

Számítógép architektúrák

A memória

Tartalom

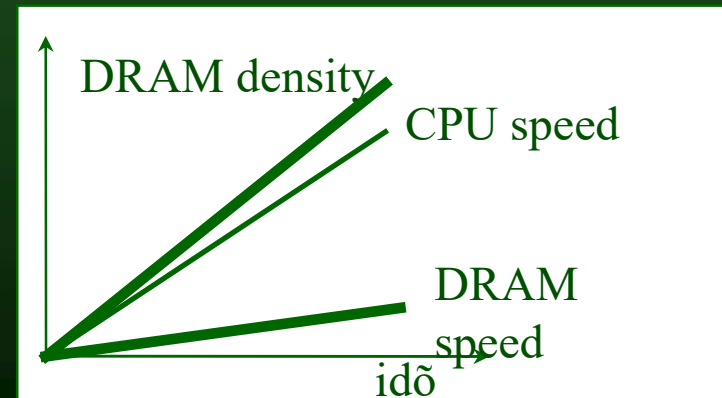
- **Félvezetű tárolók**
- **DRAM, SRAM**
- **ROM, PROM**
- **Tokozások, memóriamodulok**
- **Lokalitás elve**
- **Gyorsítótárak (cache)**

A memória

- **Tár: programok és adatok tárolására. Címezhető cellák.**
- **Központi tár: (gyors) memóriabuszon v. a rendszerbuszon keresztül kapcsolódik a processzorhoz**
- **Memória a perifériavezérlőkön is! Ezek is címezhetőek! Néha címtartományuk egybeesik, néha nem.**
- **Később figyeljük meg a memória-hierarchiát!**

A táruk implementációja

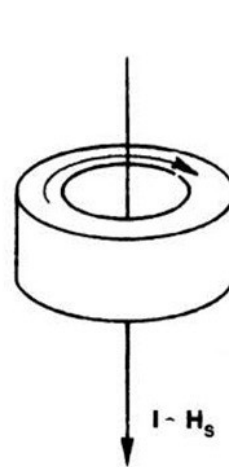
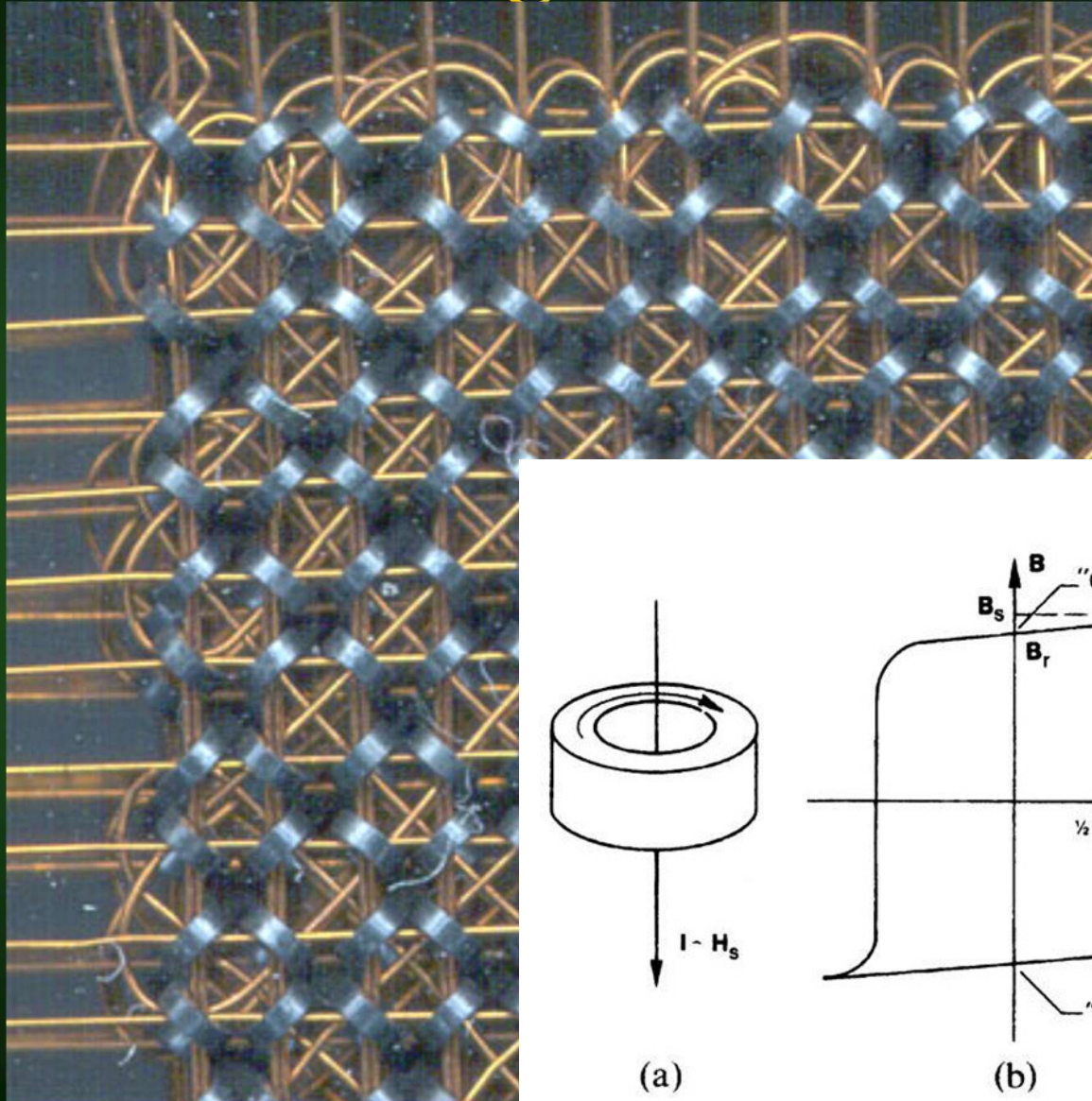
- Régebben ferritgyűrűs táruk: mágnesezhetőség-fluxusváltás elven. Nem felejtettek.
- Ma már magas-integráltságú félvezetős lapkák (milló számú tranzisztor[-kondenzátor] hálózata, tokban), memória modulra szerelve
 - ciklikus működés!
- Két trend:
 - kapacitás növelés,
 - elérési idő csökkentés.



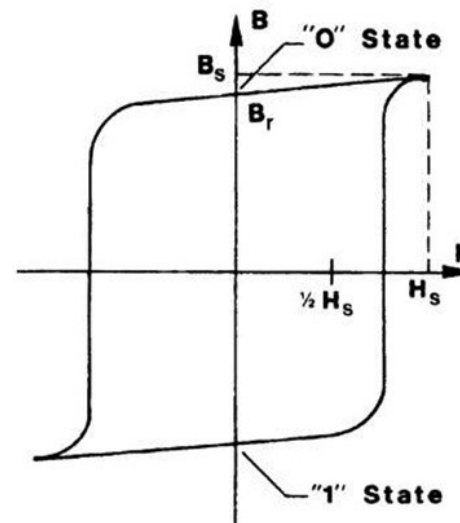
A memória© Vadász, 2007.

Ea7 4

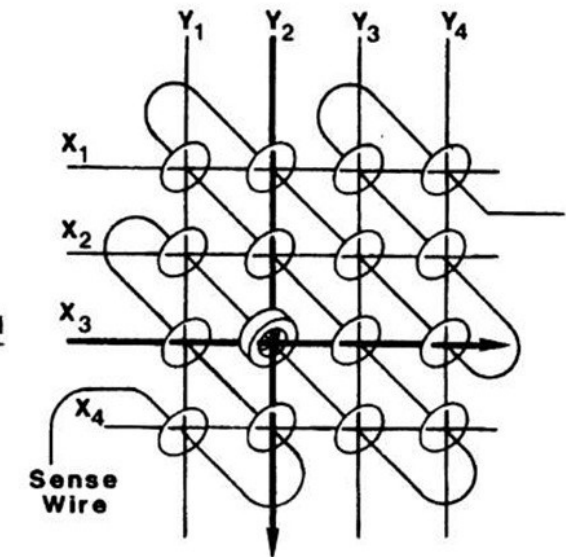
Magnetic-core memory



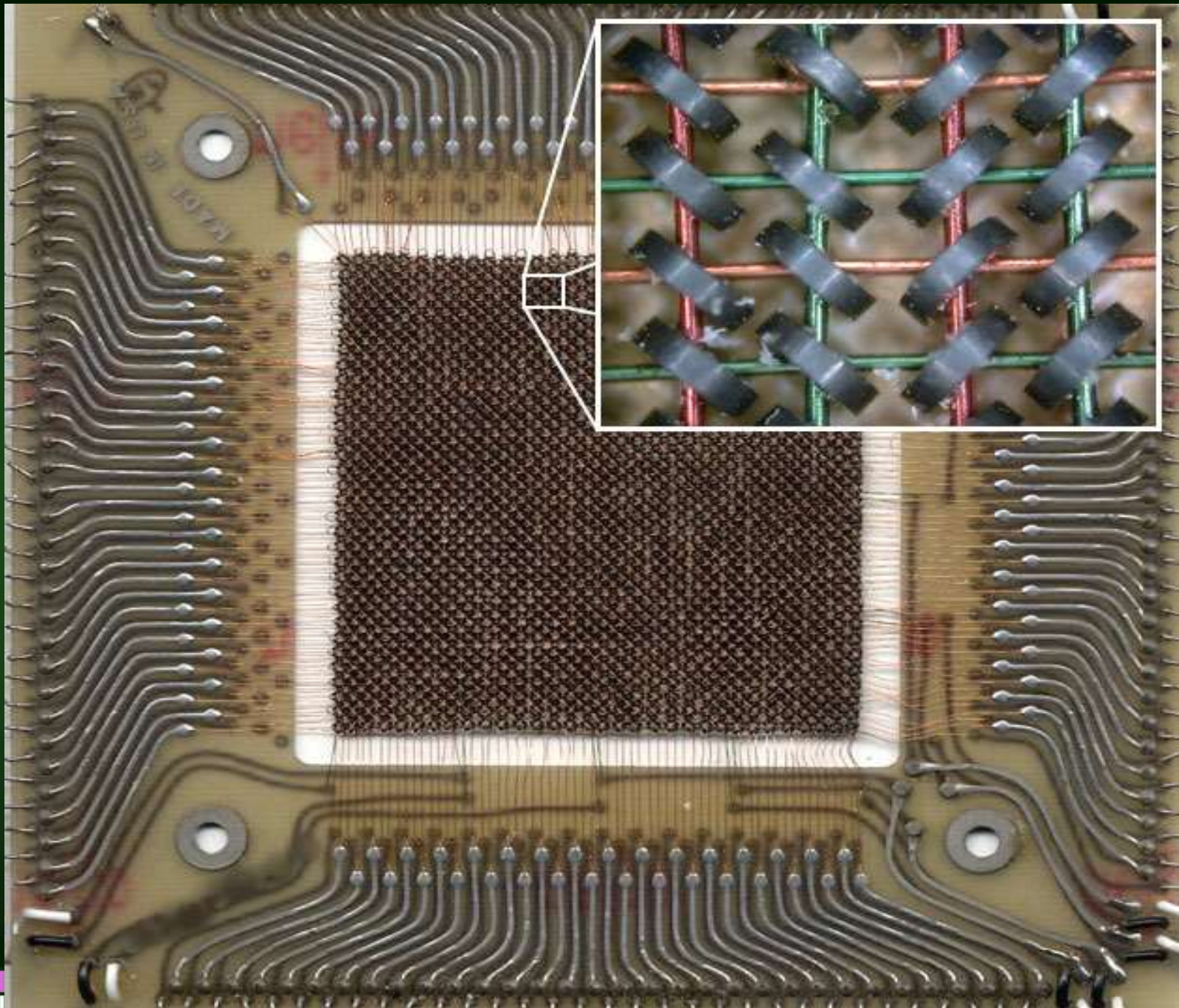
(a)



(b)



(c)



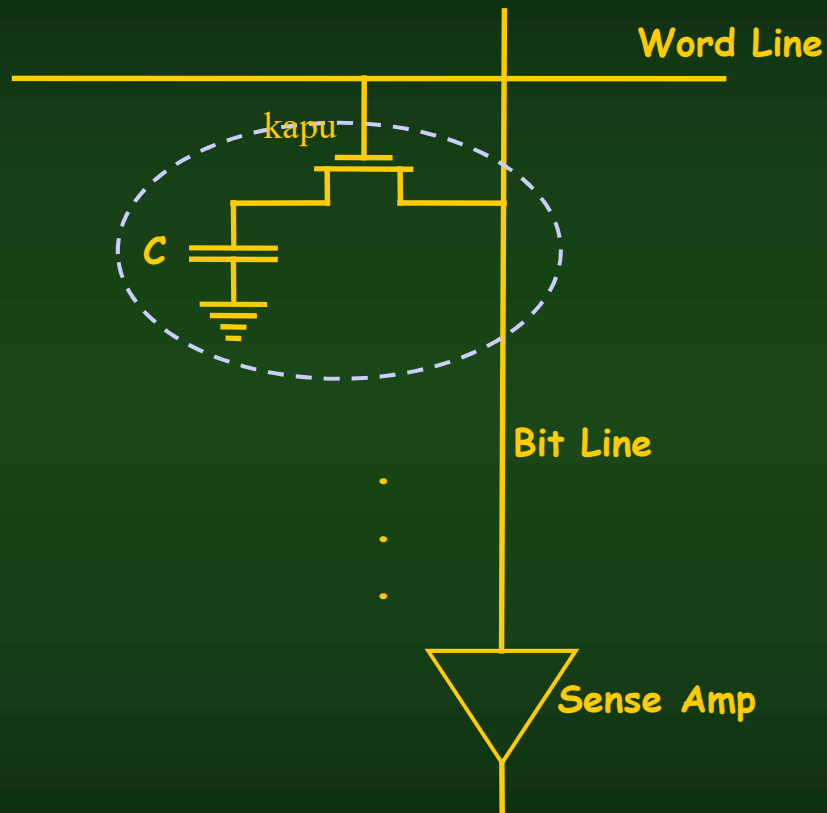
Megjegyzések

- A „kiszolgálási idő” két összetevője:
 - (1) **access time**, a memória modulhoz beérkező kérelemtől a rendelkezésre állásig eltelt idő (a memória modul válaszüzeje). Ez nagyságrendileg **50-150 ns** szokott lenni;
 - (2) a memória **buszbeli** (és a chipset-beli) **idő**. Ez átlagosan, nagyságrendben **125 ns**.
- Ezzel egy átlagos kiszolgálási idő: **195 ns**.
- Vessük össze egy (átlagos) L2 cache (ez CPU-n kívüli, external cache) kiszolgálási idővel, nos ez **45 ns**.
- Az SDRAM-ok megjelenése óta a sebességet MHz-ben adják meg. Még egy kis „hamisság”: a lapkákra lehet írva ns-ban idő: nos, az a ciklusidő!!! Azaz **15 ns → 66 MHz**; **10 ns → 100 MHz**; **8 ns → 133 MHz** sít. (Szóval ez a busz ciklus ...) **2.78 ns → 3600 MHz**
- A valódi memóriaelérési sebesség a buszciklus szorozva a szószélességgel (MHz * bit) ... Pl.
 - 100 MHz lapka, 64 bites busz = 6400 Mbit/s = 800 MByte/sec sebesség
 - RDRAM 800 MHz, 16 bites = 1,6 GByte/sec

A félvezető tárolók

- **RAM: Random Access Memory**
 - random: egy cella elérése nem függ a többitől, “tetszőlegesen” bármelyiket címezhetjük.
(Nem pl. szekvenciális elérésű)
 - Sorokból és oszlopokból álló háló, elemei a cellák
- **DRAM: Dynamic RAM**
 - Egy cella egy tranzisztor-kondenzátor pár, egy bithez.
 - Dinamikusság: a kiolvasás-beírás, a “frissítés” is dinamikus, időt igénylő.
 - Írható-olvasható,
 - táp megszűnésével “felejt”.
 - MOS, CMOS, NMOS technológiák.
- **SRAM: Static RAM**

Egy cella ...

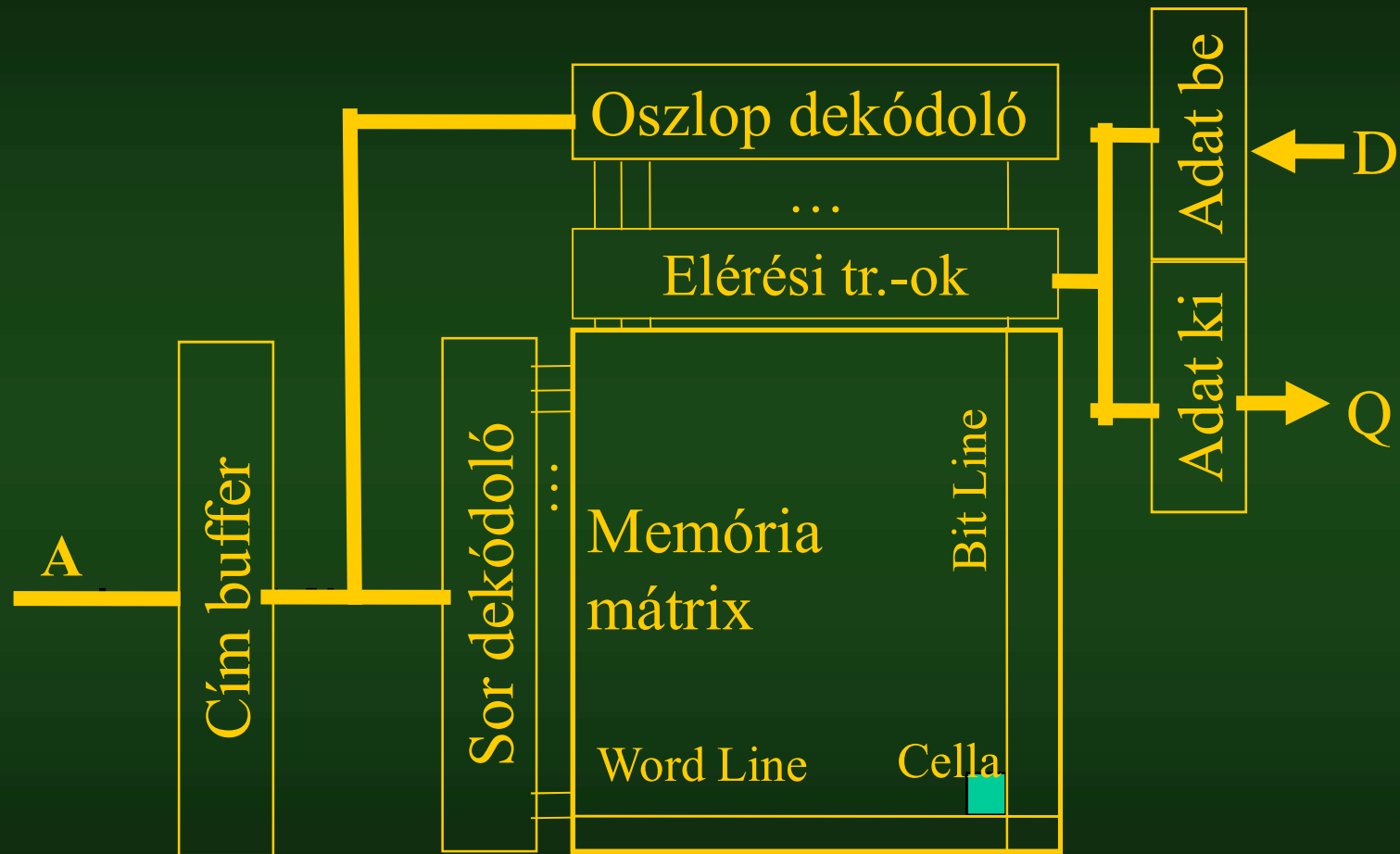


- **Írás:**
 - Állíts a bit vonalra magas v. alacsony szintet (a beírni kívánt bit szerint);
 - Nyisd a kaput: megfelelően feltöltődik a C
- **Olvasás:**
 - Állítsd a bit vonalat „fele” feszültségre;
 - Nyisd a kaput;
 - A C szintjétől függően a bit vonal feszültsége elmozdul, amit a Sense Amp érzékel.

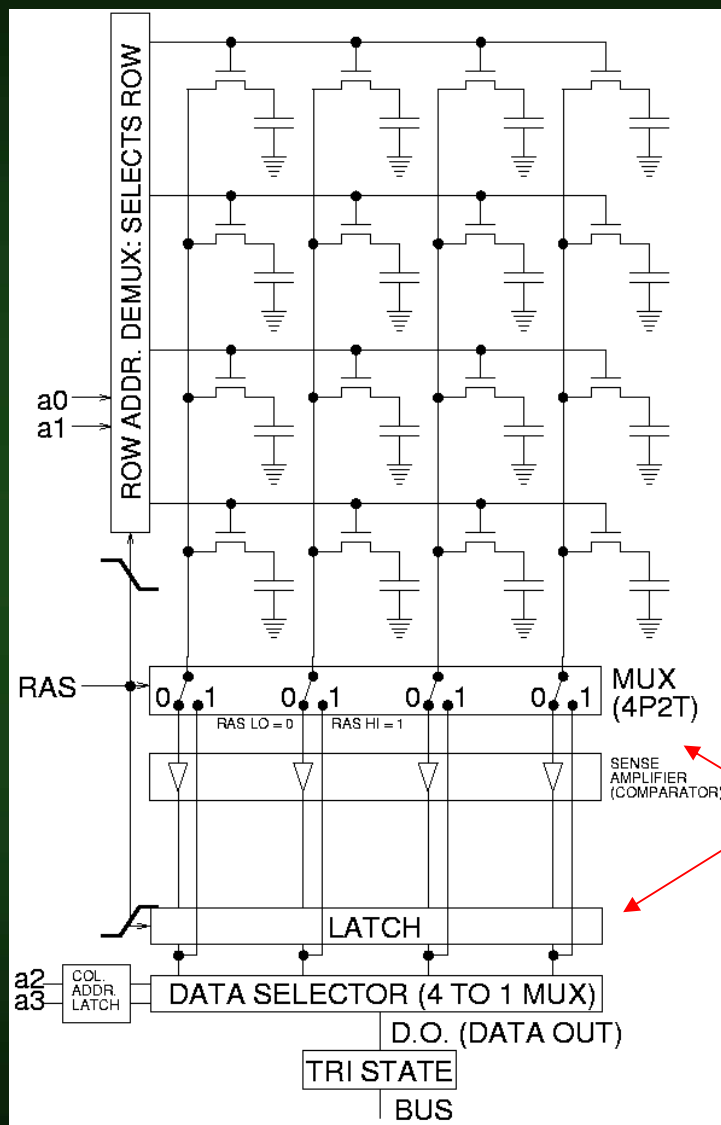
DRAM áramkörei ...

- Sorokból és oszlopokból álló háló, elemei a cellák
- A lapkában további speciális áramkörök segítenek
 - Cellák sor/oszlopainak kiválasztására (r/c address select/decoder, sor/oszlop cím bufferek)
 - A cellákból „kiolvasott” jelek tárolására (sens amplifiers: elérési tranzisztorok, output buffers)
 - Frissítési szekvenciák nyomon követésére (counter)
 - A cellák írására, töltésük „megemelésére” (write enable)
- „Kívülről” segíthet a memória kontroller (esetleg CPU)
 - Memória típus, sebesség, mennyiség azonosítás, hibakezelés

DRAM logikai felépítés

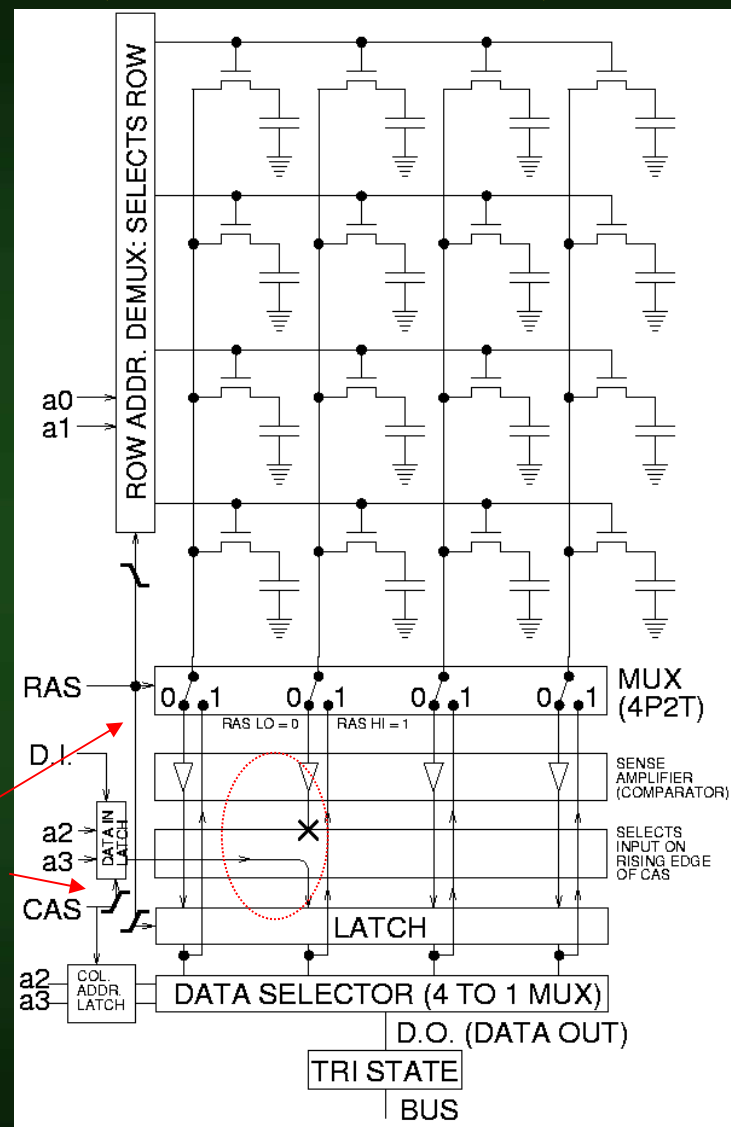


Egy 4 x 4-es mátrix (1 bit adat)



Olvasás

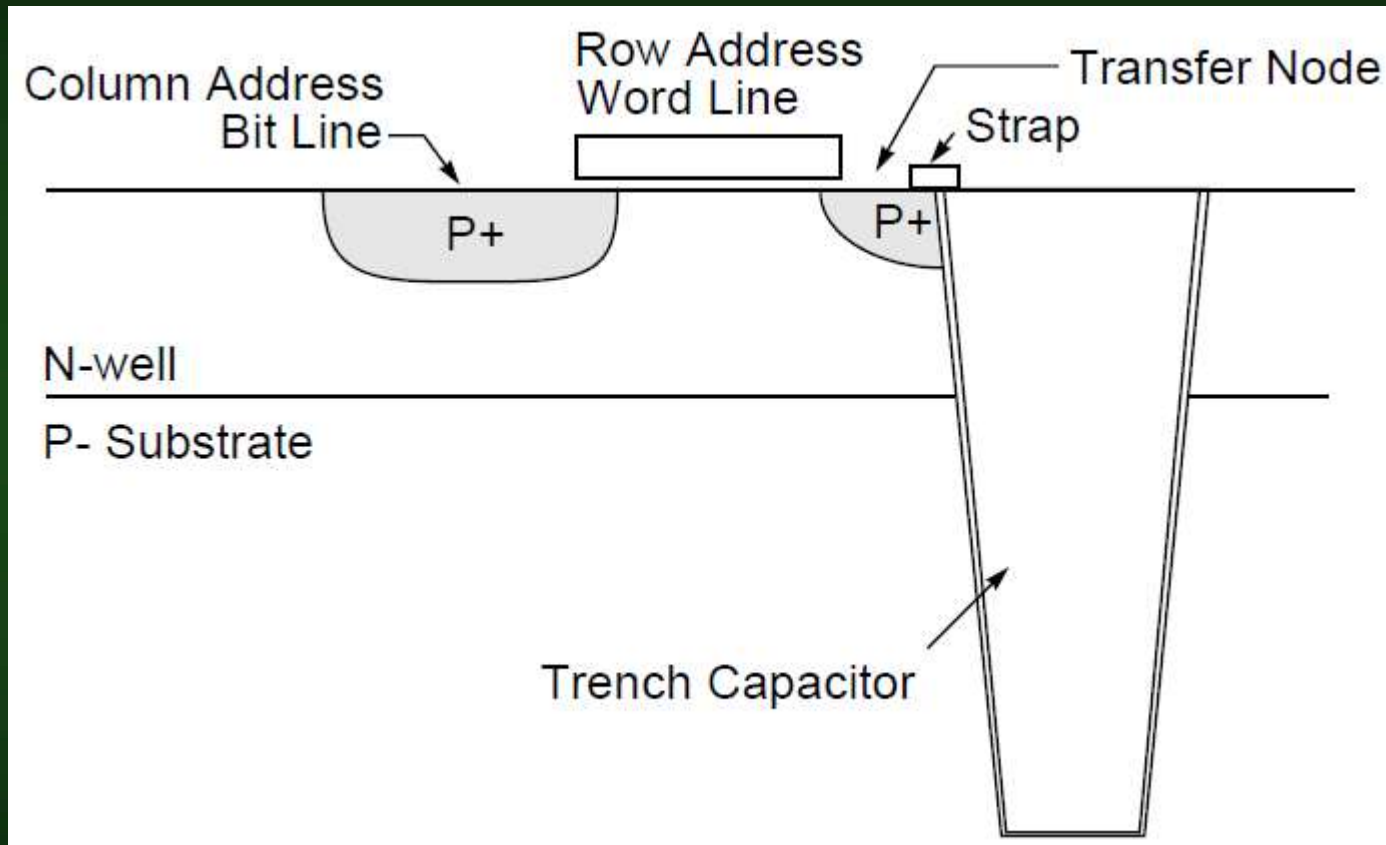
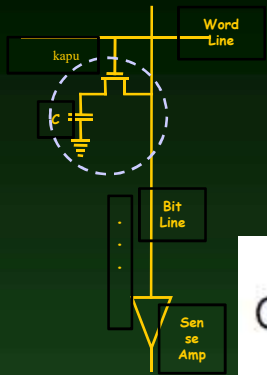
A memória © Vadász, 2007.



Írás

Ea7 12

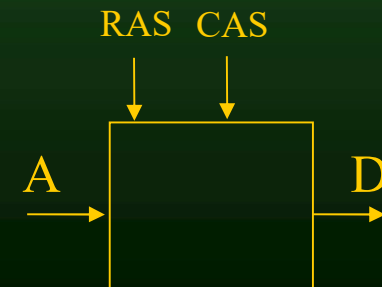
Egy cella



© IBM

DRAM operációk

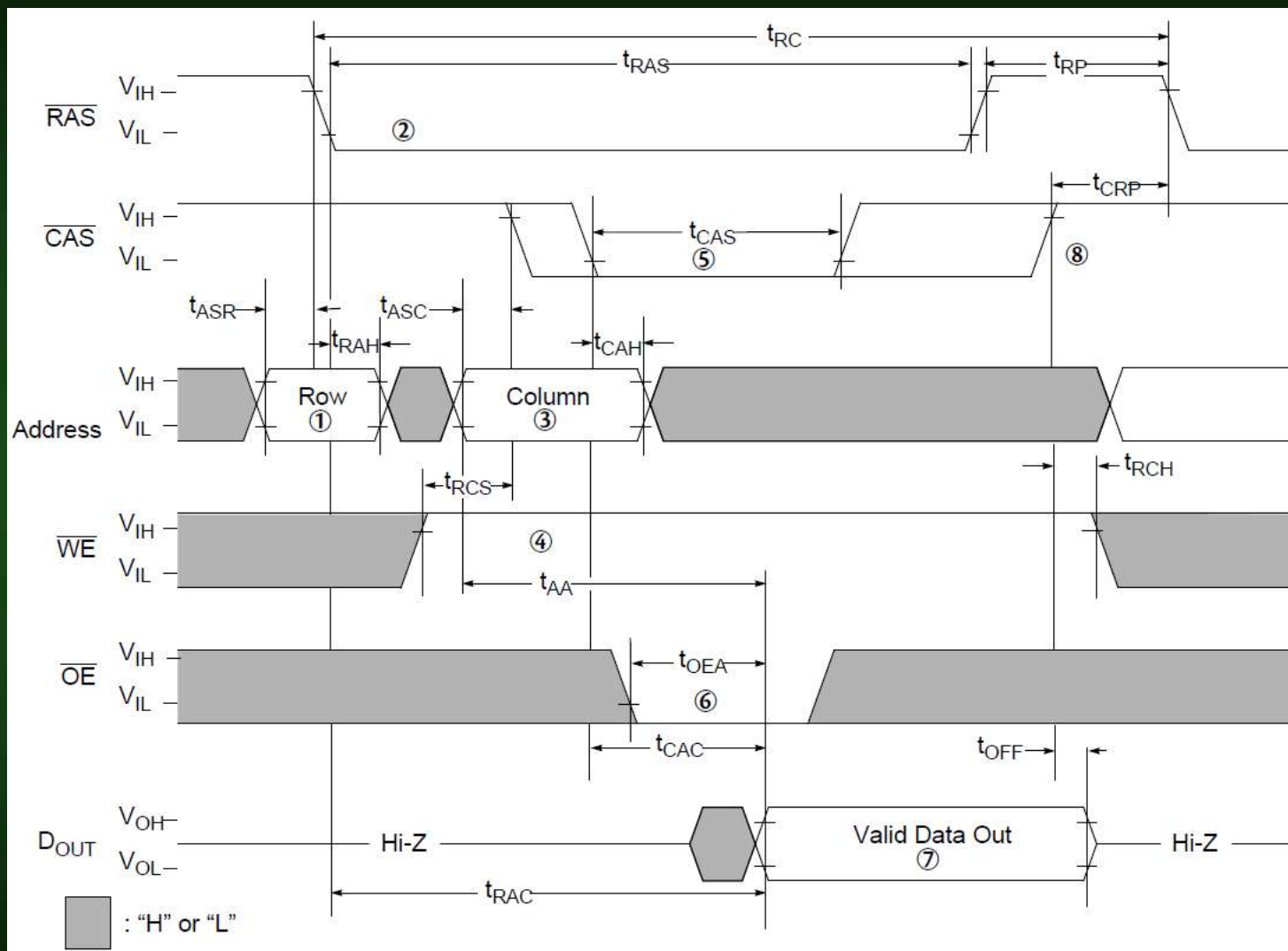
- **Tipikus memóriaelérés (olvasás):**
 - **Sorcím a cím-lábakon** → RAS jel leesik: sorcím rögzül a sor-cím-bufferben és az elérési tranzisztorok (sens amps) aktiválódnak;
 - a RAS jel stabilizálódásával a teljes sor celláinak értékét felveszik az elérési tranzisztorok;
 - **Oszlopcím a cím-lábakon** → CAS jel leesik: oszlop cím rögzül az oszlop-cím-bufferekben; a CAS stabilizálódásakor a kiválasztott rész töltődik az output bufferba.



A memória© Vadász, 2007.

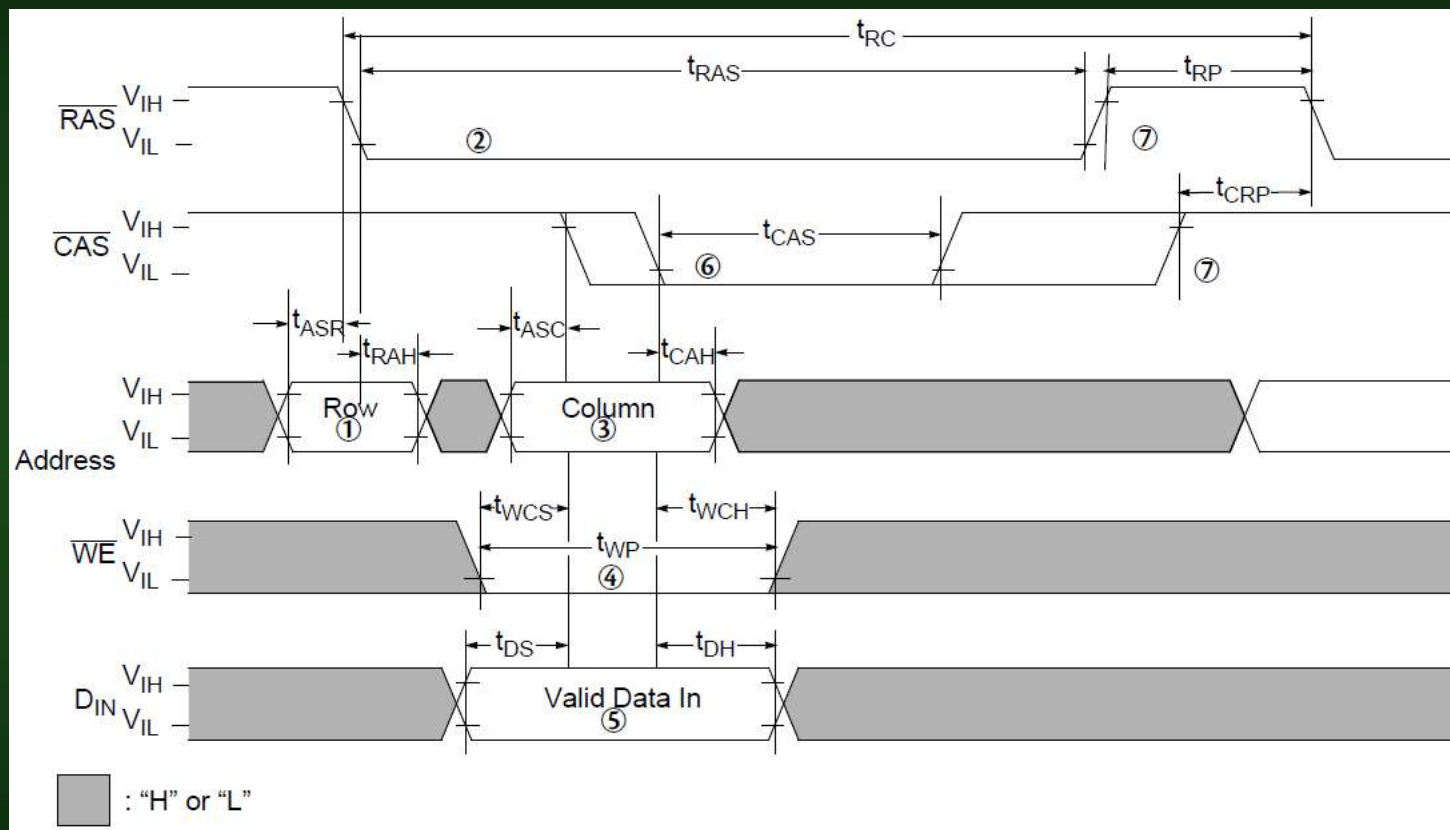
Ea7 14

DRAM operációk - olvasás



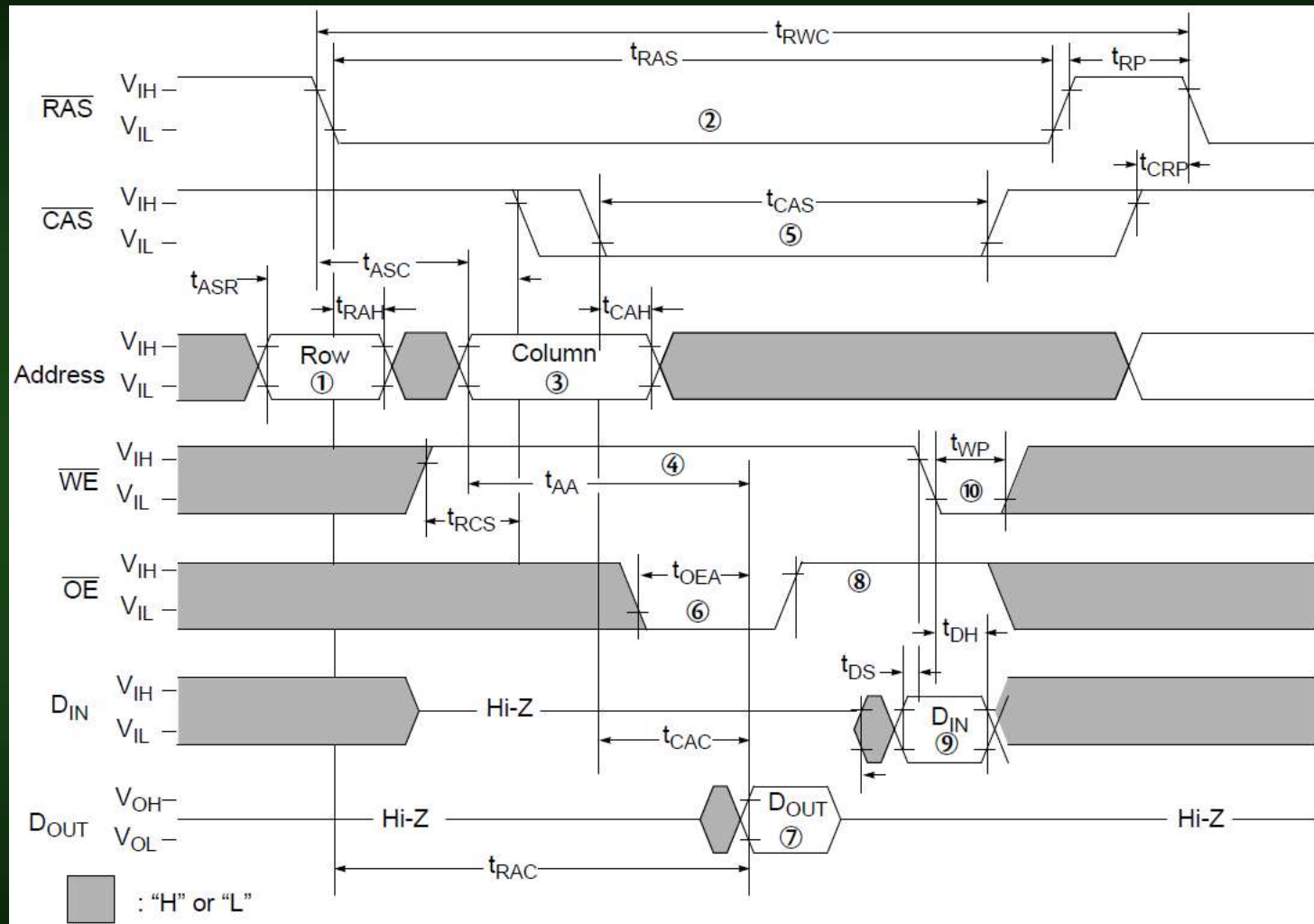
© IBM

DRAM operációk - írás



© IBM

DRAM – olvasás – módosítás - írás



© IBM

Technológiák

- **Fast Page Mode (FPM) RAM** (egy sorcímzés mellett több oszlopcímzés), **Extended Data Out (EDO)**
- **Burst Extended Data Output (BEDO) RAM** (egy sorcímzés, egy oszlopcímzés mellett 4 adatcella) (dual bank, de az EDO „halála” miatt elavult)
- **SDRAM: Synchronous DRAM** (mehetünk a 66-100MHz fölé)
A mem „zárolja” a CPU-tól jövő cím, adat és kontrol infókat, autonóm módon dolgozik (a system clock kontrollja alatt), mialatt a CPU csinálhat mást. Valamennyi idő (latency) után az outputon ott az eredmény (pipeline)...
- **ESDRAM (Enhanced ...):** a szokásos SDRAM lapkán kisebb SRAM cache is van (akár 200 MHz is)
- **DDR (Double Data Rate):** 2-burst-deep prefetch buffer,
DDR2: 4, DDR3: 8, DDR4: 8

Általános INFORMATIKAI Tanszék

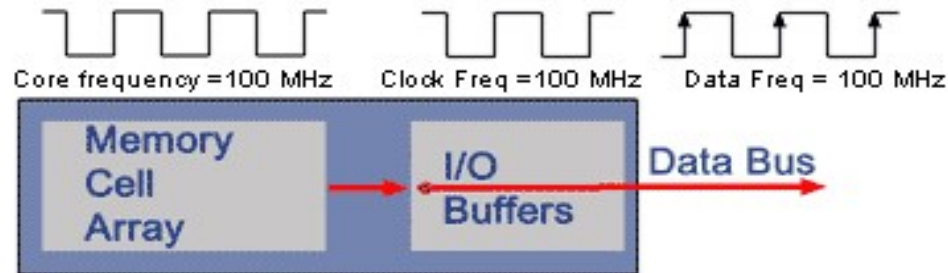


Általános INFORMATIKAI Tanszék

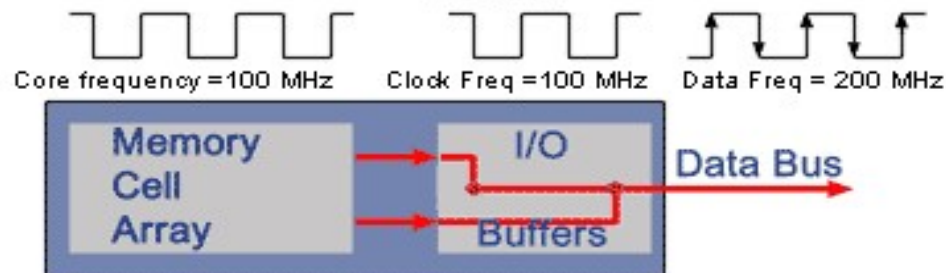


DDR SDRAM

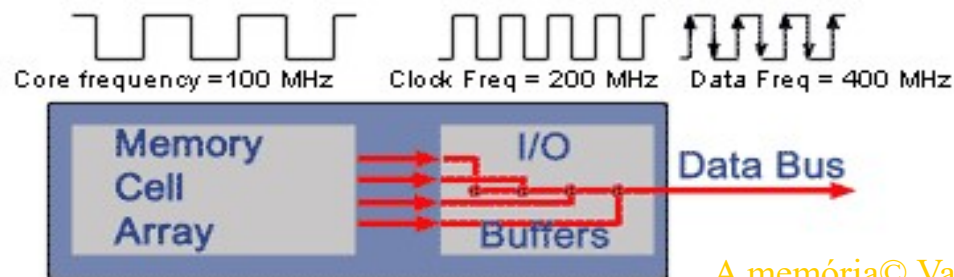
SDRAM



DDR I



DDR II



DDR3 (2007): 1.5 V

Memory: 100 – 267 MHz

I/O Bus: 400 - 1067 MHz

Data [MT/s]: (Transfer)

DDR3-800 - DDR3-2133

DDR4 (2014): 1.2 V

Memory: 200 - 400 MHz

I/O Bus: 800 - 1600 MHz

Data [MT/s]:

DDR4-1600 - DDR4-3200

DDR5 (2020): 1.1 V

2400 - 6200 MHz

DDR5-4800 - DDR5-7200

A memória© Vadász, 2007.

Ea7 21

Roadmap ...

Bemutató év	Technológia	„Sebesség” határ	Max Bps
1987	FPM	50 ns	176 MBps
1995	EDO	50 ns	186 Mbps
1997	PC66 SDRAM	60-66-83 MHz	240 MBps
1998	PC100 SDRAM	100 MHz	400 MBps
1999	RDRAM	800 MHz	1,6 GBps
1999/2000	PC133 SDRAM	133 MHz	532 MBps
2000	DDR SDRAM	266 MHz	1064 MBps

A piaci arányok ...

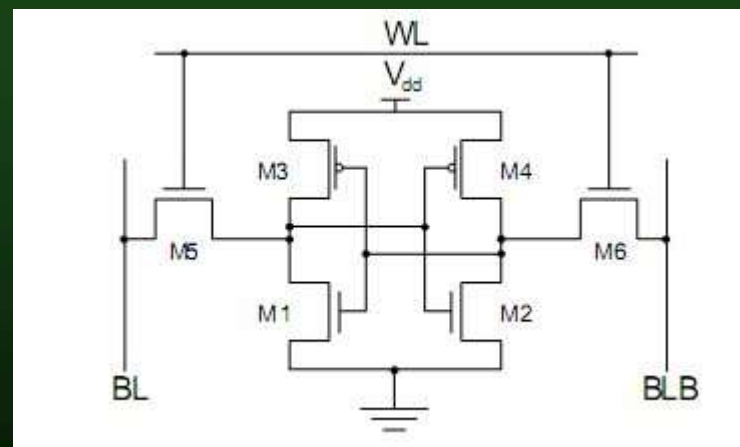
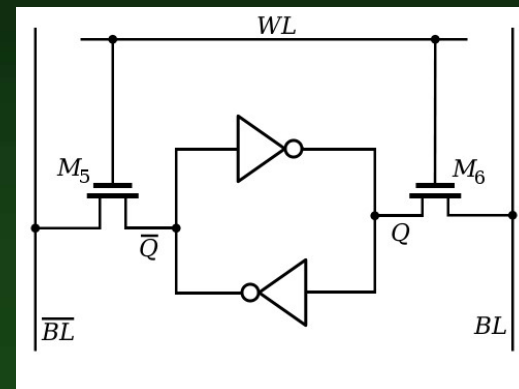
	SDRAM	DDR	RDRAM	EDO	DDR2
2002	55%	39%	5%	1%	
2003	13%	81%	3%		3%
2004	8%	83%	2%		9%
2005	5%	58%	2%		35%

Félvezető tárolók ...

SRAM: Static Random Access Memory

- ezek is írhatók, olvashatók,
- **random** elérésűek,
- 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 regiszterei)

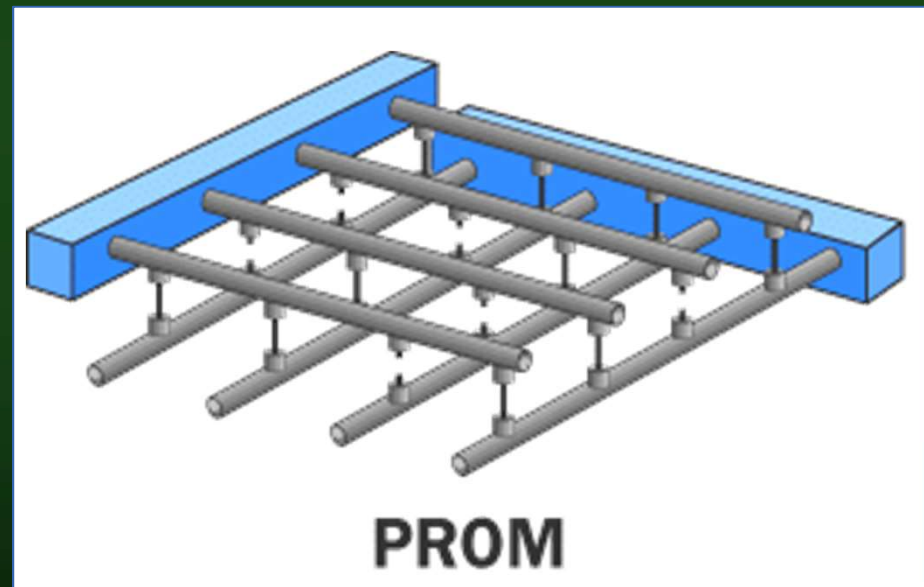


Félvezető tárolók ...

- **ROM: Read Only Memory: csak olvasható.**
 - Ez is cellák hálózata, oszlop-sor tömbbe rendezve,
 - diódák a cellákban, összekötést adnak (1 bit), nincs összekötés (0 bit).
 - **Kikapcsolva nem felejt a lapka.**
 - **Elérések itt is random jellegűek.** (oszlop-sor kiválasztás).
 - Címtartományuk, címezhetőségük egybeeshet a DRAM-okkal: **a (központi) memória részben ROM-okból, részben DRAM-ból állhat.**
- **PROM, EPROM: Beégethető a tartalom, EPROM-nál törölhető és újraírható ...**

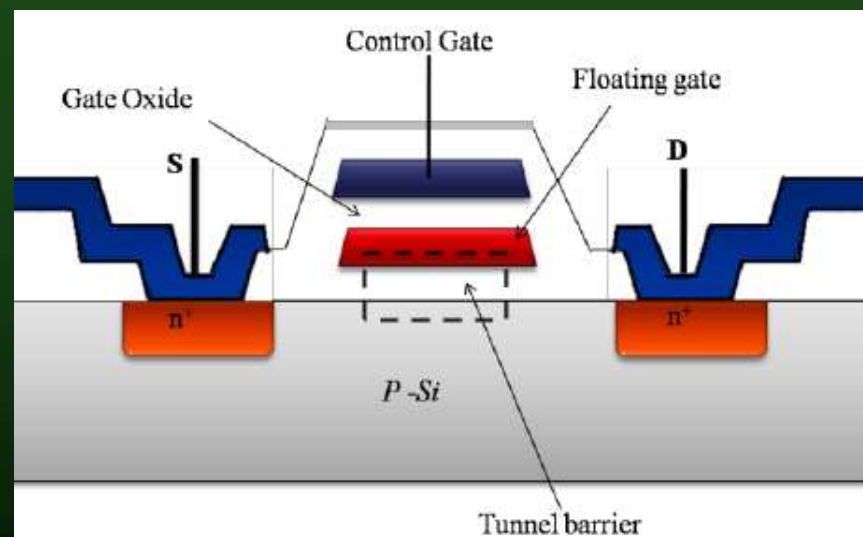
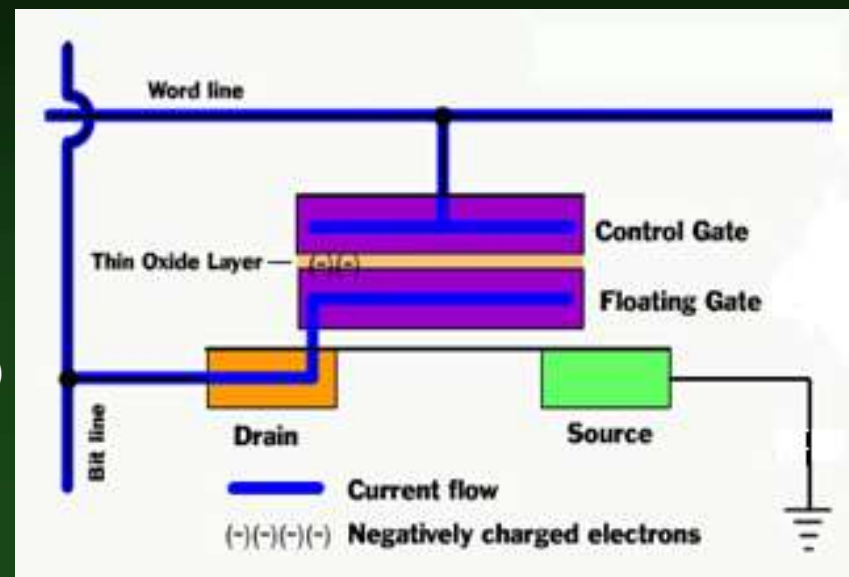
Félvezető tárolók ...

- **PROM** (Programmable ROM)
 - Sor-oszlop háló,
 - a cellákban „olvadó biztosíték” (fuse),
 - Beégethető a tartalom.

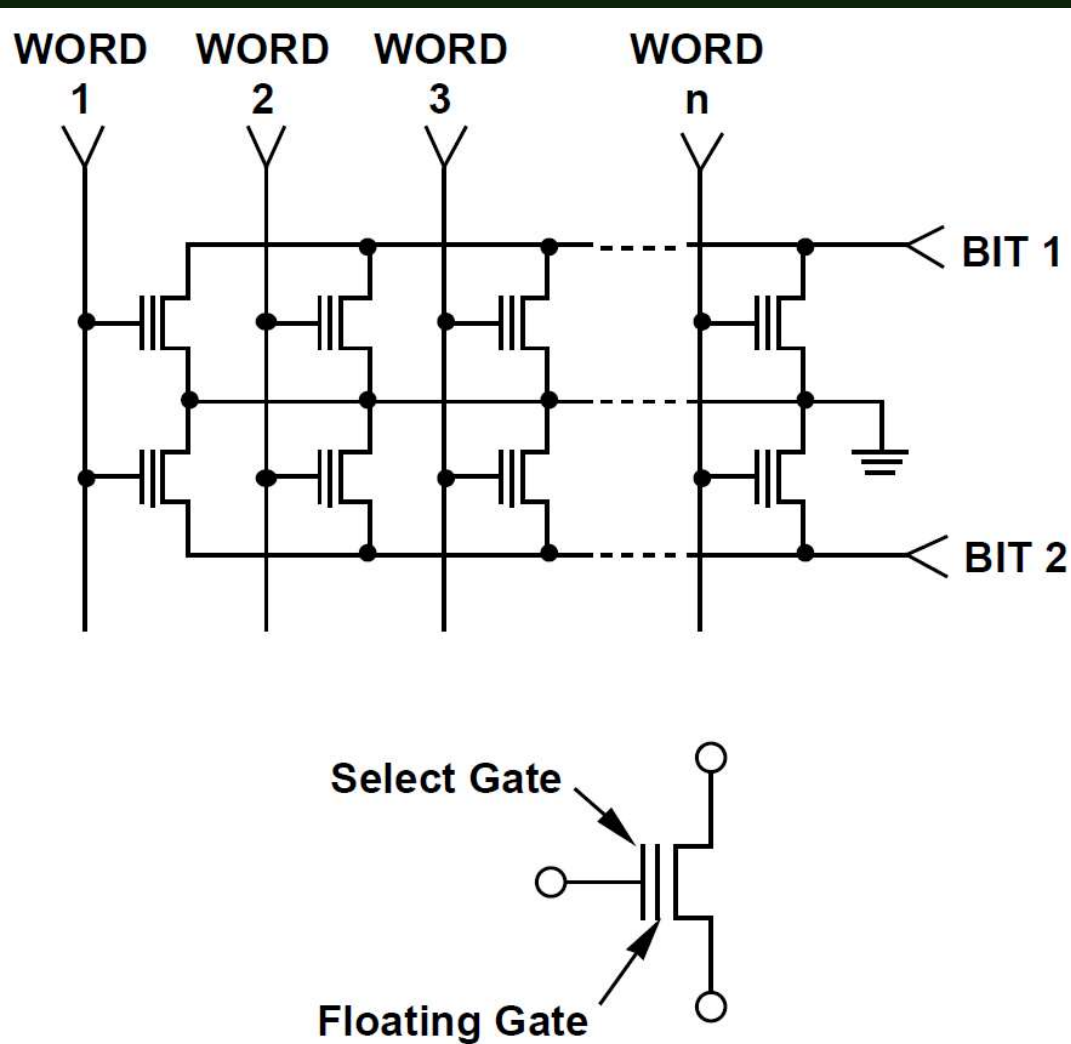


Félvezető tárolók ...

- **EPROM** (Erasable PROM) törölhető és újraírható ...
 - Cellákban két kapus FET (Field-Effect Transistor)
 - Floating gate (negatívra töltve zár)
 - Control gate
 - Köztük oxidréteg
 - 1: „összekötve” a bit-line és a föld
 - 0: a kapu „zárva”

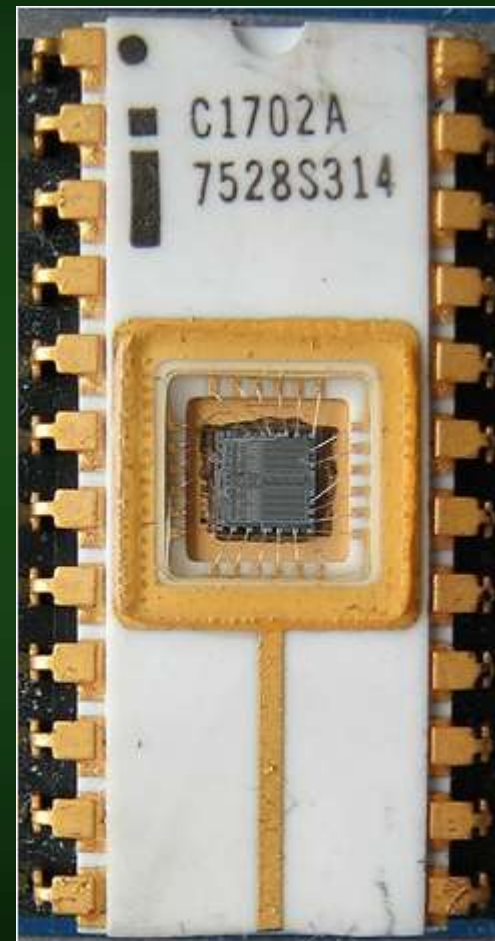


Félvezetű - EPROM



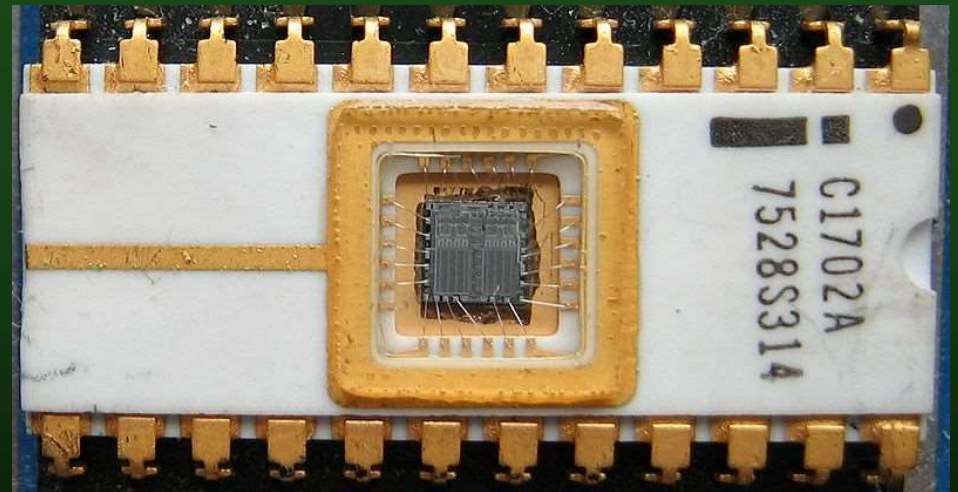
Source: ICE, "Memory 1997"

19051



Félvezető tárolók ...

- **EEPROM (Electronical Erasable PROM)**
 - Mint az EPROM, de nem UV fény „töröl”.
 - Lassú, mert 1 bájtot lehet egyszerre törölni, újraírni ...
- **FLASH memory**
 - Ez valójában EEPROM, de egyszerre blokkot (512-x kilo bájtot) lehet újraírni.
 - Már elég gyors.



Félvezető tárolók – Solide State Drive

FLASH memory

NOR flash

Nem sor, hanem csak 1 bit egyszerre

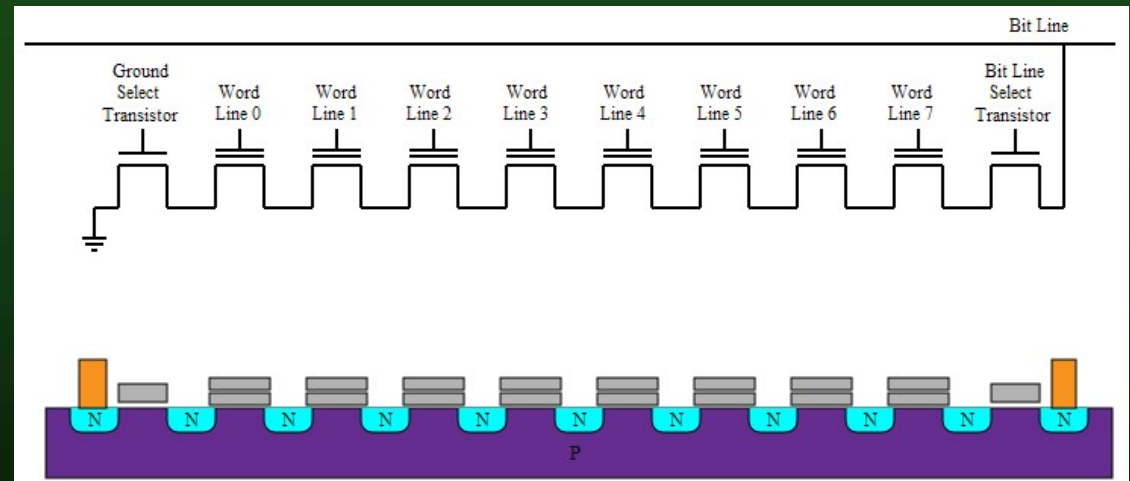
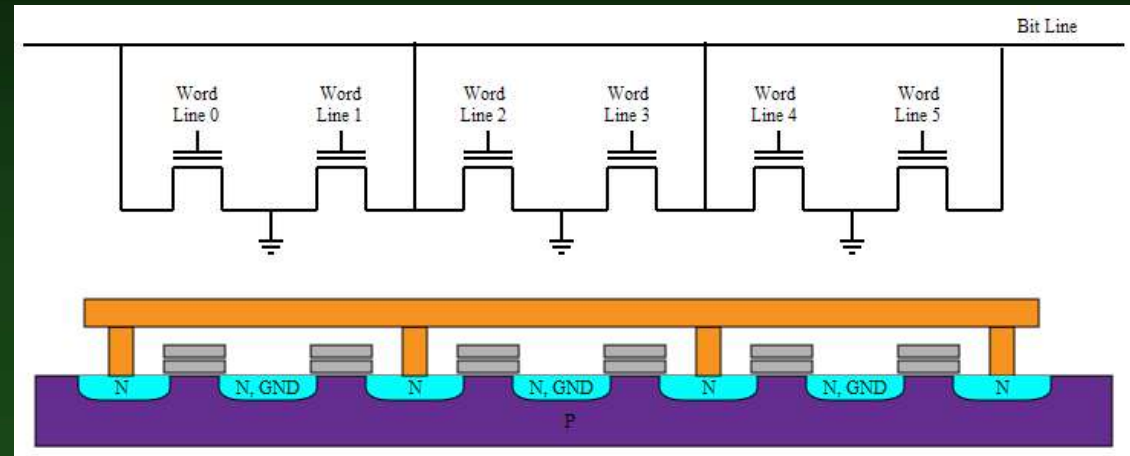
Memory wear: 100 000
program-erase (P/E) cycles

Erase sets (all) bits,
programming can only
clear bits:

1111 – 1110 – 1010 – 0010, **NAND flash**
finally 0000

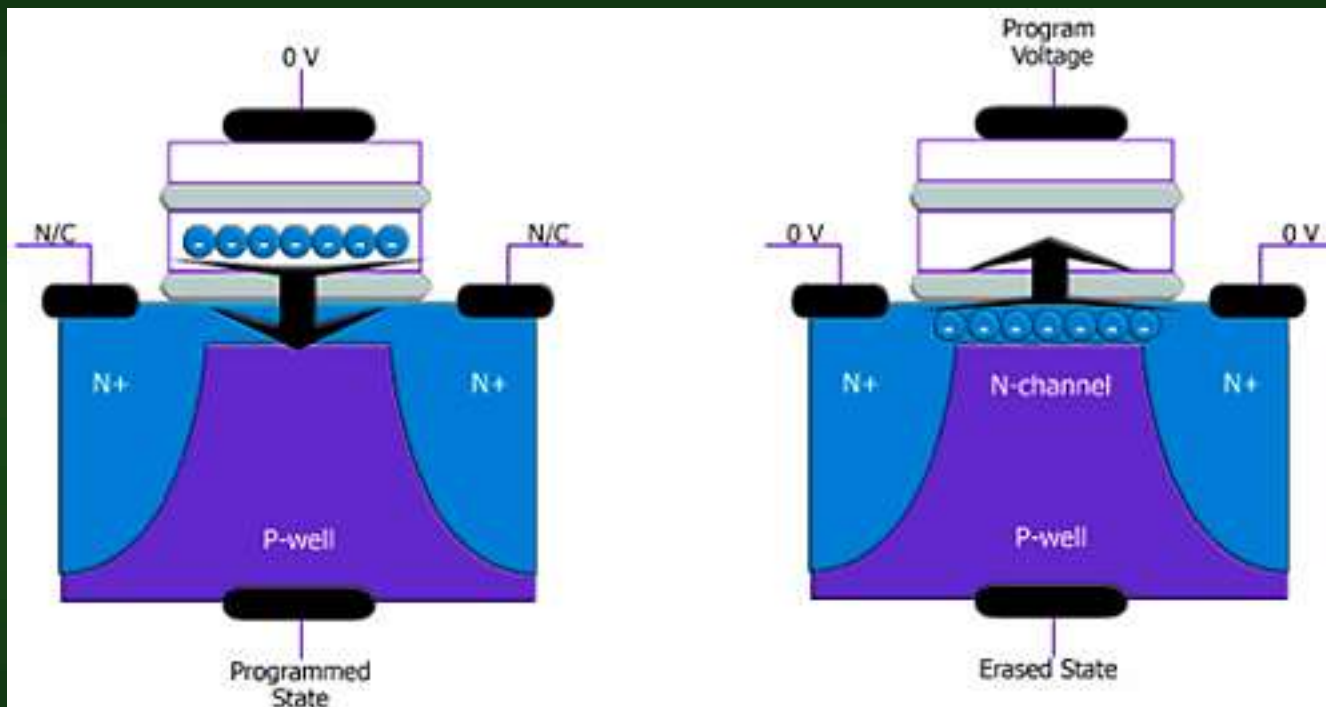
Cell read-out:

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.



Félvezető tárolók – Solide State Drive

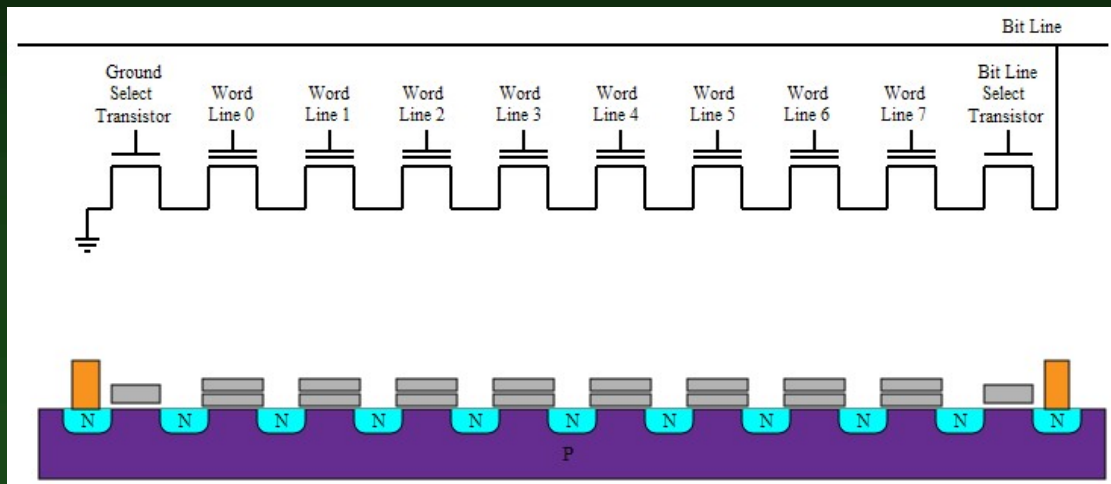
NAND Flash – programozás



(tunneling)

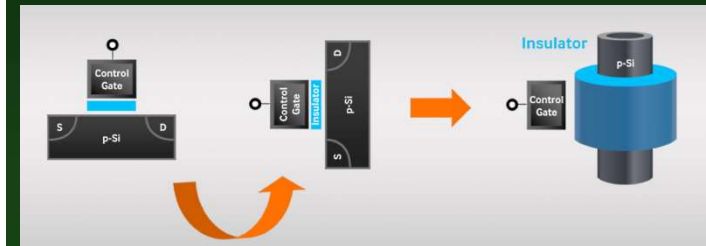
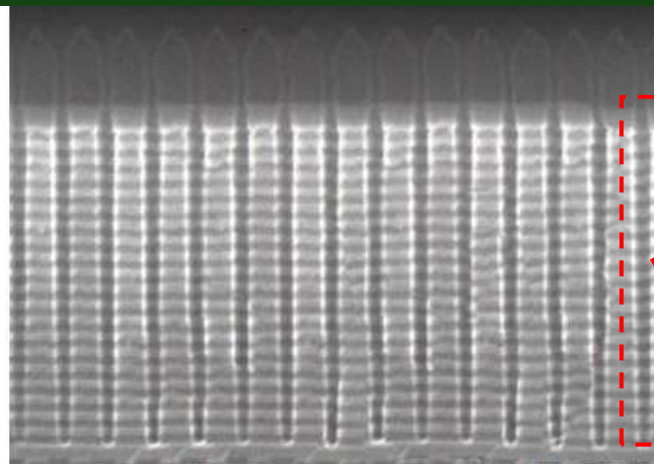
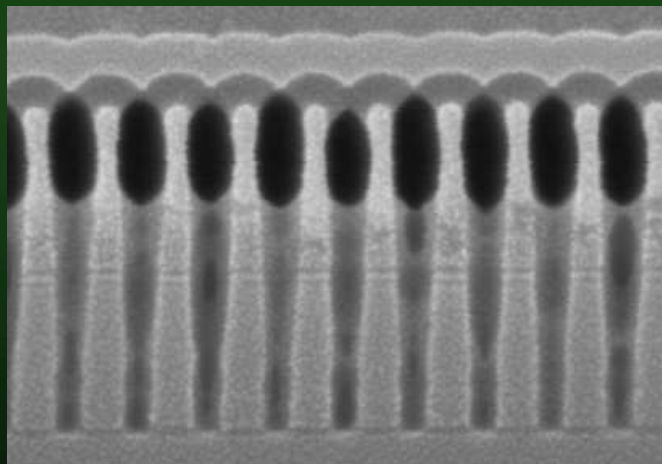
Solide State Drive – kapacitás növelés

NAND Flash, elrendezés



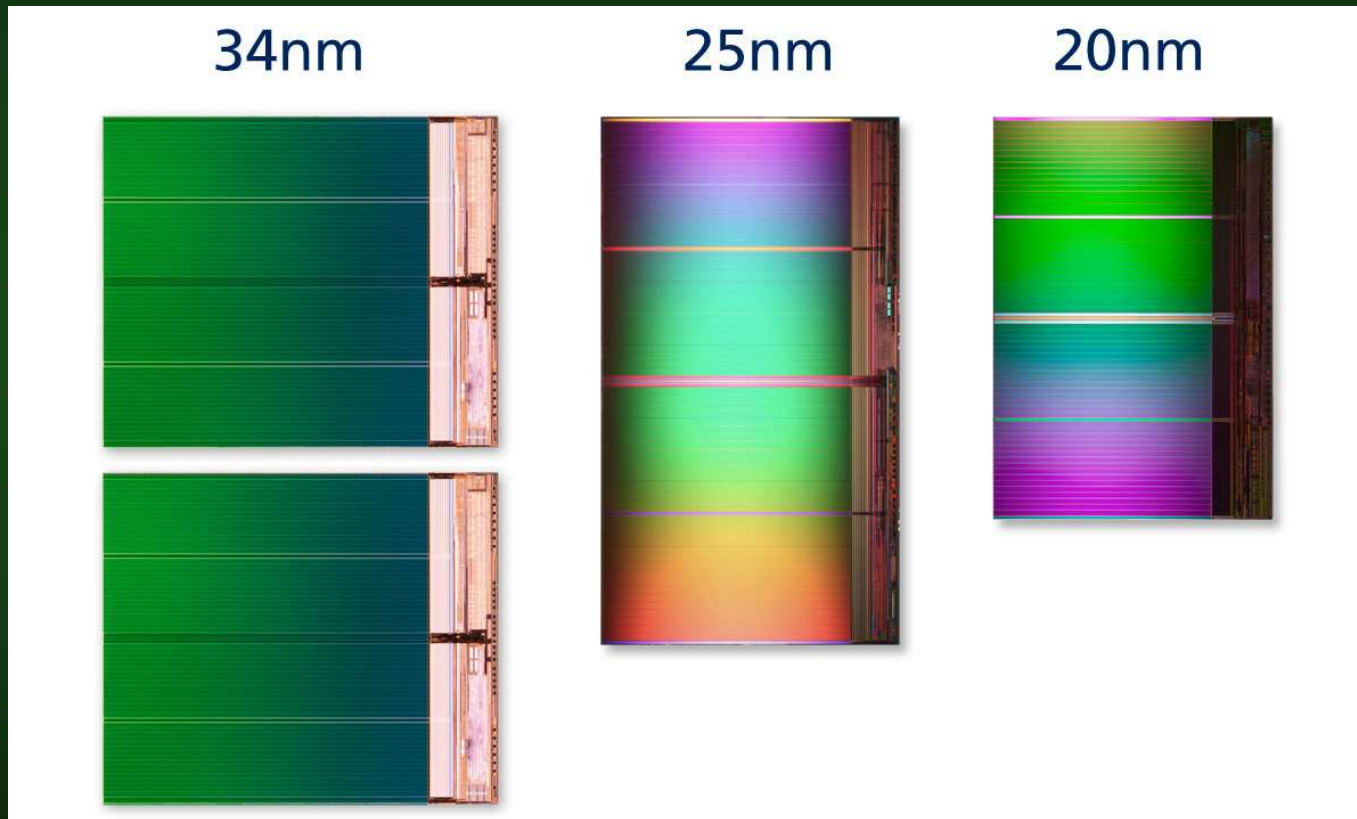
2D

3D



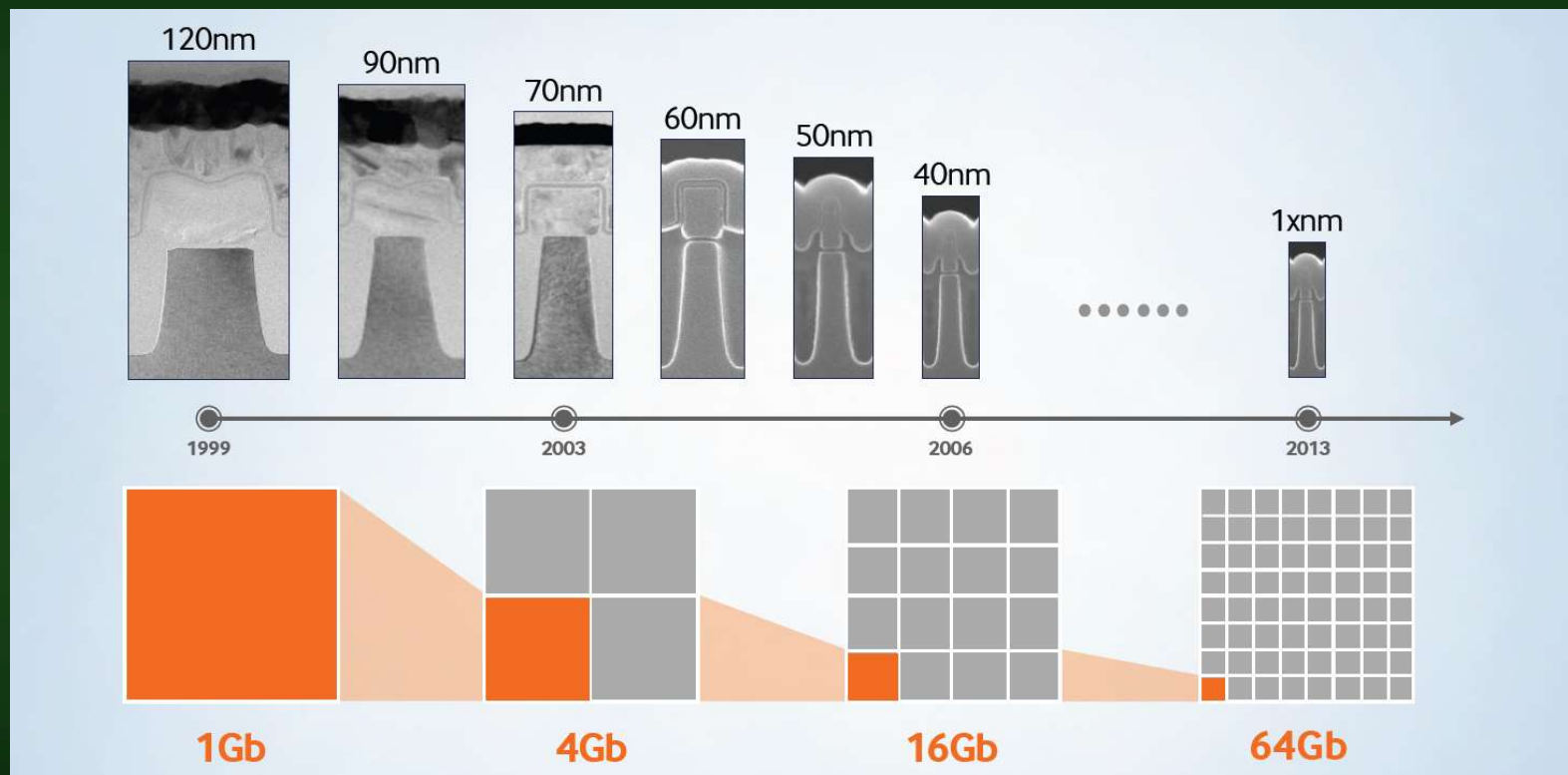
Solide State Drive – kapacitás növelés

NAND Flash – csíkszélesség (azonos kapacitás esetén)



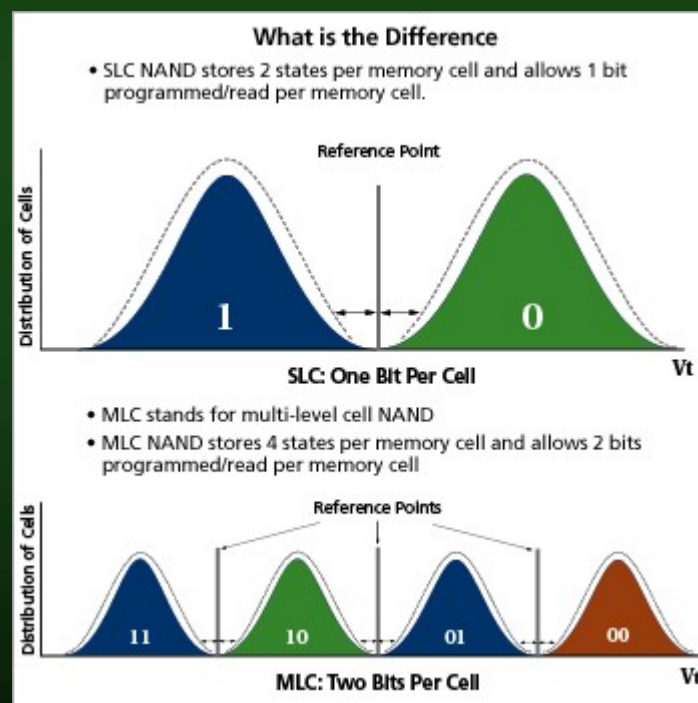
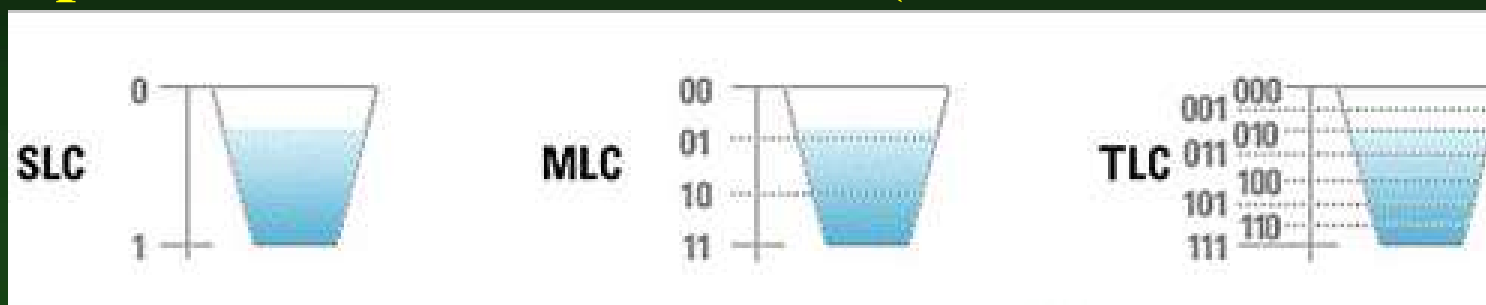
Solide State Drive – kapacitás növelés

Csíkszélesség – gond az élettartam (endurance)



Solide State Drive – kapacitás növelés

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

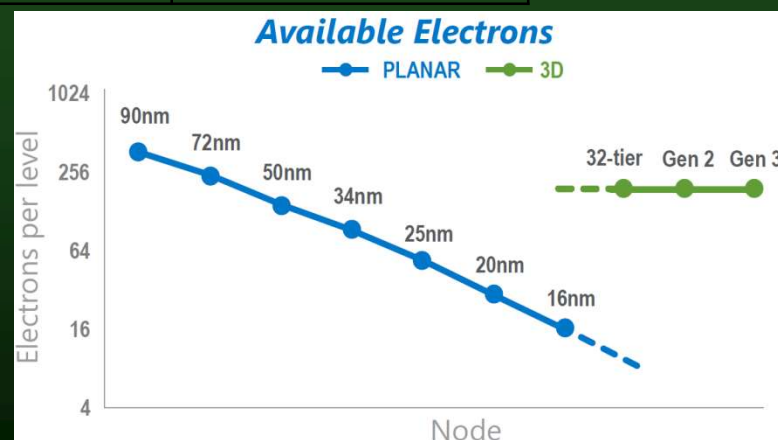


Solide State Drive – kapacitás növelés

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 μ s	50 μ s	75 μ s
Írási idő	200-300 μ s	600-1200 μ s	900-1600 μ s
Törlési idő	1,5-2 ms	2-3 ms	4,5-5 ms

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

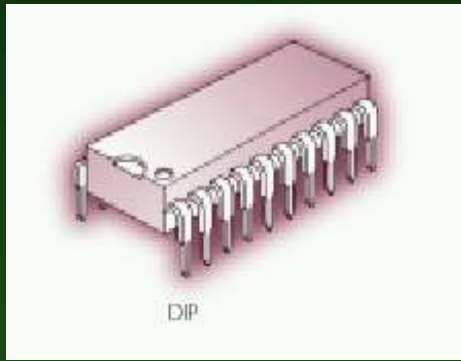


Memória (összefoglalás)

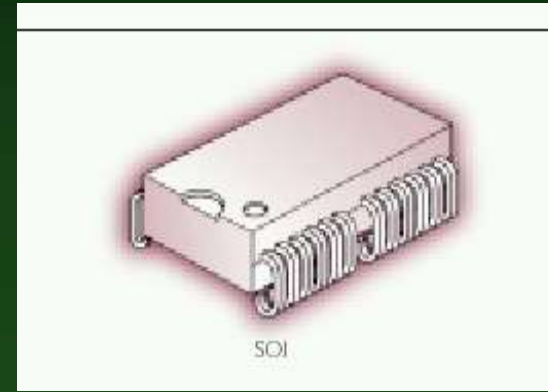
- **RAM**
 - **DRAM** (Tokozott magas integráltságú áramköri lapkák)
 - FPM, EDO, BEDO DRAM
 - SDRAM, ESDRAM
 - DDR, DDR2, DDR3, DDR4
 - RDRAM
 - **SRAM**
 - **ROM**
 - PROM
 - EPROM, EEPROM
- **FLASH**

A DRAM lapkák tokozása

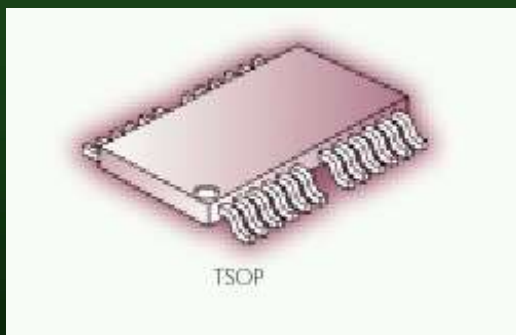
DIP: Dual In-Line Package



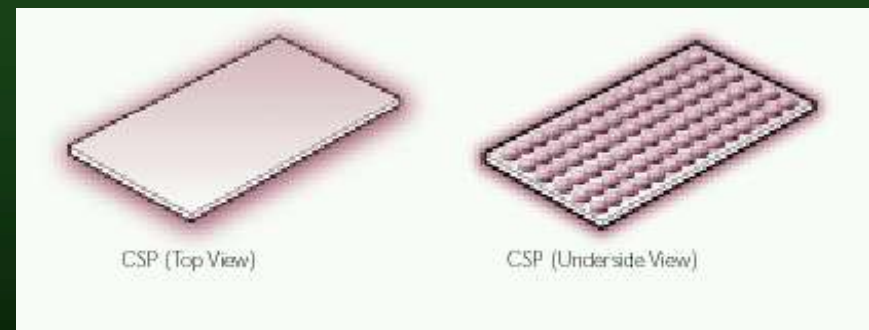
SOJ: Small Outline J-lead



TSOP: Thin Small Outline Package



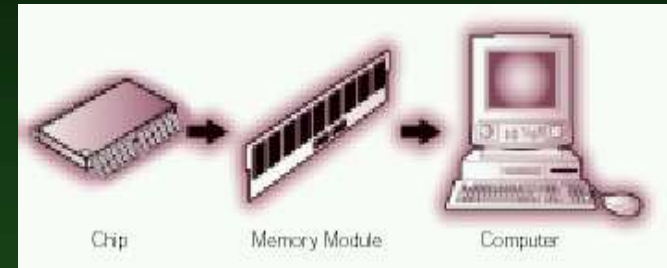
CSP: Chip Scale Package



Ball grid array (BGA)

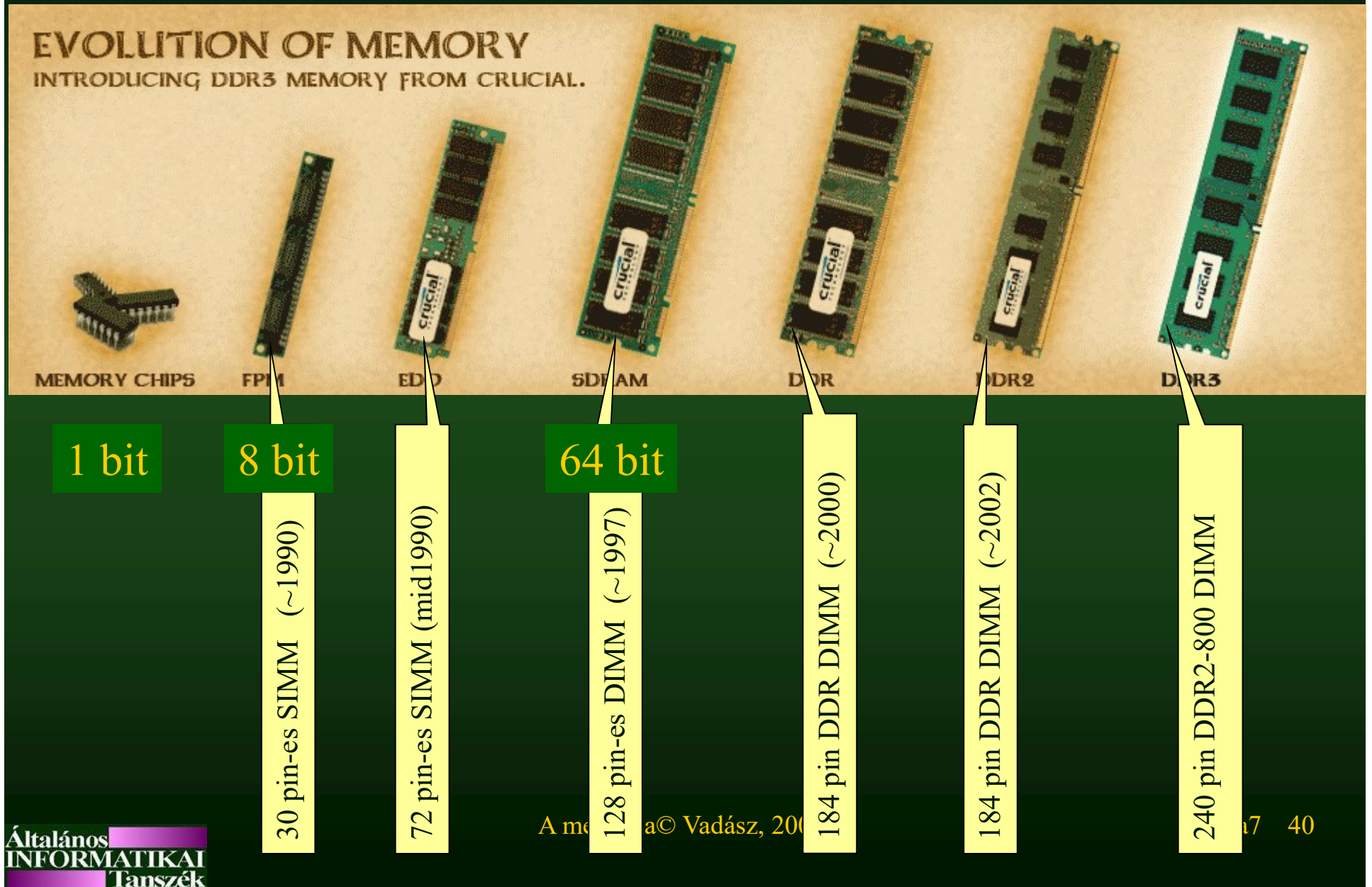
Memória modulok

- **Memória modulok (MM) hordozzák a lapkákat. Szabványok. Az alaplap(ok) foglalatába helyezhetők.**
- **Single In-line MM (SIMM)**
 - 32 bites CPU-khoz
 - 72 érintkező a modulon
- **Dual In-line MM (DIMM)**
 - 128, 184, 240 láb, 64 bites CPU-khoz is
- **Small Outline DIMM (SODIMM)**
 - 144 (72) érintkező, sokkal kisebb, notebook-okhoz
- **Rambus (RIMM, SORIMM)**
 - 16 bites adatösvény, pipelining, gyors



**DDR400 (800 MHz-es
frontside bus chipset-
hez)**

History ...



Bár növeljük a frekvenciát...

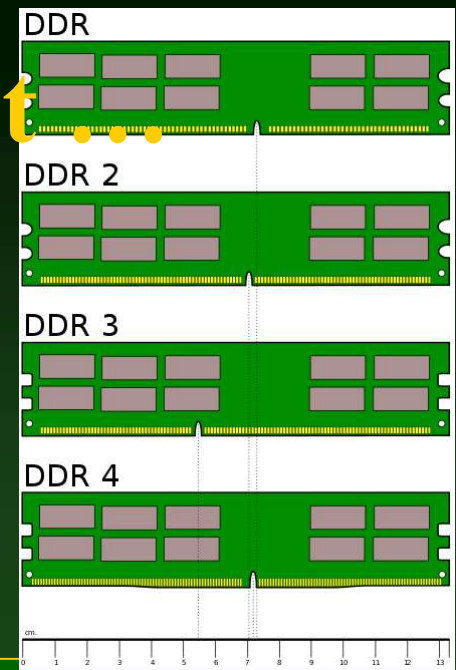
- A memória időzítések (amit lehet, kell állítani) a nagyobbak a rosszabbak

SDRAM: 2,5-2-2-5 (t_{CL} - t_{RCD} - t_{RP} - t_{RAS})

DDR (200): 2,5-3-3-8 2.5V 2-burst-deep prefetch buffer

DDR2 (400): 5-5-5-15 1.8V 4-burst-deep

DDR3 (800): 9-9-9-24 1.5V 8-burst-deep



t_{CL}	CAS Latency: a CAS leesése utáni várakozás ideje, szokásosan 2, 2.5, 3 ciklus. Miután ez letelik, az adat a DQ lábakra került.
t_{RCD}	RAS to CAS Delay: RAS leesés utáni várakozás, amíg a CAS jelet lehet küldeni. Szokásosan 2, 3 ciklus
t_{RP}	RAS Precharge: Ezalatt az idő alatt a vezérlő ismét deaktiválja a sort.
t_{RAS}	Active to precharge delay: ezalatt a sor aktív kell legyen, Csak ezután lehet deaktiválni. 5 – 8 ciklus szokott lenni. Teljesülnie kell: $t_{RCD} + t_{CL} < t_{RAS}$

http://www.tomshardware.com/2007/10/03/pc_memory/

Ma: **CAS latency (CL)** Clock cycles between sending a column address to the memory and the beginning of the data in response P1: CL14

Növeljük a teljesítményt ...

- Szélességnövelés, pipelining, bursting
- Dual channel (párhuzamosítás)



If you want maximum memory performance, you should install two memory modules into two different memory channels to have it run in dual-channel mode. This doubles memory bandwidth by providing a 128-bit data bus.



Installing two memory modules into the same channel of the memory controller will force it into single-channel mode.

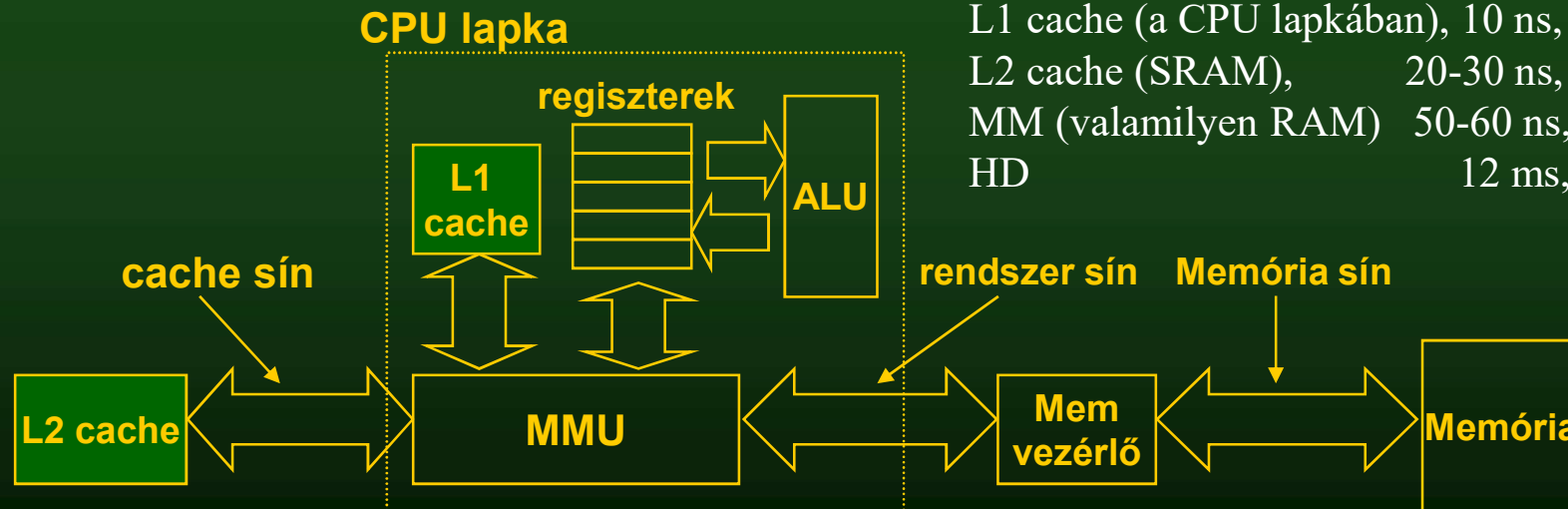
- Gyorsítótárak (cache) alkalmazásával
- Asszociatív tárok alkalmazásával

A gyorsítótárak (caches)

- **A programok lokalitása: tapasztalat**
- **Kisebb kapacitású, de a CPU-hoz “közelebbi” és gyorsabb (SRAM) memória, amiben**
- **a központi memória tartalom egy része szintén megvan.**
- **A CPU “egyszerre” címzi a cache-t is, a központi memóriát is:**
 - ha találat van a cache-ben, csak onnan vesz!
 - gond lehet az **adat-konzisztencia!**

Nagyságrendek:

L1 cache (a CPU lapkában),	10 ns,	4-16 KB
L2 cache (SRAM),	20-30 ns,	2-16 MB
MM (valamilyen RAM)	50-60 ns,	1-8 GB
HD	12 ms,	0.5-4 TB



A memória© Vadász, 2007.

A lokalitás elve

Principle of Locality

- **Processzek statisztikailag megfigyelhető tulajdonsága, hogy egy idő-intervallumban címtartományuk egy szűk részét használják ...**

- **Időbeli lokalitás**

- Hivatkozott címeiket újra ...

A 80/20 szabály: egy processz a kódja-adatai 20 %-át használja életének 80 %-ban

- **Térbeli lokalitás**

- Közeli címeiket ...

- **Az elv érvényesülése miatt van értelme kisebb, de gyorsabb átmeneti tárolók használatának ...**

- Gyorsítótár; munkakészlet; TLB; diszk buffer cache ...

Lokális elve (illusztráció)

- **Processz futása során instrukciók sorozatát hajtja végre (instrukció hivatkozás sorozat),**
- **Az instrukciókban memória-hivatkozások lehetnek (adat hivatkozás)**
- **A hivatkozások sorozata a hivatkozási lánc (Reference String):**

$\forall \omega = r_1, r_2, \dots r_t, \dots r_T$ $\# r_t$: instr. vagy adat hiv.

for(i=1; i<=n; i++) a[i]=b[i]+c[i]; // n=1000 legyen

A gépi nyelvű program regiszteres gépen kb:

Lokalitás elve (illusztráció)

Cím	Kód/adat	Megjegyzés
4000	LOAD (R1), ONE	R1 index reg inicializálás
4001	LOAD (R2), n	R2 határ reg inicializálás
4002	COMP R1, R2	$i > n$ tesztelés
4003	JG 4009	Feltételes ugrás
4004	LOAD (R3), B[R1]	B[i] betöltés R3-ba
4005	ADD (R3), C[R1]	Összeadás
4006	STOR A[R1], (R3)	A[i]-be tárolás
4007	ADD (R1), ONE	I inkrementáció
4008	JUMP 4002	Ciklus újra
4009	...	Ciklus után

Cím	Kód/adat
6000-6999	Tárolóhely A számára
7000-7999	Tárolóhely B számára
8000-8999	Tárolóhely C számára
9000	ONE
9001	Tárolóhely n számára

Ekkor a hivatkozási lánc a következő (összesen 9 instrukció-hiv. ebből 7-re 1000-szer:

4000, 9000, 4001, 9001,

(4002, 4003, 4004, 700i, 4005, 800i, 4006, 600i, 4007, 4008)¹⁰⁰⁰

4002, 4003, 4009

Lokalitás elve (illusztráció)

- Ugyanez virtuális memória menedzselésnél, lapozós rendszerben ...
- Legyen a lapméret 1000, a virtuális cím: $v = (p, o)$
(pl a[16] 6015 címe ekkor $v = (6, 15)$)
- Ekkor a lapok hivatkozási lánc (összesen 5 lap):
4, 9, 4, 9,
(4, 4, 4, 7, 4, 8, 4, 6, 4, 4)¹⁰⁰⁰
4, 4, 4
- Kisméretű munkakészlet esetén sem valószínű a laphiba.

Gyorsítótárak szintjei

A gyorsítótár koncepció minden két szint között lehetséges, ahol a felsőbb szint gyorsabb, bár kisebb kapacitású.

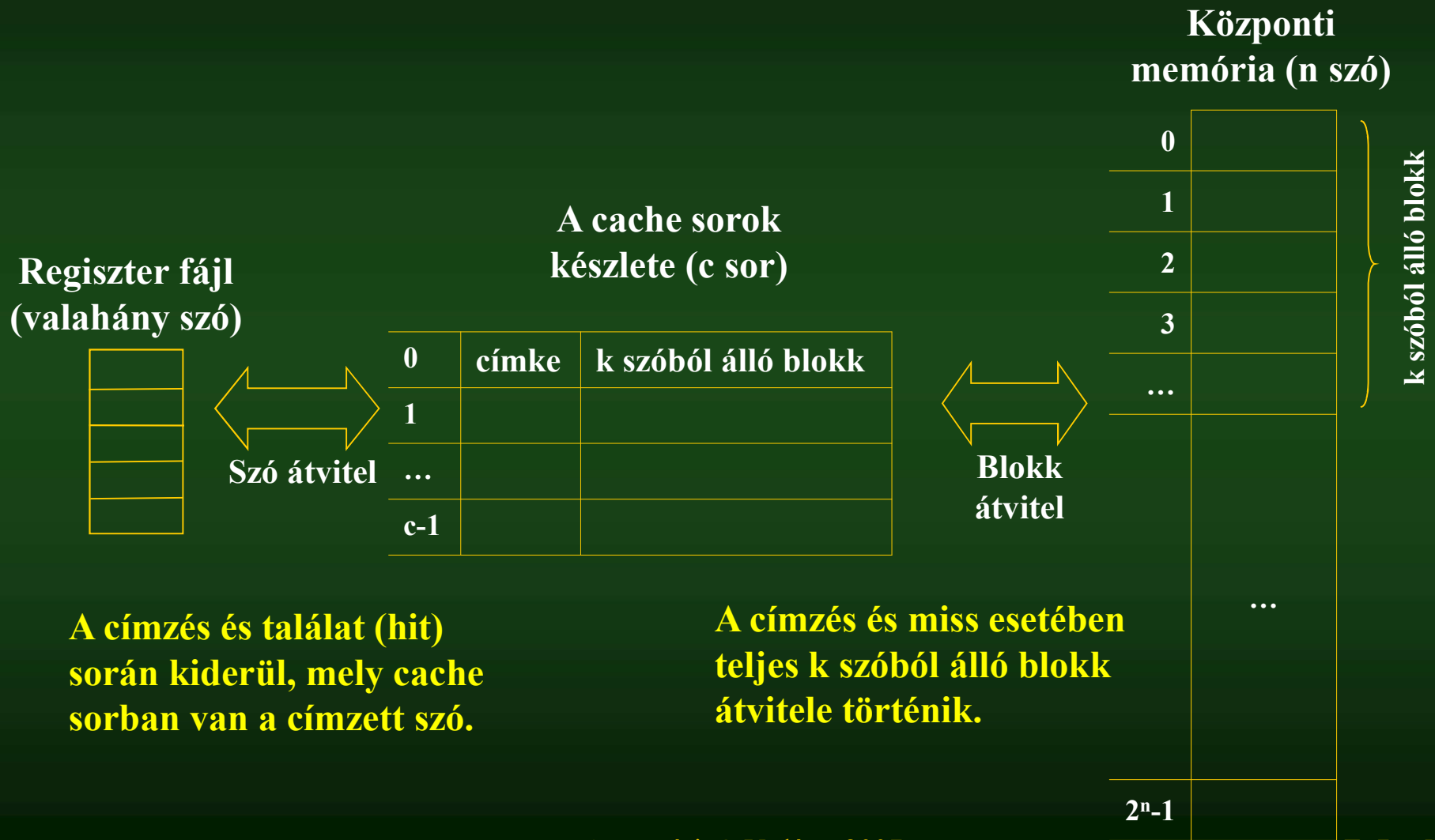
Az alsóbb szinthez fordulás gyakorisága csökkenni fog.

Level 1	CPU-n belül	~ 10 ns	4 – 16 KB
Level 2	SRAM cache sínen	~ 20 -30 ns	2 – 16 MB
Központi memória	DRAM memória sínen	~ 60 - 195 ns	128 – 8192 MB
Diszkek	Mágneses+mech I/O sínen	~ 10 - 12 ms	0.5 – 6 TB
Internet	Hálózaton	~ 1 s -	szinte határtalan

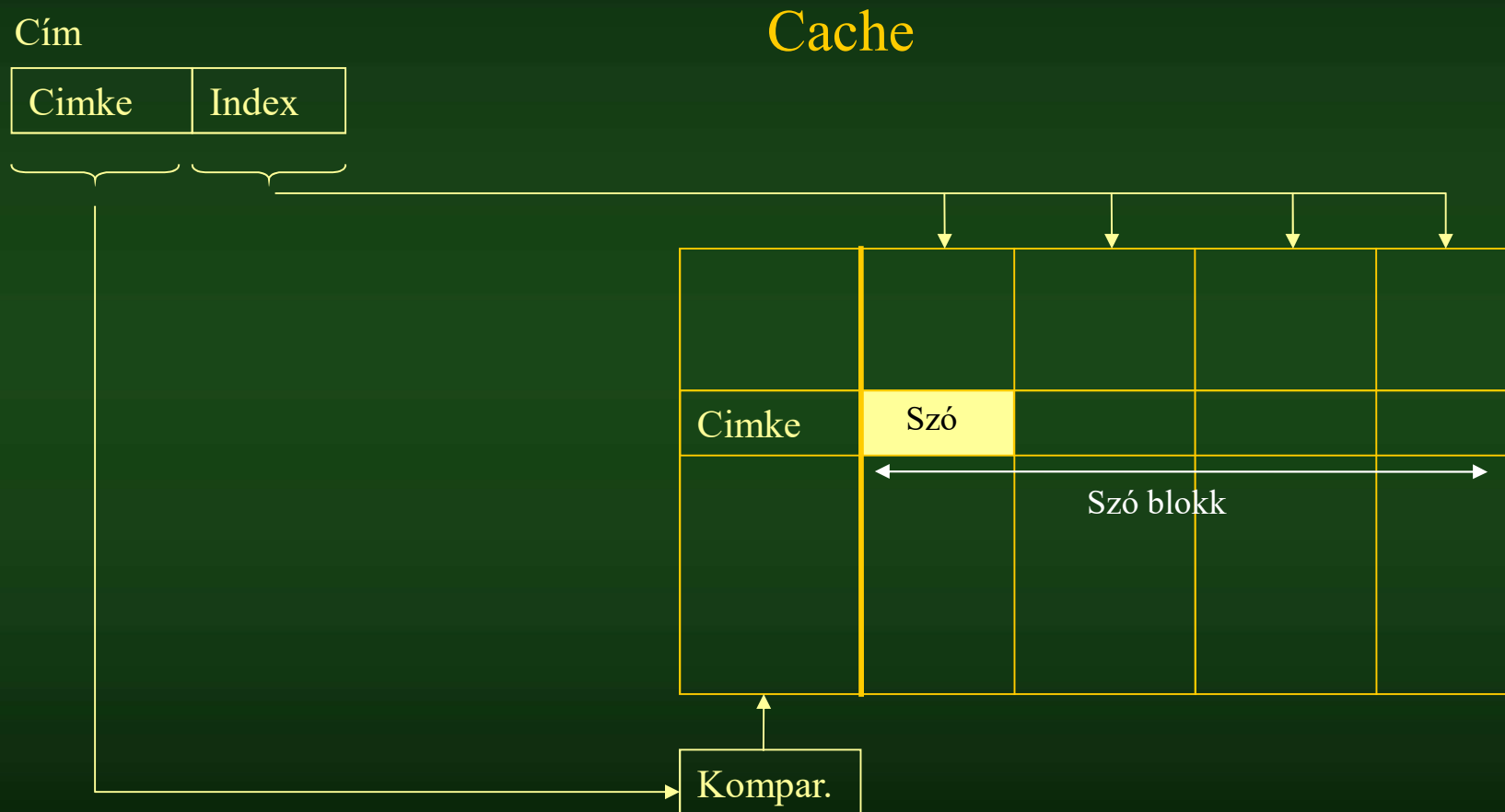
A klasszikus gyorsítótár

- **A CPU és a központi memória között**
- **Hardveres megoldás, még az OS számára is láthatatlan**
- **Mostanában két szintes (L1, és L2 szint), néha három**
- **Gyakori a szétválasztott adat- és instrukció gyorsítótár megoldás (Neumann elvet sért, de gyorsít!)**
- **A cache tervezés során ügyelni kell az adat konzisztenciára (ugyanaz a tartalom mind gyorsítótárban, mind a központi memóriában)**
 - Címzés során mind a cache, mind a memória címeződik.
 - Ha találat van a cache-ben (cache hit). Szó transzfer.
 - ha nincs találat (cache miss). Cache miss penalty fogalom. Blokk transzfer.
 - Memória írás (store) probléma: konzisztencia biztosítás

Regiszter, cache és memória struktúra



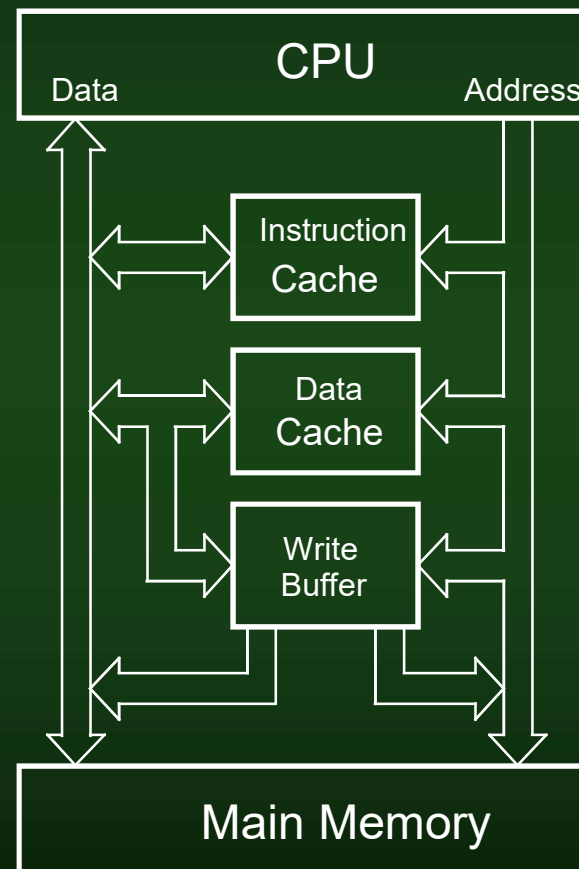
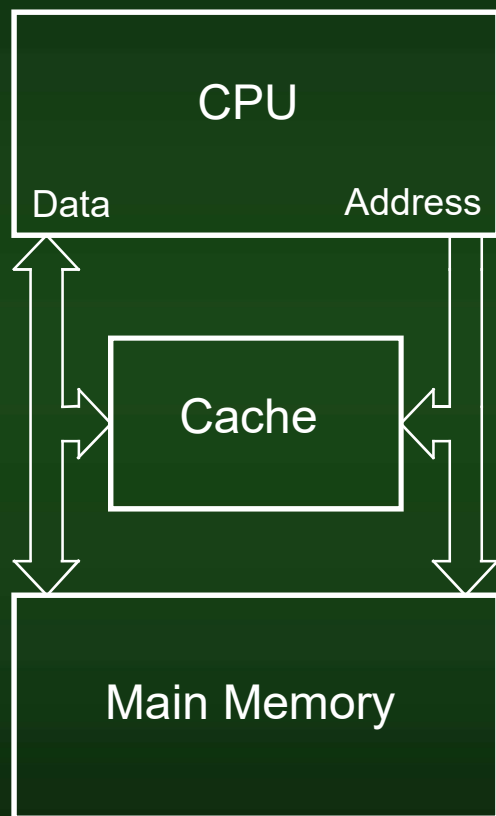
Blokkos cache



A cache tervezés

- **A cache mérete**
- **A blokk mérete**
 - Újonnan felhozott adat használatának valószínűsége
 - A cache-ből kicsorduló adat újrafelhasználásának valószínűsége
- **Mapping: mely cache helyet foglaljon el egy-egy blokk**
- **Replacement algoritmus: mely blokk kerüljön ki a cache-ből, ha egy újnak kell a hely (LRU: Least Recently Used)**
- **Write Policy (store során)**
 - Memóriába is írjunk, ha cache blokkba írtunk,
 - **Csak akkor írjunk a memóriába, amikor a blokk helyére új kerül (replacement). Write Buffer-es cache.**

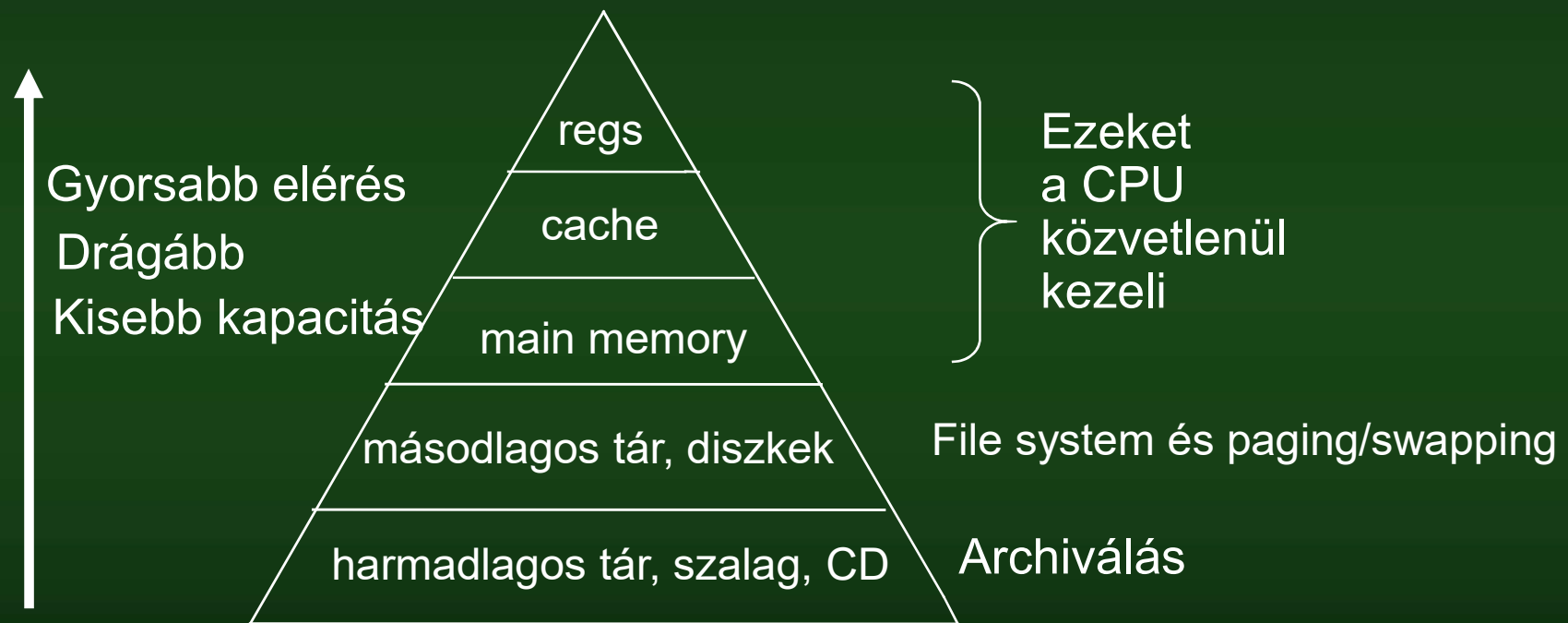
Egyszerű és Write Bufferes Cache



Az asszociatív tár

- Tartalom szerint címezhető tár
- Translation Lookaside Buffer
- A CPU-hoz közeli tár, a memóriamenedzselést segíti
- Majd az OS tárgyban vesszük.

A memória hierarchia



Számítógép architektúrák

A memória
VÉGE