Arquitecturas y Organización de Computadoras I

Memoria Cache

Noviembre 2020

Programa analitico de la asignatura

UNIDAD 3: Memoria

Características de las diferentes tecnologías de memoria. Comportamiento de los programas: principio de localidad. Jerarquía de memoria. Memoria Cache. Memoria virtual.

Memoria Caché

La memoria caché es uno de las nociones mas sencillas en computación:

Primero, se ubican los datos mas frecuentemente accedidos en una memoria rápida, la caché.

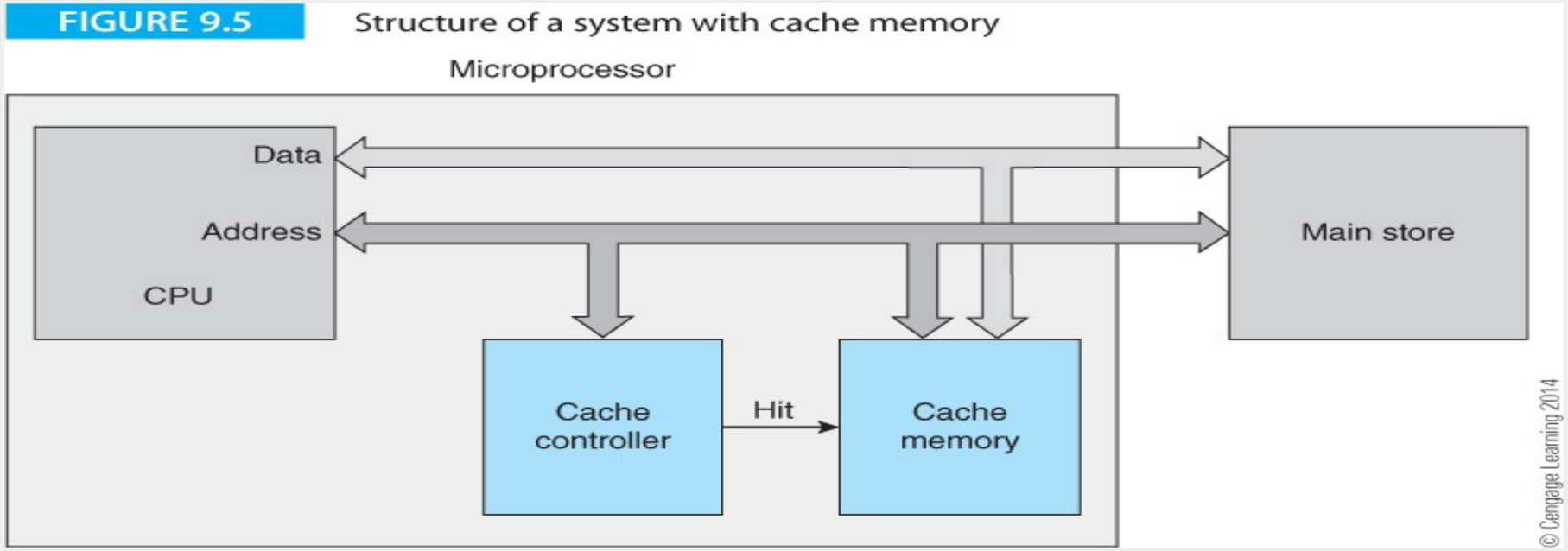
Cuando la computadora accede a estos datos lo hace mucho más rapidamente
que si los datos estuvieran en memoria principal.

La memoria caché está dentro de los microprocesadores o en la placa base (motherboard).
Funciona automáticamente a través de la electrónica del hardware.

De cualquier manera, los usuarios deben estar al tanto de su organización y funcionamiento,
para optimizar su uso.

Memoria Caché

La memoria caché puede comprenderse más facilmente con la analogía de la agenda telefónica en papel.

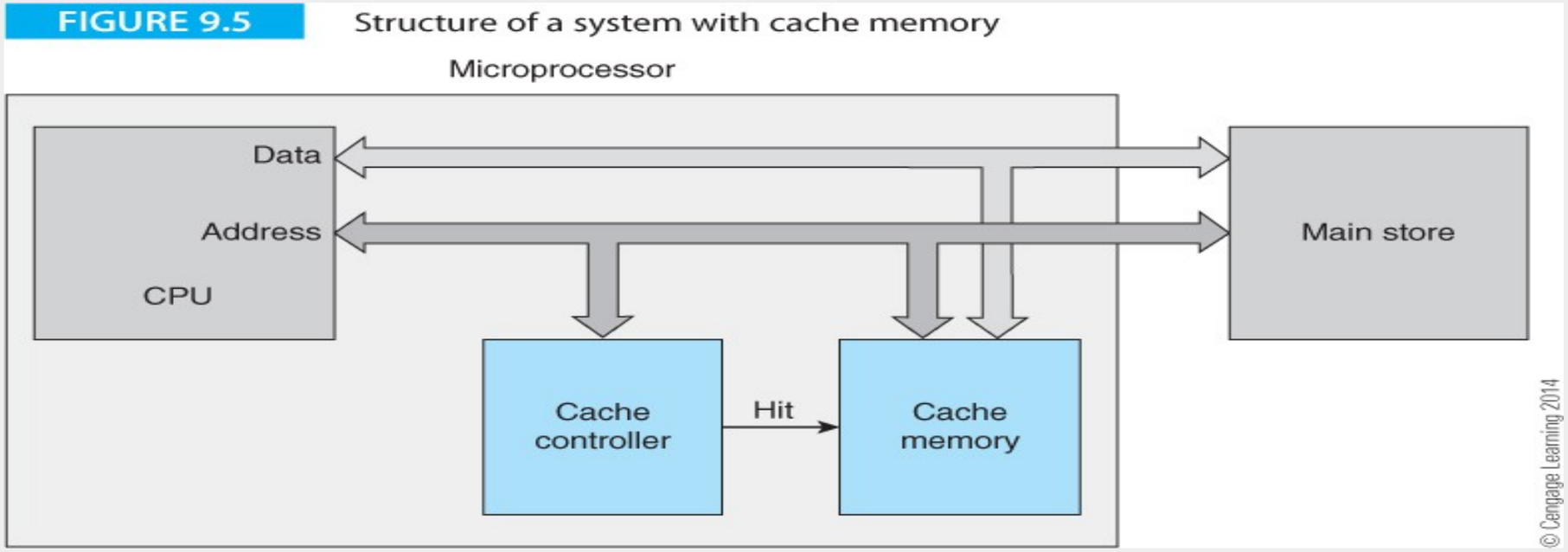


Memoria Caché

Desafortunadamente, a diferencia del cuaderno personal, la computadora no puede saber, a priori, qué datos serán posiblemente accedidos.

Las caches de computadora funcionan según un principio de aprendizaje. Por experiencia aprenden qué datos se utilizan con más frecuencia y luego lo transfieren a la memoria caché.

La observación del comportamiento de los programas muestra que las referencias a memoria realizadas en un intervalo corto de tiempo tienden a utilizar una pequeña fracción del total de la memoria. Este comportamientno de acceso ha sido llamado principio de localidad de las referencias[DENN68], o simplemente principio de localidad.

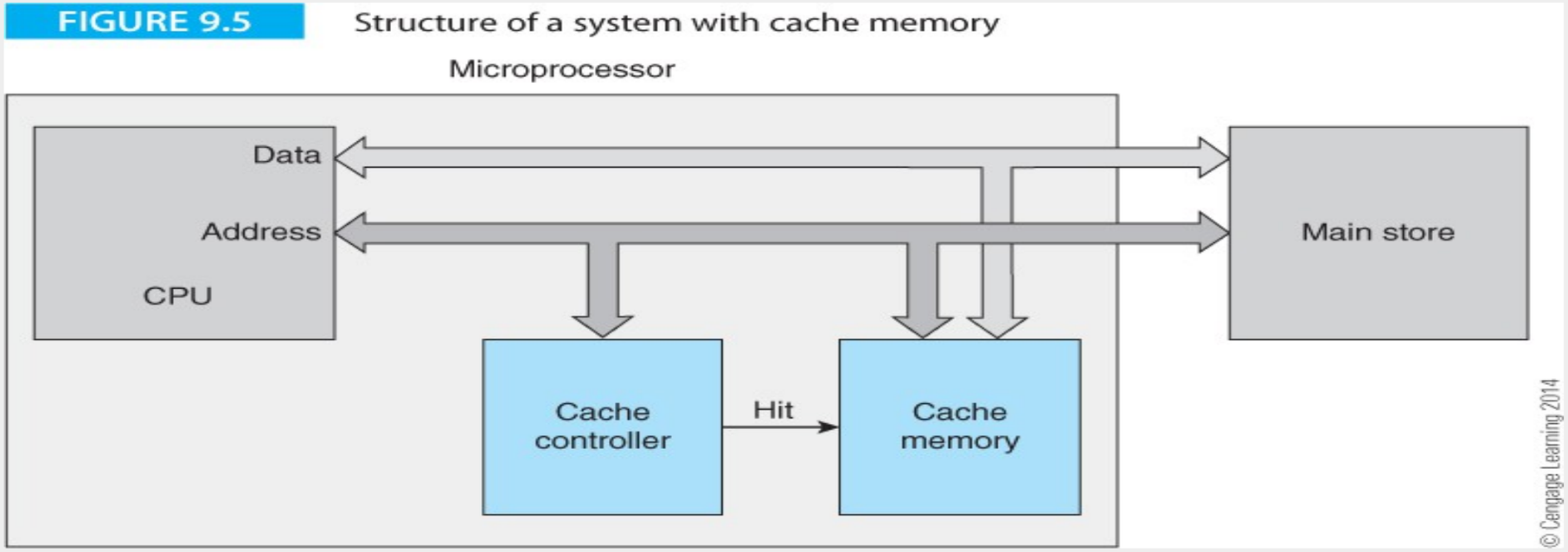


Memoria Caché

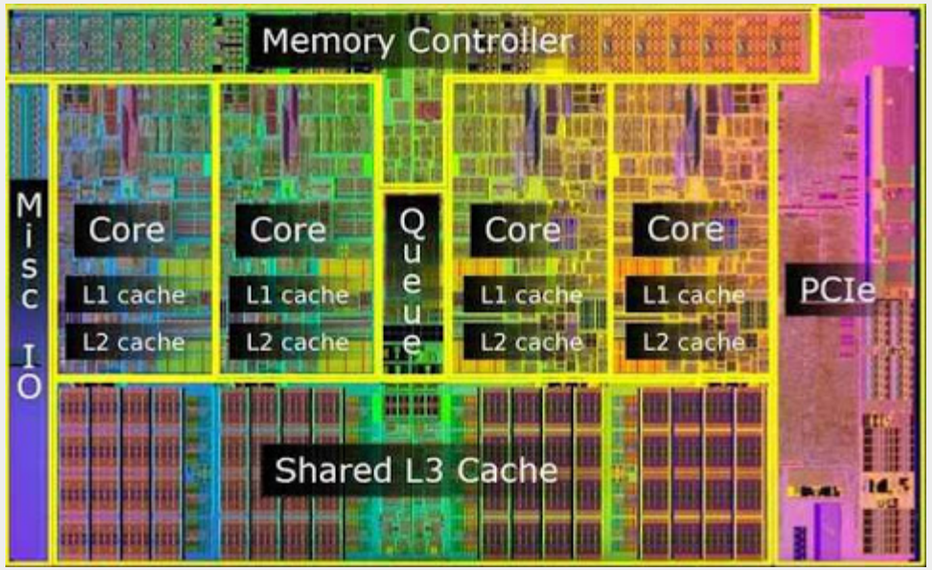
Existen al menos dos tipos básicos de localidad: localidad espacial y temporal.

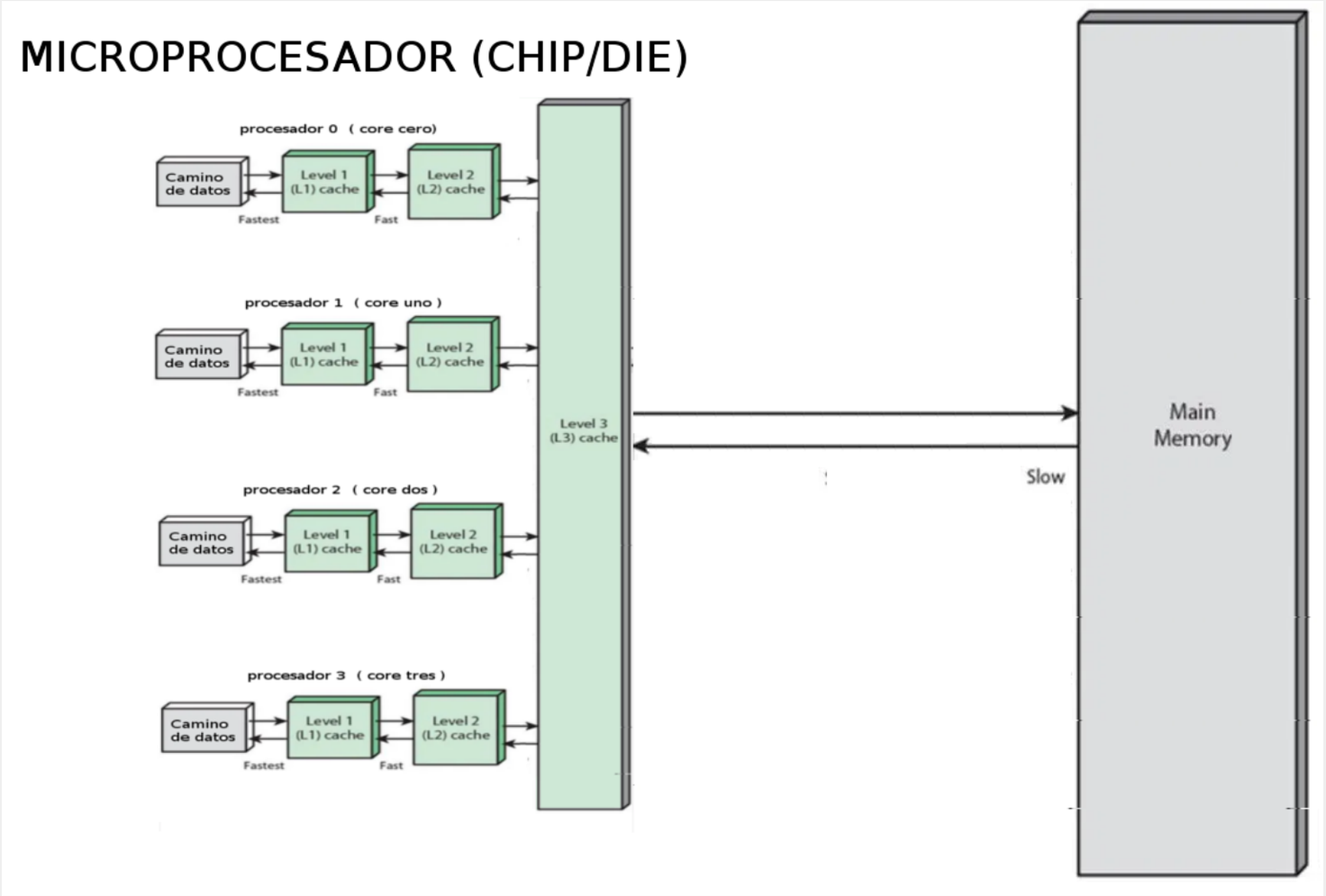
La localidad temporal se refiere a la reutilización de datos especificos, dentro de un tiempo relativamente corto. Cuando un programa accede a la ubicación en memoria de un dato o instrucción es bastante probable que vuelva a acceder a la misma ubicación pronto.

La localidad espacial, en cambio, se refiere a la utilización de datos en ubicaciones de memoria cercanas a los elementos accedidos recientemente. Si un programa accede a un dato o instruccion en memoria es altamente probable que tambien referencie a datos o instrucciones alojados en direcciones proximas.



Memoria Caché: Organización de 3 niveles





La memoria caché puede mejorar el rendimiento de una computadora dramaticamente, a cambio de un costo adicional relativamente bajo.

Memoria Caché: tiempo de acceso medio

El parámetro principal de un sistema de caché es su tasa de aciertos (hit ratio: proporción de aciertos a todos los accesos).

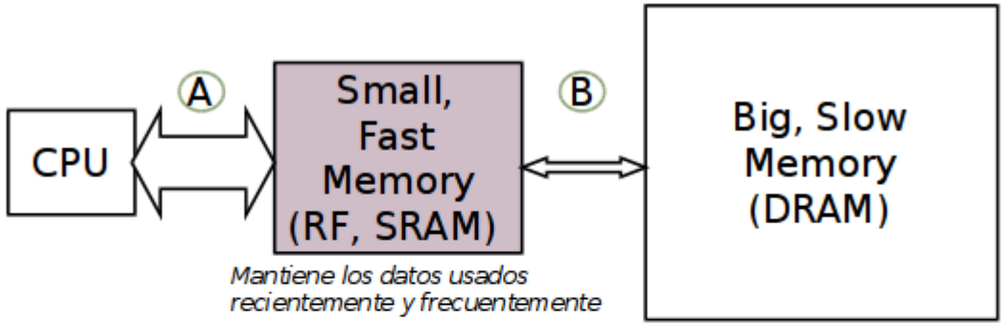
Es posible formalizar este cálculo:

- c es el tiempo de acceso a caché
- m es el tiempo de acceso a memoria principal
- h es la tasa de aciertos (fracción de las referencias que están disponibles en caché).

tiempo medio de acceso $= c + (1 - h)\, m$

Si $h \to 1$ entonces el tiempo de acceso se aproxima a c.

Si $h \to 0$ entonces el tiempo de acceso se aproxima a $c + m$.



Terminología de acceso:

Acierto/hit (data fast memory) — low latency access
Fallo/miss (data fast memory) — long latency access *(DRAM)*

Memoria Caché: Acceso

Cuando el procesador requiere un dato desde la memoria busca en la caché:

Si lo encuentra
a.k.a. HIT

La caché le entrega la copia del dato al procesador Inmediatamente

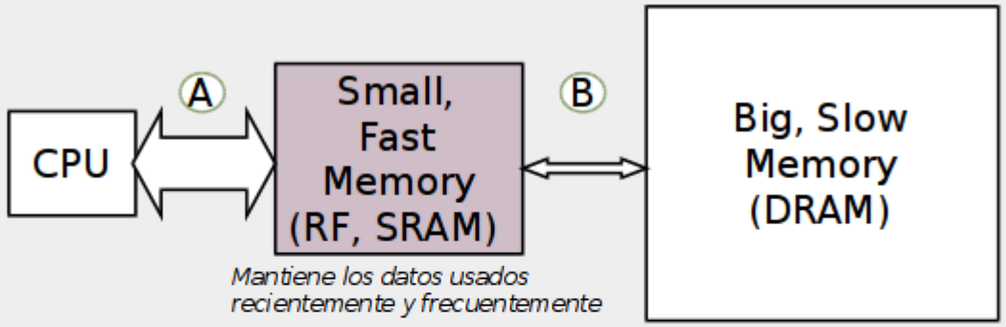
Si no se encuentra en la caché
a.k.a. MISS

Leer un bloque desde la Memoria Princial

Esperar ...

Entregar el dato al procesador y actualizar la caché

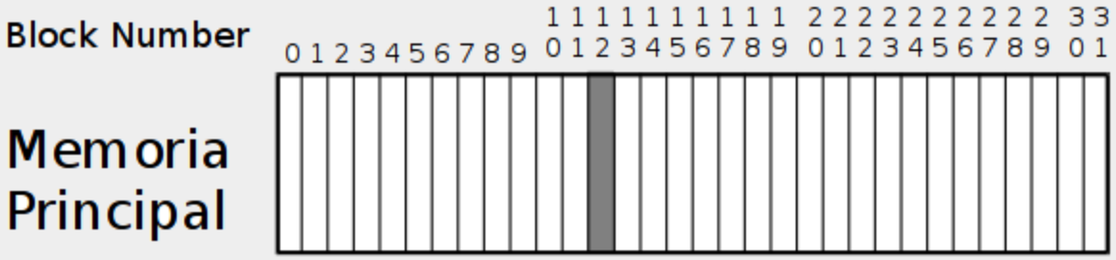
Pregunta: qué línea de caché se debe reemplazar?

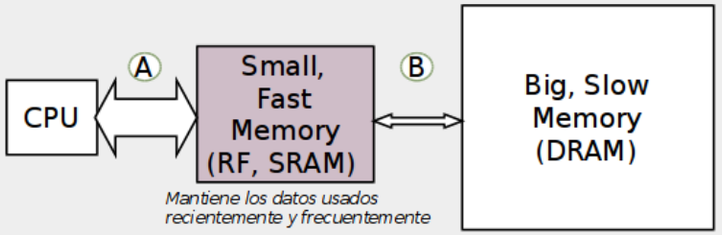


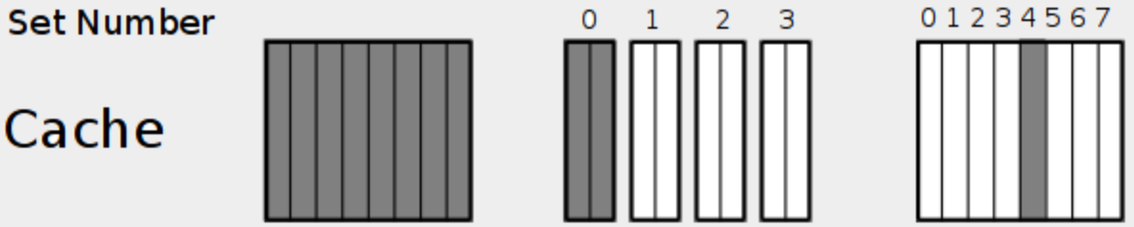
Terminología de acceso:

Acierto/hit (data fast memory) low latency access
Fallo/miss (data fast memory) long latency access *(DRAM)*

Memoria Caché: Las 3 organizaciones clásicas







Terminología de acceso:

Acierto/hit (data fast memory) low latency access
Fallo/miss (data fast memory) long latency access *(DRAM)*

El bloque 12 puede ser ubicado	En cualquier lugar	En cualquier lugar del conjuto 0 (12 mod 4)	Unicamente en el bloque 4 (12 mod 8)

Memoria Caché: tiempo de acceso medio

El parámetro principal de un sistema de caché es su tasa de aciertos (hit ratio:,

- proporción de aciertos a todos los accesos.

Es posible formalizar este cálculo:

- c es el tiempo de acceso a caché
- m es el tiempo de acceso a memoria principal
- h es la tasa de aciertos (fracción de las referencias que están disponibles en caché).

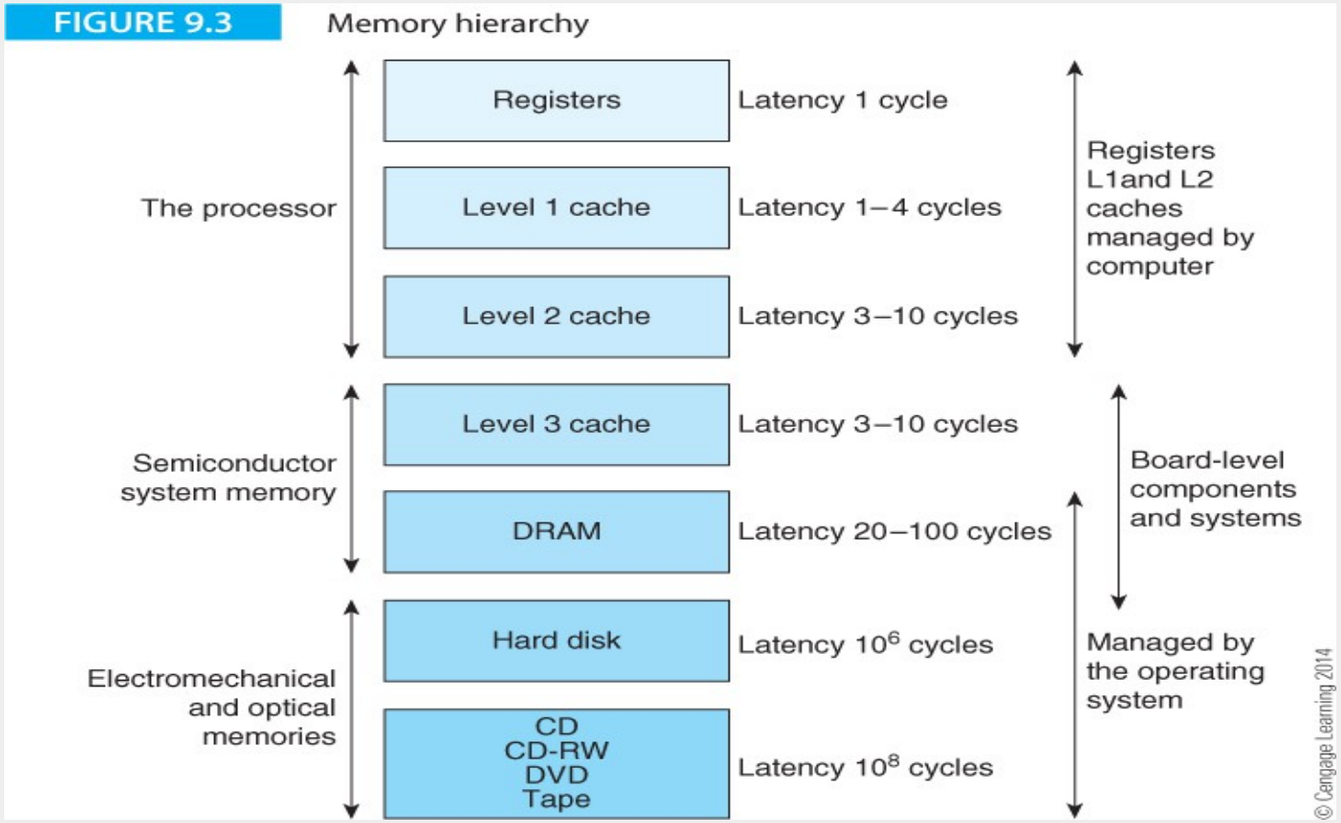
tiempo medio de acceso = c + (1 - h) m

Si h $\rightarrow$ 1 entonces el tiempo de acceso se aproxima a c.

Si h $\rightarrow$ 0 entonces el tiempo de acceso se aproxima a c + m

Memoria Caché: tiempo de acceso medio

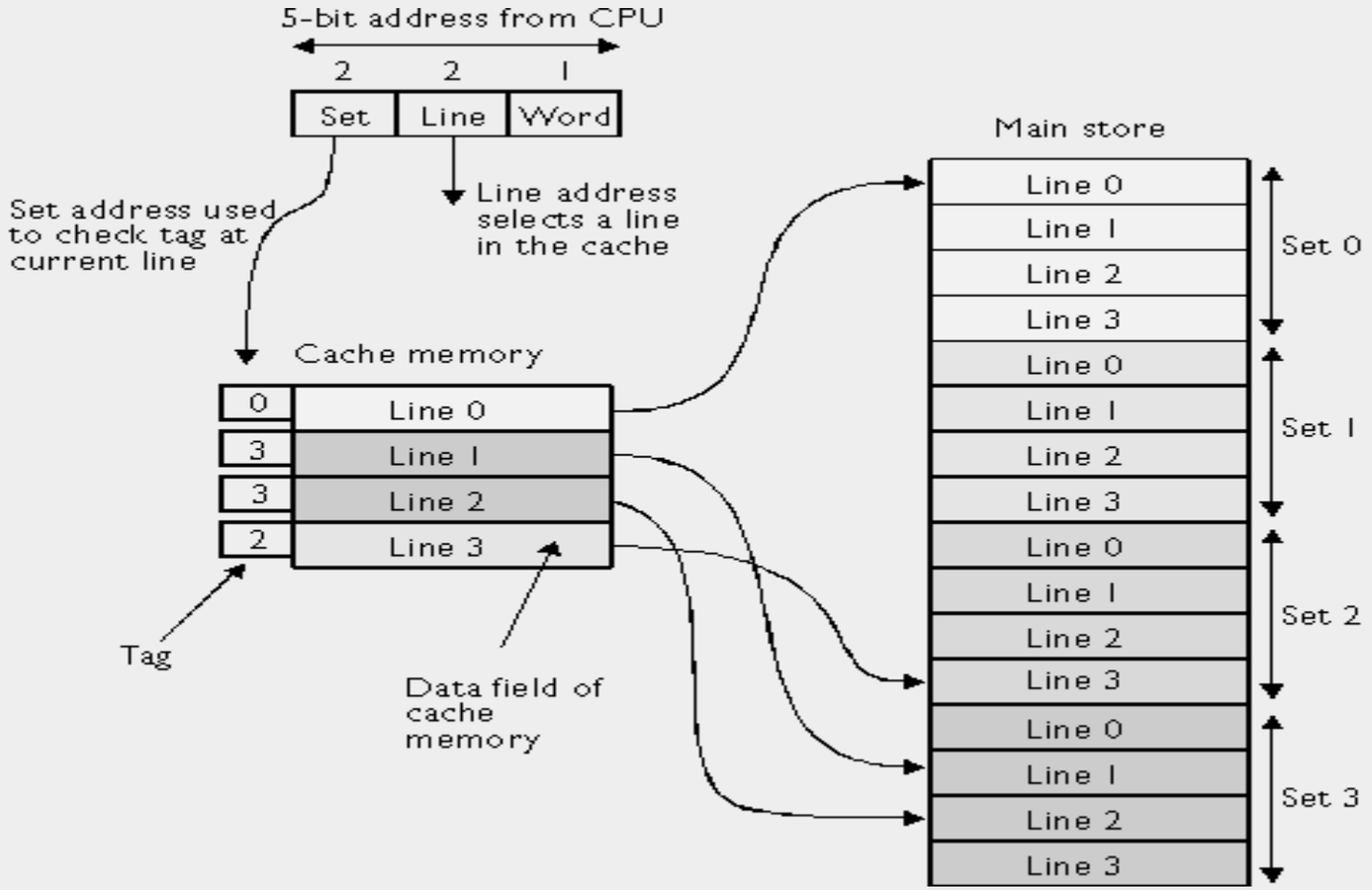
Con estas variables y definiciones el tiempo medio de acceso = c + (1 - h) m



La memoria caché puede mejorar el rendimiento de una computadora dramaticamente, a cambio de un costo adicional relativamente bajo.

Memoria Caché de mapeo directo

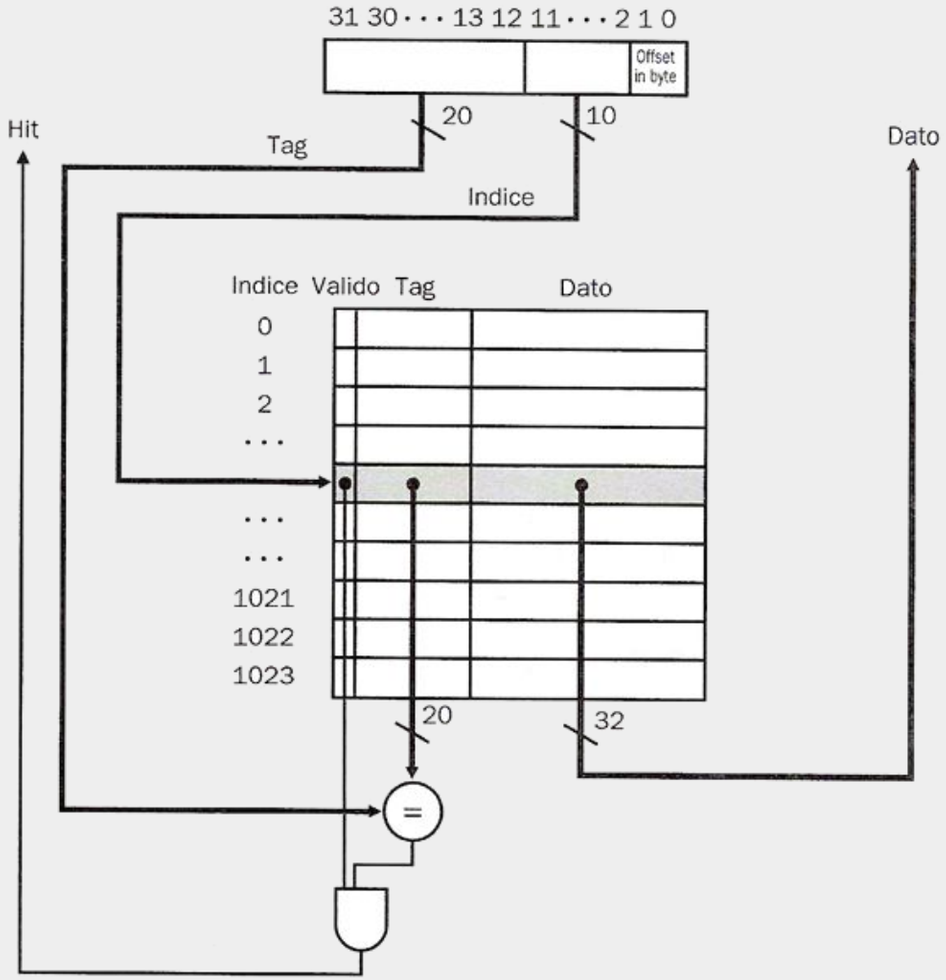
La forma más sencilla de organizar una memoria caché es utilizar un mapeo directo, que se basa en un algoritmo simple: asignar el bloque de datos i de la memoria principal al bloque de datos i en la memoria caché.



Memoria Caché de mapeo directo

Si el comparador resuelve que Tag es igual a la parte mas alta de la dirección y el bit de validez está activado el bloque está en la caché, entonces se produjo un acierto (hit) de caché.

De otro modo sucede un fallo (miss) de caché,
y la línea completa debe obtenerse desde memoria principal.



Ejercicio:

```
li t0, 0xF140
lw t1, 0(t0)
addi t0, t0, 1
lb t1, 0(t0)
addi t0, t0, 3
lw t2, 0(t0)
addi t0, t0, 0x400
lw t3, 0(t0)
addi t0, t0, -1028
lw t4, 0(t0)
```

Para los accesos a datos en memoria de las instrucciones lw/lb:

¿Que direcciones efectivas genera la CPU?
¿A qué líneas de caché accede?
¿Hay acierto o fallo en cada acceso?

Memoria Caché de mapeo directo

Si el comparador resuelve que Tag es igual a la parte mas alta de la dirección y el bit de validez está activado el bloque está en la caché, entonces se produjo un acierto (hit) de caché.

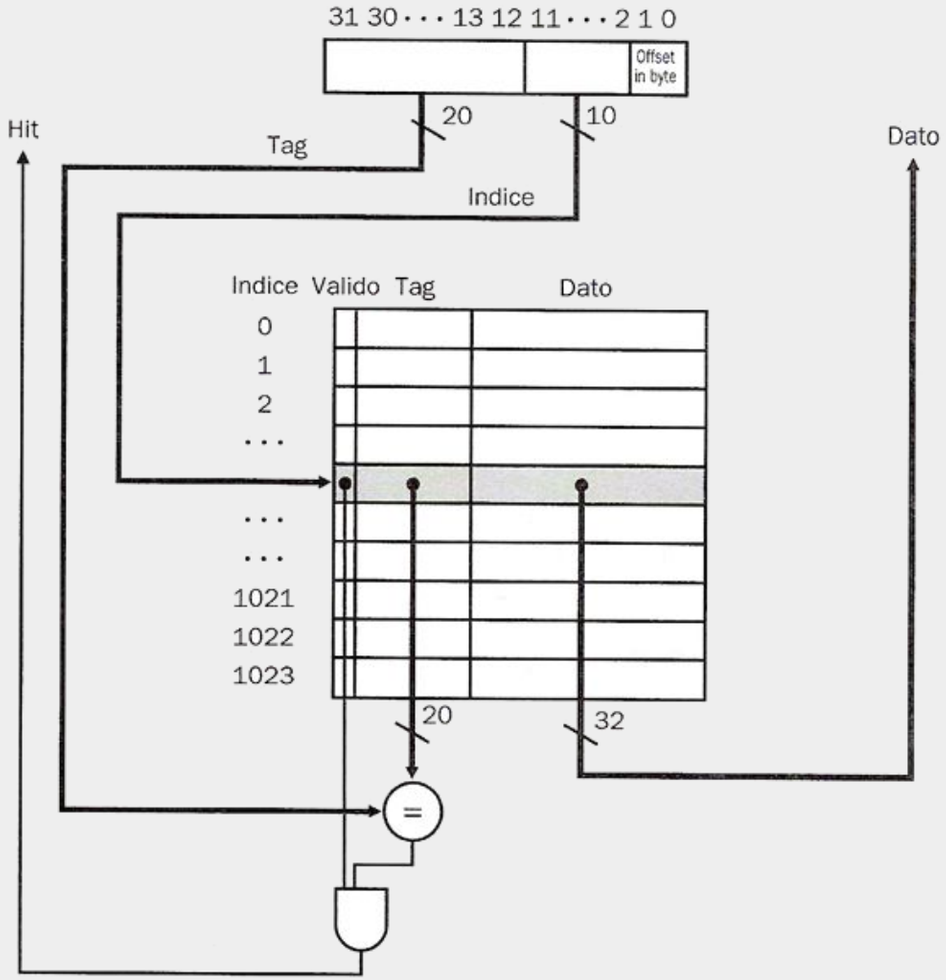
De otro modo sucede un fallo (miss) de caché,
y la línea completa debe obtenerse desde memoria principal.

Ejercicio:

```
                                  # dirección efectiva
li t0, 0xF140
lw t1, 0(t0)                      # 00000000000000001111000101000000
addi t0, t0, 1
lb t1, 0(t0)
addi t0, t0, 3
lw t2, 0(t0)
addi t0, t0, 0x400
lw t3, 0(t0)
addi t0, t0, -1028
lw t4, 0(t0)
```

Para los accesos a datos en memoria de las instrucciones lw/lb:

¿Que direcciones efectivas genera la CPU?
¿A qué líneas de caché accede?
¿Hay acierto o fallo en cada acceso?



Memoria Caché de mapeo directo

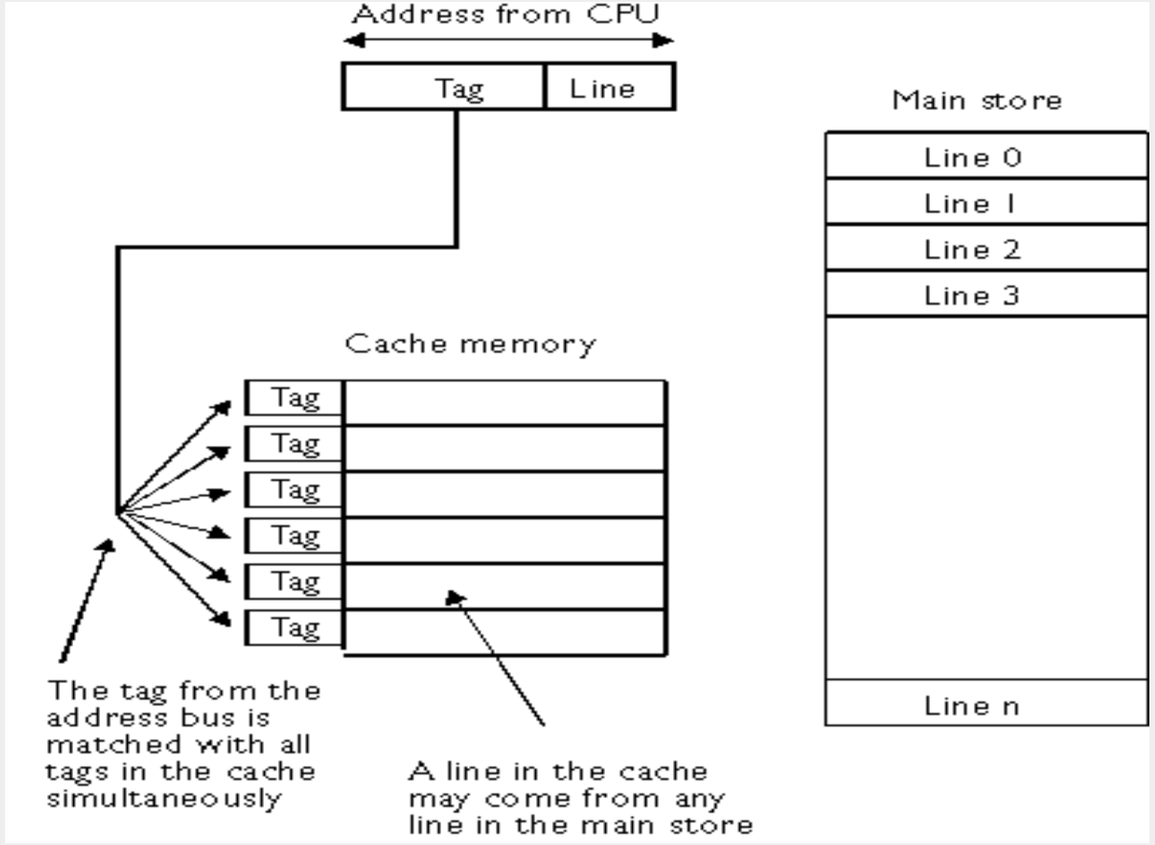
La ventaja de la memoria caché de mapeo directo es casi evidente. Debido a su sencillez es veloz y no presenta complejidad extra (costo de construcción bajo). Esta caché es un dispositivo ampliamente disponible que, aparte de su velocidad, no son más complejos que cualquier otro circuito integrado.

La desventaja de esta caché es casi un corolario de su ventaja. Un caché con n líneas tiene una restricción: en cualquier instante puede contener sólo una línea numerada x. Por lo tanto, no puede mantener una línea x del conjunto p y una línea x del conjunto q. Esta restricción existe porque hay un único bloque de datos en la caché para cada una de las lineas posibles.

Por ejemplo, supongamos que una memoria caché de mapeo directo está casi vacía, y que la mayoría de sus líneas aún no se han cargado con datos. Sin embargo, las únicas líneas con datos válidos deben reemplazarse con frecuencia debido a que los accesos a memoria son direcciones con diferentes números de conjuntos pero misma línea. Este caso tendrá un rendimiento pobre (fallos de caché frecuentes) aunque haya muchas líneas aún vacías.

Memoria Caché asociativa

Una excelente forma de organizar una memoria caché se denomina memoria caché asociativa. Este tipo de caché no tiene restricciones en cuanto a qué datos puede contener cada entrada. En otras palabras, cada entrada en esta caché puede almacenar cualquier bloque (línea) de la memoria principal.

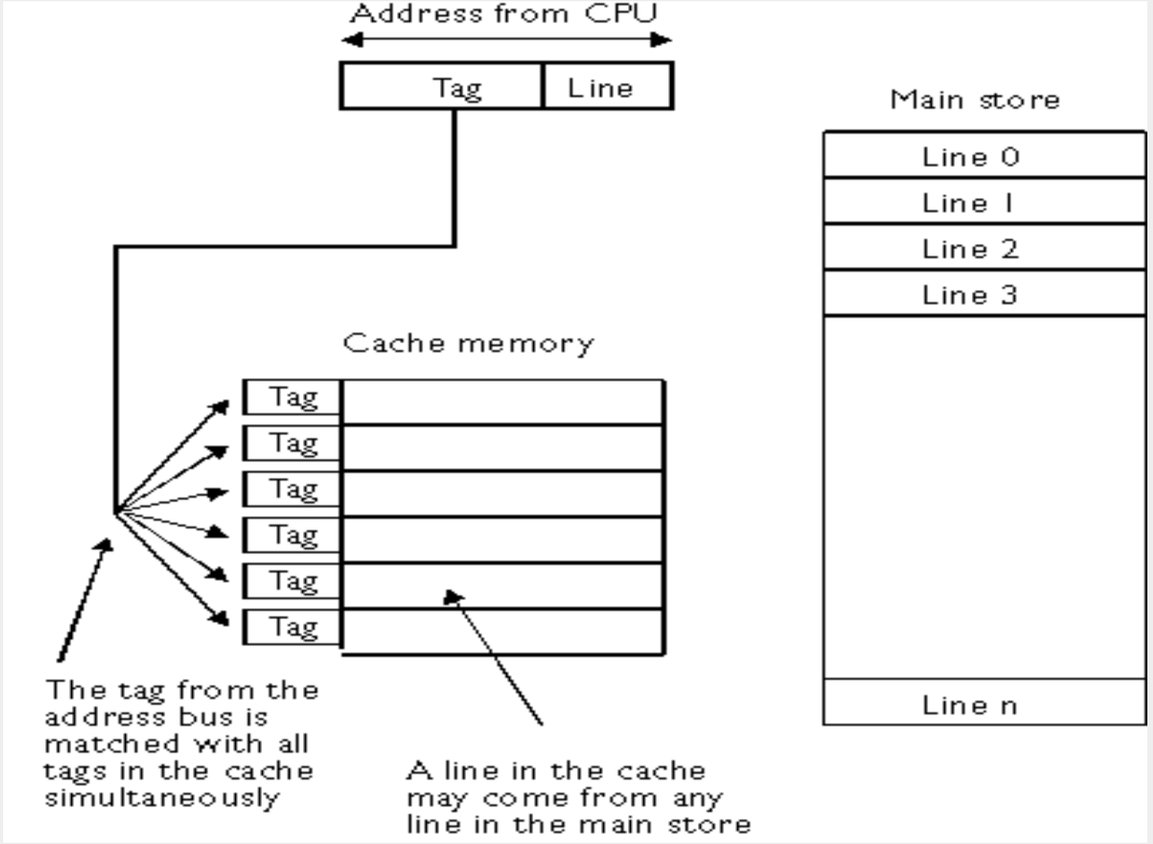


Memoria Caché asociativa

Una memoria asociativa es accedida preguntando, "¿tiene este elemento almacenado en alguna parte?"

También, necesita una política de reemplazo.

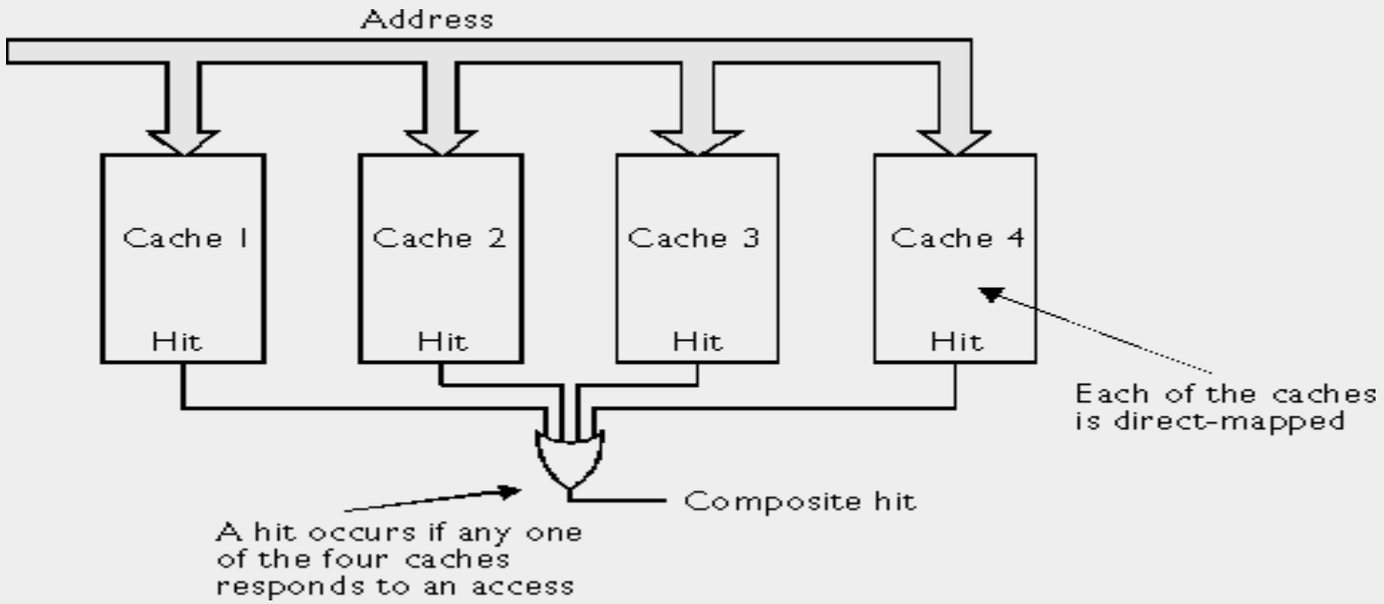
Lamentablemente, las caché completamente asociativas son muy costosas de fabricar.



Memoria Caché asociativa por conjuntos de n vías

La mayoría de las computadoras emplean un tipo de caché organizada de maneria mixta, utilizando algunos mecanismos de las cachés de mapeo directo y algunos de las cachés totalmente asociativas. Este sistema combinado se llama caché asociativa por conjuntos de n vías.

Ejemplo de caché asociativa por conjuntos de 4 vías:



Memoria Caché asociativa por conjuntos de n vías

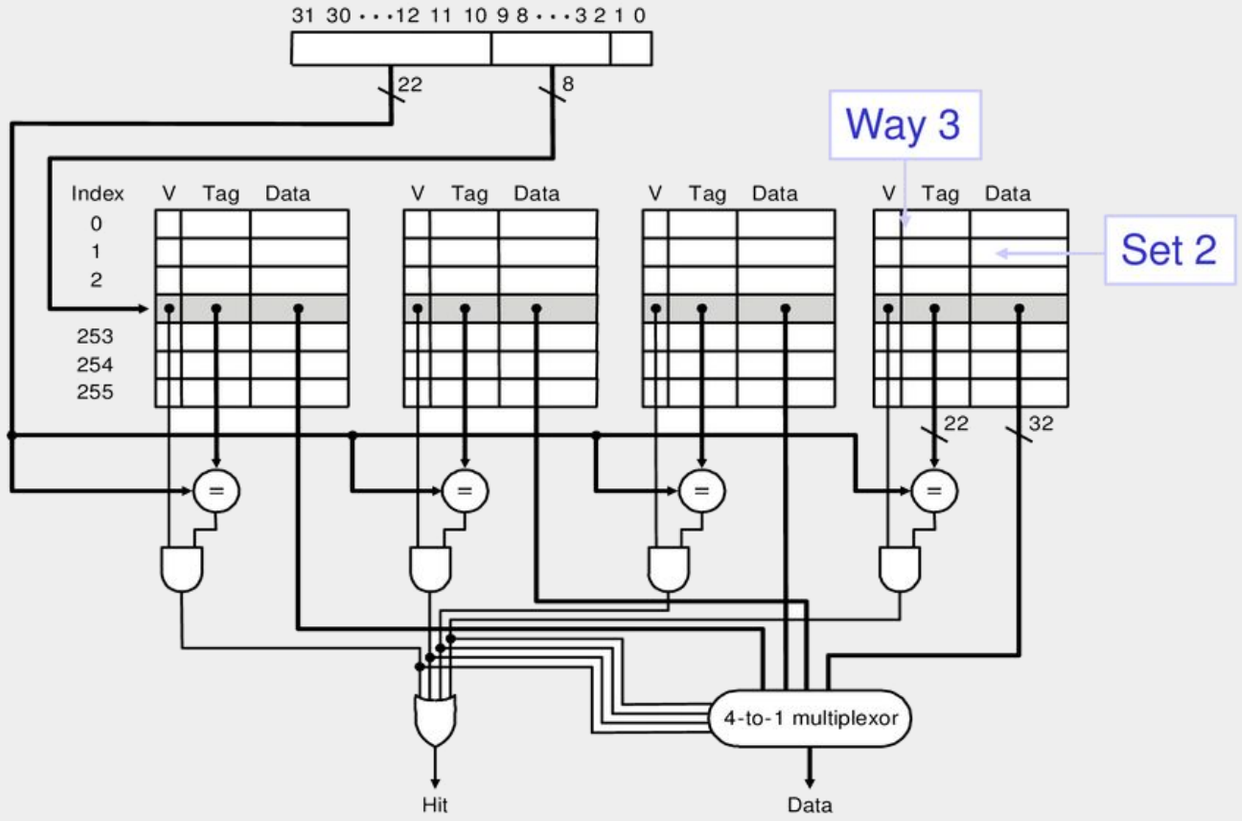
Ejemplo de caché asociativa por conjuntos de 4 vías:

```
Ejercicio:


li t0, 0xF140
lw t1, 0(t0)
addi t0, t0, 1
lb t1, 0(t0)
addi t0, t0, 3
lw t2, 0(t0)
addi t0, t0, 0x400
lw t3, 0(t0)
addi t0, t0, -1028
lw t4, 0(t0)


Para los accesos a datos en memoria de las instrucciones
lw/lb:

¿Que direcciones efectivas genera la CPU?
¿A qué líneas de caché accede?
¿Hay acierto o fallo en cada acceso?
```



Memoria Caché: estrategias en ciclos de escritura

Una tecnica común es modificar el dato en la entrada de la caché, sin modificar la memoria principal.

Este método es llamado estrategia de escritura diferida (write-back), y reduce el número de escrituras a memoria principal en muchas escrituras consecutivas de la misma entrada.
Un bit extra para cada entrada está presente en la caché, y se utiliza para indicar si la línea en la caché fue modificada.

Otra estrategia mas sencilla es la escritura directa (write-through). Cuando se realiza un requisito de escritura el nuevo dato es escrito en ambas memorias, en la caché y en la memoria principal.

De esta manera, tanto la caché como la memoria principal tienen siempre copias validas de todos los datos. Aunque existe un rendimiento mas bajo que la estrategia anterior, muchas veces el procesador puede continuar ejecutando instrucciones mientras la escritura a memoria principal procede.

Memoria Caché: otras consideraciones

- Coherencia de caché
- Políticas de reemplazo
- Múltiples memorias caché en paralelo (arquitectura Harvard)
- Multiniveles de memoria caché

Bibliografia

Textos:

- **Apuntes de cátedra**