

## 2o Control Estructura de Computadores II

curso 2006-2007 Q2

Nombre: . . . . .

### Pregunta 1 (2 puntos)

Dado el siguiente código escrito en ensamblador del IA32:

```
        movl $0, %ebx
        movl $0, %esi
for:    cmpl $4000, %esi
        jge end
(a)    movl (%ebx, %esi, 4), %eax
(b)    addl %eax, 16*1024(%ebx, %esi, 4)
(c)    movl %eax, 32*1024(%ebx, %esi, 4)
        incl %esi
        jmp for
end:
```

Suponiendo una memoria cache de datos con **mapeo directo**, escritura **write through + write NO allocate** de **4 Kbytes** y **líneas de 16 bytes**, responde a las siguientes preguntas:

1) Para cada uno de los accesos (etiquetas a, b, c), indica a qué línea de la memoria cache se accede en cada una de las 13 primeras iteraciones.

	0	1	2	3	4	5	6	7	8	9	10	11	12
a													
b													
c													

2) Calcula la cantidad de **aciertos**, en todo el bucle, para la

- referencia a: . . . . .
- referencia b: . . . . .
- referencia c: . . . . .

Suponiendo una memoria cache de datos **2-asociativa**, escritura **write through + write NO allocate** de **4 Kbytes** y **líneas de 16 bytes**, responde a las siguientes preguntas:

3) Para cada uno de los accesos (etiquetas a, b, c), indica a qué conjunto de la memoria cache se accede en cada una de las 13 primeras iteraciones.

	0	1	2	3	4	5	6	7	8	9	10	11	12
a													
b													
c													

4) Calcula la cantidad de **aciertos**, en todo el bucle, para la

- referencia a: . . . . .
- referencia b: . . . . .
- referencia c: . . . . .

## Problema 2 (2 puntos)

**a )** Disponemos de un procesador de 16 bits con direcciones de 16 bits que tiene una memoria cache de datos con las siguientes características:

- 3-asociativa, con algoritmo de reemplazo LRU
- 64 bytes por línea
- 12 líneas
- política de escritura: copy back + write allocate

El contenido inicial de la memoria de etiquetas (tags) es el siguiente:

conjunto 0	DB	conjunto 1	DB	conjunto 2	DB	conjunto 3	DB
0A	1	0A	1	0A	0	0A	0
1B	1	1B	1	1B	0	1B	0
2C	1	2C	1	2C	1	2C	1

El DB=1 indica que la línea correspondiente ha sido modificada. La información de reemplazo está implícita en la posición. Las posiciones inferiores corresponden a las líneas que llevan más tiempo sin utilizarse. Las posiciones superiores corresponden a las últimas líneas utilizadas. Por ejemplo, en el conjunto 3, la línea con tag 0A es la última utilizada, y la línea con tag 2C la que lleva más tiempo sin ser utilizada.

Rellenad la siguiente tabla, indicando para cada referencia, el número de línea de MP que le corresponde, la etiqueta (TAG), a qué conjunto de MC va a parar, si es acierto o fallo (A/F), la línea reemplazada cuando proceda, el número de bytes leídos de MP (si se lee de MP) y el número de bytes escritos en MP (si se escribe en MP)

tipo	dirección (hex)	línea MP (hex)	TAG (hex)	conjunto MC	¿acierto o fallo? (A/F)	¿Línea reemplazada?	bytes escritura MP	bytes lectura MP
LECT	EC2B							
LECT	EC45							
LECT	ECA3							
LECT	ECC0							
ESCR	1B87							
LECT	000F							
ESCR	0099							
LECT	00AA							

¿Cuál es el contenido final de la Memoria Cache?

conjunto 0	DB	conjunto 1	DB	conjunto 2	DB	conjunto 3	DB
	0						
	1						

**b)** A la cache anterior le añadimos un **buffer de prefetch** de una entrada. En este buffer se hace prebúsqueda hardware de la línea **i+1** cuando se accede (tanto en acierto como en fallo) a la línea **i**, siempre que la **i+1** no esté ya en la cache o en el buffer. En este último caso, no se realiza prefetch.

**Nombre:** \_\_\_\_\_

Rellenad la siguiente tabla (mismas referencias que la anterior) indicando, para cada referencia, el número de línea de MP que le corresponde, la etiqueta (TAG), a qué conjunto de MC va a parar, si se produce acierto o fallo en la cache (A/F), el número de bytes leídos de MP hacia la MC (si se lee de MP), sin contar los que se traen al buffer, el número de bytes escritos en MP (si se escribe en MP), la línea de MP que se encuentra en el buffer (si procede), si se produce acierto o fallo (A/F) en el buffer y la línea que se prebusca de MP (si procede).

tipo	dirección (hex)	línea MP (hex)	TAG (hex)	conjunto MC	Cache ¿acierto o fallo?	Buffer ¿acierto o fallo?	bytes escritura	bytes lectura	línea actual buffer	línea prefetch buffer
LECT	EC2B									
LECT	EC45									
LECT	ECA3									
LECT	ECC0									
ESCR	1B87									
LECT	000F									
ESCR	0099									
LECT	00AA									

### Problema 3 (1 punto)

Para cada una de las siguientes afirmaciones, **indica** si son ciertas (**C**) o falsas (**F**). Hay que contestarlas todas (1 fallo: 0.5 puntos, 2 o más fallos: 0 puntos).

- ☐ En una memoria cache copy back-write no allocate, en caso de fallo de escritura el dato se trae a la cache.
- ☐ El tiempo de penalización en caso de fallo de escritura de una memoria cache write through-write allocate depende de la longitud de la línea
- ☐ Para el mismo tamaño de memoria y el mismo programa, en general una memoria cache directa copy back y write allocate tiene una tasa de fallos menor que una memoria cache directa write through y write no allocate.
- ☐ Una victim cache sólo tiene sentido si se usa en paralelo con una memoria cache directa, ya que las asociativas por conjuntos no tienen los problemas de fallos de conflicto que tienen las caches de mapeo directo.
- ☐ En una cache asociativa por conjuntos se busca simultáneamente en la memoria de etiquetas y en la de datos.

### Problema 4 (1 punto)

Para cada una de las siguientes afirmaciones, **indica** si son ciertas (**C**) o falsas (**F**). Hay que contestarlas todas (1 fallo: 0.5 puntos, 2 o más fallos: 0 puntos).

- ☐ Una memoria RAM dinámica tiene un tiempo de acceso menor que una memoria RAM estática del mismo tamaño y grado de asociatividad.
- ☐ Para escribir en una DRAM se envían dato, fila y columna en este orden.
- ☐ Una DDR2 tiene el doble de rendimiento que una DDR normal, suministrando 4 datos por ciclo en lugar de solamente 2.
- ☐ El tamaño de una línea de la memoria principal depende del entrelazado que tenga.
- ☐ Un chip de DRAM que se direcciona con  $n$  bits de fila y  $n$  bits de columna tiene  $(2n)^2$  bits de datos.

- CPI ideal = 1,25 ciclos por instrucción
- Número de referencias por instrucción (nr) = 1,5
- Tiempo de ciclo ( $T_c$ ) = 5ns
- La Memoria Principal tiene las siguientes características:
  - MP organizada en DIMMs de 8 bytes de ancho
  - Latencia de fila de los módulos de MP = 2 ciclos
  - Latencia de columna de los módulos de MP = 3 ciclos
  - Ancho de banda del bus entre MP y MC: 8 bytes por ciclo
- Caches de datos e instrucciones separadas con las siguientes características:

Característica	Memoria Cache	
	Instrucciones	Datos
Tamaño de línea	32 bytes	64 bytes
Número de referencias a memoria por instrucción (nr)	1	0.5
Tasa de fallos (m)	5 %	10 %
Tiempo de servicio en caso de acierto	1	1
Política de escritura	---	write through + write no allocate
Porcentaje de lecturas	100%	75%
Porcentaje de líneas modificadas	0 %	20%

- Acierto en lectura en la cache de instrucciones.

[illegible]

- [illegible]

- [illegible]

- [illegible]

- [illegible]

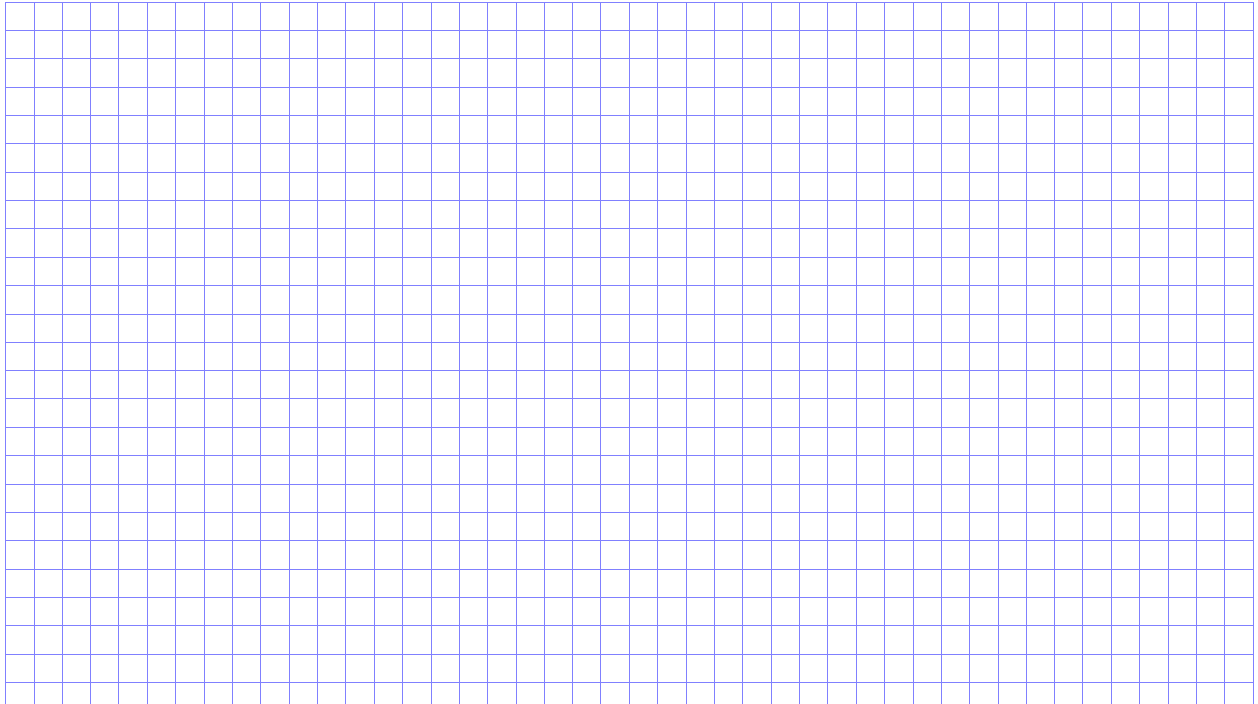
- [illegible]

- 2) **Calcula** el tiempo medio de acceso en la cache de instrucciones
- 3) **Calcula** el tiempo medio de acceso en la cache de datos
- 4) **Calcula** el tiempo de ejecución EN SEGUNDOS de 1 millón de instrucciones

[illegible]

### Problema 6 (1 punto)

Define claramente y en pocas palabras los conceptos de localidad espacial y localidad temporal.



### Problema 7 (1 punto)

Dibuja una memoria cache de mapeo directo de 4 líneas, con 8 bytes por línea, sabiendo que está en un procesador que lanza direcciones de 14 bits. Indica claramente las conexiones entre los diferentes bloques de la memoria y con los bits del bus de direcciones del procesador, así como el tamaño de la memoria de tags.

