

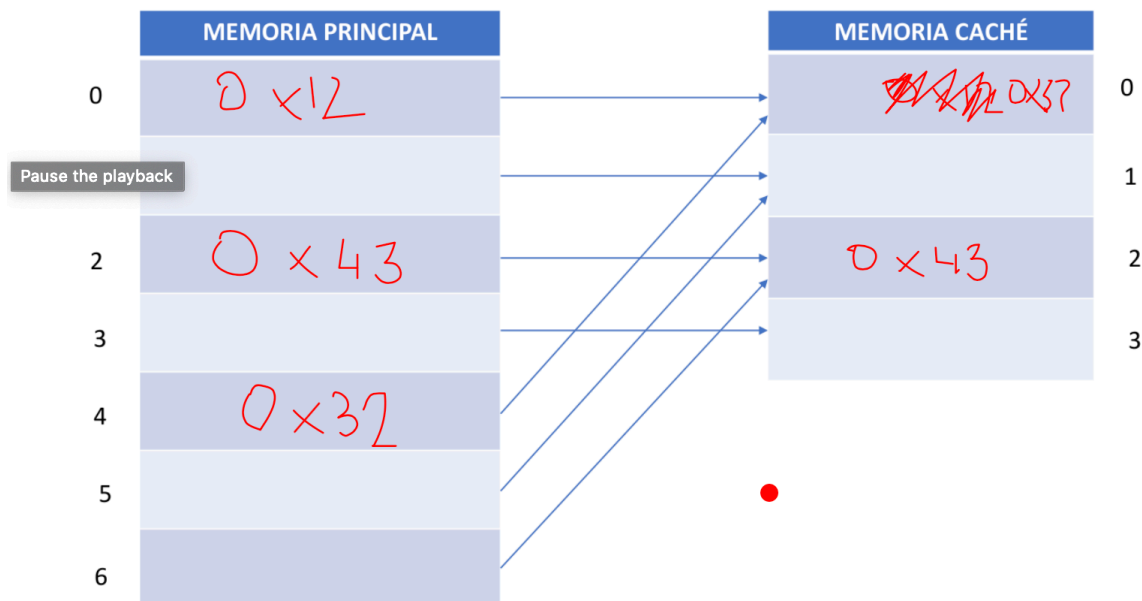
MEMORIA CACHE

Tipos de emplazamiento:

-Correspondencia directa: se guarda el dato en la memoria caché según el orden al que se accede a los datos en la línea que toca según la imagen.

Si estuvieran todas las líneas llenas, como el dato nuevo que tenemos que copiar ya tiene asignado una línea, no se siguen las políticas de reemplazamiento, simplemente se reescribe la línea a la cual debe añadirse el dato.

Correspondencia directa



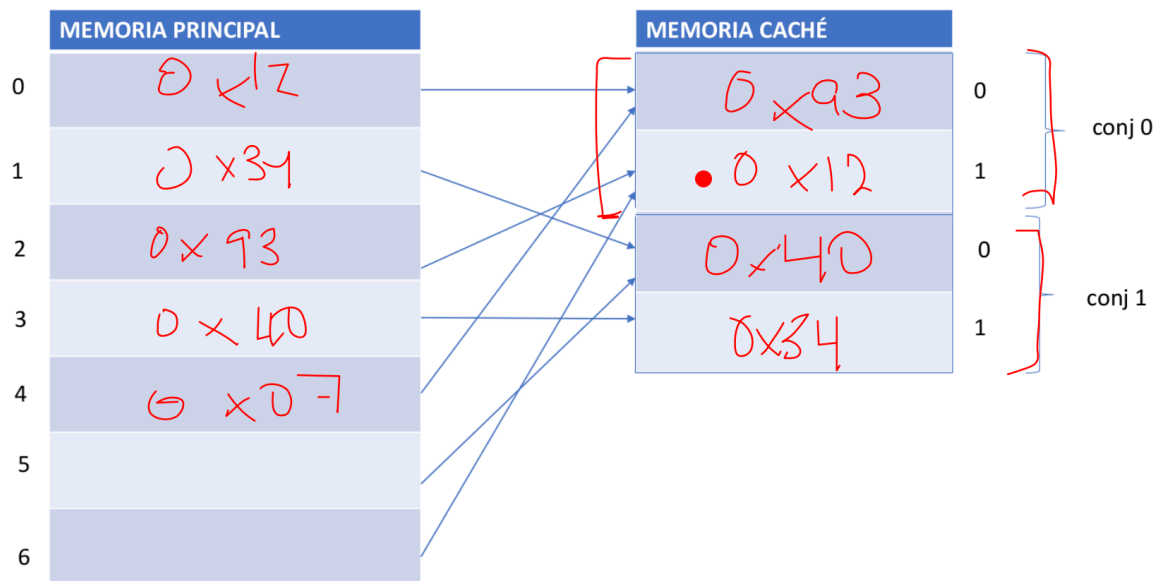
-Asociativa por conjuntos:

Se dividen en conjuntos.

Dentro del conjunto, el dato se puede escribir en cualquier línea.

Si el conjunto está lleno, para saber cuál hemos de eliminar, utilizamos las políticas de reemplazamiento.

Asociativa por conjuntos

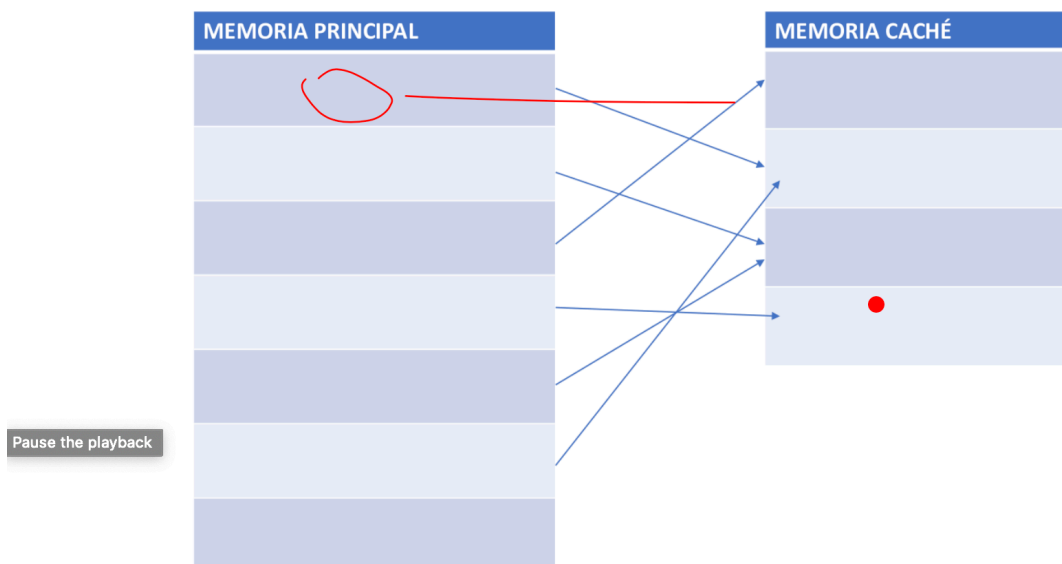


-Completamente asociativa: random

Coge una línea que este vacía y añade el valor allí.

Si estuvieran todas las líneas llenas, hay que seguir las políticas de reemplazamiento para saber cuán reescribir.

Completamente asociativa



POLÍTICAS DE REEMPLAZAMIENTO:

Truquito para LRU: ir marcando con un contador cual es el último dato al que hemos accedido.

Algoritmos de reemplazamiento

- **FIFO** (First in first out): la línea en la que se ha hecho el primer acceso es la primera que se reemplaza
- **LRU** (Last recently used): la línea que hace más tiempo que no se accede es la que se reemplaza (hace falta llevar un recuento de los accesos)
- **Aleatorio**: la línea que se va a reemplazar se escoge aleatoriamente

POLÍTICAS DE ESCRITURA:

Fallo de lectura: accedo a un dato que no encuentro en memoria caché, lo busco en la principal, lo copio, y lo añado en memoria caché. Entonces lo leo.

Fallo de escritura: depende de las siguientes políticas:

Políticas de escritura

- **Write Through / CopyBack**
 - **Write Through** (escritura inmediata): se escribe en MP y MC a la vez
 - **CopyBack** (escritura retardada): solo se escribe en MC (dirtybit)
- **Write Allocate / Write no Allocate**
 - **Write Allocate** (con asignación): copiamos el bloque a MC y escribimos en MC
 - **Write no Allocate** (sin asignación): no copiamos el bloque a MC, escribimos directamente en MP

Write Through: se modifica memoria caché y memoria principal a la vez.

CopyBack: se modifica en memoria caché pero no en la principal. Tenemos el dirtybit que es un bit que nos indica si el dato correcto solo está en memoria caché (dirtybit=1). Lo actualizaremos en memoria principal una vez saquemos el dato de la memoria caché para reemplazarlo por otro.

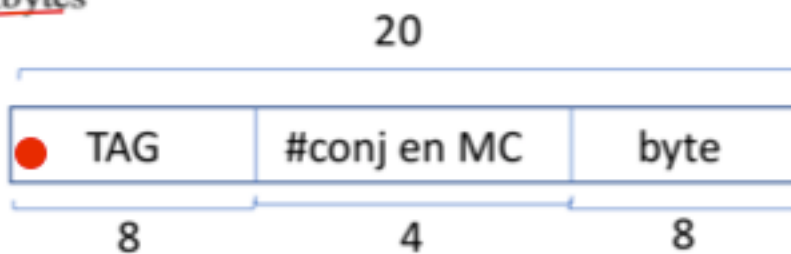
ESCRITURA INMEDIATA SIN ASIGNACION: no se copia a MC si hay fallo de escritura

ESCRITURA RETARDADA CON ASIGNACIÓN: no se actualiza el dato en MP hasta que no se elimina de MC (en memoria caché si está actualizado, nos lo indica el dirtybit). Cada vez que hacemos una escritura de un dato se pone el dirtybit a 1, es decir, se actualiza el dato solo en memoria caché.

Para calcular el número de conjuntos en la línea caché: tam MC (por ejemplo 8KB) / tamLinea (por ejemplo 256B) = $2^{13}/2^8 = 2^5$.

Si tenemos por ejemplo 2 bloques por conjunto = 32 líneas (2^5) / 2 líneas por conjunto = 16 conjuntos = 2^4

8 Kbytes



a memòria
lectures o
número de
(MP) lle-

#lineas en MC = tamMC / tamLinea =
= 8KB / 256B = $2^{13} / 2^8 = 2^5$ lineas

32 lineas, 2 lineas por conjunto, 16
conjuntos = 2^4