Logical basics of digital computers

Numeral systems

Numeric system **is a system for displaying numbers using characters**

Numeral systems

- **non positional :**
- **During the Roman Empire or Ancient Greece, the Roman numerals were used, the value of which did not depend on the position where the digit number is located.**
- **Equivalents of Roman and decimal numerals gives the following table:**

Decimal numeral	1	2	3	4	5	6	7	8	9
Roman numeral	I	II	III	IV	V	VI	VII	VIII	IX
Decimal number	10	50	100	500	1000				
Roman number	X	L	C	D	M				

Numeral systems

- **positional:**
- **numbers and theirs record, as we know them today introduced Arabs**
- **Recording numbers using characters 0 to 9 and position to express units, hundreds, tens, etc.**
- **Positional notation is one that reflects any polynomial number N:**

$$N = Z_n P^n + Z_{n-1} P^{n-1} + \ldots + Z_1 P^1 + Z_0 P^0 + Z_{-1} P^{-1} + Z_{-2} P^{-2} + \ldots +$$

$$+ Z_{-m} P^{-m} = \sum_{i=-m}^{n} Z_i P^i$$

P – base of numeral system,
Z_i – characters used in numeral sys.
$i \in \langle -m, n \rangle$.

Base of numeral system

- **can be any number**
- **practical significance in terms of information technology are just some of the numeral system:**
- **decimal (decimal) – base NS = 10**
- **usable characters: 0,1,2,3,4,5,6,7,8,9**
- **binary (Binary) – base NS = 2**
- **applicable codes: 0,1**
- **Octal - base NS = 8**
- **applicable codes: 0,1,2,3,4,5,6,7**
- **Hexadecimal – base NS = 16**
- **usable characters: 0,1,2,3,4,5,6,7,8,9, A, B, C, D, E and F.**

Decimal NS

- In everyday life, the most often used NS in calculations is the decimal numeric system that uses 10 characters (numbers 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9).
- Eg. decimal number 328,75 can be entered as: $3x10^2 +2 x10^1 +8 x10^0 +7 x10^{-1} +5 x10^{-2}$
- The full registration of each decimal number can be entered using the polynomial:

$$Z_n 10^n + Z_{n-1} 10^{n-1} + Z_{n-2} 10^{n-2} + \ldots + Z_1 10^1 + Z_0 10^0 + Z_{-1} 10^{-1} + Z_{-2} 10^{-2} + \ldots +$$

$$+ Z_{-m+1} 10^{-m+1} + Z_{-m} 10^{-m}$$

Base of numeral system
P = 10

Binary NS

- binary notation makes sense since the founding of the first electronic computers,
- the fastest and most reliable electronic components of PCs are those, which have two stable states,
- the physical elements of their activities directly models characters of binary notation,
- all information in the present computers are stored using two digits: 0 and 1 (not voltage = 0, the voltage = 1).
- Basic unit of information is called 1 bit (Binary digit - a binary number).

Writing numbers in binary notation

P = 10	P = 2	
0	0.2^0	0
1	1.2^0	1
2	$1.2^1+0.2^0$	10
3	$1.2^1+1.2^0$	11
4	$1.2^2+0.2^1+0.2^0$	100
5	$1.2^2+0.2^1+1.2^0$	101
6	$1.2^2+1.2^1+0.2^0$	110
7	$1.2^2+1.2^1+1.2^0$	111
8	$1.2^3+0.2^2+0.2^1+0.2^0$	1000
9	$1.2^3+0.2^2+0.2^1+1.2^0$	1001
10	$1.2^3+0.2^2+1.2^1+0.2^0$	1010

Octal and hexadecimal NS

- **Base of Octal numerical system is P = 8, and allowed characters are the digits 0, 1, 2, 3, 4, 5, 6, 7 (Zi)**
- **Base of hex (hexadecimal) notation is P = 16 and allowed characters are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E and F (Zi)**
- **In NS with lower basis < 10, no problem with characters. Use the same digits as decimal from 0 to P.**
- **In hexadecimal notation, it is necessary to add six more characters to decimal digits - a big alphabet letters.**

Numbers in Octal and Hexadecimal NS

P = 10	P = 8		P = 16	
0	0.8^0	0	0.16^0	0
1	1.8^0	1	1.16^0	1
2	2.8^0	2	2.16^0	2
3	3.8^0	3	3.16^0	3
4	4.8^0	4	4.16^0	4
5	5.8^0	5	5.16^0	5
6	6.8^0	6	6.16^0	6
7	7.8^0	7	7.16^0	7
8	$1.8^1+0.8^0$	10	8.16^0	8
9	$1.8^1+1.8^0$	11	9.16^0	9
10	$1.8^1+2.8^0$	12	10.16^0	A
11	$1.8^1+3.8^0$	13	11.16^0	B
12	$1.8^1+4.8^0$	14	12.16^0	C
13	$1.8^1+5.8^0$	15	13.16^0	D
14	$1.8^1+6.8^0$	16	14.16^0	E
15	$1.8^1+7.8^0$	17	15.16^0	F
16	$2.8^1+0.8^0$	20	$1.16^1+0.16^0$	10
17	$2.8^1+1.8^0$	21	$1.16^1+1.16^0$	11
18	$2.8^1+2.8^0$	22	$1.16^1+2.16^0$	12
19	$2.8^1+3.8^0$	23	$1.16^1+3.16^0$	13
20	$2.8^1+4.8^0$	24	$1.16^1+4.16^0$	14

NS in terms of computer work

- The computer works with numbers in binary notation.
- Binary numbers are usually very long and disarranged sequence of zeros and units.
- For easier recording of binary numbers are used octal or hexadecimal notation.

Transfer between binary, octal and hexadecimal is very simple.

Transfer numbers from decimal NS to system with base P

The best way to transfer is dividing decimal numbers *N10* with the base P and recording residues after dividing, which is actually the number *NP* in the chosen notation.

Transferring numbers is realized by repeated division by based P.

Example:

Transfer of the number $[39]_{10}$ to binary numeral system.

39:2 = 19	1	
19:2 = 9	1	
9:2 = 4	1	Remains after cutting
4:2 = 2	0	
2:2 = 1	0	
1:2 = 0	1	Way of the reading after the transfer.

$[39]_{10} = [100111]_2$

Test of an accuracy: $1.2^0 + 1.2^1 + 1.2^2 + 0.2^3 + 0.2^4 + 1.2^5 =$
$=1+2+4+32 = [39]_{10}$

Transfers between numeral systems

Transfer from a numerical system with base P to decimal is easy.

Transfer via calculating a formula:

$$N = \sum_{i=-m}^{n} Z_i . P^i$$

Transfer of the binary number system [1 1 0 1 0 1 0 1 0 1]2 **to decimal.**

orders

9. 8. 7. 6. 5. 4. 3. 2. 1. 0.

$[1\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1]_2 =$

$= 1.2^0 + 0.2^1 + 1.2^2 + 0.2^3 + 1.2^4 + 0.2^5 + 1.2^6 + 0.2^7 + 1.2^8 + 1.2^9 =$

$= 1 + 0 + 4 + 0 + 16 + 0 + 64 + 0 + 256 + 512 = [853]_{10}$

Transfer numbers from decimal NS to system with base P

Transfer from decimal to octal NS:

Example:

Transfer of the number $[250]_{10}$ to octal numeral system.

250 : 8 = 31	2	Remains after cutting
31 : 8 = 3	7	
3 : 8 = 0	3	

$[250]_{10} = [372]_8$

Test of an accuracy: $2.8^0 + 7.8^1 + 3.8^2 = 2+56+192 = [250]_{10}$

Transfer between binary and octal notation:

Rule for bases: $2^3 = 8^1$,

-> three orders of magnitude of binary number are shown by one order of magnitude of octal number.

Transfer of the number $[11101011010]_2$ *to octal numeral system.*

Conversion process is as follows:

- *divide the number by three digits from right to left,*
- *each triplet of digits is converted to an octal numeral system.*

$$[\ 11\,|\,101\,|\,011\,|\,010]_2 = [3\ 5\ 3\ 2]_8$$

←

Transfer of the number $[351]_8$ *to binary numeral system.*

Process: each digit of octal number is converted to three-digit number in binary numeral system (from the left enter zero, for example number $[1]_8$ *is* $[001]_2$*).*

$$[\ 3\quad 5\quad 1\ ]_8 = [\ 11\ 101\ 001]_2$$

011 101 001

Transfer between binary and hexadecimal notation:

Rule for bases : $2^4 = 16^1$***,***

-> four orders of magnitude of binary number are shown by one order of magnitude of hexadecimal number.

Transfer of the number $[11101011010]_2$ to hexadecimal numeral system.

Conversion process is as follows:

- *divide the number by four digits from right to left,*
- *each quartet of digits is converted to an hexadecimal numeral system.*

$$[\ 111\ 0101\ 1010]_2 = [7\ 5\ A]_{16}$$

Transfer of the number $[B71C]_{16}$ to binary numeral system.

Process: each digit of hexadecimal number is converted to four-digit number in binary numeral system (from the left enter zero, for example number $[1]_{16}$ is $[0001]_2$).

$$[\ B\quad 7\quad 1\quad C\ ]_{16} = [\ 1011\ 0111\ 0001\ 1100]_2$$

1011 0111 0001 1100

Arithmetic operations in binary NS

- are easier than in decimal system,
- rules for the implementation of arithmetic operations should be remembered only for two digits, which may occur in a row.
- The rules for arithmetic operations in binary notation are as follows:

Addition:	Subtraction:	Multiplication:
0 + 0 = 0	0 – 0 = 0	0 x 0 = 0
0 + 1 = 1	1 – 0 = 1	0 x 1 = 0
1 + 0 = 1	1 – 1 = 0	1 x 0 = 0
1 + 1 = 10	10 – 1 = 1	1 x 1 = 1

Examples of binary addition

Ex.1.: Addition of two numers $[9]_{10}$ and $[6]_{10}$ in binary NS.

$[6]_{10} = [\ \ 1\ 1\ 0]_2$
$[9]_{10} = [1\ 0\ 0\ 1]_2$

$[15]_{10} = [1\ 1\ 1\ 1]_2$

Ex.1.: Addition of two numers [94]10 and [90]10 in binary NS.

When adding two units[1] in a row, it is used a transfer to higher order

1 1 1 1

$[94]_{10} = [1\ 0\ 1\ 1\ 1\ 1\ 0]_2$
$[90]_{10} = [1\ 0\ 1\ 1\ 0\ 1\ 0]_2$

$[184]_{10} = [1\ 0\ 1\ 1\ 1\ 0\ 0\ 0]_2$

The End