

— Fundamentos de Computadores. Examen oportunidad ordinaria. 26/05/2017. —

Nome: \_\_\_\_\_ DNI: \_\_\_\_\_

1. El resultado del benchmark SPEC CPU2006 para el juego GO, ejecutándose en un Intel Core i7 920, (\_\_\_\_\_/2,5 pts)  
con una frecuencia de reloj de 2,66 GHz, indica un número de instrucciones de  $1274 \times 10^6$  y un tiempo de ejecución  $T_{CPU} = 527$  s. Supón que estamos desarrollando una nueva versión del procesador con un reloj de 4 GHz. Hemos añadido algunas instrucciones máquina adicionales a la ISA que producen una reducción del 20 % en el número de instrucciones del GO. Sabiendo que el SPECratio en esta nueva versión del procesador es de 24,8, obtén las valores del CPI para el GO en el procesador original y en el modificado. El tiempo de referencia para el SPECratio del GO es de 10490 s. ¿Es el nuevo procesador más rápido que el original a la hora de ejecutar el GO? ¿tiene un mayor o menor CPI? Justifica la respuesta. (NOTA: recuerda que  $SPECratio = T_{referencia} / T_{CPU}$ ) (\_\_\_\_\_/1,5 pts)

2. Indica el valor decimal de los siguientes números en formato IEEE 754 de precisión simple: (\_\_\_\_\_/1,5 pts)

- 0xC0500000
- 0x80200000
- 0x7FA00000

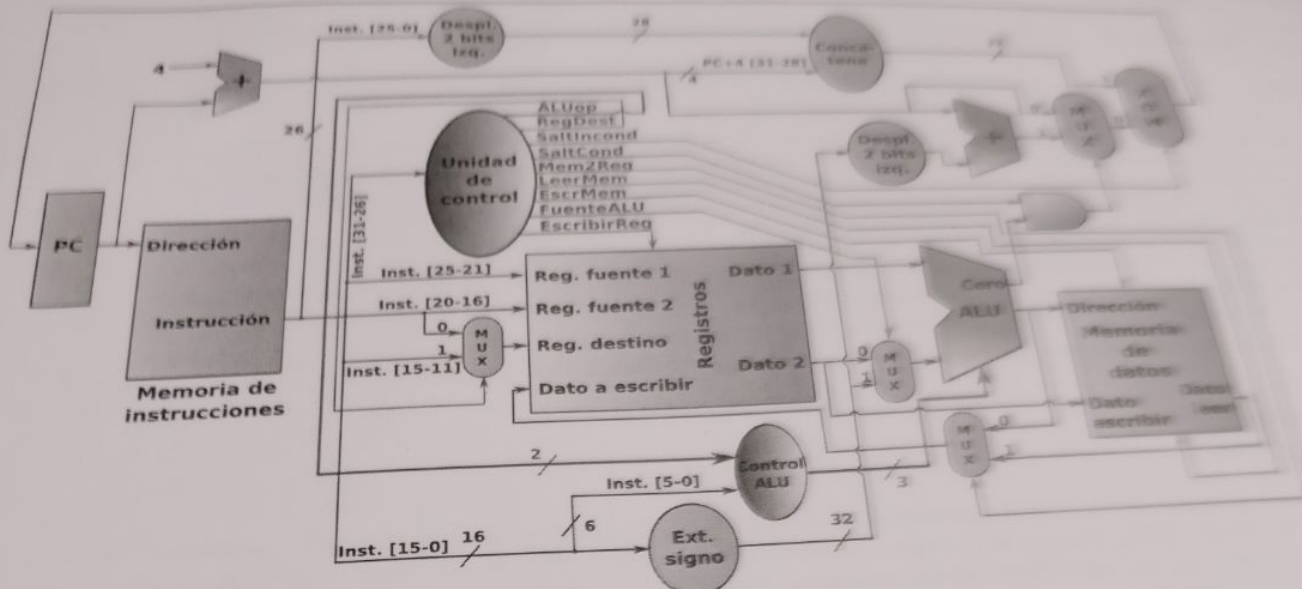
3. Se desea añadir la instrucción `jlw` al procesador tipo MIPS de la figura de la página siguiente (implementación vista en clase). Esta instrucción realiza un salto incondicional, pero la dirección destino de salto se obtiene del contenido de una dirección de memoria. Esta dirección de memoria se obtiene sumando el contenido de un registro y una constante, con el signo extendido, especificada en la propia instrucción. (\_\_\_\_\_/2,5 pts)

`jlw $s1, cte → PC = Mem[R[$s1]+sig.ext.cte]`

Indica, razonando la respuesta:

- a) Tipo y formato de la nueva instrucción.
  - b) Especifica la ruta de datos sobre el dibujo indicando, en caso de ser necesario, qué hardware adicional se requeriría, dónde sería necesario colocarlo y para qué.
  - c) Indica en la segunda tabla de la siguiente página los valores que deben tomar todas las señales de control para que se pueda ejecutar la nueva instrucción, añadiendo nuevas filas, si es necesario para controlar el nuevo hardware<sup>1</sup>. (\_\_\_\_\_/2,5 pts)
4. Supón un sistema BigEndian, con ISA MIPS de 32 bits (palabra de 32 bits), y direccionamiento de memoria a nivel de byte. El sistema tiene una cache de datos de 64 KiB, asociativa por conjuntos de 2 vías y con tamaño de línea de 8 palabras. Considera además que en la posición de memoria M está almacenado el valor  $M \% 256$ , para cualquier dirección M de memoria (es decir, en cada dirección de memoria se encuentra almacenado un byte igual a los dos últimos dígitos hexadecimales de la dirección; por ejemplo, en la dirección 0x1234ABCD se encuentra el byte 0xCD). (\_\_\_\_\_/2,5 pts)
- Si inicialmente la cache está vacía, responde a las siguientes preguntas, justificando tu respuesta.
- a) Supón que el registro `$t1` guarda el valor 0x10100058, y se ejecuta la instrucción `lw $t2, 0($t1)`. Indica qué datos se traen de memoria a la cache y en qué conjunto de esta se guardan. ¿Qué valor se guarda en el campo etiqueta? ¿Qué valor se guarda en `$t2`?
  - b) Si inmediatamente después de esa instrucción, se ejecutan las instrucciones `lw $t3, 4($t1)` y `lw $t4, 8($t1)`, ¿qué pasa en la cache?, ¿qué valores se guardan en los registros `$t3` y `$t4`? (\_\_\_\_\_/1,0 pts)
5. Razona la verdad o falsedad de las siguientes afirmaciones
- a) El algoritmo de Booth de multiplicación de enteros es más eficiente que el algoritmo simple (sumas y desplazamientos) cuando en la representación binaria del multiplicador hay largas secuencias de 1's.
  - b) Un carácter Unicode almacenado en UTF-8 ocupa siempre 8 bytes.
  - c) Una implementación de una cache asociativa por conjuntos necesita tantos comparadores como conjuntos.
  - d) Si a un sistema con una cache de primer nivel o L1 se le añade otra de segundo nivel o L2, la tasa de fallos de la cache L1 se reduce.

<sup>1</sup> Indica con una X aquellos casos en los que el valor de la señal de control sea indiferente para esta instrucción. Usa los datos de la primera tabla para elegir el valor apropiado de ALUop.



| ALUop | func   | Acción de la ALU | Control de la ALU |
|-------|--------|------------------|-------------------|
| 00    | XXXXXX | suma             | 010               |
| 01    | XXXXXX | resta            | 110               |
| 10    | 100000 | suma             | 010               |
| 10    | 100010 | resta            | 110               |
| 10    | 100100 | and              | 000               |
| 10    | 100101 | or               | 001               |

| Señal de control | Valor |
|------------------|-------|
| ALUop            |       |
| RegDest          |       |
| SaltIncond       |       |
| SaltCond         |       |
| Mem2Reg          |       |
| LeerMem          |       |
| EscrMem          |       |
| FuenteALU        |       |
| EscribirReg      |       |