Járműinformatika
Beágyazott rendszerek

2021/2022. tanév, II. félév

Dr. Kovács Szilveszter

E-mail: szkovacs@iit.uni-miskolc.hu

Informatika Intézet 107/a.

Tel: (46) 565-111 / 21-07

Ajánlott és felhasznált irodalom

Az előadás anyag forrása:

- Pannon Egyetem, Mérnöki Kar, Gépészmérnöki Intézet Járműmechatronika Intézeti Tanszék
- **Dr. Fodor Dénes, Speiser Ferenc: Autóipari beágyazott rendszerek**
- http://moodle.autolab.uni-pannon.hu/Mecha_tananyag/autoipari_beagyazott_rendszerek/index.html

Dr. Fodor Dénes intézetigazgató, tanszékvezető egyetemi docens
szoba: C218
telefon: 88/624-000/6082, 88/624-776
e-mail: fodor@almos.uni-pannon.hu

Speiser Ferenc egyetemi tanársegéd
szoba: C217
telefon: 88/624-000/6083
e-mail: speiserf@almos.uni-pannon.hu

Beágyazott rendszerek

- **Tartalom**
  - **Mit nevezünk beágyazott rendszernek?**
  - **Központi vezérlőegységek típusai**
  - **Mikrokontrollerek alapvető felépítése**
  - **Mikrokontrollerek perifériái**
  - **Mikrokontrollerek memóriái**
  - **Mikrokontroller architektúrák**
  - **Utasításkészletek**

Beágyazott rendszerek

- **Tartalom**
  - **Mit nevezünk beágyazott rendszernek?**
  - **Központi vezérlőegységek típusai**
  - **Mikrokontrollerek alapvető felépítése**
  - **Mikrokontrollerek perifériái**
  - **Mikrokontrollerek memóriái**
  - **Mikrokontroller architektúrák**
  - **Utasításkészletek**

Mit nevezünk beágyazott rendszernek?

- A beágyazott rendszereket számos eltérő felépítésben és eltérő célokra szoktak alkalmazni
- Egy adott feladatot ellátó kis számítógépet akkor neveznek beágyazott rendszernek (angolul: embedded system):
  - Cél specifikusan lett megtervezve
  - Adott jól ismert feladat megoldására illetve ellátására lett kialakítva
- Az általános célú számítógépekkel szemben:
  - Csupán néhány, előre meghatározott feladatot képes ellátni
  - Sokszor tartalmaz feladat-specifikus mechanikus és elektronikus alkatrészeket

Mit nevezünk beágyazott rendszernek?

- **Sokszor tartalmazhat olyan feladat-specifikus mechanikus és elektronikus alkatrészeket, melyek nem találhatók egy általános célú számítógépben**
- **A rendszer feladatai a tervezés idején is jól ismertek, a tervezők a feladatnak megfelelően tudják optimalizálni a rendszert, csökkenteni a költségét és méretét, növelni megbízhatóságát**
- **Ezen eszközök kialakítása és tulajdonságai a végrehajtandó feladat függvényében széles skálán mozoghatnak**

Mit nevezünk beágyazott rendszernek?

- A beágyazott rendszerek cél-specifikusak
- Szenzorokon, indikátorokon és aktuátorokon keresztül tartják a kapcsolatot a környezetükkel,
- Ha nem autonóm rendszerként működnek, akkor többnyire valamilyen szabványos kommunikációs interfésszel is rendelkeznek, például:
  - SPI
  - IIC (I^2C)
  - U(S)ART
  - CAN
  - FlexRay,
  - Ethernet,
  - WiFi
  - Bluetooth

Mit nevezünk beágyazott rendszernek?

- **Sokszor tévesen szokták állítani, hogy egy rendszer a központi egységtől lesz beágyazott rendszer**
- **Sokkal inkább a hardver – szoftver – felhasználás hármas határozza meg, hogy egy adott rendszer beágyazott rendszernek minősül-e**
- **Több felhasználási területen összemosódnak a határok az általános célú és beágyazott rendszerek között**
  - **Ipari kisméretű számítógép alkalmazható teljes értékű asztali számítógépként és beágyazott rendszerként is a műszaki környezetétől és a rajta futó programoktól függően**

Mit nevezünk beágyazott rendszernek?

- **Számtalan különböző felépítésű beágyazott rendszer létezik**
- **Teljesen eltérő architektúrával**
- **Mindegyikben közös, hogy a rendszer feladatai a tervezés idején is egyértelműen specifikálva vannak, így a tervezők a feladatnak megfelelően tudják optimalizálni a rendszert**
  - **Az adott feladathoz igazítható a rendszer mind hardver, mind szoftver szempontjából**
    - **Csökkenthetőek a költségek és méret**
    - **Növelni lehet a megbízhatóságot**

Mit nevezünk beágyazott rendszernek?

- **Tervezésük gyakran több fajta ismeret szintézisét igényli, hardveres ismeretek**
  - **Az alkalmazandó kommunikációs szabványok ismerete**
  - **Hardver közeli, beágyazott szoftverfejlesztés**
  - **PC-s eszközillesztők és magas szintű szoftver ismeretek (PC-s felhasználói felület fejlesztéséhez)**
  - **Intelligens algoritmusok ismerete nagy bonyolultságú feladatok megoldásához**
  - **Jelfeldolgozási ismeretek és egyéb alkalmazás-specifikus ismeretek (pl. képfeldolgozás, motorvezérlés)**

Beágyazott rendszerek

- **Tartalom**
  - Mit nevezünk beágyazott rendszernek?
  - **Központi vezérlőegységek típusai**
  - Mikrokontrollerek alapvető felépítése
  - Mikrokontrollerek perifériái
  - Mikrokontrollerek memóriái
  - Mikrokontroller architektúrák
  - Utasításkészletek

Központi vezérlőegységek típusai

- A központi vezérlőegységek a beágyazott rendszerek fő komponensei
- Egy adott rendszer több - akár eltérő típusú - vezérlőegységet is tartalmazhat, az ellátandó feladattól függően
- A speciális funkció sok esetben ellátható kis számítási teljesítménnyel is
  - jellemzően a rendszer fogyasztása és költségei egyaránt alacsonyak.
- A központi vezérlőegység lehet akár mikrokontroller vagy nagy komplexitású vezérlő logika (FPGA) illetve a hagyományos számítógépeknél alkalmazott processzor is

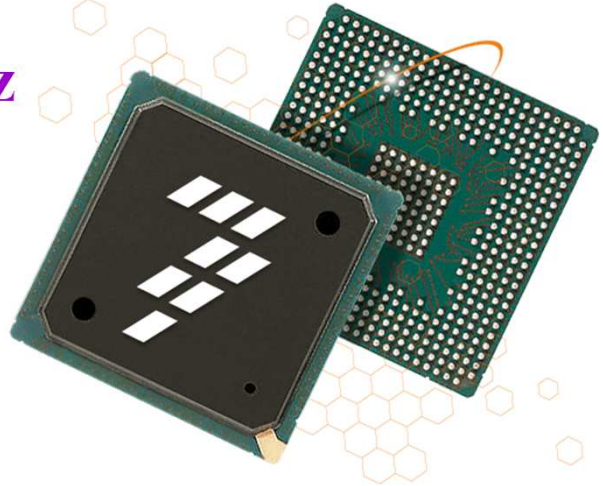
Központi vezérlőegységek típusai

- Nem mindegyik vezérlőegység igényli operációs rendszer meglétét (FPGA, CPLD)
- A kategóriák sokszor nem egyértelműek
- A leggyakrabban alkalmazott vezérlőegység típusok:
  - ASIC (Application Specific Integrated Circuits)
  - ASIP (Application-Specific Instruction-set Processor)
  - DSP (Digital Signal Processor)
  - CPLD (Complex Programmable Logic Device)
  - FPGA (Field Programmable Gate Array)
  - SoC (System On a Chip)
  - Mikrokontrollerek
  - GPU (Graphics Processing Unit)

ASIC (Application Specific Integrated Circuits)

- **Alkalmazás-specifikus integrált áramkör**
- **Mindig egy adott specifikus feladat ellátására tervezik, nem általános felhasználásra**
- **Megtervezése és a fejlesztői szériák legyártása nagy költségekkel járhat bonyolult feladatok megoldása esetén**
- **Csak egy adott célra lehet alkalmazni**
- **Nagy tételben történő felhasználás esetén költséghatékony, valamint olyan egyedi feladatoknál, amelyek más eszközzel nem oldhatóak meg**

ASIP (Application-Specific Instruction-set Processor) és DSP (Digital Signal Processor)



- Utasításkészletük egy adott célfeladathoz lett optimalizálva
- A célfeladat ellátásához legszükségesebb utasításokat tartalmazzák
- Ezek a speciális központi egységek sokszor nem önállóan, hanem társprocesszorként jelennek meg egy mikrokontroller vagy más központi egység mellett
  - gyakran azzal egy tokba integrálva
- Alkalmazás orientált processzorok, ASIP (Application-Specific Instruction-set Processor)
- Digitális jelfeldolgozó processzorok DSP (Digital Signal Processor)

ASIP és DSP

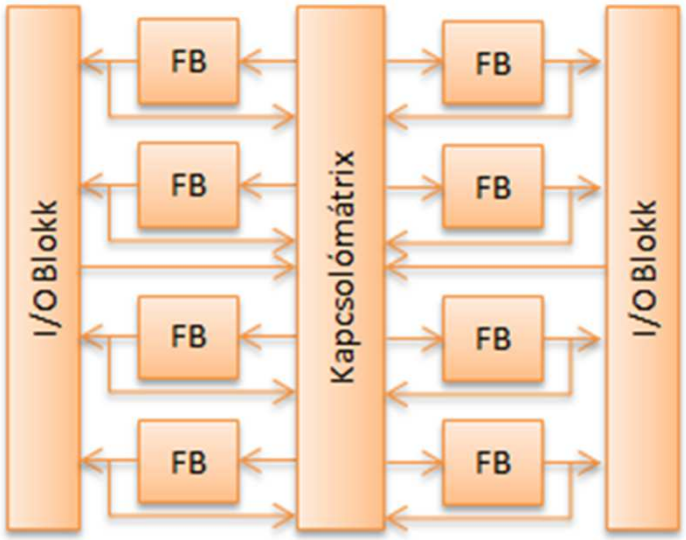
- **Digitális jelfeldolgozó processzorok DSP (Digital Signal Processor)**
  - **A jelfeldolgozáshoz optimalizált processzorok**
  - **Harvard-architektúrát alkalmaznak**
  - **Gyors, speciális feladatú hardver szorzó-akkumulátor modullal rendelkeznek (MAC egységek (Multiply-Accumulate unit))**
    - **Több műveletet egy lépésben képesek elvégezni (pl.: a ← a + (b x c))**
    - **Egy órajel alatt több memóriacím elérésre is képesek**
  - **Processzorral egybeintegrált gyors memória és/vagy gyorsítótár**

CPLD (Complex Programmable Logic Device)

- A CPLD vagy más néven összetett programozható logikai áramkör
- Lényegében több, egyszerű programozható logikai egység egybeintegrálása
- Az egységek kimenetei és bemenetei összekapcsolhatóak egymással
- A programozható logikai egységek (PLD) lényegében olyan logikai kapuk, flip-flop-ok stb. együttese, melyet a felhasználó tud konfigurálni
  - Létrehozhatóak különböző logikai kapcsolások
- A logikai egységeket a CPLD-k esetében általában makrocelláknak hívják
  - A funkcionális blokkokon belül helyezkednek el (többnyire 4-16 ilyen egység található egy funkcionális blokkban)

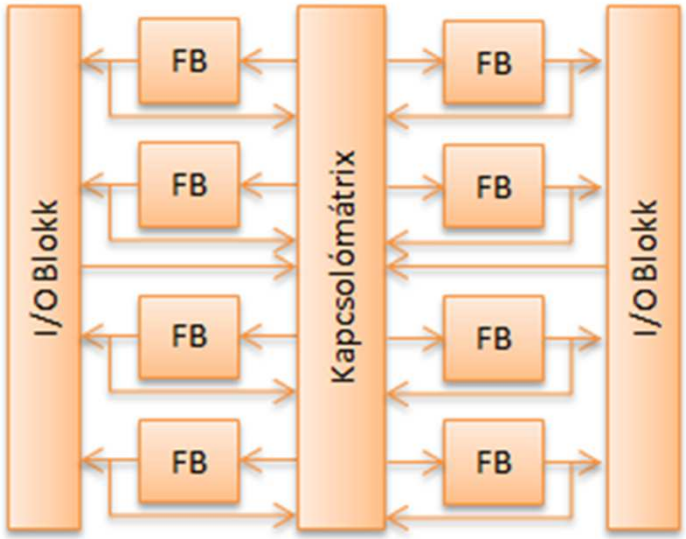
CPLD (Complex Programmable Logic Device)

- A logikai hálózat a funkció blokkok (FB) programozásával hozható létre
- Egy kapcsolómátrix segítségével lehet összekötni a funkció blokkok ki- és bemeneteit, valamint a tokozás ki- és bemeneteit, azaz az I/O blokkokat



CPLD (Complex Programmable Logic Device)

- **A CPLD-k előnyös tulajdonságai**
  - **Az áramkör a nyomtatott áramköri panelra történő beültetés után is programozható, illetve újraprogramozható**
  - **A kapcsolatok és a makrocellák konfigurációját, azaz jelen esetben a *programot* flash típusú memóriában tárolja**
    - **A kikapcsolás után is megőrzi tartalmát.**

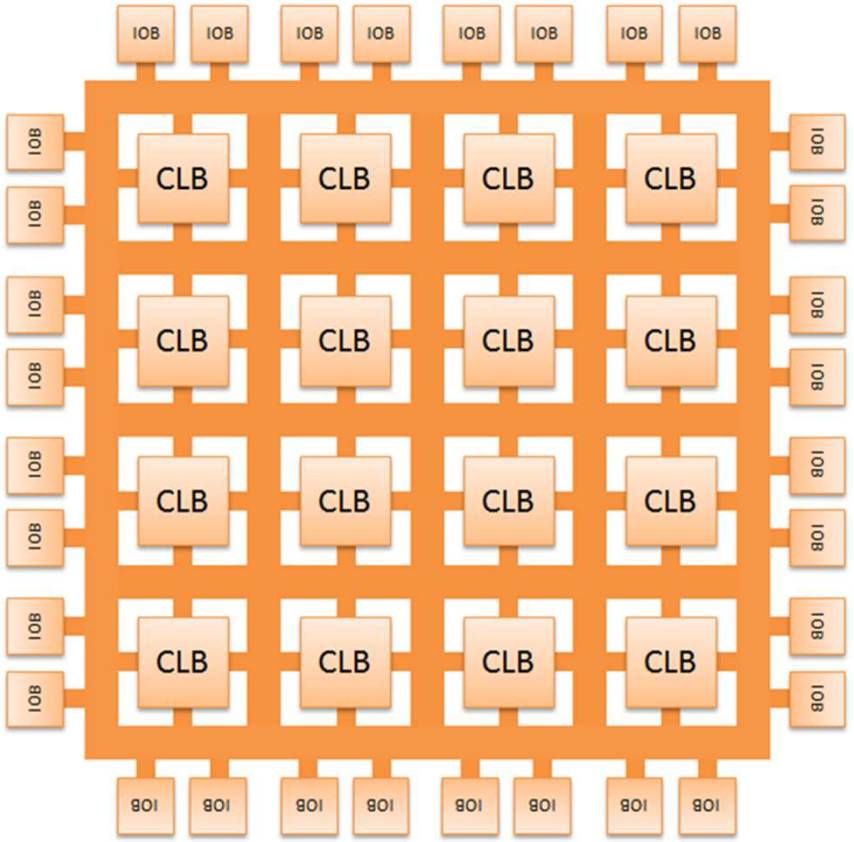


FPGA (Field Programmable Gate Array)

- Az FPGA vagy más néven programozható kapu mezők
- A felhasználó által programozható kapu-áramkörök
- Logikai hálózat kialakításánál a konfigurálható logikai blokkokat (CLB) és az ezek közötti összeköttetéseket kell programozni
- Ezek a blokkok belső huzalozási utak felhasználásával tetszőlegesen összeköthetők egymással
  - Lényegében a konfigurálható logikai blokkok valósítják meg a felhasználónak szükséges logikai kapcsolatokat
- A programozható ki- és bemeneti blokkok, azaz az IOB-k teremtik meg a kapcsolatot a tokozás kivezetései és a belső logikai kapcsolás között
  - Általában mindegyik IOB definiálható bemenet vagy kimenet illetve kétirányú csatlakozásként is

FPGA (Field Programmable Gate Array)

- A programozható kötések segítségével egymáshoz kapcsolhatóak a konfigurálható logikai blokkok valamint a ki- és bemeneti blokkok kivezetései
  - Az összeköttetések állapotait egy konfigurációs memória tárolja.
- Az FPGA logikai erőforrásaiból eredő párhuzamosság jelentős számítási teljesítményt tesz lehetővé
- Jellemzője a hosszú fordítási és optimalizálási idő



FPGA (Field Programmable Gate Array)

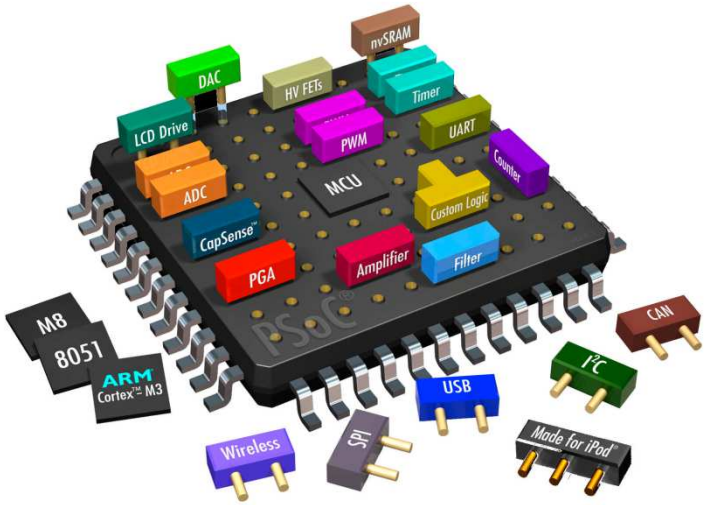
- **Sokszor össze szokták keverni a CPLD és az FPGA alapú központi egységeket**
  - **Többnyire ezek architektúrája valamint felhasználási körük is jelentősen eltér**
    - **Az egyszerűbb, gyorsabb válaszidőt igénylő feladatok esetén többnyire CPLD-t alkalmaznak**
    - **Az összetettebb, több feladatszálat futtató feladatok esetén FPGA-t szoktak alkalmazni**

FPGA (Field Programmable Gate Array)

- Általánosságban elmondható, kicsit egyszerűsítve a dolgot:
  - Az FPGA adott számú kaput tartalmaz, amiket rugalmasan lehet felhasználni adott blokkok kialakítására
  - A CPLD esetében ezen blokkok fixek (ezek a makrocellák), ezeken belül lehet logikai kapcsolásokat létrehozni, majd a makrocellákat lehet összekötni
  - Az FPGA SRAM alapú
    - Kikapcsolás után újra fel kell rá tölteni a programot egy külső, nem felejtő memóriából
  - A CPLD többnyire flash alapú memóriában tárolja a programot, így az kikapcsolás után is megőrződik
    - A felhasználás területén ökölszabályként elmondható, hogy az egyszerűbb, gyorsabb válaszidőt igénylő feladatok esetén többnyire CPLD-t, míg az összetettebb, több feladatszálat futtató feladatok esetén FPGA-t szoktak alkalmazni.

SoC (System On a Chip)

- **Olyan integrált áramkörök, melyeknek részét képezik:**
  - **A perifériakezelő rendszerek**
  - **A CPU-mag(ok)**
  - **Jellemzően az integrált grafikus vezérlő**
  - **Definíció szerint ezen rendszereket csak egy hajszál választja el a mikrokontrollerektől**
    - **Jellemzőjük a nagyobb teljesítmény, nagyobb memória méret, stb.**
      - **Gyakran x86 (x64) PowerPC vagy ARM architektúrájú rendszerek**
      - **Nagyobb teljesítményűek és általánosabb felhasználásúak, mint a mikrokontrollerek**

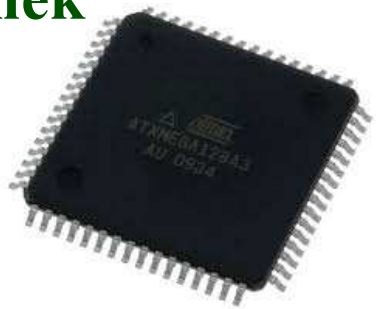


Mikrokontrollerek

- A mikrokontrollerek egychipes áramkörök
- Egy mikroszámítógép konfiguráció minden elemét tartalmazza
  - CPU
  - memória (RAM, ROM)
  - I/O egységek
  - rendszer órajel generátor
  - stb.
- Túlnyomórészt Harvard architektúrát és csökkentett utasításkészletet tartalmaznak (Reduced Instruction Set Computing, RISC)
- A külső memóriával általában nem vagy csak az I/O vonalak felhasználásával bővíthetőek

Mikrokontrollerek

- Általában jellegzetes erőforrásokkal rendelkeznek
  - Watchdog timer
  - Külső és belső megszakítás (interrupt) vonalak
  - Számláló/időzítő áramkörök,
  - Digitális és analóg be- és kimeneti vonalak



- A kivezetések számának csökkentése céljából többcélú kivezetéseket használnak
  - Egy kivezetéshez több funkció van rendelve
- Általában többféle hardveresen integrált kommunikációs interfész modult tartalmaznak, például:
  - UART
  - SPI
  - CAN
  - Ethernet, stb.

GPU (Graphics Processing Unit)

- A grafikai számítások elvégzésére optimalizált központi vezérlőegység
- Eredeti feladata a CPU tehermentesítése volt a grafikai számítások elvégzése alól
- A grafikai számítások eltérő igényekkel rendelkeznek, mint a központi egységgel támasztott követelmények
- A GPU speciális célokat szolgál, míg a CPU általános feladatokat lát el
- Manapság már nem csak grafikai számítások elvégzésére alkalmazzák a GPU-kat, mert felépítésük több esetben is előnyösebb
  - Gyorsabb program végrehajtást tesznek lehetővé bizonyos esetekben

GPU (Graphics Processing Unit)

- Lényeges a dedikált memória (vagy a rendszermemória közvetlen elérése) és annak mennyisége és sebessége, illetve a sávszélesség a GPU és a memória között
- Futószalag elv (pipeline)
  - A problémát részfeladatokra kell bontani
  - Minden feladatra egy dedikált eszközt alkalmazni
  - Az eszközök egymással párhuzamosan végzik a feldolgozást
  - A dedikált eszközök sora egymás mellé redundánsan lemásolható
- Korábban a futószalag elvet a GPU-k hardveres szinten is követték azonban manapság már általánosabb architektúrával készülnek
  - A futószalag sokszor már inkább csak logikailag, szoftveres absztrakcióként létezik

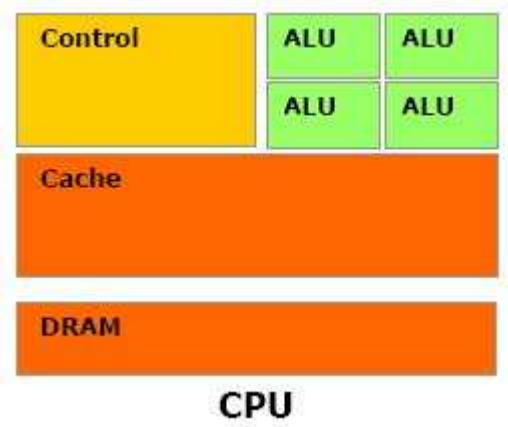
GPU (Graphics Processing Unit)

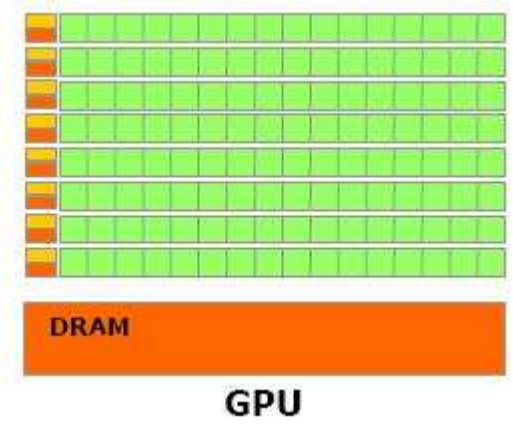
- **Manapság már nem csak grafikai számítások elvégzésére alkalmazzák a GPU-kat, mert felépítésük több esetben is előnyösebb**
  - **Gyorsabb program végrehajtást tesznek lehetővé bizonyos esetekben**
- **A CPU erőforrásai nagy részét a programok vezérlésére a gyorsabb utasítás kiválasztásra fordítja**
  - **A GPU erre kevésbé alkalmas**
- **Aritmetikai logikai egységekkel (ALU) a GPU van jobban ellátva**

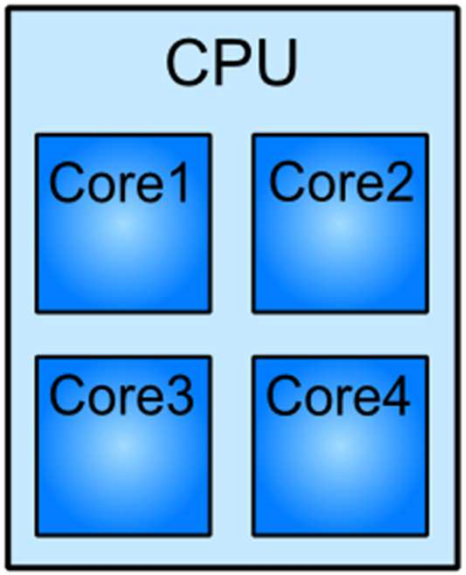
GPU (Graphics Processing Unit)

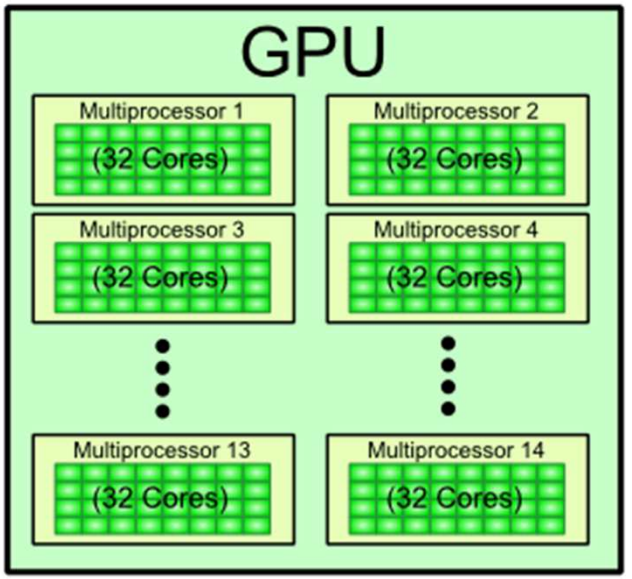
- **Egy program GPU-ra való portolása előtt mindig azt kell elsőként megvizsgálni, hogy az adott feladat mennyire adat és számolás intenzív**
- **Ha a vezérlés átadó utasítások száma elenyésző, és nagy tömegű adaton kell elvégezni ugyanazt a számítást, akkor a portolás valószínűleg eredményes lesz**

GPU (Graphics Processing Unit)







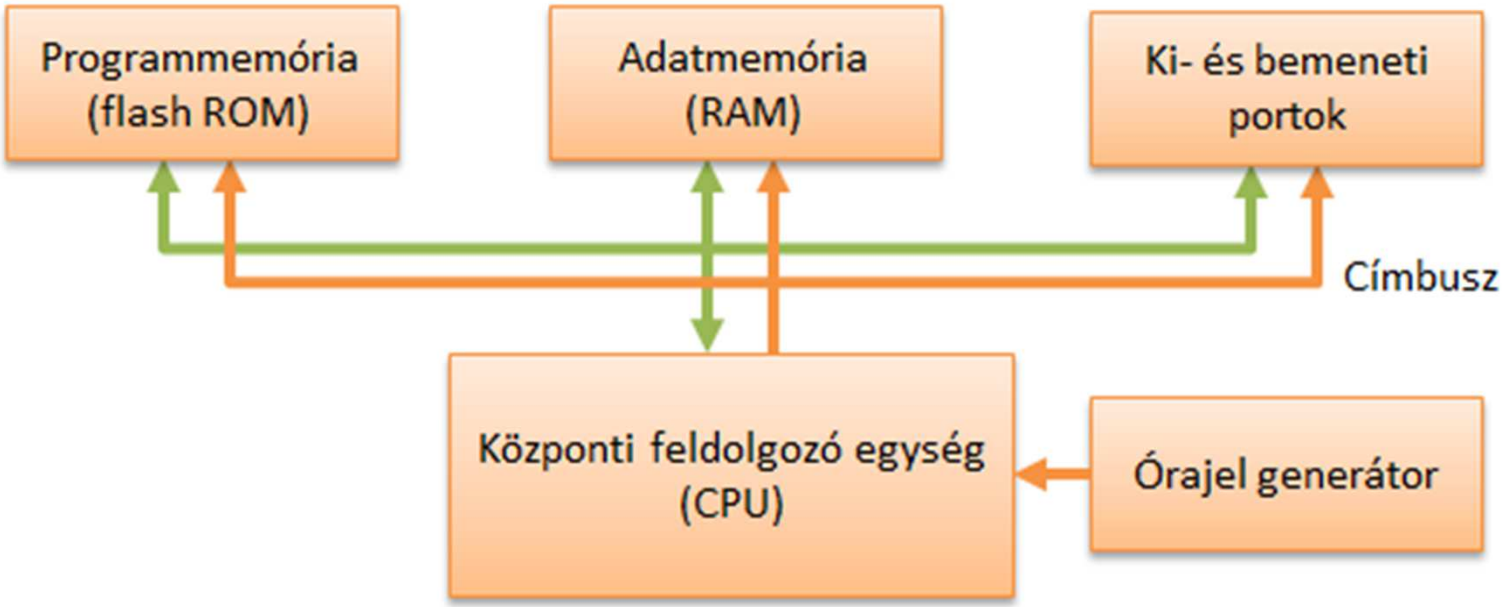


Beágyazott rendszerek

- **Tartalom**
  - Mit nevezünk beágyazott rendszernek?
  - Központi vezérlőegységek típusai
  - **Mikrokontrollerek alapvető felépítése**
  - Mikrokontrollerek perifériái
  - Mikrokontrollerek memóriái
  - Mikrokontroller architektúrák
  - Utasításkészletek

Mikrokontrollerek alapvető felépítése

- **A beágyazott rendszerekben a legelterjedtebb vezérlőegység típus a mikrokontroller**
  - **Érdemes jobban megismerni a központi vezérlő és a hozzá tartozó kiegészítő elemek főbb típusait**

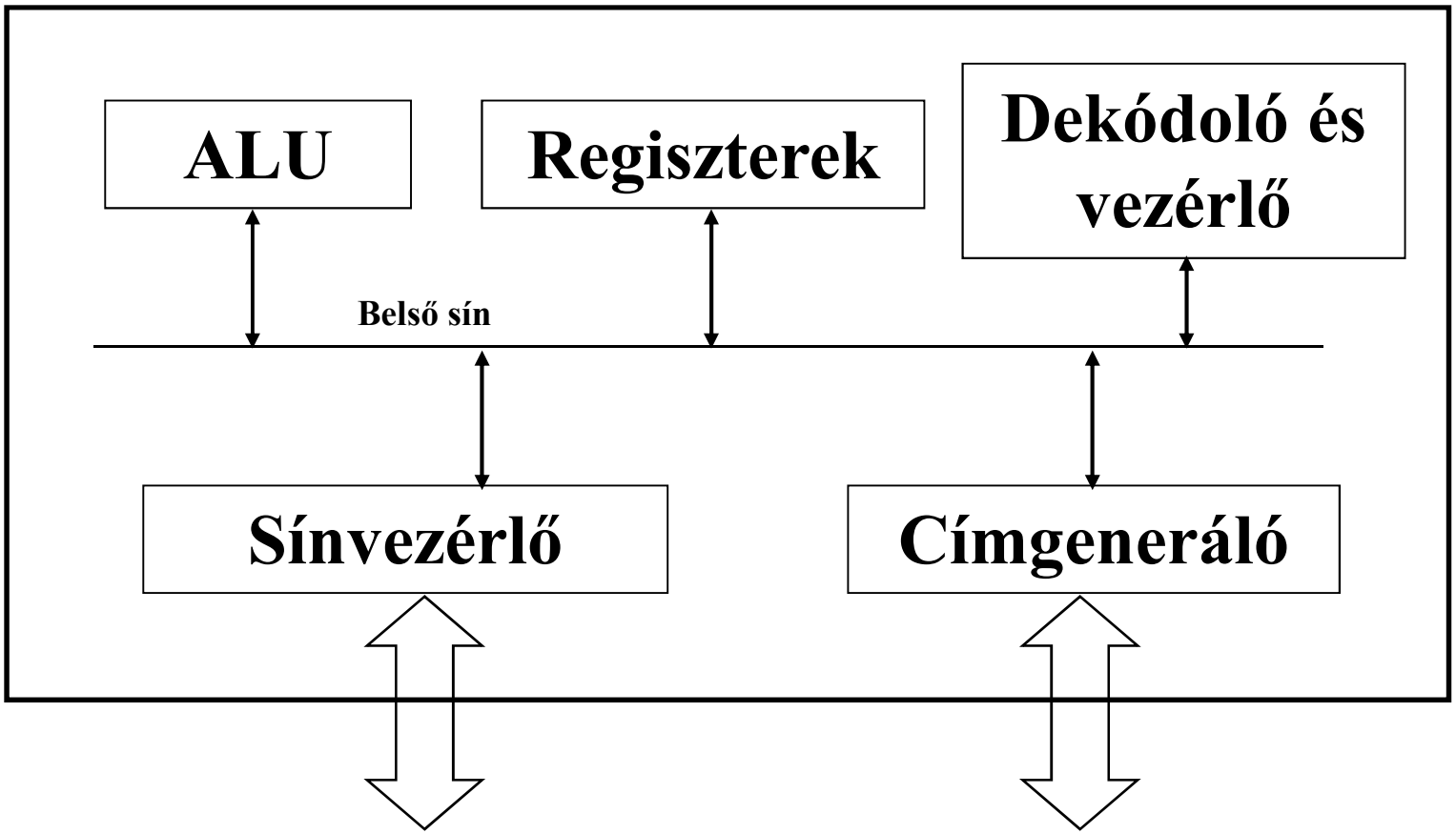


Mikrokontrollerek alapvető felépítése

- **Központi feldolgozó egység (CPU)**
  - **A processzor működésének vezérlését és a feladatok végrehajtásának ütemezését a *vezérlő és dekódoló egység* (Control Unit) végzi**
  - **Az *aritmetikai-logikai egység* (ALU) felelős a számítások illetve műveletek végrehajtásáért.**
    - **Gyakran ki szokta egészíteni egy FPU, azaz lebegőpontos egység, mely a lebegőpontos számokon történő számítások elvégzését hivatott felgyorsítani**
  - **Alapvető *regiszterek* a műveletek végrehajtásához, például:**
    - **Programszámláló**
    - **Verem mutató**
    - **Státusz regiszter**
  - **Az átmeneti eredmények tárolására szolgáló, igen gyors elérésű regiszterek**
  - **Egy kontroller több magot is tartalmazhat, melyek önállóan, egymással párhuzamosan képesek utasításokat végrehajtani**
    - **A műveletek párhuzamosításánál van jelentősége**

Mikrokontrollerek alapvető felépítése

- **Központi feldolgozó egység**



Mikrokontrollerek alapvető felépítése

- **A végrehajtandó programot tároló memória a *programmemória***
  - **Nem felejtő**
  - **Alapesetben csak olvasható (read-only memory (ROM))**
  - **Kikapcsolás után is megőrzi a tartalmát**
  - **A mikrokontroller hagyományos működése közben nem íródhat felül**
- **Az adatok tárolására szolgáló memória az *adatmemória***
  - **Tetszőleges hozzáférésű memória (random access memory (RAM))**
  - **Működés közben bármely valós memóriacímén írható és olvasható**

Mikrokontrollerek alapvető felépítése

- *Vezérlőbusz*
  - Ezen keresztül utasítja a központi egység a többi elemet a megfelelő működésre
  - kétirányú
- *Címbusz*
  - A memória (vagy a periféria) megfelelő tároló rekeszét címzi
  - Egyirányú
- *Adatbusz*
  - Ezen mozognak a különféle adatok
  - Kétirányú

Mikrokontrollerek alapvető felépítése

- **Órajel generátor**
  - Azért felel, hogy a mikrokontroller összes komponense összhangban legyen
    - A processzor részegységei az órajel ütemére végzik feladataikat
      - Amikor egy részegység megkapja az órajelet egy elektronikus jel formájában, akkor elvégzi a soron következő műveletet
      - A műveletet nem szabad összetéveszteni az utasítással, egy utasítás végrehajtása több órajel ciklust is igénybe vehet.
  - Az órajel származhat:
    - A mikrokontrollerbe integrált órajel generátortól
    - Egy külső órajel generátortól
  - Az órajel fontos jellemzője a processzornak, de nem jellemzi egyértelműen a teljesítményét
    - Sok processzor egy órajel alatt több műveletet is el tud végezni
      - IPC (Instructions Per Cycle) érték: az egy órajel ciklus alatt elvégzett műveletek száma

Mikrokontrollerek alapvető felépítése

- **Be- és kimeneti portok (más néven a periféria (I/O) egység)**
  - **A külvilággal történő kommunikációra szolgálnak**

Beágyazott rendszerek

- **Tartalom**
  - Mit nevezünk beágyazott rendszernek?
  - Központi vezérlőegységek típusai
  - Mikrokontrollerek alapvető felépítése
  - **Mikrokontrollerek perifériái**
  - Mikrokontrollerek memóriái
  - Mikrokontroller architektúrák
  - Utasításkészletek

Mikrokontrollerek perifériái

- A perifériák terén komoly eltérések lehetnek a különböző mikrokontrollerek között
- A perifériák kezeléséért többnyire különálló komponensek felelnek, melyeket kontrollerbe integrálnak
  - Kialakításuk és szerepük eltérő lehet
- A perifériák kezelése többnyire meghatározott memóriaterületek írásával és olvasásával történik (Special Function Register, SFR)
  - Az alapvető felépítésben nem igényel különösebb változtatásokat
- Vannak olyan perifériák, amelyek a kontrollerek többségében megtalálhatóak:
  - Beépített időzítő
  - Analóg-digitális átalakító (ADC)
  - Stb.

Mikrokontrollerek perifériái

- A beépített időzítő
  - többnyire legalább egy megtalálható a mikrokontrollerekben Számos feladata lehet:
    - Használható időmérésre
    - Különböző feladatok ütemezésére
- Biztonsági időzítő áramkör, azaz más néven WatchDog Timer (WDT)
  - Feladata, hogy újraindítsa a kontrollert abban az esetben, ha az végtelen ciklusba kerülne valamely műveletnél
- A valósidejű óra (RealTime Clock, RTC) generátor
  - Feladata a hosszútávon történő pontos időmérés
  - Gyakran külső elemes vagy akkumulátoros tápellátás is szükséges a működtetéséhez
    - Abban az esetben is tudja mérni az időt, amikor a mikrokontroller kikapcsolt állapotban van

Mikrokontrollerek perifériái

- **Analóg-digitális átalakító (ADC)**
  - **Feladata, hogy a beérkező analóg jelet mintavételezés és kvantálás után, a központi feldolgozó egység által értelmezhető digitális formára alakítsa**
- **Digitális-analóg átalakító (DAC)**
  - **Feladata, hogy a digitális jeleket analóg jellé alakítsa**
- **Kommunikációs interfészek**
  - **A külvilággal illetve más mikrokontrollerekkel történő kommunikációt tesznek lehetővé**
  - **Ilyen lehet például:**
    - **UART**
    - **I2C**
    - **SPI**
    - **stb.**

Mikrokontrollerek perifériái

- Adatok tárolására szolgáló, nem felejtő memóriaterület
  - Például azonosítók, hálózati címek tárolására
- A fejlesztés során alkalmazott komponensek
  - Fő feladatuk:
    - A fejlesztési stádiumban elősegítsék a mikrokontroller felprogramozását és
    - a programban történő hibakeresést (debugger)

Beágyazott rendszerek

- **Tartalom**
  - Mit nevezünk beágyazott rendszernek?
  - Központi vezérlőegységek típusai
  - Mikrokontrollerek alapvető felépítése
  - Mikrokontrollerek perifériái
  - **Mikrokontrollerek memóriái**
  - Mikrokontroller architektúrák
  - Utasításkészletek

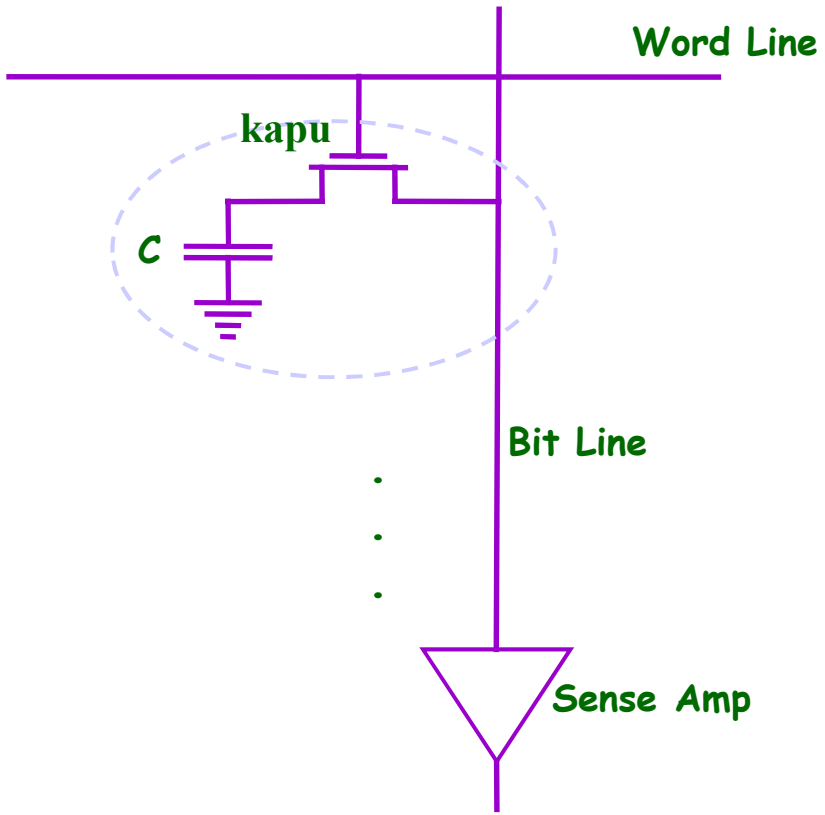
Mikrokontrollerek memóriái

- A memóriáknak alapvetően két fajtája van
  - Az egyik a CPU-ba került integrálásra, melyet *regiszter*nek hívnak
  - A másik, amit a CPU buszrendszeren keresztül ér el, hagyományosan memóriának (operatív tárnak) szokás hívni
- A memória lehet alapesetben:
  - Csak olvasható (ROM)
  - Írható és olvasható is (RAM)
- Ezen felül egy gyakori csoportosítás még:
  - Felejtő memóriák
  - Nem felejtő memóriák

Felejtő memóriák

- A felejtő memóriákat gyakran félrevezetően RAM-nak szokták nevezni
- A felejtő memória lényege:
  - Írni és olvasni is lehet
  - Ha megszűnik a kontroller tápellátása, akkor elveszíti a tartalmát
- A kontrollerek többsége kevés felejtő memóriát szokott tartalmazni
  - Nagy helyigény
  - Magas ár
- Nem a számítógépeknél alkalmazott dinamikus RAM-ot (DRAM), hanem statikus RAM-ot (SRAM) szoktak használni
  - Csak akkor kell frissíteni a tartalmát, amikor változik
  - Nem igényel folyamatos újraírást, mint ahogy az a dinamikus RAM esetében történik

Dinamikus RAM (DRAM) egy cellája



Írás:

Állíts a bit vonalra magas v. alacsony szintet (a beírni kívánt bit szerint);

Nyisd a kaput: megfelelően feltöltődik a C

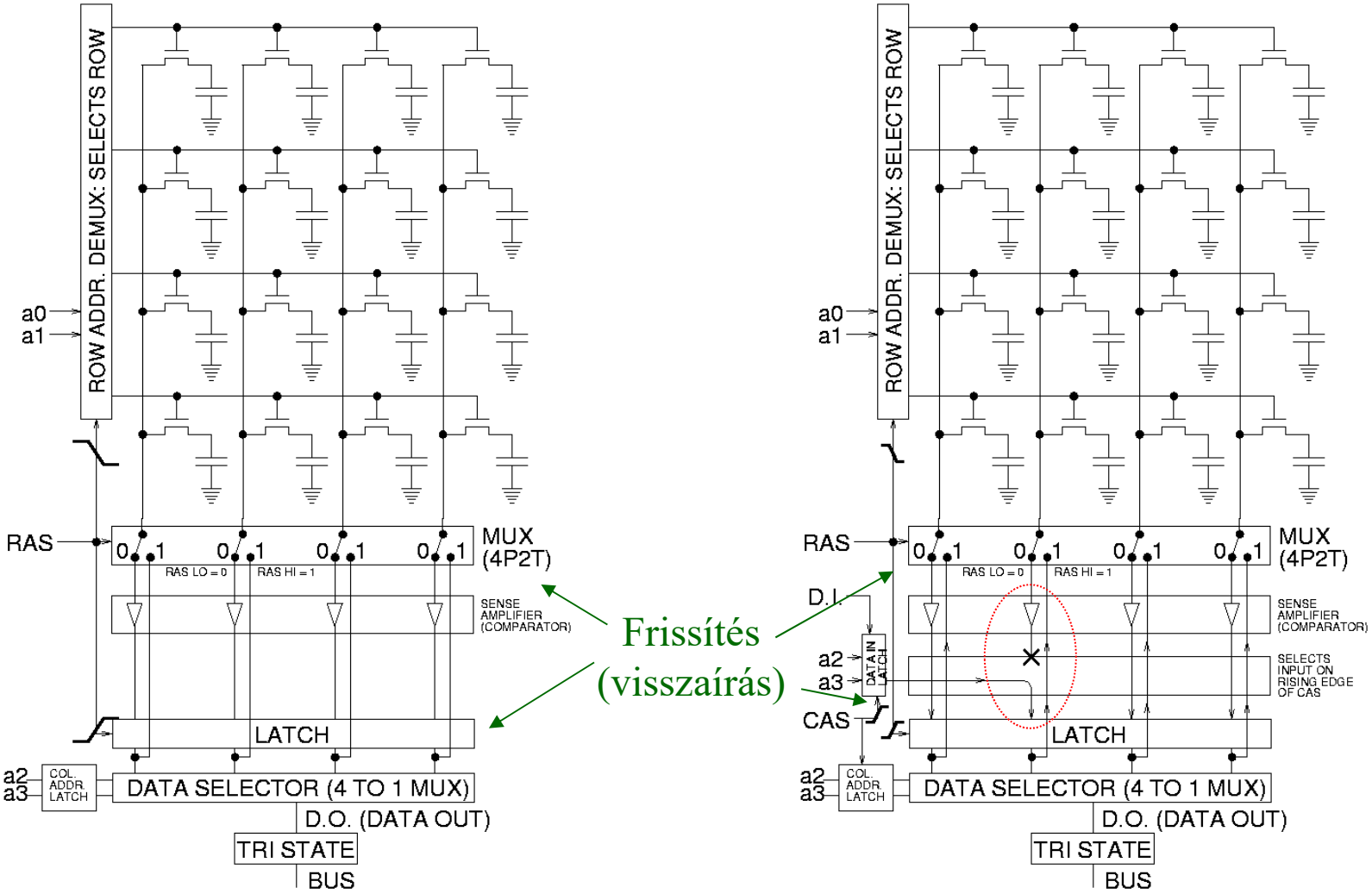
Olvasás:

Állítsd a bit vonalat „fele” feszültségre;

Nyisd a kaput;

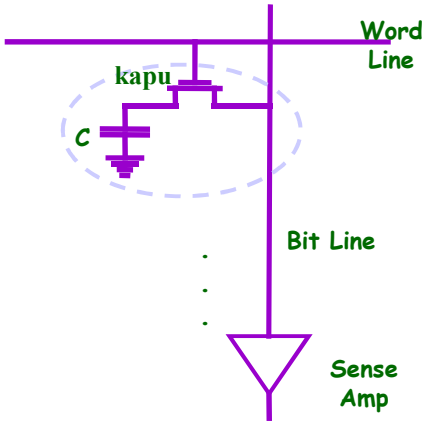
A C szintjétől függően a bit vonal feszültsége elmozdul, amit a Sense Amp érzékel.

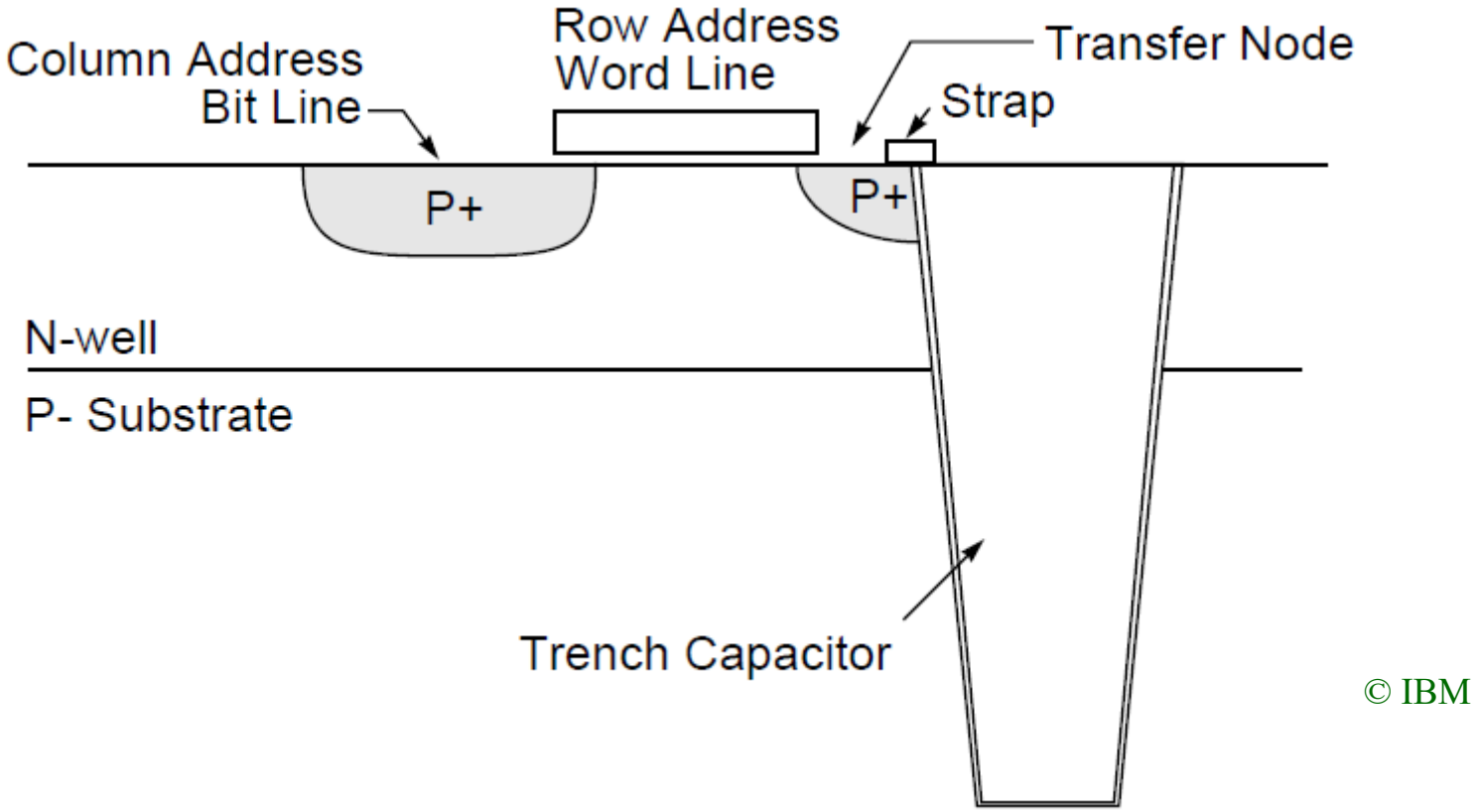
DRAM 4 x 4-es mátrix (1 bit adat)



Olvasás **Írás**

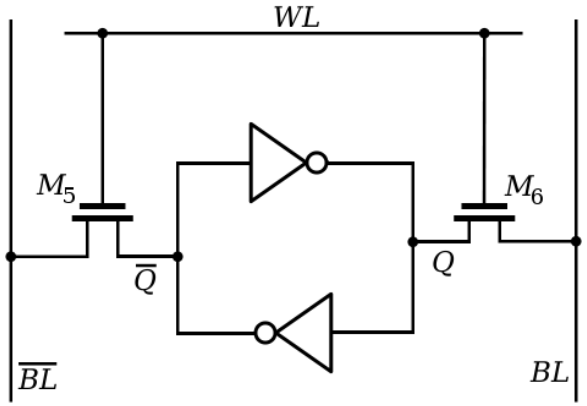
DRAM egy cellája

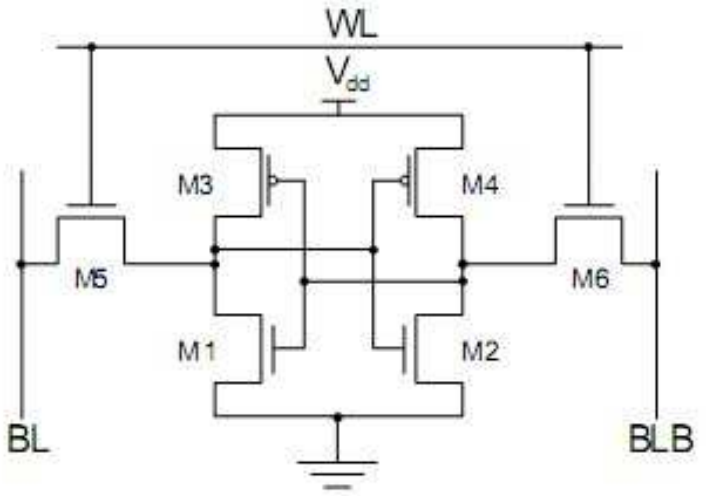




SRAM: Static Random Access Memory

- ezek is írhatók, olvashatók,
- random elérésűek (RAM),
- kiolvasási idejük hallatlanul gyors,
- de drágák és
- energiaigényesek (ezért melegednek, hűtendők!)
- gyorsító-tárakhoz (cache) használják.
- egy cellájuk 4-6 tranzisztorból álló flip-flop áramkör. Nincs bennük kondenzátor. (Olyanok, mint a CPU-k regiszeterei)

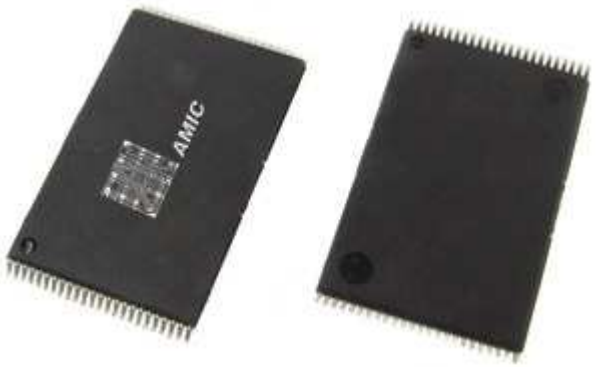




Nem felejtő memóriák

- A nem felejtő memóriát gyakran félrevezetően ROM-nak szokták nevezni
- Előnyük:
  - Tartalmukat a tápellátás megszűnésekor is megőrzik
  - A kontrollerek többsége írni is képes a nem felejtő memória területet
- Hátrányuk, hogy az írási művelet sokszor jóval lassabb, mint a felejtő memória esetében
- A nem felejtő memóriáknak számos fajtája van:
  - Maszkolt ROM
  - EPROM
  - OTP
  - EEPROM
  - Flash memória
  - FRAM

Maszkolt ROM



- A maszkolt ROM ténylegesen csak egyszer írható
- Tartalmát például fotólitográfiás eljárással a gyártás során ténylegesen beleégetik a memóriába
- Első sorban nagy szériás gyártásban gazdaságos

EPROM

- Erasable Programmable Read Only Memory), elektronikusan programozható ROM
- Ahogy a neve is mutatja elektronikusan programozható
- Törölni elektronikus úton nem lehet, csak például ultraibolya fénnyel
- A Flash memóriák elterjedése előtt széles körben alkalmazták

OTP

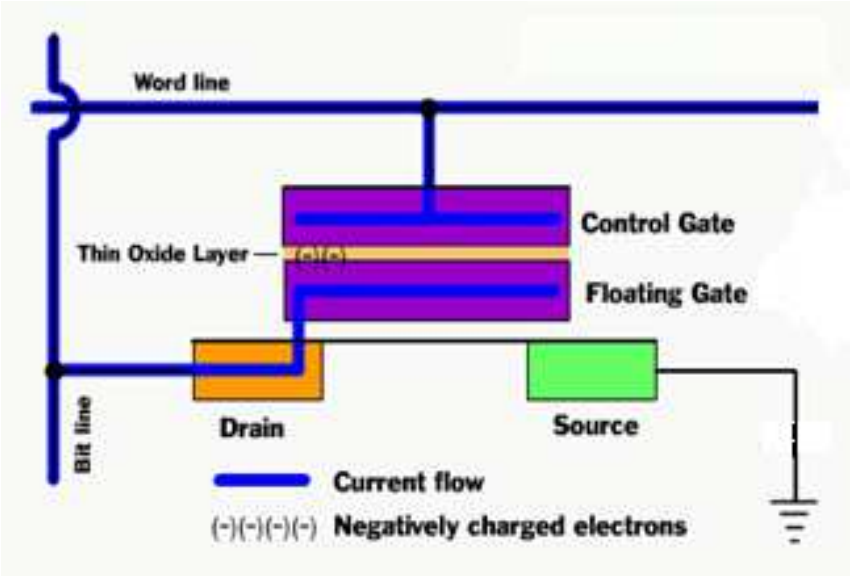
- One-Time Programable memory
- Egyszer írható ROM
- Lényegében egy EPROM, amelyet nem lehet törölni

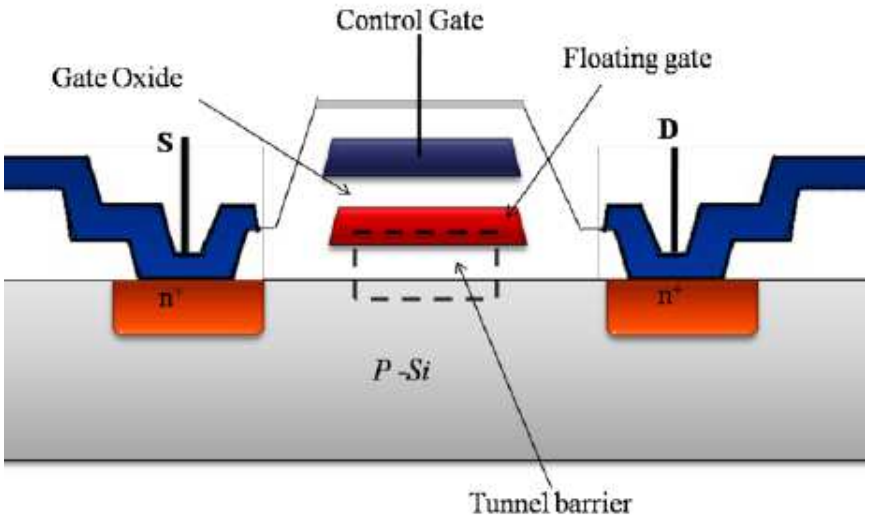
EEPROM

- **Electrically Erasable Programmable Read-Only Memory**
- **Elektronikusan törölhető és programozható ROM**
- **Hasonló az EPROM-hoz, ugyanakkor törlése történhet elektromosan is**

EPROM törölhető és újraírható

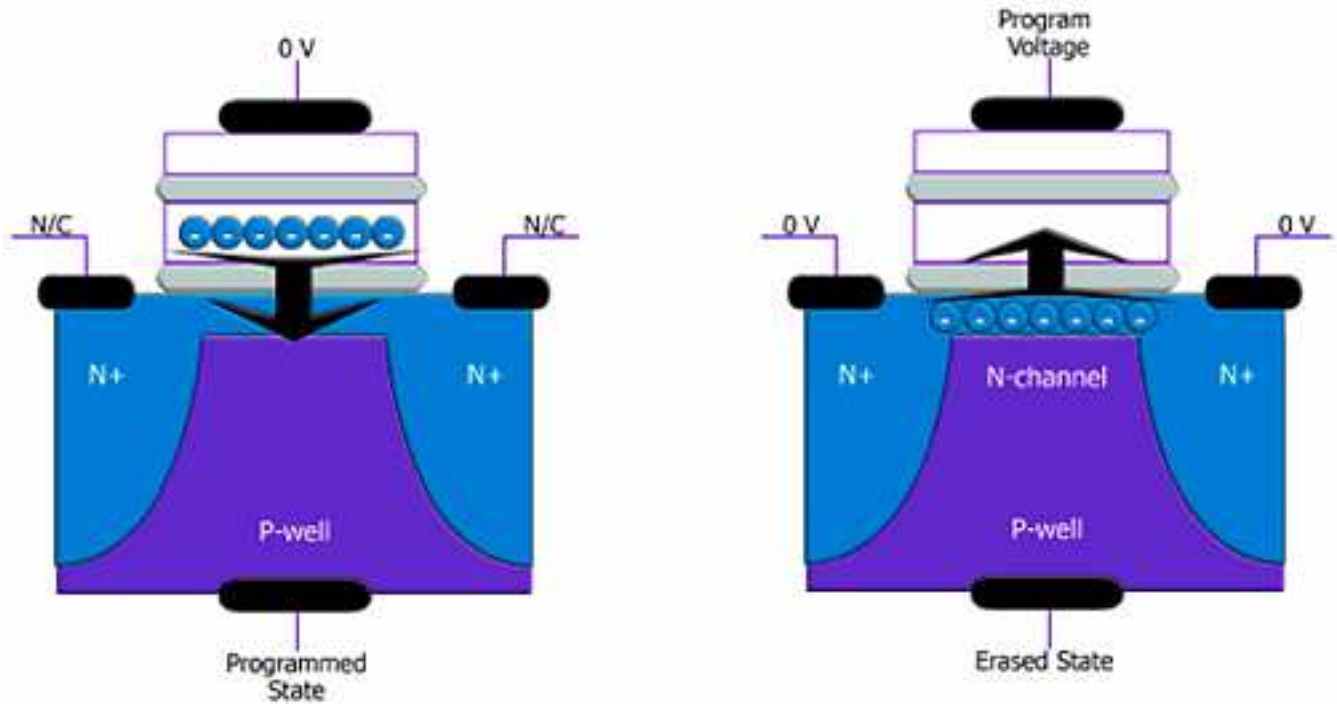
- Cellákban két kapus FET (Field-Effect Transistot)
  - Floating gate (negatívra töltve zár)
  - Control gate
  - Köztük oxidréteg
- 1 bit: „összekötve” a bit- és word line …
- 0 bit: a kapu „zárva”





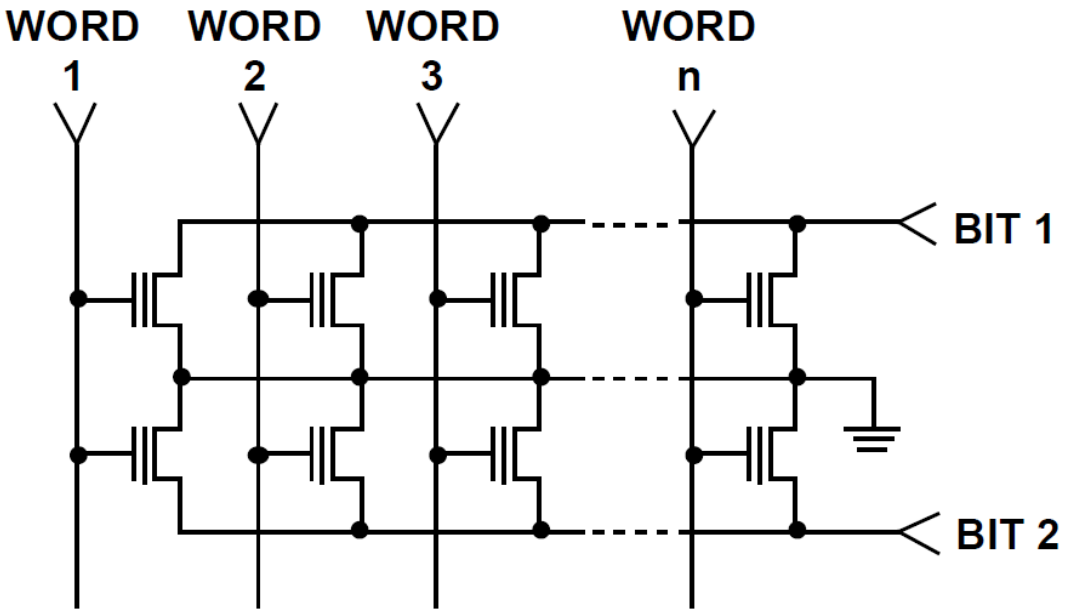
EEPROM cella programozás

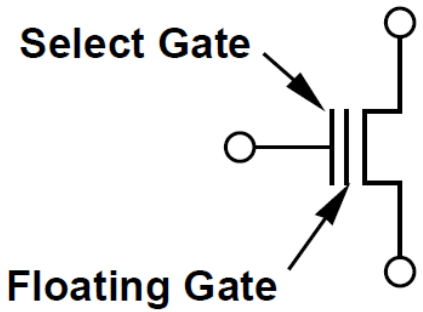
NAND Flash – programozás

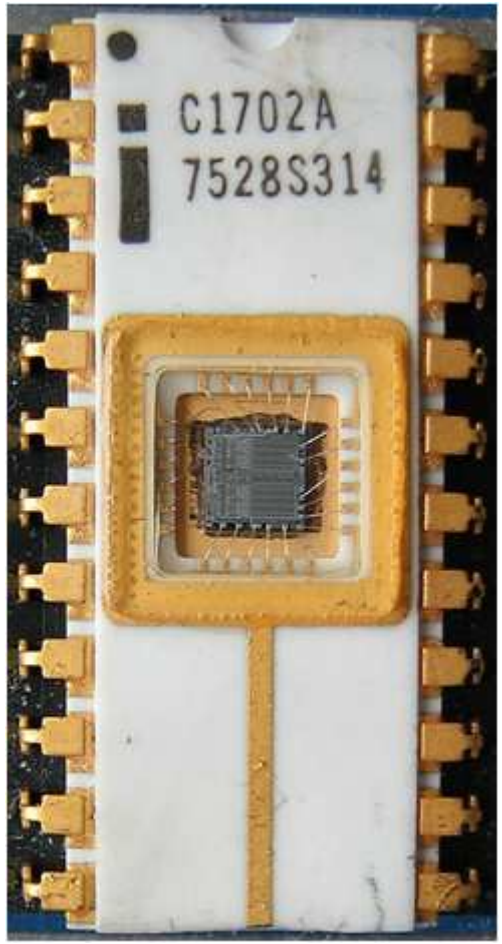


(tunneling)

EPROM





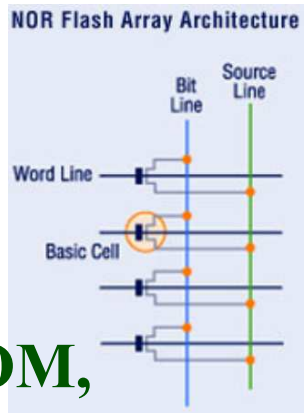


Source: ICE, "Memory 1997"

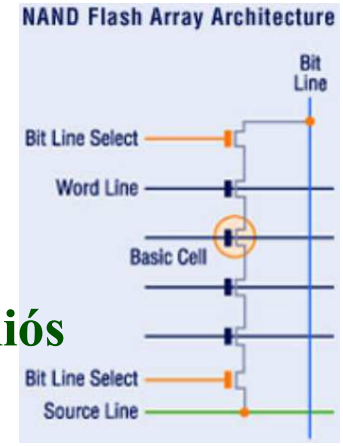
19051

Flash memória

- Leginkább az EEPROM-hoz hasonlít
- Írni és törölni is lehet elektromosan
- A fő különbség a Flash és az EEPROM között:
  - Felépítésbeli eltérések
  - Az EEPROM-ot bájtonként, míg a NAND Flash memóriát blokkonként lehet törölni
    - A NOR flash bájtonként is címezhető



- Fontos megjegyezni, hogy mind az EPROM, EEPROM, valamint a Flash memória esetében a memóriacelláknak van egy elhasználódási száma
  - Nem lehet végtelenszer törölni majd újraírni őket
    - Fizikai működési elvükből adódóan egy idő után „elhasználódnak”
  - A Flash memóriáknál a törlési/írási ciklusok száma a milliós vagy még magasabb nagyságrendet közelít



Félvezető tárolók – Solide State Drive

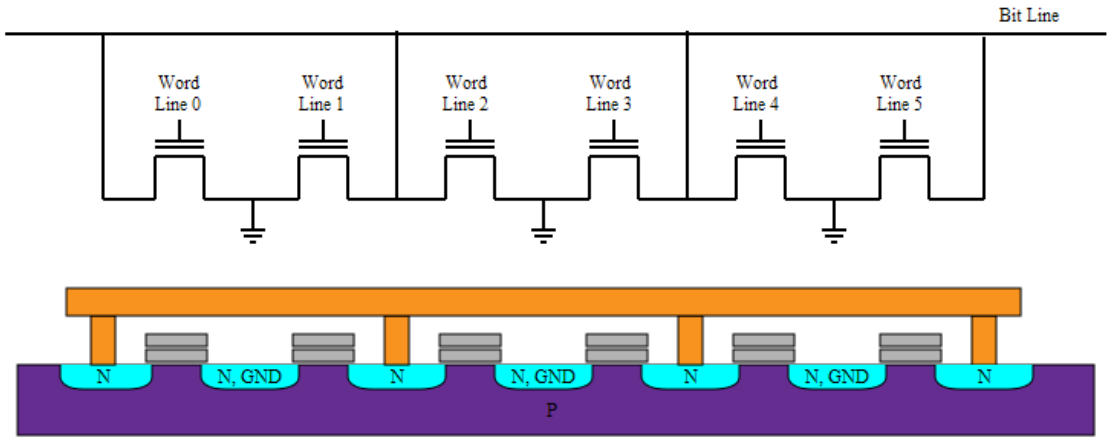
FLASH memory

Memory wear: 100 000
program-erase (P/E) cycles

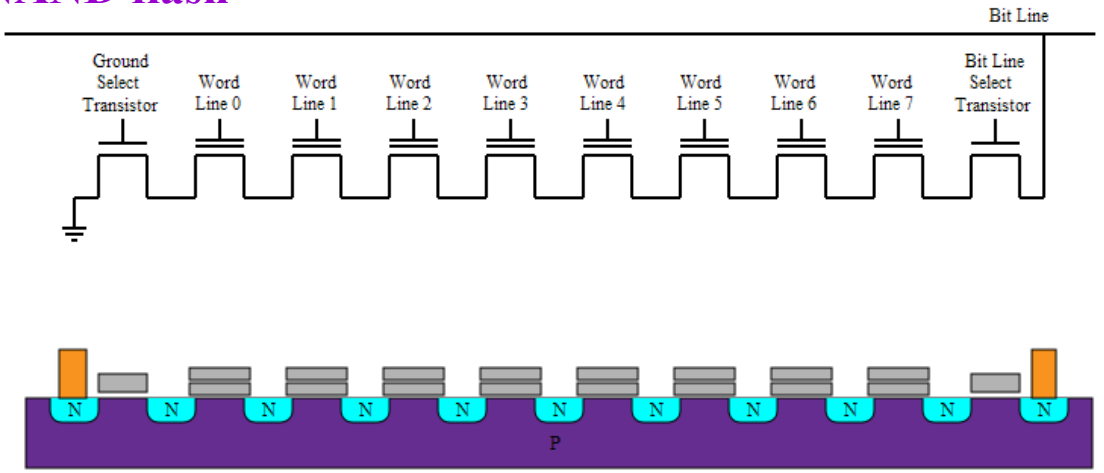
Erasure sets (all) bits,
programming can only
clear bits:
1111 – 1110 – 1010 - 0010,
finally 0000

Cell read-out:
a voltage intermediate between the threshold voltages is applied to the CG, and the MOSFET channel will conducting or remain insulating, depending on the V_T of the cell, which is in turn controlled by charge on the FG.

NOR flash Nem sor, hanem csak 1 bit egyszerre

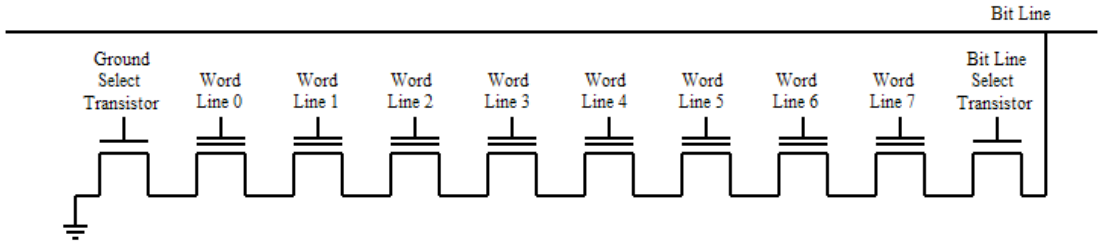


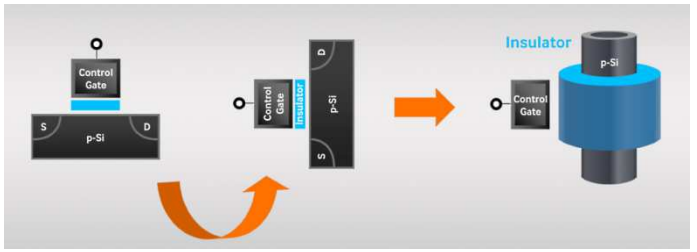
NAND flash

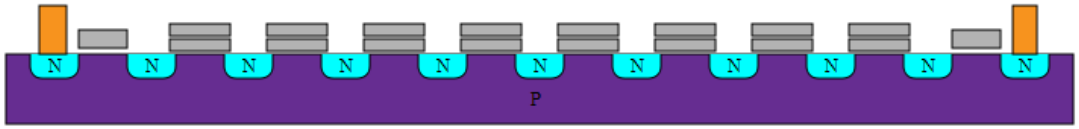


Solide State Drive – kapacitás növelés

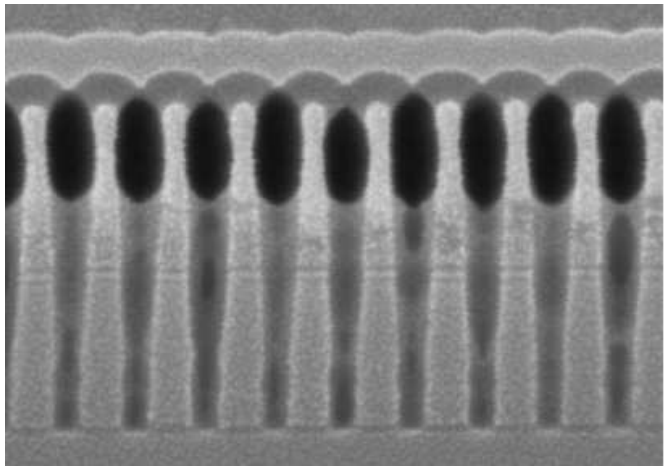
NAND Flash, elrendezés

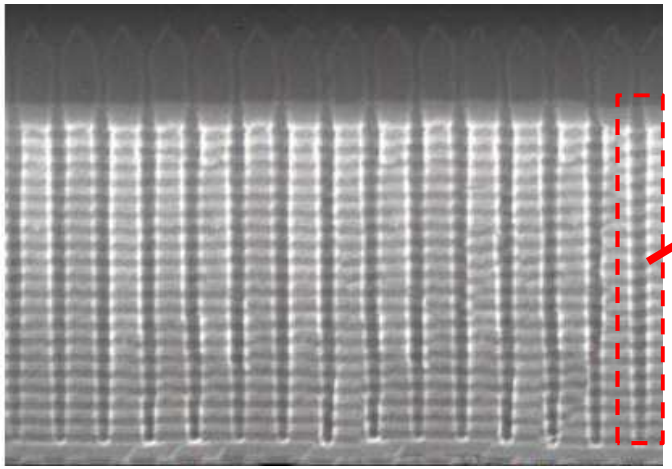






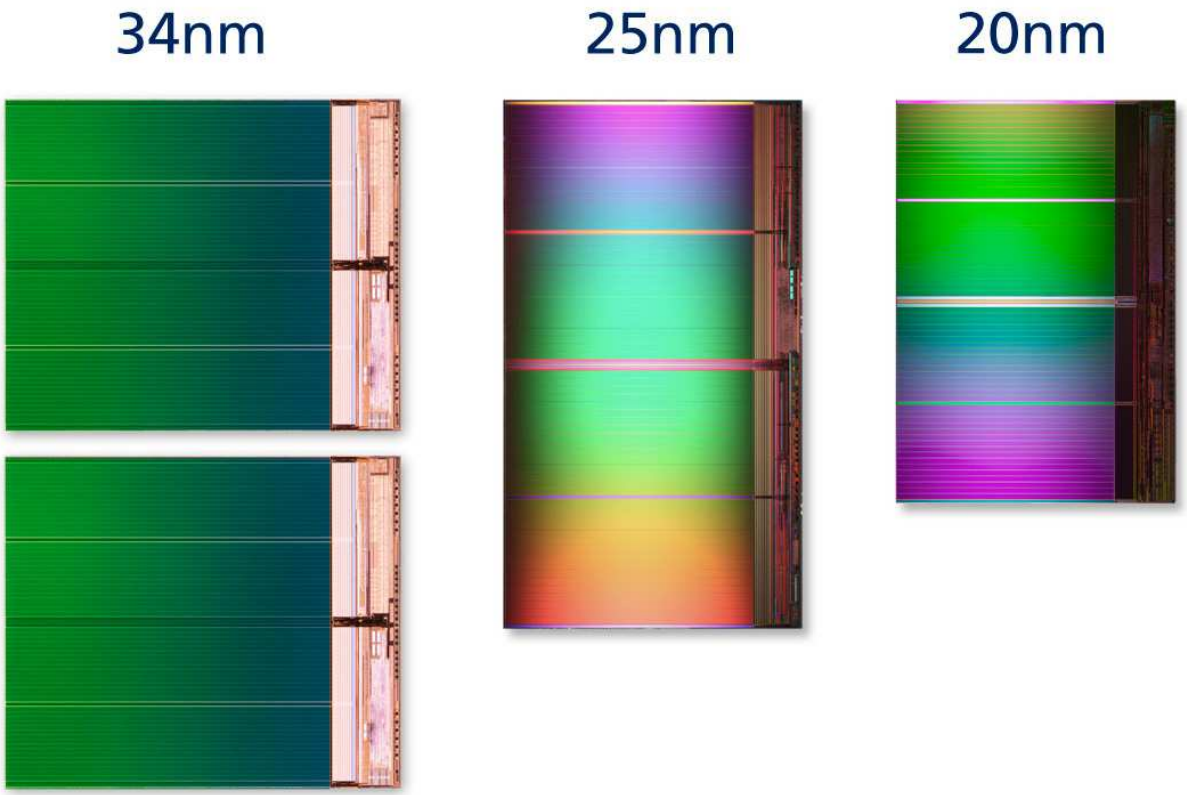
2D **3D**





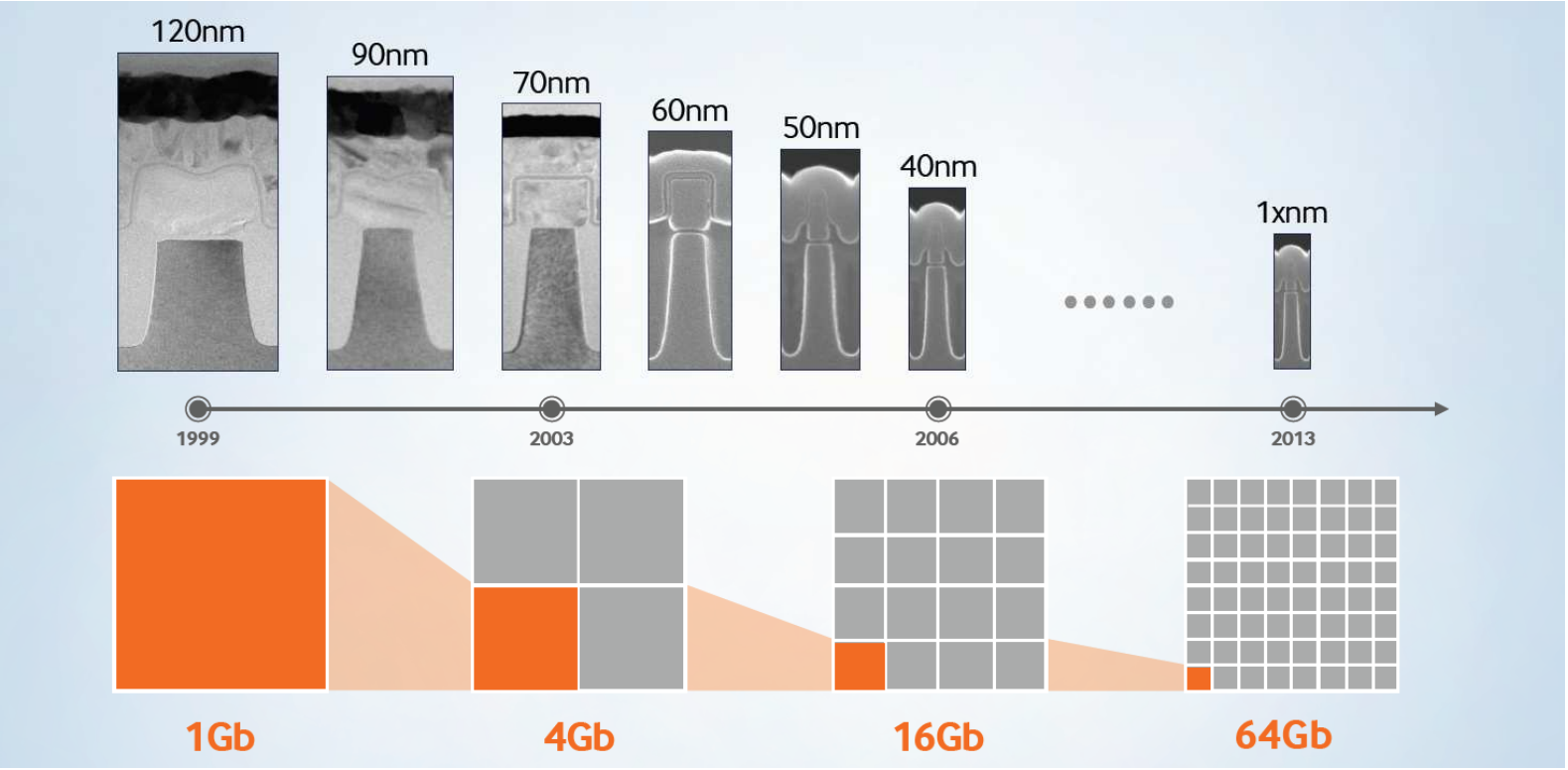
Solide State Drive – kapacitás növelés

NAND Flash – csíkszélesség (azonos kapacitás esetén)



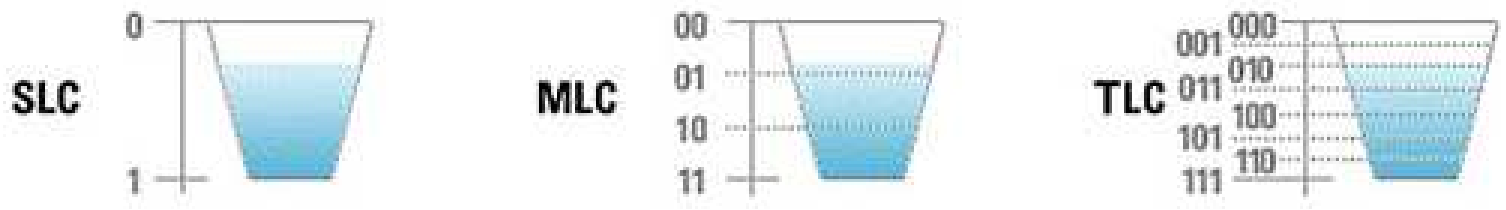
Solide State Drive – kapacitás növelés

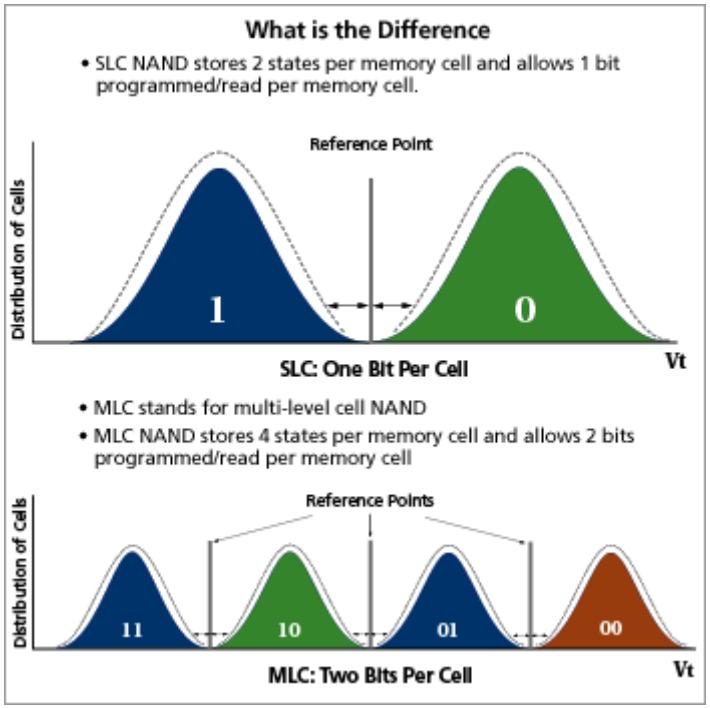
Csíkszélesség – gond az élettartam (endurance)



Solide State Drive – kapacitás növelés

Állapotok számának növelése (SLC:1, MLC:2, TLC:3 bit)



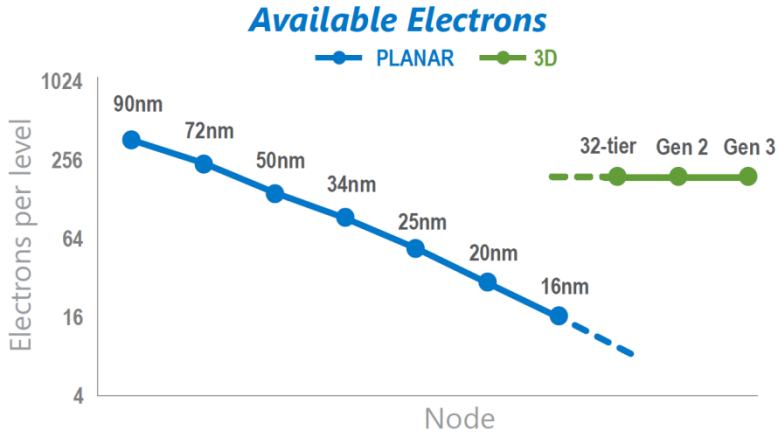


Solide State Drive – kapacitás növelés

Gond az élettartam (endurance)

NAND típus (20nm)	SLC	MLC	TLC (3-bit MLC)
Cellánkénti bitek száma	1	2	3
Max. programozási ciklus	100 000	3000	1000
Olvasási idő	25 µs	50 µs	75 µs
Írási idő	200-300 µs	600-1200 µs	900-1600 µs
Törlési idő	1,5-2 ms	2-3 ms	4,5-5 ms

Megoldás: tartalékok és használat kiegyenlítés (wear-leveling algorithms)



FRAM (Ferroelectric Random Access Memory)

- Hagyományos nem felejtő memóriákkal szembeni előnyök:
  - Felépítése leginkább a dinamikus memóriákéhoz hasonlít
    - Azonban a dielektromos réteg helyett ferroelektromos réteget alkalmaz a memóriacelláknál
      - Kiküszöböli annak állandó frissítési igényét
      - Nem felejtővé is teszi
  - FRAM-ok esetében sokszor kisebb az elhasználódás mértéke, mint a Flash memóriáknál
    - Elsősorban az alacsony feszültségen működő FRAM-okra igaz
  - Az SRAM-okhoz hasonló írási idővel rendelkeznek (100 ns környéki vagy még kisebb)
  - Jelentősen gyorsabbak, mint a Flash memóriák
- A Flash memóriákhoz mérten magas ár és nagyobb fizikai méret jellemzi

Beágyazott rendszerek

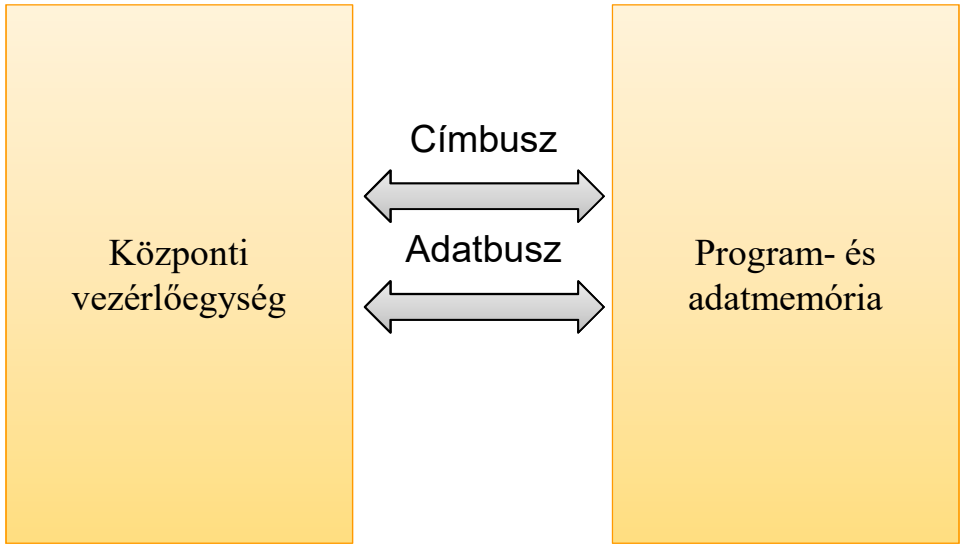
- **Tartalom**
  - Mit nevezünk beágyazott rendszernek?
  - Központi vezérlőegységek típusai
  - Mikrokontrollerek alapvető felépítése
  - Mikrokontrollerek perifériái
  - Mikrokontrollerek memóriái
  - **Mikrokontroller architektúrák**
  - Utasításkészletek

Mikrokontroller architektúrák

- **A beágyazott rendszerekben alkalmazott kontrollerek többnyire két különböző architektúrát alkalmaznak**
- **A fő különbség a memóriaelérés, valamint memória felépítésének szempontjából van**
- **A két architektúra**
  - ***Harvard-architektúra***
  - ***Neumann-architektúra***
- **Mindkét felépítés esetén a fő részegységek funkciója megegyezik**

Neumann-architektúra

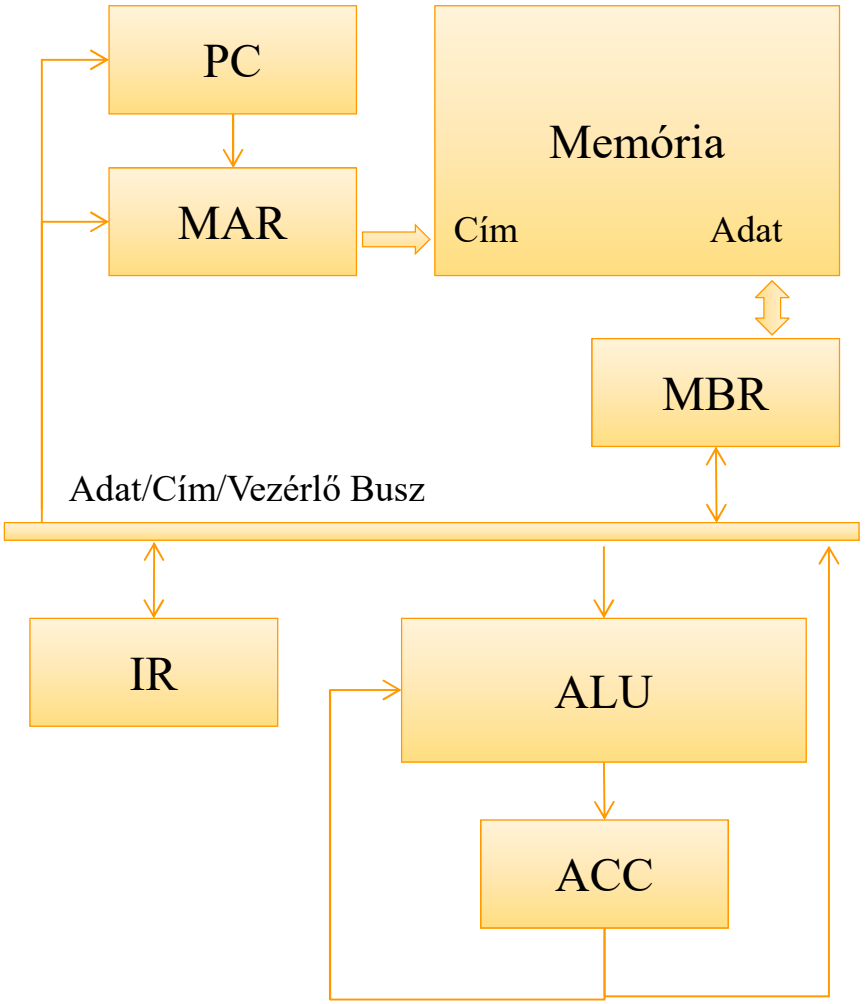
- **Az adat- (nem felejtő) és a programmemória (felejtő) fizikailag nem választhatóak szét**
- **Szekvenciális utasítás végrehajtás**



Neumann-architektúra

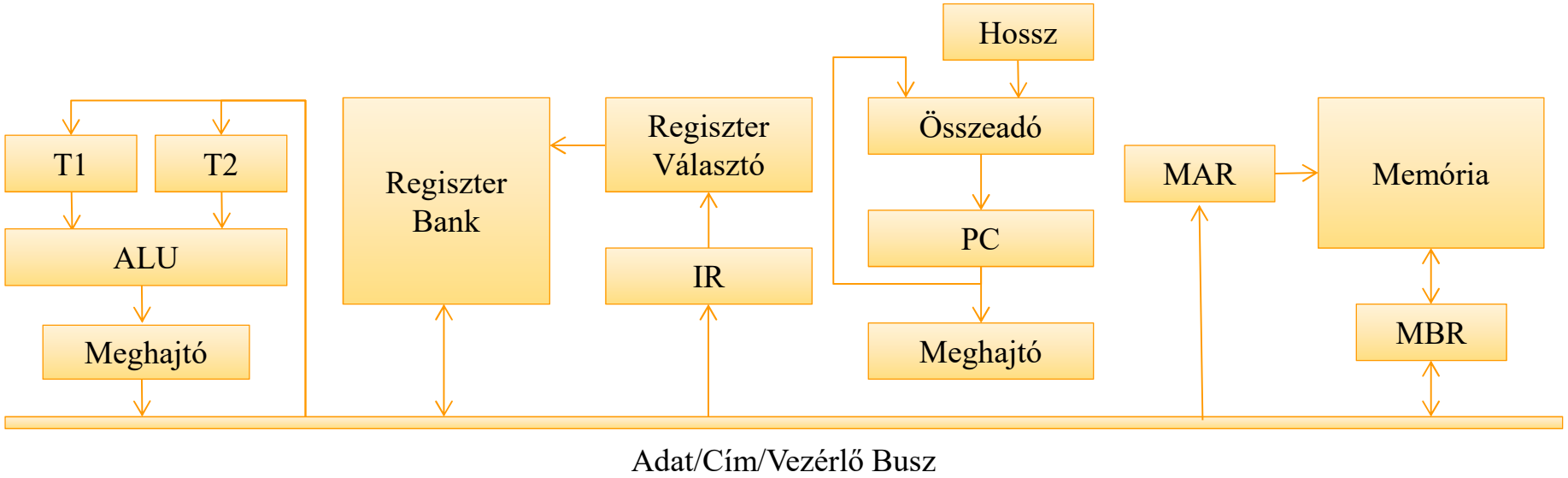
Neumann elvű számítógép

- **Egycímű gépek**
  - **ALU: (Aritmetic Logic Unit) aritmetikai-logikai egység, matematikai és logikai műveleteket végez el**
  - **ACC: (ACCumulator) akkumulátor, speciális regiszter, a legtöbb ALU-t érintő művelet egyik operandusát (két operandus esetén) és a művelet eredményét tartalmazza**
  - **PC: (Program Counter) programszámláló, a soron következő (végrehajtandó) utasítás címét azonosítja**
  - **IR: (Instruction Register) utasításregiszter, a végrehajtás alatt álló utasítás kódját tartalmazza**
  - **MAR: Memory Address Register**
  - **MBR: Memory Buffer Register**



Neumann elvű számítógép

- **Többcímű gép (regiszteres)**
  - **Egy utasítással több műveletet lehet azonosítani**
  - **A többcímű utasítások meghatározzák, mind a forrás mind a cél (utolsó operandus címe) információt**

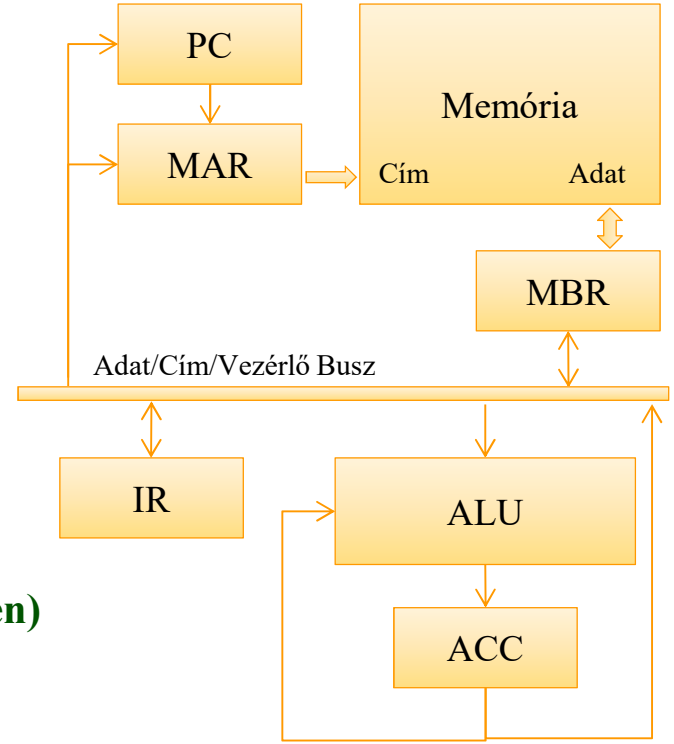


Utasítás kódok

- Az egyes egységek között végbemenő tranzakciók leírása
- Az utasítások végrehajtásának leírására szolgáló nyelv az assembly (az utasítások gyűjteménye a gépi utasításkészlet)
- Egy utasítás végrehajtásának három fő lépése:
  - Fetch: az utasítás betöltődik a memóriából az utasításregiszterbe (lehívás)
  - Decode: utasítás értelmezése (azonosítja az utasítást)
  - Execute: a dekódolt utasítás végrehajtása az adatokon, majd az eredmény visszaírása a memóriába (WriteBack)

Példa egy utasítás leírására

- **Egycímű ADD utasítás RTL leírása**
  - **Fetch**
    - **PC → MAR** PC-ből a következő utasítás címe a MAR-ba töltődik
    - **M[MAR] → MBR** Memóriában levő utasítás beírása az MBR-be
    - **MBR → IR** MBR-ben lévő adat az IR-be íródik
    - **PC + I_len → PC** Az utasítás hosszával (I_len) növekszik a PC értéke
  - **Decode**
  - **Execute**
    - **IR<addr> → MAR** Operandus címe beíródik a MAR-ba
    - **M[MAR] → MBR** Ez az érték összeadódik az ACC-vel
    - **ACC + MBR → ACC** összeadás (eredménye bekerül az ACC-be)

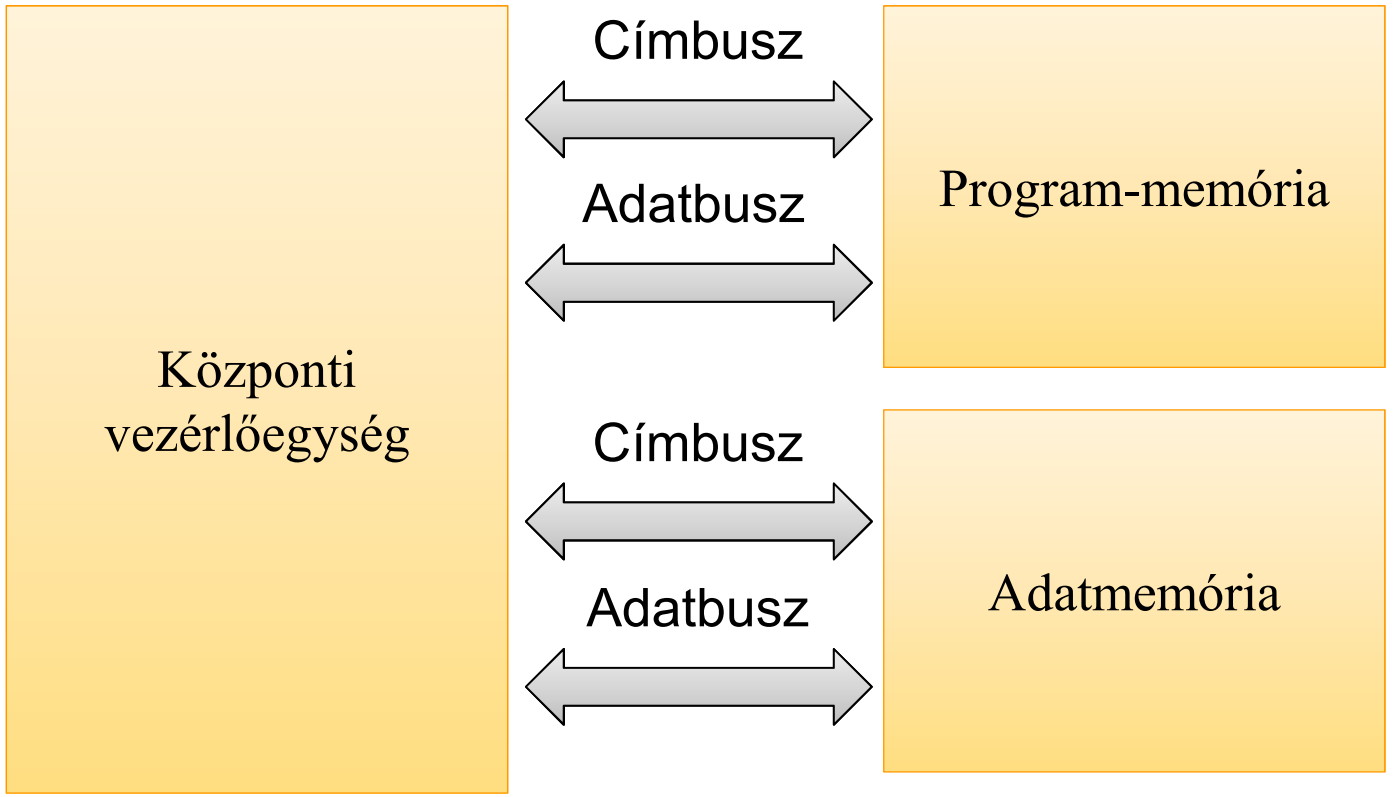


- ***RTL: register transfer level***

Harvard-architektúra

- A program- és adatmemória különbözik
- Minden utasítás 1 szavas, a programmemória szóhosszúsága tetszőleges lehet, így nem jelent korlátozó tényezőt az adatmemória általában bájtos kialakítása
- A speciális kialakítás miatt nem alakulhat ki versenyhelyzet az adat és kód elérésnél
- Egyszerre (párhuzamosan) is kezelhetőek a memóriák
- A beágyazott rendszerek többségénél alkalmazott vezérlőegységek Harvard-architektúrát használnak

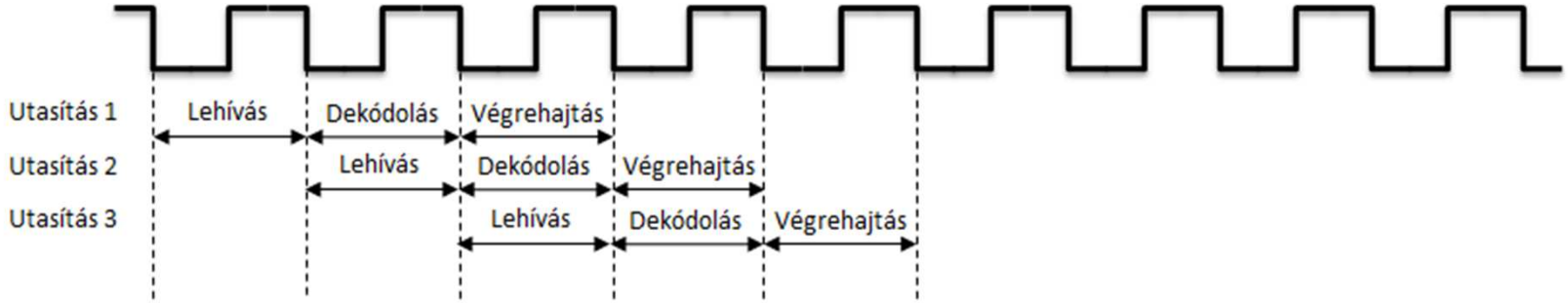
Harvard-architektúra



Harvard-architektúra

Pipeline-elv

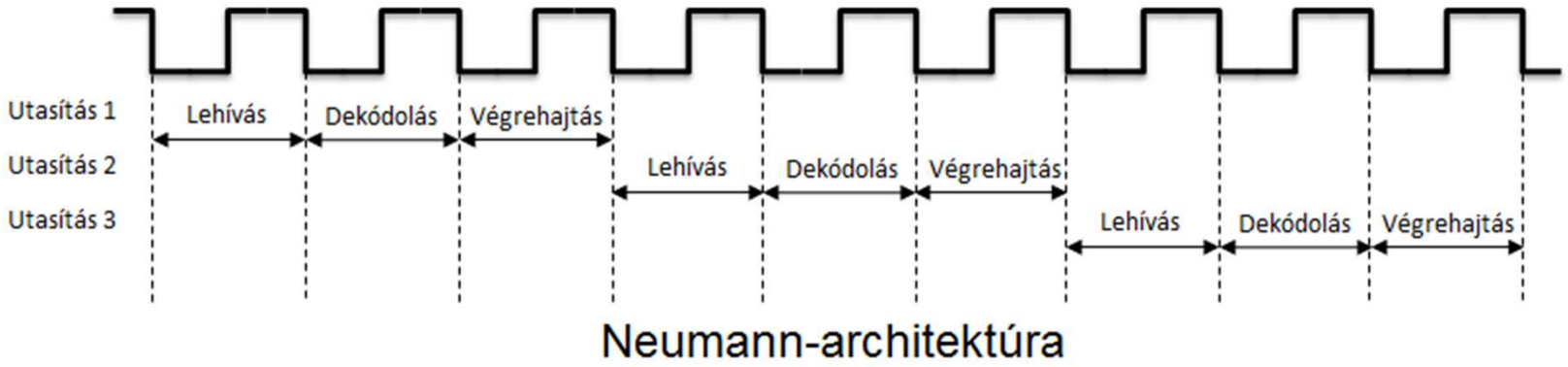
- A Neumann- és Harvard-architektúra közötti egyik legfontosabb különbség
- Az utasítások feldolgozása három részre osztható: utasítás lehívás, dekódolás, végrehajtás
- Harvard-architektúrájú processzorok esetén egyszerre címezhető a program- és az adatmemória
- Az első utasítás dekódolásával, egyidejűleg lehívható a második utasítás kódja és így tovább



Harvard-architektúra

Pipeline-elv

- **A Neumann architektúra ezt nem teszi lehetővé, mivel a címbuszon egyszerre csak egy cím lehet**



Neumann-architektúra

Beágyazott rendszerek

- **Tartalom**
  - Mit nevezünk beágyazott rendszernek?
  - Központi vezérlőegységek típusai
  - Mikrokontrollerek alapvető felépítése
  - Mikrokontrollerek perifériái
  - Mikrokontrollerek memóriái
  - Mikrokontroller architektúrák
  - **Utasításkészletek**

Utasításkészletek

- A különböző processzorok illetve mikrokontrollerek eltérő utasításkészletekkel rendelkeznek
- A különböző utasítás készleteket, illetve a kontrollereket ezek alapján két fő csoportba szokták besorolni:
  - Csökkentett utasításkészletű eszköz (Reduced Instruction Set Computer (RISC))
  - Összetett utasításkészlettel rendelkező eszköz (Complex Instruction Set Computer (CISC))

RISC (Reduced Instruction Set Computer)

- **Csak a felhasználás szempontjából legszükségesebb utasítástípusok**
- **Az utasítások végrehajtásához nincs mikroprogram, az igen bonyolult fordítóprogram állítja elő a végső formát (futás gyorsabb, a hardver egyszerűbb)**
- **Azonos hosszúságú utasításformátum (dekódolási folyamat (a végrehajtandó utasítás azonosítása) minimális idejű)**
- **Egyszeres ciklusvégrehajtás (cél, hogy minden egyes ciklusban egy utasítás kerüljön végrehajtásra)**
- **Memóriaelérés csak load és store műveletek segítségével (aritmetika csak regisztereken végezhető)**
- **Pipeline (utasítás feldolgozás párhuzamosítása, többszörös adatvonalak, nagyszámú gyors regiszter)**

Példa a RISC utasításkészletre

- **PIC16C84 utasításkészlete**

Mnemonic, Operands		Description	Cycles	14-Bit Opcode MSb			LSb	Status Affected	Notes
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0000	0011	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff	None	1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff	None	1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff	None	
NOP	-	No Operation	1	00	0000	0xx0	0000	None	
RLF	f, d	Rotate left f through carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate right f through carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff	None	1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff	None	1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff	None	1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff	None	3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff	None	3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kkkk	kkkk	None	
IORLW	k	Inclusive OR literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kkkk	kkkk	None	
RETFIE	-	Return from interrupt	2	00	0000	0000	1001	None	
RETLW	k	Return with literal in W	2	11	01xx	kkkk	kkkk	None	
RETURN	-	Return from subroutine	2	00	0000	0000	1000	None	
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kkkk	kkkk	Z	

CISC (Complex Instruction Set Computer)

- **Bonyolult hardver**
- **Nagyszámú utasítás-típus és címzési mód**
  - **Egy utasítással több elemi feladat végrehajtása (az utasítás dekódolás hosszú (azonosítani kell az utasítás hosszát))**
- **Mikro-programozott vezérlési mód**
  - **A fordító a programot egy egyszerűbb formára fordítja, majd a mikroprogram veszi át a vezérlést, azaz a végrehajtás alatt lévő utasítás egy mikroprogramtárban lévő programot indít el**
  - **A mikorprogram utasításai a mikroutasítások, melyek elemi vezérlési lépések szerint vezérlik a hardver működését**

CISC - Mikroprogram

- **Digitális számítógépek gépi utasításainak (pl. szorzás, osztás) végrehajtására szolgáló elemi lépésekből - mikroutasításokból - álló program**
- **A mikroutasítások hatására egy-egy elemi műveletvégzés történik (átvitel, léptetés stb.)**
- **A gépi utasítás végrehajtása ily módon mikroutasítások sorozatának végrehajtása**
- **A mikroutasítások egy speciális tárolóban, a mikroprogramtárban találhatóak, amelynek felépítése hasonlít a gépi utasítások tárolására használatos operatív memóriaegység felépítéséhez**
- **A legfontosabb eltérés, hogy a mikroprogramtár rendszerint csak olvasható tartalmát nem lehet a gép működése közben módosítani**

CISC - Mikroprogram

- **Egy utasítás végrehajtása úgy történik, hogy az utasításszerkezetben előírt műveleti kód (pl. osztás) meghatározza a megfelelő - kezdőcímét a mikroprogramtárban, majd elindítja ettől a címtől kezdve a mikroutasítások végrehajtását**
- **Az egymás után végrehajtásra kerülő mikroutasítások megvalósítják az utasítás mikrolépéseinek végrehajtását**
- **Ahol nincs mikroprogram ott a hardver hajtja végre a gépi kódú utasításokat**

Ajánlott és felhasznált irodalom

Az előadás anyag forrása:

- Pannon Egyetem, Mérnöki Kar, Gépészmérnöki Intézet Járműmechatronika Intézeti Tanszék
- **Dr. Fodor Dénes, Speiser Ferenc: Autóipari beágyazott rendszerek**
- http://moodle.autolab.uni-pannon.hu/Mecha_tananyag/autoipari_beagyazott_rendszerek/index.html

Dr. Fodor Dénes intézetigazgató, tanszékvezető egyetemi docens
szoba: C218
telefon: 88/624-000/6082, 88/624-776
e-mail: fodor@almos.uni-pannon.hu

Speiser Ferenc egyetemi tanársegéd
szoba: C217
telefon: 88/624-000/6083
e-mail: speiserf@almos.uni-pannon.hu