

## Arquitectura de los Computadores. Primera convocatoria 2014

### 1. (1,75 puntos)

Supongamos que estamos considerando la transformación de una arquitectura carga/almacenamiento (R-R) en una máquina R-M, con objeto de mejorar el rendimiento reduciendo el recuento de instrucciones. Para ello se está estudiando la incorporación de instrucciones aritméticas con una referencia a memoria ALU(R-M) además de las que operan con registros ALU(R-R). La idea es sustituir las secuencias de instrucciones LOAD; ALU(R-R); por ALU(R-M)\* para reducir el número de instrucciones LOAD. El compilador realizará estas sustituciones en caso de que el operando cargado y utilizado por una instrucción ALU(R-R) no se utilice de nuevo, situación que se produce con una frecuencia del 30% de las instrucciones ALU(R-R) de la CPU carga almacenamiento. Estas nuevas instrucciones de registro memoria emplean dos ciclos de reloj. En la tabla se observan los CPI por tipo de instrucción de las dos CPUs.

CPU A (R-R)		
Operación	Frecuencia	Cuenta de ciclos de reloj
ALU (R-R)	45%	1
Cargas	20%	2
Almacenamientos	10%	2
Salto	25%	2
CPU B (R-M)		
Operación	Frecuencia	Cuenta de ciclos de reloj
ALU (R-R)	¿?	1
ALU (R-M)	¿?	2
Cargas	¿?	2
Almacenamientos	¿?	2
Salto	¿?	2

- ¿Calcula la aceleración en caso de no haber contrapartida, ni en CPI por tipo de instrucción, ni en clk?
  - ¿Calcula la aceleración en caso de encontrar la contrapartida del incremento del clk de la CPU B en un 10%?
- \* **LOAD R1,0(R7)**  
**ALU(R-R) R2, R1**  
 Por  
**ALU(R-M) R2, 0(R7)**

### 2. (1,75 puntos) Para las cuatro siguientes preguntas, supongamos que se está considerando mejorar una máquina añadiéndole una unidad aritmética segmentada (UAS). Cuando se ejecuta un cálculo mediante la UAS, es 20 veces más rápido que en el modo normal de ejecución. Llamamos al porcentaje de tiempo que puede emplearse el modo UAS porcentaje UAS.

- Dibujar un gráfico donde se muestre la aceleración en relación al porcentaje UAS. Rotular el eje y con "aceleración neta" y el eje x con "porcentaje UAS".
- ¿Qué porcentaje UAS se necesita para conseguir una aceleración de 2?
- ¿Qué porcentaje UAS se necesita para conseguir la mitad de la aceleración máxima alcanzable utilizando el modo UAS?
- Supongamos que hemos medido el porcentaje UAS de programas, obteniendo que es del 70%. El grupo de diseño hardware dice que puede duplicar la velocidad de la UAS con una inversión significativa de ingeniería adicional. Se desea saber si el equipo de compilación puede incrementar la utilización del modo UAS como otra aproximación para incrementar el rendimiento. ¿Qué incremento en el porcentaje UAS (relativo a la utilización actual) se necesitará para obtener la misma ganancia de rendimiento?

### 3. (2,5 puntos) El mismo equipo de diseño de la pregunta 1, ha pensado rediseñar su arquitectura del repertorio de instrucciones para contemplar la máquina R-M con una referencia a memoria (CPU B). A partir del análisis realizado, se contempla que sean 48 el número de operaciones a realizar incluyendo las aritméticas, lógicas, saltos y referencias a memoria que permiten abordar las tareas de un procesador de propósito general. Dado que los programas que se ejecuten en el procesador se desarrollarán con lenguajes de alto nivel, se contempla un banco de 32 registros de 32 bits. Por último, el espacio de direcciones de acceso a memoria es de 4GB.

- (1 punto) Indica qué decisiones de manera **justificada** debería tomar el equipo de diseño, valorando las diferentes alternativas, en las siguientes componentes de la arquitectura.
  - Número de operandos de instrucciones ALU
  - Modos de direccionamiento
  - Codificación de los modos de direccionamiento
  - Tipos de instrucciones en el repertorio
  - Formas de especificar el destino del salto y la condición de salto
- (1,5 puntos) Determina los formatos de instrucciones incluyendo los campos de la instrucción y su tamaño según las decisiones tomadas en el apartado anterior

### 4. (1,5 puntos) Suponer que el siguiente fragmento de código se ejecuta en la máquina segmentada de 5 etapas de teoría con un valor inicial para R8 de R2+16:

```

Loop:   LW R5, 0(R2)
        LW R4, 0(R3)
        ADD R5, R5, R4
        ADD R6, R2, R6
        SW R5, 0(R6)
        ADDI R2, R2, 4
        ADDI R3, R3, 4
        SUB R7, R8, R2
        BNZ R7, LOOP
  
```

Nombre: \_\_\_\_\_

- (0.3 puntos) Suponed que el fragmento de código se ejecuta en la máquina sin implementación de la técnica de adelantamiento (forwarding) y que el cálculo de la dirección de salto se realiza en la etapa de ejecución. ¿Cuántos ciclos de reloj tardaría el fragmento de código en ejecutarse?
- (0.4 puntos) Suponed ahora que se adelanta el cálculo de salto a la etapa de decodificación y que el procesador implementa todos los posibles caminos de adelantamiento. Muestra la ejecución del código en el diagrama de temporización siguiente suponiendo que se predican los saltos como no efectivos. ¿Cuántos ciclos de reloj tardaría el fragmento de código en ejecutarse?. (Añade más filas en el diagrama si los necesitas).

[illegible]

- c) (0.4 puntos) Suponed ahora que además de forwarding se utiliza la técnica de saltos retardados (delay slot). Si es posible, **reordena** la secuencia de instrucciones para evitar el máximo número de paradas y que se ejecute el fragmento de código en el mínimo número de ciclos de reloj posible. ¿En cuántos ciclos de reloj se ejecutaría el nuevo código?
- d) (0.4 puntos) Suponed que la etapa de segmentación más larga necesita 0.8ns y que el retardo de los registros de segmentación es de 1ns. Determina el CPI del bucle para los apartados b y c anteriores y calcula el tiempo medio de ejecución por instrucción para cada caso.

5. (1 punto) Considerad un versión simplificada del MIPS en la que los datos inmediatos de las instrucciones LW y SW han de ser cero. Es decir la instrucción lw \$t0, 0(\$t1) es correcta pero no así la instrucción lw \$t0, 4(\$t1). Esta nueva máquina se puede implementar con cuatro etapas de segmentación: IF, ID, EM, WB donde la etapa EM realiza en paralelo las tareas que se hacen en las etapas EX y MEM.

- a) (0.4 puntos) Considerad la secuencia de instrucciones :

```
lw $t0, 0($a0)
```

```
lw $t0, 0($t0)
```

Suponiendo que hay adelantamiento, rellena el diagrama de temporización siguiente dónde se señalen (si hay) las paradas para resolver los riesgos de datos y los adelantamientos.

[illegible]

- b) (0.3 puntos) Si la instrucción `sw` se sustituye por una instrucción de salto condicional, ¿Bajo qué circunstancias sería necesario realizar una parada entre las dos instrucciones?
- c) (0.3 puntos) Si la instrucción `sw` se reemplazara por `addi $t0, $0, 1` y no estuviera implementado la técnica de adelantamiento. ¿Cuántas paradas serían necesarias? Razona la respuesta

6. **(1,5 puntos)** El sistema de memoria del ejercicio 1 cuenta con un sistema ideal de acceso a la misma. Supongamos en este ejercicio una jerarquía de memoria real con una caché de 2MB que tiene una penalización de fallos de la caché de 150 ciclos de reloj. A partir de los análisis del equipo de diseño, la frecuencia de fallos para ambos sistemas es de 5%. La CPU A tiene una media de 1.25 referencias a memoria por instrucción mientras que la CPU B una media de 1.5 referencias por instrucción (al incluir las instrucciones R-M) ¿Cuál es el impacto en el rendimiento de ambos sistemas cuando se incluye el comportamiento de la caché? Calcularlo para el caso en el que ambas máquinas tienen la misma duración del ciclo de reloj (clk).