



Bevezetés a számítástechnikába

#04 – Adattárolás, adatkommunikáció

2023. október 2–6.

Siklósi Bálint <siklosi.balint@itk.ppke.hu>
Naszlady Márton Bese <naszlady@itk.ppke.hu>

#04/1 – Adattárolás

Memóriák

- ▶ *volatile*: folyamatos energia kell a tároláshoz, ha nincs áram alatt, akkor felejt, sérül az adat
pl. élőlények emlékezete (ha nem eteted → meghal → nem nyered vissza az emlékeit)
- ▶ *nonvolatile*: nincs folyamatos energiaigénye, akkor is benne marad az adat, ha nincs áram alatt
pl. barlangrajz (ha nem eteted → nem érdekli → megmarad az infó energia nélkül is)

Memóriák

- ▶ *Dinamikus memória* (dynamic): folyamatosan frissíteni kell a tárolt értéket, különben elsikkad
pl. konyhakert (ha nem gyomlálgom, akkor eltűnik a paradicsom)
pl. bevszámtech tudás (ha nem gyakorlom, elfelejtem)
- ▶ *Statikus memória* (static): nem kell folyamatosan frissíteni és visszaújítani az értéket
pl. őserdő (több ezer éve ott van az a fa)
pl. saját neved (elég nehéz elfelejteni, úgy beléd van égetve)

Memóriák

- ▶ *Read-Only Memory* (ROM): csak olvasható nonvolatile memória (egyszer, vagy esetleg nehézkes törlés után többször írható)
 - ▶ *Programmable ROM*: speciális eszközzel gyártás után programozható, egyetlen alkalommal
 - ▶ *Erasable PROM*: törölhető (pl.: UV fénnel) és újraírható más adattal
 - ▶ *Eletronically Erasable PROM*: elektronikusan törölhető és néhány ezerszer (milliószor) újraírható
- ▶ *Random Access Memory* (RAM): bármely címet ugyanannyi idő elérni ellentétben pl. a merevlemezzel a legtöbb volatile memóriatípust is valamilyen RAM-nak hívják

Memória-hierarchia

Probléma: a gyors memóriák drágák és kicsik, a nagy memóriák nem férnek el közel a CPU-hoz

Megoldás ötletek:

- ▶ Alkalmazzunk sokféle memóriát
- ▶ A software válogasson közöttük (registerek, fő memória, merevlemez)

Memória-hierarchia

Különböző technológiák, gyorsabbtól a lassabbig:

- ▶ Register: CPU chipen rajta, fordító dönti el általában, hogy mi kerül regiszterekbe
- ▶ L1, L2, ... cache (gyorsítótár): egyre nagyobb és messzebb levő DRAM memóriák a hardware dönti el, hogy mi kerül bele
- ▶ fő memória: általában valamilyen DRAM operációs rendszer ide cache-eli a háttértárat
- ▶ Solid State Drive (SSD): gyors nonvolatile (perzisztens) tároló
- ▶ Hard Disk Drive (HDD): lassabb, mágneslemezes tároló

Háttértárak

Perzisztens tár (nonvolatile): kikapcsolás után megmarad az adat

Jellemzők:

- ▶ sávszélesség
- ▶ hozzáférési idő
- ▶ tárolókapacitás
- ▶ élettartam
- ▶ ár

(lásd: memóriahierarchia)

Optikai tárolók

Egy lemez egy hosszú reflektív spirál mentén tárolja az adatokat.

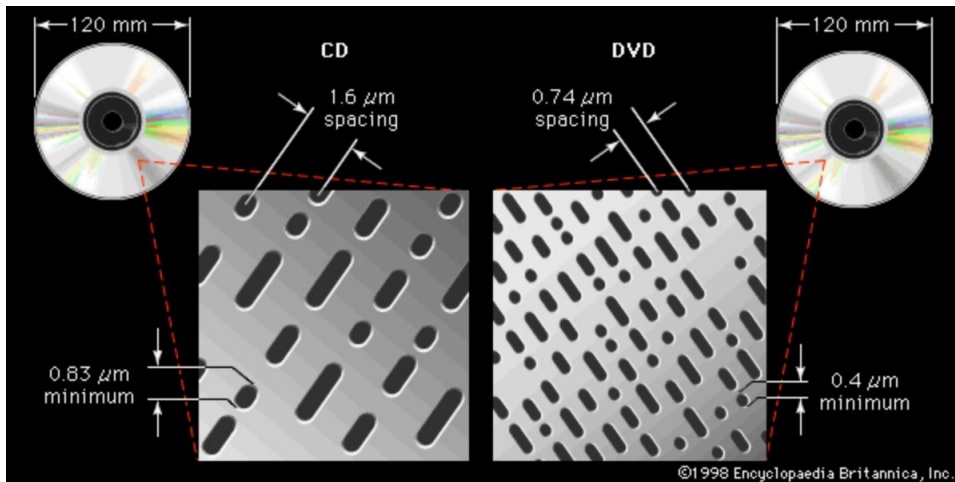
Az adatok az anyag felületén (pit (0) és land (1)) tárolódnak, kiolvasáskor a felpörgetik a lemezt, és egy lézersugárral megvilágítják.

A land visszatükrözi a sugarat, a pit nem (ön-interferál).

Típusai:

- ▶ Technológiától függően: CD, DVD, BluRay
- ▶ Írhatóságtól függően: csak olvasható (ROM), egyszer írható (R), újraírható (RW)
- ▶ Alkalmazástól függően: zene, film, program, képek

Optikai tárolók



Merevlemez

Hard Disk Drive (HDD)

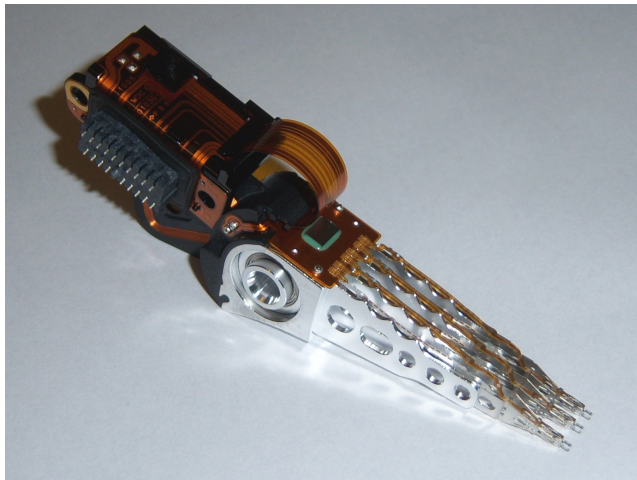
Mágnesezhető, forgó lemezekre írja az adatot.

Ellentétben az optikai lemezekkel, koncentrikus körökre. (Egyenletes sebességgel lehet forgatni.)

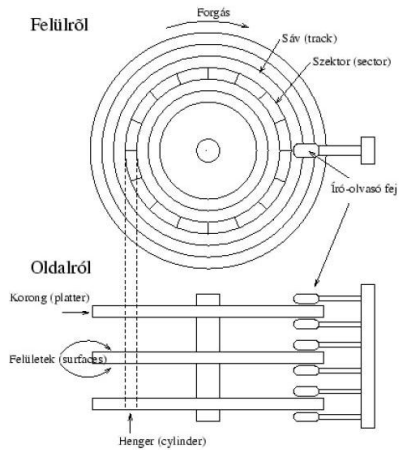
Az író/olvasó fej a lemezek között mozog néhány nanométerre légpárnát generál maga alá.

Régebben a cylinder-fej-szektor (cylinder-head-sector) alapján címezték (megfelelően az írási procedúrának) manapság inkább logikailag, folytonosan (Logical Block Addressing)

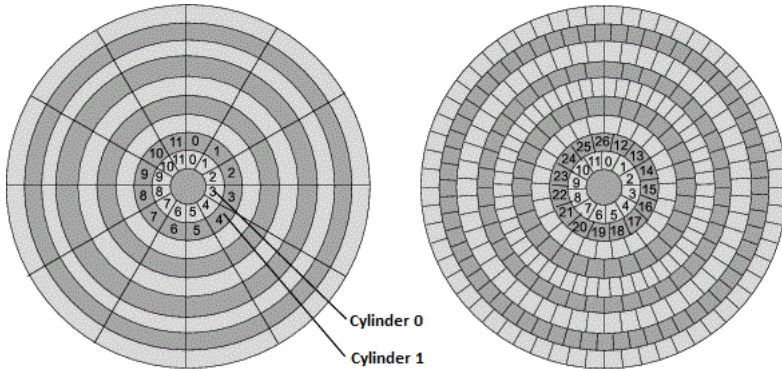
Merevlemez



Merevlemez



Merevlemez



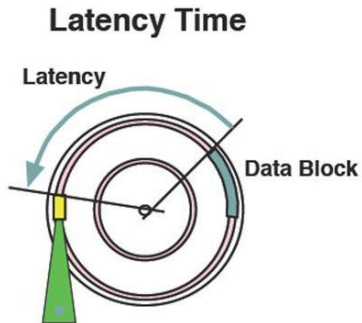
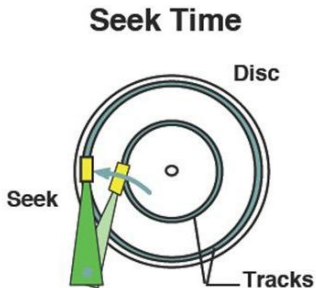
CHS Addressing

LBA Addressing

Merevlemez

Elérési idő áll:

- ▶ **keresési időből** (seek time): oda kell vinni a fejet arra a rádiusra
- ▶ **forgatási időből** (latency time): arra a szögre kell forgatni a lemezt



Flash alapú háttértárak

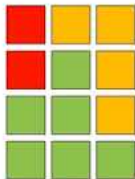
Floating-gate tranzisztor technológián alapuló random access, perzisztens tár.

Az újraírás fizikailag terheli az eszközt

Erre megoldás, hogy az írásokat egyenletesen elosztjuk az eszközön (wear levelling) Pl.: pendrive, Solid State Drive (SSD)

Solid State Storage – Wear Levelling

Without Wear
Levelling






With Wear
Levelling
(worst case)



With Wear
Levelling
(best case)



-  Nearing end of usable life
-  Early to Midway through usable life
-  Early in it's usable life



Egyéb technológiák

- ▶ mágnesszalag: hosszú távú (évtizedek) adattárolásra, amit ritkán olvasnak
- ▶ memrisztor alapú technológiák:
a memrisztor egy nemlineáris passzív elem, amelynek ellenállása függ a múltbeli töltésáramlástól
az ellenállás az állapot
- ▶ DNS alapú adattárolás

#04/2 – Particionálás

Partícionálás

Egy háttértáron belül lehetőség van több területet, **partíciót** kialakítani

- ▶ több operációs rendszernek
- ▶ többféle fájlrendszert alkalmazni
- ▶ külön partíció az adatoknak és rendszernek

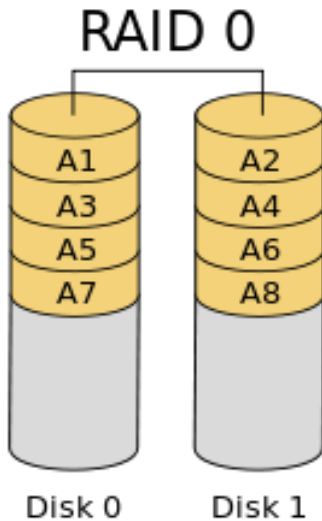
Két nagy formátum van a partíciók kialakítására: MBR és GPT

Redundant Array of Independent (Inexpensive) Disks (RAID)

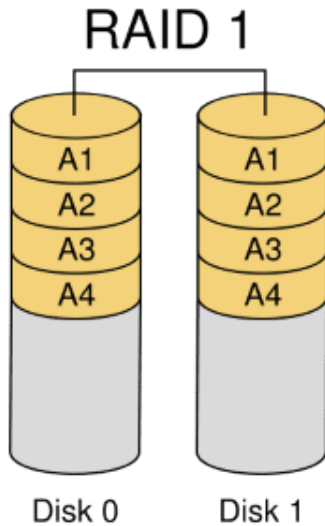
Redundánsan tárol több háttértáron, hogy néhány elromlása esetén se történjen adatvesztés

- ▶ **RAID0**: csíkozás (striping)
nem redundáns, csak a sávszélességet növeli
- ▶ **RAID1**: tükrözés:
két eszköz ugyanazt az adatot tartalmazza
- ▶ **RAID5**: blokkonkénti csíkozás elosztott paritással
ellentétben az előzőekkel, a paritás nem egy háttértáron van, hanem elosztva az összesen
- ▶ **RAID10**: csíkozás és tükrözés
két háttértár is kieshet egyszerre

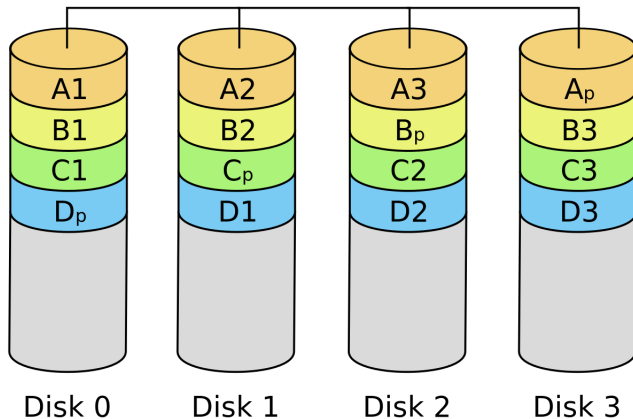
RAID



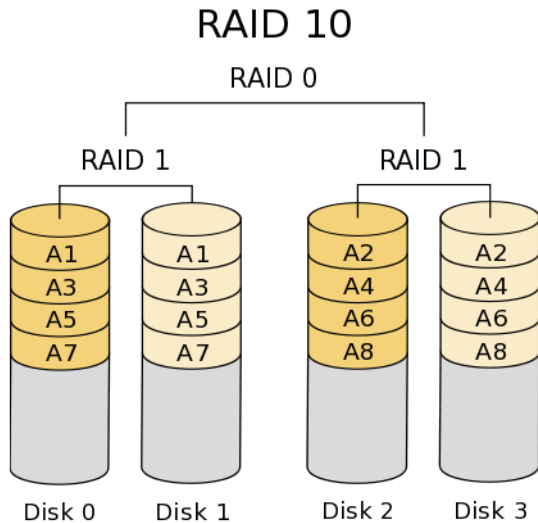
RAID



RAID 5



RAID



#04/3 – Fájrendszer

Fájlrendszerek (file systems)

A fájlrendszerek feladata, hogy a rendelkezésre álló helyen elhelyezze az adatokat: hogyan reprezentálja, milyen stratégiával hozza létre és törölje stb., és adminisztrálja a fájlokat (metaadatok).

A felhasználó felé ez fájlok egy (Linux, Mac) vagy több (Windows) fa struktúráját jelenti. A belső csomópontok a könyvtárak vagy mappák.

Metaadatok (adatok az adatokról) lehetnek:

- ▶ A fájl tulajdonosa
- ▶ Készítés, hozzáférés, módosítás dátuma
- ▶ Hozzáférési jogosultságok
- ▶ Kiterjesztett attribútumok, pl.: előkép-ikon, cím, összefoglaló
ezeket nem értelmezi a fájlrendszer, csak tárolja

Néhány népszerű fájlrendszer

- ▶ NTFS (NT file system / New Technology File System)
 - ▶ Max. fájl méret: 256 TB
 - ▶ Jogosultság- és tömörítéskezelés
 - ▶ Megbízhatóság (áramszünet, hibás szektor)

Néhány népszerű fájlrendszer

- ▶ NTFS (NT file system / New Technology File System)
 - ▶ Max. fájl méret: 256 TB
 - ▶ Jogosultság- és tömörítéskezelés
 - ▶ Megbízhatóság (áramszünet, hibás szektor)
- ▶ Ext4
 - ▶ Max. fájl méret: 16 TB
 - ▶ Linux rendszerhez fejlesztették ki
 - ▶ Könnyű menedzselhetőség (pl. üzem közbeni bővítés)

Néhány népszerű fájlrendszer

- ▶ NTFS (NT file system / New Technology File System)
 - ▶ Max. fájl méret: 256 TB
 - ▶ Jogosultság- és tömörítéskezelés
 - ▶ Megbízhatóság (áramszünet, hibás szektor)
- ▶ Ext4
 - ▶ Max. fájl méret: 16 TB
 - ▶ Linux rendszerhez fejlesztették ki
 - ▶ Könnyű menedzselhetőség (pl. üzem közbeni bővítés)
- ▶ File Allocation Table (FAT)
 - ▶ Max. fájl méret: 4 GB
 - ▶ Egyszerű, régóta használt fájlrendszer
 - ▶ Sok oprendszer támogatja

Töredezettség (file fragmentation)

Általában sok fájlt tárolunk, és ezek mérete folyamatosan változik.

Külső töredezettség

A fájl nem fér bele az előre lefoglalt vagy rendelkezésre álló szabad helyre. Ezért a fájlt több darabban tároljuk.

Probléma: nehéz megoldani, hogy fizikailag egymás után következő blokkokban tároljuk a fájl belüli adatokat. Ez nem RAM esetén fontos az elérési idő miatt.

Belső töredezettség

Ha a szabad hely töredezik, mert a fájl nem tölti ki a neki lefoglalt helyet, és a fennmaradó lyukat már nem tudjuk mivel betömni (mert a blokkméret egységes).

Probléma: hiába van sok apró szabad helyünk, ezek nem használhatók fel ténylegesen szabad helyként.

Külső töredezettség



Külső töredezettség



Külső töredezettség



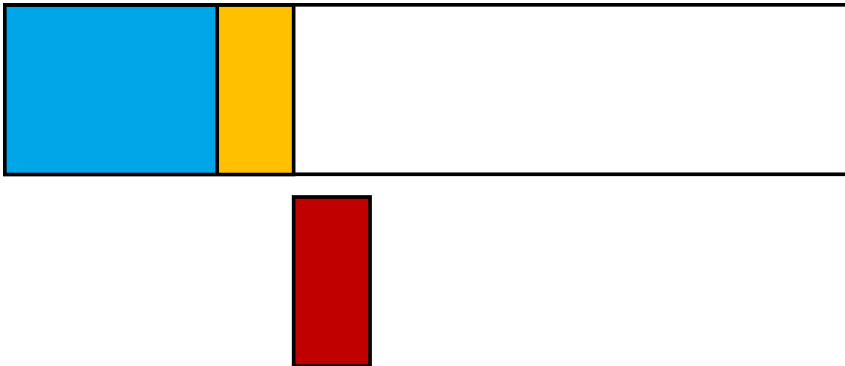
Külső töredezettség



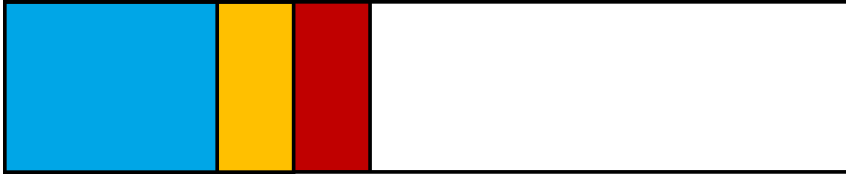
Külső töredezettség



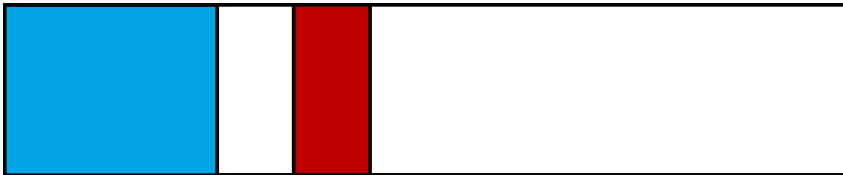
Külső töredezettség



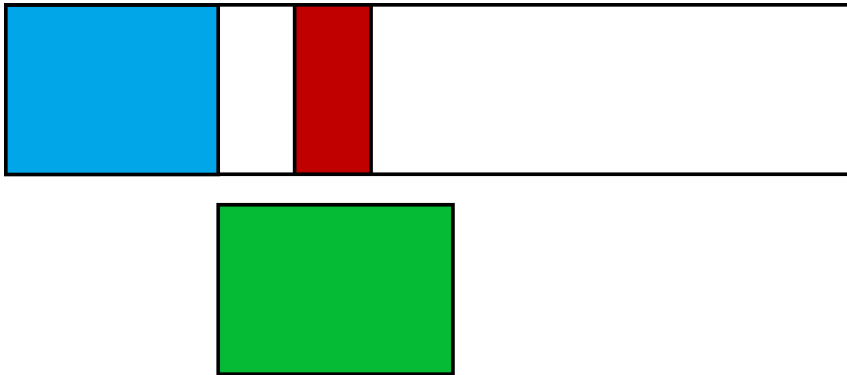
Külső töredezettség



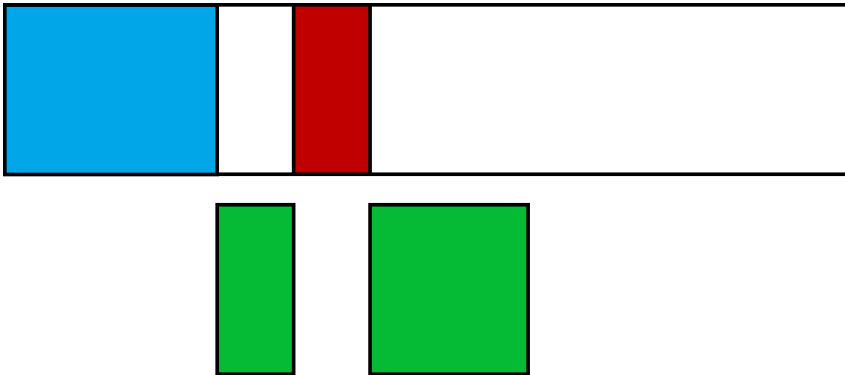
Külső töredezettség



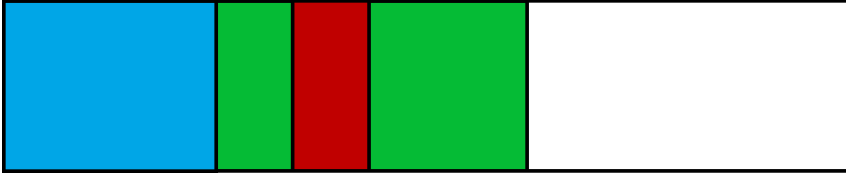
Külső töredezettség



Külső töredezettség



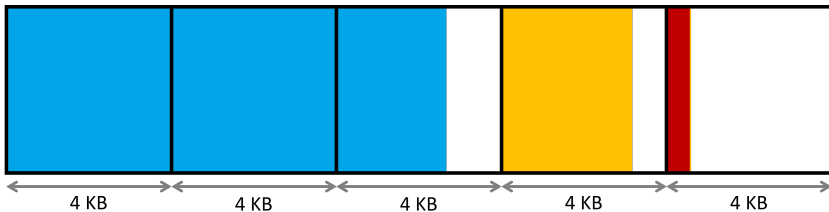
Külső töredezettség



Külső töredezettség



Belső töredezettség



#04/4 – Adatkommunikáció

Soros és párhuzamos adatátvitel

Az adat (pl. 8 bites szám) egyes bitjei egyszerre vagy időben eltérően érkeznek meg?

Párhuzamos átvitel

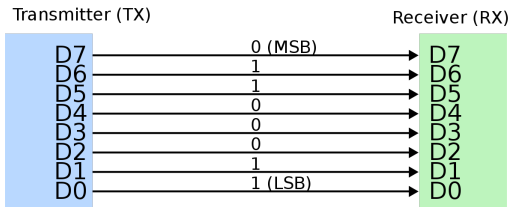
- ▶ annyi darab vezeték, ahány bit → helyigény
- ▶ egyszerre közlekedik rajta az adat → gyors
- ▶ pl. memória és CPU között

Soros átvitel

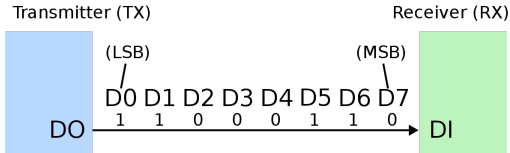
- ▶ kevesebb vezeték, mint ahány bit → kis helyen elfér
- ▶ a bitek egymást követve érkeznek meg → lassabb
- ▶ pl. USB kábel

Soros és párhuzamos adatátvitel

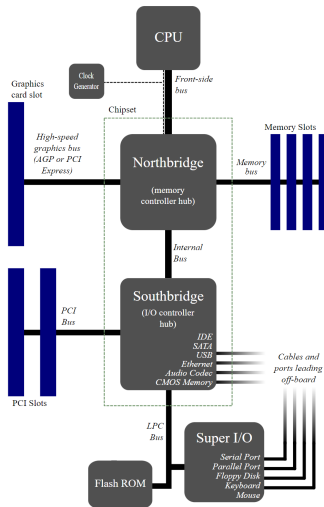
Parallel interface example



Serial interface example



Egy modern alaplap blokkvázlata



Különböző komponensek közötti összeköttetés

- ▶ vezérlőbusz: kommunikáció lefolytatása
- ▶ címbusz: üzenet célja
- ▶ adatbusz: adat megy rajta

A kommunikáció valamilyen protokoll alapján történik:

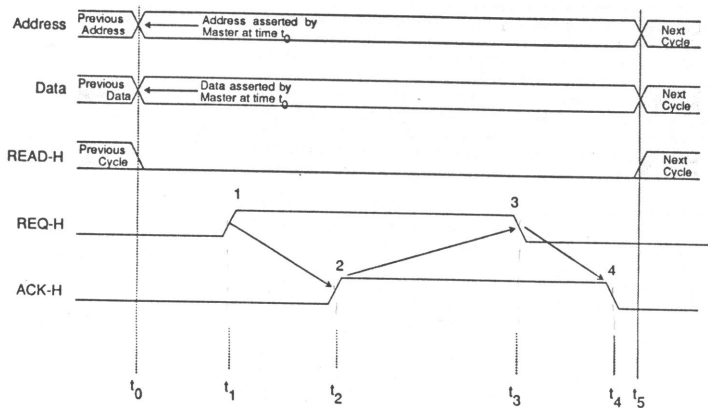
- ▶ aszinkron
- ▶ szinkron

A kommunikációban a felek lehetnek:

- ▶ egyenrangúak
- ▶ alá- és fölérendelt viszonyban

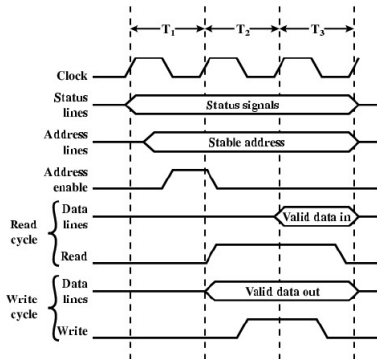
Aszinkron kommunikáció

- ▶ Nincs közös órajel, vezérlőjelekkel történik az időzítés. (Handshake)
- ▶ Előny: egyszerűbb, nem kell órajel
- ▶ Hátrány: késleltetések (meg kell várni, amíg a jel eljut a címzetthez)



Szinkron kommunikáció

- ▶ A kommunikációt közös órajel szinkronizálja.
- ▶ Órajel: valamilyen periodikus jel, meghatározott frekvenciával és szélességgel
- ▶ Előny: gyorsabb, nincs szükség handshake-re
- ▶ Hátrány: az órajelet a leglassabb eszközhöz kell igazítani



VÉGE



PÁZMÁNY

Pázmány Péter Katolikus Egyetem
Információs Technológiai és Bionikai Kar