

# Tema 6. Memòria Cache

## Estructura de Computadors (EC)

Rubèn Tous

[rtous@ac.upc.edu](mailto:rtous@ac.upc.edu)

Computer Architecture Department  
Universitat Politècnica de Catalunya



UNIVERSITAT POLITÈCNICA  
DE CATALUNYA  
BARCELONATECH

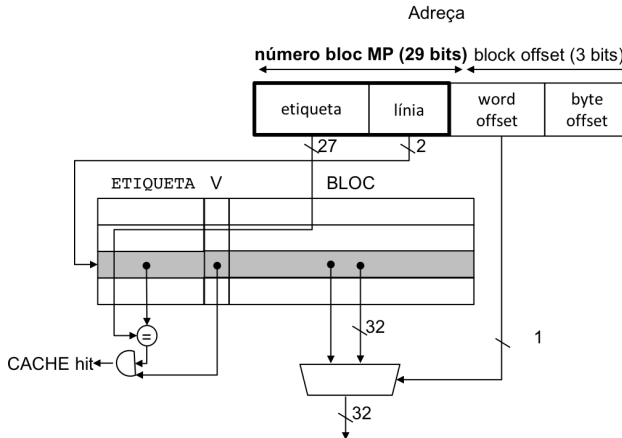
# Índex

- 1 6.2 Disseny bàsic d'una cache
  - 6.2.3 Diagrama de blocs (cache de correspondència directa)
  - 6.2.3 Grandària de bloc
  - 6.2.5 Gestió de les escriptures
  - 6.2.6 Exemple amb política d'escriptura immediata sense assignació
  - 6.2.7 Exemple amb política d'escriptura retardada amb assignació
  - 6.2.8 Disseny de la memòria per suportar caches

# Índex

- 1 6.2 Disseny bàsic d'una cache
  - 6.2.3 Diagrama de blocs (cache de correspondència directa)
  - 6.2.3 Grandària de bloc
  - 6.2.5 Gestió de les escriptures
  - 6.2.6 Exemple amb política d'escriptura immediata sense assignació
  - 6.2.7 Exemple amb política d'escriptura retardada amb assignació
  - 6.2.8 Disseny de la memòria per suportar caches

# Diagrama de blocs (cache de correspondència directa)



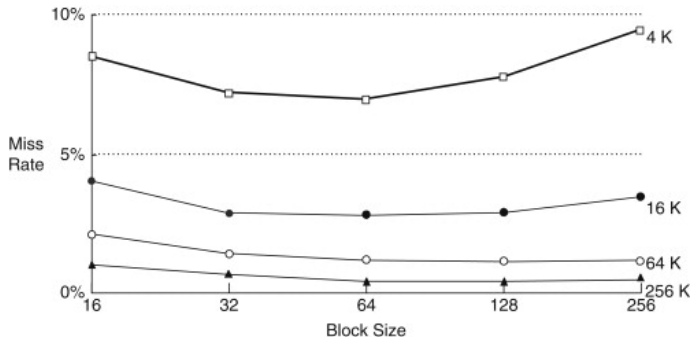
# Índex

- 1 6.2 Disseny bàsic d'una cache
  - 6.2.3 Diagrama de blocs (cache de correspondència directa)
  - **6.2.3 Grandària de bloc**
  - 6.2.5 Gestió de les escriptures
  - 6.2.6 Exemple amb política d'escriptura immediata sense assignació
  - 6.2.7 Exemple amb política d'escriptura retardada amb assignació
  - 6.2.8 Disseny de la memòria per suportar caches

## 6.2.4 Grandària de bloc

- Blocs grans  $\Rightarrow$  major localitat espacial  $\Rightarrow$  taxa de fallades (m)  $\downarrow$ .
- Però hi ha múltiples blocs competint per ocupar espai a l'MC.
- Si grandària del bloc massa gran respecte a la grandària total de l'MC  $\Rightarrow$  taxa de fallades (m)  $\uparrow$ .
- La grandària òptima del bloc mitjançant *benchmarking*.

## 6.2.4 Grandària de bloc



Taxa de miss, grandària del bloc i grandària de la cache. Benchmark SPEC92. Font: Hennessy and Patterson, Computer Architecture: A Quantitative Approach, 5th ed., Morgan Kaufmann, 2012.

## 6.2.4 Grandària de bloc

- La grandària del bloc té també un impacte sobre el temps que es triga en transferir blocs entre MP i MC.
- Això afectarà al *temps de penalització de les fallades*, que estudiarem més endavant.



# Índex

- 1 6.2 Disseny bàsic d'una cache
  - 6.2.3 Diagrama de blocs (cache de correspondència directa)
  - 6.2.3 Grandària de bloc
  - **6.2.5 Gestió de les escriptures**
  - 6.2.6 Exemple amb política d'escriptura immediata sense assignació
  - 6.2.7 Exemple amb política d'escriptura retardada amb assignació
  - 6.2.8 Disseny de la memòria per suportar caches

## 6.2.5 Gestió de les escriptures

- Fins ara hem suposat tot lectures.
- Copiar blocs de dades a MC provoca que, temporalment, hi hagi dades duplicades (redundància de dades).
- Problema de *coherència de les dades*:
  - Haurem d'assegurar-nos de que les modificacions no es perden.
  - Haurem d'assegurar-nos de que el CPU sempre treballa amb la darrera versió.

## 6.2.5 Gestió de les escriptures

- Una escriptura (e.g. un *sw*) pot donar lloc a 2 situacions diferents:
  - La dada està a MC : encert d'MC (hi ha dues còpies de la dada).
  - La dada no està a l'MC: fallada d'MC (de moment només hi ha una còpia de la dada).
- Cadascuna d'aquestes situacions es pot resoldre mitjançant diferents tècniques, la combinació de les quals dona lloc al que s'anomena *política d'escriptura* (*write policy*).

## Situació 1: encert d'escriptura (gestió de la coherència de dades)

Esriptura + encert de cache  $\Rightarrow$  Tècnica 1:

### Esriptura immediata (*write-through*)

S'escriu la dada **simultàniament a la memòria cache i a la memòria principal**. Quan calgui reemplaçar el contingut d'una línia, es podrà fer directament, ja que es té la certesa de que a la memòria principal es conserva la mateixa informació.

## Situació 1: encert d'escriptura (gestió de la coherència de dades)

Esriptura + encert de cache  $\Rightarrow$  Tècnica 2:

### Esriptura retardada (*write-back* o *copy-back*)

S'escriu la dada **únicament a la memòria cache**. Hi haurà al mateix temps dues versions diferents de la dada. Quan calgui reemplaçar el contingut d'una línia, s'haurà de conèixer si la línia conté alguna modificació o no. Caldrà afegir un bit a la memòria cache, el bit D (**dirty bit**).

# Situació 1: encert d'escriptura (gestió de la coherència de dades)

NÚM. LÍNIA	MC				
	V	D	etiqueta	WORD 1	WORD 0
0	0	0	0x0		
1	1	1	0x1		
2	0	0	0x0		
3	0	0	0x0		

El bit D (*dirty bit*) assenyalava quines línies contenen dades modificades si s'utilitza una escriptura retardada.

## Situació 1: encert d'escriptura (gestió de la coherència de dades)

Si el bloc que cal reemplaçar conté una dada modificada, caldrà procedir de la següent manera:

- 1 Copiar **tot el bloc** modificat (el que volem reemplaçar) a la memòria principal.
- 2 Transferir a la memòria cache el nou bloc, sobreescrivint el bloc antic.

Això té també un impacte en el funcionament de la cache quan es produeixen lectures.

## Situació 2: fallada d'escriptura. *Polítiques de fallada d'escriptura* (write-miss policies)

Espectura + fallada de cache  $\Rightarrow$  Tècnica 1:

Espectura sense assignació (*no-write allocate*)

Si es produeix una fallada de cache, escrivim **la dada** a la memòria principal, com si no hi hagués cache.



## Situació 2: fallada d'escriptura. *Polítiques de fallada d'escriptura* (write-miss policies)

Espectura + fallada de cache  $\Rightarrow$  Tècnica 2:

### Espectura amb assignació (*write allocate*)

Si es produeix una fallada de cache, copiem **el bloc** a la memòria cache de la mateixa manera que ho fem amb una lectura.

## Combinacions de tècniques. La política d'escriptura

- Els dos enfocaments per a la gestió de la coherència (escriptura immediata o retardada) es poden combinar amb qualsevol dels enfocaments per a la gestió de les fallades d'escriptura (escriptura amb assignació o sense assignació).
- La combinació escollida dona lloc al que s'anomena *política d'escriptura (write policy)*.
- A EC treballarem principalment dues combinacions diferents:
  - 1 Escriptura immediata sense assignació.
  - 2 Escriptura retardada amb assignació.
- Una excepció serà el laboratori, ja que el Mars fa servir escriptura immediata amb assignació.

# Índex

- 1 6.2 Disseny bàsic d'una cache
  - 6.2.3 Diagrama de blocs (cache de correspondència directa)
  - 6.2.3 Grandària de bloc
  - 6.2.5 Gestió de les escriptures
  - **6.2.6 Exemple amb política d'escriptura immediata sense assignació**
  - 6.2.7 Exemple amb política d'escriptura retardada amb assignació
  - 6.2.8 Disseny de la memòria per suportar caches

# Exemple amb política d'escriptura immediata sense assignació

MC correspondència directa 4 línies de 2 paraules. Inicialment:

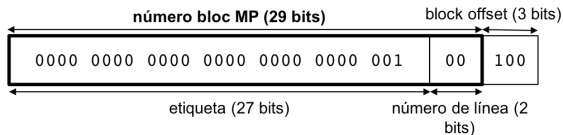
MC				
NÚM. LÍNIA	V	etiqueta	WORD 1	WORD 0
0	0	0x0	0x00000000	0x00000000
1	0	0x0	0x00000000	0x00000000
2	0	0x0	0x00000000	0x00000000
3	0	0x0	0x00000000	0x00000000

Adreça	MP	Número bloc MP
0x00000000	0x00	Bloc MP 0
0x00000001	0x00	
0x00000002	0x00	
0x00000003	0x00	
0x00000004	0x00	Bloc MP 0
0x00000005	0x00	
0x00000006	0x00	
0x00000007	0x00	
0x00000008	0x00	Bloc MP 1
0x00000009	0x00	
0x0000000A	0x00	
0x0000000B	0x00	
0x0000000C	0x00	Bloc MP 1
0x0000000D	0x00	
0x0000000E	0x00	
0x0000000F	0x00	
...		
0x00000020	0xAA	Bloc MP 4
0x00000021	0x00	
0x00000022	0x00	
0x00000023	0x00	
0x00000024	0xBB	Bloc MP 4
0x00000025	0x00	
0x00000026	0x00	
0x00000027	0x00	

A continuació s'executa el següent codi:

```
li $t0, 36
lw $t1, 0($t0)
```

**adreça**

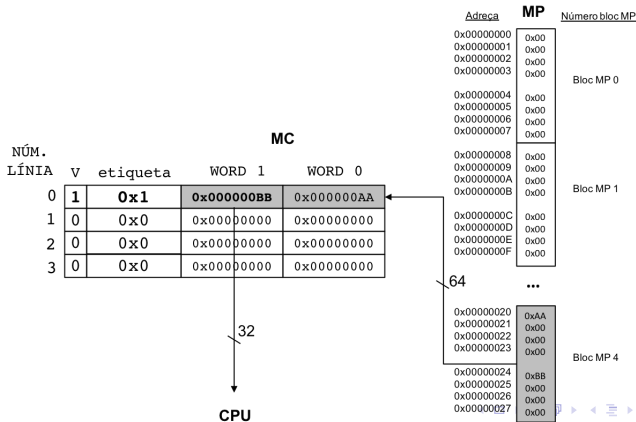


### Anàlisi de l'adreça 36.

## Exemple (esc. immediata sense assignació): **lectura amb fallada**

- 1 Donat que la línia 0 té el bit  $V = 0$  es produirà una fallada de lectura.
- 2 Es copiarà el bloc 4 d'MP a la línia 0. Es posarà el bit  $V$  de la línia 0 a 1. I s'escriurà l'etiqueta 1 a la línia 0.
- 3 Es servirà la dada a la CPU.

# Exemple (esc. immediata sense assignació): lectura amb fallada



## Exemple (esc. immediata sense assignació): **lectura amb encert**

A continuació s'executa el següent codi:

```
li $t0, 32  
lw $t1, 0($t0)
```

L'adreça 32 correspon a la primera paraula del bloc 4 d'MP.



## Exemple (esc. immediata sense assignació): **lectura amb encert**

El que succeirà al subsistema de memòria serà el següent:

- 1 La línia 0 té el bit  $V = 1$  i l'etiqueta 1 (la mateixa que el bloc 4): encert de lectura.
- 2 Es servirà la dada a la CPU.

*NOTA: L'accés a les etiquetes per comprovar si la referència és un encert i el servei de la referència es poden realitzar en el mateix cicle. En general direm que totes dues coses es realitzen durant el temps  $t_h$ .*

# Exemple (esc. immediata sense assignació): lectura amb encert

		MC		
NÚM.				
LÍNIA	V	etiqueta	WORD 1	WORD 0
0	1	0x1	0x000000BB	0x000000AA
1	0	0x0	0x00000000	0x00000000
2	0	0x0	0x00000000	0x00000000
3	0	0x0	0x00000000	0x00000000

CPU

Adreça	MP	Número bloc MP
0x00000000	0x00	Bloc MP 0
0x00000001	0x00	
0x00000002	0x00	
0x00000003	0x00	
0x00000004	0x00	
0x00000005	0x00	
0x00000006	0x00	
0x00000007	0x00	
0x00000008	0x00	Bloc MP 1
0x00000009	0x00	
0x0000000A	0x00	
0x0000000B	0x00	
0x0000000C	0x00	
0x0000000D	0x00	
0x0000000E	0x00	
0x0000000F	0x00	
...		
0x00000020	0xAA	Bloc MP 4
0x00000021	0x00	
0x00000022	0x00	
0x00000023	0x00	
0x00000024	0xBB	
0x00000025	0x00	
0x00000026	0x00	
0x00000027	0x00	

## Exemple (esc. immediata sense assignació): escriptura amb encert

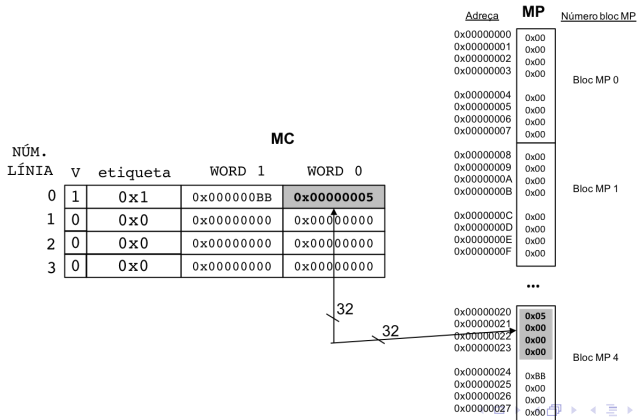
A continuació s'executa el següent codi:

```
li $t0, 32  
li $t1, 5  
sw $t1, 0($t0)
```

El que succeirà al subsistema de memòria serà el següent:

- 1 Donat que la línia 0 té el bit  $V = 1$  i l'etiqueta 1 (la mateixa que el bloc 4) es produirà encert d'escriptura.
- 2 S'escriurà la dada en paral·lel a l'MC i a la memòria.

## Exemple (esc. immediata sense assignació): escriptura amb encert



## Exemple (esc. immediata sense assignació): escriptura amb fallada

A continuació s'executa el següent codi:

```
li $t0, 0  
li $t1, 6  
sw $t1, 0($t0)
```

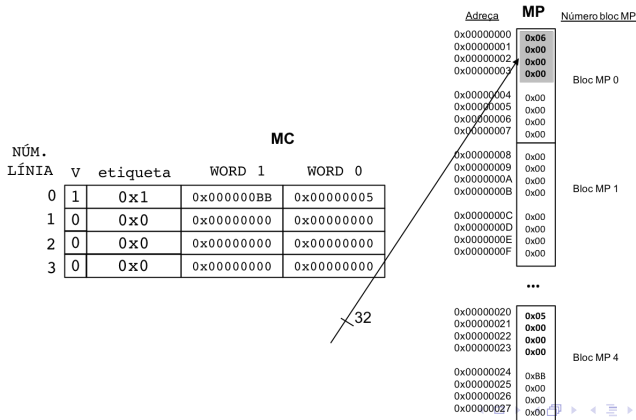
L'adreça 0 correspon a la primera paraula del bloc 0 d'MP.

## Exemple (esc. immediata sense assignació): escriptura amb fallada

El que succeirà al subsistema de memòria serà el següent:

- 1 Tot i que la línia 0 té el bit  $V = 1$  també té etiqueta = 1, diferent a l'etiqueta que correspon a l'adreça (etiqueta 0). Per tant, es produirà una fallada d'escriptura.
- 2 Donat que tenim una política d'escriptura **sense assignació**, no modificarem res a l'MC, simplement escriurem la dada a la memòria principal.

## Exemple (esc. immediata sense assignació): escriptura amb fallada



# Índex

- 1 6.2 Disseny bàsic d'una cache
  - 6.2.3 Diagrama de blocs (cache de correspondència directa)
  - 6.2.3 Grandària de bloc
  - 6.2.5 Gestió de les escriptures
  - 6.2.6 Exemple amb política d'escriptura immediata sense assignació
  - **6.2.7 Exemple amb política d'escriptura retardada amb assignació**
  - 6.2.8 Disseny de la memòria per suportar caches



# Exemple amb política d'escriptura retardada amb assignació

## Estat inicial de l'MC i l'MP

		MC			
NÚM.					
LÍNIA	V	D	etiqueta	WORD 1	WORD 0
0	0	0	0x0	0x00000000	0x00000000
1	0	0	0x0	0x00000000	0x00000000
2	0	0	0x0	0x00000000	0x00000000
3	0	0	0x0	0x00000000	0x00000000

Adreça	MP	Número bloc MP
0x00000000	0x00	Bloc MP 0
0x00000001	0x00	
0x00000002	0x00	
0x00000003	0x00	
0x00000004	0x00	Bloc MP 0
0x00000005	0x00	
0x00000006	0x00	
0x00000007	0x00	
0x00000008	0x00	Bloc MP 1
0x00000009	0x00	
0x0000000A	0x00	
0x0000000B	0x00	
0x0000000C	0x00	Bloc MP 1
0x0000000D	0x00	
0x0000000E	0x00	
0x0000000F	0x00	
...		
0x00000020	0xAA	Bloc MP 4
0x00000021	0x00	
0x00000022	0x00	
0x00000023	0x00	
0x00000024	0xBB	Bloc MP 4
0x00000025	0x00	
0x00000026	0x00	

## Exemple (esc. retardada amb assignació): **lectura amb fallada d'un bloc no modificat**

A continuació s'executa el següent codi:

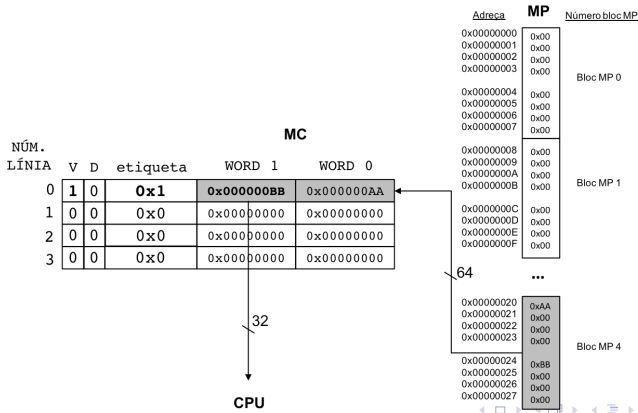
```
li $t0, 36  
lw $t1, 0($t0)
```

L'adreça 36 correspon a la segona paraula del bloc 4 d'MP (línia 0 d'MC).

## Exemple (esc. retardada amb assignació): **lectura amb fallada d'un bloc no modificat**

- 1 Donat que la línia 0 té el bit  $V = 0$  es produirà una fallada de lectura.
- 2 Copiarem el bloc 4 d'MP a la línia 0. En aquest cas la línia encara no contenia cap bloc, però si n'hagués contingut un ( $V = 1$ ) però amb una etiqueta diferent, l'haguéssim pogut reemplaçar directament si  $D = 0$ .
- 3 Es servirà la dada a la CPU.

# Exemple (esc. retardada amb assignació): **lectura** amb fallada d'un bloc no modificat



## Exemple (esc. retardada amb assignació): **lectura amb encert**

A continuació s'executa el següent codi:

```
li $t0, 32  
lw $t1, 0($t0)
```

L'adreça 32 correspon a la primera paraula del bloc 4 d'MP.

## Exemple (esc. retardada amb assignació): **lectura amb encert**

El que succeirà al subsistema de memòria serà el següent (no hi ha cap diferència amb l'exemple amb escriptura immediata sense assignació):

- 1 Donat que la línia 0 té el bit  $V = 1$  i l'etiqueta 1 (la mateixa que el bloc 4) es produirà encert de lectura.
- 2 Es servirà la dada a la CPU.

# Exemple (esc. retardada amb assignació): lectura amb encert

NÚM.			MC		
LÍÑIA	V	D	etiqueta	WORD 1	WORD 0
0	1	0	0x1	0x000000BB	0x000000AA
1	0	0	0x0	0x00000000	0x00000000
2	0	0	0x0	0x00000000	0x00000000
3	0	0	0x0	0x00000000	0x00000000

32

CPU

Adreça	MP	Número bloc MP
0x00000000	0x00	Bloc MP 0
0x00000001	0x00	
0x00000002	0x00	
0x00000003	0x00	
0x00000004	0x00	Bloc MP 0
0x00000005	0x00	
0x00000006	0x00	
0x00000007	0x00	
0x00000008	0x00	Bloc MP 1
0x00000009	0x00	
0x0000000A	0x00	
0x0000000B	0x00	
0x0000000C	0x00	Bloc MP 1
0x0000000D	0x00	
0x0000000E	0x00	
0x0000000F	0x00	
...		
0x00000020	0xAA	Bloc MP 4
0x00000021	0x00	
0x00000022	0x00	
0x00000023	0x00	
0x00000024	0xBB	Bloc MP 4
0x00000025	0x00	
0x00000026	0x00	
0x00000027	0x00	

## Exemple (esc. retardada amb assignació): **escriptura amb encert**

A continuació s'executa el següent codi:

```
li $t0, 32  
li $t1, 5  
sw $t1, 0($t0)
```

El que succeirà al subsistema de memòria serà el següent:

- 1 Donat que la línia 0 té el bit  $V = 1$  i l'etiqueta 1 (la mateixa que el bloc 4) es produirà encert d'escriptura.
- 2 S'escriurà la dada **només** a l'MC i es marcarà la línia com a modificada ( $D = 1$ ).



# Exemple (esc. retardada amb assignació): **escriptura amb encert**

NÚM.				MC	
LÍÑIA	V	D	etiqueta	WORD 1	WORD 0
0	1	1	0x1	0x000000BB	0x00000005
1	0	0	0x0	0x00000000	0x00000000
2	0	0	0x0	0x00000000	0x00000000
3	0	0	0x0	0x00000000	0x00000000

32

32

Adreça	MP	Número bloc MP
0x00000000	0x00	Bloc MP 0
0x00000001	0x00	
0x00000002	0x00	
0x00000003	0x00	
0x00000004	0x00	Bloc MP 1
0x00000005	0x00	
0x00000006	0x00	
0x00000007	0x00	
0x00000008	0x00	Bloc MP 1
0x00000009	0x00	
0x0000000A	0x00	
0x0000000B	0x00	
0x0000000C	0x00	Bloc MP 1
0x0000000D	0x00	
0x0000000E	0x00	
0x0000000F	0x00	
...		
0x00000020	0xAA	Bloc MP 4
0x00000021	0x00	
0x00000022	0x00	
0x00000023	0x00	
0x00000024	0xBB	Bloc MP 4
0x00000025	0x00	
0x00000026	0x00	
0x00000027	0x00	

## Exemple (esc. retardada amb assignació): **lectura amb fallada d'un bloc modificat**

A continuació s'executa el següent codi:

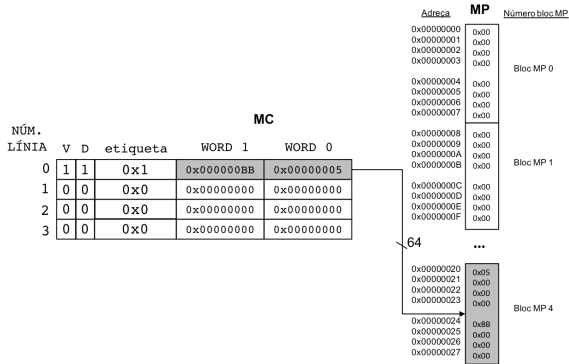
```
li $t0, 0  
lw $t1, 0($t0)
```

L'adreça 0 correspon a la primera paraula del bloc 0 d'MP.

## Exemple (esc. retardada amb assignació): **lectura amb fallada d'un bloc modificat**

- Tot i que la línia 0 té el bit  $V = 1$  també té etiqueta = 1, diferent a l'etiqueta que correspon a l'adreça (etiqueta 0). Per tant, es produirà una fallada de lectura.
- Donat que la línia 0 conté un bloc modificat ( $D = 1$ ) no podem reemplaçar el seu contingut directament. Ens cal primer copiar el bloc modificat (era el bloc 4 d'MP) a l'MP.

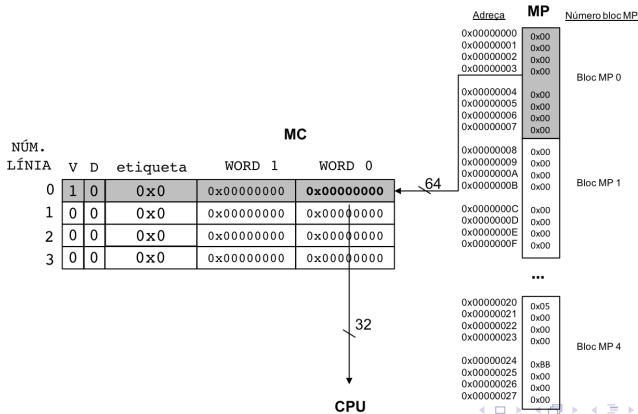
# Exemple (esc. retardada amb assignació): **lectura** amb fallada d'un bloc modificat



## Exemple (esc. retardada amb assignació): **lectura amb fallada d'un bloc modificat**

- Un cop copiat el bloc modificat a l'MP podem copiar el nou bloc, el bloc 0 d'MP, a la línia 0.
- Finalment podem servir la dada.

## Exemple (esc. retardada amb assignació): **lectura** amb fallada d'un bloc modificat



## Exemple (esc. retardada amb assignació): **escriptura amb fallada d'un bloc no modificat**

A continuació s'executa el següent codi:

```
li $t0, 36  
li $t1, 7  
sw $t1, 0($t0)
```

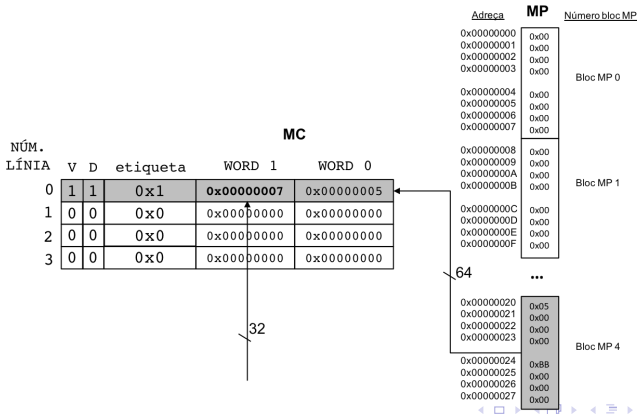
L'adreça 36 correspon a la segona paraula del bloc 4 d'MP  
(línia 0 d'MC)

## Exemple (esc. retardada amb assignació): **escriptura amb fallada d'un bloc no modificat**

- 1 Tot i que la línia 0 té el bit  $V = 1$  també té etiqueta = 0, diferent a l'etiqueta que correspon a l'adreça (etiqueta 1). Per tant, es produirà una fallada d'escriptura.
- 2 Donat que la línia té el bit  $D = 0$ , copiarem el bloc 4 d'MP a la línia 0 directament, reemplaçant el seu contingut.
- 3 Escriurem únicament a MC i marcarem la línia com a modificada ( $D = 1$ ).



# Exemple (esc. retardada amb assignació): **escriptura amb fallada d'un bloc no modificat**



## Exemple (esc. retardada amb assignació): **escriptura amb fallada d'un bloc modificat**

A continuació s'executa el següent codi:

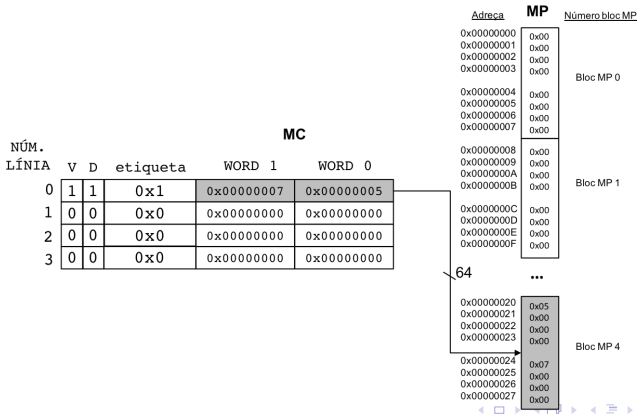
```
li $t0, 0  
li $t1, 3  
sw $t1, 0($t0)
```

L'adreça 0 correspon a la primera paraula del bloc 0 d'MP

## Exemple (esc. retardada amb assignació): **escriptura amb fallada d'un bloc modificat**

- Tot i que la línia 0 té el bit  $V = 1$  també té etiqueta = 1, diferent a l'etiqueta que correspon a l'adreça (etiqueta 0). Per tant, es produirà una fallada d'escriptura.
- Donat que la línia 0 conté un bloc modificat ( $D = 1$ ) no podem reemplaçar el seu contingut directament. Ens cal primer copiar el bloc modificat (era el bloc 4 d'MP) a l'MP.

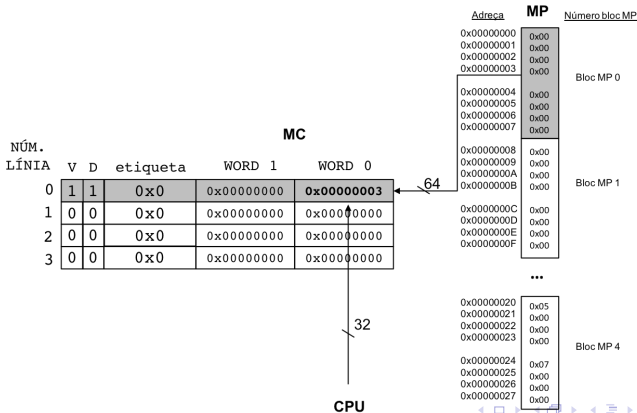
## Exemple (esc. retardada amb assignació): **escriptura amb fallada d'un bloc modificat**



## Exemple (esc. retardada amb assignació): **escriptura amb fallada d'un bloc modificat**

- Un cop copiat el bloc modificat a l'MP podem copiar el nou bloc, el bloc 0 d'MP, a la línia 0.
- Finalment escriurem la dada (1 paraula) només a l'MC.

# Exemple (esc. retardada amb assignació): **escriptura amb fallada d'un bloc modificat**



# Índex

- 1 6.2 Disseny bàsic d'una cache
  - 6.2.3 Diagrama de blocs (cache de correspondència directa)
  - 6.2.3 Grandària de bloc
  - 6.2.5 Gestió de les escriptures
  - 6.2.6 Exemple amb política d'escriptura immediata sense assignació
  - 6.2.7 Exemple amb política d'escriptura retardada amb assignació
  - 6.2.8 Disseny de la memòria per suportar caches

## 6.2.8 Disseny de la memòria per suportar caches

- El temps de penalització de les fallades ( $t_p$ ) depèn en gran mesura del que fins ara hem anomenat  $t_{block}$ .
- $t_{block}$  inclou diferents aspectes, com el temps que triga la memòria en llegir/escriure el bloc o el temps que es triga en transferir-lo pel bus.
- Anem a veure com es dissenya la memòria per poder minimitzar el temps de penalització sense encarir-ne massa la producció.



## 6.2.8 Disseny de la memòria per suportar caches

- Dues interconnexions principals: CPU  $\longleftrightarrow$  MC i MC  $\longleftrightarrow$  MP.
- Es realitzen mitjançant *busos* de dades.
- Normalment, l'amplada en bits de cada bus de coincidirà amb l'amplada de la sortida de la memòria a la que va connectat.
- L'amplada de la sortida de la memòria no necessàriament coincideix amb el que la memòria pot llegir/escriure cada vegada.

### 6.2.8 Disseny de la memòria per suportar caches

Possibilitat 1: l'MC, l'MP i els busos amplada d'una paraula.



## 6.2.8 Disseny de la memòria per suportar caches

- Temps que triga el processador en enviar l'adreça = 1 cicle.
- Temps que triga l'MP en llegir **una paraula** = 15 cicles (caldrà fer-ho 4 vegades).
- Temps que es triga en enviar **una paraula** pel bus = 1 cicle (caldrà fer-ho 4 vegades).
- Grandària del bloc: 4 paraules.

El temps necessari per a copiar un bloc d'MP a MC així:

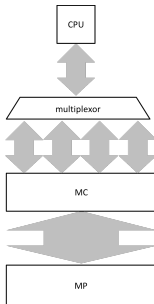
$$t_p = 1 + 4 * 15 + 4 * 1 = 65 \text{ cicles}$$

El nombre de bytes transferits per cicle seria:

$$\frac{4 * 4}{65} = 0.25 \text{ bytes/cicle}$$

### 6.2.8 Disseny de la memòria per suportar caches

Possibilitat 2: MC i MP de major àmplada", per exemple capaces de llegir/escriure dues paraules de cop.



## 6.2.8 Disseny de la memòria per suportar caches

- Temps que triga el processador en enviar l'adreça = 1 cicle (igual que abans).
- Temps que triga l'MP en llegir **dues** paraules = 15 cicles (caldrà fer-ho 2 vegades).
- Temps que es triga en enviar **dues** paraules pel bus = 1 cicle (caldrà fer-ho 4 vegades).

El temps necessari per a copiar un bloc d'MP a MC així:

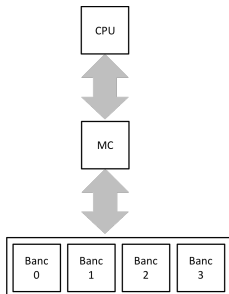
$$t_p = 1 + 2 * 15 + 2 * 1 = 33 \text{ cicles}$$

El nombre de bytes transferits per cicle seria:

$$\frac{4 * 4}{33} = 0.48 \text{ bytes/cicle}$$

## 6.2.8 Disseny de la memòria per suportar caches

Possibilitat 3: memòria *entrellaçada*:  $b$  bancs de memòria (per exemple  $b = 4$ ), d'amplada  $p$  paraules cadascun (per exemple  $p = 1$ ), capaços de treballar simultàniament. Sortida de menor amplada (per exemple 1 paraula).



## 6.2.8 Disseny de la memòria per suportar caches

- Temps que triga el processador en enviar l'adreça = 1 cicle (igual que abans).
- Temps que triga l'MP en llegir **quatre** paraules = 15 cicles.
- Temps que es triga en enviar **una** paraula pel bus = 1 cicle (caldrà fer-ho 4 vegades).

El temps necessari per a copiar un bloc d'MP a MC així:

$$t_p = 1 + 1 * 15 + 4 * 1 = 20 \text{ cicles}$$

El nombre de bytes transferits per cicle seria:

$$\frac{4 * 4}{33} = 0.80 \text{ bytes/cicle}$$