Apellidos _____ Nombre _____ Grupo___

Arquitectura e Ingeniería de Computadores. Examen Final (Teoría – parte primer cuatrimestre). 22/06/2009

Instrucciones.- Cada pregunta consta de cinco respuestas, y cada una de las respuestas puede ser cierta o falsa. Marque con un aspa las respuestas que considere ciertas y deje en blanco las que considere falsas. Si considera que alguna respuesta es ambigua y, por tanto, podría considerarse cierta o falsa en función de la interpretación, ponga una llamada y explique sus argumentos al dorso de la hoja.
No se permite la utilización de calculadora.
Puntuación.- Pregunta con todas las respuestas acertadas: 1 punto. Pregunta con un fallo: 0,6 puntos. Pregunta dos fallos 0,2 puntos. Pregunta con más de dos fallos 0 puntos. La teoría del primer cuatrimestre supone la mitad de la nota del primer cuatrimestre. Tanto la nota de teoría como la de problemas se normalizarán para que el primer cuatrimestre tenga un peso del 65% en la nota final de la asignatura.

1. Supongamos la arquitectura básica del DLX, pero segmentado en siete etapas, donde el acceso a la memoria de instrucciones consume dos ciclos de reloj y el acceso a la memoria de datos también. La máquina posee anticipación de operandos (forwarding). Las instrucciones de salto se resuelven en la etapa EX, con predicción estática de "salto no tomado". Además la etapa EX incluye un multiplicador de punto flotante, un sumador de punto flotante y un divisor con tiempos de cálculo 5, 2 y 20 ciclos de reloj, respectivamente, todos ellos segmentados con intervalo de iniciación 1. Marque cuáles de las siguientes afirmaciones son correctas:

☒ a) Un grupo de instrucciones consecutivas de la forma
```
ld    f8, 0(r5)
addd  f4, f8, f2
addd  f6, f8, f10
```
provoca una penalización de dos ciclos de reloj.

☒ b) Por cada salto tomado se produce una penalización de tres ciclos de reloj

☒ c) La presencia en un programa de dos instrucciones consecutivas de la forma
```
muld  f8, f4, f6
muld  f6, f0, f8
```
provocará cuatro ciclos de penalización

☐ d) Si en un cierto programa el 30% de las instrucciones son saltos, el 60% de los saltos se toman, y no existen riesgos estructurales ni de datos, entonces el CPI es 1,18.

☒ e) En esta arquitectura es necesario que exista un mecanismo para la detección y el tratamiento de los riesgos EDE.

2. Marque cuáles de las siguientes afirmaciones sobre multithreading son correctas:

☒ a) La conmutación entre threads debe ser mucho más rápida que la conmutación entre procesos.

☐ b) En multithreading simultáneo se realiza una conmutación de thread en cada ciclo de reloj.

☐ c) La explotación de paralelismo a nivel de thread, en la ejecución de un programa, es transparente al programador.

☒ d) En multithreading de grano grueso todas las instrucciones lanzadas a ejecución en un ciclo de reloj deben pertenecer al mismo thread

☐ e) La arquitectura IBM Power 5 es un ejemplo de multithreading de grano grueso

3. Marque cuáles de las siguientes afirmaciones sobre memoria cache son ciertas:

☒ a) En un sistema de memoria cache de dos niveles (L1, L2) la tasa de fallos local en L1 coincide con la tasa de fallos global en L1.

☐ b) Si el procesador genera 10.000 referencias a memoria y se producen 200 fallos en L1 y 40 en L2, la tasa de fallos global de L1 es del 2,4%.

☒ c) En las mismas condiciones del apartado b), la tasa de fallos global de L2 es del 0,4%.

☐ d) Supongamos un sistema con memoria virtual paginada con páginas de 64KB. Si implementamos en este sistema una cache directa virtualmente accedida, pero físicamente marcada, con tamaño de bloque de 128 bytes, el nº máximo de marcos de bloque (o líneas) de la cache es 1024.

☐ e) La cache no bloqueante reduce la tasa de fallos.

Arquitectura e Ingeniería de Computadores. Examen Parcial (Problemas). 22/06/2009

Problemas correspondientes al 1er cuatrimestre

1) Tras compilar un programa escrito en lenguaje de alto nivel se ha obtenido el código para DLX que se muestra a continuación. Las etapas del procesador segmentado son las siguientes:

IF: Búsqueda de la instrucción a ejecutar.
ID: Decodificación de la instrucción y lectura de los registros operando (2º semiciclo).
EX: Ejecución en la ALU.
MEM : Acceso a memoria (instrucciones de carga y almacenamiento).
WB: Escritura del resultado sobre el registro destino (1º semiciclo).

```
L0: sub r1, r2, r3
    ld r5, 5 (r1)
    add r2, r1, r5
    add r3, r5, r2
    sd  10(r6), r3
    sgt r4, r6, r11          ; si r6 > r11 → r4 = 1, si no r4 = 0
    bnez r4,end
    nop                      ; slot 1
    beqz r0, L1
    nop                      ; slot 2
    add r2, r3, r4
L1: ld r1, 15 (r8)
    and r3, r8, r5
    add r3, r1, r7
    seqi r2, r7, #0          ; si r7 = 0 → r2 = 1, si no r2 = 0
    beqz r2, end
    nop                      ; slot 3
    sub r1, r7, r9
end: subi r1, #1, r1
```

El slot de salto es de una instrucción (el PC se escribe en la etapa ID en las instrucciones de salto). Existen memorias separadas para datos e instrucciones. No se aplica la técnica de cortocircuito. Suponiendo que inicialmente el registro r11 contiene el valor 1000, el registro r6 contiene el valor 9 y el registro r7 contiene un valor no nulo, se pide:

i) ¿Cuántos ciclos toma la ejecución completa del código? Indicar las dependencias existentes y los ciclos de penalización de cada una. (**1.5 puntos**)

ii) Si la implementación del procesador utiliza saltos retardados, indica para cada uno de los dos saltos beqz (beqz r0, L1 y beqz r2, end) cuál es la mejor opción para rellenar el delay slot: instrucción previa al salto, instrucción del destino o instrucción de la rama que sigue en secuencia (fall-through). ¿Cuántos ciclos toma ahora la ejecución completa del código como resultado de aplicar esta optimización? **(1 punto)**

iii) Si sobre la situación inicial (sin optimizar los saltos) se aplica la técnica de cortocircuito, ¿Qué nuevo número de ciclos totales obtenemos en la ejecución? **(1 punto)**

PUNTUACIÓN: 3.5 PUNTOS

2) Sea un programa compuesto por dos procedimientos N y M que se ejecutan en secuencia. N contiene cálculos en punto flotante, mientras que M sólo contiene cálculos en punto fijo.
Este programa se ejecuta sobre un procesador A que no tiene UFs de punto flotante, repartiéndose el tiempo total de ejecución (150 segundos) a partes iguales entre los dos procedimientos. Después se ejecuta de nuevo sobre un procesador B, con las mismas características del A, pero que contiene que contiene UFs de punto flotante, resultando un speedup de 1.5 al pasar del procesador A al B.

Se pide:

a) ¿En qué factor se ha reducido el tiempo de ejecución del procedimiento N en el procesador B respecto al A? (**1 punto**)
b) En la ejecución sobre el procesador B ¿qué porcentaje del tiempo de ejecución es consumido por el procedimiento N? (**0.5 puntos**)

PUNTUACIÓN: 1.5 PUNTOS

SOLUCIÓN

1)

i) Dependencias LDE (lectura después de escritura):

1-2 → 2 ciclos
2-3 → 2 ciclos
3-4 → 2 ciclos
4-5 → 2 ciclos
6-7 → 2 ciclos
12-14 → 1 ciclo
15-16 → 2 ciclos

Ciclos = 17 (instrucciones) + 13 (penalizaciones) + 4 (llenado pipe) = 34 ciclos

ii) Slot 2 (beqz r0, L1): el salto es incondicional con lo que nos servirían la instrucción destino y la previa; cogeremos la destino, ya que la previa es una nop.

Slot 3 (beqz r2, end): el salto se toma, luego cogeremos la instrucción destino para rellenar el slot.

Nos ahorramos por tanto 2 ciclos de detención como resultado de la desaparición de las dos instrucciones nop. Con ello el nuevo número de ciclos es:

Ciclos = 34 - 2= 32

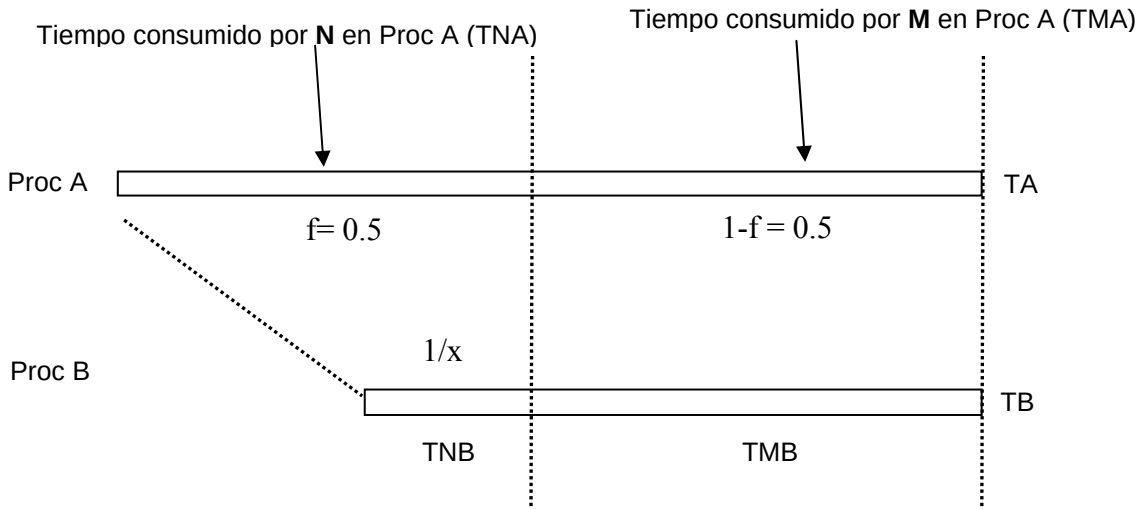
iii) Si aplicamos ahora la técnica de circuito tendríamos las siguientes penalizaciones:

Dependencias LDE (lectura después de escritura):

2-3 → 1 ciclo
6-7 → 1 ciclo (evitable si asumimos posibilidad de realimentación combinacional)
15-16 → 1 ciclo (evitable si asumimos posibilidad de realimentación combinacional)

Ciclos = 17 (instrucciones) + 3 (penalizaciones) + 4 (llenado pipe) = 24 ciclos
Los ciclos serían 22 asumiendo realimentación combinacional

2)



a) Speedup = TA/TB = 1.5 = (TNA+TMA)/(TNB+TMB)

Como TNA=TMA, TNB=TNA/x, y TMA=TMB, entonces

1.5 = (2TMA / ((TMA/x) + TMA)), de donde x = 3 (factor de reducción)

b) (TNA/3) / ((TNA/3) + TNA) = ¼ , de donde el porcentaje pedido es el 25%