



HÁLÓZATI RENDSZEREK
ÉS SZOLGÁLTATÁSOK
TANSZÉK



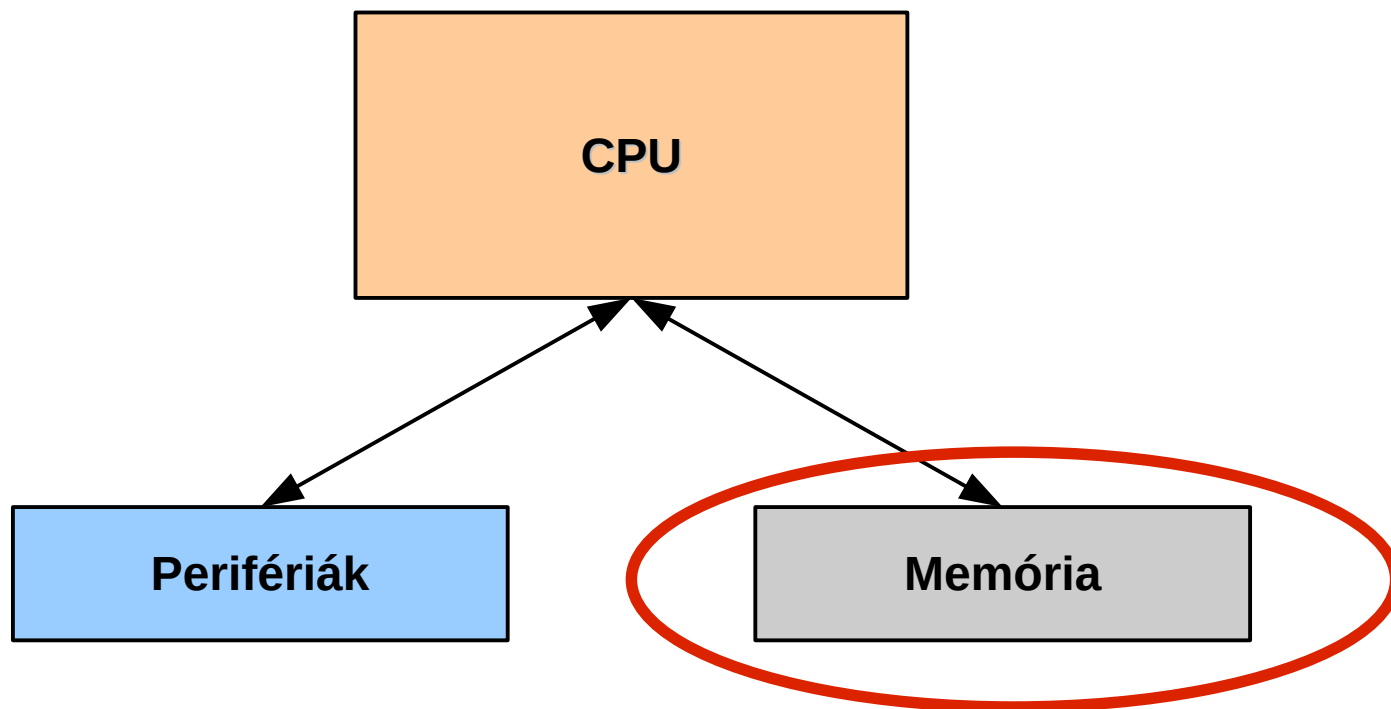
Budapest,
2025.03.11.

SZÁMÍTÓGÉP ARCHITEKTÚRÁK

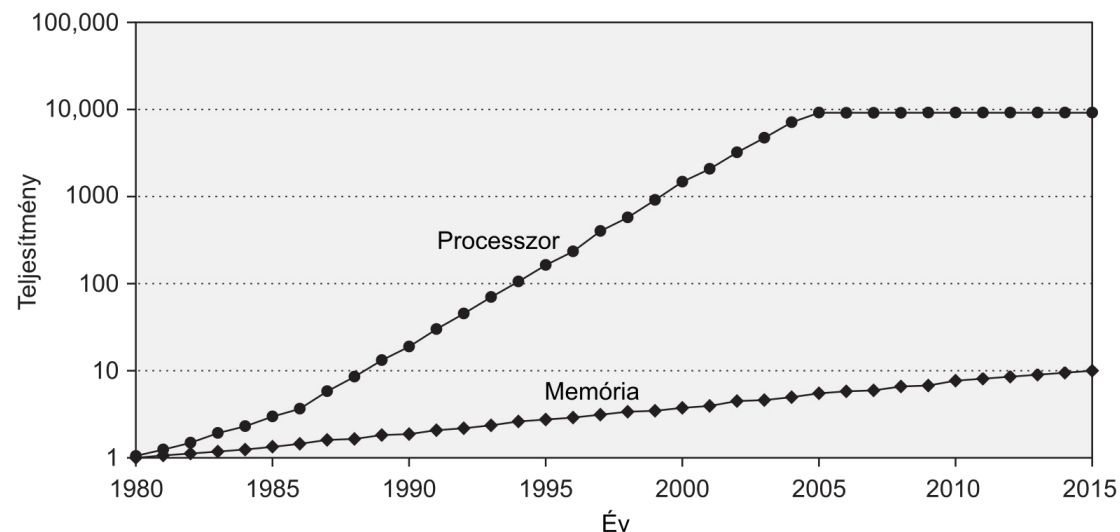
Memória technológiák

Horváth Gábor, Belső Zoltán

BME Hálózati Rendszerek és Szolgáltatások Tanszék
ghorvath@hit.bme.hu, belso@hit.bme.hu



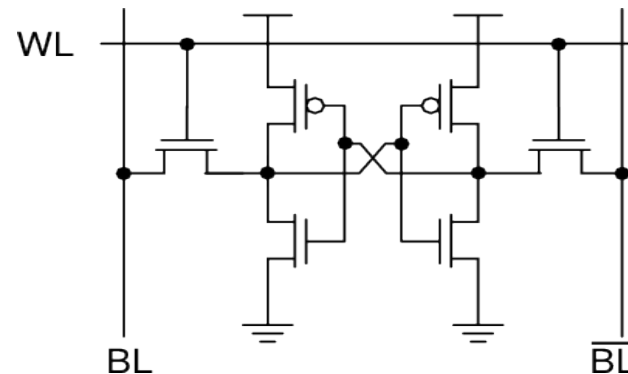
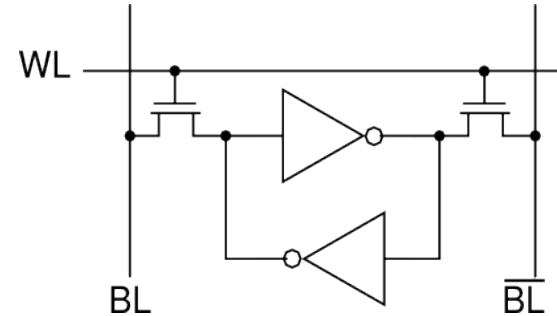
- Motiváció: Neumann-architektúrában szűk keresztmetszet



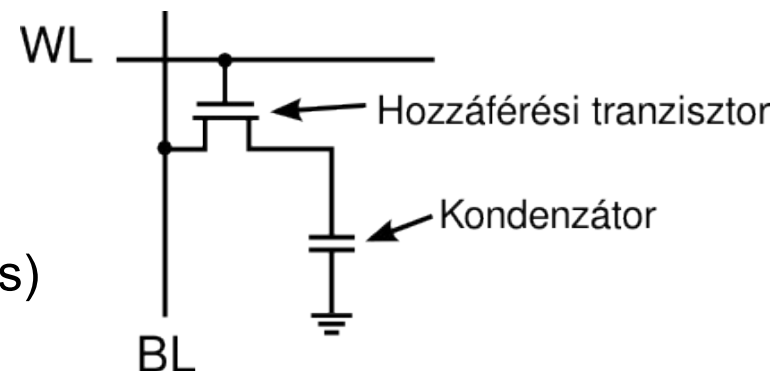
- Témakörök:
 - Memóriatechnológiák
 - DRAM, SRAM, memóriamodulok, többcsatornás memória-vezérlők, időzítések, DDR1-2-3-4-5, GDDR, stb.
 - Virtuális tárkezelés
 - Ha a valós memória mennyiségét a programok elől el akarjuk takarni
 - Gyorsítótár (cache memória)
 - A gyakran használt tartalmak elhelyezése egy gyors, de kicsi tárban

- Egyetlen bit tárolása:
 - SRAM-mal
 - DRAM-mal

- 1 bit tárolása: két szembekötött inverter
 - Olyan, mint egy flip-flop!
 - **1 bit → 6 tranzistor**: 6T
 - 2 hozzáférési: BL-ra és $\overline{BL}$ -ra köti a flip-flopot
 - 4 a flip-flop megvalósítása
- **Olvasás:**
 - WL-re logikai 1
 - Bit értéke: BL-re, negáltja: $\overline{BL}$ -re
 - Érzékelő erősítők: BL és $\overline{BL}$ különbségét figyelik
- **Írás:**
 - BL-t és $\overline{BL}$ -t a tárolni kívánt bit értékre kényszerítjük
 - WL-re logikai 1
 - A kényszerítés erősebb, mint az inverterek



- 1 bit tárolása: kondenzátorral
 - Töltött: bit=1, üres: bit=0
 - + kell 1 hozzáférési tranzisztor
 - **1 bit → 1 tranzisztor + 1 kondenzátor**: 1T1C
- **Olvasás:**
 - BL-t 0 és 1 közé félútra töltjük (precharge, előfeszítés)
 - WL-re logikai 1
 - Érzékelő erősítők: Merre változik a BL szintje?
 - Nő: bit=1
 - Csökken: bit=0
 - Kifolyt a töltés a kondenzátorból!
 - az olvasás destruktív
 - a végén vissza kell írni bele a kiolvasott bitet
- **Írás:**
 - WL-re logikai 1
 - A BL-en keresztül feltöltjük a kondenzátort, vagy kiengedjük a töltését
- **Frissítés:**
 - Magától is elszivárog a kondenzátor töltése!
 - Frissíteni kell (néhány 10 ms-onként) → kiolvasás + visszaírás

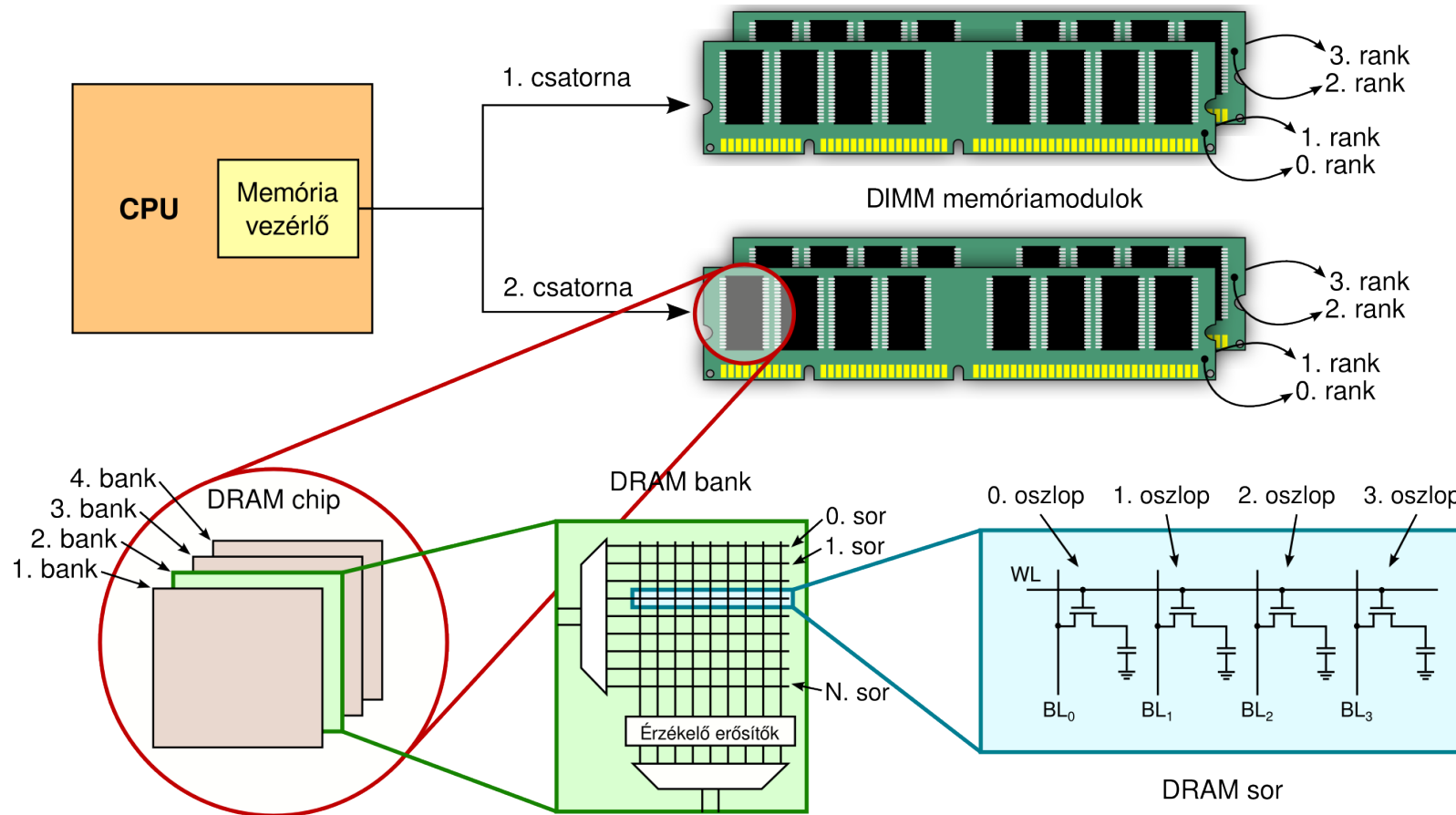


- Melyik a *gyorsabb*?
 - SRAM: tranzisztor hajtja a bitvezetést
 - DRAM: a kondenzátorból kitévelygő elektronok állítják a bitvezetést
 - ... és a DRAM-ot még frissíteni is kell!
 - **Gyorsabb: SRAM**
- Melyiknek nagyobb az *adatsűrűsége*?
 - SRAM: 6T
 - **DRAM: 1T1C → nagyobb adatsűrűség**
- Mire használják?
 - SRAM: cache
 - DRAM: rendszermemória
- Melyiket lehet a CPU mellé integrálni?
 - SRAM: könnyen. Csupa tranzisztor, mint a CPU.
 - DRAM: nehezen. Kondenzátort kell csinálni (CPU-hoz nem)
 - eDRAM: CPU mellé tett DRAM
 - 2009, POWER7, 32MB L3 cache eDRAM-ból
 - 2013, Intel Haswell GT3e, 128MB L4 cache eDRAM-ból
 - Konzolok: PlayStation 2, PlayStation Portable, Nintendo Wii U, stb.



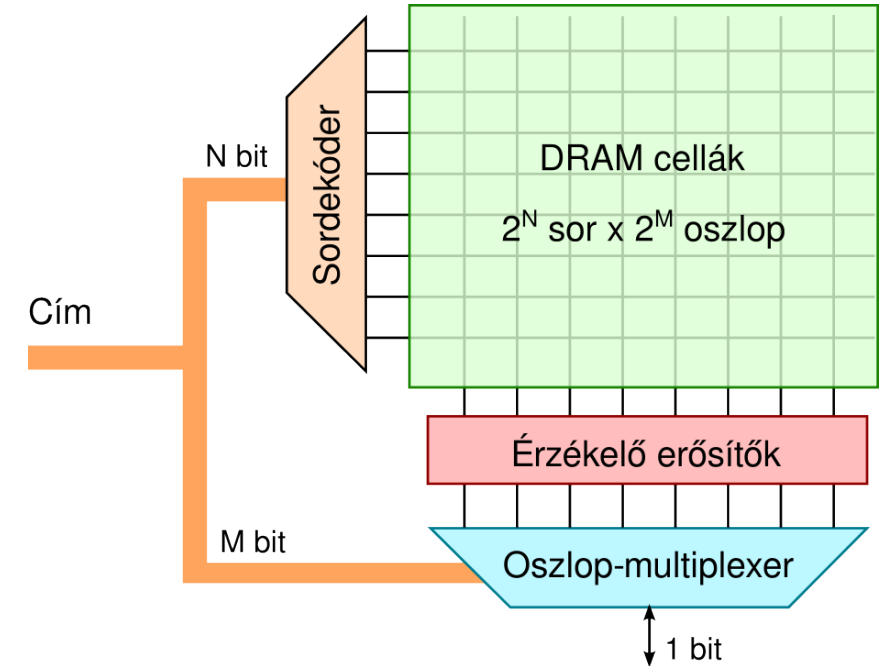
DRAM alapú rendszermemóriák

- Hogy csináljunk DRAM cellából memóriát?
 - Úgy, hogy nagy, olcsó legyen, kicsi legyen a késleltetése, nagy legyen az átviteli sebessége



Cellák mátrixban

- Egy sor: szóvezetékek összekötve
- Egy oszlop: bitvezetékek összekötve
- Olvasás:
 - A **sordekóder** kijelöl egy sort
 - A sor összes bitjét kiolvassák az **érzékelő erősítők**
 - A sorból a kívánt oszlopot kiválasztja az **oszlop-multiplexer**
- Két-fázisú műveletek:
 - Spórolunk a címbusz szélességével
 - Címbusz: sorcím → várunk → címbusz: oszlopcím → adatbusz: megjelenik az adat



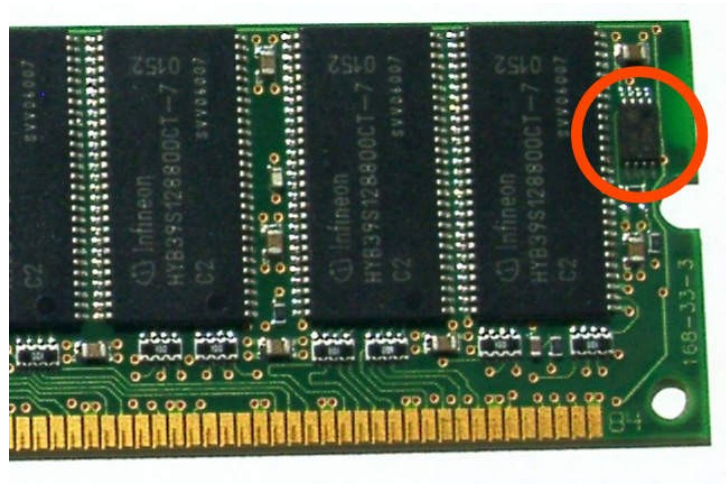
- Hogy lesz lineáris címből **sor** és **oszlop**cím?
- Példa: decimális rendszerben (!)
 - Memória kapacitás: 1 millió (cím: 0 ... 999999)
 - Tárolómező: 1000x1000
- Lineáris cím: 123456
- Sor és oszlopcím: **123**|**456** → 123. sor, 456. oszlop
- Csak „vágni” kell, osztani nem!
- Bináris rendszerben ugyanez bitekkel

- Az 5 legfontosabb:
 - **ACTIVATE**
 - Megnyit egy sort (adatok → érzékelő erősítőkbé)
 - **READ**
 - A nyitott sorból olvas egy oszlopot
 - Igazából az érzékelő erősítőkből olvas
 - **WRITE**
 - A nyitott sorba ír egy oszlopot
 - Igazából az érzékelő erősítőkbé ír
 - **PRECHARGE**
 - Bezárja a nyitott sort
 - Előfeszíti a bitvezetékeket is, hogy a köv. nyitás gyors legyen
 - **REFRESH**
 - Frissít egy sort
 - Majdnem egy sormegnyitás+lezárás
 - De ennek nem kell sorcím. Auto-inkremens.

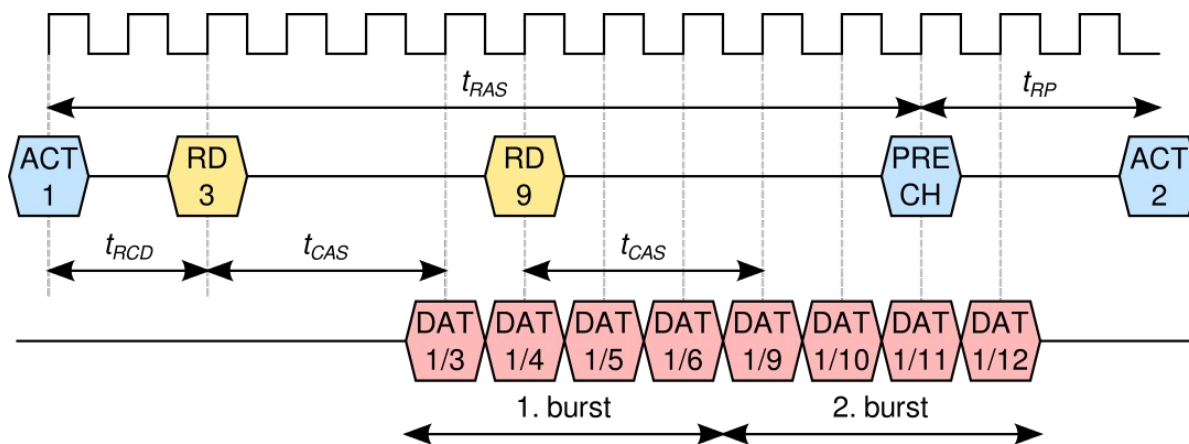
- Példa olvasási kérések:
 - (3. sor, 8. oszlop)
 - (3. sor, 14. oszlop)
 - (1. sor, 3. oszlop)
 - (1. sor, 4. oszlop)
- Parancsok (tfh. előfeszített):
 - ACTIVATE 3
 - READ 8
 - READ 14
 - PRECHARGE
 - ACTIVATE 1
 - READ 3
 - READ 4

- A parancsok végrehajtási ideje
- 4 legfontosabb:
 - T_{RCD} : A sor megnyitás ideje
 - T_{CAS} (CL): oszlopcím ráadásától az adat megjelenéséig tartó idő
 - T_{RP} : Előfeszítés (PRECHARGE) ideje
 - T_{RAS} : Minimális sor nyitvatartási idő
- Sok-sok ilyen van még
- Mértékegység: órajel (szinkron DRAM), ns (aszinkron)
- Boltban veszünk memóriát 8-9-10-11 időzítéssel:
 - $T_{CAS}=8$, $T_{RCD}=9$, $T_{RP}=10$, $T_{RAS}=11$
- Ha csak az van ráírva, hogy CL7: $T_{CAS}=7$

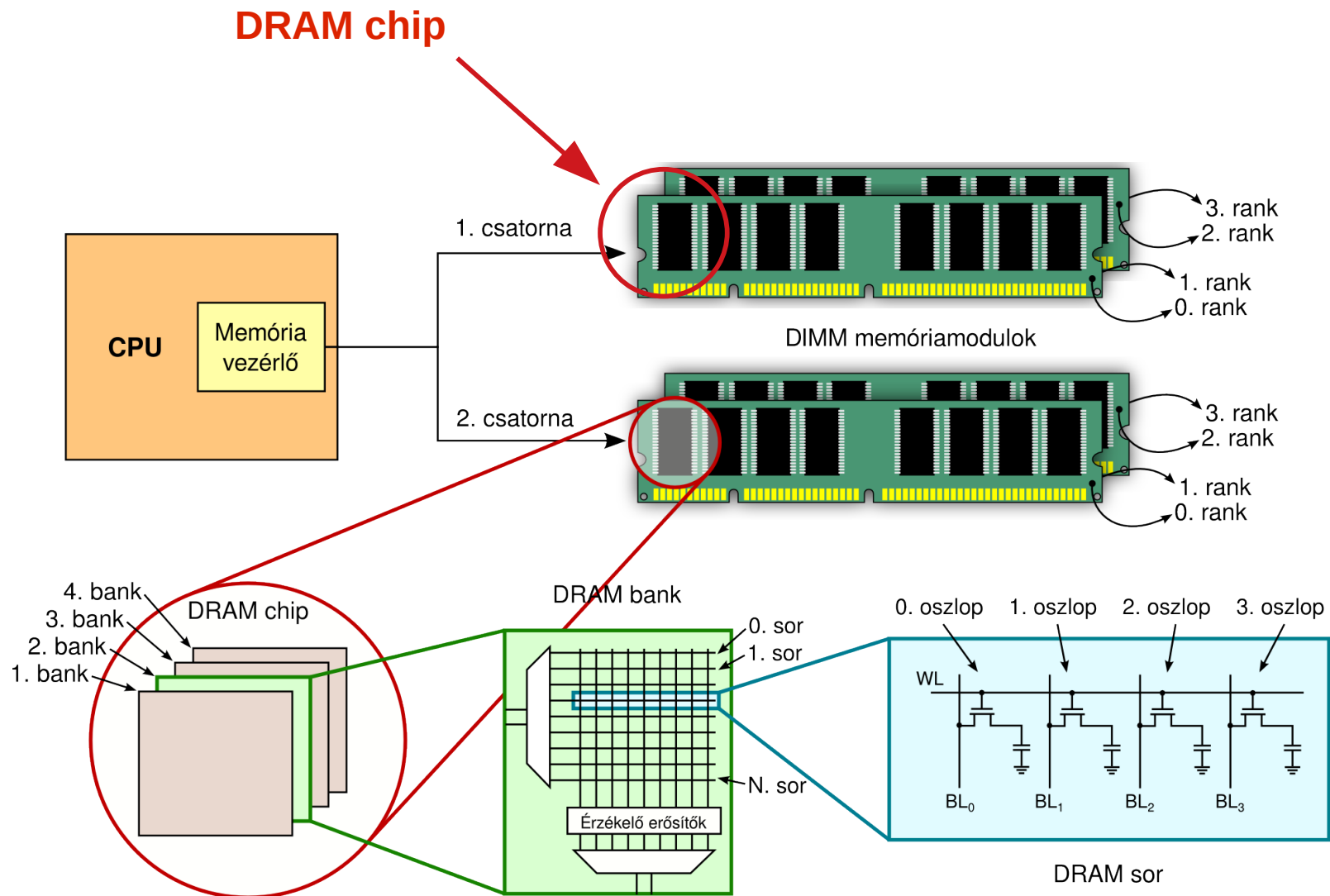
- Honnan tudja az időzítéseket a memóriavezérlő?
 - Megkérdezi a memóriamodultól
 - Memóriamodul tartalma:
 - DRAM chip-ek
 - ...és egy **SPD** chip! Ebben vannak a jellemzők.



- A sor minden kiolvasott oszlopánál:
 - Oszlopcím $\rightarrow T_{CAS} \rightarrow$ adat elvétele
 - Ez túl pazarló!
- **Burst mód**
 - Egyszer adok rá oszlopcímet
... nem egy oszlopot ad, hanem egy egész sorozatot!
 - Burst hossz: konfigurációs paranccsal beállítható

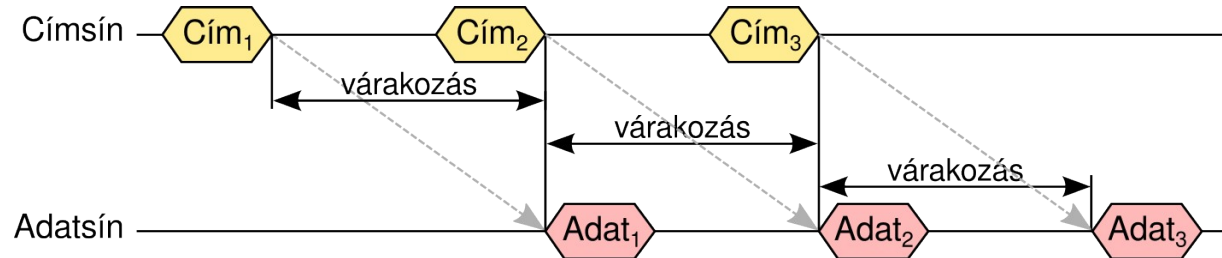


- **Átlapolt parancskiadás és adatátvitel**
 - Nem kell megvárni az előző parancsra a választ

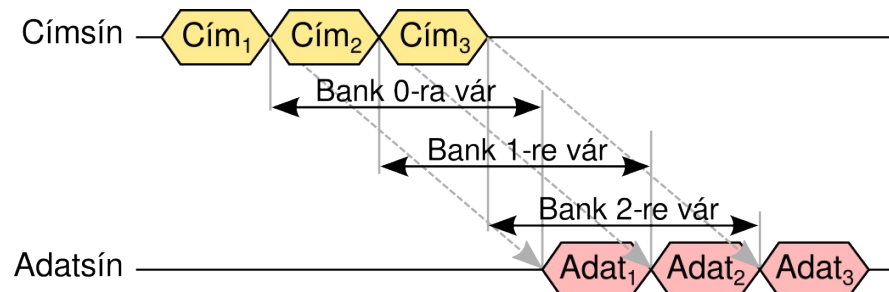


- Egy DRAM chip bankokból áll
- **Bank:**
 - Független DRAM cella mátrixok
 - Saját sordekóder, érzékelő erősítő, oszlopmultiplexer
 - Mindegyikben lehet 1-1 nyitott sor
 - Egy DRAM chip-ben több nyitott sor lehet
 - **Átlapolt adatátvitel!**

- Átlapolás nélkül:

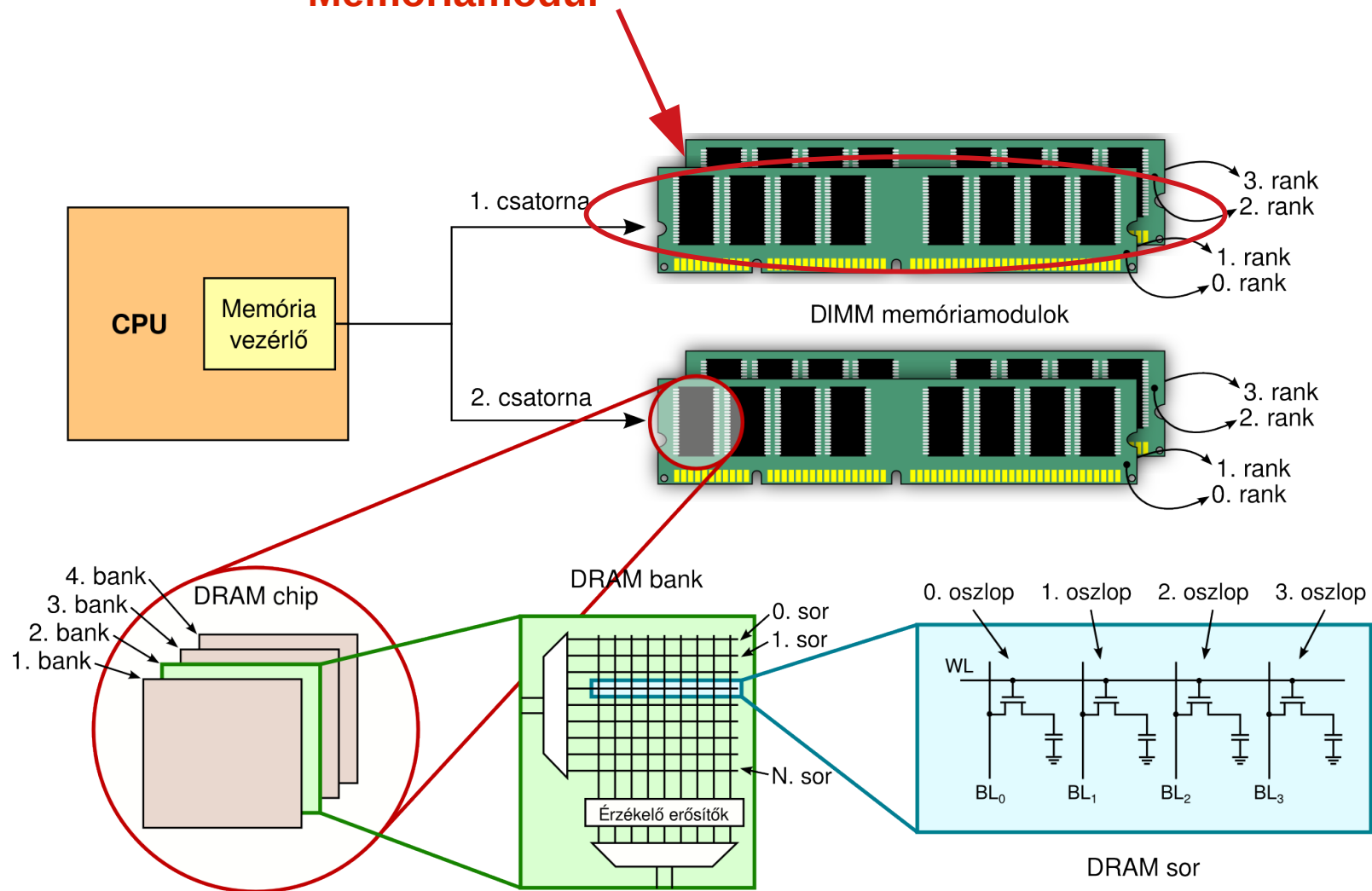


- Átlapolással:

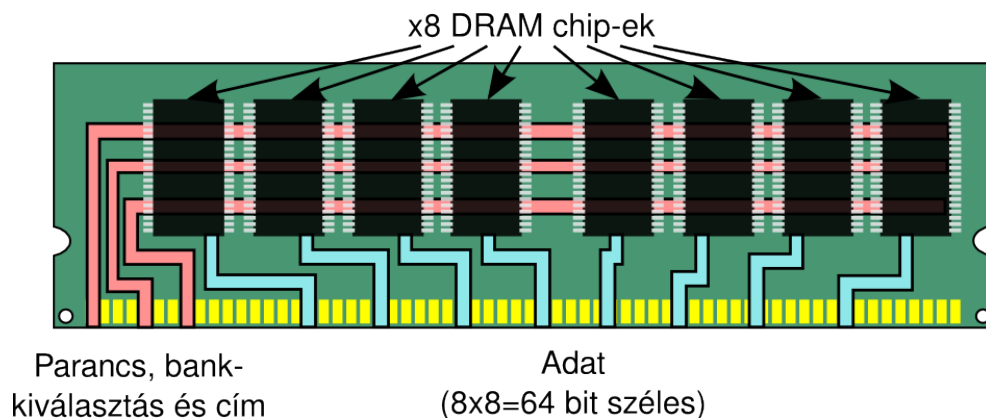


- Bankok nem biteket tárolnak
 - 1 oszlop: 4 bit, 8 bit, 16 bit (x4, x8, x16)
- Interfész:
 - **Parancs vezetékek**
 - Mit kell csinálnia
 - **Bank kiválasztó vezetékek**
 - Melyik bank-ra vonatkozik
 - **Címvezeték**
 - ACTIVATE: sorcím
 - READ/WRITE: oszlopcím
 - **Adatvezetékek**
 - 4, 8, vagy 16 bit

Memóriamodul

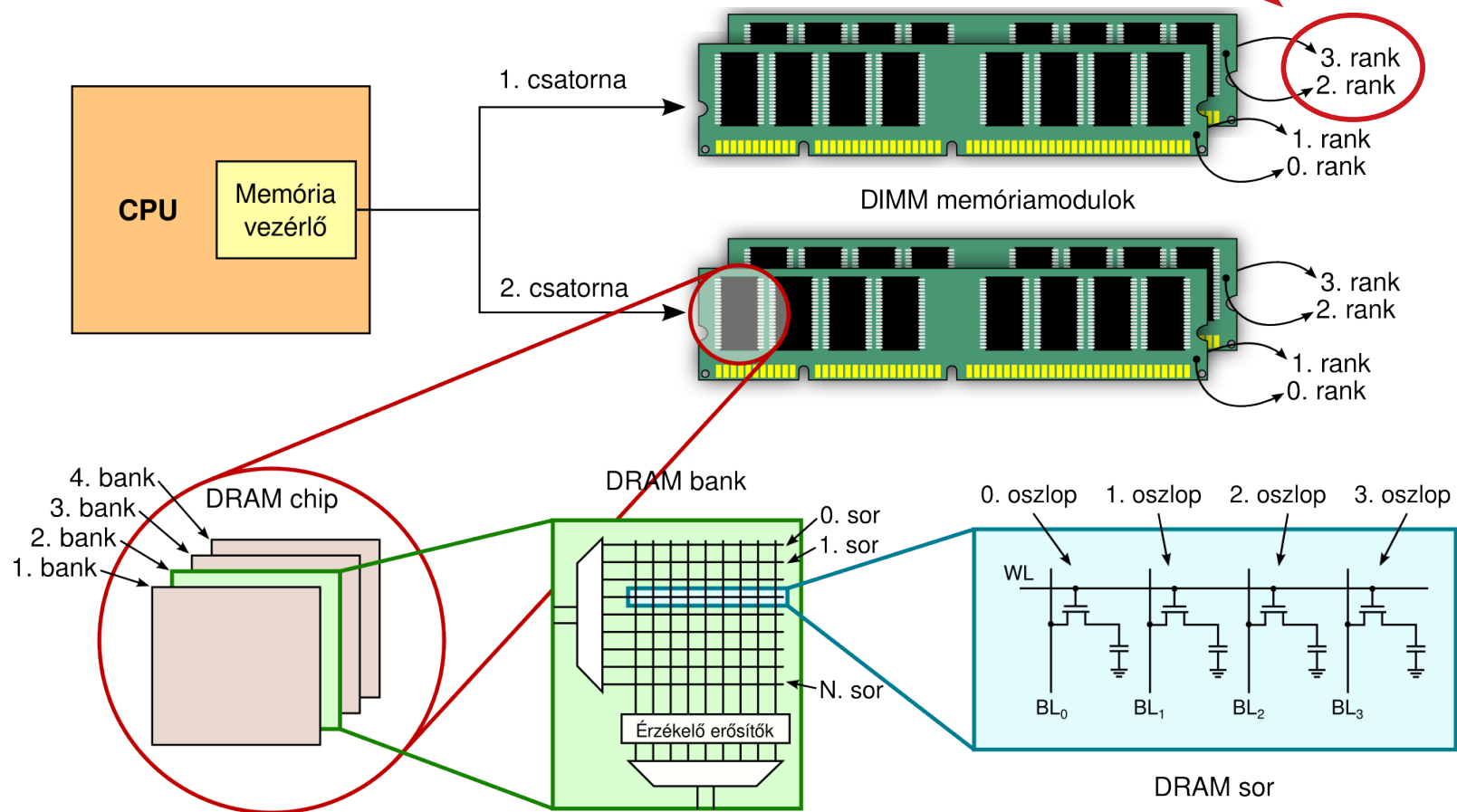


- Egy memóriamodul több DRAM chip-ből áll
- Parancs, bank kiválasztás, cím: osztott
- Adatvezetékek: összefogva

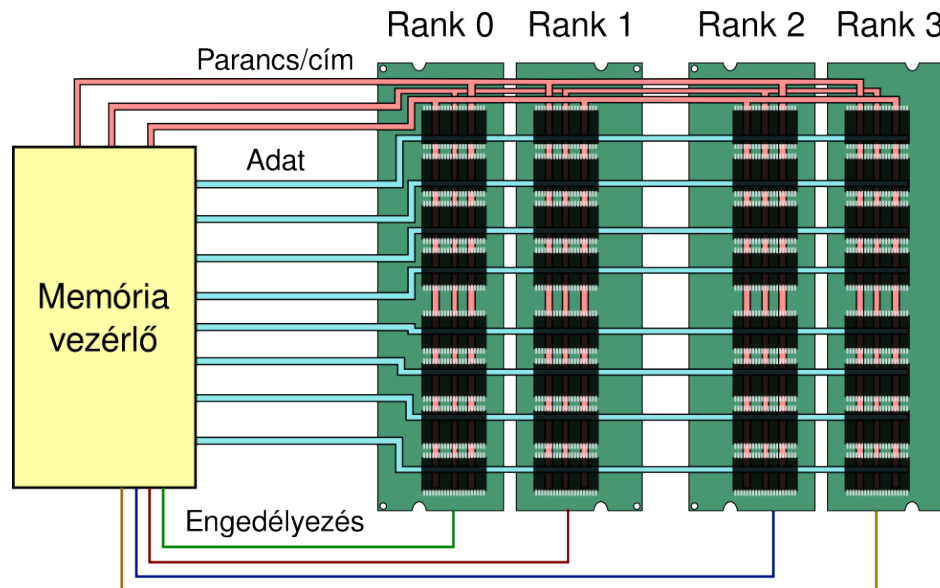


- Minden parancsot minden chip megkap
→ A megfelelő bankokban ugyanazok a sorok vannak nyitva
- Hatása:
 - Adatátviteli sebesség: 8x
 - Késleltetés: ugyanaz

DRAM rank-ek

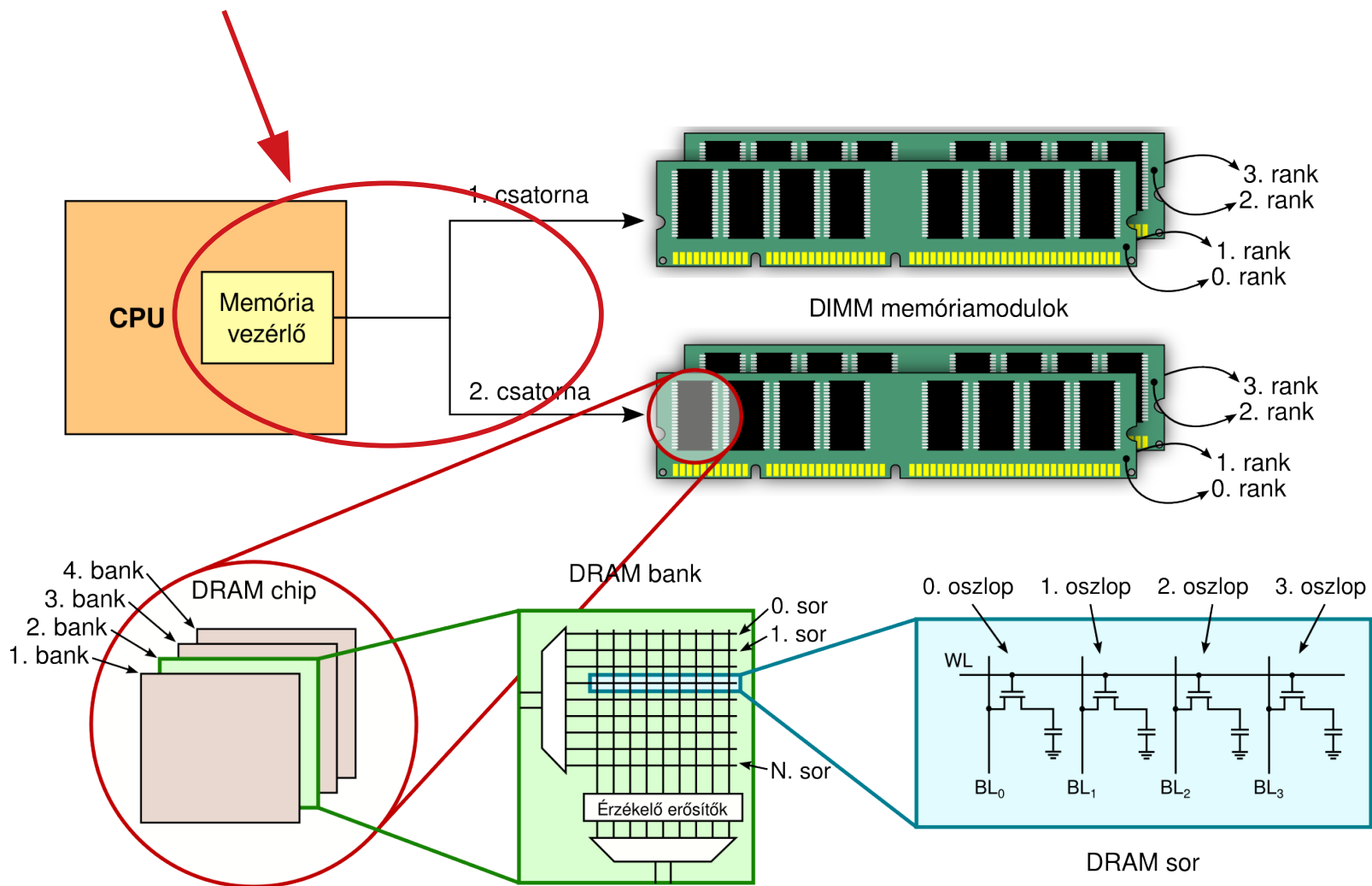


- A tárolási kapacitás növeléséhez
- Független egységek közös buszon
- Minden vezetékek osztozott
 - Egyszerre csak 1 lehet bekapcsolva → chip select vezetékek



- Hatása:
 - Adatátviteli sebesség: ugyanaz
 - Késleltetés: jobb (több bank, több nyitott sor)

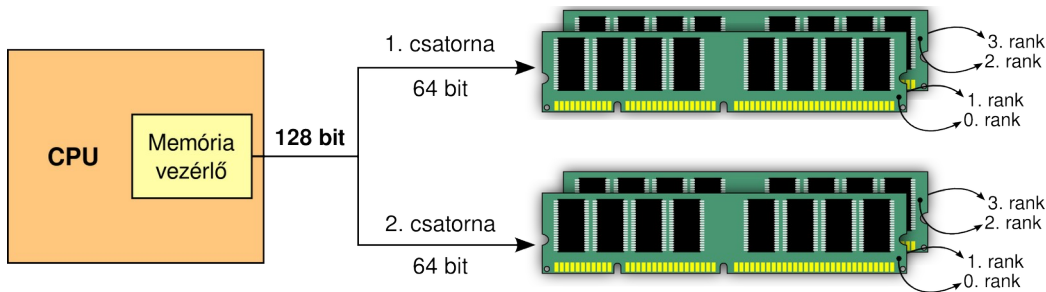
Memóriavezérlő



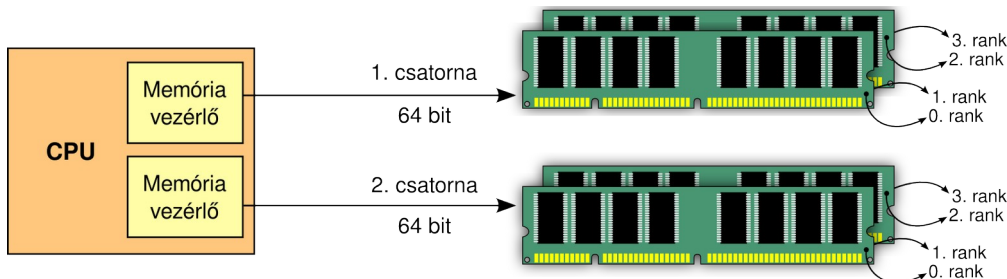
- Feladatai:
 - Memóriakérések kiszolgálása (CPU és perifériák felől)
- Teendők:
 - Címek leképzése csatornára/rank-re/bankra/sorra/oszlopra
 - Kérések sorrendjének optimalizálása
 - Gazdálkodás a nyitott sorokkal
 - Periodikus frissítések (REFRESH paranccsal)

- Memóriavezérlő több csatornát is használhat
- *Szinkronizált* eset
 - Azonos modulok kellene (darabra, méretre, időzítésekre)
 - Tökéletes szinkron működés

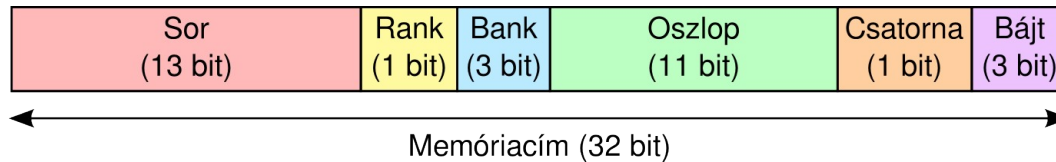
2 db 64 bites csatorna → 1 db 128 bites csatorna



- *Független* csatornák esetén
 - Nem szükségesek azonos modulok
 - Annyi független memóriavezérlő, ahány csatorna



- Címleképzés:
 - Cél: lehető legkevesebb sorváltás
 - Memóriakérések lokalitása kihasználható
 - Az egybefüggő, sorváltás nélkül elérhető címtartomány legyen nagy
 - Példa:
(4 GB, 8 bank/chip, 2^{13} sor/bank, 2 rank, 2 csatorna, 64 bit adatszélesség)



- 512 kB elérhető sorváltás nélkül

- Kérések kiszolgálási sorrendjének optimalizálása:
 - Spóroljunk a drága sor nyitás/zárással
 - Sorrendi kiszolgálás (FCFS)
 - Gyorsakat előre (FR-FCFS)

FCFS

Kérések:

(3.sor, 8. oszlop)
(1.sor, 3. oszlop)
(3.sor, 14. oszlop)

FR-FCFS

(3.sor, 8. oszlop)
(3.sor, 14. oszlop)
(1.sor, 3. oszlop)

Parancsok:

ACTIVATE 3
READ 8
PRECHARGE
ACTIVATE 1
READ 3
PRECHARGE
ACTIVATE 3
READ 14

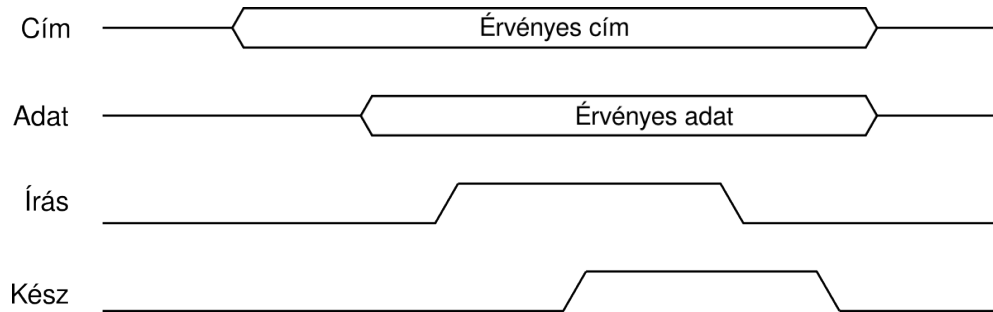
ACTIVATE 3
READ 8
READ 14
PRECHARGE
ACTIVATE 1
READ 3

- Gazdálkodás a nyitott sorokkal:
 - Elfogynak a kérések. Bezárjuk az aktuális sort?
 - **Ne:**
 - Ha ugyanarra a sorra jön a köv. kérés, nem kell megnyitni
 - Ha más sorra jön a köv. kérés, be kell zárni (késleltetés)
 - **Igen:**
 - Ha ugyanarra a sorra jön a köv. kérés, újra meg kell megnyitni (késleltetés)
 - Ha más sorra jön a köv. kérés, nem kell bezárni
 - **Adaptív:**
 - Megtippeli, hogy ugyanerre a sorra jön-e a köv. kérés
 - APM (Active Page Management)
Core i7: „Adaptive Page Closing” opció a BIOS-ban

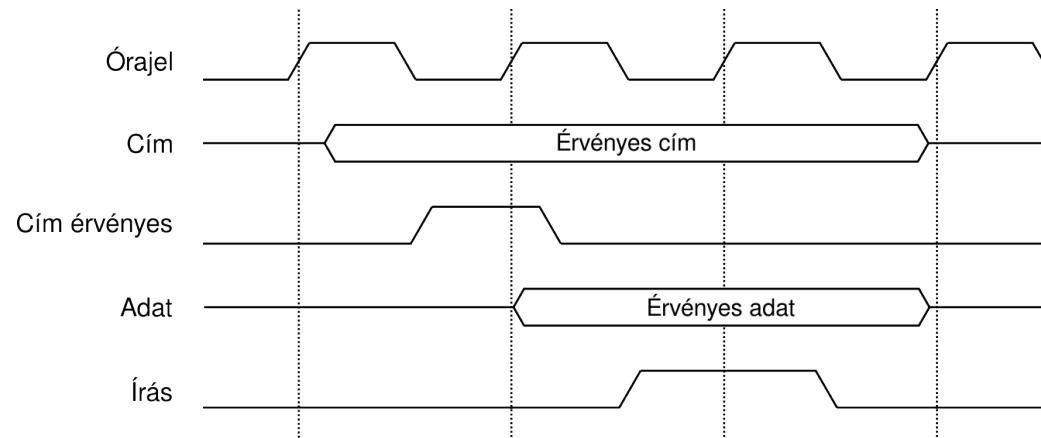


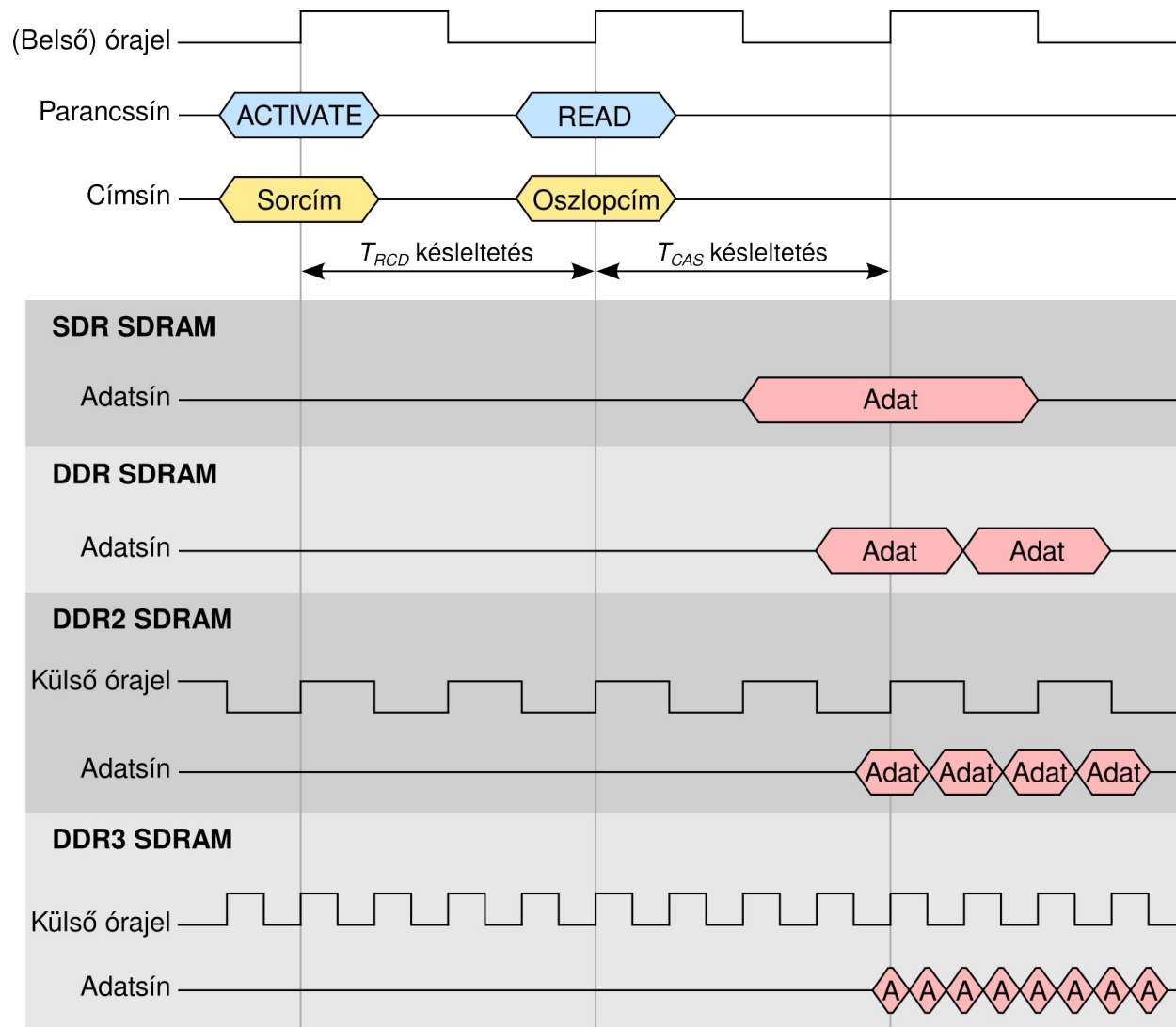
A DRAM technológiák evolúciója

- 15 évvel ezelőttig: aszinkron interfész



- Ma: szinkron – SDRAM
 - Órajelet használ





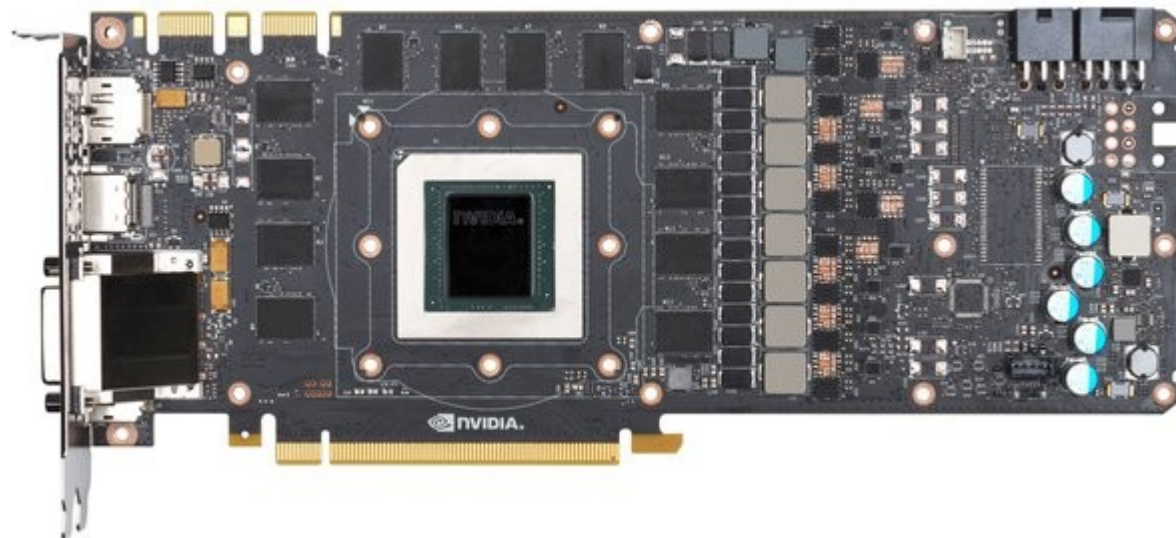
- Szabványos jelölés:
 - Az ekvivalens SDR órajel megadásával:
 - DDR-400, DDR2-800, DDR3-1600 órajele: 200 MHz
 - **A parancsok késleltetése egyenlő!!!**
 - ... csak egyre gyorsabban küldik át az adatot
 - Az adatátviteli sebesség megadásával:
 - pl. DDR2-800: adatátvitel 800 MHz-en, 8 byte/adategység
→ **PC2-6400** ($800 \times 8 = 6400$)
 - pl. DDR3-1600: adatátvitel 1600 MHz-en, 8 byte/adategység
→ **PC3-12800** ($1600 \times 8 = 12800$)
- Figyelem! Ez egy példa. Az órajel nem mindig 200 MHz!

- **A külső ↔ belső órajel aránya (burst hossz) nem nő tovább**
- Teljesítmény fokozása:
 - **Belső órajel növelése**, ehhez:
 - kisebb tápfeszültség (1.2V)
→ kisebb fogyasztás, kevésbé melegszik
 - busz órajele nő, áthallás, csúszás kezelése nehezedik
→ CRC (adatbusz) / paritásbit (cím és parancs) kell,
kalibrálás szükséges
 - 8 helyett 16 bank (nagyobb párhuzamosság)
 - bank csoportok bevezetése, bonyolultabb időzítés
 - Nagyobb méretű (kapacitású) modulok, ehhez:
 - maximum 4 rank / modul
 - a bank-ek nem négyzet alakúak
→ több sorcím bit, mint oszlopcím bit → command jelek
a felső címbitekkel multiplexálva (kevesebb kivezetés)
 - Max. 8 csatornás memória vezérlő

- Megjelenés: 2020 július
- **Burst hossz újra nő → =16**
- Teljesítmény fokozása:
 - Még kisebb tápfeszültség (1.1V)
 - 2x annyi bank
 - **Modulonként** 2 csatorna (64 bit helyett 32+32 bites adategységek)
 - Kétszeres burst méret fele szélességgel → 64 byte/burst
 - Kétszer annyi ideig tart
 - De! Két független csatornán egy időben
→ Ugyanaz a throughput
→ Jobb késleltetés
- Támogatás:
 - Intel 12. generáció, Alder Lake, Xeon Sapphire Rappids
 - AMD jelenlegi Zen3 Ryzen 5000 series még nem, Zen-4 és Zen-3+ igen

	SDR	DDR	DDR2	DDR3	DDR4	DDR5
Belső órajel	66-133 MHz	133-200 MHz	100-200 MHz	100-200 MHz	200-533 MHz	300-450 MHz
Adat/b. órajel	1	2	4	8	8	8
Adatátv. MB/s	528-1064	2128-3200	3200-6400	6400-12800	12800-34112	38400-57600
Burst hossz	1-8	2-8	4-8	8	8	16
Bank szám	2-4	2-4	4-8	8	16	32
Feszültség	3.3V	2.5V	1.8V	1.5V	1.2V	1.1V

- Cél: extrém nagy sávszélesség → grafikus kártyákhoz
- Egyszerűbb:
 - Nincs rank, nincs modul
 - Minden chip közvetlenül a vezérlőhöz kötve
- x32 adatszélesség chip-enként
 - 8 chip → 256 bites adatszélesség
- 16 bank, de bankonként 2 sor lehet nyitva
- Egyszerre tud írni és olvasni
- 3 -féle órajel:
 - DRAM cella órajel, pl. 375 MHz
 - Parancsórajel: 4-szeres: 1500 MHz
 - Adatátviteli órajel: 4-szeres: 6 GHz !!!
→ 256 bites adatszélességgel **196 GB/s** !
- Problémák:
 - Ár (4-5-szörös)
 - Rövid chip-vezérlő távolság
 - Nem moduláris / nem bővíthető
 - Melegszik, sokat fogyaszt
- GDDR6:
 - x32 helyett x16, de 2 csatorna + jobb gyártástechnológia → 768 GB/s



- DRAM alapú memóriák:
 - Késleltetéssel problémák vannak (évtizedek óta változatlan)
 - Grafikus kártyának nem fontos
 - CPU-nak nagyon fontos (lenne)
 - Sáv szélesség még növelhető (egy darabig)



HÁLÓZATI RENDSZEREK
ÉS SZOLGÁLTATÁSOK
TANSZÉK

