

COGNOMS: NOM:

2on Control Arquitectura de Computadors

Curs 2013-2014 Q2

- Temps: 13:30 a 15:30
- Poseu clarament amb LLETRES MAJÚSCULES a cada full els cognoms i el nom

Problema 1. (4 puntos)

Tenemos un procesador con un camino de datos de 8 bits (todos los accesos a memoria son a byte). Por simplicidad ignoraremos los accesos a instrucciones en todo el problema. Este procesador tiene una cache de datos de mapeo directo con un tamaño de 256 bytes y bloques (lineas) de 16 bytes. Esta cache sigue una política de escritura retardada (Copy Back) con asignación en caso de fallo (Write Allocate). A este sistema de procesador con cache lo denominamos PC.

- a) **Describe** el funcionamiento de las políticas de escritura de esta cache en caso de **escritura-acierto** y de **escritura-fallo**. **Indica** claramente cuantos accesos a memoria principal se realizan en cada uno de los casos.

Escritura acierto:

Escribe en cache y activa Dirty Bit.

Escritura fallo:

Lee MP y escribe en MC activando DB, si DB estaba activado también escribe en MP reemplazando línea.

A este procesador se le ha añadido un mecanismo de memoria virtual con páginas de 4 Kbytes, direcciones lógicas de 20 bits y direcciones físicas de 16 bits.

- b) **Dibuja** una dirección lógica con los campos de bits relevantes para el acceso a cache, indicando claramente el nombre y tamaño de cada uno de ellos. **Dibuja** también una dirección lógica con los campos de bits relevantes para la traducción de direcciones, indicando claramente el nombre y tamaño de cada uno de ellos.

12Bits	4Bits	4Bits
TAG	#Linea MC	#Byte

8Bits	12Bits
VPN	Desplazamiento

Para acelerar la traducción, al sistema se le ha añadido un TLB de datos (DTLB) totalmente asociativo de cuatro entradas, con algoritmo de reemplazo LRU. Se sigue un esquema de **traducción después de acceder a memoria cache**, por lo que la **cache se accede con direcciones lógicas**. A este sistema de procesador con cache virtual y TLB lo denominamos PCVT.

Hemos ejecutado una aplicación en el sistema PC descrito al principio. Para esta aplicación, en media, se realizan 0,6 accesos a datos por instrucción (de los que un tercio son escrituras), la cache tiene una tasa de fallos del 20% y el 25% de los bloques de cache tienen el *dirty bit* activado cuando son reemplazados. Para esta aplicación se ha medido, en el sistema PC, un CPI de 3,5 ciclos/instrucción.

Sabemos que en el sistema PCVT, cada acceso a DTLB introduce una penalización de 1 ciclo en caso de acierto en DTLB, respecto al sistema PC.

c) **Calcula** el CPI del sistema PCVT suponiendo que nunca hay fallo en el DTLB

$$\text{CPI} = \text{CPI}_{\text{ideal}} + \text{CPI}_{\text{mem}} = \text{CPI}_{\text{ideal}} + nr * m * \text{Coste Acceso a DTLB} * t_{sa}(\text{DTLB}) =$$

$$= 3.5 + 0.6 * 0.2 * (1 * 0.75 + 2 * 0.25) * 1 = 3.65 \text{ c/i}$$

Para la aplicación anterior se ha medido en el sistema PCVT un 1% de fallos en el DTLB con respecto a los accesos a DTLB, y una penalización adicional de 200 ciclos por cada fallo del DTLB (además de la penalización de 1 ciclo en el acceso a DTLB).

d) **Calcula** el CPI del sistema PCVT teniendo en cuenta los fallos de DTLB

$$\text{CPI} = \text{CPI}_{\text{ideal}} + \text{CPI}_{\text{mem}} =$$

$$1. = 3.5 + (0.99 * 0.6 * 0.2 * (1 * 0.75 + 2 * 0.25) * 1 + 0.01 * 0.6 * 0.2 * 1 * 200) = 3.888$$

$$2. = 3.5 + (0.99 * 0.6 * 0.2 * (1 * 0.75 + 2 * 0.25) * 1 + 0.01 * 0.6 * 0.2 * (1 * 0.75 + 2 * 0.25) * 200) = 3.9485$$

No se cual de las dos esta bien... lo único que cambio es el valor de CosteAccesoADTLB

Sabemos que la cache está inicialmente vacia y que el contenido del DTLB es el siguiente (en hexadecimal):

VPN	PPN
00	4
01	5
02	6
03	7

e) **Rellena** la siguiente tabla a partir de la secuencia de referencias a memoria dada:

@lógica: Dirección lógica (en hexadecimal) generada por el procesador.

L/E: El acceso es lectura (L) o escritura (E).

TAG: Etiqueta del acceso a cache (en hexadecimal).

BMC: Bloque (línea) de cache que se accede (en hexadecimal).

Cache: Acierto de Cache (A) o Fallo de Cache (F).

LMP: Numero de bytes leídos de memoria principal (**vacío si no se lee**).

EMP: Numero de bytes escritos en memoria principal (**vacío si no se escribe**).

@f MP lec: Dirección física en caso de lectura de bloque de MP (en hexadecimal) (**vacío si no se lee**).

@f MP esc: Dirección física en caso de escritura de bloque en MP (en hexadecimal) (**vacío si no se escribe**).

@lógica	L/E	TAG	BMC	Cache	LMP	EMP	@f MP lec	@f MP esc
00001	L	000	0	F	16	-	4000	-
01112	L	011	1	F	16	-	5110	-
00003	L	000	0	A	-	-	-	-
01004	E	010	0	F	16	-	5000	-
01005	L	010	0	A	-	-	-	-
01116	E	011	1	A	-	-	-	-
02117	L	021	1	F	16	16	6110	5110

.....COGNOMS:NOM:

2on Control Arquitectura de Computadores

Curs 2013-2014 Q2

Problema 2. (3 puntos)

Dado el siguiente código escrito en C:

```
typedef struct
{
    int a;
    int b;
} smplstr;

int Sub (short a, char b, smplstr c, smplstr d[3])
{
    short local1;
    char local2;
    smplstr local3;
    smplstr local4[3];
    ...
    for (i=0;i<10000;i++) // (2)
        sum+=v1[i]*v2[i]; // (2)
    ...
    return d[1].b + (int) a; // (1)
}
```

- a) **Dibuja** el bloque de activación de la rutina Sub, indicando claramente los desplazamientos respecto a %ebp y el tamaño de todos los campos

Local1	-36
Local2	-34
---	-35
Local3	-32
Local4 [0]	-24
Local4 [1]	-16
Local4 [2]	-8
Ebp	0
@ret	4
a	8
---	10
b	12
---	13
c	16
@d	24
	28

- b) **Traduce** a ensamblador del x86 la instrucción (1) de la rutina Sub.

```
movl 24(%ebp), %ecx  
  
movswl 8(%ebp), %eax  
  
addl 12(%ecx), %eax
```

Tenemos el siguiente bucle, del que sabemos que cada iteración se traduce en 6 instrucciones ensamblador:

```
for (i=0; i<1000000; i++)  
    sum+=v1[i]*v2[i];
```

Este bucle se ejecuta en un procesador a 2GHz que tiene un CPI ideal de 1.5 y una cache de datos 2 asociativa con líneas de 32 bytes y un tiempo de servicio en caso de acierto de 1 ciclo. Además, sabemos que i y sum se almacenan en registros y que tanto v1 como v2 son vectores globales de floats. El bucle tarda 10 ms en ejecutarse.

- c) **Calcula** el tpf de la cache.

$CPI = Teje / (N * 1/f) = 10 * 10^{-3} / (6 * 10^6 * 1/2 * 10^9) = 10/3$
 $CPI_{mem} = CPI - CPI_{ideal} = 10/3 - 1.5 = 11/6$
 $CICLO_{mem} = CPI_{mem} * N = 11/6 * 6 * 10^6 = 11 * 10^6$
 $Tpf = CICLO_{mem} / (nr * m) = 11 * 10^6 / (2 * 10^6 * 1/8) = 44 \text{ ciclos}$

NOTA: La penalización por fallos tiene que ser de memoria, por eso calculo CICLO_{mem}

- d) Para mejorar el sistema decidimos añadir en el bucle dos instrucciones software de prefetch (una por vector) al inicio del bucle. Suponiendo que el CPI ideal del bucle no varía al insertar estas instrucciones y que realizan un prefetch perfecto anulando los fallos de carga de datos, **calcula** en cuanto tiempo se ejecutaría ahora el bucle.

Enunciado nos está diciendo que: $CPI_{mem} = 0$ y que hay 2 instr. más en el bucle.

$Teje = N * CPI_{ideal} * Tc = 8 * 10^6 * 1.5 * 1/2 * 10^9 = 6 * 10^{-3} \text{ s}$

($Tc = 1/f$)

- e) **Calcula** cuantas iteraciones como mínimo deberíamos lanzar la precarga por adelantado. Justifica la respuesta..

$Ciclos = CPI * N = 1.5 * 8 = 12 \text{ ciclos} \rightarrow 1 \text{ iteracion! (8 instr)}$

El prefetch tarda 44 ciclos (tpf), con 2 prefetch $\rightarrow 88 \text{ ciclos}$ (Creo que tmbn dan por buena la solución de usar solo los 44 ciclos de 1 prefetch, pero yo lo hago así que tmbn esta bnn creo)

$Iteraciones = 88 \text{ ciclos} / 12 \text{ ciclos/it} = 7.33 \rightarrow 8 \text{ iteraciones}$

.....COGNOMS:NOM:

2on Control Arquitectura de Computadores

Curs 2013-2014 Q2

Problema 3. (3 puntos)

Se dispone de un procesador de tipo Load/Store conectado a un sistema de memoria con las siguientes características:

- Memoria cache (MC) directa con política de escritura write through y write NO allocate.
- Memoria Victim Cache (VC) con remplazo LRU de 8 bloques de capacidad y acceso simultáneo a MC

Se han obtenido por simulación las siguientes medidas:

- tasa de aciertos en MC: 80% de los accesos totales, tanto para lecturas como para escrituras
- Tasa de aciertos en VC: 90% de los accesos a VC que son fallo en MC

El tiempo de acceso (Tsa) a memoria cache (MC) o a la victim cache (VC) es de 10 ns. tanto en lectura como en escritura. El tiempo para intercambiar una línea entre MC y VC es de 8 ns. El tiempo de acceso a memoria principal (MP) para escribir una palabra es de 60 ns. Para leer o escribir un bloque en la MP se emplean 80 ns.

- a) **Describe**, para cada uno de los siguientes casos: (1) Acierto en MC, (2) Fallo en MC y acierto en VC, (3) Fallo en MC y fallo en VC, las acciones a realizar para acceder al sistema de memoria en un **acceso de lectura** e indica el tiempo que tarda en completarse cada caso, justificando cómo lo has calculado..

(1) Acierto en MC

10ns del acceso a cache

(2) Fallo en MC y acierto en VC

10ns + 8ns = 18ns
(acceso a VC) + (swap líneas)

(3) Fallo en MC y fallo en VC

10ns + 80ns + 10ns = 100ns
(acceso cache/VC) + lees MP + acceso MC

- b) **Calcula** el tiempo empleado en realizar 1000 accesos de lectura consecutivos. Para realizar el cálculo, calcula por separado cuantos accesos hay de cada uno de los 3 casos descritos en el apartado a). No se corregirá este apartado si no se ha contestado el apartado a).

(1) Acierto en MC

0.8 * 1000 = 800accesos

(2) Fallo en MC y acierto en VC

0.2 * 0.9 * 1000 = 180 accesos

(3) Fallo en MC y fallo en VC

0.2 * 0.1 * 1000 = 20accesos

Tiempo total

(1) 10ns*800acc=8000ns (2) 18ns*180acc=3240ns (3) 100ns*20acc = 2000ns

- c) **Describe**, para cada uno de los siguientes casos: (1) Acierto en MC, (2) Fallo en MC y acierto en VC, (3) Fallo en MC y fallo en VC, las acciones a realizar para acceder al sistema de memoria en un acceso de escritura e indica el tiempo que tarda cada caso, justificando cómo lo has calculado..

(1) Acierto en MC

$$10ns + 60ns = 70ns \\ (\text{acc MC}) + (\text{esc MP})$$

Solucion profe: 60

(2) Fallo en MC y acierto en VC

$$10ns + 60ns + 8ns = 78ns \\ (\text{MC}) + \text{esc MP} + \text{swap}$$

Solucion profe: 60

(3) Fallo en MC y fallo en VC

$$10ns + 60ns = 70ns \\ (\text{MC}) + \text{esc MP}$$

Solucion profe: 60

Con el objeto de reducir el tiempo de las escrituras, se ha añadido al sistema un buffer de escritura de 50 entradas conectado entre la CPU y la memoria principal. El tiempo de acceso al buffer de escritura es de 12 ns.

- d) Suponiendo que siempre hay una entrada disponible en el buffer cuando se produce un acceso de escritura, para cada uno de los siguientes casos: (1) Acierto en MC, (2) Fallo en MC y acierto en VC, (3) Fallo en MC y fallo en VC, **describe** las acciones a realizar para acceder al sistema de memoria en un acceso de escritura e indica el tiempo que tarda cada caso, justificando cómo lo has calculado..

(1) Acierto en MC

$$10ns + 12ns = 22ns \\ (\text{acc MC}) + (\text{esc MP})$$

Solucion profe: 12

(2) Fallo en MC y acierto en VC

$$10ns + 12ns + 8ns = 30ns \\ (\text{MC}) + \text{esc MP} + \text{swap}$$

Solucion profe: 28

(3) Fallo en MC y fallo en VC

$$10ns + 12ns = 22ns \\ (\text{MC}) + \text{esc MP}$$

Solucion profe: 12