Hardver alapok

(VIIIBA01)

Bevezetés

Pilászy György gpilaszy@iit.bme.hu

A számítógép fejlődése

Cél: matematikai számítások elvégzése

Blaise Pascal, 1642 „Pascaline”

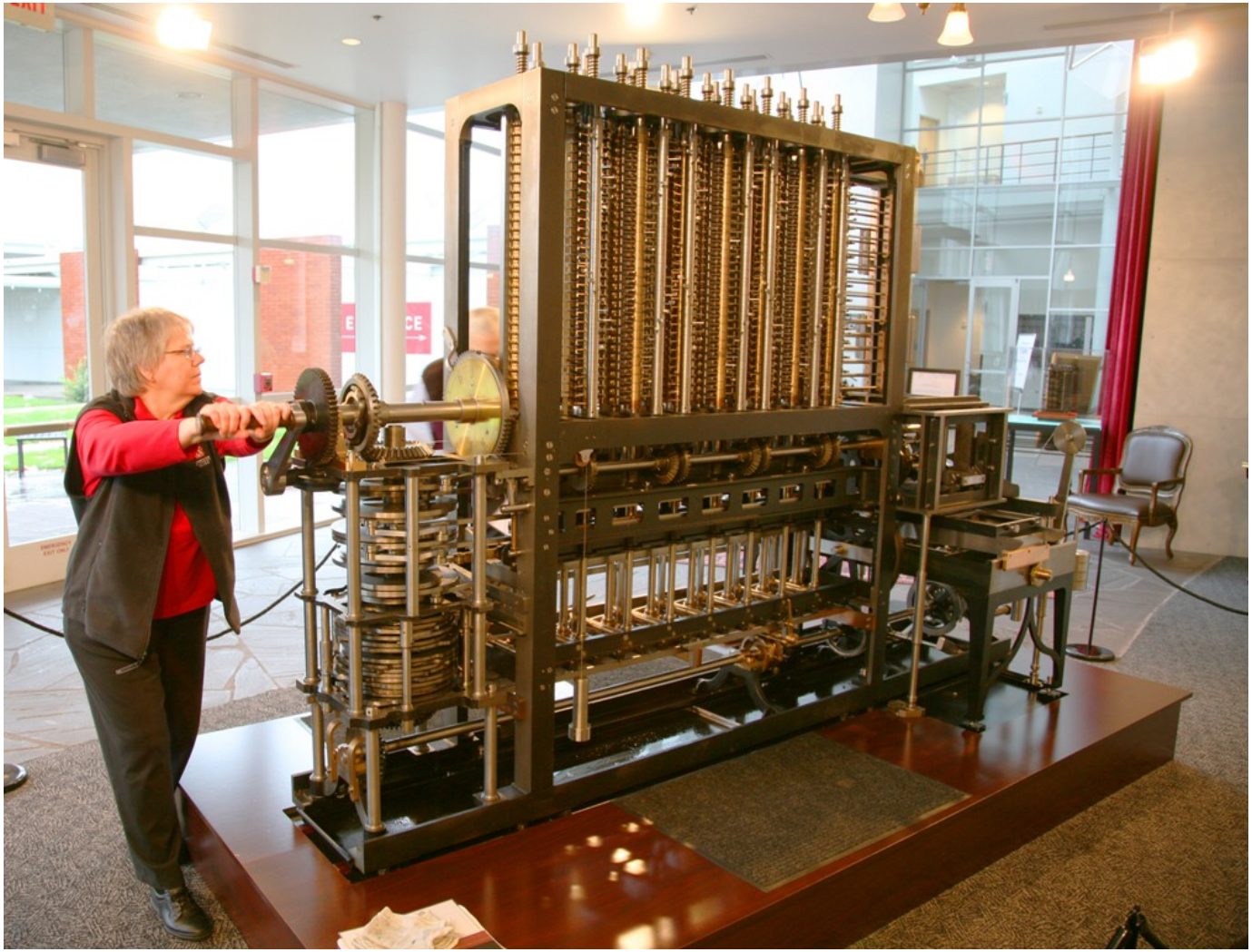
Összeadás, kivonás, (szorzás, osztás)

© Mathematisch-Physikalischer Salon, Dresden

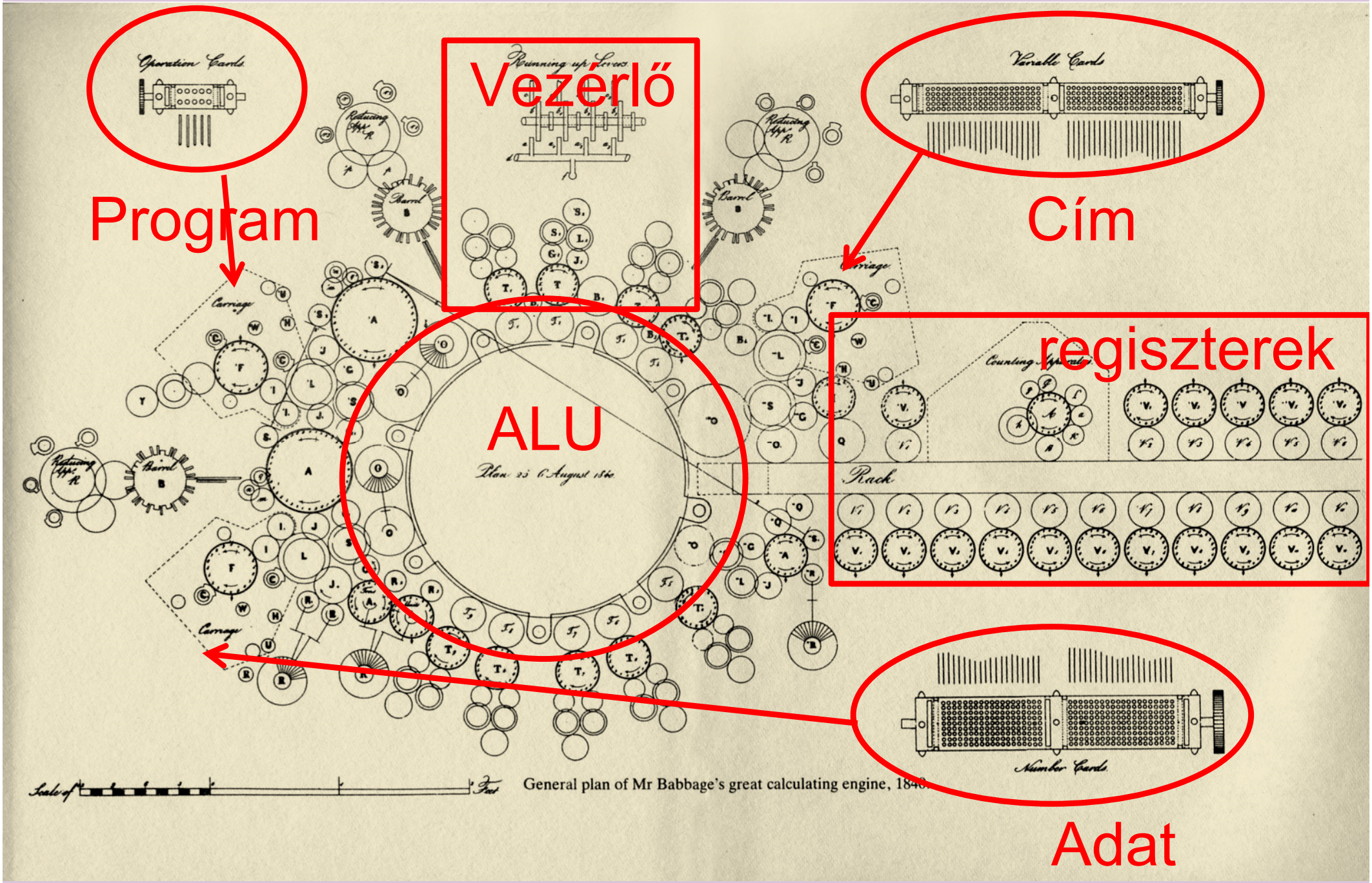
A számítógép fejlődése

Charles Babbage 1822 „Difference engine”

Függvények hatványsoros közelítése



Analytical engine 1840



A számítógép fejlődése

- További mechanikus, elektromechanikus és elektronikus megoldások, de a cél nem változott

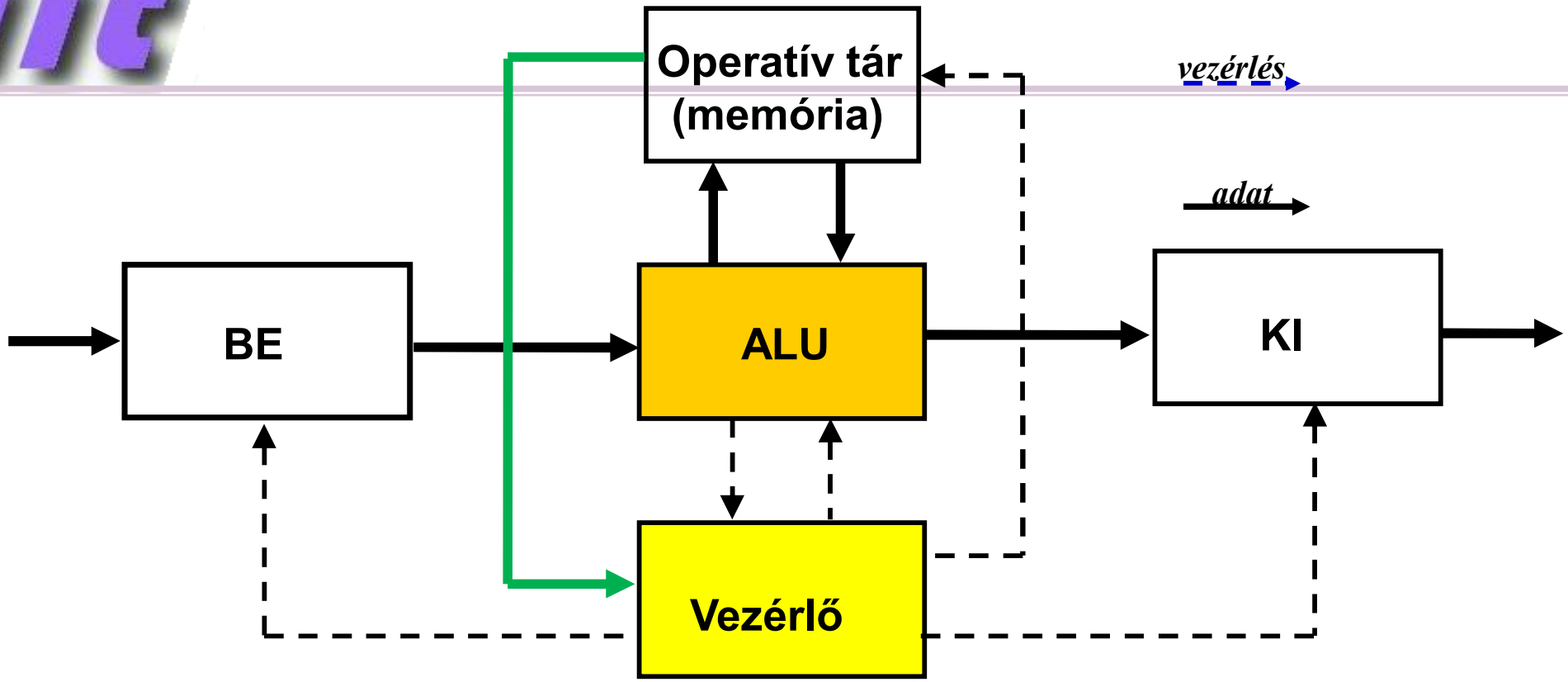
A számítógép fejlődése

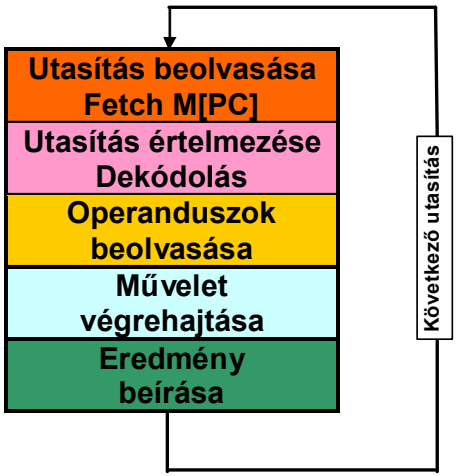
A számítások elvégzéséhez szükség van:

- Változókra → memória, regiszter
- Műveletvégzőre → ALU (Aritmetic Logic Unit)
- Megfelelő sorrendű végrehajtásra→program+vezérlő egység
- Felhasználói felület → bevitel, kijelzés, nyomtatás

- ALU+vezérlő egység→CPU (Central Processing Unit)

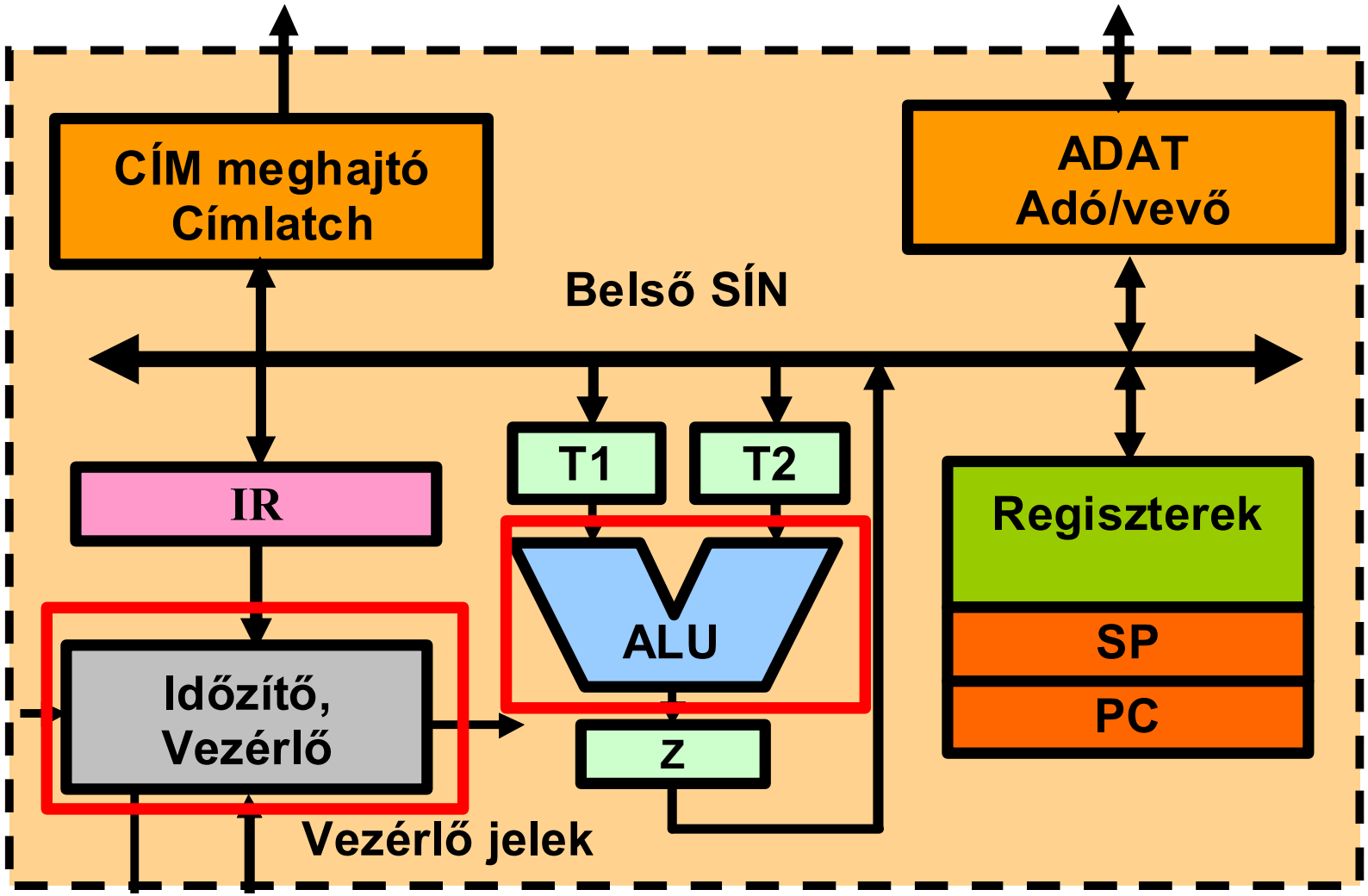
Neumann modell





-Belső programtárolás
-Utasítás és adat azonos közegben
-Értelmezés az algoritmus / PC szerint
-BINÁRIS számábrázolás
-Adattípusok műveletekhez rendeltek
-Szekvenciális utasítás végrehajtás
-Egy dimenziós lineáris címzésű memória
-Utasítások programból módosíthatók

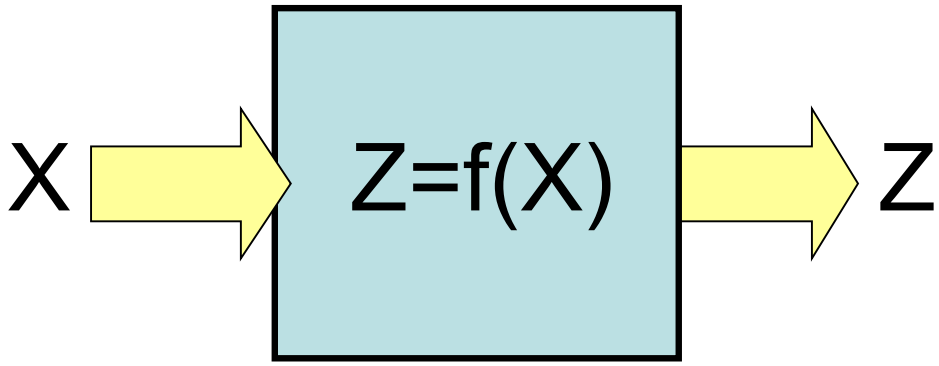
CPU elvi felépítése



- Bináris számábrázolás
- Kombinációs hálózatok
- Sorrendi hálózatok

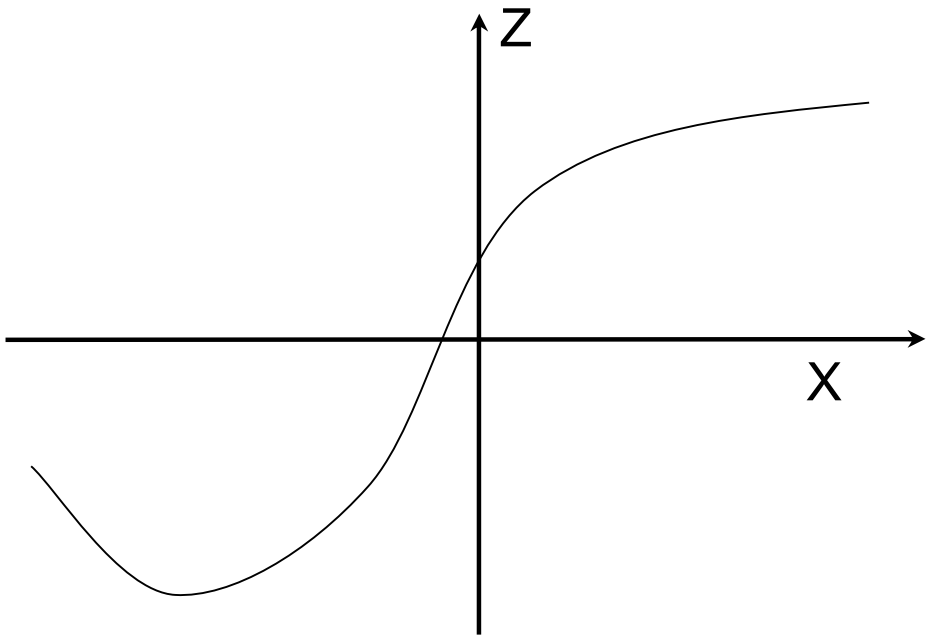
A továbbiakban ezekkel a témakörökkel foglalkozunk részletesen

Analóg←→Digitális rendszerek

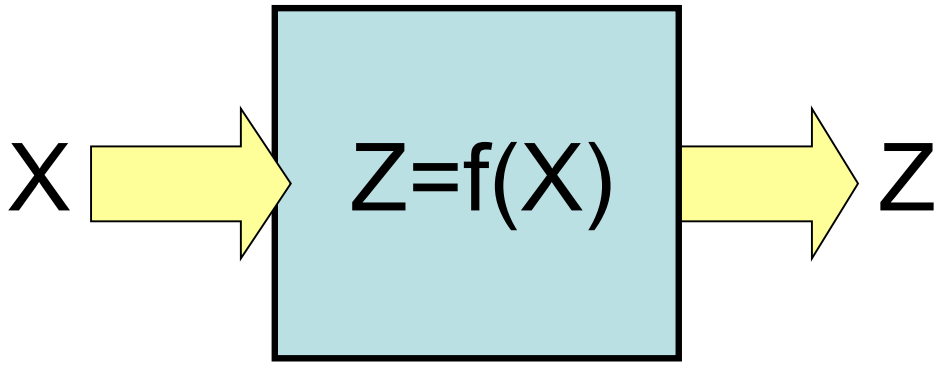


Analóg rendszer

- X folytonos
- Z folytonos
- f(X) folytonos fv kapcsolat

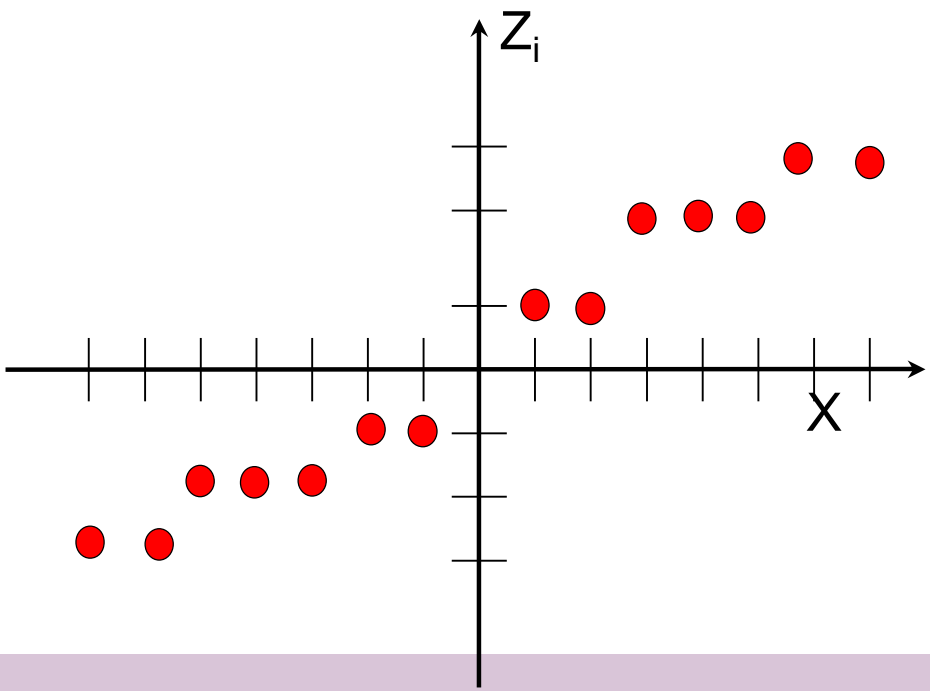


Analóg←→Digitális rendszerek



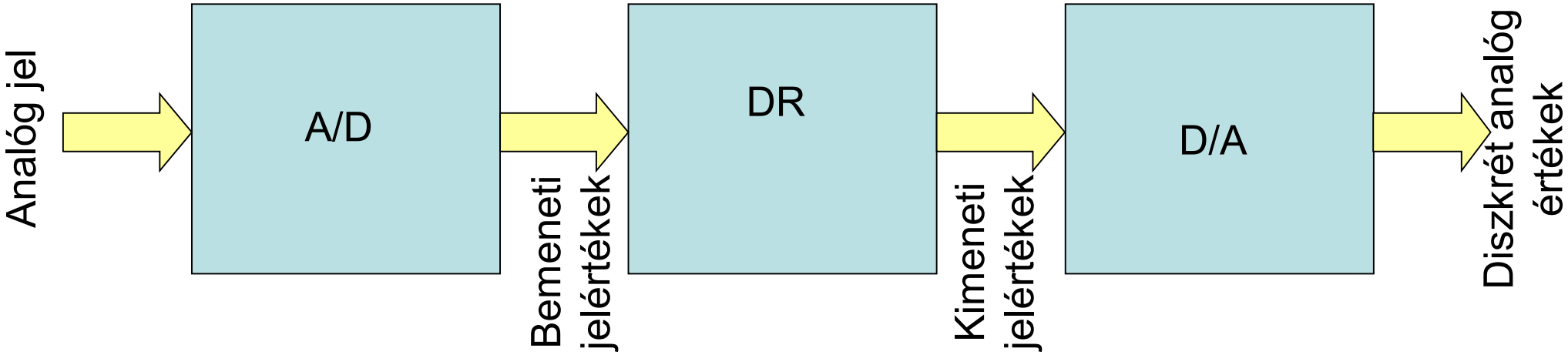
Digitális rendszer

- X diszkrét
- Z diszkrét
- f(X) diszkrét



Analóg←→Digitális rendszerek

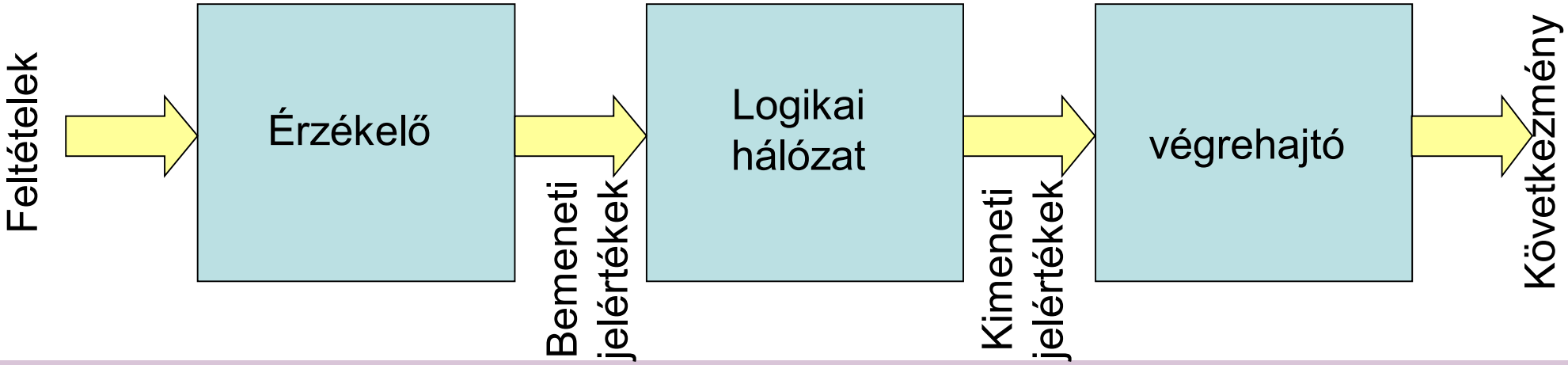
- Átjárás a két rendszer között →A/D és D/A
- f(X) megvalósítása egyszerűbb (olcsóbb) lehet



Analóg←→Digitális rendszerek

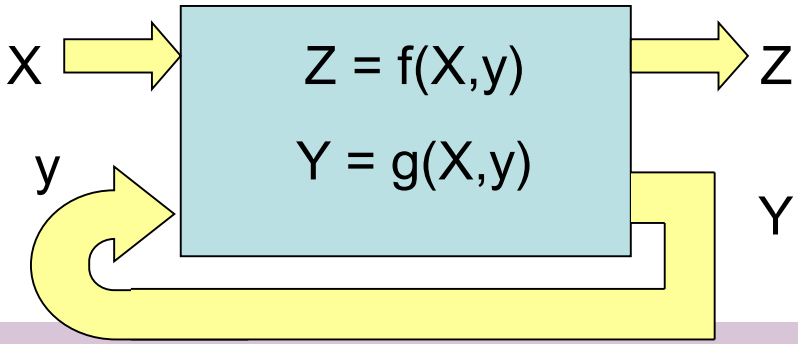
- Digitális rendszer: végesszámú diszkrét értékhez véges számú diszkrét értéket rendel adott feltételek alapján

  pl.: analóg jel→ komparátor→diszkrét jel
- Logikai rendszer, Logikai hálózat
- Feltételek és következmények összerendelése



Kombinációs ↔ sorrendi hálózatok

- Ugyanazon bemeneti kombinációhoz eltérő kimeneti érték is tartozhat a hálózat „előélete” alapján
- A hálózat „emlékezik”
- Szekunder bemeneti változók (y)
- Z = f(X,y), Y = g(X,y)
- Visszacsatolás Y→y



Számok ábrázolása

Számok ábrázolása

10-es számrendszer

$$2019 = 2 \cdot 10^3 + 0 \cdot 10^2 + 1 \cdot 10^1 + 9 \cdot 10^0$$

i-edik számjegy (h_i)

Számrendszer alapja (radix) (r) $\quad r = 10$

Egész szám esetén:

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

$$\mathrm{h}_3 = 2 \qquad r^3 = 10^3 = 1000$$

$$\mathrm{h}_2 = 0 \qquad r^2 = 10^2 = 100$$

$$\mathrm{h}_1 = 1 \qquad r^1 = 10^1 = 10$$

$$\mathrm{h}_0 = 9 \qquad r^0 = 10^0 = 1$$

10-es számrendszerben: $h_i \in \{0,1,2,3,4,5,6,7,8,9\}$

Átváltás másik számrendszerre

$$N = \sum_{i=0}^{n} h_i \cdot r^i = h_n \cdot r^n + h_{n-1} \cdot r^{n-1} + \cdots + h_1 \cdot r^1 + h_0$$

Osszuk r-el

$M = \frac{N}{r} = h_n \cdot r^{n-1} + h_{n-1} \cdot r^{n-2} + \cdots + h_2 \cdot r^1 + h_1$	Maradék: h_0	Osszuk r-el
$L = \frac{M}{r} = h_n \cdot r^{n-2} + h_{n-1} \cdot r^{n-3} + \cdots + h_3 \cdot r^1 + h_2$	Maradék: h_1	Osszuk r-el
$\vdots$		
$C = \frac{D}{r} = h_n \cdot r^1 + h_{n-1}$	Maradék: h_{n-2}	Osszuk r-el
$B = \frac{C}{r} = h_n$	Maradék: h_{n-1}	Osszuk r-el
$A = \frac{B}{r} = 0$	Maradék: h_n	

Megkaptuk a számjegyeket

2-es számrendszer

2022	:2	0
1011	:2	1
505	:2	1
252	:2	0
126	:2	0
63	:2	1
31	:2	1
15	:2	1
7	:2	1
3	:2	1
1	:2	1
0		

$$r = 2 \qquad h_i \in \{0,1\}$$

Átalakítás 10→2

kettővel vett osztási maradékok meghatározása

Eredmény kiolvasása

$$(2022)_{10} = (11111100110)_2$$

2-es számrendszer

Átalakítás 2→10

2 hatványoknak megfelelő összeadás

$$(11111100110)_2 = ?$$

$$N = 1\cdot 2^{10} + 1\cdot 2^{9} + 1\cdot 2^{8} + 1\cdot 2^{7} + 1\cdot 2^{6} + 1\cdot 2^{5} + 0\cdot 2^{4} + 0\cdot 2^{3} + 1\cdot 2^{2} + 1\cdot 2^{1} + 0\cdot 2^{0}$$

$$N = 1\cdot 1024 + 1\cdot 512 + 1\cdot 256 + 1\cdot 128 + 1\cdot 64 + 1\cdot 32 + 0\cdot 16$$

$$+ 0\cdot 8 + 1\cdot 4 + 1\cdot 2 + 0\cdot 1 = \underline{\underline{2022}}$$

$$(11111100110)_2 = (2022)_{10}$$

Horner szabály

A hatványozás helyett kiemelések…

$$N = \sum_{i=0}^{n} h_i \cdot r^i = h_n \cdot r^n + h_{n-1} \cdot r^{n-1} + \cdots + h_1 \cdot r^1 + h_0 =$$

$$\underbrace{(\ldots(}_{\text{n-1db}}h_n) \cdot r + h_{n-1}) \cdot r + h_{n-2}) \cdot r + \ldots) \cdot r + h_0$$

Az előző példánál maradva: $(11111100110)_2$ = ?

$$N = ((((((((((1) \cdot 2 + 1) \cdot 2 + 1) \cdot 2 + 1) \cdot 2 + 1) \cdot 2 + 0) \cdot 2 + 0) \cdot 2 + 1) \cdot 2 + 1) \cdot 2 + 0$$
$$= 2022$$

Előny: jóval kevesebb a művelet igény:

9db összeadás, 9db szorzás

16-os számrendszer

$$r = 16 \qquad h_i \in \{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$$

A hiányzó „számjegyek” az ABC betűi

2022 :16	6 →6
126 :16	14→E
7 :16	7 →7
0	

$$(2022)_{10} = (7E6)_{16}$$

2→16 közvetlen felírás

(4 bites csoportokra bontjuk, mert $16 = 2^4$)

$$(2022)_{10} = (\underbrace{0111}_{7}\ \underbrace{1110}_{E}\ \underbrace{0110}_{6})_2$$

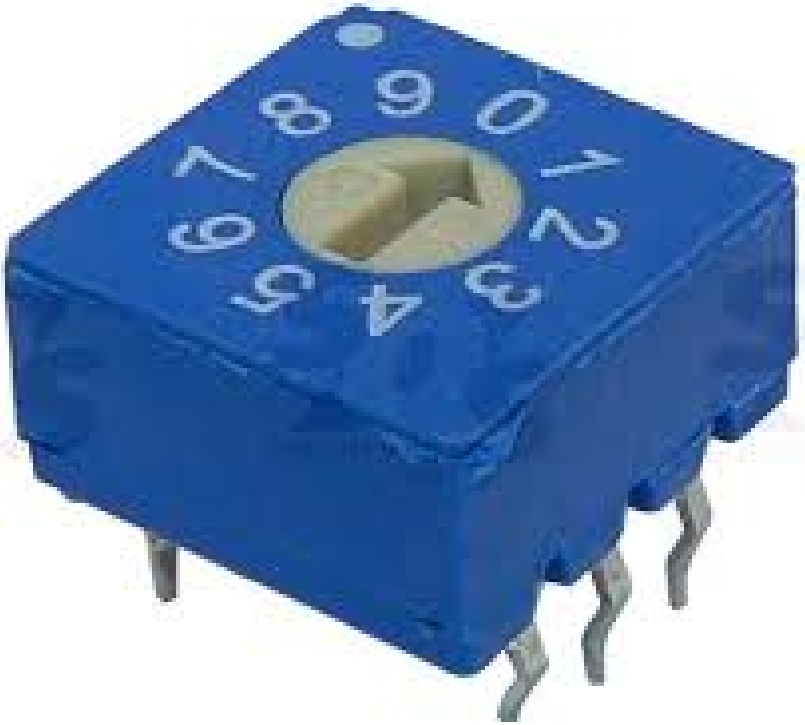
$$(7 \quad E \quad 6)_{16}$$

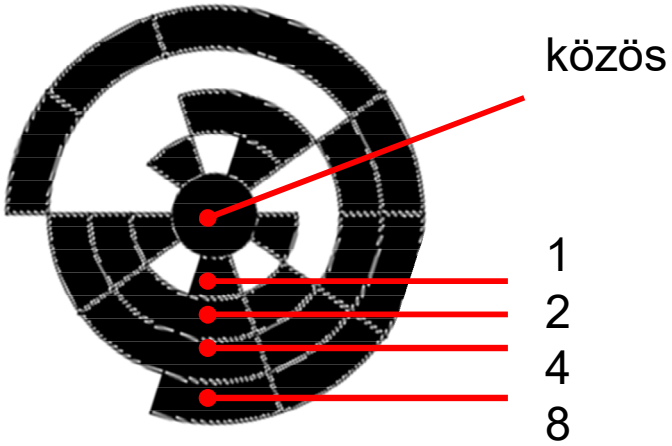
BCD számábrázolás

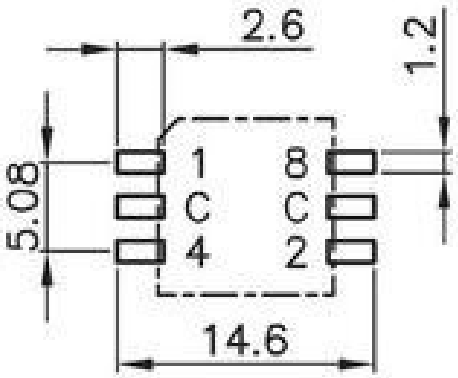
- BCD:Binary-Coded Decimal
- Tízes számrendszer számjegyeit 4 bites bináris számokként ábrázoljuk

$$(2022)_{10} = (\underbrace{0010}_{2}\ \underbrace{0000}_{0}\ \underbrace{0010}_{2}\ \underbrace{0010}_{2})_{BCD}$$

Kitekintés - BCD kódkapcsoló

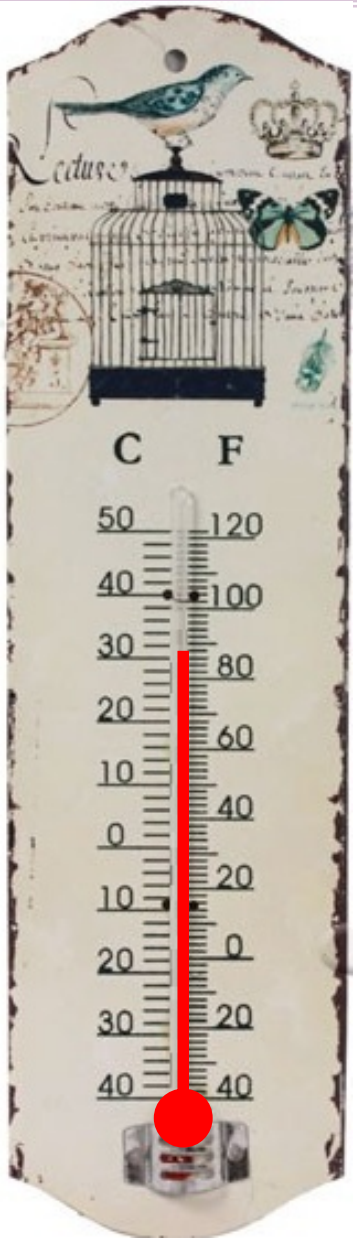




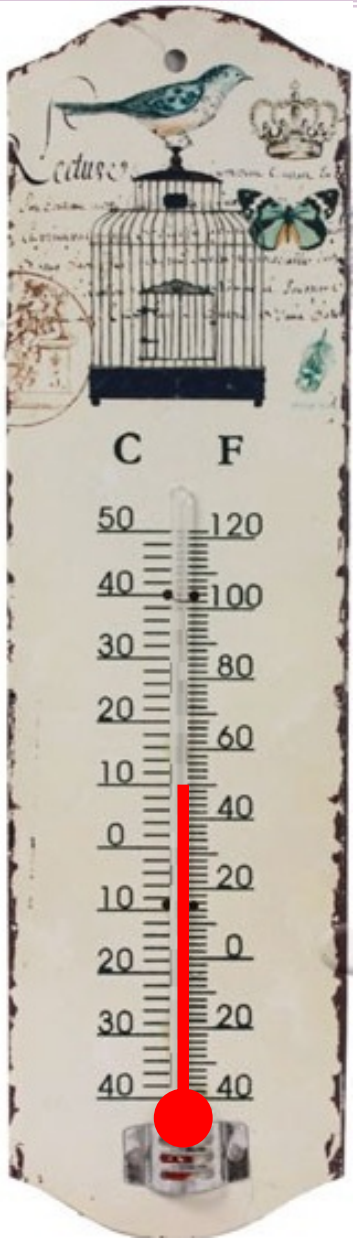


Code	POSITION									
	0	1	2	3	4	5	6	7	8	9
1		●		●		●		●		●
2			●	●			●	●		
4					●	●	●	●		
8									●	●

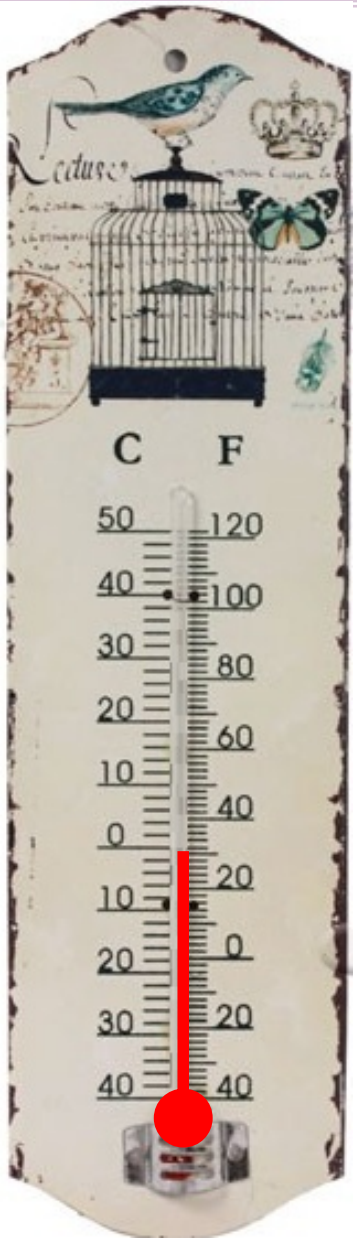
Előjel kezelés



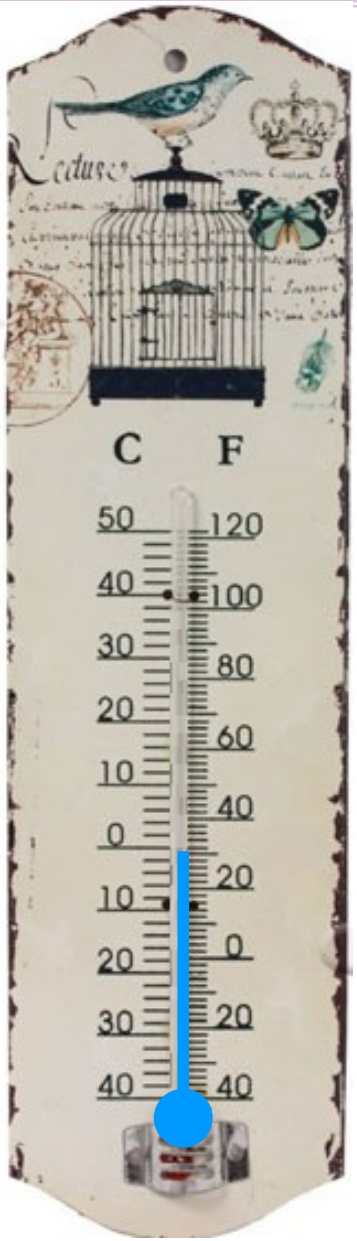
Előjel kezelés



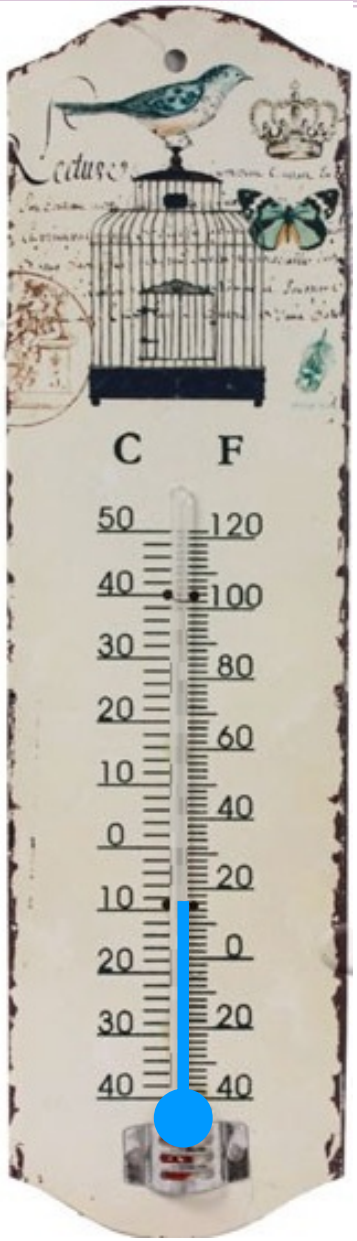
Előjel kezelés



Előjel kezelés



Előjel kezelés



Előjel kezelés

- Előjel + abszolút értékes ábrázolás

0	0	0	0	0	1	0	0	1	→ +9
1	0	0	0	0	1	0	0	1	→ -9

- Hátrány:
  - dupla nulla probléma
  - bonyolult kezelés
    - Összehasonlítás +1 < +10 vagy -10 < -1 !
    - Aritmetika 2+ (-5) = -(5-2)

Komplemens ábrázolás

A kilométer óra 0-án áll, mi van, ha vissza felé megyünk?

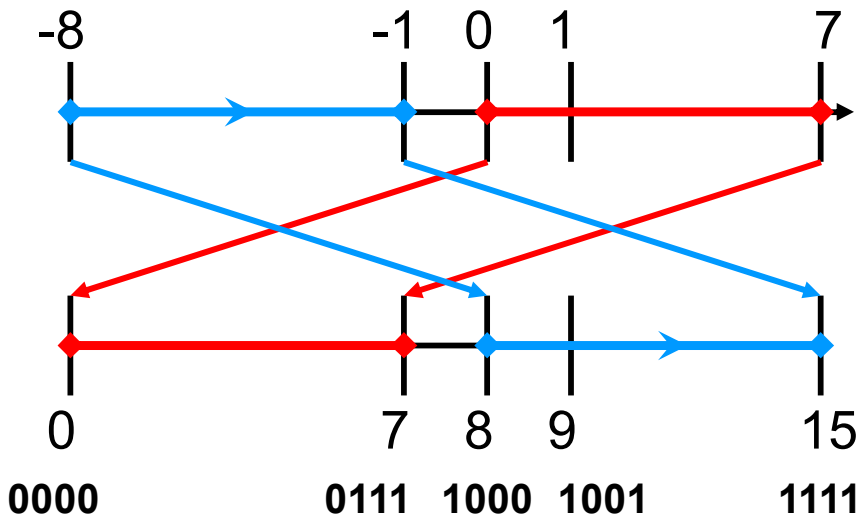
Alulcsordulás… ”-1” → ”999999”

Komplemens ábrázolás

- Rögzített a számjegyek száma
- Ugyanannyi negatív és nem negatív szám
- Pozitív számok alakja nem változik
- Negatív számok abszolút érték helyett komplemensben
  (1 000 000-ból kivonjuk az abszolút értéket)
- -1→999 999
- Az ábrázolható tartomány:
  -500 000…-1,0,1,…499 999 (1 000 000db érték)

Kettes komplemens ábrázolás

- r = 2
- A helyértékek száma legyen pl. négy
- Az ábrázolható értékek száma: $2^4 = 16$ db
- Nem negatív számok: 0…7 (8 db)
- Negatív számok: -8…-1 (8 db)
- Leképzés:



A legfelső bit előjelként viselkedik!

Kettes komplemens tulajdonságai

- Legyen m pozitív szám r bites
  - kettes komplemense: m’’
  - bitenkénti invertáltja (1-es komplemense): m’

    $m + m'' = M = 2^r$

    $m + m' = M-1$

    Átrendezve: $m + m' = m + m'' - 1$

    $m'' = m' + 1$ vagyis:
- Képzése: bitenként invertáljuk + 1 (a legkisebb helyiértéken)
- -1 + 1 = 0
- Az összeadás és kivonás szabályai mod_M érvényesek
- Nincs „dupla nulla”
- Legfelső bit megadja a szám előjelét

Gyakorló feladatok

- Legalább hány bites ábrázolást kell definiálnunk, ha előjel nélküli számokat szeretnénk ábrázolni 0..50 tartományban?

- Egészrész: $\log_2(50) = 5.64 \rightarrow 6$ bit

Számítsuk ki 8 bites kettes komplemensben a 64+65 művelet eredményét.

$$
\begin{array}{r}
64 = 0100\ 0000 \\
+65 = 0100\ 0001 \\
\hline
1000\ 0001
\end{array}
$$

→ -127

→ **Túlcsordulás!**

Adjuk meg előjeles decimális formában az **1001111** 7 bites kettes komplemensben ábrázolt számot.

- MSB = 1 → negatív szám
- Abszolút érték: 0110001 →49
- Előjeles decimális alak: **-49**