Tema 1: Memoria cache (parte 1)

Iñigo Perona Balda (CAS)
Nestor Garay (EUS) — Olatz Arbelaitz (CAS/EUS)

Universidad del País Vasco (UPV/EHU)
Grado en Ingeniería Informática
Arquitectura de Computadores

6 de septiembre de 2023

Índice de parte 1 + parte 2

Índice

Tabla de contenidos

Introducción

Problema

- El procesador es más rápido que la memoria y esta diferencia de velocidad entre la CPU y la memoria (gap) es cada vez mayor.
  - La CPU debe utilizar la memoria lo más rápido posible para acceder a las instrucciones y a los datos
  - Atención: **el componente más lento, la memoria, determina la velocidad de todo el sistema**
- El tiempo de acceso a memoria aumenta con el tamaño de la misma: las memorias pequeñas serán más rápidas. Sin embargo, necesitamos memorias grandes.
  - ¿Cómo estructurar el sistema de memoria para realizar las operaciones (lectura/escritura) lo más rápido posible? Repasemos los conceptos principales.

Tipos de memoria RAM

1. **RAM estática (SRAM): rápida**, pero necesita 4-5 transistores para cada bit de memoria (tipo biestable D).
2. **RAM dinámica (DRAM):** mayor integración (un transistor por cada bit de memoria: carga de un condensador), pero **más lenta**.

   Si el parámetro crítico es la velocidad, se utiliza RAM estática; sin embargo, si es la capacidad, mejor RAM dinámica.
3. **Memorias asociativas**: La búsqueda no se realiza por dirección, sino mediante el contenido. Dada una palabra, el resultado puede ser: **sí**, está en memoria; o **no**, no está (además de la información asociada a esa palabra; por ejemplo, dónde está). Son más complejas que las memorias RAM habituales y tienen un uso especial en las memorias cache.

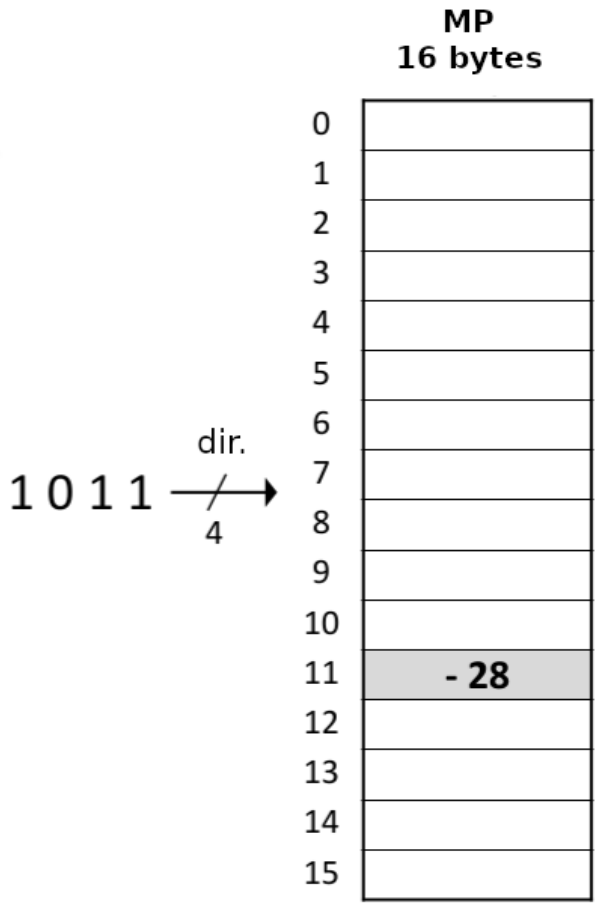
Tabla de contenidos

Estructura de la memoria principal (RAM)

- Una posición de memoria se accede mediante su dirección, tanto para leer como escribir.
- Las direcciones de una memoria con P posiciones son de $log_2 P$ bits; una dirección de n bits direcciona 2^n posiciones.
- **Comunicación entre la CPU y la MP**
  - Direcciones $\Rightarrow$ bus de dirección
  - Datos $\Rightarrow$ bus de datos
  - Operación (rd/wr) $\Rightarrow$ bus de control

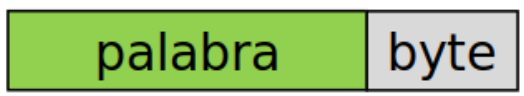
Introducción

Estructura de la memoria principal (RAM)



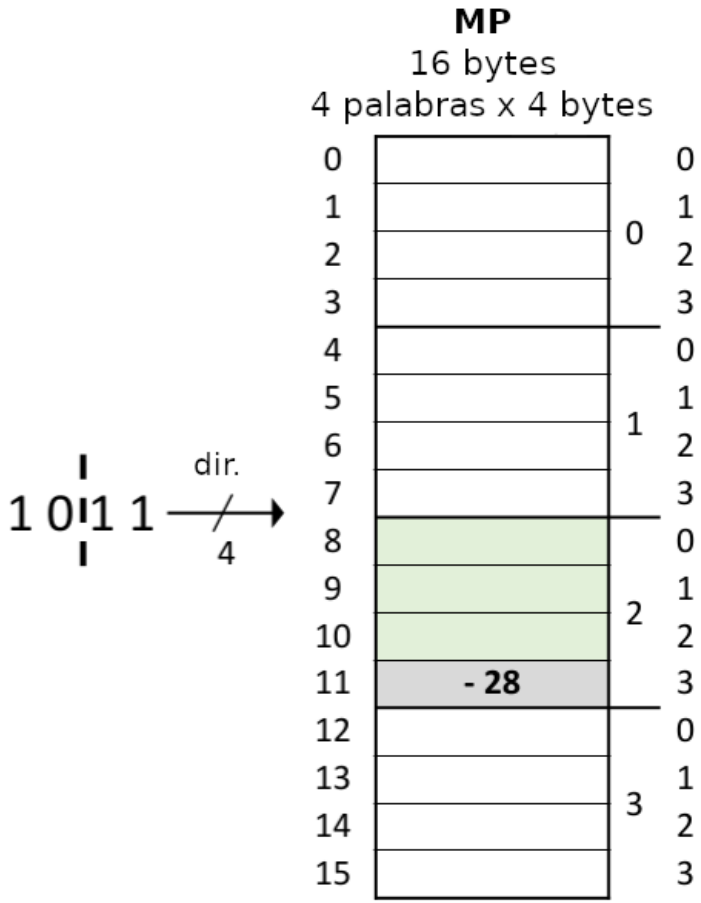
Estructura de la memoria principal (RAM)

- **Unidad de información** direccionada por el procesador:
  - **byte** (habitual)
  - palabra: 4 u **8 bytes**
- En general, las direcciones que genera el procesador suelen ser alineadas a la palabra (indican el comienzo de una palabra).
  Estructura de una dirección:



  - **palabra** = dir **div** tam_pal (división entera: $11/4 = 2$)
  - **byte** = dir **mod** tam_pal (resto de la división: $11 \bmod 4 = 3$)

Estructura de la memoria principal (RAM)



Introducción

Un apunte sobre las direcciones de memoria

- Al cargar el programa/datos en la MP, hay que decidir en qué posiciones se ubican, según qué espacio de memoria está libre en ese momento. Por tanto, las posiciones en las que se carga un programa en memoria no son fijas. El programa cargador decide en qué posiciones físicas se cargan el programa y sus datos.
- Por ello, el procesador utiliza **direcciones lógicas**, no direcciones físicas concretas. Las direcciones lógicas deben **traducirse** a **direcciones físicas** (de memoria). Esta traducción se realiza a través de un hardware especial: **TLB** (translation look-ahead buffer).
- Para simplificar, en este tema trabajaremos con **direcciones físicas**.

Tabla de contenidos

Estructura de la memoria principal (RAM)

- La memoria principal se organiza en varios módulos, de forma que las direcciones/posiciones se entrelazan entre los módulos.
- De esta forma, **se puede acceder de forma simultánea a direcciones consecutivas**, ya que estarán en módulos diferentes. El contenido de estas direcciones se guarda en los buffers de entrelazado y desde aquí, mediante el bus de datos, se transfieren al procesador

Estructura de la memoria principal (RAM)

- El nivel de entrelazado define el **tamaño del bloque** (en el ejemplo, 4 palabras)

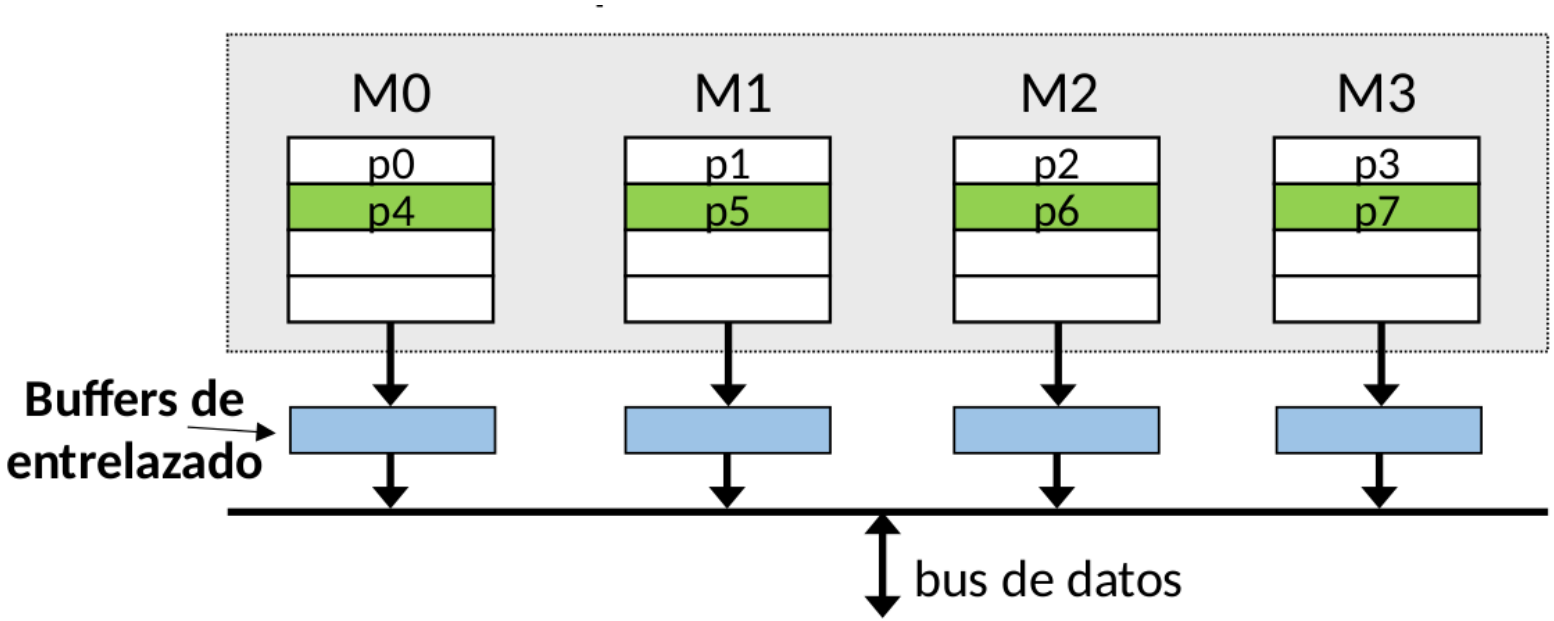


Tabla de contenidos

Características generales

Principio de localidad (locality)

- ¿Por qué se entrelaza la memoria? El acceso a las instrucciones y a los datos **no es aleatorio**
- "El 90 % del tiempo de acceso se consume en el 10 % del código"
- **localidad temporal**: si se utiliza una palabra, seguramente pronto se volverá a utilizar: bucles...

$$t \rightarrow @i \quad \Rightarrow \quad t + \Delta t \rightarrow @i$$

- **localidad espacial**: si se utiliza una palabra, seguramente la siguiente palabra será la palabra consecutiva: ejecución secuencial, acceso a vectores...

$$t \rightarrow @i \quad \Rightarrow \quad t + 1 \rightarrow @(i + \Delta)$$

Características generales

Principio de localidad (locality)

- Debido al principio de localidad, el sistema de memoria se organiza en varios niveles, definiendo una **jerarquía de memoria**
  - En los niveles superiores: **memoria cache** (MC)
    - Memoria pequeña pero rápida (cerca). RAM estática.
    - Instrucciones y datos que más utiliza el procesador.
    - Repartida en varios niveles (L1, el más pequeño; L2, L3).
    - Dado que se pueden integrar muchos transistores en un chip, L1 y L2 se integran dentro del chip: acceso rápido.
  - En los niveles inferiores: **memoria principal** (MP)
    - Memoria más grande (1.000 veces) pero más lenta (5–10 veces)
    - RAM dinámica

Características generales

Jerarquía de memoria

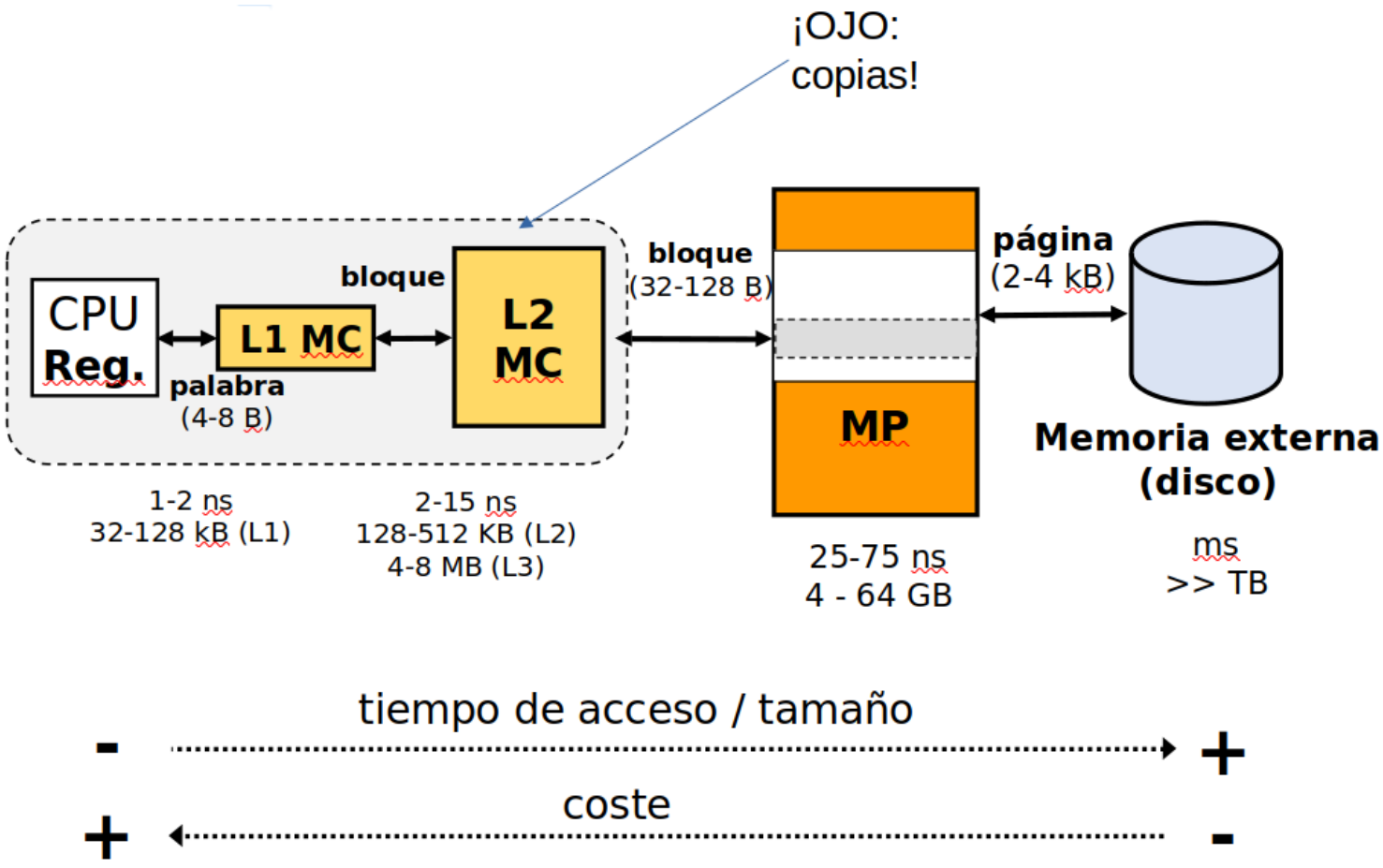


Tabla de contenidos

Características generales

Principio de inclusión

- **El contenido de un nivel** de la jerarquía de memoria **es un subconjunto del nivel anterior**. Por ello, puede haber varias copias de un bloque de datos, tal y como se indica en el esquema.

L1	1	-	-	-
L2	1	1	-	-
MP	1	1	1	-

  - No obstante, en algunos procesadores no se cumple esta condición y la cache L2 no es inclusiva: un bloque puede estar en L1 sin estar en L2.

Características generales

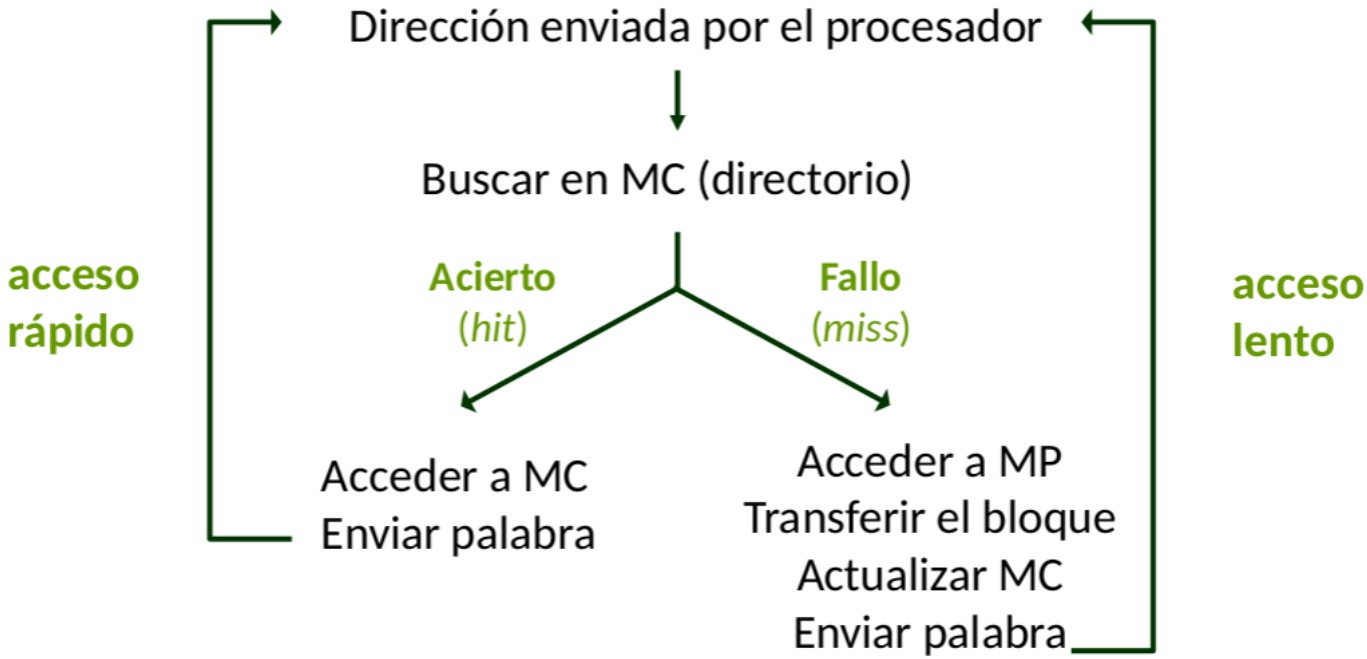
Principio de inclusión

- El procesador buscará la información en la **memoria más cercana**. Si no están en ese nivel, buscará en el siguiente nivel y así sucesivamente. Por ello, **el tiempo de acceso a la información es variable**, dependiendo de su localización.
  - Acceso rápido: la información que se busca está en un nivel cercano (en MC). Pero en la MC no cabe toda la información!
  - Hay que hacer una **APUESTA**: ¿qué se lleva a MC?

Tabla de contenidos

Características generales

- A la hora de buscar la información en la memoria cache, se puede producir un **acierto** (la información está en MC) o un **fallo** (no está en MC). Por ejemplo, al leer una palabra (solamente con L1 y MP):



Características generales

- En resumen, así se puede expresar el tiempo de acceso a la jerarquía de memoria:

$$T_{acceso} = h \times T_{MC} + (1 - h) \times T_{fallo}$$

  - h: tasa de acierto ($1 - h$, tasa de fallos)
  - T_{MC}: tiempo de acceso a memoria cache
  - T_{fallo}: tiempo para acceder a la información (instrucción/ dato) en el siguiente nivel (en función de la tasa de aciertos de cada nivel)
- **Para que sea eficiente**, la tasa de aciertos, **h, debe ser muy alta** (>95 %).

Características generales

Bloque

- **A la unidad de transferencia entre MC y MP se le llama bloque** (line)
  - Bloque: 2^n palabras consecutivas en memoria.
  - Por ejemplo, 64 bytes: 8 palabras de 8 bytes.
- ¿Por qué bloque y no palabra? **¡Localidad!**
  - Si se transfiere una palabra a MC para usarse en un determinado momento, **seguramente el siguiente acceso será a una palabra consecutiva**. Eso sí, se trata de una apuesta.
  - Por ello, se aprovecha la transferencia MP $\rightarrow$ MC para transferir más de una palabra, casi al mismo coste.
  - En general, el tamaño del bloque coincide con el entrelazado de la memoria.

Tabla de contenidos

Características generales

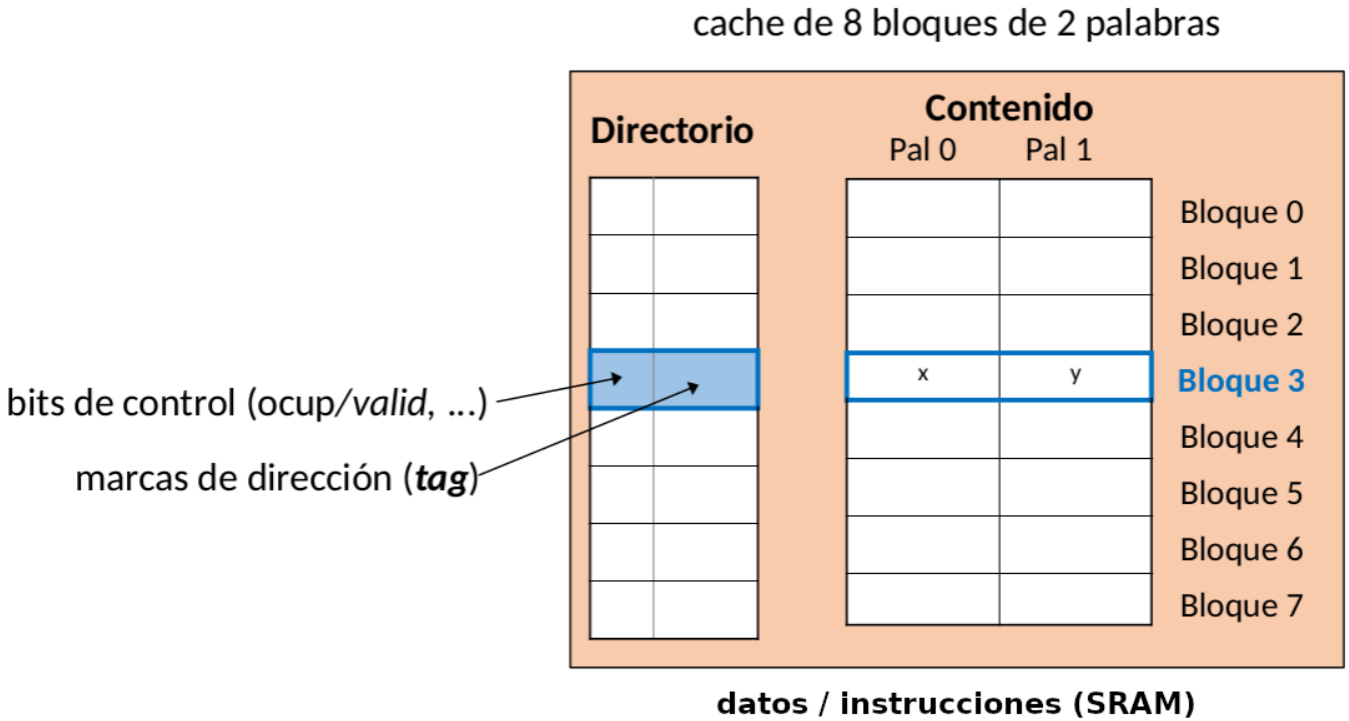
Estructura de la memoria cache: directorio / contenido

- La dirección de una palabra indica su posición en MP. ¿Pero en la memoria cache, cuál es su dirección?
- Una memoria cache tiene dos partes:
  - **contenido**: SRAM para almacenar los bloques de datos.
  - **directorio**: información acerca de los bloques de datos almacenados en la cache, una palabra por cada bloque de datos: marcas de dirección (tag) y varios bits de control del bloque.
    Dependiendo de la organización de la cache, el directorio puede ser una memoria asociativa (búsqueda por contenido).
- Además, hardware habitual: multiplexores, codificadores...

Características generales

Estructura de la memoria cache: directorio / contenido

- Directorio: hay información necesaria para saber si la palabra que se busca está en MC o no (SRAM / M. asoc.)



Tema 1: Memoria cache (parte 1)

Iñigo Perona Balda (CAS)
Nestor Garay (EUS) — Olatz Arbelaitz (CAS/EUS)

Universidad del País Vasco (UPV/EHU)
Grado en Ingeniería Informática
Arquitectura de Computadores

6 de septiembre de 2023