

Primer Examen Parcial

Arquitectura de Computadoras

20/05/2011

Instrucciones

- En el presente parcial, que consta de 4 ejercicios, podrá emplear todo material de consulta que considere necesario.
- En cada ejercicio se indica el tiempo estimado para resolverlo y el puntaje que otorga el mismo.
- La duración total del parcial es de dos horas.
- Las hojas que no estén caratuladas en la forma indicada por la cátedra no serán tenidas en cuenta.
- No se permitirá intercambiar ni prestar ningún tipo de elemento entre los estudiantes.
- La buena presentación, ortografía y redacción serán tenidas en cuenta para la calificación final.

1) **CISC Vs RISC. (30%, 40')** En el presente ejercicio se va a analizar las ventajas y desventajas de estos dos tipos de máquinas desde el punto de vista del ISA. Como máquina RISC se tomará el MIPS visto en clase.

La siguiente instrucción CISC opera únicamente con memoria, mediante referencia directa:

ADD (20H), (21H), (22H) ;

suma el contenido de memoria en la posición 21H con el contenido de la posición 22H y guarda el resultado en la posición 20H.

- a) Escriba un segmento de programa en la MIPS que haga lo mismo que la instrucción previamente descrita.
- b) Suponiendo que el espacio de memoria total es de 4GB, y suponiendo que la dimensión de los OP-Codes fuera de 8 bits en la máquina CISC. ¿Cuál segmento de programa es más grande? ¿Por cuánto?
- c) Suponiendo que la máquina CISC tiene instrucciones de largo fijo, y debe poder direccionar, en todos sus operandos, tanto a GPRs como así también a Memoria. Se limita el direccionamiento directo únicamente a página cero, con 16 bits. El desplazamiento debe ser de 16 bits y se definen sólo 4 registros índice. También se define que las constantes no superarán los 16 bits para direccionamiento inmediato. No considere más que estos modos de direccionamiento. El formato de todas las instrucciones es único. ¿Cuántos bits deben dedicarse a especificar cada operando? (sugerencia, considere un formato ortogonal).
- d) Si la máquina CISC tuviera un máximo de 27 operaciones de 3 operandos y cualquier operando podría tener uno de los siguientes modos de direccionamiento: registro GPR (32 registros), indexado (constante 16 bits y 4 índices posibles), inmediato (con constantes de 16 bits), registro indirecto autoincrementado pre (el GPR se incrementa en 1 y luego se lo usa como puntero a memoria para traer el operando). Se pide:
 - i) Dimensionar el tamaño máximo del formato de la instrucción y especificar cada uno de sus campos.
 - ii) Ventajas y desventajas de tener un largo fijo para una CISC.

2) **Pipeline. (40%, 45')** Considere la máquina CISC del punto 1d, con largo fijo de instrucción. Por simplicidad, tanto el banco de registros, como la Memoria pueden ser accedidos simultáneamente por la cantidad de direcciones que considere convenientes. En ambos casos es posible leer y escribir al mismo tiempo.

- a) Se pide que dibuje un esquema de pipeline simple para realizar el camino de datos de esta máquina CISC. (Sugerencia: inspírese en el pipe del MIPS). Confeccione una tabla en la que se detalle cada etapa y la función de la misma. Numere las etapas a partir del 1 comenzando desde la izquierda. Este pipeline no tendrá anticipaciones.
- b) Para implementar la instrucción de salto relativo BEQ, similar a la del MIPS, la cual puede tener hasta 2 operandos en memoria.
 - i) ¿En qué etapa se calcula la dirección del salto?
 - ii) ¿En qué etapa se evalúa la condición?
- c) Si en total hay un 20% de instrucciones de salto condicionales, y suponiendo que el resto de las instrucciones no tienen riesgos. ¿Cuál es el CPI promedio en esta máquina?

- d) Si el compilador logra llenar exitosamente el 20% de los huecos, ¿qué mejora de performance se obtendría?
- e) Asegúrese de que su máquina no tiene riesgos estructurales. Indique para ello qué Hw colocará en cada etapa y cuántos puertos en Memoria y Banco de registros.
- f) Indique qué riesgos de datos se presentan y la cantidad máxima de burbujas que habría que introducir si no se resuelve los mismos:

- i) Operaciones entre registros.
- ii) Operaciones entre memoria.
- iii) ¿Operaciones entre registros y Memoria?

g) Considere ahora el modo autoincrementado pre. Dibuje el esquema del pipeline modificado para aceptar este modo.

- i) ¿Se puede generar un riesgo WAR? Explique su respuesta, si es positiva escriba una secuencia de instrucciones que muestre el riesgo, si es negativa demuéstrela.

3) **Pipeline. (10%, 15')** Suponga que un procesador en pipeline tiene 6 etapas, cuyas latencias son las siguientes: 2 ns, 3 ns, 4 ns., 7 ns., 3 ns. y 2 ns. Las etapas se interconectan por medio de latches cuya latencia es de 1 ns c/u.

- a) ¿Cuál es la frecuencia máxima de reloj que se puede conectar a este procesador?
- b) ¿Cuál es la latencia del procesador?
- c) Suponiendo que fuera posible agrupar en una única etapa a dos o más etapas adyacentes del procesador en pipeline, sin cambiar la frecuencia de reloj calculada en el punto anterior. ¿Con cuántas etapas quedaría este pipe?
- d) ¿Vale la pena reducir el número de etapas del procesador, como se hizo en el punto anterior? Fundamente claramente su respuesta.

4) **ILP: Dependencias y Renombramiento de Registros, Ejecución Fuera de Orden. (20%, 20').** Dado el siguiente código, en el cual todas las instrucciones duran 1T y no requiere paradas después de un LOAD:

```

1)      lw      r1, (r2)
2)      add     r3, r4, r1
3)      sub     r4, r5, r6
4)      mul     r7, r4, r8
5)      sllv    r8, r9, r10
6)      sub     r11, r8, r12
7)      div     r12, r13, r14
8)      sw      r12, (r15)
    
```

- a) Indique cuáles son las dependencias WAR y WAW, y en qué registro(s), suponiendo que todas las instrucciones podrían ejecutarse al mismo tiempo.
- b) Reescriba el código renombrando los registros para evitar estas dependencias.
- c) ¿En cuántos T se podría ejecutar este código fuera de orden y con todo el paralelismo que fuera necesario? ¿Cuánto paralelismo sería necesario?

¡Éxito!

E. D. Cohen

Soluciones propuestas

1)

a)

```
lw    $t0, 0x21($zero)
lw    $t1, 0x22($zero)
add   $t2, $t0, $t1
sw    $t2, 0x20($zero)
```

b) CISC ocupa 13 Bytes, y MIPS 16 Bytes, por lo que MIPS es un 23% más grande.

c) Son necesarios 20 bits para especificar cada operando.

d) Son necesarios 65 bits para especificar el formato, pero por conveniencia se lo lleva a 72 bits. Es mejor para pipelining, pero desperdicia mucho espacio.

2)

a) Tiene 6 etapas, a saber: Instruction Fetch; Instruction Decode; Address Calculation, donde se calculan tres direcciones; Data Memory Access; Execute; y Write Back, en registros y en memoria

b) Es posible realizar el cálculo de la dirección en la etapa 2, y es conveniente evaluar la condición de salto a fines de la etapa 4.

c) $CPI = 1.6$ d) $CPI = 1.48$, un 8.1% mejor que la versión anterior.

e) La memoria necesita dos puertos de lectura y uno de escritura, y el banco de registros necesita tres puertos de lectura y uno de escritura.

f) En operaciones entre registros, serán necesarias 3 burbujas en el peor caso, y en operaciones entre memoria, una sola burbuja.

g) No es necesario modificar la estructura del pipeline, pero sí ampliar la etapa 3. Es imposible la ocurrencia de riesgos WAR si se cumplen las tres premisas de diseño.

3)

a) Frecuencia = 125 MHz.

b) Latencia = 48 ns.

c) Cuatro etapas.

d) Si es conveniente, para reducir la cantidad y el impacto de los riesgos.

4)

a) Hay tres dependencias WAR, y ninguna WAW.

b)

```
1)    lw    r1, (r2)
2)    add   r3, r4, r1
3)    sub   r16, r5, r6
4)    mul   r7, r16, r8
5)    sllv  r17, r9, r10
6)    sub   r11, r17, r12
7)    div   r18, r13, r14
8)    sw    r18, (r15)
```

c) Con 4 unidades en paralelo, se puede ejecutar el código en 2 T.