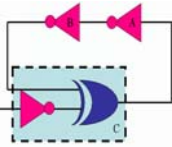
38



## Cấu trúc của RAM 64x4

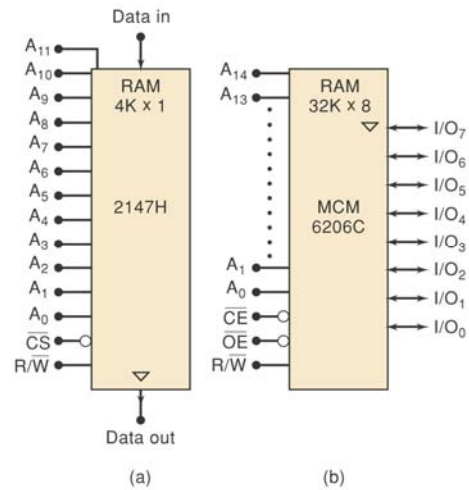
- RAM này chứa 64 từ, mỗi từ 4 bit.
- Mỗi thanh ghi có một địa chỉ tương ứng 0-63<sub>10</sub>. Do vậy cần tất cả 6 đường địa chỉ.
- 6 đường địa chỉ được đưa qua một bộ giải mã 6->64.
- Ngõ ra nào ở mức cao thì thanh ghi tương ứng được chọn.

40

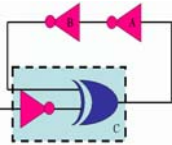


## Bài tập 05

- Hãy tính dung lượng của những RAM sau đây



41

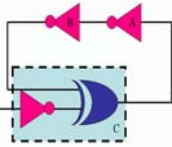


## SRAM (Static RAM)

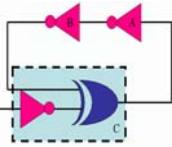
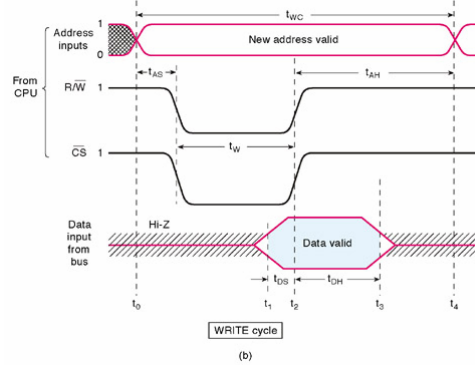
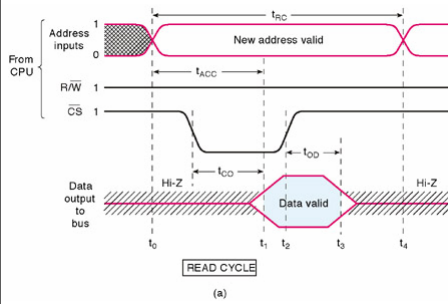
- RAM tĩnh là RAM mà dữ liệu được lưu trữ trong RAM trong suốt thời gian RAM được cấp nguồn.
- Mỗi cell của SRAM chứa 1 bit và được cấu tạo từ Flip-Flop.
- Giảm thời gian truy cập SRAM như sau:

42



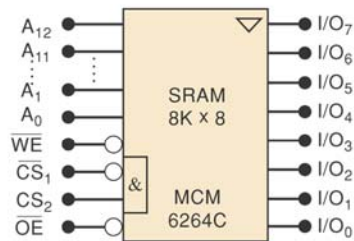


# SRAM (Static RAM)



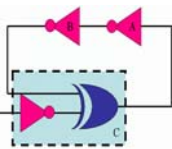
# SRAM (Static RAM)

## ■ Một SRAM thực tế 6264C



Mode	Inputs				I/O pins
	WE	CS <sub>1</sub>	CS <sub>2</sub>	OE	
READ	1	0	1	0	DATA <sub>OUT</sub>
WRITE	0	0	1	X	DATA <sub>IN</sub>
Output disable	1	X	X	1	High Z
Not selected (power down)	X	1	X	X	High Z

X = don't care

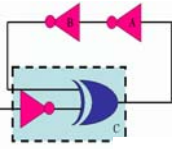


## DRAM (Dynamic RAM)

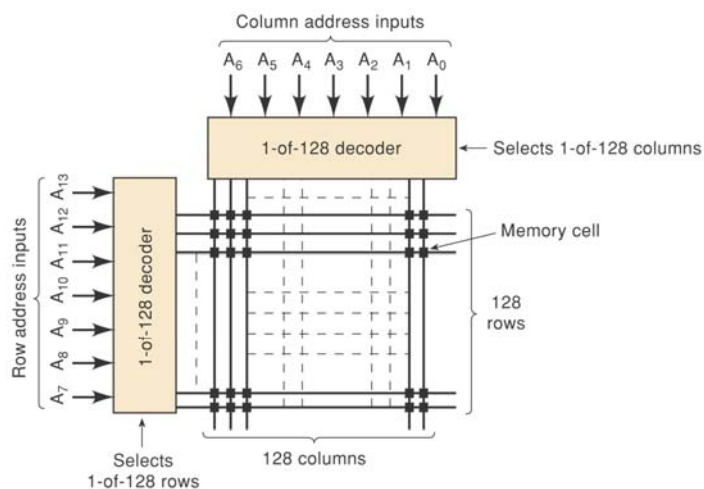


- DRAM được chế tạo từ công nghệ MOS. So với SRAM chúng có dung lượng cao hơn và yêu cầu công suất cung cấp thấp hơn.
- Giá trị của DRAM được lưu trong những tụ điện.
- Do sự rò rỉ điện tích của tụ điện nên DRAM yêu cầu phải được nạp lại điện sau một khoảng thời gian nhất định.
- Thông thường SRAM yêu cầu nạp lại dữ liệu sau 2-10ms.

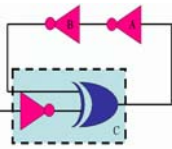
45



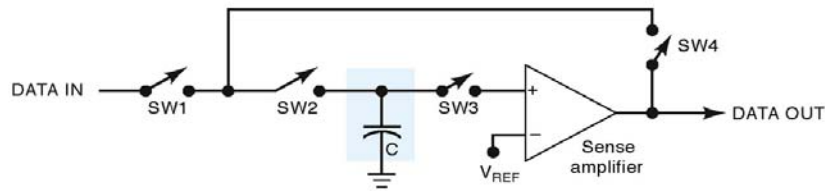
## Cấu trúc của DRAM



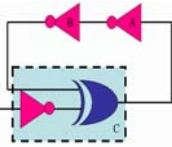
46



# Hoạt động của DRAM

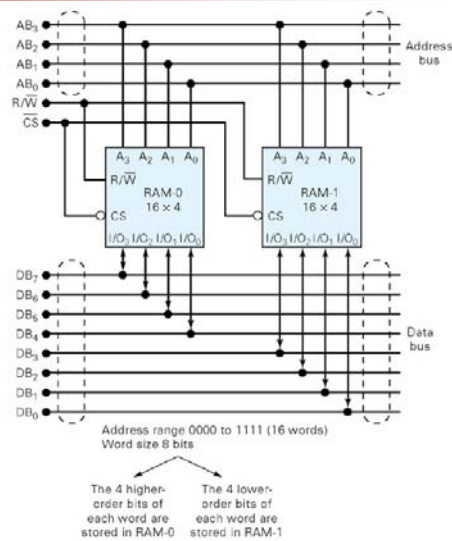


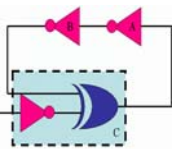
- Trong quá trình ghi công tắc SW1 và SW2 đóng còn SW3 và SW4 mở.
- Trong quá trình đọc SW2, SW3, SW4 đóng còn SW1 mở.



# Mở rộng từ và dung lượng

- Mở rộng từ
- Kết hợp hai RAM 16x4 thành RAM 16x8

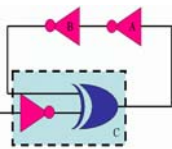
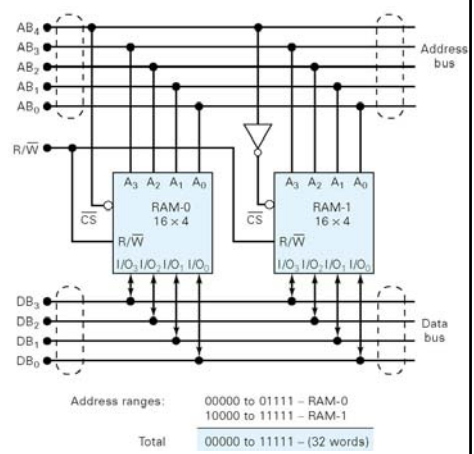




# Mở rộng từ và dung lượng



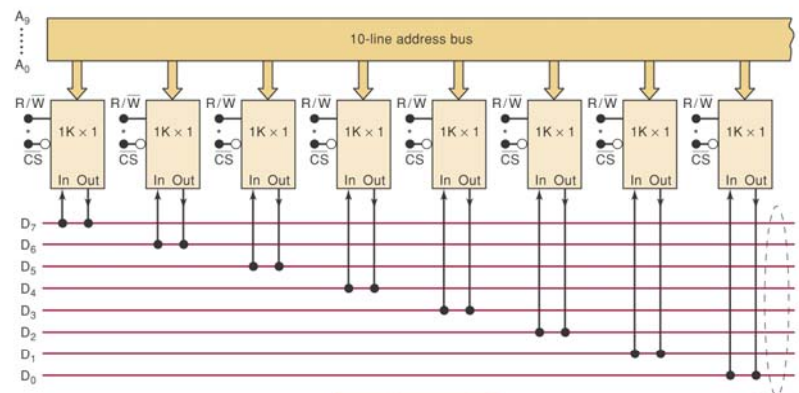
- Mở rộng dung lượng
- Kết hợp hai chip 16x4 thành bộ nhớ 32x4



# Bài tập 06



- Tính dung lượng của mạch sau đây

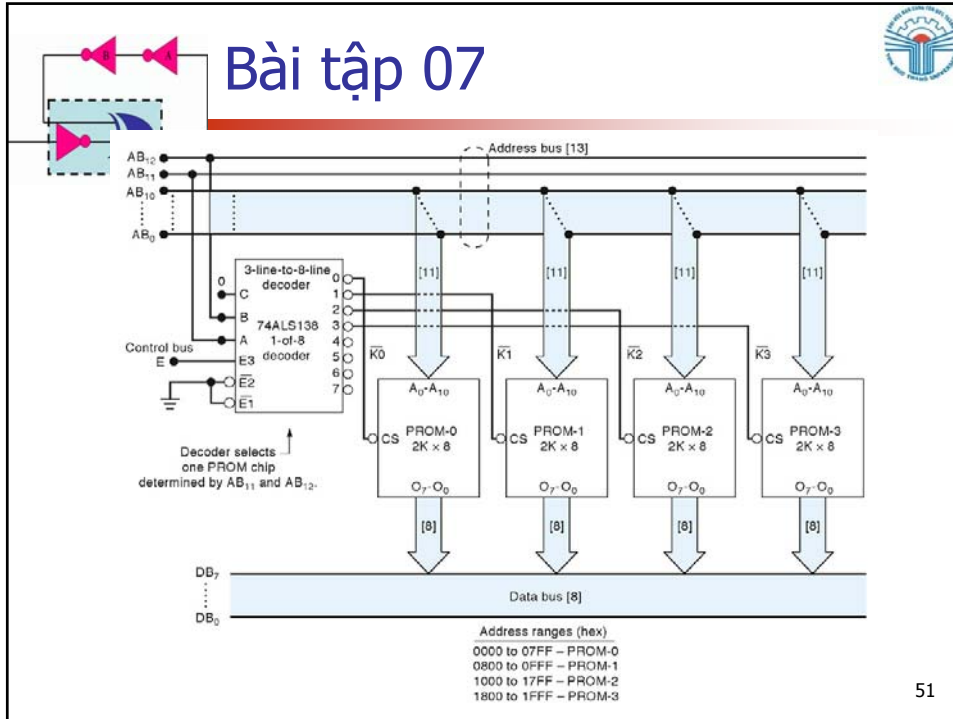


\* All R/W inputs and CS inputs are connected in common

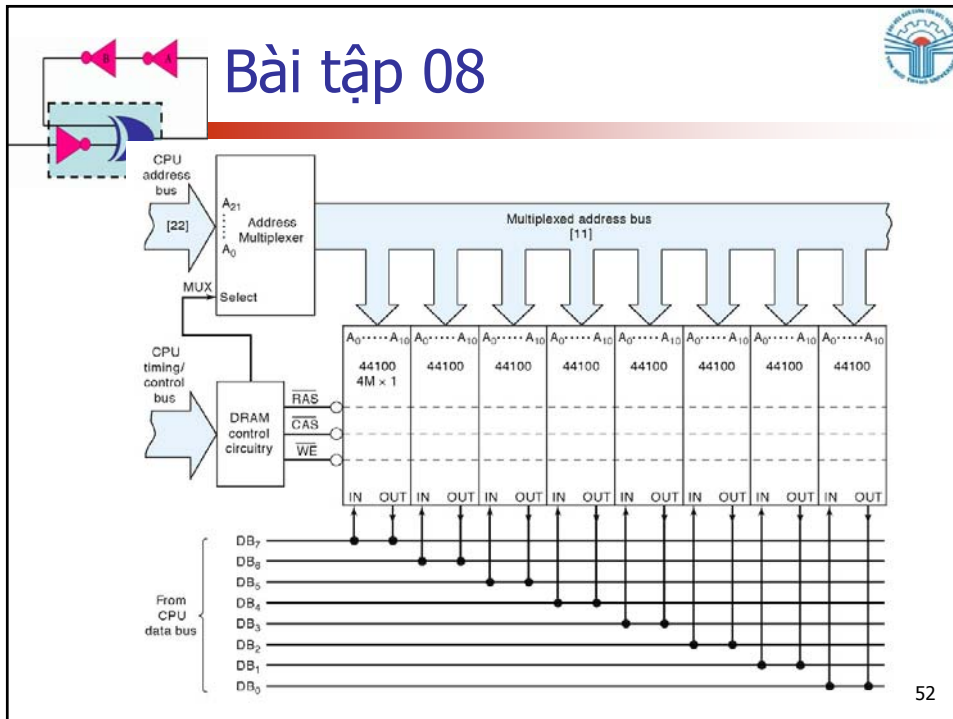
Data bus

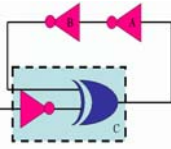


# Bài tập 07



# Bài tập 08

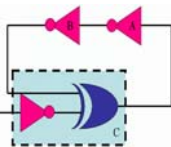




## Bài tập 09

- Từ RAM 6206 (slide 41) hãy thiết kế bộ nhớ 4K x 8.

53



## Câu hỏi?

54