Tízes számrendszer

A mindennapi életünkben gyakran találkozunk számokkal. A legtöbb általánosan használt számot úgynevezett tízes számrendszerben ábrázoljuk.
Vegyük például az 5678 decimális számot (öt**ezer**-hat**száz**-hetven-nyolc). Ha a szám értékére vagyunk kíváncsiak, akkor a következő formában határozhatjuk meg:

$$5678 = 5 \cdot 10^3 + 6 \cdot 10^2 + 7 \cdot 10^1 + 8 \cdot 10^0$$

A fenti formában az egyes számjegyeket szoroztuk a számrendszer alapjának (10) megfelelő kitevőjű hatványaival ezeket nevezzük helyiértékeknek. Az egyes számjegyek 0..9 közötti értékeket vehetnek fel. Jelöljük a számrendszer alapját r-el, az egyes számjegyeket h_i-vel, ahol i egy futó indexet jelöl, mely a fenti példánkban 0..3 közötti értékeket vehet fel. Ezen jelölésekkel a következő képen írhatjuk fel tömörebb alakban egy tetszőleges N egész szám értékének kiszámítását:

$$N = \sum_{i=0}^{n} h_i \cdot r^i$$

$$h_i \in \{0,1,2,3,4,5,6,7,8,9\}$$

$$5678 = \sum_{i=0}^{3} h_i \cdot r^i = 5 \cdot 10^3 + 6 \cdot 10^2 + 7 \cdot 10^1 + 8 \cdot 10^0 = 5000 + 600 + 70 + 8$$

Vagyis $h_3=5$, $h_2=6$, $h_1=7$, $h_0=8$; $r^3=10^3=1000$, $r^2=10^2=100$, $r^1=10^1=10$, $r^0=10^0=1$

Kettes számrendszer

A számítástechnikában elterjedten használnak kétállapotú jeleket. Ezek elsősorban logikai jelek, azonban később látni fogjuk, hogy a logikai jelek leírására használt Boole algebrát felhasználva készíthetünk aritmetikai egységet is, mely a kettes számrendszeren alapul. A fenti jelöléseket használva a kettes számrendszer esetén r = 2, és az egyes h_i számjegyek kizárólag 0 vagy 1 értéket vehetnek fel.

$$h_i \in \{0,1\}$$

Átváltás 10→2

Egy tízes számrendszerben ábrázolt számot egyszerűen átalakíthatunk kettes számrendszerbe, ha elkezdjük kettővel osztani majd minden osztásnál külön vesszük az osztási maradékokat. Határozzuk meg a fenti példában szereplő 5678 decimális szám kettes számrendszerbeli alakját.

Osztás	**Maradék**	**Számjegyek (h_i)**	**Helyi értékek (r^i)**
5678:2 = 2839	0	= h_0	1
2839:2 = 1419	1	= h_1	2
1419:2 = 709	1	= h_2	4
709:2 = 354	1	= h_3	8
354:2 = 177	0	= h_4	16
177:2 = 88	1	= h_5	32
88:2 = 44	0	= h_6	64
44:2 = 22	0	= h_7	128
22:2 = 11	0	= h_8	256
11:2 = 5	1	= h_9	512
5:2 = 2	1	= h_{10}	1024
2:2 = 1	0	= h_{11}	2048
1:2 = 0	1	= h_{12}	4096

Vagyis $(5678)_{10} = (1011000101110)_2$
Megfigyelhetjük, hogy amíg tízes számrendszerben négy számjegy elegendő volt az ábrázoláshoz, addig kettes számrendszerben összesen tizenhárom számjegyre volt szükségük.
A sok leírt számjegy közül a két szélsőre gyakran hivatkozunk az MSB (Most Significant Bit = legmagasabb helyiértékű bit) és az LSB (Least Significant Bit = legalacsonyabb helyiértékű bit) rövidítésekkel.

Átváltás 2→10

Kettes számrendszerben ábrázolt számok decimális megfelelőjét többféleképpen is kiszámíthatjuk. Kevés számjegy esetén talán a legegyszerűbb, ha egyszerűen összeadjuk az 1 értékű számjegyekhez tartozó helyi értékeket:
$(1011000101110)_2 = 4096+1024+512+32+8+4+2 = (5678)_{10}$

Ha nem tudjuk a kettő hatványokat, akkor az úgynevezett Horner szabály segítségével is átválthatjuk:
$(1011000101110)_2 = ((((((((((((1\cdot 2+0)\ 2+1)\ 2+1)\ 2+0)\ 2+0)\ 2+0)\ 2+1)\ 2+0)\ 2+1)\ 2+1)\ 2+1)\ 2+0 = (5678)_{10}$
Itt tulajdonképpen a hatványozás a zárójelek és a 2-es kiemelések segítségével jön létre. Például a legbelső zárójelben szereplő számjegy pont 2^{12}-el lesz megszorozva. A fenti számítás előnye, hogy kevesebb művelet elvégzése szükséges, mintha a hatványozást is ismételt szorzásokkal végeznénk.

Tizenhatos számrendszer

A kettes számrendszer mellett elterjedten használjuk még a tizenhatos vagy hexadecimális számrendszert. A fenti jelöléseket használva a hexadecimális számrendszer esetén r = 16, és az egyes h_i számjegyek tizenhat különböző értéket vehetnek fel. Mivel a nyelvünk írásjeleiben nincs ennyi számok ábrázolására szolgáló szimbólum, ezért az ABC néhány betűjét használjuk a hiányzó számjegyek leírására. A tizenhatos számrendszerben minden számjegy 0..9, vagy A..F értéket vehet fel.
$h_i \in \{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$

Átváltás 10→16

Egy tízes számrendszerben ábrázolt számot egyszerűen átalakíthatunk tizenhatos számrendszerbe, ha elkezdjük tizenhattal osztani majd minden osztásnál külön vesszük az osztási maradékokat. Alakítsuk át a fenti példában szereplő 5678 decimális számot tizenhatos számrendszerbeli számmá.

Osztás	**Maradék**	**Számjegyek (h_i)**	**Helyi értékek (r^i)**
5678:16 = 354	14→E	$= h_0$	1
354:16 = 22	2	$= h_1$	16
22:16 = 1	6	$= h_2$	256
1:16 = 0	1	$= h_3$	4096

Vagyis $(5678)_{10} = (162E)_{16}$

Átváltás 2→16

Egész számok esetén, a legkisebb helyi értéktől kezdve négybites csoportokat képezünk a kettes számrendszerbeli alakból, majd ezeket a csoportokat külön-külön átírjuk tizenhatos számjegyekké, akkor megkapjuk a szám tizenhatos számrendszerbeli alakját. A legmagasabb helyi értéken 0-kal pótoljuk a hiányzó számjegyeket.
$(1011000101110)_2 = (0001)(0110)(0010)(1110)_2 = (162E)_{16}$

Tört számok kezelése kettes számrendszerben

Matematikai számítások során gyakran van igény tört számok kezelésére. Ezt az alkalmazott számrendszer alapjának negatív kitevőjű hatványai (-m..-1) segítségével tudjuk előállítani.

$$N = \sum_{i=-m}^{n} h_i \cdot r^i$$

Ilyenkor külön meghatározzuk a szám egész részét, majd egy kettedes ponttal elválasztva mellé írjuk a tört részét.
Például váltsuk át az alábbi négy+2 bites bináris számot decimális alakra.

$$(1011.11)_2 = (1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2})_{10} = (8 + 2 + 1 + 0.5 + 0.25)_{10} = (11.75)_{10}$$

Ezt az ábrázolási módot gyakran hívjuk **fix pontos** ábrázolásnak, mert a számítás során nem változik az egész és a tört rész ábrázolásához használt számjegyek száma.

BCD ábrázolás

A BCD ábrázolás a tízes számrendszerben ábrázolt szám minden számjegyét négybiten, binárisan jeleníti meg. Ez az ábrázolás akkor hasznos, ha tízes számrendszerbeli számokat szeretnénk kezelni bináris formában. A fenti példában ábrázolt számot a következőképen ábrázoljuk BCD formában:
$(5678)_{10} = (0101\ 0110\ 0111\ 1000)_{BCD}$

Kettes komplemens ábrázolás

Az előjeles számok kezelésére szolgáló számábrázolás. Csak előre rögzített bitszámmal működik. Egy n bites ábrázolás esetén az aritmetikai műveletek eredményei moduló 2^n szerint érvényesek (2^n-el vett osztás maradéka).
A **pozitív számok** alakja megegyezik a kettes számrendszerbeli ábrázolással, a **negatív számokat** azonban komplemens formában ábrázoljuk. A komplemens képzése: a negatív szám abszolút értékét kivonjuk a fenti 2^n modulusból. A gyakorlatban a komplemenst egyszerűbb úgy kiszámítani, hogy a negatív szám abszolút értékét (pozitív megfelelőjét) felírjuk binárisan az ábrázolásnak megfelelő bitszámmal (szükség esetén a felső biteken megfelelő mennyiségű nullával kiegészítve), majd bitenként invertáljuk, végül a legkisebb helyiértéken 1-et hozzáadunk.
A példánkban használt számot 16 biten ábrázolva a következőket kapjuk:
$(+5678)_{10} = (0001\ 0110\ 0010\ 1110)_{2kompl}$
$(-5678)_{10} = (1110\ 1001\ 1101\ 0010)_{2kompl}$
A kettes komplemens számábrázolás előnyös tulajdonsága, hogy a legmagasabb helyiértékű bit előjel bitként viselkedik, 1 értékű a negatív és 0 értékű a nem negatív számok esetén. A pozitív és negatív számok kezelésére egységes, bináris aritmetika használható a számok összeadására, kivonására.
A számábrázolási tartomány egyik fele a negatív, másik fele a nem negatív számok (nulla és a pozitív számok) ábrázolására szolgál. Ha n bites kettes komplemens formában egész számokat ábrázolunk, akkor az ábrázolható tartomány határai:

$$-\frac{2^n}{2} \cdots\cdots + \left(\frac{2^n}{2} - 1\right)$$

Ez 8 bit esetén (n=8) azt jelenti, hogy -128…+127 közötti számokat tudunk ábrázolni.