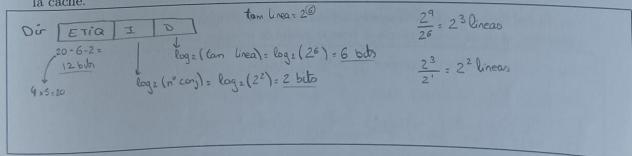
ESTRUCTURA DE COMPUTADORES Segunda evaluación (ejercicios)

1. Un computador tiene una memoria principal de 1 MiB. Tiene una caché de 512 Bytes, asociativa por conjuntos de 2 vías con algoritmo de reemplazo LRU, con un tamaño de línea de 64 Bytes y política de post-escritura (write-back). Estando la caché inicialmente vacía, se ejecuta el siguiente código. La matriz m se encuentra almacenada por filas a partir de la dirección de memoria 0xA0000, y el vector b a partir de la dirección 0xA0200. El resto de variables no provocan accesos a memoria caché y cada elemento de tipo "int" ocupa exactamente 4 Bytes.

a) [0.3p] Determina las direcciones de memoria a las que accede este código.

$$m [i] [o] = 0 \times A0000 + i \times 32 \times 4 = 0 \times A0000 + i \times 80$$
 $b [i] = 0 \times A0200 + i \times 4$
 $i = 0$
 $i = 1$
 $i = 2$
 $i = 3$
 $i =$

b) [0.2p] Indica razonadamente cómo se divide una dirección física desde el punto de vista de la caché.



sustituines la i en el aporteda a) y calcularros

c) [0.4p] Rellena los campos de la siguiente tabla para los accesos a memoria caché: dirección, etiqueta e índice/conjunto (en hexadecimal); si el acceso es un acierto o un fallo; en este último caso, qué tipo de fallo; y si se produce una escritura en memoria principal (MP WR).

_	UIOIII	timo case, que espe de fano, y si se produce una escritura en memoria principal (M1						
		Dirección	Etiqueta	Índice Acierto/Fallo		Tipo de fallo	MP WR	
	$m_{0,0}$	0×A0000	A00	0	Sallo	Sorzoso	NO	
	b_0	Ox A 0200	AO 2	0	Sallo	gor2050	NO	
*	$m_{0,0}$	0×A0000	A00	0	acierto		NO	
	סוגחי	0× A0080	A00	2.	Sallo	gorzoso	NO	
	ы	0× A0204	A02	0	aciento		NO	
*	mo, i	Y0004 ×0	A00	0	averto		NO	
	M2,0	0×A0 100	104	0	والمو	reemploso, garzoso	NO	
	bz	0 x A0208	A02	0	Sollo	conflictu	si xq rem	
*	m5,2	0x A0008	A00	0	عماه	conflicts	No	
	m3,0	0× A 0180	AOI	2	Salls	gorzoso	No	
	b3	0×A020C	A02	0	aciento	Martin Tolland	No	
*	mo,3	Dx A000C	A00	0	acients		100	

d) [0.3p] Indica cómo quedaría el directorio caché en las líneas ocupadas tras la ejecución de este código. BV = Bit de validez, BM = Bit de modificación

Conjunto	Vía	BV	ВМ	Etiqueta
0	0	1	0	A02
0	1	1	1 *	A 00
2	0	1	0	A00
2	1	1	0	AOI
			1741	

e) [0.2p] Sabiendo que el tiempo de acierto caché es de 2 ciclos y el tiempo de acceso a memoria principal es de 48 ciclos, calcula el tiempo medio de acceso a memoria.

- 2. [0.6p] Discute la veracidad de las siguientes afirmaciones
 - a) La política de **ubicar en escritura** es mejor tanto para cachés de post-escritura como de escritura directa.

En las cachés de post lescritura podemos realizar escrituras en linear de la caché sin recepcidad de escribir en la memora principal hasta que esa linea se expulse de la caché, por lo que resprovechon si se cumplen los principios de localidad.

En la escritura directa las escrituras se haven siempre en memoria principal, por logue el hecho de tenor la linea en lacadré no supore ninguin beneficio en el trempo empleado en una escritura.

b) En el sistema descrito en el ejercicio 1, incorporando una caché de segundo nivel de mayor tamaño se reduciría el tiempo medio de acceso a memoria

Los reemplasos en fallos de conflictos se resuduen en la cadré de nivel 2, porque aurque exe bloque haya sido expulsado antes de la cadré de nivel 1, es muy probable que siga estando en la de nivel 2 y por lo tanto no sea necesario boyar de nuevo a la memoria principal.

Esto no ocurre con los gallos gorzosos porque al ser la primera ver que se acceden no estaró en ninguna de las cachiés.

c) En el sistema descrito en el ejercicio 1, se podrían reducir los fallos de conflicto **aumentando la asociatividad** de la caché a 4 vías

No es recersorio hacer la labla de nuevo, pensemos:

Si duplicamos la asociatividad, la longitud del indice se reduce en un bit y por lo tento lodos los aciesos del ejercicio caerán en el conjunto.

Se accede a 5 bloques de memoria diferentes, osí que no cabián todos en el miomo conjunto y por lotante habra remplano en algún momento.

Sin embargo, la línea que se reemplanará será la ADO, indice 2, que no se reutiliza, y por lotante no habra gallos de conflicto (todos los Julios son for zosos).

COMPUTER STRUCTURE Second test (exercises)

LAST NAME:	FIRST NAME:

A210.24=214 direcciones de 14 bits 1. A computer has 16 KiB of main memory and a 2-way set-associative 1 KiB cache memory [1.4p] with 16-byte lines. The cache uses a FIFO replacement algorithm. The write policies are write-back and write-allocate. Starting with an empty cache, we execute the following code. The matrix A is stored by rows starting on address 0x3000, and each float element takes up exactly 4 bytes. Scalar variables are stored in CPU registers, and do not generate memory accesses.

> float A[8][128]; for (i=0; i<4; i++) A[0][i] = -1 * A[i][0];

a) [0.3p] Calculate the memory addresses accessed by this code.

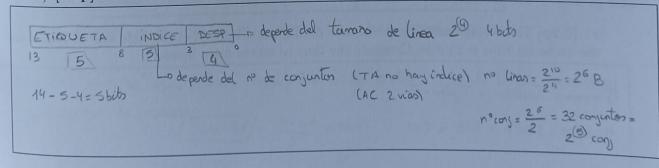
$$&A \text{ $Ci3COZ} = 0 \times 3000 + i \times 128 \times 4 = 0 \times 3000 + i 0 \times 200 \text{ (lectura)}$$

$$2^{7} \cdot 2^{2} = 2^{9} = 2.00$$

$$&A \text{ $Co3COZ} = 0 \times 3000 + i \times 4 \text{ (excittura)}$$

$$&\text{ $Ii=0$} & A \text{ $Co3COZ} = 0 \times 3000$$

b) [0.2p] Reason the breakdown of a physical address from the cache point of view.



c) [0.4p] Fill all fields on the following table for memory accesses: address, label and index/set (in hex format); whether the access results in a hit or a miss; in the latter case, what type of miss is it; and whether the access triggers a write to Main Main Main (MM) or not

	-		Lobel Label				av		
		Address	Label	Index	Hit/Miss	Miss type	MM WR	BM	covo
	$a_{0,}$	0 0x 3000	11000	00000	F	Sar 2050	NO		COVO
5	a_{0}	0 0×3000	11000	00000	A		NO		covi
	$a_{1,}$	0 0x3200	11001	00000	F	8012050	20		covo
*	a.,	1 0x 3004	11000	00000	A		20	1	COVO
reemplano E	0,2,	o 0 × 3400	11010	00000	F	Sarzosa	,sr	0	COVI
remplotos *			11000	00000	F	conflicto	/ NO		co vo
remplaso e			11011	00000	F	3012050	20		
*	a o,	0×300 C	11000	00000	А	_/	70	1	COVI
						aa estavo	en la caché		9_
				rempla	famos algo	u no esde	copavdod × q		-
					0	la caché	esta casi vac	ía .	

d) [0.3p] Write down the final state of the cache directory in the used lines after running this code.

Set	Way	Valid bit	Dirty bit	Label
0	0	1	0	11011
0	1	٨	1	11000
	TO REA			
	Hayba	EMP. NO		
	9000	AVA B		
	MAN			

e) [0.2p] The average memory access time during the execution of the code was 46 cycles, and the cache hit time is 1 cycle. Compute the time to access main memory.

2. [0.6p] Discuss the truthfulness of the following statements:

6

a) In general, the average memory access time improves when increasing the cache associativity because there are fewer capacity misses.

da asociatividad reduce los gallos de conflicto, no de capacidad. Podría reducir el tiempo medio de acesa por eliminar gallos de conflicto, pero no los atros

b) In a multi-level cache, it is possible to observe a low global miss rate while the lower level miss rate is close to $100\,\%$.

En una cadié de multiples nivelos es possible observar la tasa de gallos global boja mientras la tasa do gallos de la de último nivel es del 100% VOF

Verdodero, la caché de rivel más bajo en la jerorquia normalmente llegan pocos accros y la mayora pueden producir fallos que tienen que ser resueltos con un aceso a la memoria principal.

c) In the system desribed in exercise 1, if the replacement algorithm was LRU instead of FIFO, misses would not trigger a write to Main Memory.

Si que combienia, porque al usar el CRU necesitamos madificar la via 2 que no tiene BM, entonos no se excibe en memoria y hay menos accesos.

Con LRU ya no se retira la linea modificada, sino la otra y por lo tanto ya no hay que escubir en memoria.