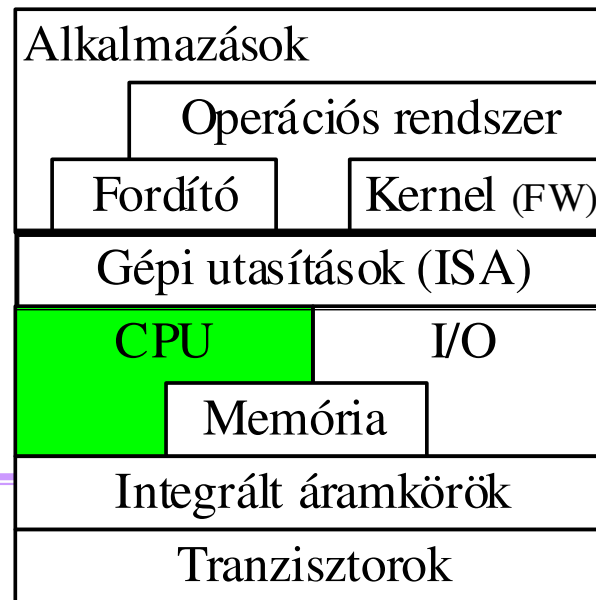


INFORMATIKA I.

BMEVIIIAB08

Számítógép architektúrák Többprocesszoros rendszerek

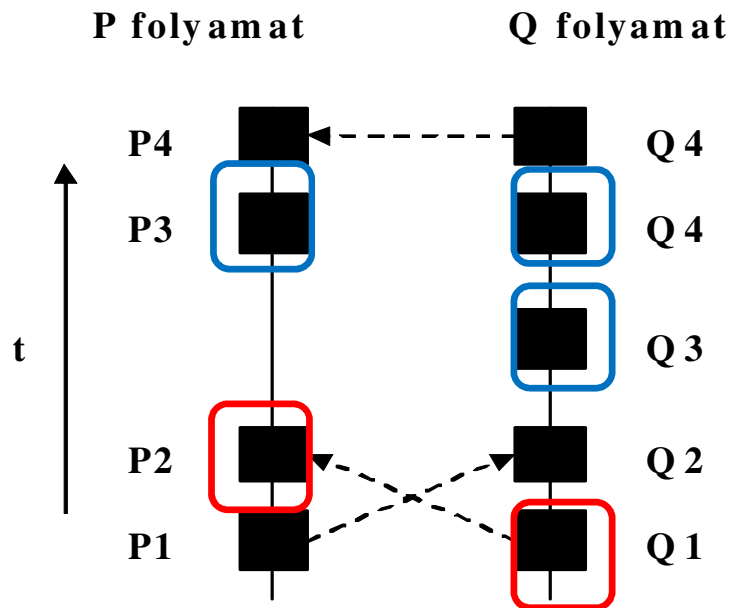


- *A mikroprocesszoros technika /és technológia/ lehetővé teszi, hogy egy nagyobb feladat kisebb elkülöníthető részeinek végrehajtását egy-egy külön processzorra bizzuk*
 - Kell egy mechanizmus, amelynek segítségével az egyes processzorok koordinálhatják működésüket a közös cél elérése érdekében

❖ **Multiprocesszoros rendszer**

Olyan struktúra, melyben ugyanazon rendszeralgorithmus egymástól független feladatainak konkurens végrehajtása folyik

Konkurencia



Két eseményt konkurensnek nevezünk, ha egyik sem tudja okszerűen befolyásolni a másikat

pl.: P_2 nem konkurens Q_1 -el

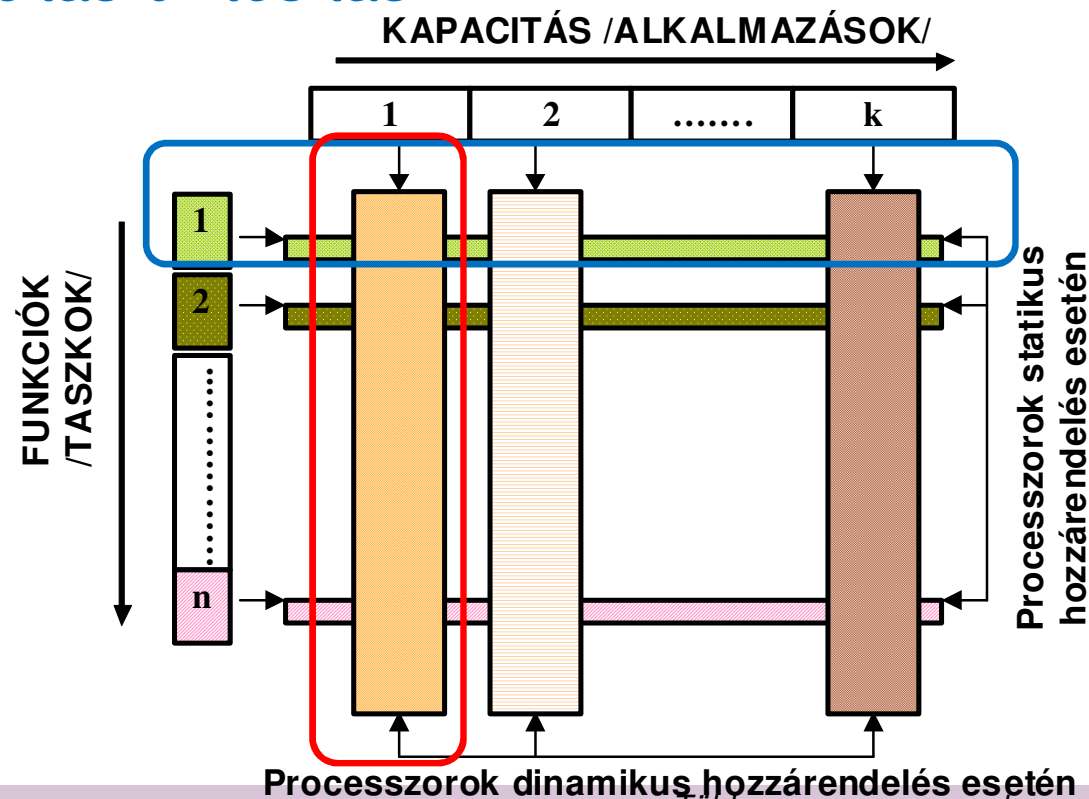
P_3 konkurens Q_3, Q_4 eseménnyel

- **A multiprocesszoros rendszereket osztályozhatjuk a feladat-hozzárendelés módja és a processzorközi kapcsolat jellege szerint**

- **Feladat-hozzárendelés**

- ◆ **Dinamikus feladat-hozzárendelés**
- ◆ **Statikus feladat-hozzárendelés**

- A multiprocesszoros rendszernek bizonyos funkciókat /szolgáltatásokat/ kell biztosítania a felhasználó /alkalmazások/ számára
- Lehetővé kell tennie minél több felhasználó kiszolgálását
→ kapacitás biztosítás



- A rendszerkapacitás egy részére *minden funkciót* elláthat egy processzor
Egy feladat-terhelési viszonyoktól függően *bármelyik processzornak* odaadható →
Logikailag egyenértékűek a processzorok
- Előnyök ↔ Hátrányok
 - 👍 Egy processzor meghibásodása a rendszernek csak egy részére terjed ki
Megfelelő hibaterjedés védelem esetén
 - 👍 A processzorok teljesítőképessége jól kihasználható
Megfelelő terheléselosztó algoritmus esetén
 - 👎 A processzorok teljesítőképessége *korlátozza* a megvalósítható funkciókat / *szolgáltatásokat* /
 - 👎 *Bonyolult szoftvert* → sok taszkváltást igényel

Statikus feladat-hozzárendelés

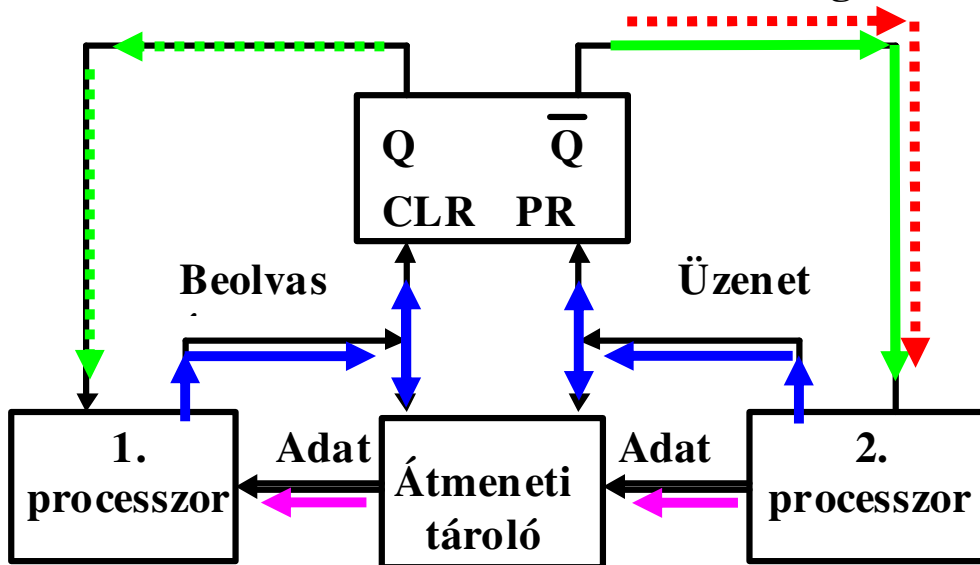
- ❑ Egy processzor **egy meghatározott** szolgáltatást /*funkciót*/ valósít meg az egész rendszer számára
Pl.: I/O processzor
- Előnyök \leftrightarrow Hátrányok
 - 👍 Egyszerű szoftver \rightarrow *minimális a kölcsönhatás az egyes processzorok feladatai között*
 - 👍 Elmarad a taszkváltással összefüggő adminisztráció
 - 👍 **Egyszerűbb** az operációs rendszer ütemezése
 - 👍 A processzorok felépítése a funkcióhoz **optimalizálható** \rightarrow *jobb teljesítmény*
 - 👎 Az egyes processzorok **terhelése jelentősen eltérhet**
 - 👎 **Érzékenyebb** a meghibásodásra \rightarrow *feladatát általában csak ugyanolyan típusú processzor veheti át*

Processzorközi kapcsolat

- Az együttműködés megvalósításához mind *hardver*, mind *szoftver* szinten *kapcsolatot* kell kialakítani az egyes *processzorok* között.
 - *Lazán* csatolt rendszerek
 - *Szorosan* csatolt rendszerek
- *Lazán* csatolt rendszerek
 - A processzorközi *kapcsolat* *üzenetorientált*
→ *Lassú és kevés üzenet*
 - Az egyes processzorokon *különálló* operációs rendszerek futhatnak
 - A processzorokat *kommunikációs alrendszerekből* megvalósított csatornák kötik össze.
 - A logikai és fizikai közeg függ
 - A rendszer *nagyságától*
 - Térbeli* elhelyezkedésétől.

- *Elv: az üzenet átvitelének idejére a forrás processzor a fogadót perifériájának tekinti.*
 - *Kis rendszernél, kevés információ átvitelére*
Pl.: hardverre alapozott szemafor

Pont-pont közötti feltételes bevitel elve
 Üzenet kész Üzenet elfogadása



Előnyök ← → Hátrányok

👍 **Egyszerű**

👎 **Lassú**

→ *DMA segíthet*

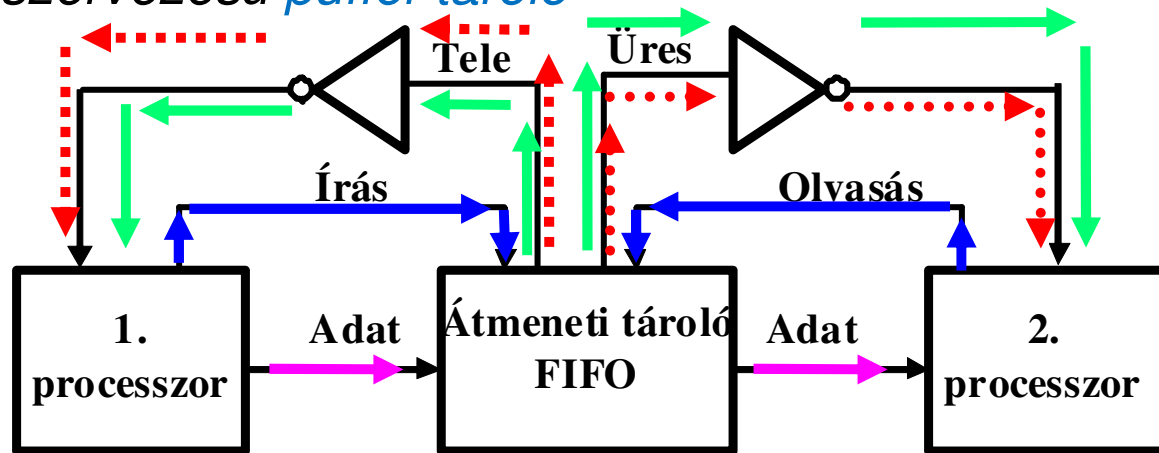
👎 **Merev**

→ *Hálózatok segíthetnek*

👎 **Nagyobb rendszereknél jelentőssé válhat az operációs rendszerek együttműködésének biztosítása**

- 👉 **Probléma:** a forrás és a nyelő processzorok között nagymértékben ingadozik az adatátvitel sebessége, ez függ az egyéb feladataiktól is

→ Megoldás : Mindkét oldal felé külön szemafor és szervezésű *puffer tároló* FIFO



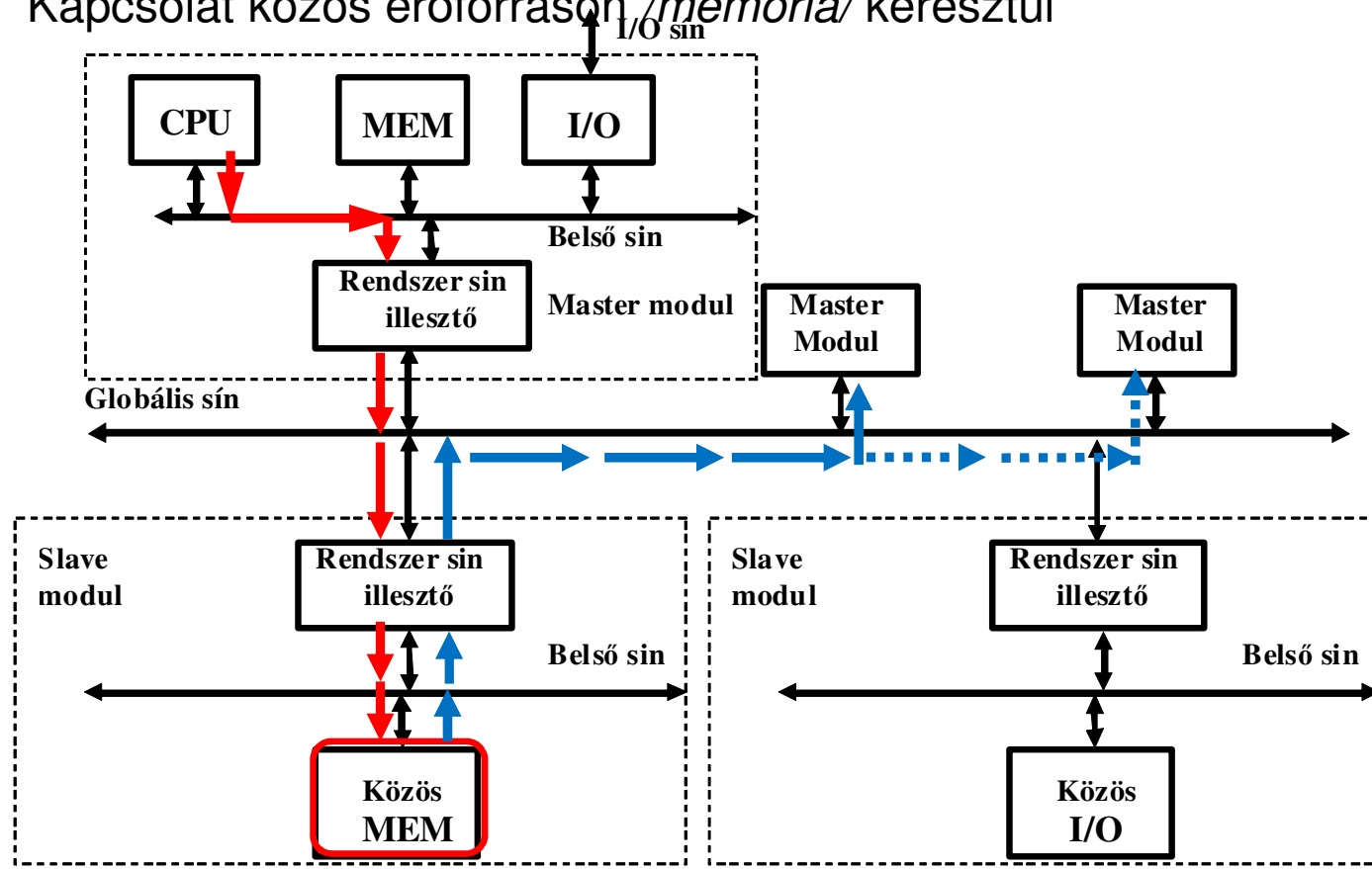
Van hely az
adat bevitelére

Van kiolvasandó
adat a FIFO-ban

✦ 👍 Számítógép hálózatok → lokális és globális →
Kommunikációs processzorok 👍

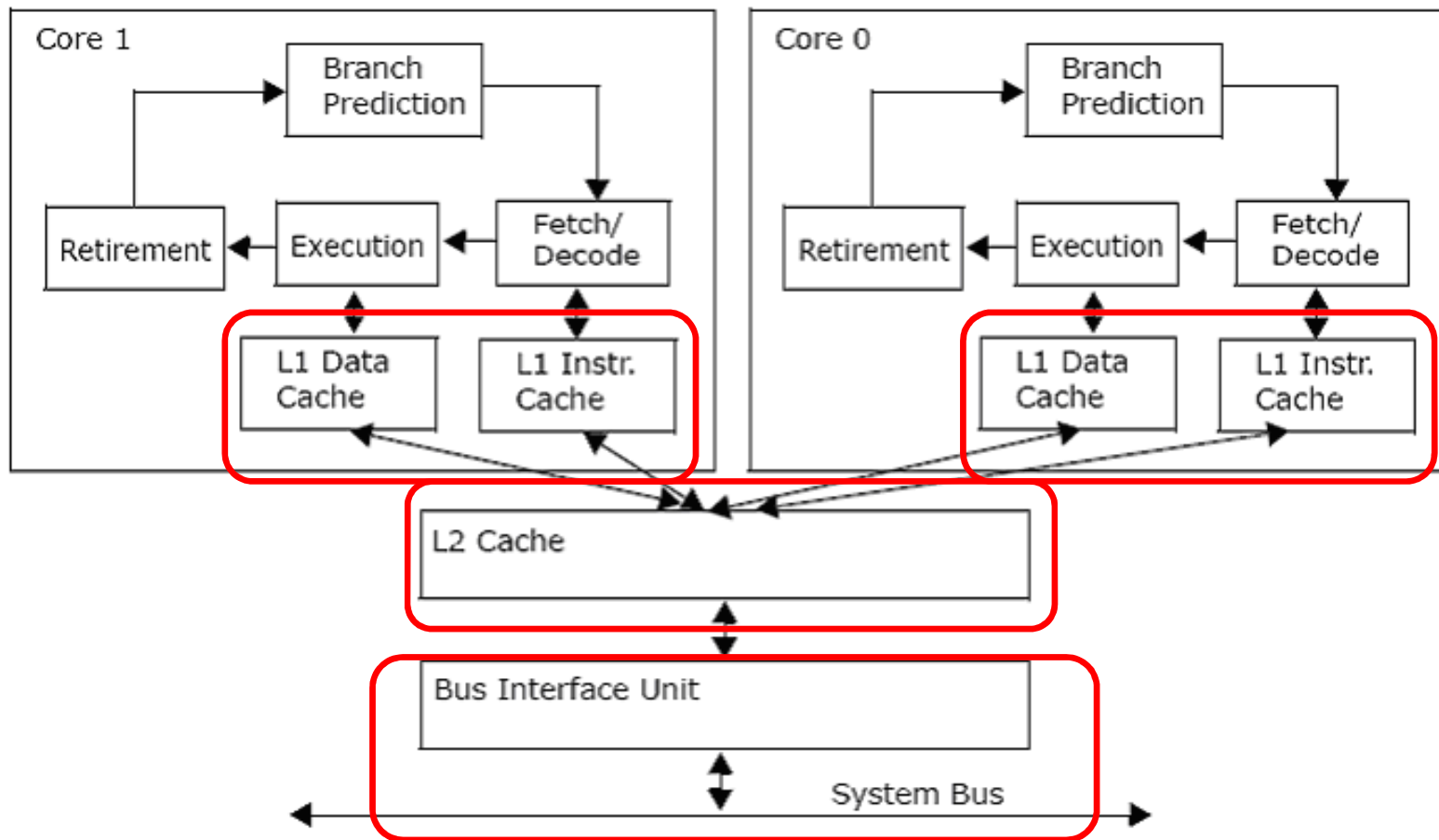
Szorosan csatolt rendszerek

- A processzorközi kapcsolat **közös erőforráson** keresztül valósul meg
- Közös az operációs rendszer**
 - Kapcsolat közös erőforráson */memória/* keresztül

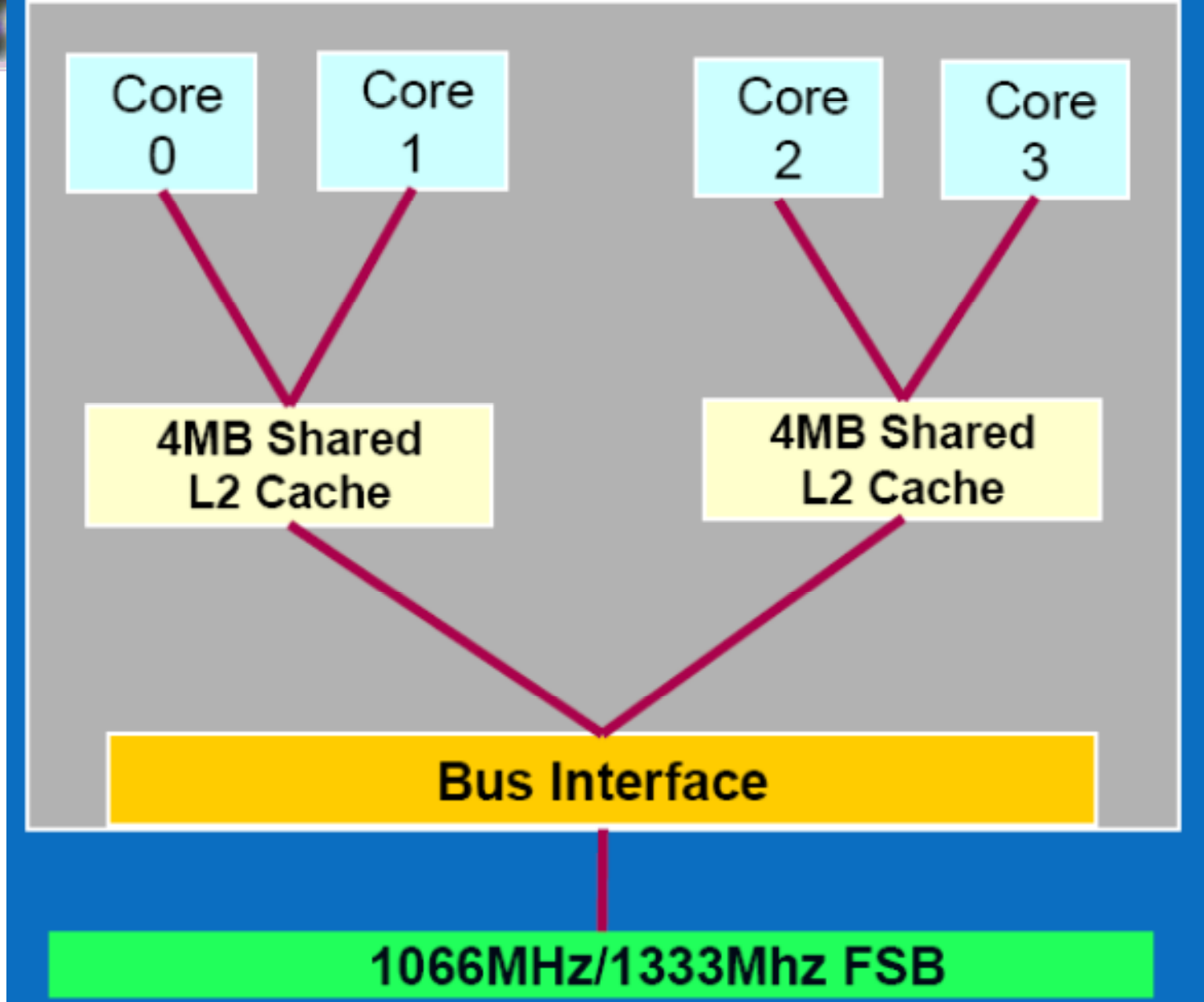


Kétsínés multiprocesszoros struktúra

kétmagos multiprocesszoros struktúra



Esettanulmány - Pentium négymagos processzor



Típus	mag	szál	cache	órajel
Core i3	2	4	3MB	1,6-2,4GHz
Core i5	2-4	4	3-6MB	2,7-3,1GHz
Core i7	2-4	4-8	4-6MB	1,7-2,9GHz
Xeon E7	10	20	30MB	2,4GHz

Mag → **Core** → **AS** /Architectural state / → ~**CPU** → **Hw**

Szál → **Thread** → **SW**

❑ A processzorközi kommunikáció

- Logikai szinten

Folyamatok közötti *adat-és vezérlés kommunikáció*

→ Postafiók elv

A mesterek üzeneteket helyezhetnek el és olvashatnak ki egy közös memóriaterületről → postafiók → Az adatok elhelyezését - kiolvasását megfelelő szemaforokkal koordinálják

- Előnyök ↔ Hátrányok
- 👍 A megoldást tekintve a fogadó modul szempontjából érdektelen ki tölti be az adatokat
- 👍 Fordítva ez igaz a küldő modulra is
- 👍 A többi modul a rendszersín művelet alatt dolgozhat a saját erőforrásaival
- 👎 A rendszersín szűk keresztmetszetté válhat

- Fizikai szinten
 - A mester modul rendszersínhez → /*közös erőforrás*/ való hozzáférését kell koordinálni → *ARBITER*
 - Központosított → *centralizált*
 - Elosztott → *decentralizált*
 - Soros ↔ Párhuzamos megoldású
 - A logikai szintű szinkronizáció /*szemaforkezelés*/ megvalósításához hardvertámogatás kell
 - *Kölcsönös kizárás* megvalósítása
- Mester_k számára a postafiók szemaforjának ellenőrzése és visszaírása idejére ki kell zárni mester_k modult /→ más mestereket/ annak használatából*
- RMW típusú utasítás → *elemi oszthatatlan művelet*
 - BUS LOCK, BUS UNLOCK
 - Szoftver+hardver megoldás /*P és V primitív*/