Introduction to Informatics

Piroska Biró

Labor 04

Revision

- Convert the following numbers to decimal system. Solve each of them with both learnt methods.

- $3254.26'405'405\ldots_{(7}$

Revision

- Fixed-point
  - sign-and-magnitude method (absolute value)
  - one’s complement
  - two’s complement
  - excess K
- Floating point
  - IEEE 754

Sign-and-magnitude method

- sign bit
  - the highest place value (the first bit on the left)
    - 0: +
    - 1: –
- remaining bits
  - binary
  - magnitude or absolute value of the number
- properties
  - two ways to represent 0:
    00000000 (+0) and 10000000 (–0)
  - the smallest number: –127: 11111111
  - the biggest number: +127: 01111111

1's complement

- sign bit
  - the highest place value (the first bit on the left)
    - 0: +
    - 1: –
- remaining bits
  - binary
  - positive number: number
  - negative number: number*(–1) (negative binary number – bitwise NOT)
- properties
  - two ways to represent 0: 00000000 (+0) and 11111111 (–0)
  - the smallest number: –127: 11111111
  - the biggest number: +127: 01111111

2's complement

- sign bit
  - the highest place value (the first bit on the left)
    - 0: +
    - 1: −
- remaining bits
  - binary
  - positive number: number
  - negative number: 1's complement+1
- properties
  - one ways to represent 0 (00000000)
  - the smallest number: −128
  - the biggest number: +127

Excess-K number representation Offset binary/biased representation

- represent the sum of the number and the excess in a binary form
  - positive number
  - the excess in the case of n bit number: $2^{n-1}-1$, 2^{n-1}
    - the higest place value is 1, the remaining 0
    - the higest place value is 0, the remaining 1
- properties (excess-128)
  - 0 can be represented definitely
  - the biggest number: +127 ($2^{8-1}-1$)
  - the smallest number: −128 (2^{8-1})
- observations
  - the system is the same as the two's complement, with changed sign
  - using: floating point numbers in exponents part

Exercise

- Represent the given decimal numbers in **8 bits** with the following fixed-pointed methods.
  - sign-and-magnitude
  - 1's complement
  - 2's complement
  - excess-127
  - excess-128

1. 78_{10}
2. -117_{10}

- Convert the results to hexadecimal form.

Exercise

- Represent the given decimal numbers in **16 bits** with the following fixed-pointed methods.
  - sign-and-magnitude
  - 1's complement
  - 2's complement
  - excess-2^{15} -1
  - excess-2^{15}

  1. 8789_{10}
  2. -27269_{10}

- Convert the results to hexadecimal form.

BCD code–Binary Coded Decimal

- We represent just the digits with the following two methods.
- **Uncompressed**: each numeral is encoded into one byte .
- **Packed**: every decimal digit is represented in four bits (1 nibble).

- Encoding the decimal number **91** using
  - uncompressed BCD results in the following binary pattern of two bytes:

    0000 1001 0000 0001
  - in packed BCD, the same number would fit into a single byte:

    1001 0001

BCD code

- representation of negative numbers: nine's and ten's complement

$$6892_{(10} = 0110|1000|1001|0010$$ BCD

$$+301_{(10} = 0000|0011|0000|0001$$ BCD

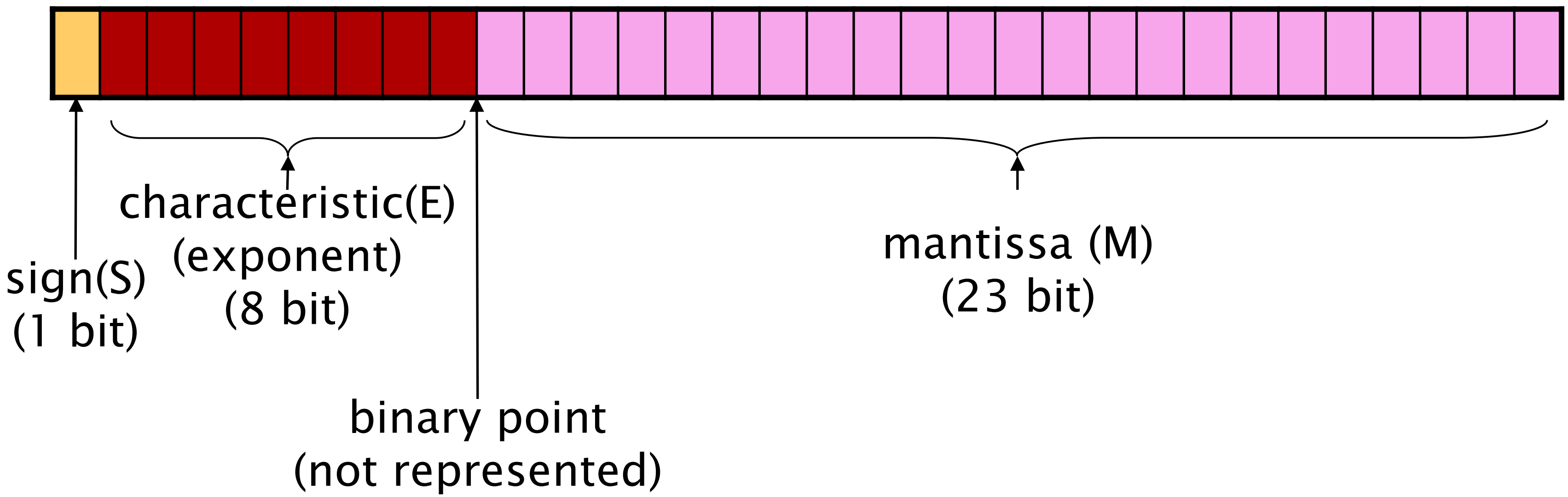
$$-301_{(10} = 1001|0110|1001|1000$$ 9's complement

$$-301_{(10} = 1001|0110|1001|1001$$ 10's complement

Exercise

- Define the packed BCD code of the following numbers (with negative numbers use the nine's and ten's complement).
  - $378_{10}=$
  - $-864_{10}=$
  - $2546_{10}=$
  - $-4124_{10}=$

Floating point representation IEEE 754



- normalized in binary number system
- normalized to ineger
- characteristic: excess-127
- sign
  - positive number: 0
  - negative number: 1

$$N = (-1)^S \cdot \left(2^{E-127}\right) \cdot (1.M)$$

Exercise

$$\underset{S}{0}\Big|\underbrace{10001000}_{E}\Big|\underbrace{00010101001000000000000}_{M}$$

- $S = 0$
- $E = 1000\ 1000_{(2} = 136_{(10}$
- $M = .0001010101001_{(2} = .082519531_{(10}$
- $\text{Number} = 1.082519531 \cdot 2^9 = 554.25$

Exercise

- Which numbers were represented with the IEEE 754 floating point standard?

  - 11000100100000110100000000000000
  - 01000010111011000000000000000000

Exercise

$$554.25_{(10} = 1000101010.01_{(2} = 1.00010101001 \cdot 2^9$$

- S = 0
- E = 127+9= $136_{(10} = 1000\ 1000_{(2}$
- M = $.00010101001_{(2}$

0100	0100	0000	1010	1001	0000	0000	0000
4	4	0	A	9	0	0	0

Exercise

- Represent the following decimal numbers in 32 bits using the IEEE 754 floating point standard.

  - $164_{(10}$
  - $-343.62_{(10}$

Revision – Exercise

- Which numbers were represented with the IEEE 754 floating point standard?

01000100111110111000000000000000

- Represent the following decimal numbers in **32 bits** using the IEEE 754 floating point standard.

–1011,125

Floating point number representation with excess characteristic

- Represent $148_{(10}$ number in **octal** system.
  - starting with sign bit
  - the exponent will be 1 digit (3 bits), excess-4
  - the fraction part 3 digits

$$148_{(10} = 224_{(8} = 0.224 \cdot 8^{3}$$

$$0111010010100$$

0 7 2 2 4

Floating point number representation with excess characteristic

- Represent $1048_{(10}$ number in **hexadecimal** system.
  - starting with sign bit
  - the exponent will be 1 nibble (4 bits), excess-8
  - the fraction part 4 digits

$$1048_{(10} = 418_{(16} = 0.4180 \cdot 16^3$$

$$\underset{0}{0}\,\underset{B}{1101}\,\underset{4}{0100}\,\underset{1}{0001}\,\underset{8}{1000}\,\underset{0}{0000}$$

Exercise

- Represent the following numbers in **octal** system.
  - starting with sign bit
  - the exponent will be 1 digit (3 bits), excess-4
  - the fraction part 4 digit

  a. $-215_{(10}$
  b. $289_{(10}$

- Represent the following numbers in **hexadecimal** system.
  - starting with sign bit
  - the exponent will be 1 nibble (4 bits), excess-8
  - the fraction part 4 digit

  a. $1641.5_{(10}$
  b. $-12621_{(10}$