

# Ampliación de Estructura de Computadores

## Soluciones de los ejercicios del Tema 4

Aquí se muestra la solución final de algunos de los ejercicios propuestos en el boletín del tema 4. No se muestra el desarrollo completo en todos los casos. El objetivo es que el alumno pueda comprobar si ha obtenido el resultado correcto una vez que ha realizado él mismo el ejercicio.

### Ejercicio 1

#### Apartado a

El tiempo medio de acceso a memoria se calcula a partir del tiempo de acceso a la caché en caso de acierto, la tasa de fallos de la caché y la penalización en caso de fallo:

$$T_{am} = T_{sa} + m \times T_{pf}$$

Ambas cachés tienen un tiempo de acceso de un ciclo. La tasa de fallos ( $m$ ) depende de la configuración de la caché como se muestra en la tabla del enunciado. La penalización por fallo es de 40 ciclos en el caso de la caché de mapeo directo, pero se ve afectada por el tiempo de ciclo en el caso de la caché asociativa, ya que se incrementa un 10%. Por tanto, al tener ciclos más largos el tiempo (en ciclos) que la memoria tardará en contestar al procesador en el caso de las cachés asociativas será:

$$T_{pf} = \left\lceil \frac{40}{1.1} \right\rceil = 37 \text{ ciclos (ciclos 10\% más largos)}$$

Por tanto:

$$T_{am\_1via} = 1 + 0.072 \times 40 = 3.880 \text{ ciclos}$$

$$T_{am\_2vias} = 1 + 0.057 \times 37 = 3.109 \text{ ciclos}$$

#### Apartado b

Para decidir qué organización de memoria es más rápida, y teniendo en cuenta que el tiempo de ciclo cambia, tenemos que compararlas en base al tiempo de ejecución de la aplicación.

$$T_{ej} = NI \times CPI \times T_{ciclo}$$

El número de instrucciones no cambiará de una caché a otra para el mismo programa. El tiempo de ciclo se incrementará en un 10% para la caché de dos vías, por tanto:

$$T_{ciclo\_2vias} = 1.1 \times T_{ciclo\_1via}$$

Ahora solo nos queda calcular el CPI que se obtiene con ambas cachés, teniendo en cuenta el CPI base, la tasa de accesos a memoria por instrucción, el porcentaje de estos accesos que tendrán penalización y el tiempo de penalización por fallo.

$$CPI = CPI_{base} + M/I \times m \times T_{pf}$$

Sabemos que el 30% de las instrucciones acceden a datos, por tanto cada instrucción requerirá un acceso a memoria (para obtener la instrucción) y un 30% de ellas un acceso a dato. Por tanto el número de accesos a memoria por instrucción es de 1.3.

Aplicando la fórmula anterior, obtenemos los siguientes resultados:

$$CPI_{1via} = CPI_{base} + M/I \times m_{1via} \times T_{pf\_1via} = 1 + 1.3 \times 0.072 \times 40 = 4.7440$$

$$CPI_{2vias} = CPI_{base} + M/I \times m_{2vias} \times T_{pf\_2vias} = 1 + 1.3 \times 0.057 \times 37 = 3.7417$$

El tiempo de ejecución en cada caso será:

$$T_{ej\_1via} = NI \times CPI_{1via} \times T_{ciclo\_1via} = NI \times 4.7440 \times T_{ciclo\_1via}$$

$$T_{ej\_2vias} = NI \times CPI_{2vias} \times T_{ciclo\_2vias} = NI \times 3.7417 \times 1.1 \times T_{ciclo\_1via} = NI \times 4.11587 \times T_{ciclo\_1via}$$

Dividiendo  $T_{ej\_1via}$  entre  $T_{ej\_2vias}$  obtenemos cómo de rápido es la opción asociativa de dos vías respecto a la de mapeo directo:

$$\frac{T_{ej\_1via}}{T_{ej\_2vias}} = \frac{NI \times 4.7440 \times T_{ciclo\_1via}}{NI \times 4.11587 \times T_{ciclo\_1via}} = 1.1526$$

Por tanto la configuración de dos vías es un 15.26% más rápida que la de mapeo directo.

## Ejercicio 2

### Apartado a

- Primera configuración:

- Accesos a instrucciones:

$$T_{am_i} = T_{sa} + m \times T_{pf} = 1 + 0.01 \times 50 = 1.5 \text{ ciclos}$$

- Accesos a datos:

$$f_{sucio} = 0.20 \text{ (Probabilidad de reemplazar un bloque sucio)}$$

$$T_{psucio} = T_{pf} = 50 \text{ ciclos (Penalización por hacer postescritura*)}$$

$$T_{am_d} = T_{sa} + m \times (T_{pf} + f_{sucio} \times T_{psucio}) = 1 + 0.10 \times (50 + 0.20 \times 50) = 7 \text{ ciclos}$$

Teniendo en cuenta que hay un 30% de cargas/almacenamientos, 30 de cada 130 accesos a memoria serán a datos, y el resto a instrucciones. Por tanto:

$$T_{am} = \frac{100}{130} \times T_{am_i} + \frac{30}{130} \times T_{am_d} = 2.77 \text{ ciclos}$$

- Segunda configuración:

- Accesos a instrucciones:

Supondremos que el acceso a memoria no empieza hasta que acaba el acceso al buffer de prelectura (es decir, accedemos secuencialmente a las dos estructuras).

$$p_{buffer} = 0.60 \text{ (Probabilidad de acertar en el buffer de prelectura)}$$

$$T_{pbuffer} = 1 \text{ ciclo (Tiempo para llevar el bloque del buffer de prelectura a la caché)}$$

$$T_{am_i} = T_{sa} + m \times (T_{pbuffer} + (1 - p_{buffer}) \times T_{pf})$$

$$= 1 + 0.01 \times (1 + 0.40 \times 50) = 1.21 \text{ ciclos}$$

- Accesos a datos:

Supondremos que el acceso a memoria no empieza hasta que acaba el acceso al buffer de escritura (es decir, también accedemos secuencialmente a las dos estructuras).

$$\begin{aligned}
 f_{buffer\ escritura} &= 0.30 \text{ (Probabilidad de encontrar el dato en el buffer de escritura)} \\
 T_{buffer\ escritura} &= 2 \text{ ciclos (Tiempo para llevar el bloque del buffer de escritura a la caché)} \\
 T_{am_d} &= T_{sa} + m \times (T_{buffer\ escritura} + (1 - f_{buffer\ escritura}) \times T_{pf}) \\
 &= 1 + 0.1 \times (2 + 0.70 \times 50) = 4.70 \text{ ciclos}
 \end{aligned}$$

Por tanto:

$$T_{am} = \frac{100}{130} \times T_{am_i} + \frac{30}{130} \times T_{am_d} = 2.0154 \text{ ciclos}$$

Con lo que la mejora que se obtiene es de  $\frac{2.77}{2.0154} - 1 = 37.44\%$ .

## Apartado b

Como el tiempo de ciclo y el número de instrucciones se mantienen constantes, usaremos el CPI para comparar las dos configuraciones:

- Primera configuración:

$$\begin{aligned}
 f_{sucio} &= 0.20 \text{ (Probabilidad de reemplazar un bloque sucio)} \\
 T_{psucio} &= T_{pf} = 50 \text{ ciclos (Penalización por hacer postescritura)} \\
 CPI &= 1 + m_i \times T_{pf} + AMI \times m_d \times (T_{pf} + f_{sucio} \times T_{psucio}) = \\
 &= 1 + 0.01 \times 50 + 0.30 \times 0.10 \times (50 + 0.20 \times 50) = 3.3 \text{ ciclos}
 \end{aligned}$$

- Segunda configuración:

Se hacen las mismas suposiciones que en el apartado anterior.

$$\begin{aligned}
 p_{buffer} &= 0.60 \text{ (Probabilidad de acertar en el buffer de prelectura)} \\
 T_{pbuffer} &= 1 \text{ ciclo (Tiempo para llevar el bloque del buffer de prelectura a la caché)} \\
 f_{buffer\ escritura} &= 0.30 \text{ (Probabilidad de encontrar el dato en el buffer de escritura)} \\
 T_{buffer\ escritura} &= 2 \text{ ciclos (Tiempo para llevar el bloque del buffer de escritura a la caché)} \\
 CPI &= 1 + m_i \times (T_{pbuffer} + (1 - p_{buffer}) \times T_{pf}) \\
 &\quad + AMI \times m_d \times (T_{buffer\ escritura} + (1 - f_{buffer\ escritura}) \times T_{pf}) = \\
 &= 1 + 0.01 \times (1 + 0.40 \times 50) + 0.30 \times 0.1 \times (2 + 0.70 \times 50) = 2.32 \text{ ciclos}
 \end{aligned}$$

Y la mejora que se obtiene es de  $1 - \frac{3.3}{2.32} = 42.24\%$ .

## Ejercicio 3

$$f_{detBufEsc} = 0.20$$

$$P_{detBufEsc} = 50 \text{ ciclos}$$

$$PF = 100 \text{ ciclos}$$

$$TF_{inst} = 0.001$$

$$TF_{datos} = 0.05$$

$$\begin{aligned} CPI_{base} &= f_{alu} \times CPI_{alu} + f_{load} \times CPI_{load} + f_{store} \times CPI_{store} + f_{saltos} \times CPI_{saltos} \\ &= 0.50 \times 1 + 0.20 \times 1 + 0.10 \times 1 + 0.20 \times 1.2 = 1.04 \text{ ciclos} \end{aligned}$$

$$\begin{aligned} CPI_{detMem} &= TF_{inst} \times PF + f_{load} \times TF_{datos} \times PF + f_{store} \times f_{detBufEsc} \times P_{detBufEsc} \\ &= 0.001 \times 100 + 0.20 \times 0.05 \times 100 + 0.10 \times 0.20 \times 50 = 2.10 \text{ ciclos} \end{aligned}$$

$$CPI = CPI_{base} + CPI_{detMem} = 1.04 + 2.10 = 3.14 \text{ ciclos}$$

## Ejercicio 4

### Apartado a

En el caso de las escrituras, suponemos que no se espera a acceder a la caché antes de comenzar la escritura a memoria.

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

$$TF = 1 - 0.90 = 0.10$$

$$PF = T_{acceso \text{ memoria}} = 50 \text{ ciclos}$$

$$f_{load+store} = f_{load} + f_{store} = 0.30$$

$$f_{load} = 2 \times f_{store}$$

$$T_{am_{load}} = T_{sa} + TF \times PF = 1 + 0.10 \times 50 = 6 \text{ ciclos}$$

$$T_{am_{store}} = T_{acceso \text{ memoria}} = 50 \text{ ciclos}$$

$$T_{am} = \frac{f_{load} \times T_{am_{load}} + f_{store} \times T_{am_{store}}}{f_{load} + f_{store}} = \frac{2 \times T_{am_{load}} + T_{am_{store}}}{3} = \frac{2 \times 6 + 50}{3} = 20.67 \text{ ciclos}$$

### Apartado b

$$T_{sa_i} = T_{sa_d} = 1 \text{ ciclo}$$

$$TF_d = 1 - 0.90 = 0.10$$

$$TF_i = 1 - 0.98 = 0.02$$

$$PF_d = PF_i = T_{acceso \text{ memoria}} = 50 \text{ ciclos}$$

$$f_{load+store} = f_{load} + f_{store} = 0.30$$

$$f_{load} = 2 \times f_{store} \Rightarrow f_{load} = 0.20$$

$$f_{store} = 0.10$$

$$T_{am_{inst}} = T_{sa_i} + TF_i \times PF_i = 1 + 0.02 \times 50 = 2 \text{ ciclos}$$

$$T_{am_{load}} = T_{sa_d} + TF_d \times PF_d = 1 + 0.10 \times 50 = 6 \text{ ciclos}$$

$$T_{am_{store}} = T_{acceso \text{ memoria}} = 50 \text{ ciclos}$$

$$T_{am} = \frac{T_{am_{inst}} + f_{load} \times T_{am_{load}} + f_{store} \times T_{am_{store}}}{1 + f_{load} + f_{store}} = \frac{2 + 0.20 \times 6 + 0.10 \times 50}{1 + 0.30} = 6.3077 \text{ ciclos}$$

### Apartado c

$$\begin{aligned}TF_d &= 1 - 0.90 = 0.10 \\TF_i &= 1 - 0.98 = 0.02 \\PF_{loads} &= PF_i = T_{\text{acceso memoria}} = 50 \text{ ciclos} \\f_{load+store} &= f_{load} + f_{store} = 0.30 \\f_{load} &= 2 \times f_{store} \Rightarrow f_{load} = 0.20 \\f_{store} &= 0.10 \\CPI_{base} &= 1 \\CPI &= CPI_{base} + TF_i \times PF_i + f_{load} \times TF_d \times PF_{loads} + f_{store} \times T_{\text{acceso memoria}} \\&= 1 + 0.02 \times 50 + 0.20 \times 0.10 \times 50 + 0.10 \times 50 = 8 \text{ ciclos}\end{aligned}$$

### Apartado d

$$\begin{aligned}T_{sa_i} &= T_{sa_d} = 1 \text{ ciclo} \\TF_d &= 1 - 0.90 = 0.10 \\TF_i &= 1 - 0.98 = 0.02 \\PF_d &= PF_i = T_{\text{acceso memoria}} = 50 \text{ ciclos} \\f_{\text{bloque sucio}} &= 0.20 \\T_{\text{bloque sucio}} &= T_{\text{acceso memoria}} = 50 \text{ ciclos} \\f_{load+store} &= f_{load} + f_{store} = 0.30 \\f_{load} &= 2 \times f_{store} \Rightarrow f_{load} = 0.20 \\f_{store} &= 0.10 \\T_{am_{inst}} &= T_{sa_i} + TF_i \times PF_i = 1 + 0.02 \times 50 = 2 \text{ ciclos} \\T_{am_{load}} &= T_{am_{store}} = T_{sa_d} + TF_d \times (PF_d + f_{\text{bloque sucio}} \times T_{\text{bloque sucio}}) \\&= 1 + 0.10 \times (50 + 0.20 \times 50) = 7 \text{ ciclos} \\T_{am} &= \frac{T_{am_{inst}} + f_{load} \times T_{am_{load}} + f_{store} \times T_{am_{store}}}{1 + f_{load} + f_{store}} \\&= \frac{2 + 0.20 \times 7 + 0.10 \times 7}{1 + 0.30} = 3.1538 \text{ ciclos}\end{aligned}$$

### Apartado e

Suponemos que los accesos al buffer de escritura y a la memoria se realizan de forma secuencial.

$$\begin{aligned}
T_{sa_i} &= T_{sa_d} = 1 \text{ ciclo} \\
TF_d &= 1 - 0.90 = 0.10 \\
TF_i &= 1 - 0.98 = 0.02 \\
PF_d &= PF_i = T_{\text{acceso memoria}} = 50 \text{ ciclos} \\
f_{\text{bloque sucio}} &= 0.20 \\
f_{\text{buffer escritura lleno}} &= 0.10 \\
T_{\text{bloque sucio}} &= f_{\text{buffer escritura lleno}} \times T_{\text{acceso memoria}} = 0.10 \times 50 = 5 \text{ ciclos} \\
f_{\text{load+store}} &= f_{\text{load}} + f_{\text{store}} = 0.30 \\
f_{\text{load}} &= 2 \times f_{\text{store}} \Rightarrow f_{\text{load}} = 0.20 \\
f_{\text{store}} &= 0.10 \\
f_{\text{aciertoBuffEscr}} &= 0.15 \\
T_{\text{accesoBuffEscr}} &= 2 \\
T_{am_{inst}} &= T_{sa_i} + TF_i \times PF_i = 1 + 0.02 \times 50 = 2 \text{ ciclos} \\
T_{am_{load}} &= T_{am_{store}} = T_{sa_d} + TF_d \times (T_{\text{accesoBuffEscr}} \\
&\quad + (1 - f_{\text{aciertoBuffEscr}}) \times (PF_d + f_{\text{bloque sucio}} \times T_{\text{bloque sucio}})) \\
&= 1 + 0.10 \times (2 + (1 - 0.15) \times (50 + 0.20 \times 5)) = 5.535 \text{ ciclos} \\
T_{am} &= \frac{T_{am_{inst}} + f_{\text{load}} \times T_{am_{load}} + f_{\text{store}} \times T_{am_{store}}}{1 + f_{\text{load}} + f_{\text{store}}} \\
&= \frac{2 + (0.20 + 0.10) \times 5.535}{1 + 0.30} = 2.816 \text{ ciclos}
\end{aligned}$$

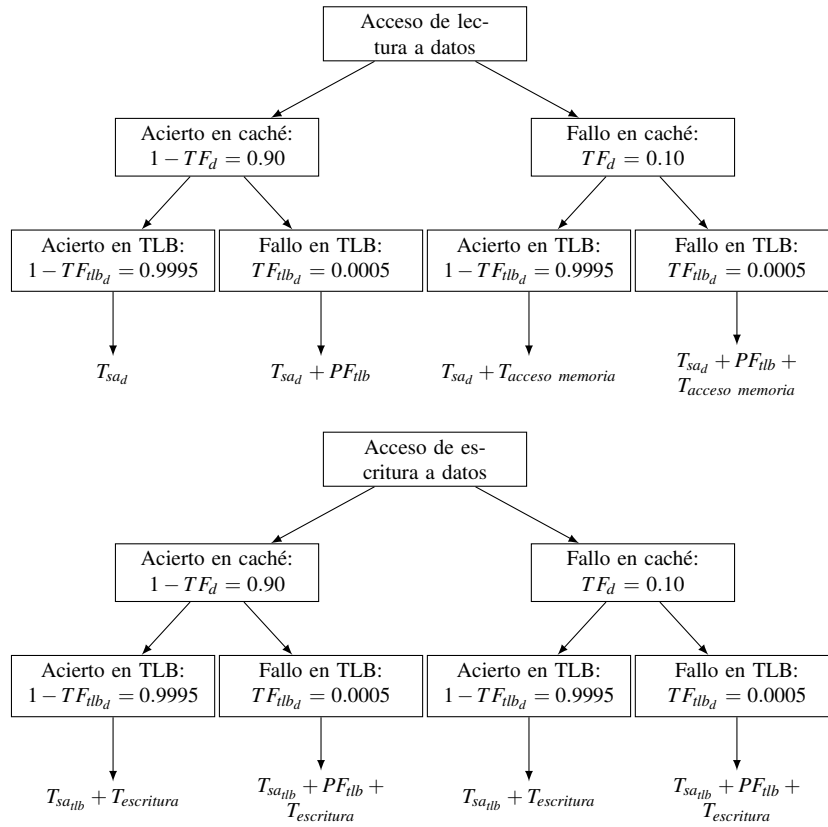
## Ejercicio 5

### Apartado a

$$\begin{aligned}
T_{sa_i} &= 2 \text{ ciclos (Dos etapas de búsqueda de instrucciones)} \\
TF_i &= 0.03 \\
T_{sa_{tlb}} &= 1 \text{ ciclo} \\
TF_{tlb_i} &= 0.0001 \\
PF_{tlb} &= 200 \text{ ns} = \frac{200 \times 10^{-9}}{1/(2 \times 10^9)} \text{ ciclos} = 400 \text{ ciclos} \\
T_{\text{traducción}} &= T_{sa_{tlb}} + TF_{tlb_i} \times PF_{tlb} = 1 + 0.0001 \times 400 = 1.04 \text{ ciclos} \\
T_{\text{acceso memoria}} &= 200 \text{ ns} = 400 \text{ ciclos} \\
PF_i &= T_{\text{traducción}} + T_{\text{acceso memoria}} = 1.04 + 400 = 401.04 \text{ ciclos} \\
T_{am_i} &= T_{sa_i} + TF_i \times PF_i = 2 + 0.03 \times 401.04 = 14.0312 \text{ ciclos}
\end{aligned}$$

### Apartado b

Análisis por casos:



$T_{sa_d} = 3$  ciclos (Dos etapas de acceso a memoria)

$TF_d = 0.10$

$T_{sa_tlb} = 1$  ciclo (incluido en  $T_{sa_d}$ )

$TF_{tlb_d} = 0.0005$

$PF_{tlb} = 200\text{ns} = \frac{200 \times 10^{-9}}{1/(2 \times 10^9)} \text{ ciclos} = 400 \text{ ciclos}$

$T_{acceso\ memoria} = 200\text{ns} = 400 \text{ ciclos}$

$T_{escritura} = T_{acceso\ memoria} = 400 \text{ ciclos}$

$T_{amload} = T_{sa_d} + TF_{tlb_d} \times PF_{tlb} + TF_d \times T_{acceso\ memoria}$   
 $= 3 + 0.0005 \times 400 + 0.10 \times 400 = 43.2 \text{ ciclos}$

$T_{amstore} = T_{sa_tlb} + TF_{tlb_d} \times PF_{tlb} + T_{escritura}$   
 $= 1 + 0.0005 \times 400 + 400 = 401.2 \text{ ciclos}$

$f_{load} = 0.24$

$f_{store} = 0.15$

$T_{am_d} = \frac{f_{load} \times T_{amload} + f_{store} \times T_{amstore}}{f_{load} + f_{store}} = \frac{0.24 \times 43.2 + 0.15 \times 401.2}{0.24 + 0.15} = 180.89 \text{ ciclos}$

## Apartado c

$$\begin{aligned}T_{ciclo} &= \frac{10^9 \text{ ns}}{2 \times 10^9} = 0.5 \text{ ns} \\T_{am} &= \frac{T_{am_i} + f_{load} \times T_{am_{load}} + f_{store} \times T_{am_{store}}}{1 + f_{load} + f_{store}} \\&= \frac{14.0312 + 0.24 \times 43.2 + 0.15 \times 401.2}{1 + 0.24 + 0.15} = 60.848 \text{ ciclos} \\&= 60.848 \times 0.5 \text{ ns} = 30.42 \text{ ns}\end{aligned}$$

## Apartado d

$$\begin{aligned}f_{alu} &= 0.36 \\f_{load} &= 0.24 \\f_{store} &= 0.15 \\f_{saltos} &= 0.25 \\CPI_{alu} &= CPI_{store} = 1 \\CPI_{load} &= 2.5 \text{ (Teniendo en cuenta solo detenciones por dependencias, de los consumidores)} \\CPI_{saltos} &= 1.27 \\CPI_{base} &= f_{alu} \times CPI_{alu} + f_{load} \times CPI_{load} + f_{store} \times CPI_{store} + f_{saltos} \times CPI_{saltos} \\&= 0.36 \times 1 + 0.24 \times 2.5 + 0.15 \times 1 + 0.25 \times 1.27 = 1.4275 \text{ ciclos} \\TF_i &= 0.03 \\T_{sa_{tlb}} &= 1 \text{ ciclo} \\TF_{tlb_i} &= 0.0001 \\PF_{tlb} &= 200 \text{ ns} = \frac{200 \times 10^{-9}}{1/(2 \times 10^9)} \text{ ciclos} = 400 \text{ ciclos} \\T_{traducción} &= T_{sa_{tlb}} + TF_{tlb_i} \times PF_{tlb} = 1 + 0.0001 \times 400 = 1.04 \text{ ciclos} \\PF_i &= T_{traducción} + T_{acceso memoria} = 1.04 + 400 = 401.04 \text{ ciclos} \\TF_d &= 0.10 \\TF_{tlb_d} &= 0.0005 \\PF_{load} &= T_{acceso memoria} = 200 \text{ ns} = 400 \text{ ciclos} \\T_{escritura} &= T_{acceso memoria} = 400 \text{ ciclos} \\CPI_{detMem} &= TF_i \times PF_i + f_{load} \times (TF_{tlb_d} \times PF_{tlb} + TF_d \times PF_{load}) + f_{store} \times (TF_{tlb_d} \times PF_{tlb} + T_{escritura}) \\&= 0.03 \times 401.04 + 0.24 \times (0.0005 \times 400 + 0.10 \times 400) + 0.15 \times (0.0005 \times 400 + 400) = 81.71 \text{ ciclos} \\CPI &= CPI_{base} + CPI_{detMem} = 83.14 \text{ ciclos}\end{aligned}$$

## Ejercicio 6

### Apartado a

Deberíamos tener una caché indexada virtualmente, de forma que pudiéramos iniciar el acceso a la caché antes de realizar la traducción de direcciones.

La caché podría ser virtualmente indexada y físicamente etiquetada o completamente virtual. En el último caso, sólo sería necesario acceder al TLB en caso de fallo de caché y, en principio, no tendríamos que hacer ningún cambio más al sistema. Sí tendríamos que modificar el sistema operativo para realizar correctamente los cambios de contexto y, posiblemente, para evitar problemas creados por el aliasing. No usaremos una caché virtual en este ejercicio para evitar cualquier modificación al sistema operativo.

Sin embargo, si la caché fuera virtualmente indexada y físicamente etiquetada, tenemos que asegurarnos de que los bits que se utilizan para indexar sean iguales en la dirección física y la virtual:



Si el tamaño de página es de 4 KB, coincidirán los  $\log_2 4096 = 12$  bits menos significativos de ambas direcciones. Por otro lado, en una caché de 16 KB, correspondencia directa y tamaño de bloque  $b$  bytes habrá  $\frac{16 \times 1024}{b}$  conjuntos, por lo que necesitaremos  $\log_2 \frac{16 \times 1024}{b}$  bits para indexarlos mientras que los  $\log_2 b$  bits menos significativos de la dirección codificarán el desplazamiento dentro del bloque.

Por tanto, se tendría que cumplir la siguiente condición:

$$\log_2 \frac{16 \times 1024}{b} + \log_2 b \leq 12$$

Que, como se puede comprobar, no se cumple para ningún valor de  $b$ , ya que:

$$\log_2 \frac{16 \times 1024}{b} + \log_2 b = \log_2 16 + \log_2 1024 - \log_2 b + \log_2 b = 14$$

Por tanto, nos vemos obligados a realizar una de las siguientes acciones o una combinación de varias de ellas:

- Aumentar el tamaño de página (hasta  $2^{14}$  bytes).
- Aumentar la asociatividad de la caché (hasta 4 vías).
- Disminuir el tamaño de la caché (hasta 4 KB).

Las dos últimas opciones disminuyen el número de bits necesarios para realizar el indexado, mientras que la primera aumenta el número de bits que se mantienen inalterados entre la dirección virtual y la física.

## Apartado b

Utilizando segmentación. Necesitaríamos dedicar al menos dos etapas del cauce para la búsqueda de instrucciones (una para la traducción y otra para la lectura de la caché), y también al menos dos etapas para el acceso a memoria.

## Apartado c

Los efectos dependerán de las opciones que se tomen en el apartado a.

- 1) El tiempo de ciclo podría aumentar si se aumenta la asociatividad de la caché.
- 2) La penalización por fallo no cambiaría para una caché virtualmente indexada y físicamente etiquetada. Si se utiliza una caché completamente virtual, aumentaría en 1 ciclo porque se tendría que acceder al TLB en caso de fallo.
- 3) La tasa de fallos podría disminuir si se aumenta la asociatividad, y podría aumentar si se disminuye el tamaño de la caché.
- 4) El tamaño de bloque no se ve afectado en ningún caso.
- 5) El número de buffers de escritura no se ve afectado en ningún caso.

## Apartado d

Supondremos que la primera máquina tiene una caché asociativa de 4 vías y la segunda tiene un pipeline con 2 etapas extra (una etapa extra de búsqueda de instrucciones y otra de acceso a memoria). Por tanto, el tiempo de ciclo de la primera máquina será un 10% mayor y la tasa de fallos un 20% menor (de acuerdo con el apartado c).

También supondremos que las cachés son de post-escritura, con un buffer de post-escritura suficientemente grande para que no haya detenciones por que esté lleno.

Ajustamos el tiempo de penalización por fallo teniendo en cuenta el cambio de tiempo de ciclo:

$$T_{pf2} = 50 \text{ ciclos}_2 \quad (\text{de acuerdo con el enunciado principal})$$

$$T_{pf1} = \left\lceil \frac{50}{1.1} \right\rceil = 46 \text{ ciclos}_1 \quad (\text{ciclos 10\% más largos})$$

El tiempo medio de acceso a memoria será, en el caso de estas dos máquinas, igual para las instrucciones que para los datos:

$$\begin{aligned}
m_2 &= 0.05 \\
m_1 &= 0.8 \times m_2 = 0.04 \\
T_{am_1} &= T_{sa_1} + m_1 \times T_{pf_1} = 1 + 0.04 \times 46 = 2.84 \text{ ciclos}_1 \quad (\text{ciclos } 10\% \text{ más largos}) \\
T_{am_2} &= T_{sa_2} + m_2 \times T_{pf_2} = 2 + 0.05 \times 50 = 4.5 \text{ ciclos}_2
\end{aligned}$$

Como unidad de medida para comparar, usaremos el tiempo de ciclo de la segunda máquina::

$$\begin{aligned}
T_{am_1} &= 2.84 \text{ ciclos}_1 = 3.124 \text{ ciclos}_2 \\
T_{am_2} &= 4.5 \text{ ciclos}_2
\end{aligned}$$

## Apartado e

$$\begin{aligned}
f_{mem} &= 0.30 \\
CPI_1 &= 1 + m_1 \times T_{pf_1} + f_{mem} \times m_1 \times T_{pf_1} = 1 + 0.04 \times 46 + 0.30 \times 0.04 \times 46 = 3.392 \text{ ciclos}_1 \\
CPI_2 &= 1 + m_2 \times T_{pf_2} + f_{mem} \times m_2 \times T_{pf_2} = 1 + 0.05 \times 50 + 0.30 \times 0.05 \times 50 = 4.25 \text{ ciclos}_2 \\
T_1 &= NI \times TC_1 \times CPI_1 = 3.392 \times NI \times TC_1 \\
T_2 &= NI \times TC_2 \times CPI_2 = NI \times \frac{TC_1}{1.10} \times CPI_2 = 3.864 \times NI \times TC_1
\end{aligned}$$

Por tanto, la primera máquina es más rápida gracias a la menor tasa de fallos y a pesar del mayor tiempo de ciclo.

## Apartado f

Supondremos una penalización de 4 ciclos para los saltos mal predichos en la segunda máquina (que es la que tiene el pipeline más largo) y de 2 en la primera.

$$\begin{aligned}
f_{branch} &= 0.20 \\
f_{aciertoSalto} &= 0.90 \Rightarrow f_{falloSalto} = 1 - 0.90 = 0.10 \\
pbranch_1 &= 2 \\
pbranch_2 &= 4 \\
CPI_1 &= 1 + m_1 \times T_{pf_1} + f_{mem} \times m_1 \times T_{pf_1} + f_{branch} \times f_{falloSalto} \times pbranch_1 = \\
&= 1 + 0.04 \times 46 + 0.30 \times 0.04 \times 46 + 0.20 \times 0.10 \times 2 = 3.432 \\
CPI_2 &= 1 + m_2 \times T_{pf_2} + f_{mem} \times m_2 \times T_{pf_2} + f_{branch} \times f_{falloSalto} \times pbranch_2 = \\
&= 1 + 0.05 \times 50 + 0.30 \times 0.05 \times 50 + 0.20 \times 0.10 \times 4 = 4.33 \\
T_1 &= NI \times TC_1 \times CPI_1 = 3.432 \times NI \times TC_1 \\
T_2 &= NI \times TC_2 \times CPI_2 = NI \times \frac{TC_1}{1.10} \times CPI_2 = 3.936 \times NI \times TC_1
\end{aligned}$$

## Ejercicio 7

### Apartado a

La penalización por fallos en el caso de la caché de instrucciones es el tiempo necesario para leer un bloque de 4 palabras, es decir  $T_{pf_i} = 40 + 4 \times 4 = 56$  ciclos.

En el caso de la caché de datos, debemos tener en cuenta que cuando se produce un fallo de cache debemos desalojar un bloque que estaba en la cache de datos y que algunas veces (25%) estará marcado como sucio, en cuyo caso deberemos escribirlo en memoria principal. Por tanto,  $T_{pf_d} = 40 + 4 \times 4 + 0.25 \times (40 + 4 \times 4) = 70$  ciclos.

Por tanto, el tiempo medio de acceso a memoria a través de cada caché es:

$$T_{am_i} = T_{sa_i} + m_i \times T_{pf_i} = 1 + 0.01 \times 56 = 1.56 \text{ ciclos}$$

$$T_{am_d} = T_{sa_d} + m_d \times T_{pf_d} = 1 + 0.05 \times 70 = 4.50 \text{ ciclos}$$

Y teniendo en cuenta que el 30% de las instrucciones (20% de cargas y 10% de almacenamientos) acceden a memoria:

$$T_{am} = \frac{T_{am_i} + 0.30 \times T_{am_d}}{1 + 0.30} = 2.24 \text{ ciclos}$$

Y el CPI queda:

$$CPI = 1 + m_i \times T_{pf_i} + M/I \times m_d \times T_{pf_d} = 1 + 0.01 \times 56 + 0.30 \times 0.05 \times 70 = 2.61 \text{ ciclos}$$

## Apartado b

El comportamiento de la caché de instrucciones no cambia, pero la caché de datos ahora se comporta de forma diferente para las lecturas (0.20 de cada 0.30 accesos) y las escrituras (0.10 de cada 0.30 accesos). Ahora, cada escritura se realiza tanto en la caché como en la memoria principal si acierta, o directamente en la memoria principal, si falla. En este último caso no leemos el bloque. Además, se escribe solamente la palabra modificada, con lo que la penalización es de  $T_{pf_{sw}} = 40 + 1 \times 4 = 44$  ciclos en las escrituras. Tampoco hay bloques sucios, por lo que la penalización por fallos de las lecturas es de  $T_{pf_{lw}} = 40 + 4 \times 4 = 56$  ciclos. Por tanto:

$$T_{am_i} = T_{sa_i} + m_i \times T_{pf_i} = 1 + 0.01 \times 56 = 1.56 \text{ ciclos}$$

$$T_{am_{lw}} = T_{sa_d} + m_d \times T_{pf_{lw}} = 1 + 0.05 \times 56 = 3.80 \text{ ciclos}$$

$$T_{am_{sw}} = T_{sa_d} + T_{pf_{sw}} = 1 + 44 = 45 \text{ ciclos (suponemos que se hacen secuencialmente la escritura en caché y en memoria)}$$

Y teniendo en cuenta que hay 0.20 de lecturas a la caché de datos por cada instrucción y 0.10 escrituras por instrucción:

$$T_{am} = \frac{T_{am_i} + 0.20 \times T_{am_{lw}} + 0.10 \times T_{am_{sw}}}{1 + 0.20 + 0.10} = 5.25 \text{ ciclos}$$

Y el CPI:

$$CPI = 1 + m_i \times T_{pf_i} + f_{lw} \times m_d \times T_{pf_{lw}} + f_{sw} \times T_{pf_{sw}} = 1 + 0.01 \times 56 + 0.20 \times 0.05 \times 56 + 0.10 \times 44 = 6.52 \text{ ciclos}$$

## Apartado c

El tiempo de acceso en cada caso con los cambios mencionados en el enunciado es:

$$T_{am_i} = T_{sa_i} + m_i \times T_{pf_i} = 1 + 0.01 \times 56 = 1.56 \text{ ciclos}$$

$$T_{am_{lw}} = T_{sa_d} + m_d \times T_{pf_{lw}} + f_{lectura\ buffer\ escritura} \times T_{lectura\ buffer\ escritura} = 1 + 0.05 \times (56 + 0.12 \times 10) = 3.86 \text{ ciclos}$$

$$T_{am_{sw}} = T_{sa_d} + f_{buffer\ escritura\ lleno} \times T_{pf_{sw}} = 1 + 0.08 \times 44 = 4.52 \text{ ciclos (suponemos que se hacen secuencialmente la escritura en caché y en memoria)}$$

$$T_{am} = \frac{T_{am_i} + 0.20 \times T_{am_{lw}} + 0.10 \times T_{am_{sw}}}{1 + 0.20 + 0.10} = 2.14 \text{ ciclos}$$

## Apartado d

Las tasas de fallo se reducen a la mitad ( $m_{i-d} = 0.005$ ,  $m_{d-d} = 0.025$ ), y también hemos de tener en cuenta que el cambio en el tiempo de ciclo ( $TC_d = TC_c \times 1.11$ ,  $TC_c = TC_a$ ) afecta a los valores de las penalizaciones en caso de fallo porque si suponemos que el tiempo de acceso a memoria no cambia y los ciclos son un 11% más largos, la caché necesitará esperar menos ciclos para recibir la respuesta de memoria.

$$T_{pf_{i-d}} = \left\lceil \frac{T_{pf_i}}{1.11} \right\rceil = \left\lceil \frac{56}{1.11} \right\rceil = 51 \text{ ciclos}$$

$$T_{pf_{lw-d}} = \left\lceil \frac{T_{pf_{lw}}}{1.11} \right\rceil = \left\lceil \frac{56}{1.11} \right\rceil = 51 \text{ ciclos}$$

$$T_{pf_{sw-d}} = \left\lceil \frac{T_{pf_{sw}}}{1.11} \right\rceil = \left\lceil \frac{44}{1.11} \right\rceil = 40 \text{ ciclos}$$

$$T_{lectura\ buffer\ escritura-d} = \left\lceil \frac{T_{lectura\ buffer\ escritura}}{1.11} \right\rceil = \left\lceil \frac{10}{1.11} \right\rceil = 10 \text{ ciclos}$$

El CPI, medido en ciclos de la nueva configuración, es:

$$\begin{aligned}
CPI_d &= 1 + m_{i-d} \times T_{pfi-d} \\
&\quad + f_{lw} \times m_{d-d} \times (T_{pfiw-d} + f_{lectura\ buffer\ escritura} \times T_{lectura\ buffer\ escritura-d}) \\
&\quad + f_{sw} \times f_{buffer\ escritura\ lleno} \times T_{pfs_{sw}-d} \\
&= 1 + 0.005 \times 51 + 0.20 \times 0.025 \times (51 + 0.12 \times 10) + 0.10 \times 0.08 \times 40 = 1.84 \text{ ciclos}
\end{aligned}$$

Y comparando los tiempos de ejecución:

$$\begin{aligned}
T_a &= NI \times CPI_a \times TC_a = NI \times 2.61 \times TC_c \\
T_d &= NI \times CPI_d \times TC_d = NI \times 1.84 \times TC_c \times 1.11 = 2.0424 \times NI \times TC_c \\
\frac{T_a}{T_d} &= \frac{2.61 \times NI \times TC_c}{2.0424 \times NI \times TC_c} = 1.28 \Rightarrow 28\% \text{ de mejora}
\end{aligned}$$

## Apartado e

Al usar el nuevo compilador, la frecuencia de aparición de las instrucciones y el número total de instrucciones cambian:

$$\begin{aligned}
NI_e &= NI - NI \times f_{alu} \times 0.20 - NI \times f_{lw} \times 0.10 - NI \times f_{sw} \times 0.10 \\
&= NI \times (1 - f_{alu} \times 0.80 - f_{lw} \times 0.90 - f_{sw} \times 0.90) = \\
&= NI \times (1 - 0.50 \times 0.20 - 0.20 \times 0.10 - 0.10 \times 0.10) = 0.87 \times NI \\
f_{alu-e} &= \frac{NI \times f_{alu} - NI \times f_{alu} \times 0.20}{NI_e} = \frac{NI \times 0.50 - NI \times 0.50 \times 0.20}{0.87 \times NI} = 0.4598 \\
f_{lw-e} &= \frac{NI \times f_{lw} - NI \times f_{lw} \times 0.10}{NI_e} = \frac{NI \times 0.20 - NI \times 0.20 \times 0.10}{0.87 \times NI} = 0.2069 \\
f_{sw-e} &= \frac{NI \times f_{sw} - NI \times f_{sw} \times 0.10}{NI_e} = \frac{NI \times 0.10 - NI \times 0.10 \times 0.10}{0.87 \times NI} = 0.1034
\end{aligned}$$

El nuevo CPI es:

$$\begin{aligned}
CPI_e &= 1 + m_i \times T_{pfi} \\
&\quad + f_{lw-e} \times m_d \times (T_{pfiw} + f_{lectura\ buffer\ escritura} \times T_{lectura\ buffer\ escritura}) \\
&\quad + f_{sw-e} \times f_{buffer\ escritura\ lleno} \times T_{pfs_{sw}} \\
&= 1 + 0.01 \times 56 + 0.2069 \times 0.05 \times (54 + 0.12 \times 10) + 0.1034 \times 0.08 \times 44 = 2.52 \text{ ciclos}
\end{aligned}$$

Y el tiempo de de ejecución es:

$$T_e = NI_e \times CPI_e \times TC_e = 0.87 \times NI \times 2.52 \times TC_c = 2.19 \times NI \times TC_c$$

Como  $T_e > T_d$ , la mejora del apartado d es preferible.