

AMPLIACIÓN DE ESTRUCTURA DE COMPUTADORES

Grado en Ingeniería Informática

Examen Final – Problemas – 17 de junio de 2024

IMPORTANTE: Es necesario entregar esta hoja junto con los folios del examen.

<input type="checkbox"/>	Grupo 1
<input type="checkbox"/>	Grupo 2
<input type="checkbox"/>	Grupo 3
<input type="checkbox"/>	Grupo 9

Apellidos: _____ Nombre: _____

1. (2 puntos) Dado un procesador con el siguiente pipeline: $F D X_1 M_1 X_2 M_2 W$, donde se calcula el destino de una instrucción de salto condicional en la etapa X_1 y la condición se evalúa en la etapa X_2 . Para los saltos incondicionales, la dirección destino se determina en la etapa D . Tenemos un 25% de instrucciones de salto, siendo los saltos condicionales el doble de frecuentes que los incondicionales. Se sabe, también, que el 75% de los saltos condicionales se toman. Suponiendo que el resto de las instrucciones no introducen ningún tipo de detención (el CPI sin considerar las instrucciones de salto es 1), se pide:
 - a) (0.5 puntos) Calcula el CPI para la versión del procesador que utiliza la estrategia de detener el cauce para todos los saltos (condicionales e incondicionales).
 - b) (0.75 puntos) Calcula el CPI para la versión del procesador que sigue una política de predicción estática de saltos de tipo *No-Tomado*, ¿qué mejora de rendimiento se consigue con respecto al apartado a)?
 - c) (0.75 puntos) En lugar de la opción anterior, se añade un predictor dinámico en la etapa X_1 , pero mientras el salto llega a dicha etapa se sigue usando predicción estática como *No-Tomado*. Sabiendo que la tasa de acierto del predictor dinámico es del 99%, calcula el nuevo CPI del procesador. ¿Cuál es el rendimiento ahora frente a la estrategia del apartado b)?
2. (2 puntos) Tenemos un procesador con cachés de primer nivel separadas para instrucciones y datos. Nos piden diseñar la caché de datos. Ésta tiene que ser de 32 KB y asociativa por conjuntos de 8 vías con un tamaño de bloque de 64 bytes. Hay que decidir si la queremos virtual o físicamente direccionada o virtual y físicamente etiquetada, basándonos únicamente en su rendimiento. El tiempo necesario para leer/escribir un bloque en memoria principal es de 50 ciclos. La caché tiene un tiempo de servicio en caso de acierto de 2 ciclos, una tasa de fallos del 5%, no está segmentada y dispone de un buffer de post-escritura que nunca se llena para las aplicaciones ejecutadas. El TLB de datos tiene un tiempo de servicio en caso de acierto de 1 ciclo, una tasa de fallos del 2% y la penalización es de dos accesos a memoria, por tanto, 100 ciclos. Se pide:
 - a) (0.5 puntos) Calcula el tiempo medio de acceso para datos en caso de una caché físicamente direccionada y etiquetada.
 - b) (0.5 puntos) Calcula el tiempo medio de acceso para datos en caso de una caché virtualmente direccionada y físicamente etiquetada.
 - c) (0.5 puntos) Calcula el tiempo medio de acceso para datos en caso de una caché virtualmente direccionada y etiquetada.
 - d) (0.3 puntos) Por qué no tendría sentido calcular el tiempo medio de acceso para una caché físicamente direccionada y virtualmente etiquetada.
 - e) (0.2 puntos) Qué decisión tomarías y por qué nos vale sólo con calcular el tiempo medio de acceso.

SOLUCIÓN EJERCICIO 1

- a) Puesto que el número de instrucciones y el tiempo de ciclo no cambia, podemos realizar la comparativa centrándonos en el CPI. Para la estrategia de insertar ciclos de parada tenemos 1 ciclo de penalización para los saltos incondicionales y 4 ciclos para los condicionales:

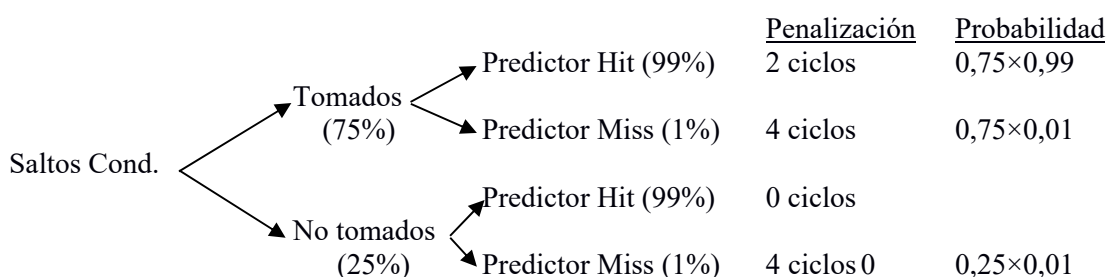
$$CPI_{\text{parada}} = 1 + 0,25 \times 1/3 \times 1 + 0,25 \times 2/3 \times 4 = \mathbf{1,75 \text{ ciclos}}$$

- b) Para la estrategia de predecir estáticamente los saltos como no tomado, solo los saltos condicionales tomados tienen penalización. Los saltos incondicionales no se ven afectados por esta estrategia:

$$CPI_{\text{NT}} = 1 + 0,25 \times 1/3 \times 1 + 0,25 \times 2/3 \times 0,75 \times 4 = \mathbf{1,58 \text{ ciclos}}$$

Hay una mejora del 11% con respecto al apartado a).

- c) Al igual que en el apartado anterior, la adición del predictor de saltos solo tiene consecuencias para los saltos condicionales, que ahora pueden saber si hay que saltar o no sin tener que esperar a la etapa X₂ (cuando el predictor acierta). Cuando añadimos el predictor en la etapa X₁ se presentan las siguientes situaciones:



Con lo que el $CPI_{\text{predictor}} = 1 + 0,25 \times 1/3 \times 1 + 0,25 \times 2/3 \times [0,75 \times 0,99 \times 2 + 0,01 \times 4] = \mathbf{1,34 \text{ ciclos}}$

En este caso la máquina con el predictor es un **18% más rápida** que la del apartado anterior.

SOLUCIÓN EJERCICIO 2

- a) Físicamente direccionada y etiquetada. Se accede primero al TLB (primer corchete) y luego a la caché (segundo corchete).

$$Tam = [1 + 0,02 \times 100] + [2 + 0,05 \times 50] = \mathbf{7,50 \text{ ciclos}}$$

- b) Virtualmente direccionada y físicamente etiquetada. Se accede en paralelo al TLB y a la caché (primer corchete). Solo en caso de fallo de TLB no se puede realizar el acceso en paralelo y añadiríamos el tiempo de fallo al TLB (segundo corchete).

$$Tam = [2 + 0,05 \times 50] + [0,02 \times 100] = \mathbf{6,50 \text{ ciclos}}$$

- c) Virtualmente direccionada y etiquetada. Se accede primero a la caché (primer corchete) y sólo en caso de fallo al TLB (segundo corchete).

$$Tam = [2] + [0,05 \times (1 + 0,02 \times 100 + 50)] = \mathbf{4,65 \text{ ciclos}}$$

- d) Porque en una caché el acceso al índice comienza antes que la comparación con la etiqueta y si tenemos ya la dirección física a la hora de acceder al índice, la podemos usar para la etiqueta también.

- e) Implementaríamos una **caché virtual**. Nos vale sólo con el tiempo medio de acceso ya que el diseño no influye en otros factores como el número de accesos a la caché.