

# Számítógép-architektúrák

## Számábrázolások

(vázlat)

A számrendszereket (különösen a 2-es, 10-es és 16-os) készség szintjén ismertnek tekintjük.

Számítógépek esetén a pozitív egész számok tárolása a legegyszerűbb esetben alapvetően megfelel a 2-es számrendszerbeli ábrázolásnak. A számokat alkotó biteket 8-asával byte-okban tároljuk. Több byte-os adatok esetén a byte-ok sorrendje általában vagy az *LSB* (Least Significant Byte First = legkisebb helyi értékű bájt van elől, vagyis az alacsonyabb memóriacímen) vagy az *MSB* sorrendet követi. (Kommunikáció esetén érdekes még a byte-ok bitsorrendje is, ami *lsb* = least significant bit first, illetve *msb* lehet.)

Bizonyos alkalmazások esetén elterjedt a BCD (Binary Coded Decimal) számábrázolás is, ahol a szám 10-es számrendszerbeli ábrázolásának megfelelő számjegyeket kódolják el binárisan. Amennyiben egy bájton csak egy számjegyet ábrázolnak akkor azt *zónázott* BCD ábrázolásnak nevezik. Lehetőség van azonban arra is, hogy egy bájt felső és alsó 4 bitjén ábrázoljunk egy-egy (tehát bájtonként kettő) decimális számjegyet (ilyenkor a magasabb helyi értékű bitek hordozzák a magasabb helyi értékű BCD számjegyet). Ezt *tömörített* BCD ábrázolásnak nevezzük.

Gyakorlásként írjuk fel az aktuális évszámot 10-es, 2-es és 16-os számrendszerben! Írjuk fel binárisan a memória egymást követő rekeszeinek tartalmát, ha az aktuális évszámot

- binárisan LSB byte sorrendben ábrázoljuk
- zónázott BCD-ben, LSB byte sorrendben ábrázoljuk
- tömörített BCD-ben, LSB byte sorrendben ábrázoljuk

### Negatív számok ábrázolása

Negatív számok ábrázolására – feltehetően az egyszerű aritmetikai műveletvégzés érdekében – leggyakrabban a kettes komplement ábrázolást szokták használni. Egy binárisan ábrázolt szám kettes komplementjét megkapjuk, ha először a szám egyes komplementjét képezzük (0 helyett 1, 1 helyett 0 lesz) majd hozzáadunk 1-et.

Gyakorlásként írjuk fel 8 biten a következő számokat kettes komplement alakban: -5, -64, -100. Számoljuk ki a következőket:  $10+(-5)$ ,  $(-64)+64$ ,  $110+(-100)$ .

### Tört számok ábrázolása

Tört számok ábrázolására egyaránt használhatunk *fixpontos* és *lebegőpontos* megoldást is.

Fixpontos megoldásnál a kettesdespont helyét rögzítjük: megmondjuk, hogy hány kettes jegyig tároljuk a törtrészt. (Például: 32 bitből használunk 8 bitet a törtrész és 24 bitet az egészek ábrázolására; használjuk az LSB sorrendet.) A kettes komplement ábrázolás itt is természetes módon adódik.

Lebegőpontos megoldásra használhatjuk az IEEE 754 lebegőpontos számformátumot [W1].

A 4 byte-os formátum röviden: 1 bit előjel (0: +, 1: -), 8 biten tároljuk a +127-tel eltolt karakterisztika értékét, 23 biten tároljuk a mantissza kettesdespont utáni részét (előtte ugyanis mindig 1-es van).

Gyakorlásként írjuk fel a 253,75-öt a fent megadott a kétféle formátumban!

A 4 byte-os IEEE 754 formátum előállítását a mellékelt példaprogramokkal ellenőrizhetjük.

Forrás:

W1: [http://hu.wikipedia.org/wiki/IEEE\\_lebeg%C5%91pontos\\_sz%C3%A1mform%C3%A1tum](http://hu.wikipedia.org/wiki/IEEE_lebeg%C5%91pontos_sz%C3%A1mform%C3%A1tum)