

### PROBLEMA 13

(con bus de 8 bytes de ancho)

L1  $\rightarrow$  bloques de 32 bytes conectado a L2

L1  $\rightarrow$   $t_a = 1$  ciclo (acierto)

$\rightarrow$  fallo  $\rightarrow$  L2  
acceso



$f = 2 \text{ GHz}$   $2s \rightarrow$  caso ideal,  $10^9$  accesos a memoria y 20% provoca fallo en L1

a) Calculad el tiempo de ciclo y los ciclos que tarda en el caso ideal

$$t_c = \frac{1}{2 \cdot 10^9} = 5 \cdot 10^{-10} \text{ s} \quad \text{Texe} = \frac{\text{ciclos}}{f} \Rightarrow \text{ciclos} = 2 \cdot 5 \cdot 10^9 = 4 \cdot 10^9 \text{ ciclos}$$

b) Calculad los ciclos de penalización de un fallo de L1 y el Texe del programa teniendo en cuenta la penalización debido a los fallos de L1.

$$\text{Ciclos penalización} = 5 (\text{latencia}) + 4 (\text{carga bus}) + 1 (\text{lectura y envío de datos}) = 10 \text{ ciclos}$$

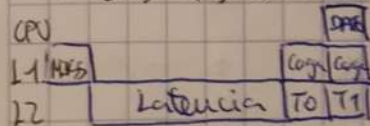
$$\text{Texe} = \text{ciclos} \cdot t_c = t_c (\text{Ciclos ideal} + nr \cdot \text{miss} \cdot \text{Ciclos penalización}) = 5 \cdot 10^{-10} (4 \cdot 10^9 + 10^9 \cdot 0.2 \cdot 10) =$$

3s

L2 a L1 (early restart). 70%  $\rightarrow$  ciclo T0 Mejora rendimiento  
10%  $\rightarrow$  ciclos T1-T3

c) Completad el siguiente cronograma en donde se ilustran las acciones a realizar en cas de fallo en L1, suponiendo que tenemos continuación anticipada y que el dato solicitado se corresponde al byte 12 del bloque

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16



Se necesita 2 ciclos para la carga porque el bus es de 8 bytes y queremos el byte 12 del bloque

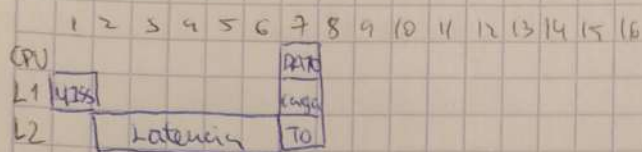
d) Calculad el tiempo medio de penalización (en ciclos) de un fallo en L1 y el tiempo de ejecución del programa para el procesador con continuación anticipada

$$\text{Ciclos penalización} = 5 + 0.7 \cdot 1 + 0.1 \cdot 2 + 0.1 \cdot 3 + 0.1 \cdot 4 = 6.6 \text{ ciclos}$$

$$\text{Texe} = t_c \cdot \text{ciclos} = 5 \cdot 10^{-10} (4 \cdot 10^9 + 10^9 \cdot 0.2 \cdot 6.6) = 2.66 \text{ s}$$

L2 → transfiere dato solicitado en T0

e) Completad el siguiente cronograma en donde se ilustran las acciones a realizar en caso de fallo en L1, suponiendo que tenemos transferencia en desorden y que el dato solicitado se corresponde al byte 12 del bloque.



f) Calculad el tiempo medio de penalización (en ciclos) de un fallo en L1 y el Tpe del programa

$$\text{Ciclos penalización} = 5 + 1 = 6 \text{ ciclos}$$

$$\text{Tpe} = 5 \cdot 10^{-10} (4 \cdot 10^9 + 10^9 \cdot 0.2 \cdot 6) = 2.6 \text{ s}$$

g) Calculad el speedup al sistema con continuación anticipada y del sistema con transferencia en desorden respecto al inicial (sin mejora)

$$\text{Speedup (early restart)} = \frac{3}{2.66} = 1.1278 \quad (12.78 \%)$$

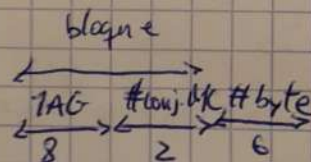
$$\text{Speedup (transferencia en desorden)} = \frac{3}{2.16} = 1.3889 \quad (15.38 \%)$$

#### PROBLEMA 14

Prefetch

16 bits con direcciones de 16 bits

$$16/3 = 4 = 2^2$$



MC: 3-asociativa, LRU, 12 bloques y 64 bytes por bloque, política escritura: copy back + write allocate

Contenido inicial:

DB = 1 = Bloque modificado

Conjunto 0	DB	Conjunto 1	DB	Conjunto 2	DB	Conjunto 3	DB
13	1	13	1	13	0	13	0
43	1	43	1	43	0	43	0
AC	0	AC	0	AC	1	AC	1

a) Rellenad la tabla siguiente

tipo	dirección	Bloque	TAG	conjunto MC	C/A/F?	Bloque reemplazado	bytes E	bytes L
LECT	B12B	2C4	B1	0	F	2B0	0	64
LECT	B145	2C5	B1	1	F	2B1	0	64
LECT	B1AF	2C6	B1	2	F	2B2	04	64
LECT	B1C4	2C7	B1	3	F	2B3	64	64
EXR	4357	10E	43	2	A	-	0	0
LECT	1108	044	11	0	F	10D	64	64
ESCR	1109	046	11	2	F	4C	0	64
LECT	11AA	046	11	2	A	-	0	0



Usamos prefetch de una entrada. Hare búsqueda de it's siempre y cuando aún no esté en HC o buffer.

b) Reemplazad la siguiente tabla

Tipo	dirección	Hoque memoria	TAG	Count HC	CA/FE?	bytes E	bytes L	Hoque buffer	buffer: CA/FE?	Hoque prefetch
LECT	B12B	2C4	B1	0	F	0	128	-	F	2C5
LECT	B145	2C5	B1	1	F	0	64	2C5	A	2C6
LECT	B1AF	2C6	B1	2	F	64	64	2C6	A	2C7
LECT	B1C4	2C7	B1	3	F	64	64	2C7	A	2C8
ESCR	4397	10E	43	2	A	0	0	2C8	-	-
LECT	1108	044	11	0	F	64	128	2C8	F	045
ESCR	1199	046	11	2	F	0	128	045	F	047
LECT	11AA	046	11	2	A	0	0	047	-	-

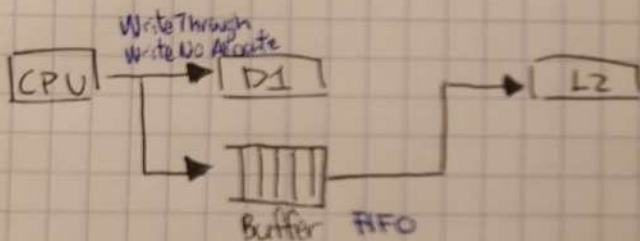
### PROBLEMA 16

```

movl %0, %esi
A: movl %eax, a(, %esi, 4); store a[i]
B: movl %eax, b(, %esi, 4); store b[i]
i: incl %esi; i++
c: cmpl $N, %esi; i < N?
j: jla

```

vector a → 0x1000 0000  
vector b → 0x2000 0000



Instrucciones a 1 ciclo todas

d) ¿Razona, a qué es debido?

Debido a que ya se ha mejorado al máximo, ya va 1 cti, añadiendo más entradas al buffer ya no mejora la CPI

# Apéndice: Cronogramas Tema 3

## Problema 16

a)

Cronograma 1: Buffer de 1 entrada.

Iteración	Iteración 0										Iteración 1										Iteración 2										Iteración 3														
Ciclo	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	
movl %eax, a(%esi,4)	A										-	-	A										-	-	A																				
movl %eax, b(%esi,4)	-	-	-	-	-	-	B																																						
incl %esi								i																																					
cmpl \$N, %esi									c																																				
jl A										j																																			
Ocupación bus																																													
# Buffer	0	1	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0	1		
Buffer[0]																																													
Buffer[1]																																													

$$CPI = 11999998 / 5000000 = 2,4$$

$$\text{Ancho de banda} = 8000000 \text{ bytes} / 11999998 = 0.66 \text{ bytes/ciclo}$$

Cronograma 2: Buffer de 2 entradas

Iteración	Iteración 0										Iteración 1										Iteración 2										Iteración 3										Iteración 4									
Ciclo	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44						
movl %eax, a(%esi,4)	A					-	A																																											
movl %eax, b(%esi,4)		B						B																																										
incl %esi			I						I																																									
cmpl \$N, %esi				C						C																																								
jl A					J						J																																							
Ocupación bus																																																		
# Buffer	0	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0	1								
Buffer[0]																																																		
Buffer[1]																																																		

$$CPI = 1,2 \text{ cli} \quad \text{Ancho de banda} = 8000000 / 6000005 = 1,3 \text{ bytes/ciclo}$$

$$\left[ 5 + 6(1000000) \right] / 5000000 = 1,2 \text{ cli}$$



Cronograma 3: Buffer de 3 entradas

Iteración	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Ciclo	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
movl %eax, a(,%esi,4)	A					A					A					A	B				A				A		B				A	B												
movl %eax, b(,%esi,4)		B					B					B										B																						
incl %esi			C					C					C					C					C																					
cmpl \$N, %esi				1					1						1					1																								
jl A																																												
Occupación bus																																												
# Buffer	0	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
Buffer[0]																																												
Buffer[1]																																												
Buffer[2]																																												

CPI =  $1 \text{ C/I}$   
Ancho de banda =  $8000000 / 5000000 = 1.6 \text{ bytes/ciclo}$

Cronograma 4: Merge buffer de 3 entradas

Iteración	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Ciclo	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
movl %eax, a(,%esi,4)	A					A					A					A	B				A				A		B				A	B												
movl %eax, b(,%esi,4)		B					B					B										B																						
incl %esi			C					C					C					C					C																					
cmpl \$N, %esi				1					1						1					1																								
jl A																																												
Occupación bus																																												
# Buffer	0	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1
Buffer[0]																																												
Buffer[1]																																												
Buffer[2]																																												

CPI =  $1 \text{ C/I}$   
Ancho de banda =  $1.6 \text{ bytes/ciclo}$