AMPLIACIÓN DE ESTRUCTURA DE COMPUTADORESº

Grado en Ingeniería Informática

Examen Final – Problemas – 14 de junio de 2023

- ❑ Grupo 1
- ❑ Grupo 2
- ❑ Grupo 3
- ❑ Grupo 9

Apellidos, Nombre:_____

1. *(1.5 puntos)* Tenemos un programa que ejecuta 4 millones de instrucciones, de las que 2 millones son operaciones aritméticas de tipo entero, 1 millón son accesos a memoria y 1 millón corresponden a operaciones en coma flotante. Sabemos también que, en un computador determinado, este programa obtiene 10 MFLOPS y su CPI es 1,5. Se pide:

a) *(0,5 puntos)* Calcule el tiempo de ejecución, la frecuencia MIPS, y la frecuencia del procesador.
b) *(0,5 puntos)* Utilizando la ley de Amdahl, calcule cuánto habría que mejorar las operaciones aritméticas enteras para obtener una aceleración global (*speedup*) del 30% (suponiendo que solo se mejoran las de tipo entero).
c) *(0,5 puntos)* Utilizando la ley de Amdahl, calcule ahora la máxima aceleración global (*speedup*) que se puede obtener sin mejorar las operaciones aritméticas enteras (es decir, mejorando todas las demás operaciones pero no las enteras, al contrario de lo supuesto en el apartado anterior).

2. *(2.5 puntos)* Tenemos 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 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. 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 maneja los saltos introduciendo ciclos de parada.
b) *(0.5 puntos)* Calcula el CPI para la versión del procesador que predice los saltos estáticamente como no tomados, ¿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%, ¿cuál es el rendimiento ahora frente a sólo predicción estática del salto como no tomado del apartado b)? Indica qué salto(s) se vería(n) beneficiado(s) por este esquema.
d) *(0.75 puntos)* Como alternativa a la opción c), se añade un BTB en la etapa F con dos puertos para permitir la actualización a la vez que se predice, sin que tengamos que detenernos. Se ha visto que la probabilidad de encontrar una entrada en el BTB para un salto es del 90%, mientras que la probabilidad de acertar la predicción de un salto condicional es del 95%. Si el salto no está en el BTB se aplica una predicción estática de no tomado. Desde el punto de vista del rendimiento obtenido, ¿es preferible esta opción o la del apartado c)?
Indica qué salto(s) se vería(n) beneficiado(s) por este esquema.

SOLUCIONES

PROBLEMA 1

a)

Dado que se obtienen 10 MFLOPS al ejecutar 1 millón de operaciones en coma flotante, aplicando la fórmula de los MFLOPS, se obtiene que:

Tiempo de ejecución = Num. Instrucciones PF / (MFLOPS×10^6) = 1000000 / (10×10^6) = 1/10 = **0,1 segundos**

Sabiendo lo anterior, los MIPS serán: (4 millones de instrucciones) / (0,1 segundos) = **40 MIPS**

Para calcular la frecuencia del procesador podemos dividir el número total de ciclos, que será el número de instrucciones multiplicado por los ciclos por instrucción (CPI = 1,5), y dividir entre el tiempo.
Por tanto, f = $4 \times 10^6 \times 1{,}5 / 0{,}1 = 6 \times 10^7$ Hz = **60 MHz**

b)

Sabemos la cantidad de instrucciones de cada tipo y sabemos que el CPI promedio global es de 1,5 ciclos. Así que podemos calcular la cantidad de ciclos usado por cada tipo de instrucción como sigue:

Instrucciones enteras = 2 Mill × 1,5 ciclos = 3 Mill. de ciclos
Instrucciones PF = 1 Mill × 1,5 ciclos = 1,5 Mill. de ciclos
Instrucciones memoria = 1 Mill × 1,5 ciclos = 1,5 Mill. de ciclos

Con los datos anteriores podemos calcular la fracción de tiempo que representan las operaciones aritméticas respecto del tiempo total (6 Mill. de ciclos): 3 Mill / 6Mill = 0,5 (o 50% de la fracción del tiempo)

Planteando la ecuación de la ley de Amdahl tenemos que:

$$1.3 = \frac{1}{(1-0.5) + \frac{0.5}{s_{\text{int}}}} \Rightarrow s_{\text{int}} = 1.857$$

Es decir, para conseguir un 30% de mejora global, habría que hacer las operaciones enteras un **85,7%** más rápidas.

c)

Para ello, supongamos aceleraciones infinitas para las operaciones en coma flotante y de memoria (que recordemos que representan un 50% de la fracción del tiempo total, como se vio en el apartado anterior) mientras que las operaciones enteras (que representa el otro 50% del tiempo) se mantienen igual. Por tanto:

$$S_{\max} = \frac{1}{0.50 + \frac{(1-0.50)}{\infty}} = \frac{1}{0.50} = 2$$

Es decir, la máxima aceleración que se podría obtener es 2× con lo que el programa iría el doble de rápido, si fuésemos capaces de optimizar hasta el infinito las operaciones en coma flotante y las de memoria.

PROBLEMA 2

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:

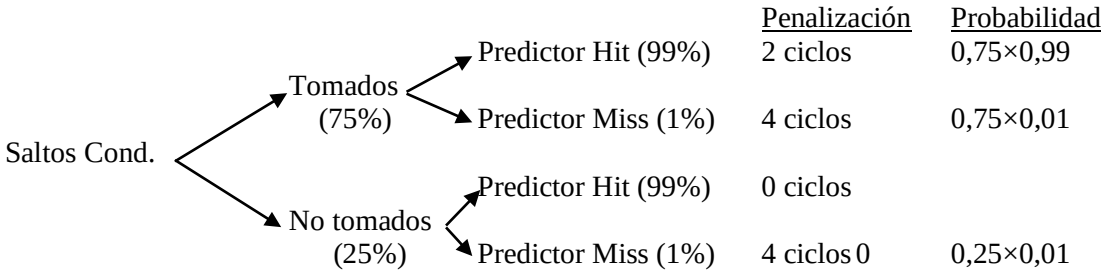
$CPI_{parada} = 1 + 0{,}25\times1/3\times1 + 0{,}25\times2/3\times4 =$ **1,75 ciclos**

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:

$CPI_{NT} = 1 + 0{,}25\times1/3\times1 + 0{,}25\times2/3\times0{,}75\times4 =$ **1,58 ciclos**

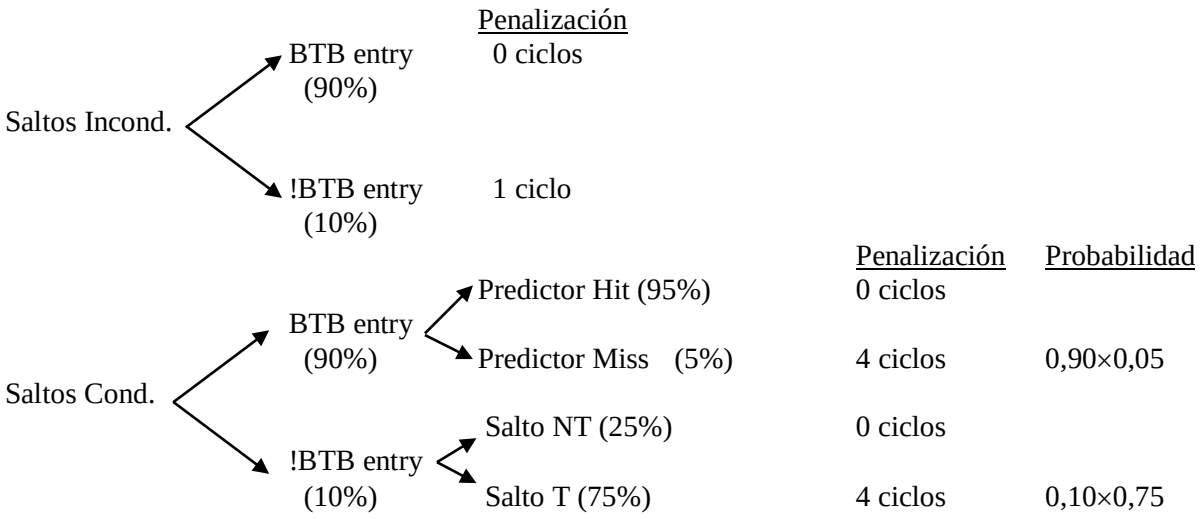
Hay una mejora del 11% con respecto al apartado a).

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 X_2 (cuando el predictor acierta). Cuando añadimos el predictor en la etapa X_1 se presentan las siguientes situaciones:



Con lo que el $CPI_{predictor} = 1 + 0{,}25\times1/3\times1 + 0{,}25\times2/3\times[0{,}75\times0{,}99\times2 + 0{,}01\times4] =$ **1,34 ciclos**
En este caso la máquina con el predictor es un **18% más rápida** que la del apartado anterior.

d. En este caso los dos tipos de salto (condicionales e incondicionales) se van a ver beneficiados de la inclusión del BTB en el diseño. Para los saltos incondicionales, cuando se encuentre una entrada en el BTB se conocerá la dirección destino en la propia etapa IF, con lo que la penalización será 0. Para los condicionales, cuando se encuentre una entrada y se acierte la predicción se reducirá a 0 la penalización. Así pues cuando añadimos el BTB, los casos que se pueden dar son los siguientes:



Luego, $CPI_{BTB} = 1 + 0{,}25\times1/3\times0{,}10\times1 + 0{,}25\times2/3\times[0{,}90\times0{,}05\times4 + 0{,}10\times0{,}75\times4] =$ **1,09 ciclos**
Por lo tanto, la máquina con el BTB es un **23% más rápida** que la del apartado c)