Computer architectures

The memory

Content

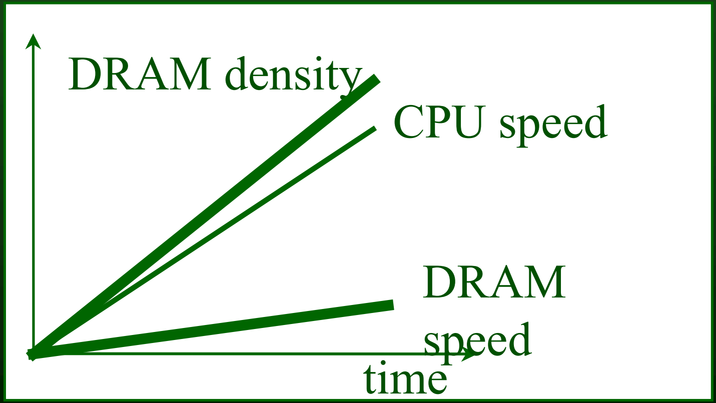
- Semiconductor storages
- DRAM, SRAM
- ROM, PROM
- Enclosures, memory modules
- The principle of locality
- Caches

The memory

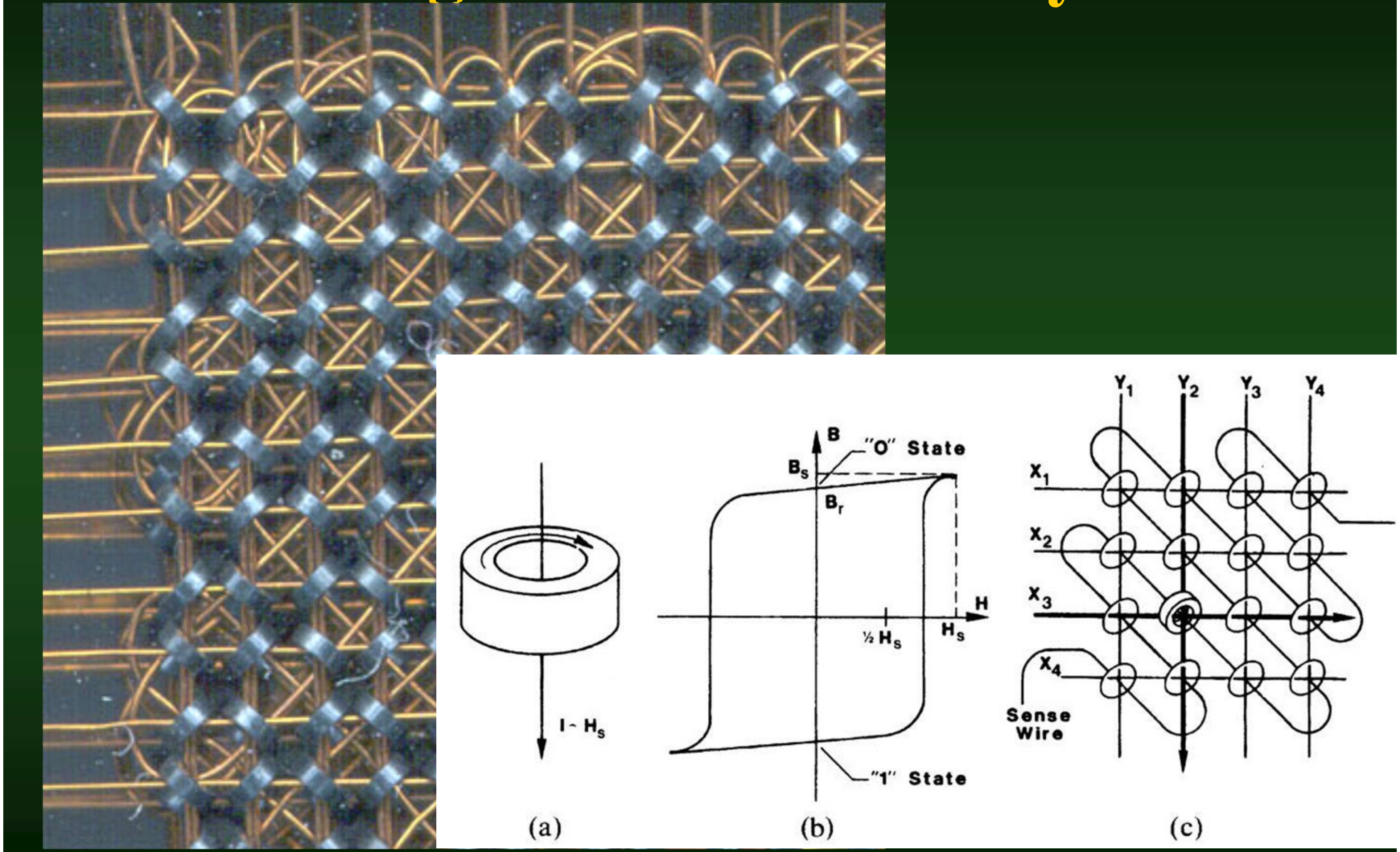
- Storage: for storing programs and data. Addressable cells.
- Central storage (Operational memory): on (fast) memory bus, or it is connected to the processor via the system bus
- Memory on the peripheral controllers too! These can also be addressed! Sometimes their address ranges match, sometimes they not overlapped.
- Let's check the memory hierarchy later!

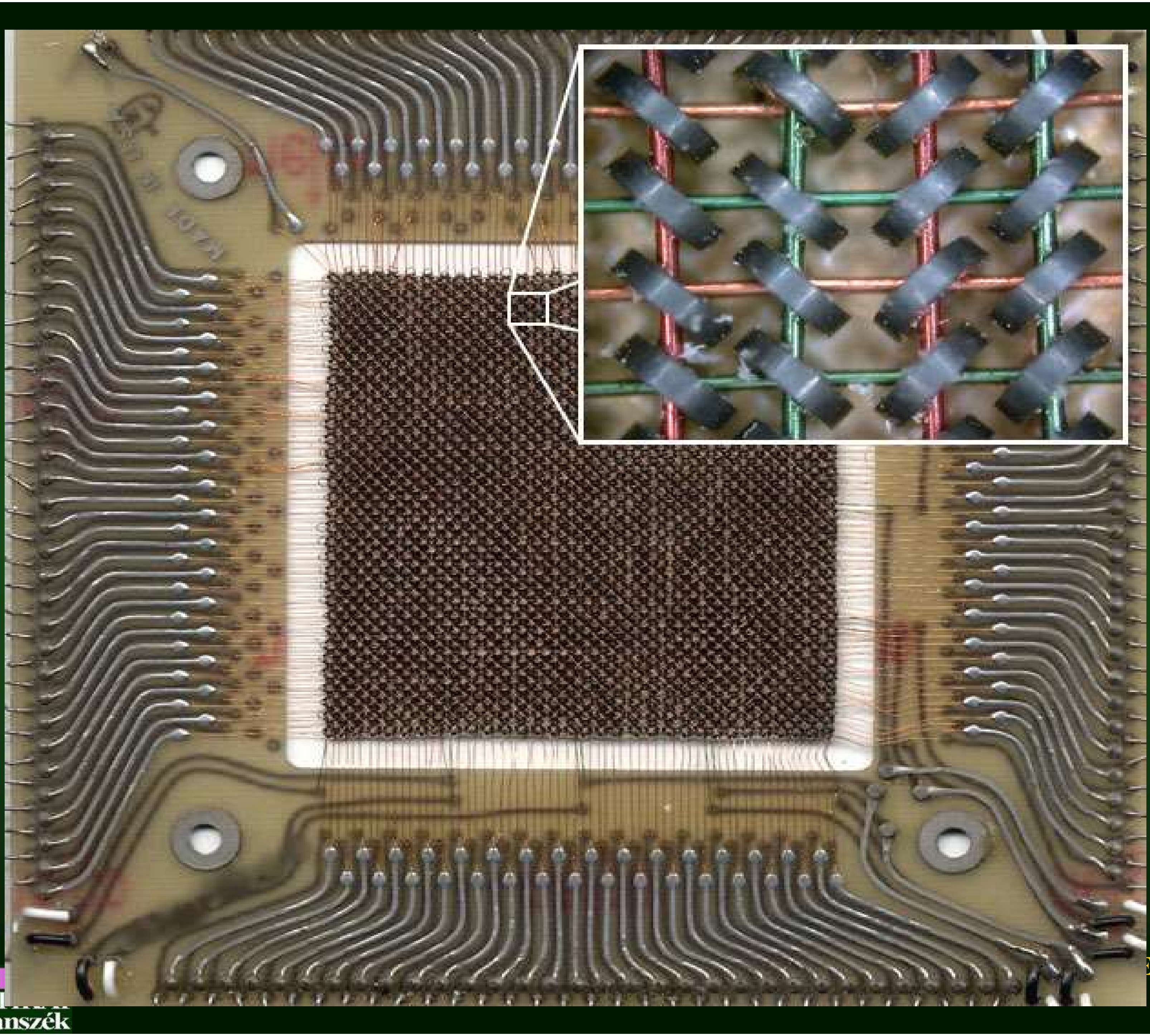
Implementation of the storages

- In the past, ferrite ring reservoirs: magnetizability-flux switching principle. Non-volatile memory.
- Today, high-integration semiconductor chips (network of transistors[-capacitors] in a case), mounted on a memory module
  - cyclic operation!
- Two trends:
  - capacity increase,
  - access time reduction.



Magnetic-core memory





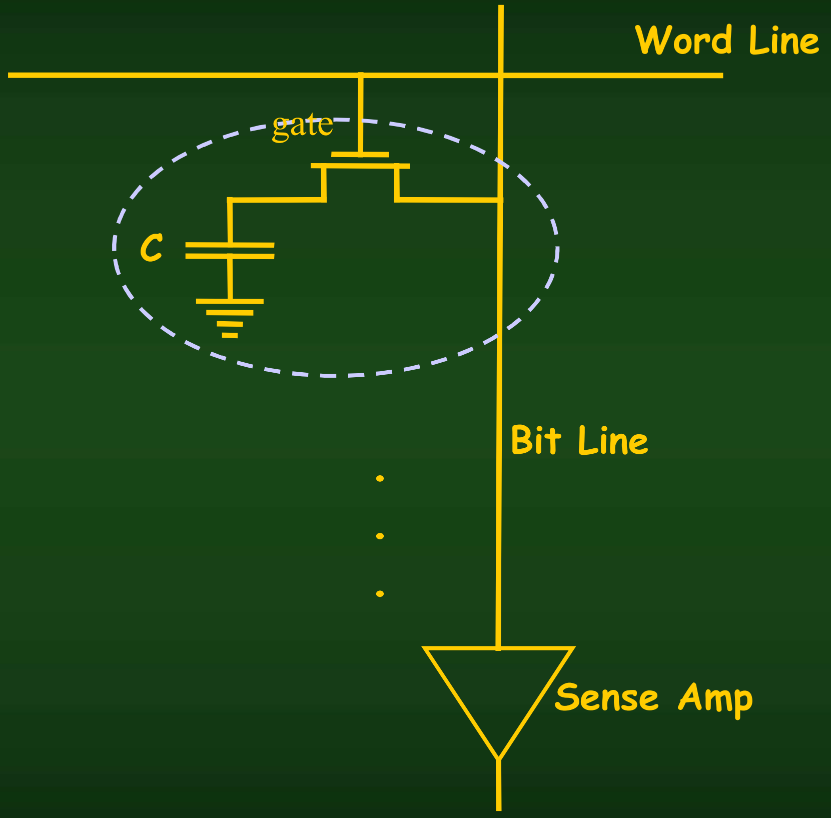
Please note

- The two main components of service time are:
  - (1) access time , the time elapsed from the request to the memory module until it is available (response time of the memory module). This is used to be on the order of *50-150 ns*;
  - (2) time of the memory bus (and in the chipset) . On average, this is in the order of *125 ns* .
- This gives an average service time of 195 ns .
- Let's compare it with an (average) L2 cache (external cache) service time, well it's 45 ns .
- Since the appearence of SDRAMs, speed has been specified in MHz. Another little "fake": time can be written on the chips in ns : well, that's the cycle time!!! That is 15 ns → 66 MHz; 10 ns → 100 MHz; 8 ns → 133 MHz syst. (So this bus cycle ...) 2.78 ns → 3600 MHz
- The real memory access speed is the bus cycle multiplied by the word width (MHz * bits) ... Ex.
  - 100 MHz chip, 64-bit bus = 6400 Mbit /s = 800 MByte /sec speed
  - RDRAM 800 MHz, 16 bits = 1.6 GB /sec

Semiconductor storage

- RAM: Random Access Memory
  - random: access to one cell does not depend on the others, we can address any of them „randomly”. (Not e.g. sequential access)
  - It is a grid consisting of rows and columns, its elements are cells
- DRAM: Dynamic RAM
  - A cell is a transistor-capacitor pair for one bit.
  - Dynamicity: read-write, „refresh” is also dynamic and takes time.
  - Writable-readable,
  - Volite, it "forgets" when the power supply cancelled.
  - MOS, CMOS, NMOS technologies.

A cell…

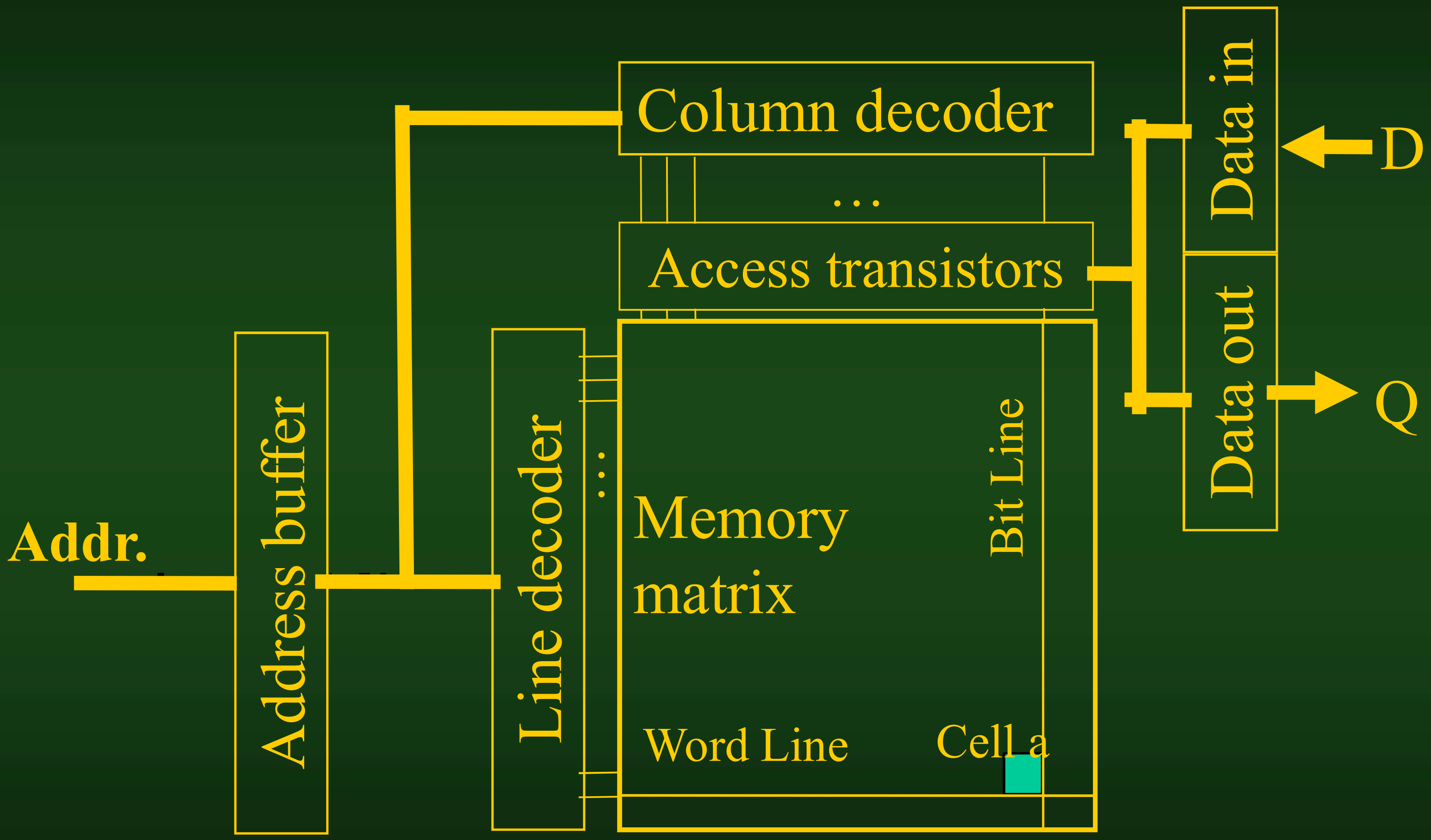


- **Writing:**
  - Set the bit line to high or low level (according to the bit to be written);
  - Open the gate: C is charged
- **Reading:**
  - Set the bit line to "half" voltage;
  - Open the gate;
  - Depending on the AC level, the bit line voltage shifts, which is sensed by the Sense Amp.

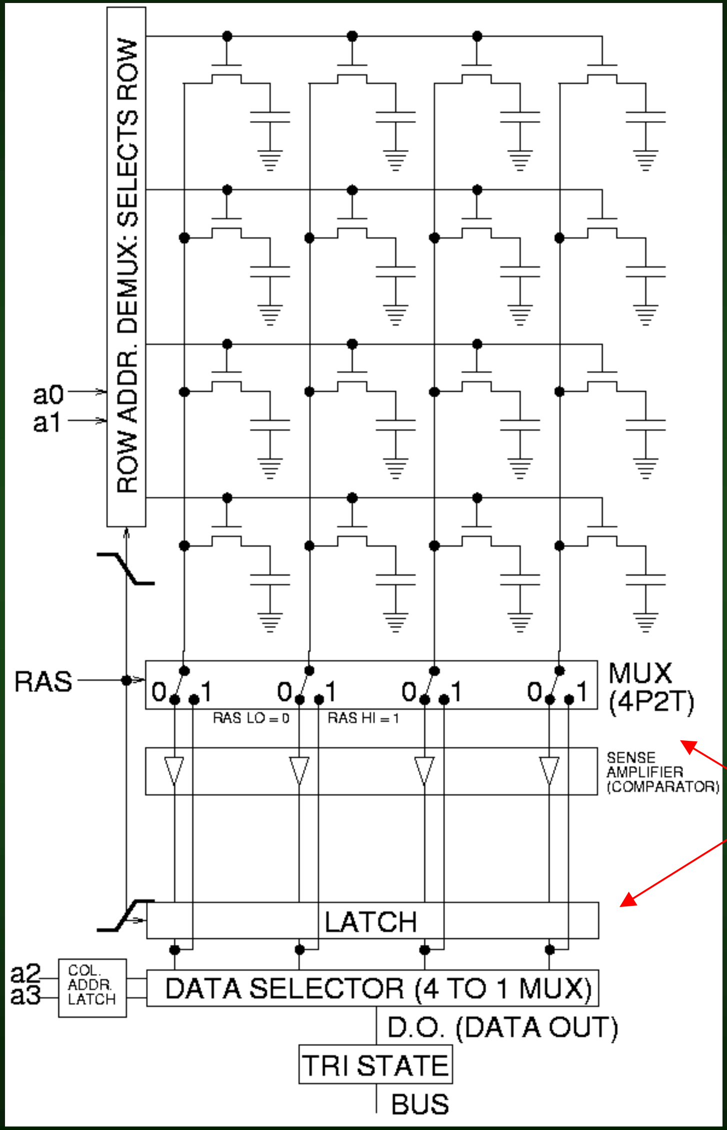
DRAM circuits...

- Grid of rows and columns, its elements are cells
- Additional special circuits in the chip help
  - to select rows/columns of cells (r/c address select/decoder , row/column address buffers)
  - to store signals "read" from the cells (sens amplifiers: access transistors, output buffers)
  - to track refresh sequences (counter)
  - to write the cells, to "raise" their charge (write enable)
- The memory controller (possibly CPU) can help "from the outside"
  - Memory type, speed, quantity identification, error handling

DRAM logic structure

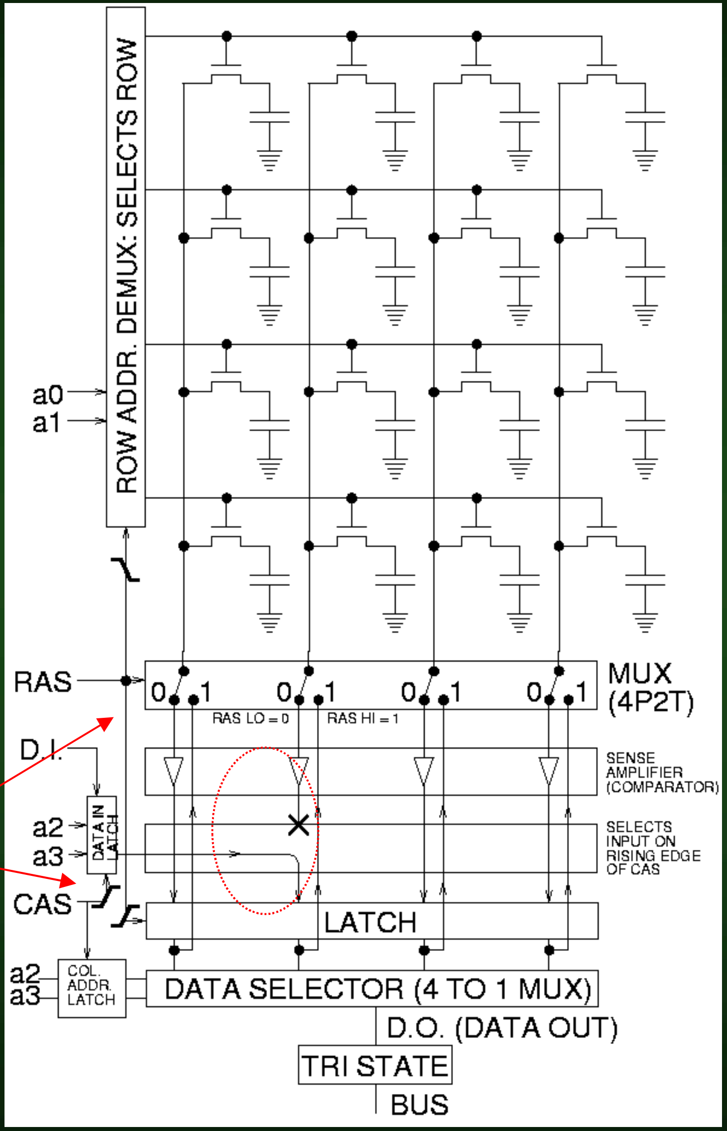


A 4 x 4 matrix (1 bit of data)



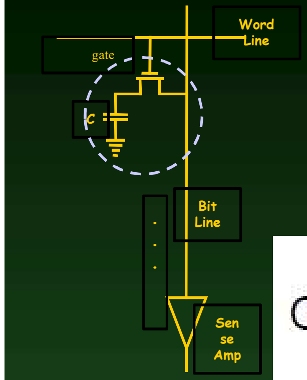
Reading

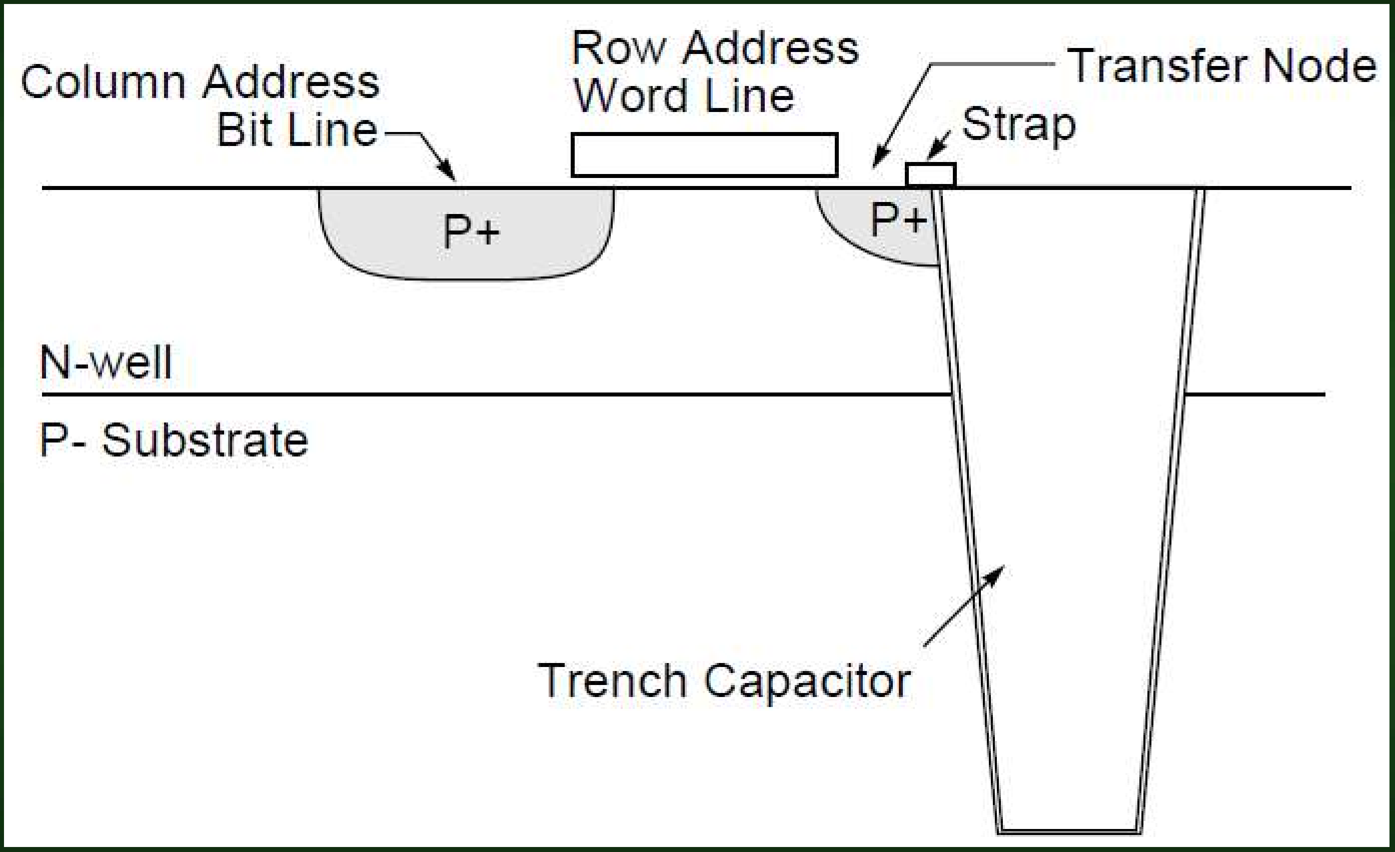
Refresh
(write back)



Writing

A cell

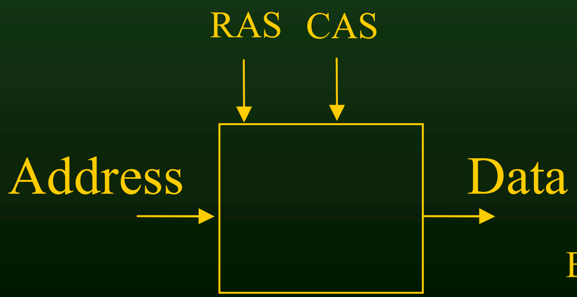

Word Line
gate
C
Bit Line
Sen
se
Amp


Column Address
Bit Line
Row Address
Word Line
Transfer Node
Strap
P+
P+
N-well
P- Substrate
Trench Capacitor

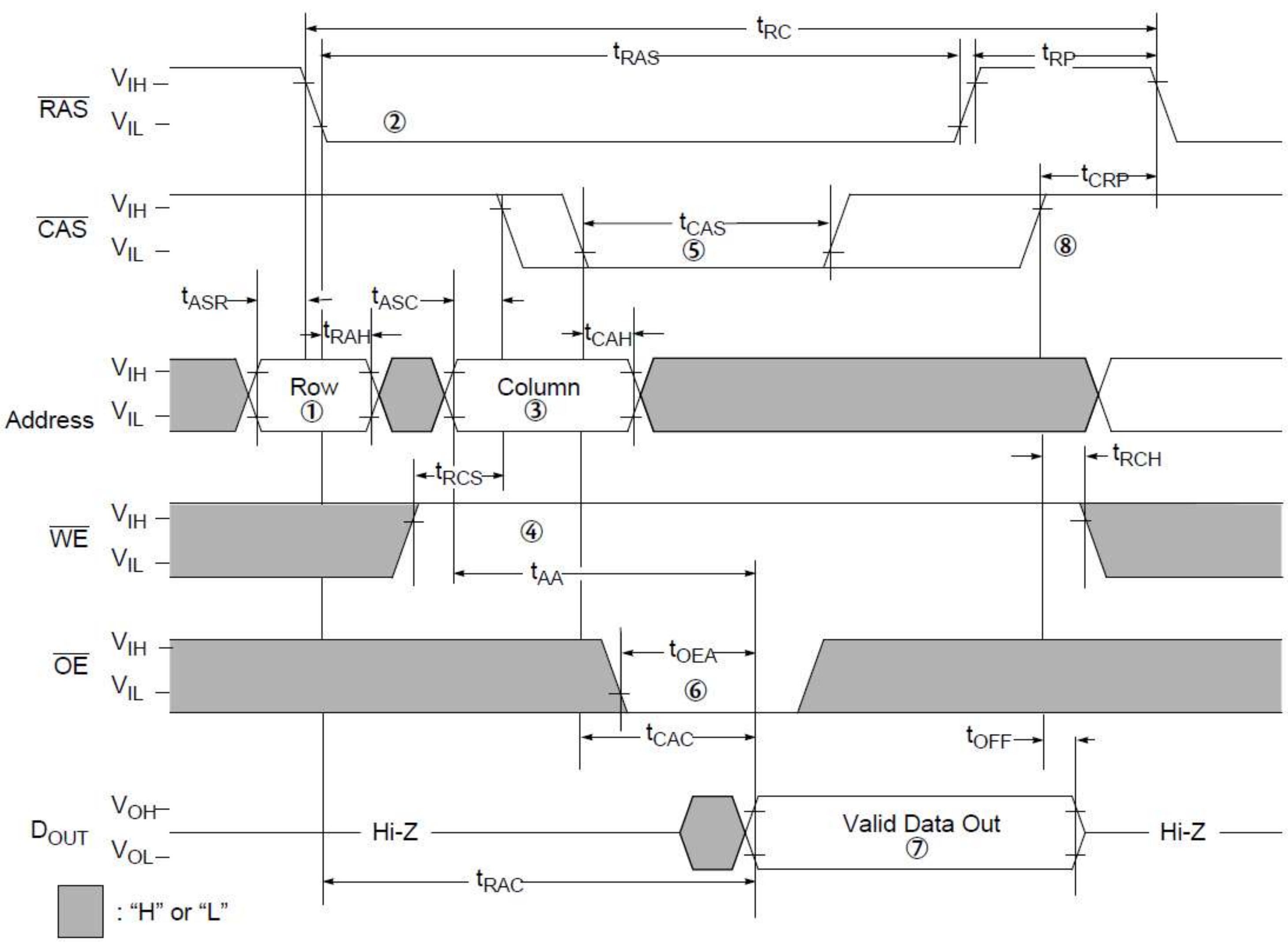
© IBM

DRAM operations

- **Typical memory access (read):**
  - **Line address on the address pins → RAS signal drops: line address is fixed in the line address buffer and the access transistors (sens amps) are activated;**
  - **with the stabilization of the RAS signal, the values of the cells of the entire row are readed by the access transistors;**
  - **Column address on the address legs → CAS signal drops: column address is recorded in the column address buffers; when the CAS stabilizes, the selected part is loaded into the output buffer.**

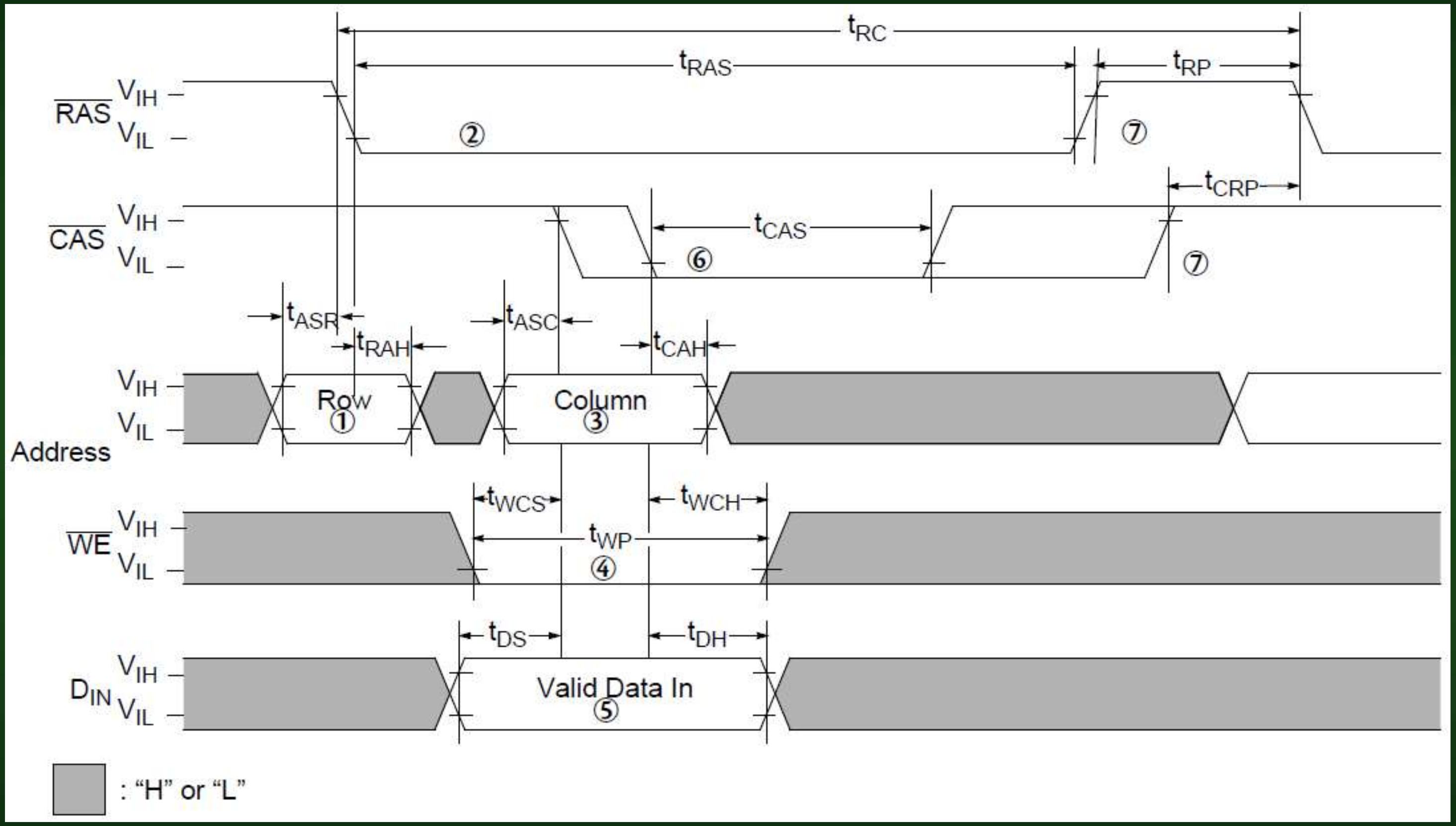


DRAM operations - reading



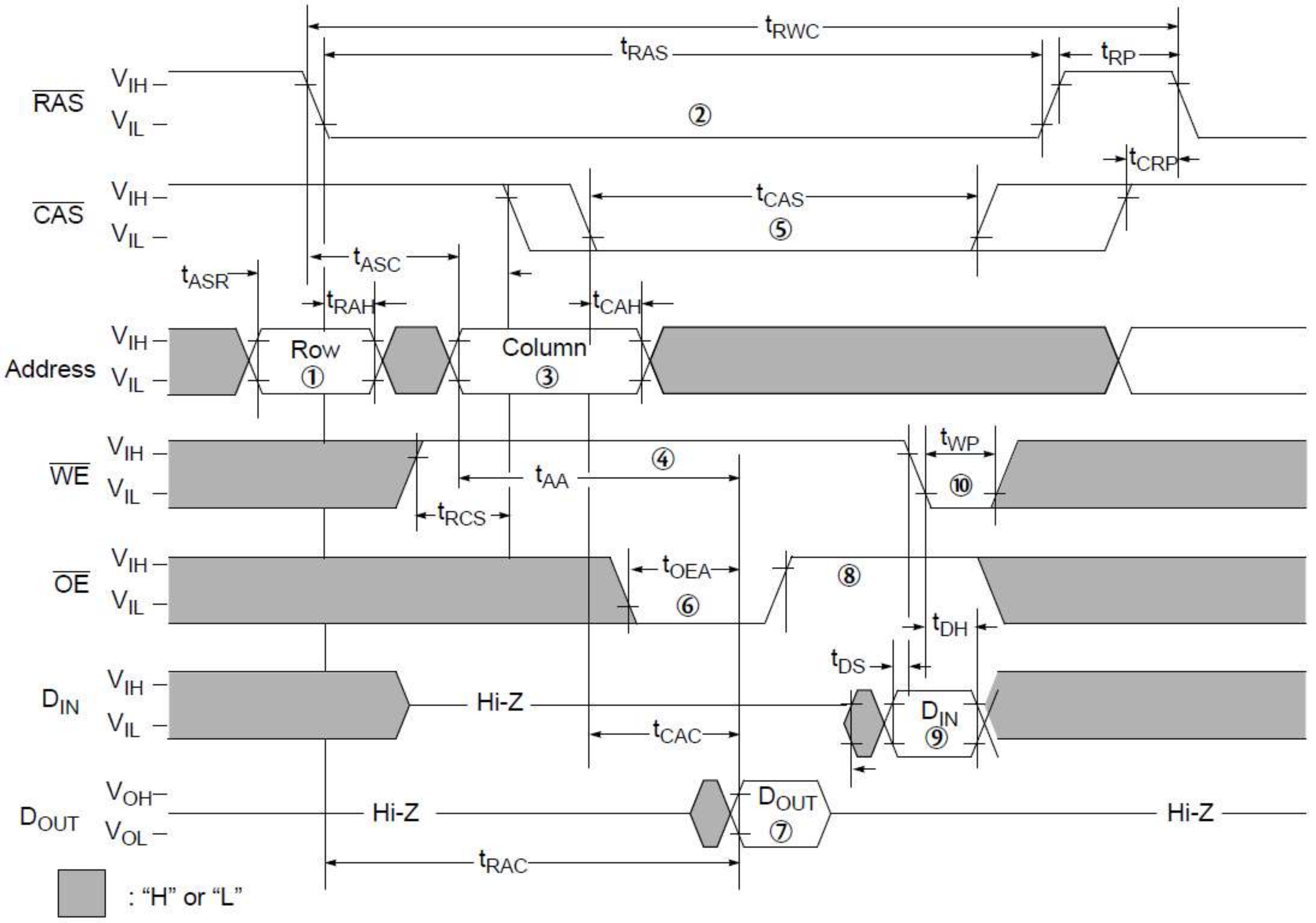
© IBM

DRAM operations - write



© IBM

DRAM - read - modify - write



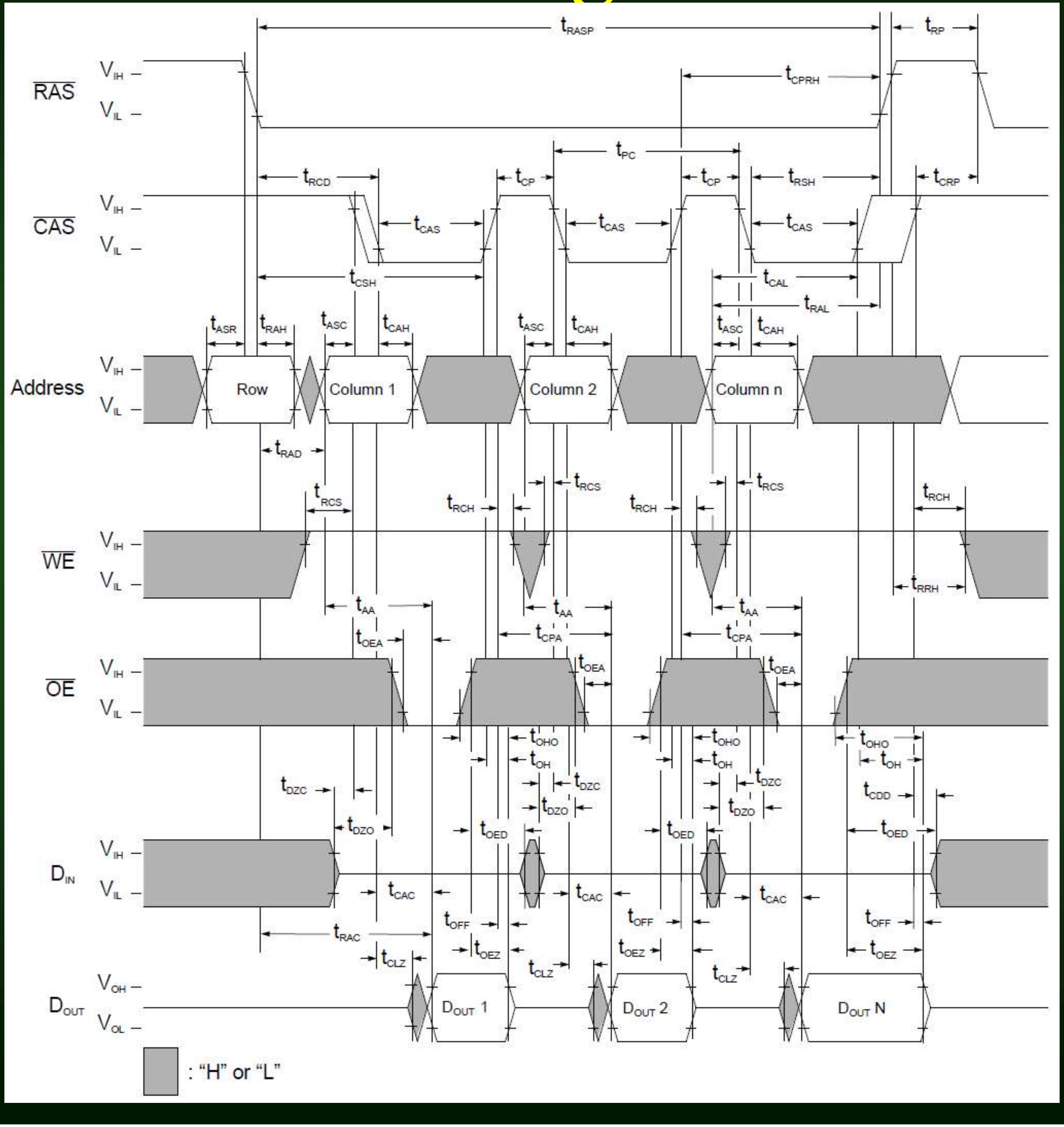
© IBM

Technologies

- Fast Page Mode (FPM) RAM (multiple column addressing in addition to one row addressing), Extended Data Out (EDO)
- Burst Extended Data Output (BEDO) RAM (one row address, one column address plus 4 data cells) (dual bank, but obsolete due to the "death" of EDO)
- SDRAM : Synchronous DRAM (we can go above 66-100MHz) The mem "locks" the address, data and control information coming from the CPU, it works autonomously (the system clock), while the CPU can do other things. After some time (latency), the output presents the result (pipeline) ...
- ESDRAM (Enhanced ...): the standard SDRAM chip also has a smaller SRAM cache (up to 200 MHZ)
- DDR (Double Data Rate): 2 - burst-deep prefetch buffer
  DDR2 : 4, DDR3 : 8, DDR4 : 8

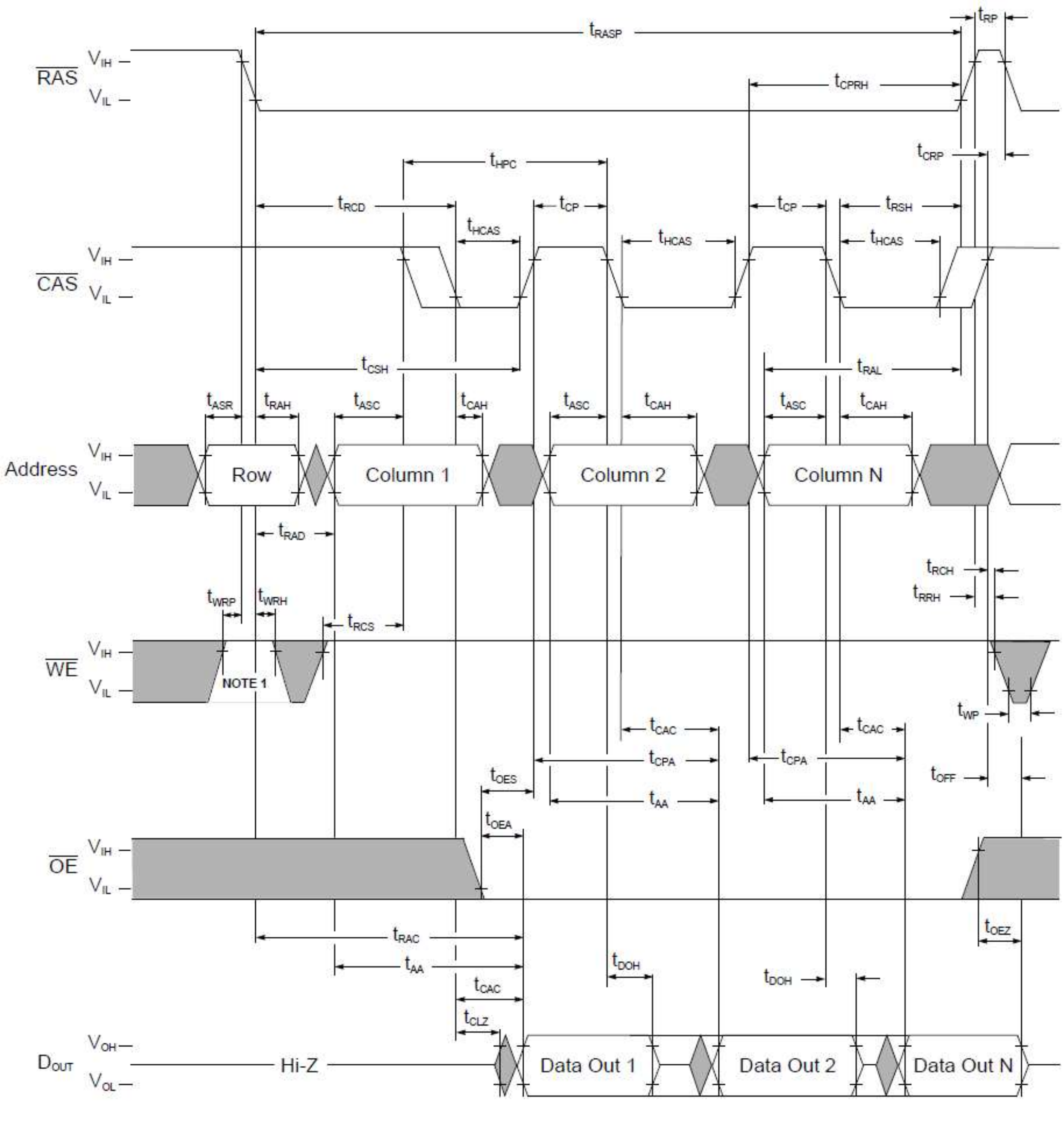
1Gb DDR3: 1 row 2,048 bits (256 Byte)

DRAM - Fast Page Mode Read



© IBM

DRAM - EDO (Hyper Page) Mode Read

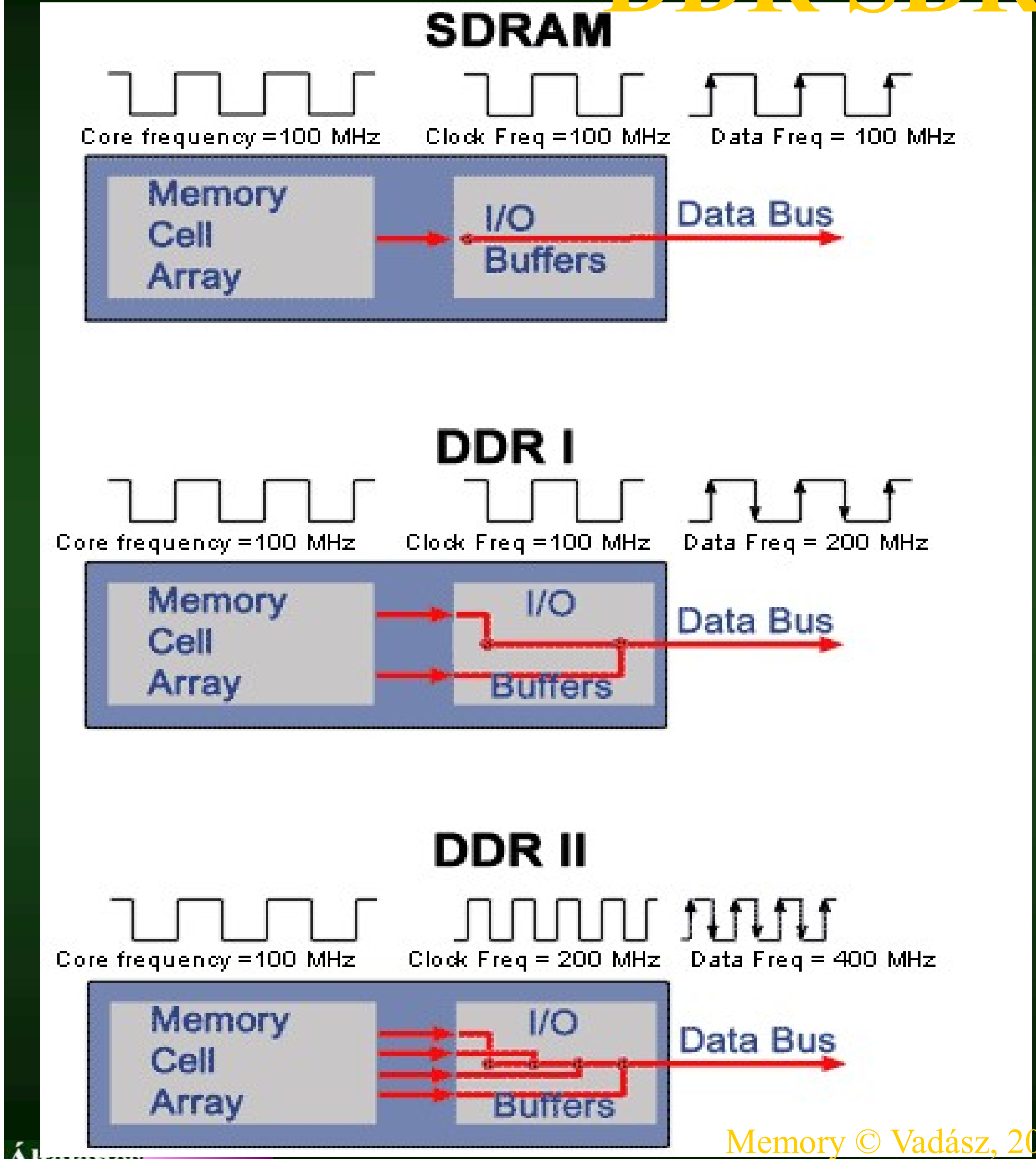


▨ : "H" or "L"

NOTE 1: Implementing $\overline{WE}$ at $\overline{RAS}$ time during a Read or Write Cycle is optional. Doing so will facilitate compatibility with future EDO DRAMs.

© IBM

DDR SDRAM



DDR3 (2007): 1.5V
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): 1.1V
2400 - 6200 MHz
DDR5-4800 - DDR5-7200

Road map...

Year of presentation	Technology	"Speed" limit	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

Market rates are…

	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%

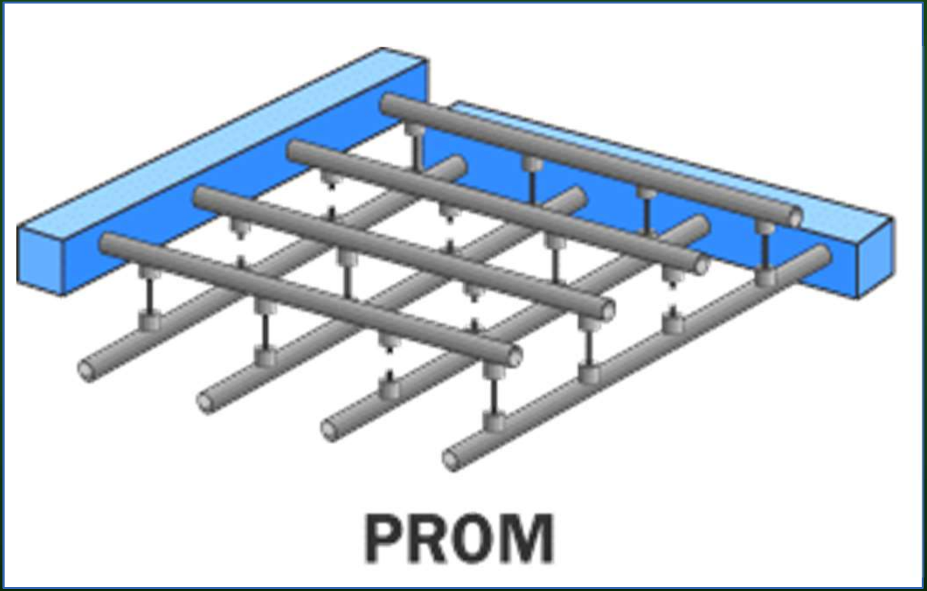
http://www20.tomshardware.com/motherboard/index.html

Semiconductor storage ...

- ROM : Read Only Memory: can only be read.
  - This is also a grid of cells, arranged in a row-by-column array,
  - diodes in the cells, give connection (1 bit), no connection (0 bit).
  - Non-Voliate (does not forget when turned off).
  - Also random access. (column-row selection).
  - Their address range and addressability can coincide with DRAMs: the (central) memory can consist partly of ROMs and partly of DRAM .
- PROM, EPROM : The content can be burned in, EPROM can be erased and rewritten...

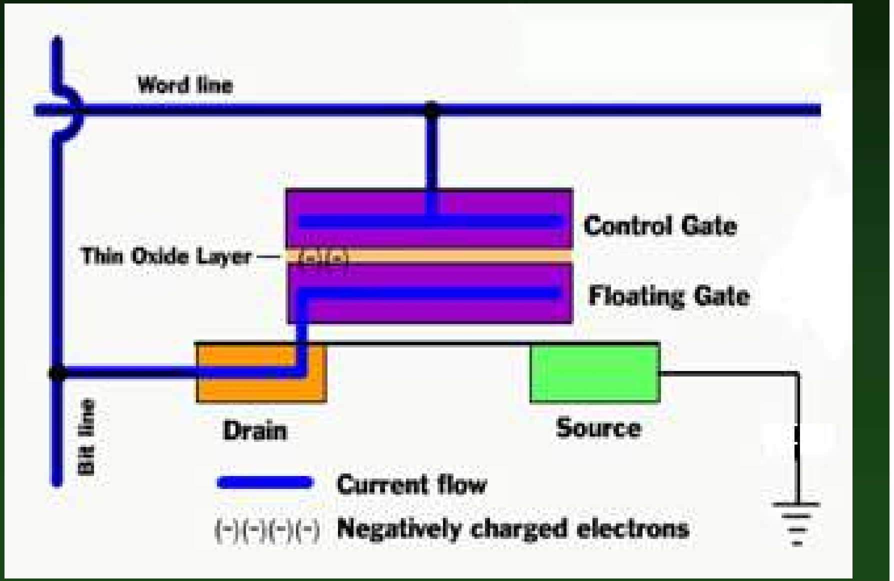
Semiconductor storage ...

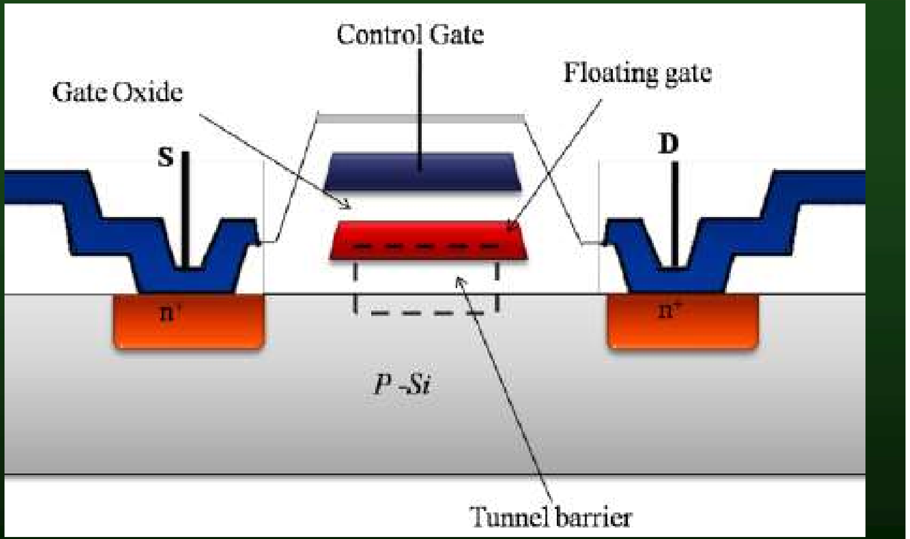
- **PROM** (Programmable ROM)
  - row-column mesh,
  - "fuse" in the cells,
  - Content can be burned in.



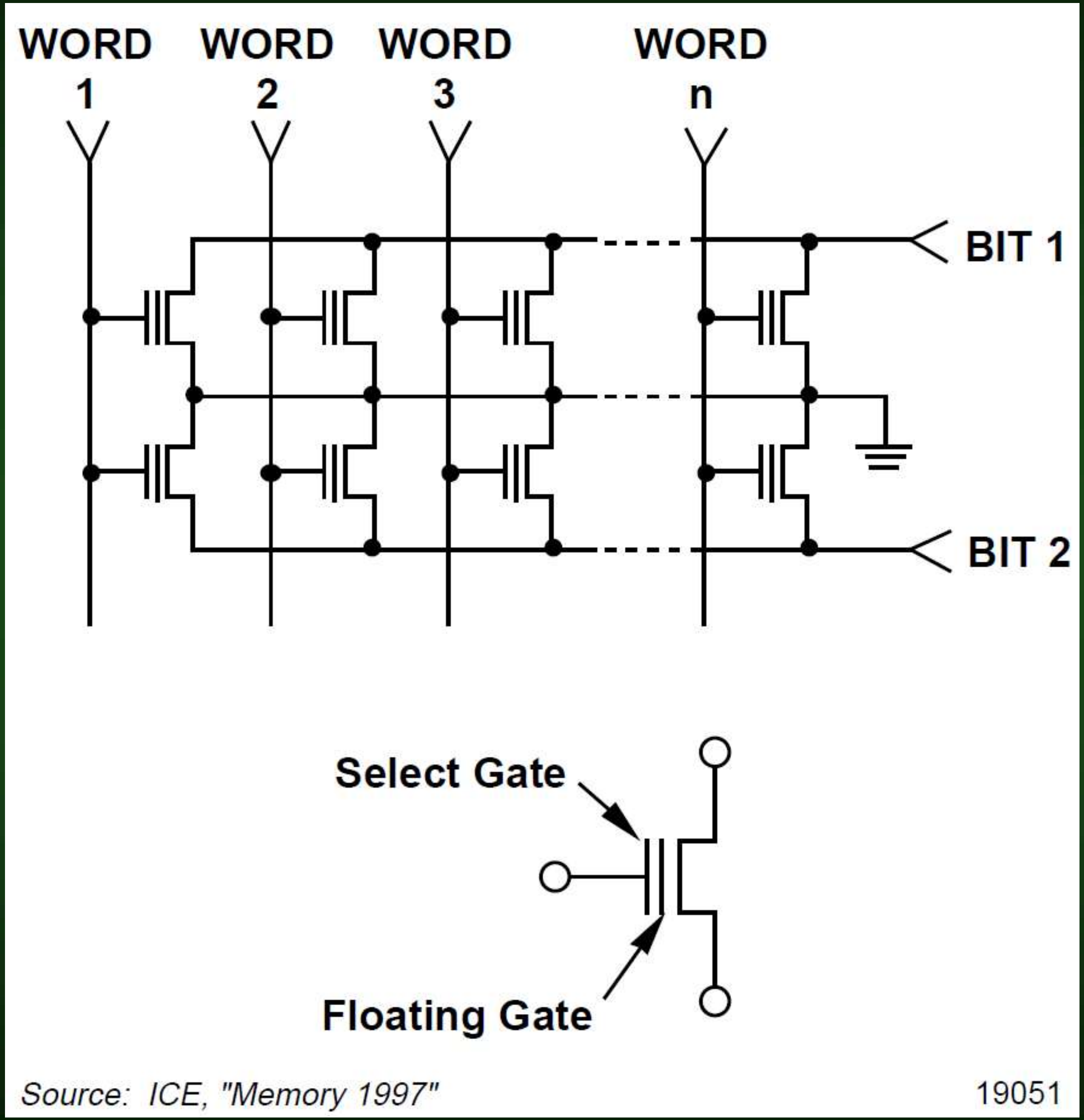
Semiconductor storage ...

- EPROM (Erasable PROM) can be erased and rewritten ...
  - Cells have two gate FETs (Field-Effect transistor)
    - Floating gate (closes when charged to negative)
    - Control gate
    - An oxide layer between them
  - 1 bit: the bit line is "connected" to the ground ...
  - 0 bit: gate "closed"

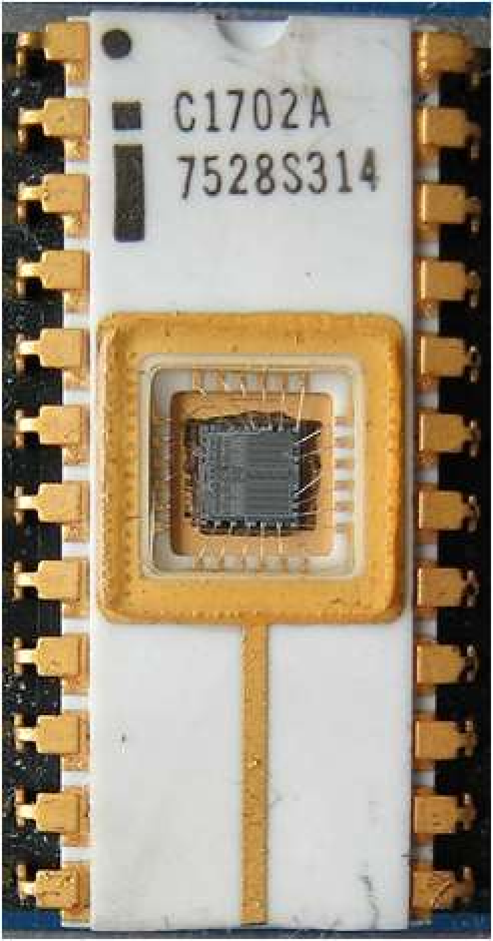




Semiconductor - EPROM

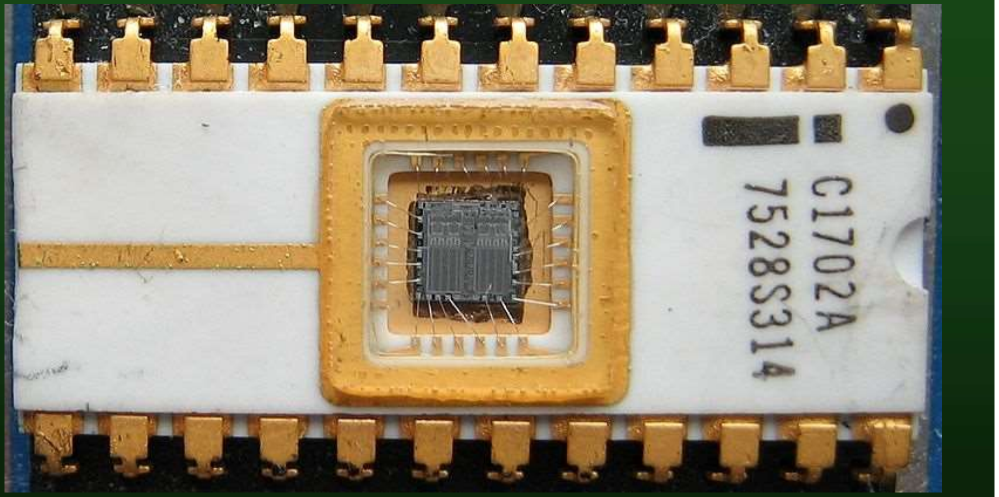


Source: ICE, "Memory 1997"



Semiconductor storage ...

- **EEPROM (Electronic Erasable PROM)**
  - Like EPROM, but can be electronically erased (not by UV light).
  - It is slow, because 1 byte can be deleted or rewritten at the same time...
- **FLASH memory**
  - This is actually EEPROM, but a block (512-x kilobytes) can be rewritten at a time.
  - It's already fast enough.



Semiconductor Storage - Solid State Drive

FLASH memory

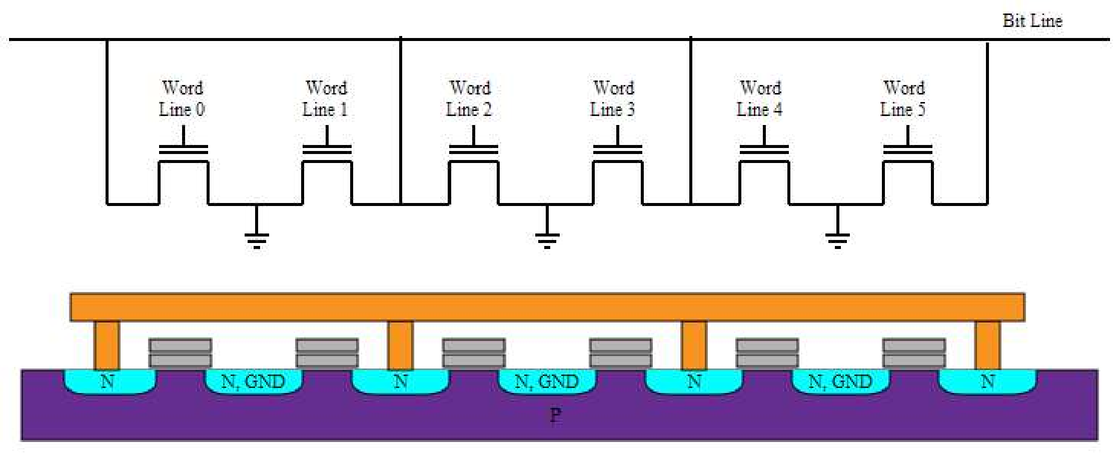
Memory wear : 100,000 program-erase (P/E) cycles

Erasure sets (all) bits, programming can only clear bits: 1111 – 1110 – 1010 - 0010, finally 0000

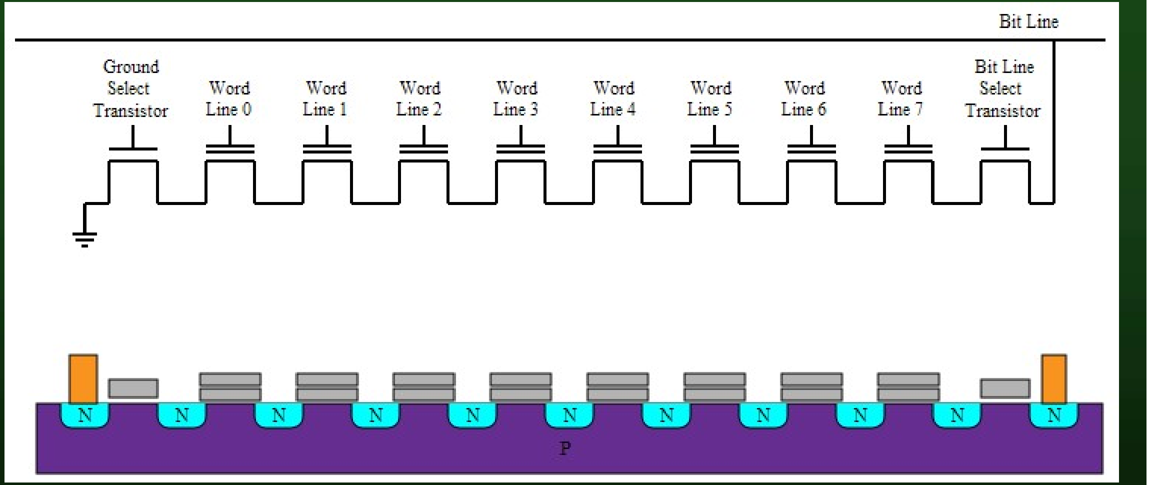
Cell read-out : a voltage intermediate between the threshold voltages is applied to the CG, and the MOSFET channel will conduct or remain insulating, depending on the V_T of the cell, which is in turn controlled by charge on the FG.

NOR flash

Not a row, but only 1 bit at a time

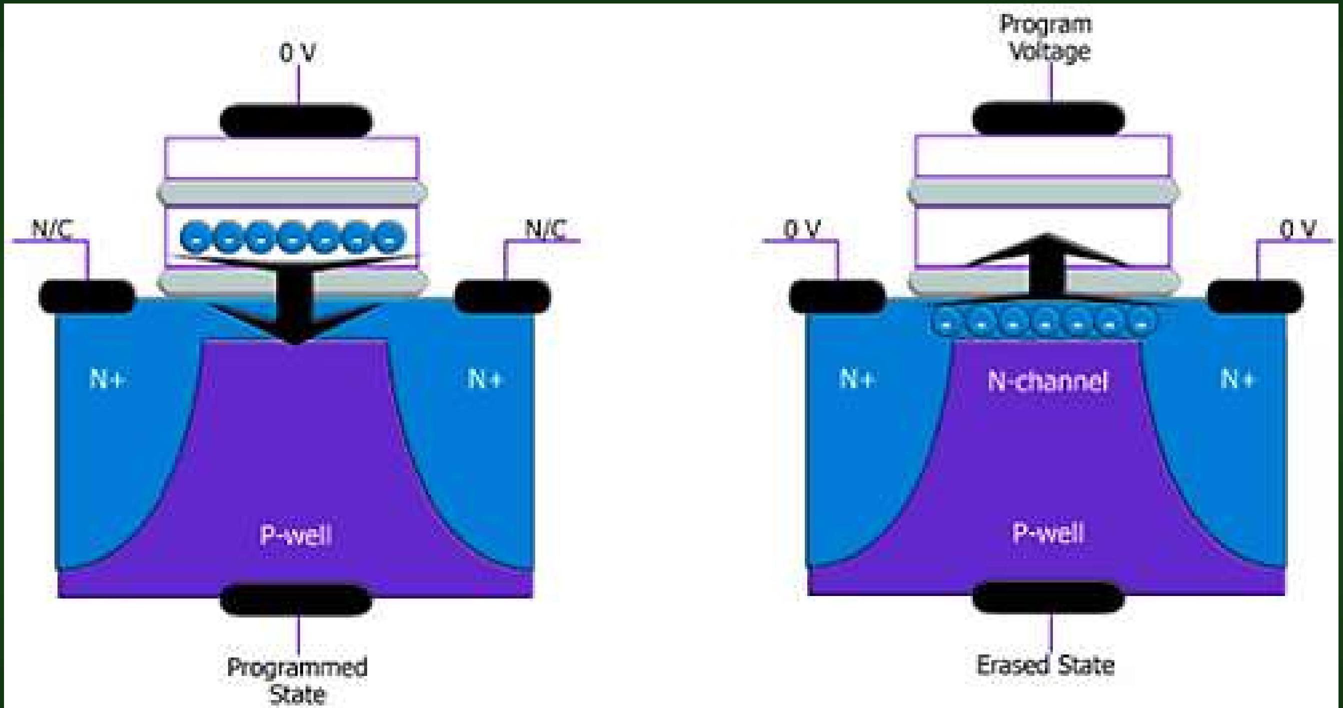


N AND flash



Semiconductor Storage - Solid State Drive

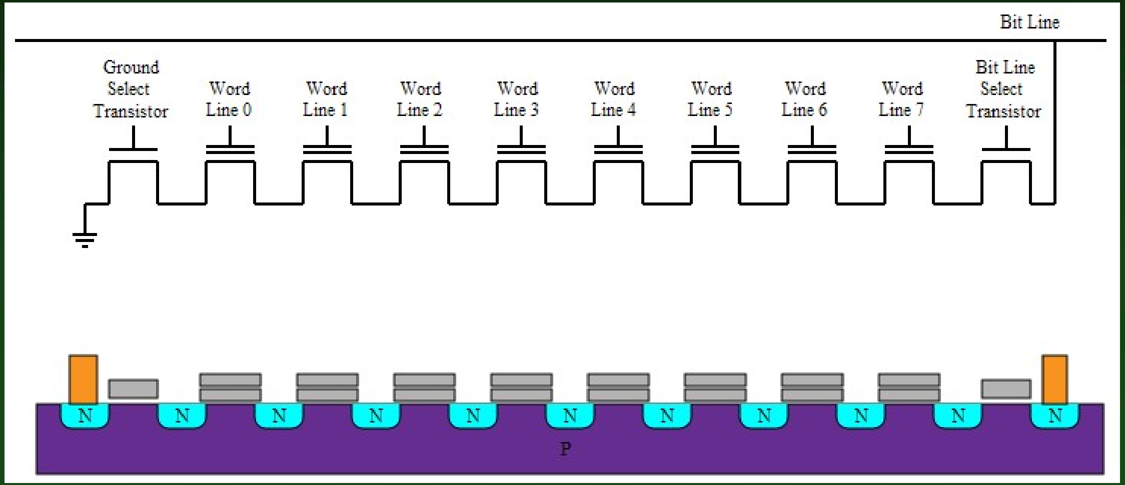
NAND Flash - programming

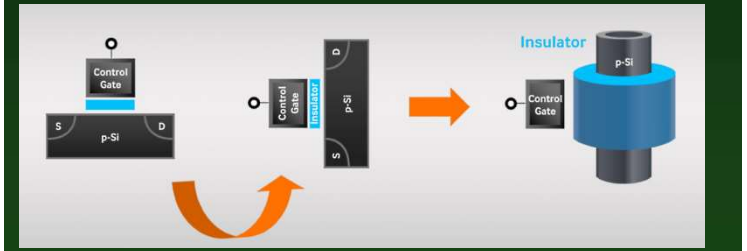


(tunneling)

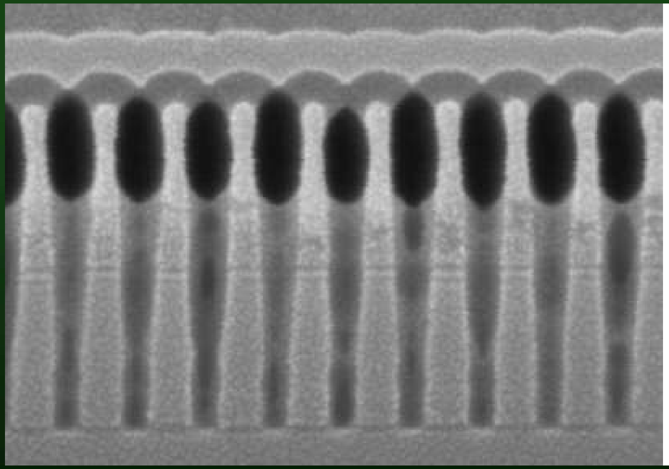
Solid State Drive - capacity increase

NAND Flash, layout

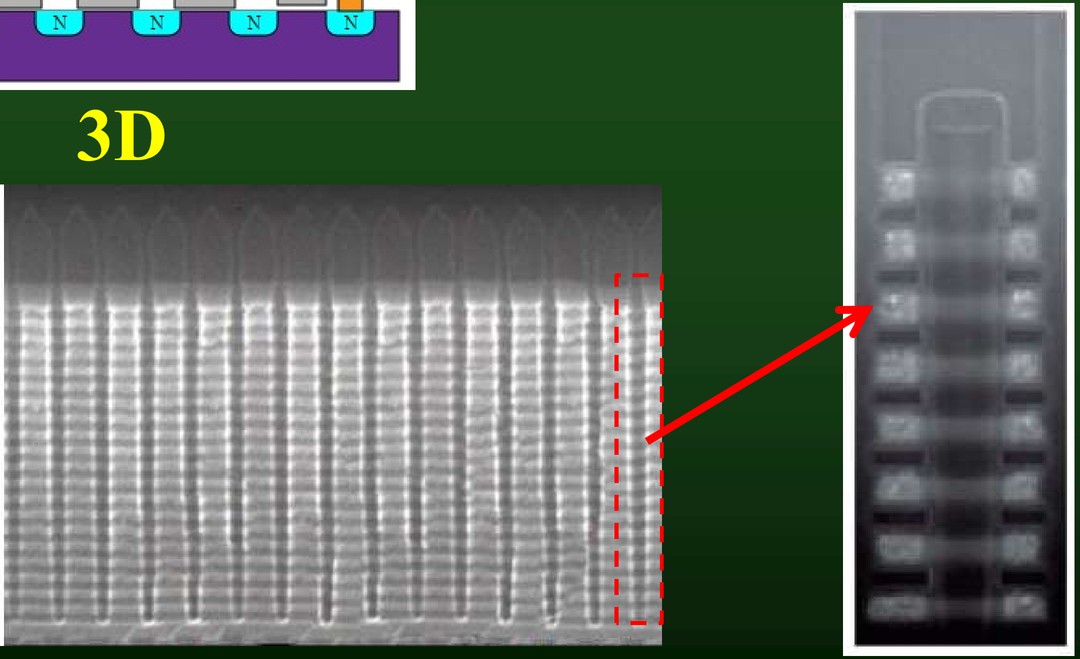




2D

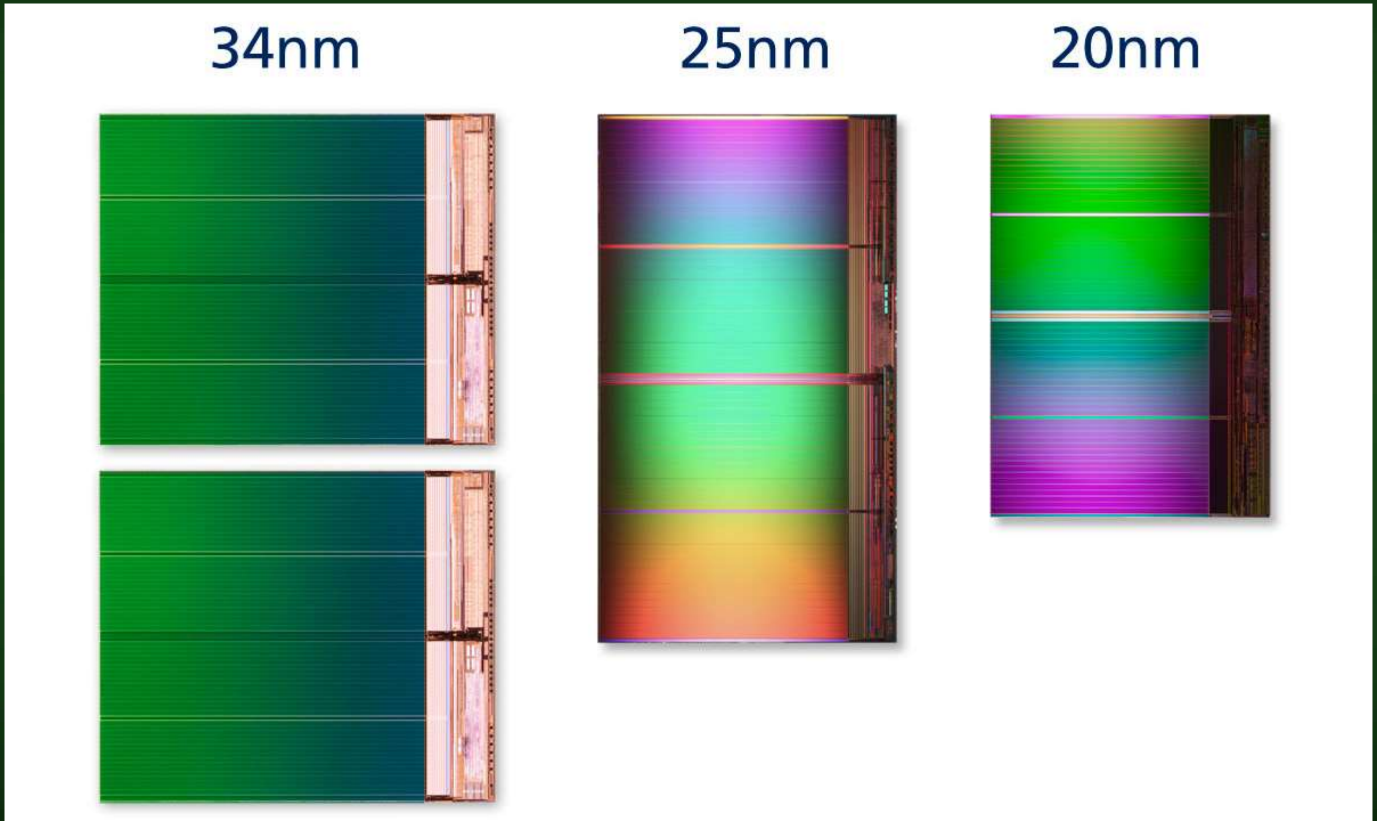


3D



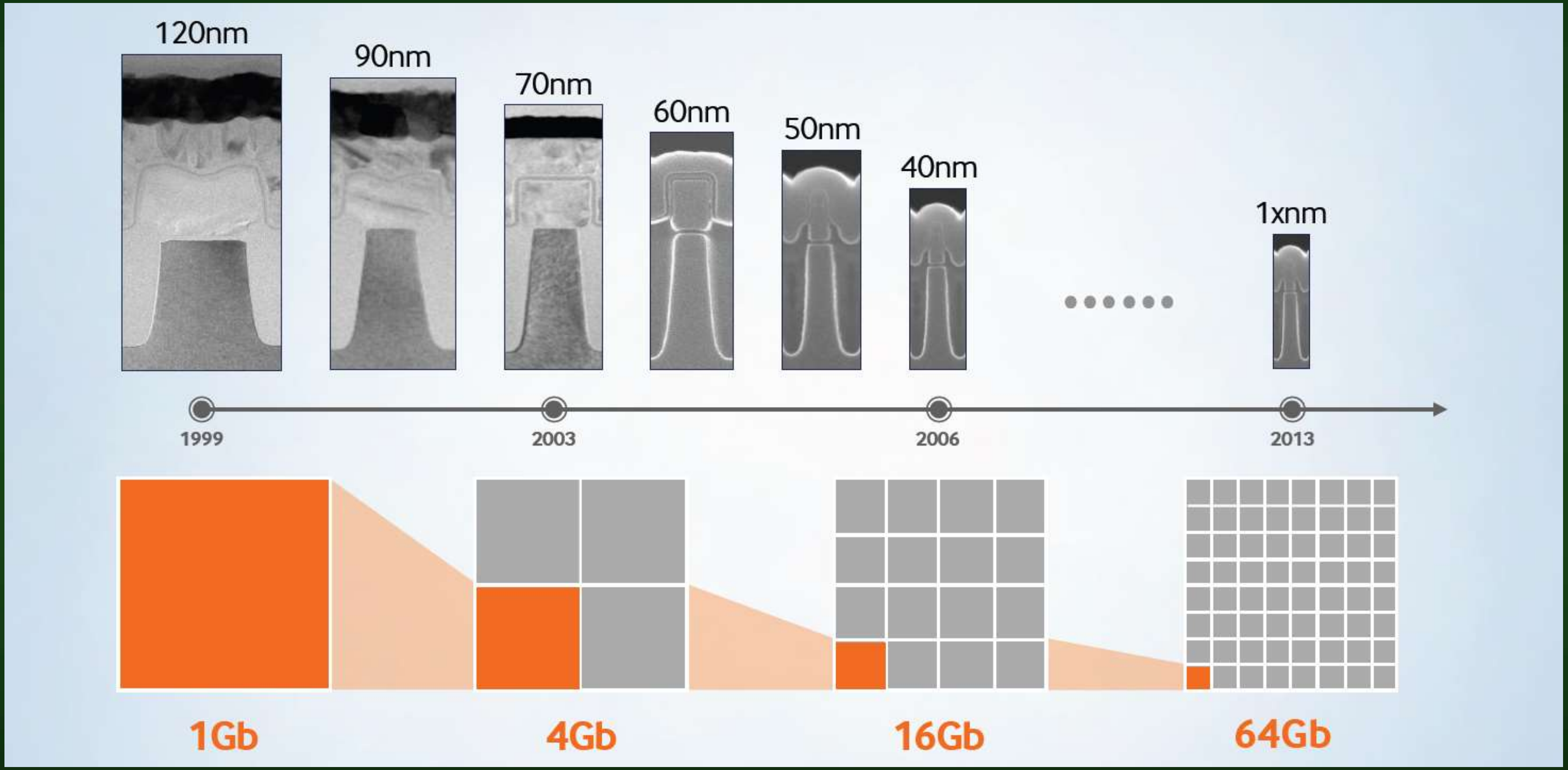
Solid State Drive - capacity increase

NAND Flash - linewidth (for the same capacity)



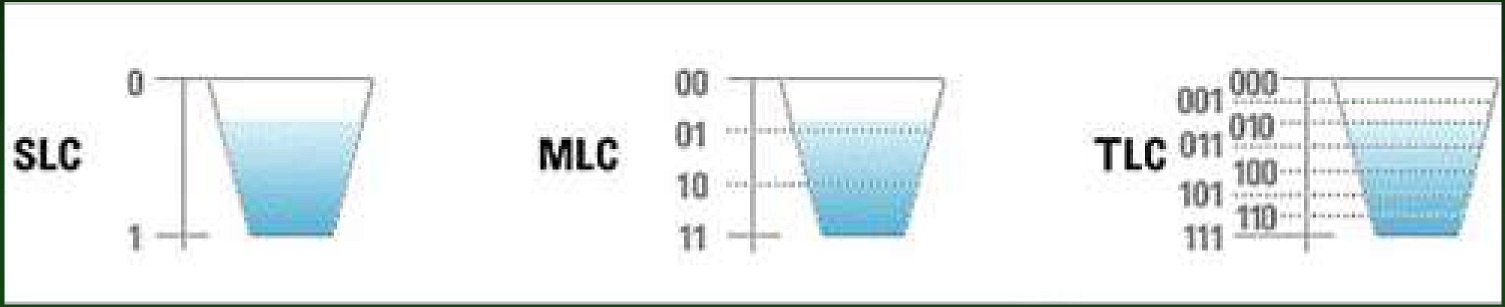
Solid State Drive - capacity increase

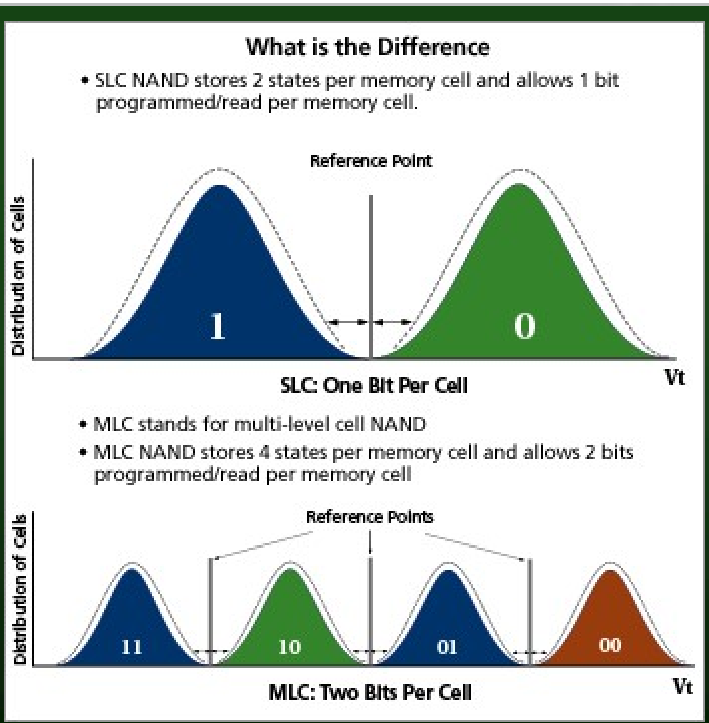
Line width - endurance is a problem



Solid State Drive - capacity increase

Increasing the number of states (SLC:1, MLC:2, TLC:3 bits)




What is the Difference

- SLC NAND stores 2 states per memory cell and allows 1 bit programmed/read per memory cell.

SLC: One Bit Per Cell

- MLC stands for multi-level cell NAND
- MLC NAND stores 4 states per memory cell and allows 2 bits programmed/read per memory cell

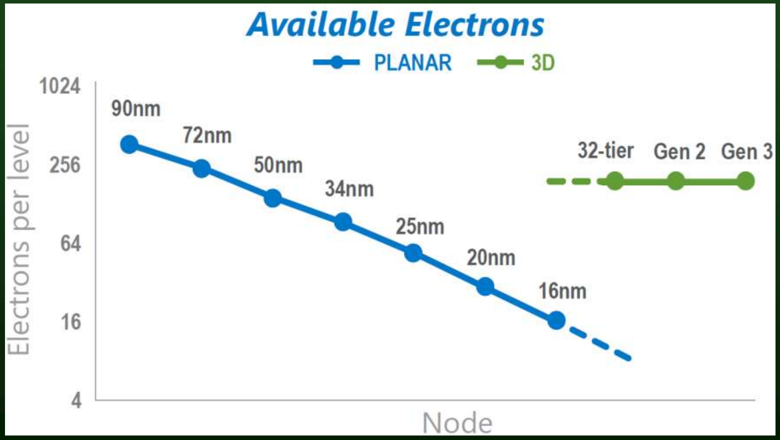
MLC: Two Bits Per Cell

Solid State Drive - capacity increase

Endurance is a concern

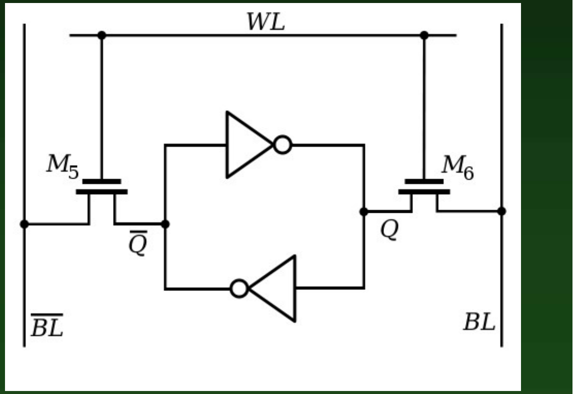
NAND type (20nm)	SLC	MLC	TLC (3-bit MLC)
Number of bits per cell	1	2	3
Max. programming cycle	100,000	3000	1000
Reading time	25 µs	50 µs	75 µs
Writing time	200-300 µs	600-1200 µs	900-1600 µs
Delete time	1.5-2 ms	2-3 ms	4.5-5 ms

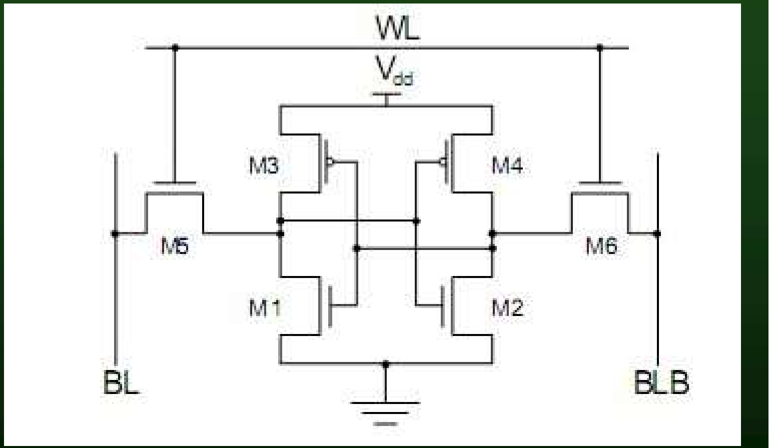
Solution: spares and wear leveling (wear-leveling algorithms)



Semiconductor storage ...

- **SRAM : Static Random Access Memory**
  - these can also **be written, read** ,
  - **are of random access** ,
  - their reading time is incredibly **fast** ,
  - but **they are expensive** and
  - **energy-intensive** (therefore they heat up and need to be cooled!)
  - used for caches.
  - **they have a cell of 4-6 transistors stationary flip-flop circuit .** They do not have a condenser . (They are like registers of CPUs)



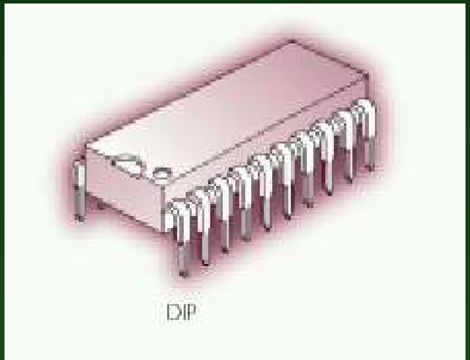


Memory (summary)

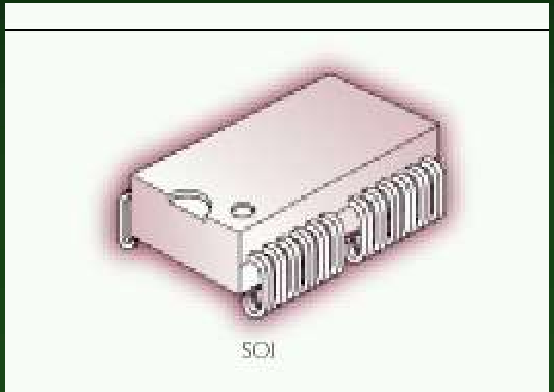
- RAM
  - DRAM (Packaged Highly Integrated Circuit Chips)
    - FPM, EDO, BEDO DRAM
    - SDRAM, ESDRAM
    - DDR, DDR2 , DDR3, DDR4
    - RDRAM
  - SRAM
  - ROM
    - PROM
    - EPROM, EEPROM
- FLASH

Packaging of DRAM chips

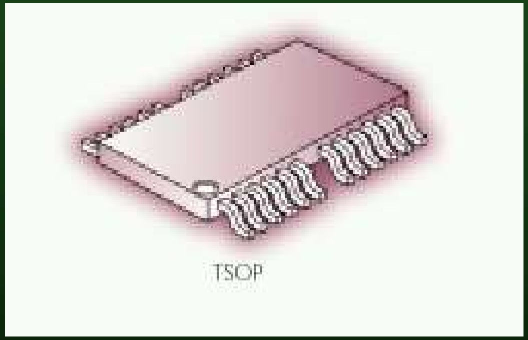
DIP: Dual In-Line Package



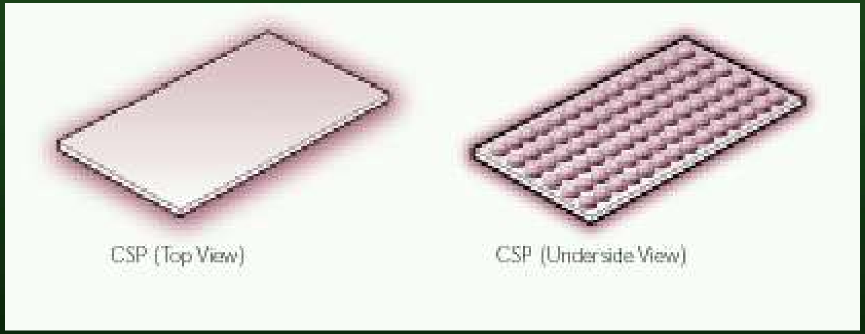
SOJ: Small Outline J-lead



TSOP: Thin Small Outline Package

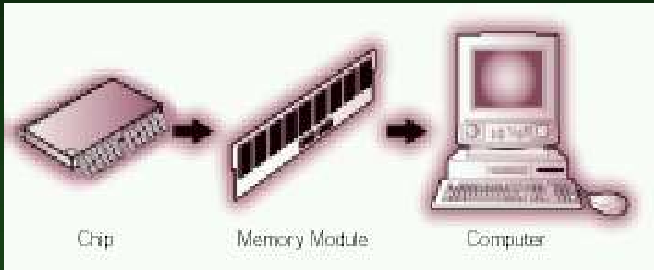


CSP: Chip Scale Package



Ball grid array (BGA)

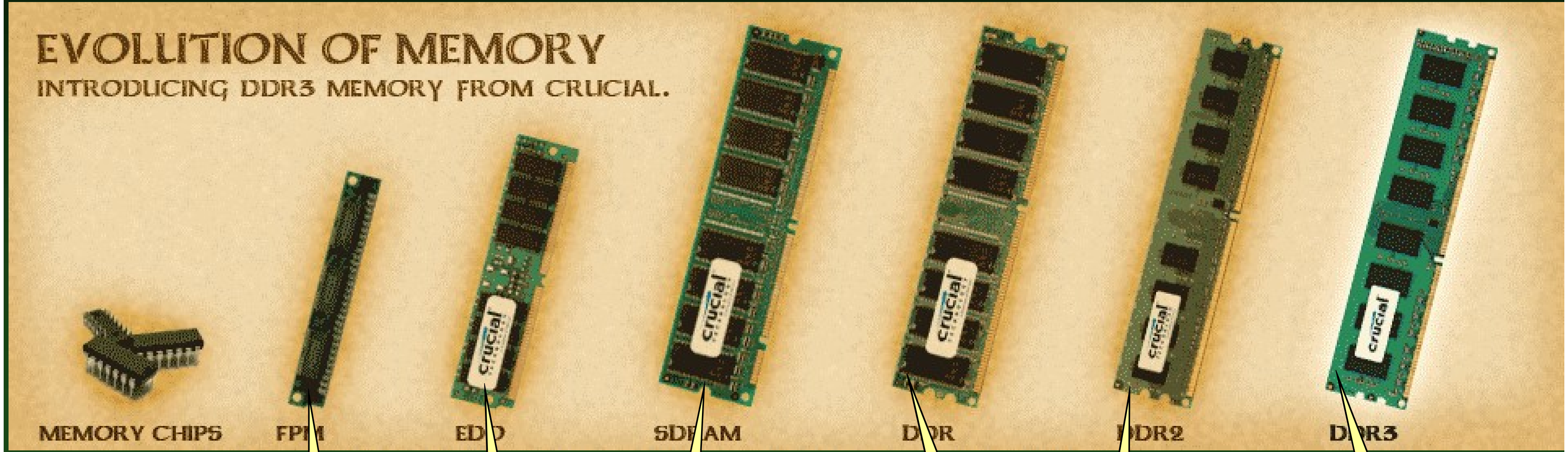
Memory modules



- Memory modules (MM) carry the chips. Standards. They can be placed in the socket of the motherboard(s).
- Single In-line MM (SIMM)
  - For 32-bit CPUs
  - 72 contacts on the module
- Dual In-line MM (DIMM)
  - Also for 128, 184, 240 feet, 64-bit CPUs
- Small Outline DIMM (SODIMM)
  - 144 (72) pins, much smaller, for notebooks
- Rambus (RIMM, SORIMM)
  - 16-bit data path, pipelining, fast

DDR400 (for 800 MHz frontside bus chipset)

History...



1 bit

8 bits

30 pin SIMM (~1990)

72-pin SIMM (mid1990)

64 bits

128-pin DIMM (~1997)

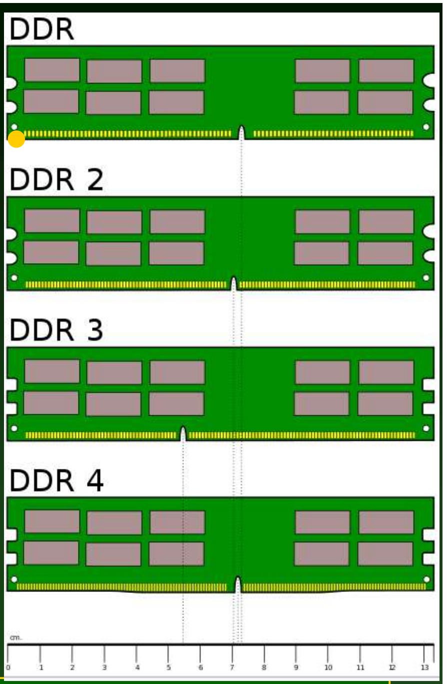
184-pin DDR DIMM (~2000)

184 pin DDR DIMM (~2002)

240 pin DDR2-800 DIMM

Általános INFORMATIKAI Tanszék

We can increase the frequency…



- **The memory timings (which can be adjusted) the bigger ones are the worse**

  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 : waiting time after the CAS drops, usually 2, 2.5, 3 cycles. Once this is done, the data is transferred to the DQ pins.

t_{RCD} RAS to CAS Delay : Wait after RAS failure until the CAS signal can be sent. Usually 2, 3 cycles

t_{RP} RAS Precharge : During this time, the controller deactivates the line again.

t_{RAS} Active lake precharge delay : the line must be active during this time, it can only be deactivated after that . There are usually 5 to 8 cycles. It must be fulfilled : $t_{RCD} + t_{CL} < t_{RAS}$

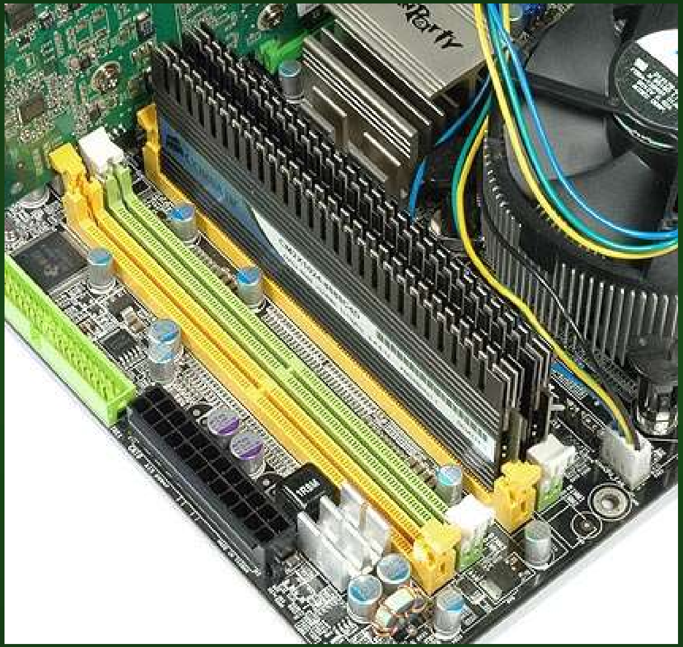
http://www.tomshardware.com/2007/10/03/pc_memory/

CAS latency (CL) Clock cycles between sending a column address to the memory and the beginning of the data in response E.g.: CL14

We can increase performance...

- Width increase, pipelining, bursting
- Dual, quad channel (parallelization)

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.



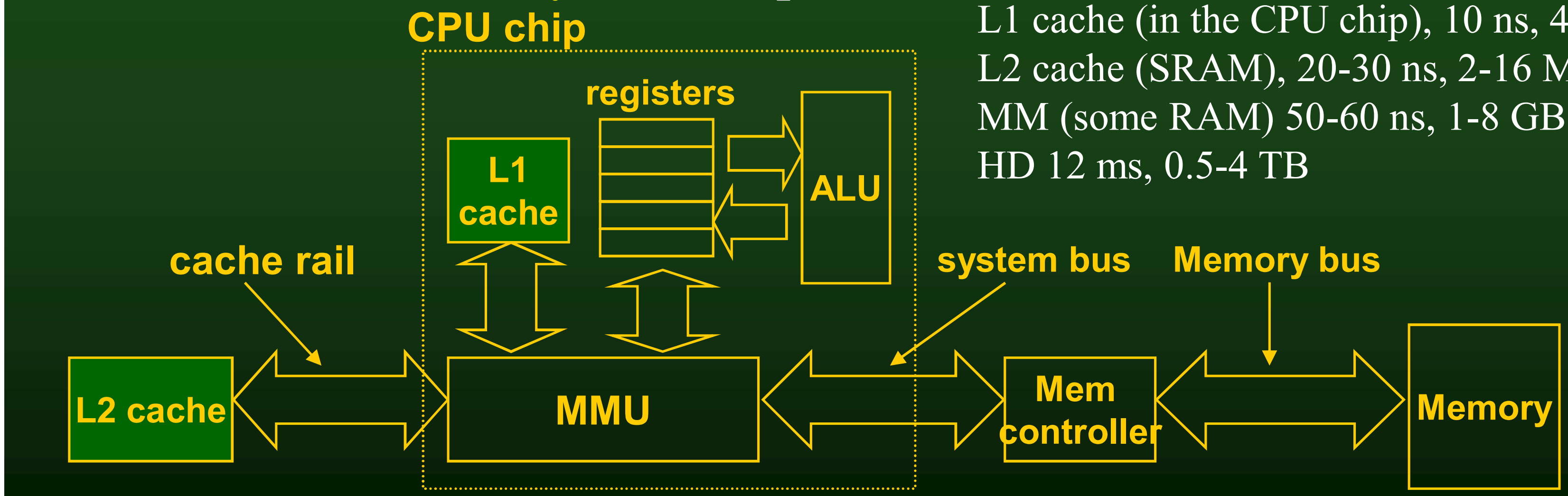
Installing two memory modules into the same channel of the memory controller will force it into single-channel mode.

- Using caches
- Using associative libraries

Caches

- Locality of the programs: experience
- Smaller capacity, but "closer" to the CPU and faster (SRAM) memory, in which
- a part of the central memory content is also there.
- The CPU addresses both the cache and the central memory "at the same time":
  - if there is a match in the cache, it will only be bought from there!
  - data consistency can be a problem !

Orders of magnitude:
L1 cache (in the CPU chip), 10 ns, 4-16 KB
L2 cache (SRAM), 20-30 ns, 2-16 MB
MM (some RAM) 50-60 ns, 1-8 GB
HD 12 ms, 0.5-4 TB



The principle of locality (caches)

- It is a statistically observable property of processes that they use a narrow part of their address range in a time interval...
  - Temporal locality
    - Linked their addresses again...
  - Spatial locality
    - Their nearby addresses...
- Due to the validity of the principle, it makes sense to use smaller but faster temporary containers...
  - cache; working set; TLB; disk buffer cache...

The 80/20 rule: a process uses 20% of its code-data for 80% of its life

Principle of locality (illustration)

- A process executes a series of instructions (instruction link series),
- Instructions may contain memory references (data references)
- A sequence of references is the reference chain (Reference String):

$\forall\ \omega = r_1, r_2, \ldots\ r_t, \ldots\ r_T$ # $r_{\ t}$: instruction or data reference

Principle of locality (illustration)

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

The machine language program on a register machine is:

Title	Code/data	Comment
4000	LOAD (R1), ONE	R1 index reg initialization
4001	LOAD (R2), n	R2 boundary reg init
4002	COMP R1, R2	i > n testing
4003	JG 4009	Conditional jump
4004	LOAD (R3), B[R1]	Load B[i] into R3
4005	ADD (R3), C[R1]	Addition
4006	STOR A[R1], (R3)	Storing in [i].
4007	ADD (R1), ONE	I increment
4008	JUMP 4002	Cycle again
4009	…	After cycle

Title	Code/data
6000-6999	Storage space for A
7000-7999	Storage space for B
8000-8999	Storage space for C
9000	YOURS
9001	Storage space for n

Then the reference chain is as follows (a total of 9 instruction references, of which 1000 times for 7:

4000, 9000, 4001, 9001,
(4002, 4003, 4004, 700i, 4005, 800i, 4006, 600i, 4007, 4008) ^{1000}
4002, 4003, 4009

Principle of locality (illustration)

- The same code with virtual memory management, in a paging system...
- Let the page size be 1000 , the virtual address: v= (p, o)

  (e.g. the address of [16] 6015 is then v=(6, 15))
- Then the reference chain of the pages (5 pages in total):

  4, 9, 4, 9,

  $(4, 4, 4, 7, 4, 8, 4, 6, 4, 4)^{1000}$

  4, 4, 4
- A page error is also unlikely for a small working set.

Levels of caches

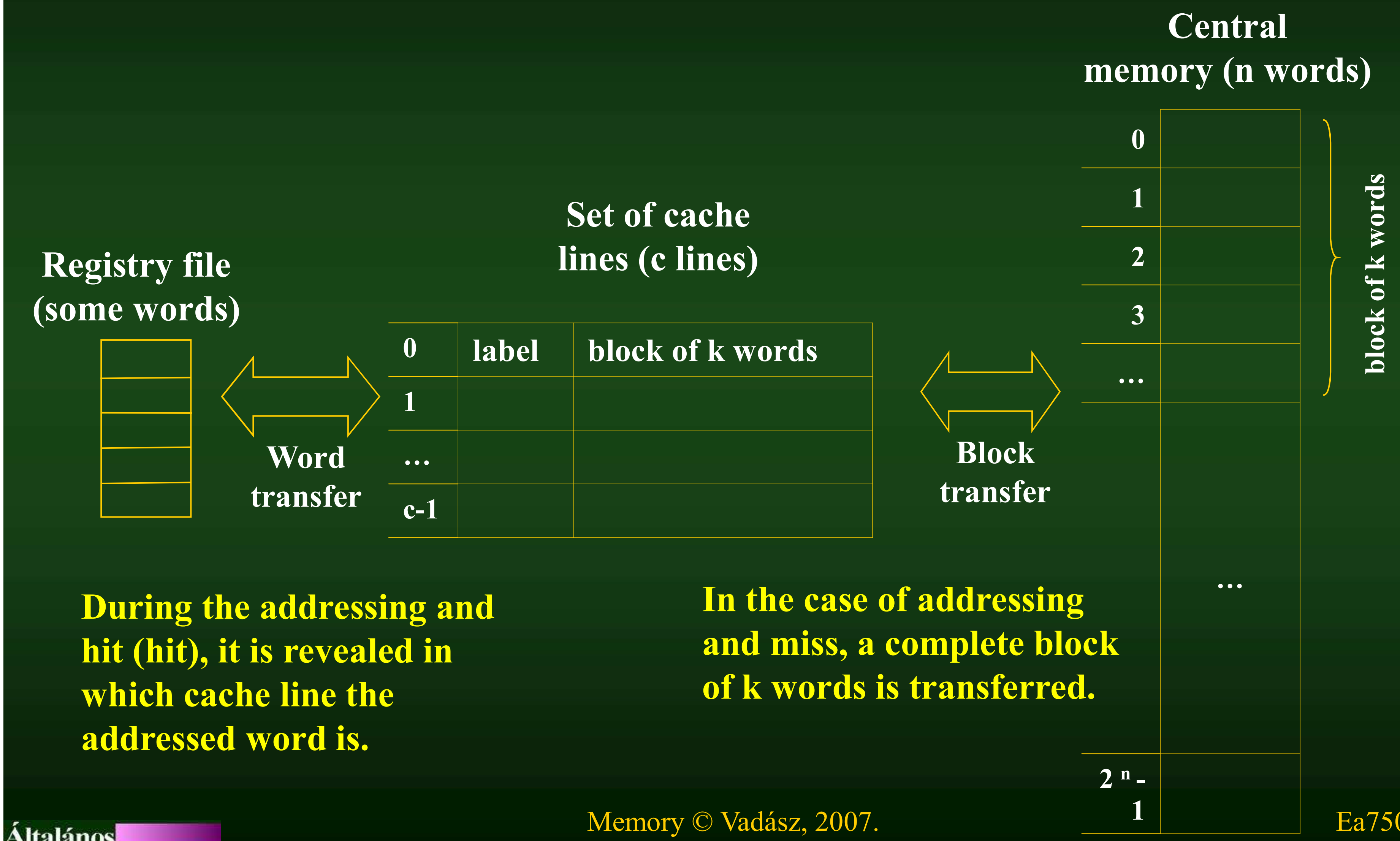
The cache concept is possible between any two levels, where the higher level is faster, although with less capacity. The frequency of turning to the lower level will decrease.

Level 1	within the CPU	~ 10 ns	4 – 16 KB
Level 2	SRAM cache on rail	~ 20 -30 ns	2 to 16 MB
Central memory	DRAM memory on rail	~ 60 - 195 ns	128 – 8192 MB
Discs	Magnetic+mech on I/O rail	~ 10 - 12 ms	0.5 – 6 TB
Internet	On a network	~ 1 s -	almost limitless

The classic cache

- Between the CPU and main memory
- Hardware solution, invisible even to the OS
- Nowadays, it has two levels (L1 and L2 levels), sometimes three
- A common solution is a separate data and instruction cache (it violates the Neumann principle, but it speeds it up!)
- Data consistency must be taken into account during cache design (the same content in both cache and central memory)
  - During addressing, both cache and memory are addressed.
  - If there is a hit in the cache (cache hit). Word transfer.
  - if there is no result (cache miss). Cache miss penalty concept. Block transfer.
  - Memory write (store) problem: consistency assurance

Register , cache and memory structure



Registry file (some words)

Word transfer

Set of cache lines (c lines)

0	label	block of k words
1		
…		
c-1		

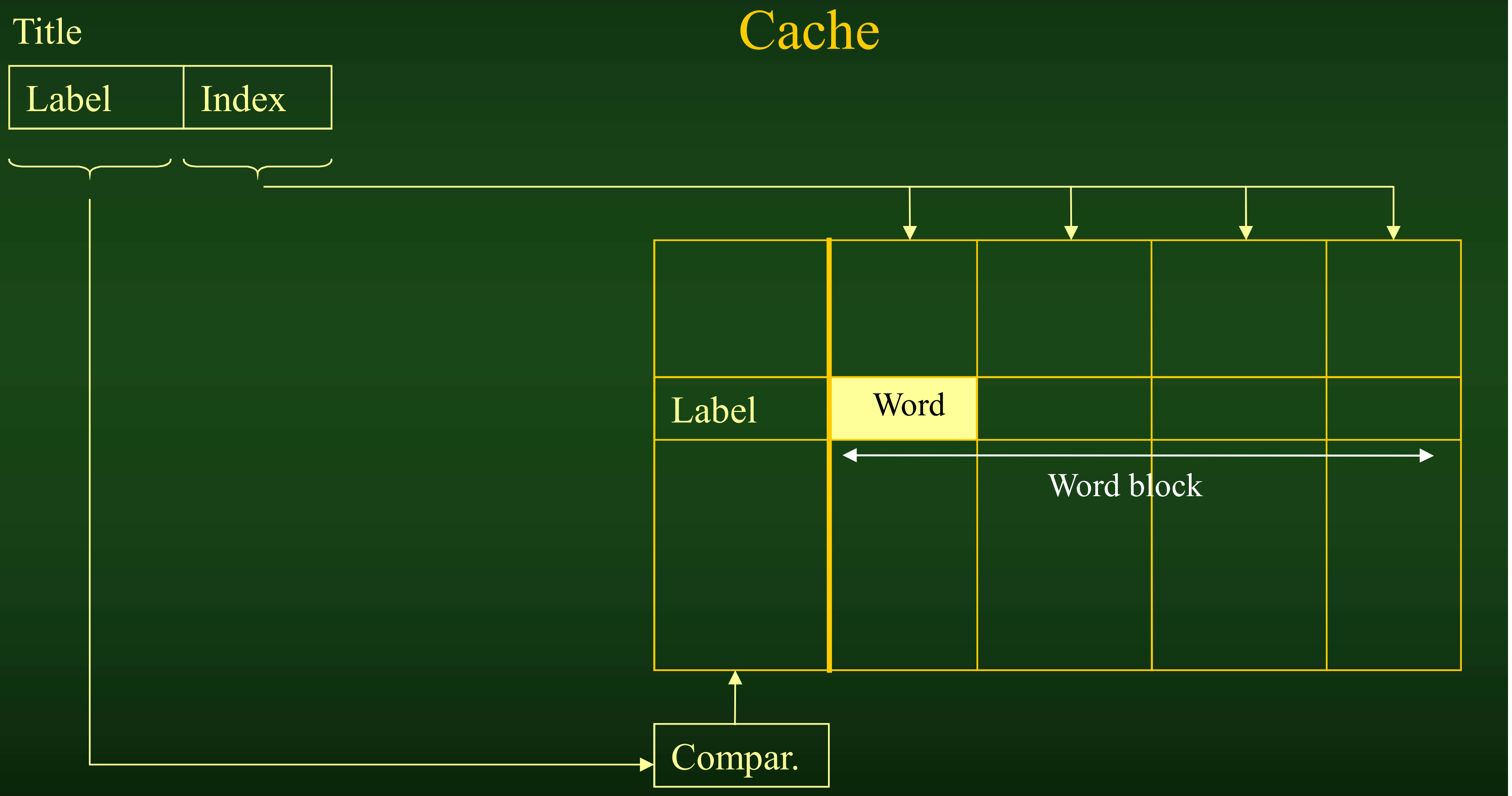
Block transfer

Central memory (n words)

During the addressing and hit (hit), it is revealed in which cache line the addressed word is.

In the case of addressing and miss, a complete block of k words is transferred.

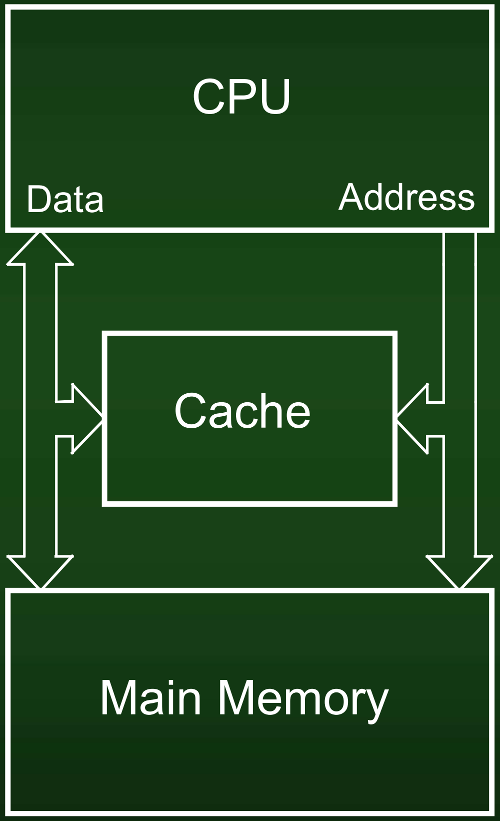
Block cache

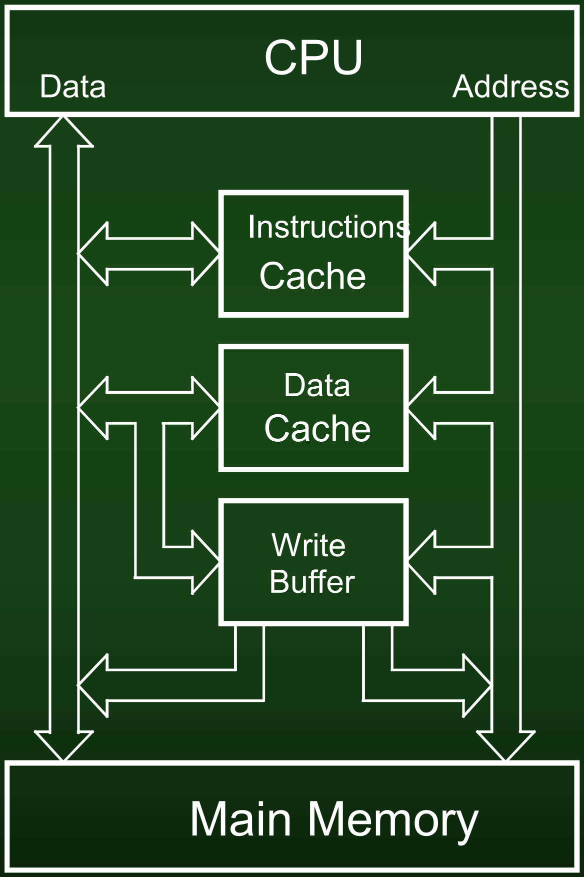


The cache design

- Cache size
- Block size
  - Probability of reusing newly retrieved data
  - Probability of reusing data removed from the cache
- Mapping : which cache space should be occupied by each block
- Replacement algorithm: which block should be removed from the cache if a new one needs space (LRU : Least Recently used)
- Write Policy (during store)
  - Also write to memory if we wrote to a cache block,
  - Write to the memory only when the block is replaced. Write Buffer cache.

Simple and Write Buffered Cache

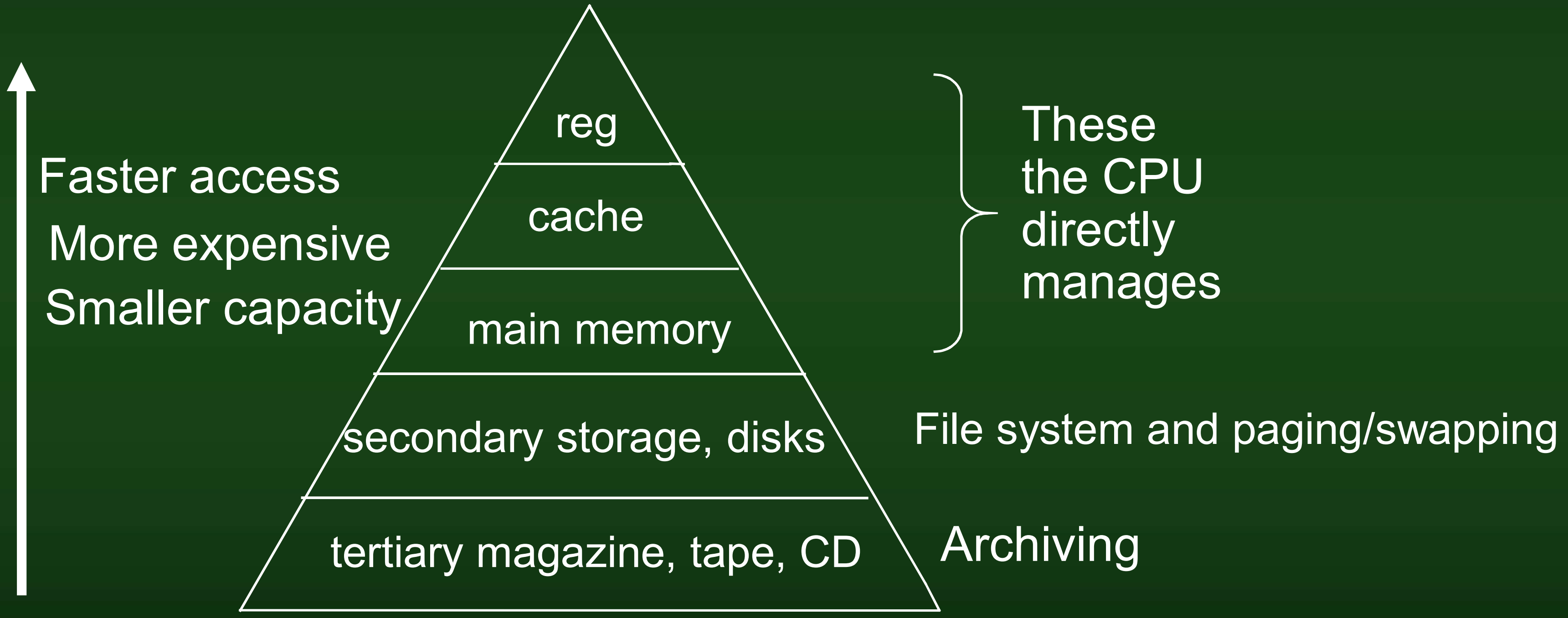




The associative library

- A memory that can be addressed by its content
- Translation Lookaside Buffer
- The storage close to the CPU helps with memory management
- We will discuss it later in the OS subject.

Memory is a hierarchy



Computer architectures

The memory

End