2. Fejezet : Számrendszerek

The Architecture of Computer Hardware and Systems Software:

An Information Technology Approach

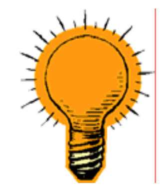
3. kiadás, Irv Englander

Wilson Wong, Bentley College

Linda Senne, Bentley College

Miért bináris?

- Korai számítógépek decimális számábrázolást használtak:
  - Mark I és ENIAC
- Neumann János javasolta a bináris számábrázolást (1945)
  - egyszerűbb a számítógépek megvalósítása
  - utasítások és adatok tárolásához is ezt egyaránt használjuk
- Természetes kapcsolat a kétállású kapcsolók és Bool logikai értékek, ill. műveletek között

	
Be	Ki
Igaz	Hamis
Igen	Nem
1	0

Számolás és aritmetika

- Decimális vagy tízes alapú számrendszer
  - Eredet: számolás az ujjakon
  - “Digit” (számjegy) *digitus* latin szóból ered, jelentése “ujj”
- *Alap*: a számrendszerben használt számjegyek száma, a nullát is beleértve
  - Példa: 10-es alapban 10 számjegy van, 0-tól 9-ig
- *Bináris* vagy *2-es alapú*
- *Bit* (aritmetikai számjegy): 2 számjegy, 0 és 1
- *Oktális* vagy *8-as alapú:* 8 számjegy, 0-tól 7-ig
- *Hexadecimális vagy 16-os alapú:*

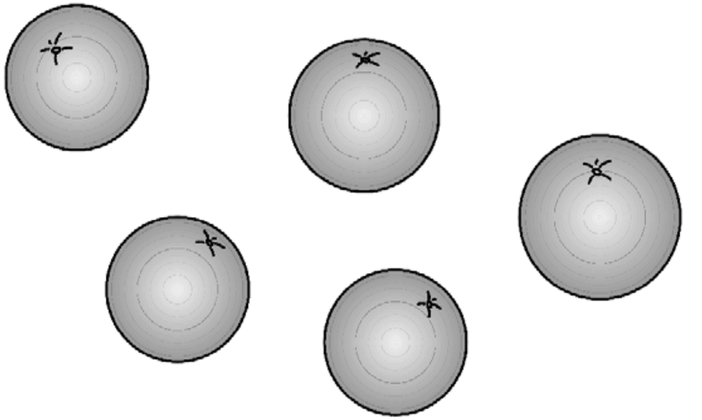
  16 számjegy, 0-tól F-ig
  - Példák: $10_{10} = A_{16}$; $11_{10} = B_{16}$

Bitek kezelése a gyakorlatban

- A bit-eket általában csoportokban tárolják és módosítják
  - 8 bit = 1 *byte*
  - 4 byte = 1 word/szó (a legtöbb rendszerben)
- Számítási műveletek során a számábrázolásra használt bit-ek száma
  - befolyásolja az eredmény pontosságát
  - korlátozza a számítógép által kezelhető számok nagyságát (ill. azok számát).

Számok, ill. azok értékének fizikai reprezentációja

- Különböző számjegyek, ugyanaz az érték
  - Rovás írás: IIIII
  - Római: V
  - Arab: 5
- Különböző alapok, ugyanazok az értékek
  - 5_{10}
  - 101_2
  - 12_3



Számrendszerek

- Római: pozíció vagy helyiérték *független,* (helyiérték nélküli)
- Modern: helyiértékes jelölésen alapszik
  - Decimális számrendszer: **helyiértékes** számábrázolás, 10 hatványain alapszik
  - Bináris rendszer: **helyiértékes** számábrázolás, 2 hatványain alapszik
  - Oktális rendszer: **helyiértékes** számábrázolás, 8 hatványain alapszik
  - Hexadecimális rendszer: **helyiértékes** számábrázolás,16 hatványain alapszik

Helyiértékes számábrázolás: 10-es alap

$43 = 4 \times 10^1 + 3 \times 10^0$

10-esek 1-esek

Helyi érték	10^1	10^0
Érték	10	1
Számítás	4 x 10	3 x1
Összeg	40	3

Helyiértékes számábrázolás : 10-es alap

$527 = 5 \times 10^2 + 2 \times 10^1 + 7 \times 10^0$

100-asok

10-esek

1-esek

Helyi érték	10^2	10^1	10^0
Érték	100	10	1
Számítás	5 x 100	2 x 10	7 x1
Összeg	500	20	7

Helyiértékes számábrázolás: Oktális

$624_8 = 404_{10}$

64-esek 8-asok 1-esek

Helyi érték	8^2	8^1	8^0
Érték	64	8	1
Számítás	6 x 64	2 x 8	4 x 1
Decimális összeg	384	16	4

Helyiértékes számábrázolás: Hexadecimális

$6.704_{16} = 26.372_{10}$

4.096-osok 256-osok 16-osok 1-esek

Helyi érték	16^3	16^2	16^1	16^0
Érték	4.096	256	16	1
Számítás	6 x 4.096	7 x 256	0 x 16	4 x 1
Decimális összeg	24.576	1.792	0	4

Helyiértékes számábrázolás: Bináris

$$1101\ 0110_2 = 214_{10}$$

Helyi érték	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Érték	128	64	32	16	8	4	2	1
Számítás	1 x 128	1 x 64	0 x 32	1 x16	0 x 8	1 x 4	1 x 2	0 x 1
Decimális összeg	128	64	0	16	0	4	2	0

Ábrázolható értékek száma

- $R = B^{K}$, ahol
  - R = ábrázolható értékek száma (range)
  - B = számrendszer alapja (base)
  - K = számjegyek száma
- 1. Példa: 10-es alap, 2 számjegy
  - $R = 10^{2} =$ 100 különböző szám (0…99)
- 2. Példa: 2-es alap, 16 számjegy
  - $R = 2^{16} =$ 65.536 vagy 64K
  - 16 bit-es PC 65.536 különböző értéket képes tárolni

Bitek decimális terjedelme

Bit-ek	Számjegyek	Terjedelem
1	0+	2 (0 és 1)
4	1+	16 (0-15)
8	2+	256
10	3+	1.024 (1K)
16	4+	65.536 (64K)
20	6+	1.048.576 (1M)
32	9+	4.294.967.296 (4G)
64	19+	~ 1,6 x 10^{19} (16 x 10^{18} [exa])
128	38+	~ 2,6 x 10^{38}

Számrendszer alapja (base, radix)

- Alap:
  - A számrendszerben számábrázoláshoz szükséges szimbólumok száma
- A *nagyobb* alap *több* számjegyet igényel
  - 10-es alap: 0,1,2,3,4,5,6,7,8,9
  - 2-es alap: 0,1
  - 8-as alap: 0,1,2,3,4,5,6,7
  - 16-os alap: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F

Szimbólumok száma vs. számjegyek száma

- Egy adott szám ábrázolásához a *nagyobb* alap esetén
  - *több* szimbólum szükséges
  - de *kevesebb* számjegy kell
- 1. Példa:
  - 65_{16} 101_{10} 145_{8} $110\ 0101_{2}$
- 2. Példa:
  - $11C_{16}$ 284_{10} 434_{8} $1\ 0001\ 1100_{2}$

Számolás 2-es alappal

Bináris szám	Egyenérték				Decimális szám
	8-asok (2^3)	4-esek (2^2)	2-esek (2^1)	1-esek (2^0)	
0				0×2^0	0
1				1×2^0	1
10			1×2^1	0×2^0	2
11			1×2^1	1×2^0	3
100		1×2^2			4
101		1×2^2		1×2^0	5
110		1×2^2	1×2^1		6
111		1×2^2	1×2^1	1×2^0	7
1000	1×2^3				8
1001	1×2^3			1×2^0	9
1010	1×2^3		1×2^1		10

10-es alapú összeadó tábla

$3_{10} + 6_{10} = 9_{10}$

+	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	10
2	2	3	4	5	6	7	8	9	10	11
3	3	4	5	6	7	8	9	10	11	12
4	4	5	6	7	8	9	10	11	12	13

stb

8-as alapú összeadó tábla

$3_8 + 6_8 = 11_8$

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

(nincs se 8 se 9 természetesen)

10-es alapú szorzótábla

3_{10} x 6_{10} = 18_{10}

x	0	1	2	3	4	5	6	7	8	9
0				0						
1		1	2	3	4	5	6	7	8	9
2		2	4	6	8	10	12	14	16	18
3		3	6	9	12	15	18	21	24	27
4	0	4	8	12	16	20	24	28	32	36
5		5	10	15	20	25	30	35	40	45
6		6	12	18	24	30	36	42	48	54
7		7	14	21	28	35	42	49	56	63

stb.

8-as alapú szorzótábla

$3_8 \times 6_8 = 22_8$

x	0	1	2	3	4	5	6	7
0				0				
1		1	2	3	4	5	6	7
2		2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4		4	10	14	20	24	30	34
5		5	12	17	24	31	36	43
6		6	14	22	30	36	44	52
7		7	16	25	34	43	52	61

Összeadás

Alap	Feladat	Legnagyobb egyjegyű szám
Decimális	6 +3	9
Oktális	6 +1	7
Hexadecimális	6 +9	F
Bináris	1 +0	1

Összeadás

Alap	Feladat	Átvitel	Eredmény
Decimális	6 +4	10	10
Oktális	6 +2	8	10
Hexadecimális	6 +A	16	10
Bináris	1 +1	2	10

Bináris aritmetika

1	1	1	1	1			
	1	1	0	1	1	0	1
+			1	0	1	1	0
1	0	0	0	0	0	1	1

Bináris aritmetika

- Összeadás
  - Bool XOR-t és AND-et használ
- Szorzás
  - AND
  - Shift (eltolás)
- Osztás

+	0	1
0	0	1
1	1	10

x	0	1
0	0	0
1	0	1

Bináris aritmetika: Bool logika

- Bool logika véges aritmetika nélkül
  - *EXCLUSIVE-OR (kizáró vagy)*
    - "1" az output ha ***nem*** mindegyik bemenet értéke "1"
  - *AND (carry bit)*
    - "1" az output ha mindegyik bemenet értéke "1"

1	1	1	1	1			
	1	1	0	1	1	0	1
+			1	0	1	1	0
1	0	0	0	0	0	1	1

Bináris szorzás

- Bool logika véges aritmetika nélkül
  - *AND (carry bit)*
    - "1" az output ha mindegyik bemenet értéke "1"
  - *Shift (eltolás)*
    - Egy szám eltolása bármelyik alapban ***balra*** egy jeggyel, az egyenlő az alappal való ***szorzással***
    - Egy szám eltolása bármelyik alapban ***jobbra*** egy jeggyel, az egyenlő az alappal való ***osztással***
    - Példák:

- 10_{10} eltolva ***balra*** = 100_{10}
- 10_{2} eltolva ***balra*** = 100_{2}
- 10_{10} eltolva ***jobbra*** = 1_{10}
- 10_{2} eltolva ***jobbra*** = 1_{2}

Bináris szorzás

			1	**1**	**0**	**1**	
		x		**1**	**0**	**1**	
			1	**1**	**0**	**1**	Első hely
					0		Második hely
	1	**1**	**0**	**1**			Harmadik hely (szorzandó bitjei ***eltolva*** eggyel feljebb a harmadik helyen)
1	**0**	**0**	**0**	**0**	**0**	**1**	Eredmény (***AND***)

Bináris szorzás

						1	1	0	1	1	0	1	
					x		1	0	0	1	1	0	
					1	1	0	1	1	0	1		Második helyiértékhez tartozik (szorzandó bitjei ***eltolva*** egy helyiértékkel balra ~ 2-vel megszoroztuk)
				1	1	0	1	1	0	1			3. helyiértékhez tartozik (2^4)
	1	1	0	1	1	0	1						6. helyiérték (2^5)
1	0	0	0	0	0	0	1	0	1	1	1	0	Eredmény (***AND***) Megjegyzés: a 0 a végén, az első helyen nem lett leírva.

megjegyzés: többszörös carry lehetséges.

Bináris szorzás

		1	1	1	1	1								
1	1	1	1	1	1	1	1							
	1	0	1	1	1	1	1	0	0	1	x	1	1	1
		1	0	1	1	1	1	1	0	0	1			
			1	0	1	1	1	1	1	0	0	1		
1	0	1	0	0	1	1	0	0	1	1	1	1		

Konvertálás 10-es alapból

- Hatvány tábla

Kitevő / Alap	8	7	6	5	4	3	2	1	0
2	256	128	64	32	16	8	4	2	1
8				32.768	4.096	512	64	8	1
16					65.536	4.096	256	16	1

10-es alapból 2-esbe

$42_{10} = 101010_2$

Kitevő / Alap	6	5	4	3	2	1	0
2	64	32	16	8	4	2	1
		1	0	1	0	1	0
Egész		42/32 = 1	10/16 = 0	10/8 = 1	2/4 = 0	2/2 = 1	0/1 = 0
Maradék		10	10	2	2	0	0

10-es alapból 2-esbe

10-es alap 42

Maradék

Hányados

2)	42	(**0**	Legkisebb helyi-értékű bit
2)	21	(**1**	
2)	10	(**0**	
2)	5	(**1**	
2)	2	(**0**	
2)	1	(**1**	Legnagyobb helyi-értékű bit

2-es alap **101010**

10-es alapból 16-osba

$5.735_{10} = 1667_{16}$

Kitevő / Alap	4	3	2	1	0
16	65.536	4.096	256	16	1
		1	**6**	**6**	**7**
Egész		5.735 /4.096 = 1	1.639 / 256 = 6	103 /16 = 6	7
Maradék		5.735 – 4.096 = 1.639	1.639 – 1.536 = 103	103 – 96 = 7	

10-es alapból 16-osba

10-es alap 5.735

Maradék

Hányados

16) 5.735 (7 Legkisebb helyi-értékű bit

16) 358 (6

16) 22 (6

16) 1 (1 Legnagyobb helyi-értékű bit

16) 0

16-os alap 1667

10-es alapból 16-osba

10-es alap 8.039

Maradék

Hányados

16)	8.039	(7	Legkisebb helyi-értékű bit
16)	502	(6	
16)	31	(15	
16)	1	(1	Legnagyobb helyi-értékű bit
16)	0		

16-os alap 1F67

8-as alapból 10-esbe

$7263_8 \quad = \quad 3.763_{10}$

Kitevő	8^3	8^2	8^1	8^0
	512	64	8	1
	x 7	x 2	x 6	x 3
Decimális összeg	3.584	128	48	3

8-as alapból 10-esbe

$7263_8 = 3.763_{10}$

7

x 8

56 + **2** = 58

x 8

464 + **6** = 470

x 8

3760 + **3** = 3.763

16-os alapból 2-esbe

- A nibble (négy bites szó) megközelítés
  - Hexadecimálist könnyebb írni és olvasni, mint binárist

16-os alap	1	F	6	7
2-es alap	0001	1111	0110	0111

  - Miért hexadecimális?
    - Operációs rendszer, ill. hálózati hibák esetén a rendszer a hibaeseményhez tartózó adatokat hexadecimális formában jelenít meg. (~ ismeretlen memóriatartalom)

Törtek

- Tizedespont vagy alappont
  - Tizedespont a 10-es alapban
  - Binárispont a 2-es alapban
- Nincs tökéletes alakja a törteknek különböző számrendszerekben
  - Pontos átalakítás nem mindig lehetséges

Tizedestörtek

- Rakjuk eggyel jobbra a tizedespontot
  - Jelenség: a szám megszorzódik a számrendszer alapjával
  - Példa: 139.0_{10} → 1390_{10}
- Rakjuk eggyel balra a tizedespontot
  - Jelenség: a szám osztódik a számrendszer alapjával
  - Példa: 139.0_{10} → 13.9_{10}

Törtek: 10-es és 2-es alap

$0{,}2589_{10}$

Helyi érték	10^{-1}	10^{-2}	10^{-3}	10^{-4}
Érték	1/10	1/100	1/1000	1/10000
Számítás	2 x 1/10	5 x 1/100	8 x 1/1000	9 x1/1000
Összeg	0,2	0,05	0,008	0,0009

$0{,}101011_2 = 0{,}671875_{10}$

Helyi érték	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
Érték	1/2	1/4	1/8	1/16	1/32	1/64
Számítás	1 x 1/2	0 x 1/4	1x 1/8	0 x 1/16	1 x 1/32	1 x 1/64
Összeg	0,5		0,125		0,03125	0,015625

Törtek: 10-es és 2-es alap

- Nincs általános összefüggés az $1/10^k$ és $1/2^k$ típusú törtek között
  - Ennek következtében néhány 10-es alapban ábrázolható szám nem ábrázolható 2-es alapban
  - De minden $1/2^k$ formájú törtet tudunk ábrázolni 10-es alapban
- Tört átalakítása két alap között akkor ér véget
  - Ha van racionális megoldás vagy
  - Ha a kívánt pontosságot elértük

Kevert számalakítás

- Egész és tört részeket külön kell konvertálni
- Alappont (tizedes pont): rögzített viszonyítás az átalakításhoz
  - A bal oldalon lévő számjegy egységnyi minden alapban
  - B^0 mindig 1, alaptól függetlenül