

Práctico 5 - Memorias

Ejercicio 1: Indicar si son correctas o no las siguientes afirmaciones, si se intenta realizar una comparación entre distintos sistemas de memoria tecnológicamente diferentes. Justificar su respuesta.

- A menor tiempo de acceso, mayor coste por bit.
 - A menor capacidad, menor coste por bit.
 - A mayor capacidad, menor tiempo de acceso.
 - A mayor tiempo de acceso, mayor capacidad.
 - A mayor frecuencia de acceso, menor capacidad y mayor coste por bit.
-
- Verdadero**, ya que si el tiempo de acceso es menor, la memoria es más rápida por lo que consecuentemente es más tecnológica en términos de performance.
 - Falso**, puede ser una memoria de poca capacidad pero muy cara en términos tecnológicos.
 - Falso**, el tiempo de acceso no está relacionado con la capacidad de la memoria, sino con la tecnología de la misma.
 - Verdadero**, si tiene menor tiempo de acceso es lenta, por lo que es una memoria grande.
 - Verdadero**, +frecuencia → menor capacidad de memoria → mejor tecnología → mayor coste por bit (es una caché).

Ejercicio 2:

Considerando un procesador que trabaja a 1.7GHz y los siguientes tiempos de acceso a memoria:

Memory technology	Typical access time	\$ per GiB in 2012
SRAM semiconductor memory	0.5–2.5 ns	\$500–\$1000
DRAM semiconductor memory	50–70 ns	\$10–\$20
Flash semiconductor memory	5,000–50,000 ns	\$0.75–\$1.00
Magnetic disk	5,000,000–20,000,000 ns	\$0.05–\$0.10

¿Cuántos ciclos de clock implica leer un dato de caché (SRAM) y de memoria principal (DRAM)?

Frecuencia del procesador → 1.7×10^9 ciclos de reloj por segundo.

Acceso a la caché → $0.5 / (1.7 \times 10^9) = 2.94 \times 10^{-10}$ y $2.5 / (1.7 \times 10^9) = 1.47 \times 10^{-9}$.

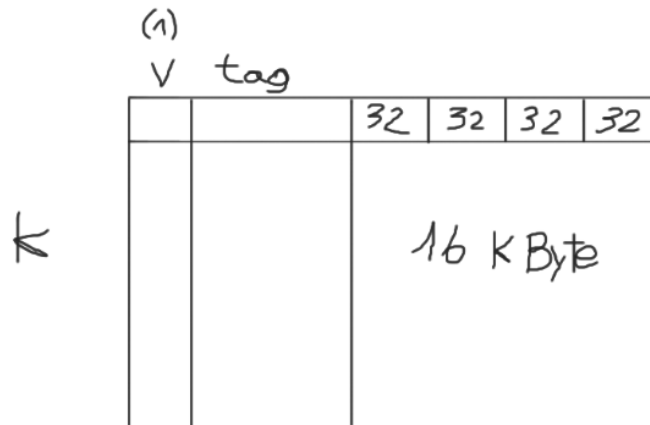
→ Acceso a la caché oscila entre 2.94 y 1.47 ciclos de reloj.

Acceso a la memoria principal → $50 / (1.7 \times 10^9)$ y $70 / (1.7 \times 10^9)$.

→ Acceso a la DRAM oscila entre 2.94 a 3.53 ciclos de reloj.

Ejercicio 3: Calcular el tamaño total (en bits) de una caché de mapeo directo de 16KiB (16K x 8 bits) de datos y tamaño de bloque de 4 palabras de 32 bits c/u. Asuma direcciones de 64 bits.

- Área de datos → 16 KiB
- Un bloque → 4 palabras de 32 bits
- Una línea → 4 palabras de 32 bits
- Un address tiene 64 bits.



$K = 16 \text{ KiB} / (4 * 32 \text{ bits}) = 2^{14} * 2^3 \text{ bits} / 2^7 \text{ bits} = 2^{10} \rightarrow \mathbf{1 \text{ Kbits.}}$

Bloques = $2^{64} \text{ bytes (palabras de memoria)} / 4(wm / wp) * 4(wp / b) = \mathbf{2^{60} \text{ bloques.}}$

Veces = bloques / líneas = $2^{60} / 2^{10} = \mathbf{2^{50} \text{ tag.}}$

→ Tamaño total de la caché: (V + tag + datos).

→ $(V + \text{tag}) * \text{líneas} + \text{caché datos.}$

→ $(1 + 50) * 2^{10} + 16 \text{ KiB.}$

→ **183296 bits.**

Ejercicio 4: Las memorias caché son fundamentales para elevar el rendimiento de un sistema de memoria jerárquico respecto del procesador. A continuación se da una lista de referencias de acceso a memoria (direcciones de 64 bits) las cuales deben ser consideradas como accesos secuenciales en ese mismo orden. El formato que se utiliza para cada dirección está reducido a sólo 16 bits, solo con fines prácticos:

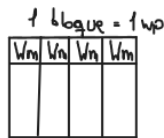
0x000C, 0x02D0, 0x00AC, 0x0008, 0x02FC, 0x0160, 0x02F8, 0x0038, 0x02D4, 0x00AC,
0x00B0, 0x0160

Se debe:

- Para cada una de estas referencias a memoria, determinar el binario de la dirección de cada palabra (cada palabra de 32 bits), la etiqueta (tag), el numero de linea (index) asignado en una cache de mapeo directo, con un tamaño de 16 bloques de 1 palabra c/u. Además liste qué referencias produjeron un acierto (hit) o un fallo (miss) de caché, suponiendo que la cache se inicializa vacía.
- Para cada una de estas referencias a memoria, determinar el binario de la dirección de cada palabra (cada palabra de 32 bits), la etiqueta (tag), el numero de linea (index) asignado en una cache de mapeo directo, con un tamaño de 8 bloques de 2 palabra c/u. Además liste qué referencias produjeron un acierto (hit) o un fallo (miss) de caché, suponiendo que la cache se inicializa vacía.

- * Word de 32 bits.
- * 16 bloques de 1 palabra.
- * 1 línea \rightarrow 1 palabra.

} Asumimos memoria $LeqV8 \therefore$



a. $0x000C \rightarrow$ $000...0000\ 0000\ 0000\ 1100$ } Analógos
tag Index 0B
 $0x02D0 \rightarrow$ $0000...0000\ 0010\ 1101\ 0000$
tag Index 0B
 $0x00AC \rightarrow$ $0000...0000\ 0000\ 1010\ 1100$
tag Index 0B

Address de 64 Bits.

	tag	V	Data
0		0	
1		0	
2		0	
3	0000...0000	1	[0x000C - 0x000F]
4	0000...000B	1	[0x02D0 - 0x02D5]
5		0	
6		0	
7		0	
8		0	
9		0	
A		0	
B	0000...0002	1	[0x00AC - 0x00AF]
C		0	
D		0	
E		0	
F		0	

1 wp

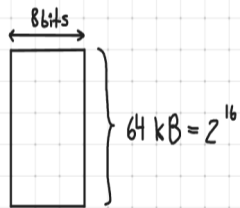
tag	index	W	Off B
58	3	1	2

0x000C = 0000 ... 0000 0000 0000 1100

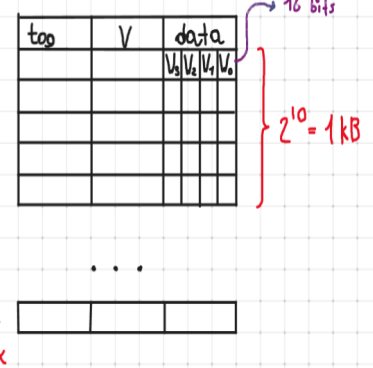
tag index

- ¿Cuántos bits hay en los diferentes campos del formato de dirección de memoria principal?
- ¿Cuántas líneas contiene la memoria caché?
- ¿Cuántos bits hay en cada línea de la memoria caché y cómo se dividen según su función?

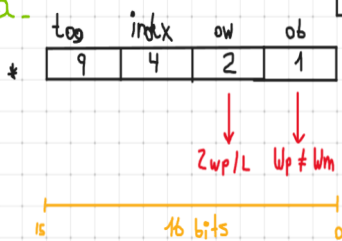
5. Memoria Principal:



Memoria Caché:



a.



b. * Número de líneas = $\frac{\text{Cache Data}}{\text{bits / línea}}$

$$\rightarrow \frac{2^{10}}{2^6} = 2^4 \rightarrow \text{index}$$

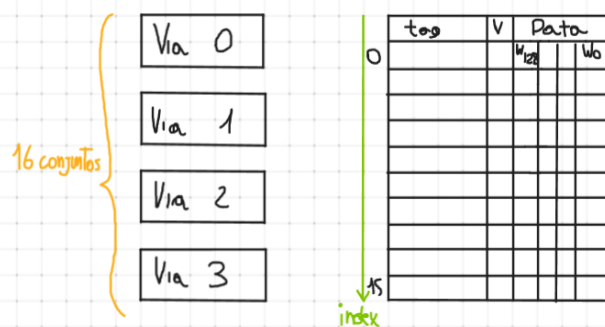
* tag = $\frac{\text{Memoria Principal}}{\text{Memoria Caché}}$

$$\rightarrow \frac{2^{16} \cdot 2^3}{2^{10}} = \frac{2^{19}}{2^{10}} = 2^9 \rightarrow \text{tag}$$

c. 9 bits (tag) + 1 bit (Validación) + 16 bits · 4 (Data) = **79 bits**

Ejercicio 6: Una caché asociativa por conjuntos consta de 64 líneas, dividida en 4 vías. La memoria principal contiene 4K bloques de 128 palabras cada uno. Muestre el formato de dirección de memoria principal suponiendo que cada palabra es direccionable directamente en memoria.

6.

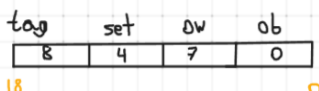


* Líneas de Caché: $\frac{64 \text{ líneas}}{4 \text{ vías}} = 16 \text{ líneas}$

* Set = 16 líneas = $2^4 \rightarrow 4 \text{ bits p/set}$

* Palabras = $4k \cdot 128 = 2^{12} \cdot 2^7 = 2^{19} \rightarrow \text{Address}$

* Ow = 128 w = 2^7



Verificación = $\frac{w_{MP}}{w_{Cache/Via}} = \frac{2^{19}}{2^7 \cdot 2^4} = 2^8$

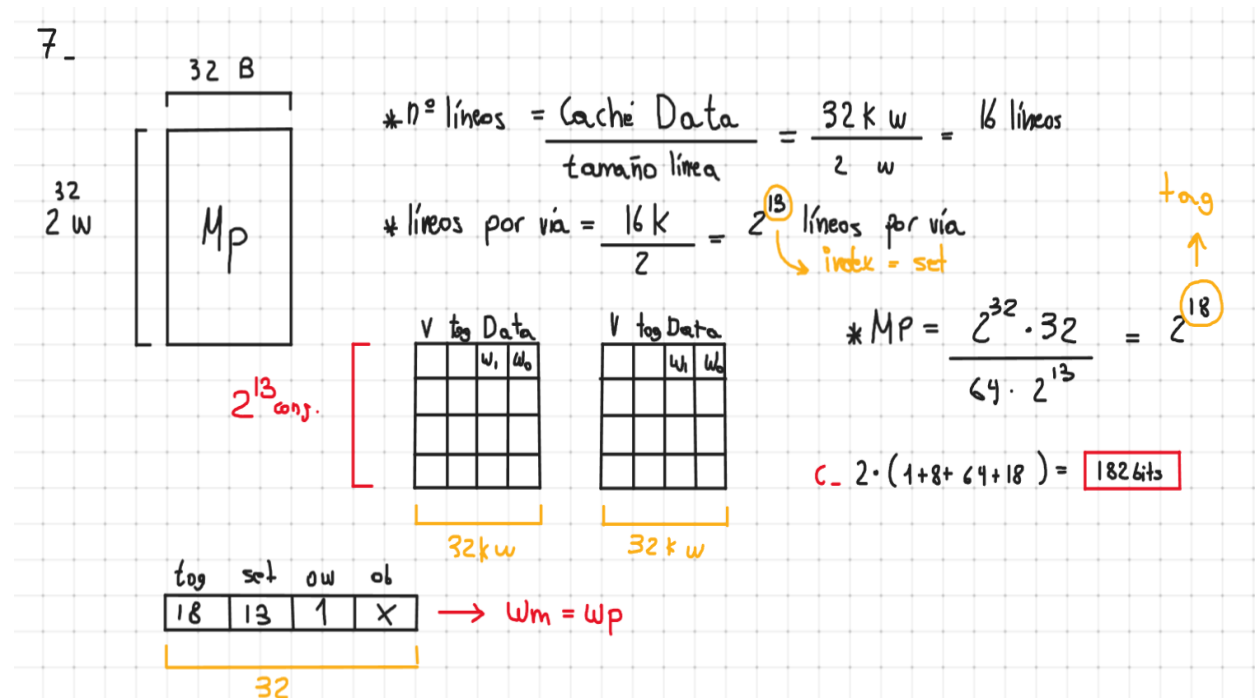
tag ↑
8

Ejercicio 7: Considere una CACHÉ con los siguientes parámetros:

- Criterio de correspondencia: Asociativa por conjuntos
- Asociatividad (N-vías): 2
- Tamaño de bloque: 2 words
- Tamaño de palabra (word): 32 bits
- Tamaño de la cache: 32K words
- Tamaño de dirección: 32 bits
- Cada palabra es directamente direccionable en memoria

Responder:

- Muestre el formato de dirección de memoria principal.
- ¿Cuál es el tamaño de toda el área de Tag de la caché, expresada en bits?
- Suponga que cada LÍNEA de la caché contiene además un bit de validación (V) y un tiempo de vida de 8 bits. Cual es el tamaño completo (expresado en bits) de un CONJUNTO de la caché, considerando datos, tags y los bits de status antes mencionados?



Ejercicio 8: Sea un sistema con una memoria principal de 1M palabras divididas en 4K bloques, donde cada palabra es direccionable directamente en memoria. Definir el formato de la dirección de memoria principal en los siguientes casos, sabiendo que la memoria caché posee 64 líneas:

- Memoria caché con función de correspondencia directa.
- Memoria caché con función de correspondencia full-asociativa.
- Memoria caché con función de correspondencia asociativa de 8 vías.

8-

a-

$WM = WP$

Bloque 0
Bloque 4k-1

4k bloques

2^{12}

1M

1 M palabras = 2^{20} → address

$$\ast \text{tag} = \frac{2^{12}}{2^6} = 2^6 \rightarrow \text{tag}$$

Index	tag	Data
0		wp_0
		wp_1
63		wp_{63}

tag	index	wp
6	6	8

20 bits

b-

tag	wp
12	8

c- $\text{líneas / vía} = \frac{2^6}{2^3} = 2^3 \rightarrow \text{set}$

tag	set	wp
9	3	8

Verificación: $\frac{4k}{2^3} = \frac{2^{12}}{2^3} = 2^9$