

Arquitectura e Ingeniería de Computadores. Examen Final (Teoría – parte primer cuatrimestre). 21/06/2010

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. Para ejecutar el programa P un procesador ejecuta 10^{12} instrucciones, invirtiendo un promedio de 2 ciclos de reloj por instrucción. El 10% de las instrucciones son de punto flotante. La frecuencia de reloj es 2 GHz. Marque cuáles de las siguientes afirmaciones son correctas.

- a) La duración media de una instrucción es 2 ns
- b) El rendimiento es 1000 MIPS.
- c) El rendimiento es 200 MFLOPS.
- d) Si los cálculos enteros consumen el 20% del tiempo de ejecución, entonces los ciclos de reloj en promedio consumidos por cada instrucción de PF son 16.
- e) Supongamos que los cálculos enteros consumen el 20% del tiempo de ejecución. Si introducimos un coprocesador matemático que reduce el tiempo de ejecución de las instrucciones en PF a una cuarta parte, entonces se obtiene un Speedup de 2.

2. Supongamos una Unidad de Ejecución que utiliza el algoritmo de Tomasulo con especulación basada en el uso del buffer de reordenamiento (ROB). Marque cuáles de las siguientes afirmaciones son correctas.

- a) En la fase WRITE, los resultados generados por las UFs se escriben en las estaciones de reserva y en el ROB, pero no en los registros.
- b) Las instrucciones de un programa realizan la fase COMMIT en el mismo orden en que realizan la fase ISSUE.
- c) Cuando una instrucción realiza la fase ISSUE puede recibir sus operandos desde los registros o desde el ROB.
- d) Cuando una instrucción de salto correctamente predicha alcanza la cabecera del ROB es suficiente con eliminar la instrucción del ROB.
- e) Cada entrada del ROB está compuesta por los siguientes campos: (Destino, TAG, Valor y Listo).

3. Marque cuáles de las siguientes afirmaciones sobre jerarquía de memoria son correctas.

- a) Al ejecutar un programa en un sistema con una memoria cache de un nivel, el número de fallos iniciales depende del tamaño de la cache.
- b) Supongamos un sistema de memoria virtual paginada con páginas de 64 Kbytes. Si implementamos en este sistema una cache directa virtualmente accedida, pero físicamente marcada, con tamaño de bloque de 32 bytes, entonces la cache puede tener un máximo de 2048 bloques.
- c) Al ejecutar un programa en un sistema con memoria cache totalmente asociativa nunca se producen fallos de conflicto.
- d) El alargamiento de arrays en una técnica para reducir el número de fallos de conflicto.
- e) La penalización media por instrucción (en ciclos) como consecuencia de no tener una cache perfecta, viene dada por la expresión:

$$\left[\left(\text{Promedio de accesos a memoria por instrucción} \right) \times \left(\text{Tasa de fallos} \right) \times \left(\text{Ciclos de penalización por fallo} \right) \right]$$

Nombre-----Grupo-----

1) En un DLX con segmentación ejecutamos el siguiente fragmento de código:

```
bucle: SUBI r1,r1,#1
      LD F0, 0(r1)
      MULF F2,F4,F0
      ADDI r2,r2,#1
      SD 0(r2), F2
      DIVF F0,F2,F4
      ADDF F4,F2,F2
      LD F0, 0(r2)
      MULF F2,F0,F0
      ADDF F4,F0,F6
      MULF F4,F4,F0
      BNEZ r1,bucle
end:  ADDI r3,r3,#1
```

Se supone que:

- Un dato se puede escribir en un registro y leer su valor en el mismo ciclo.
- Se dispone de lógica de cortocircuito (*forwarding*).
- Los saltos tienen tratamiento de instrucción entera, se resuelven en la etapa de decodificación (ID) y se espera a su resolución antes de buscar la siguiente instrucción.
- La detección de los riesgos estructurales y LDE y generación de paradas se realiza en la etapa de decodificación.
- Los riesgos EDE se resuelven mediante inhibición de escritura.
- Dos instrucciones no pueden acceder simultáneamente a la etapa de acceso a memoria ni tampoco a la de escritura en el banco de registros.
- El registro r1 contiene inicialmente el valor 2.
- Se dispone de las siguientes unidades funcionales:

UF	Cantidad	Latencia	Segmentación
FP ADDD	1	2	Sí
FP MULF	1	3	Sí
FP DIVF	1	5	No
INT ALU	1	1	No

- A) Representar el diagrama instrucción-tiempo para la primera iteración e indicar los cortocircuitos realizados. Indicar claramente las paradas y sus causas. A la vista del diagrama obtenido, y sin necesidad de desarrollar todas las iteraciones, calcular el número de ciclos que toma la ejecución completa del código. **(2.5 pts)**
- B) ¿Cómo se vería afectado el valor obtenido en el apartado anterior en el caso de que se dispusiese de un predictor de saltos del tipo BTB, al que se accediese en la etapa IF y se obtuviese su respuesta al final de dicha fase? Considerar el estado inicial del predictor como “salto tomado”. **(1 pts)**

2) Disponemos de un procesador dotado de planificación dinámica de instrucciones según el método de Tomasulo. Se dispone de las unidades funcionales que se muestran en la tabla siguiente:

UF	Cantidad	Latencia	Segmentación
FP ADDD	2	3	Sí
FP MUL/DIV	1	4/10	No

Suponer que en un mismo ciclo no se puede escribir un dato en el bus común e iniciar la ejecución de una operación que estuviera en espera de ese dato, y que existe una estación de reserva por cada unidad funcional.

Para la secuencia de instrucciones de la tabla siguiente, indica en qué ciclo (o ciclos) de reloj se realiza cada una de sus fases. Indica qué tipos de riesgos se producen. **(1.5 pts)**

Instrucción	Issue Emisión	Ejecución	Escribe resultado
ADDD F2,F2,F4			
ADDD F0,F4,F0			
MULTD F4,F2,F0			
ADDD F6, F0,F0			
DIVD F4,F4,F6			
ADDD F2,F6,F6			
ADDD F0,F2,F6			
MULTD F2,F2,F0			

SOLUCIÓN

1)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
SUBI r1,r1,#1	IF	ID	EX	M	WB																						
LD F0, 0(r1)		IF	ID	EX	M	WB																					
MULD F2,F4,F0			IF	ID	ID	M1	M2	M3	M	WB																	
ADDI r2,r2,#1				IF	IF	ID	EX	M	WB																		
SD 0(r2), F2						IF	ID	ID	EX	M	WB																
DIVD F0,F2,F4							IF	ID	D1	D2	D3	D4	D5	M	WB												
ADDD F4,F2,F2								IF	ID	A1	A2	M	WB														
LD F0, 0(r2)									IF	ID	EX	M	WB														
MULD F2,F0,F0										IF	IF	ID	M1	M2	M3	M	WB										
ADDD F4,F0,F6											IF	ID	A1	A2	M	WB											
MULD F4,F4,F0												IF	IF	ID	M1	M2	M3	M	WB								
BNEZ r1,bucle														IF	IF	ID	EX	M	WB								
SUBI r1,r1,#1																		IF	IF	ID						

XX: Parada por riesgo LDE

XX: Parada por riesgo estructural (siguiente etapa ocupada)

XX: Riesgo de control (salto espera a resolverse)

XX: Riesgo EDE (inhibición de la escritura)

XX: Riesgo estructural: dos instrucciones intentan acceder simultáneamente a memoria

A) La ejecución completa de la primera iteración conlleva desde el ciclo 1 hasta el 23. La segunda iteración comienza en el ciclo 20, pero tenemos la ejecución extra de la instrucción etiquetada como end, así pues la ejecución completa del código tomará 43 ciclos

B) Si el predictor inicialmente estuviese a "salto tomado" nos ahorraríamos el ciclo de parada de control de la primera iteración ya que el predictor acierta, pero por el contrario la primera instrucción de la segunda iteración tendría un ciclo de parada en decodificación por coincidencia de etapas de acceso a memoria con el MULD anterior.. Al finalizar la segunda iteración el procesador comenzaría a ejecutar la instrucción de resta por tercera vez, pero se cancelaría, luego se compensa con el ciclo de parada de control del apartado a. Por lo tanto el número total de ciclos queda invariable (43)

2)

Instrucción	Issue Emisión	Ejecución	Escribe resultado
ADDD F2,F2,F4	1	2-4	5
ADDD F0,F4,F0	2	3-5	6
MULTD F4,F2,F0	3	7-10 (LDE)	11
ADDD F6, F0,F0	6 (E.R)	7-9	10
DIVD F4,F4,F6	12 (E.R)	13-22	23
ADDD F2,F6,F6	13	14-16	17
ADDD F0,F2,F6	14	18-20 (LDE)	21
MULTD F2,F2,F0	24 (E.R)	25-28	29