

COGNOMS: NOM:

2on Control Arquitectura de Computadors

Curs 2014-2015 Q2

- Temps: 13:30 a 15:30

- Poseu clarament amb **LLETRES MAJÚSCULES** a cada full els cognoms i el nom

Problema 1. (3 puntos)

Se quiere diseñar la memoria cache de datos de primer nivel para un procesador de tipo Load/Store. Se barajan dos alternativas:

- (1) write through y write NO allocate.
- (2) copy back y write allocate

Se han obtenido por simulación las siguientes medidas para un determinado programa:

- porcentaje de escrituras (sobre el total de accesos): 15%
- porcentaje de bloques modificados: 10%
- tasa de aciertos caso (1): 0,9
- tasa de aciertos caso (2): 0,8

En ambos casos, la memoria cache es de mapeo directo y se leen etiquetas y datos en paralelo. En caso de fallo de lectura, el bloque de MP se escribe en la MC y posteriormente el dato se envía a la CPU desde la MC. El tiempo de acceso (T_{sa}) a memoria cache (MC) es de 10 ns tanto para lectura como escritura. El tiempo de acceso a memoria principal (MP) para escribir una palabra es de 90 ns. Para leer o escribir un bloque en la MP se emplean 130 ns.

- a) **Calcula** el tiempo empleado en realizar 1000 accesos consecutivos a datos en el caso (1)

b) **Calcula** el tiempo empleado en realizar 1000 accesos consecutivos en el caso (2)

A la cache del caso (2) le hemos añadido las siguientes mejoras: continuación anticipada (early restart) y lectura antes que escritura. Además sabemos que la latencia del siguiente nivel de memoria es de 3 ciclos, que el tamaño de bloque es de 64 bytes y el bus transfiere 8 bytes/ciclo.

c) **Dibuja** un diagrama de tiempos de las acciones a realizar para el siguiente acceso a memoria (indicando claramente el tiempo de servicio en caso de fallo y el tiempo total):

- Lectura de un byte
- Fallo con reemplazo de bloque con el Dirty Bit=1
- Fallo en el byte 30 del bloque

COGNOMS: NOM:

2on Control Arquitectura de Computadors

Curs 2014-2015 Q2

Problema 2. (3 puntos)

Dado el siguiente código escrito en ensamblador del x86:

```
    movl $0, %ebx
    movl $0, %esi
for:
    cmpl $512*1000, %esi
    jge end

    (a) movl (%ebx, %esi, 4), %eax
    (b) addl 2*8*1024(%ebx, %esi, 4), %eax
    (c) movl %eax, 4*8*1024(%ebx, %esi, 4)

    incl %esi
    jmp for
end:
```

Sabemos que se ejecuta en un sistema con memoria cache y memoria virtual. La memoria virtual utiliza páginas de tamaño 8KB y disponemos de un TLB de 4 entradas (reemplazo LRU). La memoria cache de datos (únicos accesos a memoria que contemplaremos en este problema) es Write Through + Write No Allocate, de 2 vías (reemplazo LRU), con 32 conjuntos y 8 bytes por bloque. Responde a las siguientes preguntas:

- a) **Calcula** para cada uno de los accesos (etiquetas a, b, c) el conjunto de la memoria cache al que se accede en cada una de las 9 primeras iteraciones del bucle

iteración	0	1	2	3	4	5	6	7	8
a									
b									
c									

- b) **Calcula** la cantidad de aciertos y de fallos de cache, en todo el código.

- c) Para cada uno de los accesos indicados (etiquetas a, b, c), **indica** a qué página de la memoria virtual se accede en cada una de las siguientes iteraciones del bucle (recuerda que los accesos son a 4 bytes).

iteración	0	1*512	2*512	3*512	4*512	5*512	6*512	7*512	8*512	9*512
a										
b										
c										

- d) **Calcula** la cantidad de aciertos y de fallos de TLB, en todo el código.

Sabemos que todas las direcciones de memoria son de 32 bits (tanto en las direcciones lógicas como en las físicas).

- e) **Calcula** el tamaño en bits de la memoria de datos de la cache, el de la memoria de etiquetas de la cache y también el de todo el TLB (en todos los casos podéis prescindir de los bits de estado).

COGNOMS: NOM:

2on Control Arquitectura de Computadors

Curs 2014-2015 Q2

Problema 3. (4 puntos)

En un procesador Q1 con direcciones de 32 bits, el camino crítico, y por tanto el tiempo de ciclo, está limitado por la memoria cache de datos. El tiempo (en caso de que se use) y la energía según el uso de los componentes de la memoria cache de datos se desglosa de la siguiente forma:

Componente	Tiempo	Energía
Memoria de etiquetas	0,30 ns	4 nJ/acceso
Comparación de etiquetas y (en caso necesario) selección de vía	0,20 ns	0,50 nJ/acceso
Memoria de datos y selección de byte de la línea	0,50 ns	24 nJ/acceso
Mux de vía de datos: selecciona el dato de la vía correspondiente (cuando sea necesario)	0,10 ns	1 nJ/acceso
Registro de desacoplo (cuando sea necesario)	0,05 ns	0,05 nJ/acceso

Queremos analizar 4 configuraciones para la cache de datos, todas ellas con 16KB de capacidad y líneas de 8 bytes:

- C1: Cache de mapeo directo con acceso PARALELO a etiquetas y datos.
 - C2: Cache asociativa por conjuntos de dos vías con acceso PARALELO a etiquetas y datos.
 - C3: Cache asociativa por conjuntos de dos vías con acceso SECUENCIAL a etiquetas y datos.
 - C4: Cache asociativa por conjuntos de dos vías SEGMENTADA en 2 etapas (el tiempo de acceso a la cache son 2 ciclos de procesador).
- a) **Calcula** el tiempo de ciclo de la cache de datos y el tiempo total de un acceso para las diferentes versiones del procesador Q1 con las caches C1, C2, C3 y C4, usando la distribución más adecuada de los componentes por etapas.

- b) **Calcula** la frecuencia de reloj para las diferentes versiones del procesador Q1 con las caches C1, C2, C3 y C4.

Un programa P que ejecuta $2,5 \times 10^9$ instrucciones tiene un 50% de instrucciones aritméticas, un 20% de instrucciones de salto y un 30% de instrucciones de acceso a memoria (Load/Store). Las instrucciones aritméticas tardan 4 ciclos, las de salto 3 y las de memoria 5 ciclos + los ciclos del acceso a la cache.

- c) **Calcula** el CPI del programa P para los procesadores con C1 y C4 suponiendo que nunca hay fallos en la cache de datos.

Sabemos que el programa P tiene un 10% de fallos con la cache de datos C1 y un 6% con la C2. Además, el tiempo de penalización medio por fallo en ambos casos es de 60 ciclos.

- d) ¿Que porcentaje de fallos en la cache de datos tendrá el programa P para el procesador Q1 con la cache C3? ¿y para la C4? (razona adecuadamente la respuesta)

- e) **Calcula** el speedup en tiempo de ejecución de C4 sobre C1 en % teniendo en cuenta la jerarquía de memoria completa.

Se hace una implementación multibanco (con 8 bancos) de la cache C4 para reducir el consumo de un acceso que llamaremos C4.b. Para ello dividimos la memoria de etiquetas en 8 subarrays donde cada uno consume 0,5nJ por acceso y la memoria de datos en 8 subarrays donde cada uno consume 3nJ por acceso.

- f) **Indica** cómo se desglosarían los bits de una dirección entre bits de Etiqueta, selección de Conjunto, Banco y Byte

- g) **Calcula** la energía (dinámica) consumida por un acceso a la cache C4.b (con la implementación multibanco).