Introduction to informatics

Course 03

Revision

- Who can tell me the definition of the computer?
- Which are the input peripherals?
- Which are the output peripherals?
- What is the unit of information?
- What do we know about the relationship of bit and byte?

Bit, Byte

- 1 byte = 8 bit
- 1 Kbyte = 2^{10} byte = 1024 byte = 8192 bit
- 1 Mbyte = 2^{10} Kbyte = 2^{20} byte = 1048576 byte
- 1 Gbyte = 2^{10} Mbyte = 2^{20} Kbyte = 2^{30} Kbyte
- 1 Tbyte = 2^{10} Gbyte

Data representation

- Bitseries are computerized form of data, the basic units of storage consists of 8 bits which equals one byte.

- Two methods of data storage
  - computerized number representation (calculations)
  - decoded representation

Representation of Numbers

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

- Floating point

Signed fixed-point numbers

- 256 different numbers stored in eight bits
  - $2^8 = 256$

- Which are the negative numbers?

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

Sign-and-magnitude method

$+25_{10} = 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1$

$\phantom{+25_{10} = }\ +$

$-25_{10} = 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1$

$\phantom{-25_{10} = }\ -$

$+91_{10} = 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1$

$\phantom{+91_{10} = }\ +$

$-91_{10} = 1\ 1\ 0\ 1\ 1\ 0\ 1\ 1$

$\phantom{-91_{10} = }\ -$

$+25_{10} =$

0 0 0 1	1 0 0 1
1	9

$-25_{10} =$

1 0 0 1	1 0 0 1
9	9

$+91_{10} =$

0 1 0 1	1 0 1 1
5	B

$-91_{10} =$

1 1 0 1	1 0 1 1
D	B

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)
- properties
  - two ways to represent 0:
    00000000 (+0) and 11111111 (−0)
  - the smallest number: −127: 11111111
  - the biggest number: +127: 01111111

1's complement

$$+25_{10} = 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1$$

$$-25_{10} = 1\ 1\ 1\ 0\ 0\ 1\ 1\ 0$$

$$+25_{10} = \underbrace{0\ 0\ 0\ 1}_{1}\ \Big|\ \underbrace{1\ 0\ 0\ 1}_{9}$$

$$-25_{10} = \underbrace{1\ 1\ 1\ 0}_{E}\ \Big|\ \underbrace{0\ 1\ 1\ 0}_{6}$$

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’ complement+1
- properties
  - one ways to represent 0 (00000000)
  - the smallest number: −128
  - the biggest number: +127

2's complement

$$+25_{10} = 00011001$$

$$+25_{10} = \underset{1}{0001}|\underset{9}{1001}$$

1's complement
2's complement

$$-25_{10} = 11100110$$

$$-25_{10} = \underset{E}{1110}|\underset{6}{0110}$$

1's complement

$$-25_{10} = 11100111$$

$$-25_{10} = \underset{E}{1110}|\underset{7}{0111}$$

2's complement

Binary addition

125		0	1	1	1	1	1	0	1	
-105		1	1	1	0	1	0	0	1	sign-and-magnitude

125		0	1	1	1	1	1	0	1	
-105	$+$	1	0	0	1	0	1	1	1	2's complement
$+20$		0	0	0	1	0	1	0	0	

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
Positive number with excess representation

+	2	5		0	0	0	1	1	0	0	1
1	2	8	+	1	0	0	0	0	0	0	0
				1	0	0	1	1	0	0	1

addition in binary system

$$
\begin{array}{rrrr}
 & 1 & 2 & 8 \\
+ & & 2 & 5 \\
\hline
 & 1 & 5 & 3
\end{array}
$$

$$153_{10} = 10011001_2$$

addition in decimal system, conversion

Exercise
Negative numbers with excess representation

+	2	5		0	0	0	1	1	0	0	1
−	2	5		1	1	1	0	0	1	1	1
1	2	8	+	1	0	0	0	0	0	0	0
				0	1	1	0	0	1	1	1

addition in binary system

1	2	8		1	0	0	0	0	0	0	0
+	2	5	−	0	0	0	1	1	0	0	1
				0	1	1	0	0	1	1	1

subtraction is in binary system

$$
\begin{array}{r}
128 \\
-\quad 25 \\
\hline
103
\end{array}
$$

$$103_{10} = 01100111_2$$

subtraction in decimal system, conversion

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. 67_{10}
2. -99_{10}
3. 108_{10}
4. -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. -356_{10}
2. 987_{10}
3. 8789_{10}
4. -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

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

BCD code

- representation of negative numbers
  - nine’s and ten’s complement
- Example:
- −432
- the nine's complement is 9999−432=9567
- the ten's complement is the nine's complement plus one: 9568
- −432 in signed BCD is 1001 0101 0110 1000.

BCD code

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

$$+301_{10} = 0000|0011|0000|0001$$ 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).

  - $378_{10}=$
  - $-864_{10}=$
  - $5643_{10}=$
  - $-8327_{10}=$

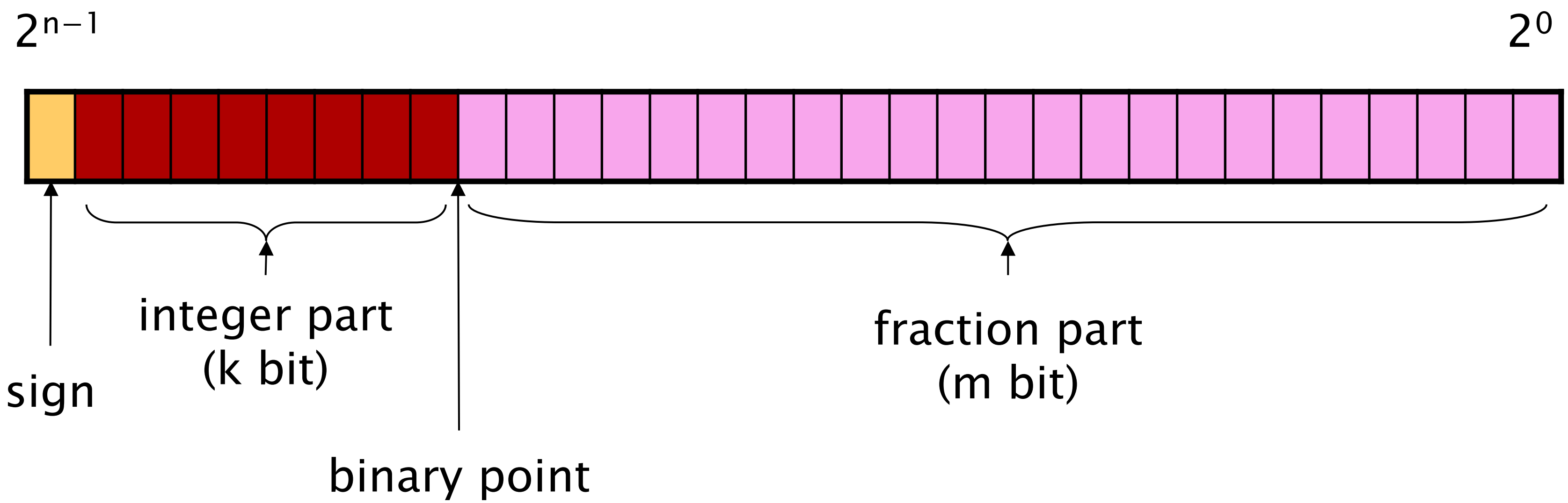
Representing real numbers

- fixed point
- floating point
  - numbers in normalized form

$$N = \pm M \cdot p^{\pm E}$$

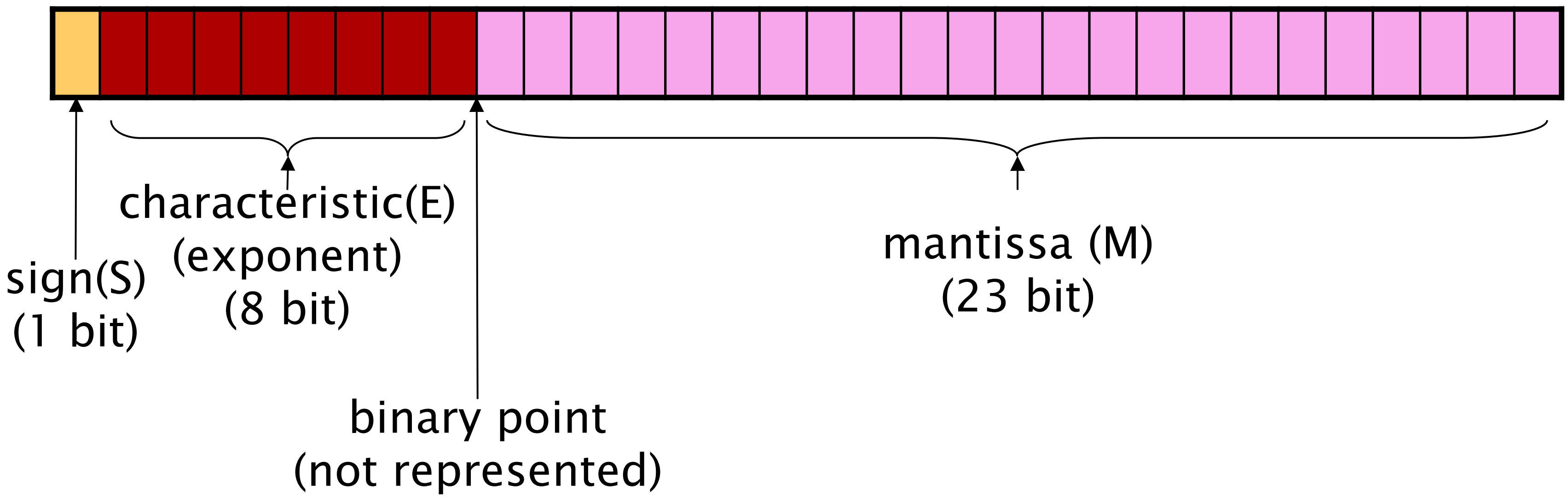
$$254.25_{10} = 2.5425 \cdot 10^{2} = 0.25425 \cdot 10^{3}$$

Fixed point



- storage capacity, tha place of binary point
  - size of numbers to be representatied
  - punctuality of representation
  - special cases
    - if the binary point is on the right edge of storage, then fixed-pointed integer
    - if the binary point is on the left edge of storage, then fixed-pointed fraction

Floating point representation IEEE 754



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

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

IEEE 754 standard

Type	Number of bits	Sign bit	Characteristic	Mantissa
single	32	1	8 bit Excess-127	23 bit
double	64	1	11 bit Excess -1023	52 bit

Exercise

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

- $S = 0$
- $E = 1000\ 1000_{(2} = 136_{(10}$
- $M = .0001010 1001_{(2} = .082519531_{(10}$
- $\text{Number} = 1.\,082519531 \cdot 2^9 = 554.25$

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