Introduction to informatics

Piroska Biró

Course 03

Revision

- History of computing, Computer generations
- Who used punched cards for operations at first?
  - Joseph Jacquard
- Who started to develop the Analytical Engine?
  - Charles Babbage
- Who designed the Z1, Z2, Z3 computers?
  - Konrad Zuse
- In which computer generation did the transistors appear?
  - The Second Generation

Revision

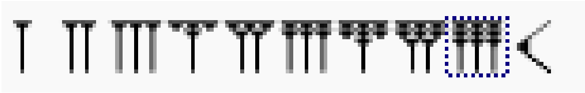
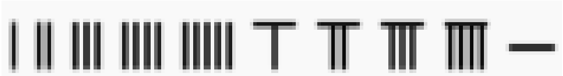
- Who can tell me the definition of the computer?
- Which are the input peripherals?
- Which are the output peripherals?
- What is the unit of information?
- What do we know about the relationship of bit and byte?

Bit, Byte

- 1 byte = 8 bit
- 1 Kbyte = 2^{10} byte = 1024 byte = 8194 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

Number/Numeral system concepts

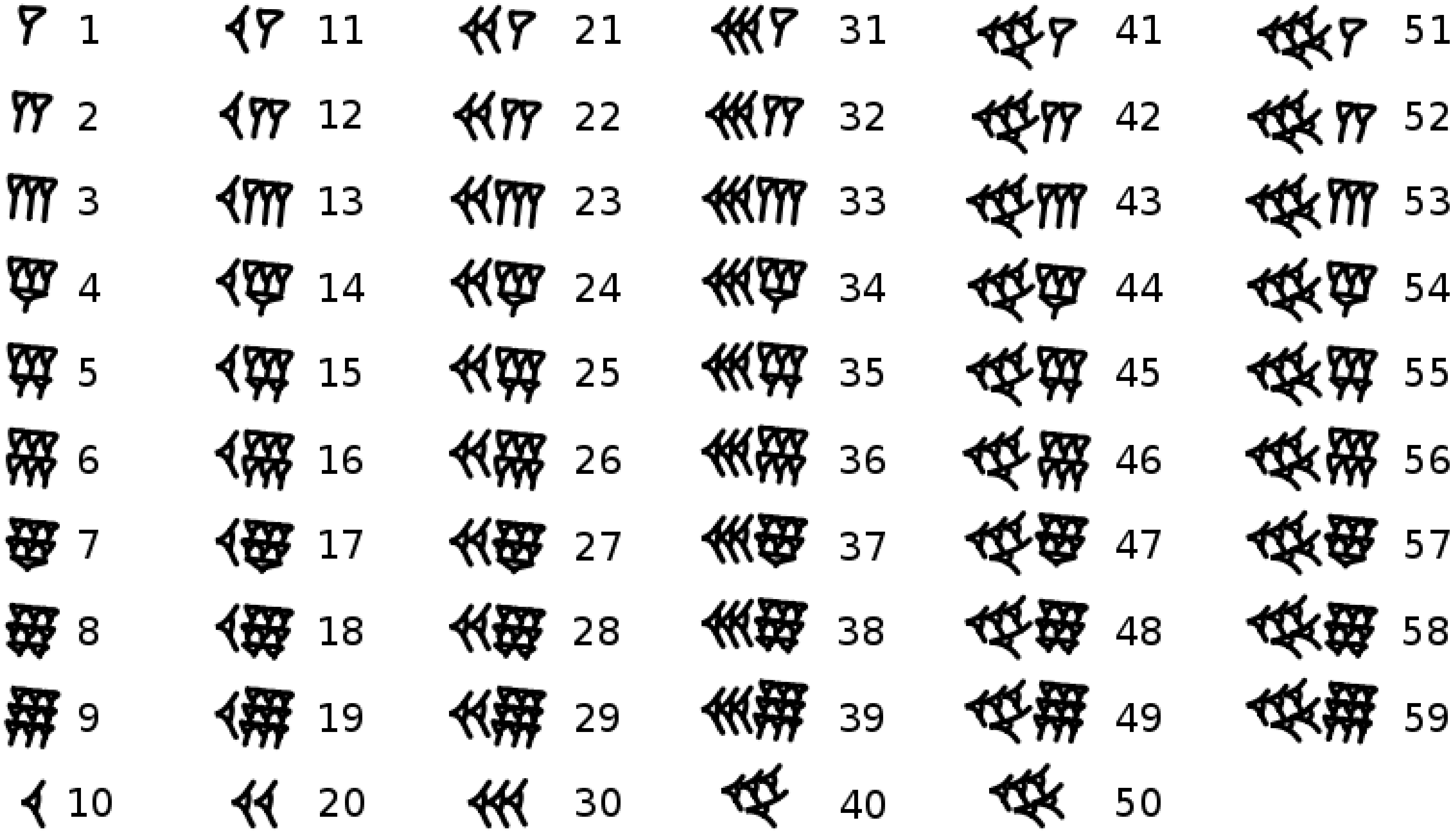
- The numeral or number systems are all the processes of the denominations and the description of numbers.
  - **non-positional** (e.q. Egyptian, Mayan, Roman; difficult calculation in them)
  - **positional notation** or **place-value notation**
    - Babylonian(B.C.1750): sexagesimal (time and angle measuring)
    - Indian (A.D. 600): decimal system (digits: 1, 2, . . . , 9)
    - Arabic (A.D. 750): appereance of the 0
    - Europe between 1200-1600 spread generally

Name	Base	Sample	Approx. first appearance
Babylonian numerals	60		3100 B.C.
Greek numerals	10	α β γ δ ϵ ϝ ζ η θ ι	
Roman numerals	10	I II III IV V VI VII VIII IX X	1000 B.C.
Chinese rod numerals	10		1st century
Arabic numerals	10	0 1 2 3 4 5 6 7 8 9 10	9th century
Nepers's Location arithmetic	2	a b c d e f g h i j	1617 in Rabdology, a non-positional binary system

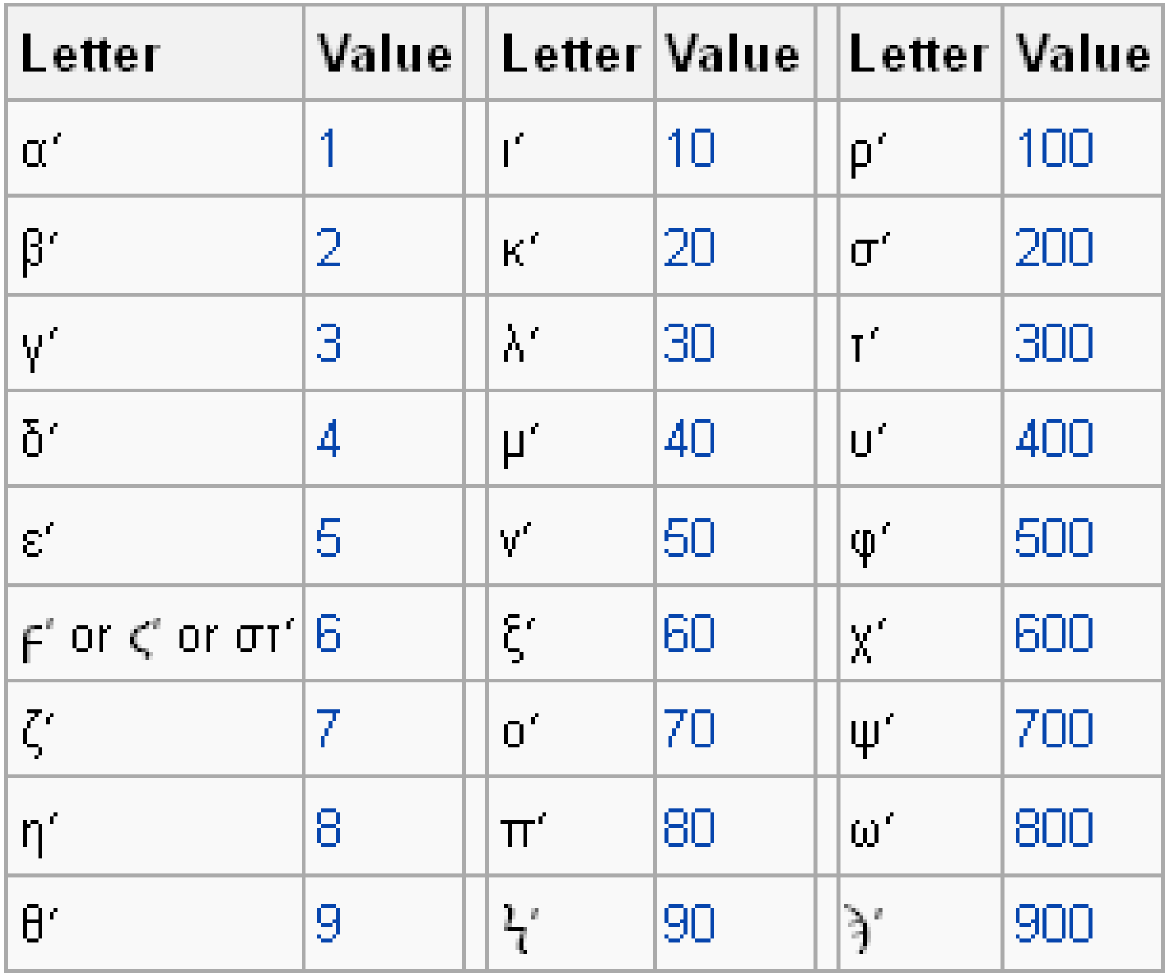
Babylonians

- sexagesimal (base–60)
- first appeared around 3100 B.C
- to count units 𒁹
- to count tens 𒌋
- 23 𒌋𒌋 𒐈
- 23 or 23×60 or 23×60×60 or 23/60

Babylonian symbols



Greek numerals



Letter	Value	Letter	Value	Letter	Value
αʹ	1	ιʹ	10	ρʹ	100
βʹ	2	κʹ	20	σʹ	200
γʹ	3	λʹ	30	τʹ	300
δʹ	4	μʹ	40	υʹ	400
εʹ	5	νʹ	50	φʹ	500
ϝʹ or ϛʹ or στʹ	6	ξʹ	60	χʹ	600
ζʹ	7	οʹ	70	ψʹ	700
ηʹ	8	πʹ	80	ωʹ	800
θʹ	9	ϟʹ	90	ϡʹ	900

Roman numerals

1 = I
2 = II
3 = III
4 = IV
5 = V
6 = VI
7 = VII
8 = VIII
9 = IX
10 = X

11 = XI
12 = XII
13 = XIII
14 = XIV
15 = XV
16 = XVI
17 = XVII
18 = XVIII
19 = XIX
20 = XX
21 = XXI

25 = XXV
30 = XXX
40 = XL
49 = XLIX
50 = L
51 = LI
60 = LX
70 = LXX
80 = LXXX
90 = XC
99 = XC

Roman numerals

I=1

V=5

X=10

L=50

C=100

D=500

M=1000

Exercise:

49 = XLIX
68 = LXVIII
156 = CLVI
496 = CDXCVI
1347 = MCCCXLVII
2014 = MMXIV

Arabic numerals

0123456789

General Numeral systems

Definition: The p-based positional notation or place-value notation number systems rule is:

$$(\ldots \bar{a}_2 \bar{a}_1 \bar{a}_0 . \bar{a}_{-1} \bar{a}_{-2} \ldots)_r =$$

$$= \sum_{i=-\infty}^{\infty} a_i p^i =$$

$$= \cdots + a_2 p^2 + a_1 p + a_0 + a_{-1} p^{-1} + a_{-2} p^{-2} + \cdots .$$

Decimal number system

3457,28 3457.28

3thousand + 4hundred + 5ten + 7one + 2tenth + 8hundreth

$$3 \cdot 10^3 + 4 \cdot 10^2 + 5 \cdot 10 + 7 + 2 \cdot 10^{-1} + 8 \cdot 10^{-2}$$

$$\bar{a}_n \bar{a}_{n-1} \bar{a}_{n-2} \dots \bar{a}_2 \bar{a}_1 \bar{a}_0 . \bar{a}_{-1} \bar{a}_{-2} \dots \bar{a}_{-m}$$

$$S_{(10} = \sum_{i=-m}^{n} a_i \cdot 10^i$$

Number systems

- p base system (any $p>1$)
  - digits: 0, 1, …, $p-1$

$$\sum_{i=-m}^{n} a_i \cdot p^i$$

- binary system
  - $p = 2$
  - digits: 0, 1

$$\sum_{i=-m}^{n} a_i \cdot 2^i$$

- octal system
  - $p = 8$
  - digits: 0, 1, 2, 3, 4, 5, 6, 7

$$\sum_{i=-m}^{n} a_i \cdot 8^i$$

- hexadecimal system
  - $p = 16$
  - digits: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

$$\sum_{i=-m}^{n} a_i \cdot 16^i$$

Rules

- Convert from p-base to decimal system.

Exercises

- Convert from p-base system to decimal.
  - $101.101_{(2}=$
  - $32121.41_{(4}=$
  - $423.45_{(6}=$
  - $293.745(_{8}=$
  - $154.485_{(9}=$

Rules

- Convert from decimal to p-base system.

Exercises

- Convert from decimal to other p-base system.
  - $45.55_{(10}=?_{(2}$
  - $111.45_{(10}=?_{(4}$
  - $23.45_{(10}=?_{(5}$
  - $23.45_{(10}=?_{(8}$
  - $54.45_{(10}=?_{(16}$

Rules

- Conections between binary, octal and hexadecimal systems.

Exercises

1. $1000\ 1001\ 1111\ 1101_{(2}=$
2. $1010\ 0111\ 1101\ 1110_{(2}=$
3. $1011\ 1100\ 0001\ 0001_{(2}=$
4. $1000\ 0101\ 0110\ 1001_{(2}=$
5. $\mathrm{BCDFA}_{(16}=$
6. $\mathrm{A94F6}_{(16}=$
7. $\mathrm{DF237}_{(16}=$
8. $\mathrm{A12F8}_{(16}=$

Exercises

- $23612.352_{(9} = ?_{(10}$
- $32918.35_{(10} = ?_{(7}$
- $B7E3DAC_{(16} = ?_{(2} = ?_{(8}$
- $10100101101111010110111_{(2} = ?_{(8} = ?_{(16}$

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

$$+25_{(10} = \underset{+}{0}\,0\,0\,1\,1\,0\,0\,1$$

$$-25_{(10} = \underset{-}{1}\,0\,0\,1\,1\,0\,0\,1$$

$$+91_{(10} = \underset{+}{0}\,1\,0\,1\,1\,0\,1\,1$$

$$-91_{(10} = \underset{-}{1}\,1\,0\,1\,1\,0\,1\,1$$

$$+25_{(10} = \begin{array}{c|c} 0\,0\,0\,1 & 1\,0\,0\,1 \\ \hline 1 & 9 \end{array}$$

$$-25_{(10} = \begin{array}{c|c} 1\,0\,0\,1 & 1\,0\,0\,1 \\ \hline 9 & 9 \end{array}$$

$$+91_{(10} = \begin{array}{c|c} 0\,1\,0\,1 & 1\,0\,1\,1 \\ \hline 5 & B \end{array}$$

$$-91_{(10} = \begin{array}{c|c} 1\,1\,0\,1 & 1\,0\,1\,1 \\ \hline D & B \end{array}$$

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

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

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

$$+25_{(10} = \begin{array}{c|c} 0\ 0\ 0\ 1 & 1\ 0\ 0\ 1 \\ \hline 1 & 9 \end{array}$$

$$-25_{(10} = \begin{array}{c|c} 1\ 1\ 1\ 0 & 0\ 1\ 1\ 0 \\ \hline \mathrm{E} & 6 \end{array}$$

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

$$\begin{array}{rccccccccccl}
125 & & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & & \\
-105 & & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & & \text{sign-and-magnitude} \\
\hline
\end{array}$$

$$\begin{array}{rccccccccccl}
125 & & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & & \\
-105 & + & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & & \text{2's complement} \\
\hline
+20 & & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & &
\end{array}$$

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

$$\begin{array}{ccc|ccccccccc} + & 2 & 5 & & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 2 & 8 & + & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline & & & & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \end{array}$$

addition in binary system

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

$$153_{(10} = 10011001_{(2}$$

addition in decimal system, conversion

Exercise
Negative numbers with excess representation

$$\begin{array}{ccc|ccccccccc}
+ & 2 & 5 & & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
\hdashline
- & 2 & 5 & & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\
1 & 2 & 8 & + & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\hline
 & & & & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1
\end{array}$$

addition in binary system

$$\begin{array}{ccc|ccccccccc}
1 & 2 & 8 & & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
+ & 2 & 5 & - & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\
\hline
 & & & & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1
\end{array}$$

subtraction is in binary system

$$\begin{array}{cccc}
 & 1 & 2 & 8 \\
- & & 2 & 5 \\
\hline
 & 1 & 0 & 3
\end{array}$$

$$103_{(10} = 01100111_{(2}$$

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

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

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

BCD

$$-301_{(10} = 1001|0110|1001|1000$$

9’s complement

10’s complement

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

Exercise

- Define the packed BCD code of the following numbers (with negative numbers use the nine’s and ten’s complement).

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

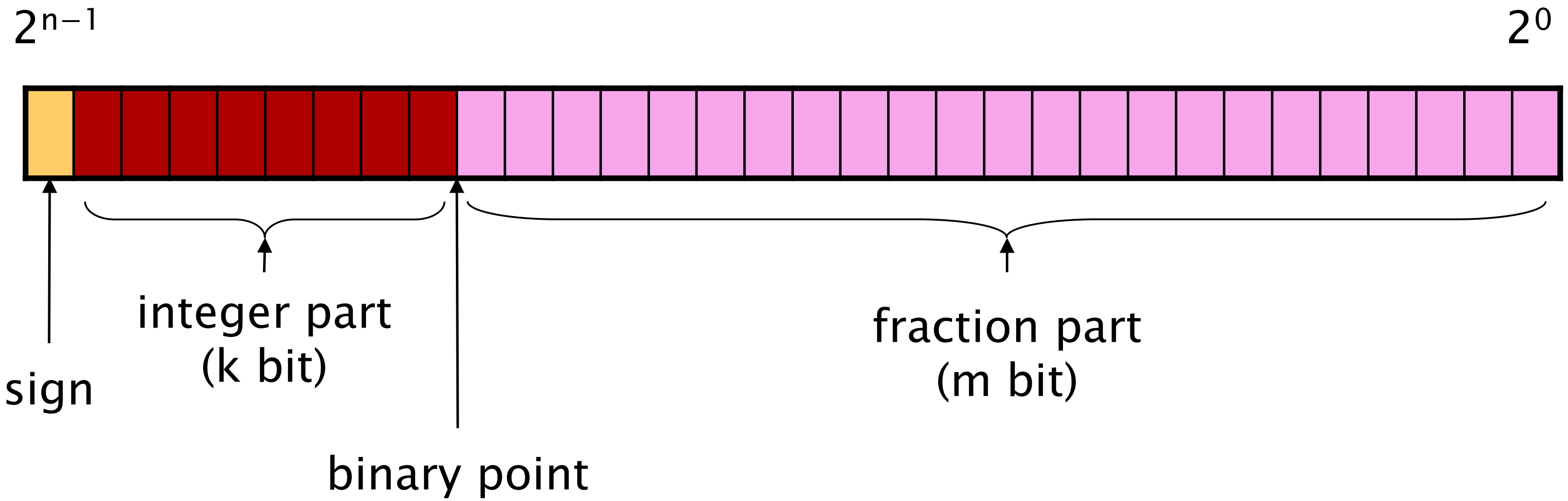
Representing real numbers

- fixed point
- floating point
  - numbers in normalized form

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

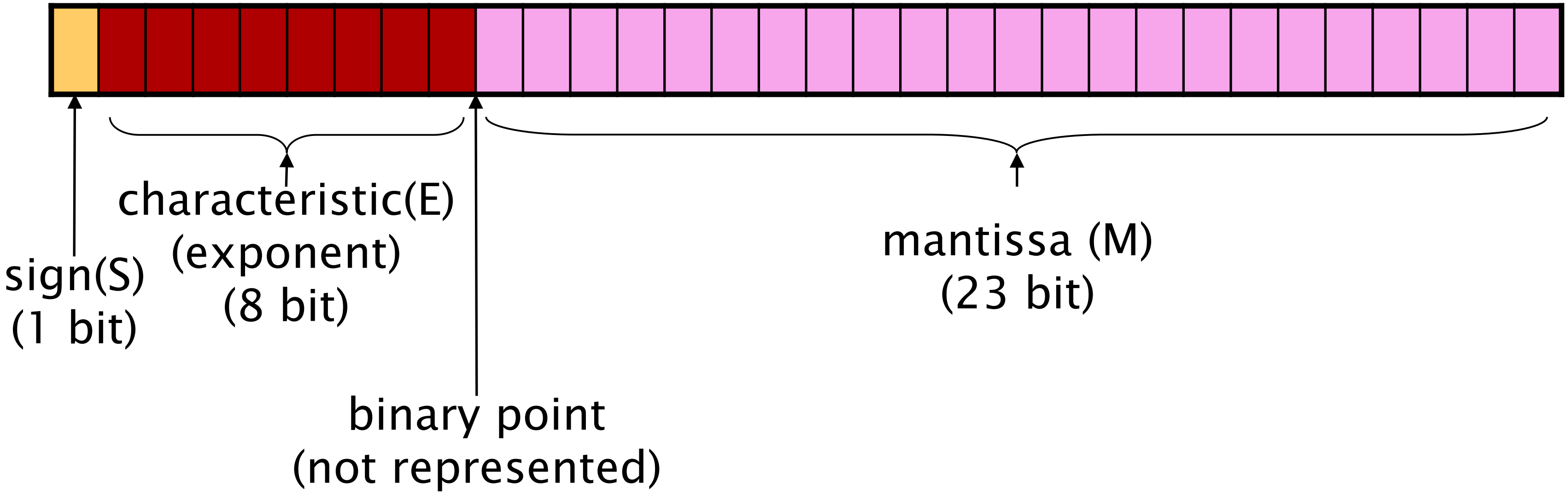
$$254.25_{(10} = 2.5425 \cdot 10^{2} = 0.25425 \cdot 10^{3}$$

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

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

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

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

- $S = 0$

- $E = 1000\ 1000_{(2} = 136_{(10}$

- $M = .00010101001_{(2} = .082519531_{(10}$

- $\text{Number} = 1.\,082519531 \cdot 2^{9} = 554.25$

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