

# RESUMEN CACHÉ – AC

@MP

#bloque MP	#byte
------------	-------

0	0	1	1	0	-
---	---	---	---	---	---

Emplazamiento  
DIRECTO

Línea MC = Bloque de memoria mod #líneas MC

TAG	#línea MC	#byte
-----	-----------	-------

0	0	1	1	0	-
---	---	---	---	---	---

4 líneas

Emplazamiento  
ASOCIATIVO  
por  
CONJUNTOS

Conjunto MC = Bloque de memoria mod #Conjuntos

TAG	#conj. MC	#byte
-----	-----------	-------

0	0	1	1	0	-
---	---	---	---	---	---

2 conjuntos

Emplazamiento  
COMPLETAMENTE  
ASOCIATIVO

TAG	#byte
-----	-------

0	0	1	1	0	-
---	---	---	---	---	---

**Write Through** - the information is written to both the block in the cache and to the block in the lower-level memory.

**Copy back** - the information is written only to the block in the cache. The modified cache block is written to main memory only when it is replaced. To reduce the frequency of writing back blocks on replacement, a dirty bit is commonly used. This status bit indicates whether the block is dirty (modified while in the cache) or clean (not modified). If it is clean the block is not written on a miss.

**Write Allocate** - the block is loaded on a write miss, followed by the write-hit action.

**No Write Allocate** - the block is modified in the main memory and not loaded into the cache.

- **WRITE THROUGH** (escritura a través o escritura inmediata)
  - Se actualiza **simultáneamente** EL DATO en la MC y la MP.
  - El tiempo de servicio es el tiempo de acceso a MP.
  - La MP siempre está actualizada.
  - Se puede reducir el tiempo de escritura utilizando buffers.
- **COPY BACK** (escritura diferida)
  - En una escritura sólo se actualiza EL DATO en MC.
  - Para cada línea se añade un bit de control (*dirty bit*) que indica si la línea ha sido modificada o no.
  - Se **actualiza el bloque en la MP** cuando la línea (si ha sido modificada) ha de ser **reemplazada**.
  - Las escrituras son rápidas (velocidad de MC).
  - El tiempo de penalización en caso de fallo aumenta.
  - Durante un tiempo existe una inconsistencia entre MP y MC.

¿Qué hacer en caso de fallo en escritura?

- **WRITE ALLOCATE** (con migración en caso de fallo)
  - Se trae el bloque de MP a MC y después se realiza la escritura.
- **WRITE NO ALLOCATE** (sin migración en caso de fallo)
  - El bloque **NO** se trae a MC. Esto obliga a realizar la escritura directamente en MP.

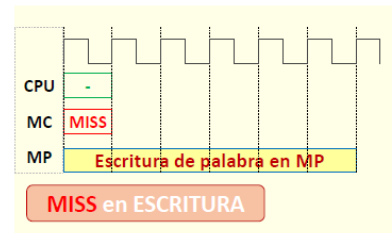
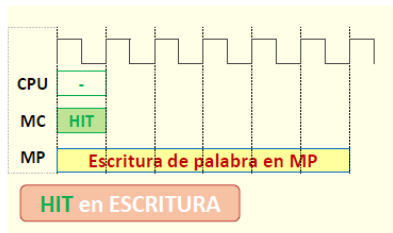
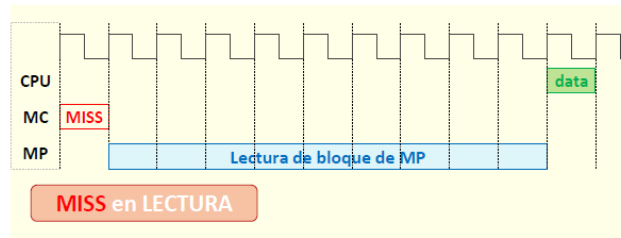
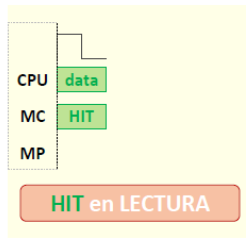
### Write Through with No Write Allocate:

- on hits it writes to cache and main memory;
- on misses it updates the block in main memory not bringing that block to the cache;
- Subsequent writes to the block will update main memory because Write Through policy is employed. So, some time is saved not bringing the block in the cache on a miss because it appears useless anyway.

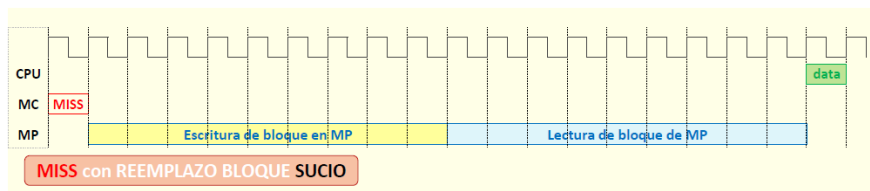
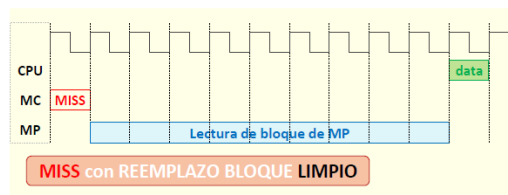
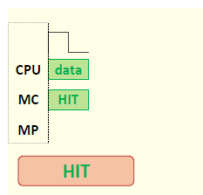
### Copy Back with Write Allocate:

- on hits it writes to cache setting *dirty bit* for the block, main memory is not updated;
- on misses it updates the block in main memory and brings the block to the cache;
- Subsequent writes to the same block, if the block originally caused a miss, will hit in the cache next time, setting dirty bit for the block. That will eliminate extra memory accesses and result in very efficient execution compared with Write Through with Write Allocate combination.

## ■ Write Through + Write NO Allocate



## ■ Copy Back + Write Allocate



- Componentes:
  - ✓ Coste de un acceso en acierto:  $t_{sa}$
  - ✓ Coste de un acceso en fallo:  $t_{sf} = t_{sa} + t_{pf}$
  - ✓ Penalización de un fallo:  $t_{pf}$

- Tiempo medio de acceso a Memoria:

$$T_{ma} = h \cdot t_{sa} + m \cdot t_{sf}$$

$$= t_{sa} + m \cdot t_{pf}$$

## ■ Medida de Rendimiento

- Tiempo de ejecución de un programa:

$$T_{\text{ejec}} = N \cdot \text{CPI} \cdot T_c$$

N: instrucciones ejecutadas

CPI: ciclos por instrucción en media

T<sub>c</sub>: tiempo de ciclo

✓  $\text{CPI} = \text{CPI}_{\text{ideal}} + \text{CPI}_{\text{mem}}$

- ✓ CPI<sub>mem</sub> son los ciclos que perdemos por tener una cache imperfecta ( $m \neq 0$ ).

$$\text{CPI}_{\text{mem}} = nr \cdot (T_{\text{ma}} - t_{\text{sa}})$$

$$\text{CPI}_{\text{mem}} = nr \cdot m \cdot t_{\text{pf}} \quad (\text{caso particular de una MC de sólo lectura})$$

(nr: número medio de referencias por instrucción)