



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Departament d'Arquitectura de Computadors

Tema 6

Memòria Cache

Estructura de Computadors (EC)

2023 - 2024 Q2

Adrià Armejach (adria.armejach@upc.edu)





Introducció



Requisits del sistema de memòria

- El programador requereix memòria que sigui...
 - ràpida
 - de gran capacitat
 - barata
- Cap tecnologia compleix amb tots els requisits

Tecnologia de memòria	Temps d'accés	\$ per GB
SRAM	0.5 - 2.5 ns	\$500 - \$1000
DRAM	50 - 70 ns	\$10 - \$40
SSD	25.000 - 100.000 ns	\$0,10 - \$0,50
HDD	5.000.000 - 20.000.000 ns	\$0,02 - \$0,05

Principi de localitat

Podem crear la **il·lusió** de una **gran** quantitat de **memòria molt ràpida** combinant múltiples tecnologies, gràcies al **principi de localitat**

- Els programes accedeixen a una porció relativament petita de l'espai d'adreces en cada instant de temps
 - No totes les adreces de memòria tenen la mateixa probabilitat de ser accedides

Principi de localitat

- **Localitat temporal**

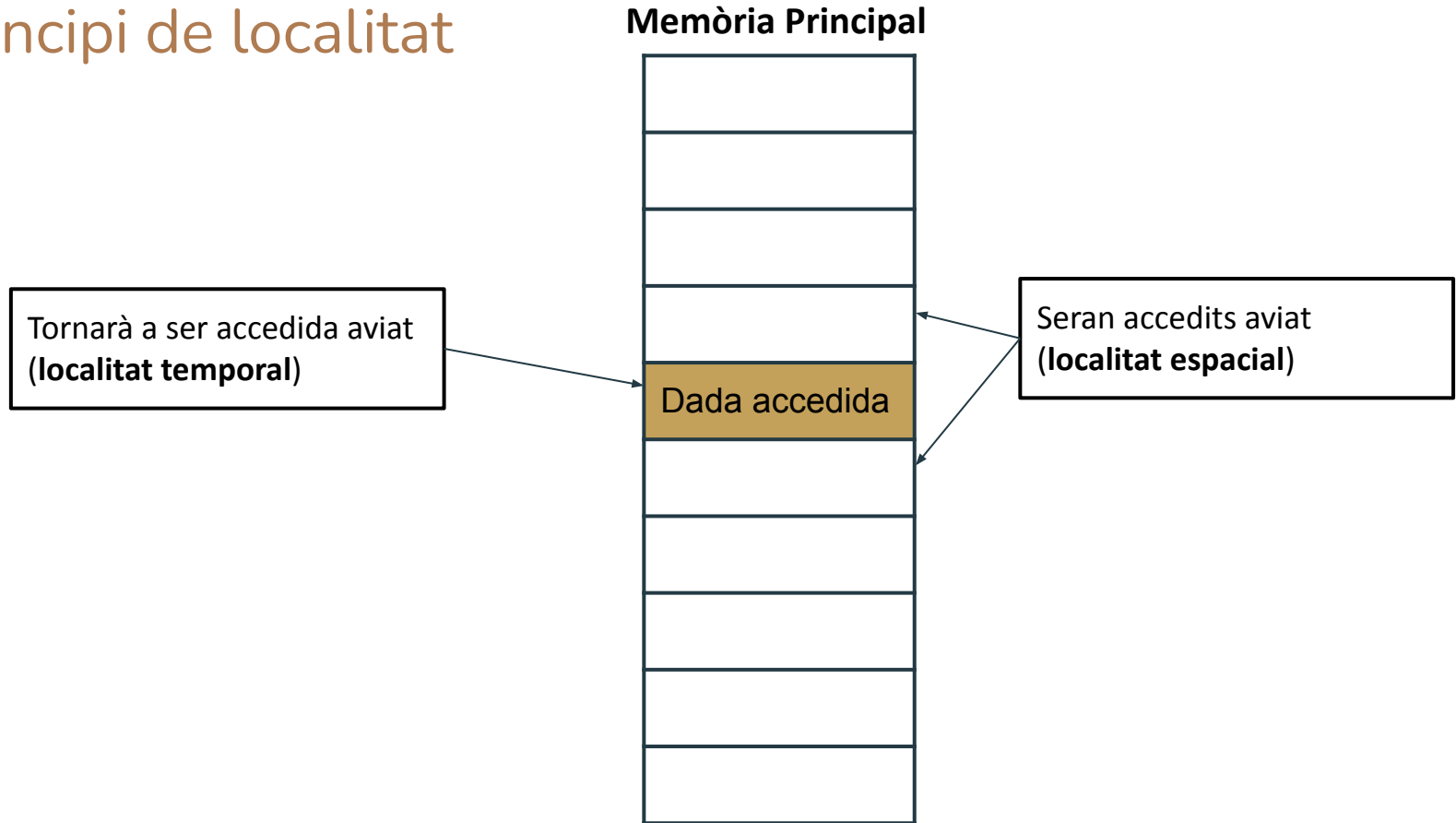
- Si accedim a una adreça de memòria, és probable que hi tornem a accedir en un futur proper.
 - bucles - s'accedeix repetidament a les mateixes instruccions i dades

- **Localitat espacial**

- Si accedim a una adreça de memòria, és probable que s'accedeixi a adreces “properes” en un futur proper.
 - instruccions
 - recorreguts de vectors i matrius

La localitat apareix de manera natural en els programes

Principi de localitat



Exemple 1: Principi de localitat

Suposem que un programa accedeix a les següents adreces de memòria:

Accés	Adreça de memòria
1	0
2	1024
3	0
4	1024
5	0
6	1024
7	0
8	1024

Quin tipus de localitat té aquest programa?

Exemple 2: Principi de localitat

Suposem que un programa accedeix a les següents adreces de memòria:

Accés	Adreça de memòria
1	0
2	4
3	8
4	12
5	16
6	20
7	24
8	28

Quin tipus de localitat té aquest programa?

Exemple 3: Principi de localitat

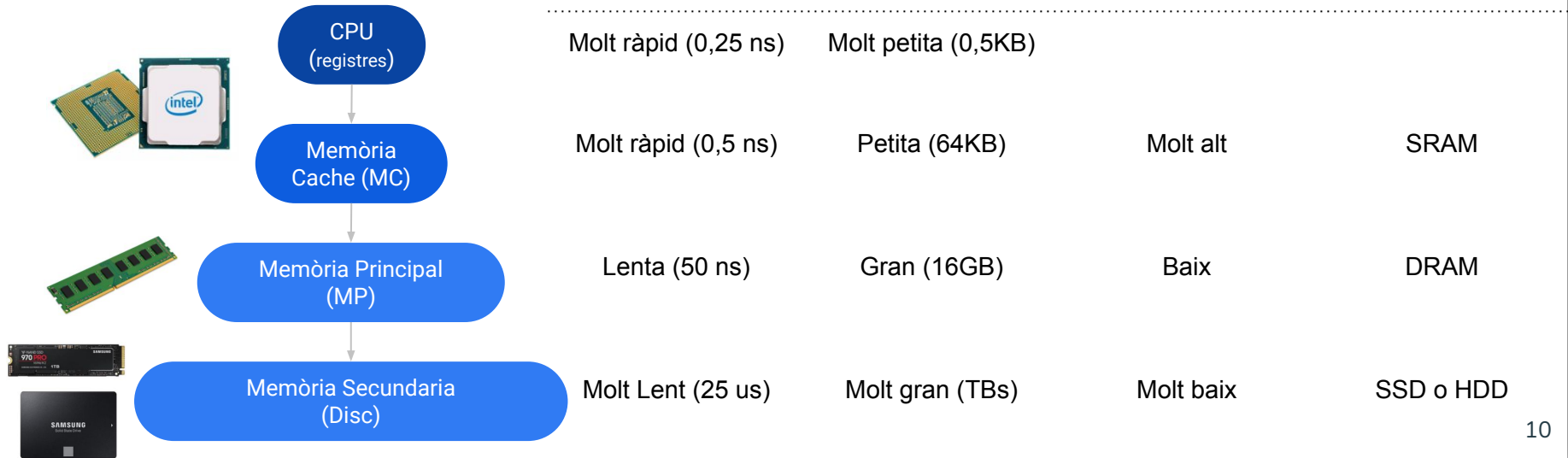
Suposem que un programa accedeix a les següents adreces de memòria:

Accés	Adreça de memòria
1	0
2	4
3	8
4	12
5	0
6	4
7	8
8	12

Quin tipus de localitat té aquest programa?

Jerarquia de memòria

Podem aprofitar el principi de localitat per implementar el sistema de memòria com una jerarquia de nivells

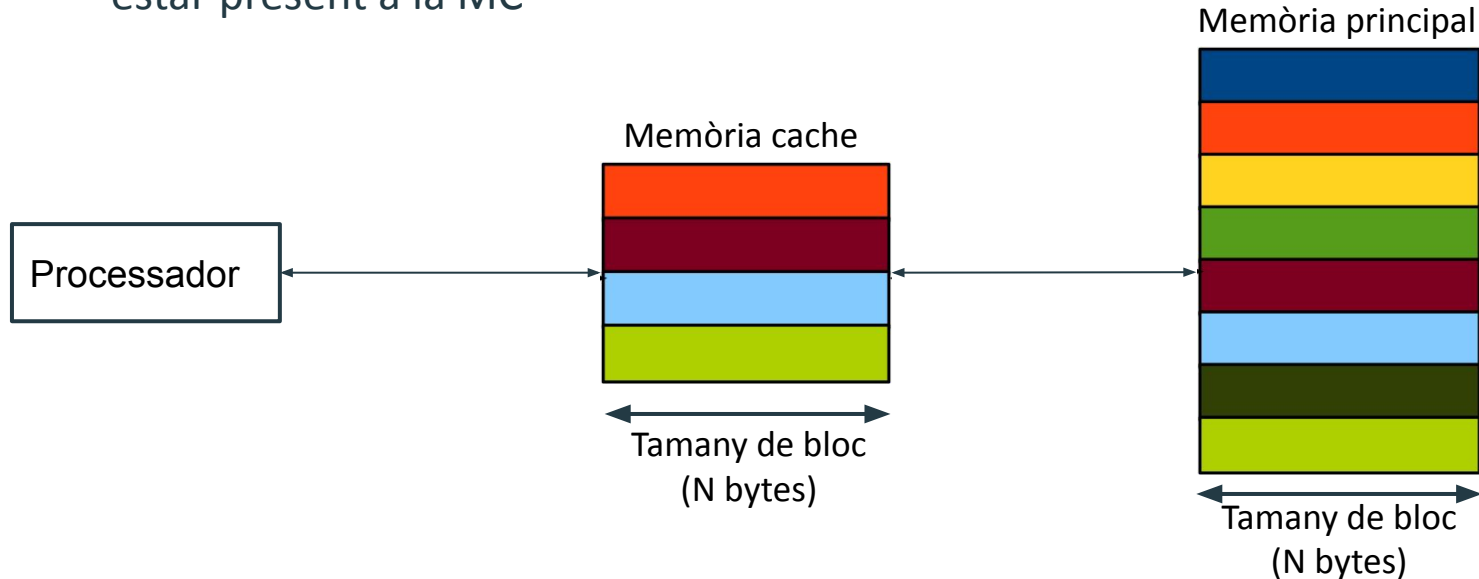


Jerarquia de memòria

- La memòria cache (MC):
 - emmagatzema un subconjunt del contingut de la MP
 - conté les dades accedides recentment
 - té una velocitat d'accés ràpida
 - té capacitat petita per reduir el cost
 - serveix la majoria d'accesos (localitat)
- El sistema de memòria ofereix gran capacitat mitjançant l'ús de DRAM (MP) i de memòria secundària (disc)
- Com més lluny de la CPU, major capacitat però major temps d'accés

Jerarquia de memòria

- L'espai d'adreces de memòria es divideix en **blocs** de memòria
 - Un bloc de memòria és la mínima unitat d'informació que pot estar o no estar present a la MC

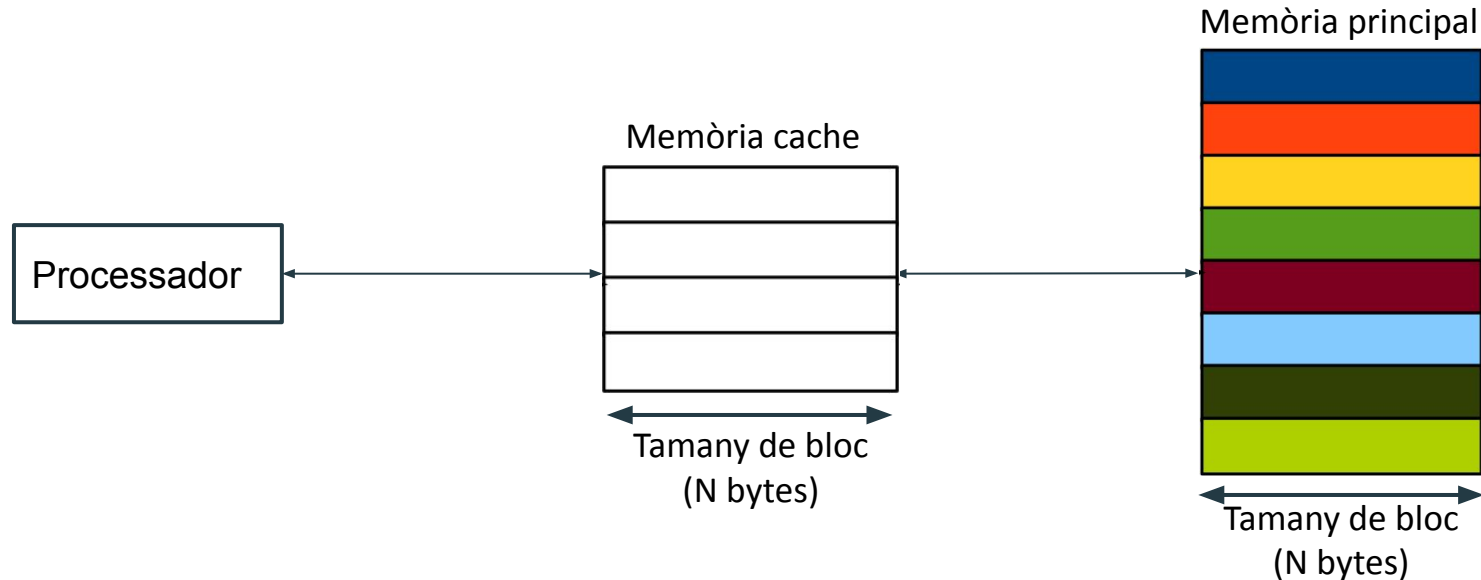


Memòria cache - Terminologia

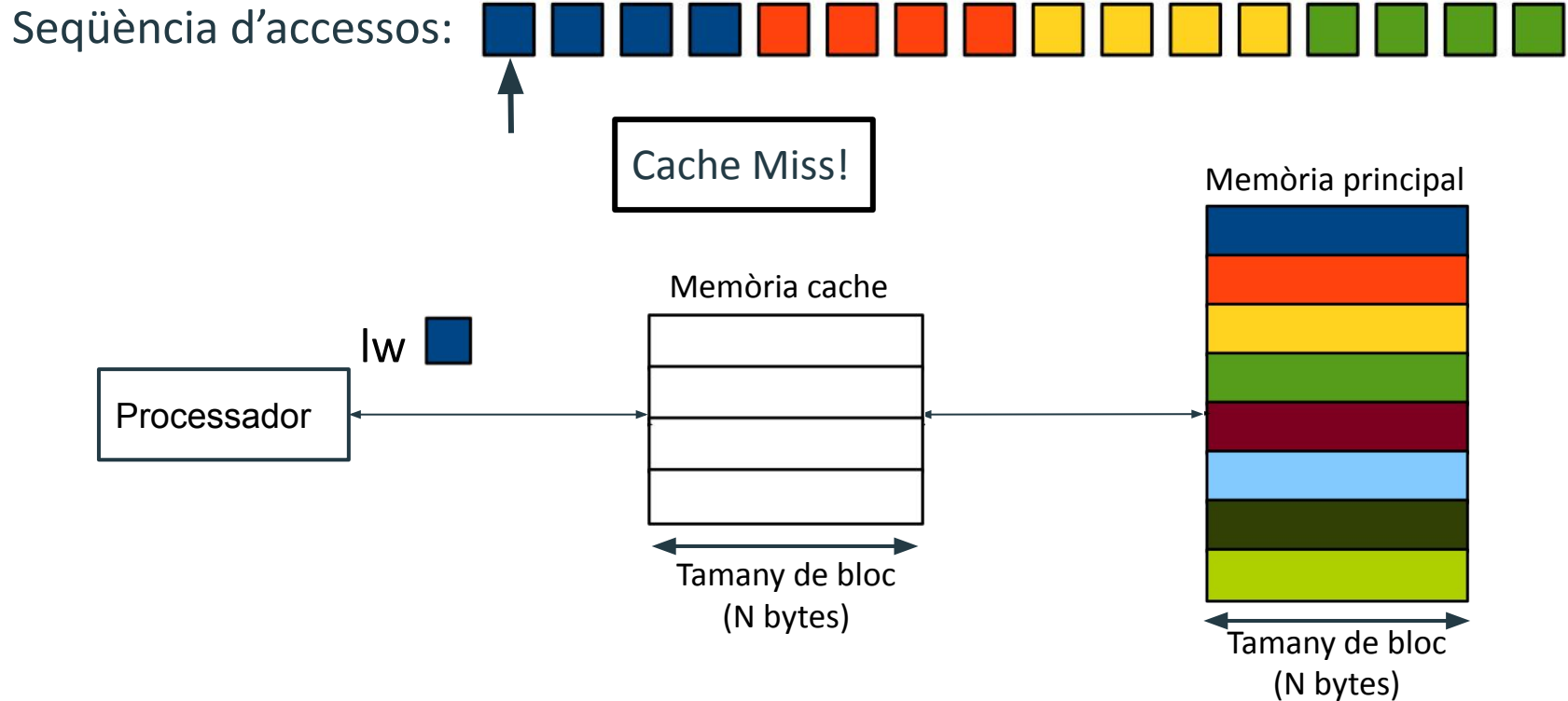
- Hit (Encert)
 - La dada accedida per el processador està present a la MC
- Hit Ratio (Taxa d'encerts)
 - Percentatge d'accessos a memòria que troben les dades a la MC (percentatge d'encerts)
- Miss (Fallada)
 - La dada accedida pel processador no està present a la MC
 - S'ha d'accedir a MP i portar el bloc sencer a la MC
- Miss Ratio (Taxa de fallades)
 - Percentatge d'accessos a memòria que NO troben les dades a la MC (percentatge de fallades)

Seqüència d'accessos

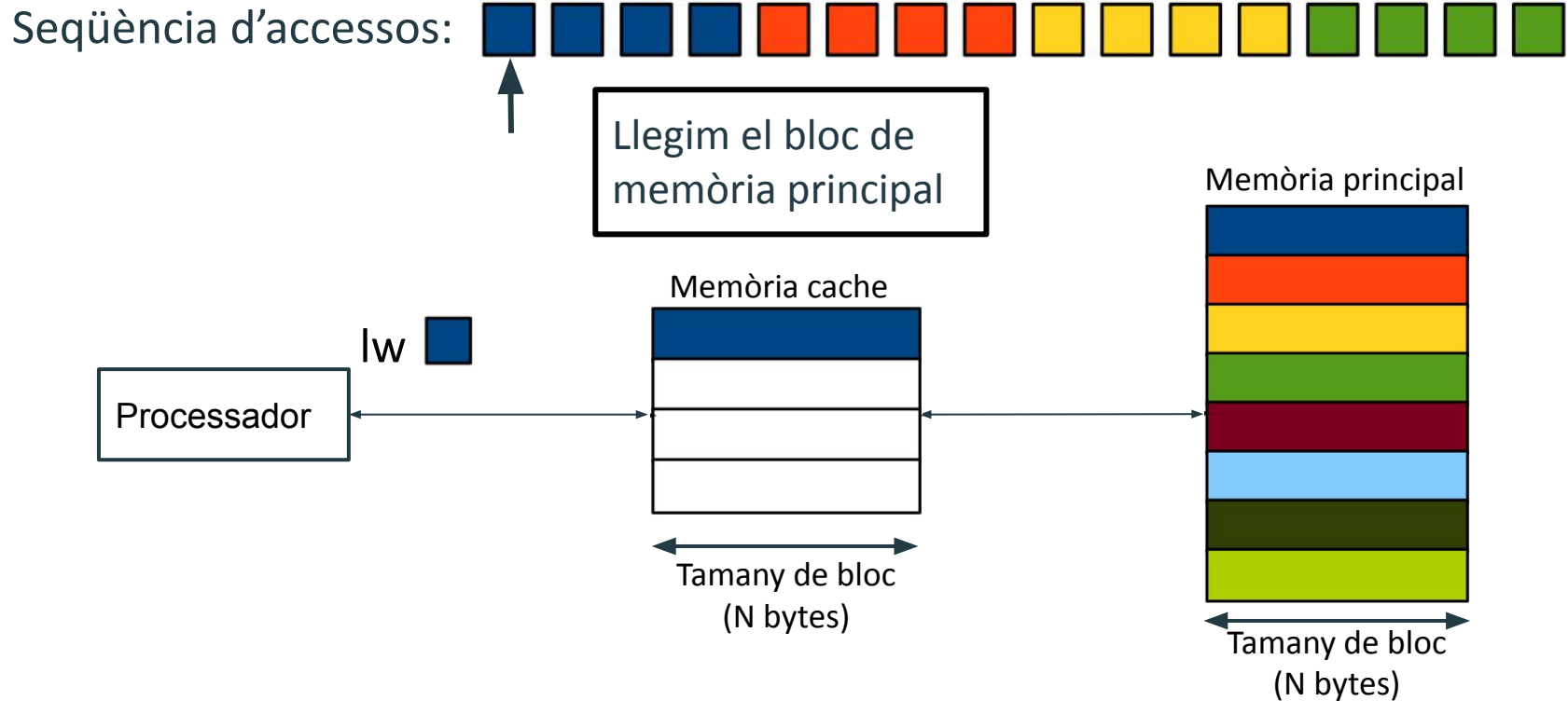
Seqüència d'accessos: 



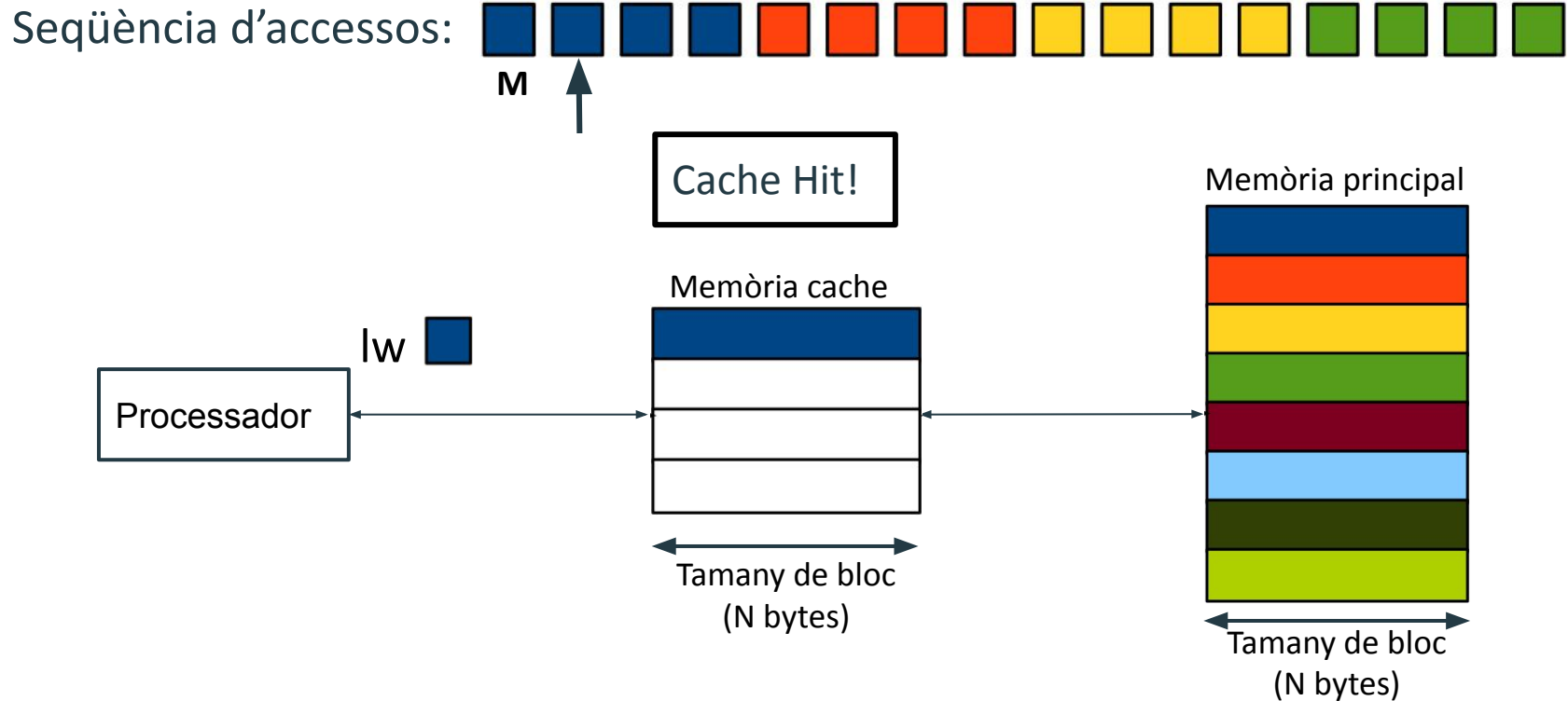
Seqüència d'accessos



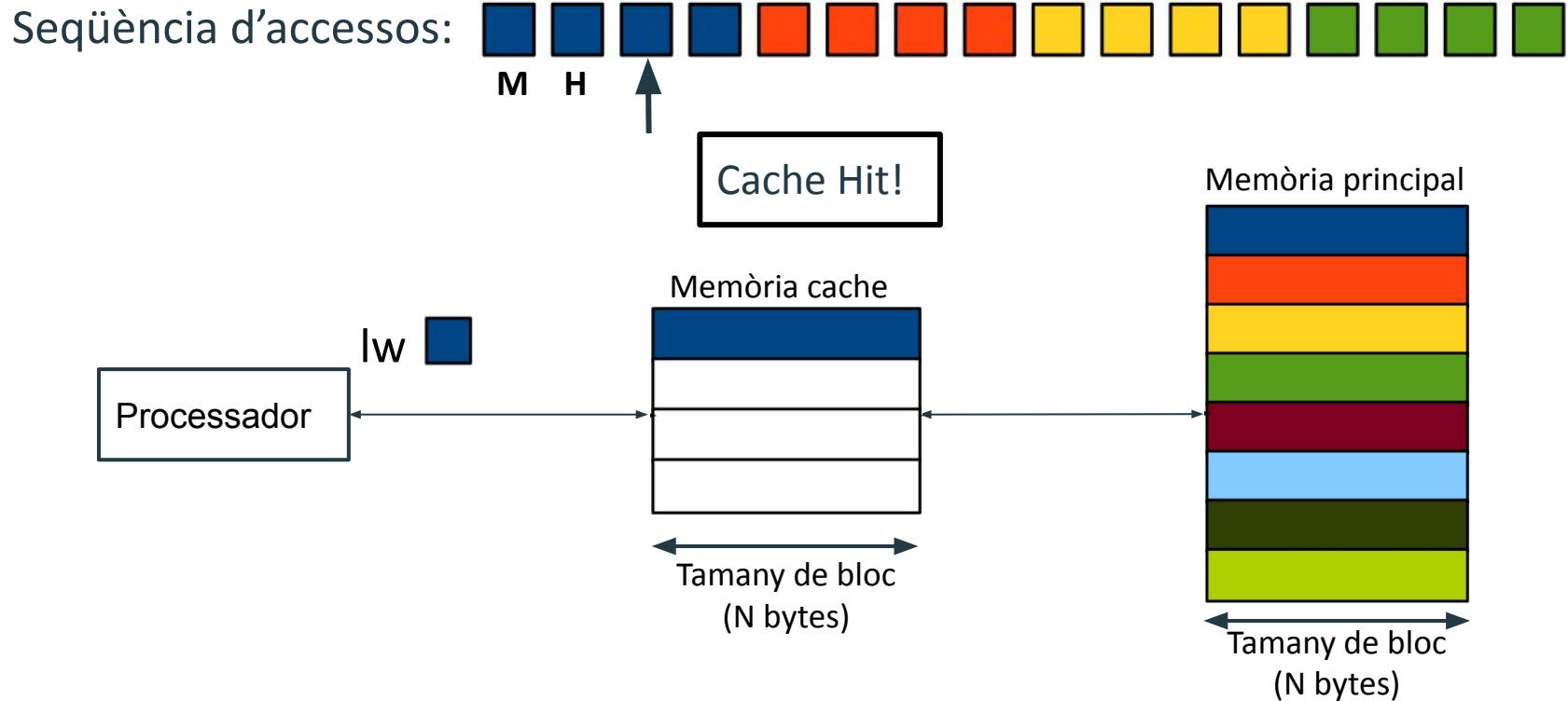
Seqüència d'accessos



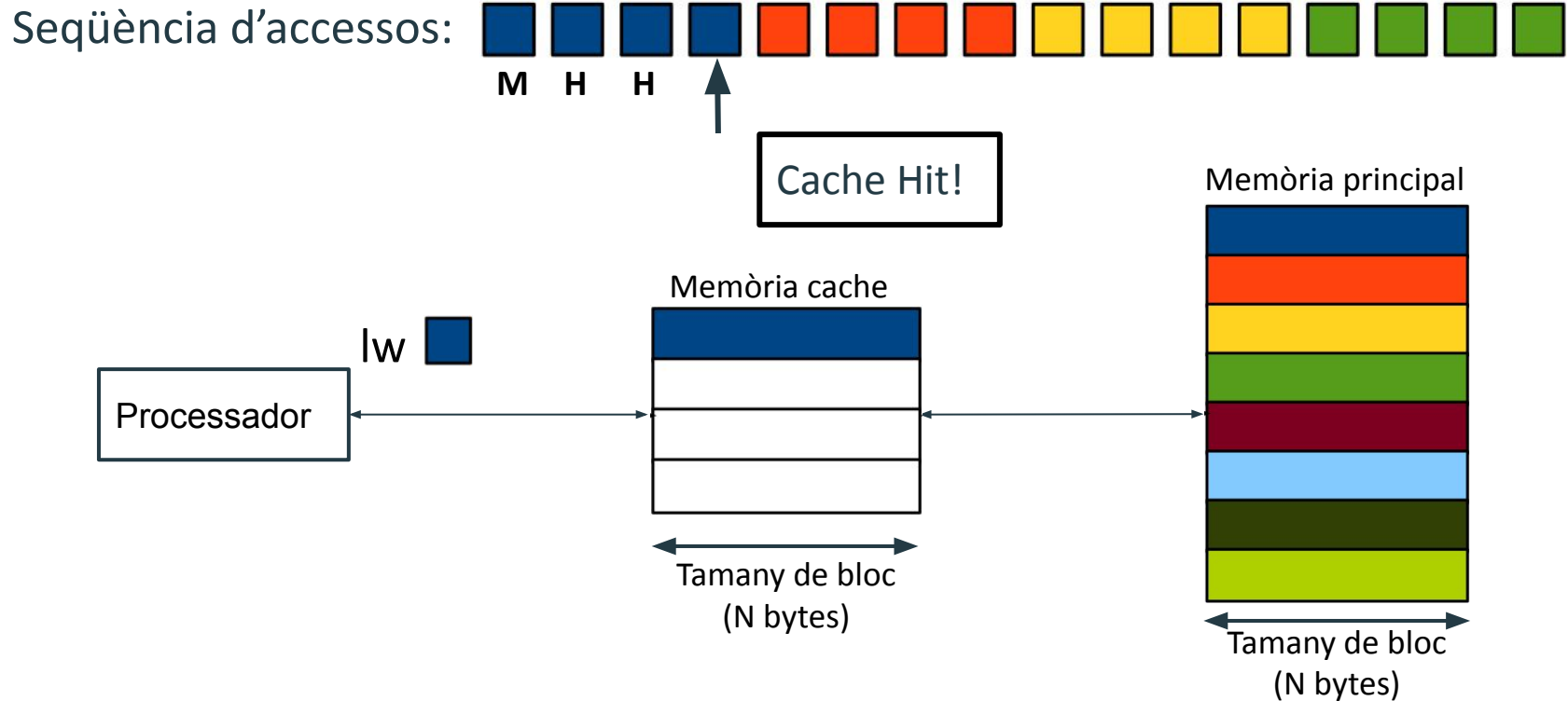
Seqüència d'accessos



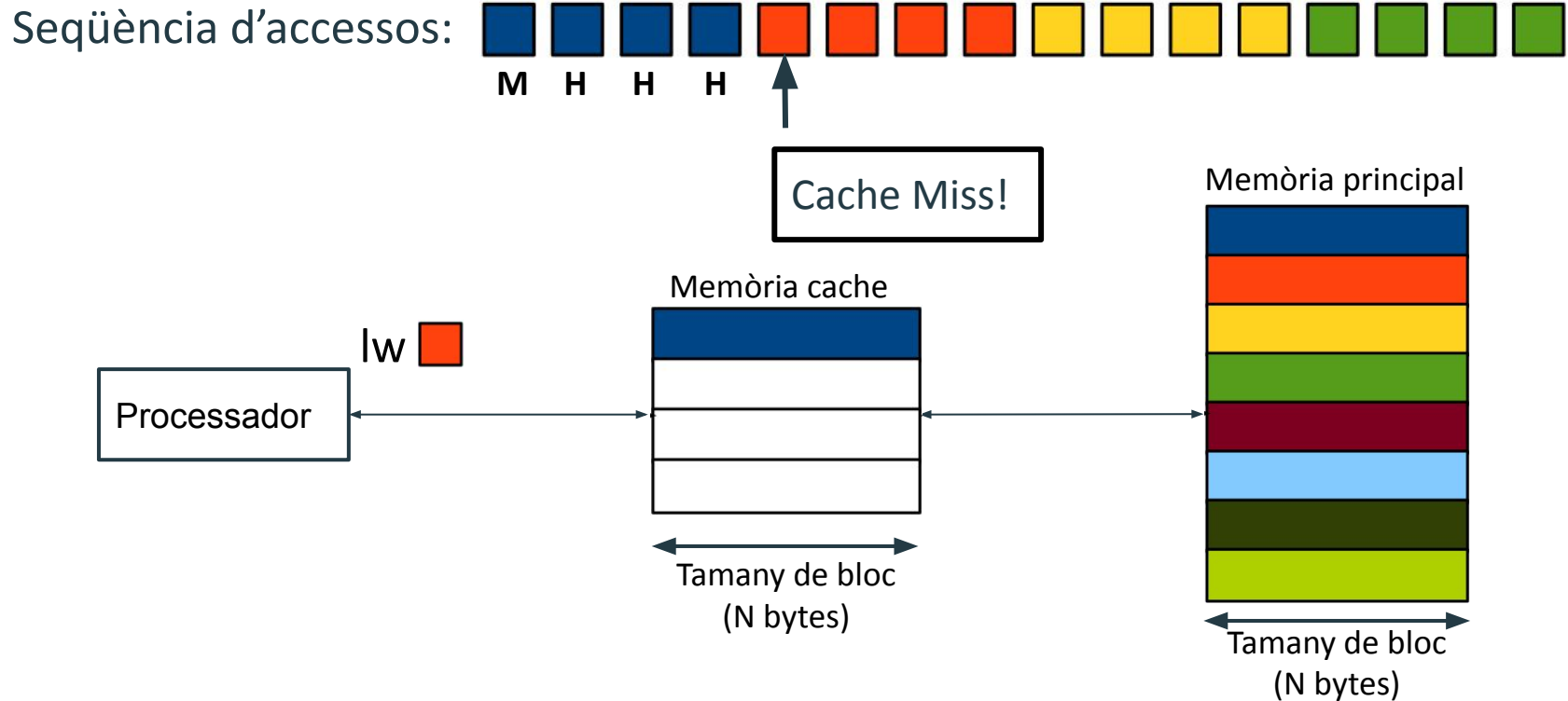
Seqüència d'accessos



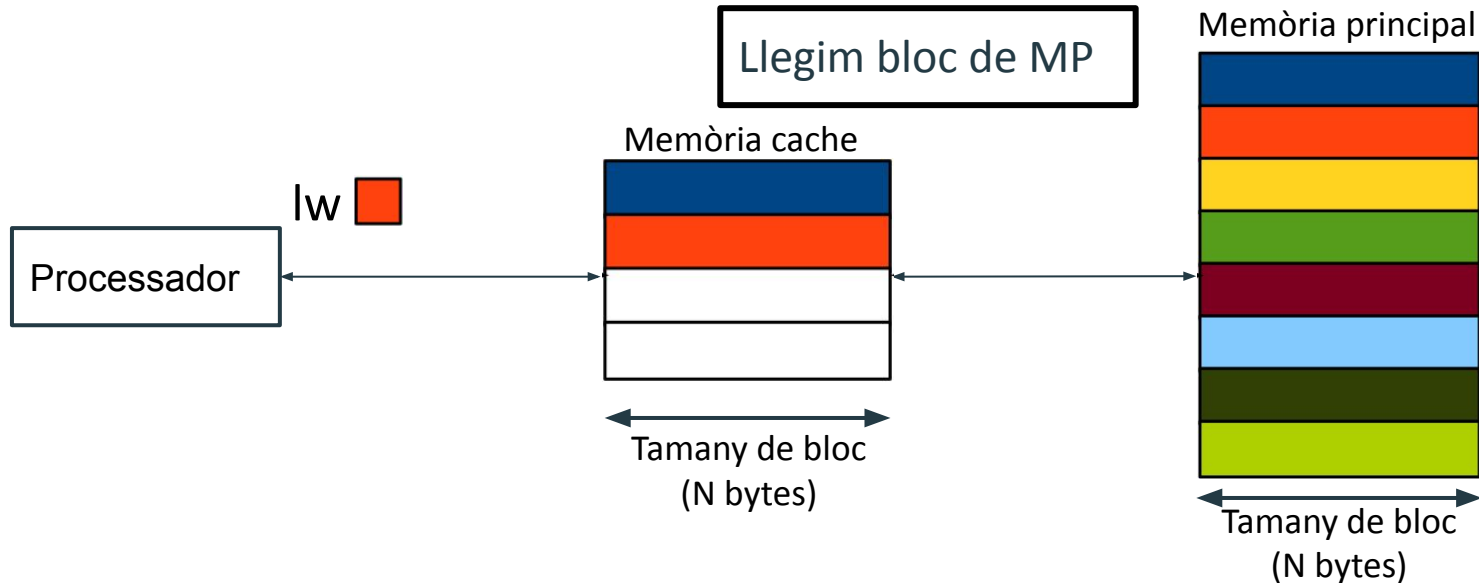
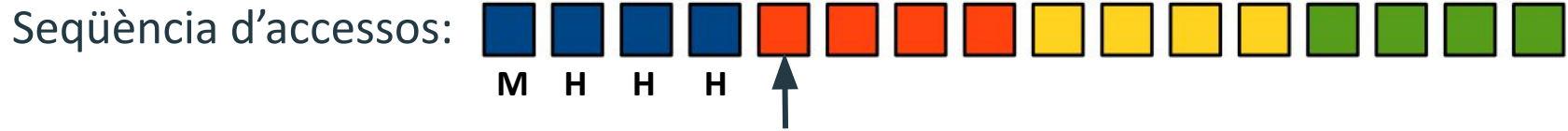
Seqüència d'accessos



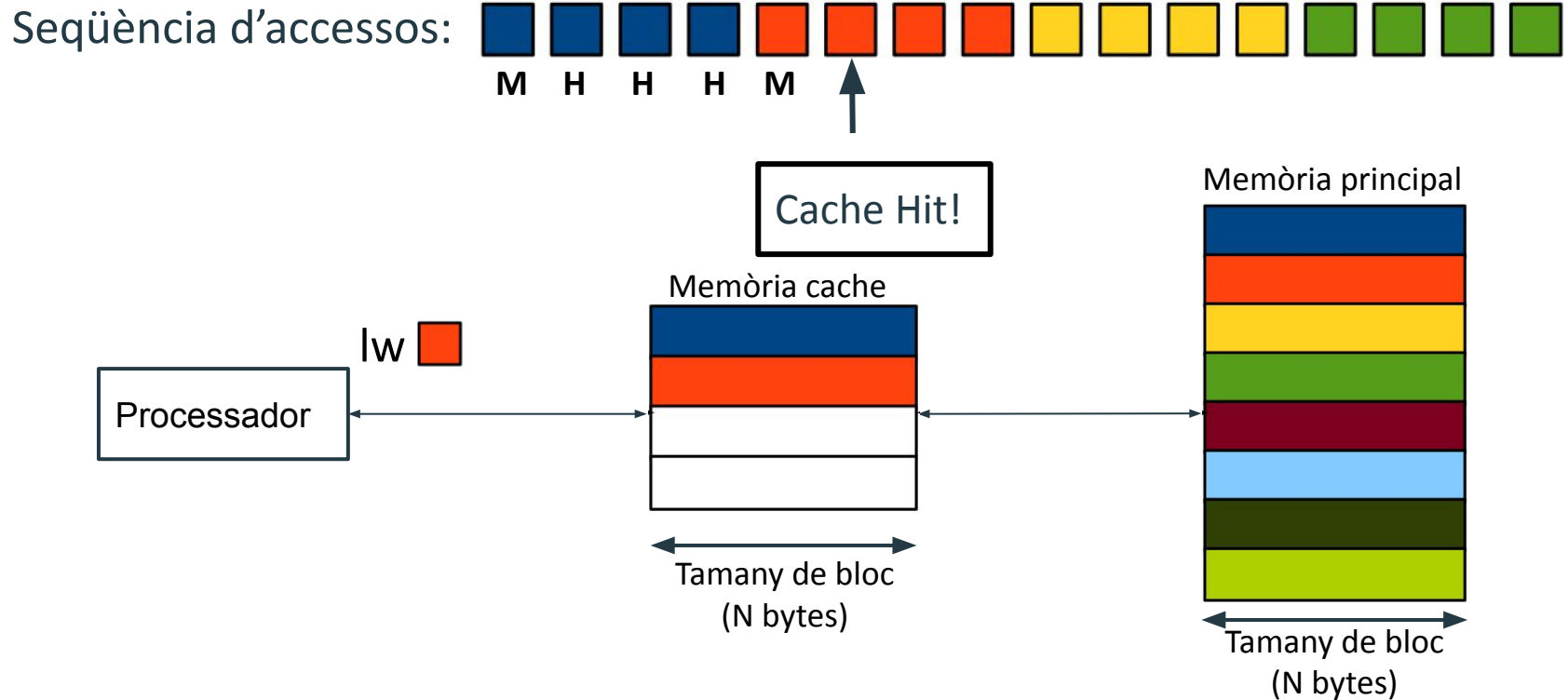
Seqüència d'accessos



Seqüència d'accessos

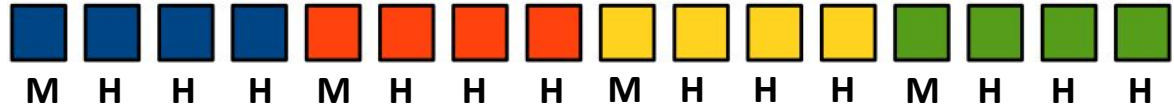


Seqüència d'accessos



Seqüència d'accessos

Seqüència d'accessos:



Taxa d'encerts = 75%

Taxa de fallades = 25%

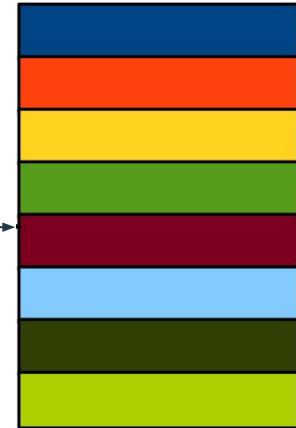
Processador

Memòria cache



Tamany de bloc
(N bytes)

Memòria principal



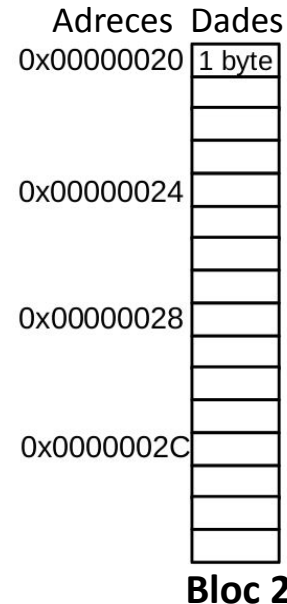
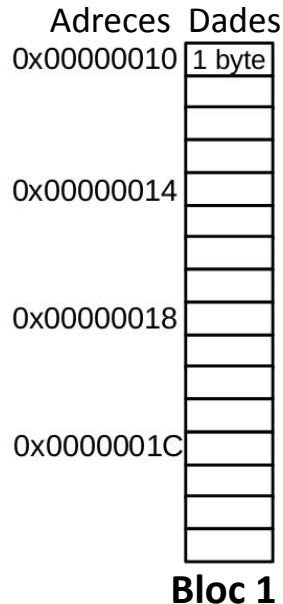
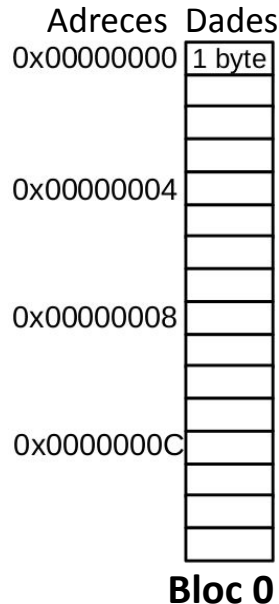
Tamany de bloc
(N bytes)



Disseny bàsic d'una cache

Organització de la memòria - blocs

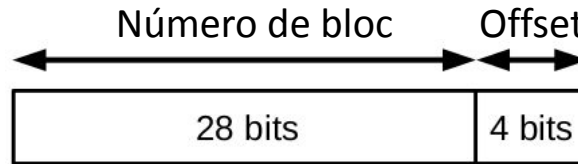
- L'espai d'adreçament de memòria es divideix en blocs de N bytes
 - Exemple: Blocs de memòria de 16 bytes



...

Organització de la memòria - blocs

- Una adreça de memòria es divideix en dos parts:
 - Número de bloc
 - Offset: byte accedit dintre del bloc
- Exemple per tamany de bloc de 16 bytes



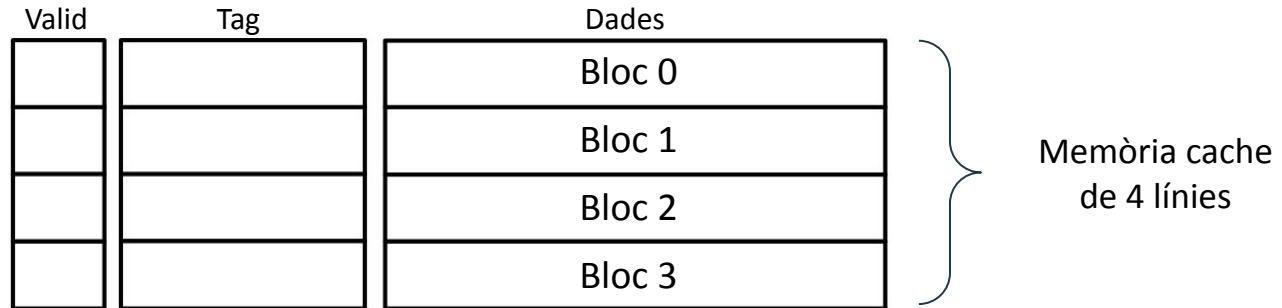
- Adreça de memòria 0x10010020
 - Número de bloc = 0x1001002
 - Offset = 0x0

$$\text{número de bloc} = \frac{\text{adreça}}{\text{tamany de bloc}}$$

$$\text{offset} = \text{adreça} \% \text{tamany de bloc}$$

Memòria cache

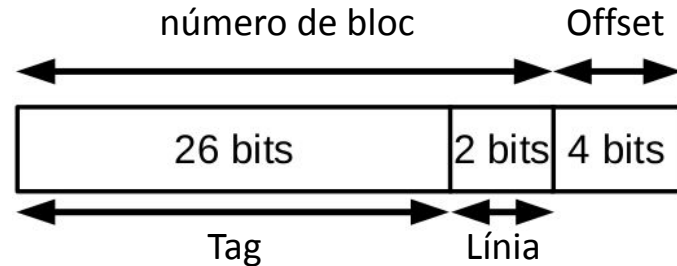
- Una memòria cache té un número determinat de línies
- Per cada línia de cache s'emmagatzema:
 - **Valid bit:** Indica si la línia de cache conté dades vàlides
 - **Tag:** Indica el número de bloc que s'emmagatzema
 - **Dades:** Conté els bytes del bloc de memòria



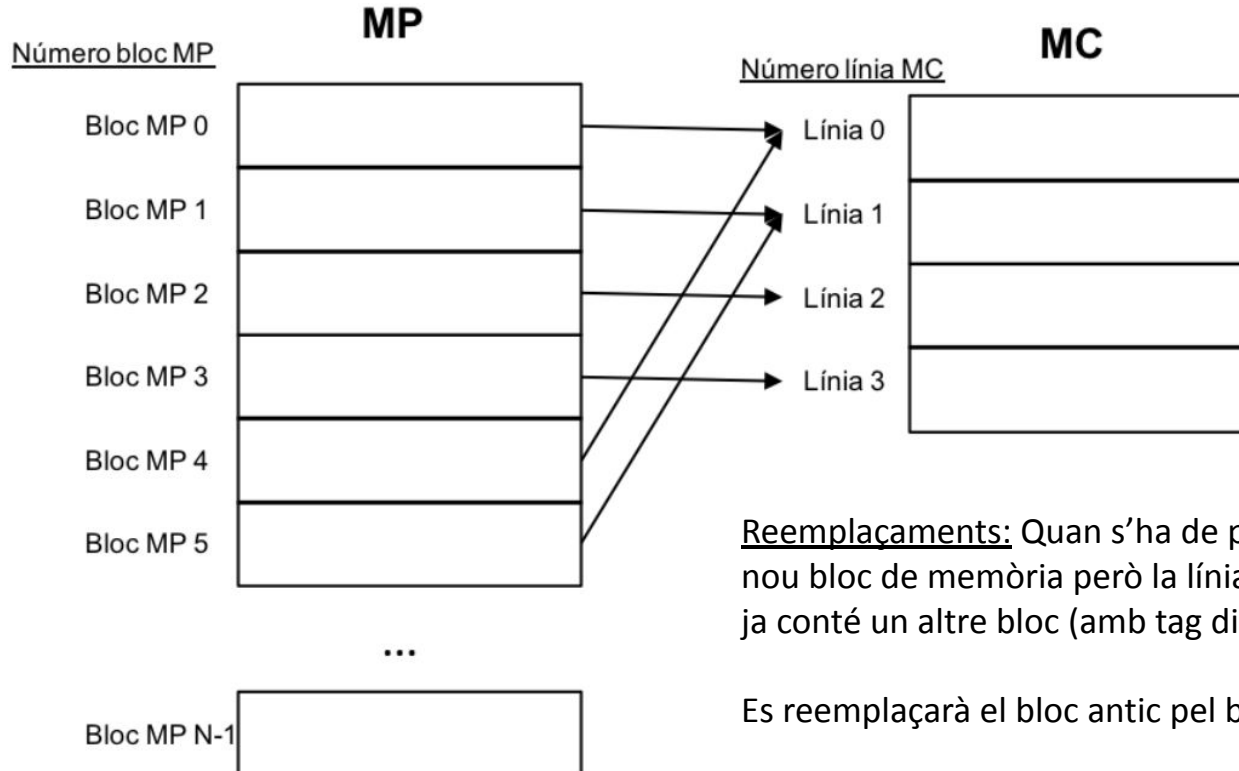
Donat un bloc de memòria, com sabem a quina línia de cache s'emmagatzema?

Política d'ubicació - Correspondència directa

- Correspondència directa
 - Cada bloc de memòria s'assigna a una línia fixe de la MC
 - $\text{número de línia MC} = \text{número_de_bloc} \% \text{num_línies}$
 - Els bits de menor pes del número de bloc indiquen la línia de cache on s'emmagatzema el bloc
- Exemple
 - Tamany de bloc de 16 bytes
 - Memòria cache de 4 línies



Correspondència directa



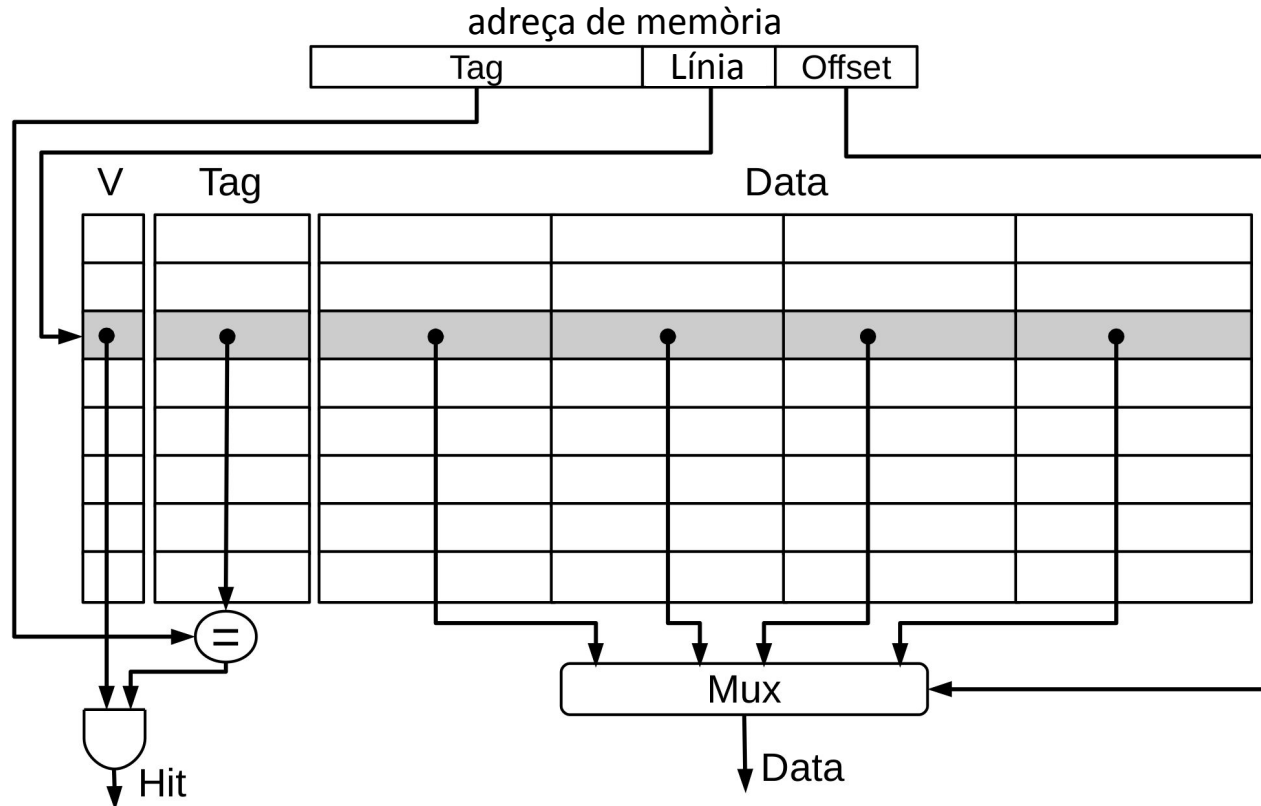
Reemplaçaments: Quan s'ha de portar un nou bloc de memòria però la línia de cache ja conté un altre bloc (amb tag different).

Es reemplaçarà el bloc antic pel bloc nou.

Exemple correspondència directa

- Suposem un tamany de bloc de 64 bytes i una memòria cache de 32 línies. Indica el número de bloc, el tag, el número de línia de cache i l'offset per les següents adreces de memòria:
 - 0x100101C0
 - 0x1001060F

Diagrama de blocs (cache correspondència directa)



Tamany del bloc de memòria

- Quin és el tamany de bloc de memòria òptim?
 - Tamany grans permeten aprofitar millor la localitat espacial
 - Tamany més petits permeten tenir més línies a la cache
- Benchmarking: executar conjunt de programes i medir el rendiment per diferents tamany de bloc

Tamany del bloc de memòria

