PRG1: A számítógépes adattárolás alapjai

2. lecke: Összeadás kettes számrendszerben

1. A lecke tartalma

- Az összeadás szabályai egy biten
- Az átvitel fogalmak
- Az összeadás szabályai az átvitelt figyelembe véve
- Példák összeadásra

2. Összeadás egy biten

Egy jegyű számok összeadása tízes számrendszerben már az első osztályban szerepel. Hasonló szabályok vonatkoznak a kettes számrendszerben való összeadásra is:

- 0+0 = 0
- 0+1 = 1
- 1+0 = 0
- 1+1 =10

A fentiek közül egyedül az utolsó szabály megértése okozhat gondot: Hogyan lesz 1+1 eredménye 10? Pontosan ugyanúgy, ahogy tízes számrendszerben 2+8 eredménye 10 lesz! Kettes számrendszerben 1+1 eredménye már nem ábrázolható egy számjegyen, mivel az eredmény *kettő*, azaz a számrendszer *alapszáma*, amit 10 alakban kell írnunk.

Ugyanaz a teendő, mint tízes számrendszerben – és minden egyéb helyiértékes számrendszerben –: az 1 értéket átvisszük a következő helyiértékre. Ezt az átvitt értéket nevezzük **átvitel**nek. Ebből az átvitelből azonban még további szabályok is következnek majd. Az alapján, hogy az átvitel egy vagy nulla, a fenti négy szabálynak lesz két változata, azaz összesen nyolc szabályunk van (amelyek egy része a szimmetria miatt elhagyható lenne, de jobb mindet kiírni).

Kettes számrendszerben egy bit összeadásának szabályai a következők:

- 0+0+0 = 00
- 0+0+1 = 01
- 0+1+0 = 01
- 0+1+1 = 10
- 1+0+0 = 01
- 1+0+1 = 10
- 1+1+0 = 10
- 1+1+1 = 11

Az összeadandók közül az első az előző helyiértékről érkezett átvitel, míg az eredmény első számjegye lesz az új átvitel, a második pedig az aktuális helyiérték. Ugyanezt foglalja össze a 1. táblázat is.

érkező átvitel	első számjegy	második számjegy	átvitel	eredmény
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

1. Táblázat: A bitenkénti összeadás műveleti táblázata

Mint látható tehát, ha a szimmetriát nem vesszük figyelembe, akkor összesen nyolc szabályt kell ismerni ahhoz, hogy kettes számrendszerben egy helyiértéken az összeadást elvégezhessük. Ha végiggondoljuk, akkor tízes számrendszerben ezeknek a szabályoknak a száma 200 lesz, mivel a 10 lehetséges számot egymással 100 féleképpen lehet összeadni, és ott is vagy 0 vagy 1 az átvitel, ami kiadja az összesen 200 lehetőséget. Általánosítva: egy n alapú számrendszerben $2n^2$ összeadási szabály létezik két szám összeadásánál egy helyiérték számára.

Mindebből már látható, hogy miért egyszerűbb a kettes számrendszerben működő számolóeszköz megvalósítása, mint bármely más számrendszerben működőé.

3. Több jegyű számok összeadása

Ha továbbra is a tízes számrendszer analógiáját használjuk, akkor két többjegyű szám összeadása úgy történik, hogy a legkisebb helyiértékű számjegyeket összeadjuk, majd haladunk a nagyobb helyiértékek felé, de ezeknél már mindig figyelembe vesszük a keletkezett átviteleket. Íme néhány példa, amin ezeket az összeadási szabályokat ki lehet próbálni:

`-|--|--`	`-|--||-`	`||-||||`	`-|||||-`
00110100	00100111	00101011	10101011
<u>00100111</u>	<u>00110110</u>	<u>01101101</u>	<u>10011110</u>
01011011	01011101	10001000	01001001

A fenti példákban a legfelső sorban a „-” azt jelzi, hogy nem volt átvitel, míg a „|” azt, hogy történt arra a helyiértékre átvitel.

Az utolsó példához kell még egy kis magyarázat. Mint látható, mind a négy példában nyolc bites számok szerepelnek. Már volt róla szó, hogy a számítógép minden számot pontosan meghatározott számú biten ábrázol. Ha a szám kevesebb bitet igényel, akkor a szám által nem igényelt bitek értéke – a szám baloldalán – nulla értéket kap. Ezt a felírási módot szemlélteti az első három példa minden száma. Ám ennek nem csak előnye, hanem hátránya is van, ahogyan az a negyedik példán látszik.

A negyedik példán ugyanis a legmagasabb helyiértéken történő összeadás átvitelt eredményezett. Azonban ezt az átvitelt – amely a kilencedik, 256 értéket jelentő biten 1 értéket adna – már nem lehet sehol felhasználni, hiszen csak nyolc biten tárolható az eredmény. Ezzel az összeadás hibás eredményt adott! Ezt a hibát, amikor az eredmény több bitet igényel, mint amennyi rendelkezésre áll, **túlcsordulás**nak nevezzük. Ha ilyen történik, azt a CPU a *túlcsordulás-jelző bit* 1 értékével jelzi. Ez a jelzőbit az egyik bit az állapotjelző regiszter bitjei közül.

3.1. Számolás 256-os számrendszerben

A tény, hogy a CPU a túlcsordulást egy meghatározott regiszterbit 1 értékével jelzi, a számítás kiterjesztésére is felhasználható. Régen, amikor a processzorok általában csak 8 bites (tehát legfeljebb 255 értékű) értékekkel tudtak csak számolni, kialakult egy szoftveres megoldás a nagyon nagy egész számokkal történő számolásra, amely bár jóval lassabb, mint ha a CPU közvetlenül tud számolni, ám nincs felső határa a számok ábrázolásának. Az eredeti, 8 bites számolás esetében ez valójában egy 256-os alapon történő összeadás volt, amely a következőképpen működött:

1. Tekintsük egyetlen számjegynek a CPU által tárolni képes szóméretet és alkossunk ezzel egy nagyobb (8 bites szavak esetén 256-os) alapszámú számrendszert.
2. Adjuk össze a legkisebb helyiértéknek számító két számot.
3. Ezután ismételjük az összeadást szópáronként, úgy, hogy
   1. ha az előző helyiértéken nem keletkezett túlcsordulás, akkor egyszerűen összeadjuk a számokat.
   2. ha az előző helyiértéken túlcsordulás keletkezett, akkor úgy adjuk össze a számokat, hogy még egyet hozzáadunk.

A 3. lépés feltételének figyelembe vételére sok CPU rendelkezik olyan utasítással, amely a túlcsordulás-bit értékét hozzáadja az összeadás eredményéhez, így támogatva ezt a kiterjesztett összeadási módot.