Digital technology

Numbering systems

Jari Hautamäki

jamk.fi

Numbering systems

Floating point number

- If the number is presented by using the complement of two
  - E.g.with an 8-bit processor only numbers -128...+127 can be presented
  - Even with a 16-bit processor only numbers -32768...+32767 are reached
  - Bigger numbers than this cannot be presented with the reserved numbers of bits
  - Direct presentation is not possible with fractions.
- Hence, in computers Floating Point Representation is used:

$$L = S*K^E$$

- L = number to be presented
- S = significand
- K = base number (in binary number presentation 2)
- E = exponent

Numbering systems

Floating point representation $L = S*K^E$

- Enables a much wider number range representation with the same number of bits as in straight?direct representation

  Examples:

  $2000000 = 2*10^6$

  $0.00\mathbf{10101}_2 = \mathbf{10101}_2*2^{-7}$ $(0.1640625 = 21*2^{-7})$

- Can represent fractions but the precise value of zero cannot be represented

Numbering systems

Floating point representation

- In the floating point representation the number is always normalized i.e., most significant bit is one
- E.g., with 8-bit significand the number 00101001_2*2^5 normalized is 10100100_2*2^3
- This type of prepresentation where the first bit is **always 1**, makes it possible that in the representation not even most significant bit needs to be shown.
  - Thus, one bit is saved
- E.g. a 32-bit floating number:

	+/-	<-	EKSPONENTTI	->	<-	MANTISSA		->
BITTI	31	30		24	23		1	0

  - Bit 31: Sign bit of significant: 1 = - and 0 = +
  - Bits 30...24: 7-bit exponent
  - bits 23...0: 24-bit significand

Numbering systems

Floating point representation

- **Exponent** is complement of two in representation when the number range is +63...-64

+63 = 0,111111

...

+1 = 0,000001

0 = 0,000000

- 1 = 1,111111

...

-64 = 1,000000

- **The representation of the significand** contains 25 bits although in the bit string?jono there are 24
  - Most significant bit can be omitted from the bit string
  - The number is normalized to the range 1...0,5, so that number 0.111111...˜1 and number 0.100000... ˜ 0,5.

Numbering systems

Floating point representation

- 32-bit floating number absolute value is in the range as represented previously
- $0.1111...*2^{63}$ ---- $0.1*2^{-64}$ i.e. with decimal numbers $0,999...*2^{63}$ ---- $0,5*2^{-64}$ i.e. $9,2*10^{18}$ - $2,7*10^{-20}$
- With a direct 32-bit number presentation we could only reach number range $+2^{31}...-(2^{31}-1)$

- Example: Represent number 238_{10} as a 32-bit floating number?
  - Convert number 238_{10} to a binary number
    -> $128+64+32+8+4+2=11101110_2$
  - Convert the representation of the number as for significant to range 1...0,5
    > $0.1110111*2^8$
- Now the floating point representation is in format:
  - M=110111 (Note: the first one could be omitted)
  - $E=8_{10}=1000_2$
  - Sign bit =0=positive number
- I.e. the Floating point representation is: 0|0001000|11011100000000000000000
-

Exercises

6. Represent in 32-bit floating number

a) -123

b) 0,56640625

c) 0,01953125

7. What is the biggest and smalles positive number that can be represented with floating number representation with 21 bits? In the character string the expontent=4 bits and significand =16 bits long.