

	VIETTEL AI RACE	TD014
	BIỂU DIỄN SỐ	Lần ban hành: 1

## 1. Biểu diễn dữ liệu trong máy tính

### 1.1 Biểu diễn số trong các hệ đếm

Hệ đếm là tập hợp các ký hiệu và qui tắc sử dụng tập ký hiệu đó để biểu diễn và xác định các giá trị các số. Mỗi hệ đếm có một số ký số (digits) hữu hạn. Tổng số ký số của mỗi hệ đếm được gọi là **cơ số** (base hay radix), ký hiệu là  $b$ .

#### 1.1.1 Hệ đếm cơ số $b$

Hệ đếm cơ số  $b$  ( $b \geq 2$  và nguyên dương) mang tính chất sau :

- Có  $b$  ký số để thể hiện giá trị số. Ký số nhỏ nhất là **0** và lớn nhất là  **$b-1$** .
- Giá trị vị trí thứ  $n$  trong một số của hệ đếm bằng cơ số  $b$  lũy thừa  $n$
- Số  $N_{(b)}$  trong hệ đếm cơ số  $(b)$  được biểu diễn bởi:

$$N_{(b)} = a_n a_{n-1} a_{n-2} \dots a_1 a_0 a_{-1} a_{-2} \dots a_{-m}$$

trong đó, số  $N_{(b)}$  có  $n+1$  ký số biểu diễn cho phần nguyên và  $m$  ký số lẻ biểu diễn cho phần  $b$  phân, và có giá trị là:

$$N_{(b)} = a_n \cdot b^n + a_{n-1} \cdot b^{n-1} + a_{n-2} \cdot b^{n-2} + \dots + a_1 \cdot b^1 + a_0 \cdot b^0 + a_{-1} \cdot b^{-1} + a_{-2} \cdot b^{-2} + \dots + a_{-m} \cdot b^{-m}$$

hay là:

$$N_{(b)} = \sum_{i=-m}^n a_i \cdot b^i$$

Trong ngành toán - tin học hiện nay phổ biến 4 hệ đếm là hệ thập phân, hệ nhị phân, hệ bát phân và hệ thập lục phân.

#### 1.1.2 Hệ đếm thập phân (Decimal system, $b=10$ )

Hệ đếm thập phân hay hệ đếm cơ số 10 là một trong các phát minh của người Ả rập cổ, bao gồm 10 ký số theo ký hiệu sau:

**0,1,2,3,4,5,6,7,8,9**

Qui tắc tính giá trị của hệ đếm này là mỗi đơn vị ở một hàng bất kỳ có giá trị bằng 10 đơn vị của hàng kế cận bên phải. Ở đây  $b=10$ . Bất kỳ số nguyên dương trong hệ thập phân có thể biểu diễn như là một tổng các số hạng, mỗi số hạng là tích của một số với 10 lũy thừa, trong đó số mũ lũy thừa được tăng thêm 1 đơn vị kể từ số mũ lũy thừa phía bên phải nó. Số mũ lũy thừa của hàng đơn vị trong hệ thập phân là 0.

Ví dụ: Số 5246 có thể được biểu diễn như sau:

3                  2                  1                  0

	<b>VIETTEL AI RACE</b>	TD014
	<b>BIỂU DIỄN SỐ</b>	Lần ban hành: 1

$$5246 = 5 \times 10 + 2 \times 10 + 4 \times 10 + 6 \times 10$$

$$= 5 \times 1000 + 2 \times 100 + 4 \times 10 + 6 \times 1$$

Thể hiện như trên gọi là ký hiệu mở rộng của số nguyên vì

$$5246 = 5000 + 200 + 40 + 6$$

Như vậy, trong số 5246 : ký số 6 trong số nguyên đại diện cho giá trị 6 đơn vị (1s), ký số 4 đại diện cho giá trị 4 chục (10s), ký số 2 đại diện cho giá trị 2 trăm (100s) và ký số 5 đại diện cho giá trị 5 ngàn (1000s). Nghĩa là, số lũy thừa của 10 tăng dần 1 đơn vị từ trái sang phải tương ứng với vị trí ký hiệu số,

$$\begin{array}{ccccccc} 0 & 1 & 2 & 3 & 4 & & \\ 10^0 = 1 & 10^1 = 10 & 10^2 = 100 & 10^3 = 1000 & 10^4 = 10000 & \dots & \end{array}$$

Mỗi ký số ở thứ tự khác nhau trong số sẽ có giá trị khác nhau, ta gọi là giá trị vị trí (place value).

Phần thập phân trong hệ thập phân sau dấu chấm phân cách thập phân (theo qui ước của Mỹ) thể hiện trong ký hiệu mở rộng bởi 10 lũy thừa âm tính từ phải sang trái kể từ dấu chấm phân cách:

$$101101 = 1011002 = 10110003 = \dots$$

$$\text{Ví dụ: } 254.68 = 2 \times 10^2 + 5 \times 10^1 + 4 \times 10^0 + 6 \times 10^{-1} + 8 \times 10^{-2}$$

### 1.1.3 Hệ đếm nhị phân (Binary system, b=2)

Với cơ số b=2, chúng ta có hệ đếm nhị phân. Đây là hệ đếm đơn giản nhất với 2 chữ số là 0 và 1, mỗi chữ số nhị phân gọi là BIT (viết tắt từ chữ BInary digiT). Vì hệ nhị phân chỉ có 2 trị số là 0 và 1, nên khi muốn diễn tả một số lớn hơn, hoặc các ký tự phức tạp hơn thì cần kết hợp nhiều bit với nhau. Ta có thể chuyển đổi số trong hệ nhị phân sang số trong hệ thập phân quen thuộc.

Ví dụ: Số  $11101.11_{(2)}$  sẽ tương đương với giá trị thập phân là :

							<u>vị trí dấu chấm cách</u>
Số nhị phân :	1	1	1	0	1	.	1 1
Số vị trí :	4	3	2	1	0	-1	-2
Trị vị trí :	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$
Hệ 10 là :	16	8	4	2	1	0.5	0.25

như vậy:

$$11101.11_{(2)} = 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 + 1 \times 0.5 + 1 \times 0.25 = 29.75_{(10)}$$

số 10101 (hệ 2) sang hệ thập phân sẽ là:

$$10101_{(2)} = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 16 + 0 + 4 + 0 + 1 = 21_{(10)}$$

	<b>VIETTEL AI RACE</b>	TD014
	<b>BIỂU DIỄN SỐ</b>	Lần ban hành: 1

#### 1.1.4 Hệ đếm bát phân (Octal system, b=8)

Nếu dùng 1 tập hợp 3 bit thì có thể biểu diễn 8 trị khác nhau : 000, 001, 010, 011, 100, 101, 110, 111. Các trị này tương đương với 8 trị trong hệ thập phân là 0, 1, 2, 3, 4, 5, 6, 7. Tập hợp các chữ

số này gọi là hệ bát phân, là hệ đếm với  $b = 8 = 2^3$ . Trong hệ bát phân, trị vị trí là lũy thừa của 8.

Ví dụ:

$$235.64_{(8)} = 2 \times 8^2 + 3 \times 8^1 + 5 \times 8^0 + 6 \times 8^{-1} + 4 \times 8^{-2} = 157.8125_{(10)}$$

#### 1.1.5 Hệ đếm thập lục phân (Hexa-decimal system, b=16)

Hệ đếm thập lục phân là hệ cơ số  $b=16 = 2^4$ , tương đương với tập hợp 4 chữ số nhị phân (4 bit). Khi thể hiện ở dạng hexa-decimal, ta có 16 ký tự gồm 10 chữ số từ 0 đến 9, và 6 chữ in A, B, C, D, E, F để biểu diễn các giá trị số tương ứng là 10, 11, 12, 13, 14, 15. Với hệ thập lục phân, trị vị trí là lũy thừa của 16.

Ví dụ:

$$34F5C_{(16)} = 3 \times 16^4 + 4 \times 16^3 + 15 \times 16^2 + 5 \times 16^1 + 12 \times 16^0 = 216294_{(10)}$$

*Ghi chú:* Một số ngôn ngữ lập trình qui định viết số hexa phải có chữ H ở cuối chữ số. Ví dụ: Số 15 viết là F<sub>H</sub>.

#### 1.1.6 Chuyển đổi một số từ hệ thập phân sang hệ đếm cơ số b

##### 1.1.6.1. Đổi phần nguyên từ hệ thập phân sang hệ b

Tổng quát: Lấy số nguyên thập phân  $N_{(10)}$  lần lượt chia cho b cho đến khi thương số bằng 0. Kết

quả số chuyển đổi  $N_{(b)}$  là các dư số trong phép chia viết ra theo thứ tự ngược lại.. Ví dụ:

Số 12<sub>(10)</sub>

= ?<sub>(2)</sub>. Dùng phép chia cho 2 liên tiếp, ta có một loạt các số dư như sau:

	VIETTEL AI RACE	TD014
	BIỂU DIỄN SỐ	Lần ban hành: 1

Ví dụ 3.11:  $0.6875_{(10)} = ?_{(2)}$

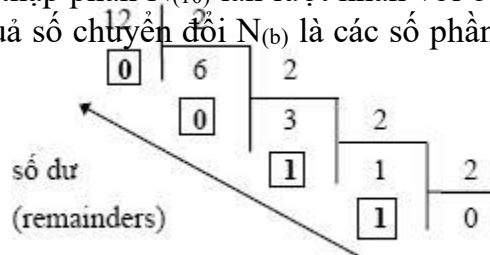
$$\begin{array}{rcl}
 0.6875 & \times 2 & = 1.375 \\
 0.3750 & \times 2 & = 0.75 \\
 0.75 & \times 2 & = 1.5 \\
 0.5 & \times 2 & = 1.0
 \end{array}$$

phần nguyên của tích  
phần thập phân của tích

Kết quả:  $0.6875_{(10)} = 0.1011_{(2)}$

### 1.1.6.2. Đổi phần thập phân từ hệ thập phân sang hệ cơ số b

Tổng quát: Lấy phần thập phân  $N_{(10)}$  lần lượt nhân với b cho đến khi phần thập phân của tích số bằng 0. Kết quả số chuyển đổi  $N_{(b)}$  là các số phần nguyên trong phép nhân viết ra theo thứ tự tính toán.



Kết quả:  $12_{(10)} = 1100_{(2)}$

## 1.2 Biểu diễn dữ liệu trong máy tính và đơn vị thông tin

### 1.2.1 Nguyên tắc chung

Thông tin và dữ liệu mà con người hiểu được tồn tại dưới nhiều dạng khác nhau, ví dụ như các số liệu, các ký tự văn bản, âm thanh, hình ảnh,... nhưng trong máy tính mọi thông tin và dữ liệu đều được biểu diễn bằng số nhị phân (chuỗi bit).

Để đưa dữ liệu vào cho máy tính, cần phải mã hoá nó về dạng nhị phân. Với các kiểu dữ liệu khác nhau cần có cách mã hoá khác nhau. Cụ thể:

- Các dữ liệu dạng số (số nguyên hay số thực) sẽ được chuyển đổi trực tiếp thành các chuỗi số nhị phân theo các chuẩn xác định.
- Các ký tự được mã hoá theo một bộ mã cụ thể, có nghĩa là mỗi ký tự sẽ tương ứng với một chuỗi số nhị phân.
- Các dữ liệu phi số khác như âm thanh, hình ảnh và nhiều đại lượng vật lý khác muốn đưa vào máy phải **số hoá (digitalizing)**. Có thể hiểu một cách đơn giản khái niệm số hoá như sau: các dữ liệu tự nhiên thường là quá trình biến đổi liên tục, vì vậy để đưa vào máy tính, nó cần được biến đổi sang một dãy hữu hạn các giá trị số (nguyên hay thực) và được biểu diễn dưới dạng nhị phân.

	<b>VIETTEL AI RACE</b>	TD014
	<b>BIỂU DIỄN SỐ</b>	Lần ban hành: 1

Tuy rằng mọi dữ liệu trong máy tính đều ở dạng nhị phân, song do bản chất của dữ liệu, người ta thường phân dữ liệu thành 2 dạng:

- **Dạng cơ bản:** gồm dạng số (nguyên hay thực) và dạng ký tự. Số nguyên không dấu được biểu diễn theo dạng nhị phân thông thường, số nguyên có dấu theo mã bù hai, còn số thực theo dạng dấu phẩy động. Để biểu diễn một dữ liệu cơ bản, người ta sử dụng 1 số bit. Các bit này ghép lại với nhau để tạo thành từ: từ 8 bit, từ 16 bit,...
- **Dạng có cấu trúc:** Trên cơ sở dữ liệu cơ bản, trong máy tính, người ta xây dựng nên các dữ liệu có cấu trúc phục vụ cho các mục đích sử dụng khác nhau. Tùy theo cách “ghép” chúng ta có mảng, tập hợp, xâu, bản ghi,...

### 1.2.2 Đơn vị thông tin

Đơn vị nhỏ nhất để biểu diễn thông tin gọi là **bit**. Một bit tương ứng với một sự kiện có 1 trong 2 trạng thái.

Ví dụ: Một mạch đèn có 2 trạng thái là:

- Tắt (Off) khi mạch điện qua công tắc là hở
- Mở (On) khi mạch điện qua công tắc là đóng

Số học nhị phân sử dụng hai ký số 0 và 1 để biểu diễn các số. Vì khả năng sử dụng hai số 0 và 1 là như nhau nên một chỉ thị chỉ gồm một chữ số nhị phân có thể xem như là đơn vị chứa thông tin nhỏ nhất.

Bit là chữ viết tắt của **BI**nary digi**T**. Trong tin học, người ta thường sử dụng các đơn vị đo thông tin lớn hơn như sau:

## 1.3 Biểu diễn số nguyên

Số nguyên gồm số nguyên không dấu và số nguyên có dấu. Về nguyên tắc đều dùng 1 chuỗi bit để biểu diễn. Đối với số nguyên có dấu, người ta sử dụng bit đầu tiên để biểu diễn dấu „-“, và bit này gọi là bit dấu.

### 1.3.1 Số nguyên không dấu

Trong biểu diễn số nguyên không dấu, mọi bit đều được sử dụng để biểu diễn giá trị số. Ví dụ 1 dãy 8 bit biểu diễn số nguyên không dấu có giá trị:

8

$2^8 = 256$  số nguyên dương, cho giá trị từ 0 (0000 0000) đến 255 (1111 1111).

Với n bits ta có thể biểu diễn 1 số nguyên có giá trị lớn nhất là  $2^n - 1$  và dải giá trị biểu diễn được từ 0 đến  $2^n - 1$ .

Thí dụ: 00000000 = 0

	<b>VIETTEL AI RACE</b>	TD014
	<b>BIỂU DIỄN SỐ</b>	Lần ban hành: 1

$$00000010 = 2$$

$$00000100 = 4$$

.....

$$11111111 = 255$$

### 1.3.2 Số nguyên có dấu

Trong biểu diễn số nguyên có dấu, bit đầu làm bit dấu: 0 là số dương và 1 cho số âm. Số nguyên có dấu thể hiện trong máy tính ở dạng nhị phân là số dùng 1 bit làm bit dấu, người ta qui ước dùng bit ở hàng đầu tiên bên trái làm bit dấu (S): 0 là số dương và 1 cho số âm. Cách phổ biến biểu diễn số âm có dấu là dùng mã bù hai:

Số bù hai được tính như sau:

- Biểu diễn số nguyên không dấu
- Nghịch đảo tất cả các bit (số bù một)
- Cộng thêm một. (số bù hai)

Chú ý: Thử biểu diễn mã bù hai của -37 để thu được số +35

### 1.3.3 Tính toán số học với số nguyên

#### 1.3.3.1. Cộng/ trừ số nguyên

##### Cộng/ trừ số nguyên không dấu

Khi cộng hai số nguyên không dấu n bits ta thu được một số nguyên không dấu cũng n bits. Vì vậy,

- Nếu tổng của hai số đó nhỏ hơn hoặc bằng  $2^n - 1$  thì kết quả nhận được là đúng.
- Nếu tổng của hai số đó lớn hơn  $2^n - 1$  thì khi đó sẽ tràn số và kết quả sẽ là sai. Thí dụ với trường hợp 8 bits, tổng nhỏ hơn 255 thì ta sẽ có kết quả đúng:

$$\begin{array}{r}
 + \quad 57 = 00111001 \\
 \quad 34 = 00100010 \\
 \hline
 \end{array}$$

$$91 = 01011011$$

$$\begin{array}{r}
 + \quad 209 = 11010001 \\
 \quad 73 = 01001001 \\
 \hline
 \end{array}$$

$$282 = 100011010$$



Bit tràn ra ngoài  $\Rightarrow$  kết quả = 26 là sai.

	<b>VIETTEL AI RACE</b>	TD014
	<b>BIỂU DIỄN SỐ</b>	Lần ban hành: 1

- Để tránh hiện tượng tràn số này ta phải sử dụng nhiều bit hơn để biểu diễn.

#### *Cộng/trừ số nguyên có dấu*

Số nguyên có dấu được biểu diễn theo mã bù hai, vậy qui tắc chung như sau:

- Cộng hai số nguyên có dấu n-bit sẽ bỏ qua giá trị nhớ ra khỏi bit có ý nghĩa cao nhất, tổng nhận được sẽ có giá trị đúng và cũng được biểu diễn theo mã bù hai, nếu kết quả nhận được nằm trong dải  $-2^{n-1}$  đến  $+2^{n-1} - 1$ .
- Để trừ hai số nguyên có dấu X và Y ( $X - Y$ ), cần lấy bù hai của Y tức  $-Y$ , sau đó cộng X với  $-Y$  theo nguyên tắc trên.

Như vậy, khi thực hiện phép tính trên sẽ thừa ra 1 bit bên trái cùng, bit này sẽ không được lưu trong kết quả và sẽ được bỏ qua.

#### *1.3.3.2. Nhân/ chia số nguyên*

So với phép cộng và phép trừ, phép nhân và phép chia phức tạp hơn nhiều. Dưới đây, chỉ giới thiệu phép nhân/phép chia với số nhị phân. Ví dụ sau mô tả phép nhân hai số nhị phân:

$$\begin{array}{r}
 \begin{array}{cc} 1011 & (11 \text{ cơ số } 10) \\ \times 1101 & (13 \text{ cơ số } 10) \\ \hline
 \end{array} \\
 \begin{array}{r}
 1011 \\
 0000 \\
 1011 \\
 1011 \\
 \hline
 \end{array}
 \end{array}$$

**10001111      kết quả 143 trong cơ số 10**

Chúng ta có một số nhận xét sau:

- Phép nhân tạo ra các tích riêng, mỗi tích thu được là kết quả của việc nhân từng bit.
- Các tích riêng dễ dàng xác định theo qui tắc:
  - Bit tương ứng số nhân là 1 thì tích riêng bằng số bị nhân
  - Bit tương ứng số nhân bằng 0 thì tích riêng bằng 0
- Tích được tính bằng tổng các tích riêng.

Phép chia phức tạp hơn phép nhân nhưng dựa trên cùng 1 nguyên tắc.