

**Beágyazott rendszerek**  
**KMABR11TNC**  
Tematika

- **Beágyazott rendszerek felépítése**
- **Beágyazott rendszerek szoftver vonatkozásai**
- **Beágyazott rendszerek hardver vonatkozásai**
- **Beágyazott rendszerek tervezése**
- **<http://www.hik.hu/tankonyvtar/site/books/b10096/index.html>**
- **Fuzzy <http://www.hik.hu/tankonyvtar/site/books/b120/index.html?highlight=>**

**Beágyazott rendszerek**

- Olyan *számítógépes eszközök*, amelyek alkalmazás-orientált célberendezésekkel, ill. komplex alkalmazói rendszerekkel *szervesen egybeépülve* azok *autonóm* működését biztosítják, vagy segítik. [IEE Guidelines].
- A *beágyazott rendszerek* szerteágazó monitorozási, vezérlési, ill. szabályozási feladatokat látnak el.
- Közös jellemzőjük a *fizikai környezettel* való intenzív *információs kapcsolat*.

**A beágyazott rendszer, mint  
alkalmazás-orientált célberendezés**

- egy chip-es rendszerek (systems on a chip)
- újrahasználható részegységek (design re-use, IP)
- integrált érzékelők és beavatkozók
- mikrovezérlők és programozható logikai eszközök
- FPGA és DSP processzoros részegységek/kártyák
- hardver-szoftver együttes tervezés
- a tervezést és a tesztelést segítő technológiák

## **Beágyazott információs rendszerek**

integrált alkalmazói rendszerek

- új minőség, újszerű szolgáltatások
- autonóm viselkedés: flexibilitás és adaptáció
- mérések beágyazott rendszerekben
- intelligens működési mechanizmusok
- új rendszertervezési elvek
- új rendszerfejlesztési technológiák

## **Tervezés, gyártás és üzemeltetés**

- a szolgáltatások komplexitásának kezelése
- a szolgáltatások minőségi kérdései
- hardver-szoftver együttes tervezés
- építkezés komponensekből
- konfigurálhatóság (tervezéskor: at design time)
- tesztelhetőség
- szolgáltatásbiztonság
- újrakonfigurálhatóság (üzem közben: at runtime)

## **A hierarchikus és több aspektusú modellezések szerepe**

- komplexitás kezelés
- a specifikáció teljesítése, mint kényszer-kielégítés
- kvantitatív és kvalitatív jellemzések
- válaszidő követelmények teljesítése
- hibadetektálás és lokalizálás támogatása
- aktív modellek létrehozása

## **Minőségbiztosítás a tervezés fázisában**

- hardver és szoftver együttes tervezés
- a szolgáltatásbiztonság mint tervezési szempont
- - biztonsági és hibatűrési képességek kialakítása
- - szoftver biztonság és hibatűrés
- - elosztott rendszerek biztonsága és hibatűrése
- - adatbázisok biztonsága és hibatűrése
- - időkezelés biztonsága és hibatűrése

### Beágyazott rendszerek felépítése

- A beágyazott rendszerekkel szemben támasztott főbb követelmények:
- - funkcionális,
- - időzítési,
- - megbízhatósági.
- Ezen követelmények hatása a tervezésre.

### Funkcionális követelmények

- Berendezéssel szembeni követelmények
- Környezeti feltételek
- Statikus követelmények
  - Villamos követelmények, Mechanikai követelmények

### Időzítési követelmények

- Real-time rendszerek követelményei
- Eseménykezelés
- Szinkron események
- Aszinkron események
- Belső időzítők
- Külső időzítőkhöz történő szinkronizálás

### Megbízhatósági követelmények

- Forrasztások száma
- Elemek száma
- Csatlakozása száma
- Pergésmentesítése
- Áramkör típus TTL, CMOS

## Valós idejű rendszerek

- Az **operációs rendszerek** egy speciális fajtája, amelyben minden egyes rendszerhívás egy előre meghatározott időn belül garantáltan végrehajtásra kerül, a konkrét körülményektől függetlenül.

A nem valós-idejű operációs rendszerek esetében ilyen korlát nem létezik, és a rendszerhívások a körülményektől függően elméletileg rendkívül sok ideig is eltarthatnak, ami azonban problémát jelenthet bizonyos szigorú időzítést és azonnali rendelkezésre állást igénylő feladatok esetében.

A valós-idejű operációs rendszereket általában beágyazott **eszközökben** alkalmazzák, amelyek a helyes és megbízható működéséhez elengedhetetlenül szükségesek speciális jellemzői.

### *Kemény és puha valós idejű rendszerek:*

- kemény valós idejű rendszer (hard real-time system (HRT)): katasztrofális következményekkel jár, ha nem tartjuk az időkorlátot (pl. járművek vezérlése).
- puha valós idejű rendszer (soft real-time system (SRT), on-line system): az eredmény értékes az időkorláton túl is, de az idővel degradálódik (pl. tranzakciós rendszerek).

#### *HRT és SRT jellemzése különböző szempontok szerint:*

- **válaszidő (response time):** HRT esetében ms, vagy annál kevesebb (pl. légzsák), az emberi beavatkozás lehetősége kizárt, a rendszer autonóm működésű és biztonságos kell legyen. SRT esetén a válaszidő másodperc nagyságrendű, az időkorlát túllépése nem okoz katasztrófát.
- **viselkedés csúcsterhelés esetén (peak-load performance):** HRT esetén jól definiált kell legyen. Tervezéskor biztosítani kell, hogy a számítógépes rendszer minden szituációban az időkorláton belül teljesítse feladatát, hiszen a HRT rendszerek éppen azért valósítják meg a velük szemben megfogalmazott elvárásokat, hogy még a ritkán előforduló csúcsterhelések idején is jószolható módon viselkedjenek. Az SRT rendszereket átlagos teljesítmény-jellemzőkre tervezzük, a ritkán előforduló csúcsterhelések következményeit - gazdaságossági megfontolásból - elviseljük.
- **az ütem vezérlése (control of pace):** A HRT rendszernek minden körülmények között szinkronban kell lennie környezetének (irányított objektum, ill. az emberi

operátor) állapotával. Az SRT rendszerek befolyásolják környezetüket, ha nem képesek eleget tenni feladatuknak (egy tranzakciós rendszer például megnöveli a válaszidejét).

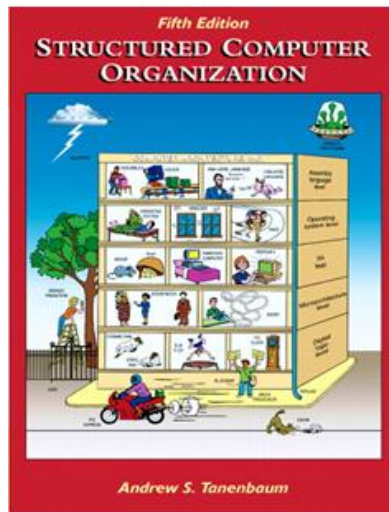
- *biztonság (safety)*: A biztonság kritikusságának mértékétől függően sokféle feladat merülhet fel tervezési időben. Autonóm hibadetektálási mechanizmusokat kell kidolgozni, amelyek valamilyen “talpra állítási” (recovery) akciót indítanak az adott alkalmazás által diktált időviszonyok mellett.
- *az adatfájlok mérete (size of data files)*: HRT rendszerek kisméretű adatfájlokra dolgoznak, amelyek valós idejű adatbázist alkotnak. Ezek jellemzője az adatintegritás rövid idejűsége, mert az idő múlásával az adatok jelentős része aktualitását veszíti. Az SRT rendszerekben éppen ellenkezőleg a hosszú idejű adatintegritás fontos.
- *a redundancia típusa (redundancy type)*: SRT rendszerekben (pl. tranzakciós rendszerek) hiba esetén a számításokat “visszagörgetik” a legutolsó ellenőrzési ponthoz, amikor még biztosan helyes volt a működés és onnan kezdik a “talpra állítást”. HRT rendszerek esetén ez a stratégia csak korlátozottan használható mert:
  - (1) az időkorlát tartása nehéz, mert a visszagörgetéshez szükséges idő nem, vagy nehezen jósolható,
  - (2) a környezetet befolyásoló “utasítás” nem tehető meg nem történné,
  - (3) az ellenőrzési pontnál érvényes adatok az idő múlásával érvényüket veszítik.
- *adat integritás (data integrity)*: HRT: rövid idejű, SRT: hosszú idejű.
- *hibadetektálás (error detection)*: HRT: autonóm, SRT: felhasználó által segített.  
*Hibás állapot kezelése:*
- (1) katasztrófa megakadályozás bénítással (*fail-safe*). Pl. vonatok jelzőlámpa rendszere: minden lámpa piros.
- (2) katasztrófa elhárítás extra eszközökkel (*fail-operational*). Pl. repülőgép: muszáj valahogy leszállni.

## Időkezelés II.

- Vezérlési módok.
- Időmérés eszközei és módszerei, időnormáliák. Idő reprezentálása. Időkényszerek.
- Szinkronizálás, órarendszerek.
- Master-slave algoritmusok. Elosztott szinkronizációs algoritmusok.

### *Vezérlési mód:*

- (1) esemény vezérelt (*event triggered*): aszinkron módon érkező megszakítások kiszolgálása, ill. ebből adódóan dinamikus ütemezés szükséges.
- (2) idő vezérelt (*time triggered*): minden kommunikáció, ill. feldolgozás központi időzítéshez (órához) szinkronizált módon történik.



### **Beágyazott rendszerek hardver vonatkozásai I.**

- Beágyazott rendszerek tipikus hardver struktúrája.
- Perifériák (AD/DA, kijelzők, billentyűzet, kommunikációs csatornák).

### **Beágyazott rendszerek hardver vonatkozásai II.**

- Periféria illesztés HW szempontjai. Regiszter, memóriába ágyazott, IO.
- Periféria szinkronizálása.
- Beágyazott rendszerek feldolgozó egységei (controller, DSP, FPGA). Integrált perifériák.

### **Számítógép rendszerek felépítése**

- Processzorok, CPU
  - Neumann architektúra
    - Adatút (data path)
    - Aritmetikai-logikai egység (ALU, Arithmetic Logic Unit)



- Utasítások
  - Regiszter-memória
  - Regiszter-regiszter
- Adatútciklus fogalma (betöltő-dekódoló-végrehajtó)
- Értelmező (interpreter)

#### — Havard architektúra

#### Fejlődés

- Egyszerű utasításkészlet, interpreter
- IBM architektúra (1950), IBM System/360 közvetlen hardveres feldolgozás
- Összetett utasítások, Digital Equipment Corporation VAX (több száz utasítás, 200 címezési mód)
- Hardveres értelmezők Motorola 68000, mikroutasítások, vezérlőtárak (control store)

#### CISC, RISC

- 1980, **Berkeley** RISC (Reduced Instruction Set Computer)
- CISC (Complex Instruction Set Computer), pl. DEC VAX
- 486-os RISC mag és CISC mag (a kompatibilitás miatt)  
Korszerű számítógépek tervezési elvei
- Minden utasítást a hardver hajtson végre
- Maximálni kell az utasítások kiadásának ütemét (MIPS, Millions of Instructions Per Second)
- Csak a betöltő és tároló utasítások hivatkozzanak a memóriára (memóriát kezelő utasítások lassúak)
- Sok regiszter kell

### Utasításszintű párhuzamosság

- Órajel frekvencia növelése???
- Párhuzamosság:
  - Utasításszintű
  - Processzorszintű

### Utasításszintű párhuzamosság

- Többfázisú csővezeték
  - Utasítás beolvasó egység
  - Utasítás dekódoló egység
  - Operandus beolvasó egység
  - Utasítás végrehajtó egység
  - Visszaíró egység

Jell: késleltetés, áteresztő képesség  
A fázisok állapota az idő függvényében (9 órajel ciklus)

### Szuperskalár architektúrák

- Kettős csővezeték (u pipeline, v pipeline – egyszerű egész és lebegőpontos művelet) a közös utasítás-beolvasó egységgel
- Szuperskalár processzor 5 funkcionális (ALU1, ALU2, LOAD, STORE, lebegőpontos egység) egységgel (1987, P4)

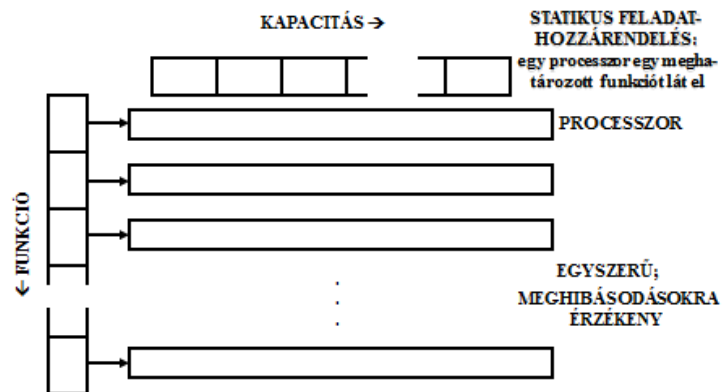
### Processzorszintű párhuzamosság

- Tömbprocesszorok (1972), adattömbök
- Vektorprocesszorok (1974), minden összeadás egyetlen csővezeték elven működő összeadó egység

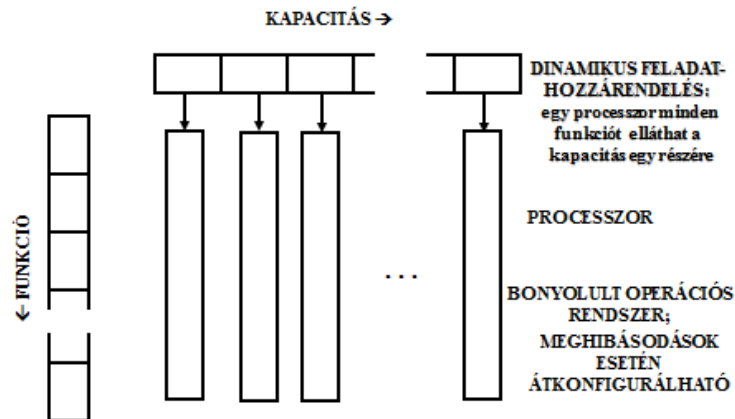
## Multiprocesszorok

- Egy sines multiprocesszorok
- Multiprocesszor lokális memóriákkal
- A mikroprocesszorok árának jelentős csökkenése lehetővé teszi, hogy azokat egy bonyolult rendszerben univerzális építőelemekként használjuk.
- **MULTIPROCESSZOROS RENDSZER:** egymástól független (rész)feladatok konkurens feldolgozását végző többprocesszoros rendszer.
- Két esemény konkurens, ha egyik sem tudja kauzálisan befolyásolni a másikat.

### Statikus feladat hozzárendelés



## Dinamikus feladat hozzárendelés



- **LAZÁN CSATOLT RENDSZER:**
  - üzenet kommunikáció,
  - minden processzornak saját operációs rendszere van.

*Egyszerű, lassú.*

- **SZOROSAN CSATOLT RENDSZER:**
  - kommunikáció közös erőforráson keresztül,
  - egyetlen operációs rendszer van.

*Bonyolult, gyors.*

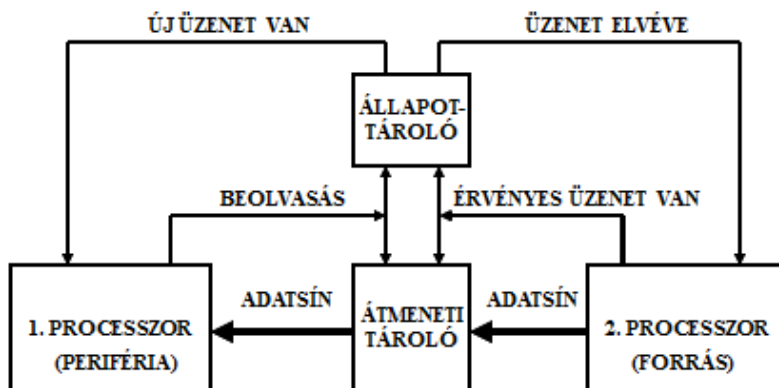
### MULTIPROCESSZOROS RENDSZEREK

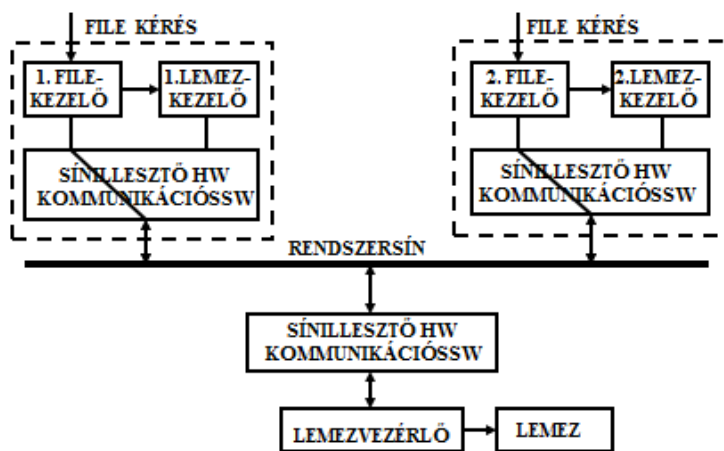
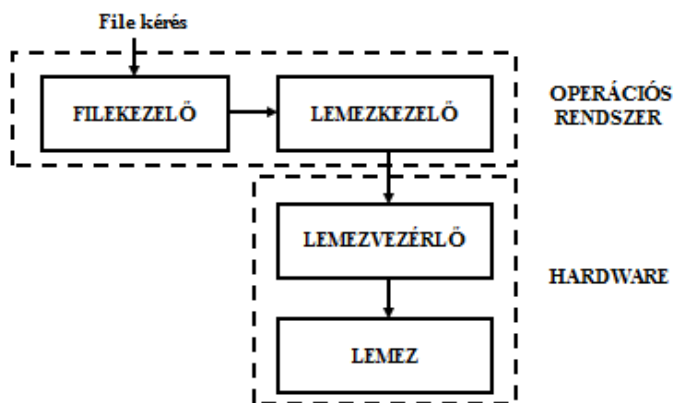
- **SZOROSAN CSATOLT RENDSZER:**
  - kommunikáció közös erőforráson keresztül,
  - egyetlen operációs rendszer van.

*Bonyolult, gyors.*

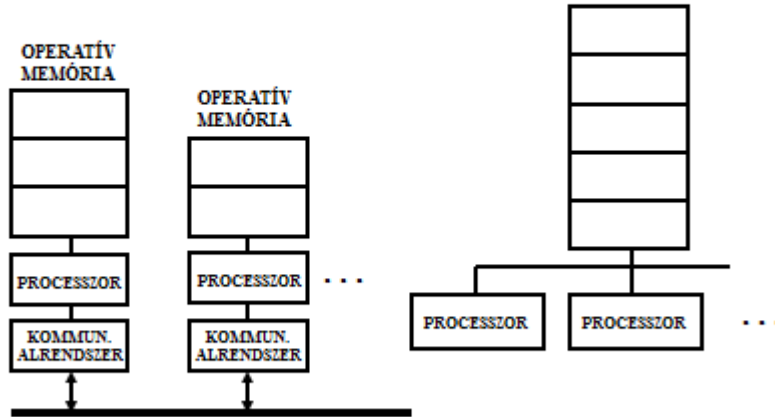
- Alapvető változatai:

- crossbar szervezés,
- multiport memóriára alapozott szervezés,
- rendszersínre alapozott szervezés.





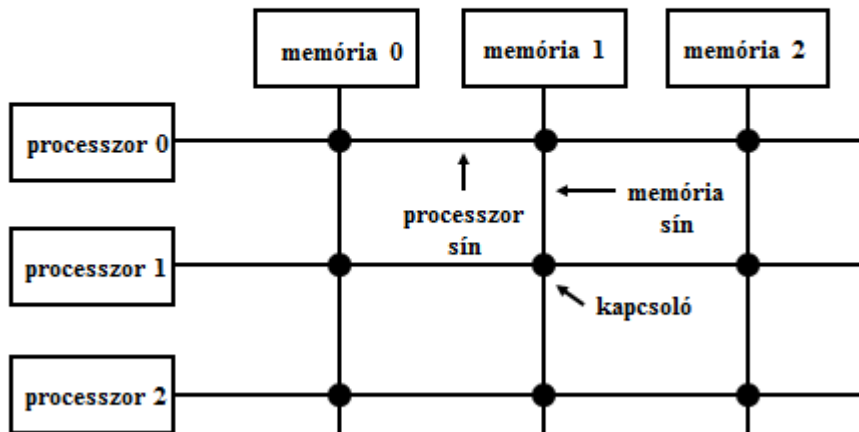
## VIRTUÁLISAN ELOSZTOTT MEMÓRIA



- A kommunikációs alrendszer a logikai pont-pont összeköttetést leképzi a fizikai átviteli közegre.
- A fizikai átviteli közeg korlátozott átviteli kapacitást biztosít, így **TORLÓDÁSVEZÉRLÉS** kell.
  - Minden processzor kap egy hitelértéket és akkor küldhet ki üzenetet, ha hitelértéke  $> 0$ .
  - Probléma: hitel elveszésének kezelése elvileg megoldhatatlan nagy rendszerekben.

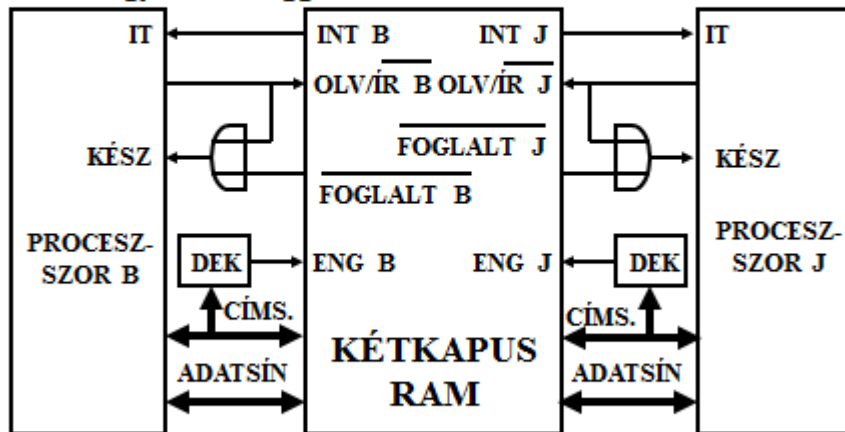
## SZOROSAN CSATOLT RENDSZEREK

### CROSSBAR SZERVEZÉS



### MULTIPOINT MEMÓRIA

- Egymástól független két bemeneti/kimeneti oldal.



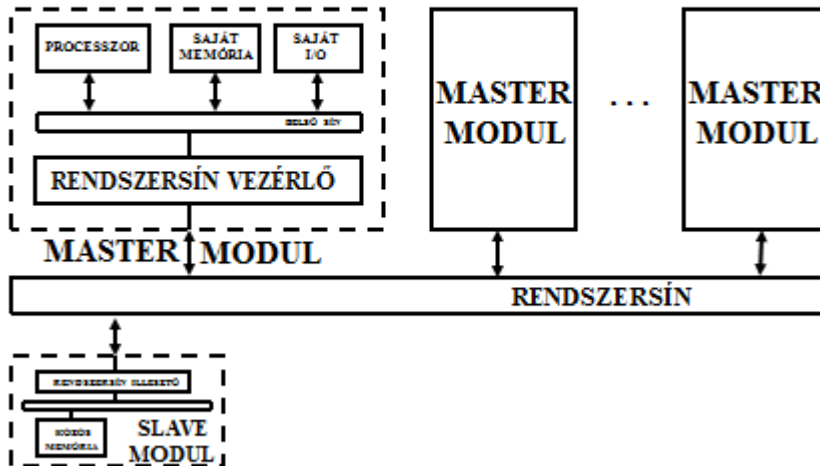
- **ELŐNY:** egyszerű.



- **HÁTRÁNYOK:**

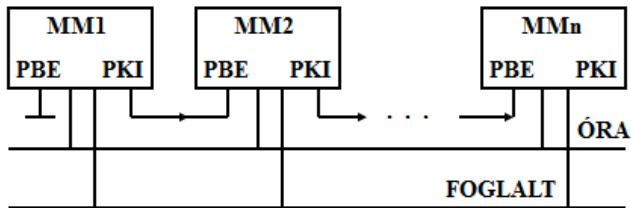
- csak kevés (2 - 4) port valósítható meg a bonyolult kapurendszer miatt;
  - ➔ csak kis rendszer építhető ki;
- kis memóriakapacitás valósítható meg a bonyolult kapurendszer miatt;
- nehéz átkonfigurálni.

## SZOROSAN CSATOLT RENDSZEREK - 8 RENDSZERSÍNRE ALAPOZOTT SZERVEZÉS



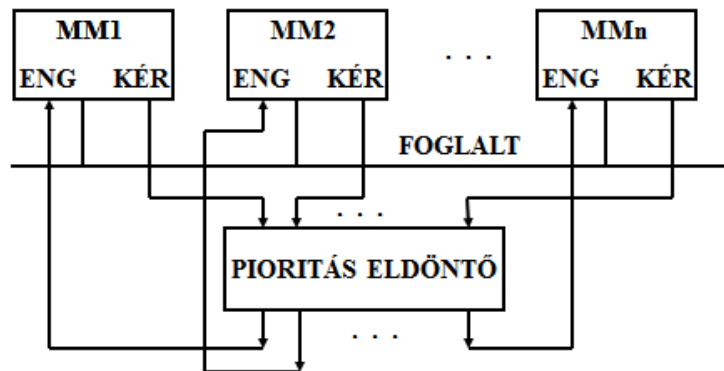
- **Master modul:**
  - magához ragadhatja a rendszersín vezérlését.
- **Slave modul:**
  - nem ragadhatja magához a rendszersín vezérlését;
  - közös erőforrás(oka)t tartalmaz.
- *A rendszer (elvben) tetszőlegesen bővíthető és átkonfigurálható.*
- Az egyes processzorokon futó folyamatok (a slave modulban lévő) közös erőforráson keresztül kívánnak kommunikálni egymással. Fel kell oldani a közös erőforrás használatáért folyó versengést.
- **DE:** a folyamatokat futtató master moduloknak előbb hozzá kell férniük a rendszersínhez.
- A master modulok versenyeznek a rendszersínhez való hozzáférés jogáért:
  - SOROS (daisy chain), vagy
  - PÁRHUZAMOS hozzáférés vezérlés.

## SZOROSAN CSATOLT RENDSZEREK - 11 SOROS HOZZÁFÉRÉS VEZÉRLÉS



- **ELŐNYE:**  
nagyon egyszerű.
- **HÁTRÁNYAI:**
  - A terjedési késleltetés miatt csak nagyon kis rendszerekben alkalmazható (meg kell várni a legbaloldalibbtól is az igény végigfutását a láncon).
  - Nehéz átkonfigurálni (vezetékét kell elvágni és összekötni), de a kis rendszerek (nagy megbízhatóság) miatt erre menetközben nincs szükség.
  - A rögzített prioritás miatt éhezés léphet fel.

## SZOROSAN CSATOLT RENDSZEREK PÁRHUZAMOS HOZZÁFÉRÉS VEZÉRLÉS



- A prioritás lehet rögzített (a prioritás eldöntő egyszerű kombinációs áramkör), vagy változó (pl. körbenforgó - round robin - ekkor hosszabb idő alatt egyenlő esélyt kap minden modul).
- A prioritás eldöntőt nagyobb rendszereknél elosztott módon valósítjuk meg (ellenkező esetben a rendszer kritikus része lenne).

## PERIFÉRIAKEZELÉSI MÓDSZEREK

- **ESZKÖZSZINTŰ KEZELÉS:** a perifériális eszköz fizikai sajátosságainak megfelelő illesztési felületet és utasításkészletet biztosítunk.
  - Kis rendszerek, beépített rendszerek esetén előnyös.
  - Jól kihasználhatók a processzor és a periféria sajátosságai.
- **LOGIKAI KEZELÉS:** általánosított illesztési felületeket és beviteli/kiviteli eljárásokat alkalmazunk.

### ESZKÖZSZINTŰ PERIFÉRIAKEZELÉS

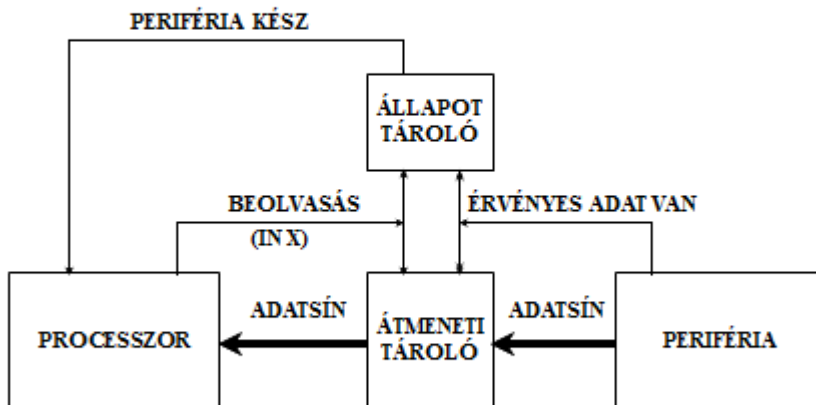
- **FELTÉTEL NÉLKÜLI BEVITEL/KIVITEL**

■ A processzor és a periféria nincs szinkronizálva.

*(A processzornak és a perifériának mindig rendelkezésre kell állnia.)*

■ Egyszerű IN és OUT utasítások.

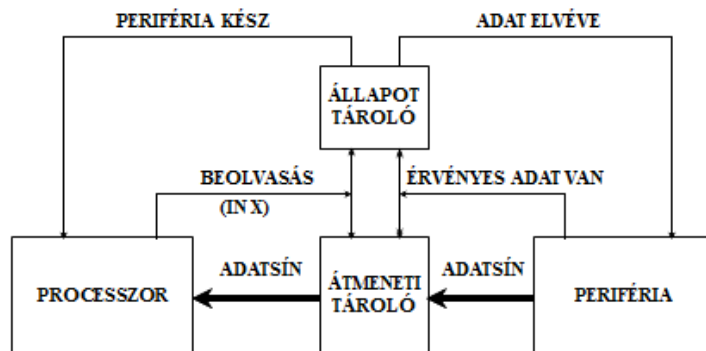
- **JELZŐBITES (FELTÉTELES) BEVITEL/KIVITEL**



■ Az együttműködésért kizárólag a processzor felelős.

*(A processzornak mindig rendelkezésre kell állnia.)*

- **SZEMAFOROS (FELTÉTELES) BEVITEL/KIVITEL**

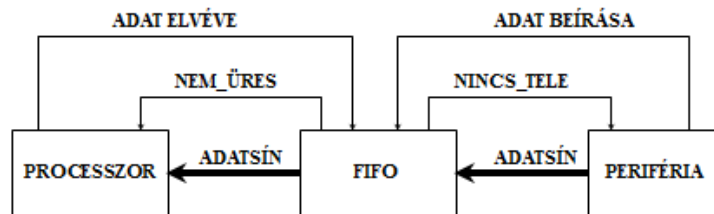


- **Kölcsönös szinkronizálás (kézfogásos üzemmód).**

*(Mindkét fél képes a másikat bevárni, illetve a másikat várakozásra kényszeríteni.)*

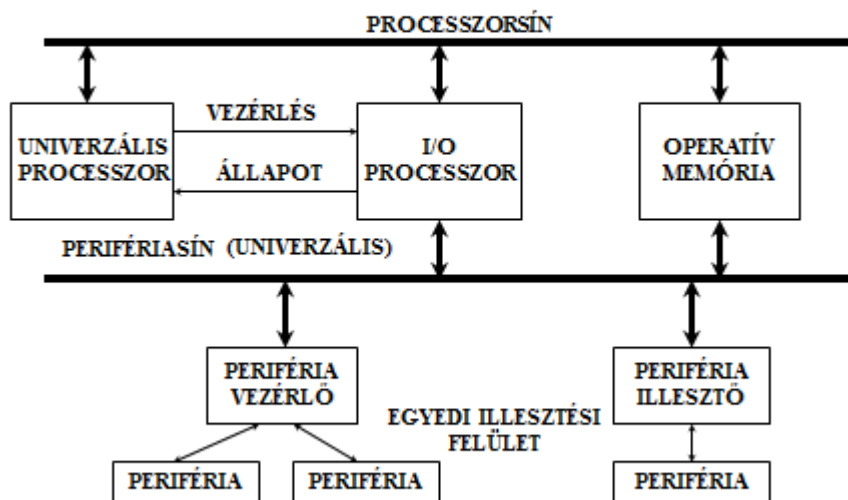
- **Nem minden periféria sebessége befolyásolható.**

## ESZKÖZSZINTŰ PERIFÉRIAKEZELÉS SZEMAFOROS BEVITEL/KIVITEL

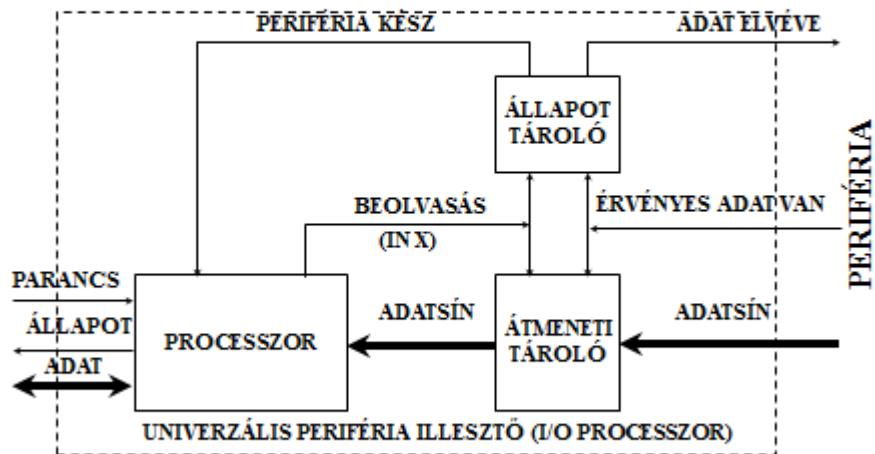


## LOGIKAI PERIFÉRIAKEZELÉS

- Sokféle perifériális eszköz:
  - ➔ általánosított beviteli/kiviteli eljárások és illesztési felület.
- I/O PROCESSZORra, ill. CSATORNÁra alapozott kezelés:
  - ➔ rögzített feladatú autonom modulok, felszabadítják a processzort a periféria részletes kezelése alól;
  - ➔ I/O processzor: átviteli műveletekre optimalizált + általános adatfeldolgozási képesség.







## KÖZVETLEN TÁROLÓ-HOZZÁFÉRÉS (Direct Memory Access, DMA)

