X36SKD

seminar

Number systems, codes, conversions, arithmetic operations

Tomáš Brabec, Miroslav Skrbek, Jiří Buček

Seminar 2

- Radix number systems and conversions
  - Binary system, converting to decimal
  - Hexadecimal system, converting to binary
- Arithmetic (1)
  - Addition
  - Multiplication
- Radix grid
- Representing negative numbers
- Arithmetic (2)
  - Subtraction

Literature

[1] Knuth, D.E., „*The Art of Computer Programming*, Vol.2“, (Third Edition), Chapter 4 – Arithmetic, Addison-Wesley, 1998, ISBN 0-201-89684-2

Radix number systems I

- Base (radix) z, $z \in N$ a $z \geq 2$
- Base-z system
- Most often used systems:

$z = 2$	binary
$z = 10$	decimal
$z = 16$	hexadecimal

Radix number systems II

- Number representation in base-z system:

$$A_z = (\underbrace{a_n\ a_{n-1} \ldots a_1\ a_0}_{\text{integer part}} \overset{\text{radix point}}{,} \underbrace{a_{-1}\ a_{-2} \ldots a_{-m}}_{\text{fractional part}})_{\underset{\text{radix of the sytem}}{z}},\quad n,m \in N$$

a_i	…	figure (digit) at the i-th position (order)
a_i	…	value of digit a_i, $0 \le \mathrm{a}_i < z$
i	…	order (position) of the digit, => weight $v_i = z^i$
n	…	highest order with nonzero digit
$-m$	…	lowest order with nonzero digit

- Value of A_z:

$$\mathrm{A} = v(A_z) = \sum_{-m}^{n} \mathrm{a}_i \cdot v_i = \sum_{-m}^{n} \mathrm{a}_i \cdot \mathrm{z}^i$$

Binary system

- System with base $z = 2 \Rightarrow$ digits 0 a 1

Ex.

$$v_i \; \ldots 2^4 \; 2^3 \; 2^2 \; 2^1 \; 2^0 \; 2^{-1} \; 2^{-2} \; 2^{-3}$$

$$1\;0\;0\;1\;1,1\;0\;1_2$$

$$v(A) = 2^4+2+1+1/2+1/8 = \underbrace{19{,}625}$$

Equivalent representation of A in decimal

- Value of the number ≈ convert from binary to decimal

Decimal → Binary (integer part)

- **Repeated division by 2** (i.e. base of the target system)

Ex. Convert 57_{10} to the binary system.

$$57_{10} \approx A_2$$

$$A_2 = 111001_2$$

57 : 2 = 28	*remainder*	1	… a_0
28 : 2 = 14	*remainder*	0	
14 : 2 = 7	*remainder*	0	
7 : 2 = 3	*remainder*	1	
3 : 2 = 1	*remainder*	1	
1 : 2 = 0	*remainder*	1	… a_5

Note. The number is written with the remainders in reverse order.

Decimal → Binary (fractional part)

- **Repeated multiplication by 2** (i.e. base of the target system)

Ex. Convert $0{,}65625_{10}$ to the binary system.

$$0{,}65625_{10} \approx A_2$$

$$A_2 = 0{,}10101_2 \leftarrow \left\{ \begin{array}{llll} 0{,}65625 & \cdot\, 2 = 1 & , & 3125 \quad \ldots\, a_{-1} \\ 0{,}3125 & \cdot\, 2 = 0 & , & 625 \\ 0{,}625 & \cdot\, 2 = 1 & , & 25 \\ 0{,}25 & \cdot\, 2 = 0 & , & 5 \\ 0{,}5 & \cdot\, 2 = 1 & , & 0 \quad \ldots\, a_{-5} \end{array} \right.$$

Exercise: Convert between binary and decimal systems.

1. $11010001{,}11_2$ → $?_{10}$
2. 1111111_2 → $?_{10}$
3. $1{,}011001_2$ → $?_{10}$

4. $147{,}15625_{10}$ → $?_2$
5. $1345{,}125_{10}$ → $?_2$

Exercise: Convert between binary and decimal systems.

1. $11010001{,}11_2$ → $209{,}75_{10}$
2. 1111111_2 → 127_{10}
3. $1{,}011001_2$ → $1{,}390625_{10}$

4. $147{,}15625_{10}$ → $1001\ 0011{,}0010\ 1_2$
5. $1345{,}125_{10}$ → $101\ 0100\ 0001{,}0010_2$

Important powers of two

n	2^n	Dec.
0	2^0	1
1	2^1	2
2	2^2	4
3	2^3	8
4	2^4	16
5	2^5	32
6	2^6	64
7	2^7	128

n	2^n	Dec.
8	2^8	256
9	2^9	512
10	2^{10}	1 024
11	2^{11}	2048
12	2^{12}	4096
13	2^{13}	8192
14	2^{14}	16384
15	2^{15}	32768
16	2^{16}	65 536

n	2^n	Dec.
20	2^{20}	1 M
30	2^{30}	1 G
32	2^{32}	4 G
40	2^{40}	1 T
-1	2^{-1}	0,5
-2	2^{-2}	0,25
-3	2^{-3}	0,125
-4	2^{-4}	0,0625

Important!

Hexadecimal system

- Number representation consists of digits 0..9 a A..F

Hex.	Dec.	Bin.
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111

Hex.	Dec.	Bin.
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

Memorize this!

Binary ←→ Hexadecimal

- Related systems: z_{16} = 16 = 2^4 = $z_2{}^4$
- ⇒ One digit in z_{16} corresponds to four digits in z_2
  ⇒ **Only a formal difference between representations in z_{16} and z_2.**

Ex. Convert numbers between related systems:

a) $1001101{,}01011_2$

$0100\ 1101\ ,\ 0101\ 1000_2$

$4\ \ D\ ,\ 5\ \ 8_{16}$

b) $734{,}051_{16}$

$0111\ 0011\ 0100{,}0000\ 0101\ 0001_2$

Exercise: Convert between binary and hexadecimal.

1. $101101011{,}010111_2 \rightarrow ?_{16}$
2. $11101011101010 0_2 \rightarrow ?_{16}$
3. $0{,}0011010111001_2 \rightarrow ?_{16}$
4. $12A5F{,}1_{16} \rightarrow ?_2$
5. $F563D{,}8_{16} \rightarrow ?_2$
6. $0{,}98736_{16} \rightarrow ?_2$

Exercise: Convert between binary and hexadecimal.

1. $101101011{,}010111_2 \rightarrow 16B{,}5C_{16}$
2. $111010111010100_2 \rightarrow 75D4_{16}$
3. $0{,}0011010111001_2 \rightarrow 0{,}35C8_{16}$

4. $12A5F{,}1_{16} \rightarrow 1\ 0010\ 1010\ 0101\ 1111{,}0001_2$
5. $F563D{,}8_{16} \rightarrow 1111\ 0101\ 0110\ 0011\ 1101{,}1_2$
6. $0{,}98736_{16} \rightarrow 0{,}1001\ 1000\ 0111\ 0011\ 0110_2$

Addition in the binary system

- The basic operation:
  addition of two 1-bit numbers

+	0	1
0	0	1
1	1	10

Carry to higher order. (the carry is the first 1 of 10)

Ex. Add together 0101_2 and 1110_2.

		0	1	0	1
+		1	1	1	0
carry		1	1	0	0
	1	0	0	1	1

Carry from the i-th order adds to the digits in the (i+1)-th order.

Note. The sum of two *N*-digit numbers can produce a (*N*+1)-digit number.

Exercise: Add in the binary system.

1. $10110001_2 + 11001101_2 = ?_2$
2. $1111_2 + 1111_2 = ?_2$
3. $111010_2 + 110_2 = ?_2$

Exercise: Add in the binary system.

1. $10110001_2 + 11001101_2 = 1\ 0111\ 1110_2$
2. $1111_2 + 1111_2 = 1\ 1110_2$
3. $111010_2 + 110_2 = 100\ 0000_2$

Multiplication in the binary system

- The basic operation:
multiplication of two 1-bit numbers

×	0	1
0	0	0
1	0	1

- Multiplication → repeated addition (and shifting)

Ex. Multiply 1110_2 and 101_2.

```
       1 1 1 0
  ×      1 0 1
  -------------
       1 1 1 0     … 1 × (1 1 1 0)
  +  0 0 0 0       … 0 × (1 1 1 0)
  +1 1 1 0         … 1 × (1 1 1 0)
  -------------
 1 0 0 0 1 1 0
```

Note. The product of *N*- and *M*-digit numbers can produce a (*N*+*M*)-digit number.

Exercise: Multiply in the binary system.

1. $1010_2 \times 101_2 = ?_2$
2. $100000_2 \times 1101_2 = ?_2$
3. $1111_2 \times 1111_2 = ?_2$

Exercise: Multiply in the binary system.

1. $1010_2 \times 101_2 = 11\ 0010_2$
2. $100000_2 \times 1101_2 = 1\ 1010\ 0000_2$
3. $1111_2 \times 1111_2 = 1110\ 0001_2$

Radix grid

- **Defines the format of representable numbers** in the computer (i.e. defines the highest order *n* and the lowest order *-m*)

Ex.

n = 3, -m = 0 n = 0, -m = -3

- Basic properties:
  - **Length** (l) – number of orders (positions).
  - **Unit** (ε) – the least nonzero number that is representable
  - **Modulus** (Z) – the least number that is **not** representable

Exercise: Determine properties of radix grids.

- Determine the properties of the following radix grids:

  a) 

  b) 

  c) 

- Determine the properties in a general form, i.e. depending on *n* and *–m*.

Exercise: Determine properties of radix grids.

- Determine the properties of the following radix grids:

a)  $l = 8,\ Z = 2_{10},\quad \varepsilon = (2^{-7})_{10}$

b)  $l = 8,\ Z = (2^{8})_{10},\quad \varepsilon = 1$

c)  $l = 6,\ Z = (2^{3})_{10},\quad \varepsilon = (2^{-3})_{10}$

- Determine the properties in a general form, i.e. depending on *n* and *–m*.

$$l = n + m + 1,\qquad Z = z^{n+1},\qquad \varepsilon = z^{-m}$$

Representing negative numbers

- Standard radix systems $\Rightarrow$ nonnegative numbers only
- Representing negative numbers $\Rightarrow$ **codes**
  - describe the transformation from a finite set of integers into a finite set of nonnegative numbers
- Most frequently used codes:
  - sign-magnitude (“direct”)
  - biased (excess-N, “additive”)
  - complement (two’s complement, ten’s complement)

Two's complement code I

- Representation of a negative number A is its complement with respect to the modulus Z of the radix grid
- Formal definition:

$$\mathcal{D}(A) = \begin{cases} A & \text{for} \quad 0 \le A < \tfrac{1}{2}Z \\ Z + A & \text{for} \quad -\tfrac{1}{2}Z \le A < 0 \end{cases}$$

Ex. The images of +5 a –5 (z = 2, Z = 16).

$\mathcal{D}(5) = 5_{10} = 101_2$ … $+101_2$ $\xrightarrow{\mathcal{D}}$ 0 1 0 1

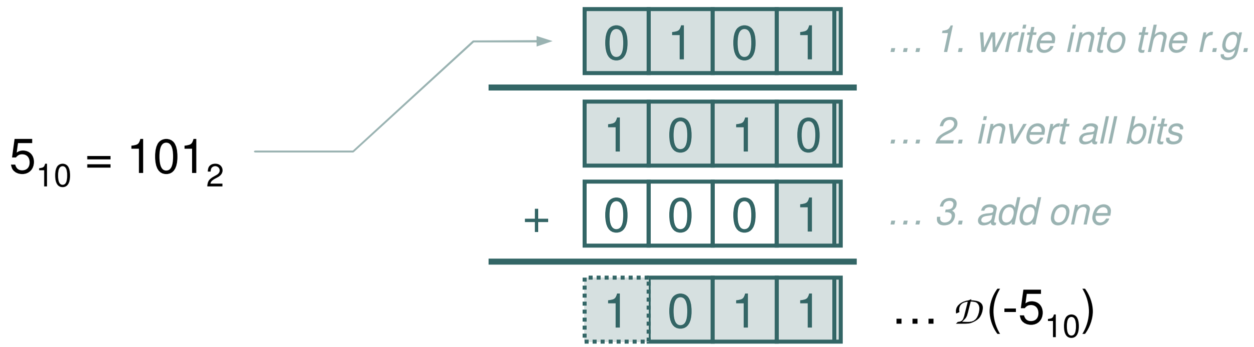
$\mathcal{D}(-5) = 16_{10} + (-5_{10}) = 11_{10} = 1011_2$ … -101_2 $\xrightarrow{\mathcal{D}}$ 1 0 1 1

the highest bit represents the sign

Two's complement code II

- Algorithm to determine the image of a negative number in the two's complement code:
  1. Write the number A_2 into the radix grid.
  2. Invert all bits.
  3. Add one.

Ex. The image of –5 (z = 2, Z = 16).



Exercise: Represent numbers in radix grids.

- Convert the numbers to binary and write them in two’s complement code into the radix grids of the following parameters.

1. $(-9)_{10} = ?_2$ $n = 5,\ m = 0$
2. $(-17)_{10} = ?_2$ $n = 7,\ m = 0$
3. $(-6C)_{16} = ?_2$ $n = 8,\ m = 0$
4. $(-0)_{16} = ?_2$ $n = 4,\ m = 0$

Exercise: Represent numbers in radix grids.

- Convert the numbers to binary and write them into the radix grids of the following parameters.

1. $(-9)_{10}$ = $?_2$ $n = 5$, $m = 0$ … 110111_2
2. $(-17)_{10}$ = $?_2$ $n = 7$, $m = 0$ … 11101111_2
3. $(-6C)_{16}$ = $?_2$ $n = 8$, $m = 0$ … 110010100_2
4. $(-0)_{16}$ = $?_2$ $n = 4$, $m = 0$ … 00000_2

Subtraction in binary

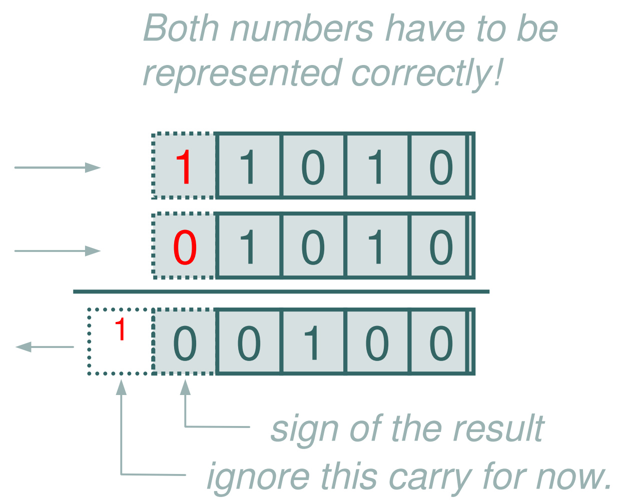
- Subtraction ≈ addition of the negated value

Ex. Compute $10_{10} - 6_{10}$ (in binary).

$6_{10} = 00110_2$ → $-6_{10} = 11010_2$

$+10_{10} = 01010_2$

$4_{10} = 00100_2$



Exercise: Subtract in binary.

- Convert the numbers to the binary system (where necessary). Subtract.

1. $6_{10} - 10_{10} = ?_2$
2. $7_{10} - 7_{10} = ?_2$
3. $1001_2 - 0110_2 = ?_2$
4. $F1_{16} - 3_{16} = ?_2$

Exercise: Subtract in binary.

- Convert the numbers to the binary system (where necessary). Subtract.

1. 6_{10} - 10_{10} = $1\ 1100_2$
2. 7_{10} - 7_{10} = 0_2
3. 1001_2 - 0110_2 = $0\ 0011_2$
4. $F1_{16}$ - 3_{16} = $0\ 1110\ 1110_2$