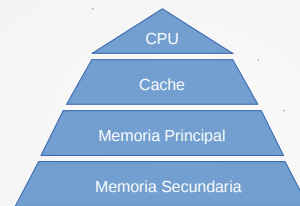


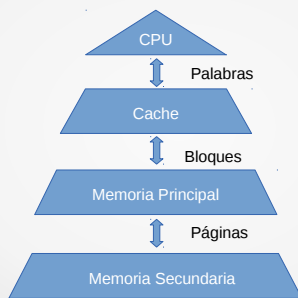
Arquitectura de Computadoras
Primer cuatrimestre de 2014
Memoria Cache

Introducción



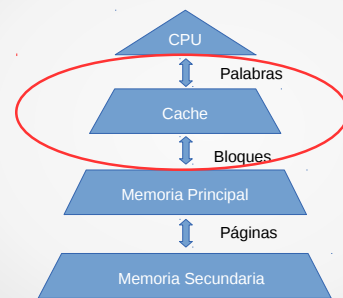
2

Introducción



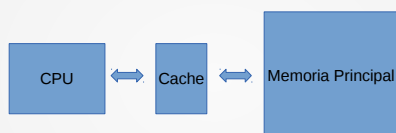
3

Introducción



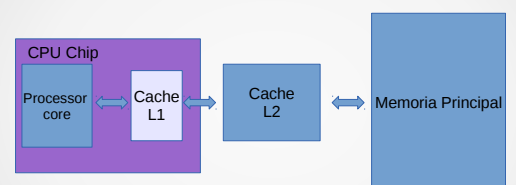
4

Introducción



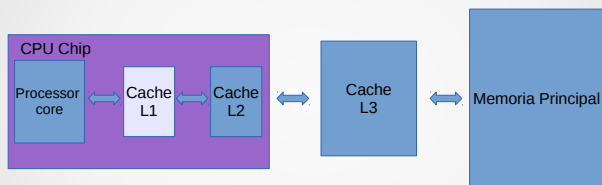
5

Introducción



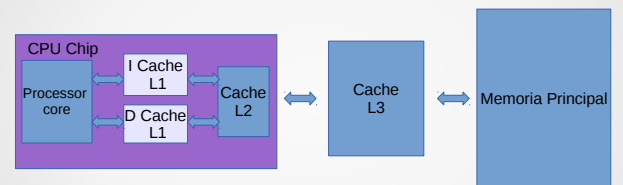
6

Introducción



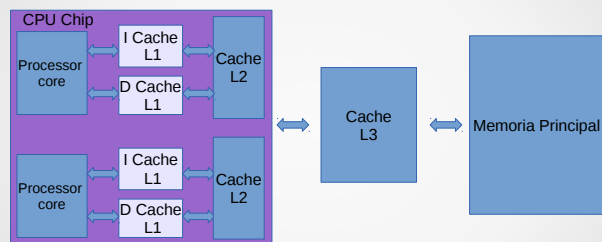
7

Introducción



8

Introducción

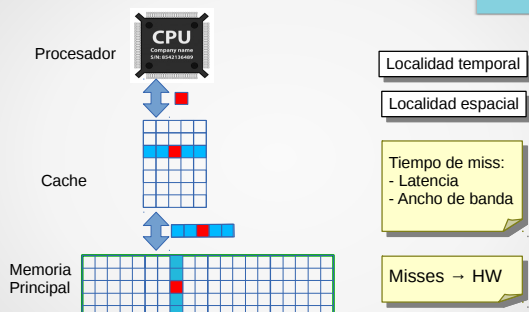


9

Memoria Cache Conceptos básicos

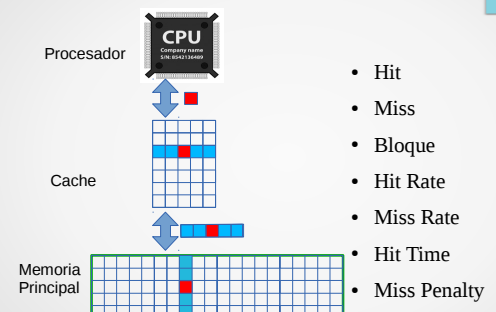
10

Conceptos básicos



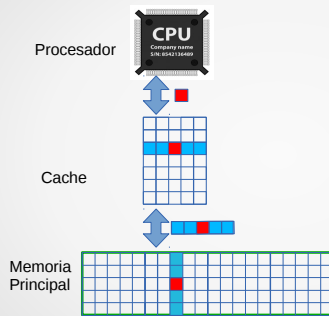
11

Conceptos básicos



12

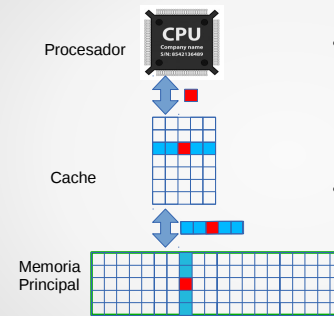
Conceptos básicos



- **Hit:** cuando *se encuentra* la información en el nivel más alto
- **Miss:** cuando la información *no se encuentra* en el nivel más alto
- **Bloque:** unidad de información transferida entre niveles

13

Conceptos básicos



- **Hit rate:** fracción de accesos a memoria que se encuentran en el nivel superior

$$\text{Hit Rate} = \frac{N^{\circ} \text{ hits}}{N^{\circ} \text{ acc a Mem}}$$

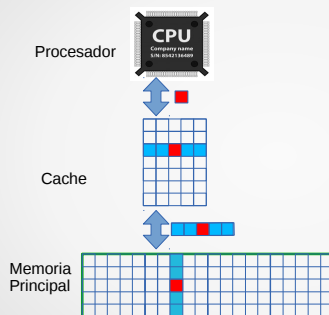
- **Miss rate:** fracción de accesos a memoria no encontrados en el nivel superior

$$\text{Miss Rate} = \frac{N^{\circ} \text{ misses}}{N^{\circ} \text{ acc a Mem}}$$

$$\text{Miss Rate} = (1 - \text{Hit rate})$$

14

Conceptos básicos



- **Hit time:** tiempo para acceder al nivel superior, incluyendo el tiempo para determinar si es un miss o un hit
- **Miss penalty:** tiempo de acceso al nivel inferior + tiempo de transferencia del bloque



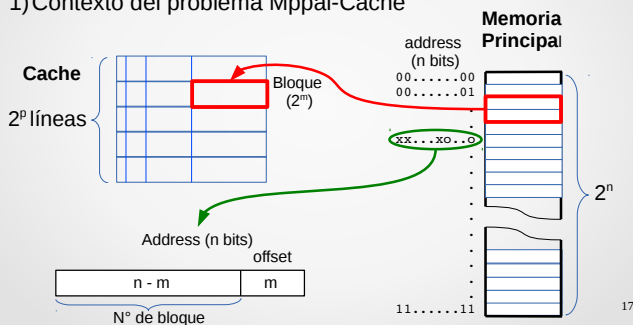
15

Memoria Cache Organización de Cache

16

Contexto del problema

1) Contexto del problema Mppal-Cache



17

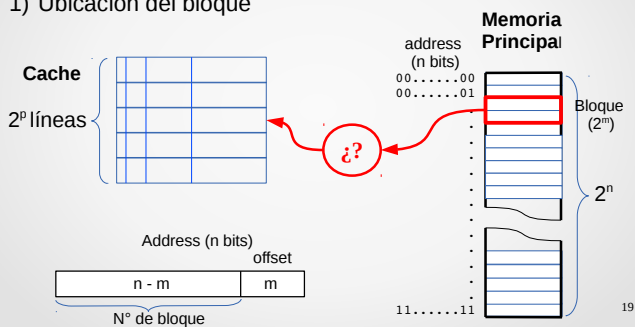
Cuatro cuestiones a resolver

- 1) Ubicación del bloque:
 - ¿Dónde colocar el bloque en el nivel superior?
- 2) Identificación del bloque:
 - ¿Cómo encontrar un bloque en el nivel superior si es que éste está en dicho nivel?
- 3) Reemplazo de bloques:
 - ¿Qué bloque debo reemplazar ante un miss?
- 4) Estrategia de escritura:
 - ¿Qué sucede ante una escritura?

18

Cuatro cuestiones a resolver

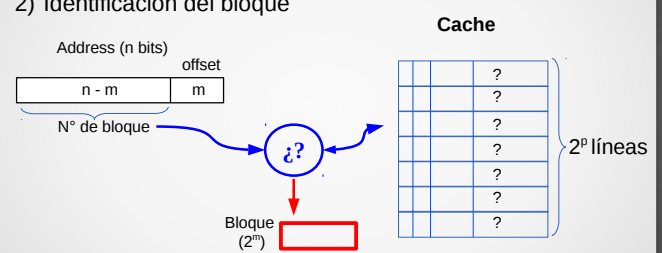
1) Ubicación del bloque



19

Cuatro cuestiones a resolver

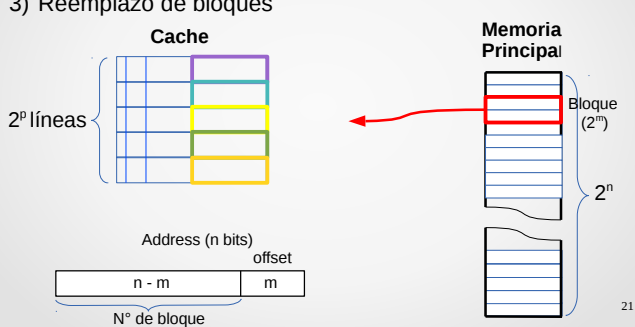
2) Identificación del bloque



20

Cuatro cuestiones a resolver

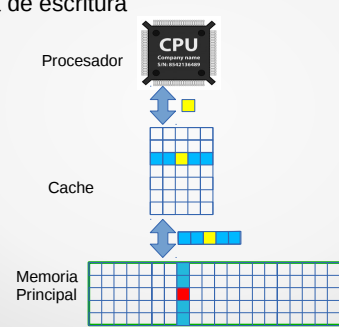
3) Reemplazo de bloques



21

Cuatro cuestiones a resolver

4) Estrategia de escritura



22

Cuatro cuestiones a resolver

1) Ubicación del bloque

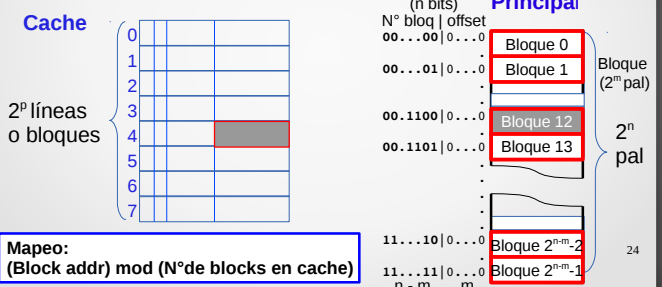
1.1) Mapeo Directo

1.2) Full asociativa

1.3) Set asociativa

1) Ubicación del bloque:

1.1) Mapeo Directo



24

23

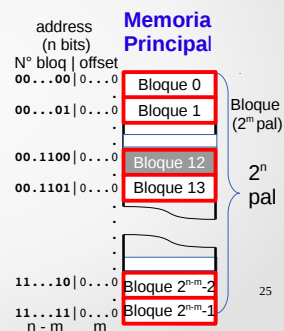
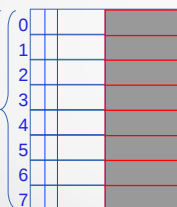
Cuatro cuestiones a resolver

1) Ubicación del bloque:

1.2) Full asociativa

Cache

2ⁿ líneas
o bloques



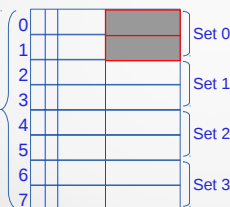
Cuatro cuestiones a resolver

1) Ubicación del bloque:

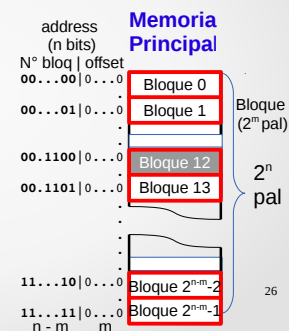
1.3) Set asociativa

Cache

2ⁿ líneas
o bloques



Set destino:
(Block addr) mod (Nº de sets en cache)



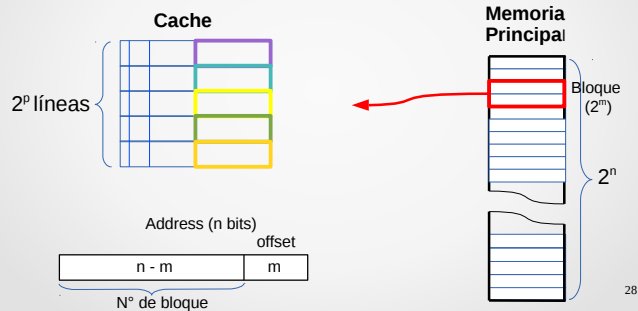
Cuatro cuestiones a resolver

2) Identificación del bloque: ejemplo



Cuatro cuestiones a resolver

3) Reemplazo de bloques



Cuatro cuestiones a resolver

3) ¿Qué bloque será reemplazado ante un miss?

- Si el esquema es mapeo directo → única opción
- ¿Cómo hacemos si el esquema es set-asociativo?
- **Tres alternativas:**
 - Random
 - LRU
 - FIFO

Cuatro cuestiones a resolver

4) Estrategia de escritura

- Primero veamos que pasa con las **lecturas**:
 - Lecturas dominan los accesos
 - El caso más común debe ser optimizado
 - En este caso, el caso más común es el caso más fácil
 - Lectura del dato // lectura y comparación de Tag (y válido)
- **Escrituras:**
 - No se pueden leer en // dato y tag
 - Tardan más que las lecturas

Cuatro cuestiones a resolver

4) Estrategia de escritura

- Ante una escritura **¿Qué hacemos con la actualización de esa información en Memoria Principal?**
 - Dos políticas:**
 - Write through:** la actualización se hace en cache y en MP en cada escritura
 - Consistencia, menor miss penalty, más simple
 - Write Back:** la actualización se hace en cache en cada escritura y en MP sólo en el reemplazo del bloque
 - Dirty bit
 - Escrituras al ritmo de cache, menor ancho de banda

31

Cuatro cuestiones a resolver

4) Estrategia de escritura

- ¿Qué pasa ante un miss en una escritura?**
 - Dos políticas:**
 - Write allocate:** traer el bloque al nivel superior y luego proceder a la escritura
 - No write allocate:** realizar la escritura en el nivel inferior sin traer el bloque al nivel superior
 - WA ↔ WB
 - NWA ↔ WT

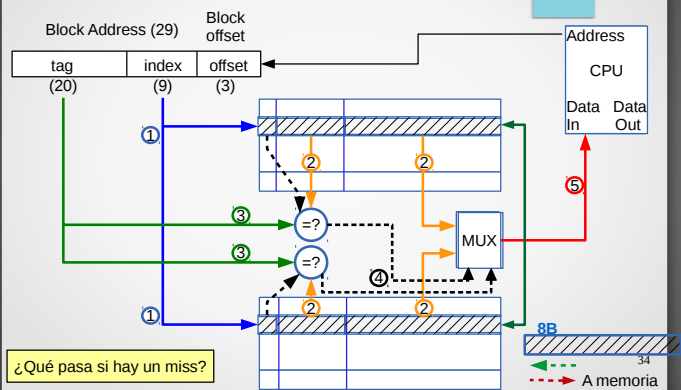
32

Ejemplo: Cache VAX11/780

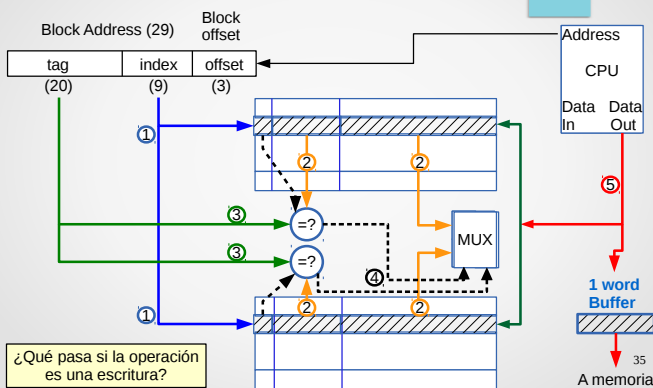
- Organización de caché:
 - 8KB para **datos**
 - Bloques de **8 bytes**
 - 32 bits** de address
 - 2-way** set asociativa
 - Algoritmo de reemplazo **Random**
 - Política de escritura: **write-through** (con buffer de 1 palabra) + **write NO allocate**
- Offset:
 - Si tamaño de bloque = 2^n
 - Offset = n bits
 - $(\log_2(\text{tam bloque}) = \log_2(8) = 3)$
- Index:
 - ¿Cuántos sets? → 2^{index}
 - $2^{\text{index}} = \frac{\text{Tam de cache (datos)}}{\text{Tam de bloque} * \text{Asociatividad}}$
 - $2^{\text{index}} = \frac{8192}{8 * 2} = \frac{2^{13}}{2^3 * 2} = 2^9$
- Tag:
 - Tag = Add – Offset – index
 - Tag = 32 – 3 – 9 = 20

¿Cómo están conformados los campos **Tag**, **Index** y **Offset**?

Ejemplo: Cache VAX11/780



Ejemplo: Cache VAX11/780



Referencias

- Hennessy, J., and Patterson, D. Computer Architecture, second ed. Morgan Kaufmann, 1996. (Capítulo 5)
- Hennessy, J., and Patterson, D. Computer Architecture, fourth ed. Morgan Kaufmann, 2006. (Apéndice C)

36