Arquitectura de computadors

Memòries

- Estructura jeràrquica de les memòries
- Memòria principal
- Memòria cau
- Memòria Virtual

Dispositius físics

- Memòries semiconductores
  - RAM i ROM

- Memòries de superfície magnètica
  - Discos durs i cintes

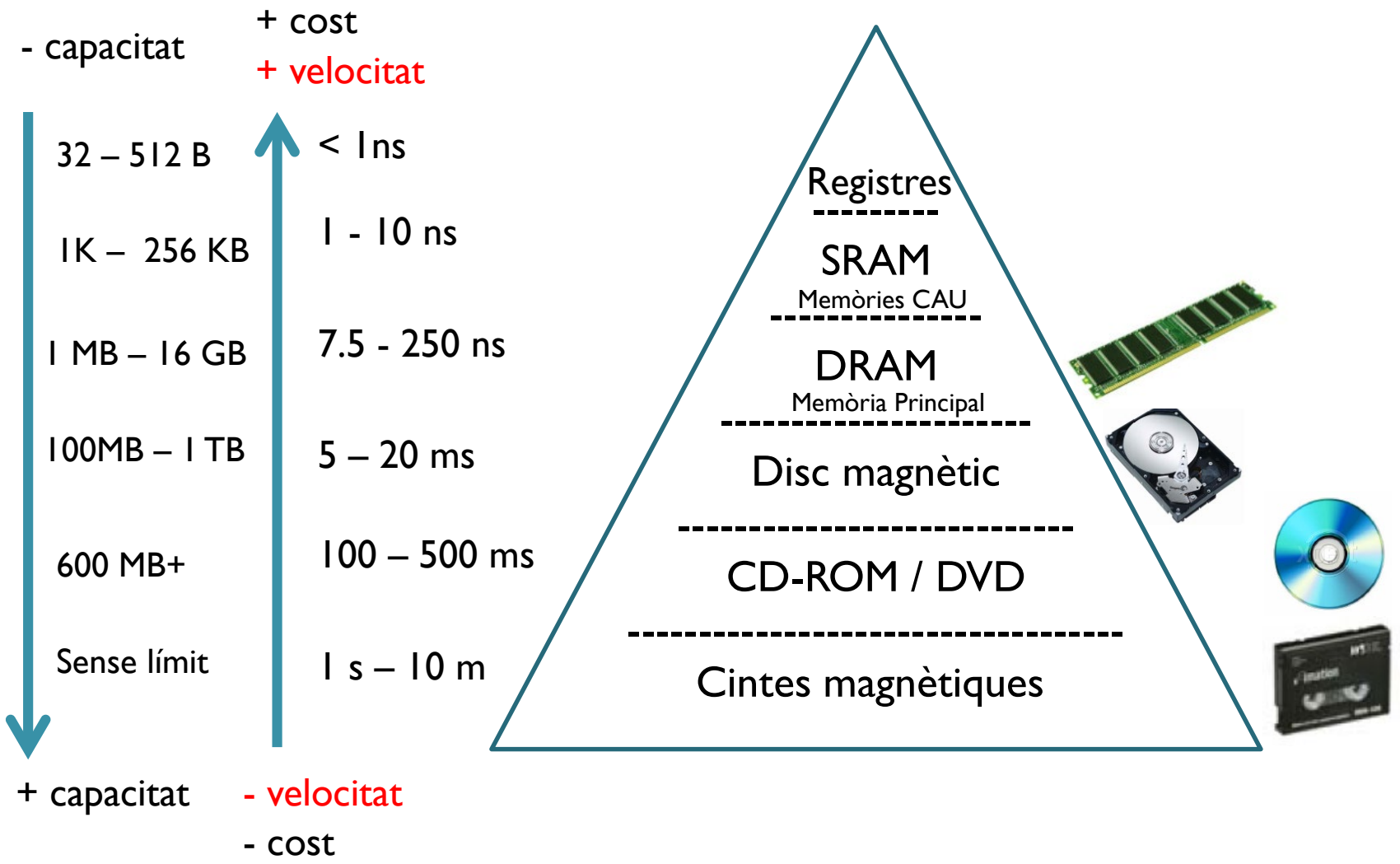
- Memòries òptiques
  - CD, DVD, HD-DVD, blu-ray,..

- Altres
  - Memòries de bombolles
  - Hologrames

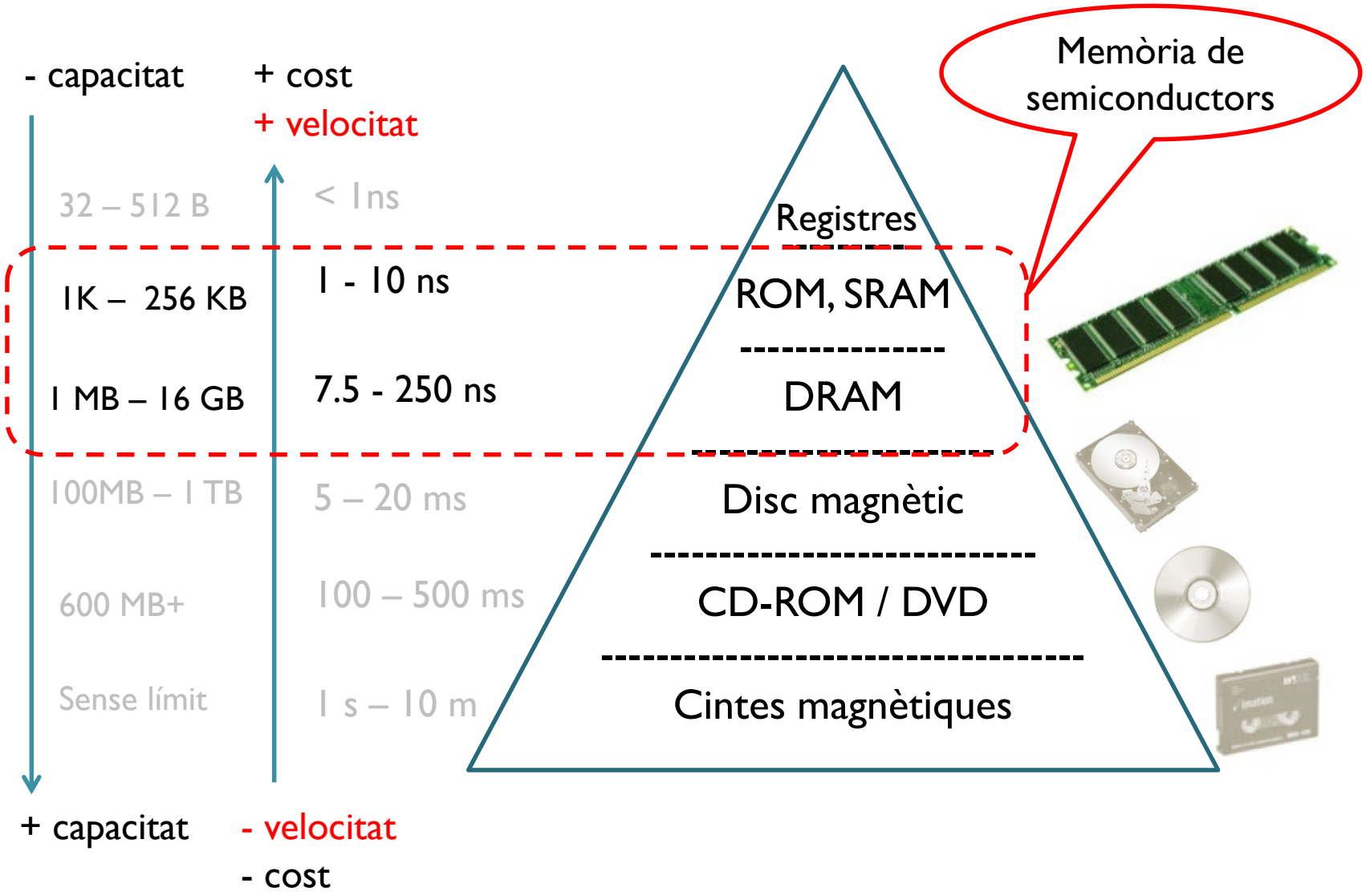
Característiques principals

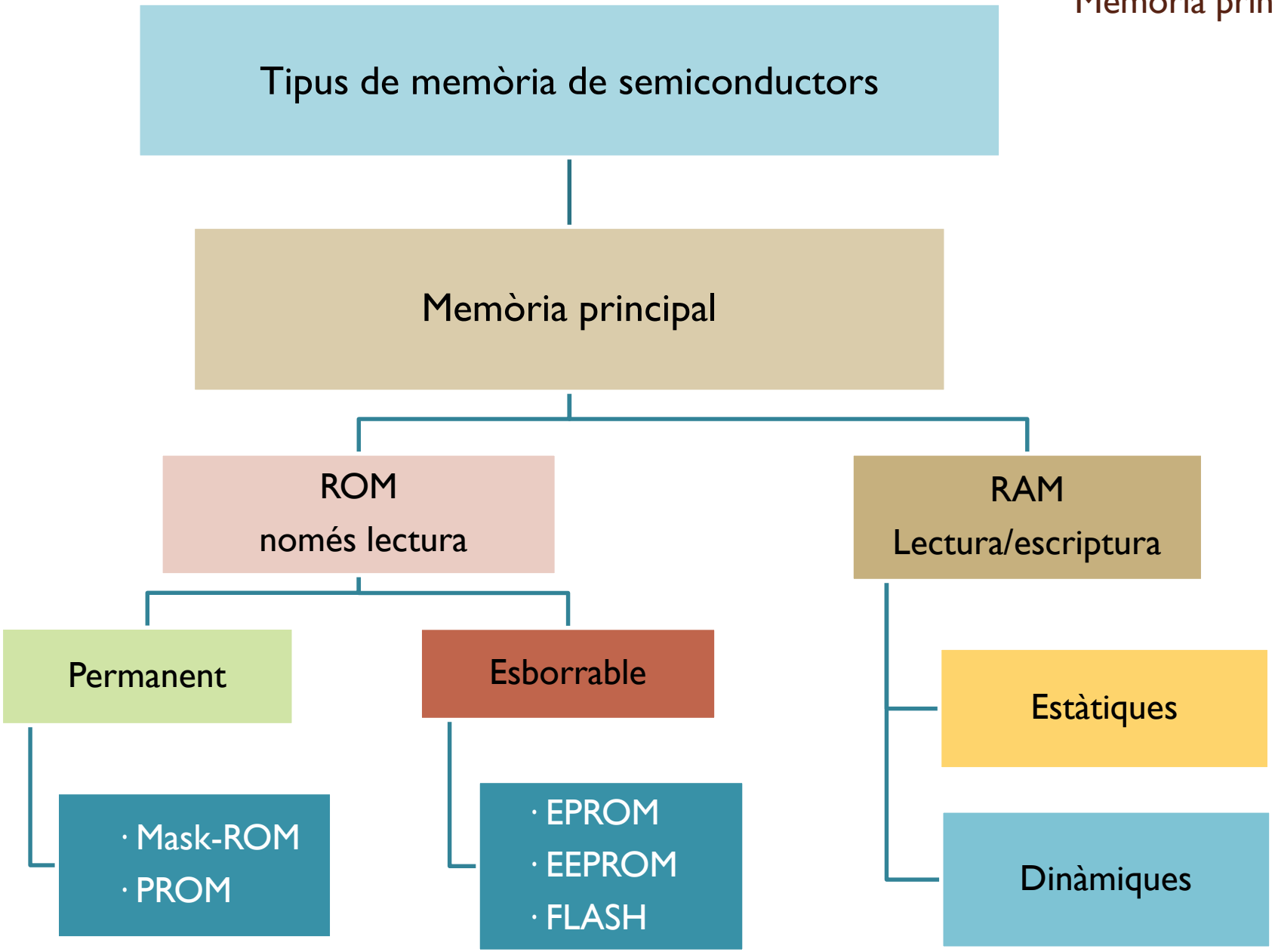
- Físiques
  - Volatilitat (RAM, DRAM, SRAM, ...)
  - No Volàtil (ROM, Flash, CD, DVD, ...)
- Organització
  - En bits, paraules, blocs, ..
  - Accés sèrie, paral·lel, ..
- Prestacions
  - Capacitat: Quantitat de dades (bits) que es poden emmagatzemar
  - Temps d'accés: Temps entre que es donen les adreces i s'obtenen les dades
  - Temps de cicle de memòria: Temps d'accés més el temps de recuperació
  - Velocitat de transferència: Quantitat de dades copiades per unitat de temps
  - Cost: Preu per unitat de dada emmagatzemada

Jerarquia de memòria



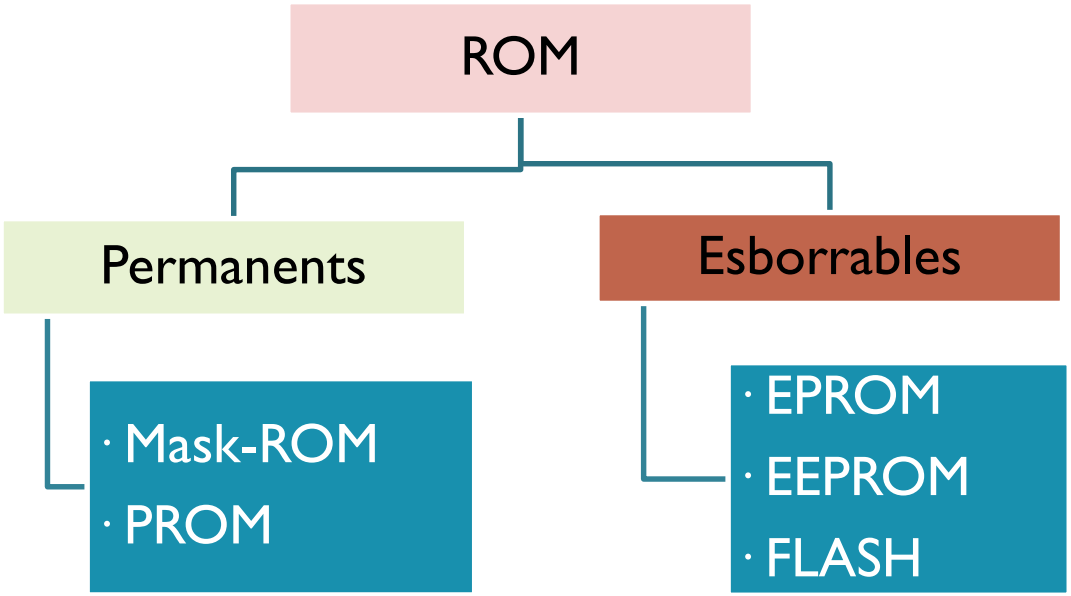
Memòria principal




Tipus de memòria de semiconductors
Memòria principal
ROM
només lectura
RAM
Lectura/escriptura
Permanent
Esborrable
Estàtiques
· Mask-ROM
· PROM
· EPROM
· EEPROM
· FLASH
Dinàmiques

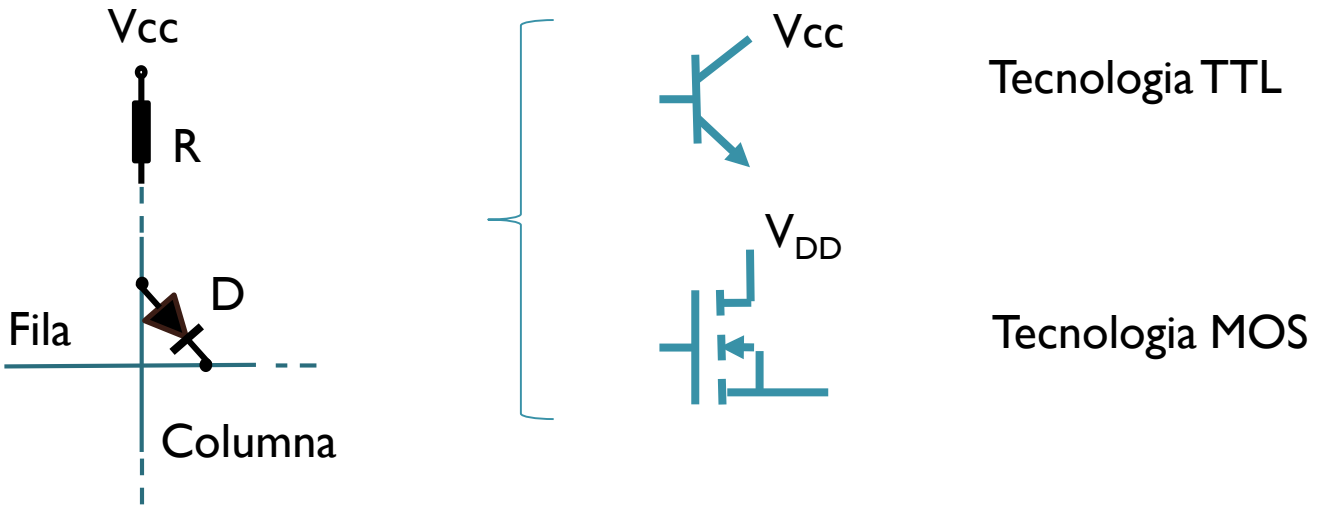
Memòria de només lectura: ROM (Read Only Memory)

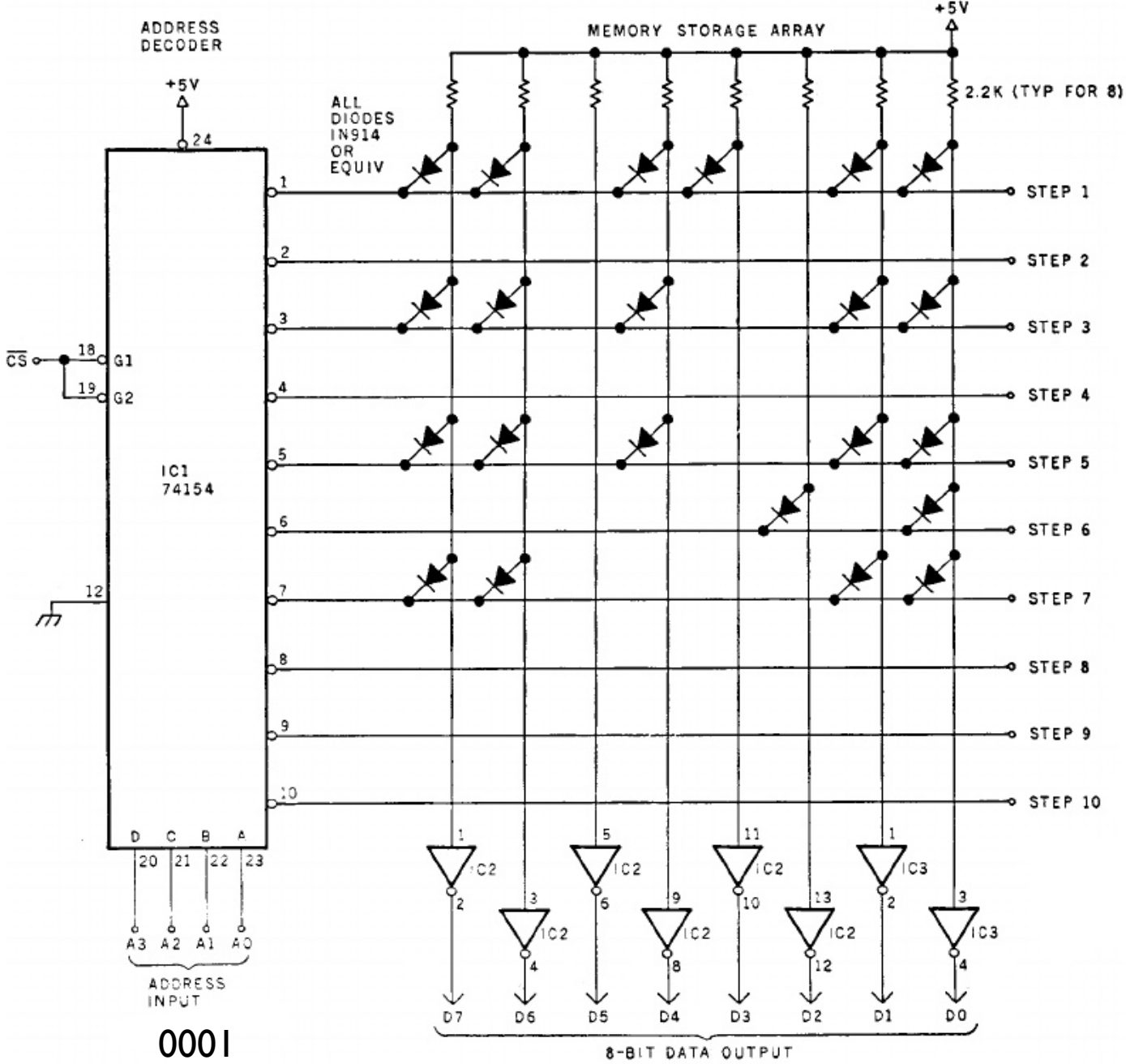
- Emmagatzemament permanent (no volàtil). La informació emmagatzemada es conserva sense necessitat d'energia.
- S'utilitza en les memòries d'arrencada dels PCs anomenades BIOS (Basic Input/Output System) boot program. També s'utilitza en sistemes encastats per emmagatzemar el programa de la aplicació.



Memòria ROM de màscara

- Aquest tipus de ROM es caracteritza perquè la informació continguda al seu interior s'emmagatzema durant la seva fabricació i no es pot alterar.
- El procés de fabricació és car, però esdevé econòmic amb la producció de grans quantitats.
- La programació es realitza mitjançant el disseny d'un negatiu fotogràfic, anomenat màscara, on s'especifiquen les connexions internes de la memòria (fotolitografia).
- Són ideals per emmagatzemar microprogrames, sistemes operatius, taules de conversió i caràcters.





Exemple de memòria ROM amb díodes

00100100 -> 24_H

00111100 -> $3C_H$

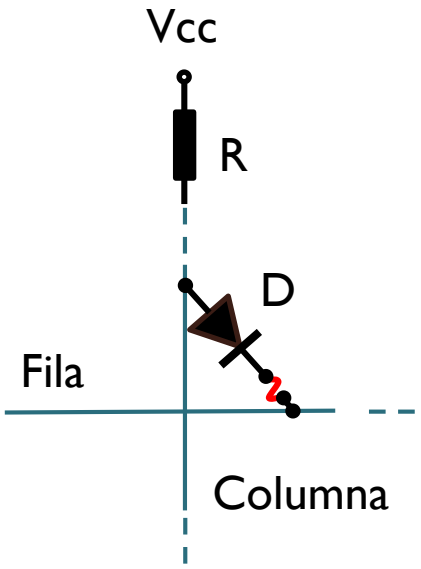
0001
0111

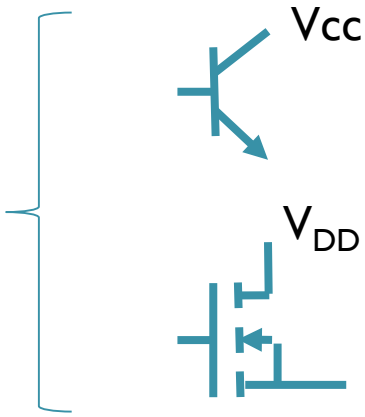
11011011 -> DB_H

11000011 -> $C3_H$

Memòria PROM

- **Programmable Read Only Memory**: ROM programable.
- Aquest tipus de memòria a diferència de la ROM no es programa durant el procés de fabricació, sinó que l'efectua l'usuari i només es pot fer una vegada, després de la qual no es pot esborrar o tornar a emmagatzemar altra informació.
- Per emmagatzemar la informació s'empren dues tècniques: per destrucció de fusible o per destrucció d'unió.
- La informació es programa en les diferents cel · les de memòria aplicant l'adreça al bus d'adreces, les dades en els buffers d'entrada de dades i un pols de 10 a 30V, en un terminal dedicat per fondre els fusibles corresponents. Quan s'aplica aquest pols a un fusible de la cel · la, s'emmagatzema un 0 lògic, en cas contrari s'emmagatzema un 1 lògic (estat per defecte), quedant d'aquesta manera la informació emmagatzemada de forma permanent.



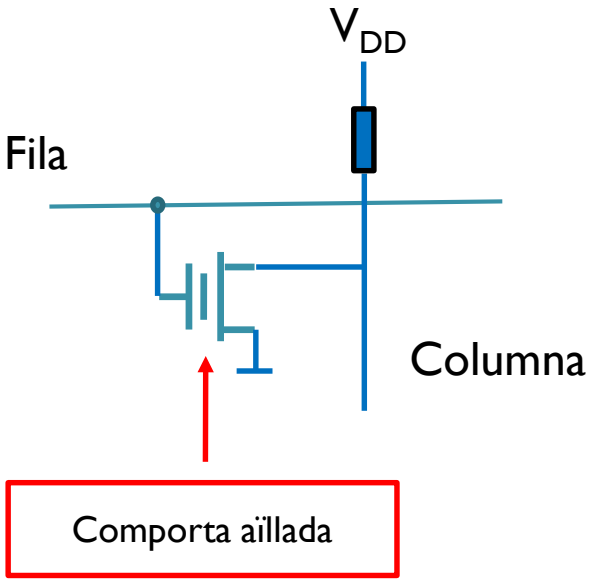


Tecnologia TTL

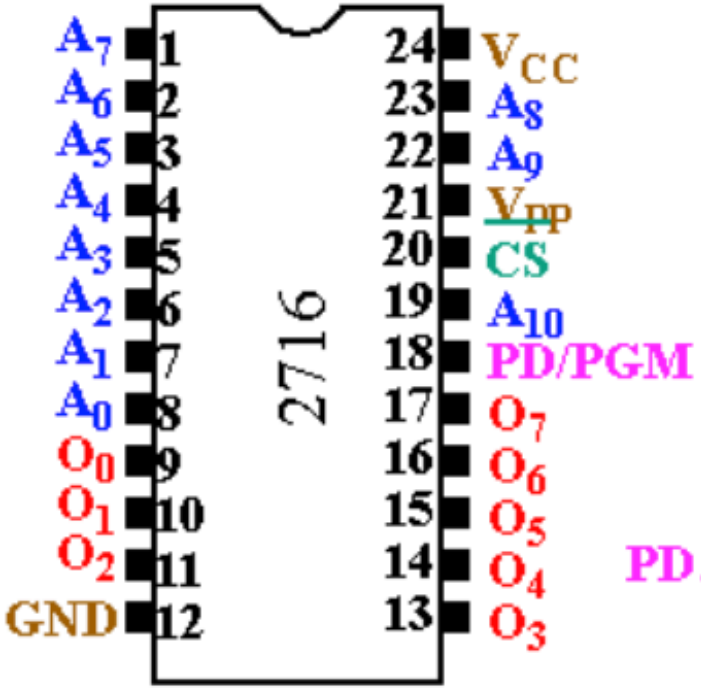
Tecnologia MOS

Memòria EPROM

- **Erasable Read Only Memory.**
- Aquest tipus de memòria és similar a la PROM amb la diferència que la informació es pot esborrar i tornar a gravar diverses vegades.
- La programació s'efectua aplicant a un pin especial de la memòria una tensió entre 10 i 25 Volts durant aproximadament 50 ms, segons el dispositiu, a la vegada que es proporciona la posició de memòria als pins d'adreça i es posa la informació a emmagatzemar en les entrades de dades.
- Aquest procés pot trigar diversos minuts depenent de la capacitat de memòria. Per esborrar la memòria s'ha d'insolar amb radiació UV durant 10-20 minuts.
- La memòria EPROM es composa d'una matriu de transistors MOSFET de Canal N de comporta aïllada, anomenats FAMOS (Floating gate Avalanche Metal Oxide Semiconductor).



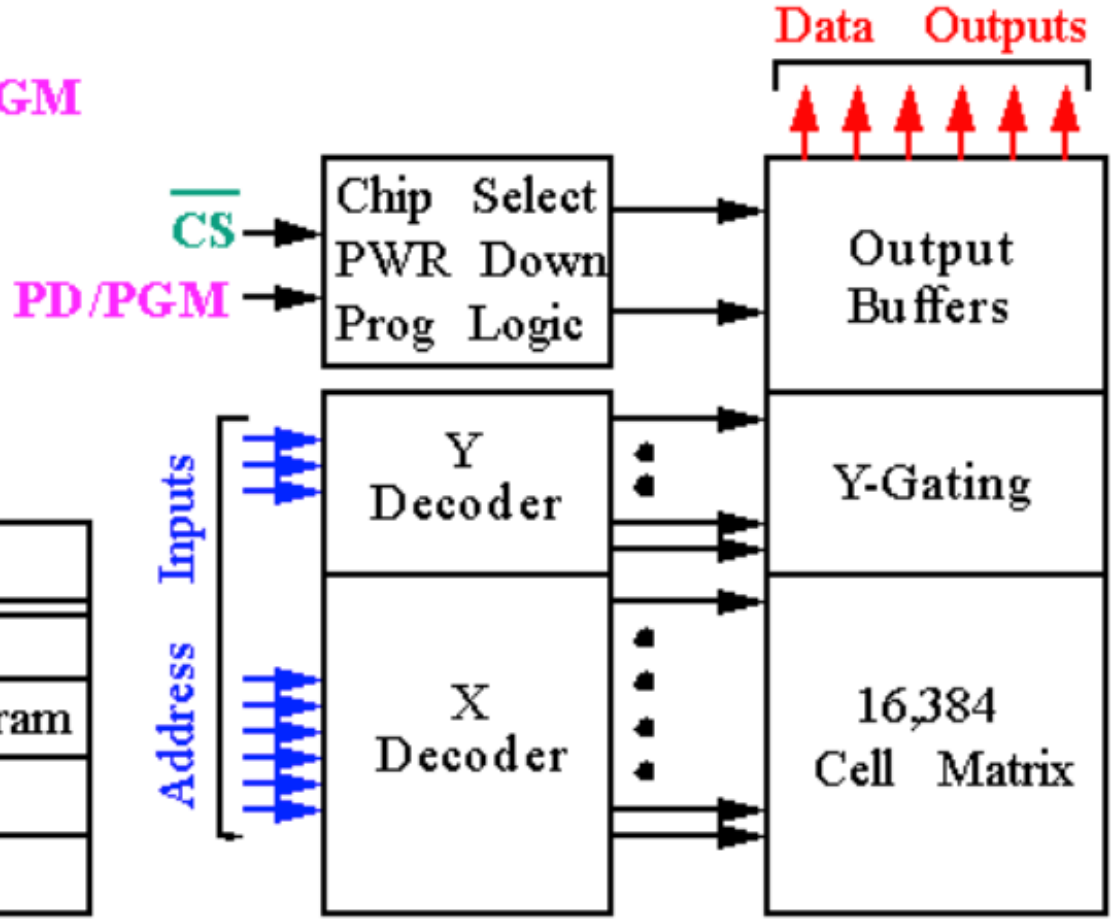
- *Intel 2716 EPROM (2K X 8):*



2K x 8 EPROM

Pin(s)	Function
A_0-A_{10}	Address
PD/PGM	Power down/Program
$\overline{CS}$	Chip Select
O_0-O_7	Outputs

V_{PP} is used to program the device by applying 25V and pulsing PGM while holding $\overline{CS}$ high.



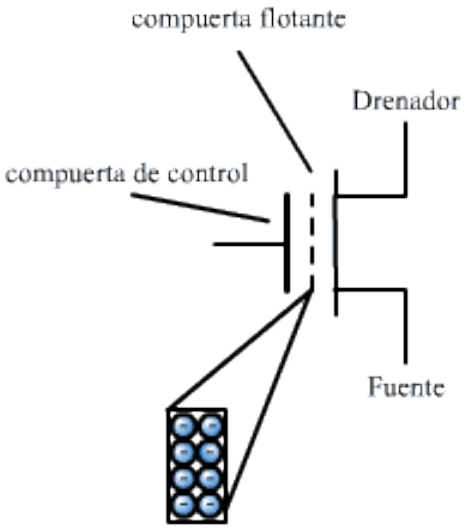
Memòria EEPROM

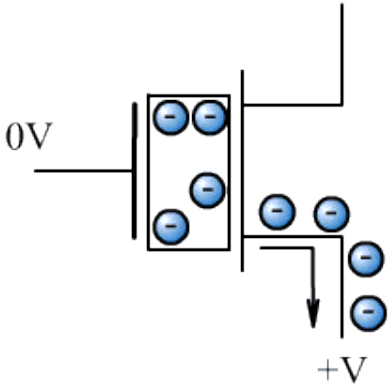
Electrically Erasable Programmable Read-Only Memory:

- Aquesta memòria és programable i esborrable elèctricament.
- Actualment aquestes memòries es construeixen amb transistors de tecnologia MOS (Metal Oxide Silicon) i MNOS (Metall Nitride-Oxide Silicon).
- Les cel · les de memòria en les EEPROM són similars a les cel · les EPROM i la diferència bàsica es troba en la capa aïllant al voltant de cada comporta flotant, que és més prima i no és fotosensible.
- La programació d'aquestes memòries és similar a la programació de la EPROM. Es realitza per aplicació d'una tensió de 21 Volts als terminals dels transistors MOSFET, deixant d'aquesta manera una càrrega elèctrica, que és suficient per activar els transistors i emmagatzemar la informació.
- L'esborrat de la memòria s'efectua aplicant tensions sobre les comportes per alliberar la càrrega elèctrica emmagatzemada en elles.

Memòria FLASH

- La memòria **FLASH** és similar a la EEPROM: Es pot programar i esborrar elèctricament. Es caracteritza per tenir alta capacitat per emmagatzemar informació i és de fabricació senzilla, el que permet fabricar models de capacitat equivalent a les EPROM a menor cost.
- Les cel · les de memòria es troben constituïdes per un transistor MOS de portes apilades, que es forma amb una porta de control i una porta aïllada.
- Per programar la cel·la, el programador aplica una tensió elevada a la porta de control de totes aquelles cel·les que han d'emmagatzemar un '0'. Aquesta tensió permet als electrons que es troben en el substrat del transistor, travessar l'aïllant a través d'un "túnel" i arribar fins a la porta flotant. S'aconsegueix un '0'.

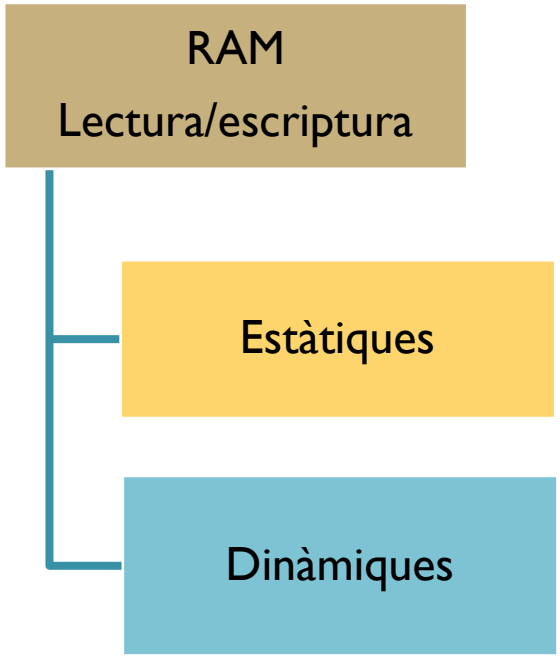




Procés de descàrrega d'una cel · la de memòria FLASH

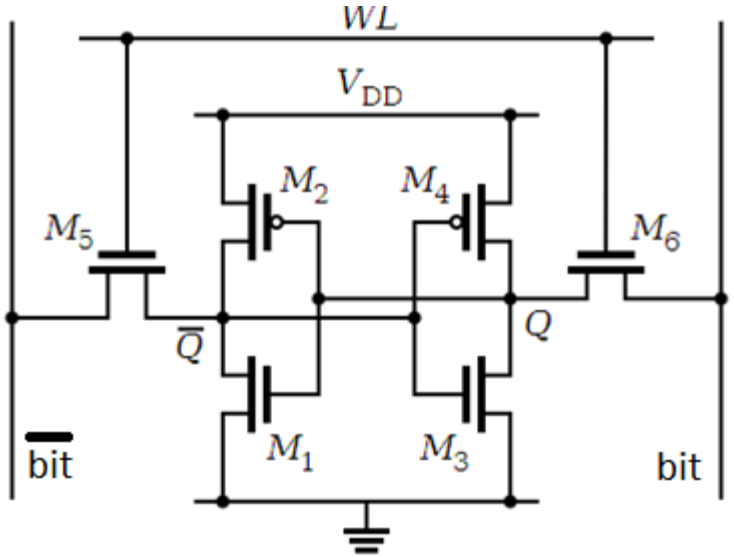
Memòria de lectura i escriptura RAM (Random Access Memory)

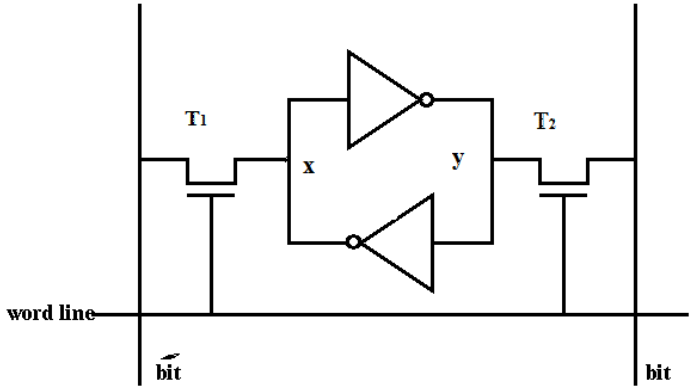
- Emmagatzemament temporal (volàtil)
- S'utilitza en la memòria principal per emmagatzemar programes i dades



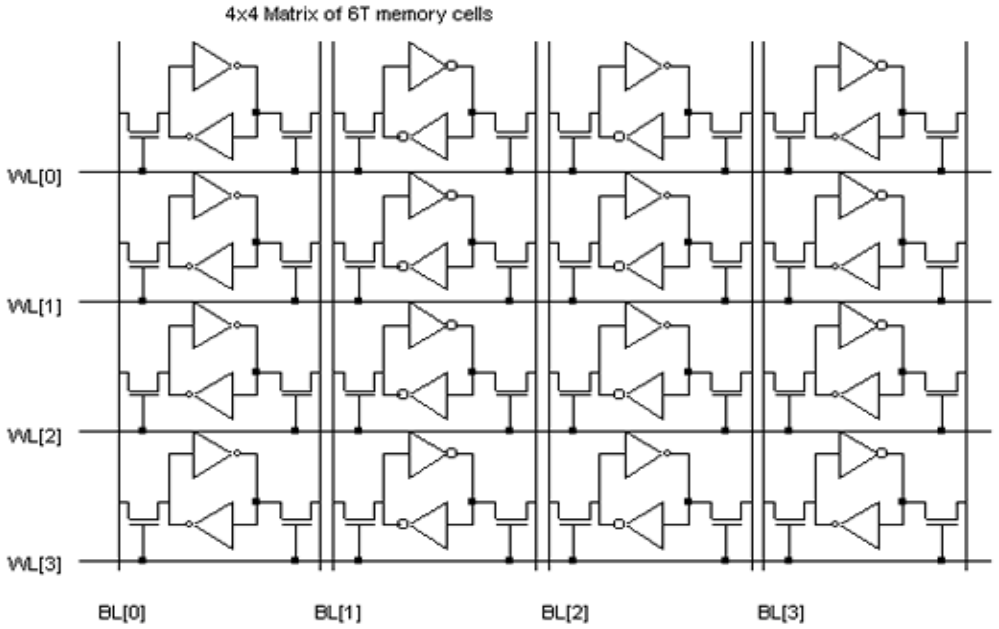
RAM estàtica (SRAM)

- Emmagatzema els bits com si fos una bàscula biestable. No es descarrega



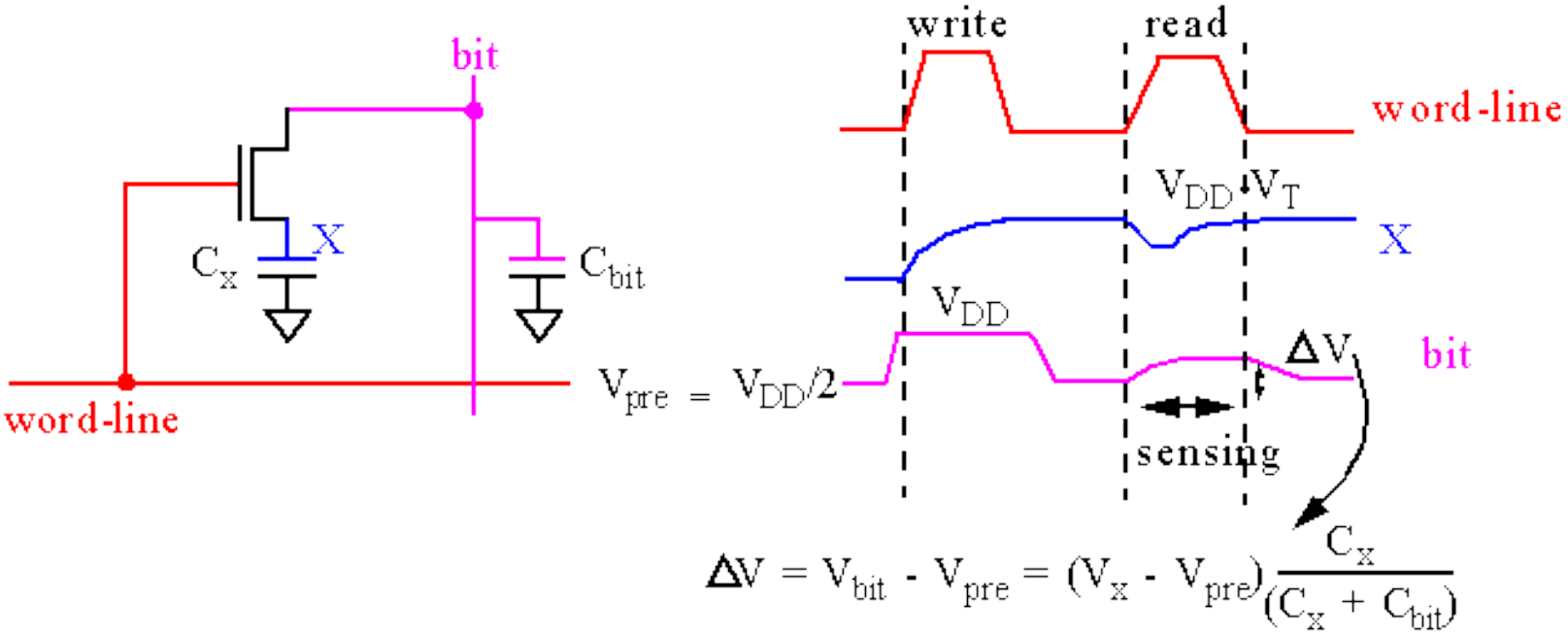


La cel·la s'activa mitjançant un nivell alt a l'entrada "Word line" i les dades es carreguen o es llegeixen a través de les línies laterals "bit".



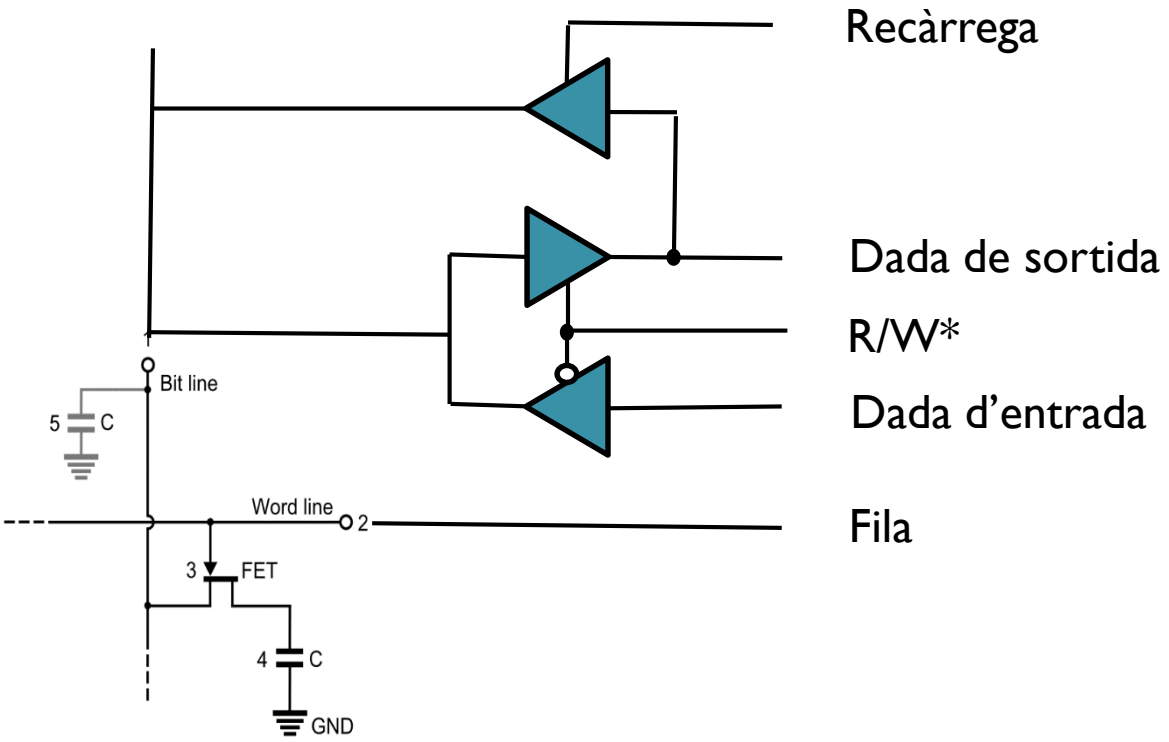
RAM dinàmica (DRAM)

- Emmagatzema els bits per la càrrega d'un condensador
- Té l'inconvenient que es descarrega i s'ha de refrescar periòdicament (2-4ms) amb circuits addicionals.
- Avantatges: Fabricació simple, gran capacitat d'emmagatzematge i molt econòmica.



RAM dinàmica (DRAM)

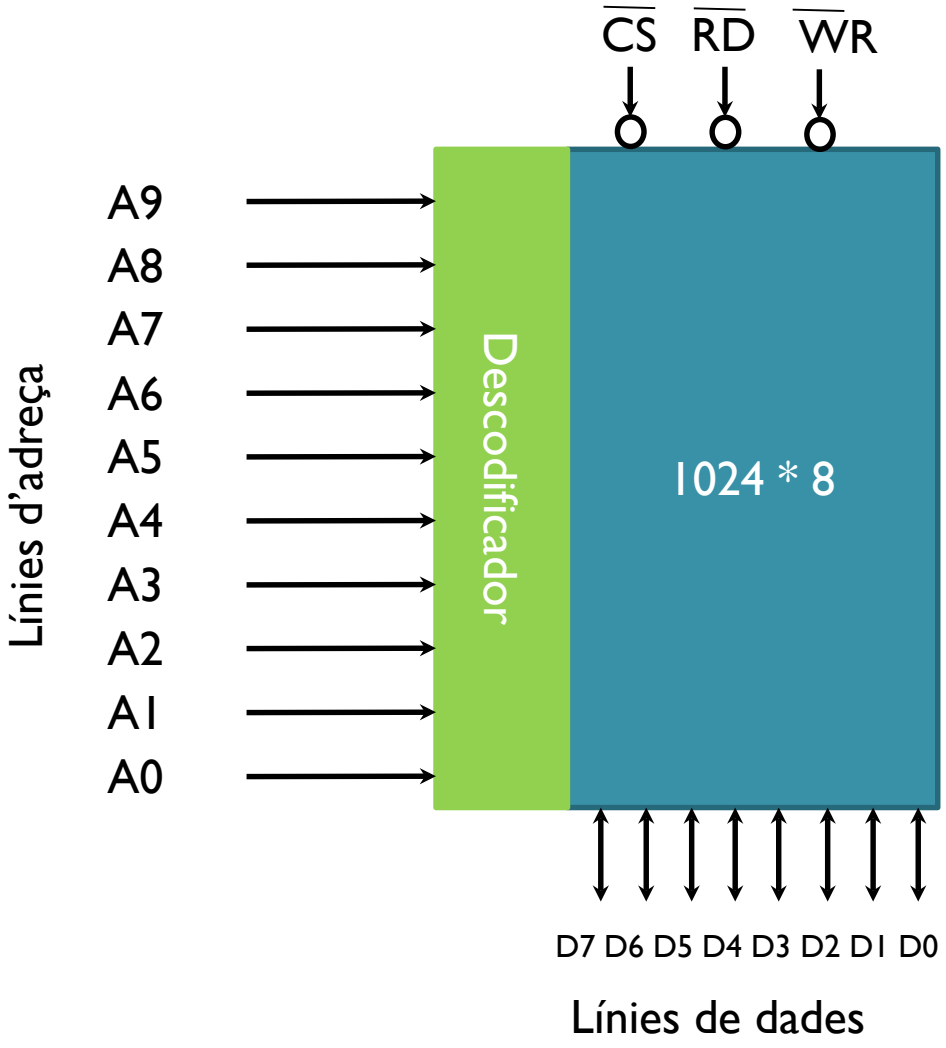
- L'operació de la cel · la és similar a la d'un interruptor, quan l'estat a la fila es a nivell alt, el transistor entra en saturació i la dada present al bus intern de la memòria (columna) s'emmagatzema al condensador, durant una operació d'escriptura i s'extreu en una operació de lectura.
- L'inconvenient que té aquest tipus de memòries és que cal recarregar la informació emmagatzemada en les cel · les, per la qual cosa aquestes cel · les requereixen de circuiteria addicional per complir aquesta funció. A aquest funció especial es diu de refresc.



SRAM vs DRAM

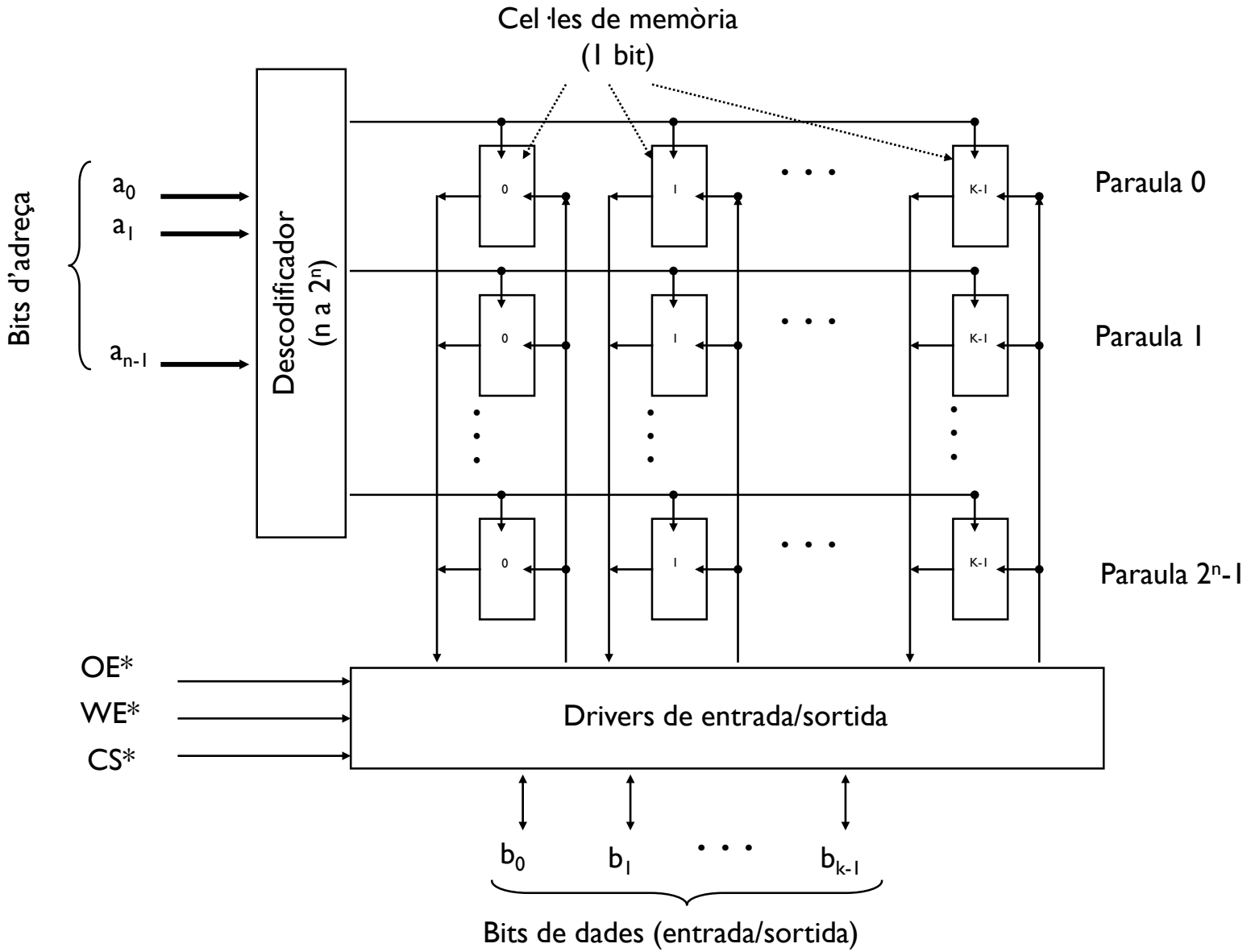
Memòria	Avantatges	Inconvenients
SRAM	• La velocitat d'accés és alta. • Per retenir les dades només necessita estar alimentada. • Són més fàcils de dissenyar.	• Menor capacitat, ja que cada cel · la d'emmagatzematge requereix més transistors. • Major cost per bit. • Més consum de Potència.
DRAM	• Més densitat i capacitat. • Menor cost per bit. • Menor consum de potència.	• La velocitat d'accés és baixa. • Necessita recàrrega de la informació. emmagatzemada per retenir (refresc). • Disseny complex.

Xip de SRAM típic de 1KByte

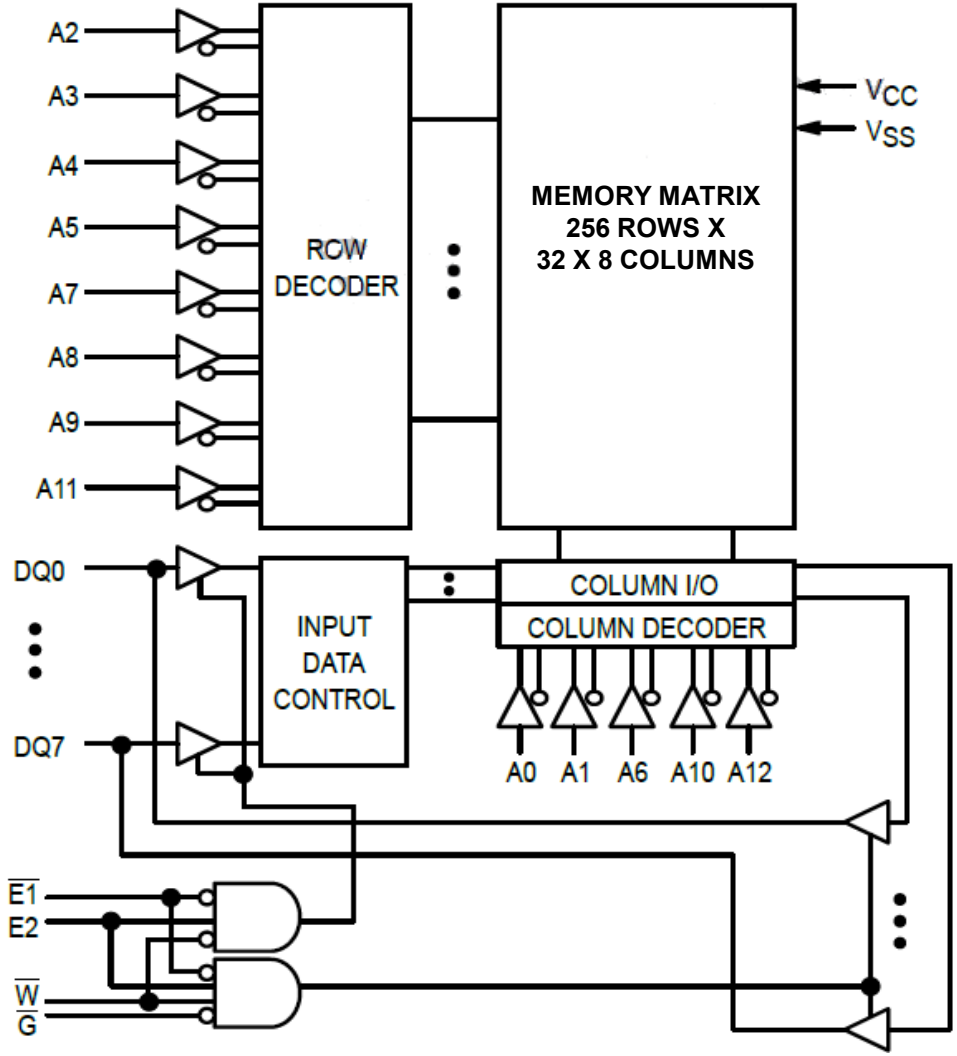


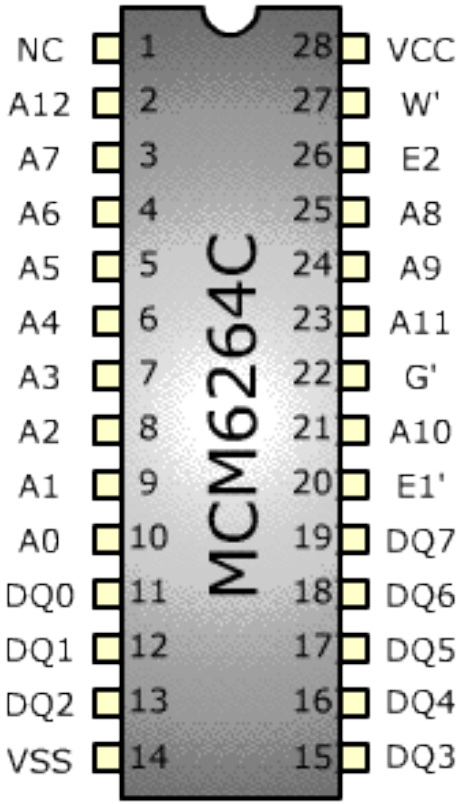
Una memòria necessita les línies d'adreces per identificar una posició de memòria, un senyal de "Chip select" (CS) per habilitar el xip i els senyals de control per llegir (/RD) o per escriure (/WR) en les posicions de memòria

RAM estàtica (SRAM)



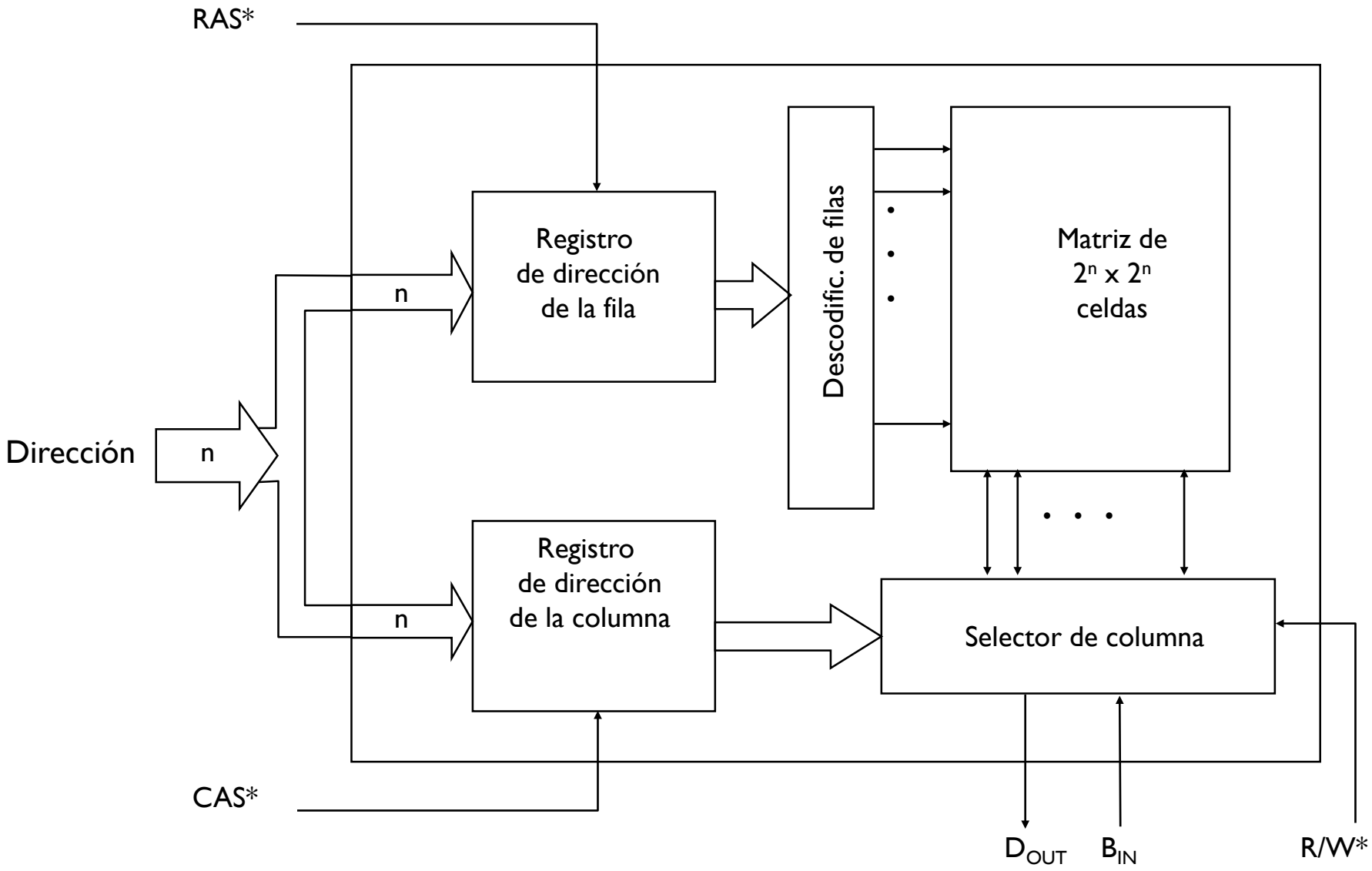
SRAM MCM6264C





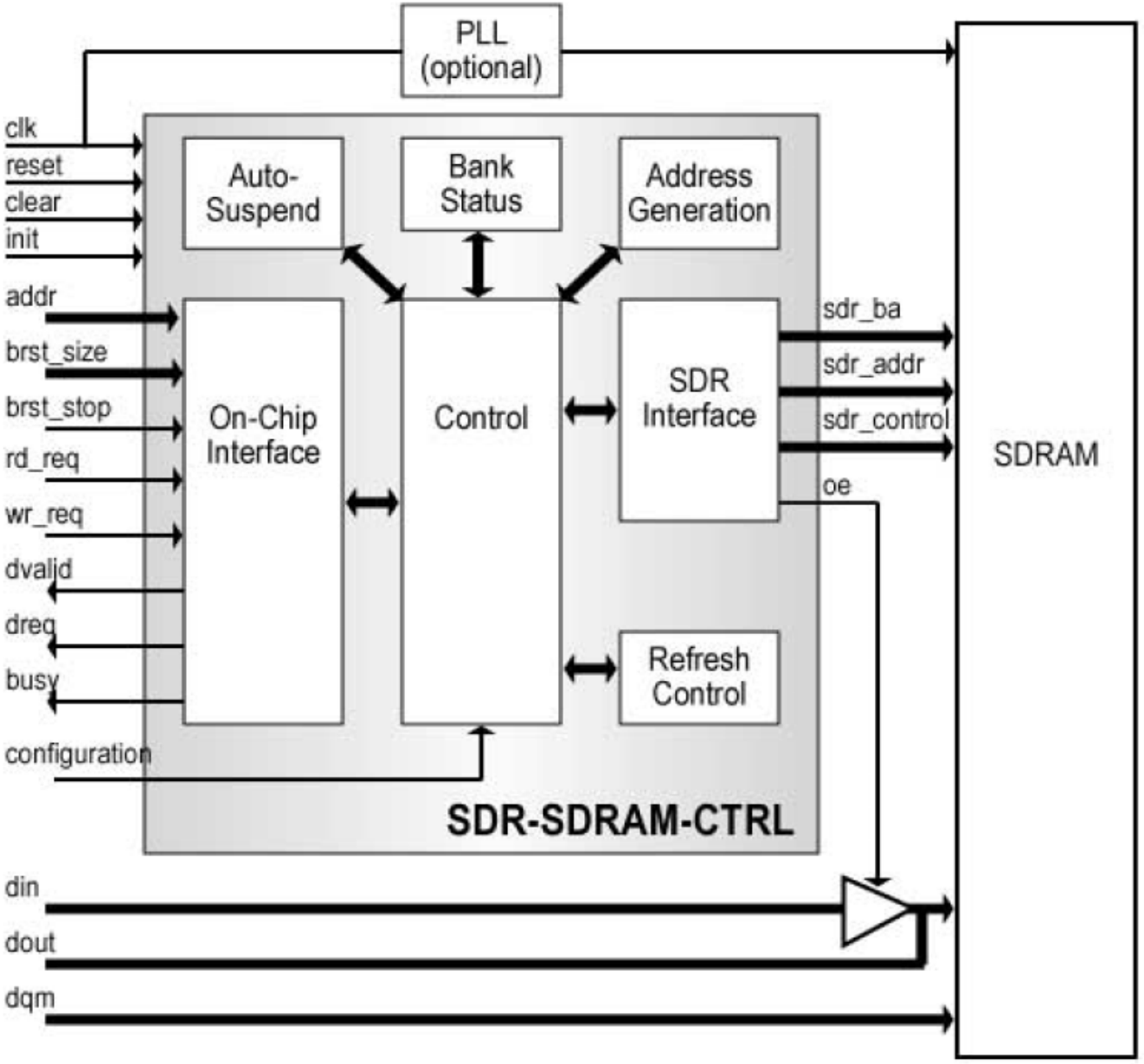
$\overline{E1}$	E2	$\overline{G}$	$\overline{W}$	Mode	V_{CC} Current	Output	Cycle
H	X	X	X	Not Selected	I_{SB1}, I_{SB2}	High–Z	—
X	L	X	X	Not Selected	I_{SB1}, I_{SB2}	High–Z	—
L	H	H	H	Output Disabled	I_{CCA}	High–Z	—
L	H	L	H	Read	I_{CCA}	D_{out}	Read Cycle
L	H	X	L	Write	I_{CCA}	High–Z	Write Cycle

RAM dinàmica (DRAM)

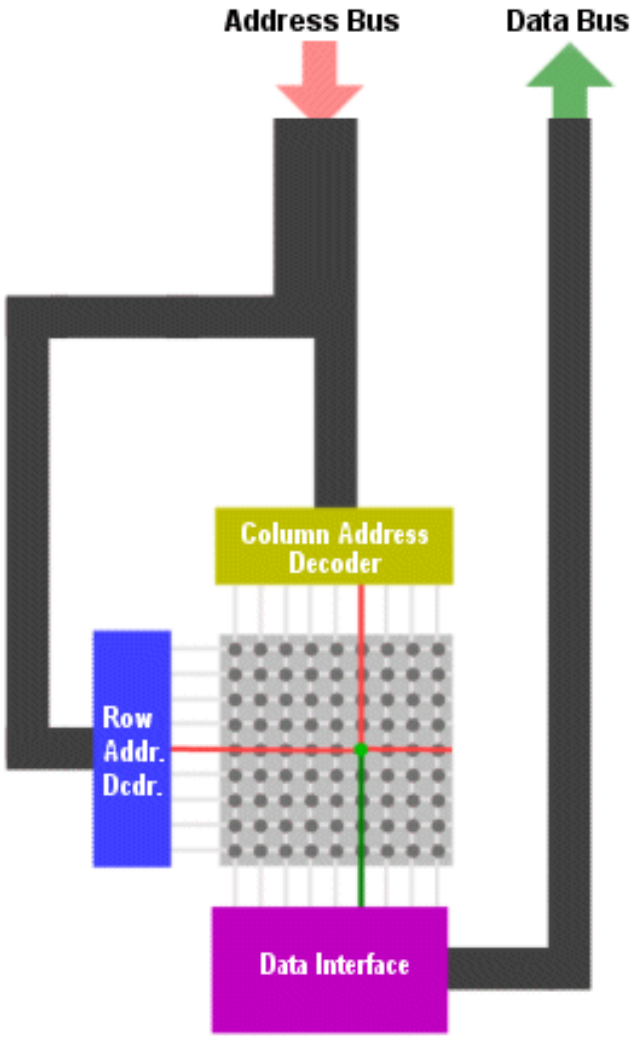


Controlador de memòria DRAM

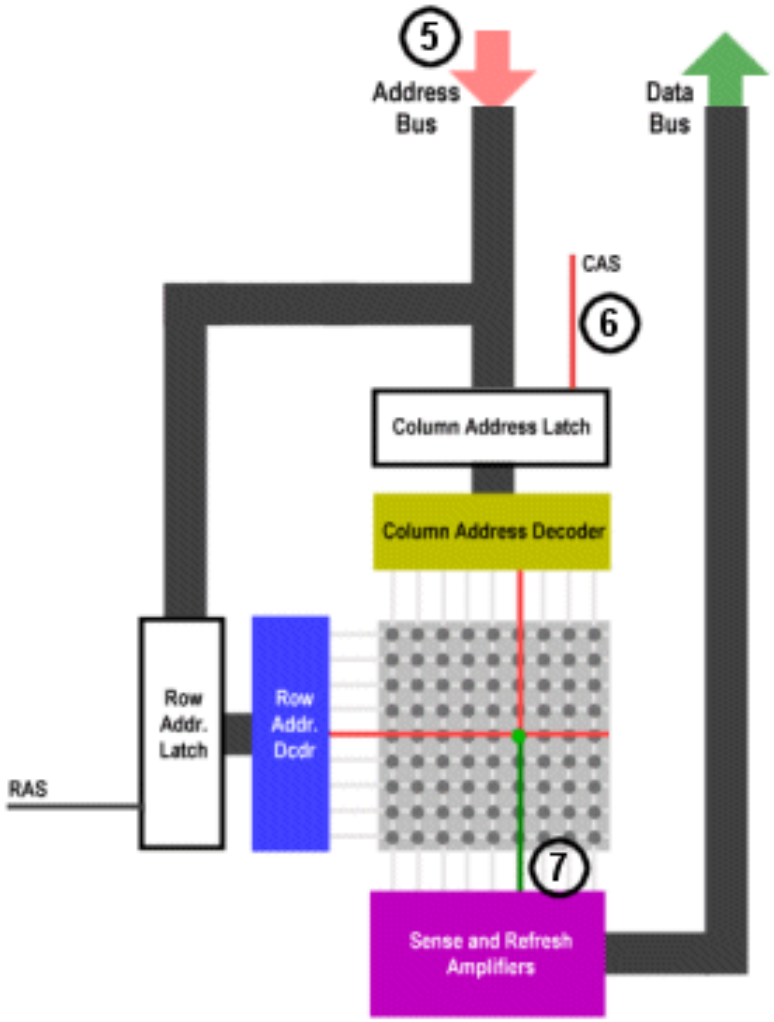
- El controlador s'encarrega del refresc i les particularitats de la DRAM.
- Amaga totes les particularitats al processador i li ofereix una interfície simple.
- El processador no depèn de la tecnologia de la memòria.



Organització interna de la memòria



Adreçament per fila/columna

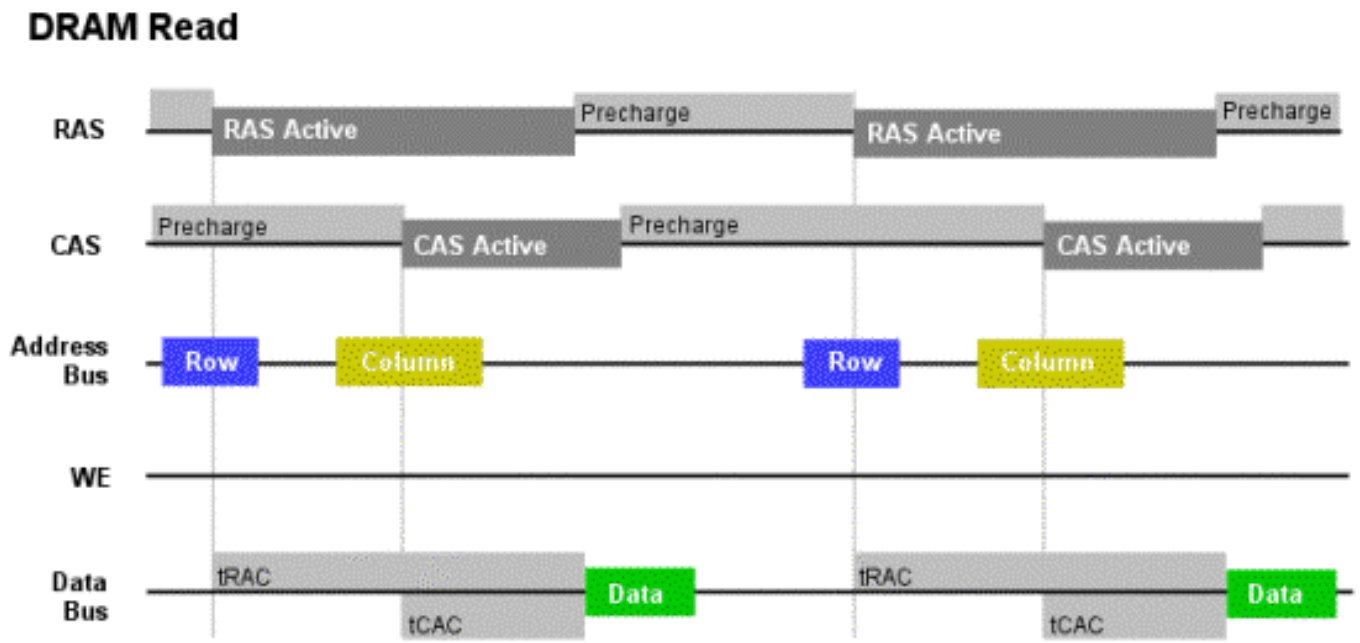


Adreçament per fila/columna amb RAS/CAS

Organització interna de la memòria

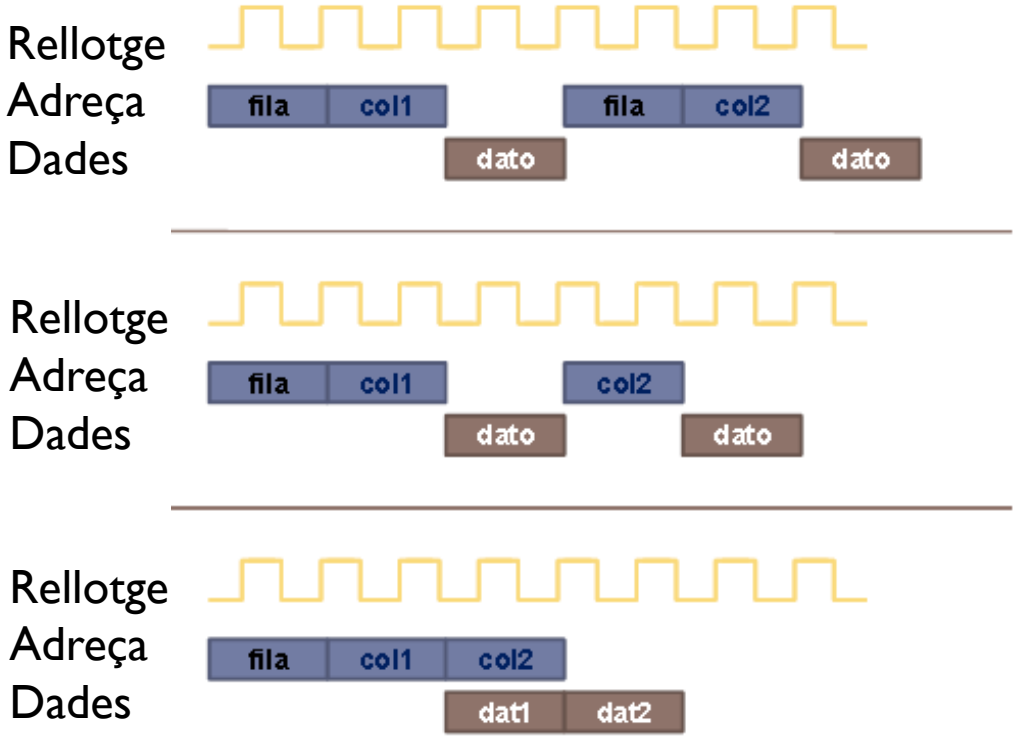
Lectura DRAM

1) L'adreça de la fila es col·loca als pins de direcció a través del bus d'adreces.
2) S'activa l'entrada RAS-> col·loca l'adreça de la fila al registre de files.
3) El descodificador de l'adreça de la fila selecciona la fila apropiada per a ser enviada als amplificadors de sortida
4) L'habilitació de l'escriptura està desactivat, de manera que la DRAM sap que no s'està fent una escriptura.
5) Es col·loca la columna als pins de l'adreça a través del bus d'adreces.
6) S'activa l'entrada CAS -> el que situa l'adreça de la columna al registre de columnes.
7) El CAS també serveix per habilitar la sortida. Quan s'estabilitza el senyal CAS els amplificadors de sortida reben les dades de la fila i la columna seleccionada per donar-la a bus de dades.
8) Es desactiva RAS i CAS per poder començar un nou cicle



Accessos típics

- Lectura DRAM



- FPM

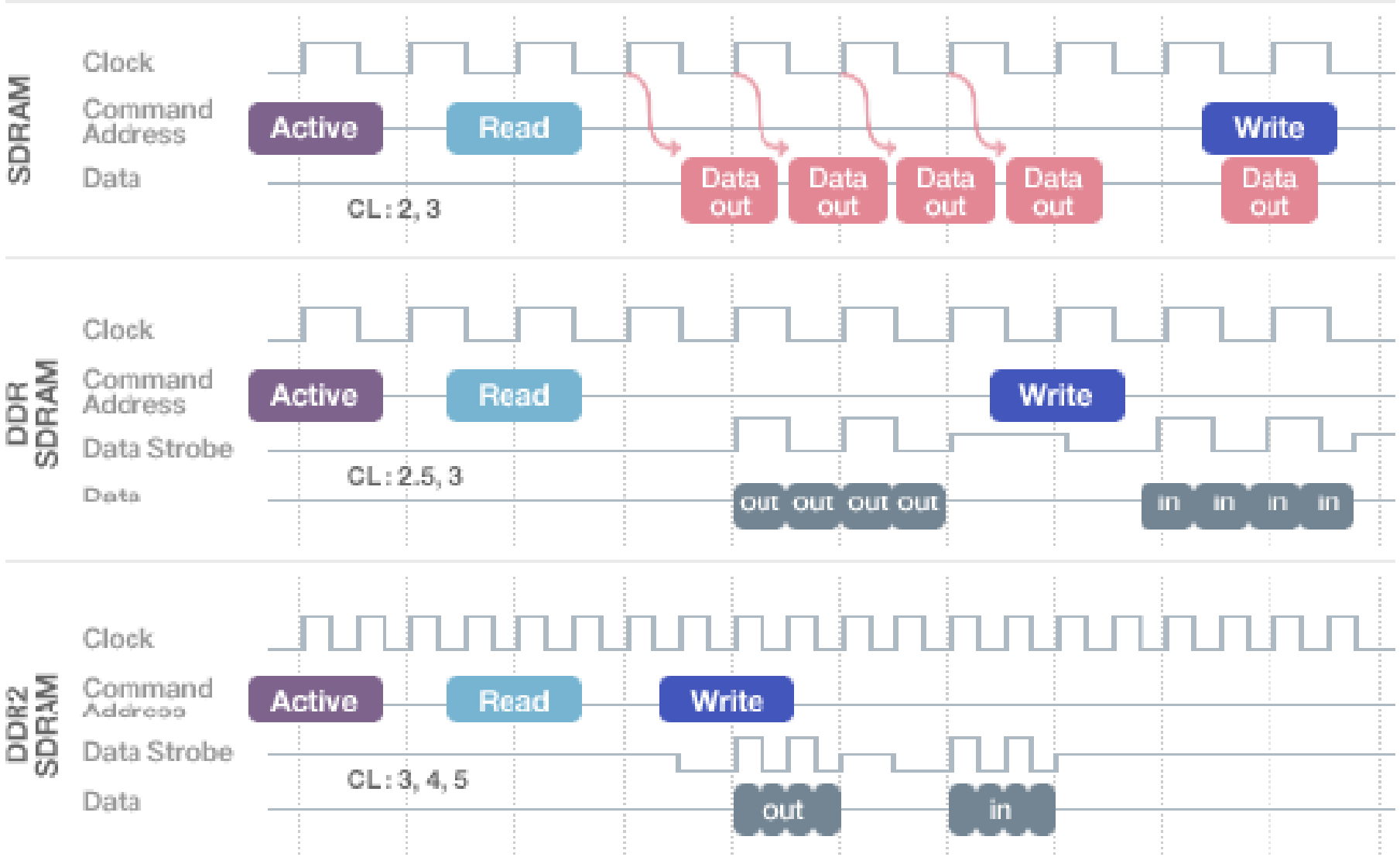
Fast Page Mode.
Paginat ràpid

- EDO

Extended Data
Output.

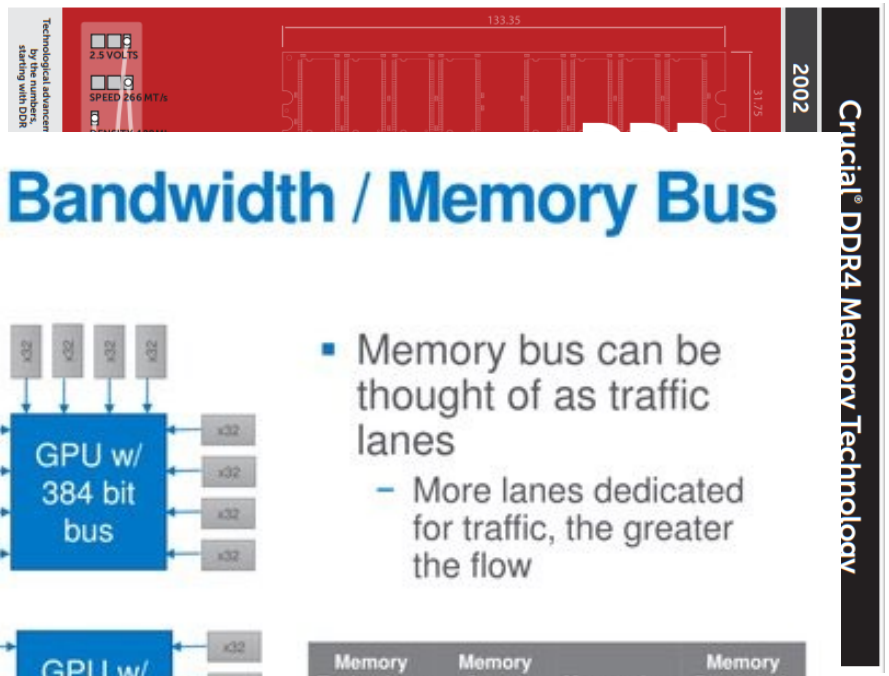
Mentre FPM només pot accedir a un sol valor cada cop, l'EDO pot moure un bloc complert a la memòria cau (es pot direccionar una posició i se'n pot llegir una altra)

Accessos típics



DDR : Doble taxa de dades (333 a 400 MHz)
DDR2 : Doble taxa de dades (533 a 1066 MHz)

Diferències



Memory Technology	Memory Speed	Memory bus	Memory Bandwidth
GDDR6	14 Gbps	384-bit	672 GB/s
GDDR5X	11 Gbps	384-bit	528 GB/s
GDDR5	7 Gbps	384-bit	336 GB/s
GDDR6	14 Gbps	256-bit	448 GB/s
GDDR5X	11 Gbps	256-bit	352 GB/s
GDDR5	7 Gbps	256-bit	224 GB/s
GDDR6	14 Gbps	192-bit	336 GB/s
GDDR5X	11 Gbps	192-bit	264 GB/s
GDDR5	7 Gbps	192-bit	168 GB/s

mes latència.

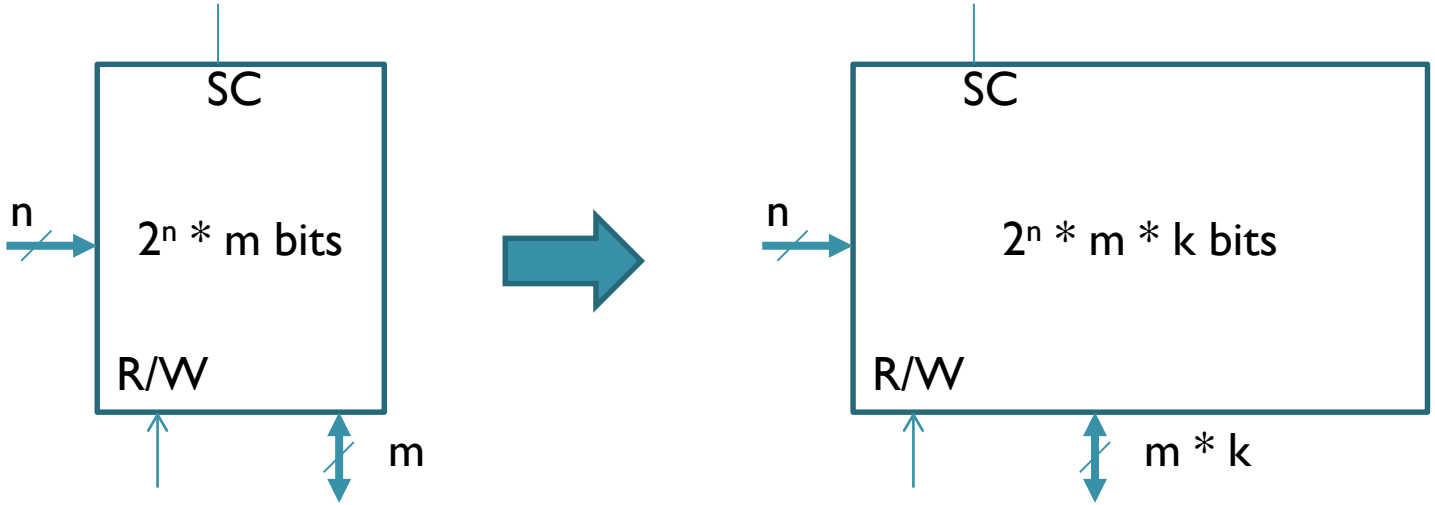
- DDR4: Velocitat entre 2 i 4 GHz. 1,05 V.
- DDR5: 6 GHz... Fins a 128 GB en un xip.... (2019)

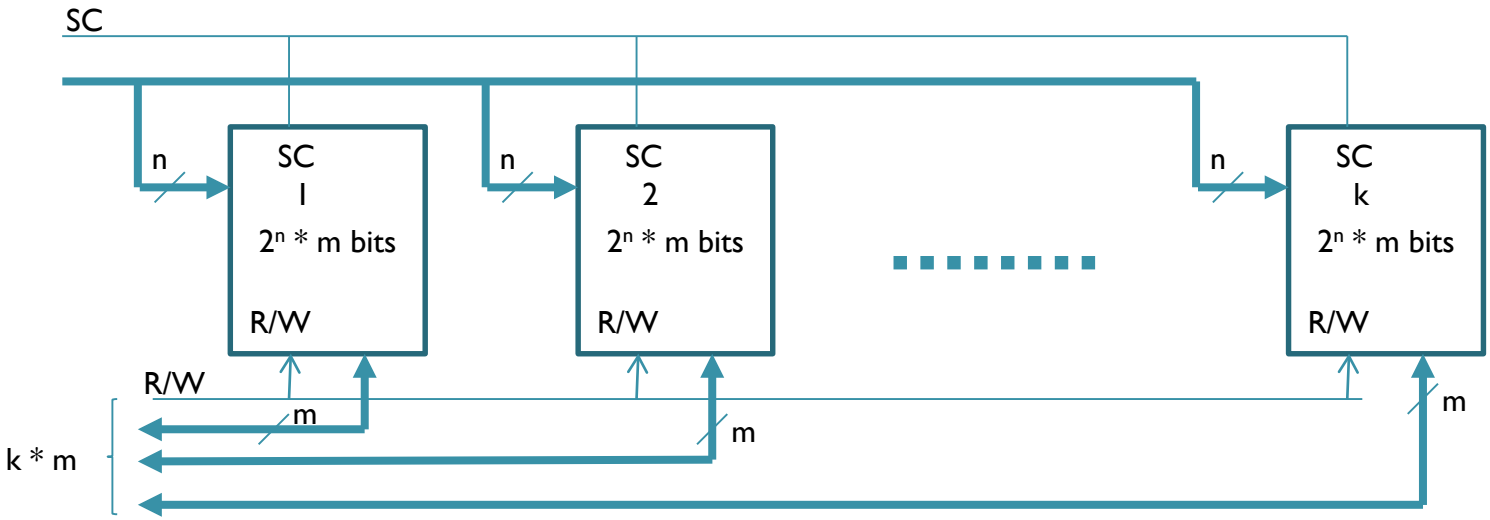
288 contactes

Disseny de memòries

- Ampliació del nombre de bits de la paraula de memòria

Es vol formar una memòria de $2^n * (m * k)$ bits a partir de xips de $2^n * m$ bits



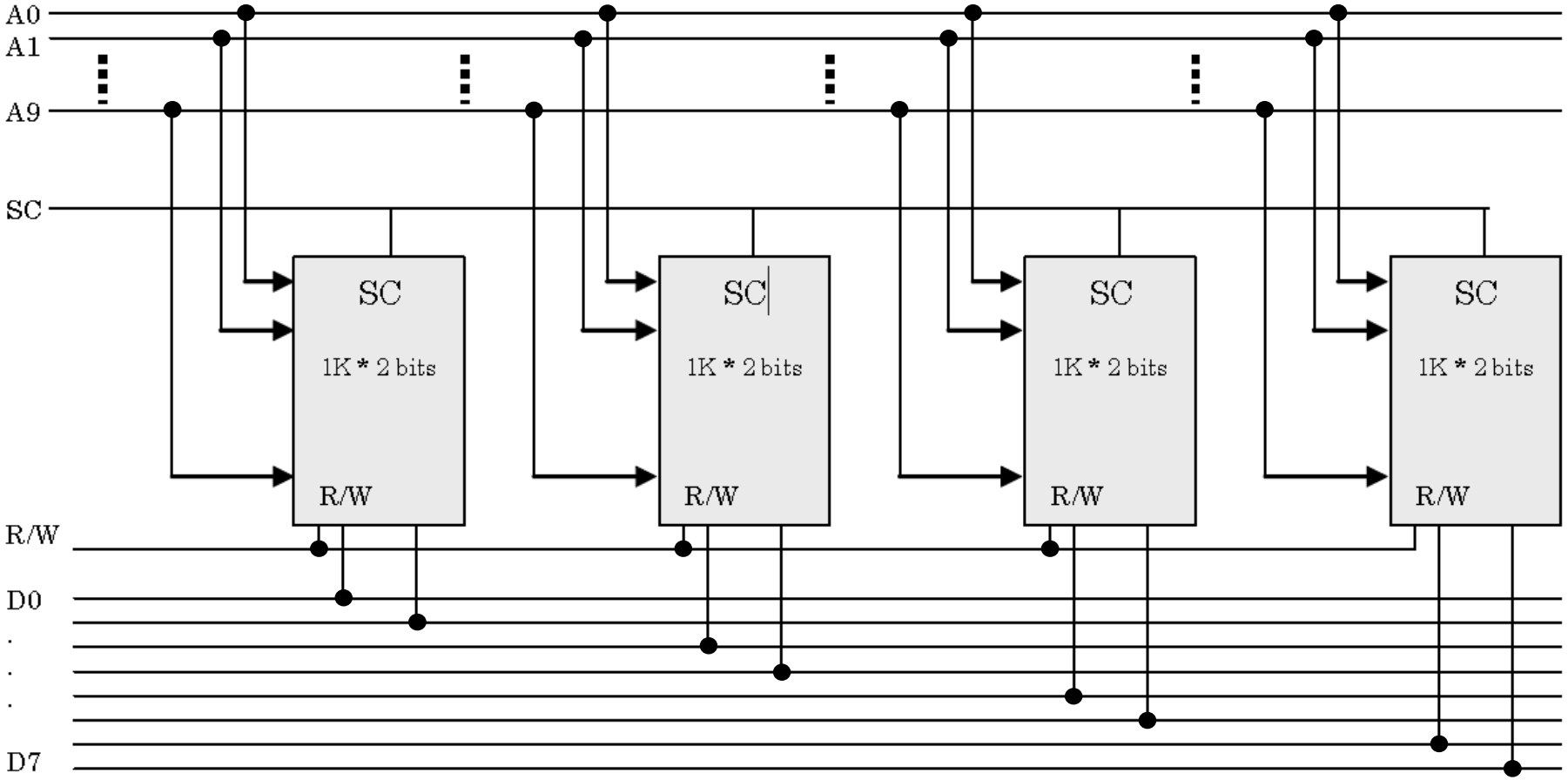


Disseny de memòries

- Ampliació del nombre de bits de la paraula de memòria

Disseny d'una memòria de 1 K * 8 amb mòduls de 1 K * 2 bits

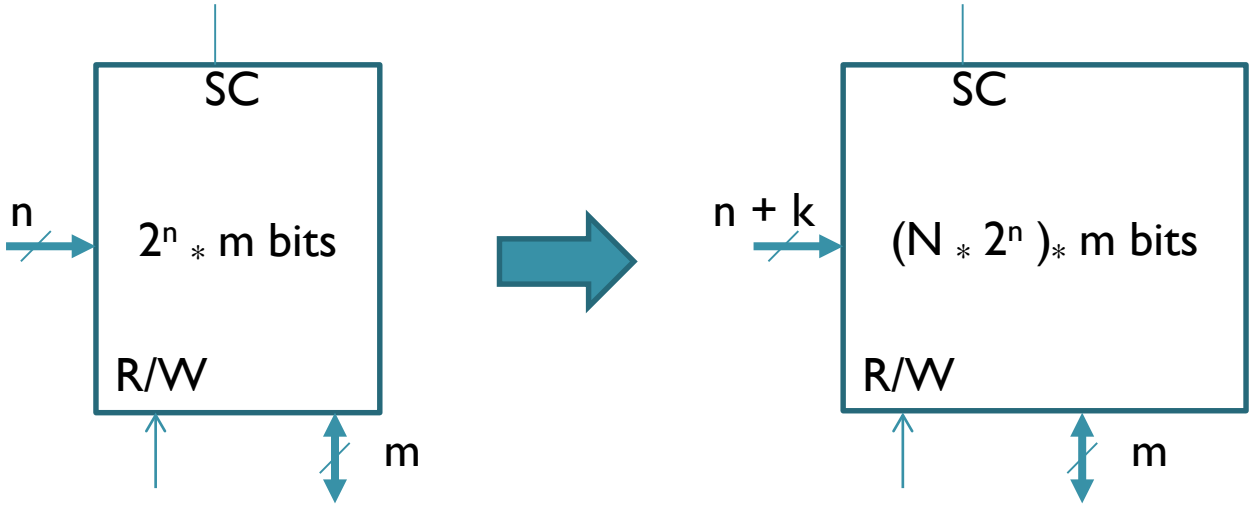
(1 K = 2^{10})

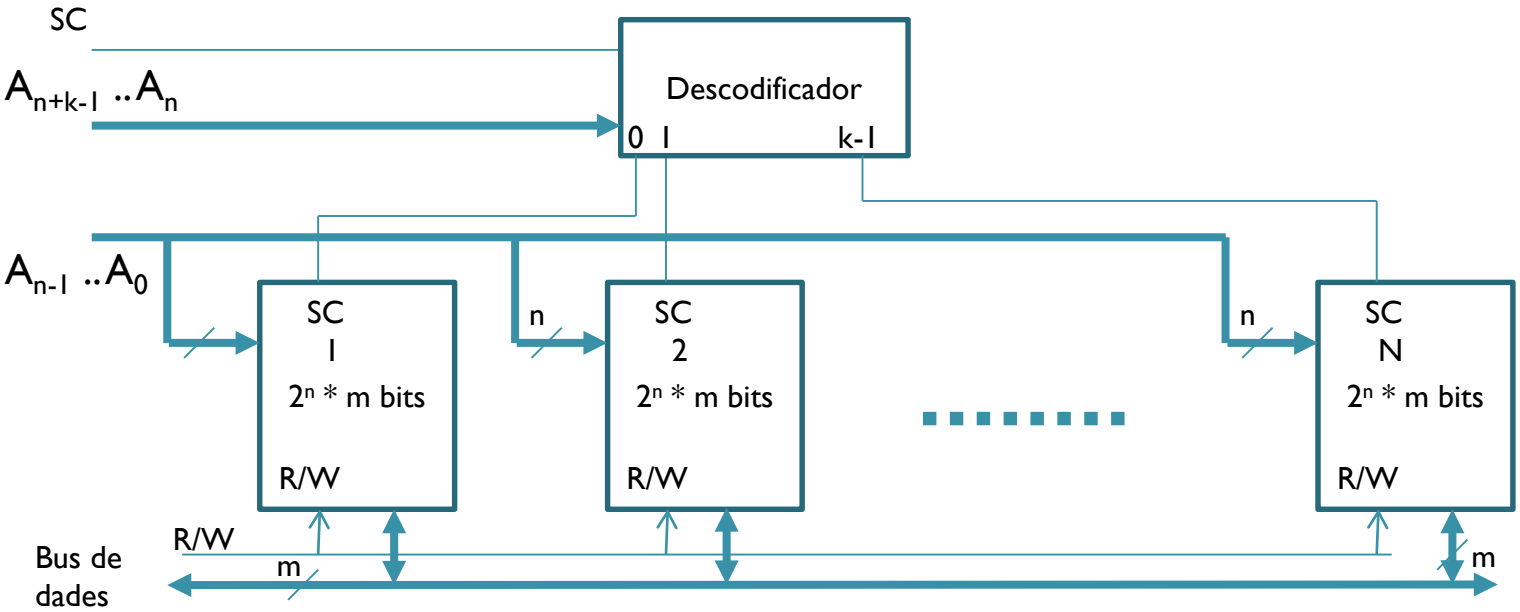


Disseny de memòries

- Ampliació de la capacitat de memòria

Es vol formar una memòria de N * 2^n * m bits a partir de xips de 2^n * m bits. (N = 2^K)

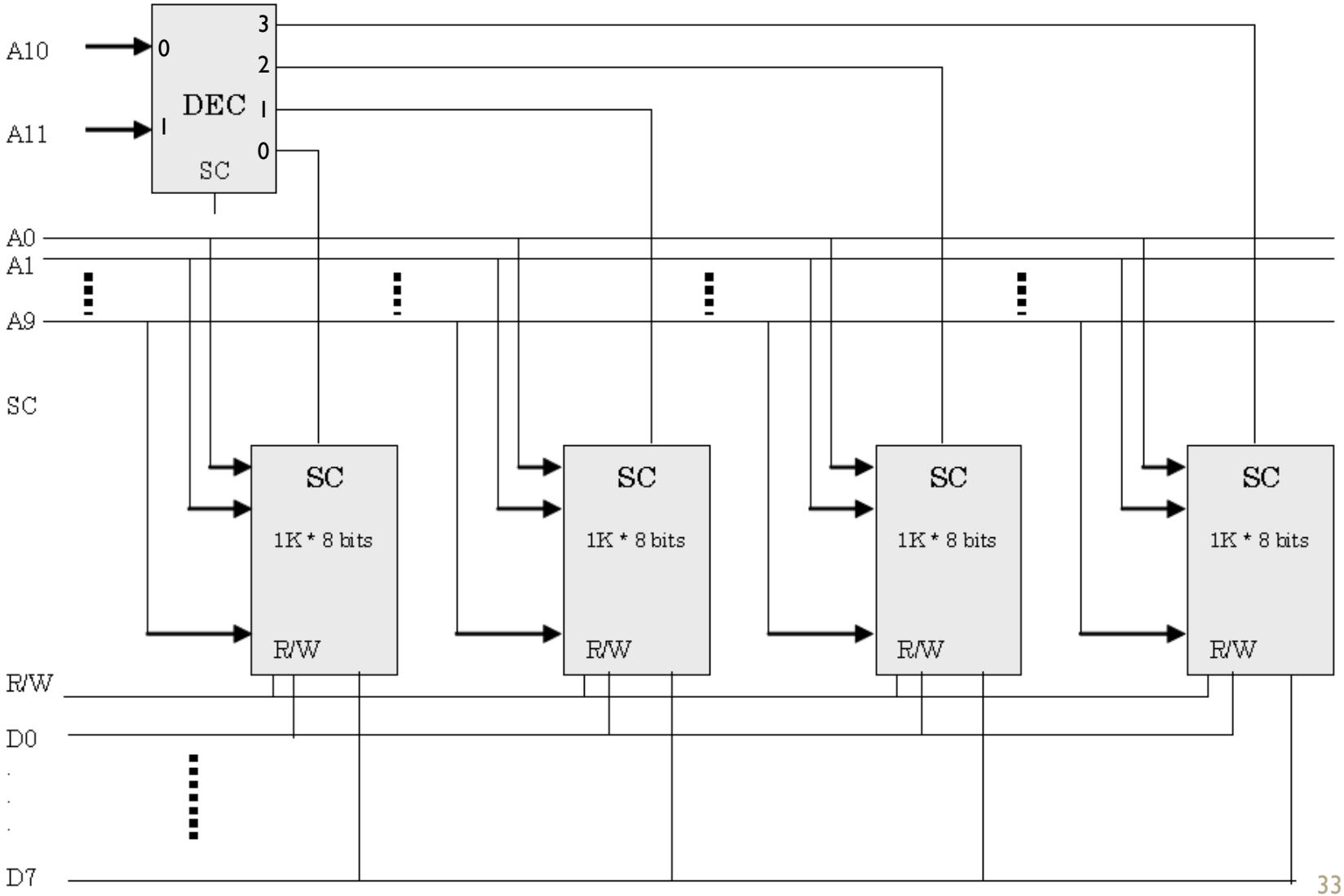




Disseny de memòries

- Ampliació de la capacitat de la memòria

Disseny d'una memòria de 4 K * 8 amb mòduls de 1 K * 8 bits



Problema

Memòria

- Xip de memòria de 256*4 bits
- Memòria RAM -> 1 K * 8
- Com es pot fer?

Arquitectura de computadors

Memòries

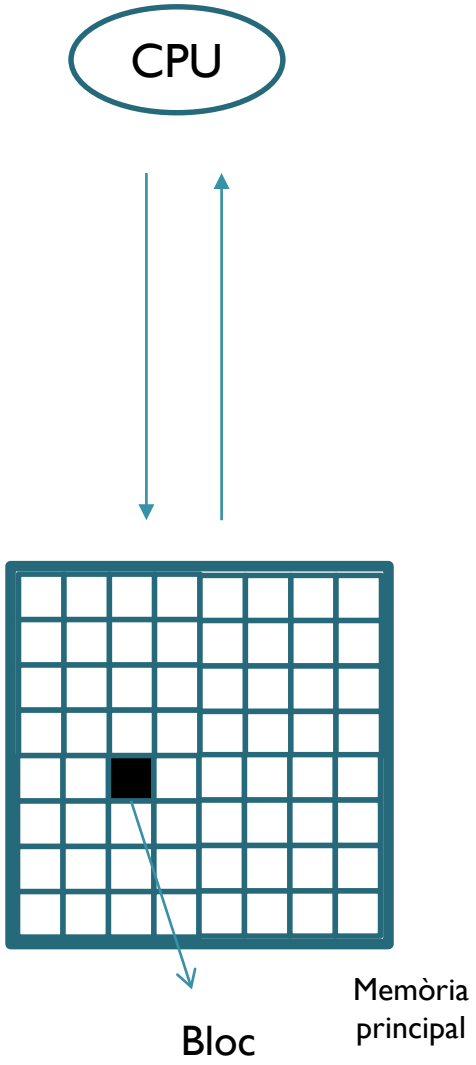
- Estructura jeràrquica de les memòries
- Memòria principal
- Memòria cau
- Memòria Virtual

Memòria cau

- Memòria petita i ràpida situada entre el processador i la memòria principal.
  - Emmagatzema una còpia de la porció de la informació que s'està utilitzant de la memòria
- L'objectiu és reduir el temps d'accés a memòria.
- L'estructura del sistema de memòria cau/principal:
  - Mp (memòria principal)
    - Formada per 2^n paraules adreçables
    - Dividida en nB blocs de 2^k paraules per bloc
  - Mc (memòria cau)
    - Formada per nM línies de 2^k paraules cadascun (nM << nB).
  - Directori (a la memòria cau)
    - Senyala quin subconjunt dels nB blocs estan situats a les nM línies de la cau
- Funcionament general
  - La CPU sol·licita continguts d'una posició de memòria
  - La Mc comprova si té les dades o instruccions

    Si els té: Els dona a la CPU

    Si no els té:
    - Es transfereix de la Mp el bloc associat a la posició
    - La Mc entrega la dada a la CPU

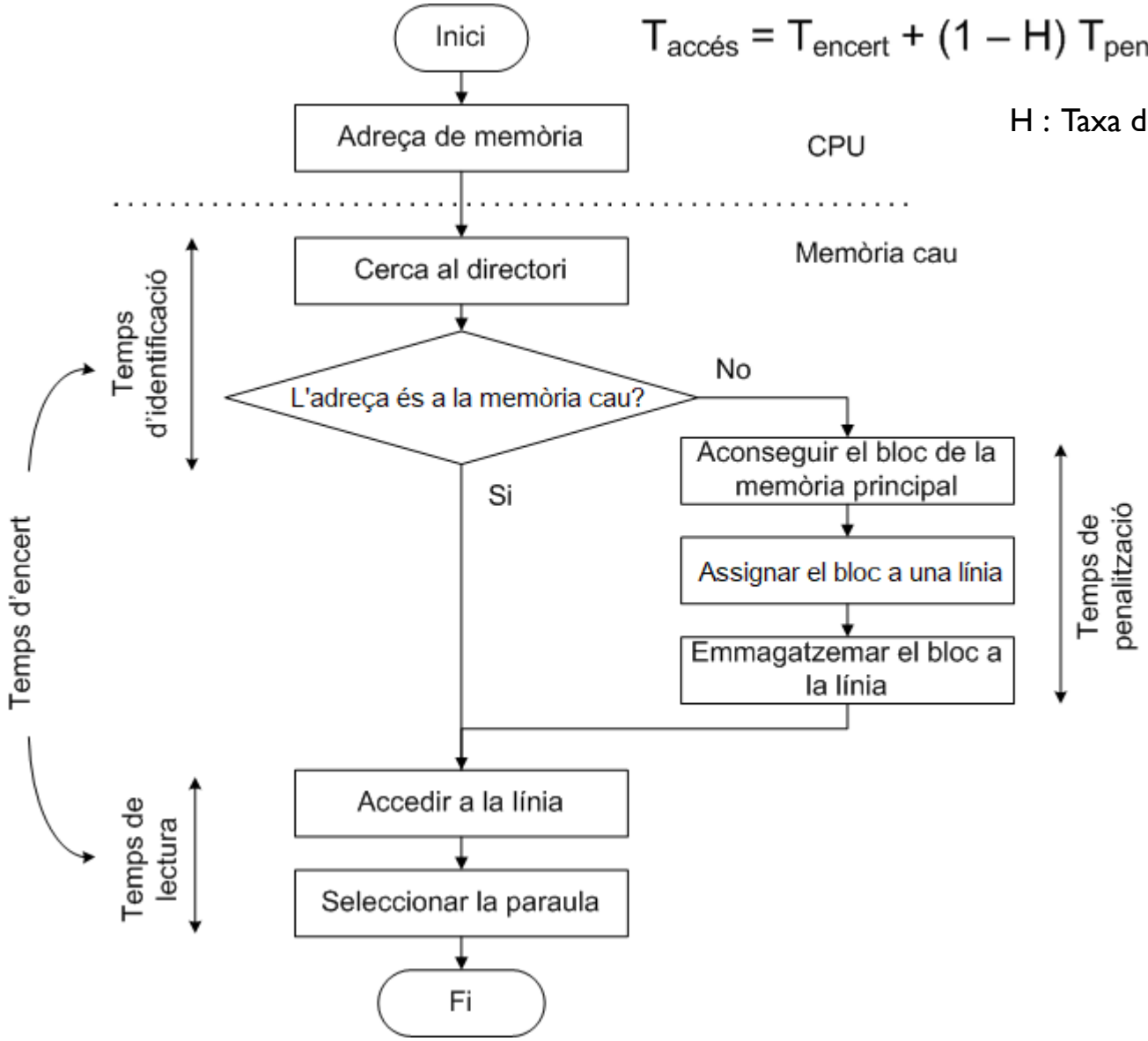


Memòria cau

Cicle d'accés

$$T_{accés} = T_{encert} + (1 - H)\, T_{penalització}$$

H : Taxa d'encert



Memòria cau

Disseny: Aspectes bàsics

- Organització:
  - Grandària de memòria
  - Dimensió de les línies
  - Quantitat de memòries cau
- Política d'ubicació: Al haver-hi menys línies a la memòria cau que blocs a la memòria principal es necessiten algoritmes que facin correspondre els blocs a les línies:
  - Directa
  - Associativa
  - Associativa per conjunts.
- Polítiques de substitució: Quan es porta un bloc nou, s'ha de substituir per un dels que hi ha. S'ha de determinar quin s'ha de treure.
- Política d'escriptura: Abans de substituir una línia de la memòria cau, s'ha de comprovar si s'ha modificat o no, per actualitzar-la o no a la memòria principal.
- Política de cerca: S'utilitza per decidir quan es transfereix un bloc o quins blocs s'han de transferir de la memòria principal a la memòria cau.

Memòria cau

Organització

- **Dimensions de la memòria**: com més gran és el sistema hi ha menor taxa d'errors però per contra, és més lent i més car. La mida estàndard sol estar entre 1K i 512K paraules .....
- **Mida del bloc**: a mesura que augmenta la grandària del bloc la taxa d'encerts primer augmenta a causa del principi de localitat, però després disminueix ja que cada paraula addicional està més lluny de la paraula requerida, i per tant és més improbable que sigui necessària a curt termini. La relació entre grandària de la línia i taxa d'encerts és complexa, depenent de les característiques de localitat de cada programa particular (entre 4 i 64 paraules).
- **Nombre de memòries caus**: amb l'augment de la densitat d'integració (llei de Moore) ha estat possible tenir una memòria cau en el mateix xip del processador: memòria cau on-xip. Això permet tenir un sistema basat en dos o més nivells de memòria cau:
  - Interna (L1): redueix el temps d'accés perquè s'elimina l'accés al bus. És petita ja que ha de cabre al costat de la CPU. Sol utilitzar emplaçament directe. (entre 8 i 128 KB)
  - Externa (L2): gran (entre 256 i 4MB) i sol utilitzar la correspondència associativa per conjunts.
- Pel que fa al seu contingut es sol separar la memòria cau en dues:
  - Cau d'instruccions (només lectura)
  - Cau de dades (lectura / escriptura)
  - Avantatge: duplica l'ample de banda.
  - Desavantatge: ofereix major taxa d'error que la memòria cau unificada.

Memòria cau

Polítiques d'ubicació

- La correspondència directa és la tècnica més simple i consisteix a fer correspondre cada bloc de memòria principal a una línia concreta de la memòria cau. La correspondència s'expressa com:

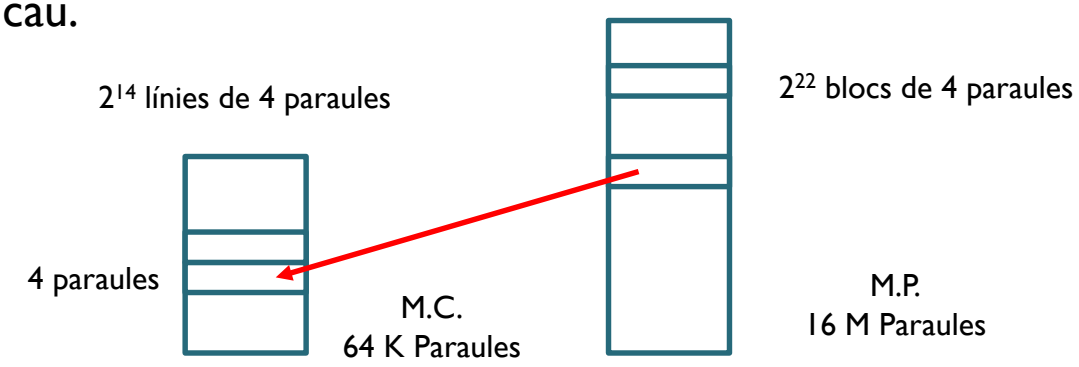
$$i = j \text{ mòdul } m$$

  - i = número de línia de la memòria cau.
  - j = nombre de bloc de memòria principal.
  - m = nombre línies de la memòria cau.

Adreça:

Etiqueta (s-r bits)	Línia (m bits)	Paraula (w bits)

  - Paraula: codifica el nombre de paraules de cada línia/bloc de memòria.
  - Línia: codifica el número de línia de la memòria cau.
  - Etiqueta: codifica el bloc de memòria associat a aquesta línia de la memòria cau.



Memòria cau

Polítiques d'ubicació

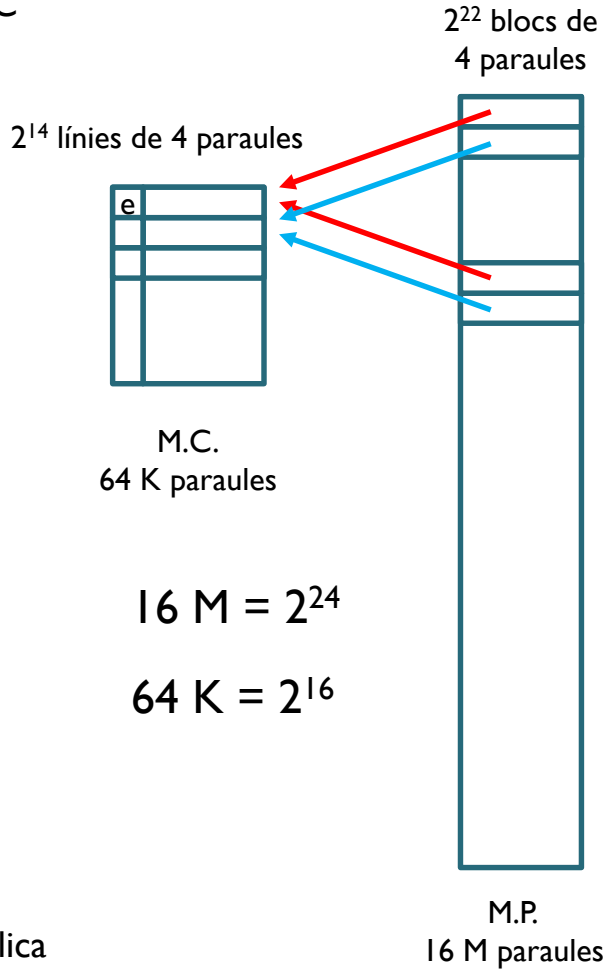
- **Directe**:
  - Cada bloc de MP li correspon una sola línia de la MC (sempre la mateixa)
  - L'adreça de MP s'interpreta com:

22-14 = 8 bits	14 bits	2 bits
Etiqueta	Línia (m)	Paraula

  - Si a la línia hi ha l'etiqueta (e) llavors el bloc és a la memòria cau.
  - Ex:

Línia cau	Bloc de la MP
0	$0, 2^{14}, 2*2^{14}, 3*2^{14}, ...$
1	$1, 2^{14}+1, 2*2^{14}+1, 3*2^{14}+1, ...$
..	....

  - Avantatges: Simple, econòmica i ràpida
    Directori petit
  - Inconvenients: Un patró d'accessos dolent, que implica més fallades. Taxa de fallades alta si varis blocs competeixen per la mateixa línia.

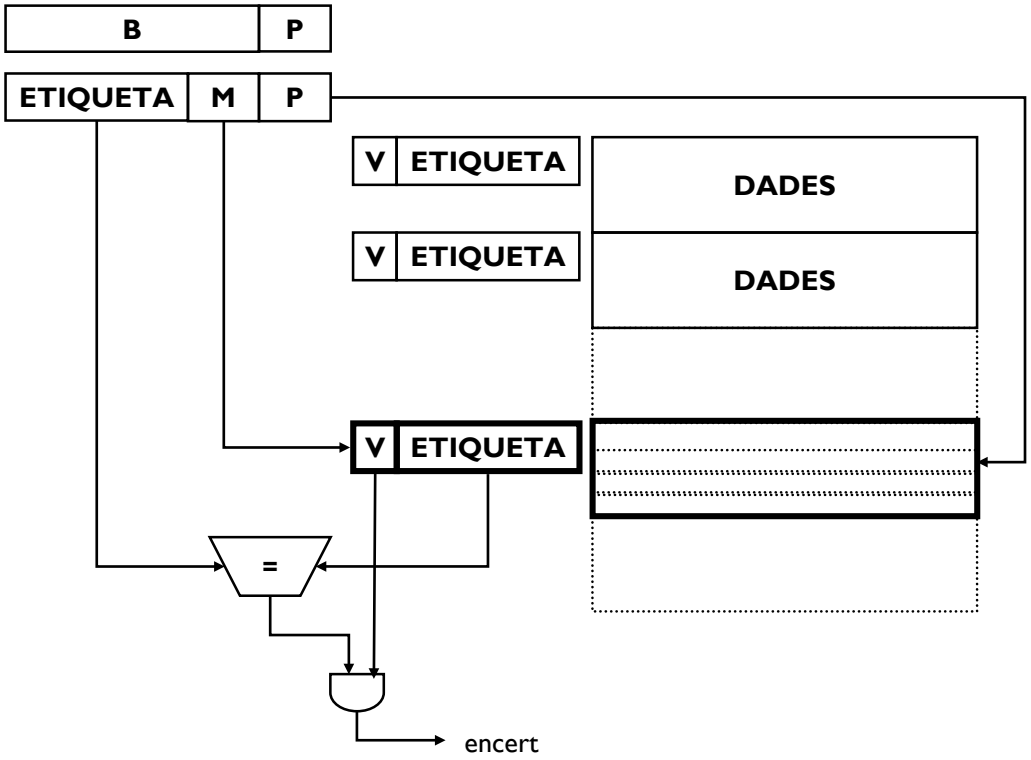


$16\ M = 2^{24}$

$64\ K = 2^{16}$

Memòria cau

Polítiques d'ubicació: Directe

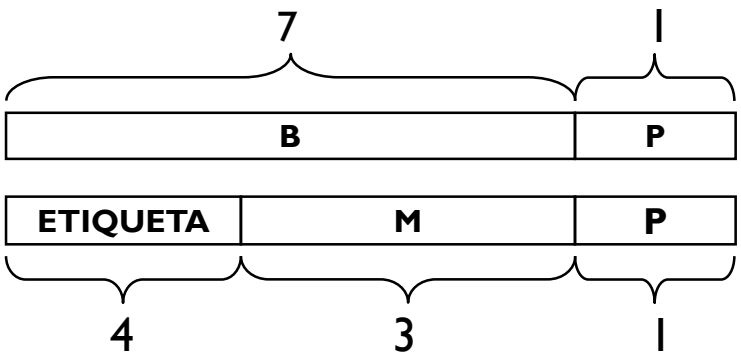


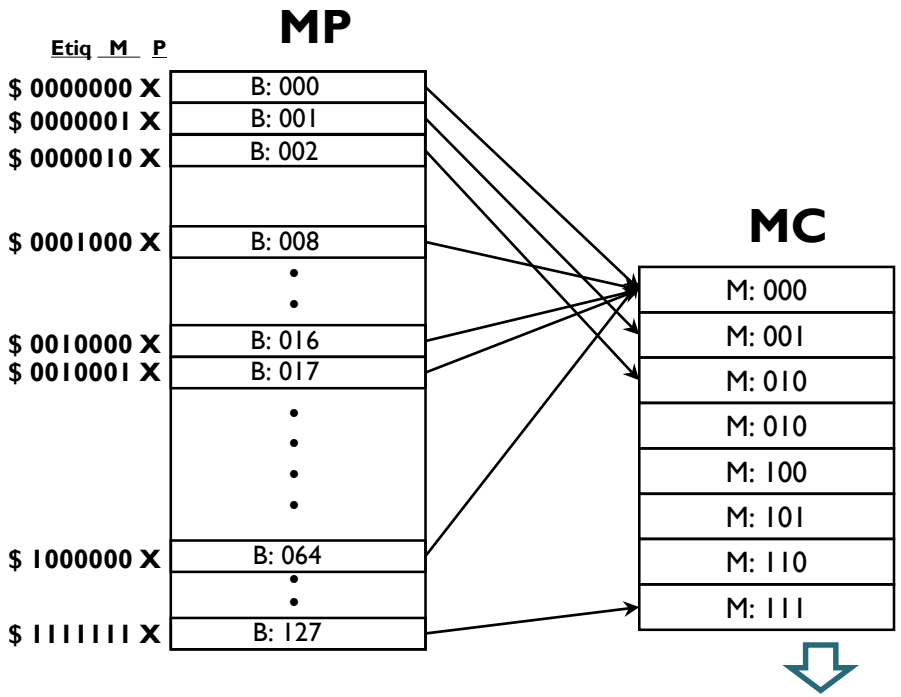
Correspondència Directe: exemple

- *Memòria principal*
  - 256 paraules (@ amb 8 bits)
- *Memòria cau*
  - 16 paraules (@ amb 4 bits)
- *Dimensió del bloc*: 2 paraules
  - Nombre de blocs a MP
    - ⇨ nB = 128 blocs (@ 7 bits)
  - Nombre de línies a MC
    - ⇨ nM = 8 línies (@ 3 bits)

Adreces de la CPU	*línia*
XXXX000X	0
XXXX001X	1
XXXX010X	2
XXXX011X	3
XXXX100X	4
XXXX101X	5
XXXX110X	6
XXXX111X	7

Informació que s'emmagatzema a l'etiqueta de la línia





Memòria cau

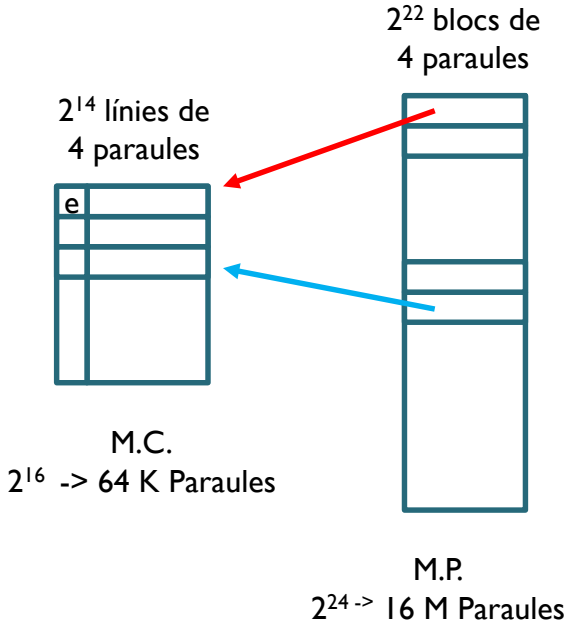
Polítiques d'ubicació

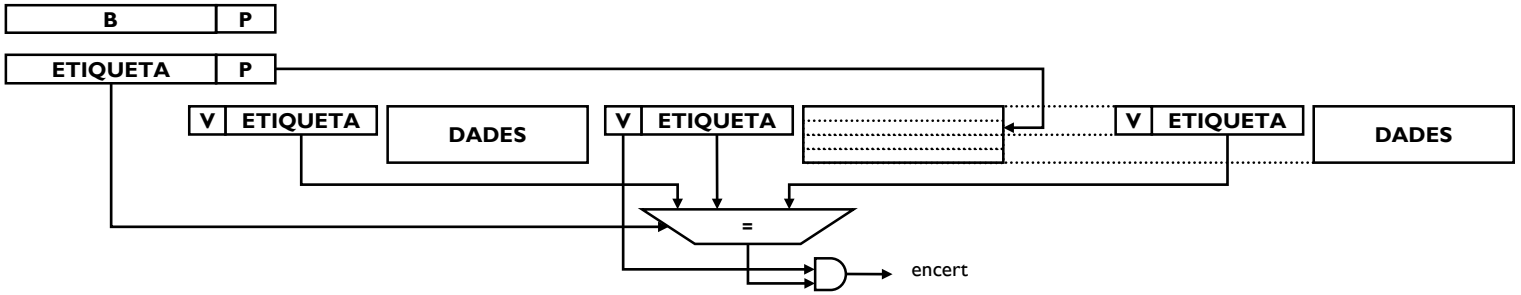
- Associativa:
  - Qualsevol bloc de MP pot carregar-se a qualsevol línia de la MC.
  - L'adreça de MP s'interpreta com:

  - Si hi ha una línia amb "etiqueta" a la memòria cau llavors hi ha el bloc.

⌦ *Avantatges*:
- ➢ baixa taxa de falles

⌦ *Inconvenients*:
- ➢ alta complexitat del maquinari
- ➢ Lentitud en la identificació
- ➢ directori gran



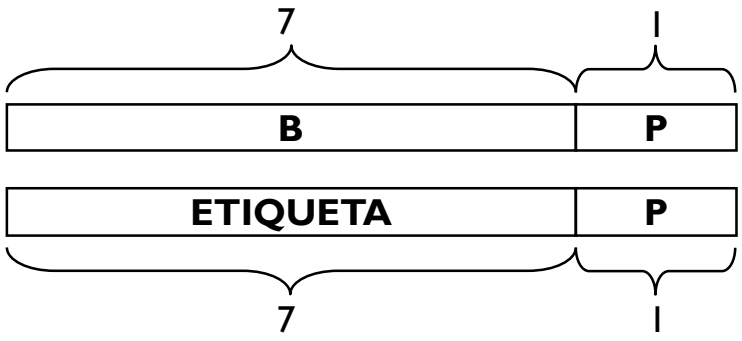


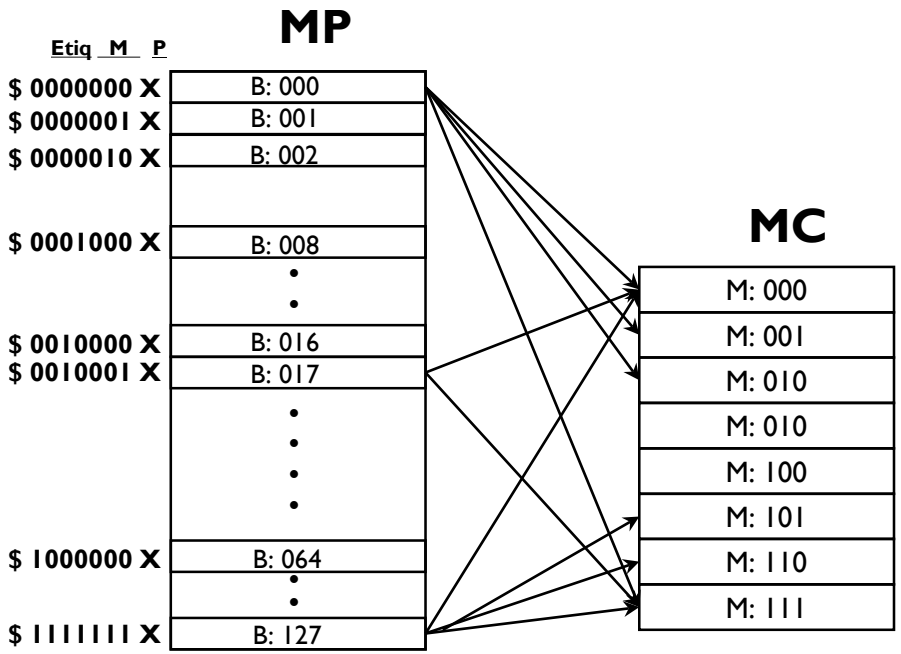
Correspondència associativa: exemple

- *Memòria principal*
  - 256 paraules (@ 8 bits)
- *Memòria cau*
  - 16 paraules (@ 4 bits)
- *Dimensió del bloc*: 2 paraules
  - Nombre de blocs a MP
    - nB = 128 blocs (@ 7 bits)
  - Nombre de línies a MC
    - nM = 8 línies (@ 3 bits)

Adreces de la CPU	*Línia*
XXXXXXX X	Qualsevol

Informació que s'emmagatzema a l'etiqueta de la línia





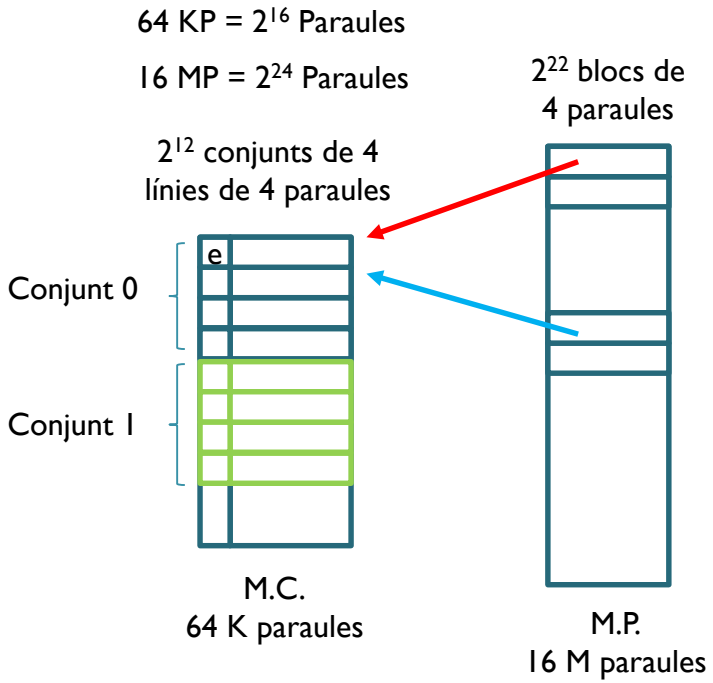
Memòria cau

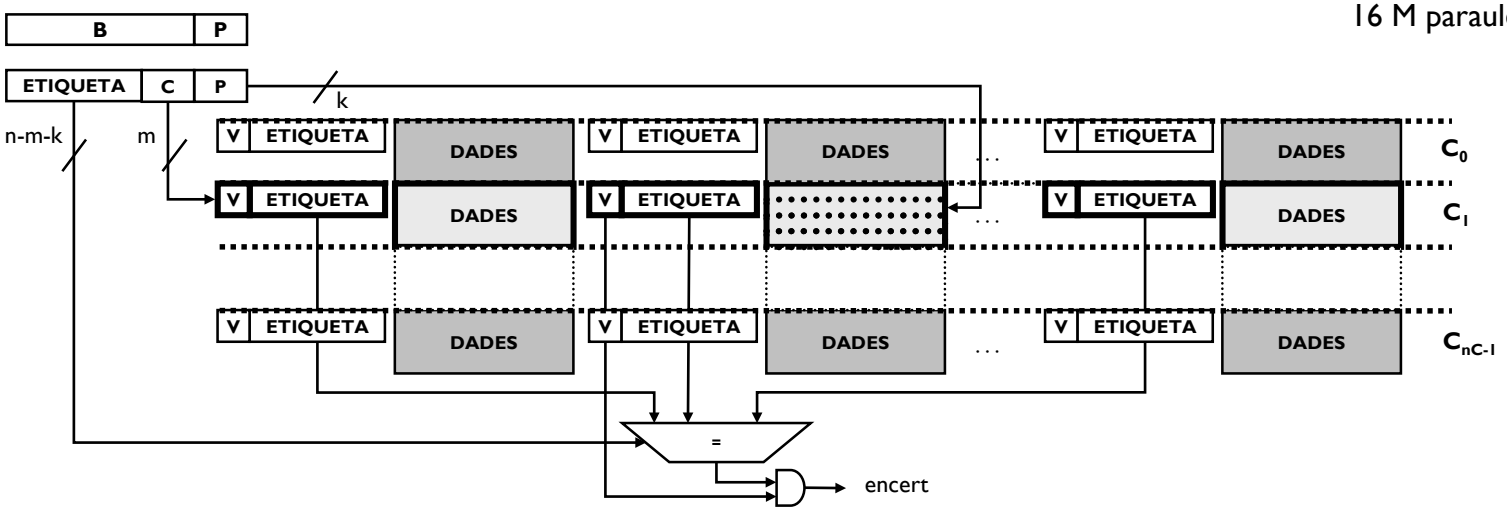
Polítiques d'ubicació

- Associativa per conjunts:
  - Un bloc de la MP pot carregar-se a qualsevol línia de la MC d'un conjunt determinat.
  - L'adreça de MP s'interpreta com:

10 bits	12 bits	2 bits
Etiqueta (e)	Conjunt (c)	Paraula

  - Si hi ha una línia amb l'etiqueta 'e' dins del conjunt 'c' llavors hi ha el bloc a la memòria cau.



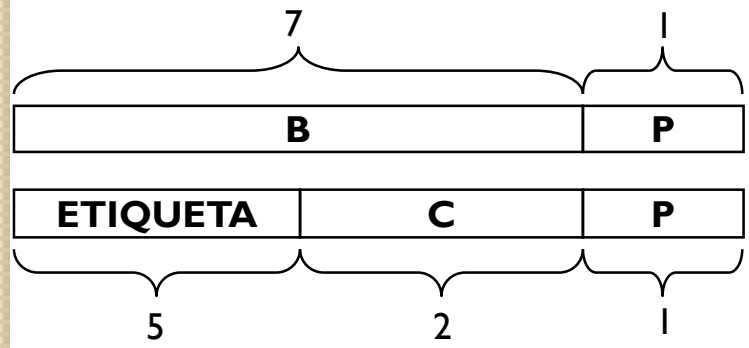


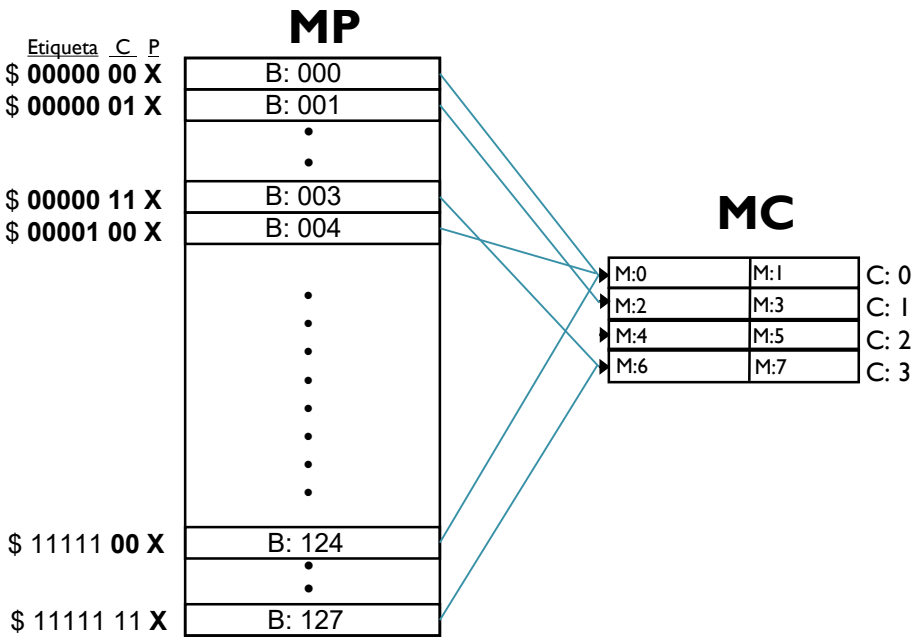
Correspondència associativa per conjunts: exemple

- *Memòria principal*
  - 256 paraules (@ 8 bits)
- *Memòria cau*
  - 16 paraules (@ 4 bits)
- *Dimensió del bloc*: 2 paraules
  - Nombre de blocs a MP
    - ⇨ nB = 128 blocs (@ 7 bits)
  - Nombre de línies a MC
    - ⇨ nM = 8 línies (@ 3 bits)
- Vies: 2
  - Nombre de línies per conjunt = 2
  - Nombre de conjunts a la MC
    - ⇨ nC = 4 (2 bits)
    - ⇨ nM / nC = 8 / 4 = 2

Adreces de la CPU	*Conjunt*	*Línia*
XXXXX 00 X	0	qualsevol de 2
XXXXX 01 X	1	qualsevol de 2
XXXXX 10 X	2	qualsevol de 2
XXXXX 11 X	3	qualsevol de 2

Informació que s'emmagatzema a l'etiqueta de la línia





Memòria cau

Polítiques d'ubicació

- Associativa per conjunts:

- *Avantatges*:
  - Es un entremig entre emplaçament directe i associatiu

- El valor nM/nC (nombre de línies de la cau / nombre de conjunts de la cau) es denomina **grau d'associativitat** o **nombre de vies** de la MC
  - Grau d'associativitat = 1, equival a l'emplaçament directe
  - Grau d'associativitat = nM, equival a l'emplaçament associatiu

- El grau d'associativitat afecta el rendiment de la política d'emplaçament
  - Un grau d'associativitat alt fa que disminueixen les fallades per competència de la línia
  - En augmentar el grau d'associativitat augmenta el temps d'accés i el cost del maquinari
  - *Grau òptim*: entre 2 i 16
  - *Grau més comú*: 2

Memòria cau

Polítiques de substitució

- *Que fer quan es produeix una fallada i totes les línies estan ocupats?*
  - És necessari elegir-ne una i sobreescriure el nou bloc
- **Espai de substitució**: conjunt de possibles línies que poden ser substituïdes pel nou bloc
  - *Directe*: a la línia que el bloc té assignat. No es necessita cap algoritme
  - *Associatiu*: qualsevol línia de la memòria cau.
  - *Associatiu por conjunts*: qualsevol línia que estigui al conjunt que té assignat el bloc.

- **Algoritmes** (implementats en maquinari):
  - **Aleatori**: s'escull una línia de l'espai de substitució a l'atzar.
  - **FIFO**: es substitueix la línia de l'espai de substitució que porti més temps carregat.
  - **LRU *(least recently used)***: es substitueix la línia de l'espai de substitució que porti més temps sense que s'hagi fet cap referència. -> Es necessita un registre d'edat.
  - **LFU (*least frequently used*):** es substitueix la línia de l'espai de substitució que hagi tingut menys referències. -> Es necessita un registre d'us.

Memòria cau

Política d'escriptura

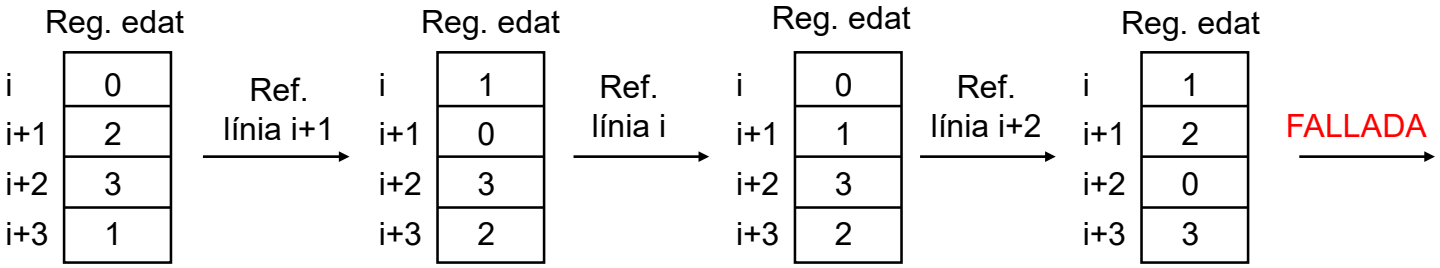
- *Que succeeix quan la CPU escriu una paraula a memòria?*
  - Si s'escriu sobre un dels blocs carregats a la cau, el contingut de la Mc i de la Mp no coincideixen ⇒ es necessari actualitzar el bloc corresponent de la Mp.
- **Escriptura immediata (*write-through*):** cada vegada que es fa una escriptura a la Mc s'actualitza immediatament a la MP.
  - *Variants* en funció del que succeeix en cas de fallada d'escriptura:
    - **Amb assignació en escriptura**: un bloc es carrega a la Mc com a conseqüència de fallades de lectura i escriptura.
    - **Sense assignació en escriptura**: un bloc es carrega a la Mc només com a conseqüència de fallades de lectura, les fallades d'escriptura no carreguen blocs i accedeixen directament a la Mp.
  - *Avantatges:* cost reduït del maquinari i consistència en tot moment
  - *Desavantatges:* augmenta el tràfic entre la Mc i la Mp.
  - Per evitar que el processador s'hagi d'esperar a que es realitzi l'actualització, s'utilitza un buffer intermedi (4 referències)
- **Post-escriptura (*copy-back*):** la Mp s'actualitza només quan es substitueix el bloc
  - A la Mc es carreguen blocs tan per fallades de lectura com d'escriptura.
  - S'utilitza un **bit d'actualització** per blocs que indica si s'ha d'actualitzar a la Mp.
    - Quan es realitza una escriptura s'activa
    - Quan es realitza una substitució es comprova si està activat o no.
  - *Avantatges:* disminueix el tràfic entre la Mc i la Mp i disminueix el temps d'accés per l'escriptura
  - *Desavantatges:* inconsistència
  - Per evitar retards en les substitucions s'utilitza un buffer intermedi (1 bloc)

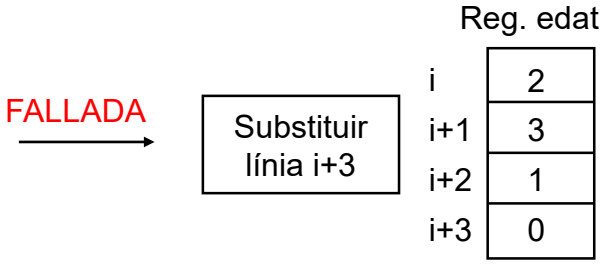
Memòria cau

Política de cerca

- **Cerca per demanda:** un bloc es porta a la Mc quan es necessita, és a dir, com a conseqüència d'una fallada.
- **Cerca anticipada:** un bloc es porta a la Mc abans de que sigui necessari per reduir la taxa de fallades.
  - El més comú és triar el següent bloc al que es fa referència (***one block lookahead***).
  - *Variants*:
    - **Cerca prèvia per fallada**: si es produeix una fallada a l'accedir a un bloc es busca aquest bloc i el següent.
      - **Exemple:** El bloc k produeix una fallada ⇒ portar a la Mc els blocs k i k+1
    - **Cerca prèvia sempre**: el primer cop que es fa referència a un bloc es busca el següent.
      - **Exemple:** referència al bloc k ⇒ portar a la Mc el bloc k+1
        referència al bloc k+1 ⇒ portar a la Mc el bloc k+2

Exemple: 4 blocs per conjunts ⇒ registre d'edat de 2 bits

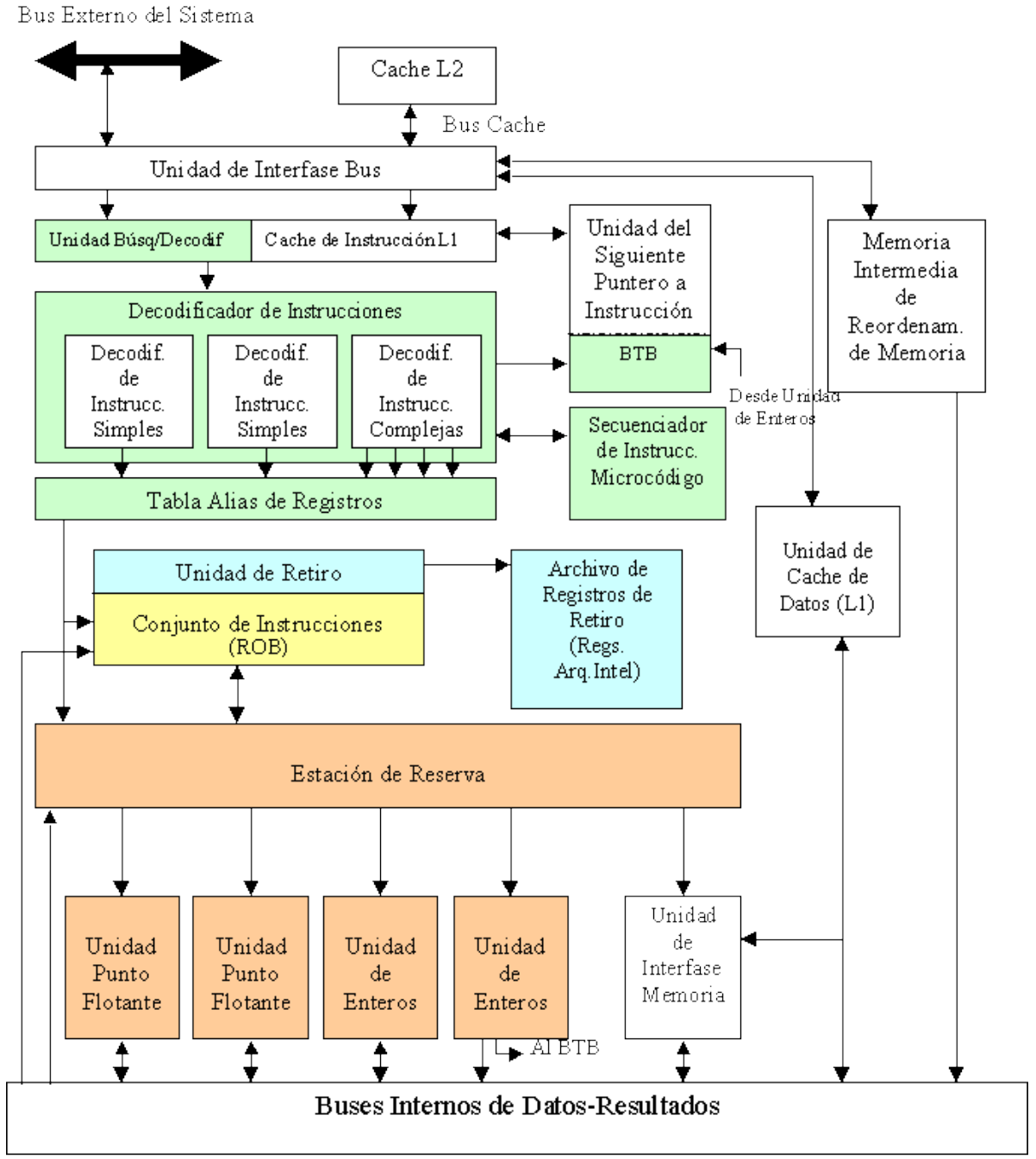




Exemple: memòria cau en un pentium

- **2 nivells de cau**
  - 2 caus internes, una de dades i una d'instruccions
  - 1 cau externa
- **Cau interna:**
  - *Grandària*: 8 Kb
  - *Grandària del bloc*: 32 bytes
  - *política d'ubicació*: associativa per conjunts de 2 vies
  - *política de substitució*: LRU (edat)
  - *política d'escriptura (per la de dades)*: post-escriptura o immediata (seleccionable)
- **Cau externa:**
  - *Grandària*: 256 o 512 Kb
  - *Grandària del bloc*: 32, 64 o 128 bytes
  - *política d'ubicació*: associativa per conjunts de 2 vies
  - *política de substitució*: LRU
  - *política d'escriptura* : post-escriptura
- **Control de la cau:**
  - En el registre d'estat hi ha 2 bits per controlar la cau interna
    - **CD** (*cache disable*): per inhabilitar la cau
    - **NW** (*not write throught*): per seleccionar la política d'escriptura
  - En el repertori d'instruccions hi ha 2 instruccions:
    - **INDV**: neteja la memòria cau
    - **WBINVD**: neteja la memòria cau, actualitzant prèviament la memòria principal

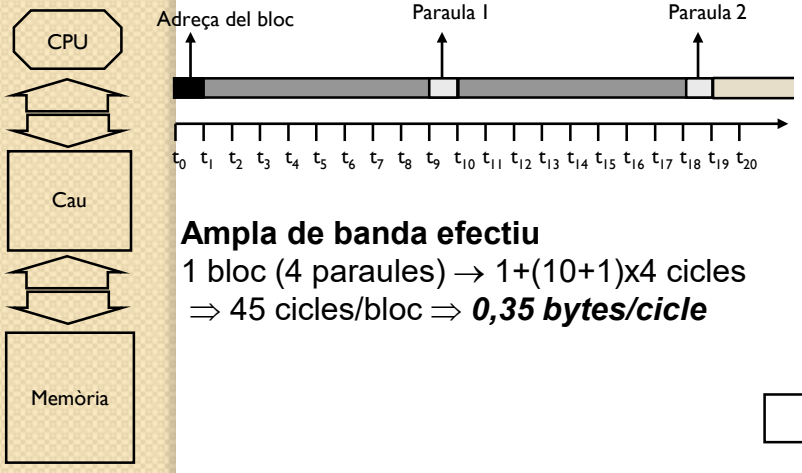
Exemple: memòria cau en un pentium



Memòria entrellaçada

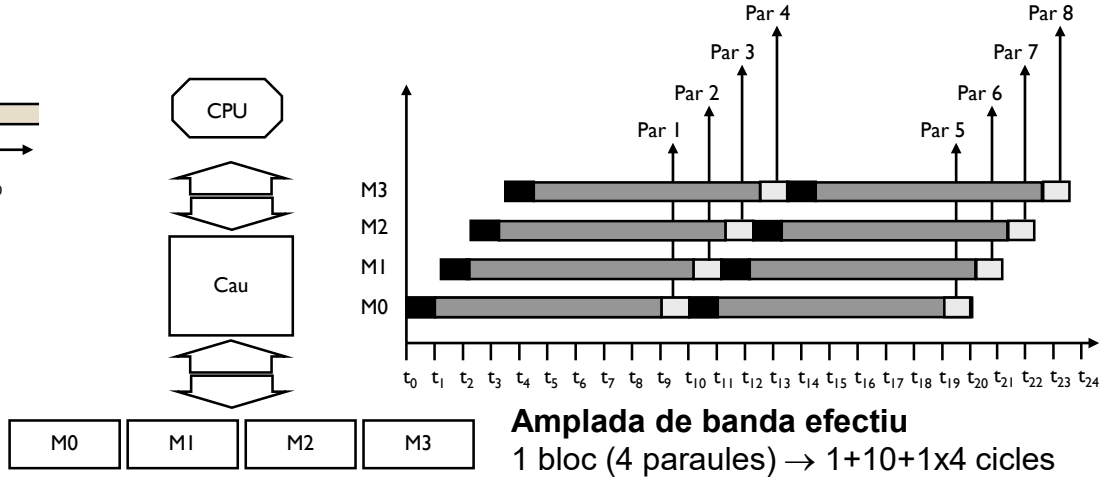
- Per accelerar la transferència de blocs entre la MP i la MC es poden utilitzar varies tècniques:
  - Organització avançada de la DRAM (EDRAM, EDO RAM, SDRAM, etc.)
  - Memòria entrellaçada
- La memòria entrellaçada és un mètode per reduir el temps d'accés efectiu a la MP.
  - Els mòduls de la MP s'organitzen de manera que es pot accedir en paral·lel a totes les dades.
- *Exemple*:
  - MP: 1 cicle per enviar l'adreça, 10 cicles per l'accés i 1 cicle per enviar la dada
  - MC: Blocs de 4 paraules (1 paraula = 4 bytes) (1 Bloc = 16 bytes)

Amplada del BUS: 1 PARAULA
Amplada de la MEMÒRIA: 1 PARAULA
ACCÉS SEQÜENCIAL



Ampla de banda efectiu
1 bloc (4 paraules) → 1+(10+1)x4 cicles
⇒ 45 cicles/bloc ⇒ ***0,35 bytes/cicle***

Amplada del BUS: 1 PARAULA
Amplada de la MEMÒRIA: 4 PARAULES
ACCÉS SOLAPAT ALS MÒDULS

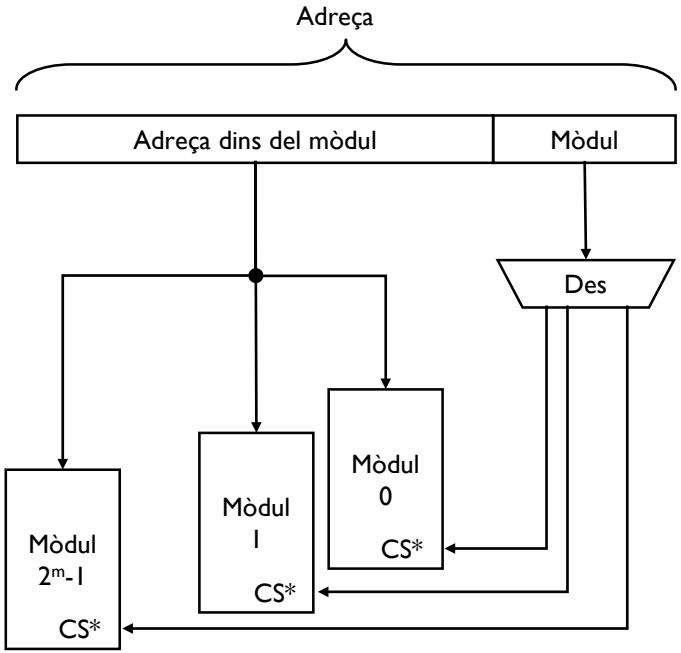


Amplada de banda efectiu
1 bloc (4 paraules) → 1+10+1x4 cicles
⇒ 15 cicles/bloc ⇒ ***~1 byte/cicle***

Memòria entrellaçada

De baix ordre: Adreces consecutives en mòduls diferents

- La MP de 2^n paraules es divideix en 2^m mòduls de 2^{n-m} paraules cadascun
  - Adreces consecutives s'ubiquen en mòduls consecutius

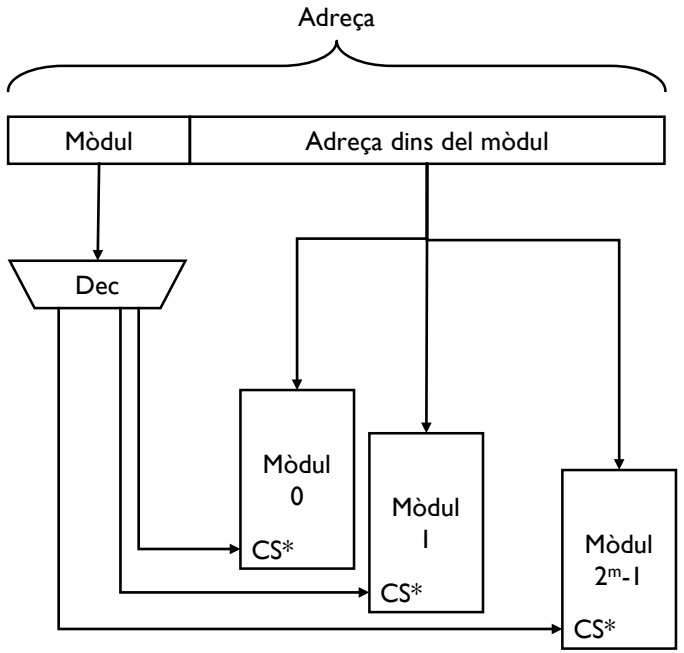


Mod 0	Mod 1	Mod 2	Mod 3
0	1	2	3
4	5	6	7
8	9	10	11
..	..	..	..

Per M = 2

D'alt ordre: Adreces consecutives en el mateix mòdul

- La MP de 2^n paraules es divideix en 2^m mòduls de 2^{n-m} paraules cadascun
  - cada mòdul emmagatzema 2^{n-m} paraules consecutives



Mod 0	Mod 1	Mod 2	Mod 3
0	2^{n-2}	$2 \cdot 2^{n-2}$	$3 \cdot 2^{n-2}$
1	$2^{n-2}+1$	$2 \cdot 2^{n-2}+1$	$3 \cdot 2^{n-2}+1$
...	..	..	..
$2^{n-2}-1$	$2 \cdot 2^{n-2}-1$	$3 \cdot 2^{n-2}-1$	$4 \cdot 2^{n-2}-1$

Arquitectura de computadors

Memòries

- Estructura jeràrquica de les memòries
- Memòria principal
- Memòria cau
- Memòria Virtual

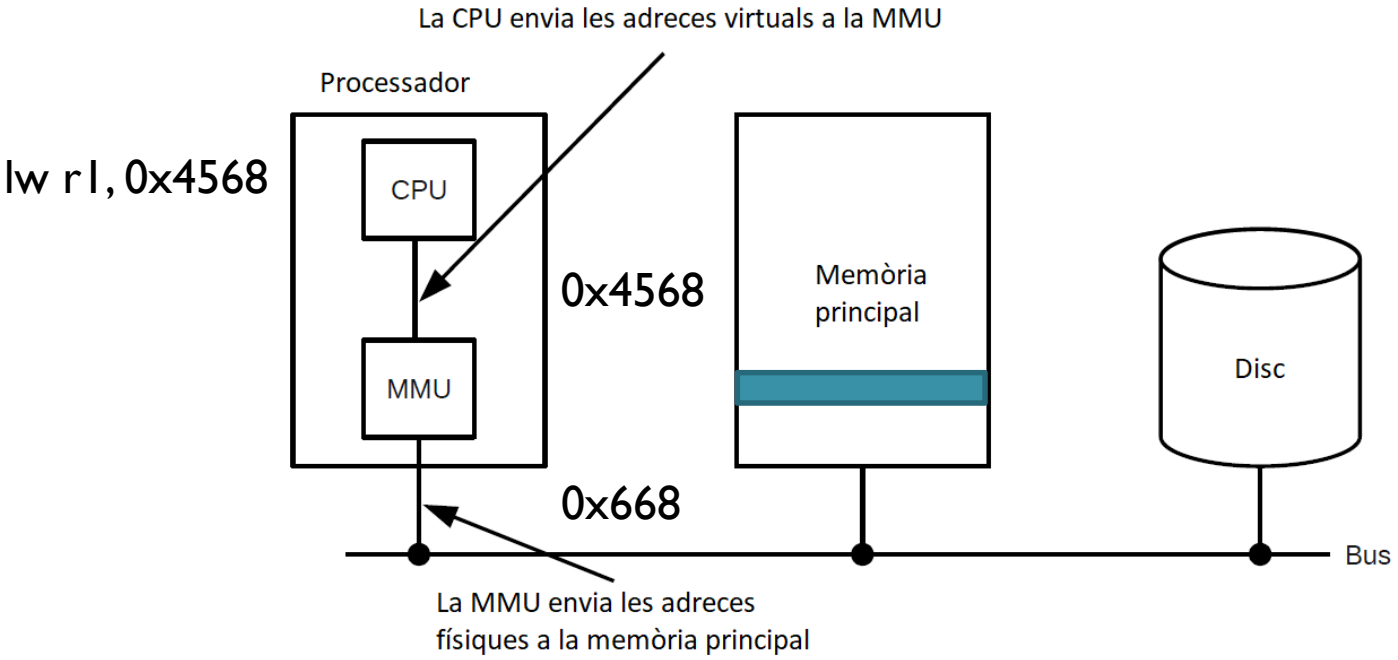
Justificació de l'ús de memòria virtual:

- Elimina la problemàtica del “poc” espai de la memòria principal:
  - Crea un espai d'adreces virtuals (és a dir, una memòria virtual) que gestiona com si hagués una gran memòria principal.
  - Gestiona automàticament les transferències d'informació entre els dos nivells de la jerarquia de memòria involucrats: la memòria principal de la màquina (física) i la memòria secundària (disc).
- Permet compartir la memòria entre múltiples processos: de forma segura i eficient
  - Si diversos processos s'estan executant al mateix temps en una màquina, la memòria que es necessita per tots aquests programes pot ser més gran que l'espai de memòria principal lliure. Però normalment, només es fa servir una part d'aquesta memòria en un moment donat, de manera que la memòria principal només ha de contenir les parts en ús dels programes en execució

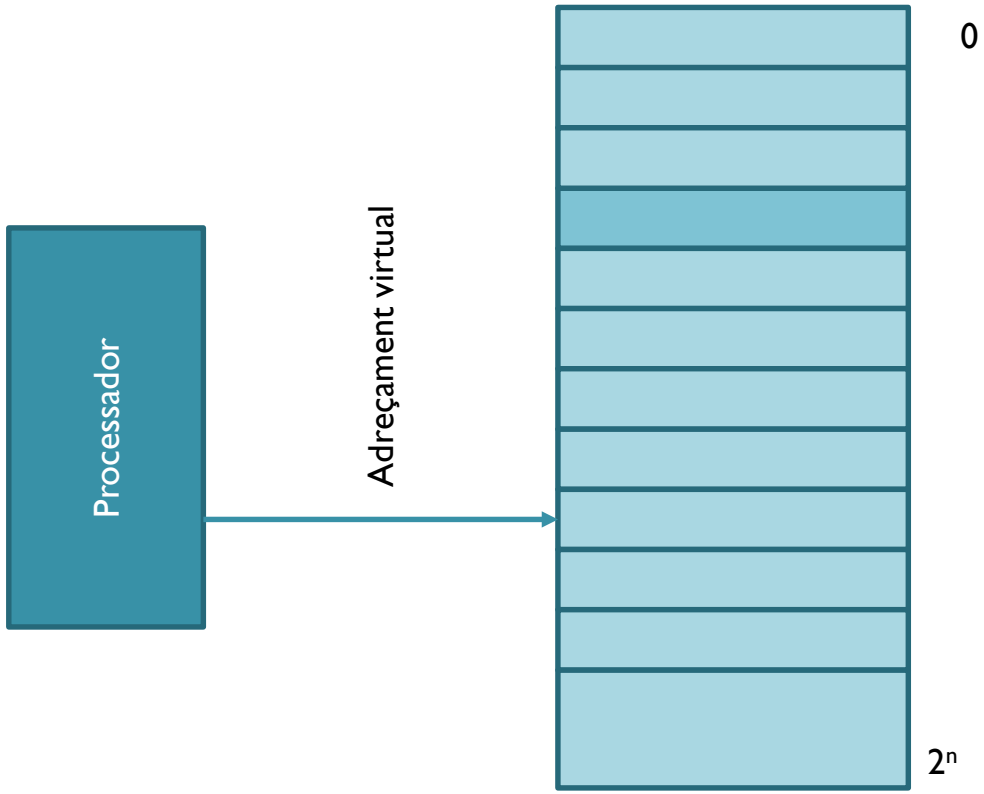
Conceptes

La CPU produeix adreces virtuals i la unitat de gestió de memòria o MMU (Memory Management Unit) les tradueix a adreces físiques, que s'utilitzaran per accedir a la memòria principal (o a la memòria cau si n'hi ha).

La clau de la memòria virtual és la traducció d'adreces (address translation).



El programa que executa el processador només entén d'adreces virtuals. Direcciona sobre un mapa de memòria, que és virtual.



Exemple de direccionament

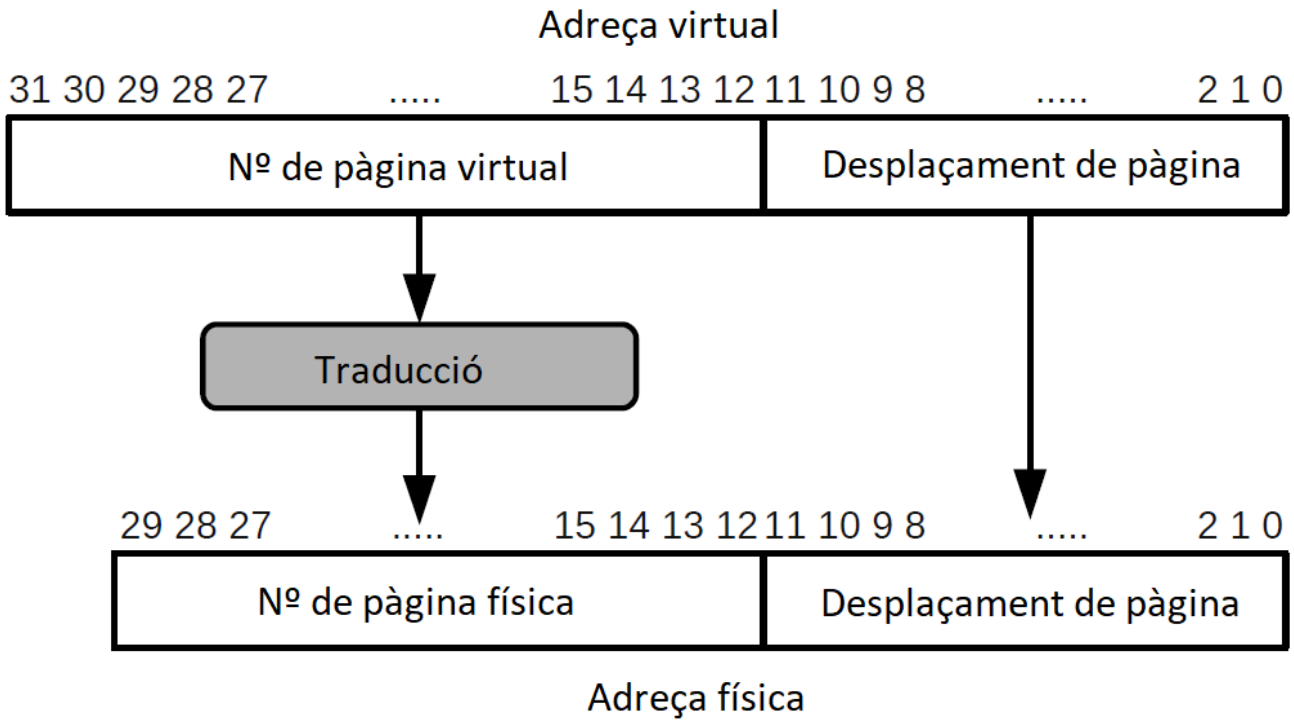
Memòria principal :

- 1 Gbytes -> 2^{30}
- Pàgina de 4 KB -> 2^{12} bytes
- Nombre de pàgines a memòria: 2^{18}

Direccionament virtual :

- 4 Gbytes -> 2^{32} => 2^{12} x 2^{20}

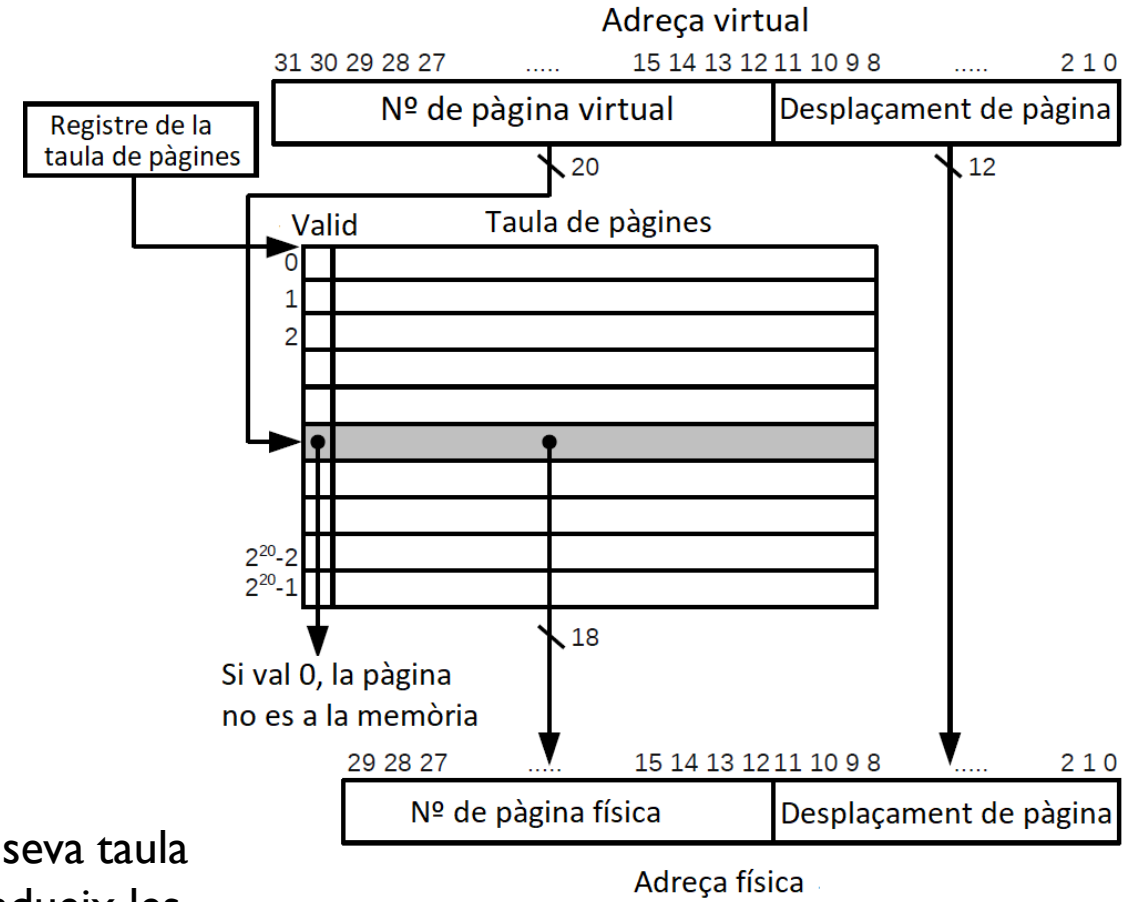
  2^{20} pàgines virtuals (marcs)



Consideracions de disseny d'un sistema de memòria virtual

- Problemàtica -> els cost de les fallades de pàgina.
  - Aquest cost és degut a la gran diferència de velocitats entre la memòria principal i la secundària. (10.000 vegades més lenta)
- Les pàgines han de ser prou grans per amortitzar el temps d'accés a disc. Normalment entre 16 KB a 64 KB
- Interessa organitzar les pàgines de la MP de manera que es redueixi al màxim la taxa de fallades. Normalment -> Totalment associatiu
- Les fallades de pàgina es poden gestionar per programari (la sobrecàrrega d'utilitzar programari en lloc de maquinari és poca comparada amb el temps d'accés a disc), i pot fer servir algoritmes més intel·ligents per decidir com col·locar les pàgines.
- La política d'escriptura directa per gestionar escriptures en memòria virtual no és apropiada perquè cada escriptura seria molt lenta. En el seu lloc, els sistemes de memòria virtual usen postescritura.

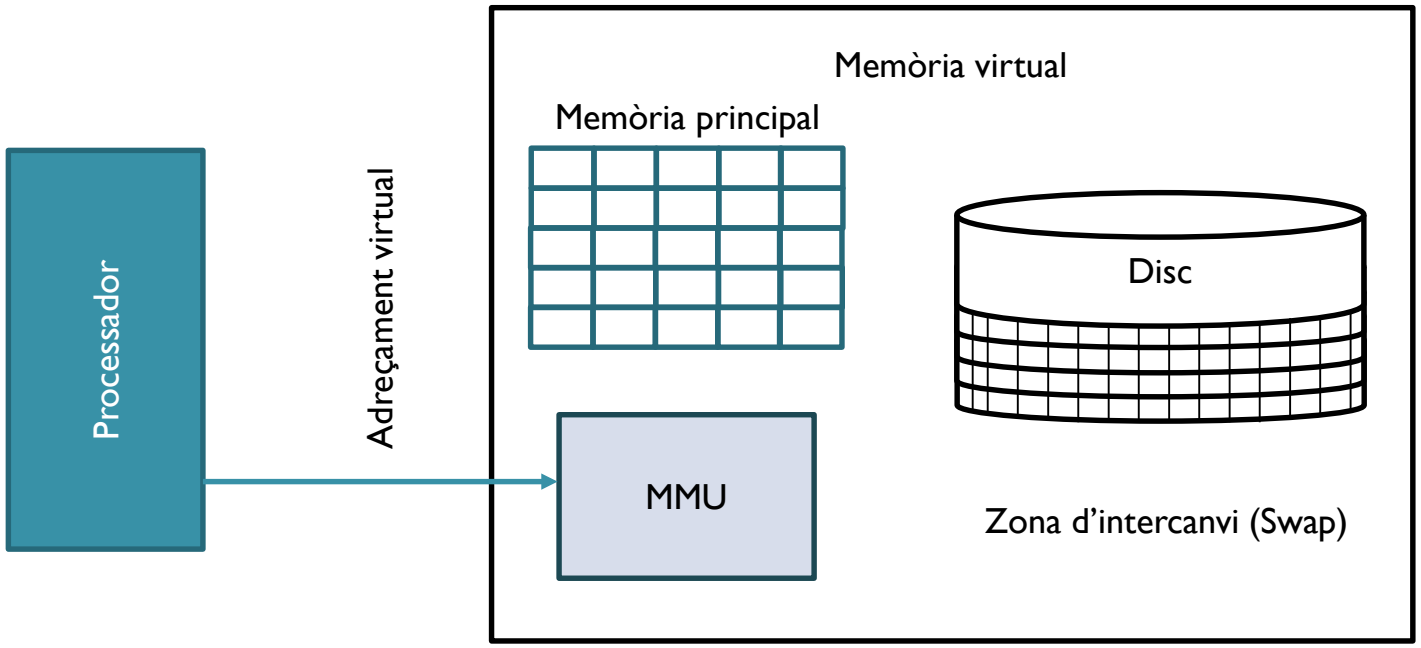
Taula de pàgines



Cada procés té la seva taula de pàgines, que tradueix les adreces virtual en físiques. S'assigna pel registre de taula de pàgines

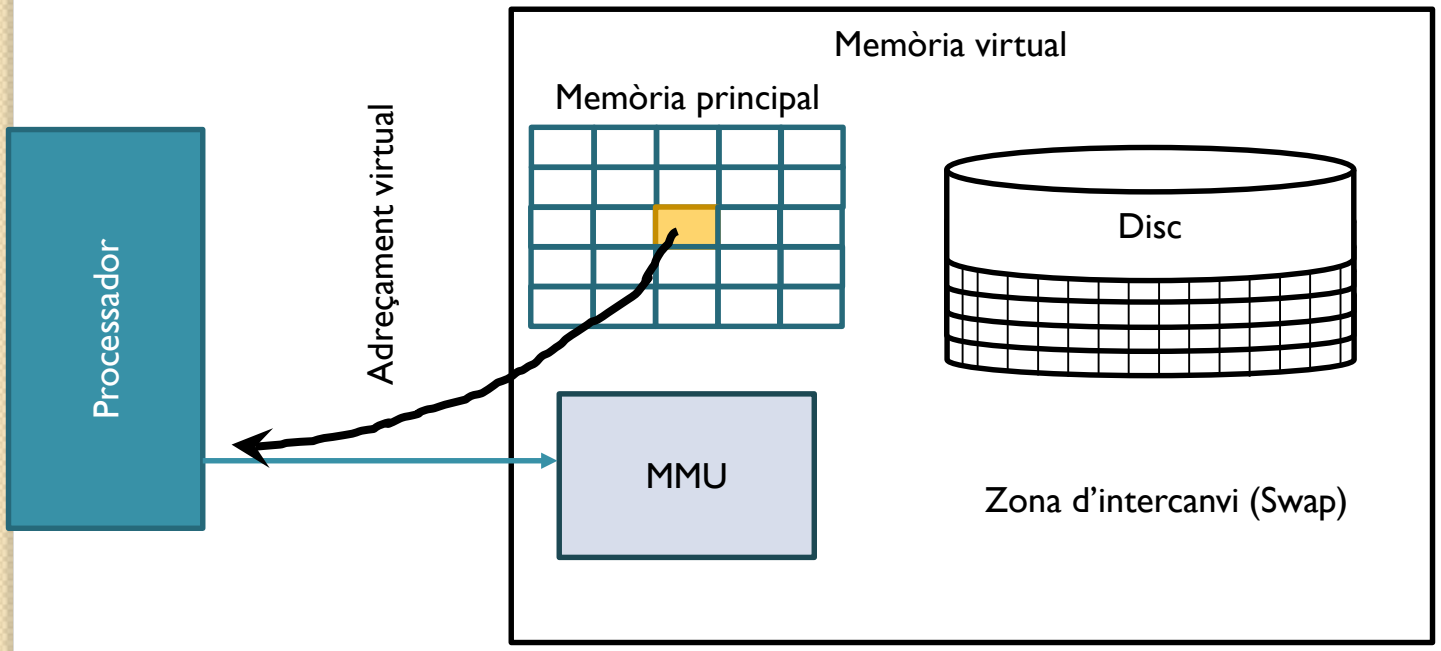
En lloc d'una memòria principal, existeix la memòria virtual. El suport físic és una zona de disc (intercanvi o swap) més memòria principal. La MMU (Memory Management Unit) rep l'adreça i:

- la tradueix o
- dona una fallada de pàgina.



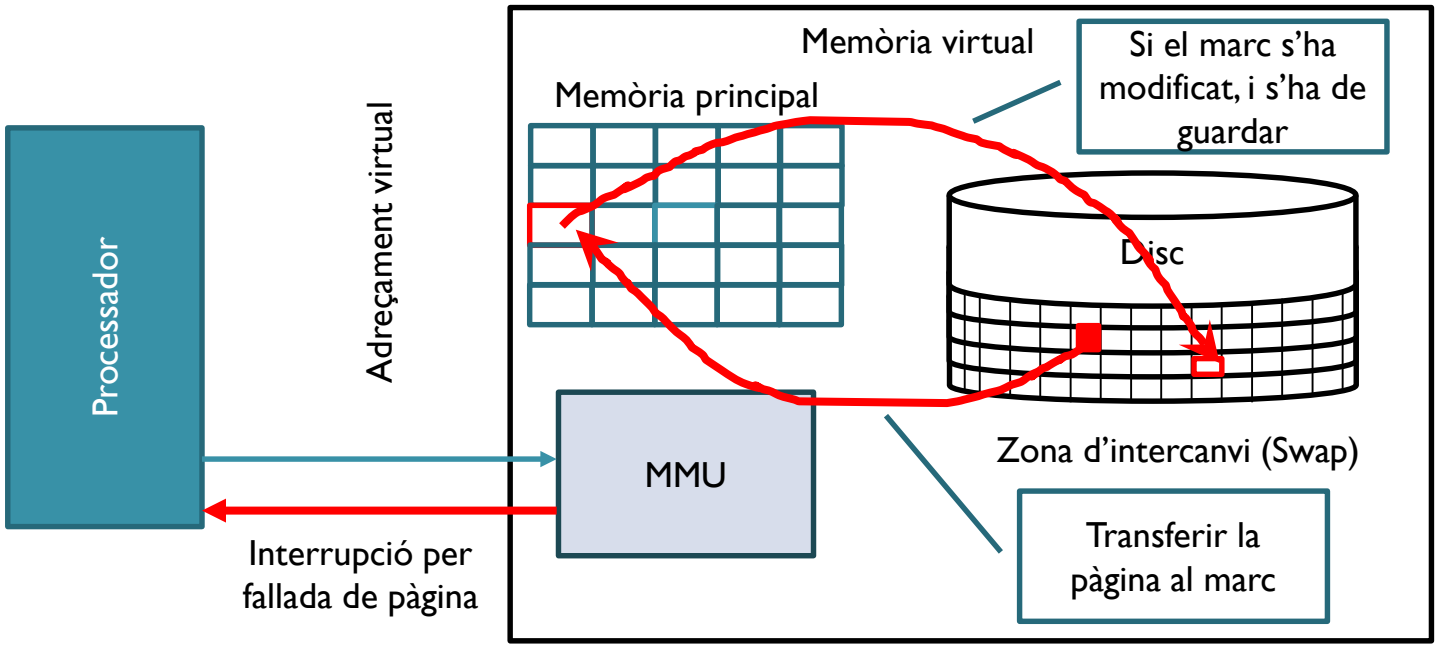
Encert

La posició generada pel processador és en algun dels marcs de pàgina de la memòria principal, la MMU tradueix l'adreça i es fa l'accés.



Fallada

La posició generada pel processador no és en cap dels marcs de pàgina de la memòria principal. No es pot fer l'accés i el programa no pot continuar l'execució. La MMU avisa al S.O. mitjançant una interrupció. El S.O. genera una petició al disc per transferir la pàgina a un dels marcs.



Un cop s'ha transferit la pàgina al marc, el programa pot continuar l'execució

Tractament de les fallades

Problemes en la implementació de la taula de pàgines:

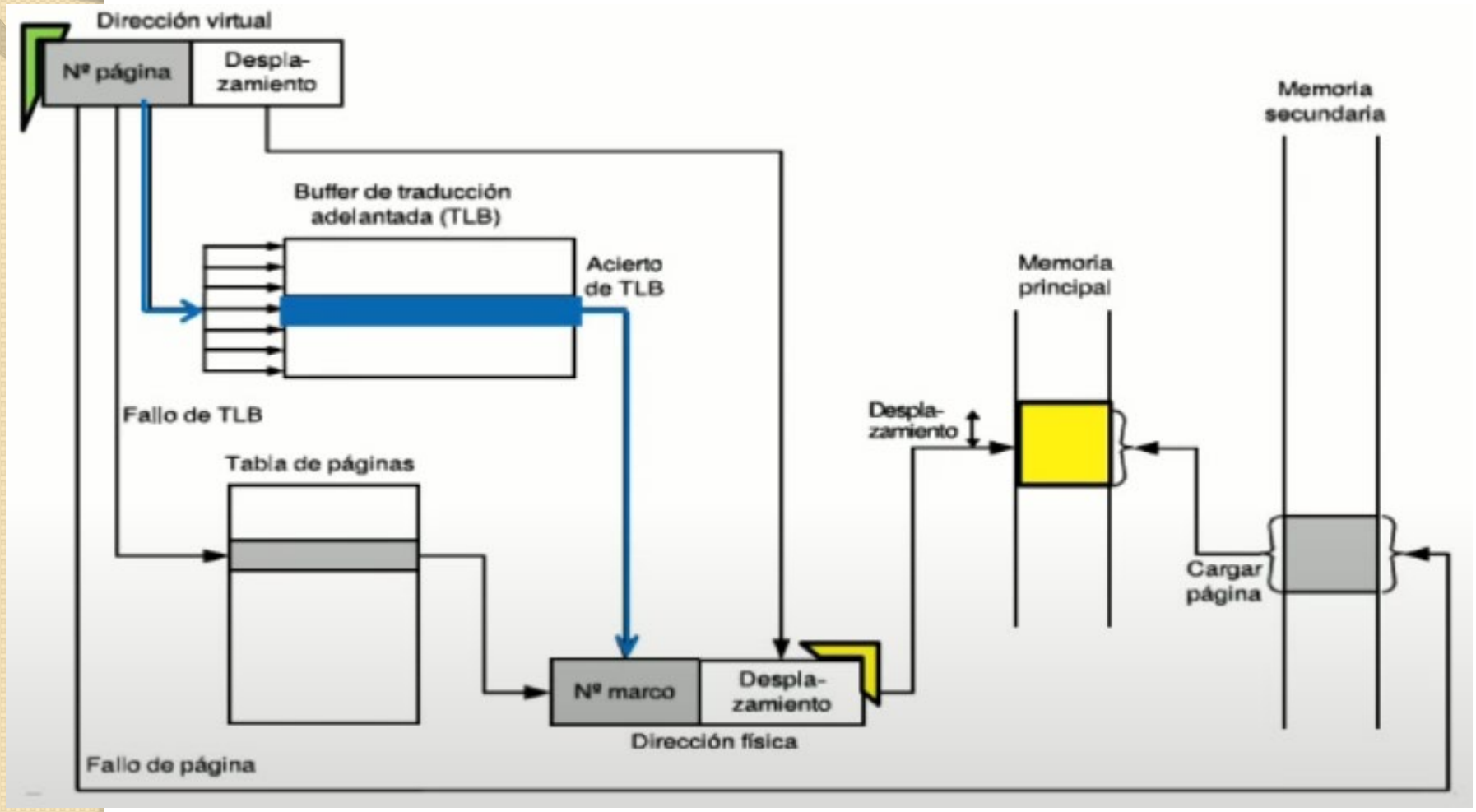
- La taula és molt gran i n'hi ha una per procés
  - Adreces de 32 bits i 4KB/pàgina (o més) -> 2^{20} entrades (taula de pàgines de 1048576 entrades)
- L'associació (correspondència) ha de ràpida
  - Cada instrucció d'accés a memòria ->

    Accés a memòria (taula de pàgines) per saber l'associació adreça virtual a física + Accés a l'adreça física
- Per solucionar-ho -> Es pot crear una TLB: Cau especial que guarda les traduccions: #Pàgina virtual a #Pàgina física més freqüents.

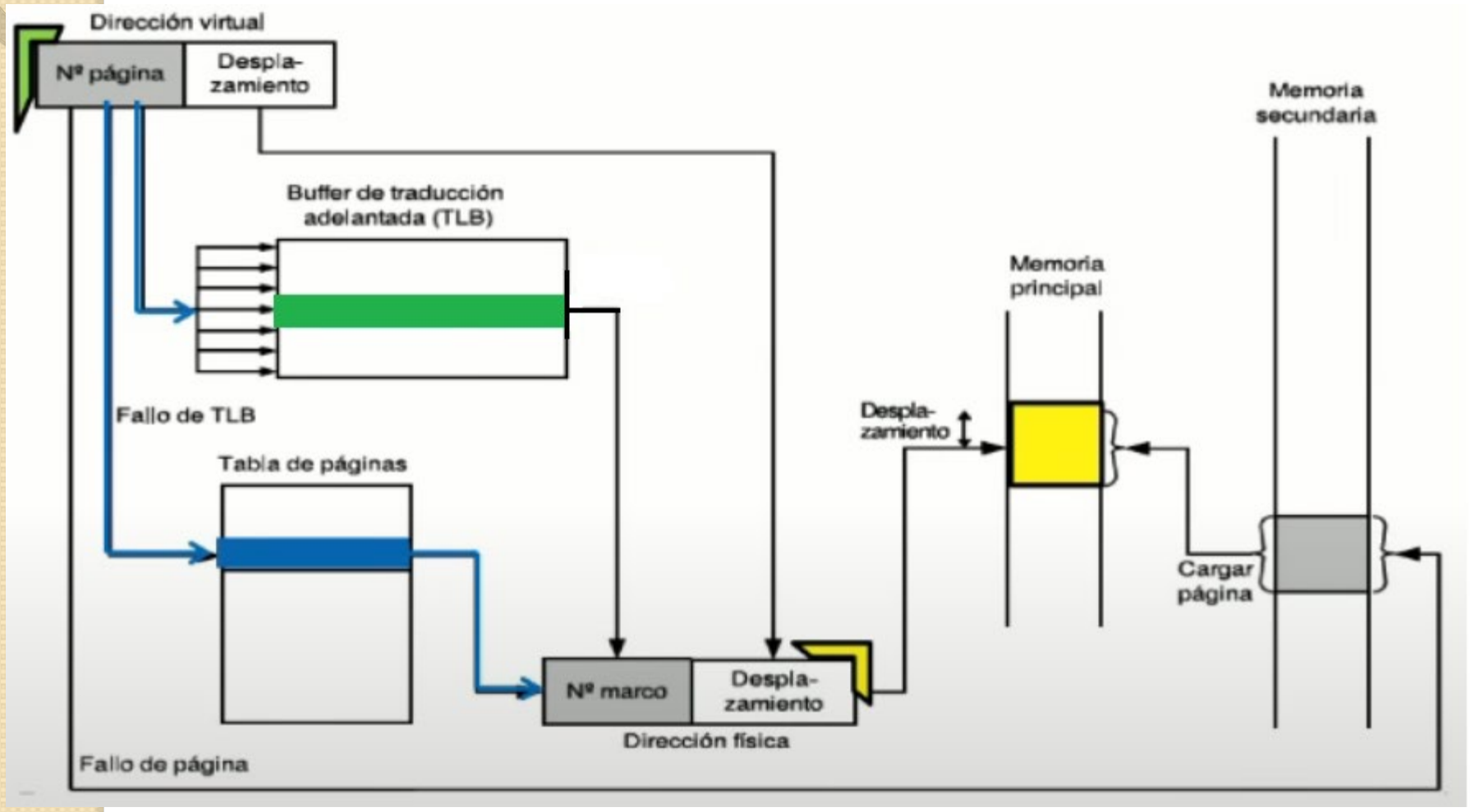
TLB (translation-lookaside buffer)

- Dispositiu de maquinari que tradueix les adreces virtuals a físiques sense accedir a la taula de pàgines
- Està ubicada dins de la MMU
- Consisteix en un número petit d'entrades (màxim 64)
- Cadascuna de les entrades conté informació sobre una pàgina:
  - Bits de control: (validesa, us, modificat, ...)
  - Etiqueta (Bits del nombre de pàgina virtual – bits índex d'accés a la TLB)
  - Nombre de pàgina física
  - Bits de control de la taula de pàgines per la pàgina virtual (M, U, ..)

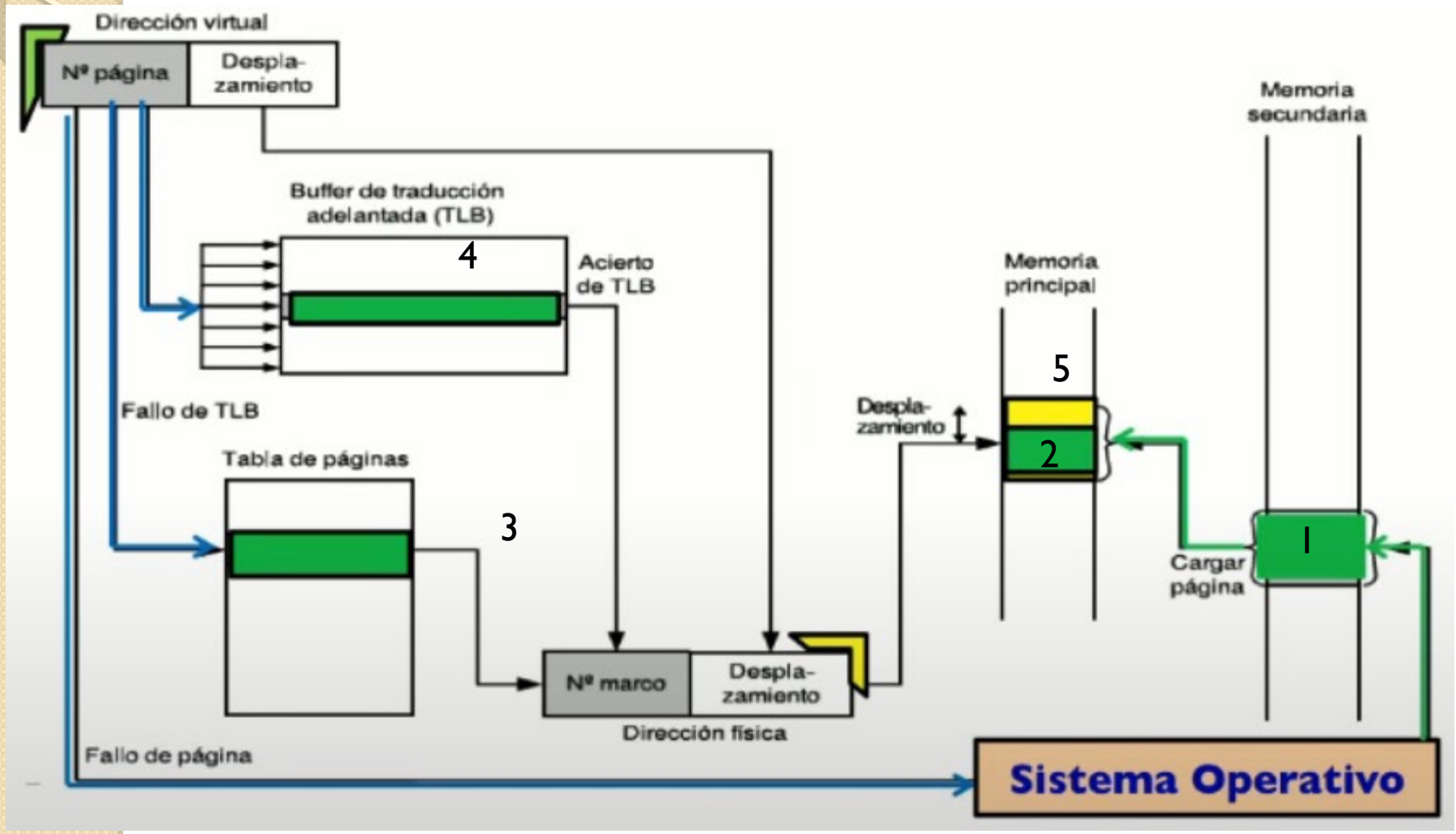
En cas d'encert a la TLB:



En cas fallada a la TLB i encert a taula de pàgines



En cas fallada a la TLB i la taula de pàgines



Característiques del TLB

Característiques	Intel Pentium	Power PC 604
Adreces virtuals	32 bits	52 bits
Adreces físiques	32 bits	32 bits
Pàgina	4KB – 4MB	4KB – 256 MB
Organització TLB	Una TLB per instruccions i una per dades. Associativa per conjunts de 4 vies LRU-simplificat TLB: • instruccions: 32 entrades • Dades: 64 entrades Fallades tractades per maquinari	Una TLB per instruccions i una per dades. Associativa per conjunts de 2 vies LRU TLB: • instruccions: 128 entrades • Dades: 128 entrades Fallades tractades per maquinari

Algoritme:

1. La MMU descomposa l'adreça virtual: número de pàgina virtual + desplaçament.
2. La MMU cerca el número de pàgina virtual al TLB:
    1. Si hi ha encert de TLB, salta al pas 6.
    2. Si hi ha error de TLB, se segueix pel pas 3.
3. Com que hi ha una fallada de TLB, aleshores la MMU no tindrà més remei que anar a la taula de pàgines i comprovar el bit de validesa de l'entrada corresponent:
    1. Si està actiu, hi ha encert de pàgina i se salta al pas 5.
    2. Si està apagat, hi ha error de pàgina i se segueix pel pas 4.
4. Com que hi ha fallada de pàgina, el procés interromp la seva execució i passa el control al SO, que ha de trobar la pàgina a la memòria secundària i decidir on col·locar la pàgina sol·licitada a la memòria principal. A més, si cal, es guarda la pàgina a la memòria secundària. Després, a l'entrada de la taula de pàgines, es posarà el número de pàgina física on s'ha col·locat i s'activa el bit de validesa.
5. La MMU copia el contingut de l'entrada de la taula de pàgines que correspon a la pàgina virtual cercada (bit d'ús, bit de modificació i número de pàgina física) al TLB a la posició que li correspongui. A més, s'actualitza l'etiqueta de la posició del TLB per identificar aquesta nova entrada.
6. La MMU forma l'adreça física usant la informació del TLB. I actualitza els bits de control (s'activa el bit d'ús i, si la referència és una escriptura, s'activa el bit de modificació).