

Relacion-resuelta-3.pdf



Macs017



Estructura de Computadores



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingeniería Informática
Universidad de Málaga



Tu ordenador lo único que necesita
programar es su jubilación.

Stealth 15M



El Stealth 15M es uno de los portátiles gaming más finos y ligeros. Siempre menos es más. Ve a donde quieras llevando siempre el máximo rendimiento.



NEW YORK A UN SALTO

La ciudad que nunca duerme

365 DÍAS POR
89,99 €
¡SOLO AHORA!



ELIGE DESTINO
Y JUÉGALO EN

PlayStation Plus
PREMIUM



110,99€

¡SOLO AHORA!
89,99 €
AL AÑO

DEL 12 AL 20 DE DICIEMBRE

Marvel's
Spider-Man y
cientos de juegos
para viajar donde
quieras.

Consigue esta oferta con las Tarjetas
Regalo PlayStation en



TE FALTA FNAC | FNAC.ES

Promoción válida en PlayStation
Store hasta el 20 de diciembre

Relación resuelta

jueves, 28 de noviembre de 2019 11:10

Ejercicio 2

- Un computador tiene un bus de direcciones de 32 bits y es direccionable a nivel de byte. La cache tiene 256 Kbytes, con palabras de 1 byte y bloques de 32 bits, asignación directa, reemplazo LRU y escritura. También tiene 1 bit de dirty para los bloques que hayan sido modificados y por tanto tienen que escribirse en memoria principal. Responde a las siguientes preguntas:
 - Descompon en campos la dirección física (campo TAG, campo índice, etc.)
 - ¿Cuál es el tamaño, en bits, del directorio de la cache? Considera que en el directorio encuentran todos los bits de control necesarios para el funcionamiento de la memoria cache.
 - ¿Son coherentes todos los datos que proporcionan sobre la memoria caché? Justifica tu respuesta.

a) Memoria Directa

$$MC = 2^8 \text{ bytes} = 2^{10} \text{ bytes}$$

$$\text{bloques de } 32 \text{ bits} = 2^5 \text{ bytes} > \frac{2^{10}}{2^2} = 2^8 \text{ bloques en cache}$$

palabras de 2^0 bytes
2 bit dirty

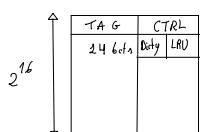
32 bits



$$32 - 10 = 14 \quad 16 \quad 2$$

b)

$$\text{Directorio} \leq 2^{16} \cdot 14 + 1 (\text{dirty}) + 5 (\text{LRU}) \leq 2^{16} \cdot 20$$



Ejercicio 2

- Una memoria caché....Una memoria cache asocial por conjuntos de 2 bloques con direcciones de 32 bits y palabras de 1 byte tiene 4 bytes en cada uno de sus bloques y una capacidad total de 128 Kbytes. El direccionamiento es a nivel de byte.

Se pide:

- El tamaño de una dirección de memoria y su descomposición en los campos número de línea o bloque (N), etiqueta (N-c), conjunto (c) y desplazamiento dentro del bloque (m).
- Indicar el valor de cada campo para las siguientes direcciones de memoria principal:
 - 0284A482h, 01148C89h, 0038CF00h, 0038CF01h
- ¿Pueden todos los bloques que incluyen las referencias anteriores estar en caché al mismo tiempo?

Asociativa por conjuntos (2 uan)

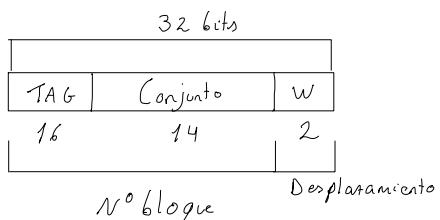
Direcciones de 32 bits

Palabras de 1 byte

$$\text{bloques de } 4 \text{ palabras} = 2^2 \text{ bytes} \quad MC = 2^7 \text{ kb} = 2^{17} \text{ bytes} \quad \frac{2^{17}}{2^2} = 2^{15} \text{ bloques}$$

a)

$$C = \frac{2^{15}}{2^2} = 2$$



b)

$$0284A482h = 101000\ 0100|1010\ 0100100000|20$$

' , , , , TAG = 644 C = 20528 W = 2

(Los casos restantes siguen el mismo procedimiento)

C) Si, ya que están en conjuntos distintos

Ejercicio 3

3. Sea un sistema de memoria de 1 MByte, con palabras de 1 byte, que incorpora una memoria caché de 64 KBytes organizada asociativamente en cuatro conjuntos con 8 palabras por bloque y algoritmo de reemplazo LRU. Se ejecuta un programa que referencia a la siguiente secuencia de direcciones (en hexadecimal):

ABC80b ABC81b ABC88b BCD90b BCD9D_b BCDA0b CDE00b CDE18b CDE20b

- a) Muestra la evolución de la cache, indicando para cada referencia si se produce un fallo o un acierto. Calcula el índice de fallos.
 - b) Repite el ejercicio anterior calculando el índice de fallos suponiendo que el número de iteraciones tiene infinito.

$$MM = 1 MB = 2^{20} \text{ bytes}$$

NEW YORK A UN SALTO

La ciudad que nunca
duerme

365 DÍAS POR

89,99 €

¡SOLO AHORA!



PlayStation®

ELIGE DESTINO Y JUÉGALO EN



PlayStation®Plus

PREMIUM

Marvel's Spider-Man y cientos de juegos
para viajar donde quieras.

119,99 €

¡SOLO AHORA!
89,99 €
AL AÑO

DEL 12 AL 20 DE DICIEMBRE



Consigue esta oferta con las Tarjetas Regalo
PlayStation en



TE FALTA FNAC | FNAC.ES



Promoción válida en PlayStation Store hasta el 20 de diciembre

palabras de 2⁰ bytes

$$MC = 64\text{ Kb} = 2^6\text{ Kb} = 2^{16}\text{ bytes}$$

$$\text{Bloques de } 8 \text{ palabras} = 2^3 \text{ bytes} \quad > \frac{2^{16}}{2^3} = \frac{2^{13} \text{ bloques}}{2^2 \text{ conjuntos}} = 2^{11} \text{ vías (bloques por conjunto)}$$

	dcr. Mem	Nº Bloque	Conjunto
ABC80h = 10101012120010000000	703616	87952	1936 = C ₀ F
ABC82h = 10101012120010000002	703617	87952	1936 = C ₀ A
ABC88h =	703624	87953	C ₁ F
BCD90h =	773520	96690	C ₂ F
BCD9Dh =	773533	96691	C ₃ F
BCDA0h =	773536	96692	C ₀ F
CDE00h =	843264	105408	C ₀ F
CDE18h =	843288	105421	C ₃ F
CDE20h =	843296	105442	C ₀ F

Indice fallo: $\frac{8}{9}$

6) Suponiendo un nº de iteraciones

infinitas, el índice de fallo tiende a 0,
ya que si se vuelve a iterar, todos los bloques
se encuentran en cache y no se produciría ningún fallo

Ejercicio 4

4. Disponemos de una memoria cache de 4 Kb con tamaño de bloque de 256 bytes, asociatividad 2 y algoritmo de reemplazo LRU. Conectamos la cache entre un procesador que trabaja con palabras de 8 bits y una memoria principal de 1 Mb. Se pide:

- a) Suponiendo la cache inicialmente vacía, describir la evolución del directorio cache para la siguiente secuencia de peticiones de direcciones a memoria principal:

319F0h, 31AF0h, 7013Ch, 77777h, 44037h, 7780Eh, A5021h

- b) Comparar el Sistema anterior con respecto a criterios de rendimiento y coste, frente a una caché organizada de forma directa y otra totalmente asociativa.

$$MC = 4\text{ Kb} = 2^2\text{ Kb} = 2^{12}\text{ bytes} \quad > \frac{2^{12}}{2^8} = 2^4 \text{ bloques en MC}$$

Bloques de 2⁸ bytes

Asociatividad 2 vías

Reemplazo LRU

$$\frac{2^4}{2^8} = 2^{-4} \text{ bloques en MM}$$

$$MM = 2^{20} \text{ bytes}$$

palabras de 2^3 bytes

$$C = \frac{2^4}{2^2} = 2^3 = 8 \text{ conjuntos}$$

TAG	C	W
8	8	

dir. Mem	Nº bloque	Mod 8	Conjunto	iter: 2
329F0h	203248	793	C2	F A
32AF0h	203504	794	C2	F A
7023Ch	489068	1793	C2	F A
77777h	489335	1911	C7	F A
44037h	278583	1088	C0	F F
778D6h	489694	1912	C0	F F
A5022h	675873	2640	C0	F F

- b) N° fallos en directa aumenta, en totalmente asociativa disminuye
 En directa, búsqueda más rápida, en totalmente asociativa más lento

Ejercicio 5

Caché Datos

$$MP = 2^4 \text{ Mbytes} = 2^{24} \text{ bytes}$$

$$MC = 2^3 \text{ kb} = 2^{13} \text{ bytes}$$

Bucle 100 iteraciones

Caché asociativa (2-was)

$$\text{Número conjuntos} = 2^7 \text{ conjuntos}$$

a)

$$\text{Nº conjuntos} = 2^7$$

Un ordenador con el que podrás jugar como mi ex jugó conmigo.



Para que el futuro nos sea más benevolente debemos empezar hoy a perseguirlo, creando para nosotros un camino que pueda llevarnos hacia el lugar donde queramos estar el día de mañana, aprovecha la serie Pulse GL76 y viaja al futuro.



$$\text{Bloques en MC} = N^{\circ} \text{vías} \cdot N^{\circ} \text{conjuntos} = 2^2 \cdot 2^3 = 2^5 \text{ bloques}$$

$$\text{Tamaño de bloque} = \frac{\text{Tamaño MC}}{\text{Bloques}} = \frac{2^{13}}{2^5} = 2^5 \text{ bytes} = 5 \text{ bytes}$$

$$\text{Tamaño TAG} = 12 \text{ bits}$$

Reemplazo LRU ó FIFO

6)

Carrot C4

$$- MC = 2^4 \text{ kb} = 2^{14} \text{ bytes}$$

$$- MP = 2^4 \text{ Mbytes} = 2^{24} \text{ bytes}$$

- Caché asociativa (4-vías)

$$- N^{\circ} \text{vías} \cdot N^{\circ} \text{conjuntos} = 2^2 \cdot 2^3 = 2^5 \text{ conjuntos}$$

$$- Bloques en MC: N^{\circ} \text{vías} \cdot N^{\circ} \text{c} = 2^2 \cdot 2^3 = 2^5 \text{ bloques}$$

$$- Tamaño bloque: \frac{7 \cdot MC}{2^5} = \frac{2^{14}}{2^5} = 2^5 \text{ bytes}$$

000A40 = 101001000000

004A40 = 0100120100100000

008A40 = 1000101001000000

00BA40 = 1011101001000000

Bloque MP	Número Conjunto	
82	82	F A
594	82	F A
1106	82	F A
1490	82	F A

Grape G5

$$- MC = 2^5 \text{ kb} = 2^{15} \text{ bytes}$$

$$- MP = 2^4 \text{ Mbytes} = 2^{24} \text{ bytes}$$

- Caché asociativa (2-vías)

$$- N^{\circ} \text{vías} \cdot N^{\circ} \text{conjuntos} = 2^2 \cdot 2^2 = 2^4 \text{ conjuntos}$$

$$- Bloques en MC = 2^2 \cdot 2^2 = 2^{10} \text{ bloques}$$

$$- Tamaño bloque: \frac{2^{15}}{2^{10}} = 2^5 \text{ bytes}$$

000A40 = 101001000000

Bloque MP	Número Conjunto	
82	82	F F

004A40	= 0100101001000000	594	82	F	F
008A40	= 1000101002000000	1106	82	F	F
00BA40	= 1012101002000000	1490	466	F	A

Ejercicio 6

6. Considérese un procesador con instrucciones de 32 bits y una caché de instrucciones de 32 bytes con líneas de 8 bytes. Para los dos fragmentos de código MIPS siguientes, indicar la organización (y sus parámetros) que da lugar a una ejecución con el mínimo número de fallos (si hay varias que produzcan el mismo rendimiento, ordenar por coste hardware). NOTA: El número a la izquierda de cada instrucción indica la dirección en memoria principal donde se encuentra.

0 lw \$1, 100(\$2)	0 lw \$1, 100(\$2)
4 add \$1, \$1, \$3	4 add \$1, \$1, \$3
8 sub \$4, \$5, \$1	8 sub \$4, \$5, \$1
12 j 8	12 j 8
...	...
24 mul \$2, \$2, \$8	24 mul \$2, \$2, \$8
28 sub \$2, \$2, \$9	28 sub \$2, \$2, \$9
32 div \$8, \$1, \$4	32 add \$8,
36 j 0	36 j 6

Instrucciones de 32 bits = 4 bytes

Caché de 32 bytes = 2^5 bytes

Bloques de 8 bytes = 2^3 bytes

Ejercicio 7

7. Sea el siguiente programa en MIPS:

```

addi $4, $0, 1
j lab1
lab0: lw $1, 16($0)
sw $1, 80($0)
j lab0
lab1: sw $0, 16($0)
lw $0, 20($0)
lw $3, 33($0)
sw $2, 40($0)
bne $3, $0, lab0

```

donde la dirección lab0 es igual a 8h y lab1 es igual a 140h. Hay una cache de instrucciones, con asignación directa, y otra cache de datos asociativa por conjuntos (con 4 conjuntos en total). Ambas caches tienen bloques de 8 bytes y un tamaño de 128 bytes. Se pide:

- a) Mostrar la secuencia de referencias que hace el programa durante su ejecución. Distingue entre referencias a instrucciones y referencias a datos, y basta con que indiques las 2 primeras iteraciones.

Instrucciones	Datos

- b) Describe cada una de las caches, indicando el tamaño tanto de la zona de control como de los datos.

- c) Muestra la evolución de cada una de las caches, indicando para cada referencia si se produce un fallo o un acierto. Calcula las tasas de fallo.

Caché de instrucciones - Directa

Caché de datos - Asociativa por conjuntos

Bloques de 8 bytes = 2^3 bytes

$$MC = 228 \text{ bytes} = 2^7 \text{ bytes}$$

Instrucciones

Datos : 26, 80, 208, 336, 400

```

0h add $4, $0, 1
4h j lab1
8h lab0: lw $1, 16($0)
12h sw $1, 80($0)
16h j lab1
...
240h lab1: sw $0, 16($0)
244h lw $2, 208($0)
248h lw $3, 336($0)
252h sw $2, 400($0)
256h bne $4, $0, lab0

```

Cache: Instrucciones

$$\text{Bloques } MC = \frac{2^7}{2^3} = 2^4$$

TAG	Bloque	MC	W
		4	3

	Bloque	MP	Mod	Por Cache
0h = -----0000	0	0	F	
4h = -----0100	0	0	A	
240h = 000201000000	40	8	F	
244h = 000201000100	40	8	A	
248h = 000201000100	41	9	F	
24Ch = 0002010001100	41	9	A	
250h = 0002010001000	42	10	F	
8h = -----1000	1	1	F	
Ch = -----1100	2	2	F	
10h = -----00020010	2	2	A	
240h = 000201000000	40	8	A	
244h = 0002010000100	40	8	A	
248h = 000201000100	41	9	A	
24Ch = 0002010001100	41	9	A	
250h = 0002010001000	42	10	A	

Indice falso; $\frac{6}{15}$

Caché datos

Conjuntos = $2^2 = 4$ conjuntos

Bloques por conjunto: $2^4 : 2^2 = 2^2$ bloques por conjunto

14G	C	w
	2	3

		Mod 4	
	: 8 Bloque MP		Conjunto
16	2	2	F
208	26	2	F
336	42	2	F
400	50	2	F
16	2	2	A
80	10	2	F
16	2	2	A
208	26	2	F
336	42	2	F
400	50	2	F

Índice fallo: $\frac{8}{10}$

Ejercicio 8

8. Sea el siguiente programa en MIPS:

```

ori $2,$0,1000h
loop:    lw $1,2800h($2)
sub $4,$1,$0          rotar: add $10,$4,$4
jal rotar           mul $7,$10,2
sw $7,7800h($2)      jr $31
sw $1, C800h($2)
sub $2,$2,4
bne $2,$0,loop

```

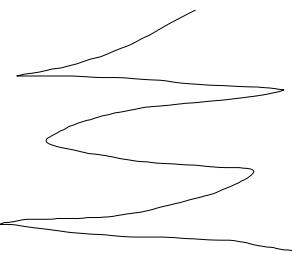
Se desea usar una memoria cache de tamaño 4 Kbytes, con asignación directa, para resolver las referencias a datos. Se pide:

- ¿Cuál es el tamaño de bloque óptimo teniendo en cuenta la tasa de fallos? Prueba distintos tamaños de bloque, empezando con 1 palabra por bloque, y calcula sus tasas de fallo.
- Si el tamaño de bloque está fijado en 256 bytes, ¿es posible reducir la tasa de fallos aumentando el nivel de asociatividad?
- ¿A partir de qué nivel de asociatividad no se obtiene mejora?
- ¿Qué política de reemplazo es mejor, LRU o FIFO?

$$MC = 2^{12} \text{ bytes}$$

Memoria directa

Si tu ordenador tiene más años que los que dura tu carrera: toca cambio.



Ejercicio 9

En la figura se proponen dos códigos en alto nivel que inicializan ambos una matriz de 20x20 números reales a cero. El código generado por el compilador almacena la matriz en la memoria en posiciones consecutivas de RAM. Las variables *i*, *j* son declaradas en el registro dinámico del procesador (RD) para memoria.

```

    i=0
    j=0
    while (i<20) do
        j=0
        while (j<20) do
            A[i][j]=0.0
            j=j+1
        end while
        i=i+1
    end while

```

Código A

```

    i=0
    j=0
    while (i<20) do
        j=0
        while (j<20) do
            A[j][i]=0.0
            j=j+1
        end while
        i=i+1
    end while

```

Código B

Se dispone de un sistema de memoria cache cuya cache es completamente asociativa y usa reemplazo FIFO. El tamaño de la cache es equivalente a diez números reales en la representación usada por el procesador y

Estructura de Computadores

permite almacenar diez bloques por página. Ambos códigos realizan la misma función, pero dadas al punto de vista del sistema de memoria el rendimiento es diferente. Analiza cuál de ellos daría más fallos de cache.

Caché totalmente asociativa

Reemplazo FIFO

$MC = 10 \text{ bytes} \Rightarrow$ bloques de 2 palabras $\rightarrow 5$ bloques en caché

Código 2 (FIFO)

	Nº Bloque	Caché
1	0	F
2	1	F
3	2	A
4	2	F
5	2	A
6	3	F
7	3	A
8	4	F
9	4	A



10	S	-	F
11	S	-	A

$$Z. Fallos = \frac{1 + 50}{100} = 51/100$$

Código 2 (FIFO)
Nº Bloque Pos cache

2	0	-	F
11	S	-	F
22	10	-	F
32	15	-	F
42	20	-	F
52	25	-	F
62	30	-	F
72	35	-	F
82	40	-	F
91	45	-	F
2	1	-	F

$$Z. Fallo = 1$$

Ejercicio 10

- Cache : 8 bytes
- Tamaño bloq.: 2 bytes

a)

Directa

Dirección MP $\xrightarrow{\wedge 2}$ Bloque MP $\xrightarrow{\text{mod } 4}$ Pos. Cache

A(0)	0	0	0	F
A(1)	1	0	0	A
A(2)	2	1	1	F
	3	1	1	A
4	2	2	2	F
5	2	2	2	A
6	3	3	3	F
7	3	3	3	A

8

4

0 F

9

4

0 A

$$\text{L.Fallo: } \frac{5}{10}$$

Totalmente asociativa: $\frac{5}{10}$ (indice fallo)

Asociativa por conjuntos: $\frac{5}{10}$ (indice fallo)

Cogeman la directa por menor conte hardware

- b) Los indices de fallo serían iguales,
pero sería diferente

c)

Directa

Dirección MP $\xrightarrow{12}$ Bloque MP $\xrightarrow{\text{mod } 4}$ Pon. c

69	34	2 F
0	0	0 F
68	34	2 A
1	0	0 A
67	33	2 F
2	2	2 F
66	33	2 F
3	2	2 P
65	32	0 F
4	2	2 A
64	32	0 F
5	2	2 A
63	31	3 F
6	3	3 F
62	31	3 F
7	3	3 P
..	.	- -

62	30	2 F
8	4	0 A
60	30	2 F
9	4	0 A

Indice Fallos: 14120

Totalmente asociativa: 10120 (2. fallos)

Asociativa por conjuntos: 10120 (2. fallos)

d)

Ejercicio 21

11. Sea un sistema de memoria de 1 MByte, con palabras de 1 byte, que incorpora una memoria caché de 64 KBytes, con 4096 palabras por bloque y algoritmo de reemplazo FIFO. Se ejecuta un programa que referencia a la siguiente secuencia de direcciones (en hexadecimal):

00000h, 01000h, 01001h, 0F000h, 10000h, 00000h, 40000h, 20000h, 08000h

- a) Compara las tasas de fallo cuando se considera una organización con asignación directa, totalmente asociativa, y asociativa con 4 conjuntos.
- b) Calcula el tamaño de cada una de las caches incluyendo la zona de directorio.

$$MM = 2^{20} \text{ bytes}$$

palabras de 2^0 bytes

$$MC = 2^6 \text{ kb} = 2^10 \text{ bytes}$$

$$\text{bloques de } 2^{12} \text{ palabras} = 2^{12} \text{ bytes}$$

FIFO



Directa

Bloque MP Pos cache

00000h = - - - - -	0	0	F
01000h =1000100000000000	2	2	F
01001h =1000100000000001	1	1	A
0F000h =1111000000000000	15	15	F
10000h = 0001000000000000	11	11	F

que esta Navidad lo único que suspendas sean planes



Tu ordenador lo único que necesita programar es su jubilación.



20000 h	= 00001000000000000000000000000000	16	0	0	F
40000 h	= 01000000000000000000000000000000	64	0	0	F
20000 h	= 00100000000000000000000000000000	32	0	0	F
08000 h	= 00001000000000000000000000000000	8	8	8	F

L.Fallo: $\frac{8}{9}$



El Stealth 15M es uno de los portátiles gaming más finos y ligeros. Siempre menos es más. Ve a donde quieras llevando siempre el máximo rendimiento.



T. asociativa

	Bloque MP	Pos cache	
00000h =	0	-	F
02000 h = 0001000000000000	2	-	F
01001 h = 0001000000000001	1	-	A
0F000 h = 1110000000000000	15	-	F
10000 h = 0001000000000000	16	-	F
00000 h =	0	-	A
40000 h = 0100000000000000	64	-	F
20000 h = 0010000000000000	32	-	F
08000 h = 0000100000000000	8	-	F

L.Fallo: $\frac{6}{9}$

A. por Conjuntos (4 conjuntos)

1AG	C	W
	2	12

Bloques por conjunto: $\frac{2^4}{2^2} = 2^2$ bloques

Bloque MP Pos cache

WUOLAH



00000h	=	0	F
01000h	=	... 00 01 0000 00000000	2	F
01001h	= 00 01 000000000001	2	A
0F000h	= 1111 000000000000	15	F
10000h	=	000100 00 000000000000	16	F
00000h	= 0	0	A
40000h	=	01000000 000000000000	64	F
20000h	=	00100000 000000000000	32	F
08000h	=	00001000 000000000000	8	F

2. Fallo: $\frac{7}{9}$

Ejercicio 12

12. Considerar una memoria principal de 64 Kbytes direccionable a nivel de byte y con palabras de este mismo tamaño. Sea la siguiente secuencia de direcciones de acceso a memoria principal:
04F5h, 11E0h, 1500h, 2000h, 241Fh, 16F7h, 1233h, 21F0h

Departamento de Arquitectura de Computadores, Universidad de Málaga

5

Problemas Tema 3

Mostrar el contenido del directorio cache, así como los fallos que se producen para cada una de las caches descritas en la siguiente tabla:

	Tamaño Cache	Tamaño Bloque	Organización	Reemplazo
C1	4 Kb	256 bytes	Directa	---
C2	2 Kb	256 bytes	Totalmente asociativa	FIFO
C3	2 Kb	256 bytes	Asociativa por conjuntos de 4 vías	LRU

$$MP = 2^6 \text{ Kb} = 2^{16} \text{ bytes}$$

Cache 1 (Directa)

$$MC = 2^2 \text{ Kb} = 2^{12} \text{ bytes} > 2^4 \text{ bloques en cache}$$

Bloques de 2^8 bytes

TAG	Bloque MP	W
4	8	

| Bloque MP | Cache |

04F5h = 200111101012	4	4	F
11E0h = 1000111100000	17	2	F
1500h = 1010100000000	21	5	F
2000h = 1000000000000	32	0	F
241Fh = 10010000011112	36	4	F
16FFh = 1011011111122	22	6	F
1233h = 1001000110022	18	2	F
21F0h = 1000111110000	33	1	F

I. Fallo = 2

Caché 2 (T. asociativa)

$$MC = 2^2 \text{ Kf} = 2^{12} \text{ bytes} \\ \text{Bloques de } 2^8 \text{ bytes} \rightarrow 2^3 \text{ bloques} \\ - \text{ FIFO}$$



	Bloque MP	Caché
04F5h = 200111101012	4	-
11E0h = 1000111100000	17	-
1500h = 1010100000000	21	-
2000h = 1000000000000	32	-
241Fh = 10010000011112	36	-
16FFh = 1011011111122	22	-
1233h = 1001000110022	18	-
21F0h = 1000111110000	33	-

I. Fallo = 2

Caché 3 (A. por Conjuntos)

$$MC = 2^{11} \text{ bytes} \\ \text{Bloques de } 2^8 \text{ bytes} \rightarrow 2^3 \text{ bloques en MC} \\ 4 \text{ ways}, 3$$

sacarte una carrera es una montaña rusa emocional, así que disfruta del trayecto

$$\text{Conjunto} : \frac{2^2}{2^2} = 2^1 = 2$$

TAG	C	W
2		8

	Bloque MP	Conjunto	
04F5h = 2010121100100101	4	0	F
11E0h = 1000101110000000	17	2	F
1500h = 1010101000000000	21	2	F
2000h = 1000000000000000	32	0	F
241Fh = 1001000000111111	36	0	F
16FFh = 1011011111111111	22	0	F
1233h = 10010001100111	18	0	F
22F0h = 10000111110000	33	2	F

1. Fallo = 1

Ejercicio 23

13. En aplicaciones multimedia para la reproducción de audio o video es habitual tener cargas de trabajo denominadas *streaming*, es decir, cargas de trabajo con una gran cantidad de datos que apenas son reusados. Los datos de audio o de video suelen ser *streams* con datos de tamaño igual a 1 byte que codifican por ejemplo un nivel de gris o de cuantificación de una onda sonora. Considera un *stream* de video que accede a 512 KB de datos secuencialmente:

0, 2, 4, 6, 8, 10, 12, 14, 16...

- a) Considera una cache de 64 Kb con asignación directa y bloques de tamaño 32 bytes. ¿Cuál es la tasa de fallo? ¿Depende del tamaño de la cache o del tamaño de los datos a los que se accede? Usando el modelo de las 3C, ¿qué tipo de fallos se producen?

- b) Recalcula las tasas de fallo considerando bloques de tamaño 16 bytes, 64 bytes y 128 bytes. ¿Qué tipo de localidad se explota en este tipo de trabajo?

La técnica de prefetching (precarga) permite mejorar el acceso a datos cuando el patrón de acceso es predecible. Esta técnica se basa en trae, de forma especulativa, un bloque al que todavía no se ha accedido; por ejemplo, una implementación sencilla consistiría en traer, a un buffer separado, el siguiente bloque adyacente al bloque que se está accediendo. Si el dato que se busca se encuentra en ese buffer se considera un acierto, se introduce en la cache, y se precarga el siguiente bloque en el buffer.

- c) Considerando que se dispone de un buffer de precarga con capacidad para dos bloques y que la latencia de la cache es menor o igual que el tiempo que necesita nuestro programa para procesar los datos de un bloque de la cache. ¿Cuál será la tasa de fallos para el stream anterior?

a)

$$\begin{aligned} \text{MP} &= 2^{\alpha} \text{ Kb} = 2^{14} \text{ bytes} \\ \text{MC} &= 2^{\beta} \text{ Kb} = 2^{11} \text{ bytes} \\ \text{bloques de } 2^5 \text{ bytes} &= 32 \end{aligned}$$

} 2^{14} \text{ bloques en MP}

} 2^{11} \text{ bloques en MC}

Directa

EXISTIMOS
ÚNICAMENTE
PARA AYUDARTE!



:32	Bloque MP	Pos. Cache
0	0	0
2	0	0
4	0	0
6	0	0
8	0	0
10	0	0
12	0	0
14	0	0
16	0	0
...
32	2	0

- Z. Fallo: $\frac{2^{14}}{2^{19}} = \frac{1}{32}$ (De todas las referencias a memoria, se produce un fallo por cada bloque (Ya que la primera vez, no se encuentra en cache))

- Depende del tamaño de los datos
- Segun el modelo de las 3C's, se producen "fallo obligatorio" (Compulsory miss) ya que se produce cuando accedemos por primera vez a un bloque.

b)

Tamaño de bloque 2^4 bytes
 $MP = 2^{15}$ bytes $\rightarrow 2^{15}$ bloques en MP

:16	Bloque MP	Pos. Cache	
0	0	0	F
2	0	0	A
4	0	0	A
6	0	0	A
8	0	0	A
...

10	0	0	A
12	0	0	A
14	0	0	A
16	1	0	F

$$Z_{\text{fallo}}: \frac{2^{15}}{2^{19}} = \frac{1}{16}$$

- Tamaño bloque de 2^6 bytes $\rightarrow Z_{\text{fallo}} = \frac{2^{13}}{2^{19}} = \frac{1}{64}$
- Tamaño bloque de 2^7 bytes $\rightarrow Z_{\text{fallo}} = \frac{2^{12}}{2^{19}} = \frac{1}{128}$

De todas las referencias a memoria, se produce un fallo por cada bloque (ya que la primera vez no se encuentra en caché)

c)

$$Z_{\text{fallo}} = \frac{1}{2^{16}} \approx 0\%$$

Ejercicio 14

14. El siguiente código en alto nivel:

```
for (i=0; i<2; i++)
    for (j=0; j<2; j++)
    {
        ac = C[i*j*2];
        for (k=0; k<2; k++) ac += A[i+k*2]*B[k+j*2];
        C[i*j*2] = ac;
    }
```

se ejecuta en un procesador que dispone de una memoria principal de 1GB con tamaño de palabra de 4 bytes, y con tamaño de línea de 8 bytes. El array C se almacena a partir de la dirección 0 de memoria principal, el array A, a partir de la dirección 20 (0x14) y el array B, a partir de la dirección 40 (0x28) (todos los

números están expresados en DECIMAL (hexadecimal)). Se sabe también que cada elemento del array ocupa 4 bytes en memoria. Se pide lo siguiente:

a) Mostrar los accesos a los elementos de los array, junto con su dirección de memoria principal y su número de bloque. Para ello, rellenad una tabla como la de abajo indicando en la columna "dato" el elemento concreto del array que se accede (p.ej. A[3]), la dirección de memoria principal donde se encuentra dicho elemento (dir MP), y número de bloque de memoria principal (bloq MP). Mostradlo en el mismo orden que se solicita en el programa teniendo en cuenta que en cada iteración del tercer bucle FOR, el orden de acceso a los arrays es: B[k+j*2], A[i+k*2].

b) Suponiendo que disponemos de una cache de datos, asociativa por conjuntos de 2 vías (2 bloques por conjunto), reemplazo LRU y tamaño 32 bytes, muestra la evolución del directorio de la cache para dicho código. Para ello, rellena en la tabla la columna "CONJ" con el número del conjunto de la caché donde dicho bloque va a ser alojado, e indica en "A/F" si se produce acierto (A) o fallo (F) de caché y un comentario opcional de lo que ocurre. Muestra también como evoluciona el directorio de la cache. Finalmente, calcula el Indice de Fallos (exprésalo en forma fraccionaria).

dato	dir MP	bloq MP	CONJ	A/F

$$MM = 2^{30} \text{ bytes}$$

palabras de 2^2 bytes

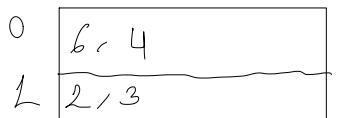
$$\text{Bloques de } 2^3 \text{ bytes} = 8 \text{ bytes} \rightarrow 2^2 \text{ bloques MC}$$

$MC = 2^5 \text{ bytes}$

Asociativa 2 vías $\rightarrow 2 \text{ conjuntos}$

2 RV

dato	dir MP	bloq MP	CONJ	A/F
C[0]	0	0	0	F
B[0]	40	5	1	F
A[0]	20	2	0	F
B[1]	44	5	1	A
A[2]	28	3	1	F
C[0]	0	0	0	A
C[2]	8	1	1	F
B[2]	48	6	0	F
A[0]	20	2	0	F
B[3]	52	6	0	A
A[2]	28	3	2	A
C[2]	8	1	2	A
C[1]	4	0	0	F
B[0]	40	5	1	F
A[1]	24	3	1	F
B[1]	44	5	1	A
A[3]	32	4	0	F
C[4]	4	0	0	A
C[3]	12	2	2	F
B[2]	48	6	0	F
A[1]	24	3	2	F
B[3]	52	6	0	A
A[3]	32	4	0	F
C[5]	12	2	1	A



I. Fallo: $\frac{15}{24}$

Ejercicio 15

15. El siguiente código en alto nivel:

```
FOR (i=1; j<=2; j++) A[i]=j;
FOR (j=1; j<=3; j++) B[j+1]=2*j;
FOR (j=2; j<=4; j++) A[j+1]=A[j-1]-B[j];
```

Se ejecuta en un procesador que dispone de una memoria principal de 1GB con tamaño de palabra de 1 byte y con tamaño de bloque de 2 bytes. El array A se almacena a partir de la dirección 0 de memoria principal y el array B a partir de la dirección 32 (ambos números están expresados en DECIMAL). Se sabe también que cada elemento del array ocupa 1 byte en memoria.

Se pide lo siguiente:

- a) Mostrar los accesos a los elementos de los array, junto con su dirección de memoria principal (MP) y su número de línea (bloque). Mostrand los resultados en una tabla similar a la del ejercicio

anterior, mostrando el elemento concreto del array que se accede, la dirección de memoria principal donde se encuentra dicho elemento y el número de bloque al que pertenece. Mostrarlo en el mismo orden que se solicita en el programa teniendo en cuenta que en cada iteración del tercer bucle FOR, el orden de acceso a los arrays es: B[J], A[J-1], A[J+1].

b) Suponiendo que disponemos de una caché de datos, asociativa por conjuntos de 2 vías (2 bloques por conjunto), reemplazo LRU y tamaño 8 bytes, muestra la evolución del directorio de la cache para dicho código. Para ello, incluye en la tabla anterior dos columnas adicionales: una con el número del conjunto de la caché donde dicho bloque va a ser alojado y otra si se produce acierto (A) o fallo (F) de caché. Complétalo con un comentario adicional de lo que va ocurriendo en cada referencia. Muestra también el estado final en el que queda el directorio de la caché. Finalmente, calcula el Índice de Fallos.

$$MM = 2^{30} \text{ bytes}$$

Palabras de 2^6 bytes

$$\begin{aligned} \text{Bloque de } 2^2 \text{ bytes} &= 2 \text{ bytes} \\ MC = 2^3 \text{ bytes} &\rightarrow 2^2 \text{ bloques en MC} \\ &\rightarrow 2 \text{ conjuntos} \\ \text{Asociativa - 2 vías} & \end{aligned}$$

LRU

dato	dir MP	bloq MP	CONJ	A/F
A[1]	1	0	0	F
A[2]	2	1	1	F
B[2]	34	17	1	F
B[3]	35	17	1	A
B[4]	36	18	0	F
B[5]	34	17	1	A
A[3]	2	0	0	A
A[3]	3	1	1	A
B[3]	35	17	1	A
A[2]	2	1	1	A
A[4]	4	2	0	F
B[4]	36	18	0	F
A[3]	3	1	1	A
A[5]	5	2	0	A

$$\text{I. Fallo: } \frac{6}{14} = \frac{3}{7}$$

Ejercicio 16

16. Calcula el tiempo medio de acceso a memoria (AMAT) en nseg. de un sistema caché donde la frecuencia de reloj es de 500 MHz, el tiempo de acierto es de 1 ciclo de reloj, la penalización por fallo de 20 ciclos, y donde el índice de aciertos es del 95%.

$$\text{Frecuencia} = 500 \text{ MHz} = 5 \cdot 10^8 \text{ Hz}$$

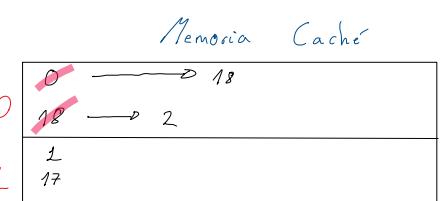
$$T_{acierto} = 1 \text{ ciclo}$$

$$\text{Penalización fallo} = 20 \text{ ciclos}$$

$$\text{Índice aciertos} = 95\%$$

$$\text{Tiempo medio de acceso a memoria (AMAT)} = \frac{\text{Total Memoria}}{N^{\circ} \text{ ref memoria}} = t_{acierto} + \text{Índice fallo} \cdot \text{Penalización por fallo} (\text{nsg})$$

que el último empujón no sea por un barranco



NEW YORK A UN SALTO

La ciudad que nunca duerme

365 DÍAS POR
89,99 €
¡SOLO AHORA!



ELIGE DESTINO
Y JUÉGALO EN

PlayStation Plus
PREMIUM



110,99 €

¡SOLO AHORA!
89,99 €
AL AÑO

DEL 12 AL 20 DE DICIEMBRE

Marvel's
Spider-Man y
cientos de juegos
para viajar donde
quieras.

Consigue esta oferta con las Tarjetas
Regalo PlayStation en



TE FALTA FNAC | FNAC.ES

Promoción válida en PlayStation
Store hasta el 20 de diciembre

$$AMAT = 2 + 0.05 \cdot 20 = 2 \text{ ciclos}$$

$$t_{\text{ciclo}} = \frac{1}{F} = \frac{1}{5 \cdot 10^9} = 2 \cdot 10^{-10} \text{ seg}$$

$$AMAT (\text{en ns}) = 2 \cdot 2 \cdot 10^{-10} \text{ seg} = 4 \cdot 10^{-9} \text{ seg} = 4 \text{ ns}$$

Ejercicio 17

17. Consider a benchmark which executes 1000 instructions in a 1.2 GHz processor, where 330 instructions are load/store. In this program, the CPI for a perfect cache is 1.8 (perfect cache: no any miss). Now we consider the stalls due to an actual cache with write-through policy, so that the miss rate for the instruction cache (IS) is 2% and for the data cache (DS) is 8%. The miss penalty is 10 cycles for I-C and 15 cycles for D-C. Calculate:

- a) Memory-stall cycles Sol. 596
- b) Effective CPI (CPI_{efectivo}) Sol.2.396
- c) Execution time of the program (CPU time) Sol.1.996 μs
- d) The average Access time (AMAT) (hit time: 1 cc) Sol.2 ns.

$$\text{Frecuencia} = 1.2 \text{ GHz} = 1.2 \cdot 10^9 \text{ Hz}$$

$$1000 \text{ instrucciones} \rightarrow 330 \text{ instrucciones } l_w/l_w = 33\%$$

$$\text{Indice fallo (Instrucciones)} = 2\% \rightarrow P_{\text{fallo}} = 10 \text{ ciclos}$$

$$\text{Indice fallo (Datos)} = 8\% \rightarrow P_{\text{fallo}} = 15 \text{ ciclos}$$

$$\text{a)} 1000 \cdot 0.02 \cdot 10 + 330 \cdot 0.08 \cdot 15 = 596$$

b)

$$CPI_{\text{efectivo}} = CPI_{\text{base}} + CPI_{\text{extra}} = CPI_{\text{base}} +$$

$$\frac{N_{\text{acceso MEM-Instrucciones}} \cdot I_{\text{fallo-Instrucciones}} \cdot P_{\text{fallo-Instrucciones}}}{NZ} + \frac{N_{\text{acceso MEM-Datos}} \cdot I_{\text{fallo-Datos}} \cdot P_{\text{fallo-Datos}}}{NZ} =$$

$$CPI_{\text{efectivo}} = 1.8 + \frac{2000 \cdot 0.02 \cdot 10}{1000} +$$

$$\frac{330 \cdot 0.08 \cdot 15}{1000} = 2.396$$

c)

$$t_{\text{cpu}} = NZ \cdot CPI \cdot T_{\text{ciclo}}$$

$$t_{\text{ciclo}} = \frac{1}{F} = \frac{1}{1.2 \cdot 10^9} = 8.3 \cdot 10^{-10} \text{ s}$$

$$t_{\text{cpu}} = 1000 \cdot 2.396 \cdot \frac{1}{1.2 \cdot 10^9} = 1.9966 \cdot 10^{-6} \text{ seg} = 1.9966 \text{ ms}$$

Ejercicio 18

Reservados todos los derechos.

No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

WUOLAH

18. Consider a processor with write-through policy with an IS with a miss rate of 0.5%, a D\$ with miss rate 1.5%, miss penalty for IS of 10 cycles and 15 for the D\$. The base CPI (that is, for a perfect cache) is 1.7, and there are a 32% of load&stores in the code. Calculate the efective CPI and the AMAT.

Sol. CPI=1,822 AMAT=1,122

Indice fallo (Instrucciones) = 0.5% $\rightarrow P_{fallo} = 10$ ciclos

Indice fallo (Dato) = 1.5% $\rightarrow P_{fallo} = 15$ ciclos

32% operaciones Lw/Sw

$$CPI = CPI_{base} + CPI_{extra} = 1.7 + 0.005 \cdot 10 + 0.015 \cdot 15 \cdot 0.32 = 1.822$$

\rightarrow (Se supone un ciclo)

AMAT: $t_{acerto} + Indice\ fallo \cdot Penalizaci\'on\ por\ fallo$ (sg)

$$AMAT = 1 + 0.005 \cdot 10 + 2 \cdot 0.015 \cdot 15 \cdot 0.32 = 2.220$$

Ejercicio 19

19. Consider a program that executes 2150 instructions in a 1.5 GHz processor which has implemented all the bypass technique possible. We know that 20% of the instructions are conditional branches (30% of them are effective), 5% are unconditional branches, 15% are load instructions followed by an instruction with direct data dependency, 10% are load with no data dependency and 8% are stores (with write-through policy). Furthermore, the hit rate for the I-C is 99.4% (with 12 cc of miss penalty) and for the D-C is 98% (with a miss penalty of 15 cc).

The processor follows the branch not taken prediction technique, where the update of the PC is carried out in the mem stage for conditional branches and in the ID stage for the unconditional branches. Calculate the CPI

CPI_{base}=1,38 CPI_{efec}=1,971

$$Frecuencia = 1.5 \text{ GHz} = 1.5 \cdot 10^9 \text{ Hz}$$

2150 instrucciones

20% Saltos condicionales (80% saltan)

5% Saltos incondicionales

15% Lw seguidos de dependencia de dato

10% Lw sin dependencia

8% Sw

Indice acierto (Instrucciones) = 99.4% $\rightarrow P_{fallo} = 12$ ciclos

Indice acierto (Dato) = 98% $\rightarrow P_{fallo} = 15$ ciclos

Salto no tomado

- MEM en Condicionales

- ID en incondicionales

- Saltos condicionales

IF JO GX M | WB

IF ID EX | :
Puedo 3 ciclos cuando salta

→ Saltos incondicionales

IF ID EX M WB
~~IF IF~~

Puedo 2 ciclo siempre

→ 2w con dependencia

IF ID EX M WB
IF ID EX M WB

Puedo 2 ciclo

→ 33% 1w / sw

$$CPI_{base} = 1 + 0.2 \cdot 0.3 \cdot 3 + 0.05 \cdot 1 + 0.15 \cdot 1 = 2.38$$

↓ CPI efectivo: $2.38 + 0.006 \cdot 12 + 0.02 \cdot 15 \cdot 0.33 = 2.552$? Esta bien?

Ejercicio 20

20. Consider a two-level cache system L1, L2. The miss rate for L1 is 3% and for L2 is 5%. Calculate:

- Global miss rate of the memory system Sol. 0.0015
- If the total number of references is 12,000, calculate the number of
 - misses of L1 Sol. 360
 - references to L2 Sol. 360
 - misses of L2 Sol. 18
- Assuming that the hit time for L1 is 1 cc (as usual), the hit time for L2 is 10 cc and the miss penalty for L2 is 50 cc, calculate the AMAT
Sol. 1.375

$$\text{Indice fallo L1} = 0.03$$

$$\text{Indice fallo L2} = 0.05$$

a)

$$\text{Global miss rate} = 0.03 \cdot 0.05 = 2.5 \cdot 10^{-3}$$

b)

Referencias a memoria → 12000

v

i) Fallas en L1 = $12000 \cdot 0.03 = 360$ fallas

ii) Referencias a L2 = Fallas en L1 = 360 (Si falla en L1, se busca en L2)

iii)

Fallas en L2 = $360 \cdot 0.05 = 18$ fallas

c)

Tacceso L1 = 1 ciclo

Tacceso L2 = 10 ciclos $\rightarrow P.fallo = 50$ ciclos

AMAT = Tacceso + Indice fallo • Penalización por fallo (Δg)

AMAT = (HitL1 + IndiceL1 • P.falloL1) + (HitL2 + IndiceL2 • P.falloL2)

AMAT = 1 + 0.03 • (10 + 0.05 • 50) = 1.375

Ejercicio 21

21. Consider a processor working at 1.7 GHz with a two-level cache system where the hit rate for L1 is 90% and for L2 it is 75%. Taking into account that the average memory access time is 5 ns and the access time for L2 is 15 ns, calculate the miss penalty for L1 and for L2 (in both number of cycles and ns.)

Frecuencia = $1.7 \cdot 10^9$ Hz

Indice acceso L1 = 90% $\rightarrow I.fallo = 10\% = 0.1$

Indice acceso L2 = 75% $\rightarrow I.fallo = 25\% = 0.25$

AMAT = 5 ciclos = AMAT_{L2}

THit_{L2} = 15 ciclos

THit_{L2} = 2 ciclo (Suponemos)

AMAT_{L2} = Tacceso_{L2} + P.fallo_{L2} • Indice fallo_{L2}

$$P.fallo_{L2} = \frac{AMAT_{L2} - Tacceso_{L2}}{I.fallo_{L2}} = \frac{5 - 1}{0.1} = 40 \text{ ciclos} \quad 1.7 \cdot 10^9 = 2.352 \cdot 10^{-9} \text{ s} = 23.52 \text{ ns}$$

La P.fallo de L2 es igual al AMAT_{L2}, ya que la penalización por fallo es el tiempo medio en acceder a L2

Un ordenador con el que podrás jugar como mi ex jugó conmigo.



Para que el futuro nos sea más benevolente debemos empezar hoy a perseguirlo, creando para nosotros un camino que pueda llevarnos hacia el lugar donde queramos estar el día de mañana, aprovecha la serie Pulse GL76 y viaja al futuro.



$$AMAT_{Tc} = T_{Hitz} + T_{Jallos} \cdot P_{Jallos}$$

$$P_{Jallos} = \frac{40 - 15}{0.25} = 100 \text{ ciclos} / 1.7 \cdot 10^9 = 5.882 \cdot 10^{-9} \text{ seg} = 58.82 \text{ ns}$$

Ejercicio 22

$$t_{dir} = 1 \text{ ciclo}$$

$$t_{DRAM} = 20, 2, 8, \dots \text{ ciclos}$$

$$t_{DUS} = 2 \text{ ciclos}$$

$$1 \text{ bloque} = 4 \text{ palabras} = 16 \text{ bytes}$$

$$BW = \frac{\text{nº bytes}}{\text{Tiempo transacción}}$$

2 acceso a MP \rightarrow 1 palabra por modulo (4 bytes)

a)

Bus de 32 bits

$$PF = "T_{dir} + t_{DRAM} + t_{bus}"$$

$$PF = 1 \text{ ciclo} + (20 + 3 \cdot 8) + 2 = 46 \text{ ciclos}$$

$$BW = \frac{16}{46} =$$

b)

2 módulos (Metiendo 2 palabras = 8 bytes)

$$\text{i) bus de } 32 \text{ bits} = 4 \text{ bytes}$$

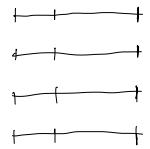
$$\text{ii) bus de } 64 \text{ bits} = 8 \text{ bytes}$$

$$(i) PF = 1 + (20 + 8) + (2 + 1) = 32 \text{ ciclos}$$

$$(ii) PF = 1 + (20 + 8) + 1 = 30 \text{ ciclos}$$

c) 4 mōdulos (4 palabras - 16 bytes)

i) $6 \times 32 \text{ bytes} = 192 \text{ bytes}$



ii) $PF = 1 + 20 + (1+1+1+1) = 25$

(ii) $PF = 1 + 20 + (1+1) = 23$

(iii) $PF = 1 + 20 + 1 = 22$