

2. előadás

- Számítógépek felépítése, alapfogalmak

Lovas Szilárd
SZE MTK MSZT
lovas.szilard@sze.hu
B607 szoba



Nem reprezentatív felmérés – kinek van ilyen számítógépe?



Nem reprezentatív felmérés – kinek van ilyen számítógépe?



Nem reprezentatív felmérés – kinek van ilyen számítógépe?



Nem számítógépnek látszó számítógépek

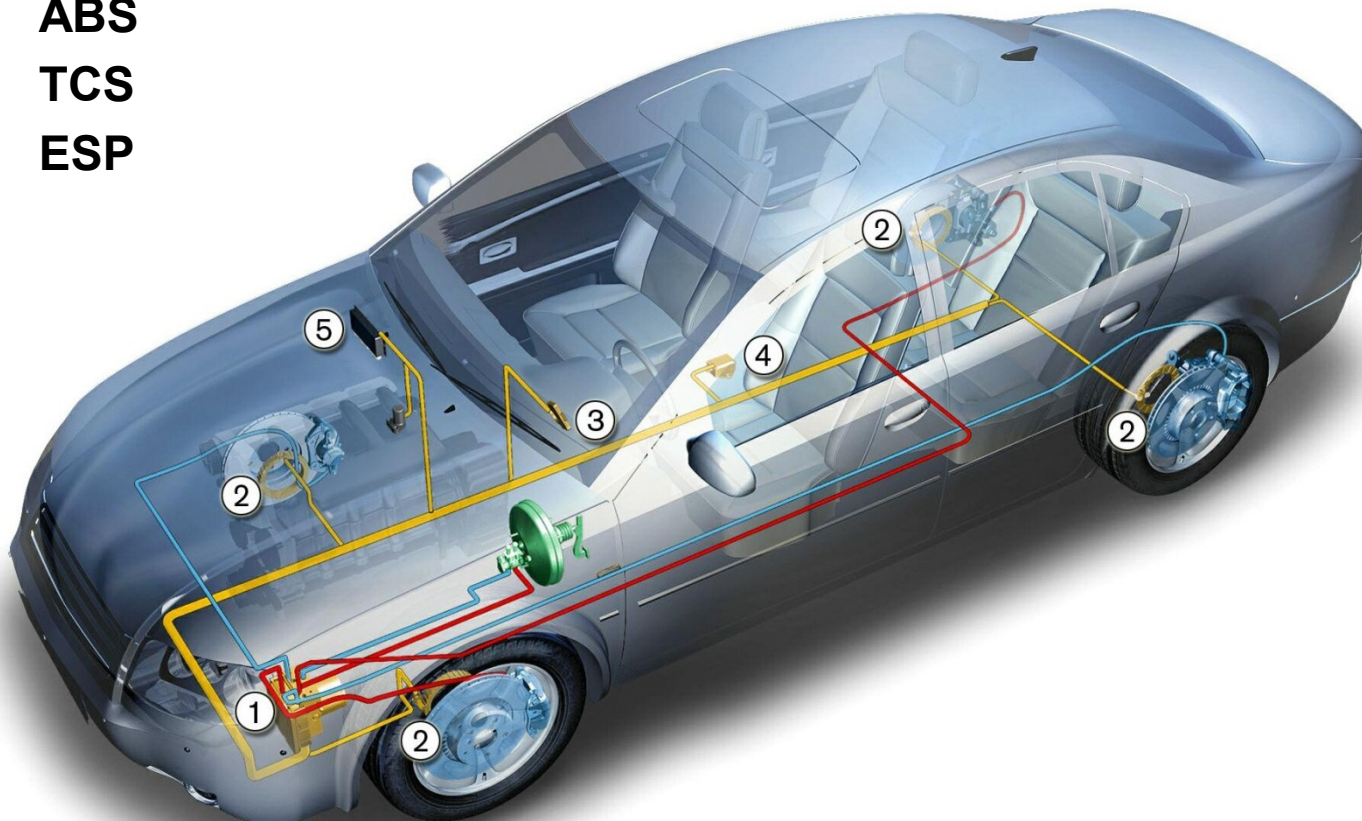


Nem számítógépnek látszó számítógépek



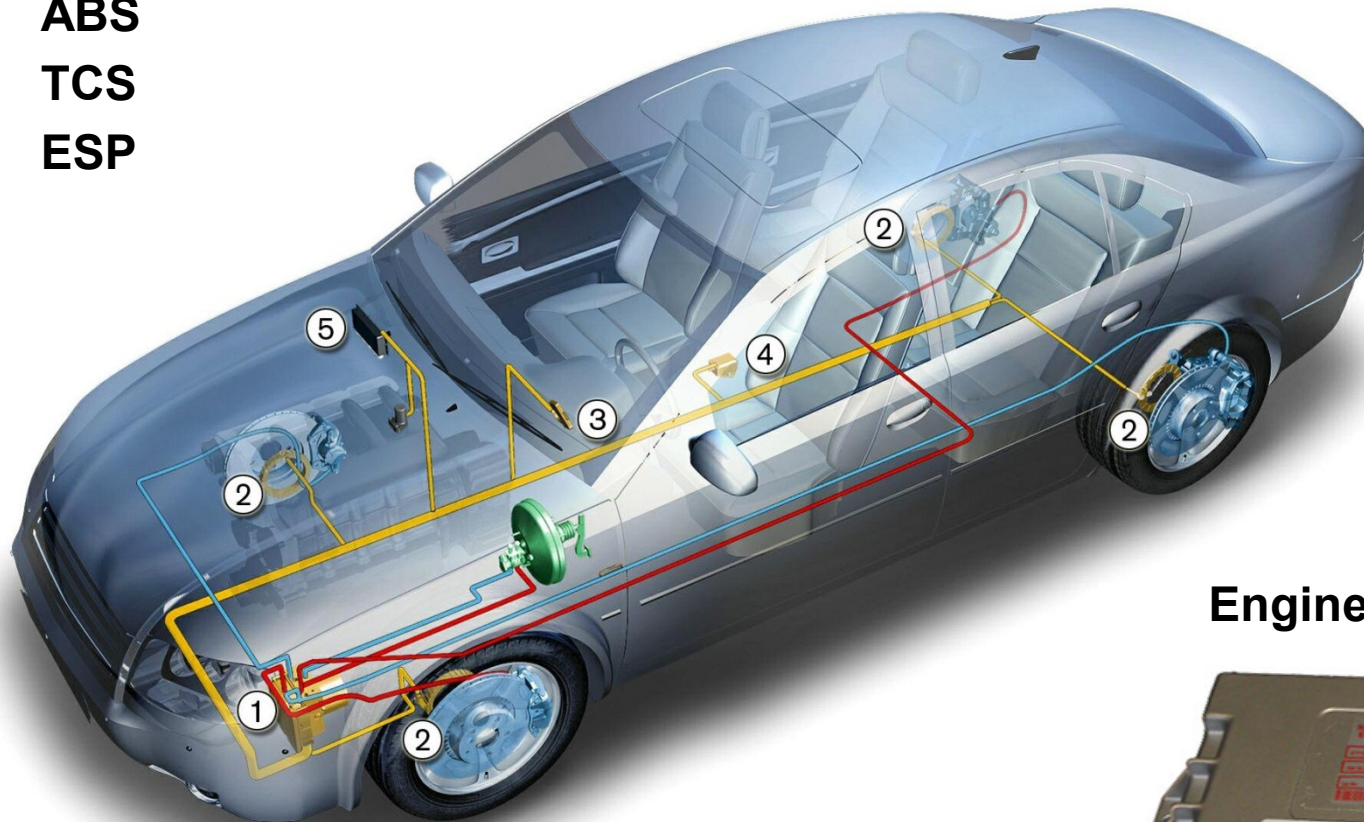
Nem számítógépnek látszó számítógépek

- ABS
- TCS
- ESP



Nem számítógépnek látszó számítógépek

- ABS
- TCS
- ESP

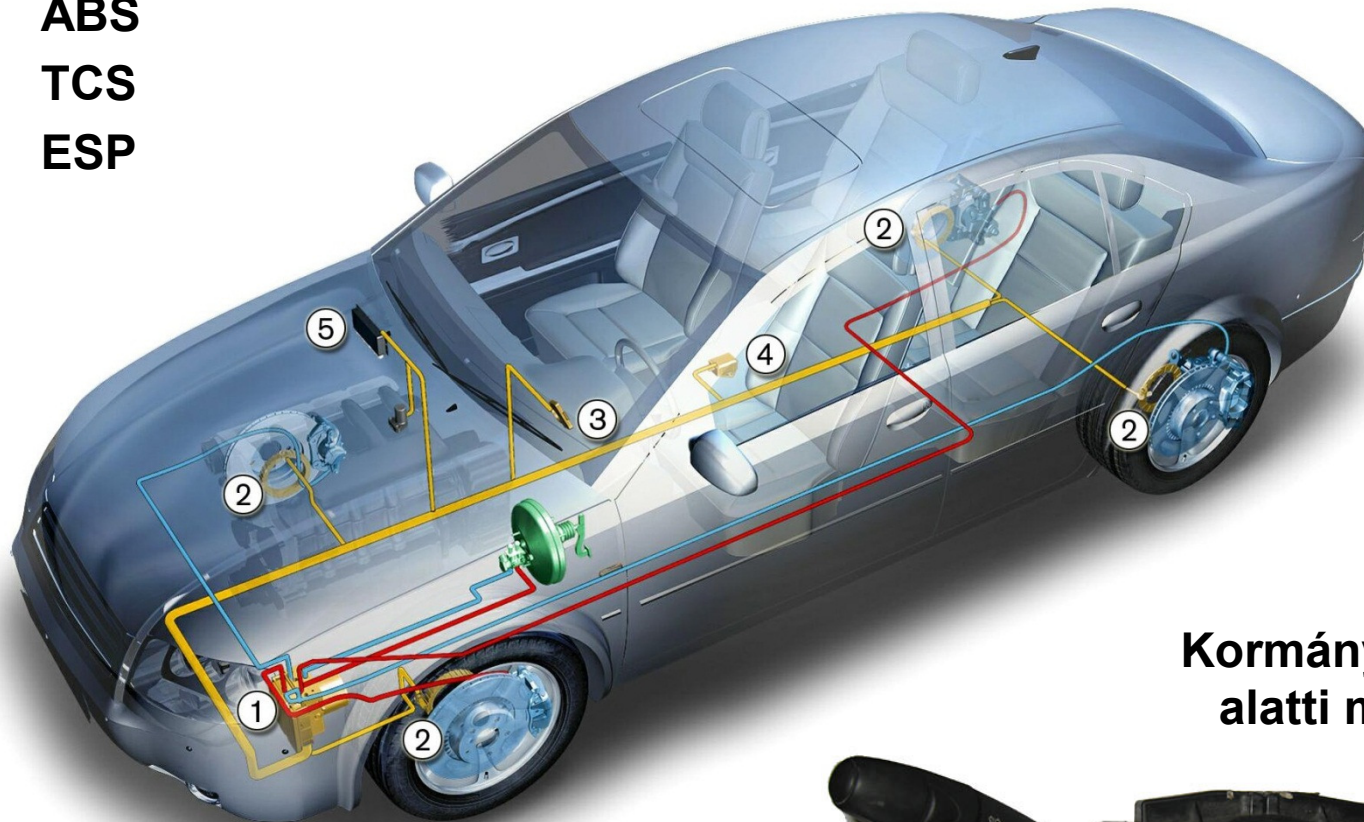


ECU
Engine Control Unit



Nem számítógépnek látszó számítógépek

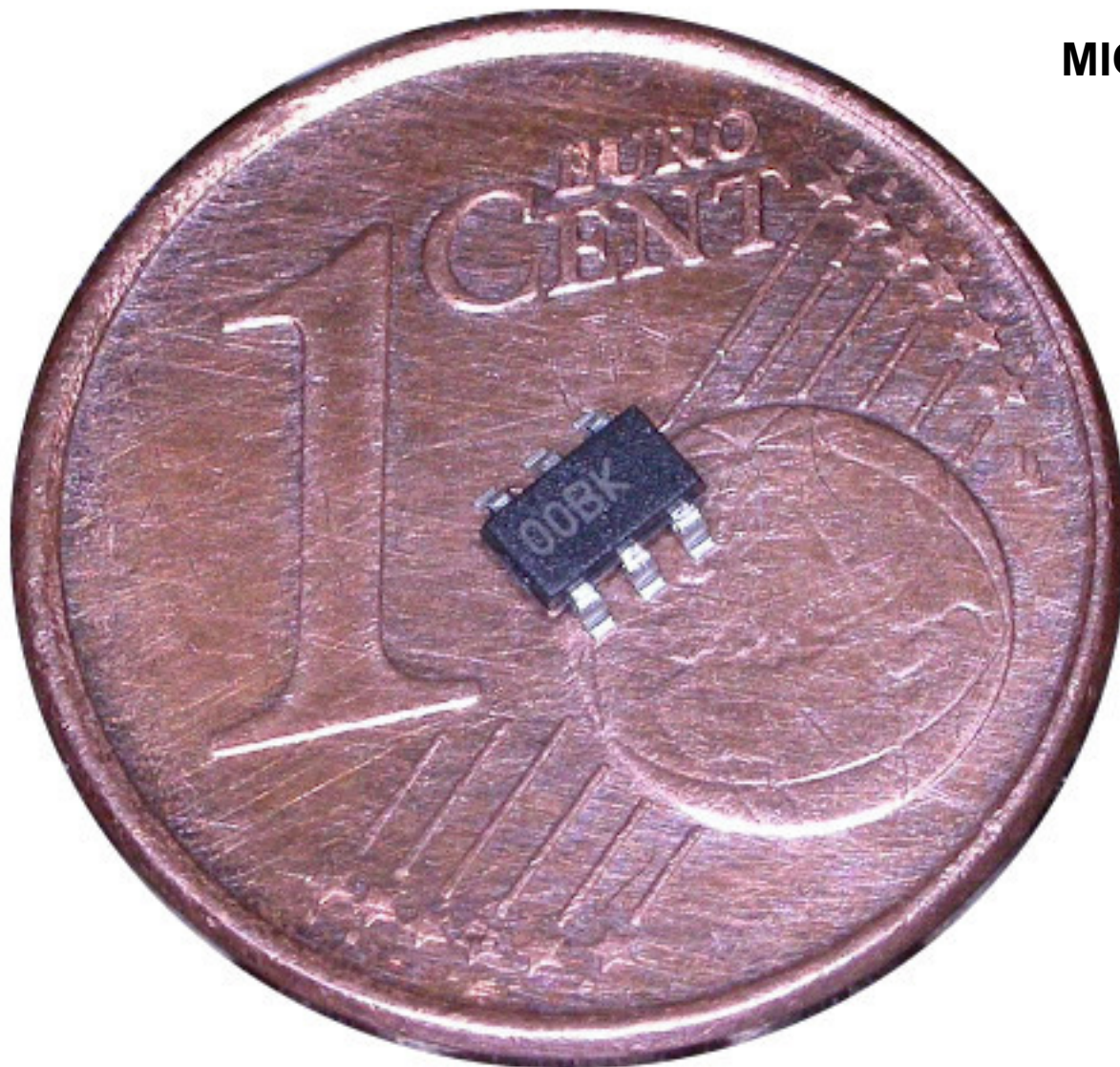
- ABS
- TCS
- ESP



**Kormánykerék
alatti modul**



Nem számítógépnek látszó számítógépek



MICROCHIP PIC10F200

Nem számítógépnek látszó számítógépek



Mi is az a számítógép?

Tágabb értelemben:



Mi is az a számítógép?

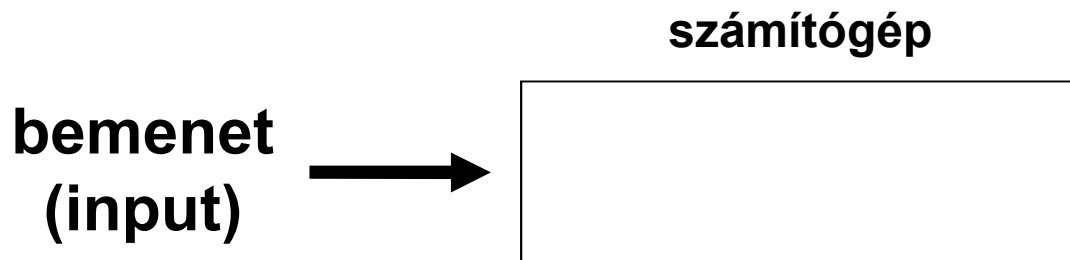
Tágabb értelemben:

számítógép



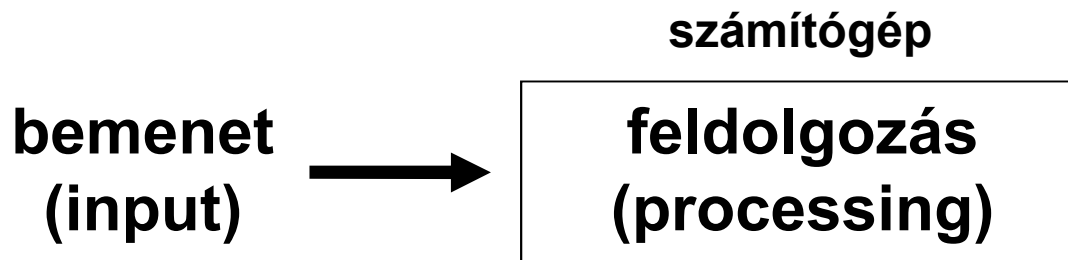
Mi is az a számítógép?

Tágabb értelemben:



Mi is az a számítógép?

Tágabb értelemben:



Mi is az a számítógép?

Tágabb értelemben:



Mi is az a számítógép?

Tágabb értelemben:



Determinisztikus működés



Mi is az a számítógép?

Szűkebb értelemben a következő felépítésű
(architektúrájú) gépeket nevezzük számítógépnek:

- Neumann architektúra
- Harvard architektúra
- Módosított Harvard architektúra



Részegységek - hogyan oldunk meg egy feladatot matematika órán?

Szükséges hozzávalók:



Részegységek - hogyan oldunk meg egy feladatot matematika órán?

Szükséges hozzávalók:

- papír
- ceruza, kéz
- számológép



Részegységek - hogyan oldunk meg egy feladatot matematika órán?

Szükséges hozzávalók:

- papír
- ceruza, kéz
- számológép
- agy
- szem, fül, száj



Részegységek - hogyan oldunk meg egy feladatot matematika órán?

Szükséges hozzávalók:

- papír
 - ceruza, kéz
 - számológép
 - agy
 - szem, fül, száj
- ➔
- memória



Részegységek - hogyan oldunk meg egy feladatot matematika órán?

Szükséges hozzávalók:

- papír → memória
- ceruza, kéz → busz
- számológép
- agy
- szem, fül, száj



Részegységek - hogyan oldunk meg egy feladatot matematika órán?

Szükséges hozzávalók:

- papír → memória
- ceruza, kéz → busz
- számológép → ALU
- agy
- szem, fül, száj



Részegységek - hogyan oldunk meg egy feladatot matematika órán?

Szükséges hozzávalók:

- papír → memória
- ceruza, kéz → busz
- számológép → ALU
- agy → CU
- szem, fül, száj



Részegységek - hogyan oldunk meg egy feladatot matematika órán?

Szükséges hozzávalók:

- | | | |
|-------------------|---|---------|
| ■ papír | → | memória |
| ■ ceruza, kéz | → | busz |
| ■ számológép | → | ALU |
| ■ agy | → | CU |
| ■ szem, fül, száj | → | I/O |



Részegységek - hogyan oldunk meg egy feladatot matematika órán?

Szükséges hozzávalók:

- | | | | |
|-------------------|---|---------|----------------------|
| ■ papír | → | memória | |
| ■ ceruza, kéz | → | busz | |
| ■ számológép | → | ALU | } CPU,
processzor |
| ■ agy | → | CU | |
| ■ szem, fül, száj | → | I/O | |



Részegységek - hogyan oldunk meg egy feladatot matematika órán?

Szükséges hozzávalók:

- | | | | |
|-------------------|---|---------|----------------------|
| ■ papír | → | memória | |
| ■ ceruza, kéz | → | busz | |
| ■ számológép | → | ALU | } CPU,
processzor |
| ■ agy | → | CU | |
| ■ szem, fül, száj | → | I/O | |

(Speciális célú alkalmazásokban nem biztos, hogy minden részegységre szükség van)



Neumann elvek

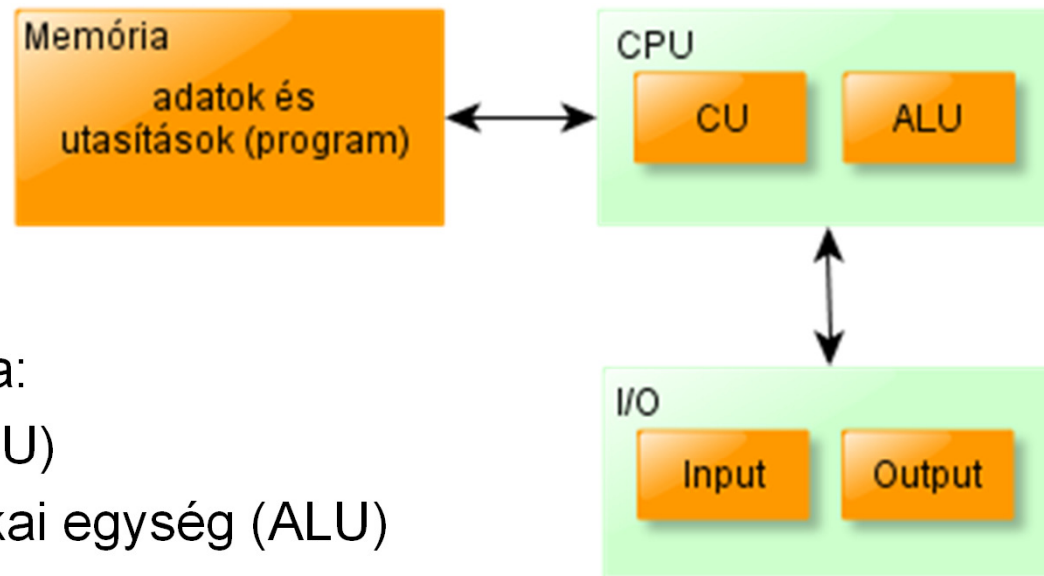
Neumann elvek

(1945 First Draft of a Report on the EDVAC):

- Teljesen elektronikus működés
- Kettes számrendszer használata
- Soros utasítás végrehajtás
- Belső memória használata, tárolt program elve
- Univerzális gép



Számítógép - Neumann architektúra

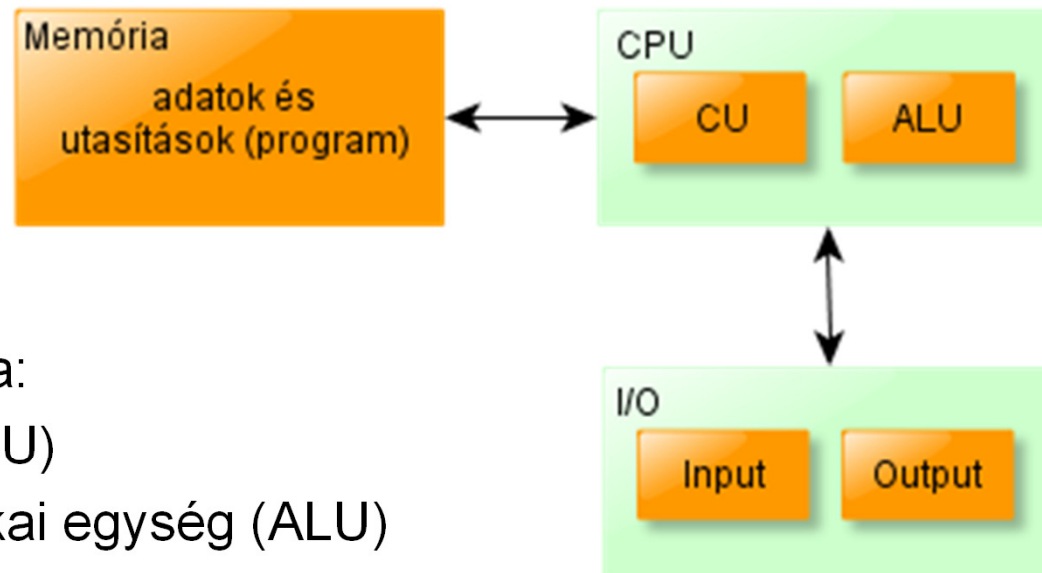


Neumann architektúra:

- Vezérlő egység (CU)
- Aritmetikai és logikai egység (ALU)
- Memória (program + adat)
- Input/output egység



Számítógép - Neumann architektúra

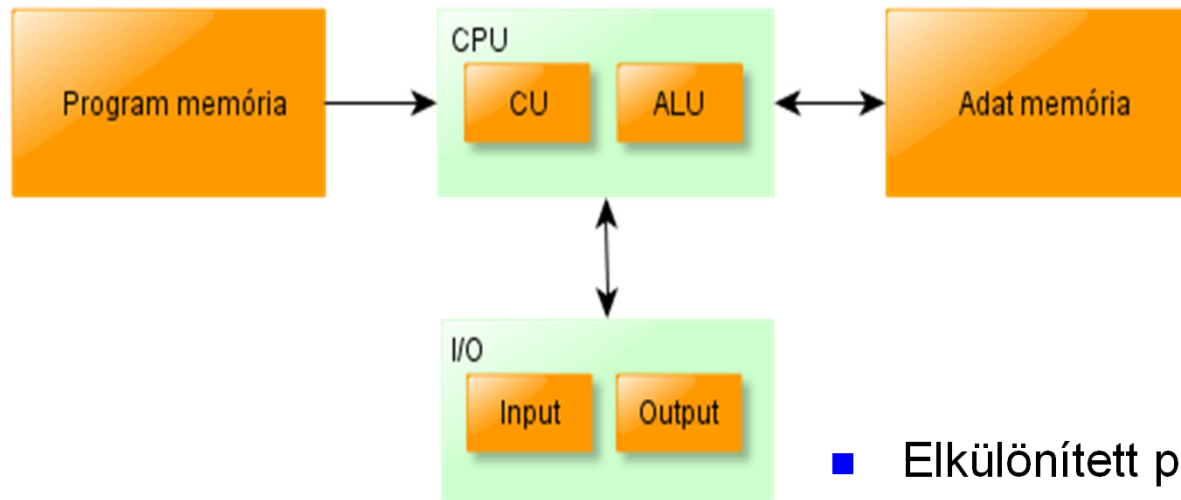


Neumann architektúra:

- Vezérlő egység (CU)
- Aritmetikai és logikai egység (ALU)
- Memória (program + adat)
- Input/output egység
- Busz



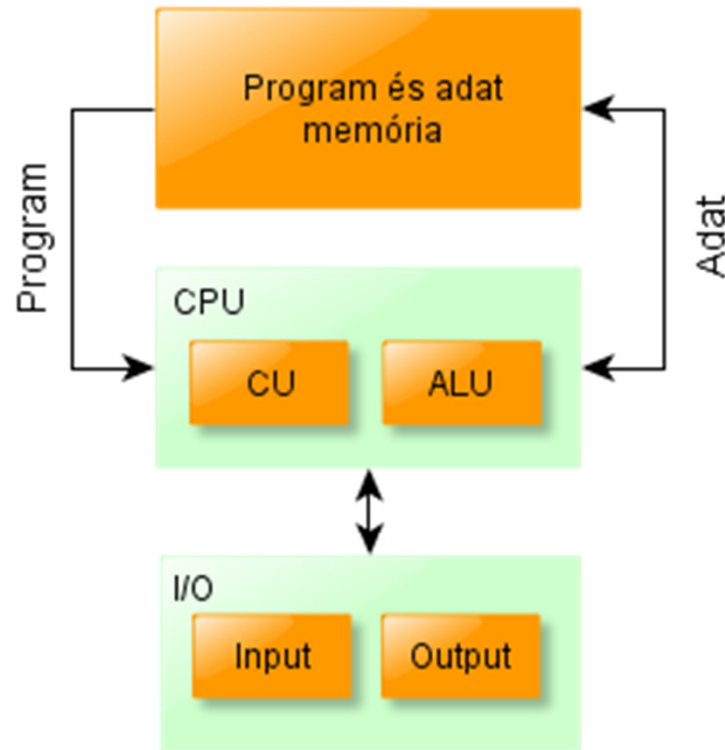
Számítógép - Harvard architektúra



- Elkülönített program és adat memória
- Nagyobb biztonság
- Kedvezőbb teljesítmény
- Lehetőség eltérő adat és program memória szervezésre
- Nem képes saját programját generálni
- Főleg kisméretű számítógépekben (mikrokontrollerekben) elterjedt



Számítógép - módosított Harvard architektúra



- Mentés a Neumann architektúra hiányosságaitól
- Képes a saját programját generálni és futtatni
- Esetleges hátrány: az adat és programmemória nem lehet különböző szervezésű
- Az általános célú számítógépekben széleskörűen használt felépítés (például a PC)

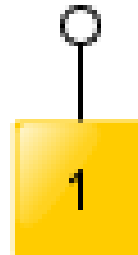


Számítógép - Memória működése

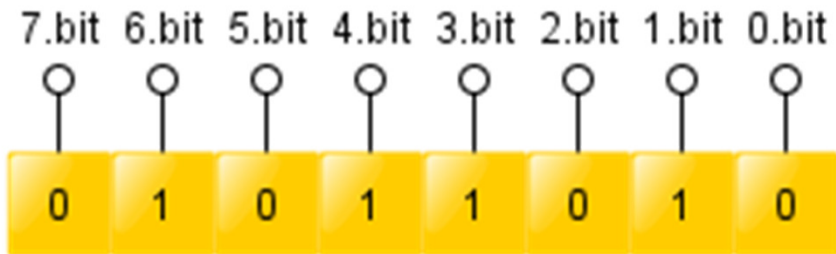
alacsony feszültség



magas feszültség



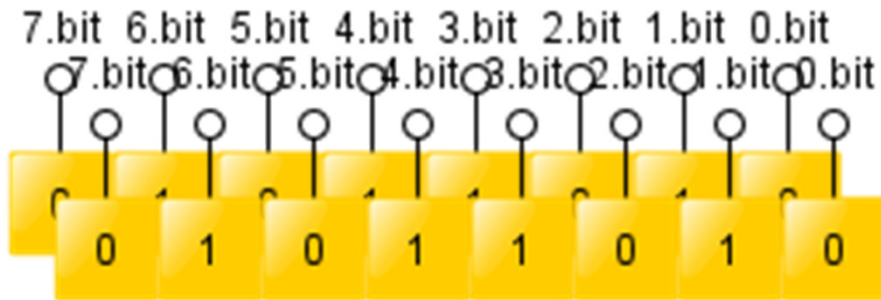
Számítógép - Memória működése



= 1 bájt (8 vezeték)



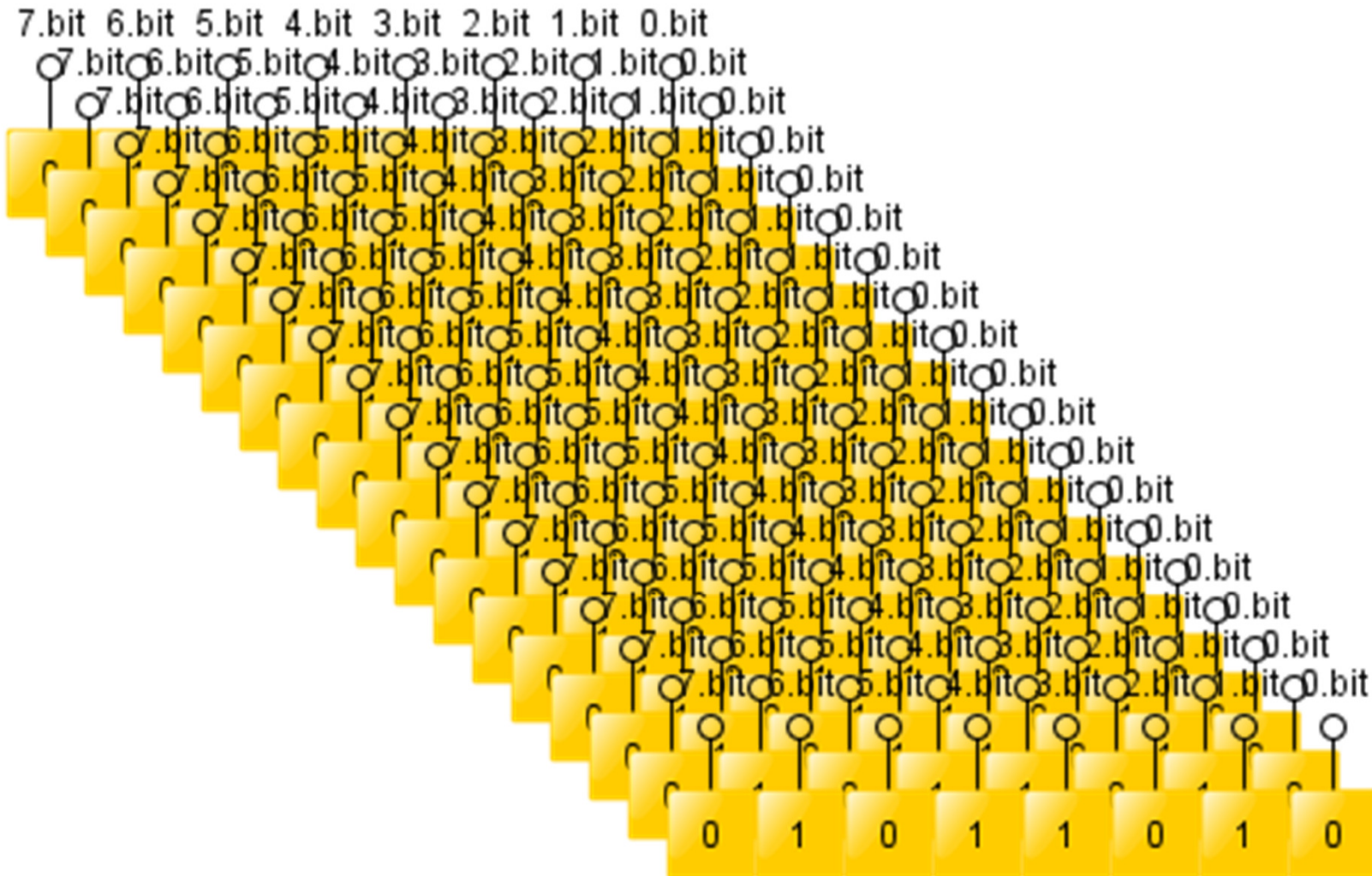
Számítógép - Memória működése



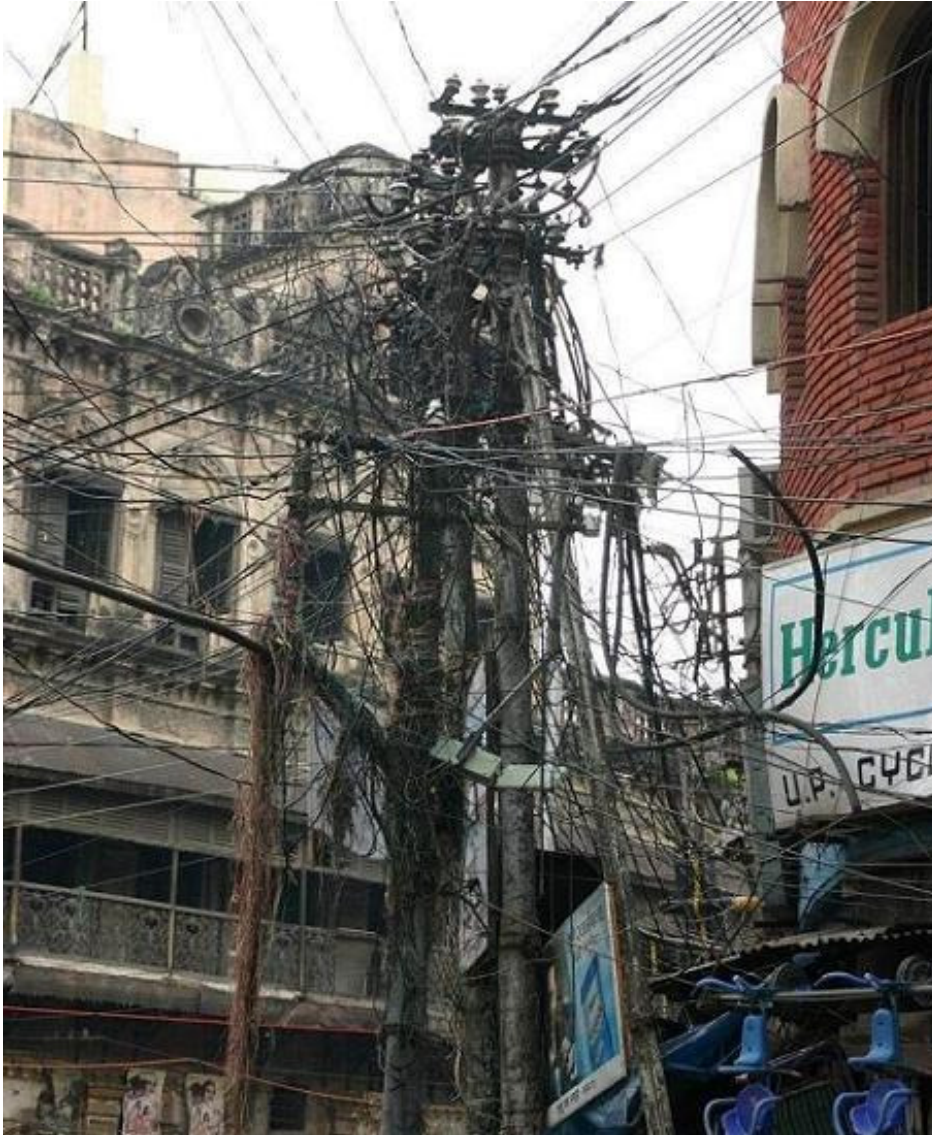
= 1 bájt (8 vezeték)



Számítógép - Memória működése



Számítógép - Memória működése



**Probléma megoldása:
multiplex kezelés
(memóriacímek)**

- **Adatvezetékek közösek**
- **Memóriát címmel látjuk el**



Számítógép - Memória működése



=

Cím

Adat

0000

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0001

0	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

0002

0	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

0003

0	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

▪

▪

▪

FFFE

0	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

FFFF

0	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---



Számítógép - Memória működése

Neumann elvek:
Teljesen elektronikus
Kettes számrendszer



=

Cím	Adat							
0000	0	0	0	0	0	0	0	0
0001	0	1	0	1	1	0	1	0
0002	0	1	0	1	1	0	1	0
0003	0	1	0	1	1	0	1	0
FFFE	0	1	0	1	1	0	1	0
FFFF	0	1	0	1	1	0	1	0



Számítógép - Memória működése

Neumann elvek:
Teljesen elektronikus
Kettes számrendszer



=

Kódolni kell a cím és
adat értékeket is !

Cím

Adat

0000

0	0	0	0	0	0	0	0
---	---	---	---	---	---	---	---

0001

0	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

0002

0	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

0003

0	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

▪

▪

▪

FFFE

0	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

FFFF

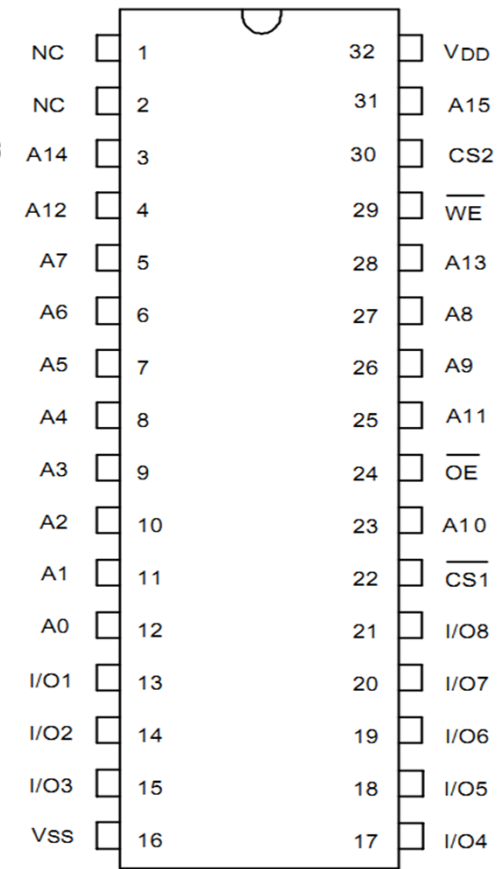
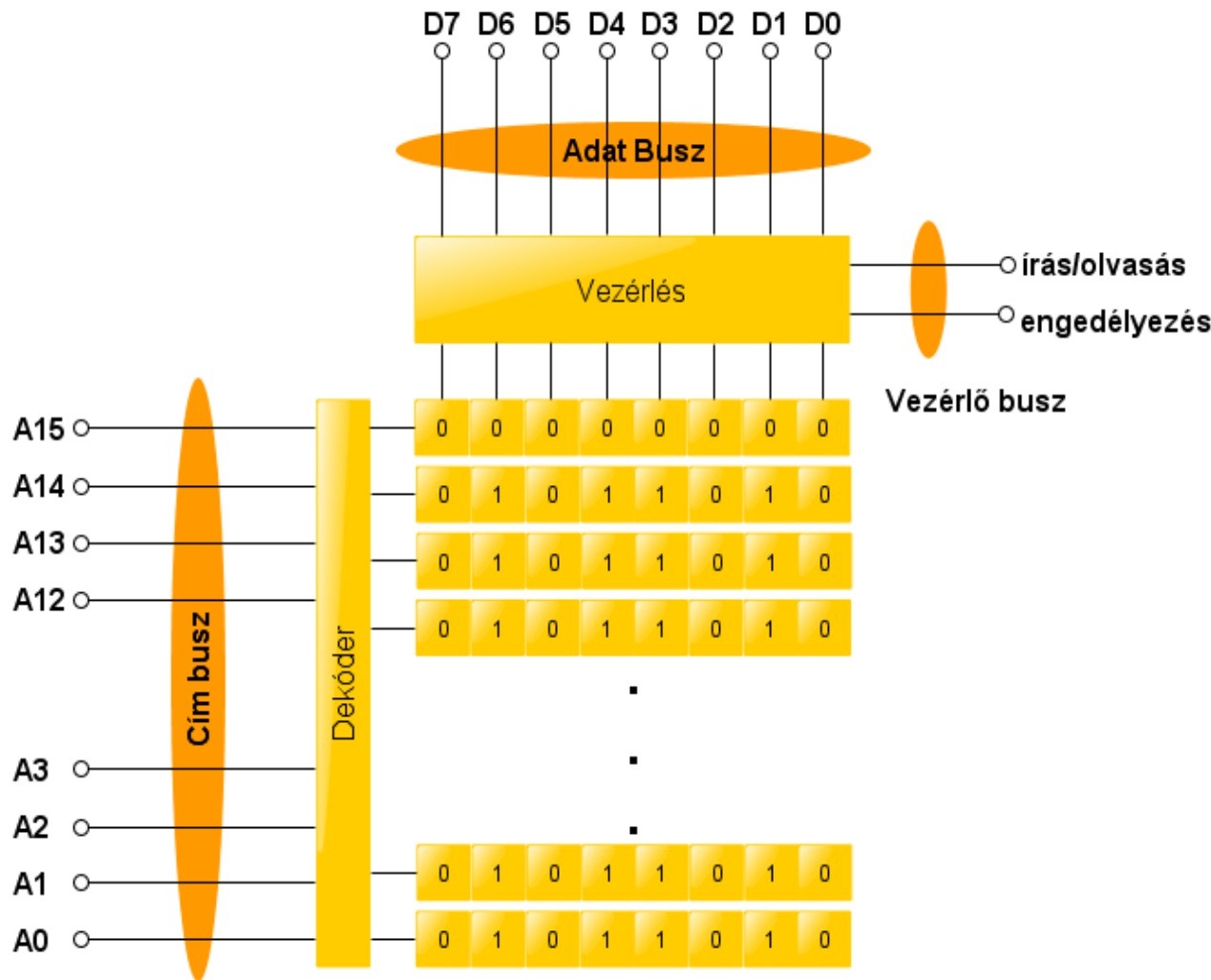
0	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---



Számítógép - Memória működése

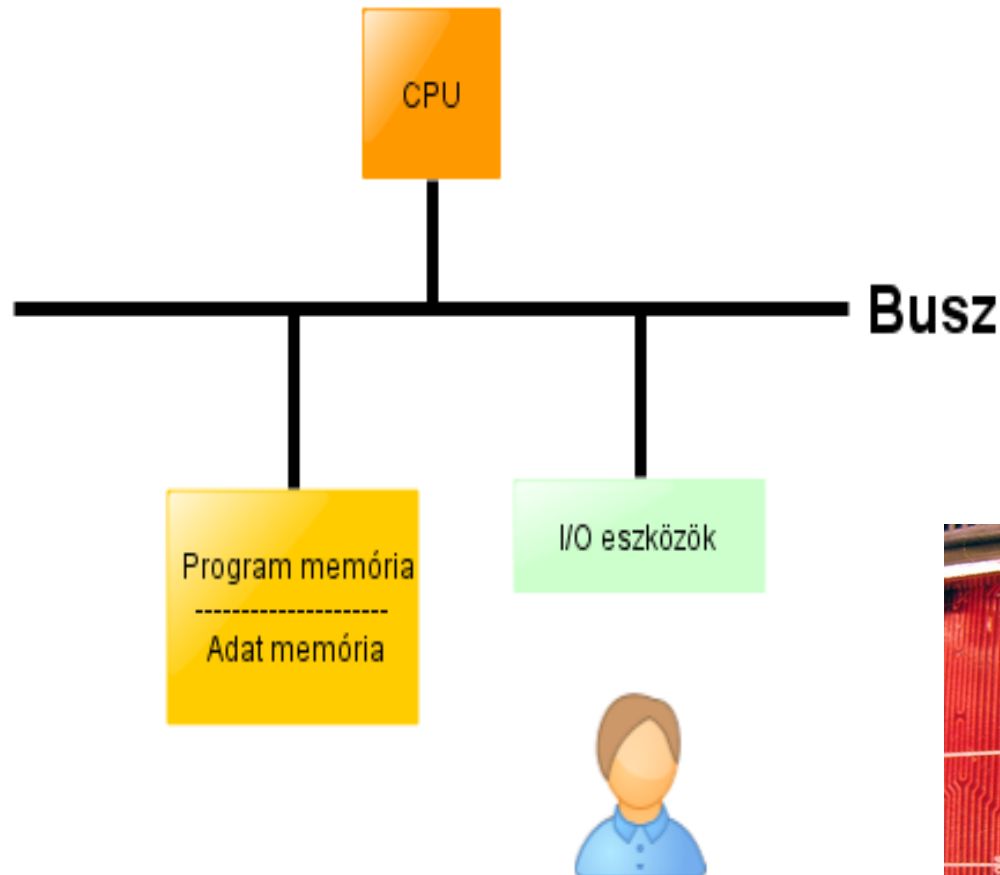
$$k = \lceil \log_2 S_m \rceil$$

$$S_m = 2^k$$



Példa: w24512A – 64Kx8bit RAM

Számítógép - Busz

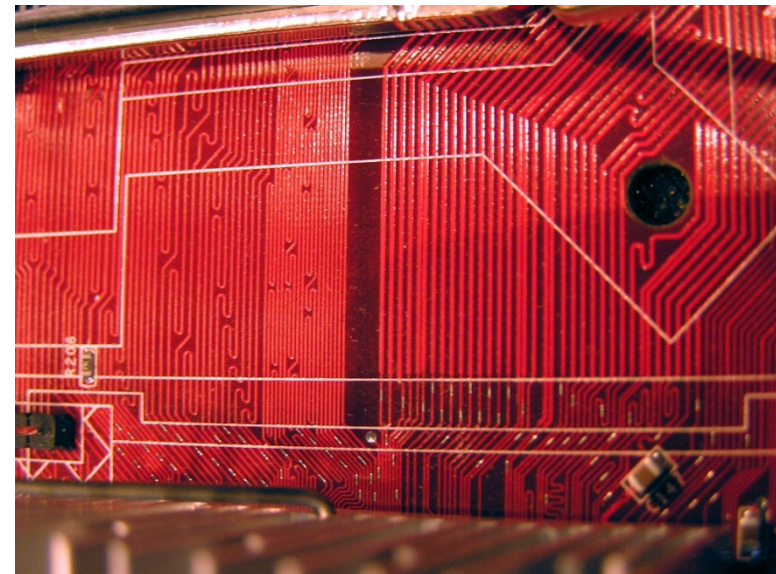


■ Busz

- Adat vezetékek
- Cím vezetékek
- Vezérlő jelek

■ Átvitel

- Soros v. párhuzamos
- Szinkron v. aszinkron



- Sáv szélesség [b/s]
- Késleltetés [s]



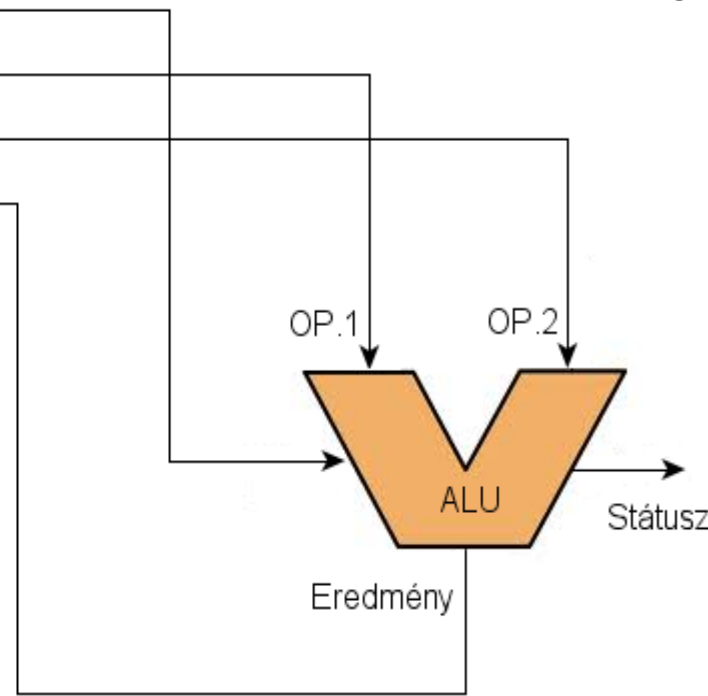
Central Processing Unit felépítése – ALU

Cím **Adat**

0000	0	1	0	1	0	0	0	1
0001	0	1	0	1	1	0	1	0
0002	0	1	0	1	1	0	1	0
0003	1	0	1	1	0	1	0	0
FFFE	0	0	0	0	0	0	0	0
FFFF	0	0	0	0	0	0	0	0

Utasítás (művelet)

ALU - Aritmetikai logikai egység:
az egyik legfontosabb összetevő -
a CPU „számológépe”



Eredmény = OP1 művelet OP2

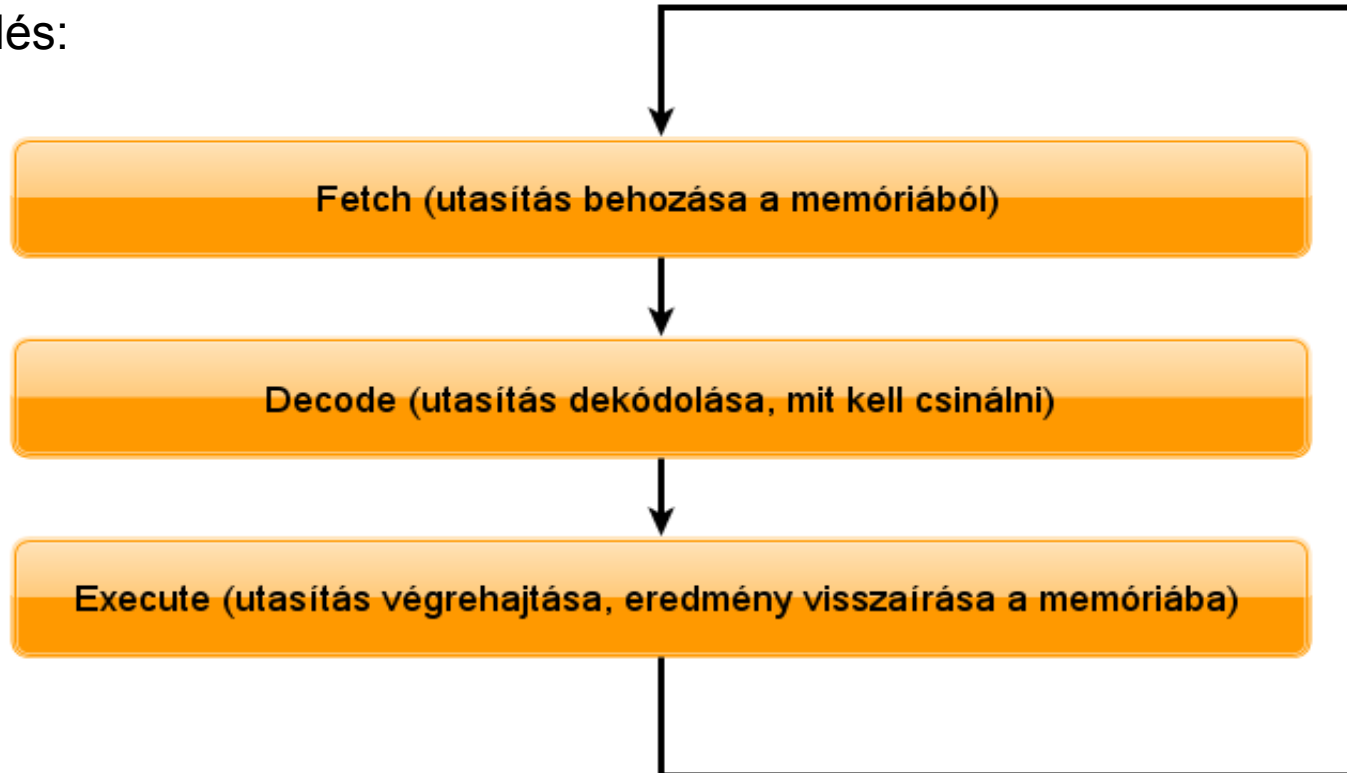
Gyakran megvalósított műveletek (CPU függő): +, - , (:,*),
AND, OR, XOR, <<, >> (bit mozgatás), NOT (egy operandus)



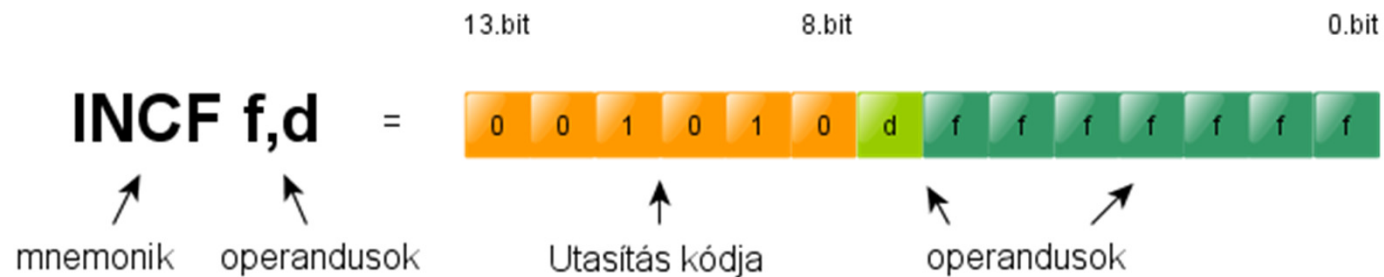
Central Processing Unit felépítése – Control Unit (CU)

Feladata a működés időzítése, Fetch, Decode, Execute ciklus vezérlése.

Működés:



CPU – példa egy gépi utasításra



Operandusok száma:

- 0,1,2 operandusú utasítások

Címzési módok (mit tartalmaz az utasítás)

- Direkt (adat, memória)
- Indirekt
- Abszolút, relatív

Ortogonalis az utasításkészlet, ha minden utasítás minden regiszterrel és címzési móddal használható.



CPU – utasításkészlet

Az utasítások összességét utasításkészletnek (instruction set) nevezzük, amely jellemző az adott processzorra.

- Adat vagy memóriakezelő utasítások (regiszter vagy memória olvasása, írása, másolása)
- Aritmetikai utasítások (+, -, /, *, bitmanipulációs műveletek például: eltolás, negáció; összehasonlítás például: kisebb, nagyobb, egyenlő)
- Vezérlő utasítások (feltételes ugrás, feltétel nélküli ugrás)
- Egyéb (például energiagazdálkodást szabályzó utasítások)

Két különböző megközelítés:

- RISC (Reduced instruction set computer)
- CISC (Complex instruction set computer)



CPU – további építőelemek

Regiszterek: kisméretű, gyorsan elérhető, tároló elemek a processzoron belül

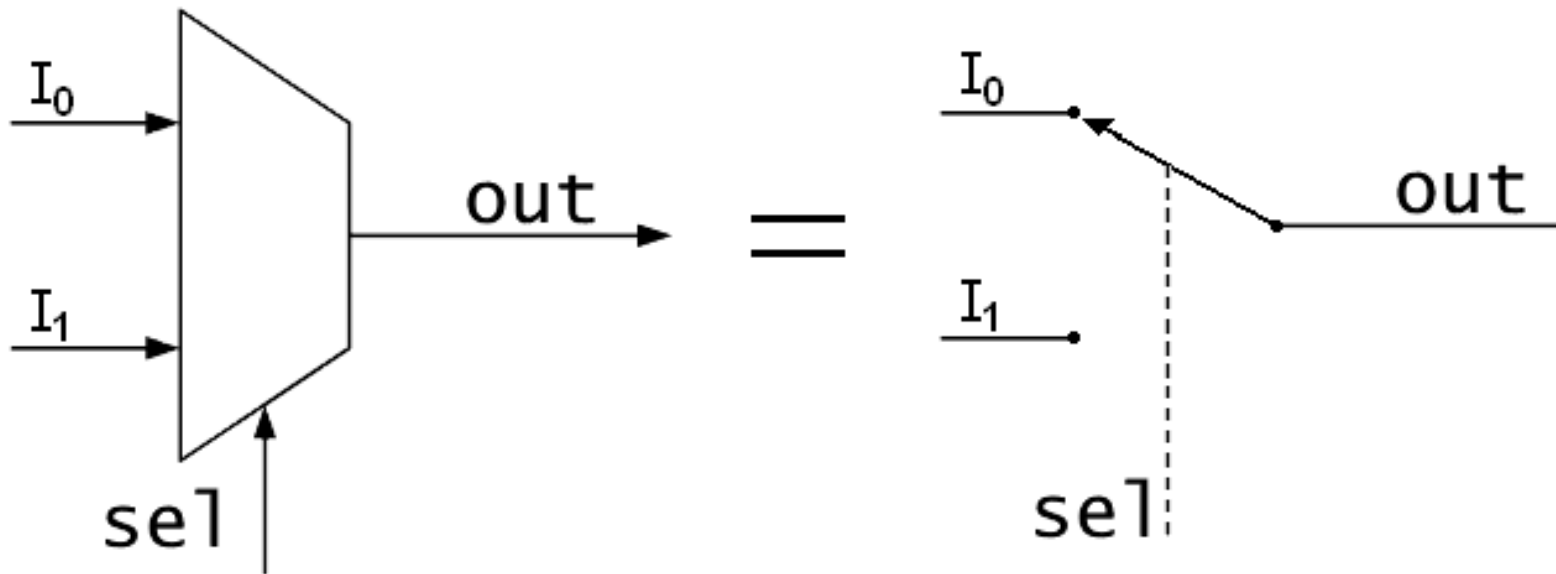
- Általános célú regiszter – szabadon használható adatok tárolására
pl.: akkumlátor (AC) – hány bites a CPU ?
- Speciális célú regiszterek – a benne található érték valamilyen speciális jelentéssel bír a CPU számára. Például:
 - Program számláló (Program Counter, PC)
 - Utasítás regiszter (Instruction Register, IR)
 - Státusz regiszter

Reset - indításkor a PC regiszter egy bizonyos meghatározott memóriacímre mutat (reset vector) például: Intel x86: FFFFFFF0H



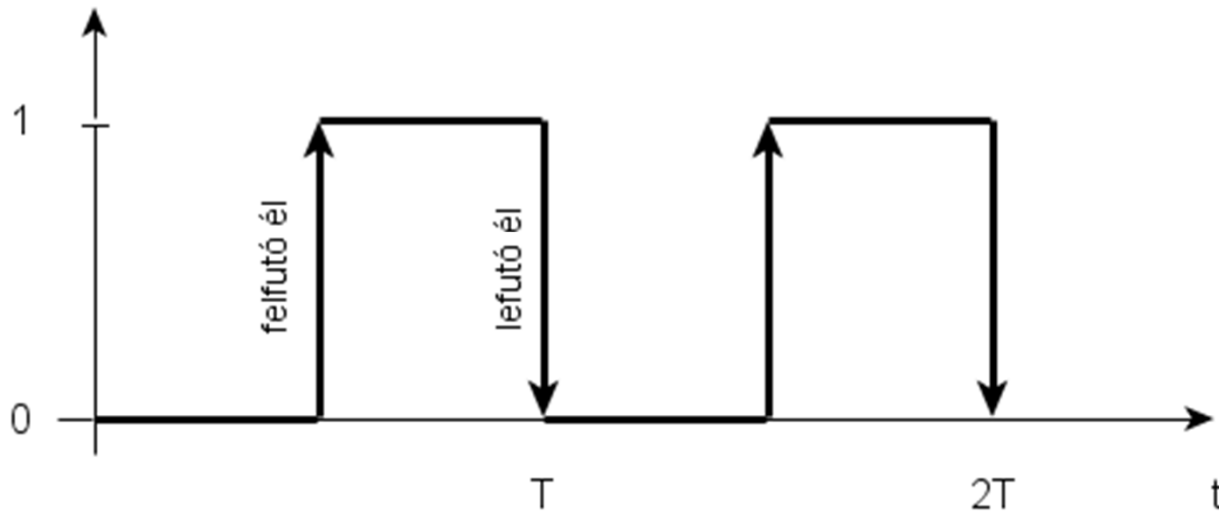
CPU – további építőelemek

Multiplexerek: útválasztók, váltókapcsolók



Órajel

A számítógépek általában szinkron, sorrendi digitális hálózatok, megfelelő működésükhöz szinkronizáló jelre van szükség.



Input / Output eszközök

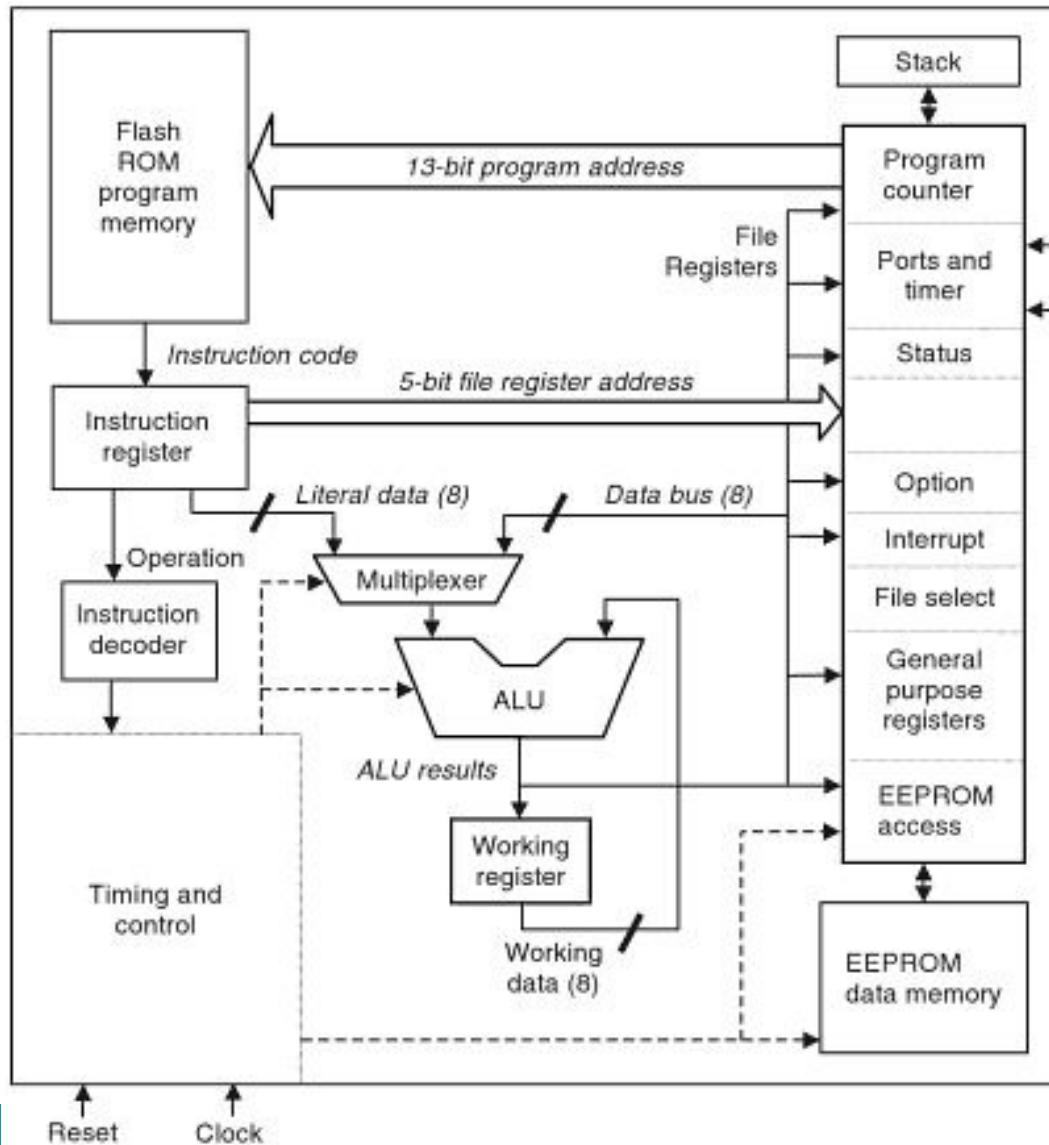
Feladatuk a számítógép és a környezet közötti kapcsolat megteremtése.

A CPU szemszögéből alapvetően két fajta megközelítés létezik:

- Memóriába ágyazott I/O. Elérés az (adat) memória eléréséhez használt load, store utasításokkal. Ebben a processzor az adott I/O eszközt egy memória területnek látja.
- Dedikált I/O portok használata. Elérés dedikált (kimondottan erre szolgáló) in, out parancsokkal.



Számítógép felépítése



PIC16f84 mikrokontroller
(komplett számítógép)

- Harvard architektúra
(14 bites utasítások, 8 bites adatok)
- (adat) memóriába ágyazott I/O



Összefoglalás

- Nem számítógépnek látszó számítógépek
- Számítógép fogalma
- Architektúrák, előnyök-hátrányok
- Részegységek (memória, busz, ALU, CU, I/O) felépítése és működése
- Kevés technikai részlet



Ajánlott irodalom, hasznos linkek

- Előadás diák, jegyzet kivonat a C100-ban
- Neumann elv
<http://web.mit.edu/STS.035/www/PDFs/edvac.pdf>
<http://www.feltalalok.hu/tudosok/neumannjanos/html/neujantal2.htm>
- PIC16F84 <http://ww1.microchip.com/downloads/en/devicedoc/35007b.pdf>
- X86 utasításkészlet http://en.wikipedia.org/wiki/X86_instruction_listings
- CPU/számítógép szimulátor <http://ozark.hendrix.edu/~burch/socs/hymn/>

