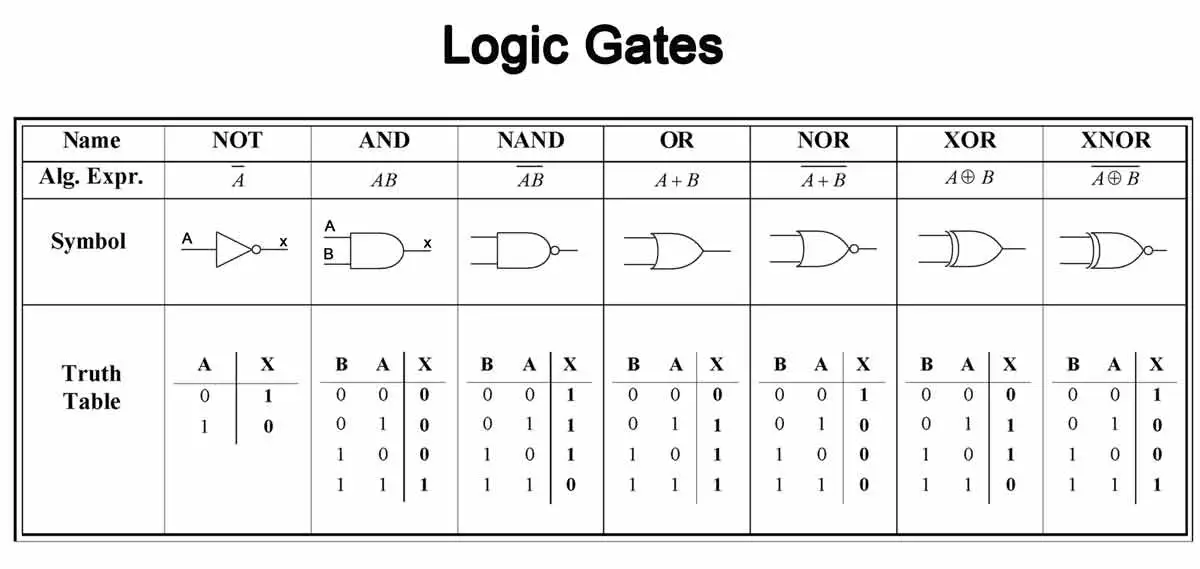
**Mikroprocesszor**

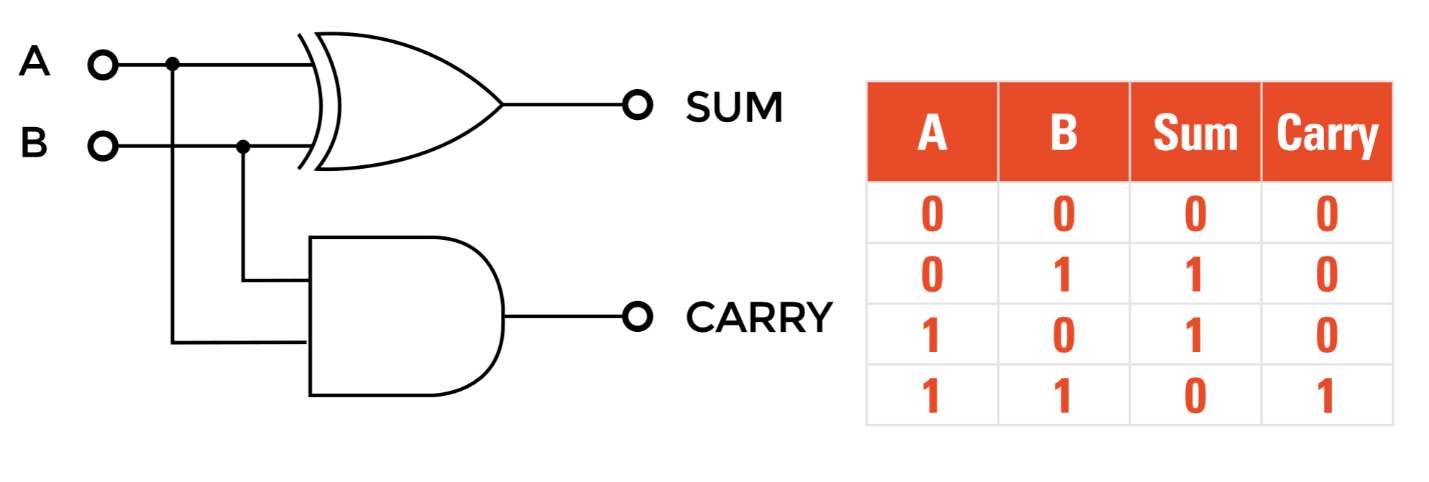
Bináris logika a működése minden számítógépes rendszernek

ALU – aritmetikai logikai egység, feladata számítások elvégzése

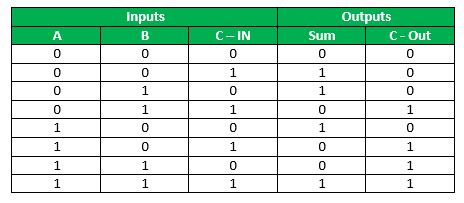
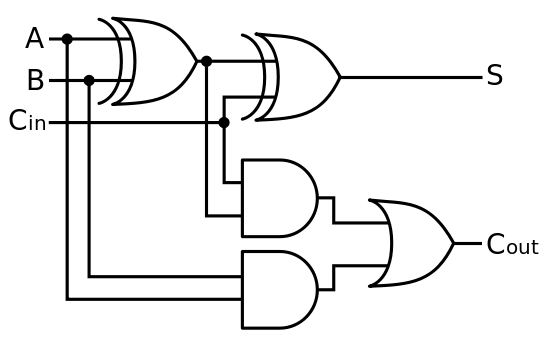
ALU kombinációs hálózat részei:

* Két adatbemenet (operandusok, A B)
* Eredménykimenet
* Állapotkimenet(ek)
* Túlcsordulás bemenet
* Működést befolyásoló eszköz(logikai kapuk)

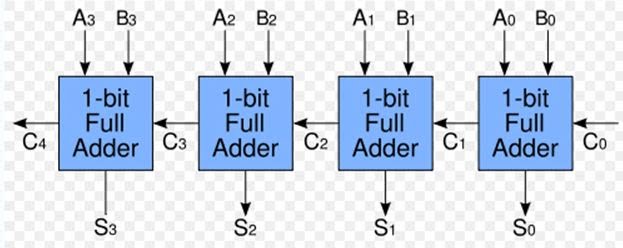
ALU kapuk:  
NOT  
OR  
AND  
NOR  
NAND  
XOR  
XNOR

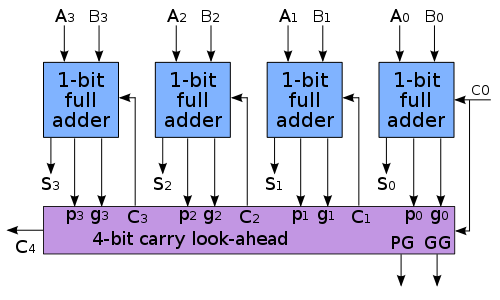
1 bites félösszeadó

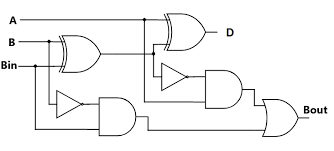
S = A XOR B  
CARRY = A AND B

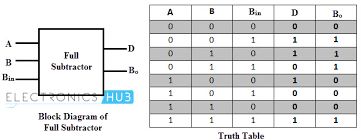
1 bites összeadó  


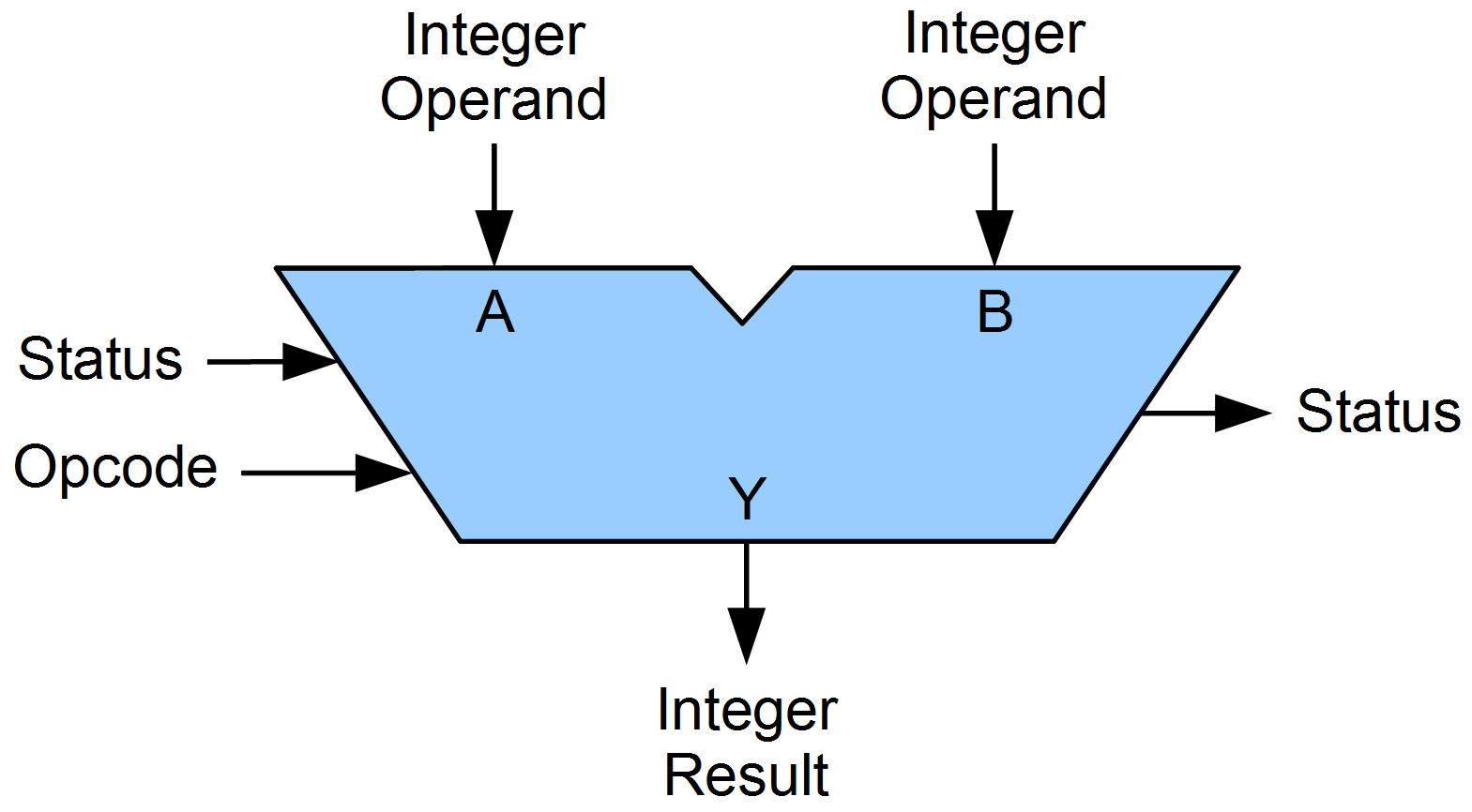
S = (A XOR B) XOR Cin  
CARRY = (A AND B) OR (Cin AND)...

4 bites teljes összeadó  
  
A teljes összeadók sorba kapcsolásával elméletben tetszőleges bitszélességű összeadó áramkör létrehozható.  
A túlcsordulás sorosan terjed végig az összeadókon, minél több bites összeadó, annál lassabb a rendszer működése.

4 bites teljes összeadó Carry Look Ahead áramkör segítségével

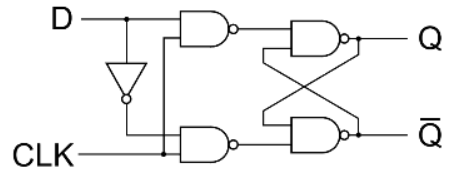
1 bites teljes kivonás, kivonás esetén Borrow bit  
R = (A XOR B) XOR Cin  
CARRY = ((NOT A) AND B) OR (Cin AND(NOT(A XOR B)))

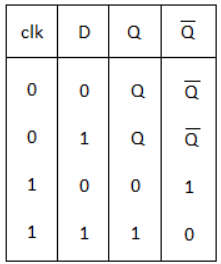


összeadó/kivonó  
Mode(status) bemenet(0-összead, 1-kivon)XOR kapuk programozható inverterek

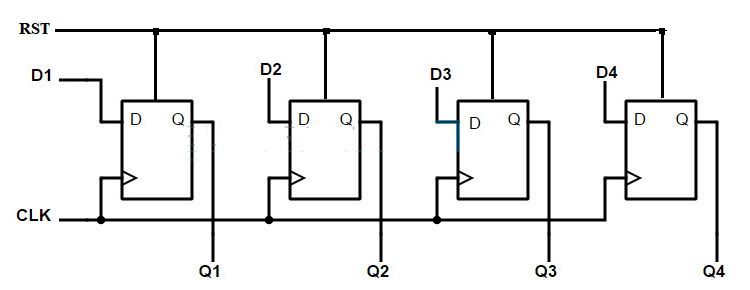
**Regiszterek, multiplexerek, számlálók**

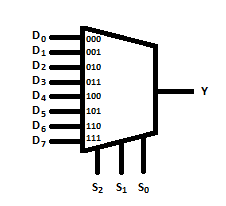
D tároló



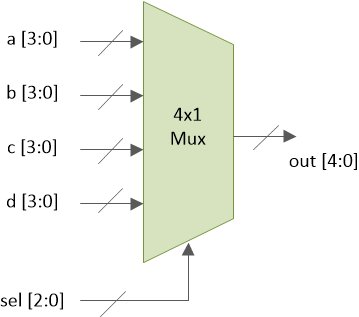


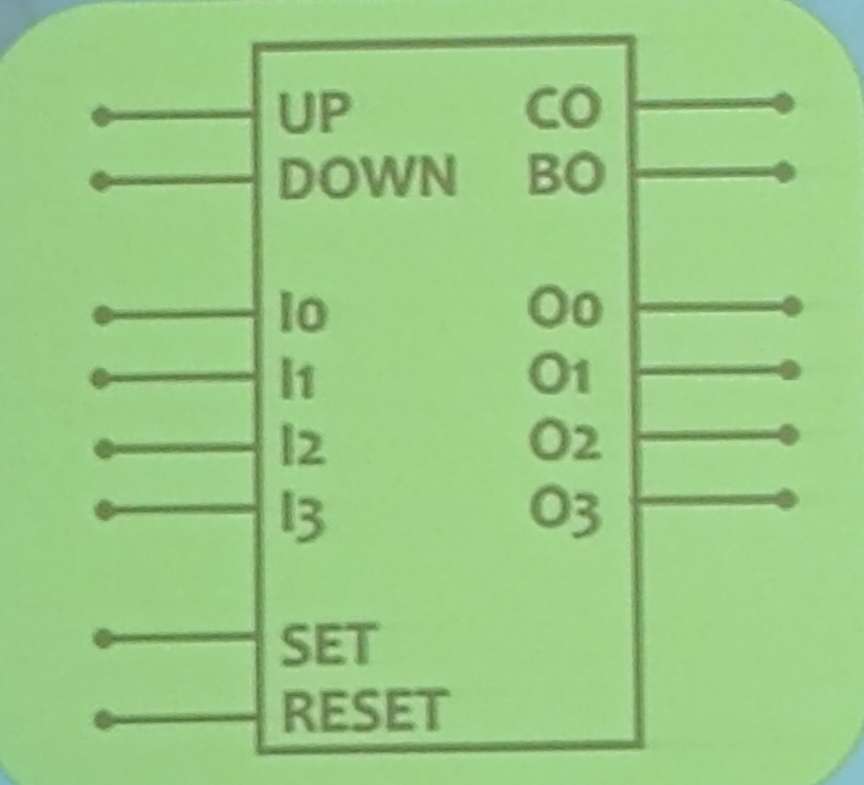
4 bites regiszter D tárolóval



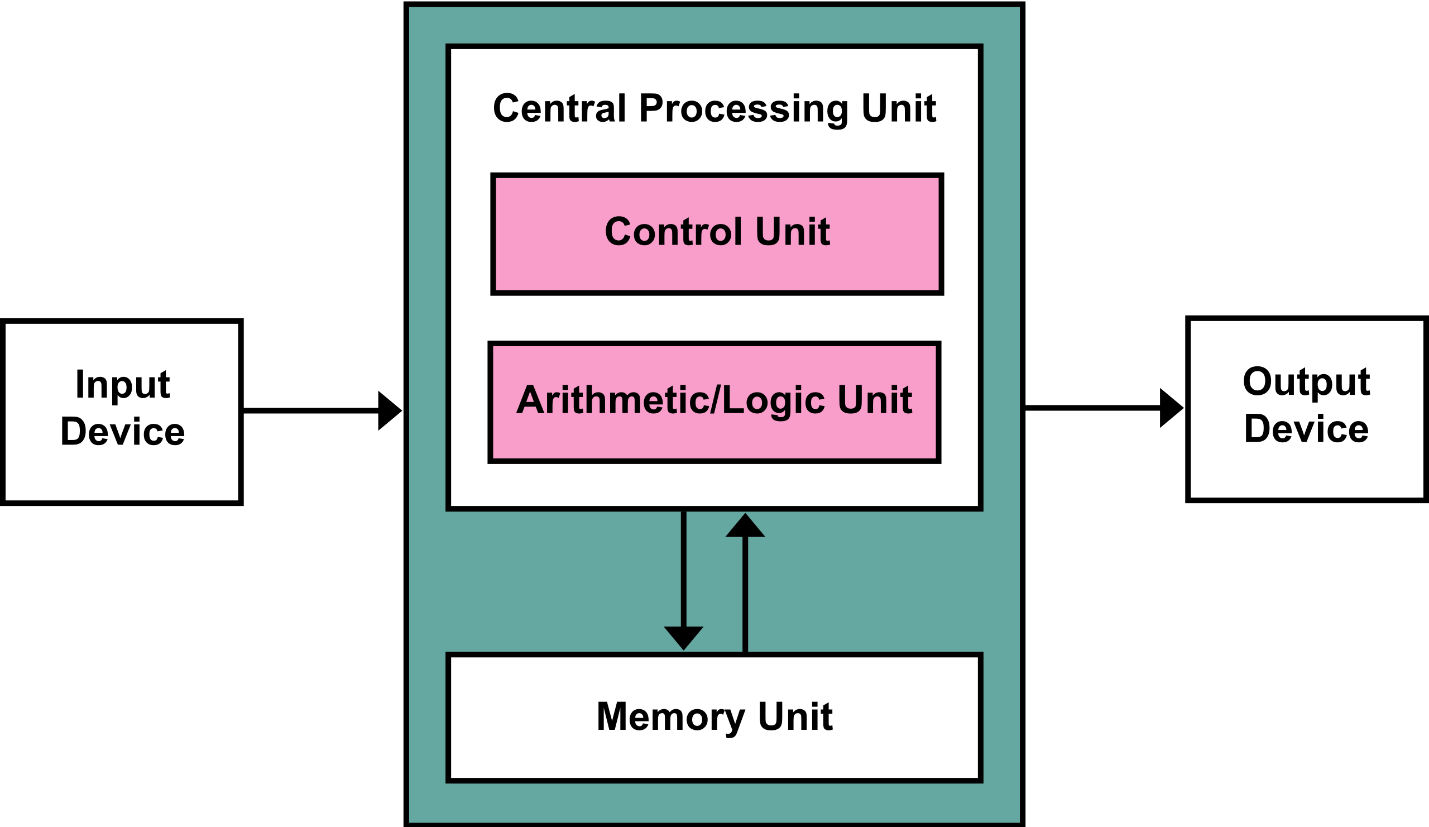
Multiplexer

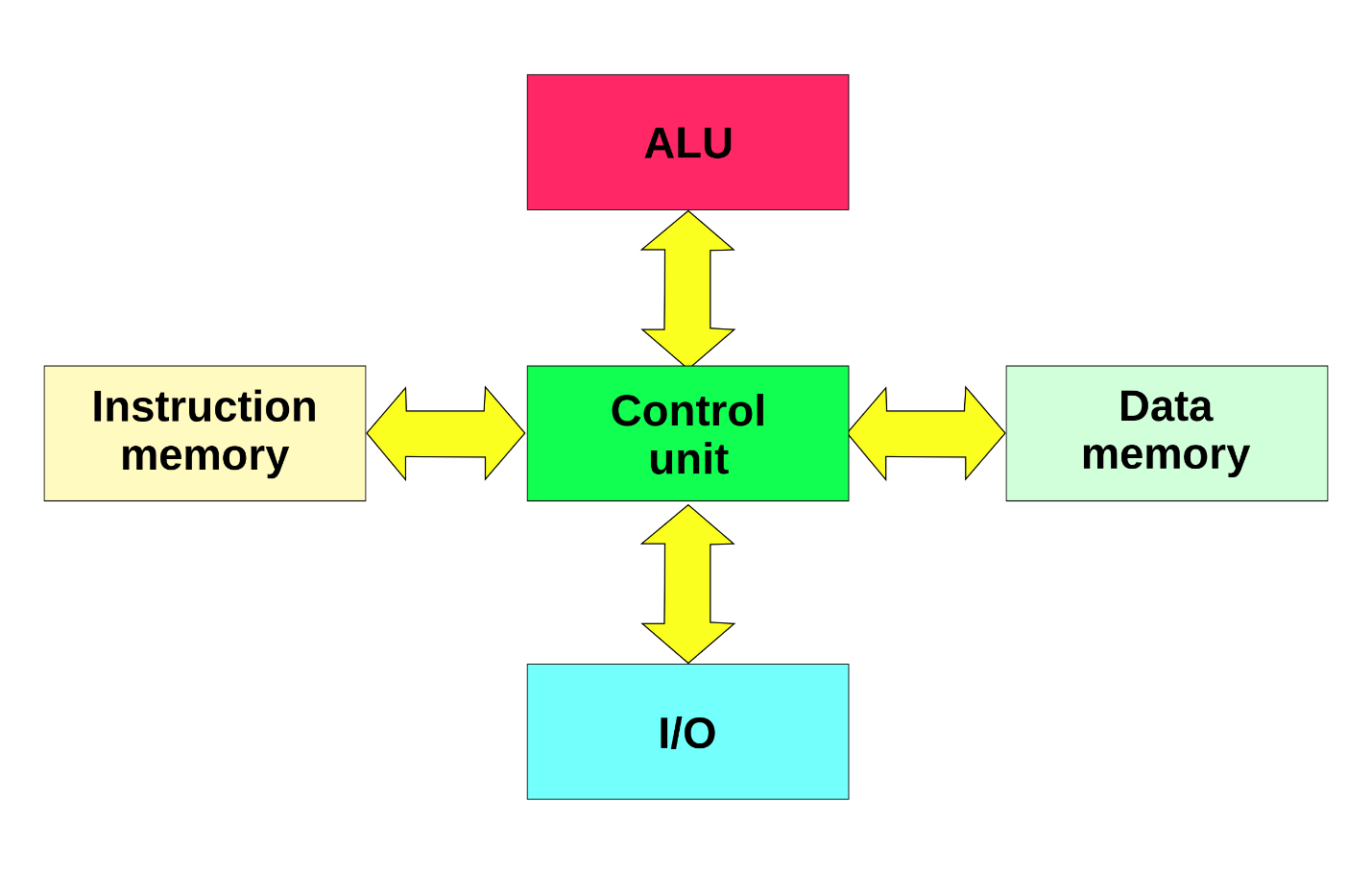
4 bites multiplexer

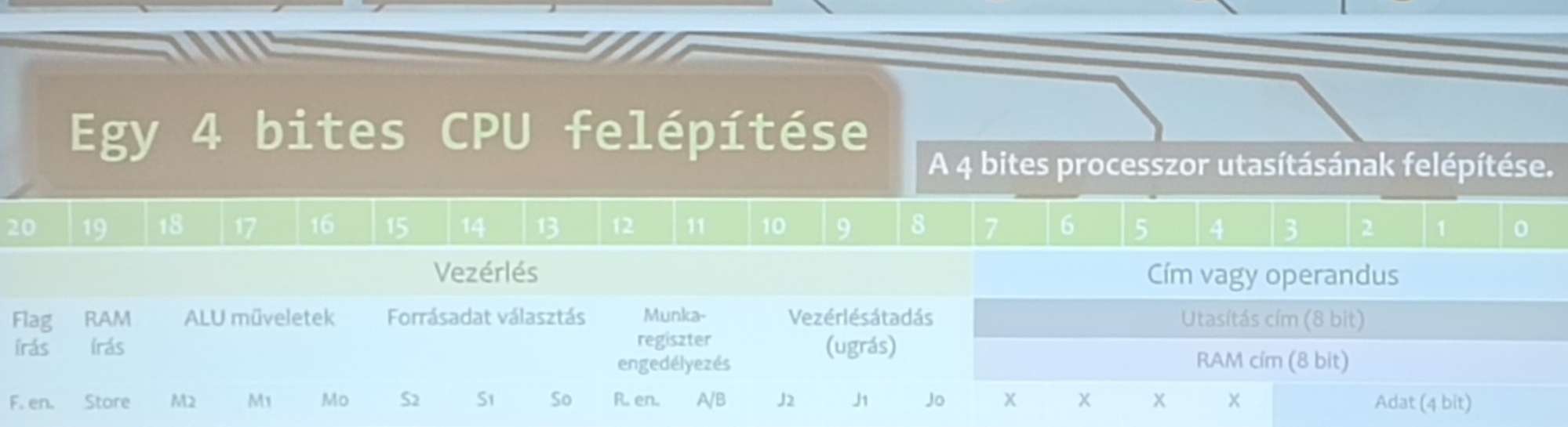


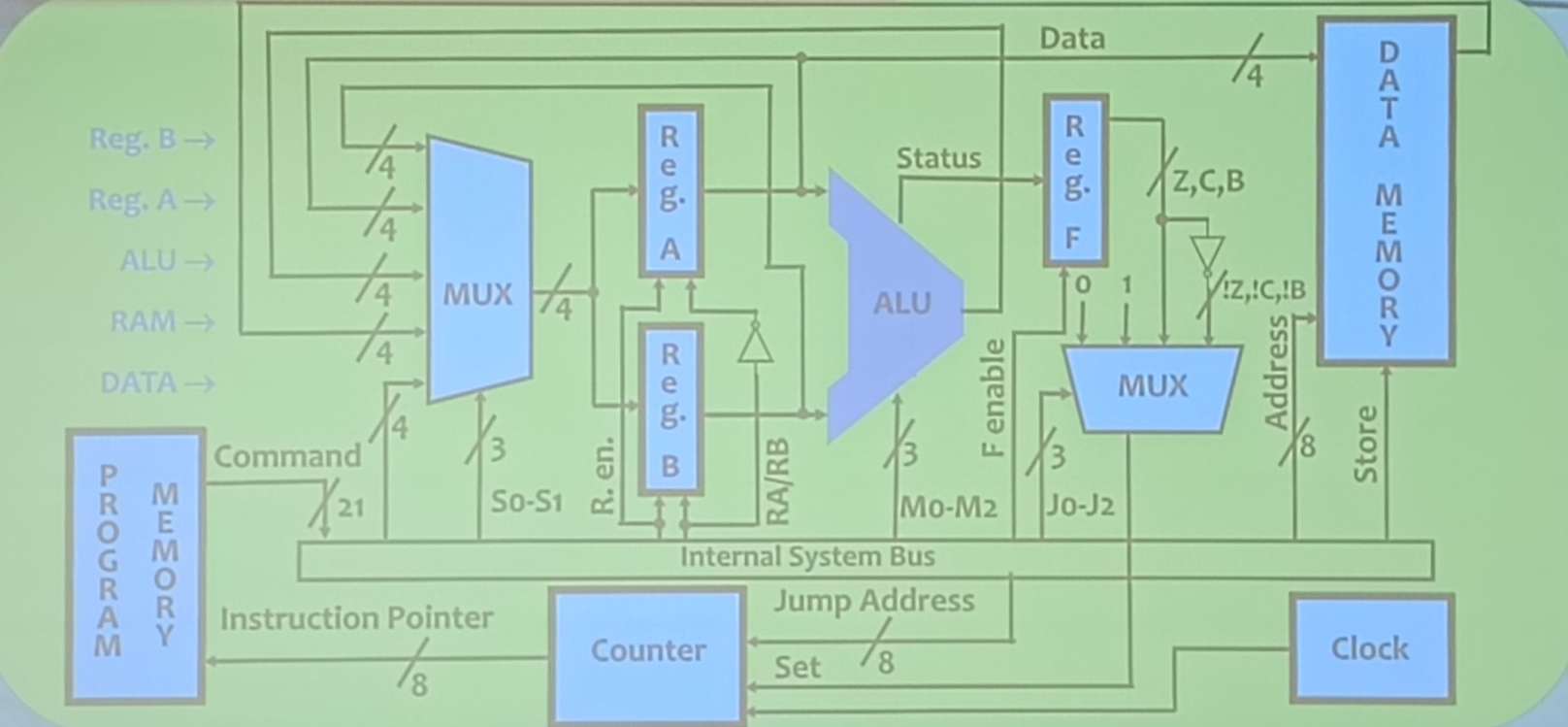
Számláló(frekvenciaosztó)

Neumann architektúra – bináris rendszer, 1 memória, ami tartalmazza a programot és az adatot, és módosítani tudja önmagát, végrehajtás szekvenciális



Harvard architektúra – megegyezik a Neumann architektúrával, kivéve, hogy külön van a program- és az adatmemória, ezáltal nem önmódosító.





## CISC – Complex Instruction Set Computing:

## Komplex utasítások (lassabb végrehajtás, bonyolultabb felépítés, nagyobb fogyasztás)

## Utasítások végrehajtása több órajel alatt

## Kevesebb általános regiszter, mert a műveletek a memóriában hajtódnak végre(memóriahatékony)

## Komplex címzés

## Több adattípus

## Olcsóbb

## Változó utasításméret

## Hardver orientált

## Kevesebb CPU ciklus utasításonként, kevesebb utasítás programonként

## RISC - Reduced Instruction Set Computing:

## Egyszerű utasítások (több utasítás CISC-hez képest a komplexebb utasításokhoz, egyszerű felépítés, kis fogyasztás)

## Egy órajel alatt elvégzi a műveleteket(gyorsabb)

## Általános regiszterek

## Kevesebb adattípus

## Egyszerű címzés

## Futószalagos

## Drágább

## Fix utasításméret

## Szoftver orientált

## Minimális utasítás programonként, de több CPU ciklus

## Mindkettő a CPU teljesítményt igyekszik növelni

## 

Időmultiplexelés – Egy vezetéken több állapot megjelenítése – Ketté osztjuk a vezetéket HIGH és LOW-ra, HIGH a felső 8 bit, LOW az alsó 8 bit, TPU vezetékkel értelmezzük melyik érvényes, időben választjuk szét.