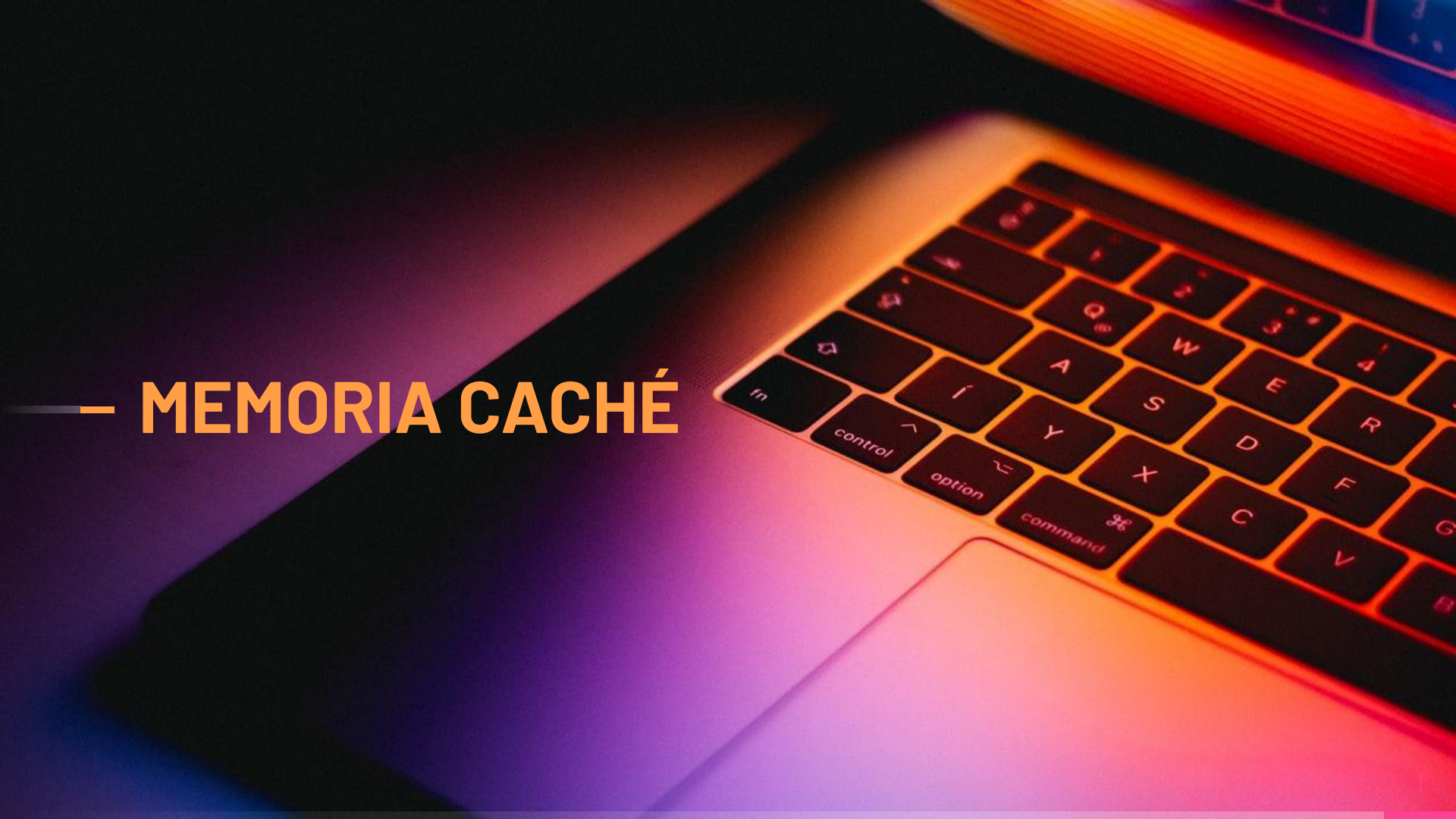
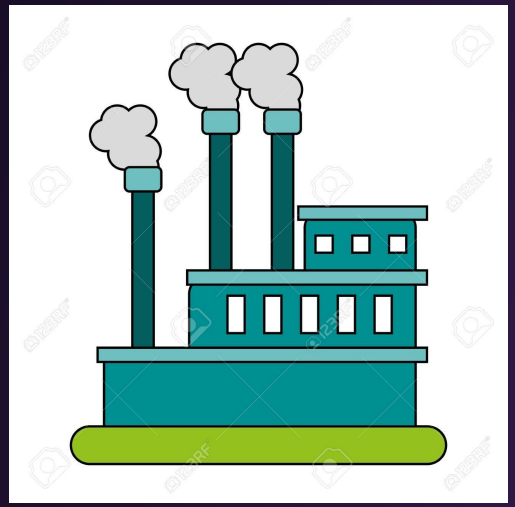
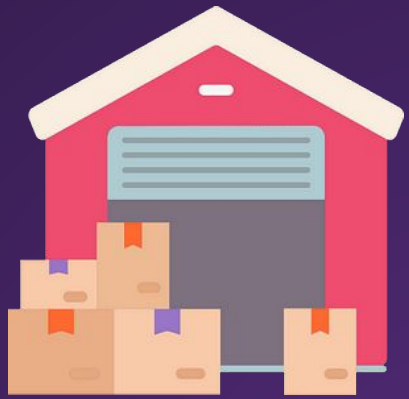
MEMORIA CACHÉ



THE BASICS OF CACHING

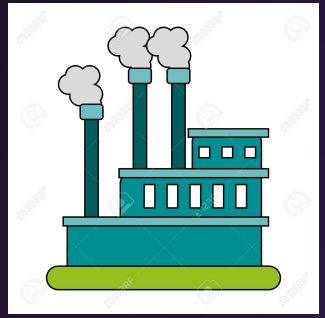


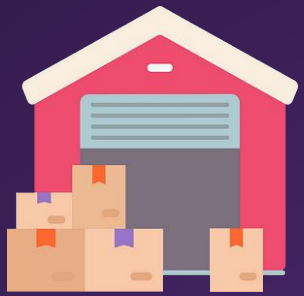


Fábrica
(CPU)

Depósito(MEMO
RIA PRINCIPAL)

THE BASICS OF CACHING





Fábrica(CPU)

Depósito(MEMORIA PRINCIPAL)

THE BASICS OF CACHING

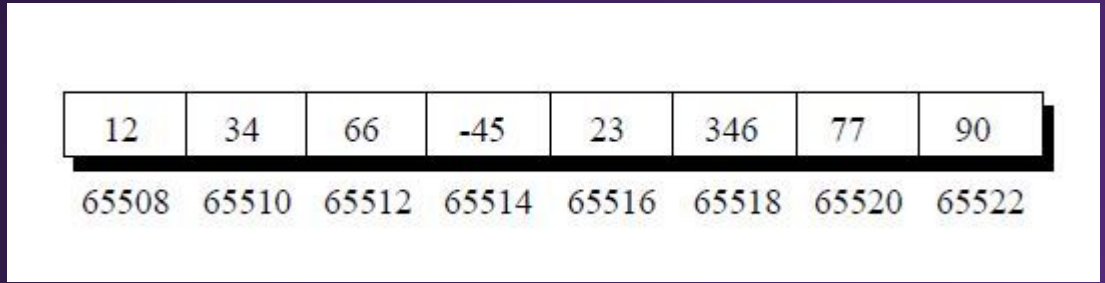
- Un caché es una colección de ubicaciones de memoria a las la CPU puede acceder en menos tiempo que algunas otras ubicaciones de memoria.
- Suele ubicarse en el mismo chip que la CPU o separado (rápido acceso)

THE BASICS OF CACHING: Principio universal

```
float z[1000];
. . .
sum = 0.0;
for (i = 0; i < 1000; i++)
    sum += z[i];
```

Principio de localidad

```
float z[1000];
. . .
sum = 0.0;
for (i = 0; i < 1000; i++)
    sum += z[i];
```



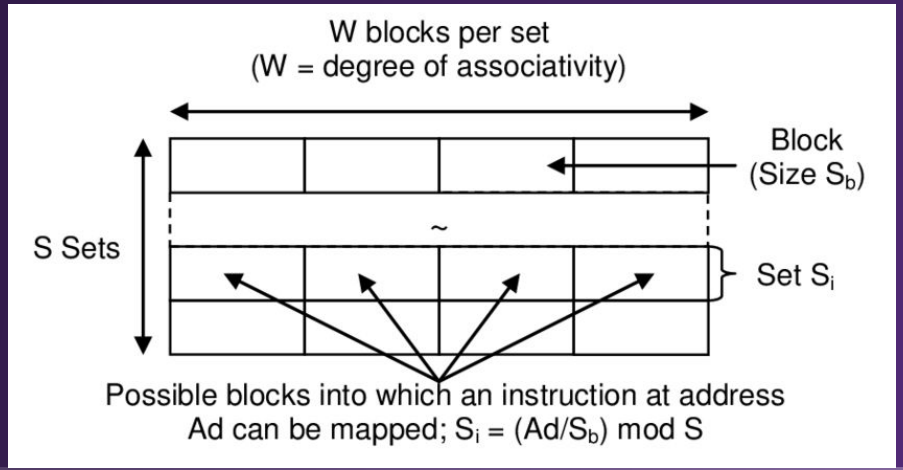
Principio de localidad

- Refiere a que luego del acceso a una ubicación es seguido por un acceso a una ubicación cercana.

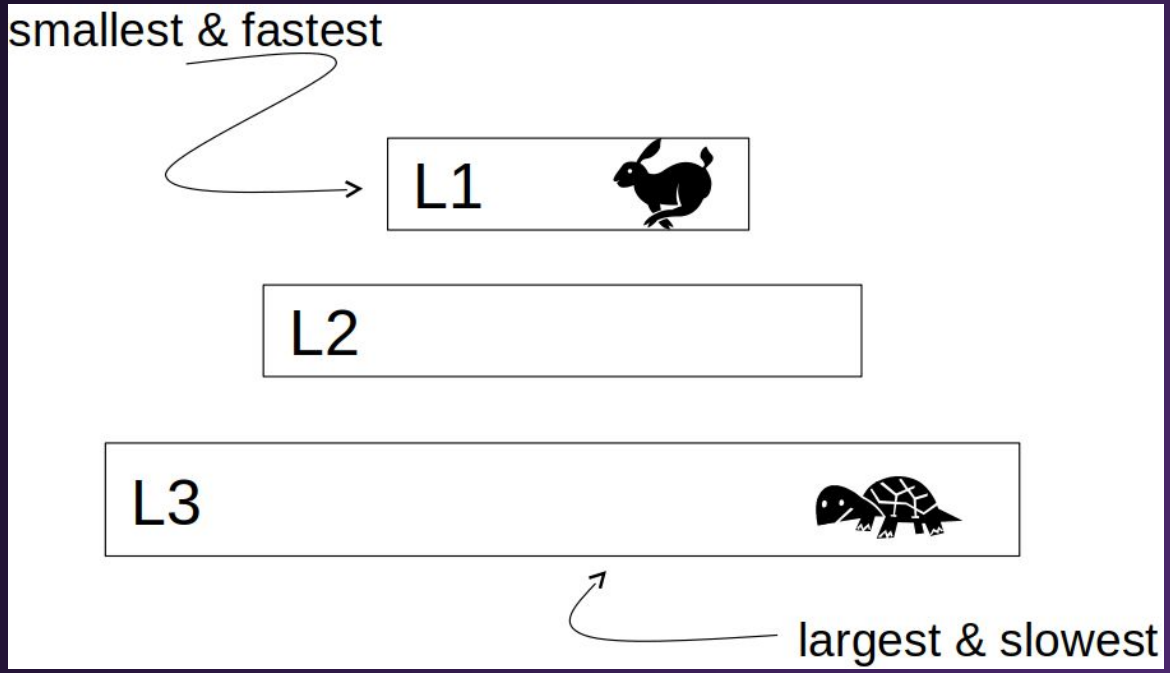
- ***Spatial locality:*** Al acceder a una ubicación de memoria, un programa normalmente accede a una ubicación cercana.
- ***Temporal locality:*** la reutilización de elementos de datos específicos dentro de un corto período de tiempo.

Cache Blocks o Cache Lines

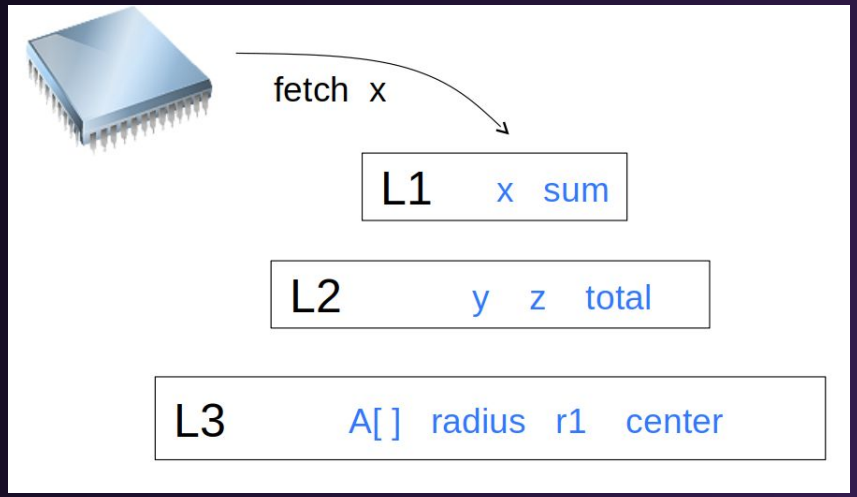
- Un acceso a la memoria operará efectivamente en bloques de datos e instrucciones.
- Una línea de caché típica almacena de 8 a 16 veces más información que una sola memoria.

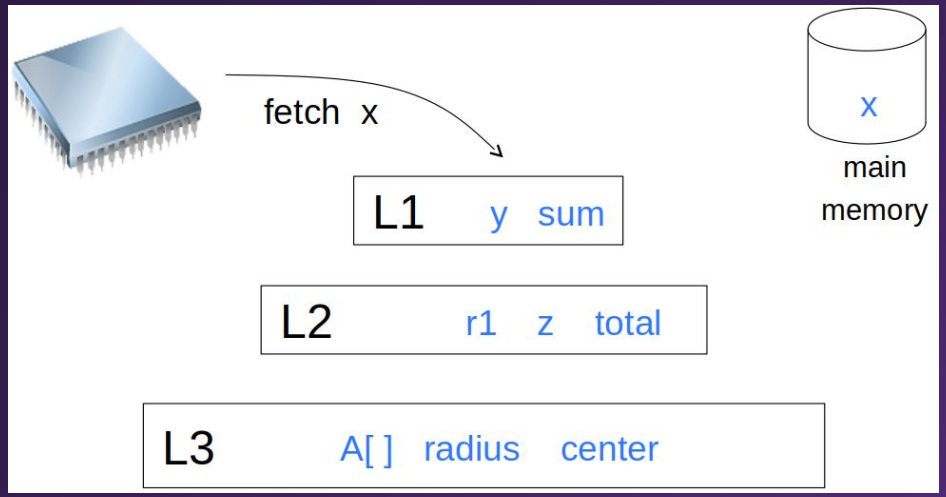


Estructura



Cache Hit and Cache miss





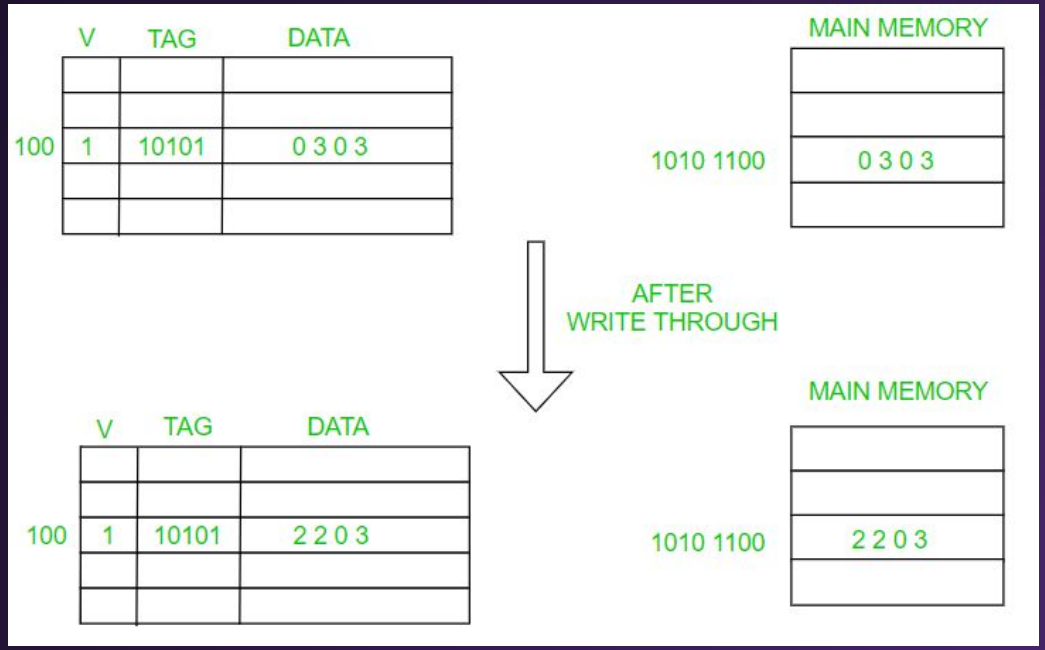
Write-through and write-back

Cuando una CPU escribe datos en la memoria caché, el valor en la memoria caché puede ser inconsistente con el valor en la memoria principal.

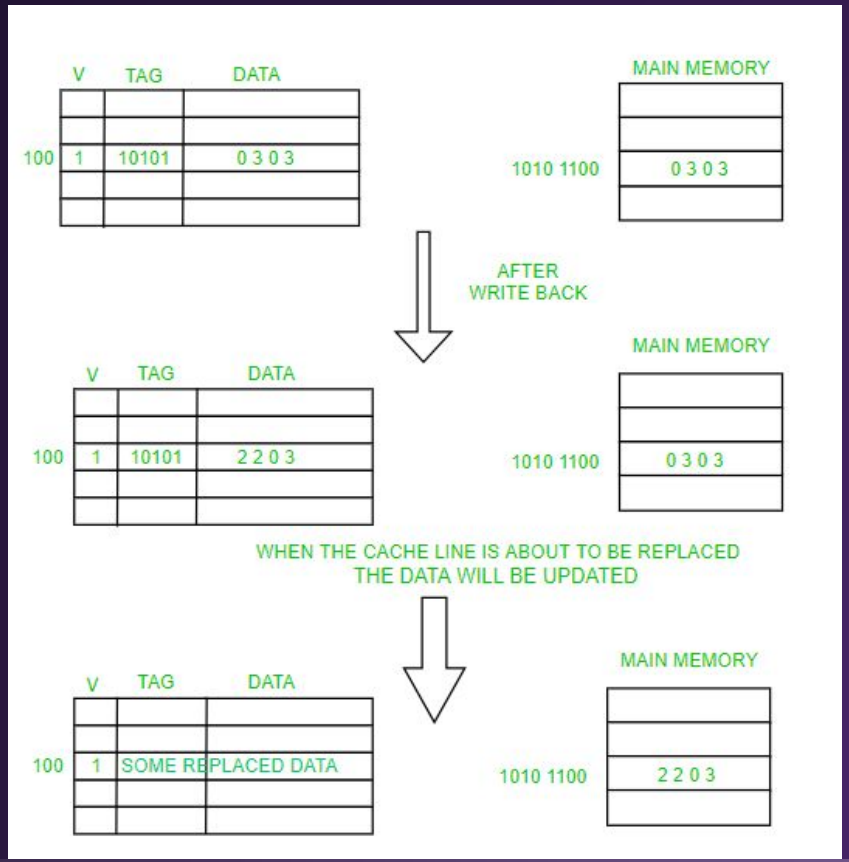
Write-through, la línea se escribe en la memoria principal cuando se escribe en el caché.

Write-back, los datos no se escriben inmediatamente sino se marcan como “dirty”, y cuando la línea de caché se reemplaza por una nueva línea de caché de la memoria, la línea “dirty” se escribe en la memoria.

Write-through



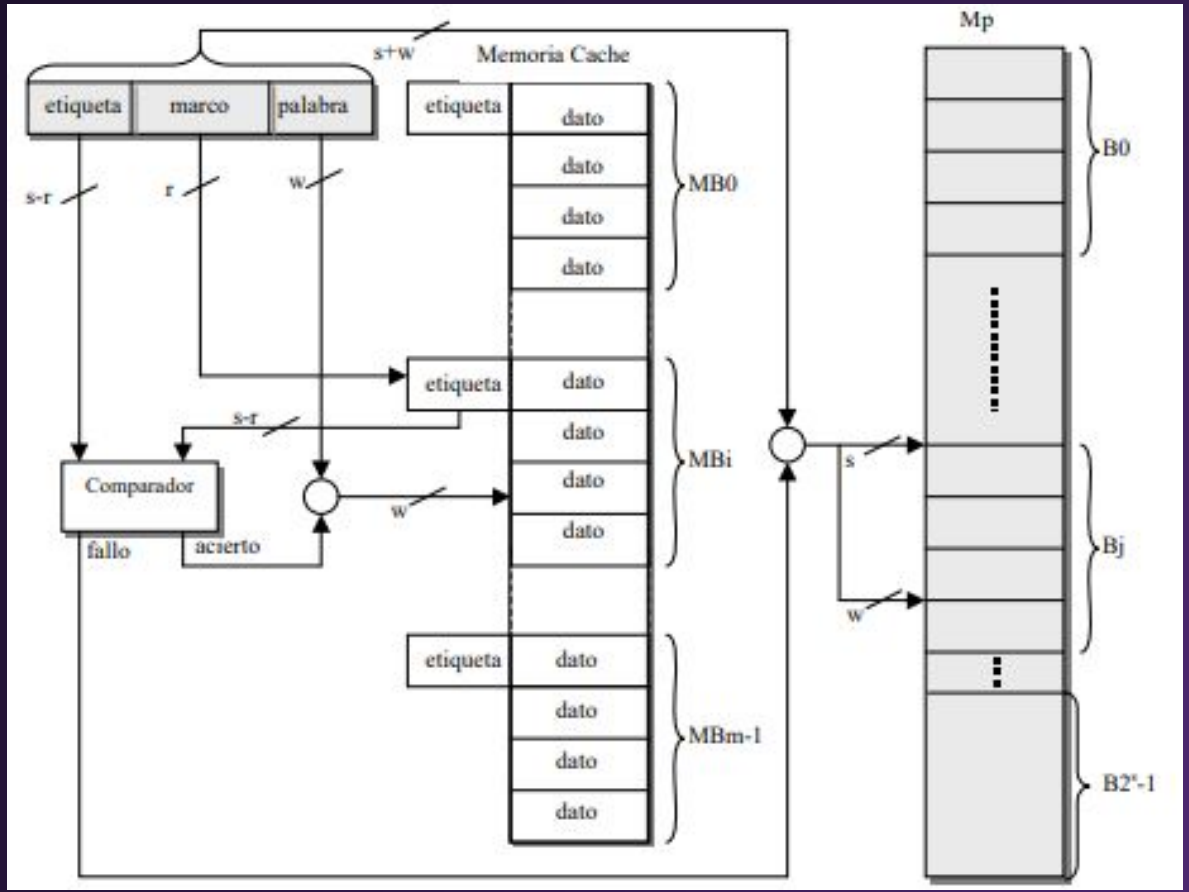
Write-Back


V
TAG
DATA
100
1
10101
0 3 0 3
MAIN MEMORY
1010 1100
0 3 0 3
AFTER
WRITE BACK
V
TAG
DATA
100
1
10101
2 2 0 3
MAIN MEMORY
1010 1100
0 3 0 3
WHEN THE CACHE LINE IS ABOUT TO BE REPLACED
THE DATA WILL BE UPDATED
V
TAG
DATA
100
1
SOME REPLACED DATA
MAIN MEMORY
1010 1100
2 2 0 3

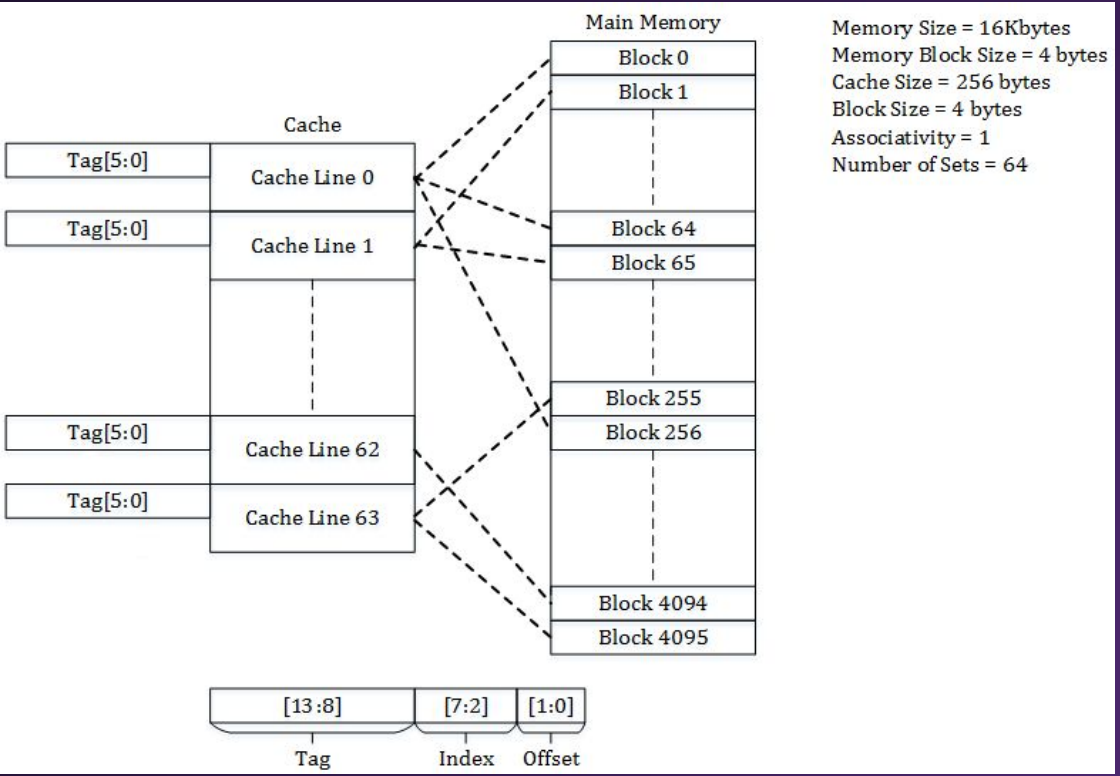
Cache Mappings

- Full associative: se puede colocar una nueva línea en cualquier lugar de la memoria caché.
- Direct mapped: cada línea de caché tiene una ubicación única en el caché a la que se asignará.
- n-way set associative: cada línea de caché se puede colocar en una de n ubicaciones diferentes en el caché.

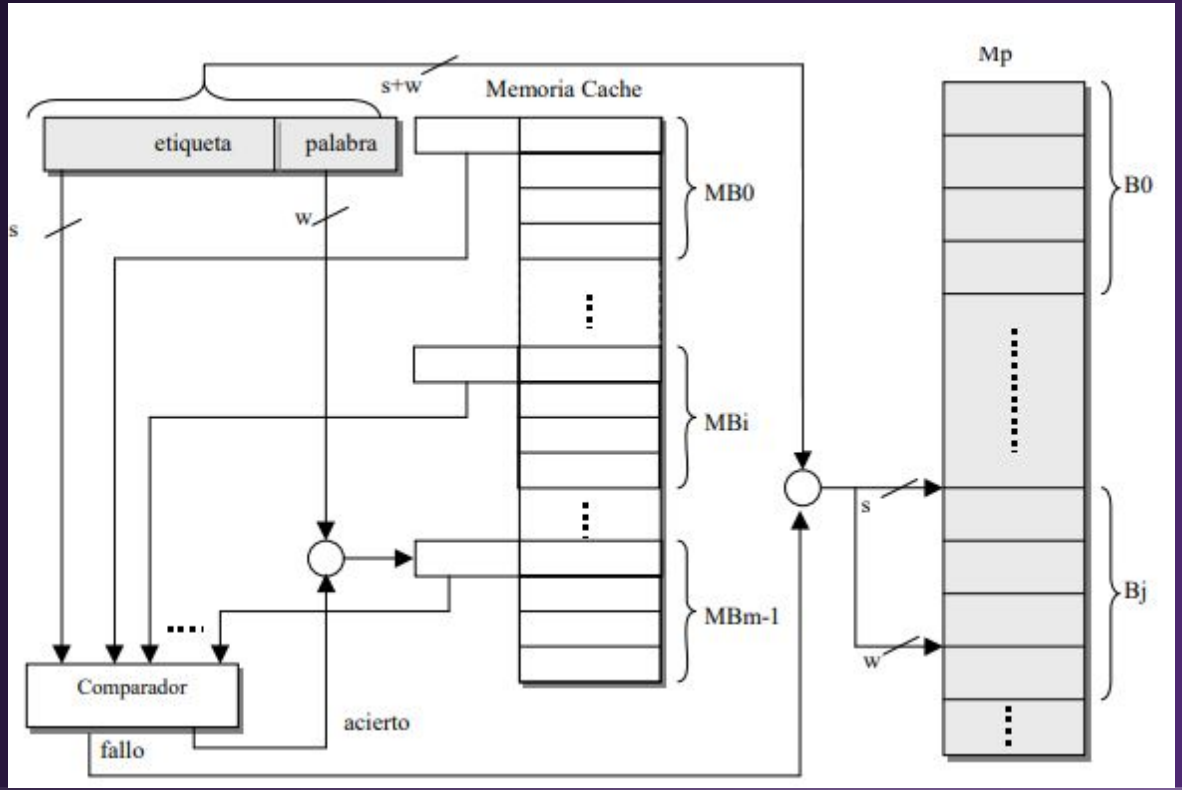
Direct Mapped cache



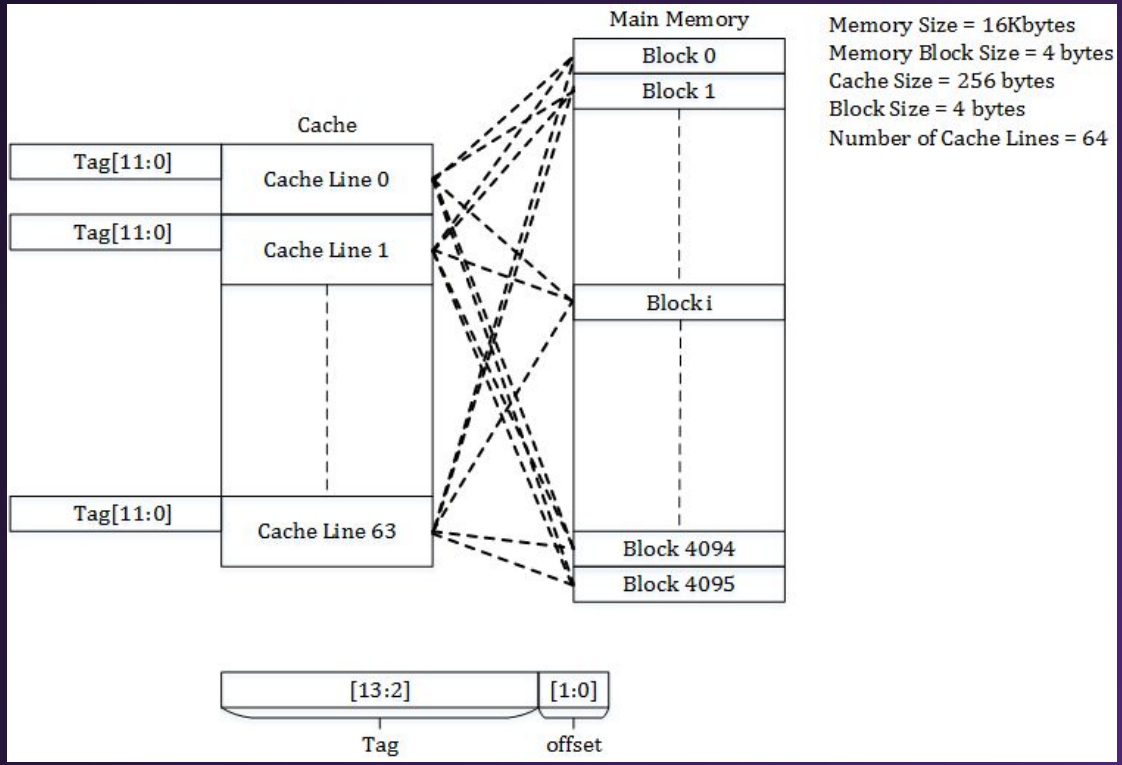
Direct Mapped cache



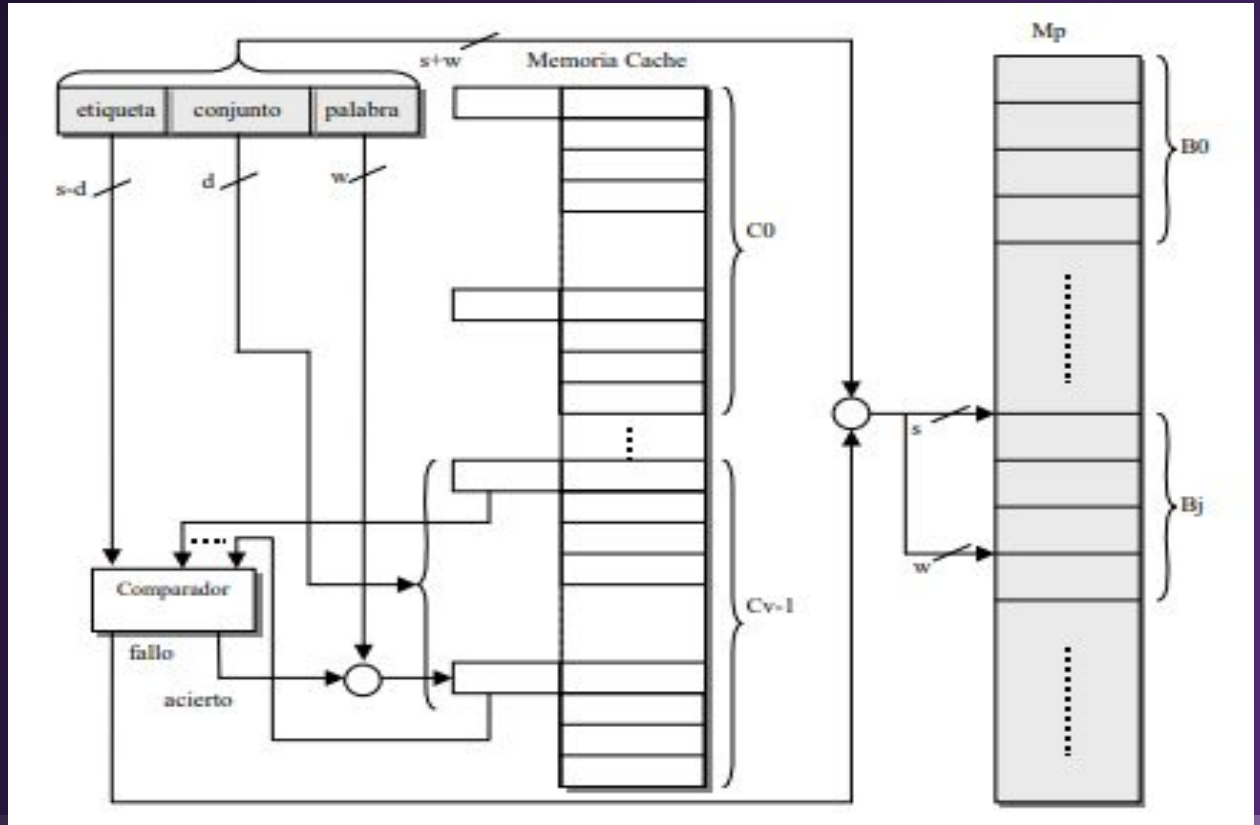
Full associative cache



Full associative cache



N-way set associative cache



N-way set associative cache

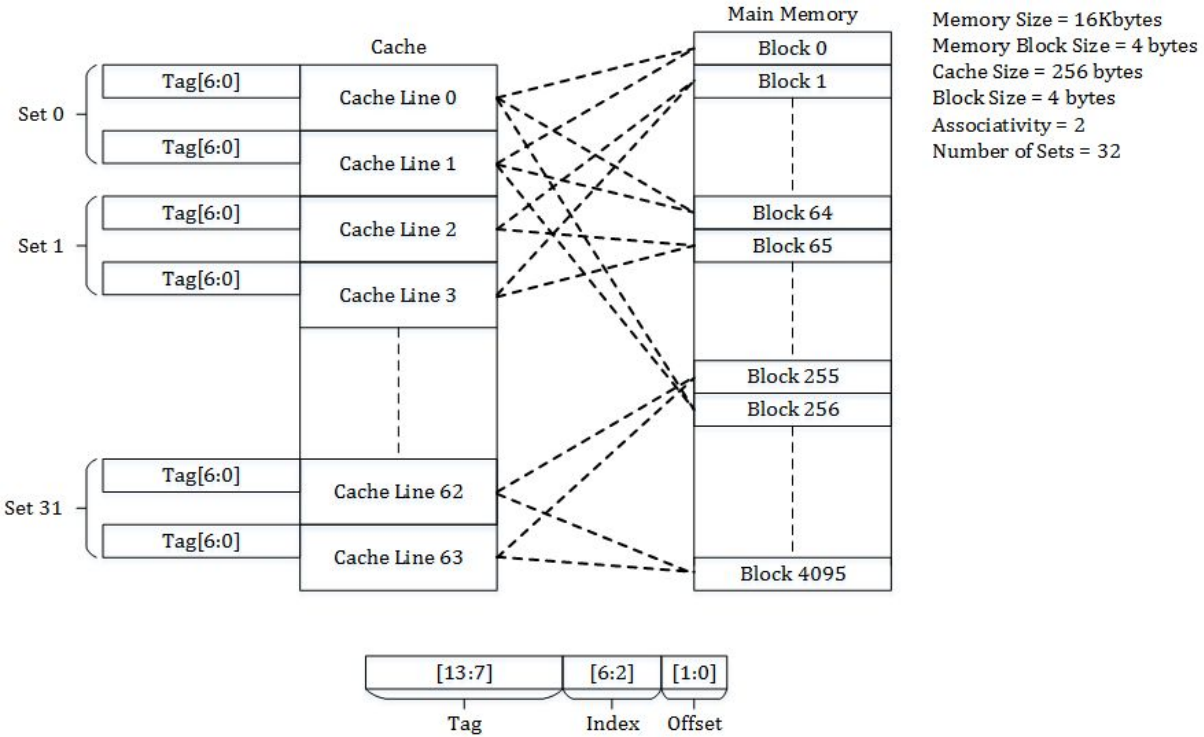


Table 2.1 Assignments of a 16-line Main Memory to a 4-line Cache

	Cache Location		
Memory Index	*Fully Assoc*	*Direct Mapped*	*2-way*
0	0, 1, 2, or 3	0	0 or 1
1	0, 1, 2, or 3	1	2 or 3
2	0, 1, 2, or 3	2	0 or 1
3	0, 1, 2, or 3	3	2 or 3
4	0, 1, 2, or 3	0	0 or 1
5	0, 1, 2, or 3	1	2 or 3
6	0, 1, 2, or 3	2	0 or 1
7	0, 1, 2, or 3	3	2 or 3
8	0, 1, 2, or 3	0	0 or 1
9	0, 1, 2, or 3	1	2 or 3
10	0, 1, 2, or 3	2	0 or 1
11	0, 1, 2, or 3	3	2 or 3
12	0, 1, 2, or 3	0	0 or 1
13	0, 1, 2, or 3	1	2 or 3
14	0, 1, 2, or 3	2	0 or 1
15	0, 1, 2, or 3	3	2 or 3

Caches and programs

- La CPU está controlada por el hardware del sistema, y los programadores, no determinan directamente qué datos y qué instrucciones están en el caché.
- Principle of spatial and temporal locality, nos permite tener cierto control indirecto sobre el almacenamiento en caché.
- C almacena arrays bidimensionales en orden de “row-major”.

```
double A[MAX][MAX], x[MAX], y[MAX];
. . .
/* Initialize A and x, assign y = 0 */
. . .
/* First pair of loops */
for (i = 0; i < MAX; i++)
   for (j = 0; j < MAX; j++)
      y[i] += A[i][j]*x[j];
. . .
/* Assign y = 0 */
. . .
/* Second pair of loops */
for (j = 0; j < MAX; j++)
   for (i = 0; i < MAX; i++)
      y[i] += A[i][j]*x[j];
```

Analisis

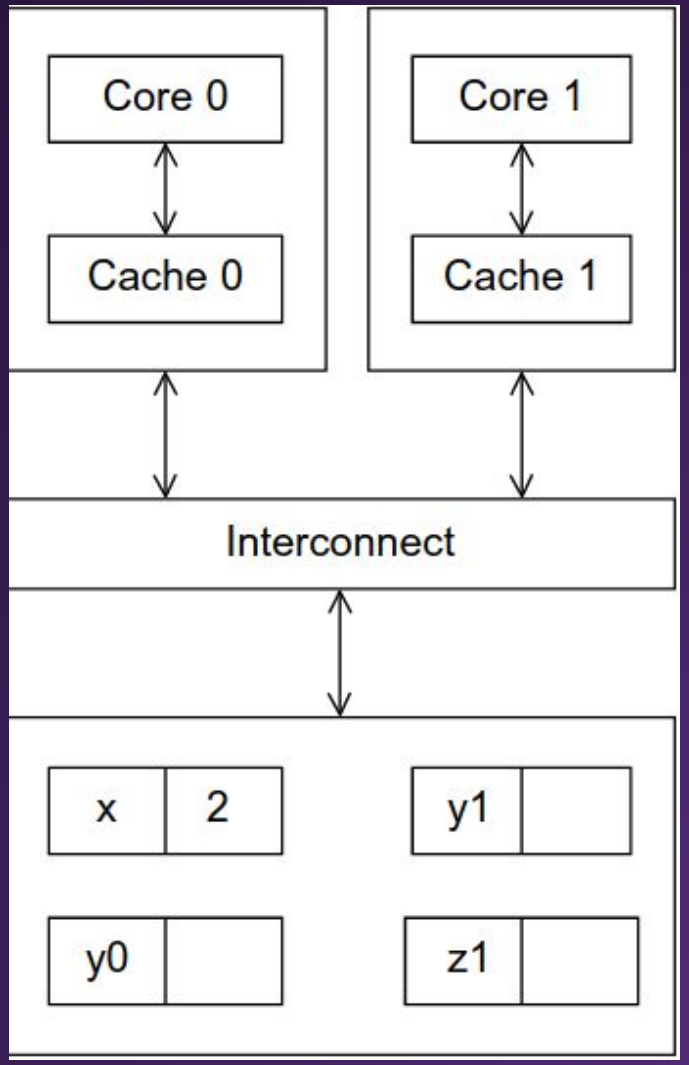
Supongamos que MAX es 4 y los elementos de A se almacenan en la memoria de la siguiente manera.

Cache Line	Elements of A			
0	A[0][0]	A[0][1]	A[0][2]	A[0][3]
1	A[1][0]	A[1][1]	A[1][2]	A[1][3]
2	A[2][0]	A[2][1]	A[2][2]	A[2][3]
3	A[3][0]	A[3][1]	A[3][2]	A[3][3]

Cache Coherence

Time	Core 0	Core 1
0	`y0 = x;`	`y1 = 3*x;`
1	`x = 7;`	Statement(s) not involving x
2	Statement(s) not involving x	`z1 = 4*x;`

Un sistema de memoria compartida con dos núcleos y dos cachés.



Snooping Cache Coherence

- Los núcleos comparten un bus.
- Cualquier señal transmitida en el bus puede ser "vista" por todos los núcleos conectados al bus.
- Cuando el núcleo 0 actualiza la copia de x almacenada en su caché, también transmite esta información a través del bus. Si el núcleo 1 está "snooping" el bus, verá que x se ha actualizado y puede marcar su copia de x como no válida

Directory Based Cache Coherence

- Utiliza una estructura de datos llamada directorio que almacena el estado de cada línea de caché.
- Cuando se actualiza una variable, se consulta el directorio y se invalidan los controladores de caché de los núcleos que tienen la línea de caché de esa variable en sus cachés.

GRACIAS!