CHƯƠNG 3:
BIỂU DIỄN DỮ LIỆU VÀ SỐ HỌC

Nội dung môn học

- Chương 1: Giới thiệu chung
- Chương 2: Tổng quan về hệ thống máy tính
- Chương 3: Biểu diễn dữ liệu và số học
- Chương 4: Đơn vị xử lý trung tâm
- Chương 5: Bộ nhớ máy tính
- Chương 6: Hệ thống vào ra
- Tổng kết – ôn tập

Nội dung

- Các hệ thống số
- Biểu diễn dữ liệu trong máy tính
- Biểu diễn số nguyên
- Thực hiện các phép toán số học với số nguyên
- Biểu diễn số thực
- Thực hiện các phép toán số học với số thực
- Biểu diễn ký tự

Các hệ thống số

- Hệ thập phân (Decimal System)
  - Con người sử dụng
- Hệ nhị phân (Binary System)
  - Máy tính sử dụng
- Hệ thập lục phân (Hexadecimal System)
  - Dùng để viết gọn số nhị phân
- Hệ bát phân(Octal System)
  - Dùng để viết gọn số nhị phân

Các hệ thống số

- Một số khái niệm cơ bản
  - Cơ số (R- Radix)
    - Là số lượng ký tự chữ số (ký số - digit) sử dụng để biểu diễn trong hệ thống số đếm
  - Trọng số (weight)
    - Đại lượng biểu diễn cho vị trí của một con số trong hệ thống số đếm
    - **Trọng số = cơ số** $^{\text{vị trí}}$
  - Giá trị (value)
    - Tính bằng tổng theo trọng số
    - Value= Σ(ký số x trọng số)

Hệ thập phân

- Cơ số r=10
  - 10 chữ số: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- Dùng n chữ số thập phân có thể biểu diễn được 10^n giá trị khác nhau:

0	0	...	0	0	=	0
0	0	...	0	1	=	1
9	9	...	9	9	=	10^n-1

$\longleftrightarrow$

n số

Hệ thập phân

Thập phân

10

Ký số

0	1	2	3	4
5	6	7	8	9

Quy tắc đếm

0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9

10 → 11 → 12 → 13 → 14 → 15 → 16 → 17 → 18 → 19

...

90 → 91 → 92 → 93 → 94 → 95 → 96 → 97 → 98 → 99

100 → 101 → 102 → 103 → 104 → 105 → 106 → 107 → 108 → 109

Hệ thập phân

- Tính giá trị số thập phân
  - Số thập phân $D=d_n d_{n-1}..d_0.d_{-1}d_{-2}..d_{-m}$
  - $D=d_n x10^n+d_{n-1}x10^{n-1}+...d_0x10^0+d_{-1}x10^{-1}+d_{-2}*10^{-2}+..d_{-m}x10^{-m}$

4	**0**	**7**	**.**	**6**	**2**	**5**
10^2	10^1	10^0	.	10^{-1}	10^{-2}	10^{-3}
$4x10^2$	$0x10^1$	$7x10^0$	.	$6x10^{-1}$	$2x10^{-2}$	$5x10^{-3}$
400	0	7	.	0.6	0.02	0.005

400 + 0 + 7 + 0.6 + 0.02 + 0.005 = *407.625*

Hệ thập phân

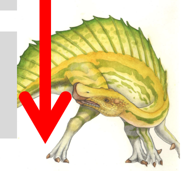
4	0	7	.	6	2	5

Các chữ số phần nguyên: phần nguyên chia 10 lấy dư

407	:	10	=	40	dư	7
40	:	10	=	4	dư	0
4	:	10	=	0	dư	4

Các chữ số thập phân: phần tp x 10 lấy phần nguyên

0.625	x	10	=	6.25	nguyên	6
0.25	x	10	=	2.5	nguyên	2
0.5	x	10	=	5	nguyên	5



Hệ nhị phân

- Cơ số r=2
  - 2 chữ số nhị phân: 0 và 1
- Chữ số nhị phân gọi là ***bit*** (binary digit)
  - Bit là đơn vị thông tin nhỏ nhất
- Dùng n bit có thể biểu diễn được 2^n giá trị khác nhau:

0	0	...	0	0	=	0
0	0	...	0	1	=	1
1	1	...	1	1	=	2^n-1

n số

Hệ nhị phân

Nhị phân

2

Ký số

0	1

Quy tắc đếm

0 → 1

10 → 11

100 → 101 → 110 → 111

Hệ nhị phân

- Tính giá trị số nhị phân
  - Số thập phân $B=b_nb_{n-1}..b_0.b_{-1}b_{-2}..b_{-m}$
  - $B=b_nx2^n+b_{n-1}x2^{n-1}+...b_0x2^0+b_{-1}x2^{-1}+b_{-2}*2^{-2}+..b_{-m}x2^{-m}$

1	0	1	.	0	1	1
2^2	2^1	2^0	.	2^{-1}	2^{-2}	2^{-3}
$1x2^2$	$0x2^1$	$1x2^0$	.	$0x2^{-1}$	$1x2^{-2}$	$1x2^{-3}$
4	0	1	.	0	0.25	0.125

4 + 0 + 1 + 0 + 0.25 + 0.125 = *5.375*

Hệ thập lục phân

- Cơ số r=16
  - 16 chữ số: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
- Dùng n bit có thể biểu diễn được 16^n giá trị khác nhau:

0 0 ... 0 0 = 0

0 0 ... 0 1 = 1

F F ... F F = 16^n-1

$\longleftrightarrow$

n số

Hệ thập lục phân

Thập lục phân

16

Ký số

0	1	2	3	4	5	6	7
8	9	A	B	C	D	E	F

Quy tắc đếm

0 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → A → B → C → D → E → F

10 → 11 → 12 → 13 → 14 → 15 → 16 → 17 → 18 → 19 → 1A → 1B → 1C → 1D → 1E → 1F

...

F0 → F1 → F2 → F3 → F4 → F5 → F6 → F7 → F8 → F9 → FA → FB → FC → FD → FE → FF

100 → 101 → 102 → 103 → 104 → 105 → 106 → 107 → 108 → 109 ...

Hệ thập lục phân

- Tính giá trị số thập lục phân quy đổi về thập phân
  - Số thập lục phân $H=h_n h_{n-1}..h_0.h_{-1}h_{-2}..h_{-m}$
  - $H=h_n x16^n+h_{n-1}x16^{n-1}+...h_0 x16^0+h_{-1}x16^{-1}+h_{-2}*16^{-2}+..h_{-m}x16^{-m}$

5	A	0	.	4	D	1
16^2	16^1	16^0	.	16^{-1}	16^{-2}	16^{-3}
$5x16^2$	$10x16^1$	$0x16^0$	.	$4x16^{-1}$	$13x16^{-2}$	$1x16^{-3}$
1280	160	0	.	0.25	0.0508	0.0002

1280 + 160 + 0 + 0.25 + 0.0508 + 0.0002 = *1440.301*

Chuyển đổi qua lại giữa các hệ thống số

- Chuyển từ số thập phân sang số nhị phân
- Từ số thập phân sang thập lục phân
- Từ số nhị phân sang số thập phân
- Từ số thập lục phân sang số nhị phân và ngược lại

Chuyển từ số thập phân sang số nhị phân

8 . 625

8 : 2 = 4 *dư* 0 *(LSB)*

4 : 2 = 2 *dư* 0

2 : 2 = 1 *dư* 0

1 : 2 = 0 *dư* 1

1 0 0 0 . 1 0 1 B

0.625 x 2 = 1.25 *Phần nguyên* 1 *(MSB)*

0.25 x 2 = 0.5 *Phần nguyên* 0

0.5 x 2 = 1.0 *Phần nguyên* 1

Chuyển từ số thập phân sang số Hex

1 4 8 0 . 4 2 9 6 8 7 5

1480 : 16 = 92 *dư* 8 *(LSB)*

92 : 16 = 5 *dư* 12

5 : 16 = 0 *dư* 5

<u>5 C 8 . 6 E H</u>

0.4296875 x 16 = 6.875 *Phần nguyên* 6 *(MSB)*

0.875 x 16 = 14.0 *Phần nguyên* 14

Chuyển từ số thập lục phân sang nhị phân

0 0 1 1 1 0 1 1 0 1 0 1 1 1 0 1 . 0 1 1 0 1 0 1 0 B

3 B 5 D . 6 A

2C9.E8 H

0 0 1 0 1 1 0 0 1 0 0 1 . 1 1 1 0 1 0 0 0

Nội dung

- Các hệ thống số
- Biểu diễn dữ liệu trong máy tính
- Biểu diễn số nguyên
- Thực hiện các phép toán số học với số nguyên
- Biểu diễn số thực
- Thực hiện các phép toán số học với số thực
- Biểu diễn ký tự

Mã hóa dữ liệu

- Thông tin và dữ liệu mà con người hiểu được tồn tại dưới nhiều dạng khác nhau, ví dụ như các số, các ký tự văn bản, âm thanh, hình ảnh...
- Trong máy tính mọi thông tin và dữ liệu được biểu diễn bằng số nhị phân (chuỗi bit). Vì vậy, dữ liệu trước khi được lưu trữ trong máy tính đều được mã hóa thành mã nhị phân.
- Việc mã hóa dữ liệu phụ thuộc vào phân loại của dữ liệu. Nhưng thông thường là gán một số lượng bit mẫu đến mỗi đối tượng. Do có nhiều loại dữ liệu nên cũng sẽ có nhiều chuẩn mã hóa.

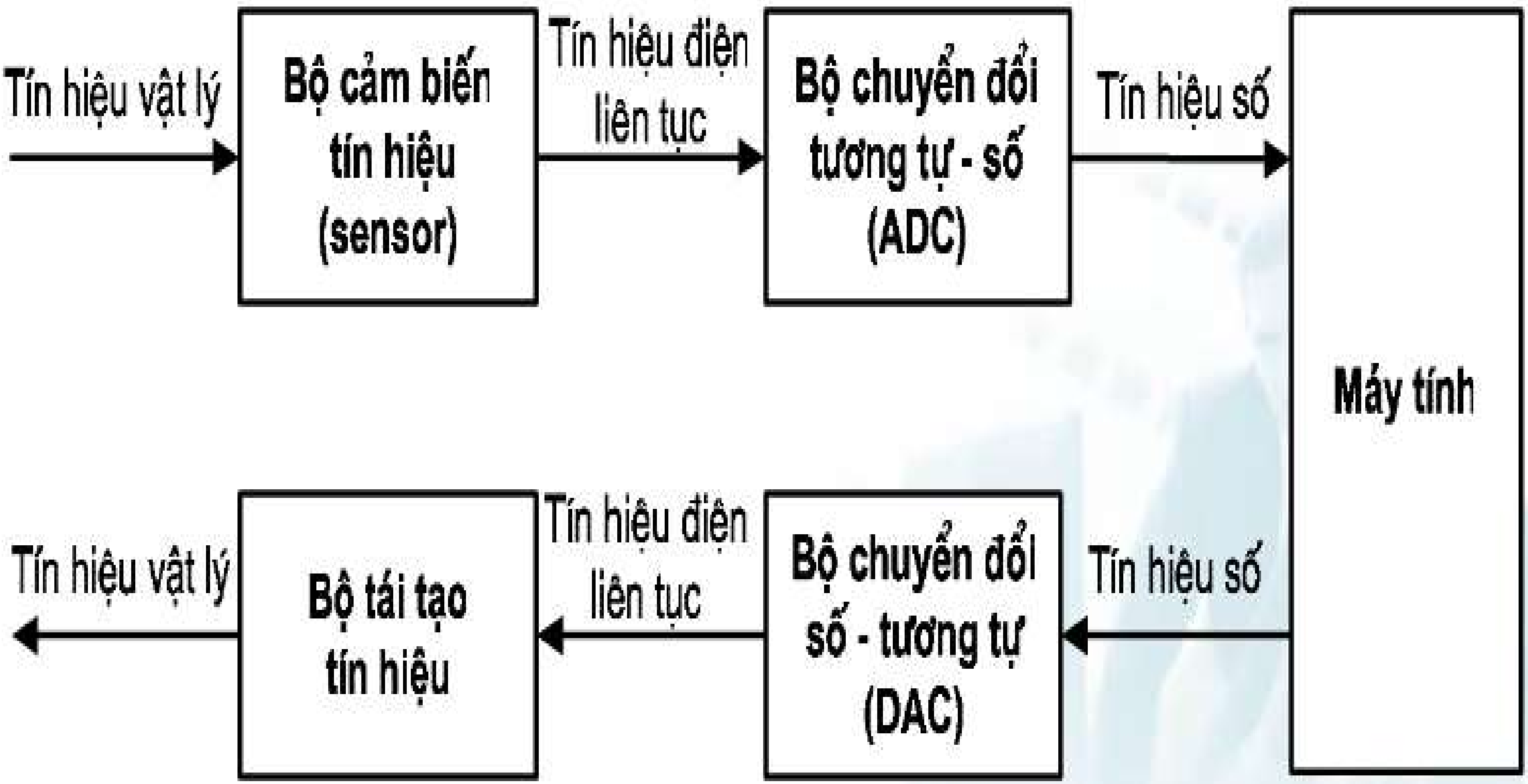
Mã hóa dữ liệu

- Các loại dữ liệu
  - Dữ liệu nhân tạo: do con người quy ước
    - Dữ liệu số nguyên
    - Dữ liệu số thực
    - Dữ liệu ký tự
  - Dữ liệu tự nhiên:
    - Tồn tại khách quan với con người.
    - Phổ biến là các tín hiệu vật lý như âm thanh, hình ảnh,…

Các nguyên tắc mã hóa dữ liệu

- Dữ liệu số nguyên: mã hoá theo một số chuẩn qui ước
- Dữ liệu số thực: mã hoá bằng số dấu chấm động
- Dữ liệu ký tự: mã hoá theo bộ mã ký tự
- Dữ liệu tự nhiên:
  - Các dữ liệu cần phải được số hóa trước khi đưa vào máy tính lưu trữ.
  - Sơ đồ mã hóa và tái tạo tín hiệu vật lý

Sơ đồ mã hóa và tái tạo tín hiệu



Ví dụ mã hóa cho 24 chữ cái

Các phép toán trên số nhị phân

- Phép cộng
  - 0 + 0 = 0
  - 0 + 1 = 1
  - 1 + 0 = 1
  - 1 + 1 = 0 nhớ 1
- Phép trừ
  - 0 - 0 = 0
  - 0 - 1 = 1 mượn 1
  - 1 - 0 = 1
  - 1 - 1 = 0

```
   1 1 1 1
   1 1 0 1 1 1 0
+      1 1 1 0 1
-----------------
 1 0 0 0 1 0 1 1
```

```
    -1       -1
   1 1 0 1 1 1 0
-      1 1 1 0 1
-----------------
     1 0 1 0 0 0 1
```

Các phép toán trên số nhị phân

- Phép nhân
  - 0 x 0 = 0
  - 0 x 1 = 0
  - 1 x 0 = 0
  - 1 x 1 = 1

$$
\begin{array}{cccccccc}
 & & & & 1 & 0 & 1 & 1 \\
\times & & & & 1 & 0 & 0 & 1 \\
\hline
 & & & & 1 & 0 & 1 & 1 \\
+ & & & 0 & 0 & 0 & 0 & \\
 & & 0 & 0 & 0 & 0 & & \\
 & 1 & 0 & 1 & 1 & & & \\
\hline
 & 1 & 1 & 0 & 0 & 0 & 1 & 1
\end{array}
$$

Các phép toán trên số nhị phân

- Phép chia

```
  1 0 0 1 0 0 0 1 | 1 0 1 1
−   1 0 1 1       |--------
  ---------       |  1 1 0 1
      1 1 1 0     |
    −  1 0 1 1
      ---------
         1 1 0 1
       − 1 0 1 1
         -------
             1 0
```

Nội dung

- Các hệ thống số
- Biểu diễn dữ liệu trong máy tính
- Biểu diễn số nguyên
- Thực hiện các phép toán số học với số nguyên
- Biểu diễn số thực
- Thực hiện các phép toán số học với số thực
- Biểu diễn ký tự

Số nguyên

- Có hai loại số nguyên:
  - Số nguyên không dấu (Unsigned Integer)
  - Số nguyên có dấu (Signed Integer)
- Biểu diễn số nguyên không dấu
  - Dùng n bit biểu diễn số nguyên không dấu A:

    $a_{n-1}a_{n-2}\ldots.a_2a_1a_0$
  - Giá trị của A được tính như sau: $A = \sum_{i=0}^{n-1} a_i 2^i$
  - Dải biểu diễn của A: $0 \text{ -:- } 2^n - 1$
    - Số 8 bit có giá trị : 0 ÷ 255
    - Số 16 bit có giá trị : 0 ÷ 65 535
    - Số 32 bit có giá trị : 0 ÷ 4 294 967 295

Số nguyên

- Biểu diễn được các giá trị từ 0 đến 255
  - 0000 0000 = 0
  - 0000 0001 = 1
  - 0000 0010 = 2
  - 0000 0011 = 3
  - …
  - 1111 1111 = 255

Chú ý:

  1111 1111
+ 0000 0001
1 0000 0000

Vậy: 255 + 1 = 0?

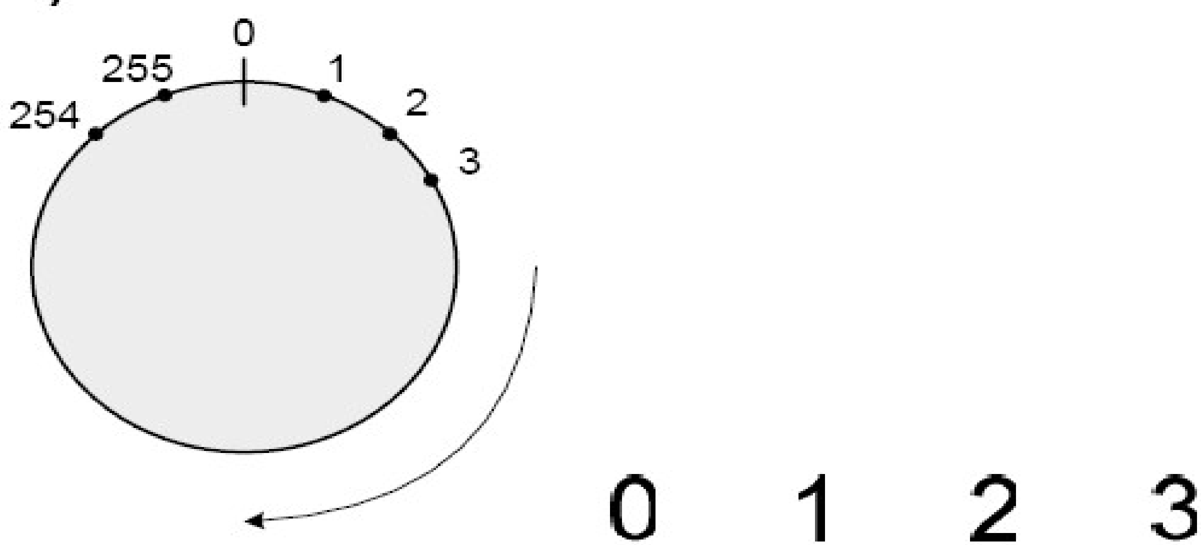
→ do tràn nhớ ra ngoài

Số nguyên

Trục số học số nguyên không dấu với n = 8 bit

- Trục số học:

- Trục số học máy tính:



Số nguyên có dấu

- Dạng tổng quát của số nguyên có dấu A:

  $a_{n-1}a_{n-2}...a_2a_1a_0$

- Qui ước: chọn bit có trọng số cao nhất (MSB) làm bit dấu
  - 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: bit $a_{n-1} = 1$
- Giá trị của A được tính theo công thức sau:

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

Số nguyên có dấu (tt)

- Dải biểu diễn của A: -2^{n-1} -:- 2^{n-1} - 1

  1 0 0 0 ... 0 0 0 0

  1 0 0 0 ... 0 0 0 1

  ...

  0 1 1 1 ... 1 1 1 1

Số nguyên có dấu (tt)

- Số nguyên dương
  - Dạng tổng quát của số dương:

$$0a_{n-2}\ldots a_2a_1a_0$$

  - Giá trị của số dương: $A = \sum_{i=0}^{n-2} a_i 2^i$
  - Dải biểu diễn của số dương: 0 -:- 2^{n-1} - 1

Số nguyên có dấu (tt)

- Số nguyên âm
  - Bit $a_{n-1} = 1$
  - Dạng tổng quát của số âm:

$$\color{red}{1}a_{n-2}\ldots a_2a_1a_0$$

  - Giá trị của số dương: $A = -2^{n-1} + \sum_{i=0}^{n-2} a_i 2^i$
  - Dải biểu diễn của số dương: -2^{n-1} -:- -1

Số nguyên có dấu (tt)

- Ví dụ: xác định giá trị các số nguyên có dấu 8 bit sau đây:

  A = 0101 1110

  B = 1101 0011

- A = $2^6 + 2^4 + 2^3 + 2^2 + 2^1 = 64 + 16 + 8 + 4 + 2 = 94$
- B = $-2^7 + 2^6 + 2^4 + 2^1 + 2^0 = -128 + 64 + 16 + 2 + 1$

  $= -45$

Số nguyên có dấu (tt)

- Ví dụ: xác định giá trị các số nguyên có dấu 16 bit sau đây:

  A = 0101 1110 0101 1110

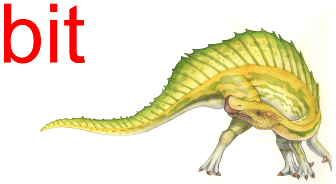
  B = 1101 0011 1101 0011

Số nguyên có dấu (tt)

- Số bù một và Số bù hai:
  - Giả sử có một số nguyên (không dấu) nhị phân A được biểu diễn bởi n bit. Khi đó ta có:
  - Số bù một của A = $(2^n-1) – A$
  - Số bù hai của A = $2^n – A$

  → Số bù hai = Số bù một + 1
  - Ví dụ:
    - Xét n = 4 bit, A = 0110
    - Số bù một của A = (2^4-1) - 0110 = 1001
    - Số bù hai của A = 2^4 - 0110 = 1010

Có thể lấy bù một của một số bằng cách đảo bit



Số nguyên có dấu (tt)

- Số bù một và Số bù hai:
  - Dùng n bit để biểu diễn số nguyên có dấu –A = Biểu diễn số bù 2 của A (sử dụng n bit)
  - Ví dụ: Biểu diễn số nguyên có dấu sau đây bằng 8 bit: A = -70
  - Biểu diễn 70 = 0 1 0 0 0 1 1 0

    Bù 1: 1 0 1 1 1 0 0 1

    +1

    Bù 2:1 0 1 1 1 0 1 0
  - Vậy: A= 1011 1010

Số nguyên có dấu (tt)

- Ví dụ: biểu diễn số nguyên có dấu sau đây bằng 8 bit. Sử dụng số bù 1 và số bù 2
  - A = -120
- Ví dụ: biểu diễn số nguyên có dấu sau đây bằng 16 bit. Sử dụng số bù 1 và số bù 2
  - A = -1170

Số nguyên (tt)

- Cộng trừ số nguyên không dấu:
  - Tiến hành cộng lần lượt từng bit từ phải qua trái.
  - Khi cộng hai số nguyên không dấu n bits ta thu được một số nguyên không dấu cũng n bits.
  - Nếu tổng của hai số đó lớn hơn 2^{n-1} thì khi đó sẽ tràn số và kết quả sẽ là sai.

Số nguyên (tt)

- Dùng 8 bit để biểu diễn số nguyên không dấu.
- Trường hợp không xảy ra tràn số (carry-out):
  - X = 1001 0110 = 150
  - Y = 0001 0011 = 19
  - S = 1010 1001 = 169
  - Cout = 0
- Trường hợp có xảy ra tràn số (carry-out):
  - X = 1100 0101 = 197
  - Y = 0100 0110 = 70
  - S = 0000 1011 ≠ 267
  - Cout = 1

Số nguyên (tt)

- Cộng hai số khác dấu: kết quả luôn đúng
- Cộng hai số cùng dấu:
  - Nếu tổng nhận được cùng dấu với 2 số hạng thì kết quả là đúng.
  - Nếu tổng nhận được khác dấu với 2 số hạng thì đã xảy ra hiện tượng *tràn số học* và kết quả nhận được là sai
- Tràn số học xảy ra khi tổng thực sự của hai số nằm ngoài dải biểu diễn của số nguyên có dấu n bit: -2^{n-1} -:- $2^{n-1}-1$

Số nguyên (tt)

Ví dụ về không tràn số

X = 01001000 = 72

Y = 00110101 = 53

S = 01111101 = 125

X = 01011101 = 93

Y = 11001000 = -56

S = 00100101 = 37

Cout = 1 bỏ qua

X = 01001101 = 77

Y = 10011101 = -99

S = 11101010 = -22

X = 10110011 = -77

Y = 11011001 = -39

S = 10001100 = -116

Cout = 1 bỏ qua

Số nguyên (tt)

Ví dụ tràn số

X = 01001101 = 77

Y = 00111011 = 59

S = 10001000 ≠ 136

X = 10001100 = -116

Y = 11001000 = -56

S = 01010100 ≠ -172

Cout = 1

Số nguyên (tt)

- Nhân số nguyên không dấu
- Chia số nguyên không dấu

Số nguyên (tt)

- Các phép toán Logic với số nhị phân

A	B	AND	OR	XOR	NOT
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

- Ví dụ:

X = 01001101
Y = 00111011
X and Y = 00001001

X = 01001101
Y = 00111011
X or Y = 01111111

X = 01001101
Y = 00111011
X xor Y = 01110110

Nội dung

- Các hệ thống số
- Biểu diễn dữ liệu trong máy tính
- Biểu diễn số nguyên
- Thực hiện các phép toán số học với số nguyên
- Biểu diễn số thực
- Thực hiện các phép toán số học với số thực
- Biểu diễn ký tự

Số thực

- Số chấm động (floating point) dùng để tính toán trên số thực.
  - Một số thực X được biểu diễn theo kiểu số dấu chấm động như sau:

    $X = \pm\ m * B^{\pm e}$

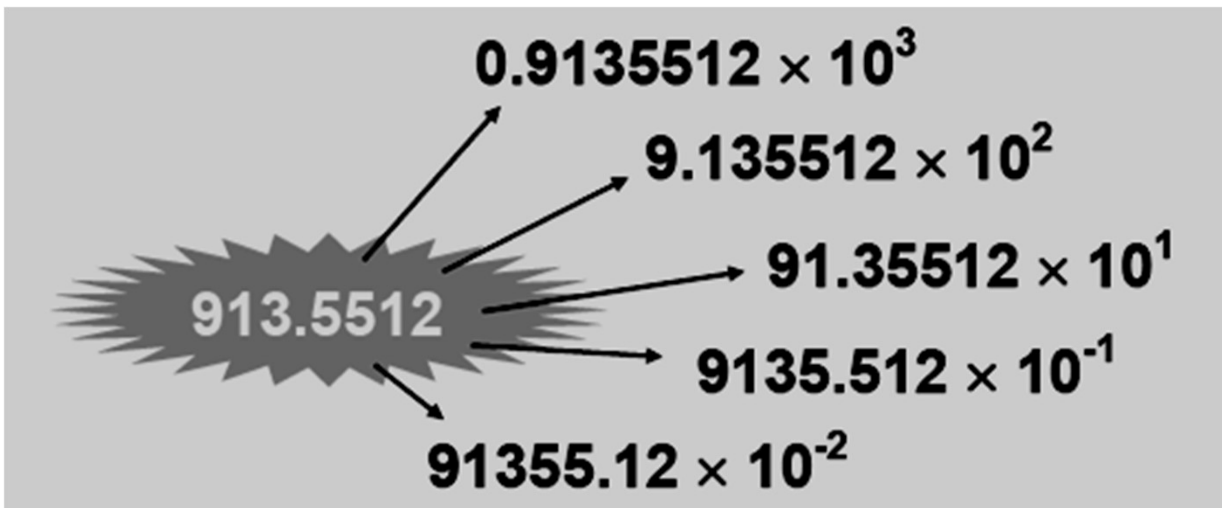
    - m là phần định trị (Mantissa),
    - B là cơ số (base),
    - e là phần mũ (Exponent).
- **m (mantissa) quyết định độ chính xác**
- **B (base)**
- **e (exponent) quyết định độ lớn/nhỏ**

Số thực

- Ví dụ: với cơ số B = 10, giả sử 2 số thực R_1 và R_2 được lưu trữ theo phần định trị và số mũ như sau:
  - $M_1 = -5$ và $E_1 = +9$
  - $M_2 = 3$ và $E_2 = -6$
  - Có nghĩa là $R_1 = M_1 \times 10^{E1} = -5 \times 10^9$
    $= -5,000,000,000$
  - $R_2 = M_2 \times 10^{E2} = 3 \times 10^{-6} = 0.000003$

Số thực

- Một giá trị có thể biểu diễn dưới nhiều dạng
  - Khó xử lý
  - Cần chuẩn hóa

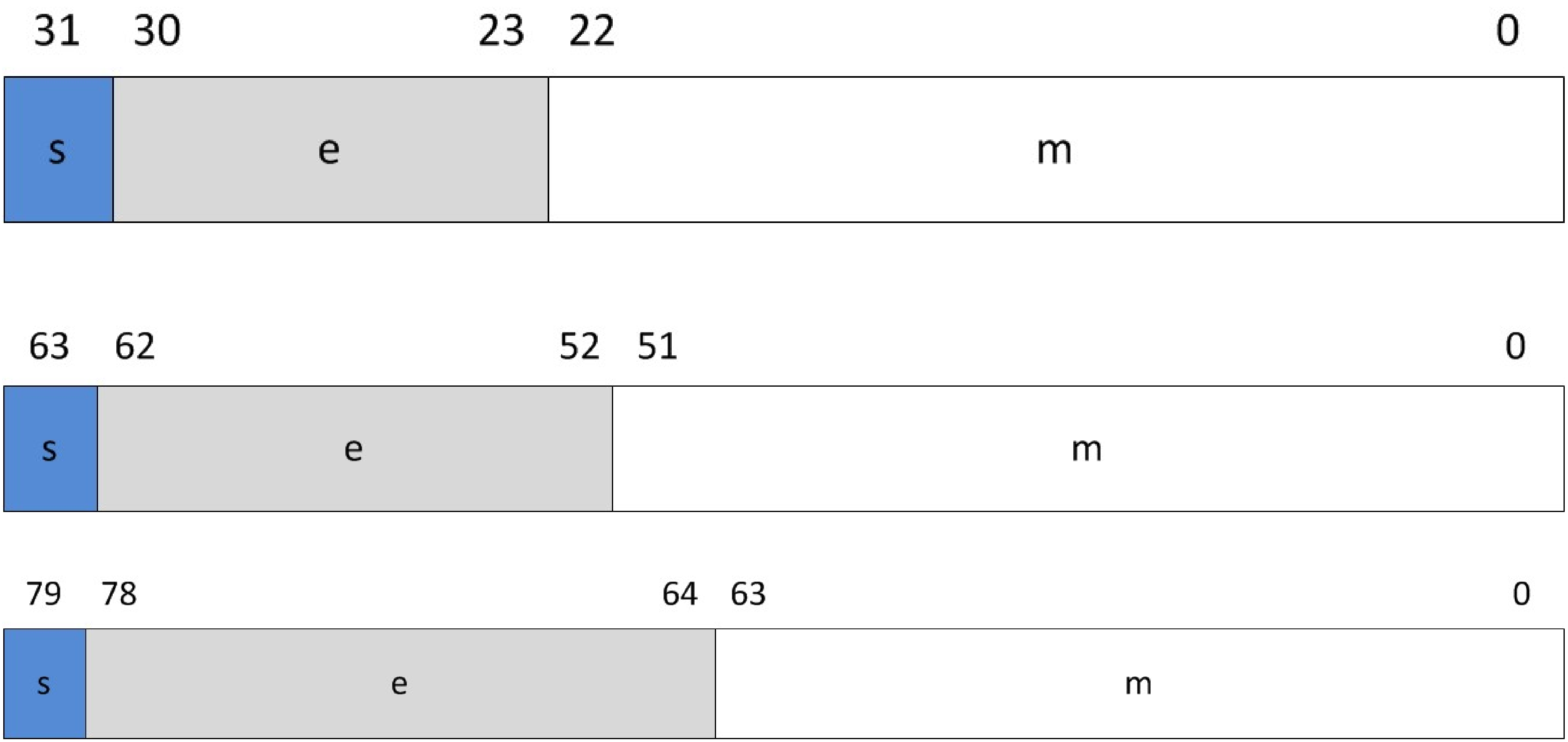


Số thực

- Chuẩn IEEE 754/85
  - Là chuẩn mã hóa số dấu chấm động
  - Cơ số R = 2
  - Có các dạng cơ bản:
    - Dạng có độ chính xác đơn, 32-bit
    - Dạng có độ chính xác kép, 64-bit
    - Dạng có độ chính xác kép mở rộng, 80-bit

Số thực

- Khuôn dạng mã hóa



Số thực

- S là bit dấu, S=0 đó là số dương, S=1 đó là số âm.
- e là mã lệch (excess) của phần mũ E, tức là: E = e –b
- Trong đó b là độ lệch (bias):
  - Dạng 32-bit : b = 127, hay E = e - 127
  - Dạng 64-bit : b = 1023, hay E = e - 1023
  - Dạng 80-bit : b = 16383, hay E = e -16383

Số thực

- m là các bit phần lẻ của phần định trị M, phần định trị được ngầm định như sau: M = 1.m
- Công thức xác định giá trị của số thực tương ứng là: $X = (-1)^S x\ 1.m \times 2^{e-b}$

Số thực

- Ví dụ 1:
  - Có một số thực X có dạng biểu diễn nhị phân theo chuẩn IEEE 754 dạng 32 bit như sau:
  - 1100 0001 0101 0110 0000 0000 0000 0000
  - Xác định giá trị thập phân của số thực đó.
- Ta có:
  - S = 1 → X là số âm
  - e = 1000 0010 = 130
  - m = 10101100...00
  - Vậy $X = (-1)^1 x\ 1.10101100...00 \times 2^{130-127}$

    $= -1.101011 \times 2^3 = -1101.011 = -13.375$

Số thực

- Ví dụ 2:
  - Xác định giá trị thập phân của số thực X có dạng biểu diễn theo chuẩn IEEE 754 dạng 32 bit như sau:
  - 0011 1111 1000 0000 0000 0000 0000 0000
- Ta có:
  - S = 0 → X là số dương
  - e = 0111 1111= 127
  - m = 000000...00
  - Vậy X = $(-1)^0$x 1.0000...00 x $2^{127-127}$

    = 1.0 x 2^0= 1

Số thực

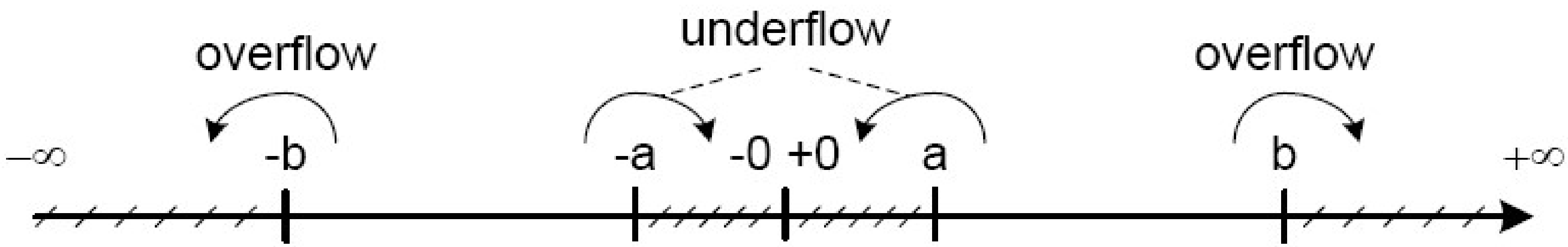
- Ví dụ 3:
  - Biểu diễn số thực X = 9.6875 về dạng số dấu chấm động theo chuẩn IEEE 754 dạng 32 bit
- Ta có:
  - $X = 9.6875_{(10)} = 1001.1011_{(2)} = 1.0011011 \times 2^3$
  - S = 0 vì đây là số dương
  - $E = e - 127$ nên $e = 127 + 3 = 130_{(10)} = 1000\ 0010_{(2)}$
  - m = 001101100...00 (23 bit)
  - Vậy: X = 0100 0001 0001 1011 0000 0000 0000 0000

Số thực

- Ví dụ 4:
  - Biểu diễn số thực X = 21. 875 về dạng số dấu chấm động theo chuẩn IEEE 754 dạng 32 bit
- Ta có:
  - $X = 21.875_{(10)} = 10101.111_{(2)} = 1.0101111 \times 2^4$
  - S = 0 vì đây là số dương
  - $E = e - 127$ nên $e = 127 + 4 = 131_{(10)} = 1000\ 0011_{(2)}$
  - m = 010111100...00 (23 bit)
  - Vậy: X = 0100 0001 1010 1111 0000 0000 0000 0000

Số thực

- Trục số biểu diễn



- Dạng 32 bit: $a = 2^{-127} \approx 10^{-38}$ $b = 2^{+127} \approx 10^{+38}$
- Dạng 64 bit: $a = 2^{-1023} \approx 10^{-308}$ $b = 2^{+1023} \approx 10^{+308}$
- Dạng 80 bit: $a = 2^{-16383} \approx 10^{-4932}$ $b = 2^{+16383} \approx 10^{+4932}$

Số thực

- Lưu ý:
  - Nếu tất cả các bit của e đều bằng 0, các bit của m đều bằng 0, thì X = 0
  - Nếu tất cả các bit của e đều bằng 1, các bit của m đều bằng 0, thì X = $\pm\infty$
  - Nếu tất cả các bit của e đều bằng 1, m có ít nhất một bit bằng 1, thì X không phải là số (not a number -NaN)

Nội dung

- Các hệ thống số
- Biểu diễn dữ liệu trong máy tính
- Biểu diễn số nguyên
- Thực hiện các phép toán số học với số nguyên
- Biểu diễn số thực
- Thực hiện các phép toán số học với số thực
- Biểu diễn ký tự

Biểu diễn ký tự

- **Nguyên tắc chung:**
  - Các ký tự cũng cần được chuyển đổi thành chuỗi bit nhị phân gọi là **mã ký tự**.
  - Số bit dùng cho mỗi ký tự theo các mã khác nhau là khác nhau.
- Ví dụ :
  - Bộ mã ASCII dùng 8 bit cho 1 ký tự.
  - Bộ mã Unicode dùng 16 bit.

Bộ mã ASCII

- Do ANSI (American National Standard Institute) thiết kế
- ASCII là bộ mã được dùng để *trao đổi thông tin chuẩn của Mỹ*. Lúc đầu chỉ dùng 7 bit (128 ký tự) sau đó mở rộng cho 8 bit và có thể biểu diễn 256 ký tự khác nhau trong máy tính
- Bộ mã 8 bit mã hóa được cho $2^8 = 256$ kí tự, có mã từ 00_{16} -:- FF_{16}, bao gồm:
  - 128 kí tự chuẩn có mã từ 00_{16} -:- $7F_{16}$
  - 128 kí tự mở rộng có mã từ 80_{16} -:- FF_{16}

Bộ mã ASCII

- 95 kí tự hiển thị được:có mã từ 20_{16} -:- $7E_{16}$
  - 26 chữ cái hoa Latin 'A' -:- 'Z' có mã từ 41_{16} -:- $5A_{16}$
  - 26 chữ cái thqờng Latin 'a' -:- 'z' có mã từ 61_{16} -:- $7A_{16}$
  - 10 chữ số thập phân '0' -:- '9' có mã từ 30_{16} -:- 39_{16}
  - Các dấu câu: . , ? ! : ; …
  - Các dấu phép toán: + -* / …
  - Một số kí tự thông dụng: #, $, &, @, ...
  - Dấu cách (mã là 20_{16})
- 33 mã điều khiển: mã từ 00_{16} -:- $1F_{16}$ và $7F_{16}$ dùng để mã hóa cho các chức năng điều khiển

Bộ mã Unicode

- Do các hãng máy tính hàng đầu thiết kế
- Bộ mã 16-bit
- Bộ mã đa ngôn ngữ
- Có hỗ trợ các ký tự tiếng Việt