

**Arquitectura e Ingeniería de Computadores. Examen Parcial (Teoría). 27/01/2005**

**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.

**Puntuación.-** Pregunta con todas las respuestas acertadas: 1 punto. Pregunta con un fallo: 0,6 puntos. Pregunta dos fallos 0,1 puntos. Pregunta con más de dos fallos 0 puntos. La teoría supone la mitad de la nota del examen.

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, un sumador de punto flotante y un divisor con tiempos de cálculo 5, 2 y 20 ciclos de reloj, respectivamente, todos ellos segmentados. Marque cuáles de las siguientes afirmaciones son correctas:

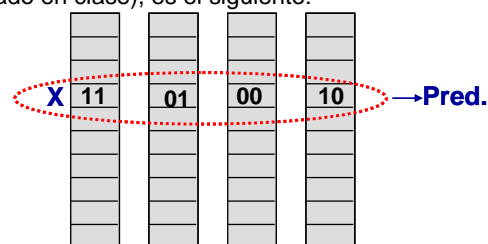
- ☐ a) Un par de instrucciones consecutivas de la forma:
- ```
LD    R3, 0(R5)
ADD   R5, R3, R2
```
- provoca una penalización de un ciclo de reloj.
- ☐ b) Si un salto no se toma, la penalización es 0 ciclos de reloj.
- ☐ c) Si un programa contiene dos instrucciones de la forma:
- ```
MULD  F2, F4, F6
ADDD  F6, F8, F10
```
- se plantea un riesgo que detendrá el ADDD en la etapa de decodificación hasta que la instrucción MULDD haya completado la lectura de registros.
- ☐ d) Las instrucciones pueden ejecutar la fase WB en desorden.
- ☐ e) La detección de las dependencias de tipo EDE en la etapa de decodificación puede hacerse mediante un registro de desplazamiento.

2. Supongamos la arquitectura básica del DLX segmentado en 5 etapas con las siguientes mejoras: existe forwarding, los saltos se resuelven en la segunda etapa y se implementa una política de saltos retardados. Marque cuáles de las siguientes afirmaciones son ciertas:

- ☐ a) Si la condición de salto no se cumple, entonces es incorrecto rellenar el "delay slot" con una instrucción del destino del salto.
- ☐ b) Cuando el "delay slot" se rellena con una instrucción previa al salto, entonces la ejecución del salto implica una penalización de un ciclo de reloj.
- ☐ c) La siguiente transformación es correcta:
- |  |   |  |
|--|---|--|
| <pre>sub r1,r2,r3 beqz r1, et nop add r6,r1,r1 xor r8,r6,r5 ... et: sub r5,r6,r7</pre> | ⇒ | <pre>sub r1,r2,r3 beqz r1, et add r6,r1,r1 xor r8,r6,r5 ... et: sub r5,r6,r7</pre> |
|--|---|--|
- ☐ d) Si el compilador rellena el "delay slot" con una instrucción del destino del salto y el salto se toma, entonces se produce una penalización de un ciclo de reloj.
- ☐ e) La instrucción que ocupa el "delay slot" siempre se ejecutará completamente, tanto si el salto se toma como si no se toma

3. Marque cuáles de las siguientes afirmaciones sobre predicción de saltos son correctas:

- ☐ a) En presencia de bucles anidados un predictor de dos bits se comporta mejor que un predictor de un bit.
- ☐ b) Supongamos un predictor de dos niveles de historia (2,2). Supongamos que el estado del predictor para la instrucción de salto S, que tiene asignada la posición X en la Tabla de Historia de Saltos (con el formato explicado en clase), es el siguiente:



**Tabla de Historia de Saltos**

Si los dos últimos saltos no se han tomado, entonces se predice que salto S se toma.

- ☐ c) En un predictor de dos niveles (n, m), para cada salto existen n predictores de  $2^m$  bits.
- ☐ d) En un predictor de dos bits, el contador saturado de dos bits asociado a cada salto se incrementa cada vez que se produce un acierto en la predicción.
- ☐ e) En general, el comportamiento de los predictores de saltos dinámicos es mejor en los programas de punto flotante que en los de punto fijo.

4. Sea una arquitectura capaz de lanzar a ejecución una instrucción por ciclo, con planificación dinámica y ejecución especulativa basada en uso de un buffer de reordenamiento (ROB). Marque las afirmaciones correctas:

- ☐ a) Las cuatro últimas fases del procesamiento de una instrucción son: lanzamiento (issue), ejecución, escritura del resultado y "commit".
- ☐ b) Una instrucción de salto puede lanzarse a ejecución cuando el ROB está lleno.
- ☐ c) El ROB permite implementar un modelo de interrupciones precisas.
- ☐ d) Cuando una instrucción de suma en punto flotante hace la fase "commit", es posible que el resultado no se almacene en ningún registro del Banco de Registros.
- ☐ e) Cuando una instrucción ejecuta la fase de escritura del resultado, escribe sobre el Bus de Datos Común el resultado acompañado del nº de la Estación de Reserva donde se encuentra la instrucción.

5. Marque cuáles de las siguientes afirmaciones sobre memoria cache son correctas:

- ☐ a) La cache no bloqueante es una técnica para reducir la tasa de fallos.
- ☐ b) En un sistema cache de dos niveles (L1,L2) la tasa de fallos local en L1 es igual a la tasa de fallos global en L1
- ☐ c) El alargamiento de arrays en una técnica para reducir el número de fallos iniciales.
- ☐ d) Al ejecutar un cierto programa en una cache totalmente asociativa se producen 4016 fallos. Con estos datos podemos afirmar que la suma de fallos iniciales y de capacidad ha sido 4016.
- ☐ e) La penalización media por instrucción (en ciclos) como consecuencia de no tener una cache perfecta, viene dada por la expresión:
- $$[(\text{Promedio de accesos a memoria por instruc}) \times (\text{Tasa de fallos}) \times (\text{Ciclos de penalización por fallo})]$$

