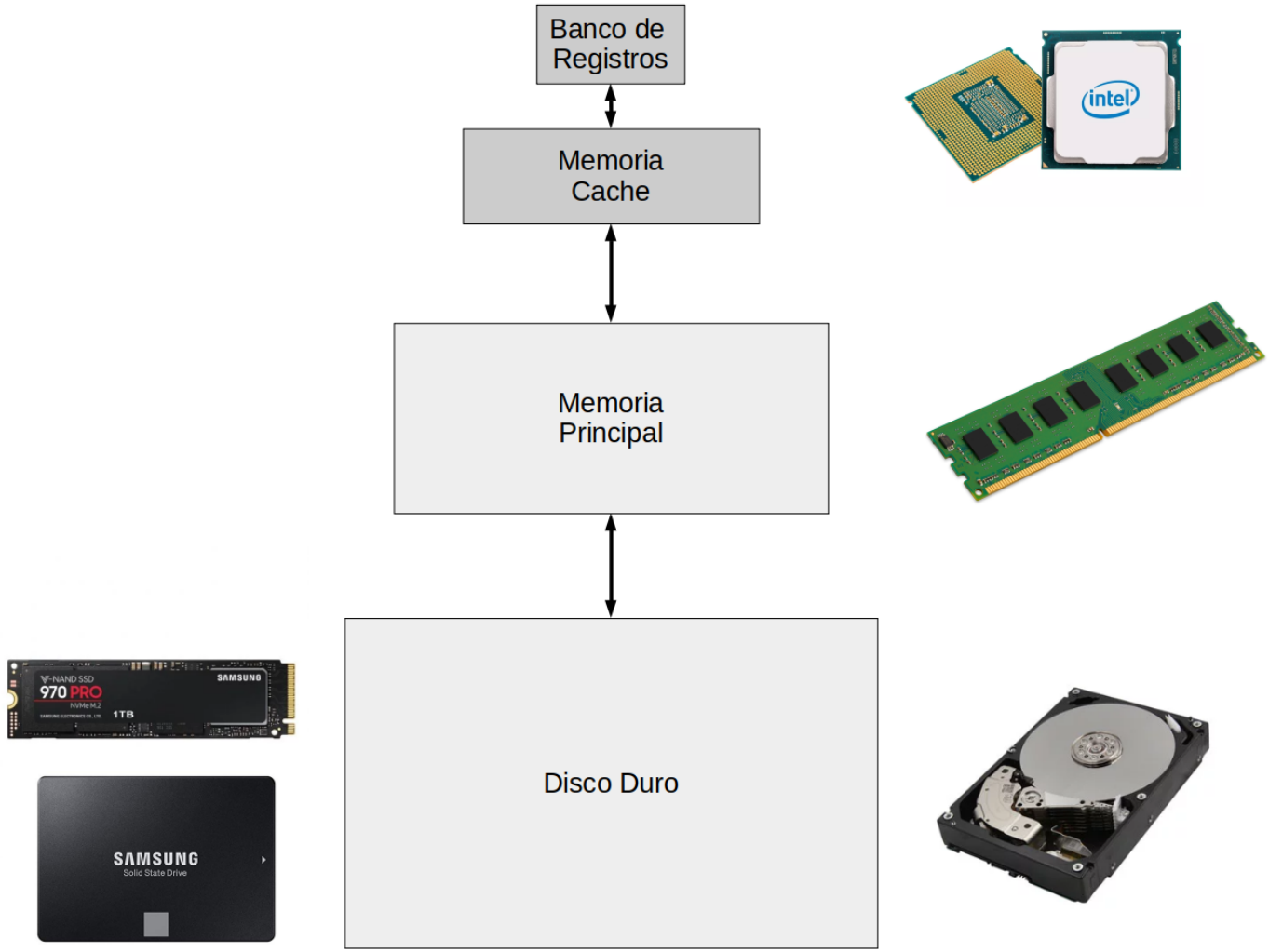
Estructura de Computadores

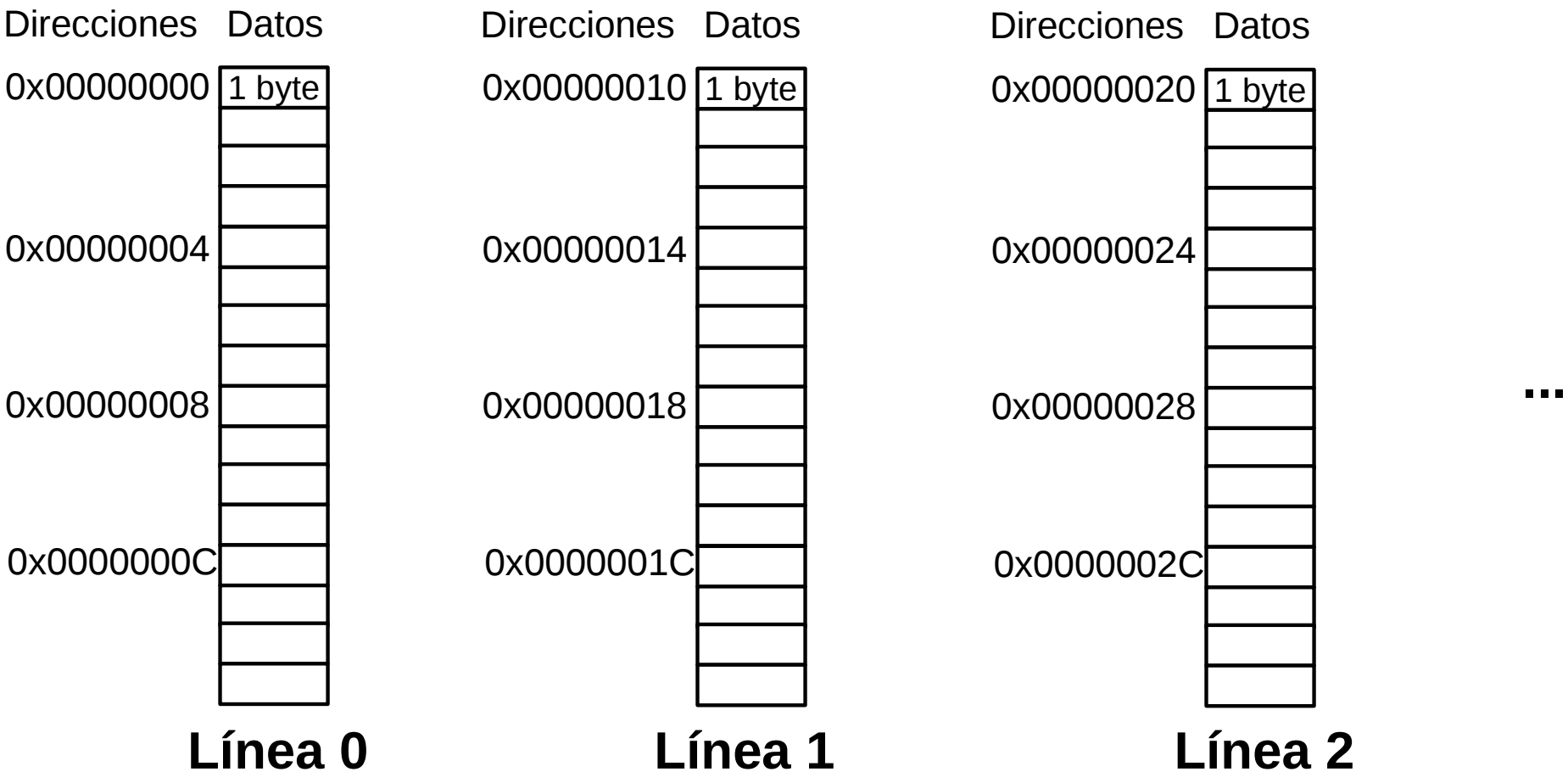
Tema 6. Memoria Cache

Jerarquía de Memoria



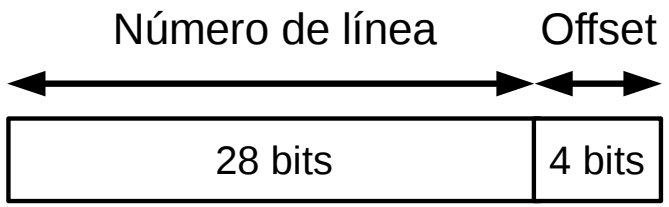
Líneas de memoria

- El espacio de direcciones se divide en líneas de N bytes
  - Ejemplo: líneas (bloques) de memoria de 16 bytes



Líneas de memoria

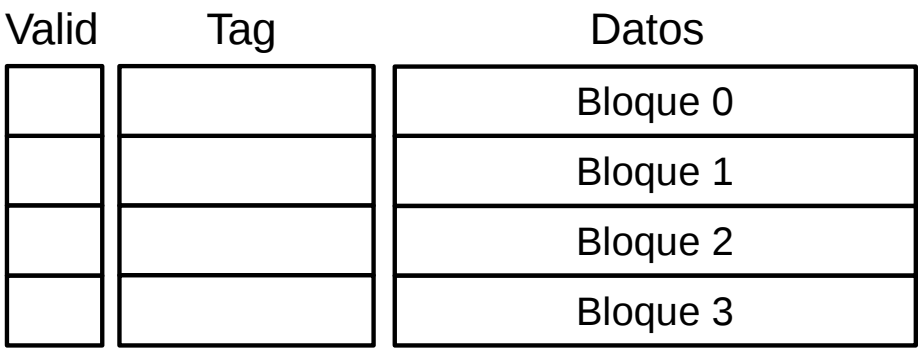
- Una dirección de memoria se divide en dos partes:
  - Número de línea
  - Offset: byte accedido dentro de la línea
- Ejemplo para tamaño de línea de 16 bytes



- Dirección de memoria 0x10010020
  - Número de línea = 0x1001002
  - Offset = 0x0

Memoria cache

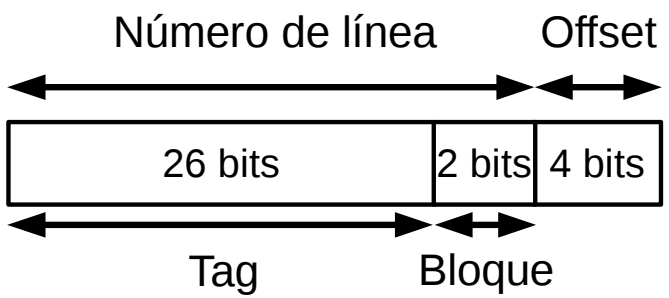
- Para cada bloque de cache se almacena:
  - **Valid bit**: Indica si el bloque de cache contiene datos válidos
  - **Tag**: Indica la línea de memoria que se almacena
  - **Datos**: Contiene los bytes de la línea de memoria



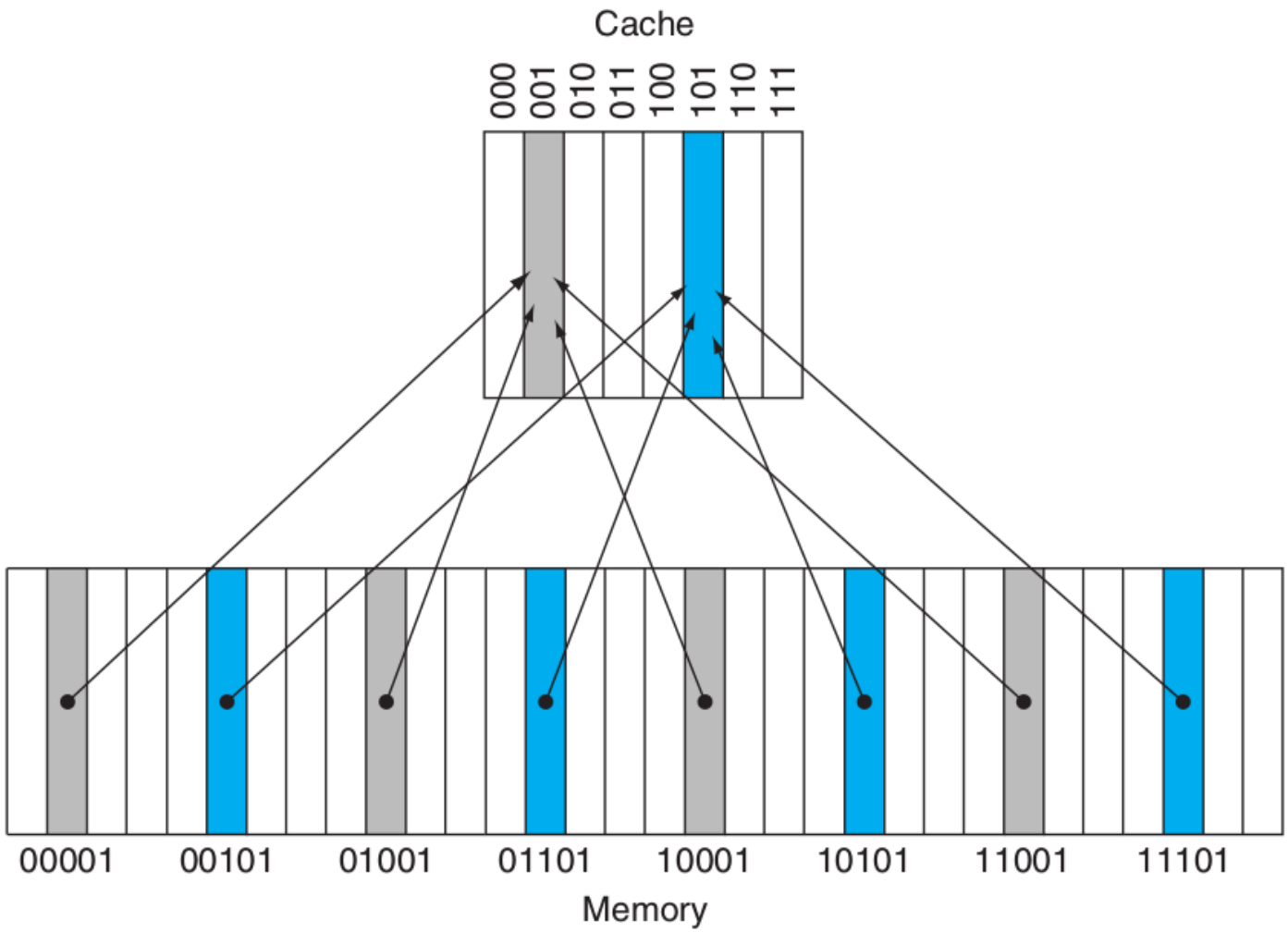
- Dada una línea de memoria, ¿cómo sabemos en qué bloque de la memoria cache se almacena?

Memoria cache

- Correspondencia directa
  - Cada línea de memoria se mapea a un bloque fijo de la memoria cache
  - (Número de línea) modulo (Número de bloques en la cache)
  - Los bits de menor peso del número de línea indican el bloque de cache donde se almacena la línea
- Ejemplo
  - Tamaño de línea de memoria de 16 bytes
  - Memoria cache con 4 bloques



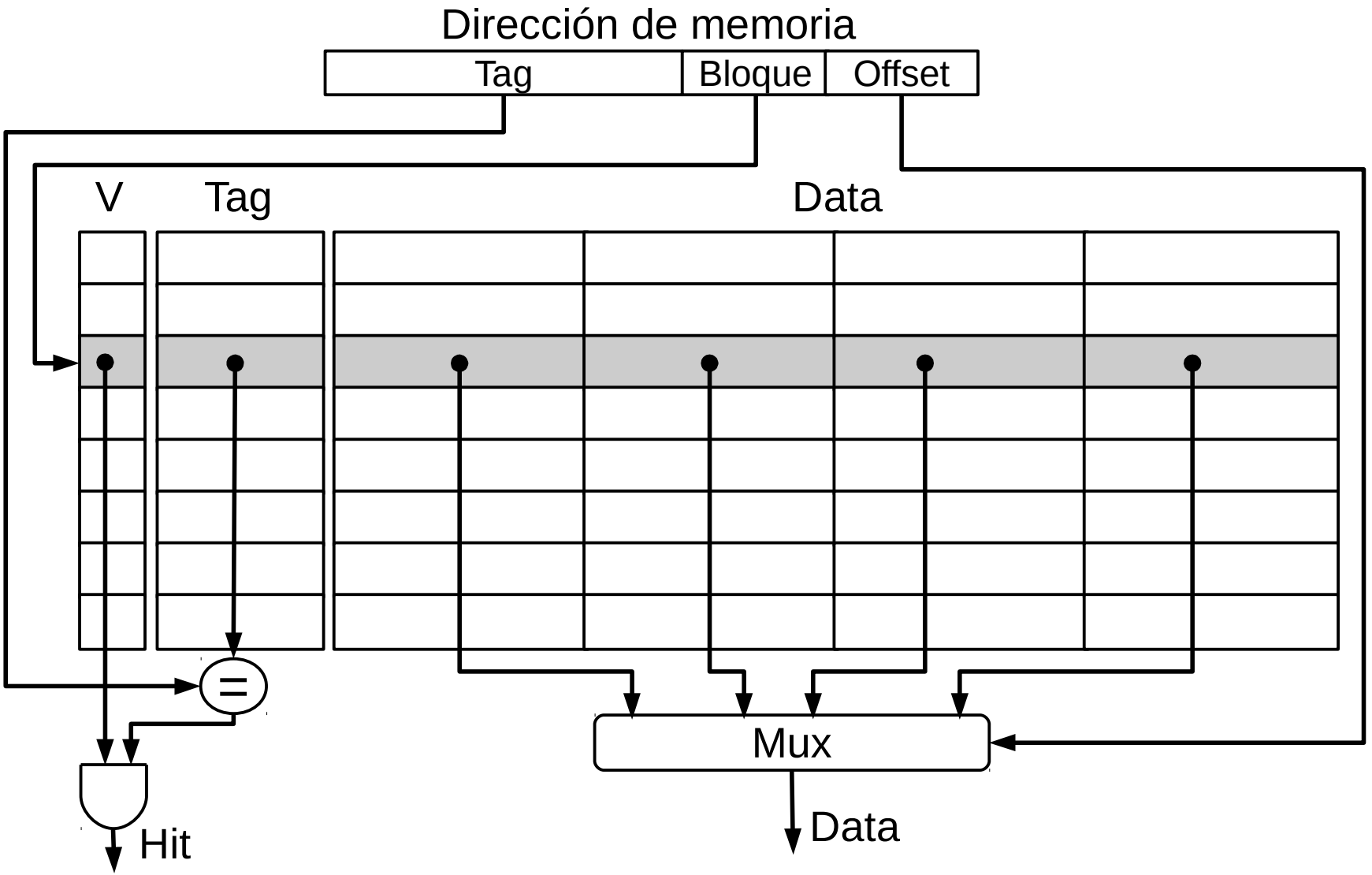
Correspondencia directa



Correspondencia directa

- Suponiendo un tamaño de línea de 64 bytes y una memoria cache con 32 bloques, indica el tag, el número de bloque y el offset para cada una de las siguientes direcciones de memoria:
  - 0x100101C0
  - 0x1001060F

Memoria cache

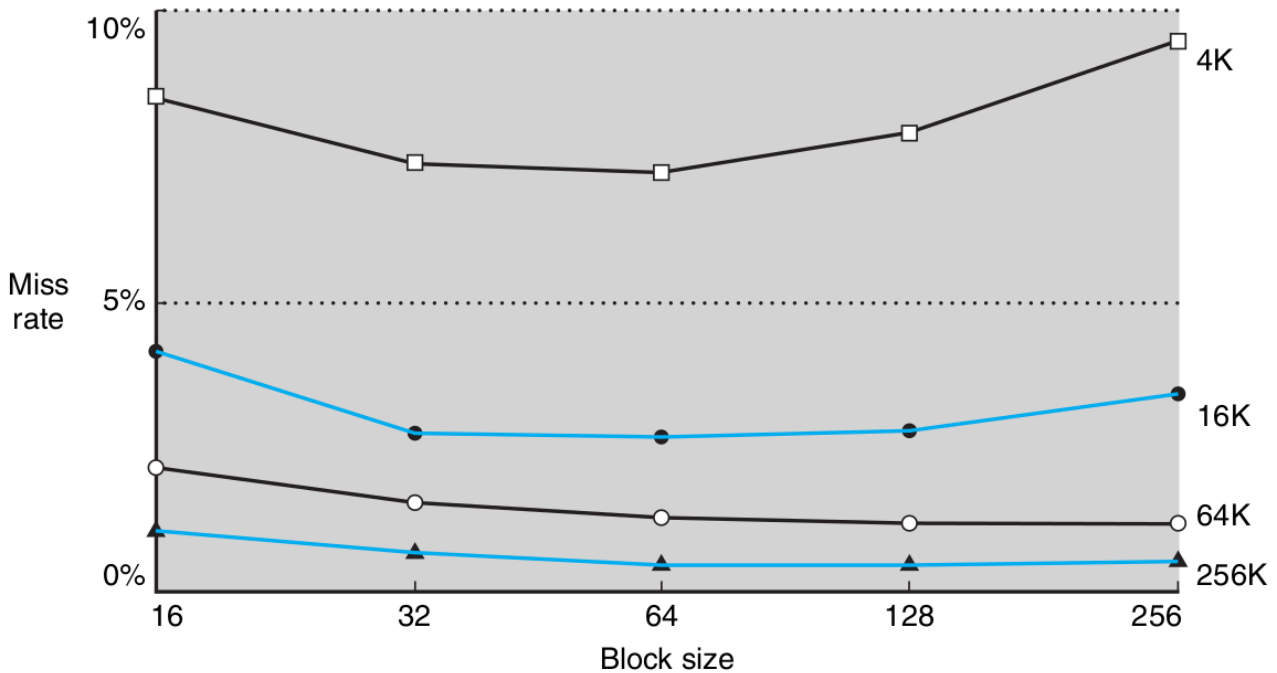


Tamaño de línea de memoria

- ¿Cuál es el tamaño de línea de memoria óptimo?
  - Tamaños más grandes permiten aprovechar mejor la localidad espacial
  - Tamaños más pequeños permiten tener más líneas en la cache

Tamaño de línea de memoria

- ¿Cuál es el tamaño de línea de memoria óptimo?
  - Tamaños más grandes permiten aprovechar mejor la localidad espacial
  - Tamaños más pequeños permiten tener más líneas en la cache



Políticas de escritura

- MC: Memoria Cache, MP: Memoria Principal
- **Escritura inmediata con asignación**
  - En caso de fallo en una escritura, se trae el bloque de MP a MC. La escritura se realiza tanto en MC como en MP. (MARS)

Políticas de escritura

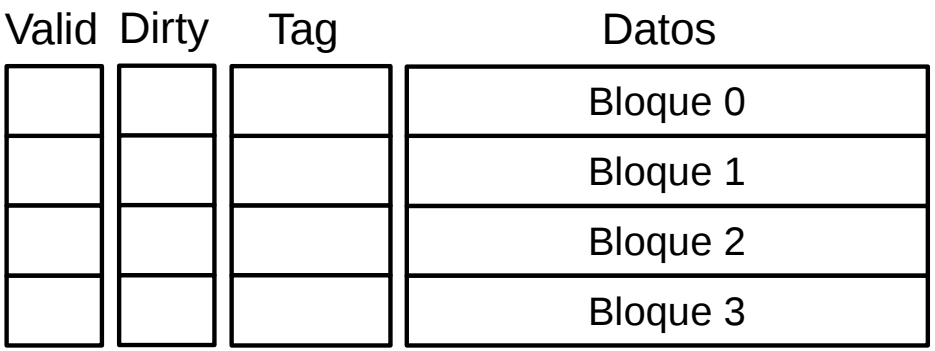
- MC: Memoria Cache, MP: Memoria Principal
- **Escritura inmediata con asignación**
  - En caso de fallo en una escritura, se trae el bloque de MP a MC. La escritura se realiza tanto en MC como en MP. (MARS)
- **Escritura inmediata sin asignación**
  - En caso de fallo en una escritura, no se trae el bloque a MC. La escritura se realiza únicamente en MP. Si la escritura es un acierto, se escribe en MC y MP.

Políticas de escritura

- MC: Memoria Cache, MP: Memoria Principal
- **Escritura inmediata con asignación**
  - En caso de fallo en una escritura, se trae el bloque de MP a MC. La escritura se realiza tanto en MC como en MP. (MARS)
- **Escritura inmediata sin asignación**
  - En caso de fallo en una escritura, no se trae el bloque a MC. La escritura se realiza únicamente en MP. Si la escritura es un acierto, se escribe en MC y MP.
- **Escritura retardada con asignación**
  - En caso de fallo en una escritura, se trae el bloque a MC. La escritura se realiza únicamente en MC. Cuando el bloque se reemplaza, se actualiza la MP.

Escritura retardada con asignación

- Dirty bit
  - Indica si el bloque ha sido modificado (escrito)
- Al reemplazar una línea, si el bit dirty está a 1 hay que copiar la línea a memoria principal



Escritura retardada con asignación

- Si la escritura es con asignación, hay que **leer** la línea de memoria principal para almacenarla en la cache
- Cuando se realiza un reemplazo de una línea modificada, hay que escribir el bloque en memoria prinicipal antes de hacer el reemplazo
  - Hay que **escribir** en memoria principal, incluso si la operación que provoca el reemplazo es una lectura

Escritura retardada con asignación

- En caso de reemplazo de una línea modificada (bit dirty = 1)
  1. Leer la línea reemplazada de la memoria cache
  2. Escribir la línea reemplazada en memoria principal
  3. Leer la nueva línea de memoria principal
  4. Escribir la nueva línea en memoria cache
  5. Reiniciar el acceso (ahora será un hit)

Escritura inmediata

- Con escritura inmediata, todos los accessos (de escritura) requiren acceder a memoria principal
  - La memoria principal tiene una latencia elevada comparada con la memoria cache
  - Se pierde el beneficio en rendimiento que aporta la memoria cache

Escritura inmediata

- Con escritura inmediata, todos los accessos (de escritura) requiren acceder a memoria principal
  - La memoria principal tiene una latencia elevada comparada con la memoria cache
  - Se pierde el beneficio en rendimiento que aporta la memoria cache
- Buffer de escritura
  - Buffer que almacena los datos que están pendientes de escribir a memoria principal
  - El procesador no necesita esperar a que se escriban los datos en memoria principal
  - En cuanto se han escrito los datos en la memoria cache y en el buffer de escritura, el procesador puede continuar la ejecución

Problema

Disponemos de un procesador de 16 bits (direcciones de 16 bits). El tamaño de bloque de memoria de 16 bytes. La memoria cache es de correspondencia directa, con 256 bytes de capacidad y con una política de escritura inmediata sin asignación. Rellena la siguiente tabla.

tipo	dirección (hex)	etiqueta (hex)	índice MC (hex)	acierto/ fallo	#bytes leidos MP	#bytes escritos MP
R	4534					
R	4568					
W	13A4					
W	13A8					
R	3560					
W	453C					
W	60A0					
R	453C					
W	3900					
R	A238					

Problema (continuación)

Rellena la misma tabla, suponiendo que la memoria cache tiene una política de escritura retardada con asignación. Calcula para cada política: tasa de fallos, número de bytes leidos de MP y número de bytes escritos a MP.

tipo	dirección (hex)	etiqueta (hex)	índice MC (hex)	acierto/ fallo	#bytes leidos MP	#bytes escritos MP
R	4534					
R	4568					
W	13A4					
W	13A8					
R	3560					
W	453C					
W	60A0					
R	453C					
W	3900					
R	A238					