Organización del Computador 1
Memoria Cache

Dr. Marcelo Risk

Departamento de Computación
Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

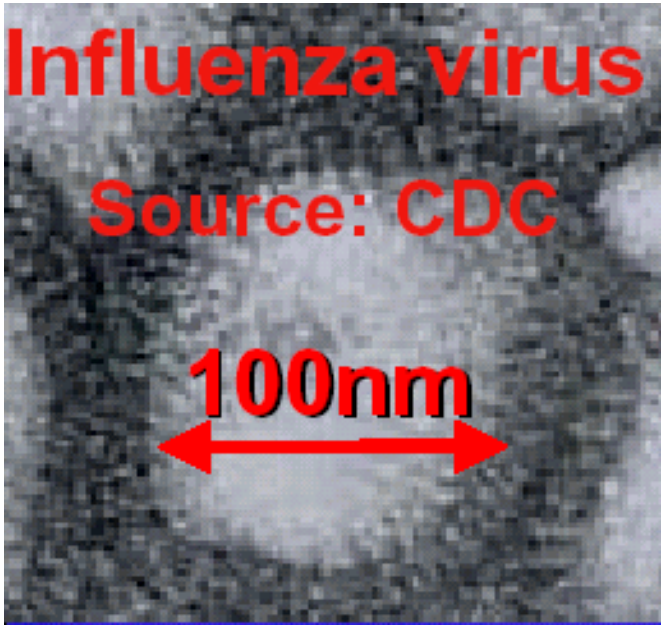
2017

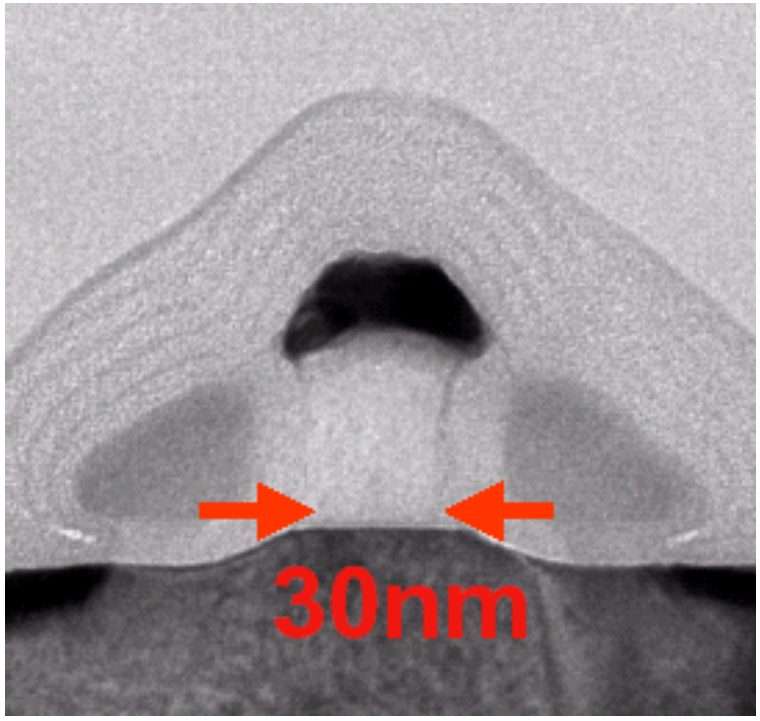
Memorias: Evolución

- Pioneros:
  Maurice Wilkes con la primer memoria de tanque de mercurio para la computadora EDSAC. 2 bytes: 1947.

- Visionarios...
  640K debe ser suficiente para cualquiera.
  Bill Gates, 1981.

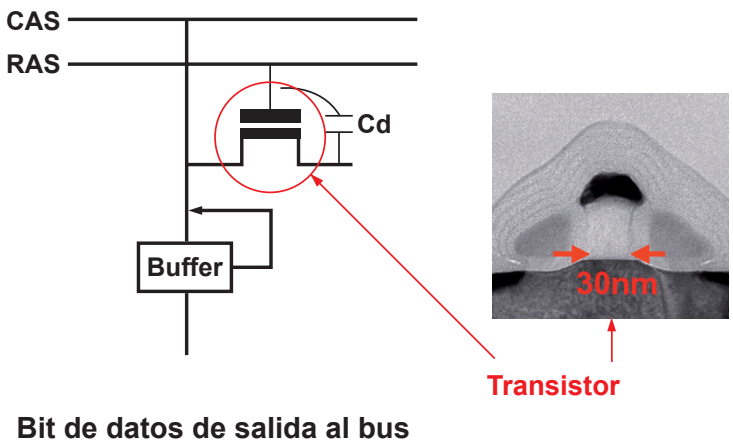
Tecnología de Integración Actual





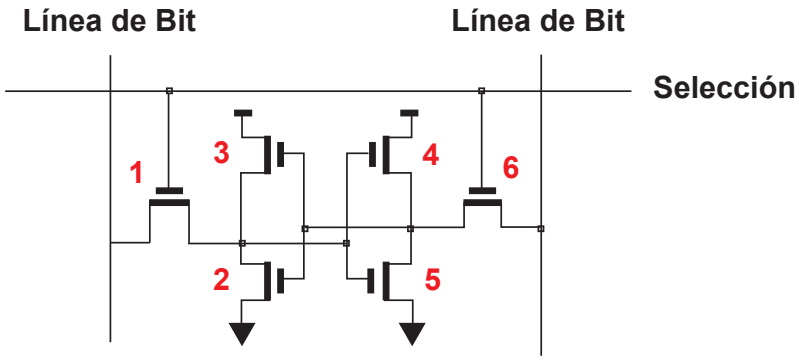
- Imágenes obtenidas con TEM (Transmission Electron Microscope) de una cepa del virus de la gripe, y de un transistor construido con la tecnología de 65 nm utilizada desde el año 2005 en el Procesador Pentium IV y posteriores.

Tecnología de Memorias: RAM Dinámica



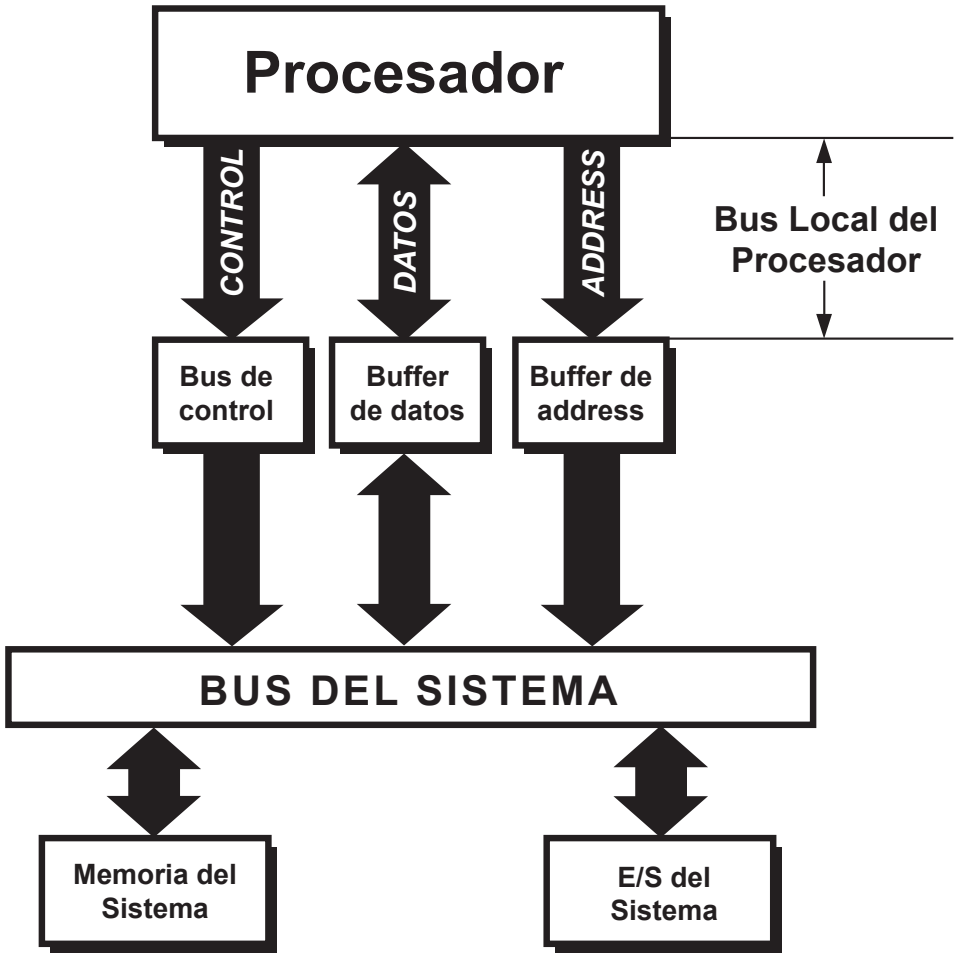
- Almacena la información como una carga en una capacidad espuria de un transistor.
- Una celda (un bit) se implementa con un solo transistor ⇒ máxima capacidad de almacenamiento por chip.
- Ese transistor consume mínima energía ⇒ Muy bajo consumo.
- Al leer el bit, se descarga la capacidad ⇒ necesita regenerar la carga ⇒ aumenta entonces el tiempo de acceso de la celda.

Tecnología de Memorias: RAM Estática



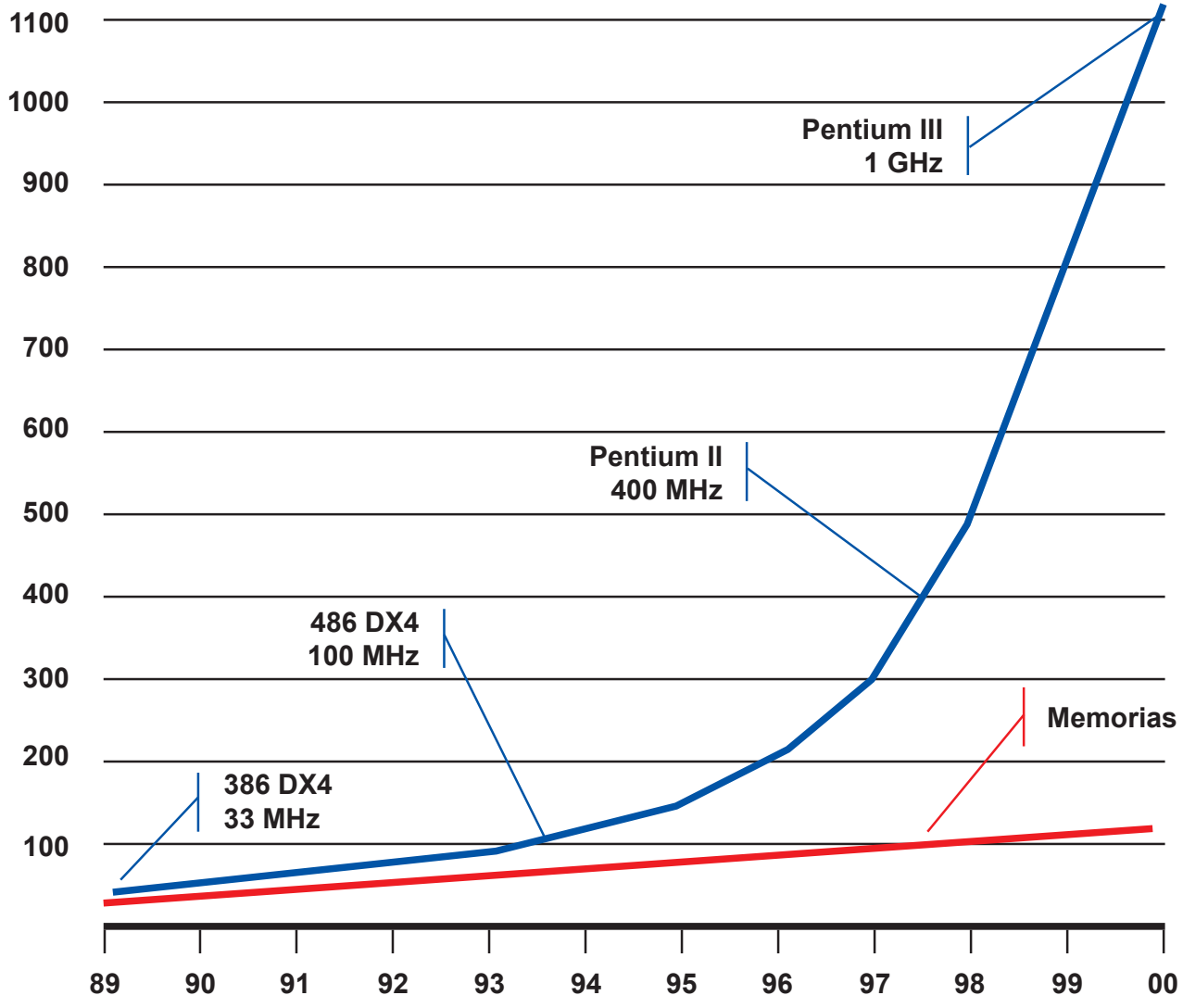
- Almacena la información en un biestable.
- Una celda (un bit) se compone de seis transistores menor capacidad de almacenamiento por chip.
- 3 transistores consumen energía máxima en forma permanente y los otros 3 consumen mínima energía ⇒ Mayor consumo.
- La lectura es directa y no destructiva ⇒ tiempo de acceso muy bajo.

Estructura de Bus clásica



- Desde fines de los años 80, los procesadores desarrollaban velocidades muy superiores a los tiempos de acceso a memoria.
- En este escenario, el procesador necesita generar wait states para esperar que la memoria esté lista *READY* para el acceso.

Crecimiento de la velocidad de clock de las CPU versus memoria



El problema

- RAM dinámica (DRAM)
  - Consumo mínimo.
  - Capacidad de almacenamiento comparativamente alta.
  - Costo por bit bajo.
  - Tiempo de acceso alto (lento), debido al circuito de regeneración de carga.
  - Si construímos el banco de memoria utilizando RAM dinámica, no aprovechamos la velocidad del procesador.
- RAM estática (SRAM)
  - Alto consumo relativo.
  - Capacidad de almacenamiento comparativamente baja.
  - Costo por bit alto.
  - Tiempo de acceso bajo (es mas rápida).
  - Si construímos el banco de memoria utilizando RAM estática, el costo y el consumo de la computadora son altos.

La solución: Memoria cache

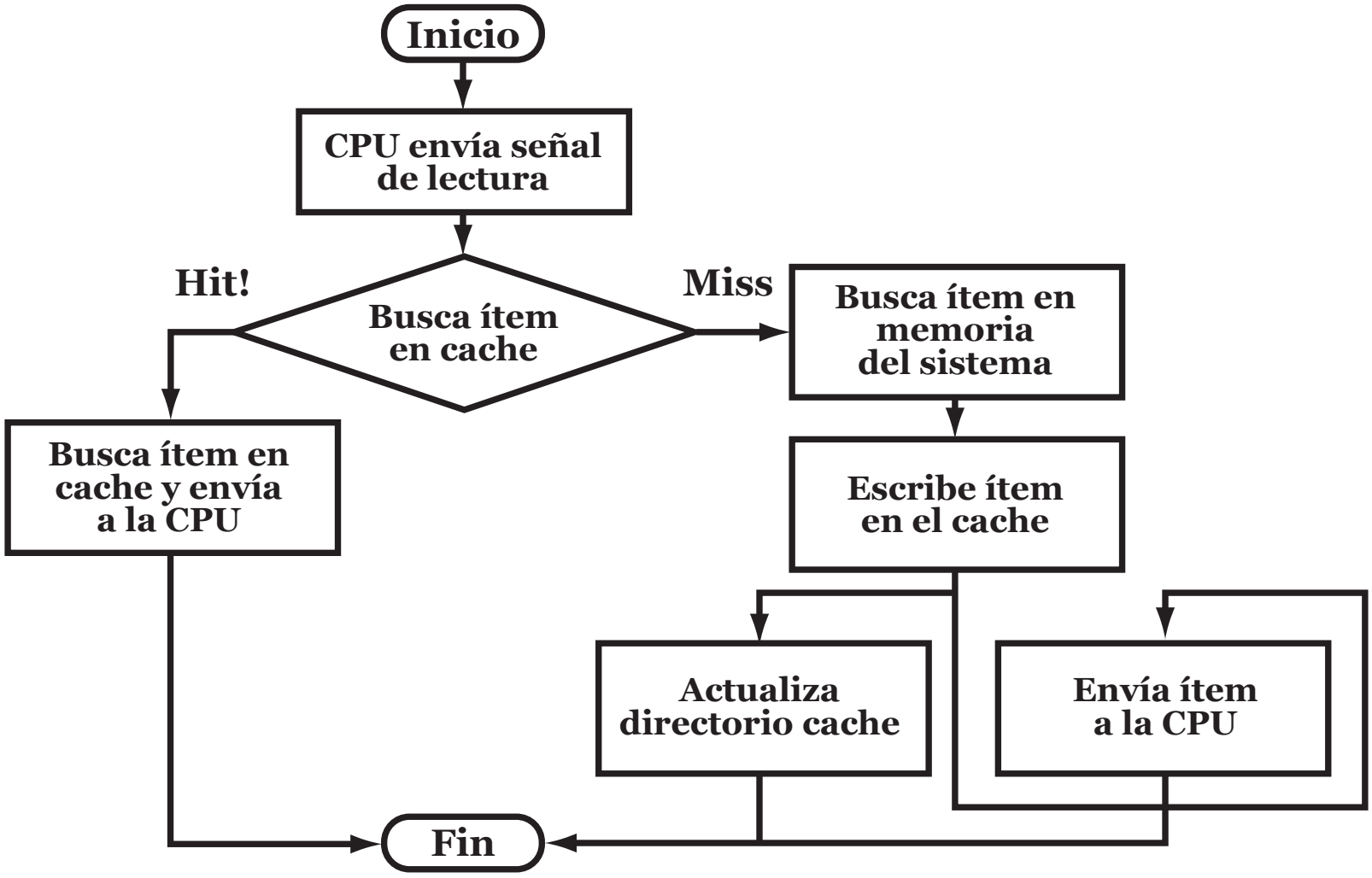
- Se trata de un banco de SRAM de muy alta velocidad, que contiene una copia de los datos e instrucciones que están en memoria principal.
- El arte consiste en que esta copia esté disponible justo cuando el procesador la necesita permitiéndole acceder a esos ítems sin recurrir a *wait states*.
- Combinada con una gran cantidad de memoria DRAM, para almacenar el resto de códigos y datos, resuelve el problema mediante una solución de compromiso típica.
- Requiere de hardware adicional que asegure que este pequeño banco de memoria cache contenga los datos e instrucciones más frecuentemente utilizados por el procesador.

Referencias

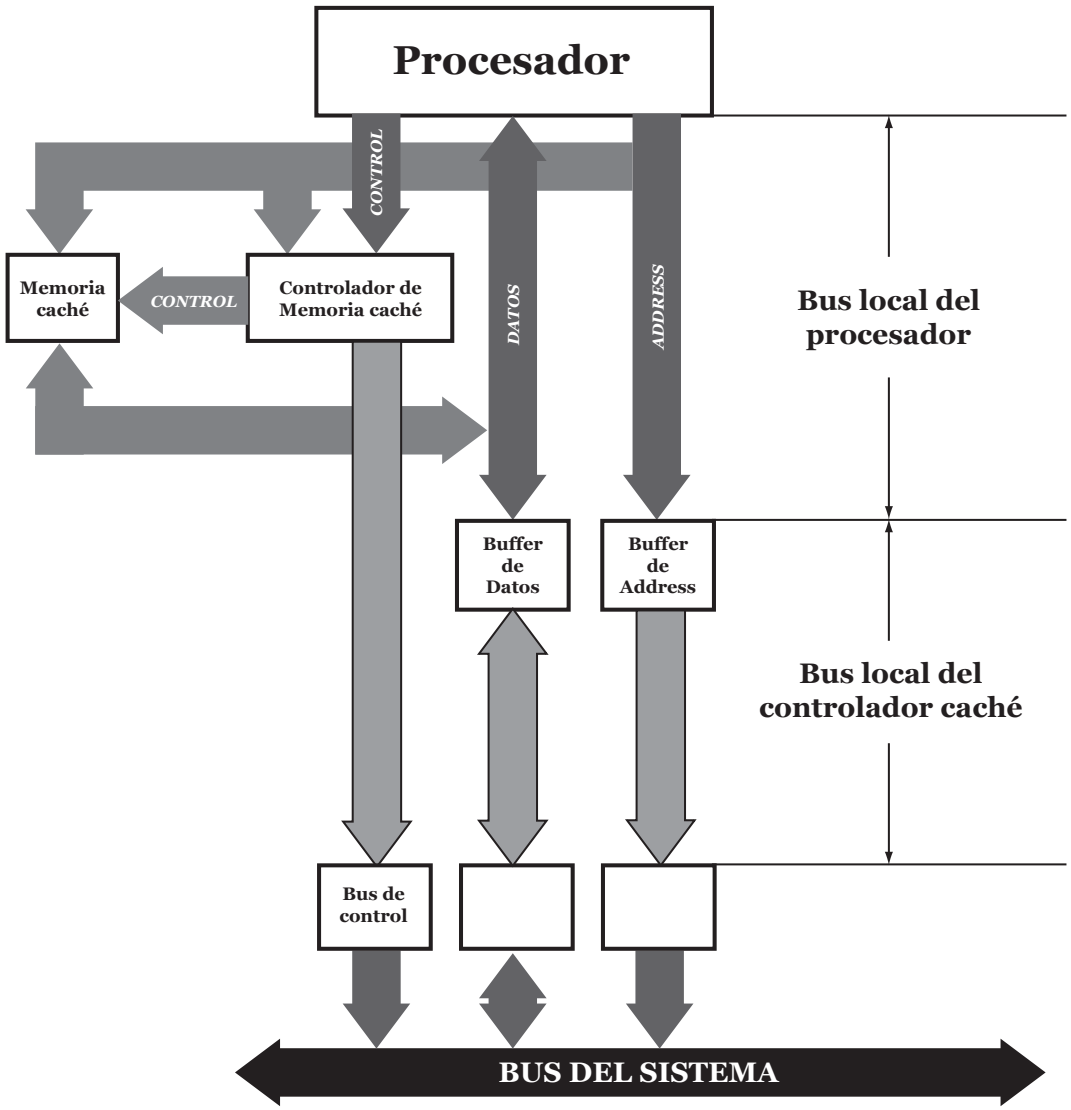
- El tamaño del banco de memoria cache debe ser:
  - Suficientemente grande para que el procesador resuelva la mayor cantidad posible de búsquedas de código y datos en esta memoria asegurando una alta performance.
  - Suficientemente pequeña para no afectar el consumo ni el costo del sistema.
- Se dice que se logra un ***hit*** cuando se accede a un ítem (dato o código) y éste se encuentra en la memoria cache.
- En caso contrario, se dice que el resultado del acceso es un ***miss***.
- Se espera un ***hit rate*** lo más alto posible

$$hit\ rate = \frac{Cantidad\ de\ accesos\ con\ presencia\ en\ Memoria\ Cache}{Cantidad\ total\ de\ accesos\ a\ memoria}$$

Operación de Lectura de memoria



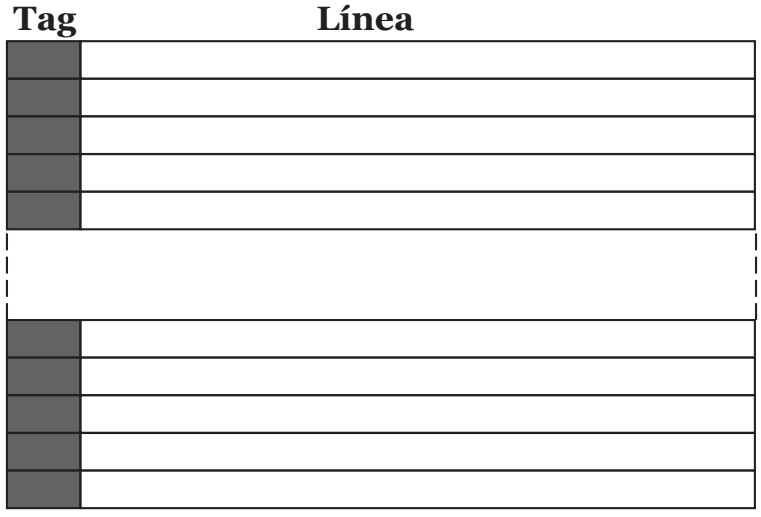
Estructura de Bus del sistema con cache

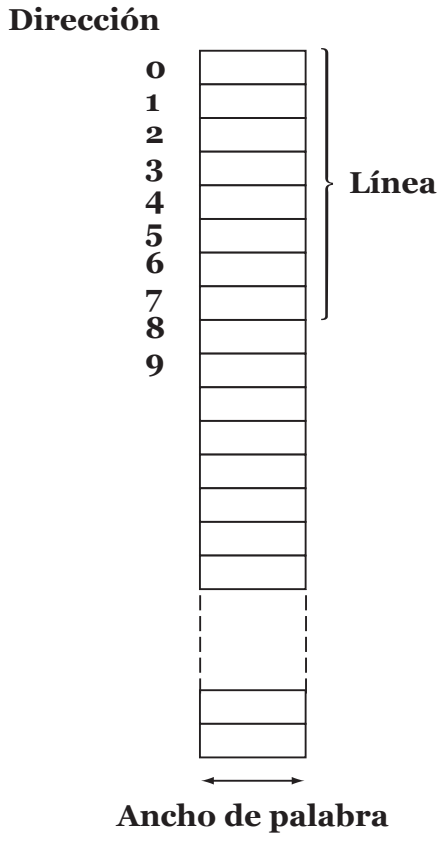


Cómo trabaja el controlador cache

- El controlador cache trabaja mediante dos principios que surgen de analizar el comportamiento de los algoritmos de software que se emplean habitualmente.
  - **Principio de vecindad temporal:** Si un ítem es referenciado, la probabilidad de ser referenciado en el futuro inmediato es alta.
  - **Principio de vecindad espacial:** Si un ítem es referenciado, es altamente probable que se referencie a los ítems vecinos a éste.
  - Ejemplo: Algoritmo de convolución
  - **i**, **j**, **suma**, se utilizan a menudo. Por lo tanto si se mantienen en el cache, el tiempo de acceso a estas variables por parte del procesador es óptimo.

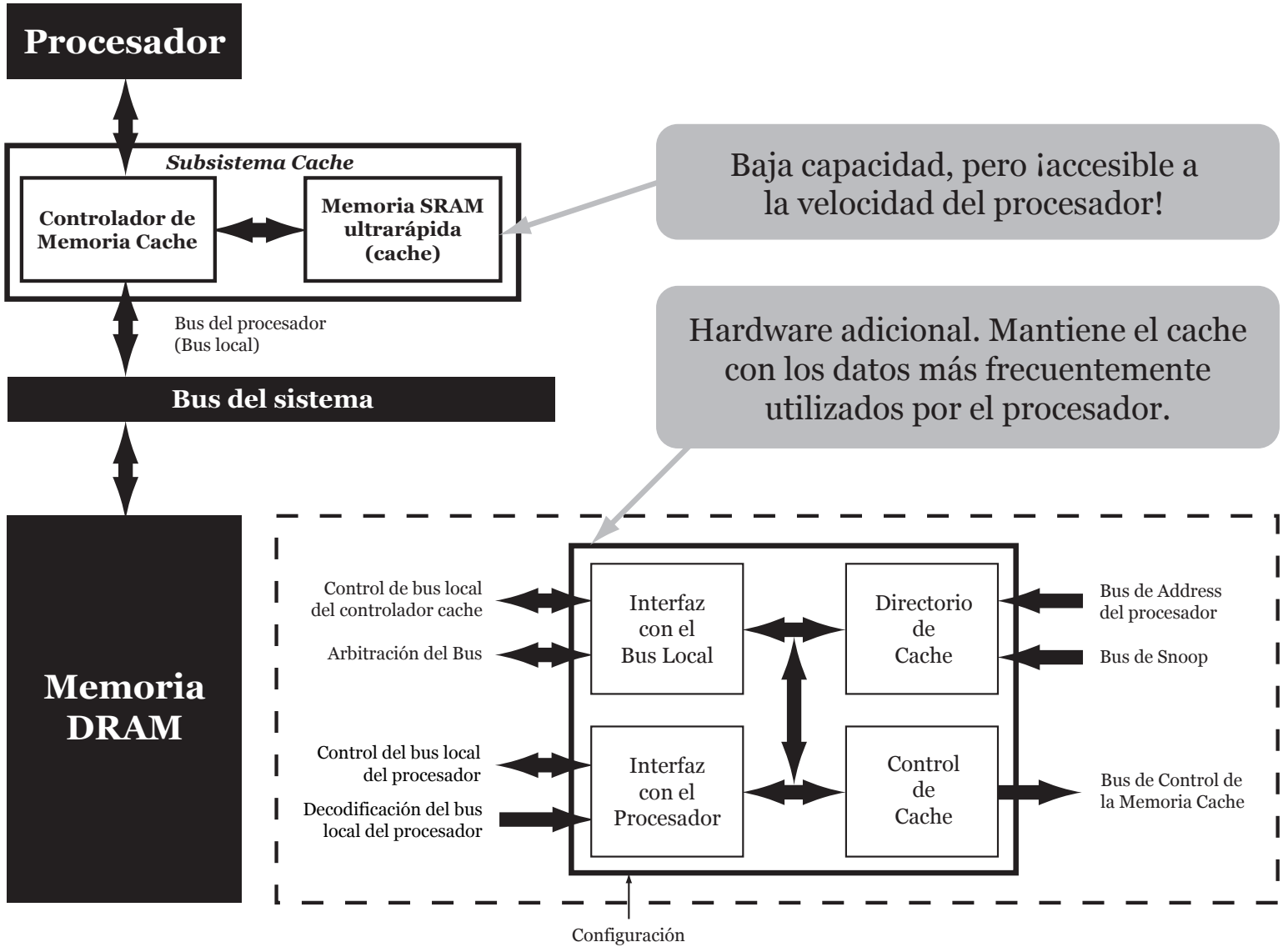
Estructura de memoria cache



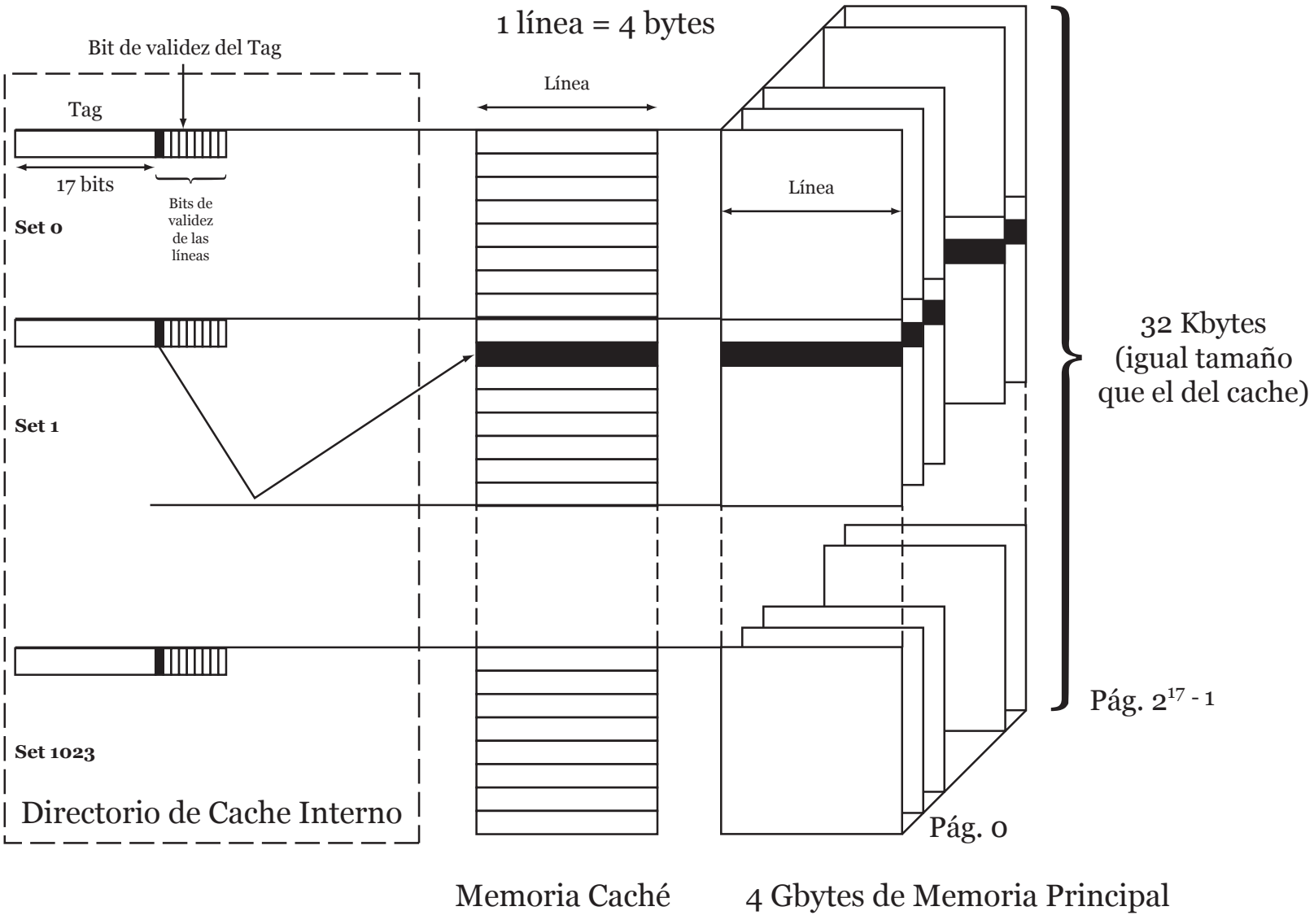


Línea: Elemento mínimo de palabra de datos dentro del cache. Corresponde a un múltiplo del tamaño de la palabra de datos de memoria. Razón: Cuando se direcciona un ítem en memoria generalmente se requerirá de los ítems que lo rodean (Principio de vecindad espacial)

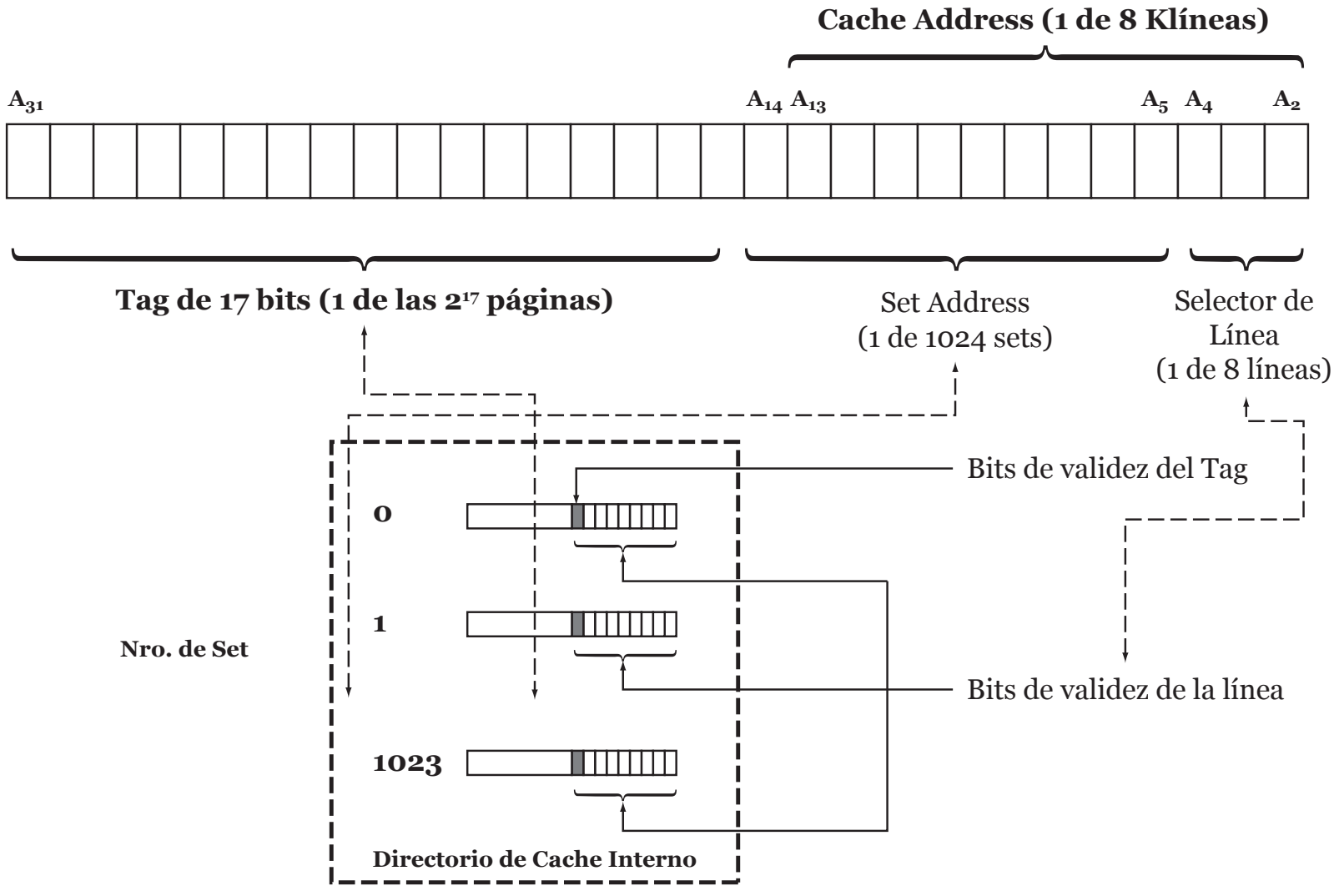
Memoria Cache



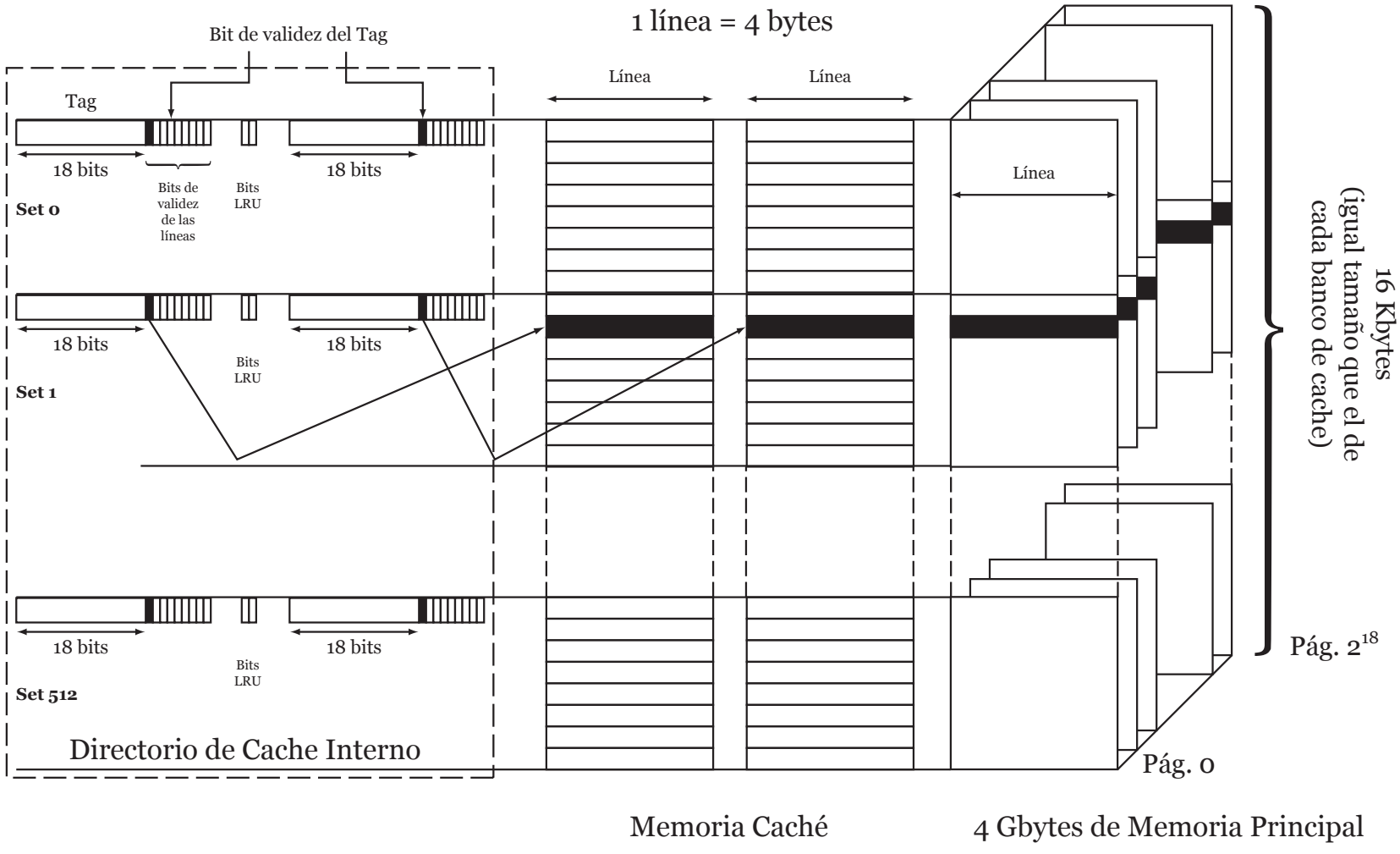
Organización del cache: Mapeo Directo



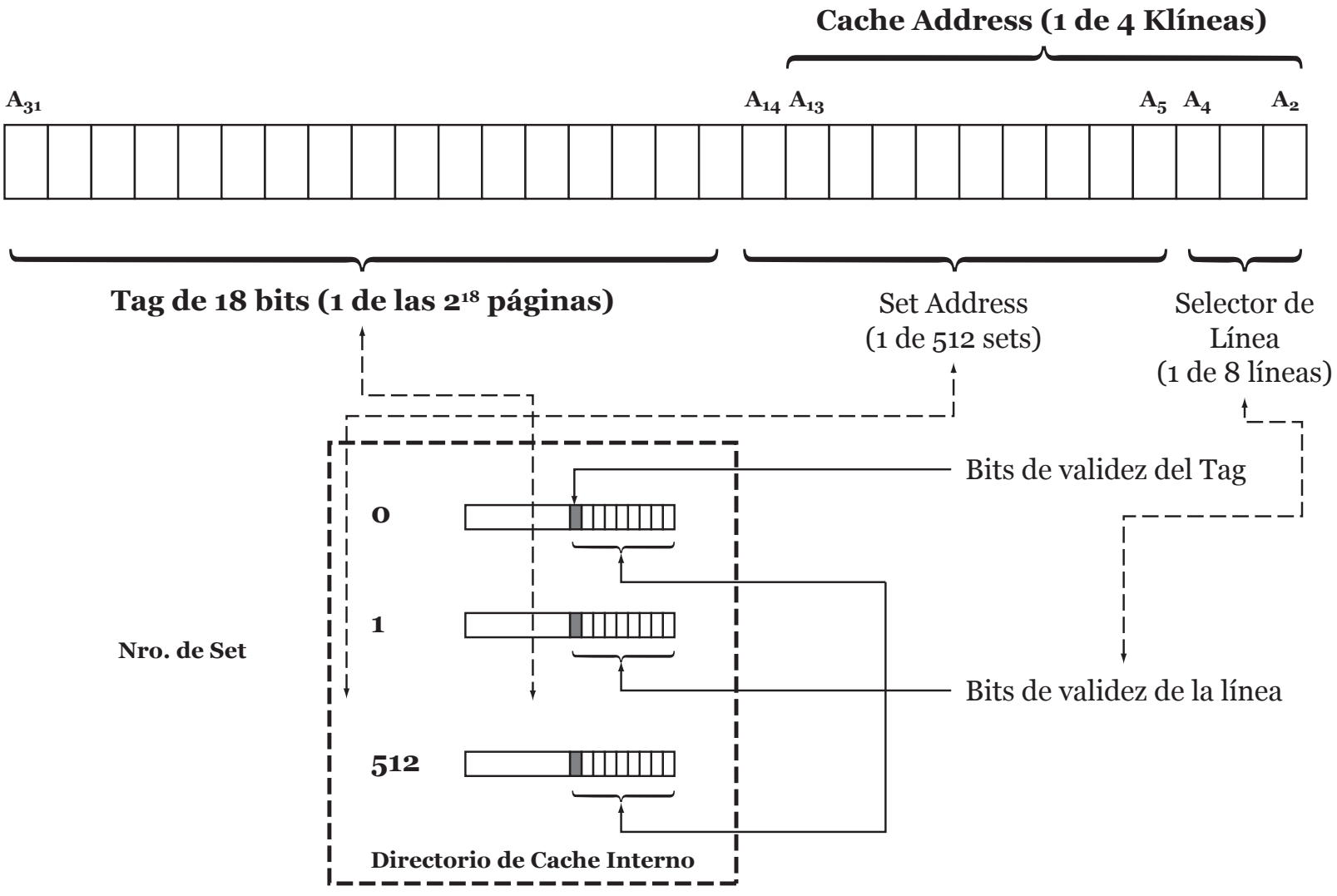
Organización del cache de mapeo directo



Organización del cache: Asociativo de dos vías



Organización del cache asociativo de dos vías

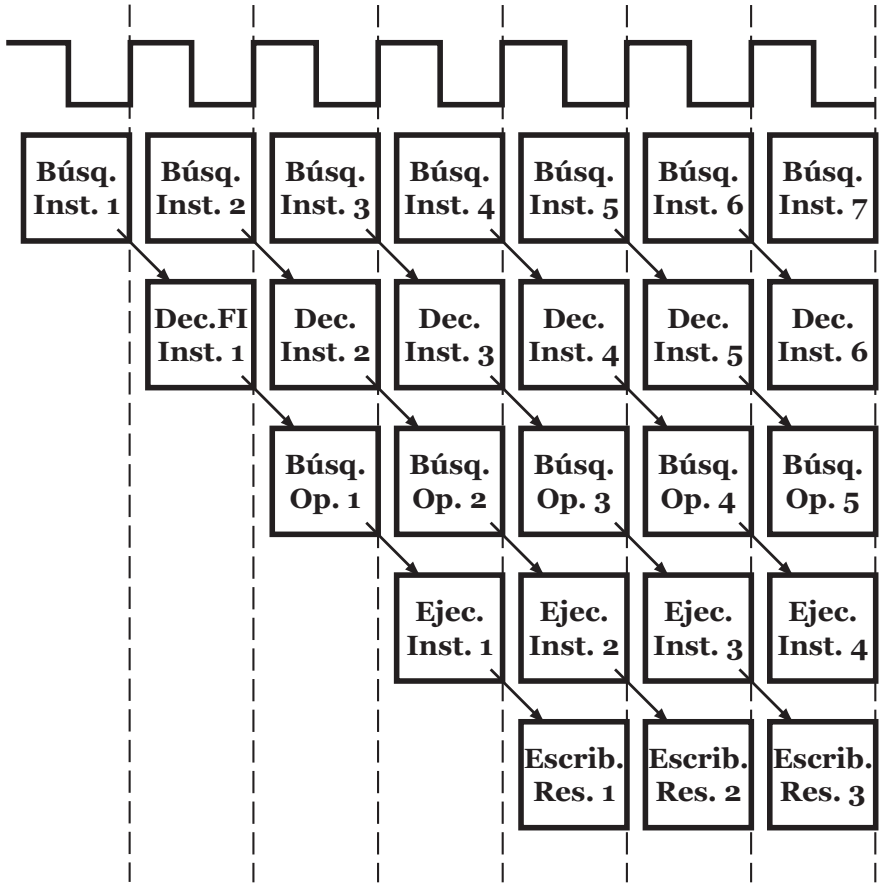


Manejo del contenido

- Algoritmos de reemplazo del contenido de la memoria cache:
  - LRU: Least Recently Used: Se corresponde con el principio de vecindad temporal.
  - LFU: Least Frecuently Used.
  - Random.
  - FIFO.

Cache miss: Impacto en el Pipeline de instrucciones

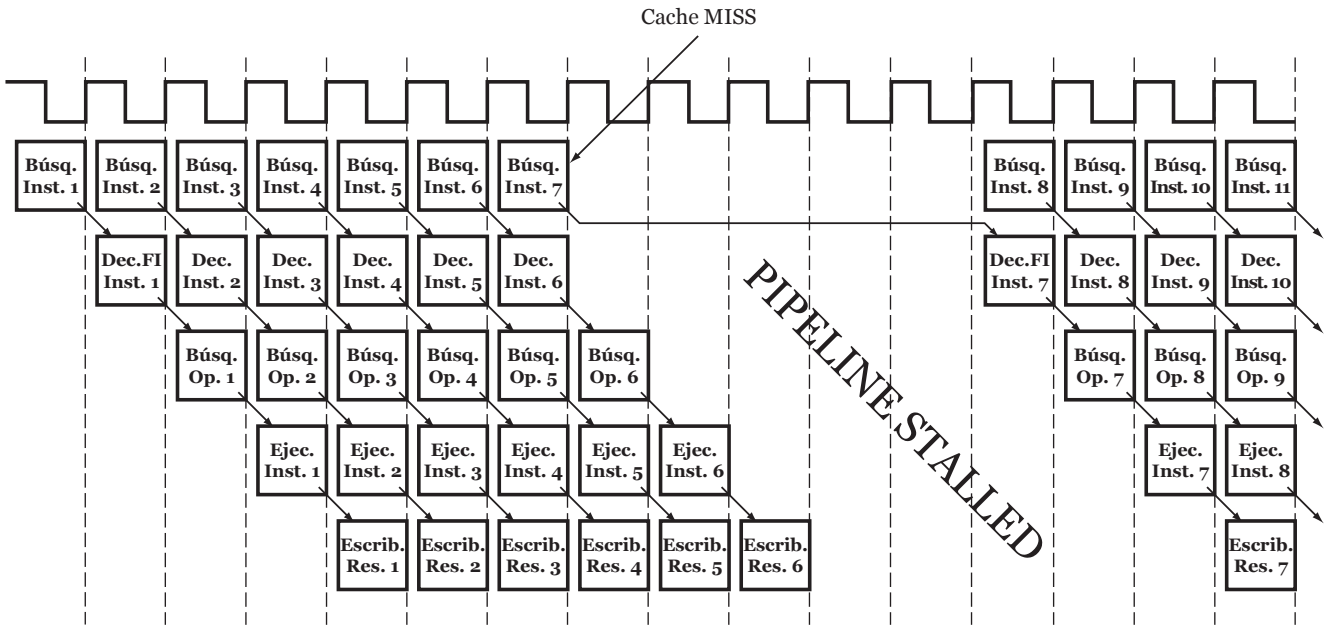
- Pipeline: permite superponer en el tiempo la ejecución de varias instrucciones a la vez.
- No requiere hardware adicional. Sólo se necesita lograr que todas las partes del procesador trabajen a la vez.
- Trabaja con el concepto de una línea de montaje:
  - Cada operación se descompone en partes.
  - Se ejecutan en un mismo momento diferentes partes de diferentes operaciones.
  - Cada parte se denomina etapa (stage).



RESULTADO: UNA VEZ EN REGIMEN EJECUTA A RAZON DE UNA INSTRUCCION POR CICLO DE CLOCK.

Cache miss: Impacto en el Pipeline de instrucciones

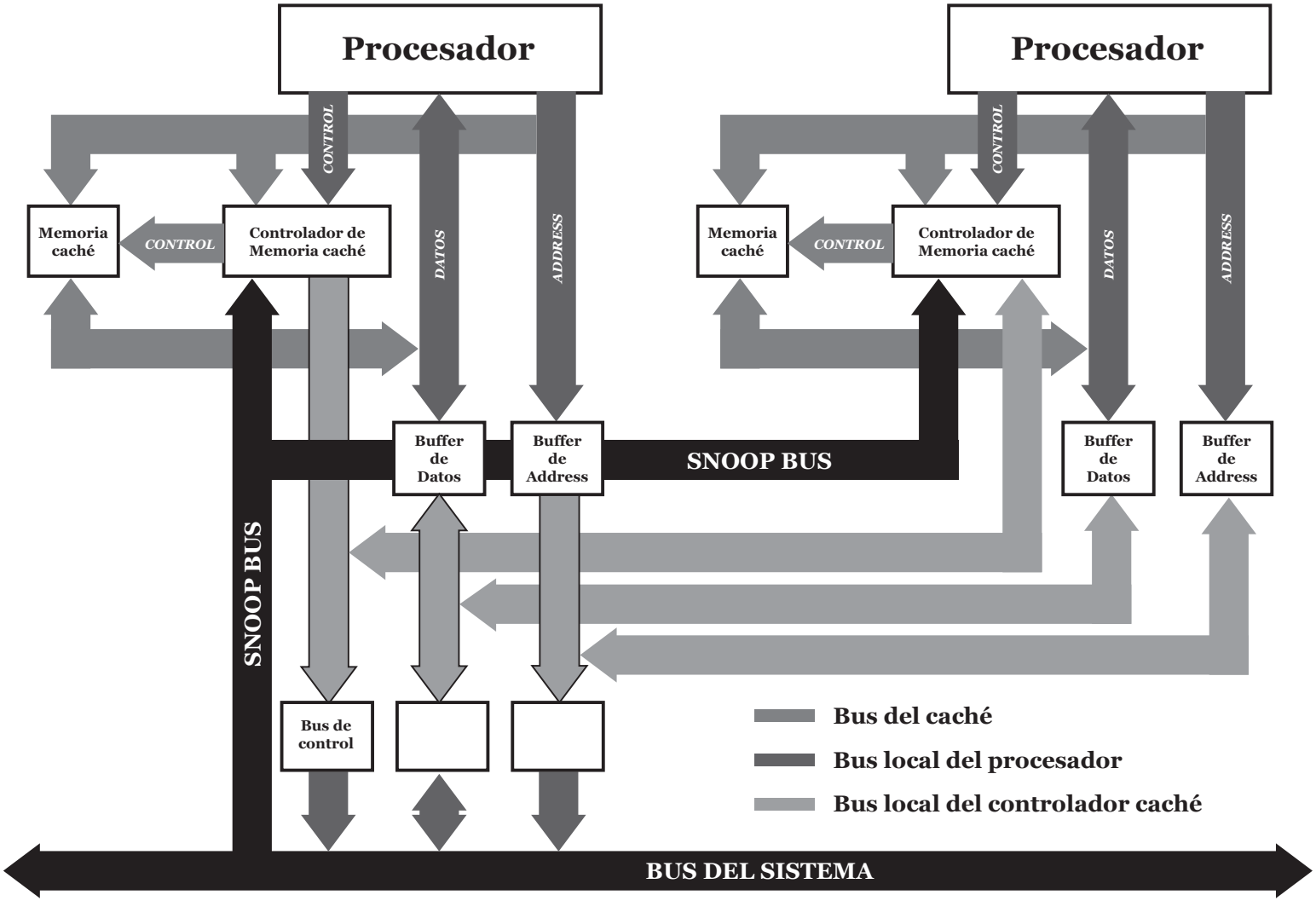
- Si la búsqueda de una instrucción o de un operando en el cache falla, entonces el procesador debe recurrir a la memoria principal.
- La demora en el acceso hace que el pipeline se atasque (stall).
- Una vez recuperado el dato de memoria principal se requieren (en este ejemplo), 5 ciclos de reloj adicionales para recuperar el ritmo de operación del pipeline!



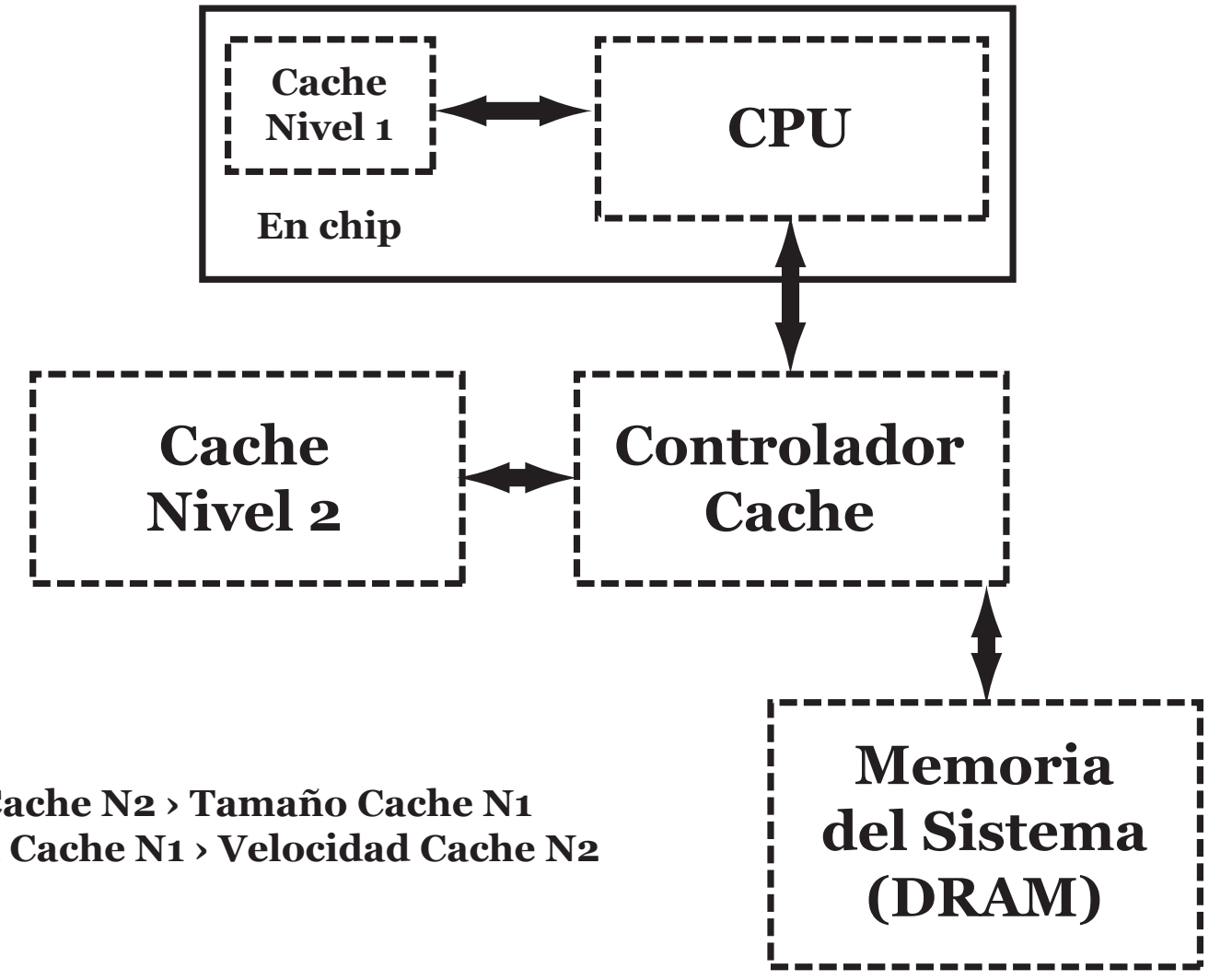
Coherencia de un cache

- Una variable que está en el caché también está alojada en alguna dirección de la DRAM.
- Ambos valores deben ser iguales.
- Cuando el procesador la modifica hay varios modos de actuar:
  - Write through: el procesador escribe en la DRAM y el controlador cache refresca el cache con el dato actualizado.
  - Write through buffered: el procesador actualiza la SRAM cache, y el controlador cache luego actualiza la copia en memoria DRAM mientras el procesador continúa ejecutando instrucciones y usando datos de la memoria cache.
  - Copy back: Se marcan las líneas de la memoria cache cuando el procesador escribe en ellas. Luego en el momento de eliminar esa línea del cache el controlador cache deberá actualizar la copia de DRAM.
- Si el procesador realiza un miss mientras el controlador cache está accediendo a la DRAM para actualizar el valor, deberá esperar hasta que controlador cache termine la actualización para recibir desde éste la habilitación de las líneas de control para acceder a la DRAM.

Estructura de Bus del sistema Multiprocesador con cache



Multilevel cache



Tamaño Cache N2 › Tamaño Cache N1
Velocidad Cache N1 › Velocidad Cache N2

Implementaciones prácticas de memoria cache (1)

- Intel 80486:
  - 8 Kbytes de cache L1 on chip.
  - Tamaño de línea: 16 bytes.
  - Organización asociativa de 4-vías.
- Pentium:
  - Dos caches on-chip, uno para datos y otro para instrucciones.
  - Tamaño de cada cache: 8 Kbytes.
  - Tamaño de línea: 32 bytes.
  - Organización asociativa de 4-vías.
- PowerPC 601:
  - Cache on-chip de 32 Kbytes.
  - Tamaño de línea: 32 bytes.
  - Organización asociativa de 8-vías.

Implementaciones prácticas de memoria cache (2)

- PowerPC 603:
  - Dos caches on-chip, una para datos y otra para instrucciones.
  - Tamaño de cada cache: 8 Kbytes.
  - Tamaño de línea: 32 bytes.
  - Organización asociativa de 2-vías (organización del cache más simple que en el 601 pero un procesador mas fuerte).

- PowerPC 604:
  - Dos caches on-chip, uno para datos y otro para instrucciones.
  - Tamaño de cada cache: 16 Kbytes.
  - Tamaño de línea: 32 bytes.
  - Organización asociativa de 4-vías.

- PowerPC 620:
  - Dos caches on-chip, uno para datos y otro para instrucciones.
  - Tamaño de cada cache: 32 bytes.
  - Tamaño de línea: 64 bytes.
  - Organización asociativa de 8-vías.