



4. Fejezet : Az egész számok (integer) ábrázolása

**The Architecture of Computer Hardware
and Systems Software:
An Information Technology Approach**
3. kiadás, Irv Englander
John Wiley and Sons Ó2003

Wilson Wong, Bentley College
Linda Senne, Bentley College

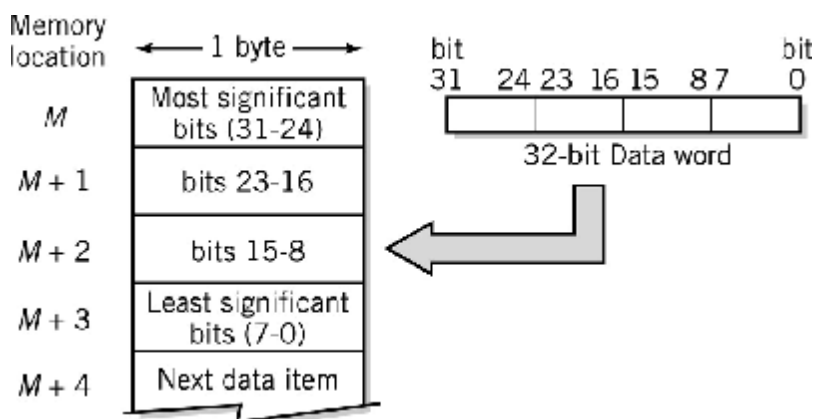


Számábrázolás

- § Egésszámok ábrázolásához a következő információt kell tárolnunk:
 - § Egésszám abszolút értéke (value, magnitude)
 - § Előjel (plusz vagy mínusz)



32 bites adat szó



4. Fejezet: Az integer ábrázolása

4-3



Előjel nélküli egész számok: Integerek

- § Bináris ábrázolás: Előjel nélküli egész szám vagy *integer*
- § BCD ábrázolás: Decimális egész közvetlen *bináris* megfelelője (Binary Coded Decimal)

§ 4 bit: 0-tól 9-ig § 16 bit: 0-tól 9.999-ig
§ 8 bit: 0-tól 99-ig § 32 bit: 0-tól 99.999.999-ig

Decimális érték	Bináris ábrázolás	BCD ábrázolás
68	$= 0100\ 0100$ $= 2^6 + 2^2 = 64 + 4 = \mathbf{68}$	$= 0110\ 1000$ $= 2^2 + 2^1 = \mathbf{6}\quad 2^3 = \mathbf{8}$
99 (legnagyobb 8 biten ábrázolható szám BCD kódolással)	$= 0110\ 0011$ $= 2^6 + 2^5 + 2^1 + 2^0 =$ $= 64 + 32 + 2 + 1 = \mathbf{99}$	$= 1001\ 1001$ $= 2^3 + 2^0\quad 2^3 + 2^0$ $= \mathbf{9}\quad \mathbf{9}$
255 (legnagyobb 8 biten ábrázolható szám bináris kódolással)	$= 1111\ 1111$ $= 2^8 - 1 = \mathbf{255}$	$= 0010\ 0101\ 0101$ $= 2^1\quad 2^2 + 2^0\quad 2^2 + 2^0$ $= \mathbf{2}\quad \mathbf{5}\quad \mathbf{5}$

4. Fejezet: Az integer ábrázolása

4-4



Ábrázolható értékek összehasonlítása: Bináris vs. BCD

- § Ábrázolható értékek BCD kódolással < ábrázolható értékek hagyományos bináris ábrázolás
 - § Bináris: 4 bit 16 különböző értéket képes tárolni (0-tól 15-ig)
 - § BCD: 4 bit csak 10 különböző értéket tud tárolni (0-tól 9-ig)

Bitek száma	BCD határ		Bináris határ	
4	0-9	1 számjegy	0-15	1+ számjegy
8	0-99	2 számjegy	0-255	2+ számjegy
12	0-999	3 számjegy	0-4.095	3+ számjegy
16	0-9.999	4 számjegy	0-65.535	4+ számjegy
20	0-99.999	5 számjegy	0-1 million	6+ számjegy
24	0-999.999	6 számjegy	0-16 million	7+ számjegy
32	0-99.999.999	8 számjegy	0-4 billion	9+ számjegy
64	0-(10 ¹⁶ -1)	16 számjegy	0-16 quintillion	19+ számjegy

4. Fejezet: Az integer ábrázolása

4-5



Hagyományos bináris vs. BCD

- § Bináris ábrázolás kedveltebb
 - § Nagyobb értéket képes tárolni adott számú biten (helyiértéken)
 - § Számolási műveletek elvégzése egyszerűbb
- § BCD gyakran használatos vállalati rendszereknél, főként a tizedes kerekítés és a tizedes pontosság miatt

4. Fejezet: Az integer ábrázolása

4-6



4. Fejezet: Az integer ábrázolása

4-7



- #### 4. Fejezet: Az integer ábrázolása

4-8



Előjel-és-érték

- § A baloldali legszélső bit használata előjelként
 - § 0 = pozitív; 1 = negatív
- § Az ábrázolható értékek száma nem változik
 - § Számok fele pozitív, fele negatív
 - § A legnagyobb ábrázolt érték feleakkora lesz, mint előjel nélkül
- § Példa 8 biten:
 - § Előjel nélküli: $1111\ 1111 = (+)255$
 - § Előjeles: $0111\ 1111 = +127$
 $1111\ 1111 = -127$
 - § Megjegyzés: kettő érték a 0-ra:
 $+0 = 0000\ 0000$ és
 $-0 = 1000\ 0000$

4. Fejezet: Az integer ábrázolása

4-9



Bonyolult számítási algoritmusok

- § Előjel-és-érték algoritmusok összetettek és bonyolultak ahhoz, hogy hardver hajtsa végre
 - § Vizsgálni kell a 0 mindkét értékét
 - § BCD kódolás esetén használható
 - § Az előjeles szám és a carry/átvitel sorrendje hibát okoz
- § Példa: Decimális összeadó algoritmus

Összeadás: 2 pozitív szám	Összeadás: 1 előjeles szám		
$\begin{array}{r} 4 \\ +2 \\ \hline 6 \end{array}$	$\begin{array}{r} 4 \\ -2 \\ \hline 2 \end{array}$	$\begin{array}{r} 2 \\ -4 \\ \hline -2 \end{array}$	$\begin{array}{r} 12 \\ -4 \\ \hline 8 \end{array}$

4. Fejezet: Az integer ábrázolása

4-10



Komplemens ábrázolás

- § A szám előjelét nem kell elkülönítve kezelni
- § Megegyezik minden különböző előjelű bemeneti számkombinációra
- § Két módszer
 - § Alapszám: alap (abszolút) érték
 - § Csökkentett alap: alap értéke mínusz 1
 - 9-es komplemens: a 10-es alap csökkentett alapja
 - 1-es komplemens: a 2-es alap csökkentett alapja

4. Fejezet: Az integer ábrázolása

4-11



9-es decimális komplemens

- § *Komplemens számítás*: az érték kivonása az alap eredeti értékéből
- § Decimális (10-es alap) rendszer csökkentett alapja
 - § Alap mínusz 1 = $10 - 1 \Rightarrow 9$
 - § 3 számjegyű példa: alap eredeti értéke = 999
 - § A számok 0-tól 999-ig terjedhetnek, választóvonal 500-nál

Számok	Negatív		Pozitív	
Megjelenítési mód	Komplemens		Szám önmaga	
Decimális szám terjedelme	-499	-000	+0	499
Számítási algoritmus	999 mínusz szám		nincs	
Ábrázolási példa	500	999	0	499

999 - 499 - növekvő érték +

4. Fejezet: Az integer ábrázolása

4-12



9-es decimális komplement

- § Fontos meghatározni a számjegyek számát vagy a szóhosszúságot
- § Példa: 3 jegyű számok ábrázolása
 - § Első számjegy = 0-tól 4-ig $\Rightarrow$ pozitív szám
 - § Első számjegy = 5-től 9-ig $\Rightarrow$ negatív szám
- § Előjel-és-érték szám konvertálása 9-es komplementű számmá
 - § **321**: eredmény = **321**
 - § **521**: vegyük a komplementjét $(999 - 521) = -478$



Modulus aritmetika (számolás)

- § A műveletek eredménye megadja a (számolás alapjaként használt értékkel) modulus történő osztás maradékát
- § Példa:
 - $4 \bmod 4 = 0$
 - $5 \bmod 4 = 1$
 - $1000 \bmod 999 = 1$
- § A modulusos aritmetika legfontosabb jellemzője
 - § A számolást nullától újratekdjük, ha a számolás során túllépnénk a *moduluson*



Számábrázolás választásának indokai

- § Legyen konzisztens (hasonló logikájú) a megszokott aritmetikával
 - § $-(-érték) = érték$
- § Ha egy értéknek kétszer kiszámoljuk a komplementjét, akkor az eredeti értéket kapjuk vissza:
 - § $komplement = alap - érték$
 - § Egy komplement értékre még egyszer alkalmazzuk a komplement számolást:
 - $alap - komplement = alap - (alap - érték) = érték (!!!)$

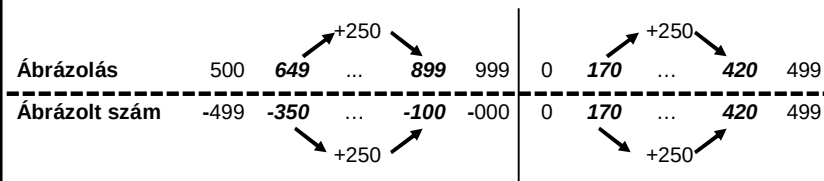
4. Fejezet: Az integer ábrázolása

4-15



Összeadás modulus aritmetikával

- § A számegyenesen való előrehaladás megegyezik az összeadással
- § Példa a 9-es komplementben:
 - § Három helyiértéken ábrázolt számok
 - § A példában a számolás során nem keresztezzük a modulust



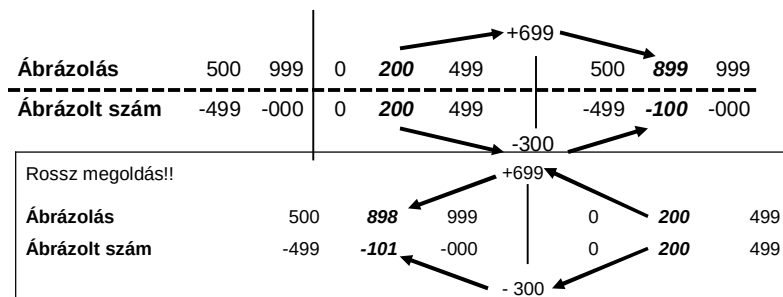
4. Fejezet: Az integer ábrázolása

4-16



Összeadás körkörös számegyenessel

- § Negatív szám hozzáadásakor is haladjunk jobbra a számegyenesen
- § *Körkörös ismétlődő* számegyenest használunk, hogy kiterjesszük a számegyenest a negatív értékek ábrázolására
 - § Balra haladás keresztezné a modulust és helytelen eredményt adna, mert két értéke van a 0-nak (+0 és -0)



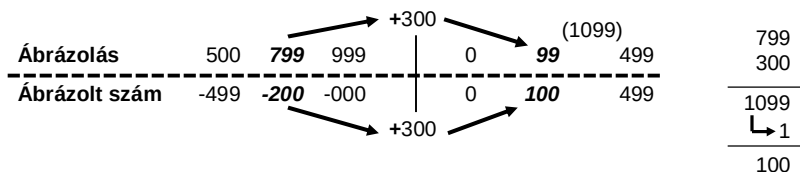
4. Fejezet: Az integer ábrázolása

4-17



Összeadás: túlcsordulás átvitel

- § A jobbra haladás keresztezné a modulust
- § Túlcsordulás átvitel
 - § 2 számjegy összeadása 9-es komplementes aritmetikával
 - § Ha az eredmény több számjegyből áll a meghatározottnál, akkor a carry-t (az átvitelt) az eredményhez adjuk



4. Fejezet: Az integer ábrázolása

4-18



Túlcsordulás

- § Rögzített szó (word) méretnek rögzített a terjedelme
- § Túlcsordulás: számok azon kombinációja, amelyek összege a határon kívül esik
- § Túlcsordulás átvitelrel, ezt a problémát kerüli ki
- § Komplementes aritmetika: *határon kívüli* számoknak ellentétes előjelük van
 - § Teszt: Ha összeadásnál mindkét bemeneti szám azonos előjelű és az eredmény előjele más, akkor túlcsordulás lépett fel

4. Fejezet: Az integer ábrázolása

4-19



1-es bináris komplement

- § *Komplement számítás*: az érték kivonása az alap eredeti értékéből
 - § Bináris (2-es alap) csökkentett alapú komplemente
 - § Alap mínusz 1 = $2 - 1 \iff 1$ mint az alap
- § *Invertálás*: egyesek és nullák felcserélése ($1 \rightarrow 0$, $0 \rightarrow 1$)
 - § 0-val kezdődő számok pozitívak
 - § 1-el kezdődő számok negatívak
 - § Két értéke van a nullának (-0 , $+0$)
- § Példa 8 bites bináris számokkal

Számok	Negatív		Pozitív	
Megjelenítési mód	Komplement		Szám önmaga	
Decimális szám terjedelme	-127_{10}	-0_{10}	$+0_{10}$	127_{10}
Számolás	Invertálás		Nincs	
Ábrázolási példa	10000000	11111111	00000000	01111111

4. Fejezet: Az integer ábrázolása

4-20



1-es bináris komplementens

§ $0101\ 1111_{1kB} = +95_D$

§ $1010\ 0000_{1kB}$

§ negatív

§ a negáltja amely egyben az abszolút értéke is(!!!): $0101\ 1111_B = |95_D|$

§ tehát az eredmény: -95_D



Komplementek közötti váltás

§ Nem lehet közvetlenül 9-es komplementusből 1-es komplementusba váltani

§ 3 számjegyű decimális modulus: 999

▫ Pozitív terjedelem 499

§ 8 bites bináris modulus:

11111111 vagy 255_{10}

▫ Pozitív terjedelem 01111111 vagy 127_{10}

§ Közbeeső lépés: előjel-és-érték ábrázolás


$$0010\ 1101 = +45$$
$$0011\ 1010 = +58$$
$$0110\ 0111 = +103$$

0010 1101 = +45

$$\mathbf{1100\ 0101} = -58$$

1111 0010 = -13

0000 1101

$$8 + 4 + 1 = |\underline{13}|$$

↑
|13|

4-23


$$0010\ 1101 = +45$$

✓ **1110 0001** = **-30**

$$\mathbf{1110\ 0001} = -30$$
$$0001\ 1110 = |30|$$
$$1|0000\ 1110 = \quad +$$
$$\text{L} \rightarrow +1 = 15$$

+15

4-24



Összeadás átvitelrel (carry-vel)

§ 8 bites szám

§ Negatív érték

hozzáadása:

§ Invertálás

0000 0010 (2_{10})

1111 1101

§ Összeadás

§ 9 bit

Túlsordulás átvitelrel

0110 1010 = +106

1111 1101 = -2

10110 0111

+1

0110 1000 = +104

4. Fejezet: Az integer ábrázolása

4-25



Kivonás átvitelrel (carry-vel)

§ 8 bites számábrázolás

§ szám kivonása = szám

-1 szerezésének

hozzáadása

§ invertálás

0101 1010 (90_{10})

1010 0101

§ Összeadás

§ 9 bit

Túlsordulás átvitelrel!

0110 1010 = +106

- 0101 1010 = - (+90)

0110 1010 = +106

+ 1010 0101 = -90

10000 1111

+1

0001 0000 = +16

4. Fejezet: Az integer ábrázolása

4-26



Kivonás

§ 8 bites számbábrázolás $0110\ 1010 = +106$

§ szám kivonása = szám -1 szeresének hozzáadása $- 1111\ 0110 = -(-9)$

§ invertálás
 $1111\ 0110 (-9_{10})$
 $0000\ 1001$

§ összeadás

$$\begin{array}{r} 0110\ 1010 = +106 \\ - 1111\ 0110 = -(-9) \\ \hline 0110\ 1010 = +106 \\ + 0000\ 1001 = +9 \\ \hline 0111\ 0011 = +115 \end{array}$$

4. Fejezet: Az integer ábrázolása

4-27



Kivonás

§ 8 bites szám $0110\ 1010 = +106$

§ invertálás
 $1010\ 0101(90_{10})$
 $0101\ 1010$

§ Összeadás

$$\begin{array}{r} 0110\ 1010 = +106 \\ - 1010\ 0101 = -90 \\ \hline 0110\ 1010 = +106 \\ + 0101\ 1010 = +90 \\ \hline 1100\ 0100 = - \\ 0011\ 1011 = |59| \end{array}$$

Túlcsordulás! Rossz megoldás!!!

4. Fejezet: Az integer ábrázolása

4-28



Túlcsordulás

- § 8 bites szám $0100\ 0000 = +64$
 - § 256 különböző szám
 - § Pozitív számok: $0100\ 0001 = +65$
0 to 127
 - § Összeadás $1000\ 0001 = -126$
 - § Túlcsordulás tesztelése
 - § 2 pozitív bemenetre negatív eredményt adott
 - **túlcsordulás!**
 - § Rossz megoldás!
- Invertáljuk, hogy megkapjuk az értéket
- 126_{10}
- § Programozók óvakodjatok: néhány magas-szintű nyelv pl.: a BASIC néhány verziója nem ellenőrzi a túlcsordulást megfelelően!

4. Fejezet: Az integer ábrázolása

4-29



10-es komplement

- § Komplement rendszer alkotása egy 0-val
- § Alap komplement: használjuk az alapot a komplement műveletekhez
- § 10-es alap: 10-es komplement
- § Példa: Modulus 1000 mint tükrözési pont

Számok	Negatív		Pozitív	
Megjelenítési mód	Komplement		Szám önmaga	
Decimális szám terjedelme	-500	-001	0	499
Számolás	1000 mínusz szám		nincs	
Ábrázolási példa	500	999	0	499

4. Fejezet: Az integer ábrázolása

4-30



Példák 3 jegyű számokkal

§ Példa 1:

§ A 247 mint szám, 10-es *komplement* ábrázolása:

▫ 247 (pozitív szám)

§ A 247 mint szám, 10-es *komplemente*:

▫ $1000 - 247 = 753$ (negatív szám)

§ Példa 2:

§ A 17 mint szám, 10-es *komplemente*:

▫ $1000 - 017 = 983$

§ Példa 3:

§ A 777 mint szám, 10-es *komplemente*:

▫ Negatív szám, mert az első számjegye 7

▫ $1000 - 777 = 223$

▫ Előjeles érték = -223



Alternatív módszer a 10-es komplementre

§ A 9-es komplementen alapszik

§ Példa 3 jegyű számokkal

§ Megjegyzés: $1000 = 999 + |1|$

§ 9-es komplement = $999 - \text{érték}$

§ Átírva

▫ 10-es komplement = $1000 - \text{érték} = 999 + |1| - \text{érték}$

§ Vagy: 10-es komplement = 9-es komplement + $|1|$

§ Számítás egyszerűbb különösen bináris számok esetén



2-es komplement

- § Modulus = 2-es alapú „1”-es után nullák
 - § 8-biten a modulus = 1000 0000
- § Két módszer van megtalálni a komplementet
 - § Kivonjuk a szám értékét a modulusból vagy invertálunk

Számok	Negatív		Pozitív	
Megjelenítési mód	Komplement		Szám önmaga	
Decimális szám értéke	-128₁₀	-1₁₀	+0₁₀	127₁₀
Számolás	Invertálás		Nincs	
Megjelenítési példa	10000000	11111111	00000000	01111111

4. Fejezet: Az integer ábrázolása

4-33



1-es vs. 2-es komplement

- § A választás a számítógép tervezőitől függ
- § 1-es komplement
 - § Egyszerű előjelet váltani
 - § Összeadásnak szüksége van egy extra túlcsondulás átvitelre
 - § Algoritmusnak tesztelnie és konvertálnia kell a „-0”-át
- § 2-es komplement egyszerűbb
 - § Negálás után egy |1| hozzáadás szükséges

4. Fejezet: Az integer ábrázolása

4-34



2-es bináris komplement

§ $0101\ 1111_{2kB} = +95_D$

§ $1010\ 0000_{2kB}$

§ negatív

§ a negáltja amely **nem** az abszolút értéke(!!!):
 $0101\ 1111_B$

§ $|1|$ hozzáadása a negálthoz

§ így: $0101\ 1111_B$

$$+ \quad \quad \quad 1_B$$

$$0110\ 0000_B = |96_D|$$

§ tehát az eredmény: -96_D

4. Fejezet: Az integer ábrázolása

4-35



Egy összetett példa

§ Határozza meg a **decimális** és **hexadecimális** értékét a következő 8 bit-es bináris számoknak, ha **közönséges binárisként**, vagy **1-es komplementként** illetve **2-es komplementként** értelmezzük őket. A hexadecimális számoknál az előjeleket kezelje a decimális számoknál megszokott módon!

§ $0111\ 0100_B$: közönséges: 116_D és 74_H

1-es: $+116_D$ és $+74_H$

2-es: $+116_D$ és $+74_H$

§ $1101\ 0101_B$: közönséges: 213_D és $D5_H$

$|0010\ 1010_B|$ 1-es: -42_D és $-2A_H$

$|0010\ 1011_B|$ 2-es: -43_D és $-2B_H$

4. Fejezet: Az integer ábrázolása

4-36



Egy még összetettebb példa

§ Végezze el a következő **kivonást** 8 bit-es bináris számokkal úgy, hogy a számpár első tagját **1-es komplementesként** a második tagját **2-es komplementesként** értelmezzük. Az eredményt adja meg bináris 1-es komplementesként, majd váltsa át azt decimális számrendszerbe:

§ $1010\ 0101_{B1K} - 1001\ 1010_{B2K} = 1010\ 0101_{B1K} - (0110\ 0101_B + |1|_B = 0110\ 0110_B \Rightarrow) 1001\ 1001_{B1K}$

§
$$\begin{array}{r} 1010\ 0101_{B1K} \\ + 0110\ 0110_B \\ \hline 1\ 0000\ 1011_{B1K} \\ \xrightarrow{\quad\quad\quad} 1_B \\ \hline 0000\ 1100_{B1K} = +12_D \end{array}$$

4. Fejezet: Az integer ábrázolása

4-37



Integer mérete

§ Pozitív számok 0-val kezdődnek

§ Kicsi, negatív számok (közel a 0-hoz) több 0-val kezdődnek

§ $1111\ 1110_{B2K} = -2$ 8-bites 2-es komplement

§ $1000\ 0000_{B2K} = -128$, nagyobb negatív szám 2-es komplemente

§ Cseréljük fel minden 1-est és 0-ást és határozzuk meg az értéket

4. Fejezet: Az integer ábrázolása

4-38



Túlcsordulás és Átvitel közötti különbség

§ Átvitel (carry) bit:

§ Jelzi, ha egy **adott** helyiértéken elvégzett művelet eredménye meghaladja az ott ábrázolható értéket.

§ Túlcsordulás (Overflow):

§ ha egy összeadás vagy kivonás művelet eredménye kívül esik az aktuális számábrázolási tartományon.



Túlcsordulás/Átvitel példák

§ Példa 1:

§ Helyes eredmény

$$0100 = (+4)$$

§ Se túlcsordulás, se átvitel

$$0010 = +(+2)$$

$$0110 = (+6)$$

§ Példa 2:

§ Helytelen eredményt kapunk

$$0100 = (+4)$$

$$0110 = +(+6)$$

$$1010 = (-6)$$

§ Van túlcsordulás, és átvitel is

Négy bit-es 2-es komplementes számábrázolás!



Túlcsordulás/Átvitel példák

§ Példa 3:

§ Az átvitelt elhagyva az eredmény helyes

$$1100 = (-4)$$

$$1110 = +(-2)$$

§ Van átvitel, de nincs túlcsordulás

$$\cancel{1}1010 = (-6)$$

§ Példa 4:

§ Helytelen eredmény

$$1100 = (-4)$$

$$1010 = +(-6)$$

§ Van túlcsordulás, átvitel mellőzve

$$\cancel{1}0110 = (+6)$$

Négy bit-es 2-es komplement számábrázolás!