ARCOS Group

uc3m | Universidad Carlos III de Madrid

L5: Memory hierarchy (1)

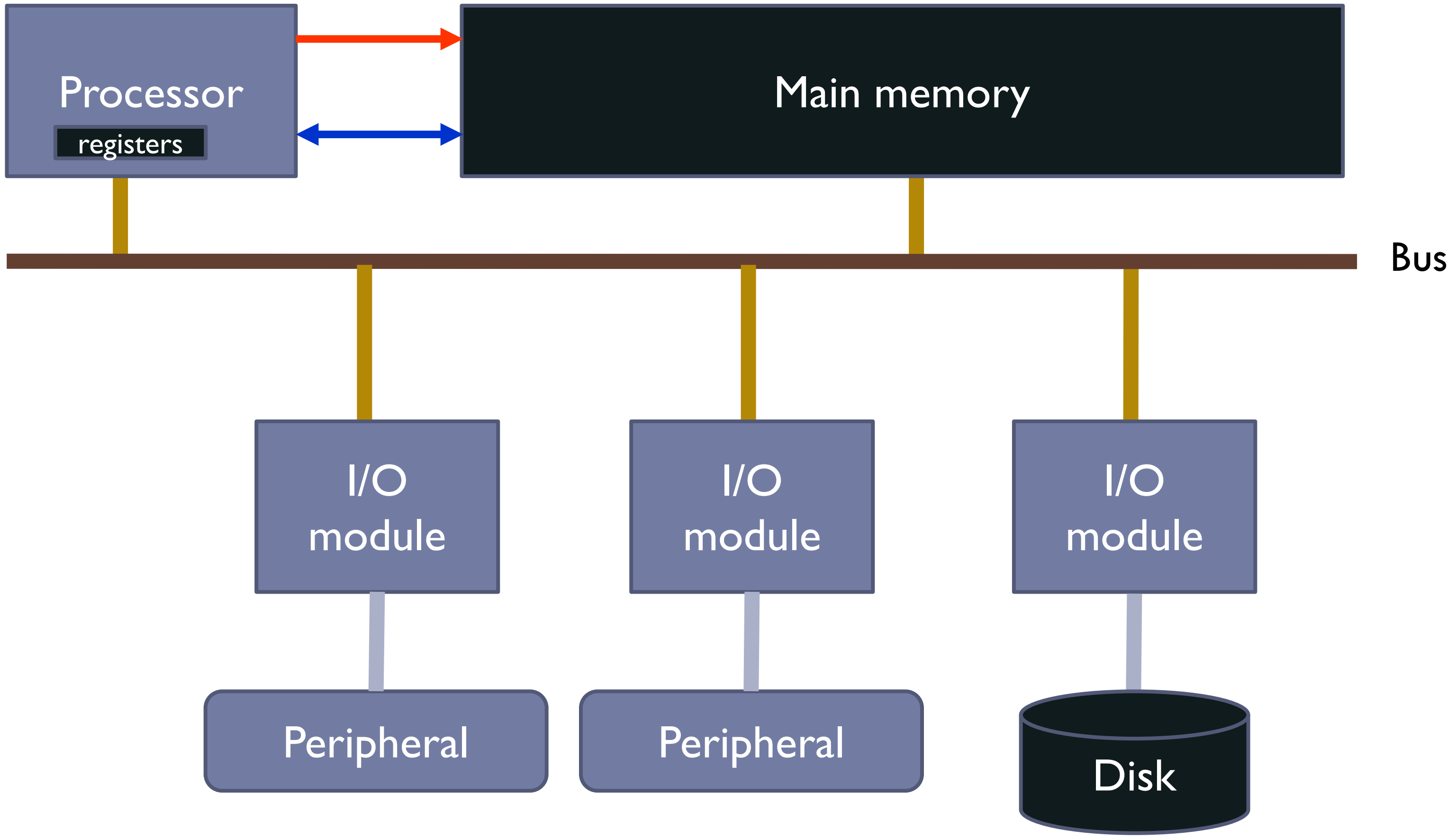
Computer Structure

Bachelor in Computer Science and Engineering
Bachelor in Applied Mathematics and Computing
Dual Bachelor in Computer Science and Engineering and Business Administration

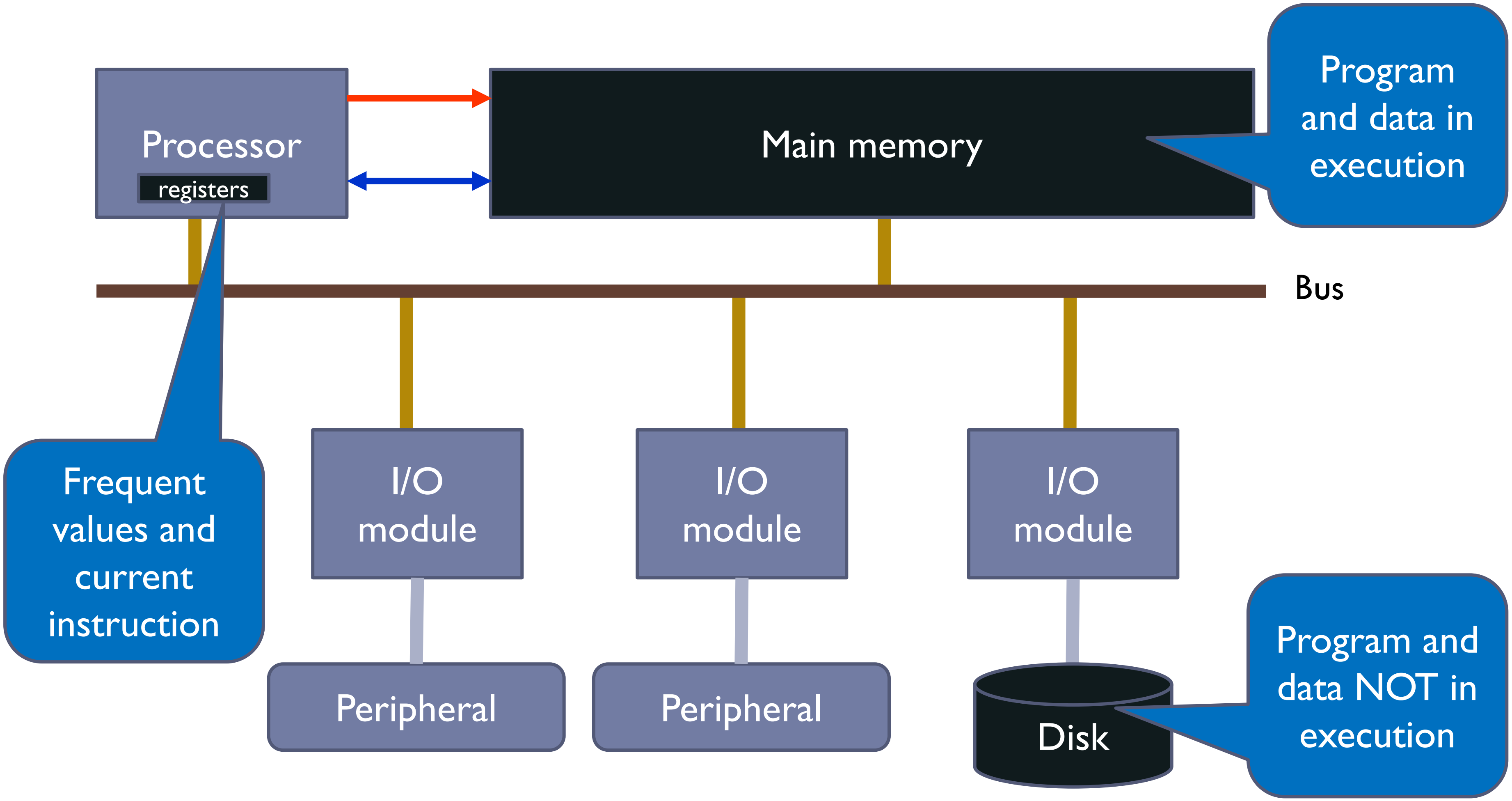
Contents

1. Types of memories
2. Memory hierarchy
3. Main memory
4. Cache memory
5. Virtual memory

Computer overview



Types of memories (hasta el momento)

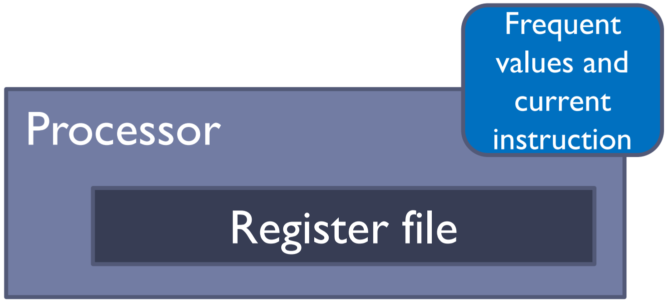


Program and data in execution

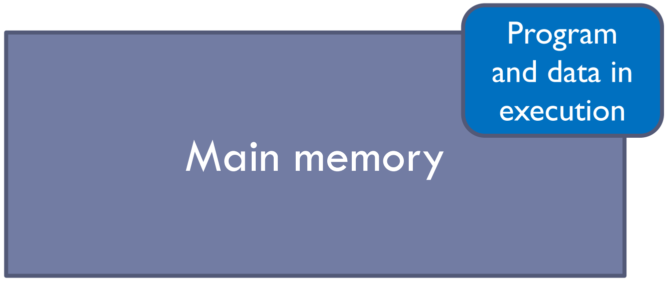
Frequent values and current instruction

Program and data NOT in execution

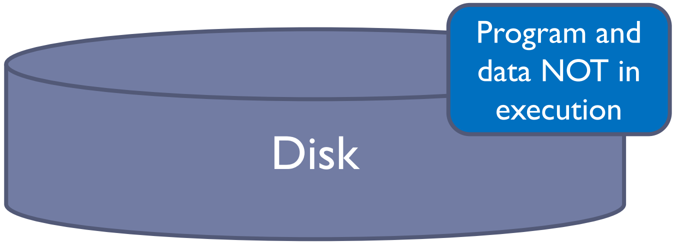
Types of memories (so far)



- Capacity: stores few data
- Access time: around ns.



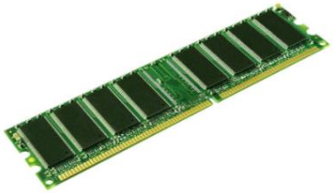
- Capacity: in the order of GB
- Access time: 40 – 100 ns.
  - 1 M.M. access = many clock cycles



- Capacity: almost unlimited (replaceable)
- Access time: ~milliseconds (slow)

Different types of physical devices

- Semiconductor memories
  - Electronic circuits
  - E.g.: RAM, ROM y Flash



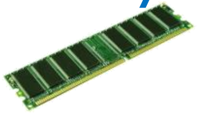
- Magnetic memories
  - Information on a magnetized surface
  - E.g.: hard disk and tapes

- Optic memories
  - Information engraved with a laser that generates perforations on a surface
  - E.g.: CD, DVD and Blu-ray

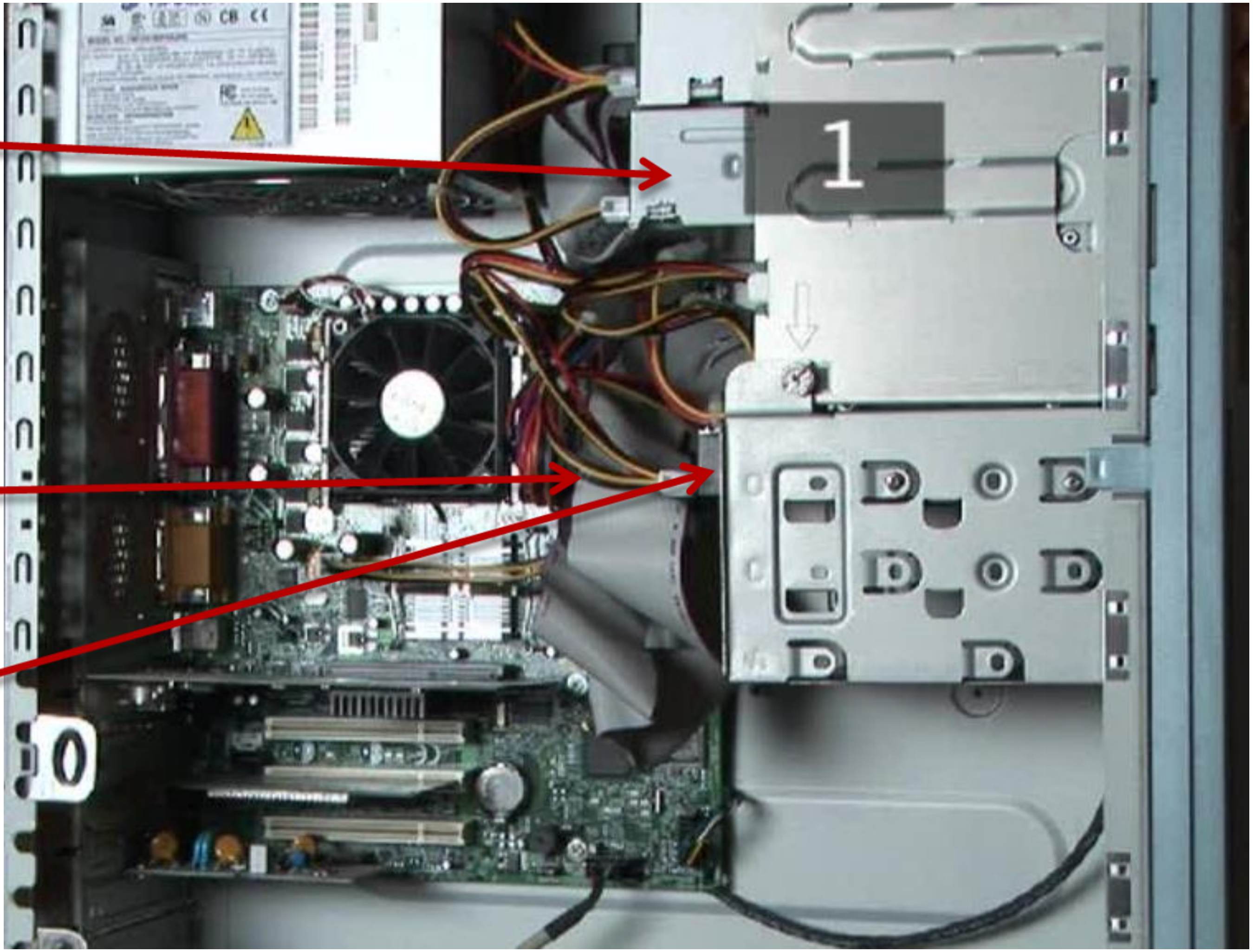
Where is it located?

CD-ROM/
DVD-ROM/
BluRay/…

RAM
memory



Hard disk



http://www.videojug.com/film/what-components-are-inside-my-computer

Main features

- Data Permanency:
  - Volatile (e.g. RAM)
  - Non-volatile (e.g. ROM, Flash)
- Types of operations:
  - Read and write: RAM
  - Read-only: ROM
- Organization:
  - Storage unit:
    - Bits, bytes, words, blocks, etc.
  - Access mode:
    - Sequential (e.g., magnetic tape),
    - Random (RAM): can be accessed in any order. Same access time
- Performance:
  - Access time: time between submitting address and obtaining data.
  - Bandwidth or Transfer rate: amount of data accessed per unit of time.
- Other:
  - Capacity: amount of data that can be stored
  - Cost: price per unit of storable data

Size units

- Usually expressed in bytes (octet):

byte	1 byte = 8 bits	
kilobyte	1 KB = 1.024 bytes	2^{10} bytes
megabyte	1 MB = 1.024 KB	2^{20} bytes
gigabyte	1 GB = 1.024 MB	2^{30} bytes
terabyte	1 TB = 1.024 GB	2^{40} bytes
petabyte	1 PB = 1.024 TB	2^{50} bytes
exabyte	1 EB = 1.024 PB	2^{60} bytes
zettabyte	1 ZB = 1.024 EB	2^{70} bytes
yottabyte	1 YB = 1.024 ZB	2^{80} bytes

Size units (with care)

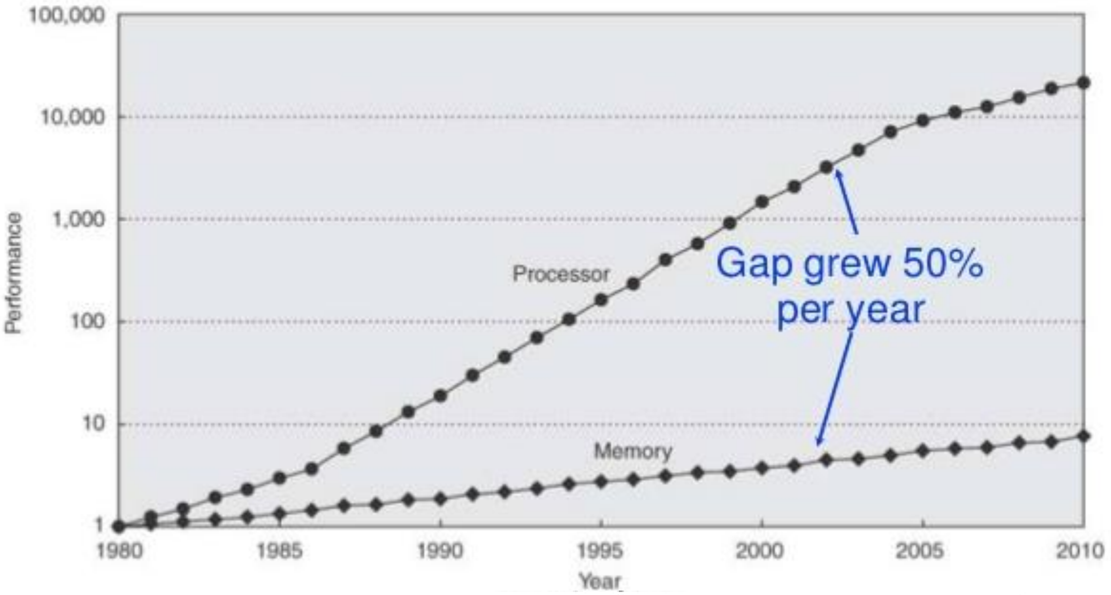
- In communication the kilobit is usually used instead of the kilobyte (1 Kb <> 1 KB) and powers of 10:
  - 1 Kb = 1.000 bits
  - 1 KB = 1.000 bytes

- In storage (hard disks) some manufacturers do not use powers of two, but powers of 10:

kilobyte	1 KB = 1.000 bytes	10^3 bytes
megabyte	1 MB = 1.000 KB	10^6 bytes
gigabyte	1 GB = 1.000 MB	10^9 bytes
terabyte	1 TB = 1.000 GB	10^{12} bytes

  - …

Performance evolution



Source: Computer Architecture, A Quantitative Approach by John L. Hennessy and David A. Patterson

- Processors
  - 1980-2000: 60% of annual average increase
- DRAM memories
  - 1980-2000: 7% of annual average increase
- Distance between memory and CPU increases every year

different access times to memory…

- Registers access time
  - ~1 ns

A library in UC3M...

- SRAM access time
  - ~2-5 ns

A library in UAB…

- DRAM access time
  - ~70-100 ns

A library in Florida…

Exercise 1

Number of memory accesses

```
int i;
int s = 0;
for (i=0; i<1000; i++)
     s = s + i;
i=0;
```

- How many memory accesses are generated in this code fragment?

Exercise 1

Number of memory accesses

```c
int i;
int s = 0;
for (i=0; i<1000; i++)
     s = s + i;
i=0;
```

```
         li   t0, 0     # s
         li   t1, 0     # i
         li   t2, 1000
bucle1: bge  t1, t2, fin1
         add  t0, t0, t1
         addi t1, t1, 1
         beq  x0, x0, bucle1
fin1:    li   t1, 0
```

- How many memory accesses are generated in this code fragment?

Exercise 1

Number of memory accesses

```
int i;
int s = 0;
for (i=0; i<1000; i++)
      s = s + i;
i=0;
```

```
         li   t0, 0     # s
         li   t1, 0     # i
         li   t2, 1000
bucle1: bge  t1, t2, fin1
         add  t0, t0, t1
         addi t1, t1, 1
         beq  x0, x0, bucle1
fin1:    li   t1, 0
```

Solution: $3 + 4 \times 1000 + 1 + 1 = 4005$

Exercise 1

Number of memory accesses

```
int i;
int s = 0;
for (i=0; i<1000; i++)
      s = s + i;
i=0;
```

```
         li   t0, 0     # s
         li   t1, 0     # i
         li   t2, 1000
bucle1:  bge  t1, t2, fin1
         add  t0, t0, t1
         addi t1, t1, 1
         beq  x0, x0, bucle1
fin1:    li   t1, 0
```

Solution: 3 + 4 × 1000 + 1 + 1 = 4005

- If memory access time is 60 ns the total time is 240,240 ns
- A processor would use more that 98% waiting for data from main memory

Exercise 2

Number of memory accesses

```
int v[1000];  // global

int i;
for (i=0; i < 1000; i++)
     v[i] = 0;
```

- How many memory accesses are generated in this code fragment?

Exercise 2

Number of memory accesses

```
int v[1000];   // global


int i;

for (i=0; i < 1000; i++)
     v[i] = 0;
```

```
.data
        v: .zero 4000
.text:
         li   t0, 0    # i
         li   t1, 0    # i de v
         li   t2, 1000 # n. eltos
bucle2:  bgt  t0, t2, fin2
         sw   0,  v(t1)
         addi t0, t0, 1
         addi t1, t1, 4
         j    bucle2
fin2:
```

- How many memory accesses are generated in this code fragment?

Exercise 2

Number of memory accesses

```
int v[1000];  // global

int i;
for (i=0; i < 1000; i++)
     v[i] = 0;
```

```
.data
        v: .zero 4000
.text:
         li   t0, 0    # i
         li   t1, 0    # i de v
         li   t2, 1000 # n. eltos
bucle2:  bgt  t0, t2, fin2
         sw   0,  v(t1)
         addi t0, t0, 1
         addi t1, t1, 4
         j    bucle2
fin2:
```

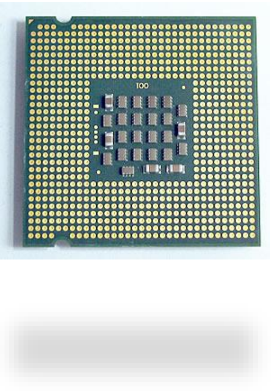
Solution:

3 + 5 × 1000 + 1 + 1000 (additional access of sw) = 6004

Contents

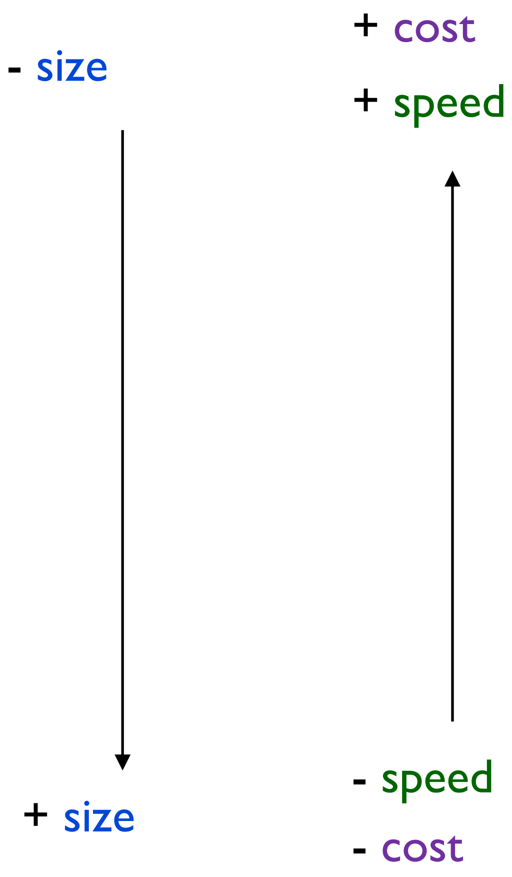
1. Types of memories
2. Memory hierarchy
3. Main memory
4. Cache memory
5. Virtual memory

What would the ideal memory system look like?



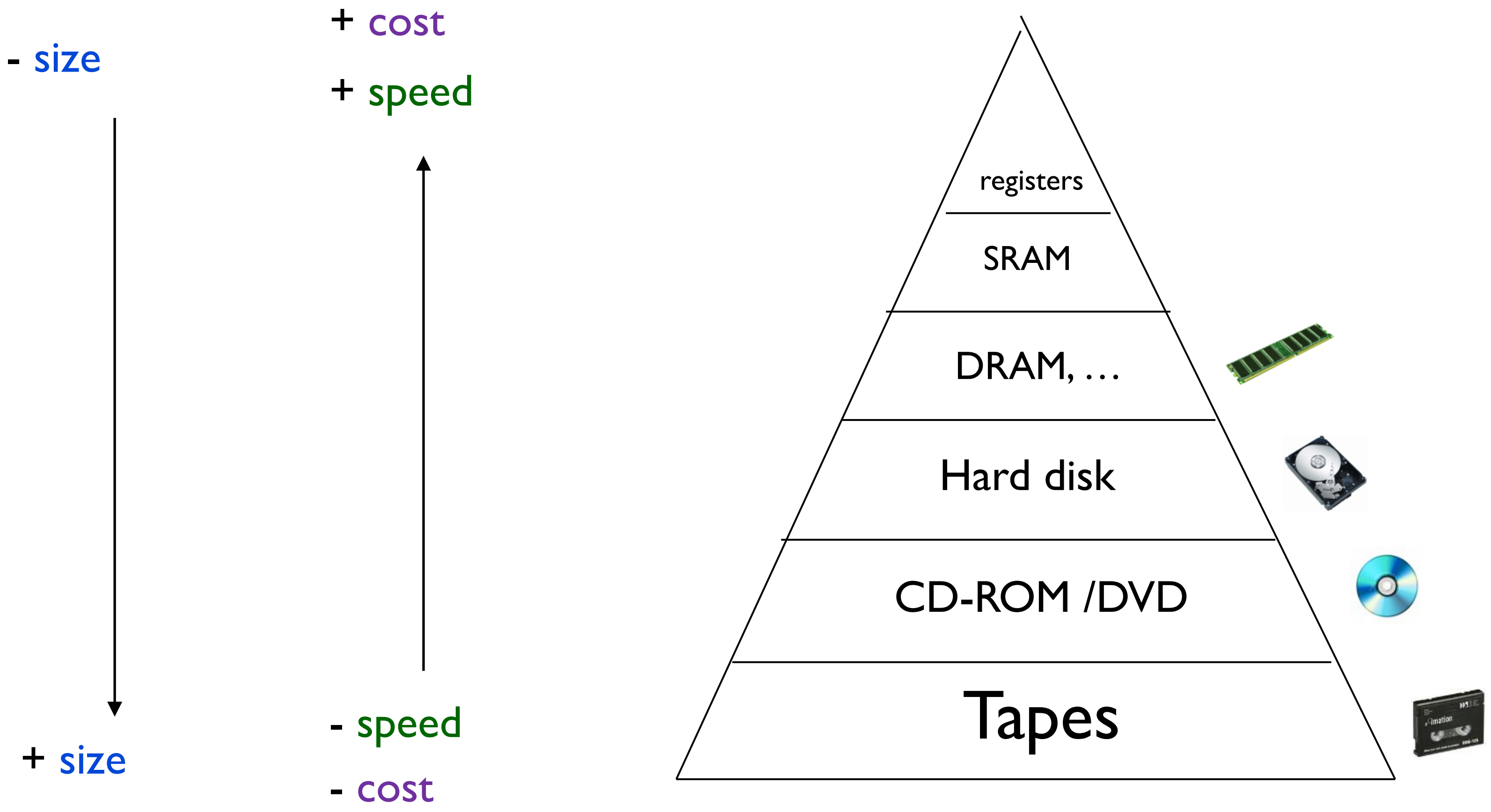
- Minimizes access time
- Maximizes capacity
- Minimizes cost

Reality



- Incompatible goals :
  - + speed - size ⇨
- Different types of memory are used:
  - DRAM, Hard disk, …
- Different types of memory are organized by access speed:
  - Memory hierarchy

Memory hierarchy



- size

+ size

+ cost

+ speed

- speed

- cost

registers

SRAM

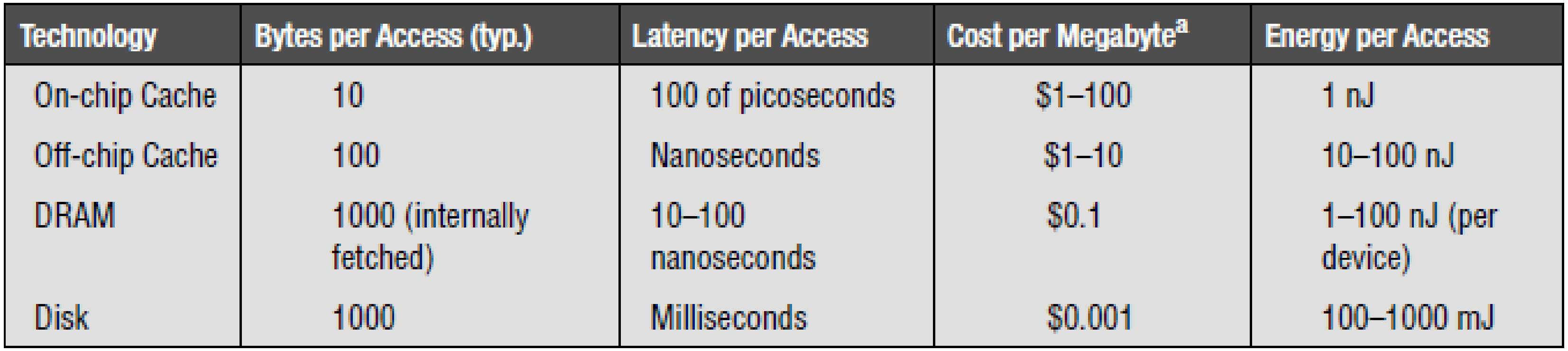
DRAM, …

Hard disk

CD-ROM /DVD

Tapes

Comparison



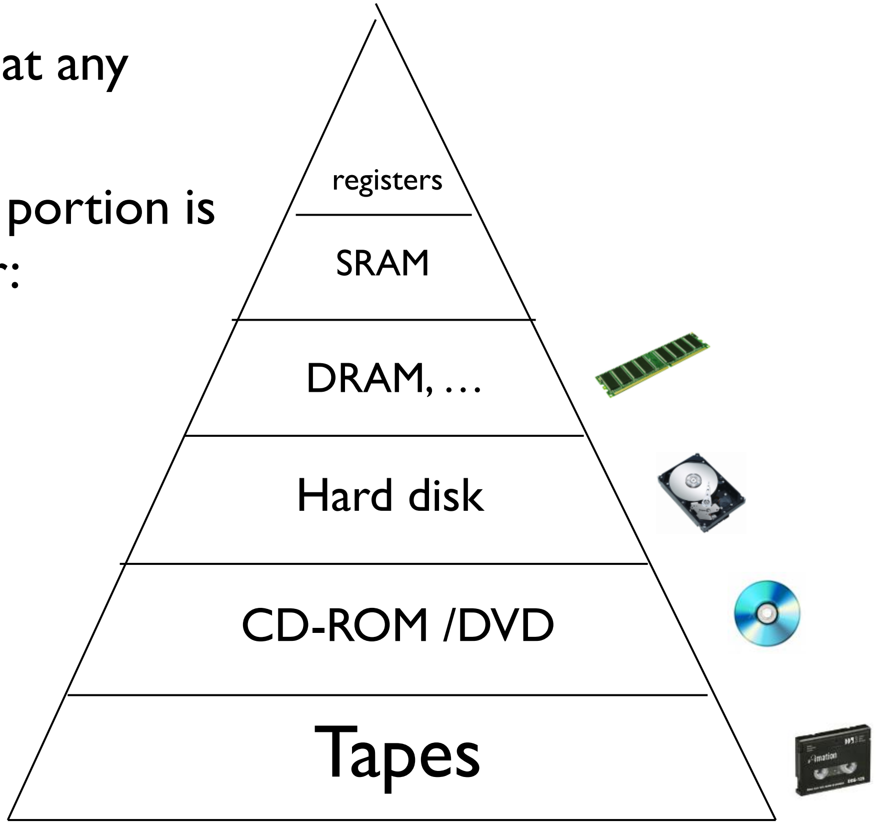
Technology	Bytes per Access (typ.)	Latency per Access	Cost per Megabyte[a]	Energy per Access
On-chip Cache	10	100 of picoseconds	$1–100	1 nJ
Off-chip Cache	100	Nanoseconds	$1–10	10–100 nJ
DRAM	1000 (internally fetched)	10–100 nanoseconds	$0.1	1–100 nJ (per device)
Disk	1000	Milliseconds	$0.001	100–1000 mJ

Memory Systems
Cache, DRAM, Disk
Bruce Jacob, Spencer Ng, David Wang
Elsevier

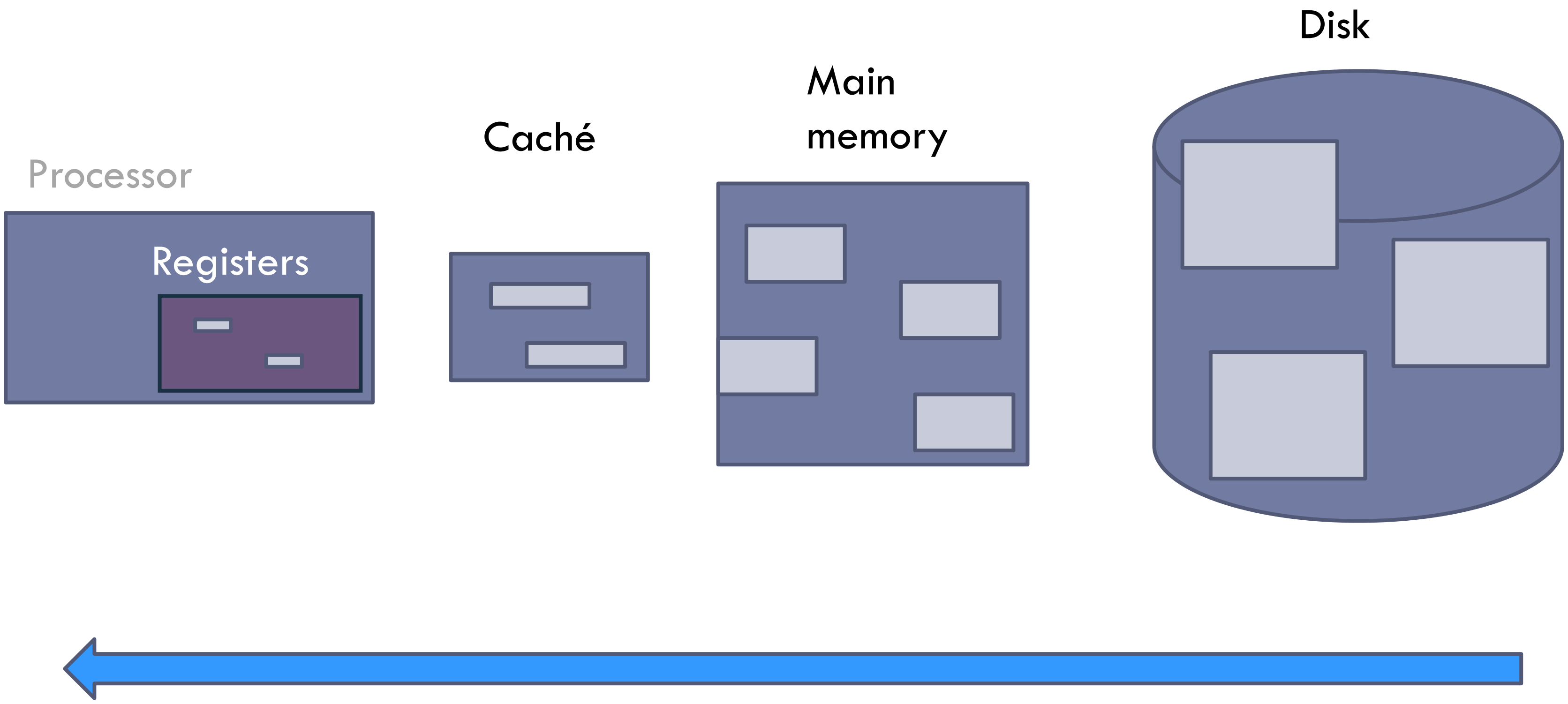
Use of memory hierarchy

- Only in memory what is needed at any given time.
- If it is not present, the necessary portion is copied from one level to another:
  - E.g.: load a program into RAM
- When it is no longer needed, the copy made is deleted.

- Access behavior supports it:
  - Proximity of references



Idea of the memory hierarchy



Memory hierarchy design

- The design of the memory hierarchy is crucial in multicore processors.
- Bandwidth increases with the number of cores
  - An Intel Core i7 generates two memory accesses per core per clock cycle
  - With 4 cores and 3.2 GHz clock frequency
    - 25.6 billion 64-bits data accesses per second +
    - 12.8 billion 128-bits data accesses for instructions = 409.6 GB/s
  - A DRAM memory offers only 6% (25GB/s)
  - It is required:
    - Multi-port memories
    - Cache levels

Contents

1. Types of memories
2. Memory hierarchy
3. Main memory
4. Cache memory
5. Virtual memory

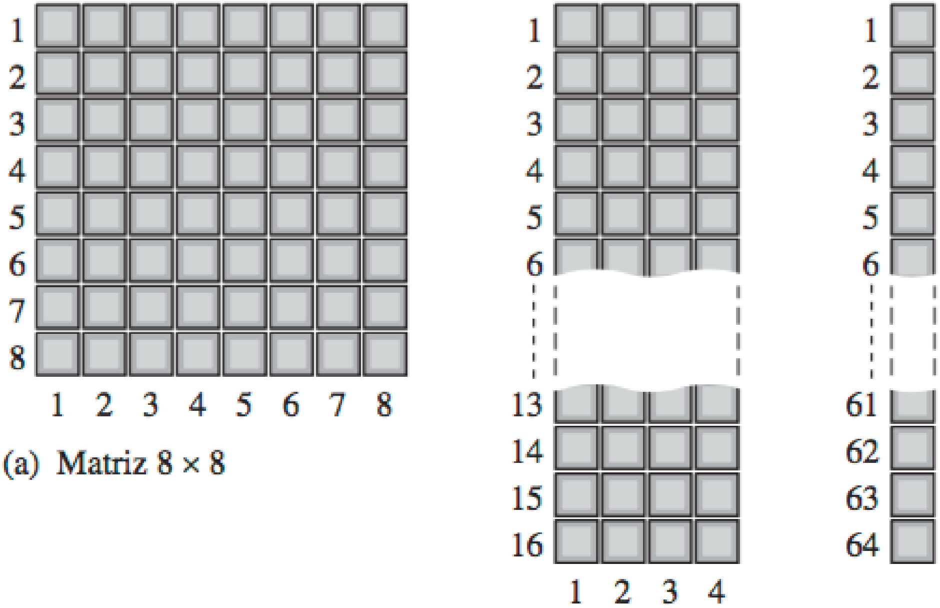
Semiconductor memories

- Read only memory (ROM)
  - Non-volatile memory
    - persistent
  - Example of use: BIOS

- Random access memory (RAM)
  - Volatile memory
    - Not persistent
  - Faster than ROM
  - Example of use: main memory

Semiconductor Memory Matrix

- Each cell stores a 1 or a 0



(a) Matriz 8 × 8

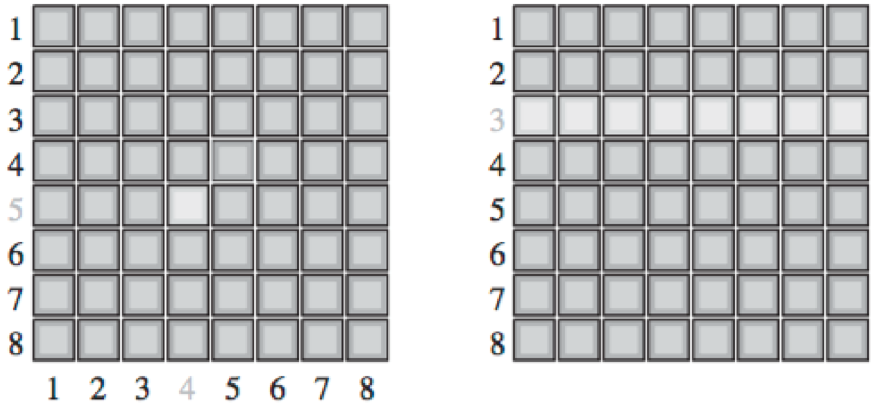
(b) Matriz 16 × 4

(c) Matriz 64 × 1

Fundamentos de Sistemas Digitales
Thomas L. Floyd

Addresses and capacity

- Address: position of a data unit in the memory matrix



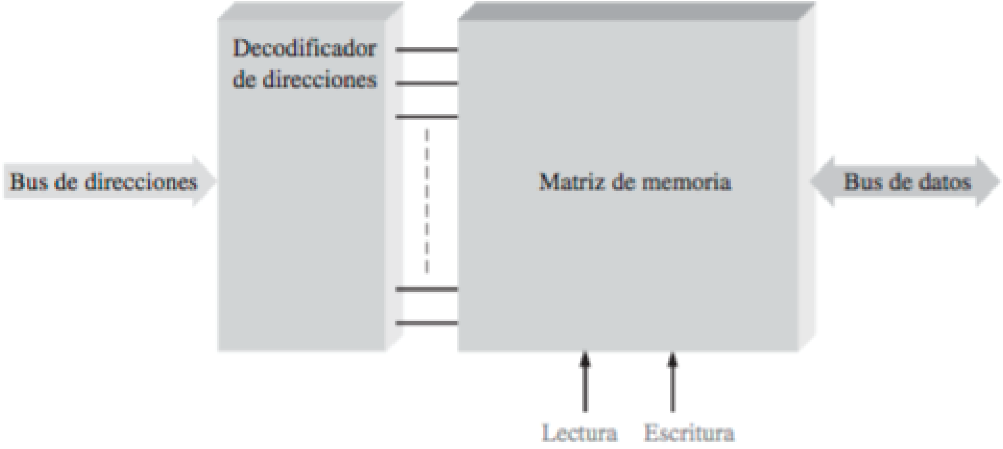
(a) La dirección del bit gris claro es fila 5, columna 4.

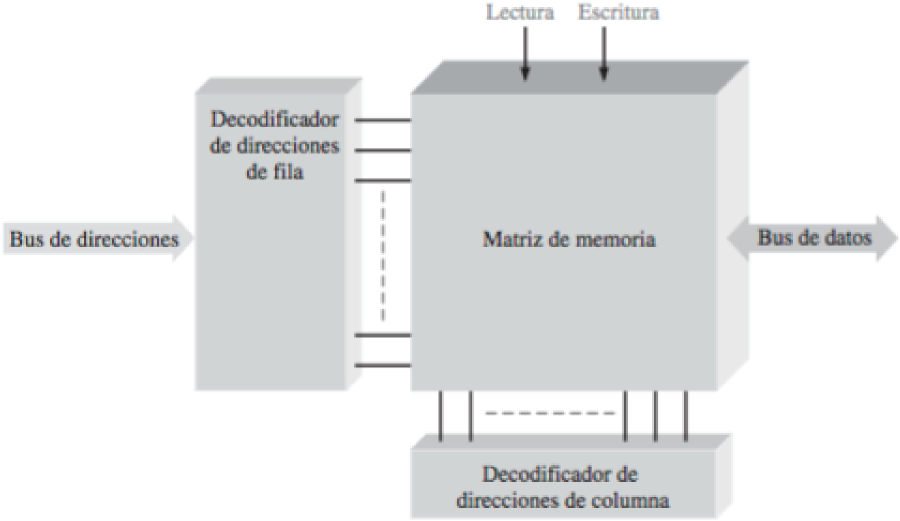
(b) La dirección del byte gris claro es la fila 3.

Fundamentos de Sistemas Digitales
Thomas L. Floyd

- Capacity: total number of data units that can be stored

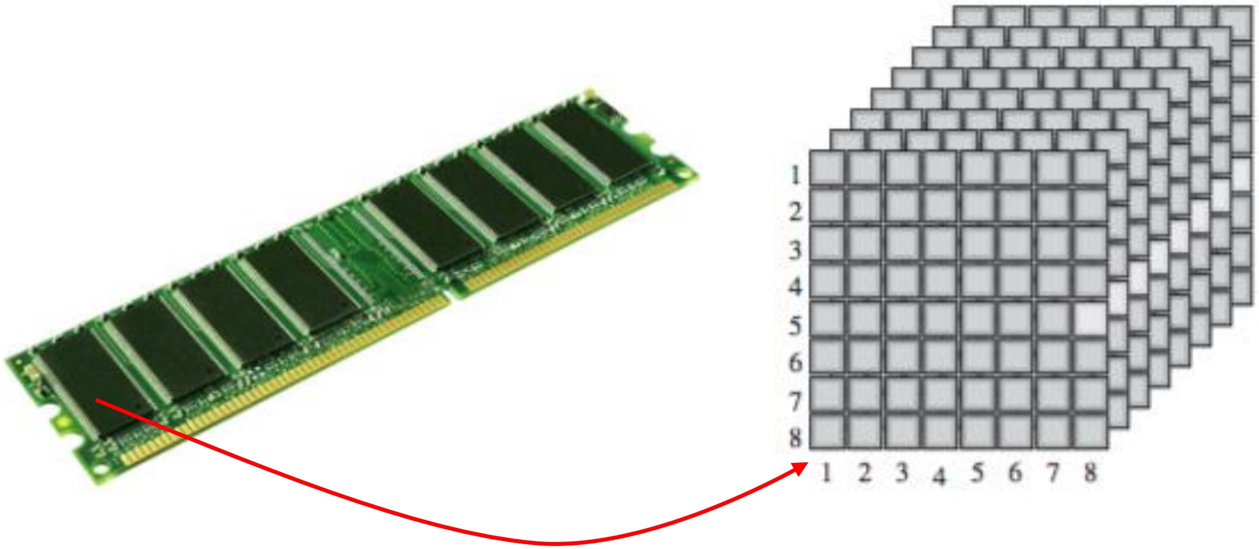
Addressing types




Lectura
Escritura
Decodificador de direcciones de fila
Bus de direcciones
Matriz de memoria
Bus de datos
Decodificador de direcciones de columna

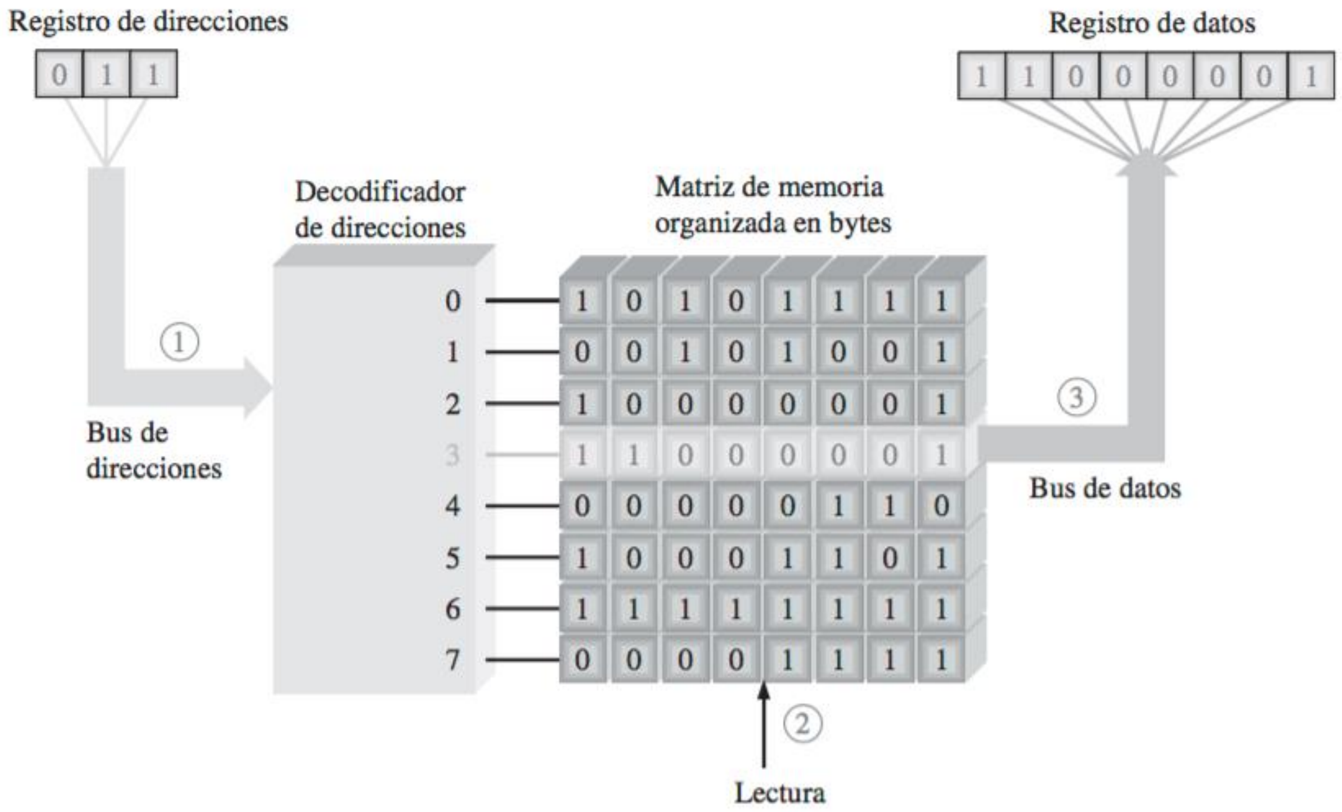
Fundamentos de Sistemas Digitales
Thomas L. Floyd

Example of organization



La dirección del byte en color gris claro corresponde a la fila 5, columna 8.

Read operation



0	1	1

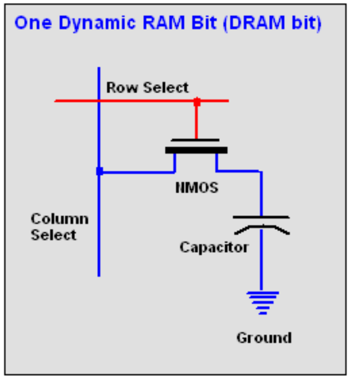
1	1	0	0	0	0	0	1

0	1	0	1	0	1	1	1	1
1	0	0	1	0	1	0	0	1
2	1	0	0	0	0	0	0	1
3	1	1	0	0	0	0	0	1
4	0	0	0	0	0	1	1	0
5	1	0	0	0	1	1	0	1
6	1	1	1	1	1	1	1	1
7	0	0	0	0	1	1	1	1

Fundamentos de Sistemas Digitales
Thomas L. Floyd

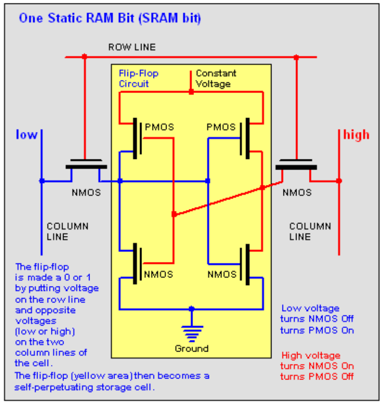
RAM (random access memories)

From Computer Desktop Encyclopedia
© 2005 The Computer Language Co. Inc.



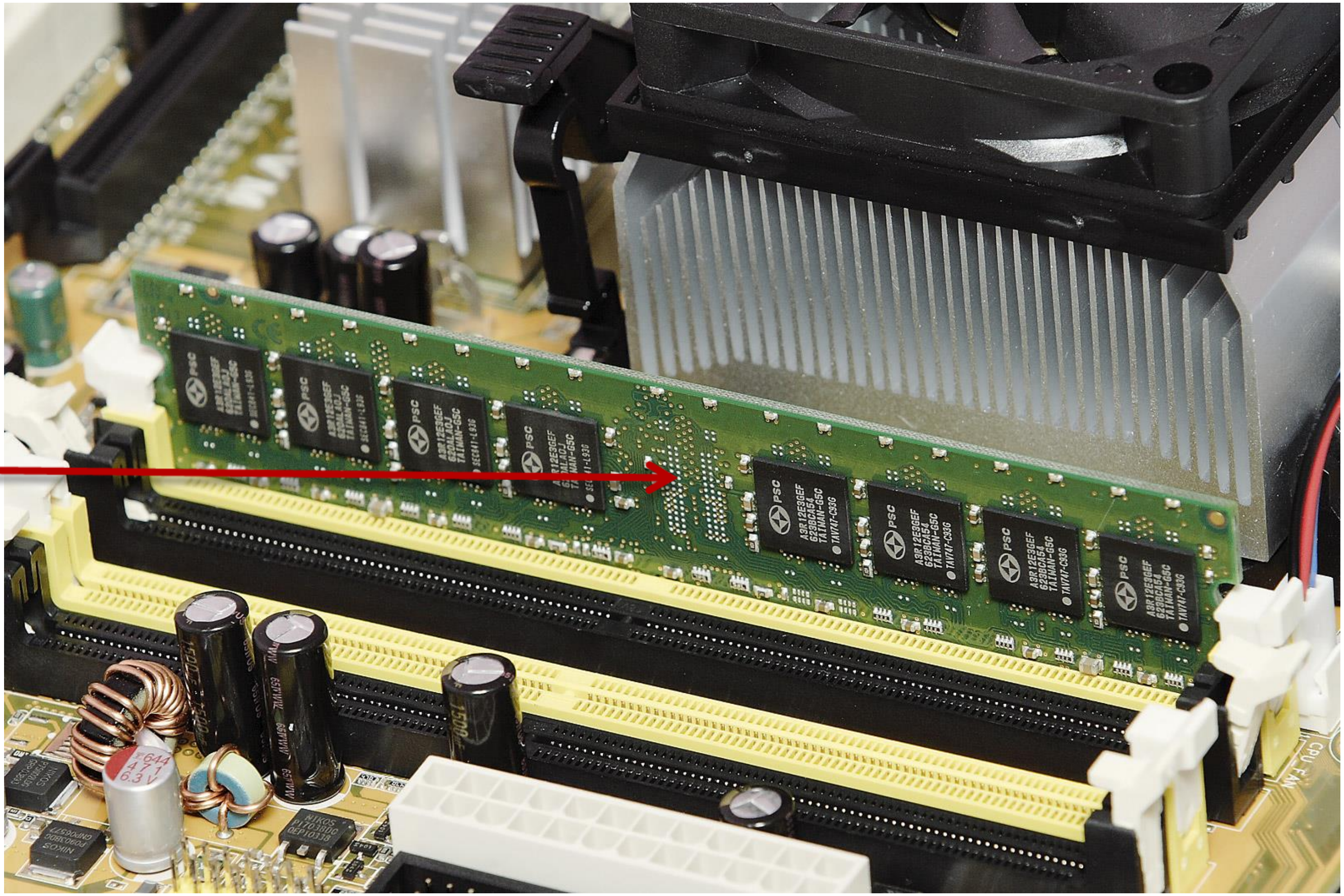
- Dynamic RAM (DRAM)
  - Stores bits as charge in capacitors.
  - Tends to discharge: needs periodic refreshing.
    - Advantage: simpler construction, **more storage**, more cost effective
    - Disadvantage: needs refreshing circuitry, **slower**.
      - □ 2%-3% of clock cycles consumed by the refresh
    - Used in main memory

From Computer Desktop Encyclopedia
© 2005 The Computer Language Co. Inc.


One Static RAM Bit (SRAM bit)
ROW LINE
Flip-Flop Circuit
Constant Voltage
PMOS
PMOS
low
high
NMOS
NMOS
COLUMN LINE
COLUMN LINE
NMOS
NMOS
Ground
The flip-flop is made a 0 or 1 by putting voltage on the row line and opposite voltages (low or high) on the two column lines of the cell.
The flip-flop (yellow area) then becomes a self-perpetuating storage cell.
Low voltage turns NMOS Off turns PMOS On
High voltage turns NMOS On turns PMOS Off

- Static RAM (SRAM)
  - Stores bits as on and off switches.
  - Tends **not** to discharge: does **not** need refreshing.
    - Advantage: No need for refresh circuitry, **faster**.
    - Disadvantage: Complex construction, **less storage**, more expensive.
    - Used in memory caches

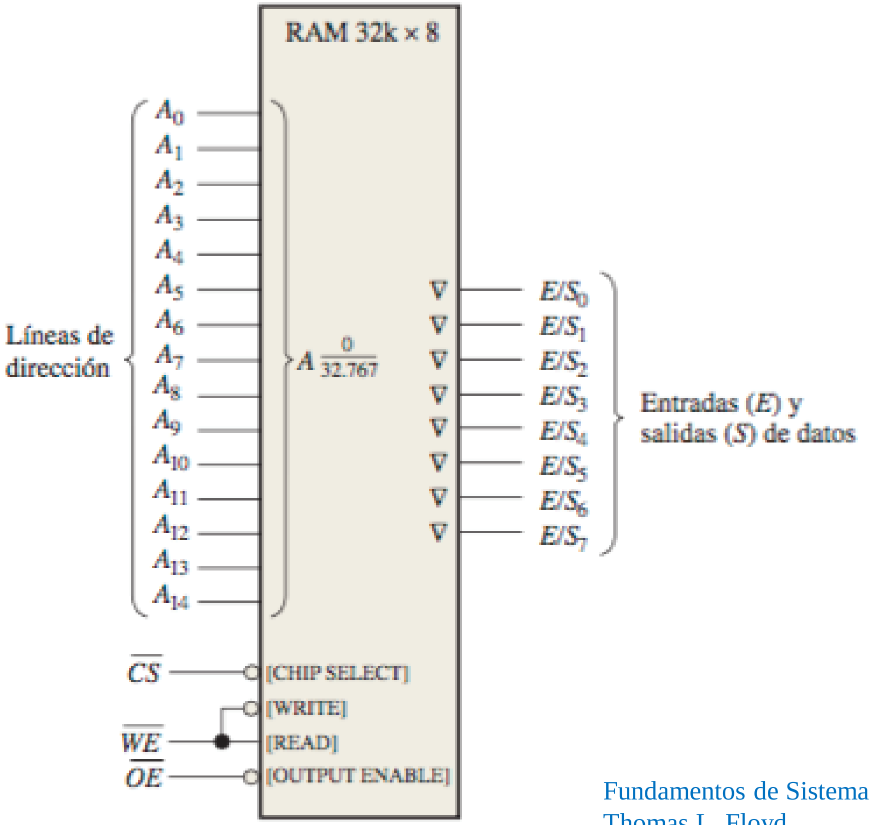
Where is the DRAM memory located?



DRAM memory

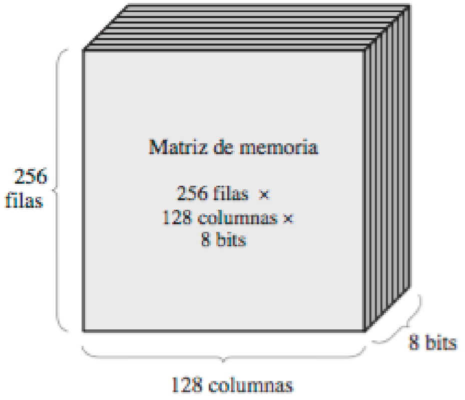
http://en.wikipedia.org/wiki/Primary_storage#Primary_storage

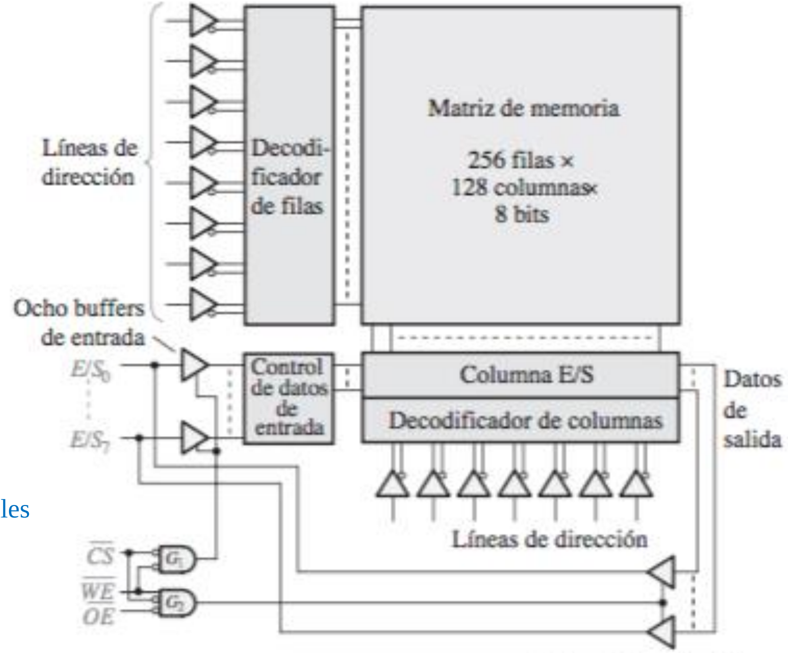
SRAM memory example



Logical organization

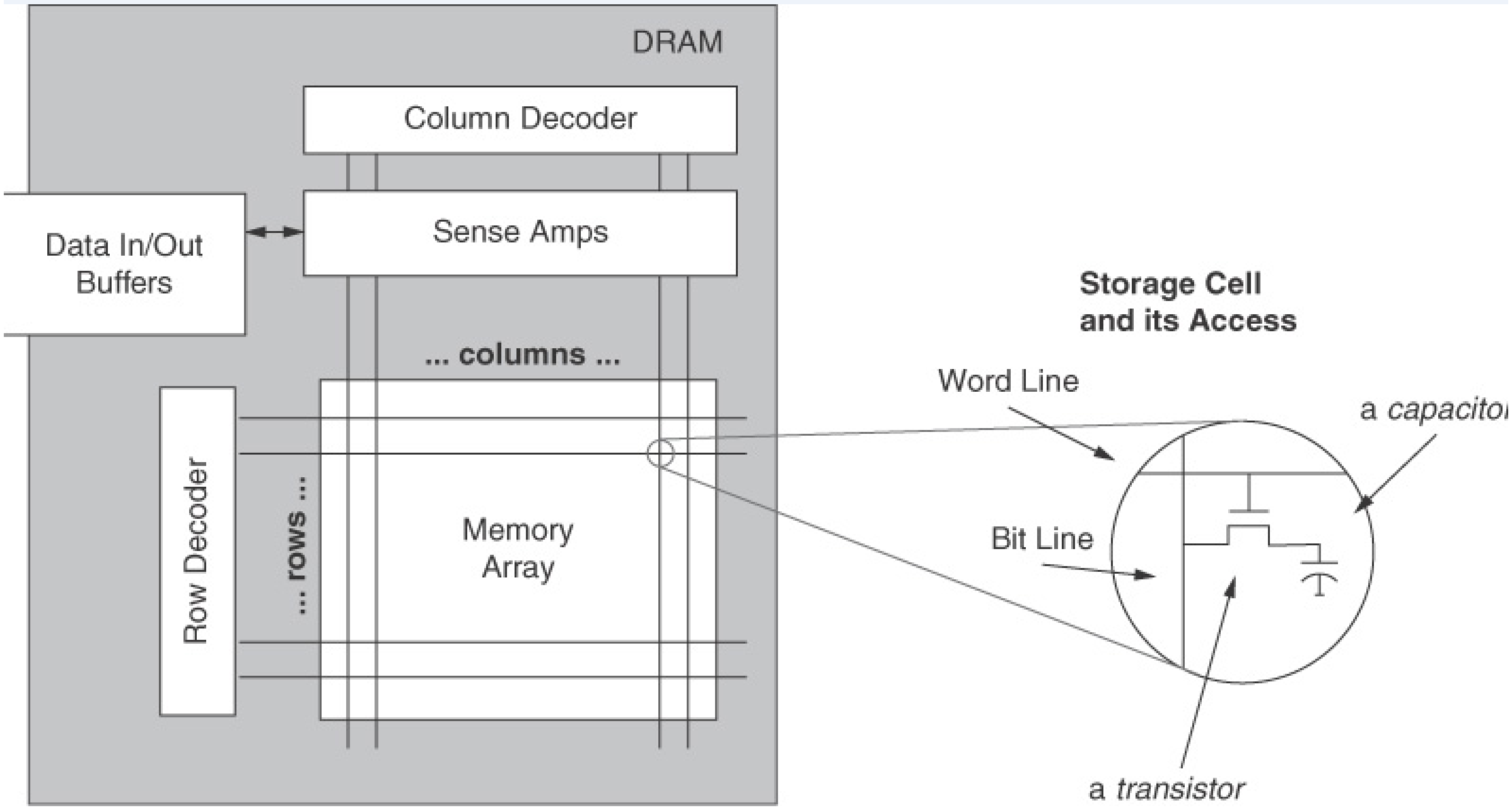
Fundamentos de Sistemas Digitales
Thomas L. Floyd





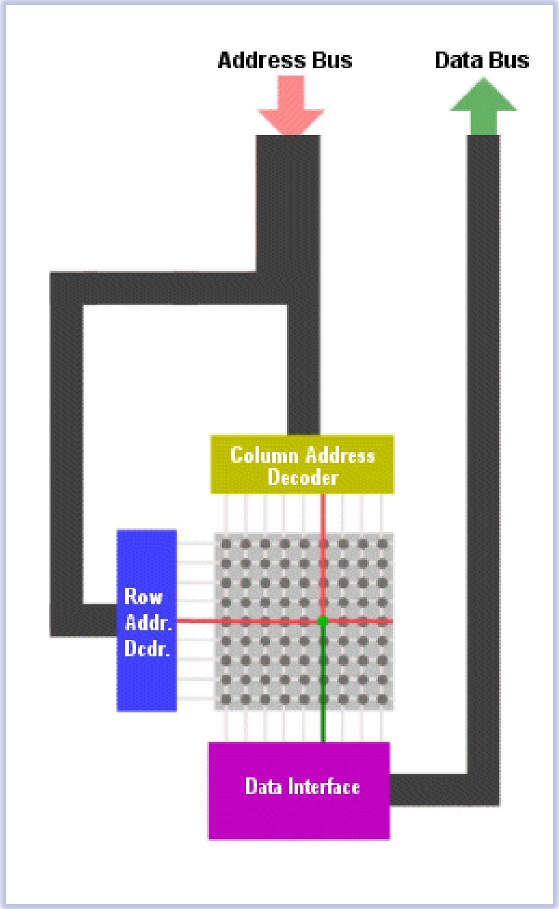
Physical organization

DRAM structure

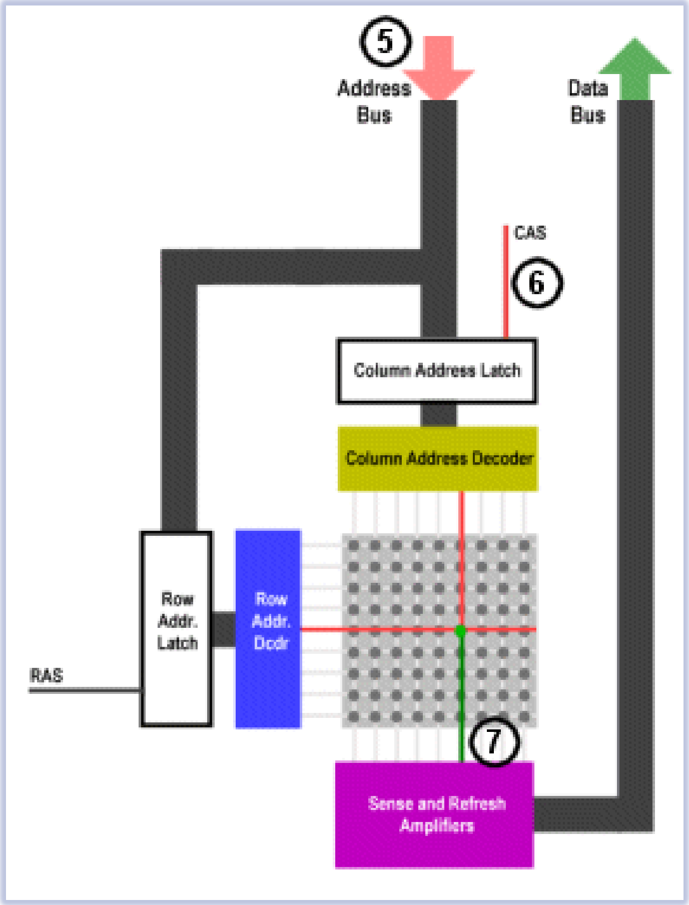


Memory Systems
Cache, DRAM, Disk
Bruce Jacob, Spencer Ng, David Wang
Elsevier

Address multiplexing in DRAM

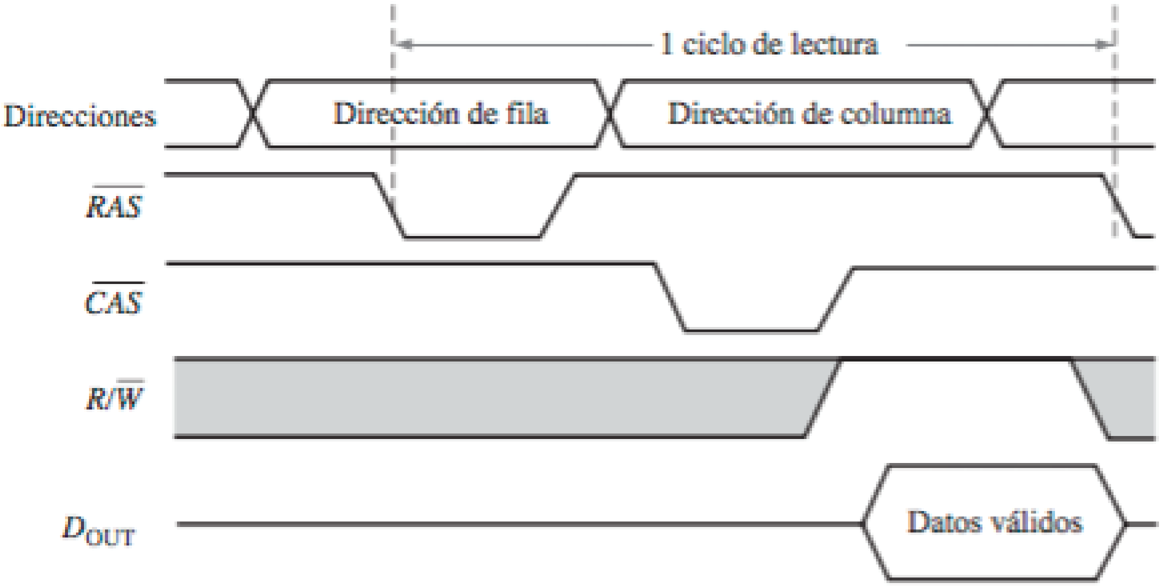


Row/column addressing



Row/column addressing with CAS/RAS

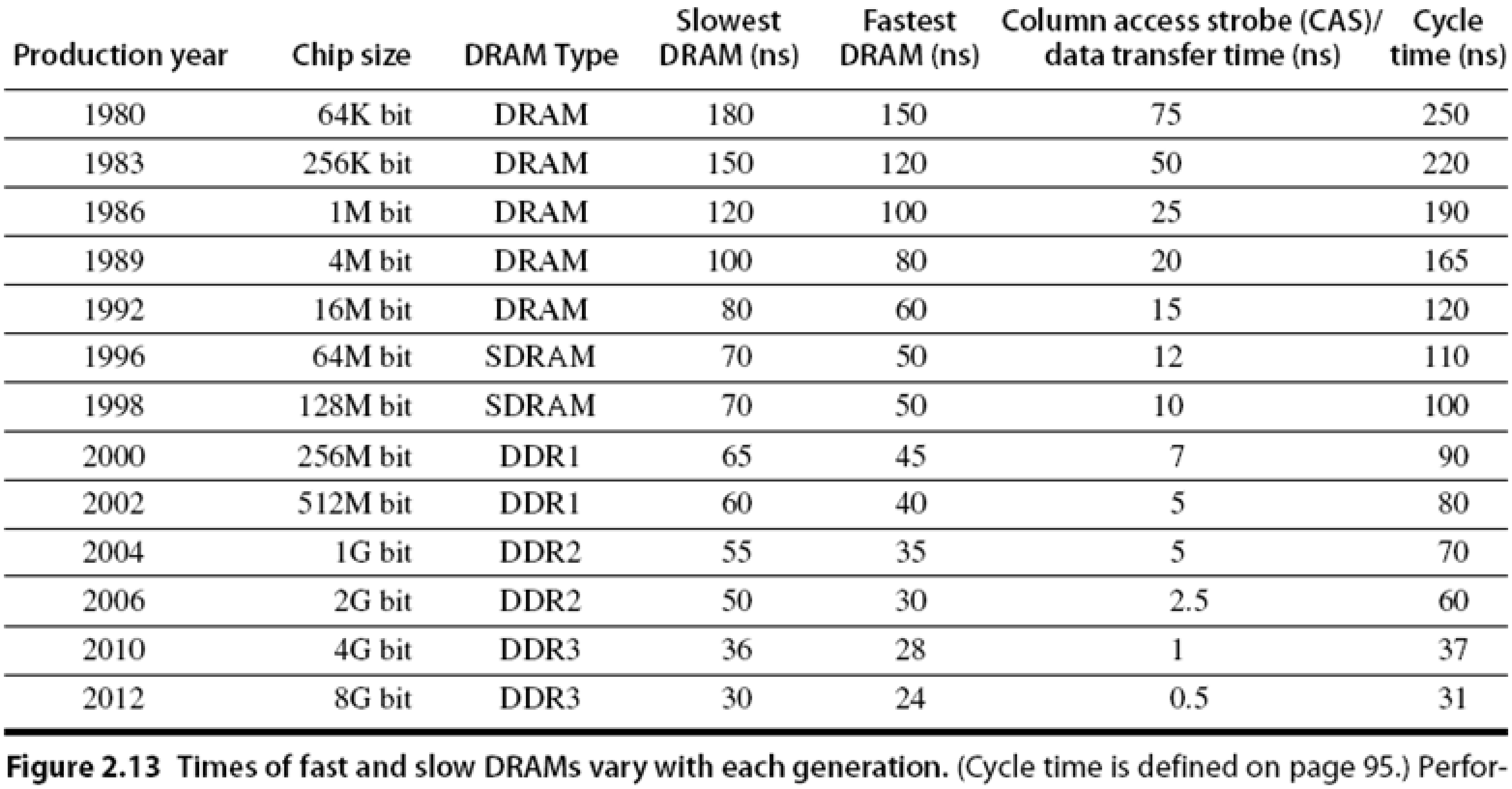
Read operation with CAS/RAS



Refresh cycles

- A DRAM stores a bit in a capacitor.
- This charge degrades with time and temperature
- Each bit needs to be refreshed
- Typically, a DRAM must be refreshed every few milliseconds.
- A read operation refreshes all the addresses in a row.
- A DRAM uses refresh cycles

DRAM memory speed

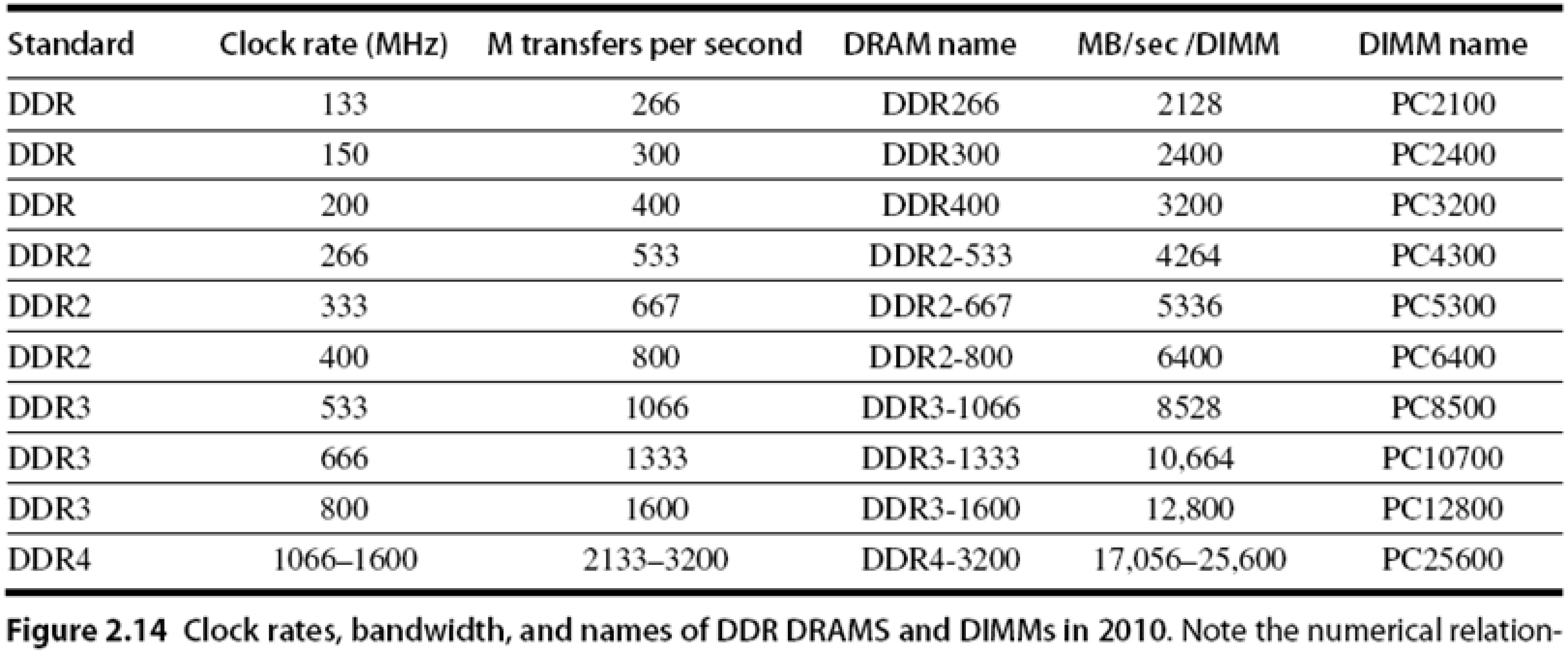


Production year	Chip size	DRAM Type	Slowest DRAM (ns)	Fastest DRAM (ns)	Column access strobe (CAS)/ data transfer time (ns)	Cycle time (ns)
1980	64K bit	DRAM	180	150	75	250
1983	256K bit	DRAM	150	120	50	220
1986	1M bit	DRAM	120	100	25	190
1989	4M bit	DRAM	100	80	20	165
1992	16M bit	DRAM	80	60	15	120
1996	64M bit	SDRAM	70	50	12	110
1998	128M bit	SDRAM	70	50	10	100
2000	256M bit	DDR1	65	45	7	90
2002	512M bit	DDR1	60	40	5	80
2004	1G bit	DDR2	55	35	5	70
2006	2G bit	DDR2	50	30	2.5	60
2010	4G bit	DDR3	36	28	1	37
2012	8G bit	DDR3	30	24	0.5	31

Figure 2.13 Times of fast and slow DRAMs vary with each generation. (Cycle time is defined on page 95.) Perfor-

Patterson y Hennesy

Types of DDR memories



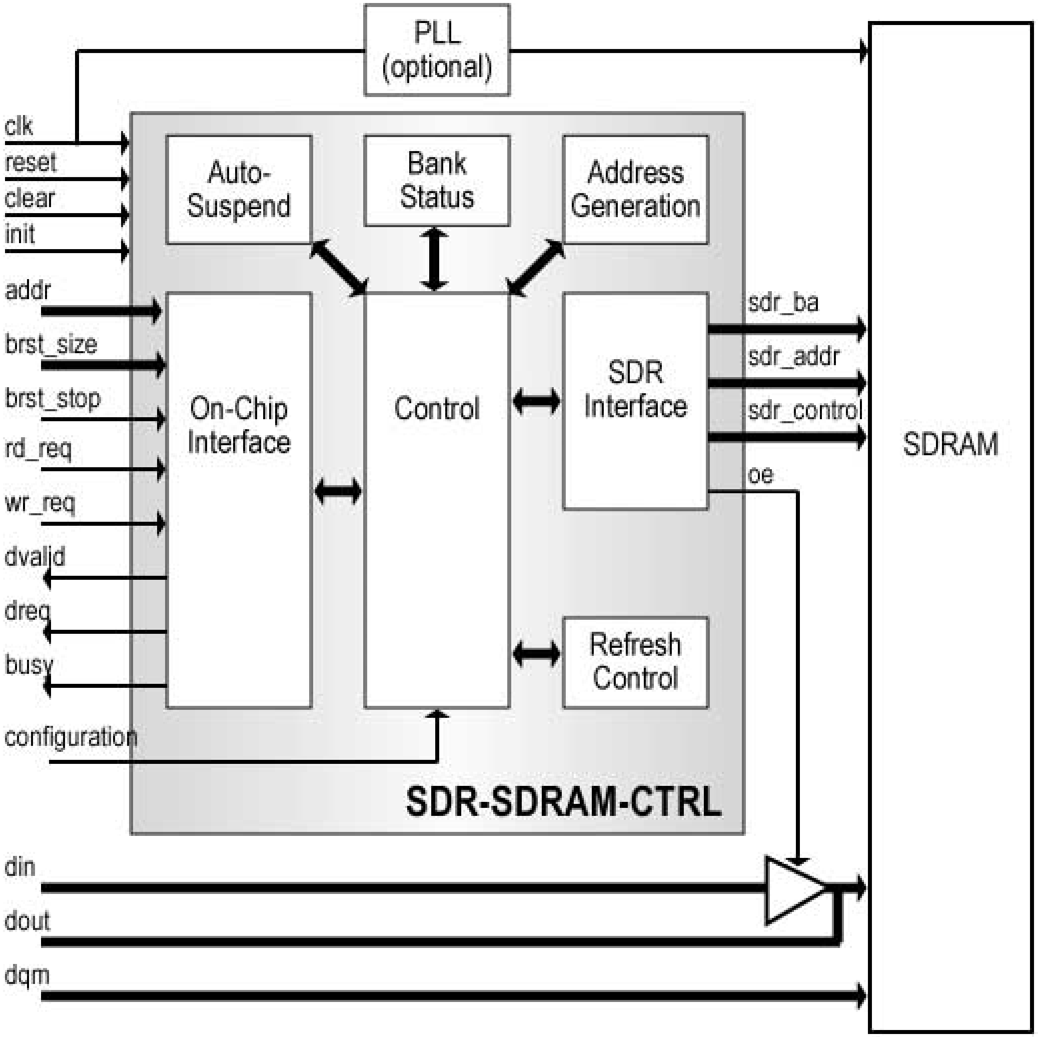
Standard	Clock rate (MHz)	M transfers per second	DRAM name	MB/sec /DIMM	DIMM name
DDR	133	266	DDR266	2128	PC2100
DDR	150	300	DDR300	2400	PC2400
DDR	200	400	DDR400	3200	PC3200
DDR2	266	533	DDR2-533	4264	PC4300
DDR2	333	667	DDR2-667	5336	PC5300
DDR2	400	800	DDR2-800	6400	PC6400
DDR3	533	1066	DDR3-1066	8528	PC8500
DDR3	666	1333	DDR3-1333	10,664	PC10700
DDR3	800	1600	DDR3-1600	12,800	PC12800
DDR4	1066–1600	2133–3200	DDR4-3200	17,056–25,600	PC25600

Figure 2.14 Clock rates, bandwidth, and names of DDR DRAMS and DIMMs in 2010. Note the numerical relation-

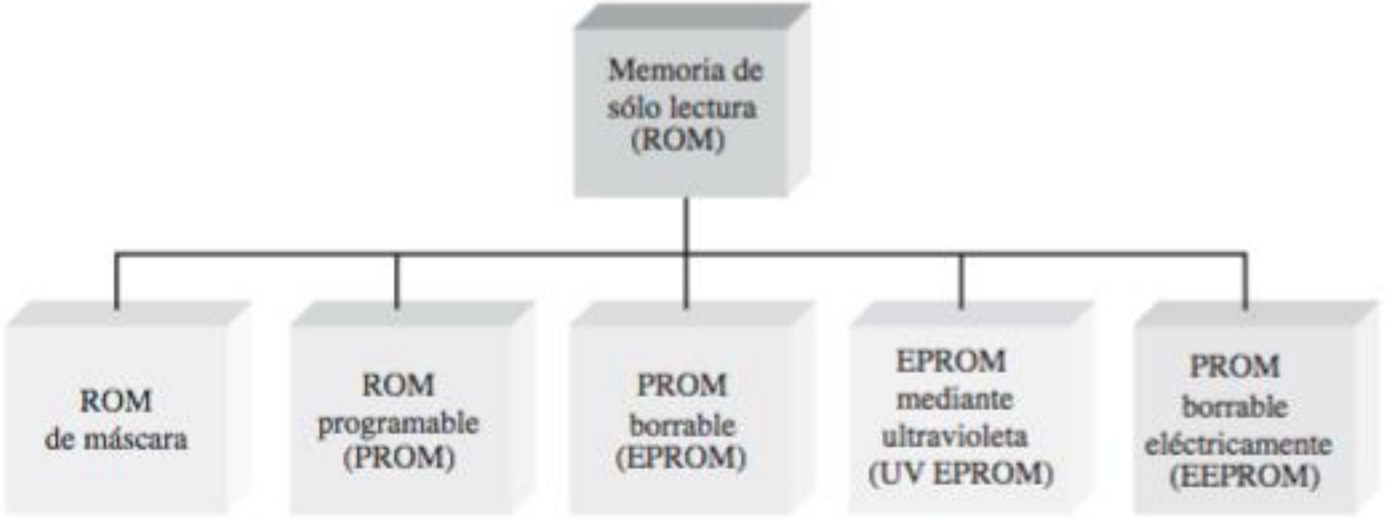
Hennesy & Patterson

DRAM memory controller

- Controller handles refresh and DRAM peculiarities
- It hides all this from the processor and offers a simple interface.
  - Processor **not** dependent on memory technology



ROM memories


Memoria de sólo lectura (ROM)
ROM de máscara
ROM programable (PROM)
PROM borrable (EPROM)
EPROM mediante ultravioleta (UV EPROM)
PROM borrable eléctricamente (EEPROM)

Fundamentos de Sistemas Digitales
Thomas L. Floyd

ARCOS Group

uc3m | Universidad Carlos III de Madrid

L5: Memory hierarchy (1)

Computer Structure

Bachelor in Computer Science and Engineering
Bachelor in Applied Mathematics and Computing
Dual Bachelor in Computer Science and Engineering and Business Administration