



# Adatok ábrázolása, adattípusok

---

**Összefoglalás**



# Adatok ábrázolása, adattípusok

---

- Számítógépes rendszerek
  - működés: információfeldolgozás
  - IPO: input-process-output modell
  - információ tárolása adatok formájában
  - adatformátum választása:
    - gépi feldolgozás, tárolás egyszerű legyen – Neumann János → bináris
    - adatformátum választása: az adatokon műveleteket egyszerűen SZISZTEMATIKUSAN



# Adatokon végzett műveletek (IPO modell)

---

- Információ számítógép által feldolgozható formába rögzíteni:
  - számítógép perifériák (bemeneti egységek)
- Információ feldolgozása
  - CPU, memória
- Eredmények megjelenítése:
  - számítógép perifériák (kimeneti egységek)



# 6. Fejezet: Little Man Computer (A kisember számítógép)

---

**A számítógép hardver és szoftver  
architektúrája:**

**Egy informatikai megközelítés**

**3. kiadás, Irv Englander**

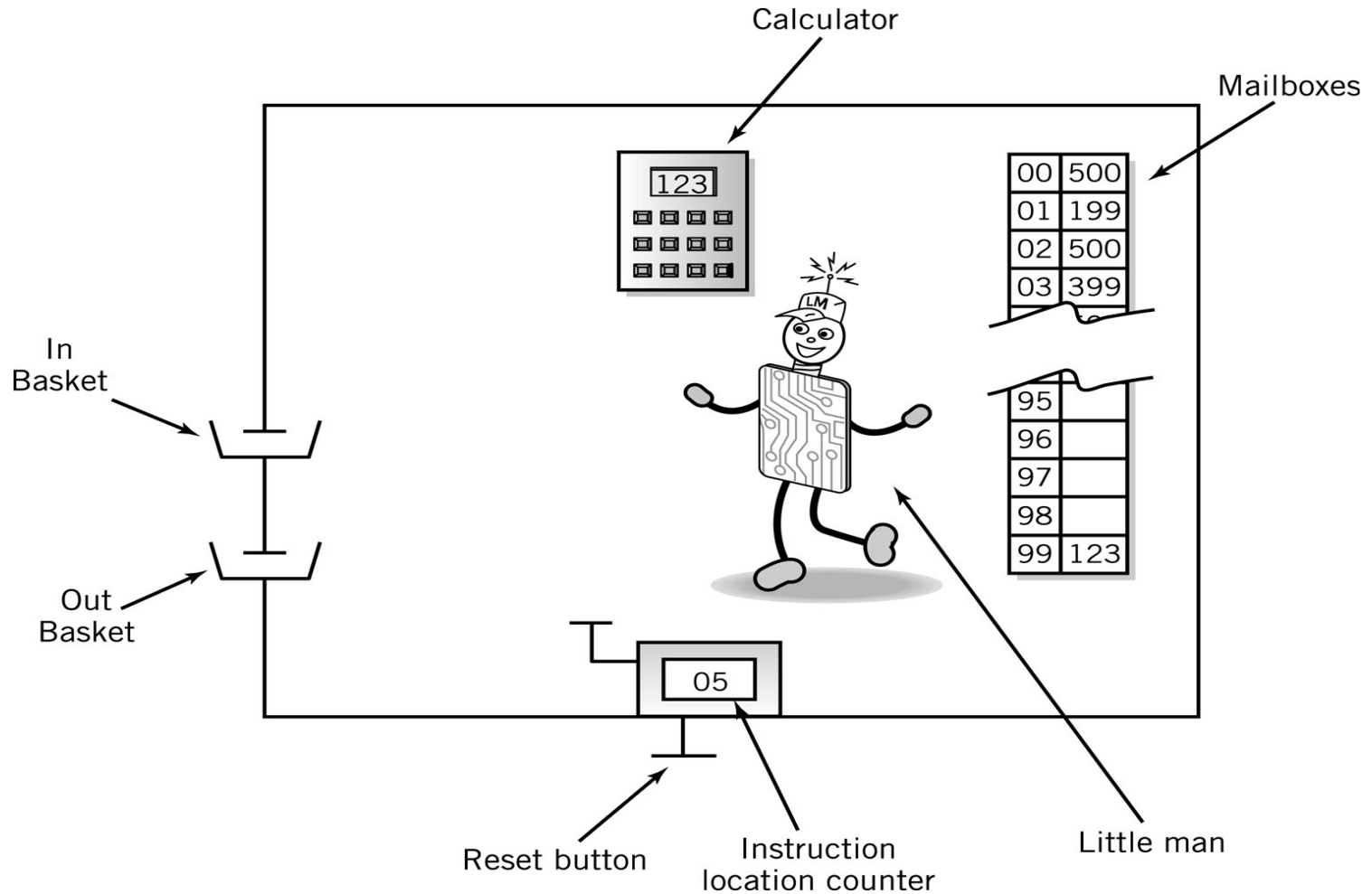
**John Wiley and Sons ©2003**

**Wilson Wong, Bentley College**

**Linda Senne, Bentley College**



# Little Man Computer (LMC)





# Postaládák: Cím vs. Tartalom

---

- A postaláda címek egymást követik
- A postaláda tartalma lehet
  - adat vagy
  - utasítás

Cím	Tartalom



# Tartalom: Utasítások

- Műveleti kód
  - művelet kódja
  - tetszőleges rövidítés (mnemonic)
- Művelet operandusa
  - módosítandó „objektum”
    - adat értéke vagy
    - adat címe, ahol az érték tárolódik

Cím	Tartalom	
	Műveleti kód	Operandus



# Varázslat!

---

- Program betöltése a memóriába
- Adatok belehelyezése a bemeneti kosárba („In Basket”)





# Assembly nyelv

- CPU specifikus
- egy az egyhez összerendelés az assembly nyelv utasításai és a bináris (CPU által végrehajtható) utasítások között
- *Mnemonic-ok* (rövid karakter sorozatok) utasításokat reprezentálnak
- Akkor használjuk, ha a programozónak a hardver pontos vezérlésére van szüksége
  - pl.: eszközmeghajtók (driver-ek) készítése



# Utasításkészlet

Aritmetikai	1xx	Összeadás
	2xx	Kivonás
Adat mozgató	3xx	Tárolás
	5xx	Betöltés
Input/Output	901	INPUT
(Be-/Kimenet)	902	OUTPUT
Gép irányítása (kávészünet)	000	Leállítás COB



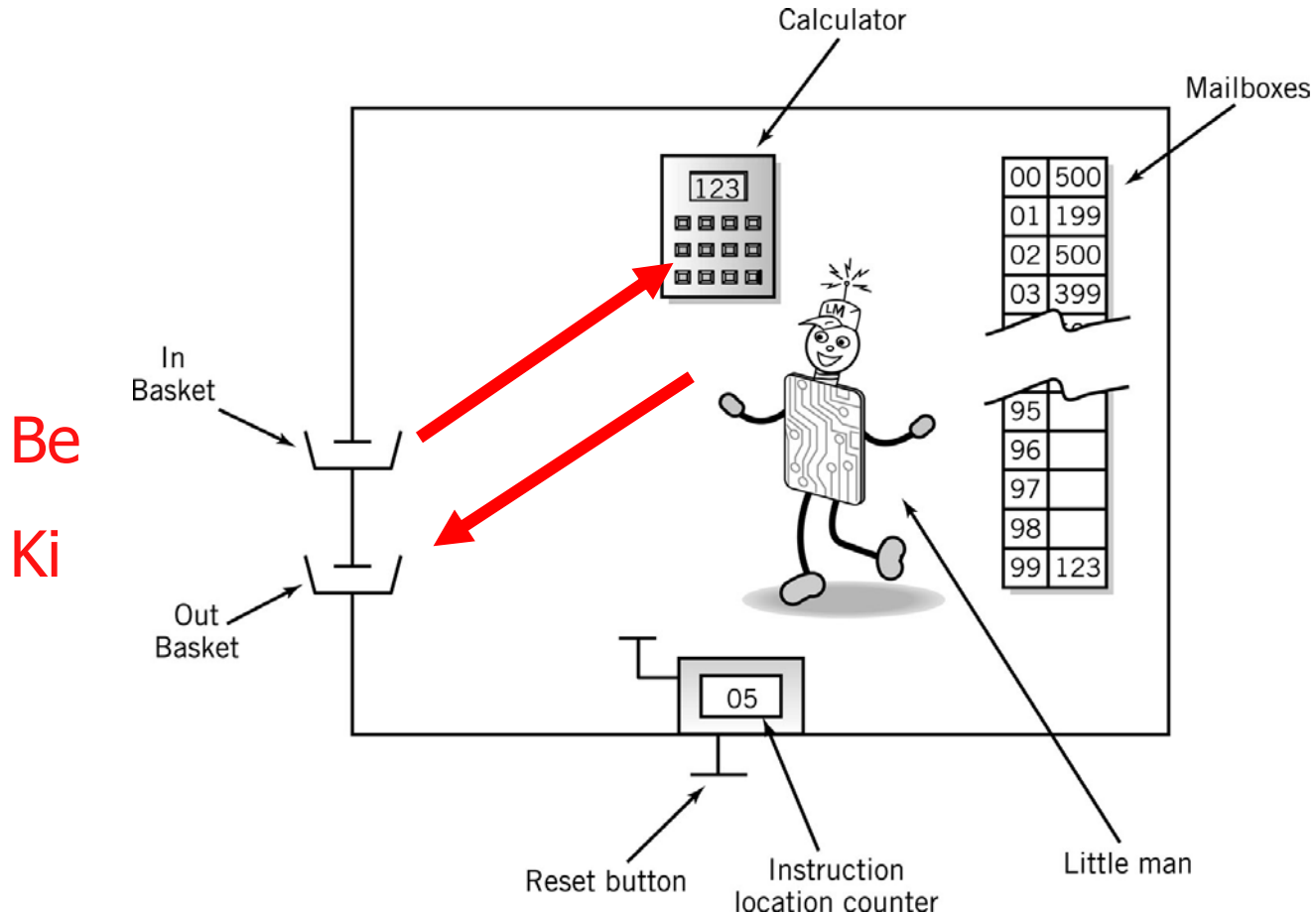
# Input/Output

- Adatok mozgatása a számítógép és a be/ki kosarak között

	Tartalom	
	Műveleti kód	Operandus
IN (input)	9	01
OUT (output)	9	02



# LMC Input/Output





# Belső adatmozgás (LMC-n belül)

- A postaláda és a számítógép között

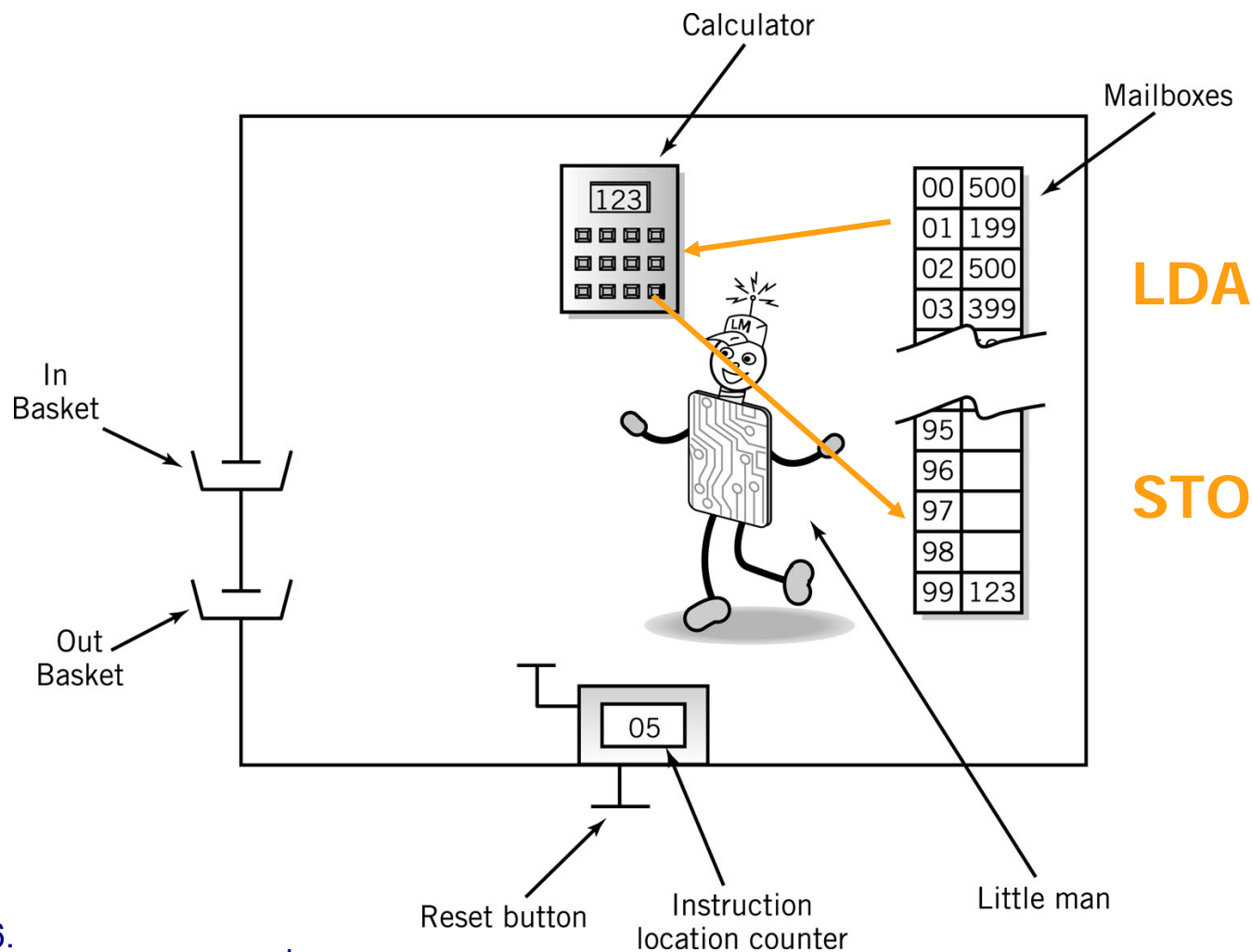
STO  
(tárolás)

LDA  
(betöltés)

Tartalom	
Műveleti kód	Operandus
3	XX
5	XX



# LMC belső adatmozgatás





# Adatok tárolása

---

- Utasításokat és az adatokat tároló postaláda fiókok fizikailag azonosak
- Adatok (általában) nincsenek beleágyazva az utasítások közé
- *DAT* mnemonic azonosítja őket az assembly kódban



# Aritmetikai utasítások

- Postaládában tárolt érték beolvasása
- Művelet elvégzése a számológép segítségével

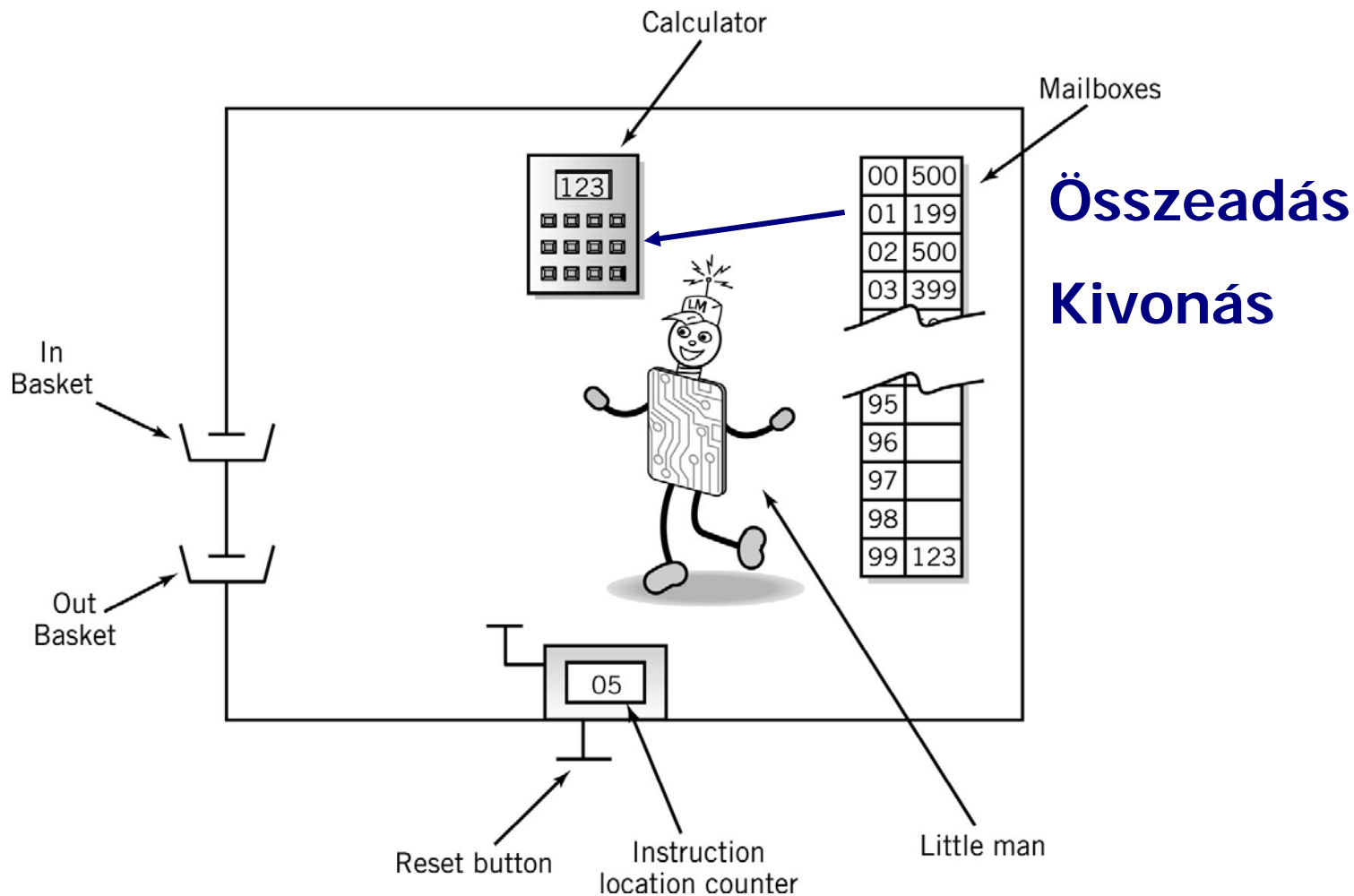
Összeadás  
Kivonás

Tartalom	
Műveleti kód	Operandus
1	xx
2	xx





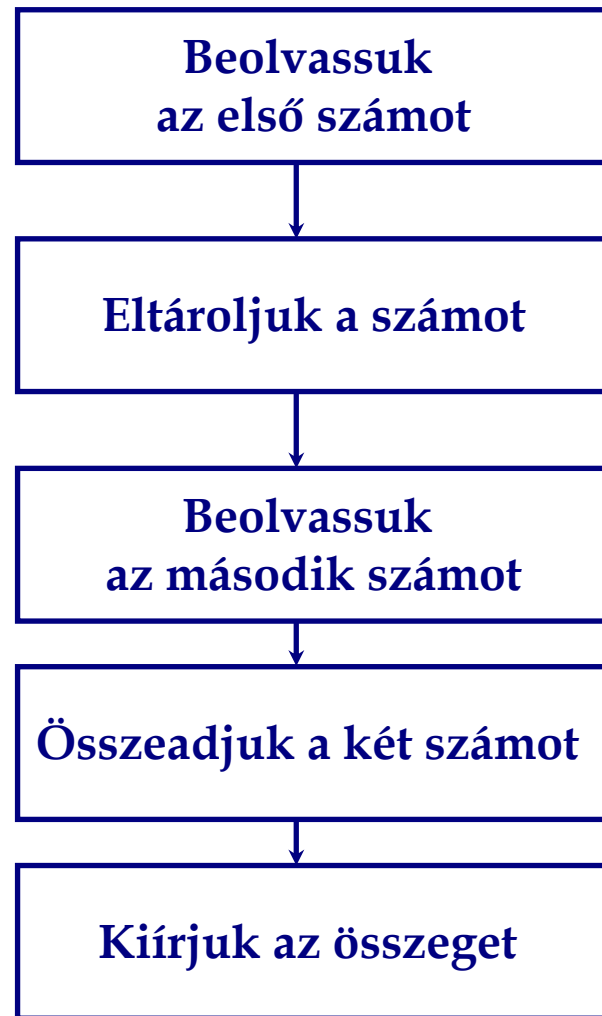
# LMC aritmetikai utasítások





# Egyszerű program: 2 szám összeadása

- Feltételezzük, hogy az adatokat 90< címekkel címzett postaládákban tároljuk
- Írjuk meg a programot!





# Program 2 szám összeadására: Mnemonic-okat használva

Postaláda	Mnemonic	Utasítás leírása
00	IN	;1. Input szám
01	STO 99	;Adat tárolása
02	IN	;2. Input szám
03	ADD 99	;Két szám összeadása
04	OUT	;Eredmény kiírása
05	COB	;Leállítás
99	DAT 00	;Adat



# Program 2 szám összeadására

Postaláda	Kód	Utasítás leírása
00	901	;1. Input szám
01	399	;Adat tárolása
02	901	;2. Input szám
03	199	;Két szám összeadása
04	902	;Eredmény kiírása
05	000	;Leállítás
99	000	;Adat



# Program vezérlés

- Elágazás utasítások (a szekvenciális utasítás-végrehajtás eltérítése)
  - Cím megváltoztatása az utasítás-számlálóban

- Leállítás

BR (Ugrás)

BRZ (Ágazás 0-ál)

BRP (Ágazás +-nál)

COB (Leállítás)

Tartalom	
Műveleti kód	Operandus
6	xx
7	xx
8	xx
0	(ignore)



# Utasításkészlet

Aritmetikai	1xx	Összeadás
	2xx	Kivonás
Adatmozgatás	3xx	Tárolás
	5xx	Betöltés
BR	6xx	Ugrás
BRZ	7xx	Ágazás 0-nál
BRP	8xx	Ágazás +-nál
Input/Output	901	INPUT
	902	OUTPUT
Gép vezérlés (kávészünet)	000	Leállítás COB



# Számoljuk ki két szám különbségének abszolútértékét (kimenet pozitív!)

00	IN	901	
01	STO 10	310	
02	IN	901	
03	STO 11	311	
04	SUB 10	210	
05	BRP 08	808	;teszt
06	LDA 10	510	;ha negatív, fordított sorrendben kell!
07	SUB 11	211	
08	OUT	902	;eredmény kiírása
09	COB	000	;leállítás
10	DAT 00	000	;adattároló
11	DAT 00	000	;adattároló



# Utasítás-végrehajtási ciklus

---

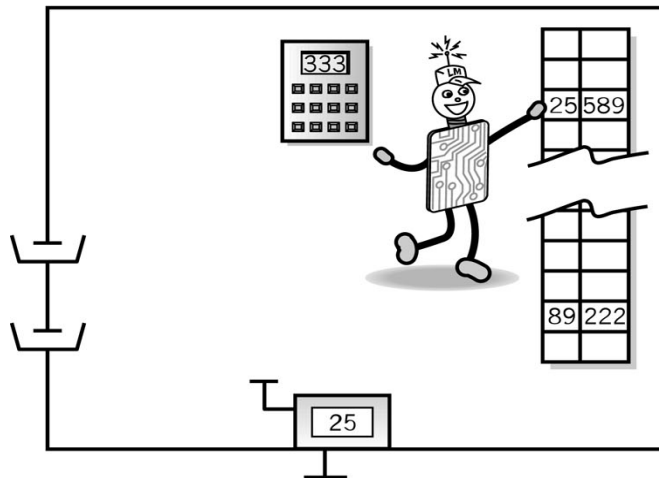
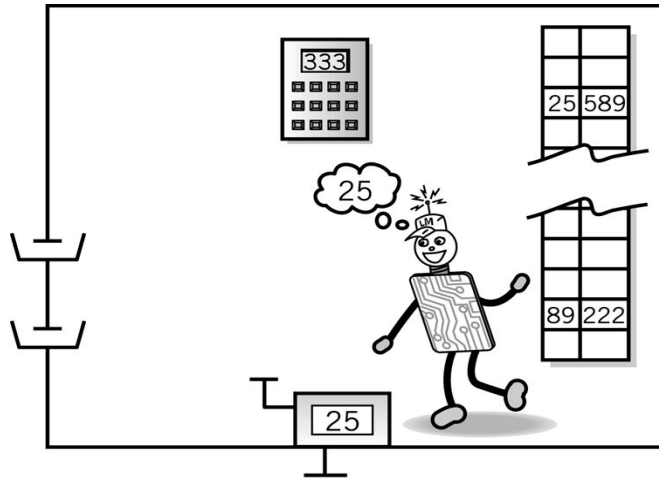
Két lépésre osztható:

- *Fetch (~kikeresés)*: Kisember kideríti, hogy milyen utasítást hajtson végre
- *Végrehajtás*: Kisember elvégzi az utasítás által definiált műveletet





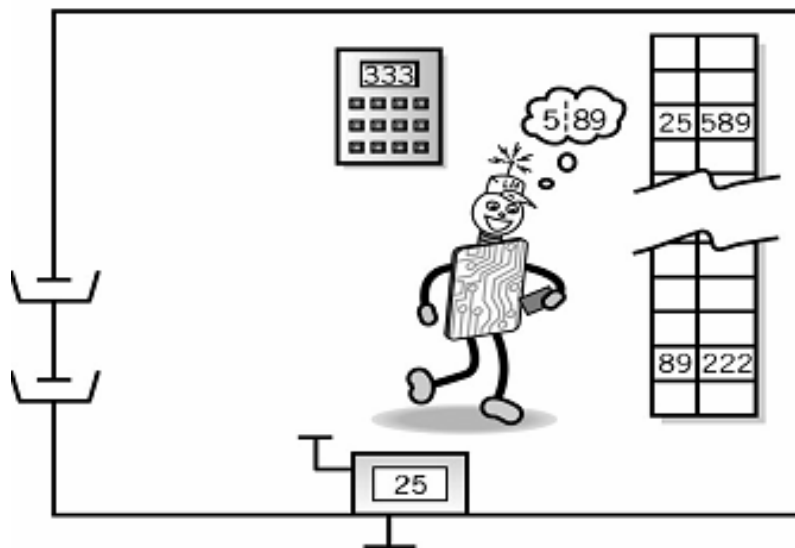
# Fetch része az utasítás- végrehajtási ciklusnak



1. Kisember beolvassa a címet az utasítás helyét tároló számlálóból (utasításszámláló)
2. Elmegy a postaládához, ami az utasítás-számlálóban tárolt értékhez tartozik



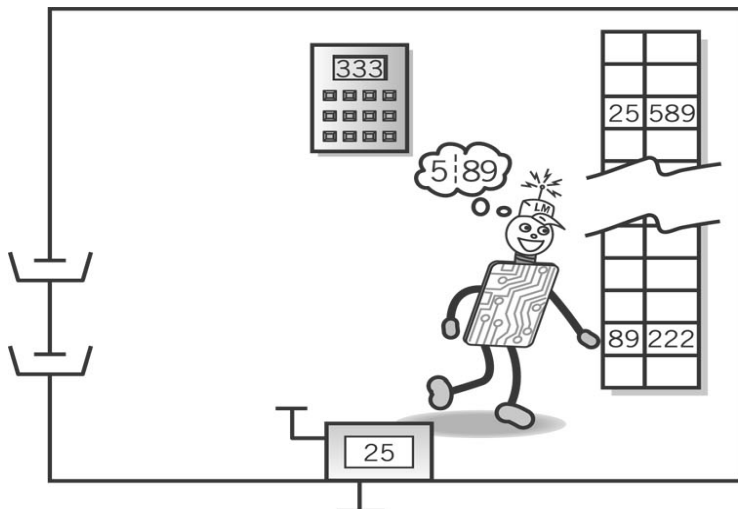
# Fetch, (folyt.)



3. A cetliről leolvassa postaládában tárolt a számot (visszarakja a cetlit, hátha szüksége lesz még rá később)

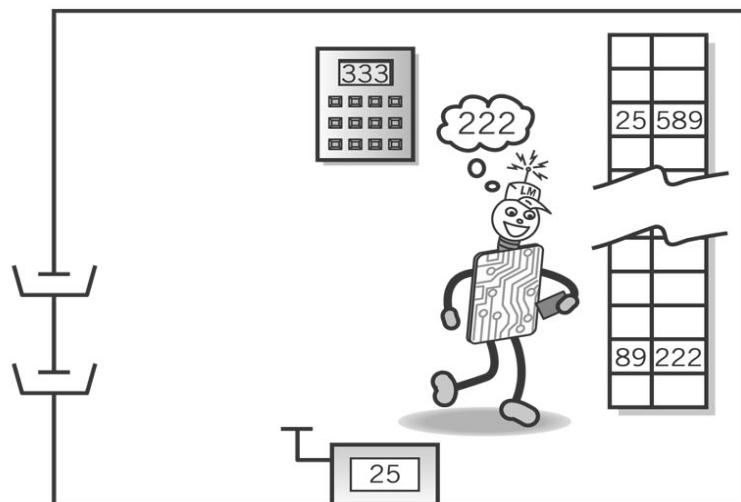


# Végrehajtás része az utasítás- végrehajtási ciklusnak



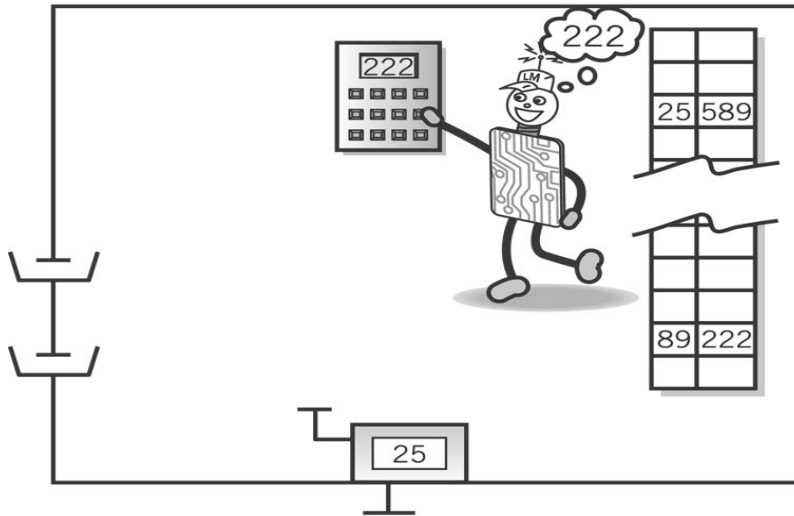
1. Kisember odamegy ahhoz a postaládához, ami az imént fetch-elt utasítás kódjában volt definiálva.

2. Kiolvassa a számot a postaládából (emlékezik a cserére, hátha szüksége lesz rá később).



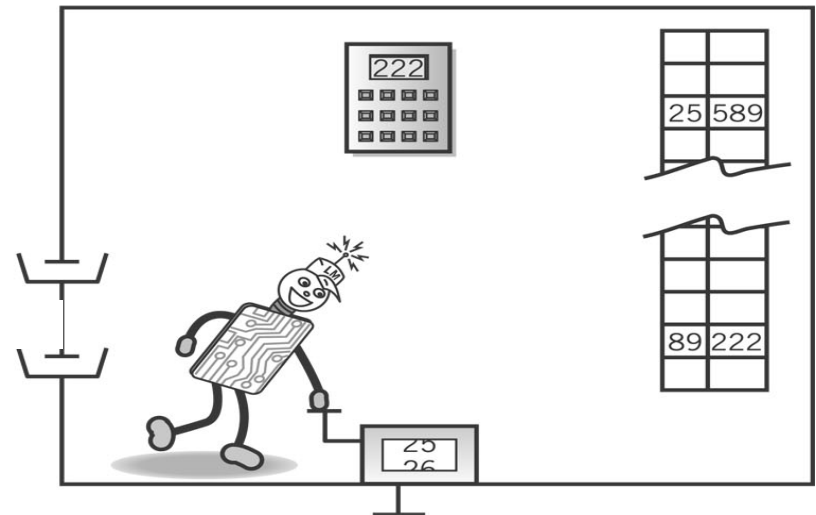


# Végrehajtás (folyt.)



3. Odamegy a számológéphez és begépel a számot.

4. Odamegy a az utasítás helyét tároló számlálóhoz (utasításszámlálóhoz) és megnyomja (inkrementálja), ami után készen áll fetch-elni a következő utasítást.





# Neumann architektúra (1945)

---

- Tárolt program elve
  - a programokat és az adatokat azonos memóriában tároljuk
- Memória használata a tartalom figyelembevétele nélkül
- Bináris számábrázolás használata
- Memória lineáris (szekvenciális) címzése