



## Chương 3 – Biểu diễn dữ liệu

---

- 3.1. Khái niệm thông tin
- 3.2. Lượng thông tin và sự mã hóa thông tin
- 3.3. Hệ thống số
- 3.4. Các phép tính số học cho hệ nhị phân
- 3.5. Số quá n (excess-n)
- 3.6. Cách biểu diễn số với dấu chấm động
- 3.7. Biểu diễn số BCD(Binary Coded Decimal)
- 3.8. Biểu diễn các ký tự



# Mục tiêu

---

- Hiểu các hệ cơ số thông dụng và cách chuyển đổi.
- Hiểu phương pháp biểu diễn số nguyên và số chấm động.
- Hiểu các phương pháp tính đơn giản với các số.
- Hiểu các phương pháp biểu diễn số BCD và ký tự



# Hình dung về “biểu diễn dữ liệu”

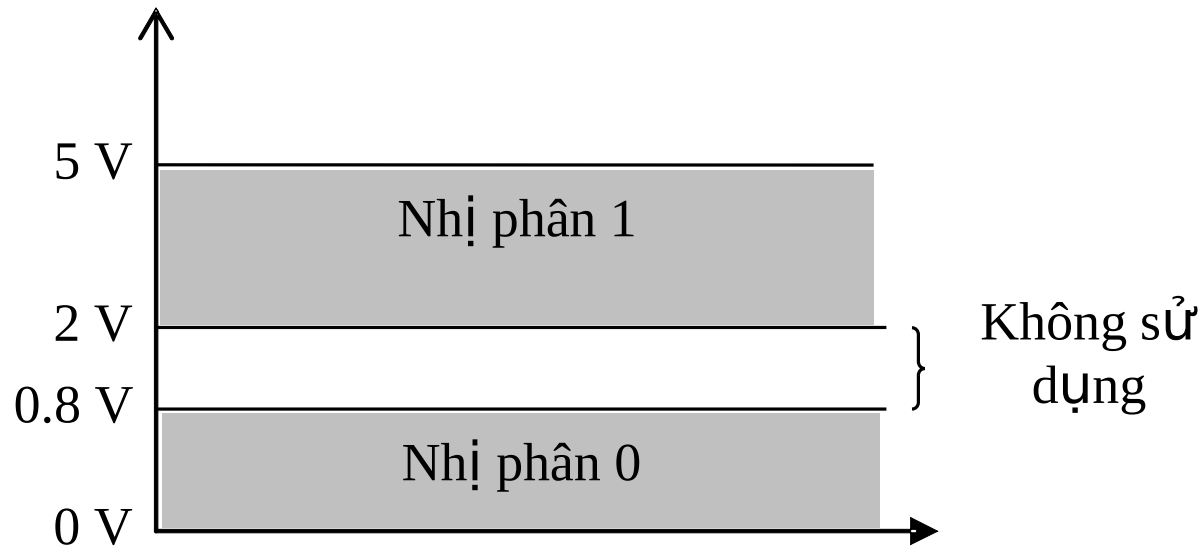
---

- ❑ Mọi thứ trong máy tính đều là 0 và 1
- ❑ Thế giới bên ngoài có nhiều khái niệm như con số, chữ cái, hình ảnh, âm thanh,...
- ❑ → biểu diễn dữ liệu = quy tắc “gắn kết” các khái niệm trong thế giới thật với một dãy số 0 và 1 trong máy tính



## 3.1. Khái niệm thông tin

- ❑ Dùng các tín hiệu điện thế
- ❑ Phân thành các vùng khác nhau



Hình 3.1. Biểu diễn nhị phân qua điện thế



## 3.2. Lượng thông tin và sự mã hoá thông tin

- ❑ Thông tin được đo lường bằng đơn vị thông tin mà ta gọi là bit.
- ❑ Lượng thông tin được định nghĩa bởi công thức:

$$I = \text{Log}_2(N)$$

- Trong đó:
  - I: là lượng thông tin tính bằng bit
  - N: là số trạng thái có thể có
- Ví dụ, để biểu diễn một trạng thái trong 8 trạng thái có thể có, ta cần một số bit ứng với một lượng thông tin là:

$$I = \text{Log}_2(8) = 3 \text{ bit}$$



## 3.3. Hệ Thống Số

□ Dạng tổng quát để biểu diễn giá trị của một số:

$$V_k = \sum_{i=-m}^{n-1} b_i \cdot k^i$$

– Trong đó:

- $V_k$ : Số cần biểu diễn giá trị
- $m$ : số thứ tự của chữ số phần lẻ (phần lẻ của số có  $m$  chữ số được đánh số thứ tự từ -1 đến - $m$ )
- $n-1$ : số thứ tự của chữ số phần nguyên (phần nguyên của số có  $n$  chữ số được đánh số thứ tự từ 0 đến  $n-1$ )
- $b_i$ : giá trị của chữ số thứ  $i$
- $k$ : hệ số ( $k=10$ : hệ thập phân;  $k=2$ : hệ nhị phân;...).



## 3.3. Hệ Thống Số

- Các hệ đếm (cơ số) thông dụng
  - **Thập phân (Decimal)**
    - 10 chữ số : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
  - **Nhị phân (Binary)**
    - 2 chữ số: 0, 1
  - **Bát phân (Octal)**
    - 8 chữ số: 0, 1, 2, 3, 4, 5, 6, 7
  - **Thập lục phân (Hexadecimal)**
    - 16 chữ số: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E.
      - A=10, B=11, C=12, D=13, E=14, F=15




# Chuyển đổi từ cơ số 10 sang b


## □ Quy tắc:

Chia số cần đổi cho b, lấy kết quả chia tiếp cho b cho đến khi **kết quả** bằng **0**. Số ở cơ số b chính là các số dư (của phép chia) viết ngược.

## □ Ví dụ:

41 ÷ 2	= 20	dư	1
20 ÷ 2	= 10	dư	0
10 ÷ 2	= 5	dư	0
5 ÷ 2	= 2	dư	1
2 ÷ 2	= 1	dư	0
1 ÷ 2	= 0	dư	1



$$41_{10} = 101001_2$$






# Chuyển đổi hệ 10 sang Nhị phân

Quy tắc: Người ta chuyển đổi từng phần nguyên và lẻ theo quy tắc sau

**Phần nguyên:** Chia liên tiếp phần nguyên cho 2 giữ lại các số dư, Số nhị phân được chuyển đổi sẽ là dãy số dư liên tiếp tính từ lần chia cuối về lần chia đầu tiên.

**Phần lẻ:** Nhân liên tiếp phần lẻ cho 2, giữ lại các phần nguyên được tạo thành. Phần lẻ của số Nhị phân sẽ là dãy liên tiếp phần nguyên sinh ra sau mỗi phép nhân tính từ lần nhân đầu đến lần nhân cuối



# Chuyển đổi hệ 10 sang Nhị phân

Ví dụ: Chuyển sang hệ Nhị phân số: 13,6875

**Thực hiện:**

**Phần nguyên:**  $13:2 = 6$  dư  $1$   
 $6:2 = 3$  dư  $0$   
 $3:2 = 1$  dư  $1$   
 $1:2 = 0$  dư  $1$



Phần nguyên của số Nhị phân là 1101

**Phần lẻ:**

$0,6875 \times 2 = 1,375$  Phần nguyên là  $1$   
 $0,375 \times 2 = 0,750$  Phần nguyên là  $0$   
 $0,750 \times 2 = 1,500$  Phần nguyên là  $1$   
 $0,5 \times 2 = 1,00$  Phần nguyên là  $1$



Phần lẻ của số Nhị phân là: 0,1011

**Ta viết kết quả là:**  $(13,6875)_{10} = (1101,1011)_2$

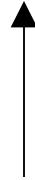


# Chuyển đổi từ cơ số 10 sang b

❑ **Quy tắc:** Chia số cần đổi cho b, lấy kết quả chia tiếp cho b cho đến khi **kết quả** bằng 0. Số ở cơ số b chính là các số dư (của phép chia) viết ngược.

❑ Ví dụ:

$$\begin{array}{rcll} 41 \div 16 & = & 2 & \text{dư } 9 \\ 2 \div 16 & = & 0 & \text{dư } 2 \end{array}$$



$$41_{10} = \underline{29}_{16}$$



**Ví dụ:** Chuyển số  $(3287,5100098)_{10}$  sang Cơ số 8.

□ Phần nguyên:

$3287:8 = 410$	dư	7	↑
$410:8 = 51$	dư	2	
$51:8 = 6$	dư	3	
$6:8 = 0$	dư	6	

Vậy  $(3287)_{10} = (6327)_8$

□ Phần lẻ:

$0,5100098 \times 8 = 4,0800784$	phần nguyên là 4	↓
$0,0800784 \times 8 = 0,6406272$	phần nguyên là 0	
$0,6406270 \times 8 = 5,1250176$	phần nguyên là 5	
$0,1250176 \times 8 = 1,0001408$	phần nguyên là 1	

Vậy  $(0,5100098)_{10} = (0,4051)_8$

Kết quả chung là:  $(3287,5100098)_{10} = (6327,4051)_8$



# Chuyển đổi hệ 2 sang hệ 10

---

**Ví dụ:** Chuyển đổi sang hệ Thập phân số:  **$m = 1101,011$**

Thực hiện: Ta lập tổng theo trọng số của từng Bit nhị phân:

$$m = 1.2^3 + 1.2^2 + 0.2^1 + 1.2^0 + 0.2^{-1} + 1.2^{-2} + 1.2^{-3}$$

$$m = 8 + 4 + 0 + 1 + 0 + 1/4 + 1/8$$

$$m = 13,375$$





**Ví dụ:** Chuyển số  $M = (574,321)_8$  sang biểu diễn nhị phân.

Thực hiện: Thay mỗi chữ số bằng nhóm nhị phân 3 bit tương ứng:

$$M = \begin{array}{ccc} 101 & 111 & 100 \\ 5 & 7 & 4 \end{array} , \begin{array}{ccc} 011 & 010 & 001 \\ 3 & 2 & 1 \end{array}$$

**Ví dụ:** Chuyển số  $M = (1001110,101001)_2$  sang cơ số 8.

$$\text{Thực hiện: } M = \begin{array}{ccc} 1 & 001 & 110 \\ & & & & 101 & 001 \end{array} ,$$

$$M = \begin{array}{ccc} 1 & 1 & 6 \\ & & & & 5 & 1 \end{array} ,$$

$$M = (116,51)_8$$



## □ Quy tắc chung (r: cơ số, n: số chữ số)

- Bù (r-1) của N =  $(r^n - 1) - N$
- Bù r của N =  $r^n - N$ 
  - Bù r của (bù r của N) = N
  - **Nhận xét:** Có tính chất giống  $-(-N) = N$

## □ Đối với hệ 10

- bù 9 của N = 9-từng ký số
  - VD: Bù 9 của 43520 là  $99999 - 43520 = 56479$
- Bù 10 của N = bù 9 + 1
  - VD: bù 10 của 43520 là  $56479 + 1 = 56480$
  - Mẹo: Bù 10 của 347200 là 652800





# Số bù (tt)

## □ Đối với hệ nhị phân:

– Bù 1 = đảo n bit của N

• Bù 1 của (1100) = 0011

– Bù 2 = bù 1 + 1

• Bù 2 của (1100) = 0011 + 1 = 0100

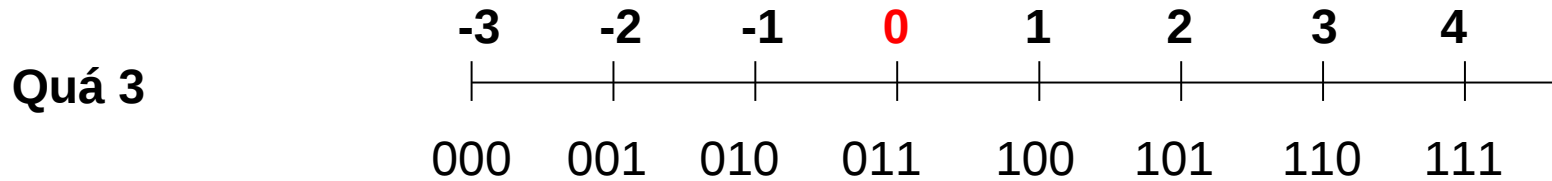
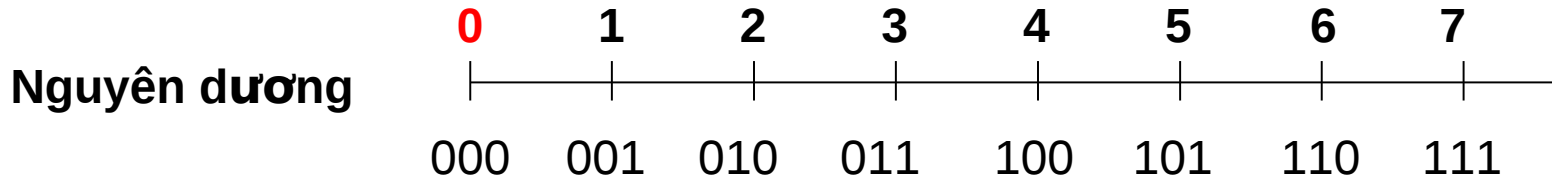
• Mẹo: giữ nguyên các số 0 bên phải cho đến khi gặp số 1, sau đó đảo

**1100**

**0100**



# Số quá n (excess-n)



**Quy tắc chung:**

Biểu diễn quá n của N = biểu diễn nguyên dương của (N + n)

**Ví dụ:**

Biểu diễn (quá 127) của 7 là:

$$127+7 = 134 = 10000110_2$$



# Phép cộng trừ nhị phân dùng bù 1

- ❑ Phép cộng giống như cộng các số nhị phân không dấu, cộng cả bit dấu.
- ❑ Cần lưu ý: Cộng số nhớ của bit lớn nhất vào bit cuối cùng
- ❑ Ví dụ:

13	001101	-13	110010
+	+	+	+
11	001011	-11	110100
-----	-----	-----	-----
24	011000	-24	Nhớ 1 100110
			+ 1
			-----
			100111

- ❑ Phép trừ thực hiện thông qua phép cộng



# Phép cộng trừ nhị phân dùng bù 2

- ❑ Quy tắc:  $-A = \text{bù 2 của } A$
- ❑  $A - B = A + (-B) = A + (\text{bù 2 của } B)$
- ❑ Ví dụ:  $13 - 6 = 13 + (-6)$

$$\begin{array}{r} 6 = 00000110 \\ -6 = 11111010 \\ \hline 13 = 00001101 \\ \hline = 100000111 \quad (7) \end{array}$$

↑  
Bỏ bit tràn (nếu có)

Số nhớ là 1 => kết quả là số dương, ta bỏ qua số nhớ không ghi trong số kết quả

- Nếu số nhớ cuối cùng là 1 thì số đó là số dương. Kết quả là những bit không kể đến bit nhớ cuối cùng đó.

- Nếu số nhớ cuối cùng bằng 0 (không có nhớ) thì số đó là số âm và dãy bit mới chỉ là bù 2 của kết quả. Muốn có kết quả thật ta lấy Bù 2 một lần nữa





# BCD (Binary Coded Decimal)

□ Biểu diễn một chữ số **thập phân** bằng 4 chữ số nhị phân (ít dùng)

$$0 = 0000$$

$$1 = 0001$$

...

$$9 = 1001$$

27	0010	0111	
+ 36	0011	0110	
<hr/>			
63	0101	<b>1101</b>	→ Ký số vượt quá => kết quả sai
	<u>0000</u>	0110	→ Sửa sai kết quả
	0110	0011	Kết quả = 63



# Ví dụ tính toán với BCD

$$\begin{array}{r} 28 \\ + 59 \\ \hline 87 \end{array} \quad \begin{array}{r} 0010 \ 1000 \\ 0101 \ 1001 \\ \hline 1000 \ 0001 \end{array} \longrightarrow \text{Có nhớ 1} \Rightarrow \text{kết quả sai}$$
$$\begin{array}{r} 0000 \ 0110 \\ 1000 \ 0111 \end{array} \longrightarrow \text{Sửa sai kết quả}$$

Kết quả = 87

$$\begin{array}{r} 61 \\ - 38 \\ \hline 23 \end{array} \quad \begin{array}{r} 0110 \ 0001 \\ 0011 \ 1000 \\ \hline 0010 \ 1001 \end{array} \longrightarrow \text{Ký số bên phải mượn 1 khi trừ}$$
$$\begin{array}{r} 0000 \ 0110 \\ 0010 \ 0011 \end{array} \longrightarrow \text{Sửa sai kết quả}$$

Kết quả = 23



# Biểu diễn ký tự

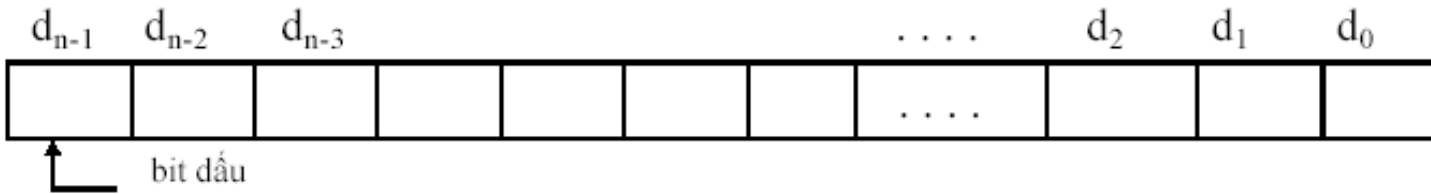
---

- ❑ Sử dụng bộ mã ASCII mở rộng (8 bit)
  - 00 – 1F: ký tự điều khiển
  - 20 – 7F: ký tự in được
  - 80 – FF: ký tự mở rộng (ký hiệu tiền tệ, vẽ khung, ...)
- ❑ Ngày nay dùng bộ mã Unicode (16 bit) (UTF-8)





# Biểu diễn số nguyên có dấu



$+2510 = \mathbf{0}0011001_2$

$-2510 = \mathbf{1}0011001_2$

Một Byte (8 bit) có thể biểu diễn các số có dấu từ -127 tới +127.

Có hai cách biểu diễn số không là 0000 0000 (+0) và 1000 0000 (-0).



# Biểu diễn chấm động

□  $F = (-1)^S \times M \times R^E$

- S: dấu
- M: định trị
- R: cơ số
- e: mũ

□ Ví dụ:  $2006 = (-1)^0 \times 2.006 \times 10^3$



# Biểu diễn chấm động

- Biểu diễn chấm động được gọi là chuẩn hóa khi phần định trị chỉ có duy nhất **một** chữ số bên trái dấu chấm thập phân và chữ số đó khác không  
→ một số chỉ có duy nhất một biểu diễn chấm động được chuẩn hóa.

$$2.006 \times 10^3 \text{ (chuẩn)}$$

$$20.06 \times 10^2 \text{ (không)}$$

$$0.2006 \times 10^4 \text{ (không)}$$



## Biểu diễn chấm động trên hệ nhị phân

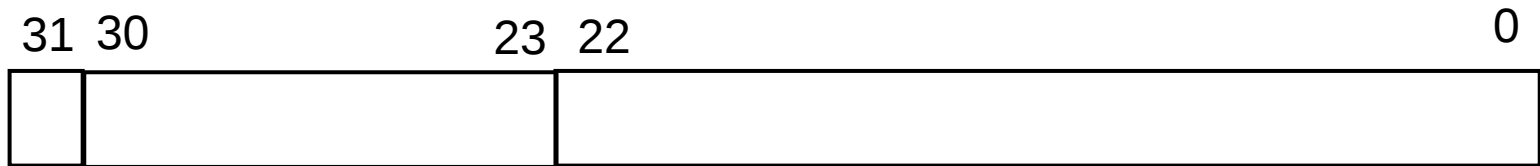
---

- Sử dụng dạng chuẩn hóa
- Dùng 1 bit cho phần dấu: 0-dương, 1-âm
- Không biểu diễn cơ số (R) vì luôn bằng 2
- Phần định trị **chỉ biểu diễn phần lẻ** (bên phải dấu chấm) vì chữ số bên trái dấu chấm luôn là 1



## Biểu diễn chấm động trên hệ nhị phân

□ Ví dụ:



- Dấu 1 bit
- Mũ: 8 bit (từ bit 23 đến bit 30) là một số quá 127 (sẽ có trị từ -127 đến 128)
- Định trị: 23 bit (từ bit 0 đến bit 22)





# CÂU HỎI VÀ BÀI TẬP CHƯƠNG III

---