

AMPLIACIÓN DE ESTRUCTURA DE COMPUTADORES

Grado en Ingeniería Informática

Examen Final – Problemas – 14 de mayo 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) Supongamos el procesador DLX estudiado en clase de 5 etapas (**F D X M W**), que no implementa la técnica de adelantamiento y en el que los saltos se resuelven en la cuarta etapa. Sabiendo que, en promedio, el 40% de las instrucciones son de tipo ALU, el 20% son cargas, el 15% son almacenamientos y el 25% son saltos condicionales (de los cuales se sabe que el 75% son tomados):
 - a) (0,50 puntos) Calcular el CPI total, sabiendo que el 20% de las instrucciones de carga y el 15% de las instrucciones de las de ALU van seguidas de una instrucción ALU que usa el dato leído de memoria y calculado por la ALU respectivamente, y que los saltos condicionales se predicen estáticamente como no tomados.
 - b) (0,50 puntos) Se mejora el procesador implementando la técnica de adelantamiento y se introduce en la etapa *D* un predictor de saltos dinámico con una tasa de aciertos del 92% (los saltos siguen siendo resueltos en la cuarta etapa y las instrucciones buscadas de forma errónea son anuladas). ¿Qué CPI se obtiene ahora?
 - c) (0,50 puntos) En lugar del predictor dinámico se incluye un BTB en su lugar, en la etapa *F*, con dos puertos para permitir la actualización a la vez que se predice sin que haya que esperar (los saltos siguen siendo resueltos en la cuarta etapa y las instrucciones buscadas de forma errónea son anuladas). La probabilidad de acertar en el BTB es del 90% y la de acertar la predicción es del 80%. ¿Qué CPI se obtiene ahora?
 - d) (0,50 puntos) Se plantea la inclusión de las instrucciones de salto condicional DBEZ y DBNZ al repertorio del DLX. Concretamente, se trata de saltos en los que el operando que se emplea para la condición (igual a cero o distinto de cero respectivamente) es el contenido de una posición de memoria después de restarle una unidad. La instrucción DBEZ $Cte(Reg), Etiqueta \text{ salta si } Memoria[Reg+Cte]-1 \text{ es igual a cero}$. La instrucción DBNZ $Cte(Reg), Etiqueta \text{ salta si } Memoria[Reg+Cte]-1 \text{ es distinto de cero}$. Indica la secuencia de instrucciones DLX a la que estas nuevas instrucciones de salto sustituirían y explica las etapas que como mínimo tendría que tener el nuevo cauce para poder ejecutar estas nuevas instrucciones.
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, asociativa por conjuntos de 8 vías con un tamaño de bloque de 64 bytes y físicamente etiquetada. Tenemos que decidir si la queremos virtual o físicamente direccionada, basándonos en su rendimiento. El tiempo necesario para leer/escribir un bloque en memoria principal es de 40 ciclos. La caché tiene un tiempo de servicio en caso de acierto de 2 ciclos, una tasa de fallos del 6%, **no** está segmentada y **no** dispone de un buffer de post-escritura. El 25% de los bloques reemplazados son sucios. El TLB de datos tiene un tiempo de servicio en caso de acierto de 1 ciclo, una tasa de fallos del 4% y la penalización es de tres accesos a memoria, por tanto, 120 ciclos.
 - a) (0,75 puntos) Calcula el tiempo medio de acceso para datos en caso de una caché físicamente direccionada.
 - b) (0,75 puntos) Calcula el tiempo medio de acceso para datos en caso de una caché virtualmente direccionada.
 - c) (0,50 puntos) El diseño se restringe a cachés físicamente etiquetadas, ya que las cachés virtualmente etiquetadas tienen 3 problemas principales. Describe brevemente dichos problemas.

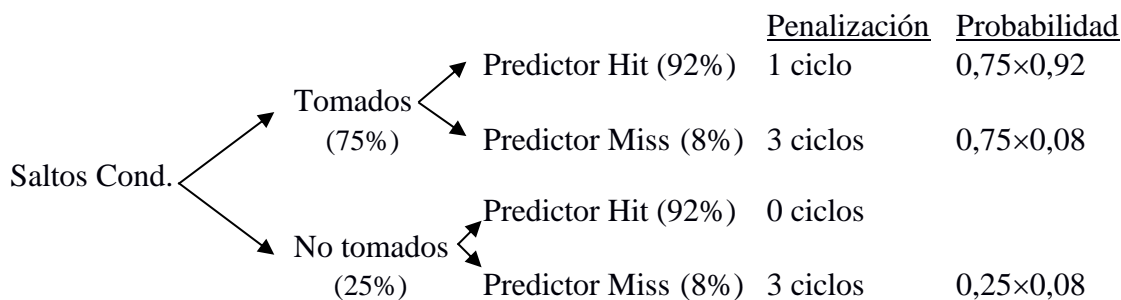
SOLUCIÓN EJERCICIO 1

- a) Para las instrucciones de carga y ALU se introducen 2 ciclos de parada cuando la instrucción siguiente (en este caso una instrucción ALU) consume el dato producido por ellas. Para las instrucciones de salto se incurre en una penalización de 3 ciclos para los saltos que son tomados. Con lo que:

$$CPI_{Total} = CPI_{Ideal} + \text{Ciclos de reloj de detención por instrucción}$$

$$CPI_{Total} = 1 + 0,4 \times 0,15 \times 2 + 0,2 \times 0,2 \times 2 + 0,25 \times 0,75 \times 3 = \mathbf{1,7625 \text{ ciclos}}$$

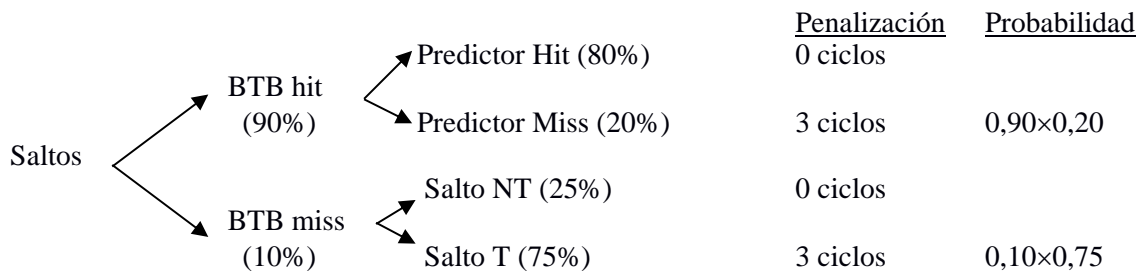
- b) Al implementar los adelantamientos, sólo las instrucciones ALU que siguen a una carga y que requieran el valor buscado de memoria habrán de detenerse 1 ciclo. Cuando la instrucción que produce el resultado es de tipo ALU se utilizan los caminos de adelantamiento y no se para. Para los saltos condicionales tenemos que se ha añadido un predictor en la etapa *D*, pero que los saltos siguen siendo resueltos al final de la etapa *M*. De esta forma se asume que en la etapa *D* se conoce la dirección destino de un salto condicional y no es sino hasta la etapa *M* cuando se comprueba la condición. Los casos que pueden darse son los siguientes:



En general, cuando el predictor falla, siempre son 3 ciclos de penalización y por tanto se puede simplificar a $0,08 \times 3$. Con lo que:

$$CPI_{Total} = 1 + 0,2 \times 0,2 \times 1 + 0,25 \times (0,75 \times 0,92 \times 1 + 0,08 \times 3) = \mathbf{1,2725 \text{ ciclos}}$$

- c) Cuando se sustituye el predictor condicional y, en su lugar, se incorpora un BTB en la etapa de *fetch*, los casos que se pueden dar son los siguientes:



Con lo que

$$CPI_{Total} = 1 + 0,2 \times 0,2 \times 1 + 0,25 \times (0,9 \times 0,2 \times 3 + 0,1 \times 0,75 \times 3) = \mathbf{1,2312 \text{ ciclos}}$$

- d) Cada instrucción DBEZ/DBNZ Cte (Reg), Etiqueta sustituiría a la siguiente secuencia de instrucciones:

```
LD Rx, 0 (Reg)
SUBI Rx, Rx, 1
BEQZ/BNEZ Rx, Etiqueta
```

Para poder incluir estas instrucciones tenemos que ampliar el número de etapas de nuestro *pipeline*. Concretamente, añadimos una nueva etapa para restar uno al dato leído de memoria. En esta etapa (que denominaremos *X₂*) se resolverán por lo tanto los saltos DBEZ/DBNZ. Así pues, el nuevo *pipeline* quedaría como sigue: ***F D X₁ M X₂ W***.

SOLUCIÓN EJERCICIO 2

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

$$T_{ma} = [1 + 0,04 \times 120] + [2 + 0,06 \times (40 + 0,25 \times 40)] = \mathbf{10,80 \text{ ciclos}}$$

b) Virtualmente direccionada. 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).

$$T_{ma} = [2 + 0,06 \times (40 + 0,25 \times 40)] + [0,04 \times 120] = \mathbf{9,80 \text{ ciclos}}$$

c) Las cachés virtuales tienen 3 problemas:

1. Comprobar la protección de memoria;
2. Cambios de contexto;
3. Sinónimos o alias. (Ver transparencias de teoría)