

---

# **Memòria Principal. Jerarquia de Memòria**

Memòria cache

Memòria Principal

Memòria Secundaria

Memòria virtual

# MEMÒRIES SEMICONDUCTORES

---

Classificació de les memòries. Es pot fer la classificació en funció de diferents paràmetres:

1.- per la permanència de les dades

2.- Tipus d'accés

3.- Tecnologia de programació

# MEMÒRIES SEMICONDUCTORES

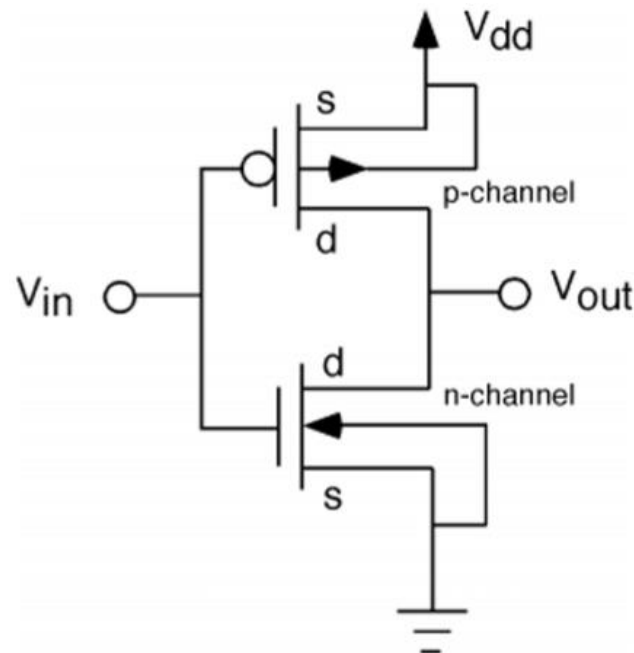
## Classificació

Permanència d'informació	Accés		Escriptura	
	Accés aleatori	Accés especial	Programació	Esborrat
Permanents	ROM PROM - OTP EPROM EEPROM Flash-EEPROM  NVRAM  FRAM	EEPROM sèrie	Màscara Elèctrica Elèctrica Elèctrica Elèctrica Elèctrica  Elèctrica  magnètica	No No Llum UV Elèctric Elèctric - blocs Elèctric  Elèctric (bat)  magnètica
Volàtils (quan $V_{cc} \Rightarrow 0$ )	RAM SRAM DRAM	LIFO FIFO CAM	Elèctrica Elèctrica Elèctrica	Elèctric Elèctric Elèctric

# Fonaments de Memòries

---

- La base en el disseny d'una memòria és el transistor MOS (Metall – Òxid – Semiconductor)
- Actua com un interruptor en funció de la tensió que tingui a l'entrada (porta)
- Normalment es fa servir tecnologia CMOS



# MEMÒRIES RAM (*Random Access Memory*)

---

Memòries d'accés aleatori: adreça – dada

S'utilitzen com a memòria principal o primària dels ordinadors,

Títol+adient seria memòries de lectura/escriptura **RWM** (*Read-Write Memory*)

Memòries de lectura-escriptura intensiva (a diferència d'EEPROM que estan pensades per un operació “Alguna escriptura/Moltes lectures”)

Es divideixen en dos tipus principals segons constitució de la cel·la d'emmagatzematge:

**Memòries RAM estàtiques (SRAM) => No necessita refresc**

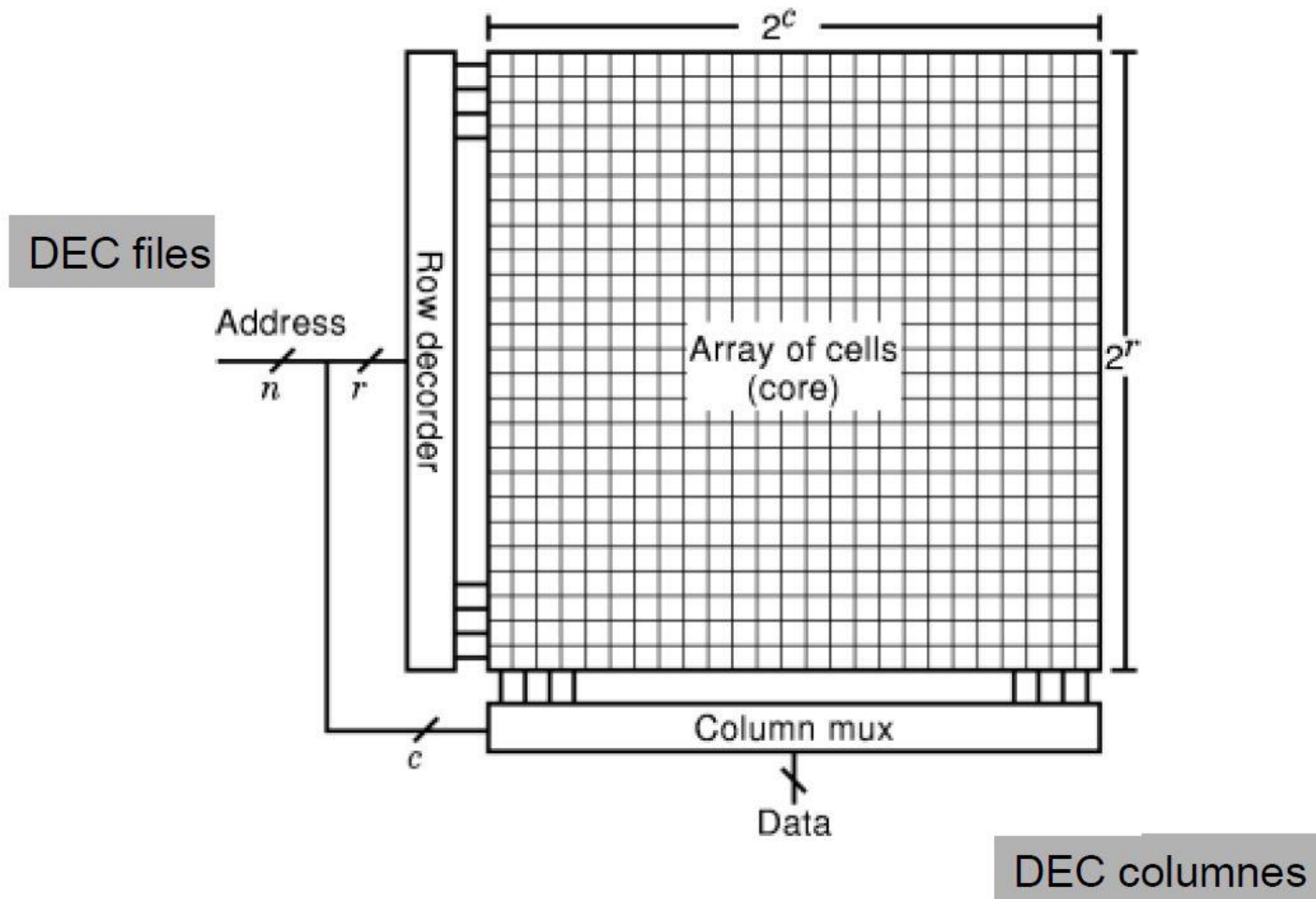
**Memòries RAM dinàmiques (DRAM) => Necessita refresc**

# Organització

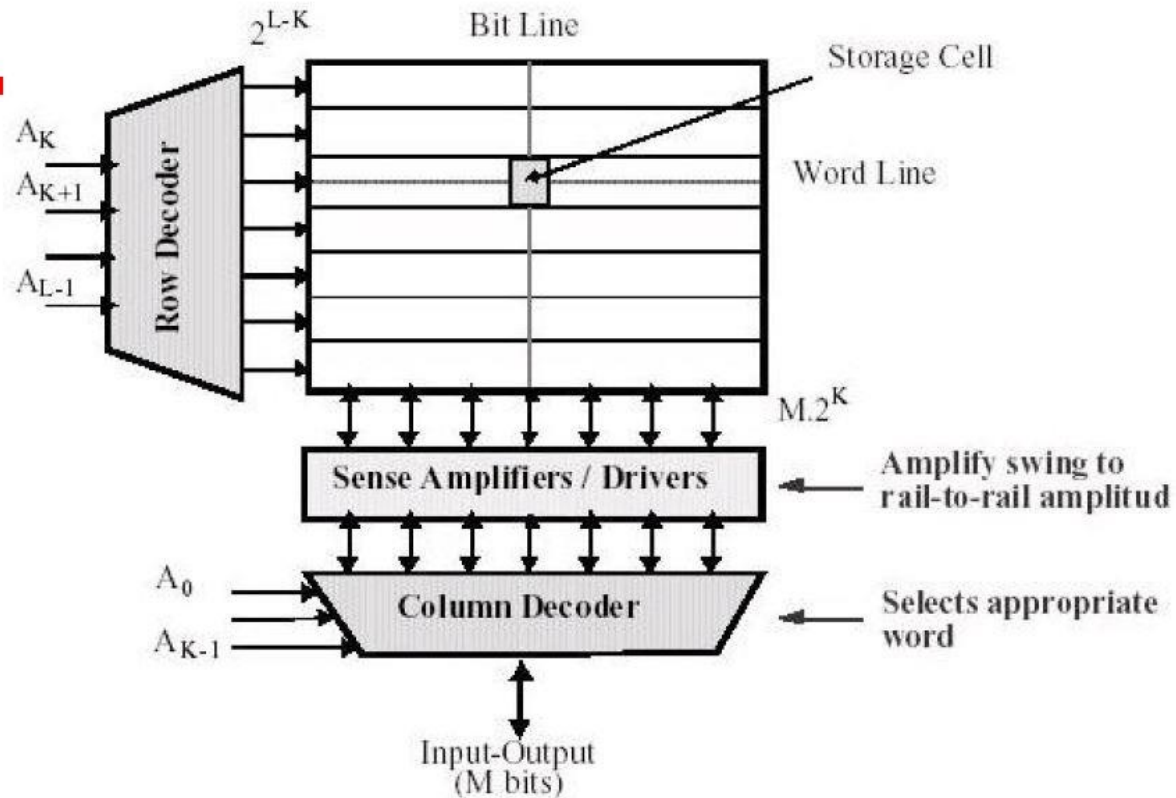
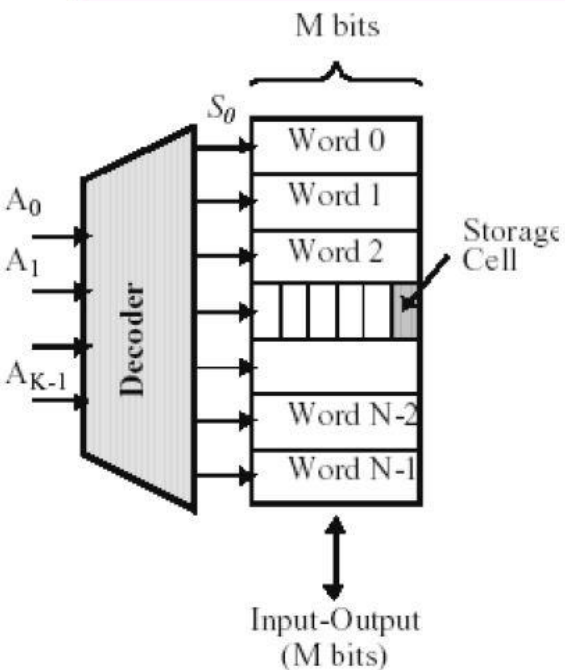
Organització files/columnes. (Relació d'aspecte +o- quadrada)

$r = \text{\#files}$

$c = \text{\#columnes}$



## Adreces: decodificació



Mida Total:  $C = M \cdot 2^K \cdot 2^{L-K} = M \cdot 2^L$

### Càlcul de Decodificadors files i columnes

Per.ex: una memòria de 256kb  $= 2^{18}$  (8kparaules x 32bits)

Files:  $2^{18} = 2^9$  files x  $2^9$  columnes **Dec. Files de 9** a 512 ( $L-K=9$ )

Columnes: Si treballem amb paraules (word) de 32b ( $2^9: 2^5=2^4$ )

Cal un **Dec Column. de 4** a 16 ( $2^4$ )





## Exemple de memòria SRAM:

**K6R4004V1D** (Samsung)  
1Mx4 Bit High Speed SRAM

Organitzada en  
paraules de 4 bits

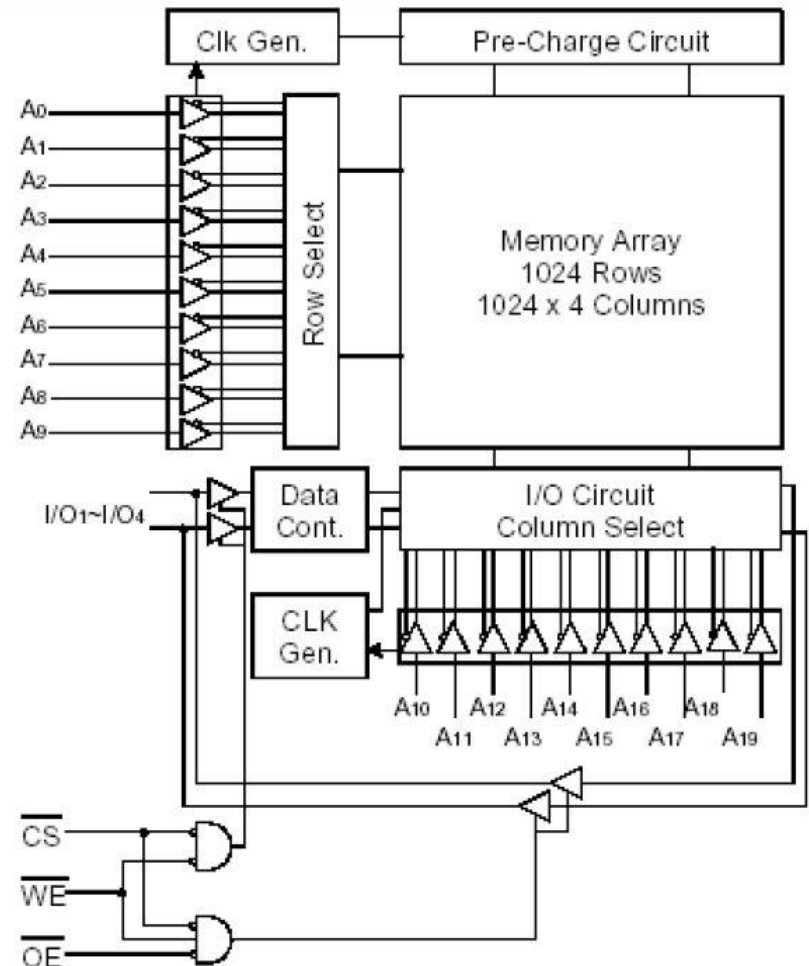
Cada posició = 4b (4 cel·les)

1Mb=2<sup>20</sup>

10 files; 10 columnes

Adr-Files: A[9..0]

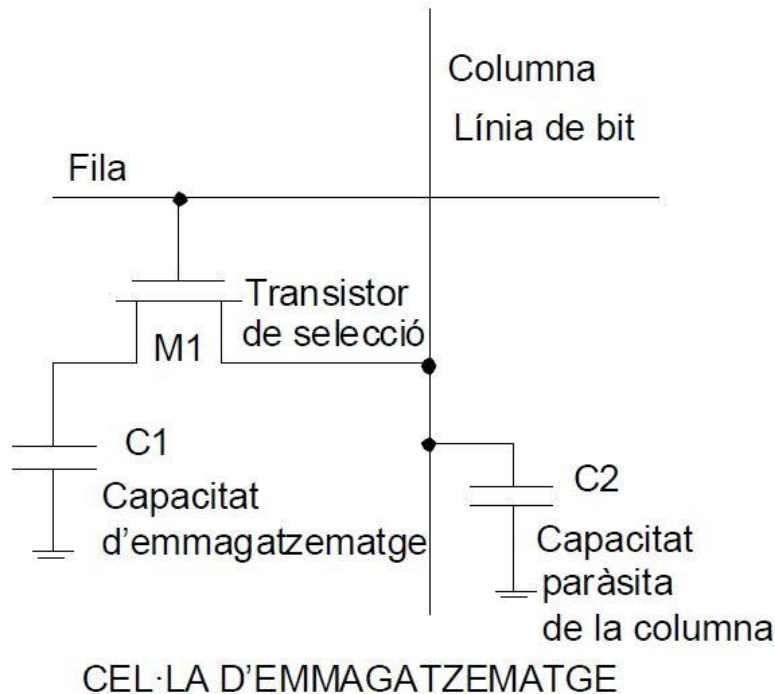
Adr-Files: A[19..10]



/CS	/WE	/OE	Mode	I/O Pin
H	X	X	No seleccionat (power down)	Alta Imp.
L	H	H	Inhibició de sortida	Alta Imp.
L	H	L	Lectura	D out
L	L	X	Espectura	D in

# Memòries DRAM

- **Dinàmiques** perquè necessiten de senyal de rellotge per mantenir les dades.
- La cèl·la de memòria està constituïda per un transistor i un condensador



Procés de **lectura** (**escriptura**) és transferència de càrrega **de C1 a C2** (**de C2 a C1**)

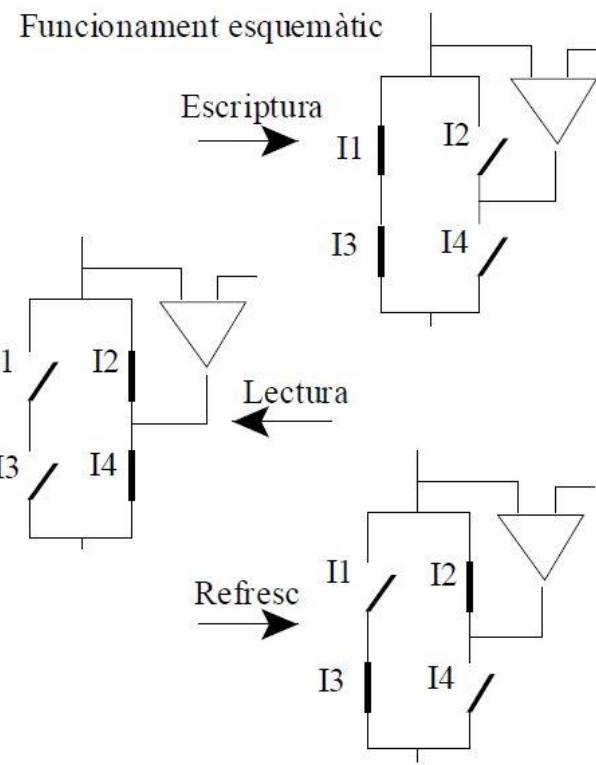
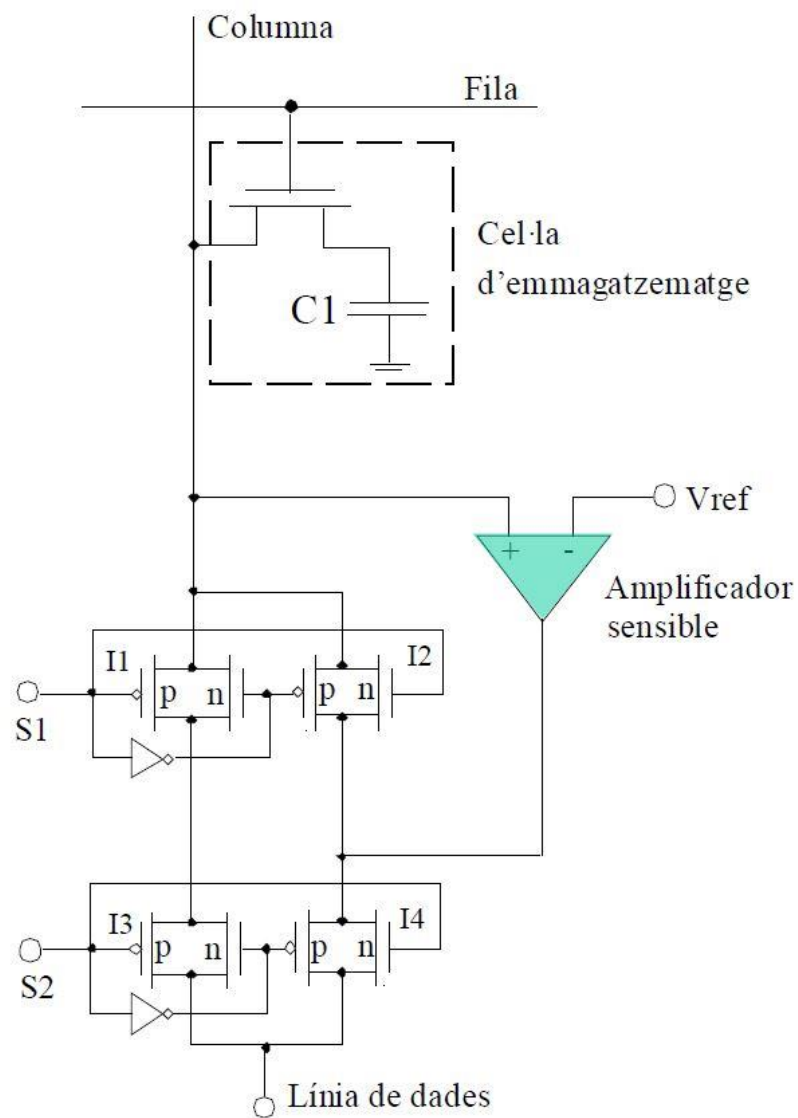
El canvi en el valor de la columna controlat per la relació entre les capacitats ( $C2 = 10 \cdot C1$ )

La transferència de càrrega és  $C2/(C1+C2)$

Excursió de tensió petita entre nivells L-H

A cada lectura (transferència de càrrega) es perd càrrega, cal reescriure.

La càrrega es perd amb el temps per fuites. Cal refrescar la dada.



Senyals de control d'operacions

	Escriptura	Lectura	Refresc
S1	0	1	1
S2	0	1	0

# SRAM vs. DRAM

---

<b>SRAM</b>	Manté dades mentre hi hagi Vcc Cel·la gran (6 trans.) Baixa capacitat Ràpida Cost elevat	Mem. Dades baixa Capac. Mem. Caches
<b>DRAM</b>	Necessita refresc periòdic (Controlador) Cel·la petita (1-3 trans) Alta Capacitat (densitat) Lenta Cost baix	Mem. Dades alta Capac. Mem. Principal Computador

# El disc dur

---

1r HD introduït per IBM al 1957 (a l'ordinador IBM350)  
anomenat RAMAC

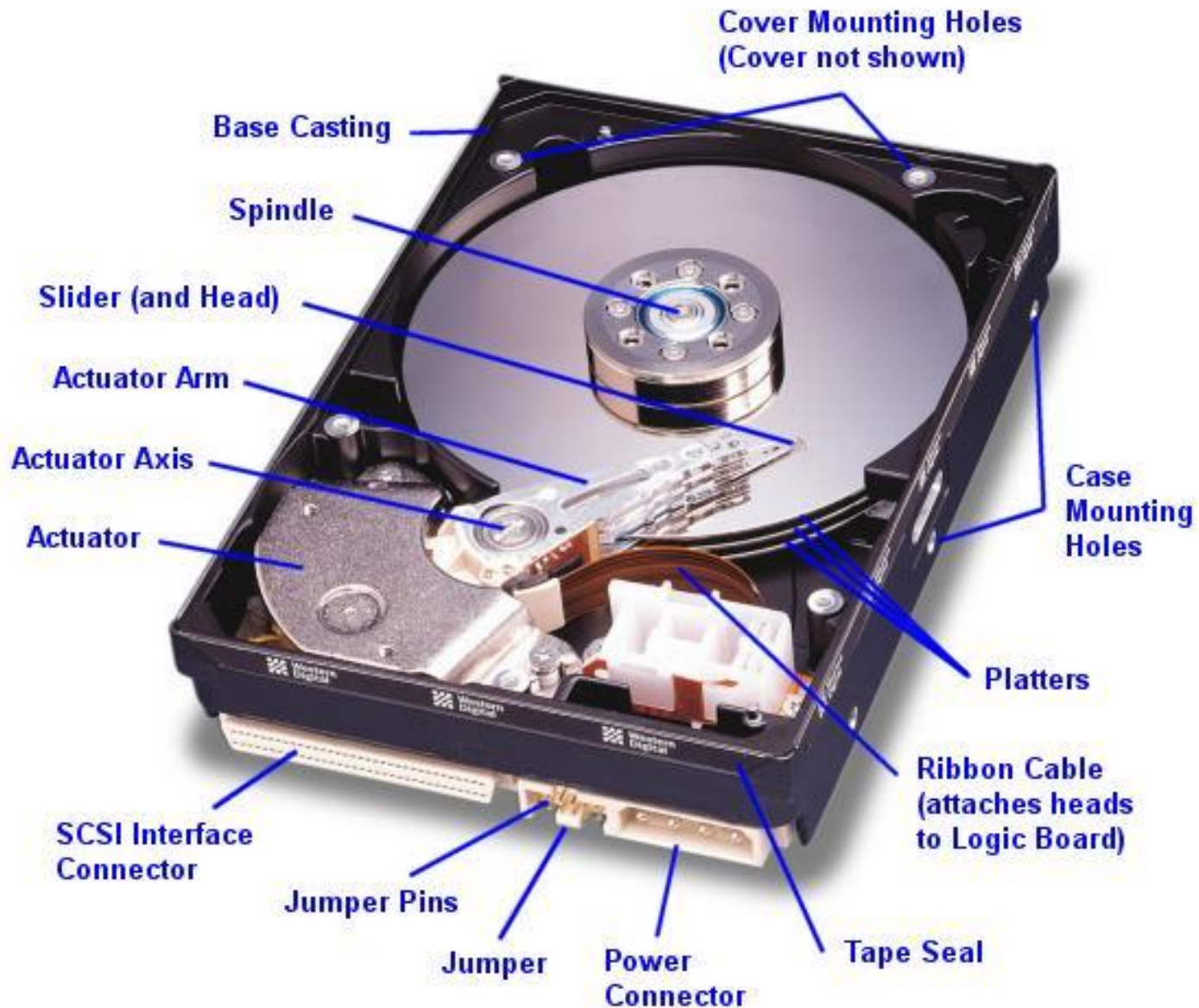
From Computer Desktop Encyclopedia  
Reproduced with permission.  
© 1996 International Business Machines Corporation  
Unauthorized use not permitted.

1. Sistema de gravació magnètic
2. Molt més lent que les memòries semiconductores
3. Molt alta capacitat d'enmagatzematge



# Introducció al disc dur

---





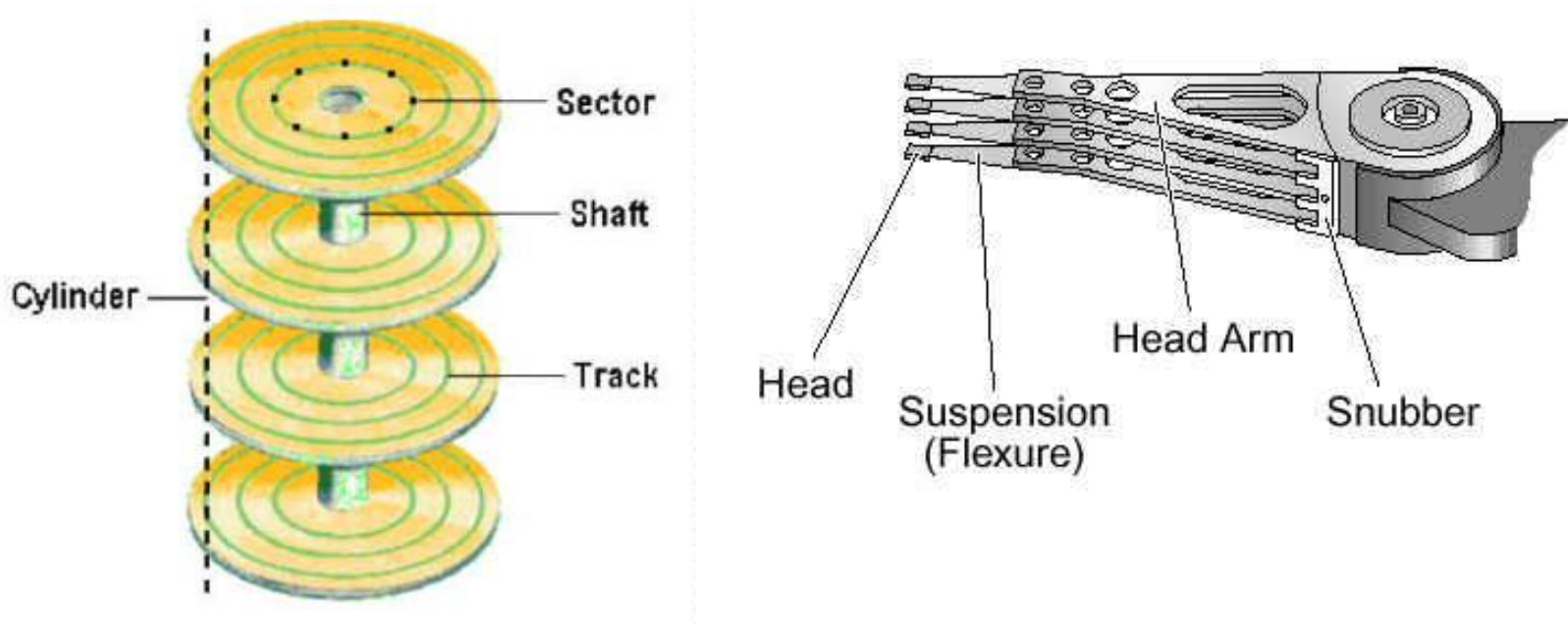
# Evolució del disc dur

---

- 2007 – First 1 terabyte<sup>[27]</sup> hard drive<sup>[28]</sup> (Hitachi GST)
- 2008 – First 1.5 terabyte<sup>[27]</sup> hard drive<sup>[29]</sup> (Seagate)
- 2009 – First 2.0 terabyte hard drive<sup>[30]</sup> (Western Digital)
- 2010 – First 3.0 terabyte hard drive<sup>[31][32]</sup> (Seagate, Western Digital)
- 2010 – First hard drive manufactured by using the [Advanced Format](#) of 4,096 bytes a block ("4K") instead of 512 bytes a block<sup>[33]</sup>
- 2011 – First 4.0 terabyte hard drive<sup>[34]</sup> (Seagate)
- 2011 – [Floods hit many hard drive factories](#). Predictions of a worldwide shortage of hard disk drives cause prices to double.<sup>[35][36][37]</sup>
- 2012 – Western Digital announces the first 2.5-inch, 5 mm thick drive, and the first 2.5-inch, 7 mm thick drive with two platters<sup>[38]</sup> (Western Digital)
- 2012 – HGST announces helium-filled hard disk drives, promising cooler operation and the ability to increase the maximum number of platters from five to seven
- 2012 – TDK demonstrates 2 TB on a single 3.5-inch platter<sup>[40]</sup>
- 2012 – [Hitachi Global Storage Technologies](#) and Western Digital merge to one hard drive manufacturer. To ensure actual competition on the hard drive market a Commission requires that Western Digital and Hitachi GST give assets and intellectual property rights to Toshiba.<sup>[41]</sup> This allows Toshiba to re-enter the 3.5" desktop to 3 TB.<sup>[42]</sup> Prior to this, Toshiba had only manufactured 2.5" laptop HDDs for many years.
- 2013 – Seagate announces that it will ship hard disk drives with capacities up to 5 TB using [shingled magnetic recording](#) (SMR), a method where tracks are written closer together, being smaller, can still read the overlapped tracks.<sup>[43]</sup>
- 2013 – HGST announces a helium-filled 6 TB hard disk drive for enterprise applications<sup>[44]</sup>
- 2013 – Western Digital demonstrates [heat assisted magnetic recording](#) (HAMR) technology<sup>[45][46][47][48]</sup>
- 2014 – Seagate introduces 6 TB hard drives that do not use helium, in turn increasing their power consumption and lowering their overall cost<sup>[49]</sup>
- 2014 – Seagate ships world's first 8 TB hard drives<sup>[50]</sup>
- 2015 – In June HGST ships Ultrastar Archive Ha<sup>10</sup> SMR HDD, the world's first 10 TB HDD<sup>[51]</sup> followed in December by a conventional PMR HDD<sup>[52]</sup>
- 2017 – 12 TB Helium-based HDD available from Western Digital.<sup>[53]</sup>
- 2017 – 14 TB Helium-filled PMR HDD announced by Toshiba, with availability being promised for 2018 Q1 or Q2.<sup>[54]</sup>

# Característiques del disc dur

---



Cluster: És la unitat mínima de informació utilitzada pel S.O. La seva grandaria oscil.la entre 1 i 64 sectors (32KB)

Un fitxer és una seqüència de clusters. Ocupa un n° sencer de clusters encara que el cluster estigui parcialment ple.



# El disc dur basat en semiconductors

---

Disc dur SSD (acrònim de Solid-State Drive) és un tipus de dispositiu d'emmagatzematge de dades que fa servir memòria del tipus FLASH (no volàtil) per guardar les dades.

Les memòries FLASH fonamenten el seu funcionament en les memòries EEPROM

Què és una memòria EEPROM?

(Electrical Erasable Programmable Read Only Memory) és una memòria que pot ser programada, esborrada i tornada a re-programar “tantes vegades” com sigui necessari.

## AVANTATGES

- Més ràpides que el disc dur magnètic
- Més silencioses
- Més lleugeres

## DESAVANTATGES

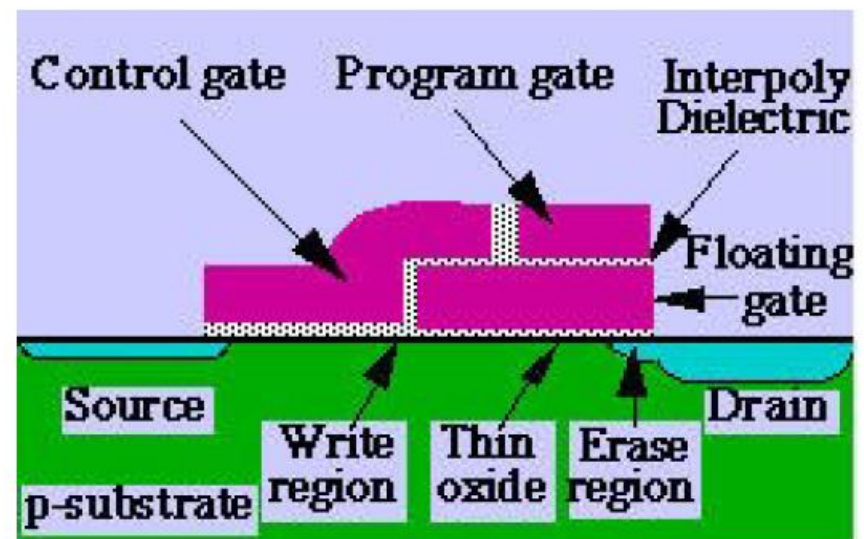
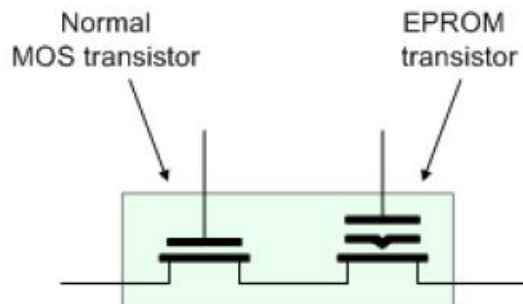
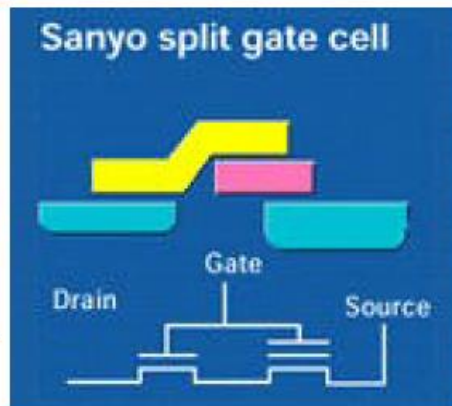
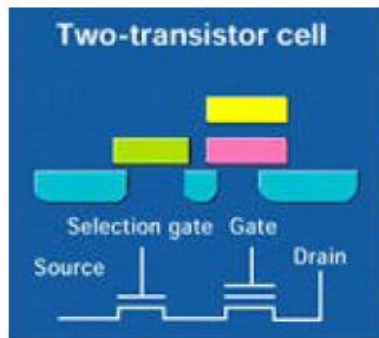
- Més cares
- Menys capacitat

# El disc dur basat en semiconductors

Basat en una estructura similar a la representada. Dos transistors per bit de memòria.

El pin del “control gate” indica si fem una lectura o una escriptura

El pin de “program gate” serveix per gravar un “1” o un “0” o per seleccionar el bit per fer una lectura



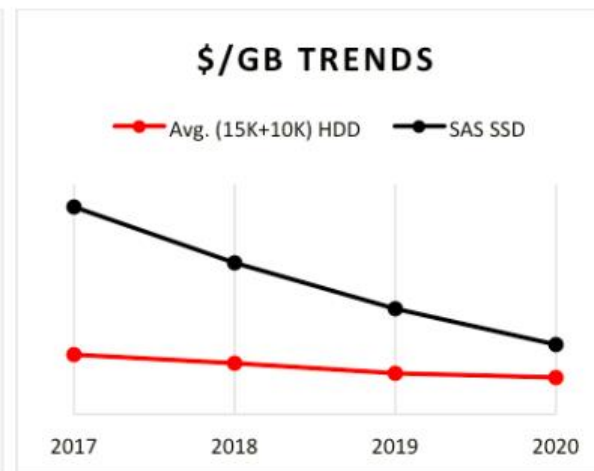
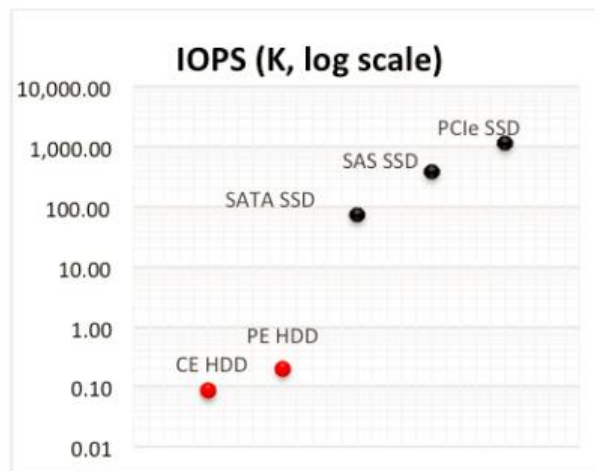
# El disc dur. SSD vs HDD

---

Attribute	SSD (Solid State Drive)	HDD (Hard Disk Drive)
<b>Power Draw / Battery Life</b>	Less power draw, averages 2 – 3 watts, resulting in 30+ minute battery boost ✓	More power draw, averages 6 – 7 watts and therefore uses more battery
<b>Cost</b>	Expensive, roughly \$0.20 per gigabyte (based on buying a 1TB drive)	Only around \$0.03 per gigabyte, very cheap (buying a 4TB model) ✓
<b>Capacity</b>	Typically not larger than 1TB for notebook size drives; 4TB max for desktops	Typically around 500GB and 2TB maximum for notebook size drives; 10TB max for desktops ✓
<b>Operating System Boot Time</b>	Around 10-13 seconds average bootup time ✓	Around 30-40 seconds average bootup time
<b>Noise</b>	There are no moving parts and as such no sound ✓	Audible clicks and spinning can be heard
<b>Vibration</b>	No vibration as there are no moving parts ✓	The spinning of the platters can sometimes result in vibration
<b>Heat Produced</b>	Lower power draw and no moving parts so little heat is produced ✓	HDD doesn't produce much heat, but it will have a measurable amount more heat than an SSD due to moving parts and higher power draw

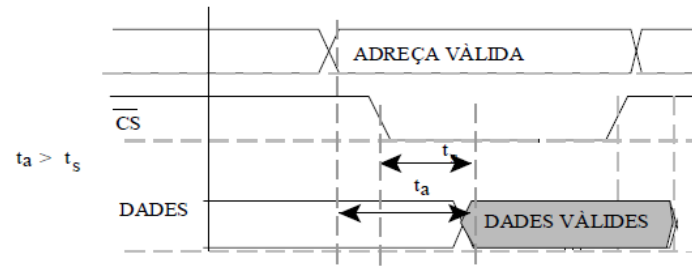
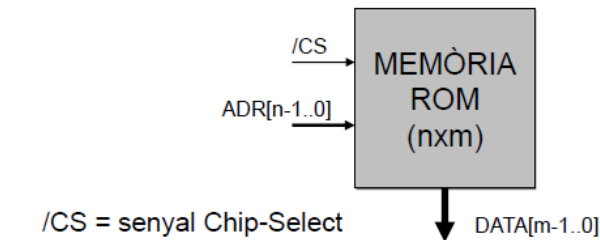
# El disc dur. SSD vs HDD

Attribute	SSD (Solid State Drive)	HDD (Hard Disk Drive)
<b>Failure Rate</b>	Mean time between failure rate of 2.0 million hours ✓	Mean time between failure rate of 1.5 million hours
<b>File Copy / Write Speed</b>	Generally above 200 MB/s ✓ and up to 550 MB/s for cutting edge drives	The range can be anywhere from 50 – 120MB / s
<b>Encryption</b>	Full Disk Encryption (FDE) ✓ Supported on some models	Full Disk Encryption (FDE) ✓ Supported on some models
<b>File Opening Speed</b>	Up to 30% faster than HDD ✓	Slower than SSD
<b>Magnetism Affected?</b>	An SSD is safe from any effects of magnetism ✓	Magnets can erase data



# JERARQUIA I MEMÒRIA: TERMINOLOGIA I CONCEPTES

Temps d'accés o latència: Es defineix el temps d'accés a una memòria com l'interval de temps que passa entre que l'adreça surt pel bus de adreces i la dada es col·loca en el bus de dades



# JERARQUIA I MEMÒRIA: TERMINOLOGIA I CONCEPTES

---

## **Nivell jerarquic superior:**

El més proper al procesador. Conté la memòria més ràpida i petita

## **Nivell jerarquic inferior:**

El més llunyà del procesador. És el més gran i el més lent. Tot el que conté les memòries de nivell superior es pot trobar en la de nivell inferior.

## **Encert (Hit)**

Es produeix quan la informació requerida pel uprocesador es troba al nivell superior

## **Fallada (Miss)**

Quan la informació requerida pel procesador no es troba en el nivell superior. S'ha d'accedir al nivell inferior i traslladar el bloc on es troba al nivell superior.

# JERARQUIA I MEMÒRIA: TERMINOLOGIA I CONCEPTES

---

## **Tassa d'encerts ( $R_e$ )**

Fracció d'accessos a memòria que s'ha trobat al nivell superior

## **Tassa de fallades ( $R_f$ )**

Fracció d'accessos a memòria que no s'ha trobat al nivell superior.  $R_f = 1 - R_e$

## **Temps d'encerts ( $t_e$ )**

És el temps necessari per realitzar un accés a un nivell superior de la jerarquia de memòria. També inclou el temps necessari per determinar si s'ha produït un encert o una fallada.

## **Penalització de fallades**

Temps requerit quan es produeix una fallada. Serà el temps necessari per buscar el bloc que conté la informació demanada en el nivell inferior, portar-ho al nivell superior i entregar-ho al procesador.

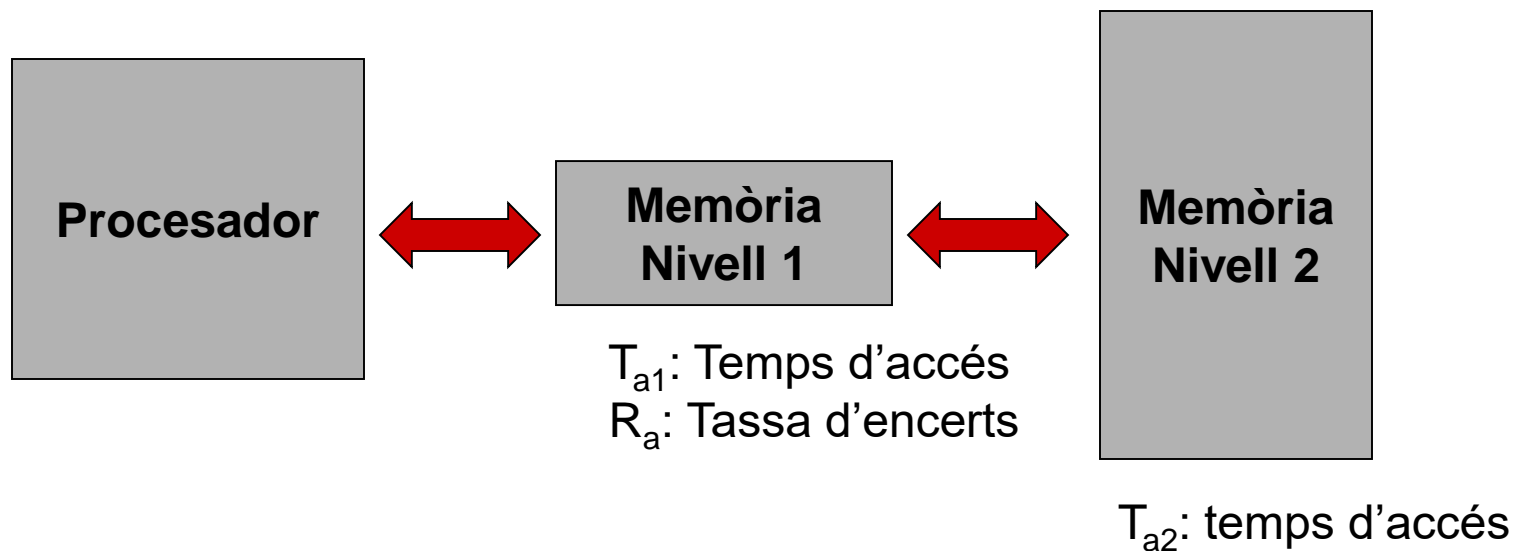
# JERARQUIA I MEMÒRIA: TERMINOLOGIA I CONCEPTES

---

## Temps d'accés mig ( $t_a^{\text{mean}}$ )

És la relació entre el temps d'encerts i el percentatge de la tasa d'encerts.

Si per exemple tenim 2 nivells de jerarquia



$$T_a^{\text{mean}} = T_{a1} \cdot R_a + \left[ T_{a1} + T_{a2} \right] \cdot \left[ 1 - R_a \right]$$



## JERARQUIA I MEMÒRIA: TERMINOLOGIA I CONCEPTES

---

- Exemple

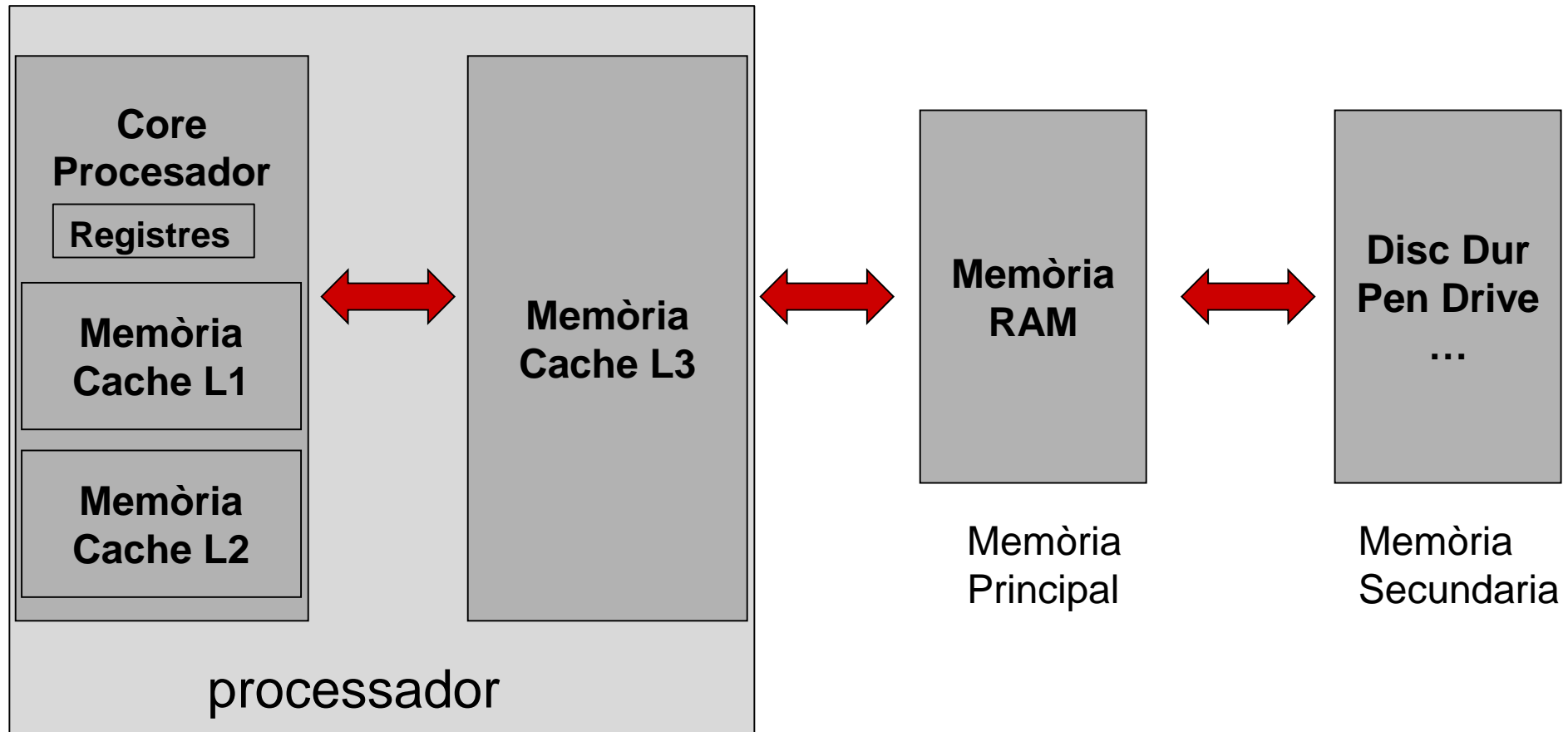
Tenim un subsistema de memòria organitzat en dos nivells de jerarquia. En el nivell superior tenim una cache de 16kB i  $T_{a1} = 20$  ns. Al nivell inferior tenim una memòria principal de 8MB i  $T_{a2} = 100$  ns. Si la taxa d'encerts és del 90% quin és el temps d'accés promig a una paraula del subsistema de memòria??

$$T_a = P_a \times T_{a1} + (1 - P_a)(T_{a1} + T_{a2})$$

$$T_a = 0.9 \times 20 + 0.1(20 + 100) = 30 \text{ ns}$$

# Jerarquia de Memòries

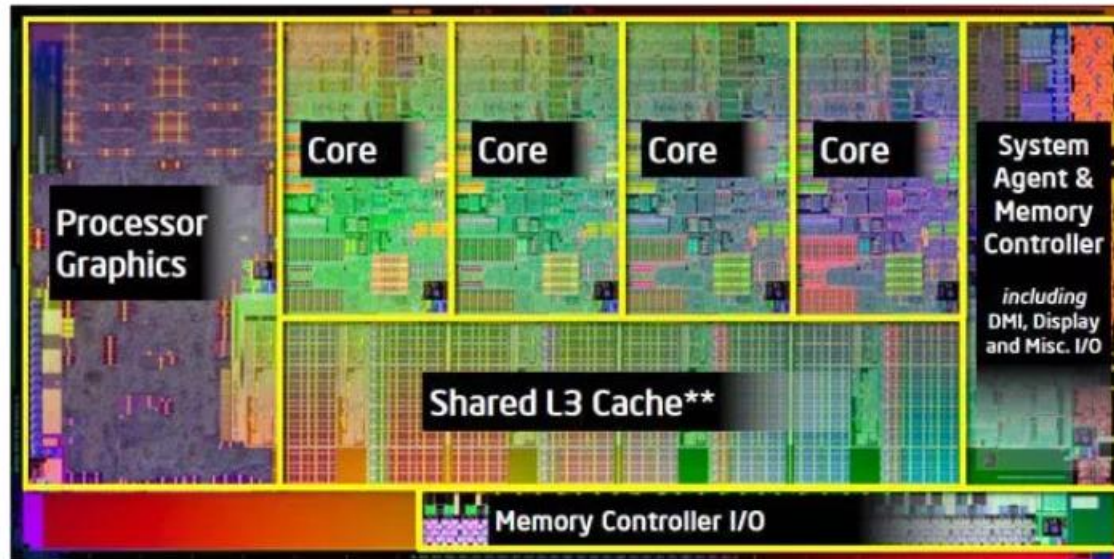
---



# Jerarquia de Memòries

---

## Die procesador Intel Core i7-4790K



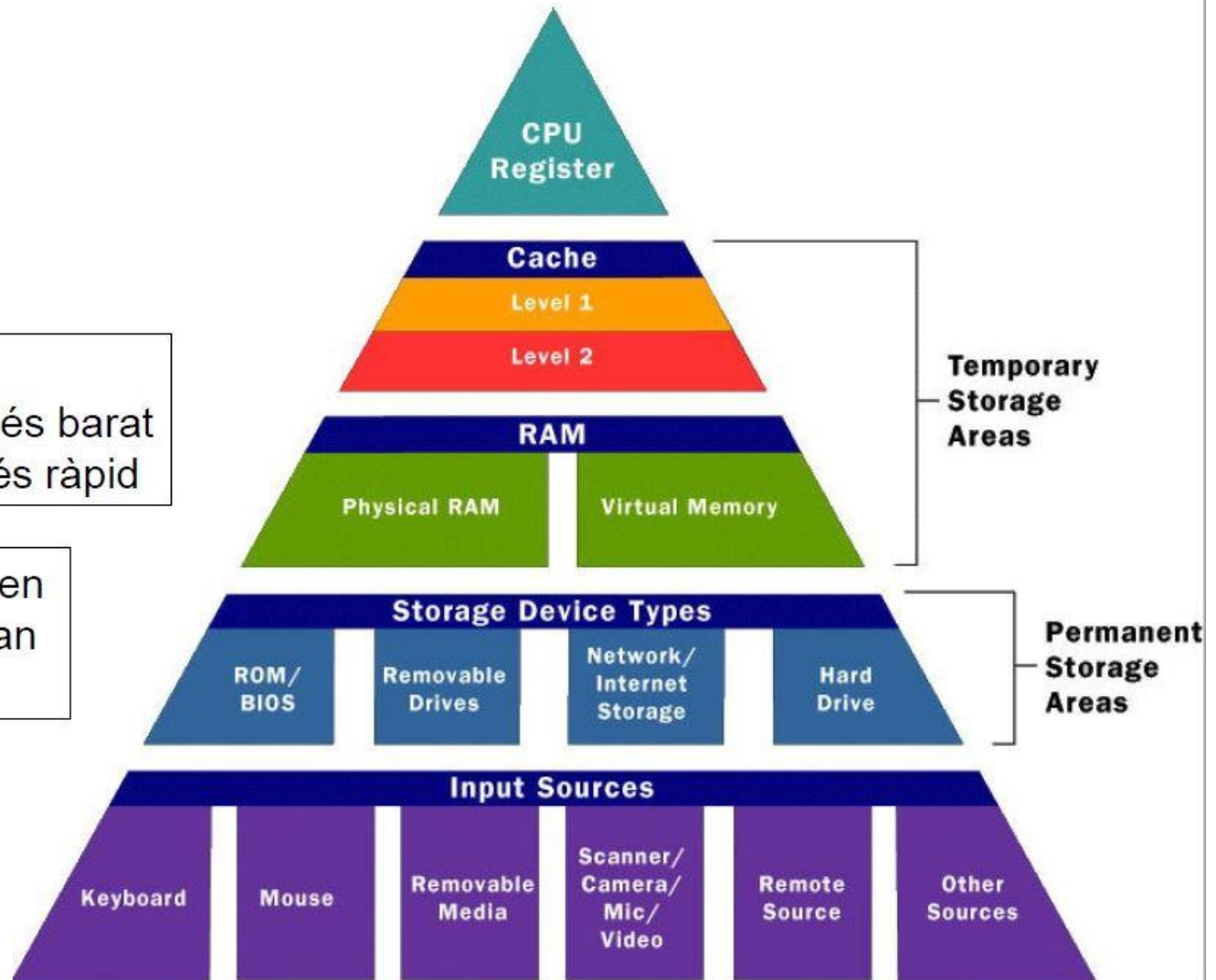
# JERARQUIA I MEMÒRIA: TERMINOLOGIA I CONCEPTES

Cada nivell és  
+petit  
+ràpid  
+car  
que l'inferior

## Objectiu:

Cost proper al nivell més barat  
Velocitat propera al més ràpid

Les dades guardades en  
un nivell també ho estan  
en el nivell inferior



## MEMÒRIA CACHE

- Nivell superior de les memòries
- Recomanable en aquelles aplicacions que precisen d'un accés molt ràpid de la informació
- Capacitat petita (16-512KB-8MB). Del tipus SRAM
- Tecnologia de fabricació bipolar
- Mode d'accés associatiu (camp d'etiqueta i camp de dades)
- Quan la paraula donada no coincideix amb el valor de cap etiqueta es produeix **Miss** ➡ **La caché no disposa de la dada demanada.** S'ha de buscar a la memòria principal

# MEMÒRIA CACHE

---

La memòria CACHE es troba actualment integrada en la CPU.  
Necessitem trobar resposta a dues preguntes:

- 1.- La dada que busquem està a la cache?
- 2.- Si està, com la trobem?

Tenim tres tipus d'estructuració de memòries cache:

- a) Mapeig directe
- b) Totalment associatives
- c) Associatives per conjunts

El mètode més simple d'entendre és el primer. Memòries cache amb mapeig directe

# MEMÒRIA CACHE

---

La memoria cache té tres parts importants:

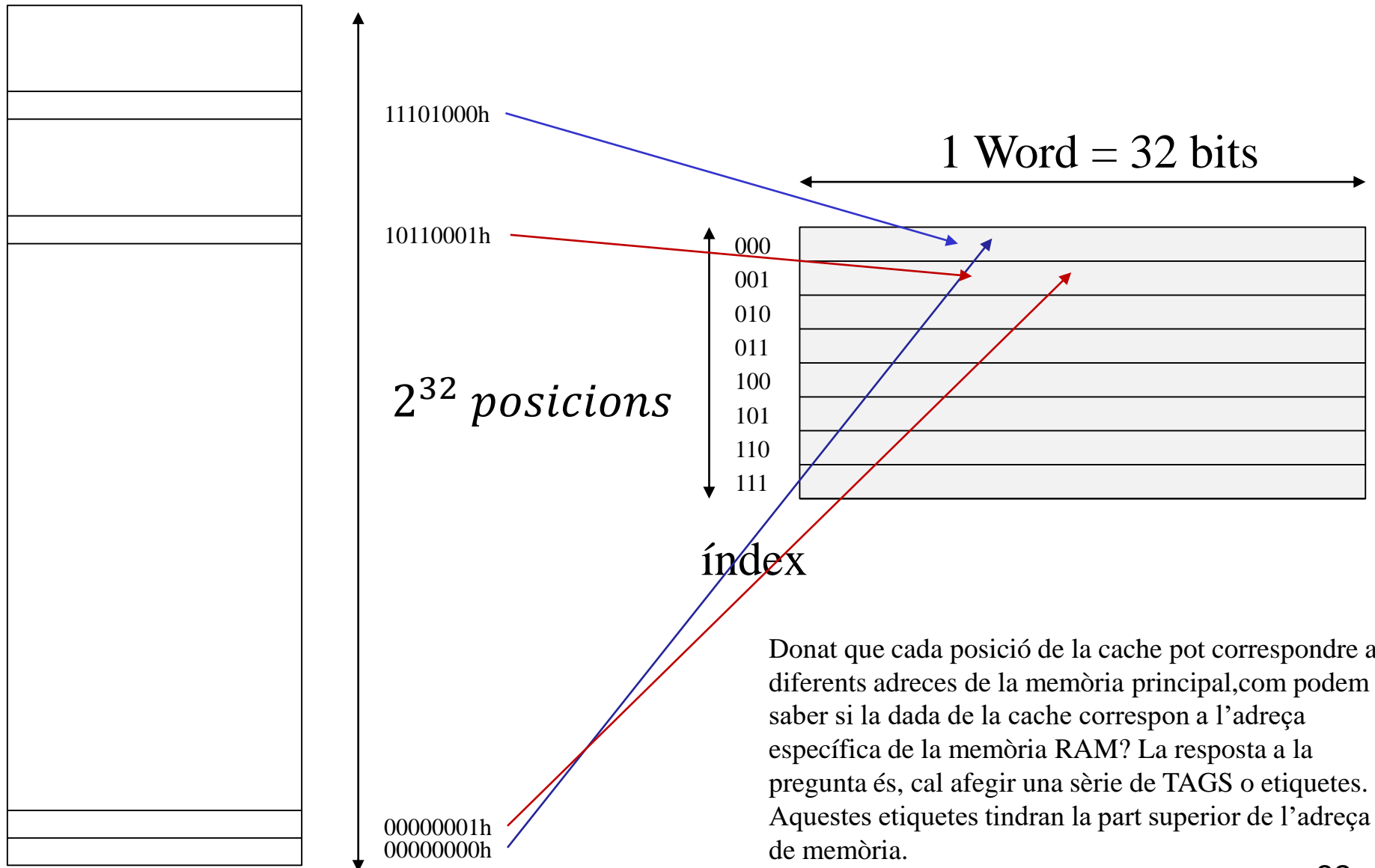
- a) Etiqueta
- b) Bloc de memoria. Un bloc de memoria pot estat constituït per N words
- c) Índex

Podem trobar un bloc en mapejat directe fent:

$(\text{block address}) \bmod (\text{Number of blocks in cache})$

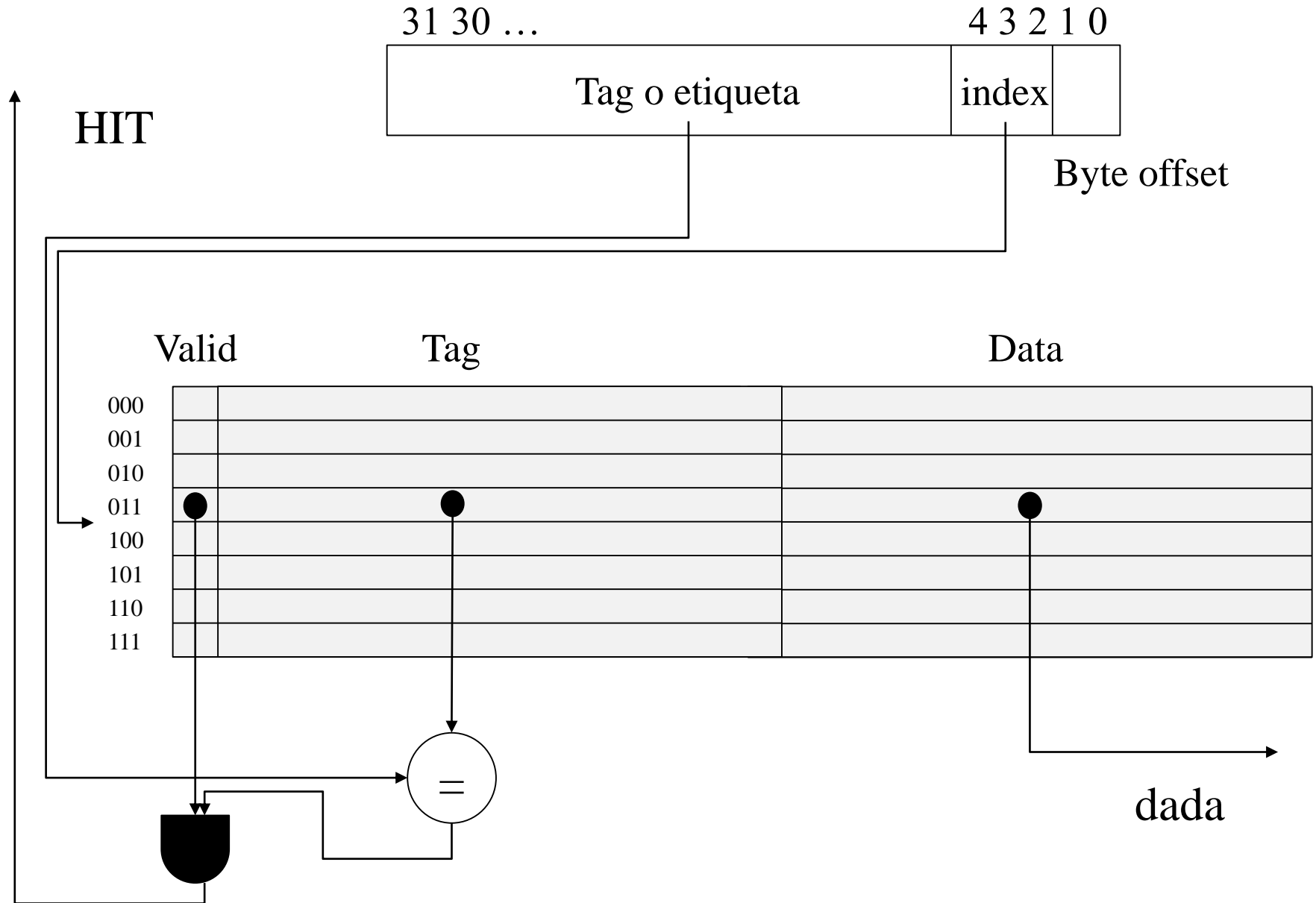
Un exemple senzill seria el següent: Tenim una cache que té 8 blocs de 1 word cada bloc. El bus d'adreces és de 32 bits. Com serà la nostra cache?

# MEMÒRIA CACHE





# MEMÒRIA CACHE



## EFICIÈNCIA DE LA CACHE

És la relació entre el temps d'accés i el temps d'accés mig.

Aquesta eficiència dependrà d'uns algorismes que s'utilitzen per carregar-la amb la informació necessària per la UCP.

Tenim dos paràmetres que influeixen en la eficiència:

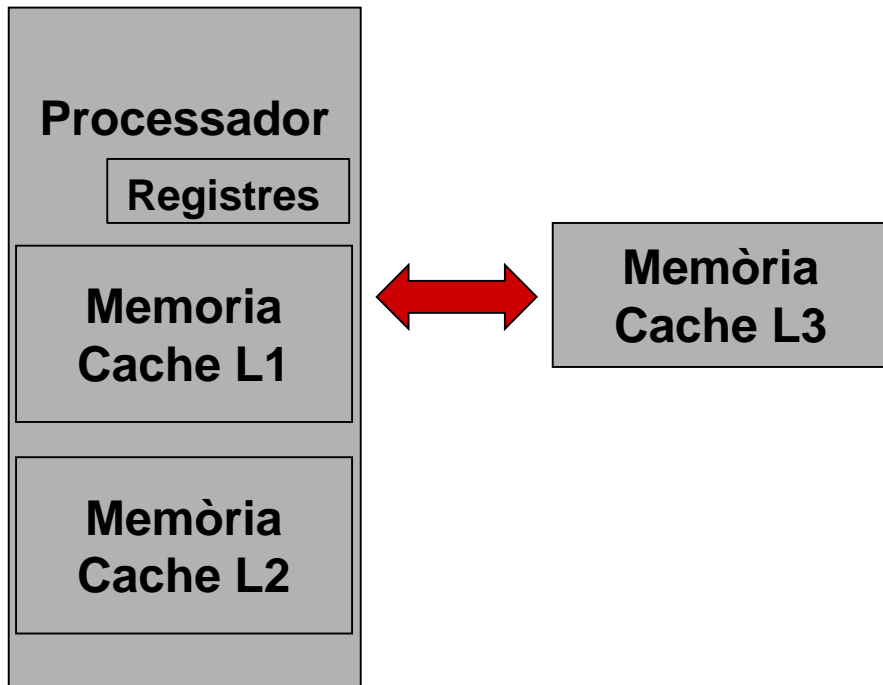
La probabilitat de presència  $(h) = \text{N}^{\circ} \text{ presències en caché} / \text{N}^{\circ} \text{ total accessos caché}$

La probabilitat de absència  $(1-h) = \text{N}^{\circ} \text{ absències caché} / \text{N}^{\circ} \text{ total accessos caché}$

Un altre paràmetre que sol donar-se és el factor de velocitat, que és la relació entre el temps d'accés de la memòria principal i el de la caché

# JERARQUIA I MEMÒRIA: TERMINOLOGIA I CONCEPTES

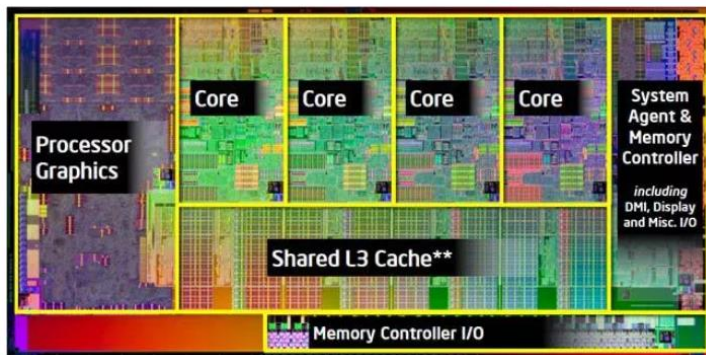
---



**Caché L1.** Integrada en la CPU. Es fa servir per accedir a dades importants i de ús freqüent. Totes les instruccions es busquen primer aquí

**Caché L2.** Utilitzada per guardar informació utilitzada recentment. També coneguda com caché secundària, està dissenyada per reduir el temps d'accés a les dades utilitzades prèviament. Es fa servir també per fer pipeline temporal d'instruccions.

**Caché L3.** Abans la Memòria L3 es trobava integrada en la placa base. Es fa servir per alimentar la memòria caché L2



## ESTRUCTURA DE LA MEMÒRIA

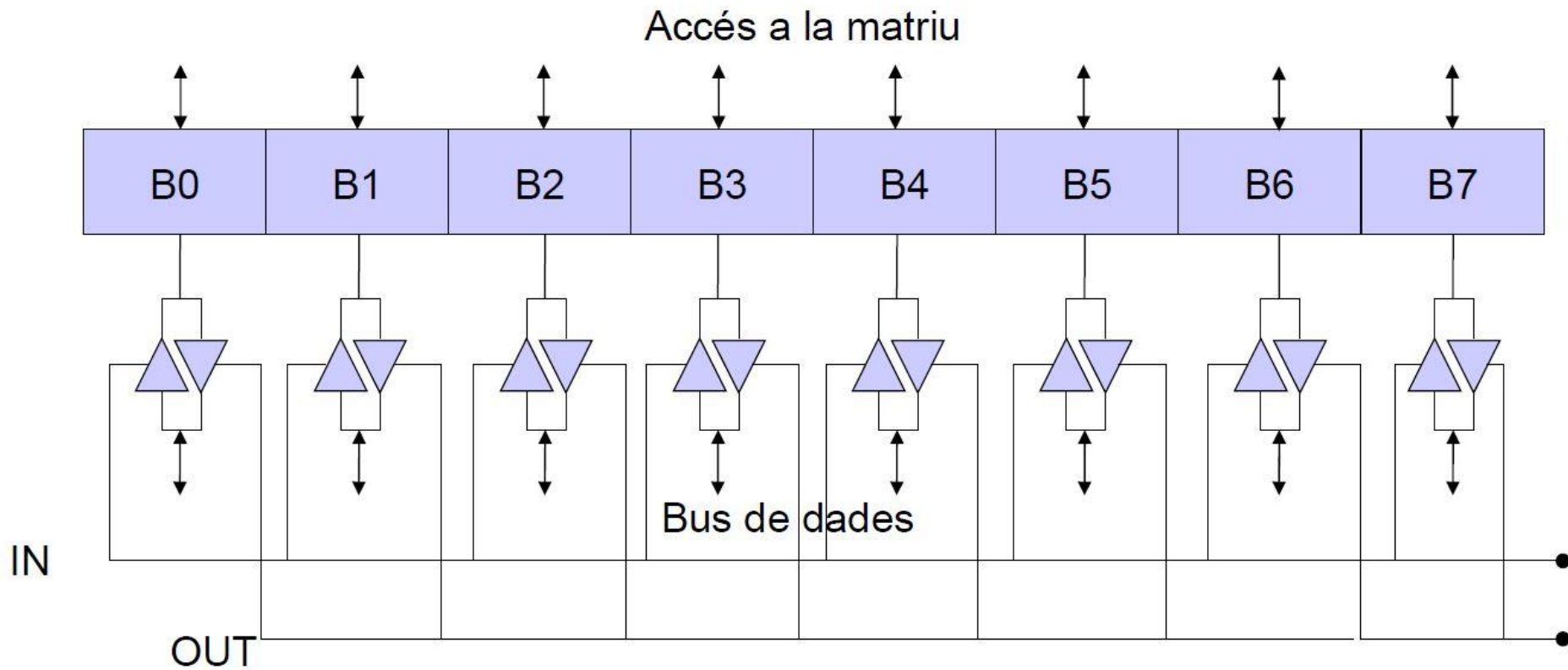
---

- Organització interna de la memòria
  - Matriu de memòria
  - Un o dos decodificadors per seleccionar les paraules guardades a la matriu de cel·les
  - Un registre d'entrada/sortida format per un n° de biestables igual a la longitud de paraula o dada guardat a la posició de memòria (Opcional. Depen de la memòria). Els buffers tri-state que té integrats determinen si la paraula és de lectura o escriptura
  - Lògica de control. A partir dels senyals externs del bus (CS, R/W, etc) genera els senyals de govern intern

Entrades		Sortides	
CS	\R/W	IN	OUT
0	X	0	0
1	0	0	1
1	1	1	0

# ESTRUCTURA DE LA MEMÒRIA

---



# ESTRUCTURA DE LA MEMÒRIA

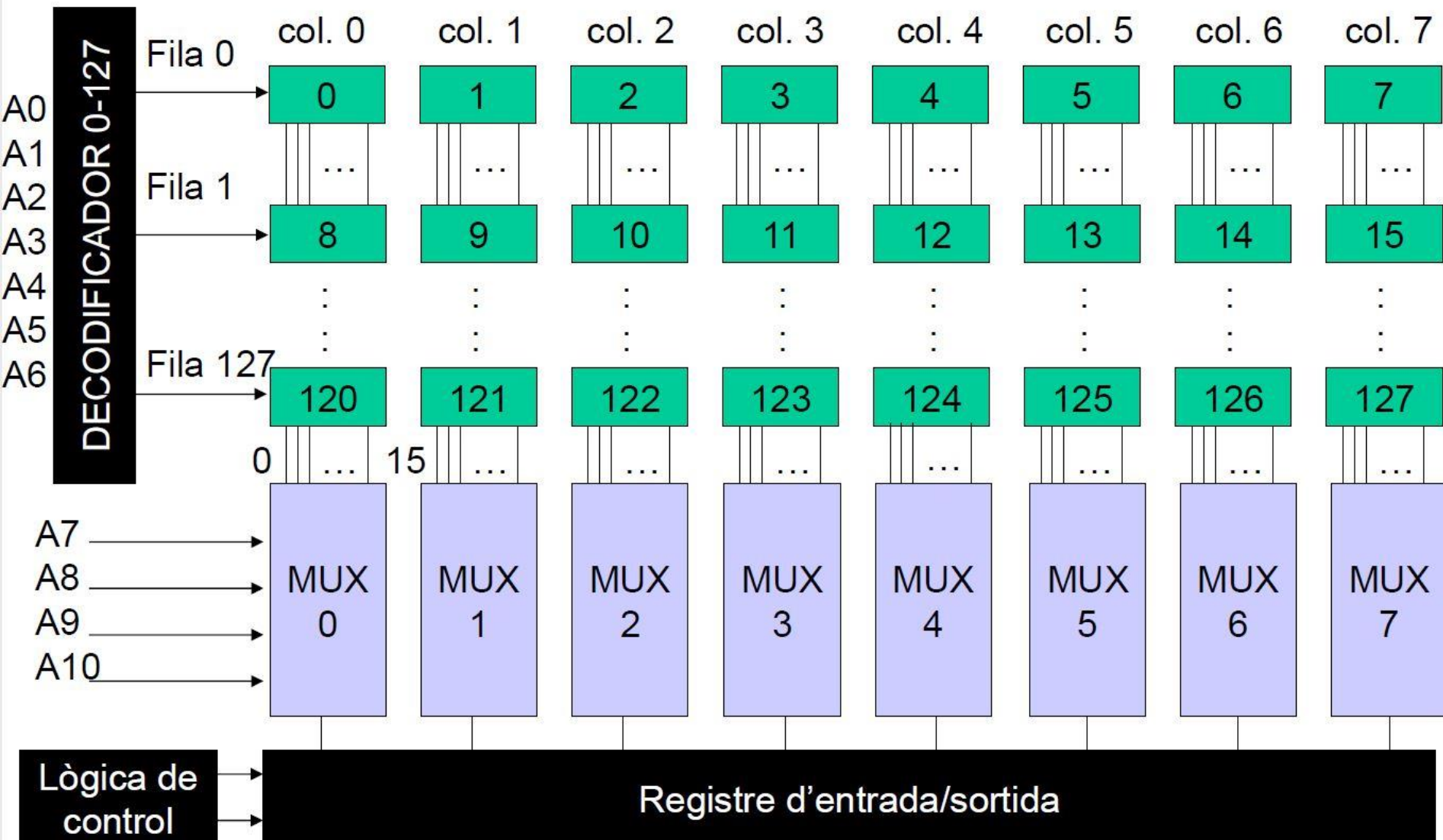
---

La distribució vista fins ara de la memòria és INEFICIENT Detalls a considerar:

- S'intenta crear una estructura matricial el més quadrat i petit possible
- Dividim les línies del decodificador en dos tipus:
  - a) Decodificador de files, format per les línies baixes del bus de direccions
  - b) Decodificador de columnes, format per les línies altes del bus de direccions

**EXEMPLE: MEMÒRIA RAM DE 2KBytes**

# ESTRUCTURA DE LA MEMÒRIA



## ESTRUCTURA DE LA MEMÒRIA

---

- Les 7 línies de menys pes del bus de direccions es decodifiquen amb el decodificador de files, seleccionant una de les 128 files possibles
- Les 4 línies restants, de A7 a A10 es decodifiquen amb el decodificador de columnes format per 8 MUX de 16 entrades de les que seleccionem 8, 1 per cada MUX.
- Amb aquest tipus d'estructura, les línies del bus de direccions es poden repartir de diferents formes en funció del fabricant.



---

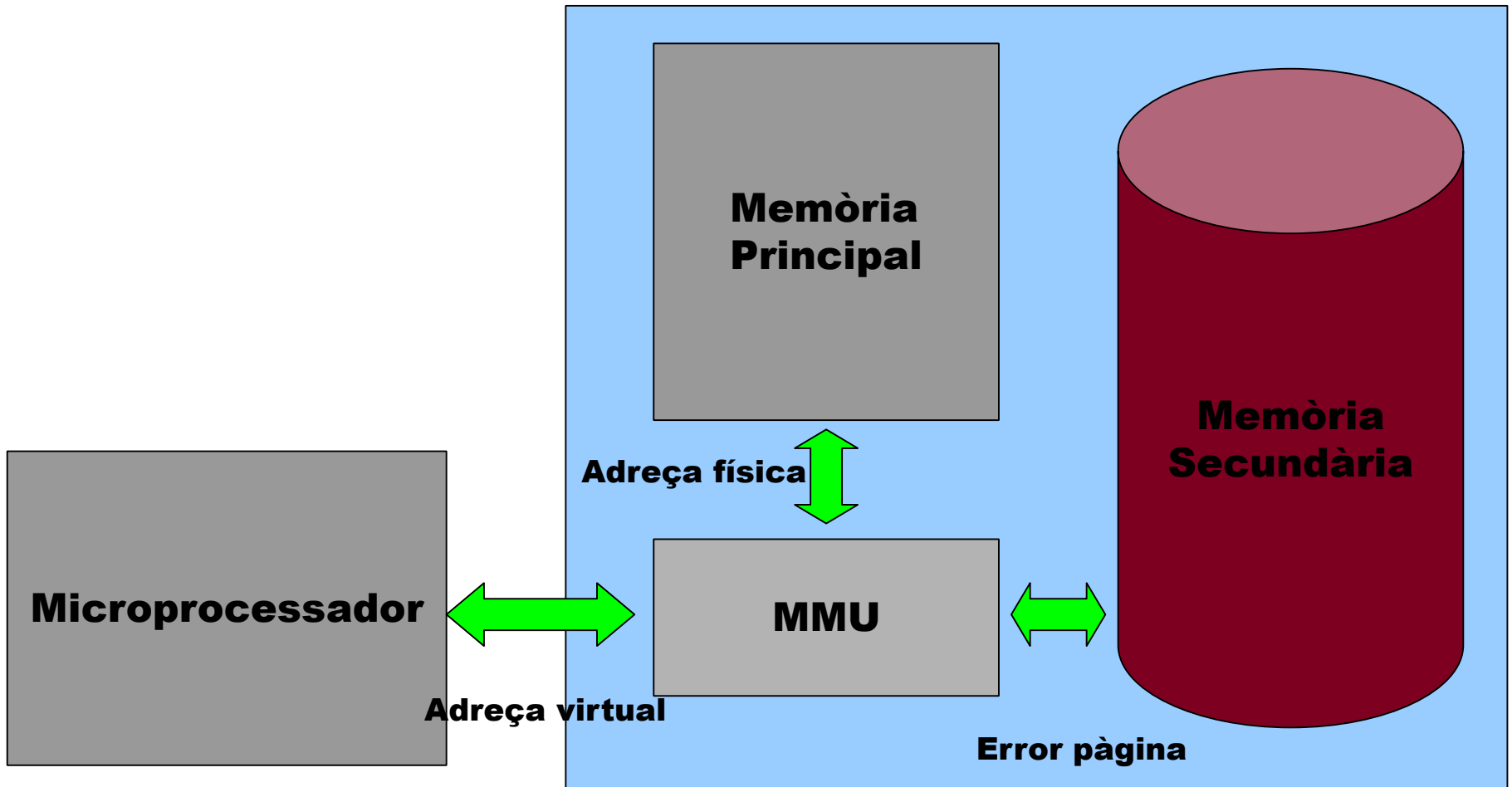
# MEMÒRIA SECUNDÀRIA | MEMÒRIA VIRTUAL

From Computer Desktop Encyclopedia  
Reproduced with permission.  
© 2005 Toshiba Corporation



# Jerarquia de memòria: Memòria Virtual

---



# Jerarquia de memòria: Memòria Virtual

---

- La memòria virtual és una tècnica d'administració de la memòria real que permet al SO proporcionar al sw d'usuari un espai d'adreces superior a la memòria física que tenim.
- La memòria virtual fa servir dos nivells de jerarquia de memòria: La memòria principal i la memòria de recolzament (memòria secundaria: Disc Dur)
- La gestió de la memòria virtual demanda una gestió automàtica que es fa per HW.
- Les fallades de pàgines són ateses pel SO. Per tant es fa per SW. El procés de migració per atendre les fallades de pàgines es denomina **paginació**

# Jerarquia de memòria: Memòria Virtual

---

- El mapa virtual associat a un programa en execució esta suportat físicament per una zona de memòria principal i una zona del disc anomenada zona d'intercanvi o **swap**
- El microprocessador genera adreces virtuals
- Tot i que el programa genera adreces virtuals, per poder executar-se ha de residir a la MP
- Si una dada no es troba a la MP, s'haurà de importar de la zona de swap per tal que el programa pugui executar-se correctament
- L'espai virtual i físic es divideix en pàgines. Pàgines virtuals (memòria virtual), pàgines d'intercanvi (swap) i marc de pàgina (MP)

# Prestacions dels Ordinadors

---

Els paràmetres característics necessaris per caracteritzar les prestacions dels ordinadors són:

1. Ample de paraula:  $n^{\circ}$  bits que fa servir el uP en paral·lel.
2. Memòria: Indica la mida de la memòria principal del uP
3. Memòria auxiliar: Expressa en Mbytes o Gbytes la mida dels perifèrics tipus disc que contingui el uP.
4. Ample de banda: Cabal d'informació capaç de transmetre un bus a una unitat
5. MIPS: (milions de instruccions per segon). Velocitat d'execució de les instruccions de la màquina
6. MFLOPS (milions d'operacions en coma flotant per segon). Velocitat de càlcul científic d'un computador

# Exemple prestaciones

	Clarkdale - 32nm		Lynnfield - 45nm		Bloomfield - 45nm
Modelos	Core i3	Core i5 6xx	Core i5 7xx	Core i7 8xx	Core i7 9xx
Velocidad	2.93 - 3.06 Ghz	3.2 - 3.46 Ghz	2.66 Ghz	2.8 Ghz	2.66 - 3.33 Ghz
Núcleos / Procesos	2/4	4/4	4/4	4/8	4/8
Memoria Caché	4MB	4MB	8MB	8MB	8MB

## Bandwidth Comparison

	Bus Clock	Internal Rate	Prefetch	Transfer Rate	Channel Bandwidth
DDR	100-200 MHz	100-200 MHz	2n	0.20-0.40 GT/s	1.60-3.20 GBps
DDR2	200-533 MHz	100-266 MHz	4n	0.40-1.06 GT/s	3.20-8.50 GBps
DDR3	400-1066 MHz	100-266 MHz	8n	0.80-2.13 GT/s	6.40-17.0 GBps
DDR4	1066-2133 MHz	100-266 MHz	8n	2.13-4.26 GT/s	12.80-25.60 GBps

# MIPS i MFLOPS

---

Calquem els MIPS a partir del nombre total d'instruccions i del temps que triguen en executar-se

$$MIPS = \frac{N^{\circ} \text{ total de instruccions}}{\text{temps que triga}} \times 10^{-6}$$

Donat que el temps depèn de la freqüència del sistema, tindrem

$$\text{temps que triga} = N^{\circ} \text{ cicles clk} \times \text{temps de cicle} = \frac{\langle N^{\circ} \text{ cicles clk} \rangle}{\text{Freqüència}}$$

$$\rightarrow MIPS = \frac{N^{\circ} \text{ total de instruccions} \times \text{Freqüència}}{\langle N^{\circ} \text{ cicles de clk} \rangle}$$

# MIPS i MFLOPS

---

De les expressions anteriors podem extreure un paràmetre clau:  
Els Cicles Per Instrucció

$$MIPS = \frac{Freqüència}{CPI} \times 10^{-6}$$

Els MIPS permeten calcular el temps que triga en executar-se un determinat programa amb un determinat nombre d'instruccions. El temps d'execució pot calcular-se com:

$$t_{exec} = \frac{N^{\circ} \text{ total d'instruccions} \times CPI}{Freqüència} = \frac{N^{\circ} \text{ total d'instruccions}}{MIPS} \times 10^{-6}$$



# Exemple

---

Un programa consta de 140 instruccions, de les quals 70 triguen en executar-se 4 cicles, 35 triguen en executar-se 5 cicles, 20 triguen en executar-se 3 cicles i les 15 restants triguen 7 cicles

- 1.- Calculeu el CPI promig per aquest programa
- 2.- Si l'ordinador funciona a una freqüència de 20 MHz, calculeu el temps que triga en executar el programa

# Rendiment d'un Ordinador

---

Definim el rendiment que té un determinat uP per executar un programa com:

$$\eta = \frac{1}{t_{exec}} = \frac{Freqüència}{N^{\circ} instruccions \times CPI}$$

On el temps d'execució no és més que el N<sup>o</sup> instruc. x Temps que triga en executar-se una instrucció.

# Rendiment d'un Ordinador

---

El rendiment d'un uP és directament proporcional a la freqüència de treball

El rendiment d'un uP és inversament proporcional al valor del CPI

La potència de processament és inversament proporcional al n° total d'instruccions que s'han d'executar.

# Comparativa del Rendiment

Ordinadors estudiats:

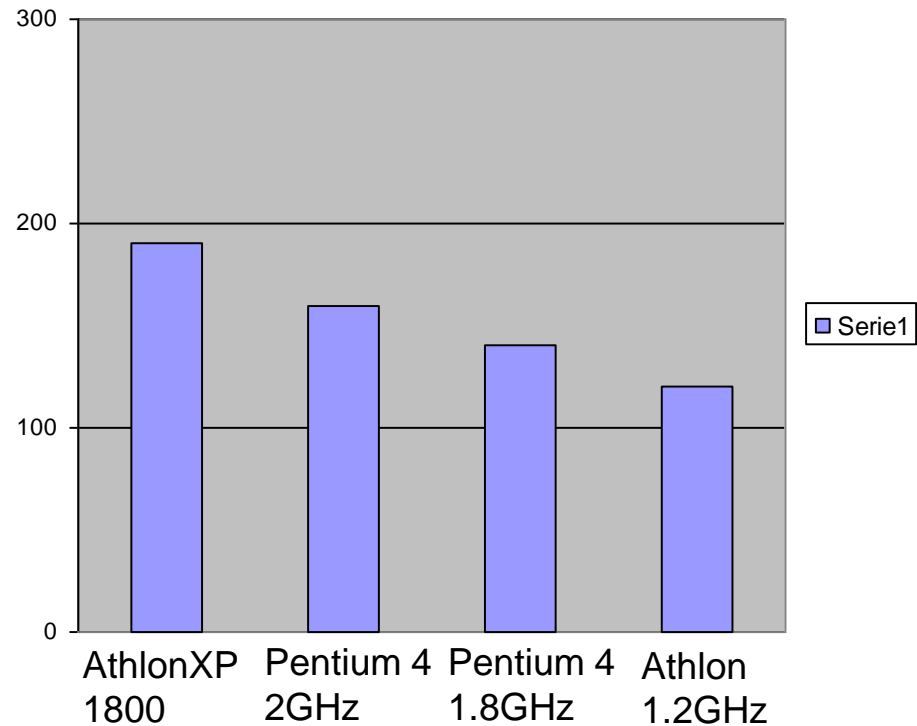
AMD Athlon XP 1800

Intel Pentium 4 2GHz

Intel Pentium 4 1.8GHz

AMD Athlon 1.2GHz

Font: Bapco Sysmark 2001 Resultat Global



# Comparativa del Rendiment (II)

---

Tot i que la freqüència del Pentium 4 es superior (2GHz), aquest processador té un CPI major que els seus predecessors per tant a igual freq. té menor rendiment. En el cas de AMD, el seu processador tot i treballar a una freq. inferior, donat que té un CPI inferior presenta un rendiment superior.

Per tal de fer aquests estudis les empreses utilitzen programes de prova o benchmarks especialment dissenyats per tal efecte.

# Exemple

---

Es disposa de les següents dades per dos ordinadors i una determinada aplicació:

- 1.- Power PC 601 de 80 MHz de freqüència i 70 Megainstruccions per segon (MIPS) de potència
- 2.- Pentium 120 MHz i 85 MIPS

Calculeu els CPI promig de cada processador

Si es fa servir un programa de 70 línies de codi, determina el temps d'execució, el rendiment i valora quin dels dos ordinadors presenta millor rendiment.

# Augment del rendiment: Lleis de Amdahl

Les lleis d'Amdahl serveixen per evaluar l'augment de rendiment en un sistema al introduir una millora.

Es recomana sempre millorar aquells elements que es fan servir més freqüentment, ja que són els que més influeixen en el rendiment.

## Primera Ley de Amdahl:

“L'augment del rendiment per una millora està limitat per el temps que s'utilitza aquesta millora”

$$t_{\text{millorat}} = t_{\text{antic}} \times \left( \frac{\text{Fracció de temps millorada}}{\text{Guany de velocitat}} + \text{Fracció de temps no millorada} \right)$$

# Augment del rendiment: Lleis de Amdahl

---

Exemple:

Es canvia la ALU d'un uP dedicat a una tarea a la qual hi dedica el 50% del temps. La nova ALU és dos vegades més ràpida que l'anterior. Quin és el temps de millora?

$$T_{\text{millora}} = T_{\text{anterior}} \times \left\{ \left( 0.5/2 \right) + 0.5 \right\} = T_{\text{anterior}} \times 0.75$$

S'ha millorat el temps en un 25%



# Augment del rendiment: Lleis de Amdahl

---

## Segona Llei de Amdahl

“ Quan s'introdueix una millora a un computador  
previament millorat, l'increment del rendiment és  
menor que si aquesta millora s'hagués introduït  
sobre el uP sense millorar”

Aquesta llei va enfocada a la variació del rendiment  
quan es realitzen millores successives

# Augment del rendiment: Lleis de Amdahl

---

## Exemple

A un uP se li fa una millora a la ALU, sent 30% més ràpida que l'anterior, i el programa fa un us d'aquesta millora el 40% del temps. Posteriorment es canvia la caché substituint-la per una 4 vegades més ràpida, amb una tasa d'encerts del 80% i tenint en compte que el programa presenta un 15% del temps en accessos a memòria.

Percentatge de millora del temps:

- a) Si només canbiem la caché
- b) Si només substituïm la ALU
- c) Si primer es canvia la ALU i després la caché

# Augment del rendiment: Lleis de Amdahl

Calculem la millora de la caché (considerem un temps inicial de 100 segons):

$$t_{\text{caché}} = 100 \times \left( \frac{0.15 \times 0.8}{4} + 0.88 \right) = 91 \text{ segons}$$

Calculem la millora de la ALU:

$$t_{\text{ALU}} = 100 \times \left( \frac{0.4}{1.3} + 0.6 \right) = 90.77 \text{ segons}$$

Aplicant les dues millores tenim:

$$t_{\text{ALU+caché}} = 90.77 \times \left( \frac{0.15 \times 0.8}{4} + 0.88 \right) = 82.6 \text{ segons}$$

# Exemple

---

Un ordinador té 512 Mbytes de memòria RAM i un processador pentium 4 a 1 GHz de freqüència. Usualment fa servir programes que utilitzen el 35% del temps d'execució en accedir a la memòria. L'accés al disc dur ocupa el 12% del temps. Ens pregunten que és millor:

Canviar la memòria per una altra que dobla la velocitat d'accés o

Cumprar un altre disc dur amb el quàdruple de la velocitat d'accés al disc

Què passa si fem les dues millores?