Tema 5. Memoria Caché

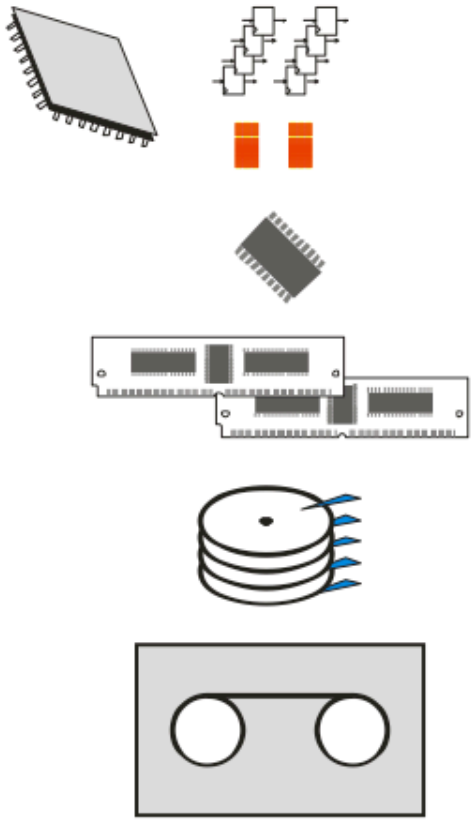
ESTRUCTURA DE COMPUTADORES

Grupo de Arquitectura de Computadores (GAC)

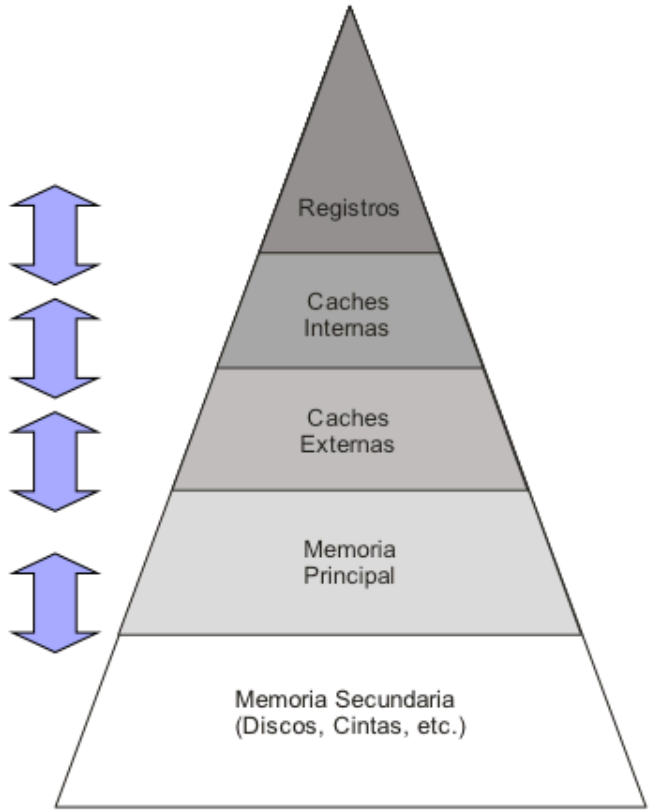
Índice

Índice

Introducción

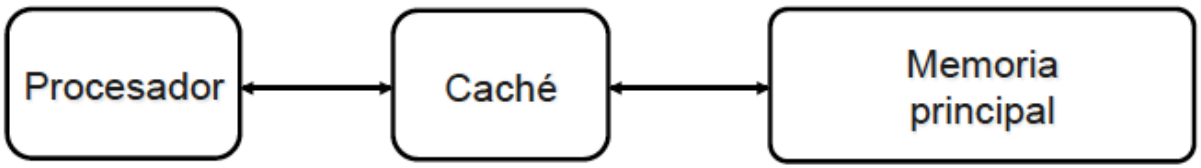


- Registros del procesador
  - Hasta 2KB
  - Hasta 57GB/seg
- Caches internas
  - 32KB - 1MB
  - 45 – 25GB/seg
- Caches externas
  - Hasta 2MB
  - Hasta 25GB/seg
- Memoria Principal
  - > 4GB
  - > 5GB/seg
- Disco duro
  - Cientos de GB
  - Hasta 150 MB/seg
- Cintas de back-up
  - 200 GB por cinta
  - 30 MB/seg



Introducción

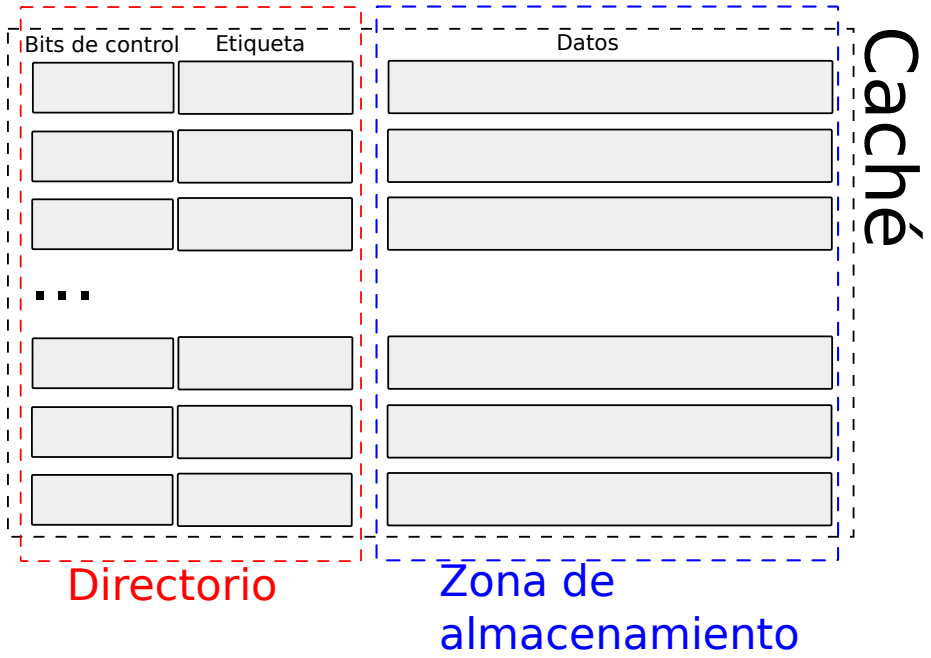
- La memoria caché es una memoria pequeña y rápida que se sitúa entre la CPU y la memoria principal
- Almacena la información actualmente en uso de la memoria principal
- En los computadores actuales está implementada dentro del chip del procesador (tecnología semiconductores)
- Es una memoria volátil
- Es una memoria asociativa: los datos se referencian a través de su contenido
  - Buscamos un dato determinado dentro de la memoria sin saber si está o no
  - Tampoco sabemos necesariamente en qué posición de la memoria se encuentra



Estructura de la caché

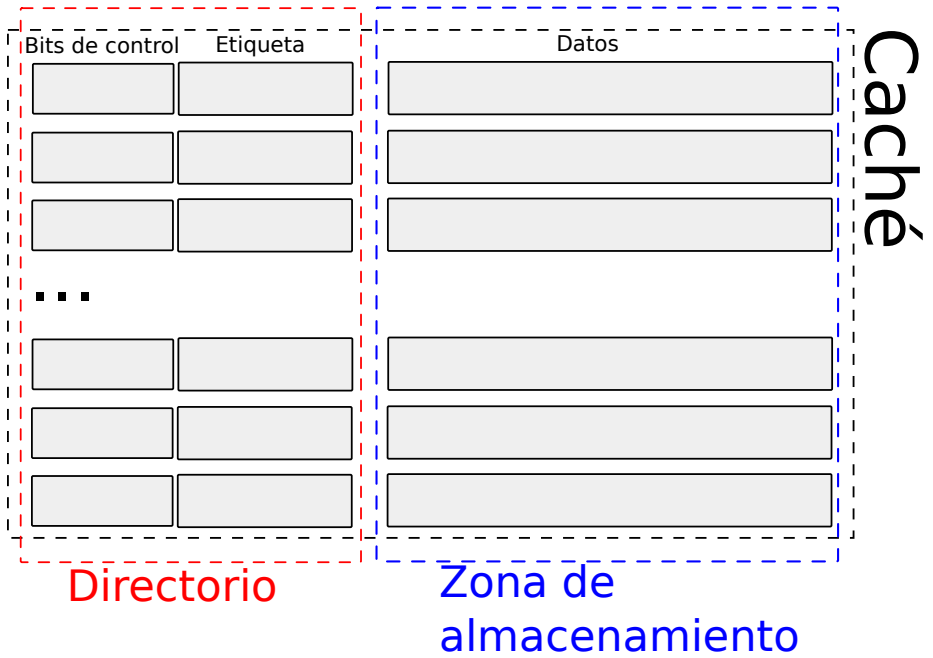
La información de la caché se divide en bloques llamados líneas

- Llamamos tamaño de línea al número de bytes que contiene una línea de la cache (1 o más palabras)
- La línea es la unidad indivisible dentro de una cache
  - No es posible cargar ni eliminar porciones de una línea
  - Cuando se carga una palabra en la cache, se carga toda la línea que la contiene
  - Esto permite aprovechar la localidad espacial



Estructura de la caché

- Zona de almacenamiento: contiene la copia de algunas líneas de memoria principal
- Directorio: contiene información sobre las líneas (bloques) de la zona de almacenamiento:
  - Los bits de control que contienen información de control relacionada con cada línea caché
    - ★ El bit de validez permite conocer si un bloque de la caché no tiene información válida
  - La etiqueta para identificar si una palabra de la caché se corresponde con la palabra buscada.



Estructura de la caché

Etiquetas y datos

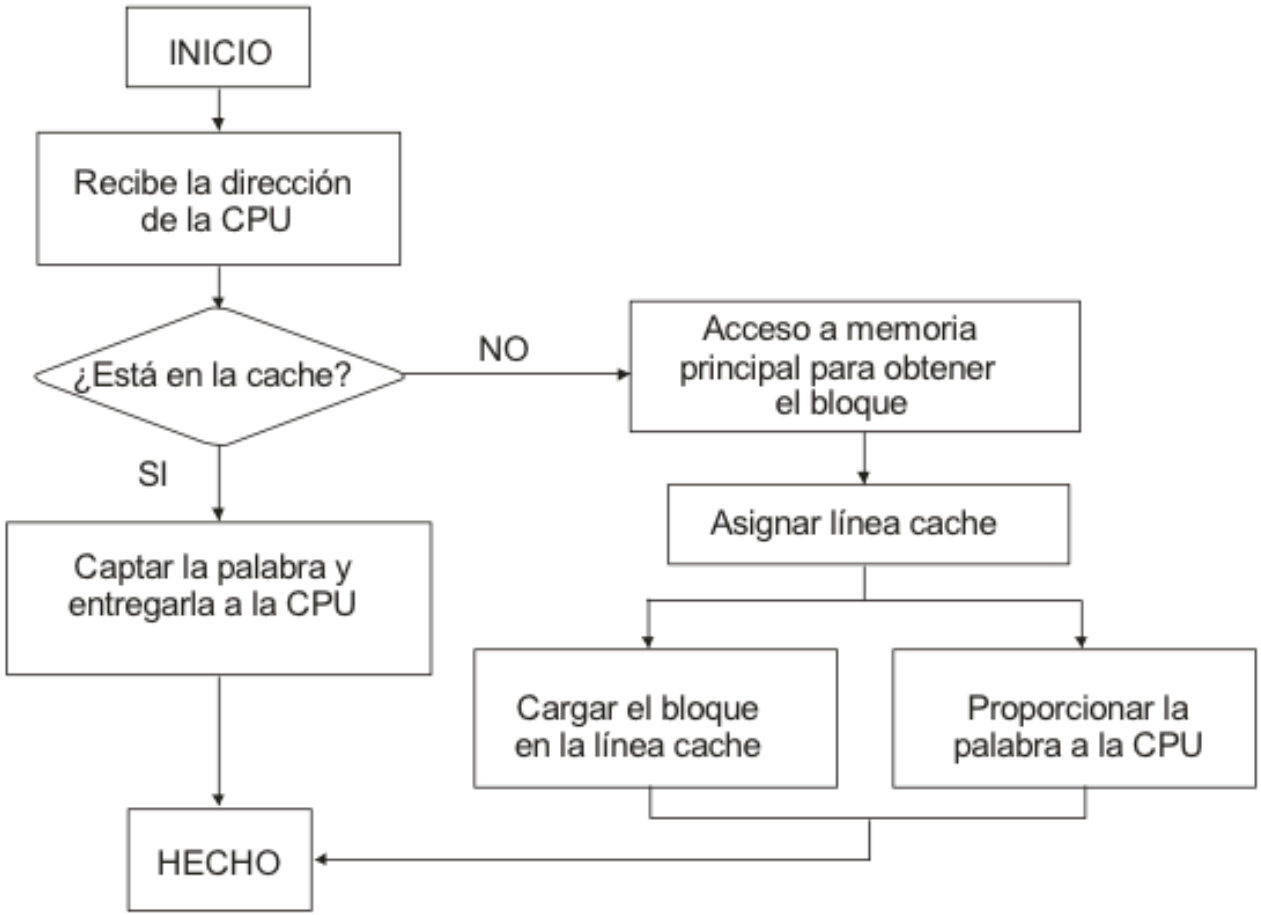
- En la cache se almacenan datos, por lo que cada línea dispone de espacio para ellos
- Los datos almacenados en una línea pueden provenir de cualquier dirección de memoria a la que esté asignada esa línea
- Por tanto, no está claro el origen de un dato almacenado en la cache
- Para resolver este problema, cada línea almacena también una etiqueta, que indica la dirección en memoria del dato que contiene
- Ejemplo: Una cache de 16 líneas de un solo byte
  - El dato en la dirección 0010 1001 0101 1011 es: 1111 0000
  - Si ese dato entra en la cache lo hará en la línea 1011
  - La etiqueta será: 0010 1001 0101
  - El dato será 1111 0000

Estructura de la caché

Reemplazos y validez

- En un principio la cache está vacía
- Cada línea tiene un **bit de validez** que indica si la línea contiene datos válidos ó no
- A medida que se llena la cache se van activando los bits de validez
- Cuando no hay posiciones libres donde poder ubicar una línea es necesario reemplazar una de las que hay cargadas

Operación de un sistema caché



Índice

Organización de un sistema caché

- Existen tres categorías de organización dependiendo de la función de correspondencia (algoritmo que hace corresponder bloques de memoria principal a líneas cache)
  - **Correspondencia directa:** hace corresponder cada bloque de memoria principal a sólo una línea posible de caché.
  - **Asociativa (o totalmente asociativas):** permite que el bloque de memoria pueda cargarse en cualquier línea de la caché.
  - **Asociativa por conjuntos:** solución de compromiso que recoge lo positivo de las dos soluciones anteriores
- Las cachés asociativas o asociativas por conjuntos permiten que cada bloque de memoria pueda ser alojado en más posiciones diferentes de la caché
  - Reduce la tasa de fallos
  - Complica el diseño:
    - Máyor consumo de energía
    - Más costosas
    - Mayor tiempo de acceso

Organización de un sistema caché

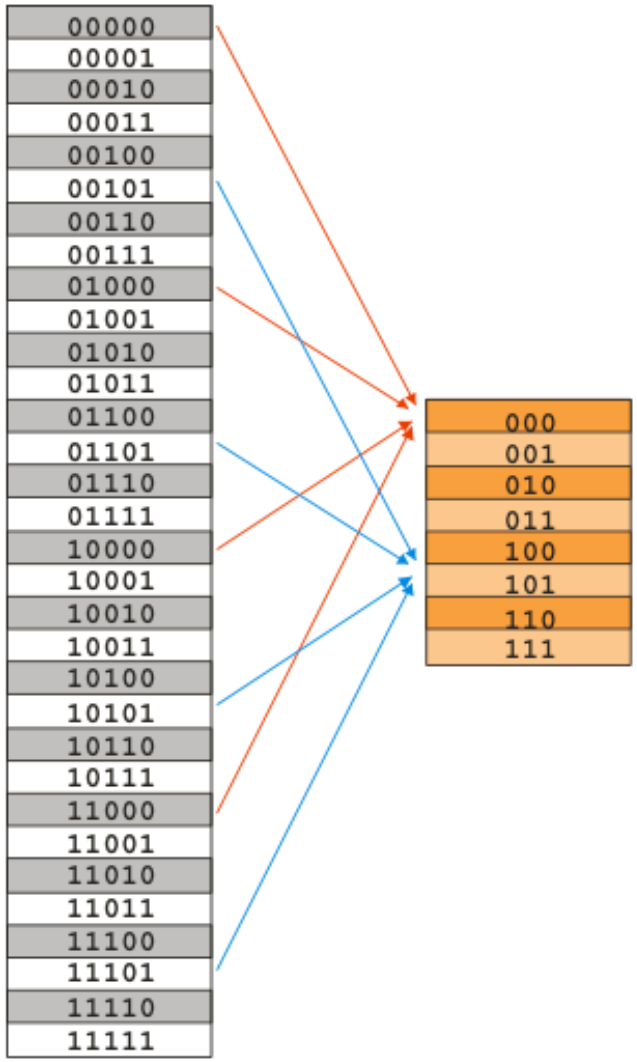
Caché de correspondencia directa

- Físicamente, una cache de correspondencia directa es una pequeña y rápida memoria estática de acceso aleatorio
- Su principal característica es que cada dato tiene asignada una posición fija en la cache

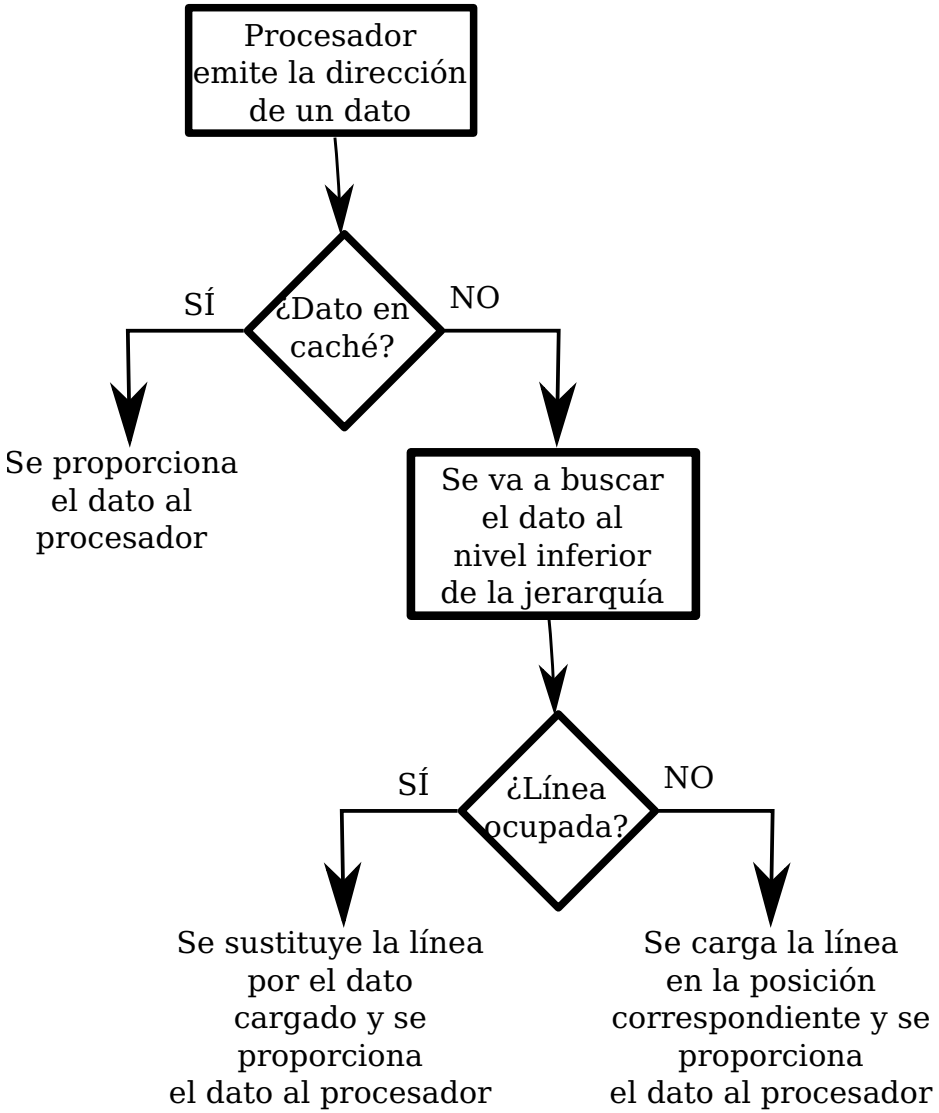
Organización de un sistema caché

Caché de correspondencia directa

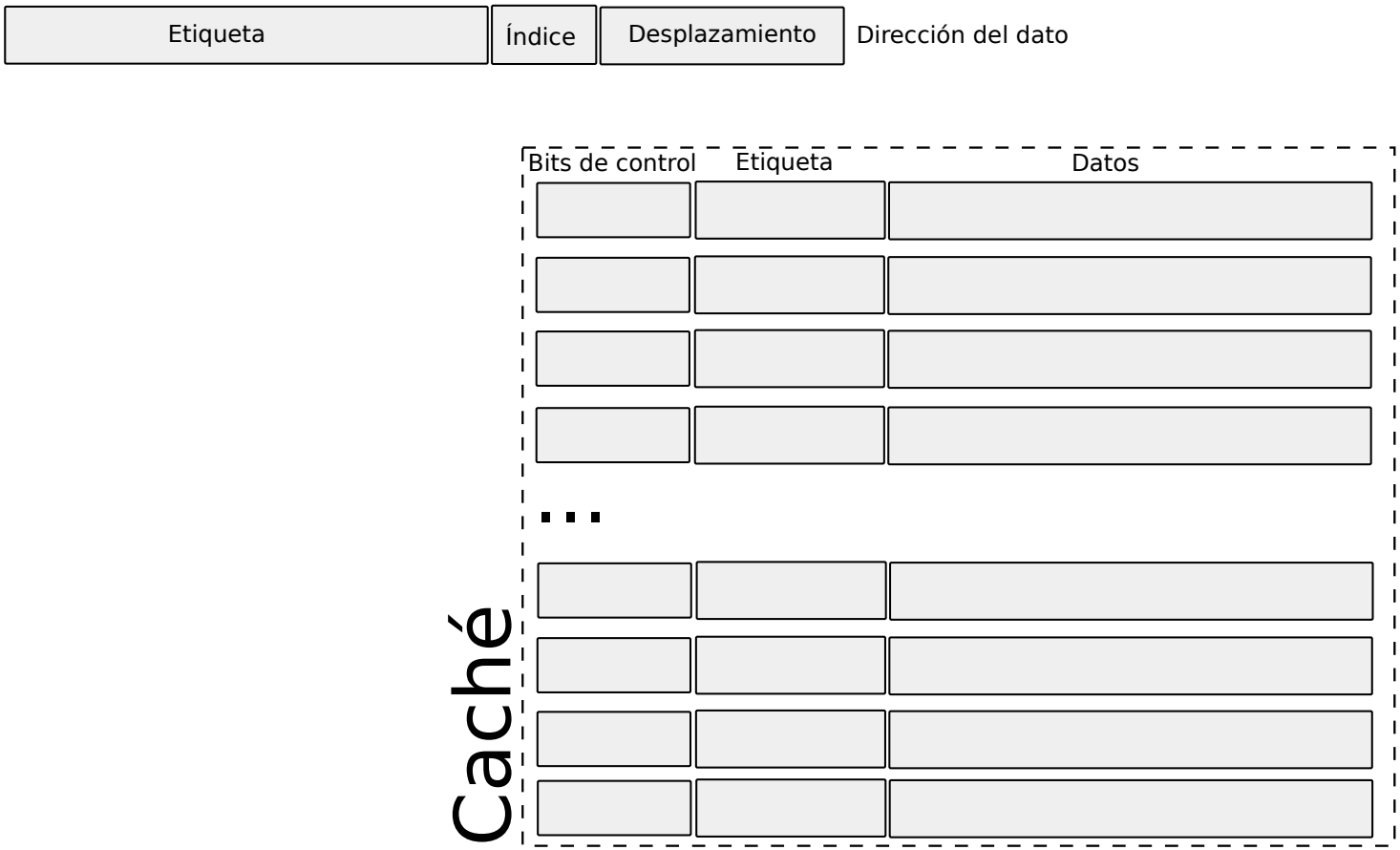
- Dada una cache de asignación directa con p posiciones, un dato almacenado en la dirección d sólo podrá alojarse en la posición **d módulo p** de la caché



Operación de una caché de correspondencia directa: General

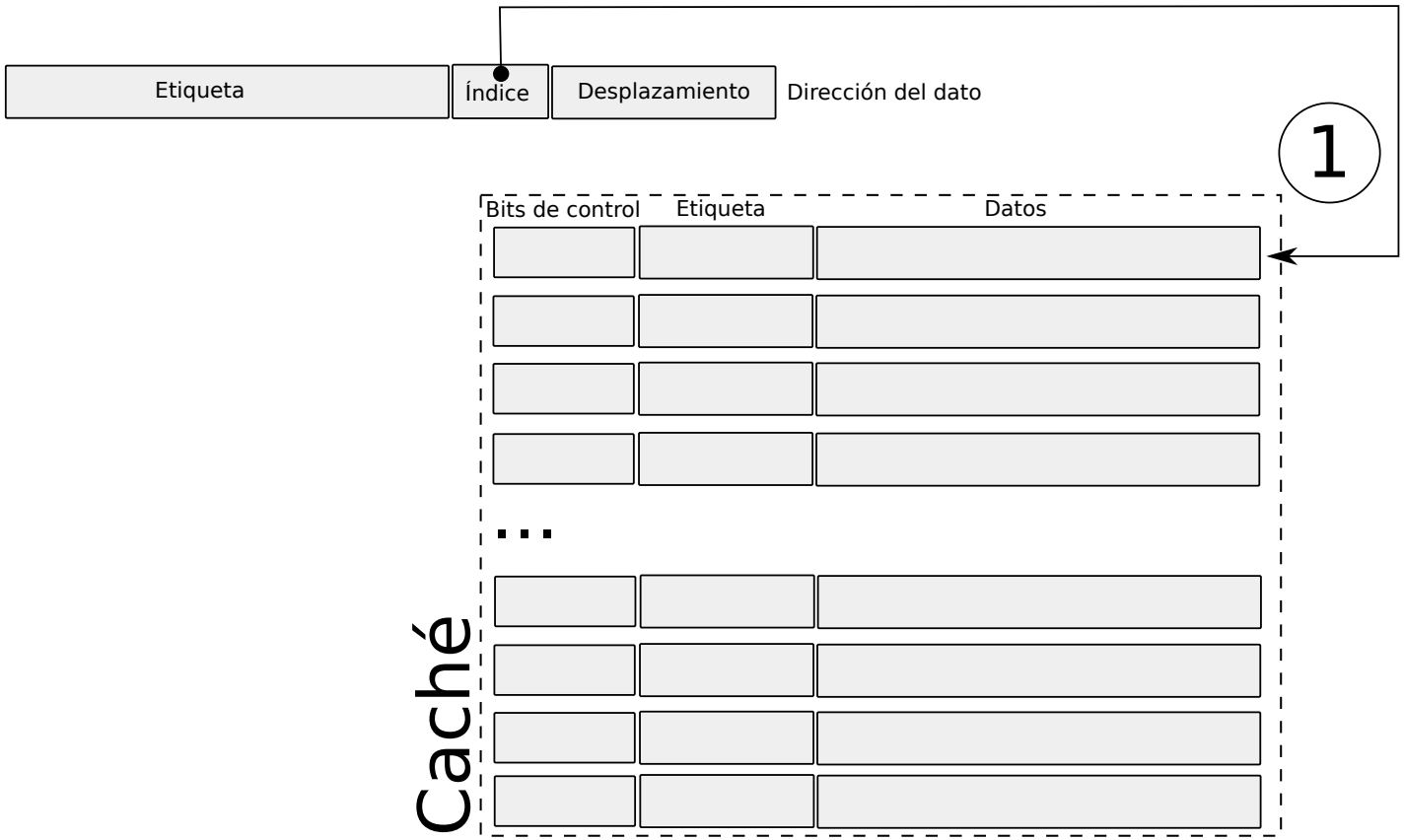


Operación de una caché de correspondencia directa: Acierto caché



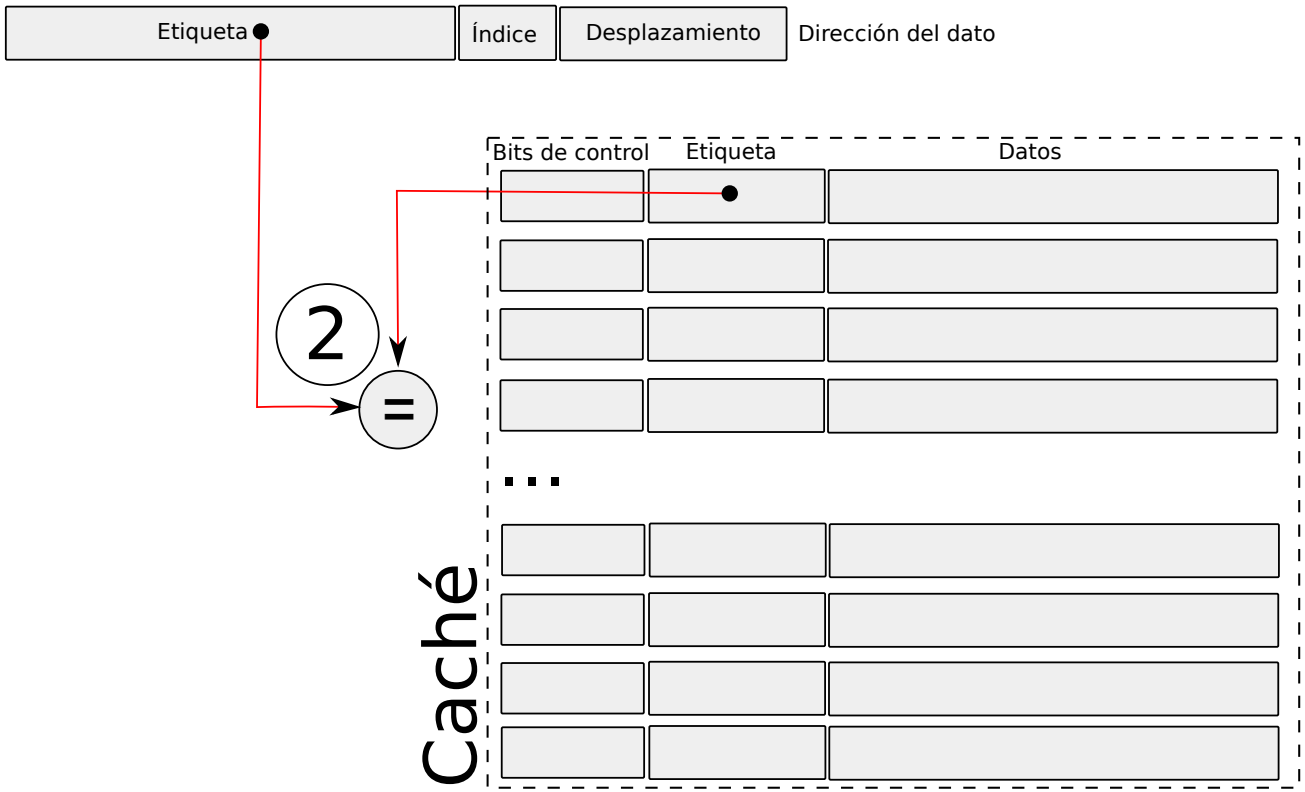
El procesador emite la dirección del dato que quiere buscar

Operación de una caché de correspondencia directa: Acierto caché



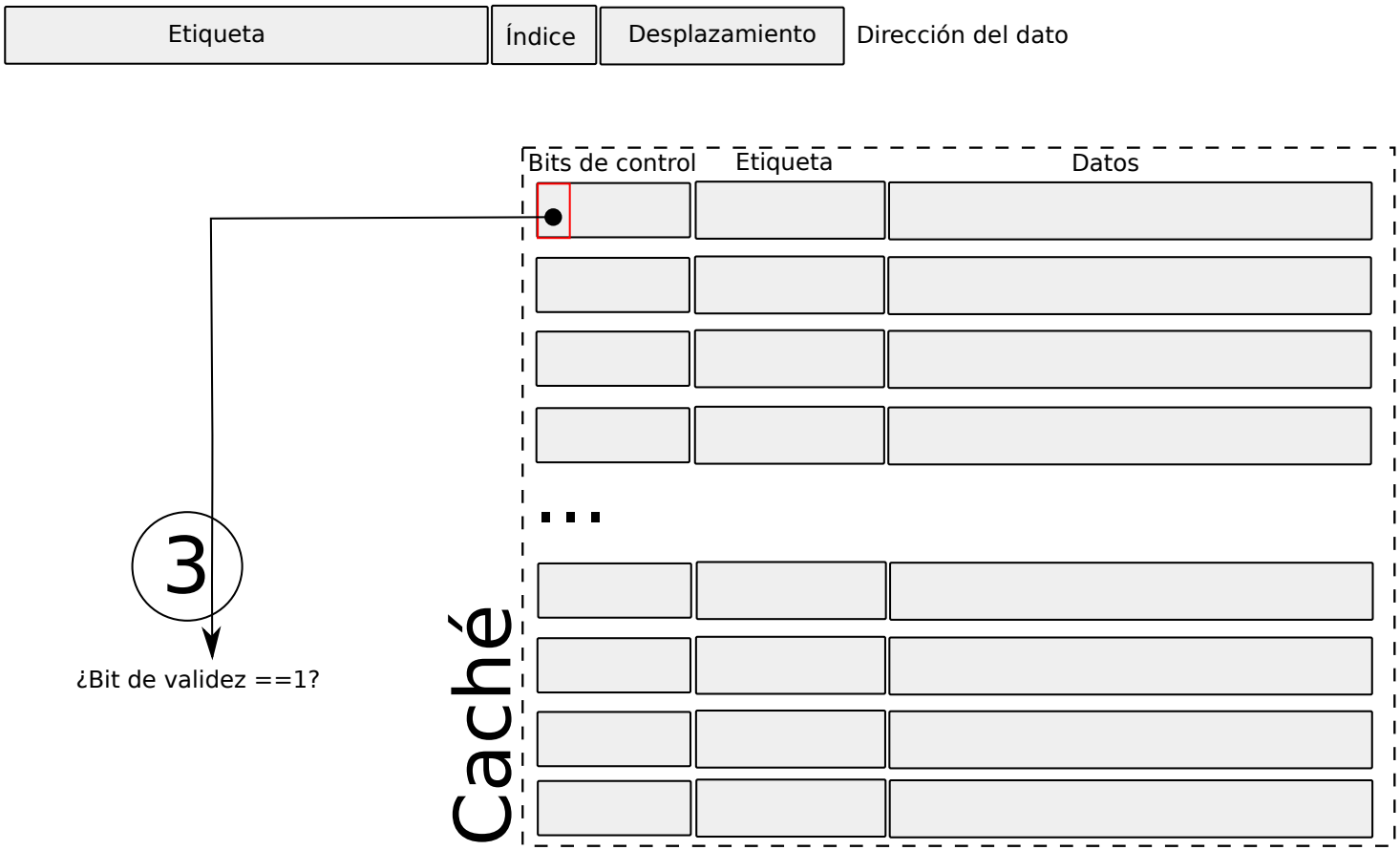
(1) Se utiliza el campo índice de la dirección para seleccionar la línea adecuada

Operación de una caché de correspondencia directa: Acierto caché



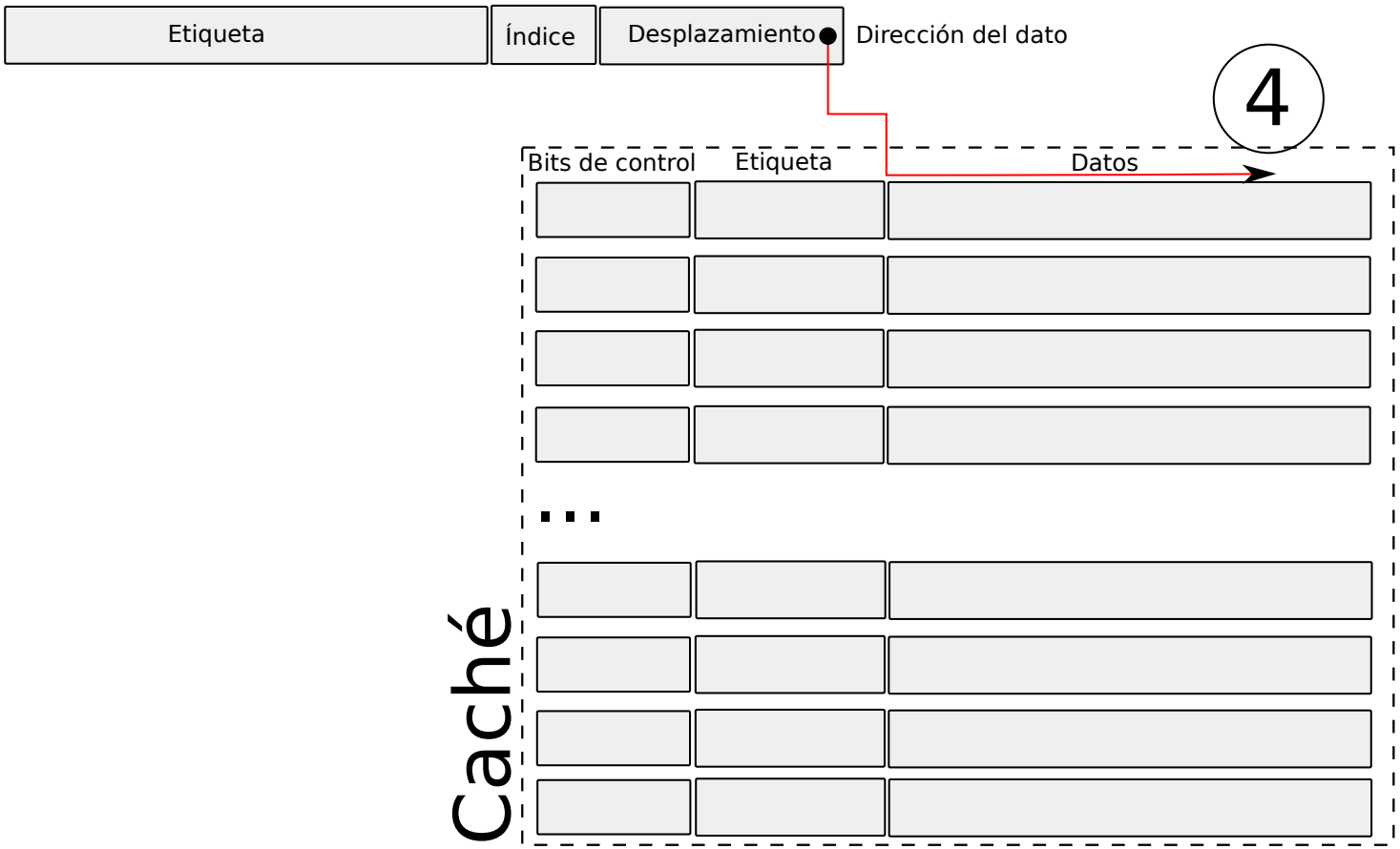
(2) Se compara la etiqueta de la línea seleccionada con la etiqueta del dato buscado

Operación de una caché de correspondencia directa: Acierto caché



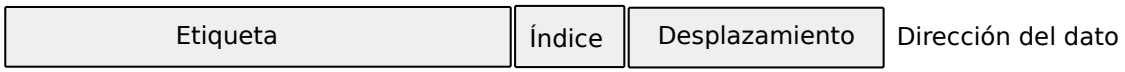
(3) Se comprueba que la línea además tiene su bit de validez a 1

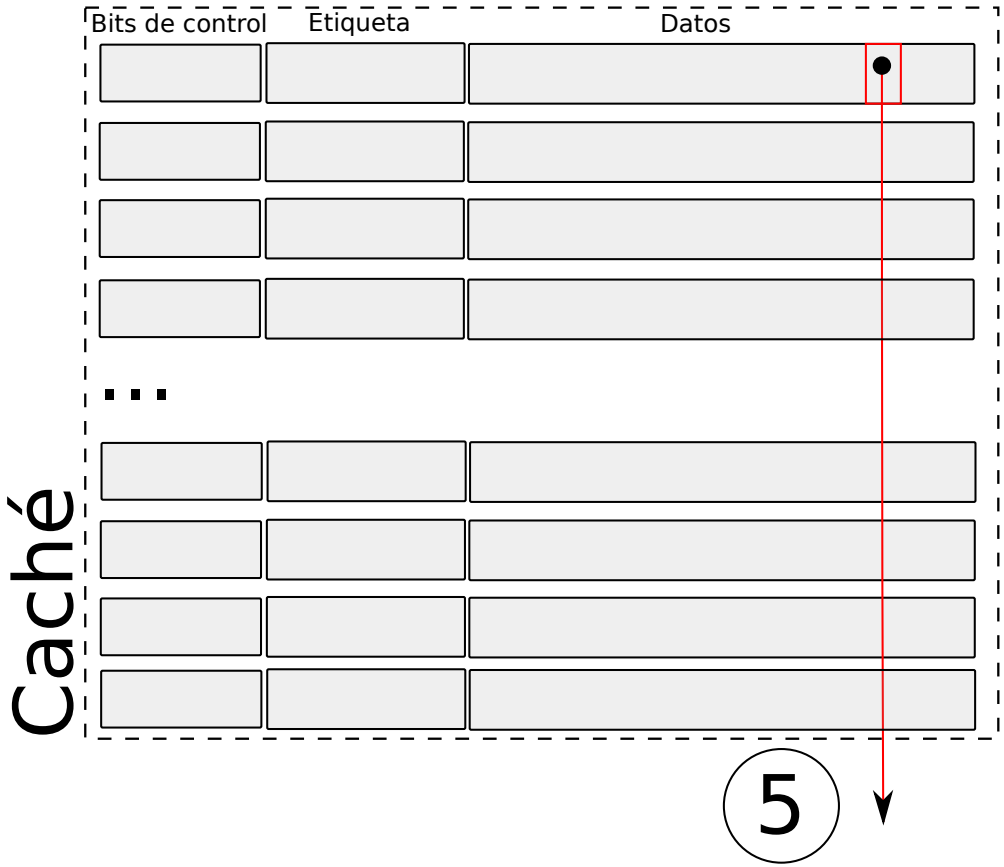
Operación de una caché de correspondencia directa: Acierto caché



(4) Se usa el desplazamiento para acceder a la parte de la línea solicitada

Operación de una caché de correspondencia directa: Acierto caché





(5) Se proporciona el dato al procesador

Organización de un sistema caché

Caché totalmente asociativa

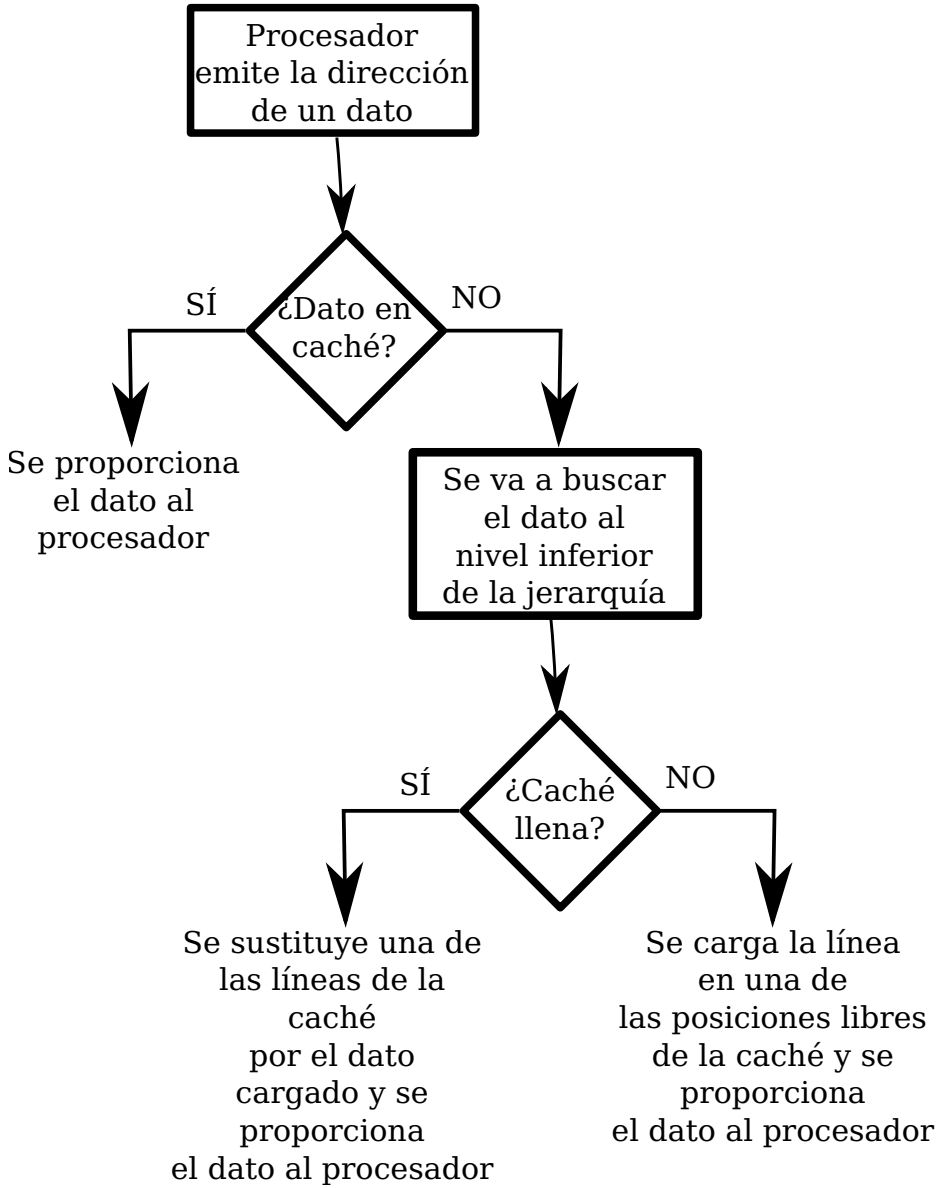
- Permite que un bloque de memoria principal se cargue en cualquier línea de la memoria cache
- Mientras la cache no está llena la asignación de línea no está fijada
- Cuando está llena se necesita un algoritmo de reemplazo
- Técnicamente, una cache asociativa es una memoria direccionable por contenido
- En lugar de utilizar una dirección, se utiliza una etiqueta, y la memoria la buscará en su directorio y devolverá el dato asociado
- Esto es muy complejo y costoso porque todas las comparaciones se realizan en paralelo

Organización de un sistema caché

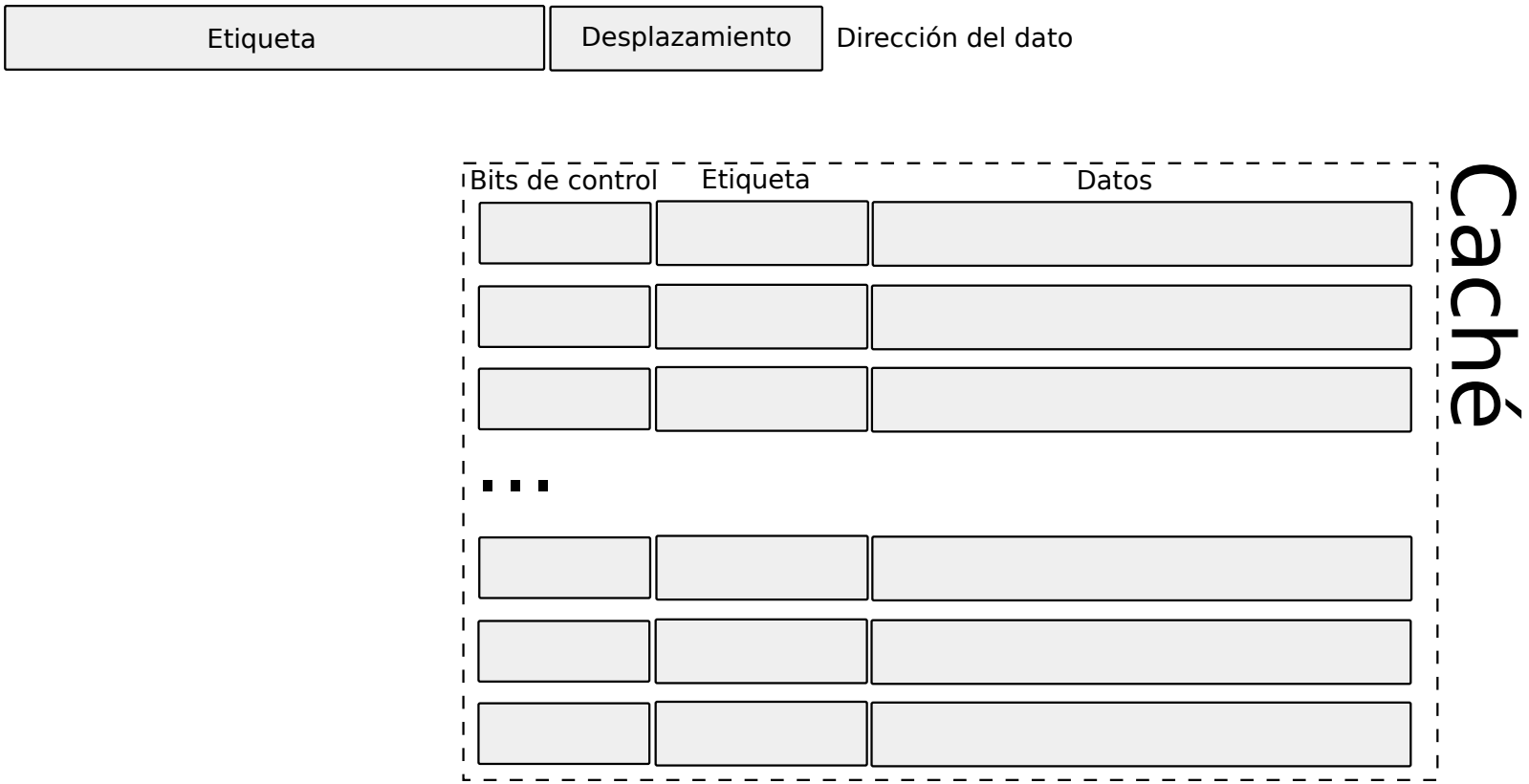
Algoritmos de reemplazo

- Su función es decidir qué línea sale de la cache cuando es necesario un reemplazo
- Son algoritmos sencillos que se implementan en hardware
  - FIFO (primero en entrar, primero en salir)
  - LRU (utilizado menos recientemente)
  - LFU (utilizado con menos frecuencia)
  - Aleatorio

Operación de una caché totalmente asociativa: General

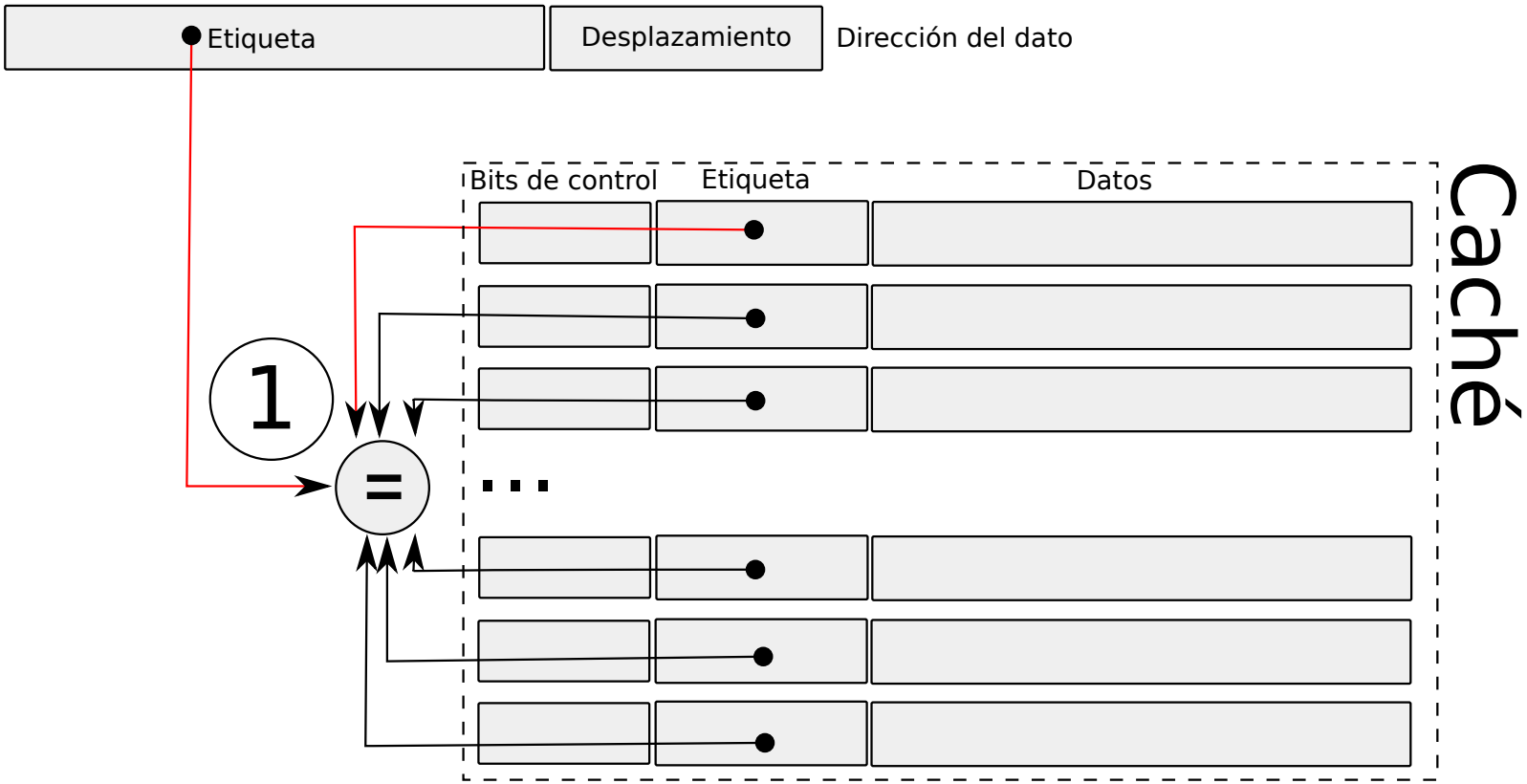


Operación de una caché totalmente asociativa: Acierto



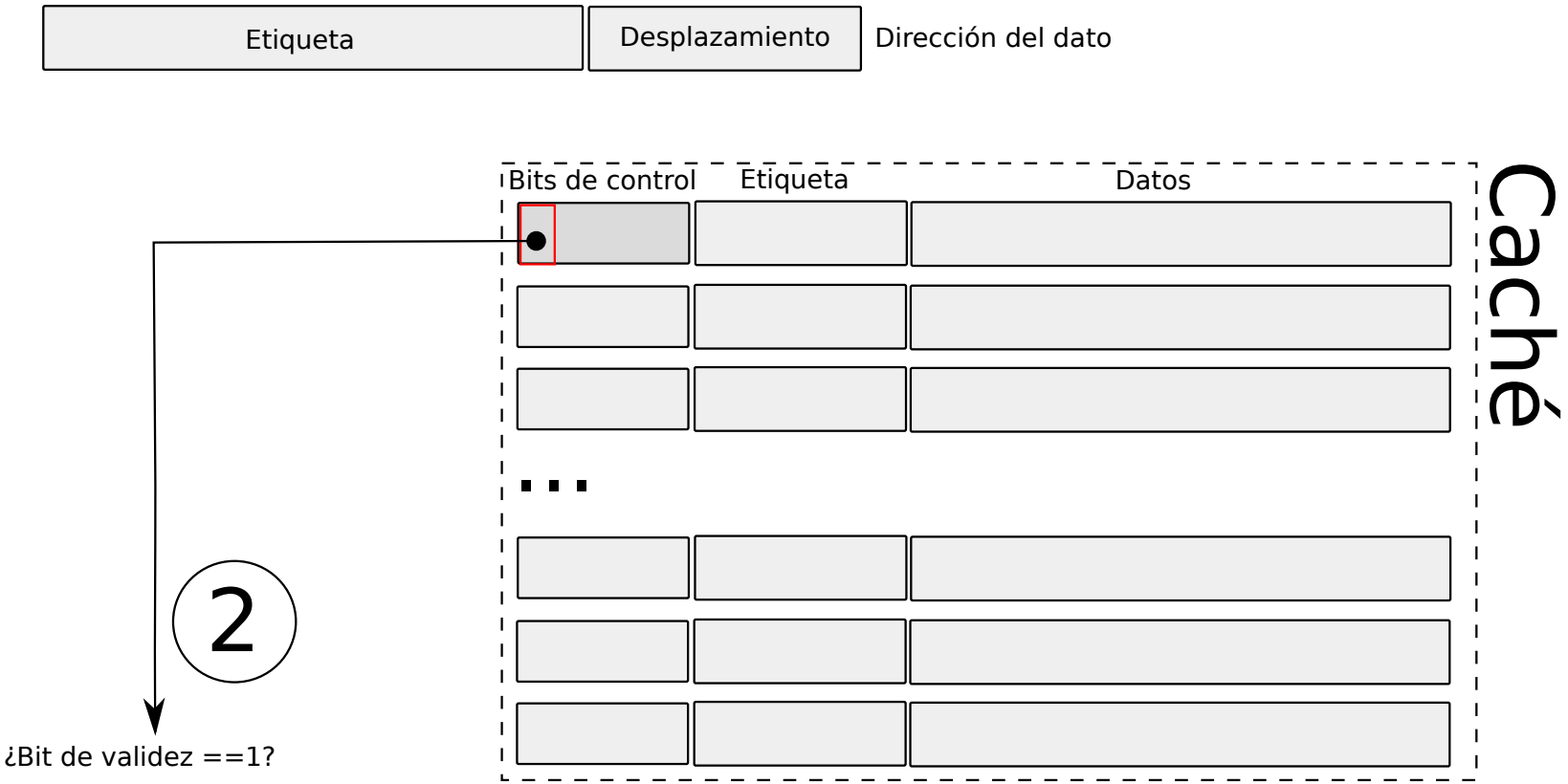
El procesador emite la dirección del dato que quiere buscar

Operación de una caché totalmente asociativa: Acierto



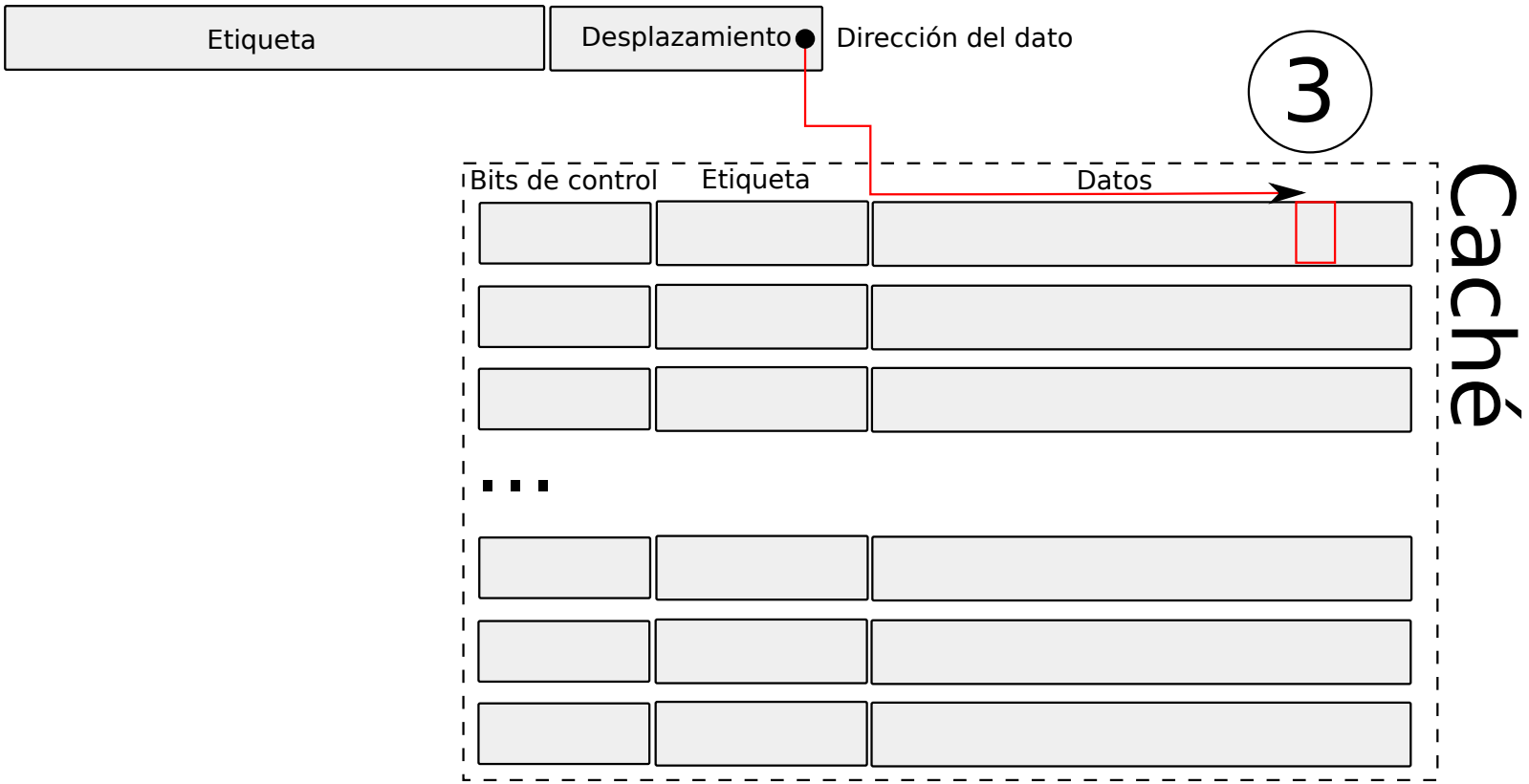
(1) Se compara la etiqueta de todas las líneas con la etiqueta del dato buscado

Operación de una caché totalmente asociativa: Acierto



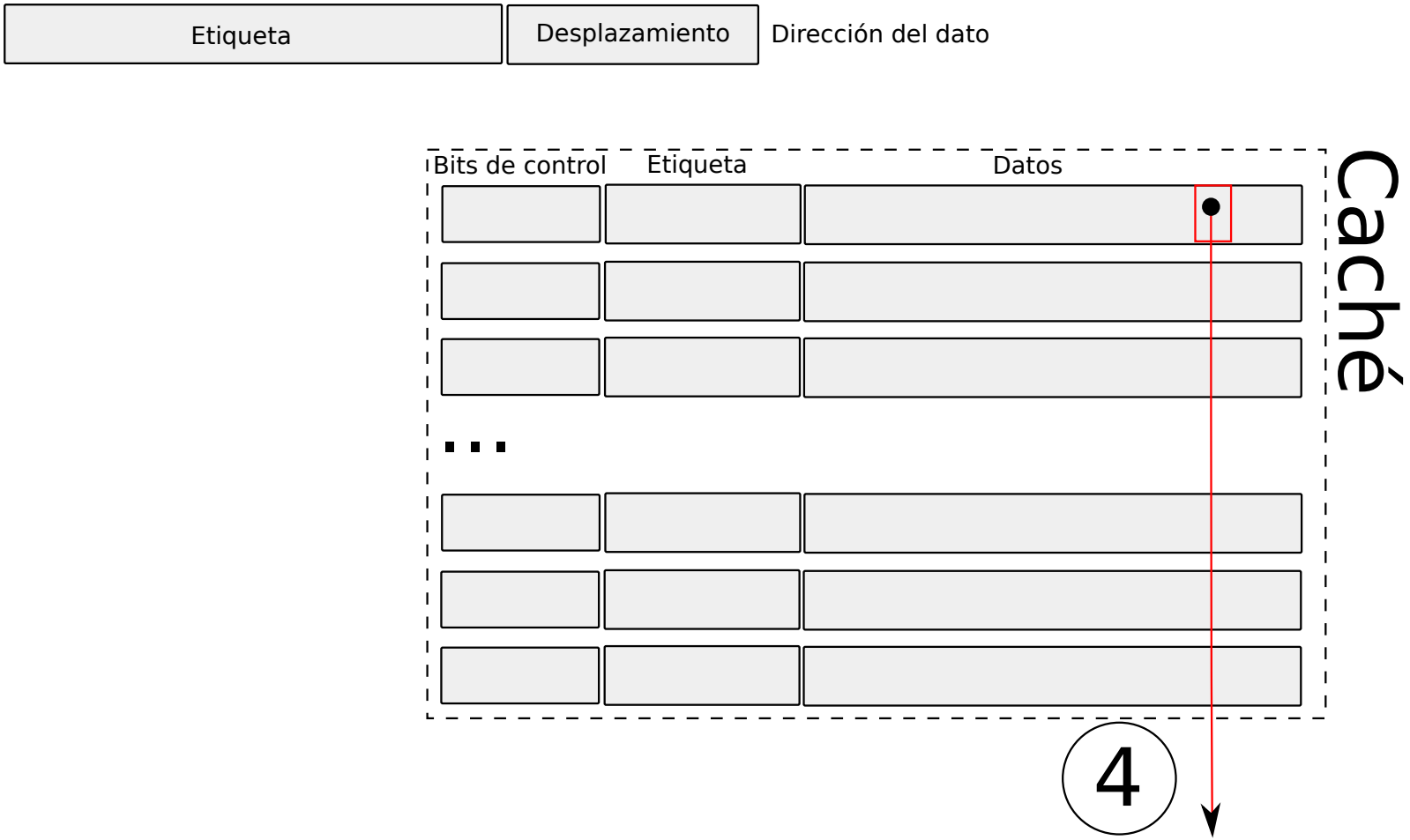
(2) Se comprueba que la línea encontrada tiene su bit de validez a 1

Operación de una caché totalmente asociativa: Acierto



(3) Se usa el desplazamiento para acceder a la parte de la línea solicitada

Operación de una caché totalmente asociativa: Acierto



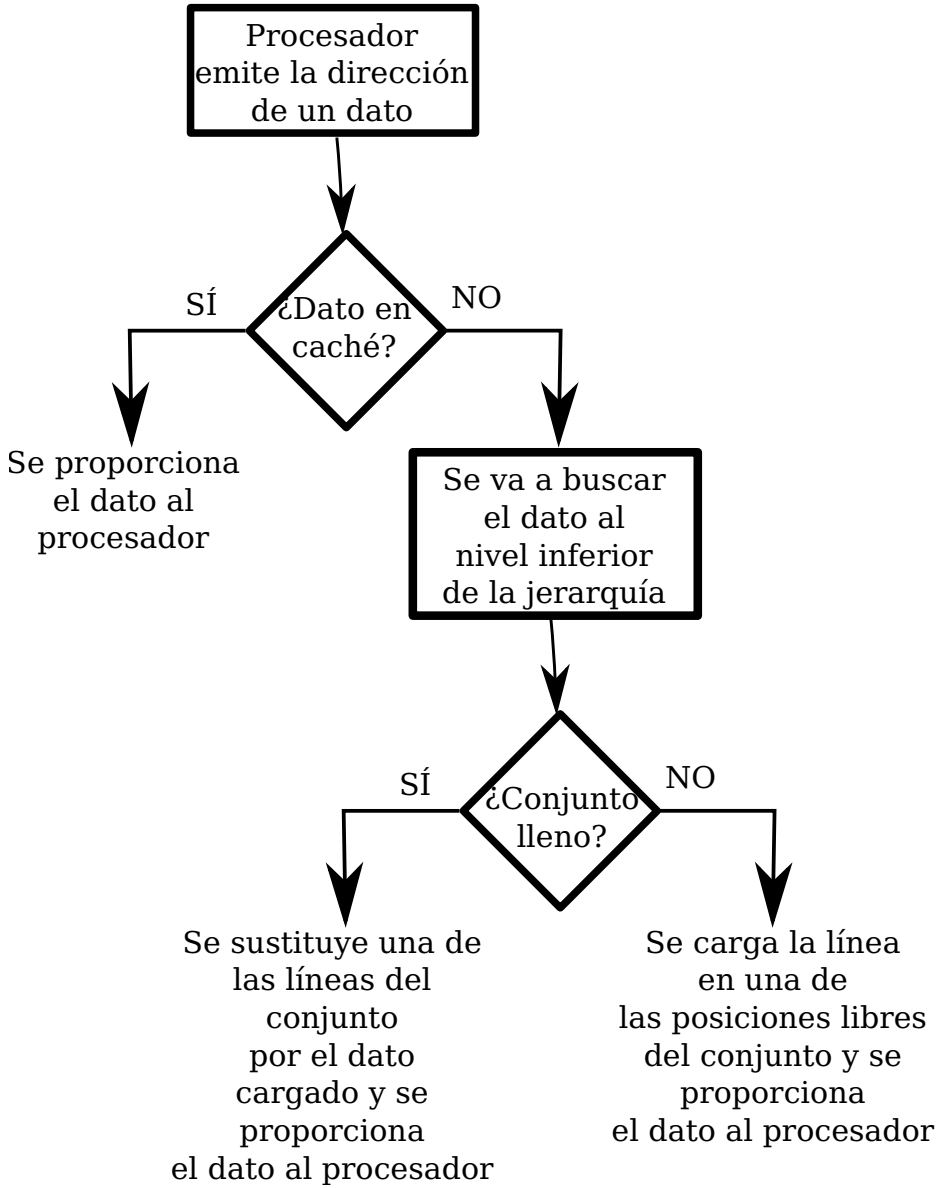
(4) Se proporciona el dato al procesador

Organización de un sistema caché

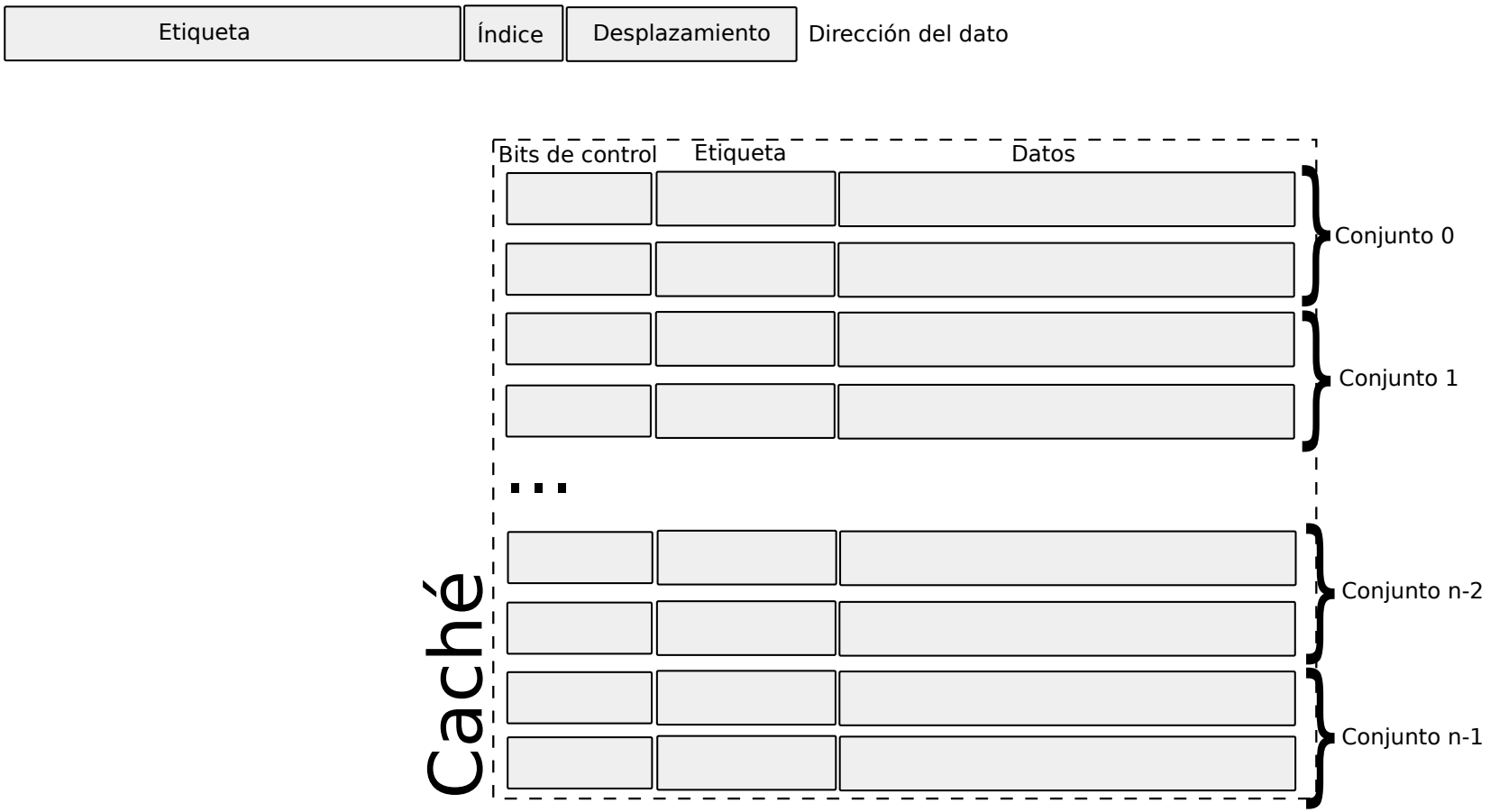
Asociatividad por conjuntos

- Consiste en dividir la cache en conjuntos de varias líneas
- Dentro de cada conjunto la asociatividad es total
- Al número de líneas por conjunto le llamamos número de vías
- Cada dirección de memoria tiene asignado un conjunto de forma directa, pero dentro de ese conjunto puede ir a cualquier línea
- Es una solución intermedia entre la asociatividad total y la correspondencia directa
- Consigue resultados próximos a la asociatividad total con un coste razonable

Operación de una caché asociativa por conjuntos: General

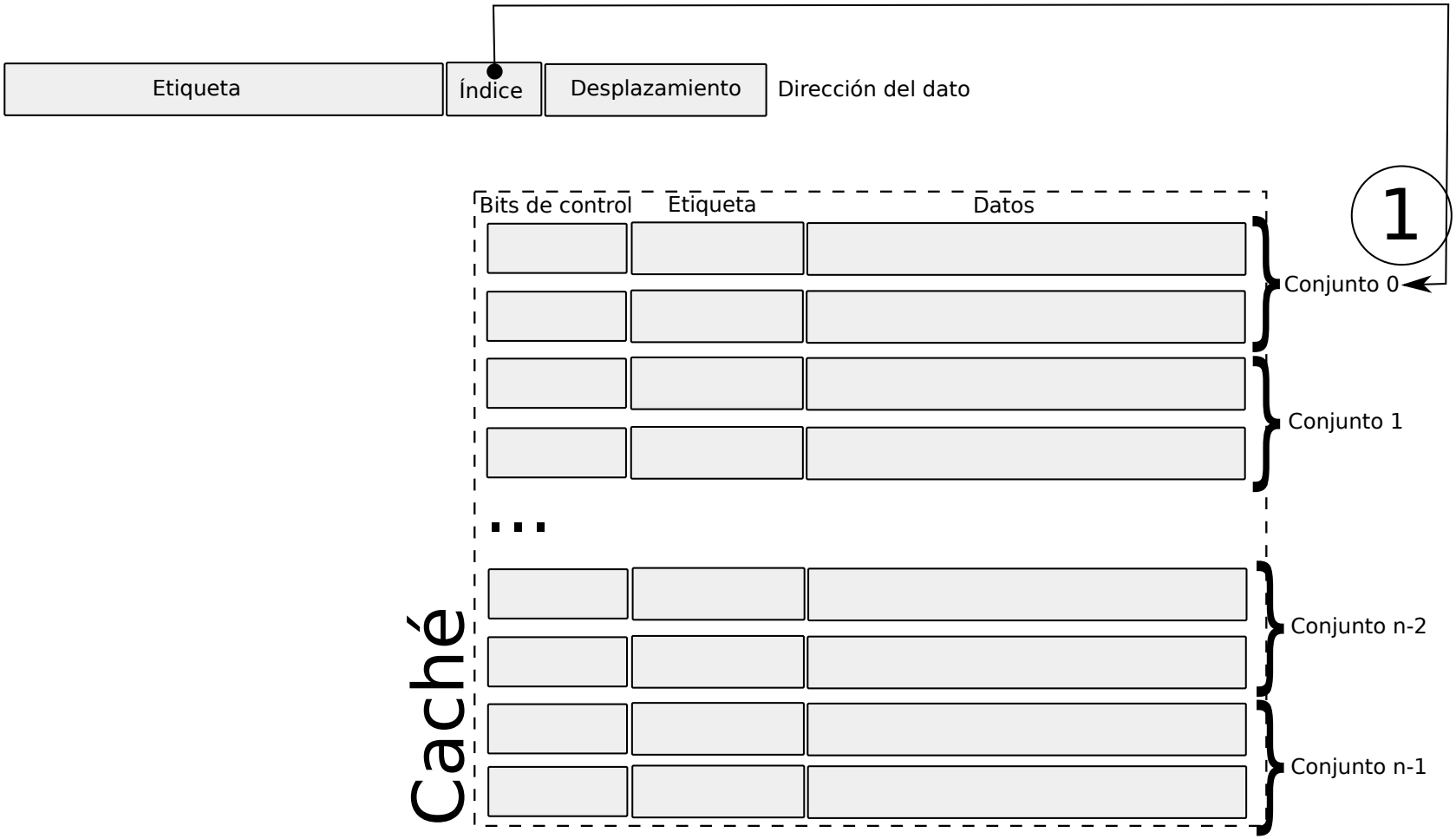


Operación de una caché asociativa por conjuntos: Acierto



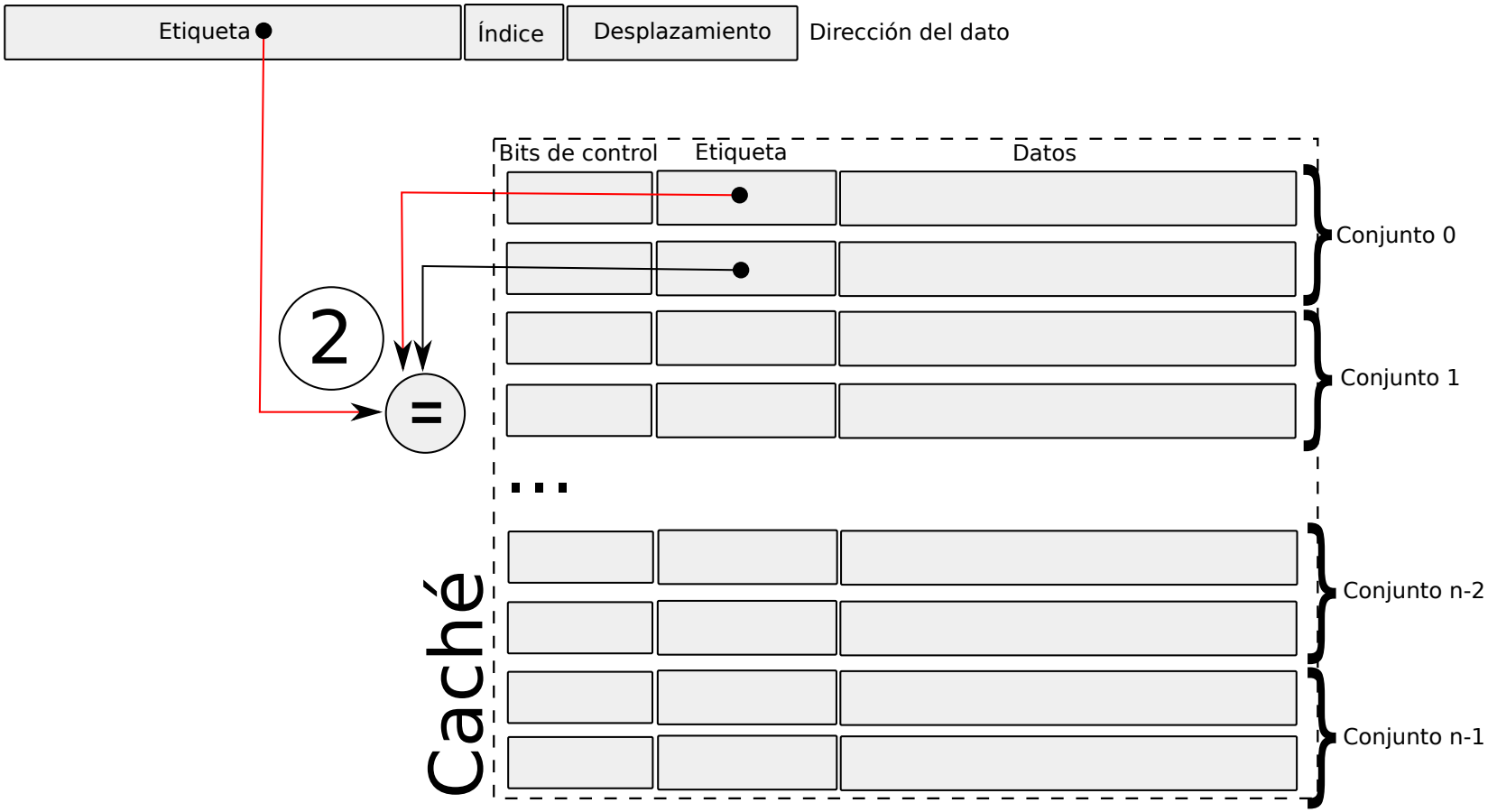
El procesador emite la dirección del dato que quiere buscar

Operación de una caché asociativa por conjuntos: Acierto



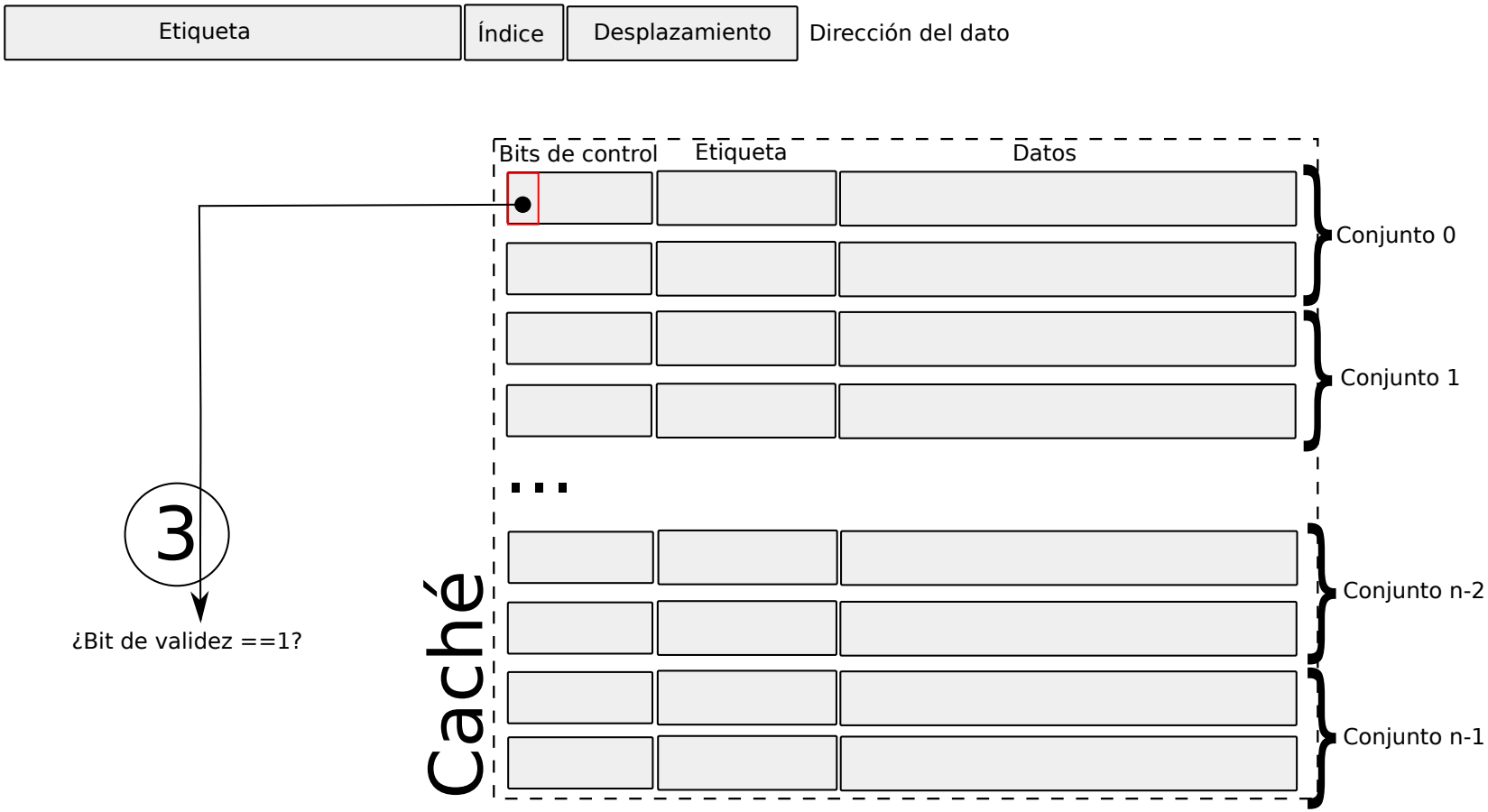
(1) Se utiliza el campo índice de la dirección para seleccionar el conjunto de líneas adeucado

Operación de una caché asociativa por conjuntos: Acierto



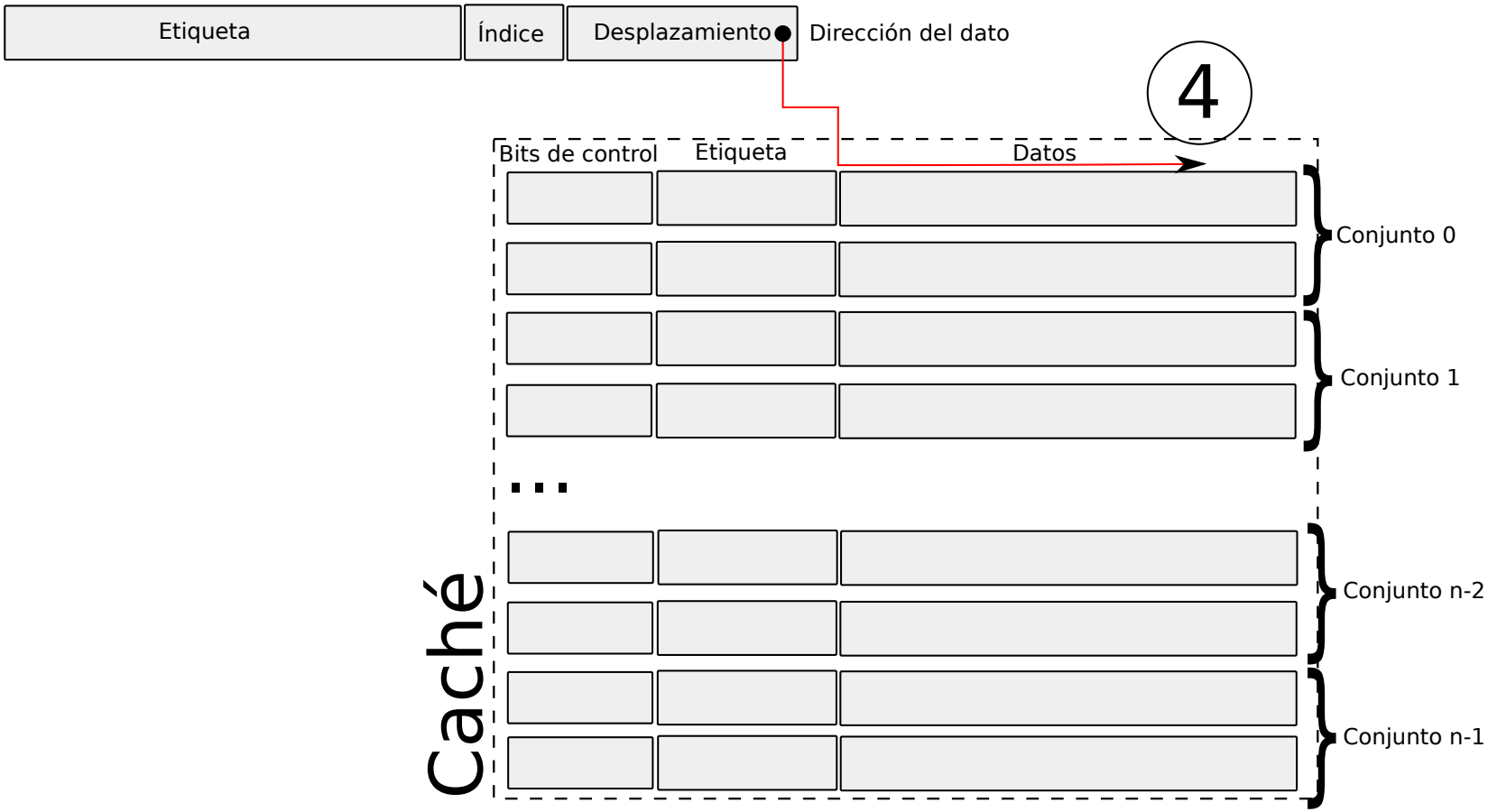
(2) Se compara la etiqueta de todas las líneas del conjunto con la etiqueta del dato buscado

Operación de una caché asociativa por conjuntos: Acierto



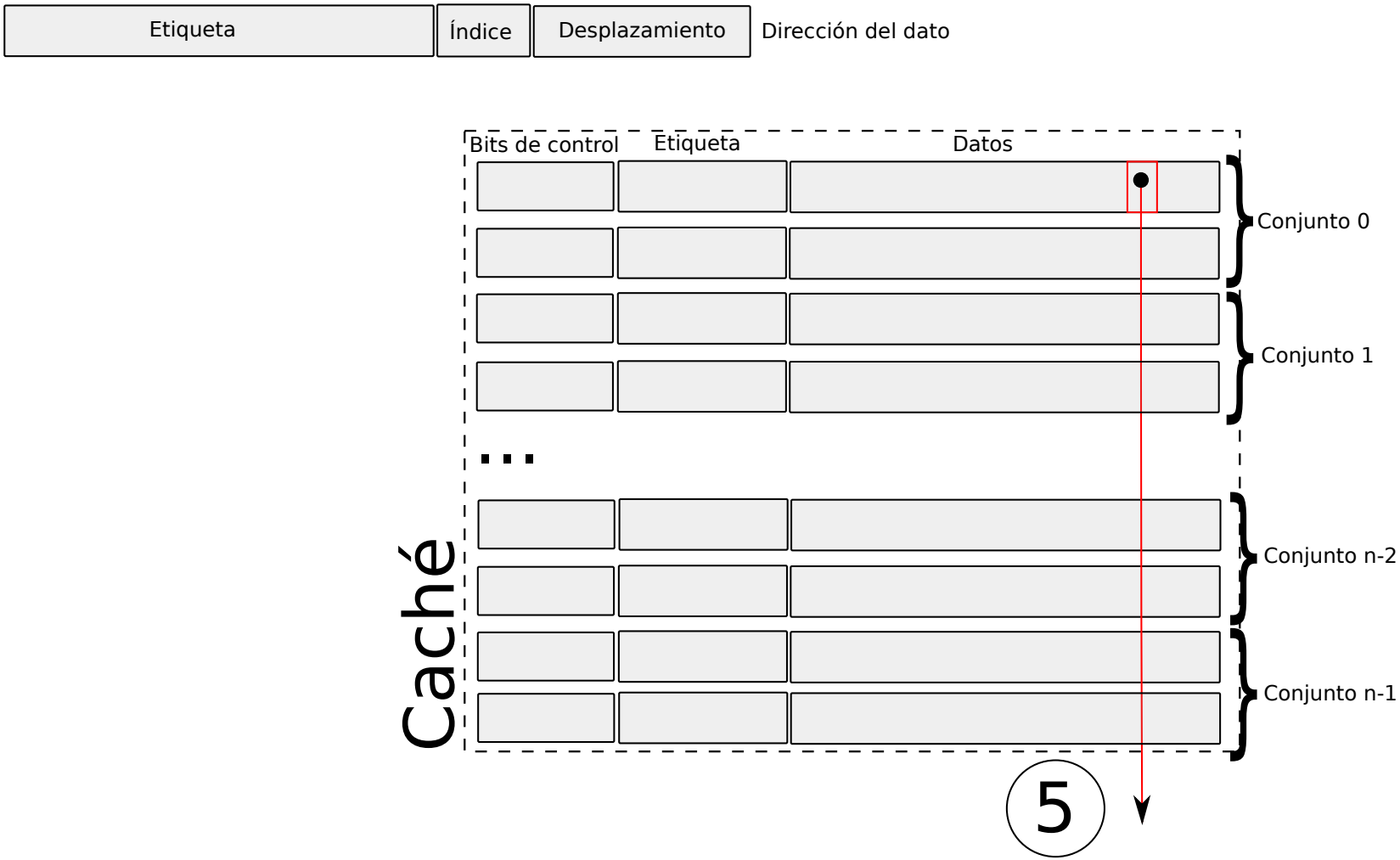
(3) Se comprueba que la línea encontrada tiene su bit de validez a 1

Operación de una caché asociativa por conjuntos: Acierto



(4) Se usa el desplazamiento para acceder a la parte de la línea solicitada

Operación de una caché asociativa por conjuntos: Acierto



(5) Se proporciona el dato al procesador

Escritura en la caché

- Cuando un programa escribe en memoria y el dato antiguo **ya está** en la cache:
  - Se escribe sólo en la cache (**Post-escritura** ó **write-back**) y se indica que esa línea ha sido modificada (mediante un **bit de modificación**) de forma que solo se escribe en niveles inferiores en caso de reemplazo
  - Se escribe en la cache y en los niveles inferiores de la jerarquía de memoria (**Escritura directa** ó **write-through**)
- Post-escritura reduce el número de escrituras a memoria principal porque:
  - Si el valor se modifica varias veces sólo se copiará 1 vez a los otros niveles de memoria
  - Si se escriben palabras sueltas no se copiará toda la línea
- Escritura directa puede resultar más segura porque:
  - Siempre habrá una copia del dato actualizado en la memoria principal (consistencia cache)
  - Cuando se produce un reemplazo con post-escritura es necesario copiar el nuevo valor a memoria al mismo tiempo que se lee el nuevo

Escritura en la caché

- Cuando un programa escribe en memoria y el dato antiguo **no está** en la cache:
  - La línea se carga (**ubicar en escritura**) y siguen las acciones de acierto de escritura (en post-escritura).
    - ⋆ Se busca que las escrituras subsiguientes en la línea no provoquen fallos
  - Se escribe directamente en otro nivel y la línea nunca viene a la cache (**no ubicar en escritura**)
    - ⋆ Las escrituras subsiguientes en la línea provocarán escrituras a memoria
    - ⋆ Este es un modo avanzado que requiere intervención del programador. Se utiliza en programas que generan gran cantidad de resultados y en los que la localidad temporal no se cumple de forma satisfactoria, p.e. muchos programas multimedia

Índice

Rendimiento de la caché

Eficiencia

- La principal medida de la eficiencia de una cache es la relación entre aciertos y fallos
  - Mejor en cachés asociativas o totalmente asociativas
- Existen otras medidas de eficiencia
  - El consumo de energía (Más bajo en cachés de correspondencia directa)
  - El tiempo de acceso (Más bajo en cachés de correspondencia directa)

Rendimiento de la caché

Tipos de fallos

- Forzosos:
  - Se producen la primera vez que se referencia un dato/instrucción
  - Se pueden reducir aumentando el tamaño de línea
- De capacidad:
  - Cuando es referencia un dato que fue reemplazado porque la cache estaba llena
  - Se pueden reducir aumentando el tamaño de la cache
- De conflicto (o de colisión):
  - En correspondencia directa ó asociativas por conjuntos cuando 2 líneas compiten por la misma posición en la cache
  - Se reducen aumentado la asociatividad de la cache
- Esta clasificación se conoce como las 3 C (compulsory, capacity, conflict)

Rendimiento de la caché

Métricas

- Tasa de fallos: número de fallos / total de referencias (TF)
- Tasa de aciertos: número de aciertos /total de referencias ($1 - TF$)
- Tiempo de acierto: tiempo de acceso a la caché ($t_{acierto}$)
- Tiempo de fallo (t_{fallo}): tiempo de acierto ($t_{acierto}$) + penalización por fallo (PF)
- Tiempo medio de acceso a memoria: tiempo de acierto + (tasa de fallos $\times$ penalización por fallo)
  $\overline{t_{acceso}} = (1 - TF) \times t_{acierto} + TF \times t_{fallo}$
  $= (1 - TF) \times t_{acierto} + TF \times (t_{cierto} + PF) = t_{acierto} + TF \times PF$

$$T_{cpu} = N \times (CPI_{ejecucion} + (\frac{Accesos\_memoria}{Instrucciones}) \times TF \times PF) \times T_{ciclo}$$

Rendimiento de la caché

Posibilidades de configuración

- Tamaño de la memoria cache
- Tamaño de línea
- Asociatividad
  - Correspondencia directa
  - Totalmente asociativa o por conjuntos
    - ⋆ Elección del algoritmo de reemplazo
- Caches de datos e instrucciones
- Caches en varios niveles
- Política de escritura

Índice

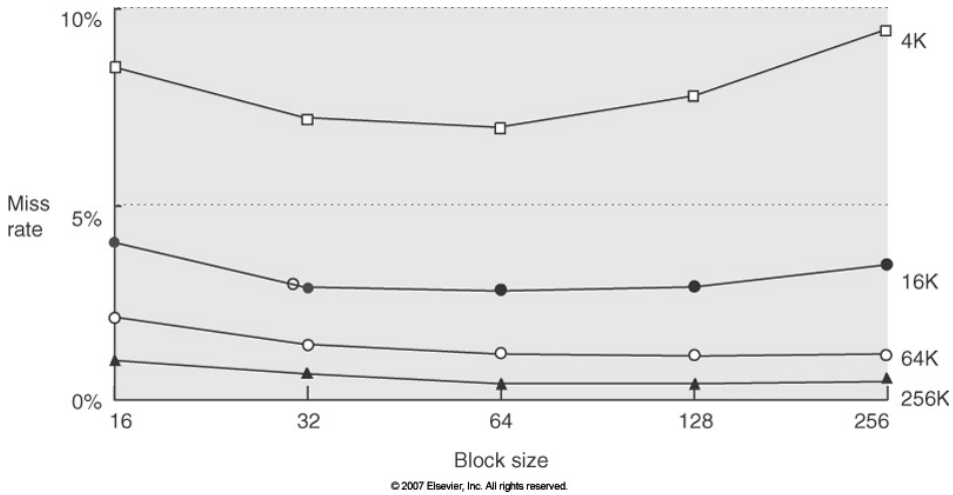
Cachés de datos e instrucciones

- Las **cachés unificadas o mixtas**, que contienen instrucciones y datos, pueden ser un cuello de botella (instrucciones *load* y *store*)
- La CPU sabe si está emitiendo la dirección de una instrucción o un dato, pudiendo separar puertos
- Ventajas de las cachés separadas:
  - Casi doblan el ancho de banda entre la CPU y la jerarquía de memoria
  - Optimización por separado de cada caché
  - Elimina fallos de conflicto entre instrucciones y datos
- Inconvenientes:
  - Espacio fijo para instrucciones y datos: peor tasa de aciertos
  - Inconsistencia: instrucciones y datos en la misma línea, instrucciones modificables

Reducción de la tasa de fallos

Incrementar el tamaño de línea

- A mayor tamaño de línea:
  - Mayores oportunidades de aprovechar la localidad espacial (secuencial)
    - ★ Reduce los fallos forzosos
  - A igual tamaño de cache, menor número de líneas
    - ★ Aumenta el número de reemplazos y puede incrementar los fallos de conflicto y los de capacidad
  - Incrementa la penalización por fallo: mayor coste temporal de transferir una línea



Reducción de la tasa de fallos

Incrementar la asociatividad

- A mayor asociatividad:
  - Menor número de fallos de conflicto
  - Caché más lenta: mayor tiempo de acierto
  - Mayor coste
- En la práctica, una caché asociativa por conjuntos de 8 vías es casi tan efectiva como una caché totalmente asociativa
- Una caché de correspondencia directa tiene aproximadamente la misma tasa de fallos que una caché asociativa por conjuntos de 2 vías de la mitad de tamaño
- Otras técnicas para reducir la tasa de fallos:
  - Caché víctima
  - Caché pseudo-asociativa

Reducción de la tasa de fallos

- Optimizaciones en tiempo de compilación
  - Fusión de arrays
  - Intercambio de lazos
  - Fusión de lazos
  - Blocking
- Prebúsqueda hardware/software

Estas técnicas se ven en las sesiones prácticas

Reducción de la penalización de fallo

Caché de dos niveles

- Caché de primer nivel de velocidad comparable a la de la CPU
- Caché de segundo nivel de gran tamaño para capturar la mayoría de los fallos que irían a memoria principal
- La ganancia radica en que la penalización de fallo del primer nivel de caché es el tiempo medio de acceso de la de segundo nivel
- La velocidad de la caché del primer nivel afecta a la velocidad de reloj de la CPU, y la de segundo nivel a la penalización de fallo de la de primer nivel
- El aumento de la asociatividad en el segundo nivel tiene poco impacto en el tiempo de acierto en este nivel, pero el aumento de la capacidad disminuye también los fallos de conflicto y por tanto los beneficios de la asociatividad

Reducción de la penalización de fallo

Caché de dos niveles

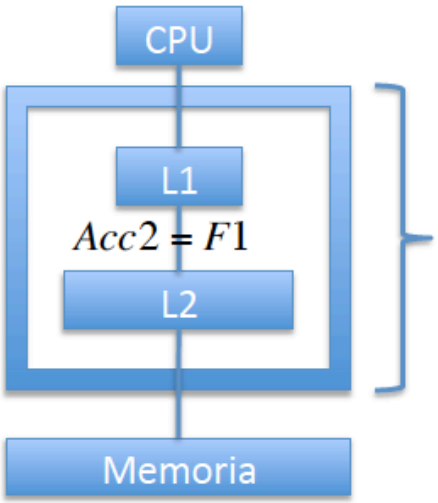
- Grandes tamaños llevan a incrementar el tamaño de la línea
  - Distintos niveles de caché pueden tener distinto tamaño de línea
- Propiedad de inclusión multinivel
  - Todos los datos de la caché de primer nivel están en la del segundo nivel
  - Deseable para mantener la consistencia
  - Algunos sistemas la cumplen y otros no
  - Se puede mantener con distintos tamaños de línea en las cachés (mediante invalidación de líneas de primer nivel)
- En las cachés del segundo nivel, el énfasis se hace en la reducción de fallos, usando cachés grandes, alta asociatividad y líneas grandes

Reducción de la penalización de fallo

Caché de dos niveles

- Tasa de fallos global

TFx = Tasa de fallos de la caché del nivel x
Accx = Número de accesos a la caché del nivel x
Fx = Número de fallos en la caché del nivel x



$$TF1 = \frac{F1}{Acc1}$$

$$TF2 = \frac{F2}{Acc2}$$

$$TF_{Global} = TF1 * TF2 = \frac{F1}{Acc1} * \frac{F2}{Acc2} = \frac{F2}{Acc1}$$

- La penalización por fallo del nivel x de la caché (PF_x) es igual al tiempo medio de acceso del nivel inmediatamente inferior ($\overline{t_{acceso}}$)

Reducción del tiempo de acierto

Caché pequeñas y simples

- El hardware pequeño es más rápido
- En una caché de correspondencia directa se puede solapar el chequeo de la etiqueta con la transmisión de los datos