

## Chương 1 Hệ thống số và mạch logic

Nội dung:

- Cơ sở biểu diễn dữ liệu bên trong máy tính
  - + Các hệ đếm cơ bản
  - + Mã hoá và lưu trữ dữ liệu bên trong máy tính
  - + Biểu diễn số nguyên
  - + **Số dấu chấm động**
  - + Biểu diễn ký tự
- Mạch logic và đại số Boole

11/14/2020

1

## Chương 1 Hệ thống số và mạch logic

1. Cơ sở biểu diễn dữ liệu bên trong máy tính

- Hệ thống số: Để biểu diễn một hệ thống số, người ta sử dụng một tập các ký hiệu khác nhau, mỗi ký hiệu được gọi là một ký số (digit). Số các ký hiệu được gọi là cơ số.

Để biểu thị một đại lượng lớn hơn ký số, các ký số được ghép lại tạo thành chuỗi các ký số gọi là một số (number). Vị trí tương đối của ký số trong số liên kết với thừa trọng (weighting factor).

11/14/2020

2

## Chương 1-Hệ thống số và mạch logic

1. Cơ sở biểu diễn dữ liệu bên trong máy tính

Công thức tổng quát

Biểu diễn số  $\longrightarrow V_n V_{n-1} \dots V_0 \cdot V_{-1} \dots V_{-m}$   
 Chấm thập phân

$$Q = V_n \times B^n + V_{n-1} \times B^{n-1} + \dots + V_0 \times B^0 + V_{-1} \times B^{-1} + \dots + V_{-m} \times B^{-m}$$

$$Q = \sum_{i=-m}^n v_i \times B^i$$

Trong đó: **B** là cơ số

$V_i$  là giá trị tương ứng của ký số ( $0 \leq v_i \leq B-1$ )

**Q** là giá trị tính trong hệ 10.

11/14/2020

3

## Chương 1-Hệ thống số và mạch logic

1. Cơ sở biểu diễn dữ liệu bên trong máy tính

Biểu diễn thông tin trong hệ nhị phân

BIT (Binary digit)	: 0 1	
BYTE = tổ hợp 8 bit	: 01001101	11111111
WORD = tổ hợp nhiều bit	: 10110	1011100101

1 KiloByte (KB)	= 1024 byte	= $2^{10}$ byte
1 MegaByte (MB)	= 1024 KB	= $2^{20}$ byte
1 GigaByte (GB)	= 1024 MB	= $2^{30}$ byte

1 số dài n bit thì biểu diễn được  $2^n$  giá trị

11/14/2020

4

## Chương 1-Hệ thống số và mạch logic

### Ví dụ số nhị phân

$$\begin{aligned}
 &1101001.1011_{(2)} \\
 &\quad \begin{array}{cccccccc} 6 & 5 & 4 & 3 & 2 & 1 & 0 & -1 & -2 & -3 & -4 \end{array} \\
 &= 2^6 + 2^5 + 2^3 + 2^0 + 2^{-1} + 2^{-3} + 2^{-4} \\
 &= 64 + 32 + 8 + 1 + 0.5 + 0.125 + 0.0625 \\
 &= 105.6875_{(10)}
 \end{aligned}$$

11/14/2020

5

## Chương 1-Hệ thống số và mạch logic

### Phương pháp chia dần cho 2

- Ví dụ: chuyển đổi  $105_{(10)}$
- $105:2 = 52$  dư 1
  - $52:2 = 26$  dư 0
  - $26:2 = 13$  dư 0
  - $13:2 = 6$  dư 1
  - $6:2 = 3$  dư 0
  - $3:2 = 1$  dư 1
  - $1:2 = 0$  dư 1
- Kết quả:  $105_{(10)} = 1101001_{(2)}$

### Chuyển số lẻ thập phân sang nhị phân

- Ví dụ 1: chuyển đổi  $0.6875_{(10)}$
- $0.6875 \times 2 = 1.375$  phần nguyên = 1
  - $0.375 \times 2 = 0.75$  phần nguyên = 0
  - $0.75 \times 2 = 1.5$  phần nguyên = 1
  - $0.5 \times 2 = 1.0$  phần nguyên = 1
- Kết quả:  $0.6875_{(10)} = 0.1011_{(2)}$

11/14/2020

6

## Chương 1-Hệ thống số và mạch logic

### Hệ mười sáu (Hexa)

Cơ số 16

16 chữ số: 0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F

Dùng để viết gọn cho số nhị phân: cứ một nhóm 4 bit sẽ được thay thế bằng 1 chữ số Hexa

### Quan hệ giữa số nhị phân và số Hexa

4-bit	Chữ số Hexa
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Ví dụ chuyển đổi số nhị phân → số Hexa:

- $0000\ 0000_2 = 00_{16}$
- $1011\ 0011_2 = B3_{16}$
- $0010\ 1101\ 1001\ 1010_2 = 2D9A_{16}$
- $1111\ 1111\ 1111\ 1111_2 = FFFF_{16}$

11/14/2020

7

## Chương 1 - Hệ thống số và mạch logic

### 2. Mã hoá và lưu trữ dữ liệu trong máy tính

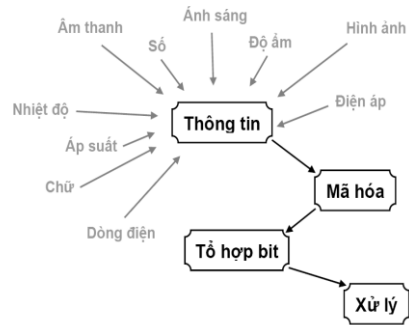
- Nguyên tắc chung:
  - Mọi dữ liệu đưa vào máy tính đều được mã hoá thành số nhị phân.
  - Các loại dữ liệu:
    - Dữ liệu số nguyên: mã hoá theo một số chuẩn quy ước
    - Dữ liệu số thực: mã hoá bằng số dấu chấm động
    - Dữ liệu ký tự: mã hoá theo bộ mã ký tự.
  - Độ dài từ dữ liệu (word): là số bit được sử dụng để mã hoá loại dữ liệu tương ứng, thường là bội số của 8 bit

11/14/2020

8

## Chương 1-Hệ thống số và mạch logic

### Mã hóa thông tin đầu vào



11/14/2020

9

## Chương 1 Hệ thống số và mạch logic

### 2. Biểu diễn số nguyên

Có 2 loại số nguyên:

- Số nguyên không dấu (Unsigned Integer)
- Số nguyên có dấu (Signed Integer)

11/14/2020

10

## Chương 1 - Hệ thống số và mạch logic

### 3. Biểu diễn số nguyên

#### Biểu diễn số

**Số không dấu** Số n bit có giá trị :  $0 \div (2^n - 1)$

Số 8 bit có giá trị :  $0 \div 255$

Số 16 bit có giá trị :  $0 \div 65\,535$

Số 32 bit có giá trị :  $0 \div 4\,294\,967\,295$

**Số có dấu** Qui ước: chọn bit có trọng số cao nhất (MSB) làm bit dấu

bit dấu = 0 là số dương - bit dấu = 1 là số âm  
sử dụng số bù 2 :  $-1 = 1111\,1111$ ,  $-2 = 1111\,1110$ , ...  
 $-127 = 1000\,0001$ ,  $-128 = 1000\,0000$

Số 8 bit có dấu có giá trị :  $-128 \div +127$

Số 16 bit có dấu có giá trị :  $-32768 \div +32767$

11/14/2020

11

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số nguyên không dấu

- Nguyên tắc tổng quát: Dùng n bit biểu diễn số nguyên không dấu A:
- Ví dụ 1: Biểu diễn các số nguyên không dấu sau đây bằng 8-bit:

A=41 ; B=150

$a_{n-1}a_{n-2}\dots a_2a_1a_0$

Giá trị của A được tính như sau:

$$A = \sum_{i=0}^{n-1} a_i 2^i$$

Dải biểu diễn của A: từ 0 đến  $2^n - 1$

**Giải:**

$$A = 41 = 32 + 8 + 1 = 2^5 + 2^3 + 2^0$$

$$41 = 0010\,1001$$

$$B = 150 = 128 + 16 + 4 + 2 = 2^7 + 2^4 + 2^2 + 2^1$$

$$150 = 1001\,0110$$

11/14/2020

12

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số nguyên không dấu

- Ví dụ 2: Cho các số nguyên không dấu M, được biểu diễn bằng 8-bit như sau:

■  $M = 0001\ 0010$

■  $N = 1011\ 1001$

Xác định giá trị của chúng?

Giải:

■  $M = 0001\ 0010 = 2^4 + 2^1 = 16 + 2 = 18$

■  $N = 1011\ 1001 = 2^7 + 2^5 + 2^4 + 2^3 + 2^0$   
 $= 128 + 32 + 16 + 8 + 1 = 185$

Với  $n = 8$  bit

Biểu diễn được các giá trị từ 0 đến 255

0000 0000 = 0

**Chú ý:**

0000 0001 = 1      1111 1111

0000 0010 = 2      + 0000 0001

0000 0011 = 3      1 0000 0000

...

Vậy:  $255 + 1 = 0?$

1111 1111 = 255      → do tràn nhớ ra ngoài

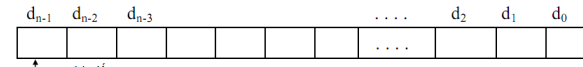
11/14/2020

13

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số nguyên có dấu

- Có nhiều cách để biểu diễn một số  $n$  bit có dấu. Trong tất cả mọi cách thì bit cao nhất luôn tượng trưng cho dấu.
- Khi đó, bit dấu có giá trị là 0 thì số nguyên dương, bit dấu có giá trị là 1 thì số nguyên âm



Số nguyên có bit  $d_{n-1}$  là bit dấu và có trị số tượng trưng bởi các bit từ  $d_0$  tới  $d_{n-2}$ .

11/14/2020

14

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số nguyên có dấu

#### a. Số bù một và Số bù hai:

- Giả sử A là một số nhị phân, ta có:

- Số bù một của A nhận được bằng cách đảo giá trị các bit của A

■  $(\text{Số bù hai của } A) = (\text{Số bù một của } A) + 1$

- Ví dụ: với  $n = 8$  bit

■ Giả sử có      A      = 0010 0101

■ Số bù một của A      = 1101 1010

                                 +      1

■ Số bù hai của A      = 1101 1011

- Vì  $A + (\text{Số bù hai của } A) = 0 \rightarrow$  dùng số bù hai để biểu diễn cho số âm

11/14/2020

15

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số nguyên có dấu

#### b. Biểu diễn số nguyên có dấu bằng mã bù hai

Nguyên tắc tổng quát: Dùng  $n$  bit biểu diễn số nguyên có dấu A:

$$a_{n-1}a_{n-2}\dots a_2a_1a_0$$

- Với A là số dương: bit  $a_{n-1} = 0$ , các bit còn lại biểu diễn độ lớn như số không dấu
- Với A là số âm: được biểu diễn bằng số bù hai của số dương tương ứng, vì vậy bit  $a_{n-1} = 1$

11/14/2020

16

## Chương 1 - Hệ thống số và mạch logic

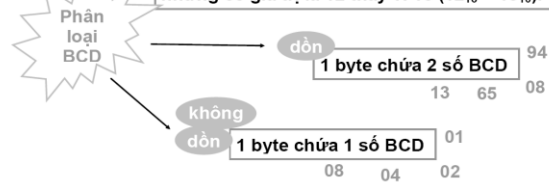
Biểu diễn số nguyên theo mã BCD (Binary Coded Decimal)

### Số BCD

Số BCD được dùng để tính toán trên số thập phân trong hệ nhị phân.

Số BCD là số viết theo hệ 16 nhưng giá trị tính theo hệ 10.

Số  $12_{BCD}$  được viết trong hệ 2 là 0001 0010 nhưng có giá trị là 12 thay vì 18 ( $12_{16} = 18_{10}$ ).



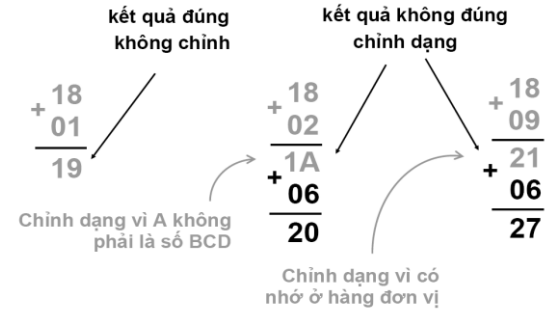
11/14/2020

17

## Chương 1 - Hệ thống số và mạch logic

Biểu diễn số nguyên theo mã BCD (Binary Coded Decimal)

### Chỉnh dạng BCD



11/14/2020

18

## Chương 1 - Hệ thống số và mạch logic

Biểu diễn số dấu chấm động

- Số chấm động được chuẩn hoá, cho phép biểu diễn gần đúng các số thập phân rất lớn hay rất nhỏ dưới dạng một số nhị phân theo một dạng qui ước.
- Thành phần của số chấm động bao gồm: phần dấu, phần mũ và phần định trị. Như vậy, cách này cho phép biểu diễn gần đúng các số thực, tất cả các số đều có cùng cách biểu diễn.

11/14/2020

19

## Chương 1 - Hệ thống số và mạch logic

Biểu diễn số dấu chấm động

### 1. Nguyên tắc chung

- Floating Point Number → biểu diễn cho số thực
- Tổng quát: một số thực  $X$  được biểu diễn theo kiểu số dấu chấm động như sau:

$$X = M * R^E$$

- $M$  là phần định trị (Mantissa),
- $R$  là cơ số (Radix),
- $E$  là phần mũ (Exponent).

11/14/2020

20

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số dấu chấm động

#### 2. Chuẩn IEEE754/85

- Cơ số  $R = 2$
- Các dạng:
  - Dạng 32-bit
  - Dạng 44-bit
  - Dạng 64-bit
  - Dạng 80-bit

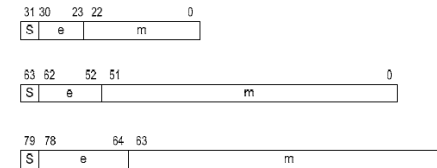
11/14

21

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số dấu chấm động

#### Các dạng biểu diễn chính



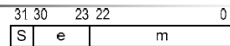
11/14/2020

22

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số dấu chấm động

#### Dạng 32 bit



- S là bit dấu:
  - $S = 0 \rightarrow$  số dương
  - $S = 1 \rightarrow$  số âm
- e (8 bit) là mã *excess-127* của phần mũ E:
  - $e = E + 127 \rightarrow E = e - 127$
  - giá trị 127 gọi là độ lệch (bias)
- m (23 bit) là phần lẻ của phần định trị M:
  - $M = 1.m$
- Công thức xác định giá trị của số thực:
 
$$X = (-1)^S \cdot 1.m \cdot 2^{e-127}$$

11/14/2020

23

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số dấu chấm động

#### Ví dụ 1

Xác định giá trị của số thực được biểu diễn bằng 32-bit như sau:

■ 1 100 0001 0101 0110 0000 0000 0000 0000

■  $S = 1 \rightarrow$  số âm

■  $e = 1000\ 0010_2 = 130 \rightarrow E = 130 - 127 = 3$

Vậy

$$X = -1.10101100 \cdot 2^3 = -1101.011 = -13.375$$

■ 0011 1111 1000 0000 0000 0000 0000 0000 = ?  
= +1.0

11/14/2020

24

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số dấu chấm động



#### Ví dụ 3

Biểu diễn số thực  $X = -0,2$  về dạng số dấu chấm động IEEE754 32-bit

Giải:

- $X = -0,2_{(10)} = -0.00110011...0011..._{(2)} = -1.100110011...0011... \times 2^{-3}$
- Ta có:
  - $S = 1$  vì đây là số âm
  - $E = e - 127 = -3 \rightarrow e = 127 - 3 = 124_{(10)} = 0111\ 1100_{(2)}$
- Vậy:
 
$$X = 1011\ 1110\ 0100\ 1100\ 1100\ 1100\ 1100\ 1100$$

11/14/2020

25

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số dấu chấm động

$$\begin{aligned}
 10.1 \quad -30.375 &= (-11110.011)_{\text{binary}} \\
 &= (-1.1110011)_{\text{binary}} \times 2^4 \\
 &= (-1)1 \times (1 + 0.1110011) \times 2^{(131-127)} \\
 &= (-1)^{\text{sign}} \times (1 + \text{fraction}) \times 2^{(\text{exponent}-127)}
 \end{aligned}$$

$$\text{sign} = 1$$

$$\text{exponent} = 131 = (1000\ 0011)_{\text{binary}}$$

$$\text{fraction} = (1110\ 0110\ 0000\ 0000\ 0000\ 000)_{\text{binary}}$$

$$\begin{aligned}
 (-30.375)_{10} &= (\text{sign exponent fraction})_{\text{binary}} \\
 &= (1\ 1000\ 0011\ 1110\ 0110\ 0000\ 0000\ 0000\ 000)_{\text{binary}} \\
 &= (1100\ 0001\ 1111\ 0011\ 0000\ 0000\ 0000\ 0000)_{\text{binary}} \\
 &= (C1F3\ 0000)_{16}
 \end{aligned}$$

11/14/2020

26

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số dấu chấm động

Bài tập:

1. Biểu diễn số thực  $-23.125$  và  $45.125$  về dạng số dấu chấm động IEEE 32 bit.

2. Xác định dãy số

$1100010100010010100100100000000$  biểu diễn số thực nào? ( $-2345.125$ )

3. Thực hiện phép cộng 2 số nhị phân:  $11.101$  và  $110.1$

KQ:  $1010.0010$

11/14/2020

27

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số dấu chấm động

Bài tập:

3. Biến đổi các số nhị phân sau thành các số thập phân:

$$(a) 11011.101010 \quad (b) 111.111111$$

4. Biểu diễn các số bù hai 8 bit của các số thập phân sau:

$$(a) -25 \quad (b) -128 \quad (c) 1010101$$

11/14/2020

28

## Chương 1 - Hệ thống số và mạch logic

### Biểu diễn số dấu chấm động

#### Bài tập:

5. Cộng 2 số nhị phân 8 bit sau (kết quả lưu ở toán hạng 8 bit) và cho nhận xét về kết quả:

- Số không dấu:  $0FFh + 01h$
- Số có dấu:  $0FFh + 01h$

6. Cho số nhị phân 8 bit  $(1010101)_2$ , số này tương ứng với số nguyên thập phân có dấu và không dấu là bao nhiêu nếu số đang được biểu diễn theo cách biểu diễn số bù 2.

11/14/2020

29

## Chương 1 - Hệ thống số và mạch logic

### 2. Đại số Boole & Mạch logic

- Các linh kiện và mạch điện tử số thuộc về phần cứng máy tính. Chúng nằm trong *cấp logic số* của cấu trúc phân cấp máy tính. Hiện nay các thành phần này được thiết kế xây dựng trên cơ sở công nghệ chế tạo mạch tích hợp IC (Integrated Circuit), thường gọi là công nghệ vi điện tử.

- Phần này chỉ nhắc lại một cách tóm tắt cơ sở toán học có liên quan và các mạch logic cơ bản

11/14/2020

30

## Chương 1 - Hệ thống số và mạch logic

### Đại số Boole

- Các mạch điện tử số được xây dựng trên cơ sở của môn đại số Logic, còn gọi là *đại số Boole*, trong đó một hàm logic có 1 hoặc nhiều biến, mỗi biến chỉ nhận một trong hai giá trị là 0 hoặc 1. Như vậy, một hàm logic có  $n$  biến có thể nhận được một trong  $2^n$  giá trị ứng với các tổ hợp khác nhau của các biến. Giá trị các tổ hợp đó thường được biểu diễn trong 1 bảng gọi là *bảng chân lý*.

- Nếu gọi  $Z$  là hàm của các biến logic  $A, B, C, \dots$  thì có 3 phép toán logic cơ bản là AND, OR và NOT thực hiện các mối quan hệ giữa các biến này được định nghĩa như sau:

11/14/2020

31

## Chương 1 - Hệ thống số và mạch logic

### Đại số Boole

-*Quan hệ AND*: hàm logic biểu diễn sự kiện:

$$Z = A.B$$

-*Quan hệ OR*: hàm logic biểu diễn sự kiện:

$$Z = A + B$$

-*Quan hệ NOT*: hàm logic biểu diễn sự kiện:

$$Z = \bar{A}$$

-*Các quan hệ logic khác có các hàm logic là tổ hợp của 3 hàm cơ bản trên, như hàm NAND, NOR*

-*Hàm XOR*:  $Z = A \oplus B$

11/14/2020

32



## Chương 1 - Hệ thống số và mạch logic

### Đại số Boole

Các biến logic tuân theo các luật sau của đại số Boole:

- Luật giao hoán:  $A.B = B.A$   
 $A + B = B + A$
- Luật kết hợp:  $A(BC) = (AB)C$   
 $A + (B + C) = (A + B) + C$
- Luật phân bố:  $A(B + C) = AB + AC$   
 $A + (BC) = (A + B)(A + C)$

11/14/2020

33

## Chương 1 - Hệ thống số và mạch logic

### Đại số Boole

Các biến logic tuân theo các luật sau của đại số Boole:

- Phép ghim:  $A(A + B) = A$   
 $A + AB = A$
- Luật lặp:  $AA = A$   
 $A + A = A$
- Luật phủ định:  $A.\bar{A} = 0$   
 $A + \bar{A} = 1$

11/14/2020

34

## Chương 1 - Hệ thống số và mạch logic

### Đại số Boole

Các biến logic tuân theo các luật sau của đại số Boole:

- Phủ định kép:  $\overline{\bar{A}} = A$
- Định lý De Morgan:  $\overline{A.B} = \bar{A} + \bar{B}$   
 $\overline{A + B} = \bar{A}.\bar{B}$

11/14/2020

35

## Chương 1 - Hệ thống số và mạch logic

### Đại số Boole

-Để xây dựng và giải các bài toán logic, việc đầu tiên là căn cứ trên các biến vào (các sự kiện vào) thiết lập bảng chân lý của các sự kiện. Từ đó viết được biểu thức của hàm logic ra, thường lấy bằng *tổng của tích các biến* tại những dòng mà hàm nhận giá trị bằng 1.

- Sau đó, căn cứ vào các luật của đại số Boole có thể *tối thiểu hoá* các biểu thức này

- Trong thực tế, còn có thể thực hiện tối thiểu hoá hàm logic theo các phương pháp khác như dùng bản đồ Karnaugh,...

11/14/2020

36

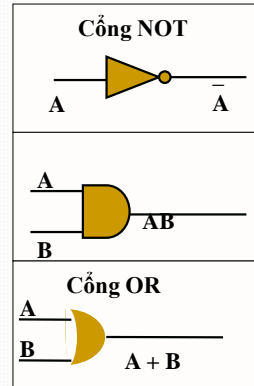
## Chương 1 - Hệ thống số và mạch logic

## Mạch logic cơ bản

## Cổng NOT

## Cổng AND

## Cổng OR



11/14/2020

37

## Chương 1 - Hệ thống số và mạch logic

## Mạch logic cơ bản

INVERTER

A	Z
0	1
1	0

NAND

A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

NOR

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	0

XNOR

A	B	Z
0	0	1
0	1	0
1	0	0
1	1	1

BUFFER

A	Z
0	0
1	1

AND

A	B	Z
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	Z
0	0	0
0	1	1
1	0	1
1	1	1

XOR

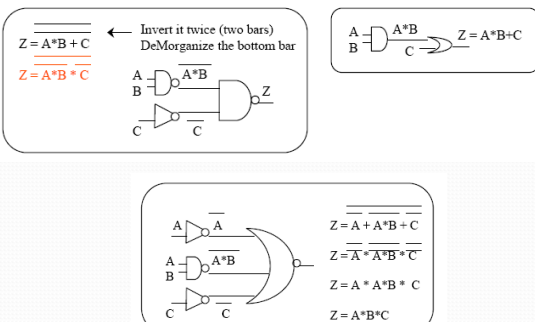
A	B	Z
0	0	0
0	1	1
1	0	1
1	1	0

11/14/2020

38

## Chương 1 - Hệ thống số và mạch logic

## Mạch logic cơ bản



11/14/2020

39

## Chương 1 - Hệ thống số và mạch logic

## Phương trình tổng tích

A	B	C	Z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

$\overline{A} \overline{B} \overline{C}$   
 $\overline{A} \overline{B} C$   
 $\overline{A} B \overline{C}$   
 $A B C$

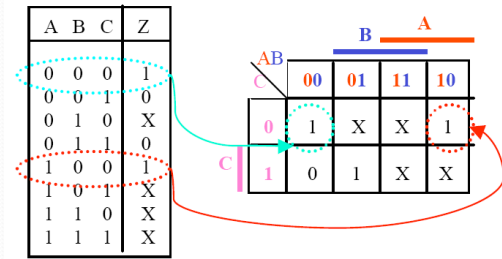
$Z = \overline{\overline{A} \overline{B} \overline{C}} + \overline{\overline{A} \overline{B} C} + \overline{\overline{A} B \overline{C}} + \overline{A B C}$

11/14/2020

40

## Chương 1 - Hệ thống số và mạch logic

## Bản đồ Karnaugh (Karnaugh Maps)

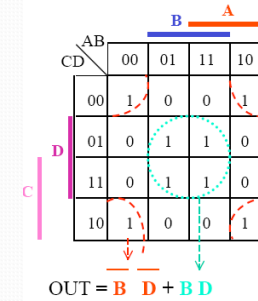


11/14/2020

41

## Chương 1 - Hệ thống số và mạch logic

## Bản đồ Karnaugh (Karnaugh Maps)



11/14/2020

42