

## 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	0.5–2.5ns	\$500–\$1000
DRAM semiconductor memory	50–70ns	\$10–\$20
Flash semiconductor memory	5,000–50,000ns	\$0.75–\$1.00
Magnetic disk	5,000,000–20,000,000ns	\$0.05–\$0.10

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

$$\text{Frecuencia} = \frac{1}{t} \Rightarrow t = \frac{1}{\text{Frecuencia}} \quad t = \frac{1}{582 \times 10^{-10}} = 0,88 \text{ ns} \quad \begin{matrix} \text{CICLO} \\ \text{CLOCK} \end{matrix}$$

SRAM = entre 1 y 4 ciclos

DRAM = 86 y 120.

## 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.

$$\text{Cantidad de bloques} = \frac{\text{Cantidad de datos}}{\text{pal} \times \text{bloque}} = \frac{131072 \text{ b}}{728} \\ = 1024 \text{ bloques} = 2^{10} \rightarrow \text{índice}$$

Memoria: como entran 4 ( $2^2$ ) palabras de mem en una línea, entonces en OW es 2, luego la dirección divide en

50	10	2	2
tag	ou 0b		

OW = n = cantidad de palabras ( $2^n$ ) que entran en una línea de cache.

V	tag	Data
1	50	128

$$= 779 \text{ bits} \times 1024 \text{ líneas} \\ = 798208$$

## 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 línea (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 línea (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.

a) tamaño cache =  $16 \times 32b = 512 \text{ bits}$ .  
 tamaño de índice = 4 bits ( $2^4$ )  $\rightarrow$  cantidad de bloques.  
 $04 = 0$ .  
 luego la dir de memoria se mapea.



$0x000C = 000...00001100$  miss  
 $0x02D0 = 000...001011010000$  miss  
 $0x00AC = 000...000010101100$  miss  
 $0x0008 = 000...00001000$  miss

I	tag	data
0		
1		
2	0	$M[0x8]$
3	0	$M[xC]$
4	11	$M[0x2D0]$
5		
6		
7		
8		
9		
10		
11	2	$M[0xAC]$

b) 8 bloques  $\Rightarrow$  índice son 3 bits  $0w = 1 \text{ bit}$ .



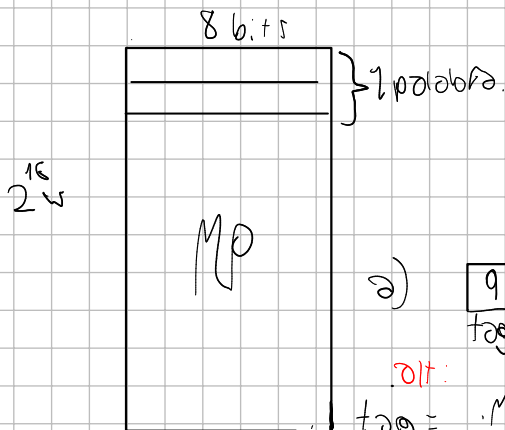
$0x000C = 000000001100$   
 $0x02D0 = 001011010000$   
 $0x00AC = 000010101100$   
 $0x0008 = 000000001000$   
 $0x02FC = 001011111100$

I	tag(b)	w <sub>1</sub>	w <sub>0</sub>
0			
1	0	$M[0xC]$	$M[0x8]$
2	1011		$M[0x2D0]$
3			
4			
5	10	$M[0xAC]$	
6			
7	1011	$M[0x2FC]$	

### Ejercicio 5:

Un computador tiene una memoria principal de 64K bytes y una memoria caché de 1K bits para datos. La memoria caché utiliza correspondencia (mapeo) directa, con un tamaño de línea de 4 palabras de 16 bits c/u. Obtener:

- ¿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?



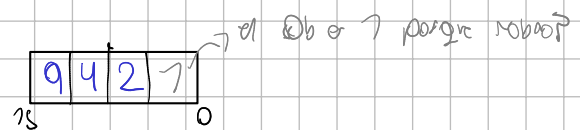
1 bloque = 4 palabras =  $4 \times 16b = 64b$ .

tengo 1024 bits de datos  $\Rightarrow$  # líneas =  $\frac{1024}{64} = 16$

índice = 4 bits,  $0b = 1$   $0w = 2$  (para diferencias entre 4 palabras)

a)  $9 \mid 4 \mid 2 \mid 1$   $\rightarrow$   $0b = 1$ ,  $1$  en es  $4b$   $0b = 2$ , que cambia?

bit:  $tag = \frac{MP}{Cache} = \frac{2^{16} \times 2^3}{2^{10}} = 2^9$

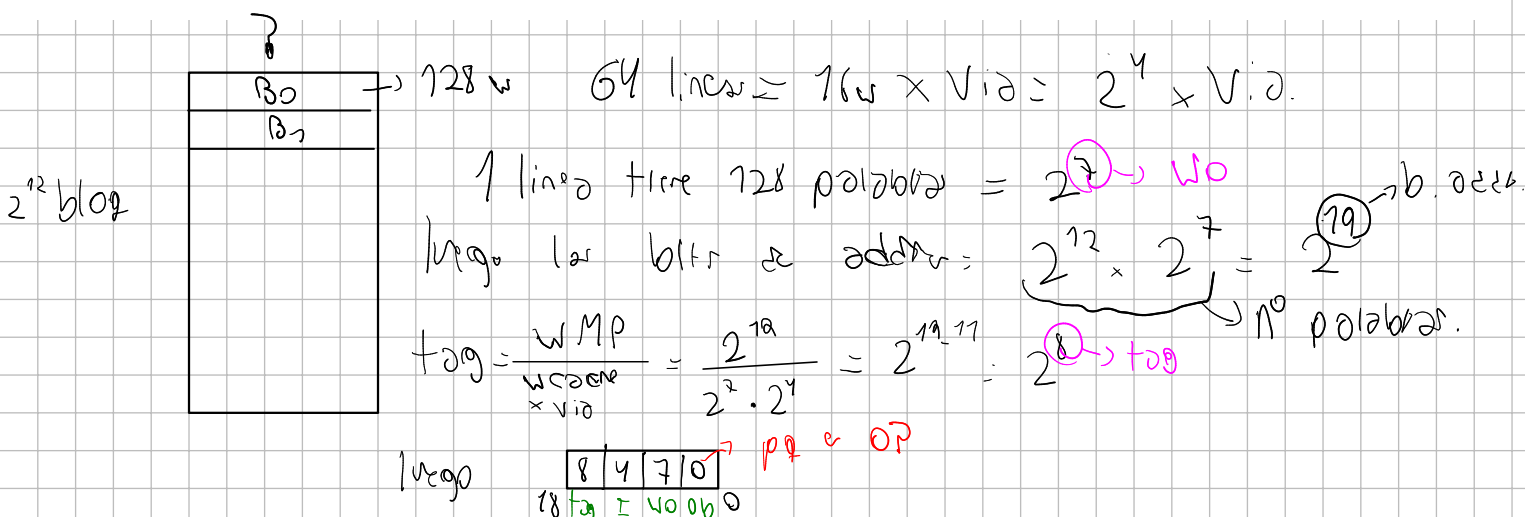


b) nº líneas: 16.

c) validación +  $tag + w_0 + w_1 + w_2 + w_3 = 24 \text{ bits} \times \text{línea}$ .

## 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.



## Ejercicio 7:

Considere una CACHE con los siguientes parámetros:

- Criterio de correspondencia: Asociativa por conjuntos
- Asociatividad (N-vías): 2
- Tamaño de bloque: 2 words = 64 bits.
- Tamaño de palabra (word): 32 bits
- Tamaño de la cache: 32K words = 16k bloques =  $n^\circ$  de líneas.
- 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 cache, expresada en bits?
- Suponga que cada LINEA de la cache 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 cache, considerando datos, tags y los bits de status antes mencionados?

$$\text{tag} = \frac{w_{MP}}{w_{cache} \times Vía} = \frac{2^{32}}{2^4 \times 2^{10}} = \frac{2^{32}}{2^{14}} = 2^{18} \text{ tag}, 2^4 \times \text{bloque} \Rightarrow w_0 = 1.$$

# bloques: 16k  $\Rightarrow$  8k bloque por vía  $\Rightarrow$  index =  $2^{13}$  index.

a) mem =

tag	I	w0	w1
18	13	1	0

b) la cache tiene la sig forma

2 <sup>13</sup>	v tag data			v tag data		
	v	tag	data	v	tag	data
	1	18	64	1	18	64

luego la cant de bits de tag que se guardan es =  $18 \times 2^{13} = 147.456$

c) una línea de conjunto tiene  $v + \text{tag} + d + 8 = 1 + 18 + 64 + 8 = 91$  bits. x línea de conj.  
luego ambos ocupan 182 bits. y un conj completo ocupa  $2^{13} \times 91 = 745.472$

## 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.

$1M = 2^{20}$  palabras  
 $4K = 2^{12}$  bloques  
 $\# w \times \text{bloque} = \frac{2^{20}}{2^{12}} = \frac{2^{20}}{2^{12}} = 2^8 = 256$  palabras por bloque  
 tengo 8b = 0.11

bde addr = 20 bits.

64 líneas = 6 index

$\text{tag} = \frac{w_{MP}}{w_{cache}} = \frac{2^{20}}{2^6 \cdot 2^8} = \frac{2^{20}}{2^{14}} = 2^6$  tag

a) mem = 

6	6	8	0
---	---	---	---

  
 tag I w0 b0

c) tengo 8 vías tengo  $\frac{2^{12}}{2^3} = 2^9$  bloques x vía  
 NO = igual.

$\text{tag} = \frac{w_{MP}}{w \times \text{vía}} = \frac{2^{20}}{2^9 \cdot 2^8} = \frac{2^{20}}{2^{17}} = 2^3$  tag

mem = 

3	9	8	0
---	---	---	---

  
 tag I w0 b0

b)

### Ejercicio 9:

Considere que un sistema tiene una CACHE para DATOS de correspondencia ASOCIATIVA POR CONJUNTOS de 2 VÍAS de 256Kbyte y 4 palabras por línea, sobre un procesador de 32bits, CPI = 1, que resuelve todos los data y control hazard sin necesidad de stalls, tiene una memoria principal de 4G palabras de 1 byte cada una.

a. Muestre el formato de memoria principal.

$\text{CONE datos} = 256 \text{ Kbyte}$        $\text{tamaño líneas} = 32 \text{ bits} = 16 \text{ bytes}$   
 $\text{Nº líneas} = \frac{\text{tamaño cone}}{\text{tamaño línea}} = \frac{256 \text{ Kbyte}}{16 \text{ bytes}} = \frac{262144 \text{ B}}{16 \text{ B}} = 16384 = 16 \text{ K líneas}$   
 $\frac{\text{Nº líneas}}{\text{vía}} = \frac{16 \text{ K}}{2} = 8 \text{ K}$        $\text{línea} \times \text{vía} = 2^{\text{index}}$  ,  $\text{OV} = 2$   
 $\text{tag} = \frac{\text{tamaño MP}}{\text{tamaño cone}} = \frac{2^{32} \cdot 2^3}{2^{10} \cdot 2^3} = \frac{2^{35}}{2^{13}} = 2^{22}$  , como 4G palabras 32 b de addr.  
 $\text{tag} = 32$        $\text{index} = 13$        $\text{vía} = 2$

b. Considere ahora la ejecución en este sistema de los siguientes segmentos de código LEGv8 equivalentes para N > 0, donde N → X3, start = 0x000203C y X6 contiene la dirección base del arreglo A[] del tipo uint32\_t. Considerar que el resto de los registros están inicializados en 0 y la CACHE vacía al inicio de la ejecución de cada segmento.

Segmento #1:	Segmento #2:
1 start: ADD X9, X3, XZR	start: LDURH X10, [X6, #0]
2 loop: CBZ X9, end X9 = 0	LDURH X11, [X6, #8]
3 LDURH X10, [X6, #0] A[X6]	ADD X12, X10, X11
4 LDURH X11, [X6, #8] A[X6+1]	LSR X12, X12, #1
5 ADD X12, X10, X11 X12 = A[X6] + A[X6+1]	ADDI X6, X6, #4
6 LSR X12, X12, #1	ADDI X9, X9, #1
7 ADDI X6, X6, #4 X6++	SUBS XZR, X9, X3
8 SUBI X9, X9, #1 X9--	B.NE start
9 B loop	end: ...
10 end: ...	

¿Cuántos MISS de CACHE se produjeron en 5 iteraciones del lazo para el segmento #1 si la dirección base del arreglo A es 0x00001008?

1. Itch.  
 3. > cache miss, porque A[0] = 0000 0000 0000 0000 0007 0000 0000 1000  
 0000 en cache línea 0x80  
 0x80  
 0x80, 1000, 1008, 100B, 100F

4. > hit  
 7. Itch.  
 3. > hit 100F  
 4. > miss