Nombre:_____

Arquitectura de los Computadores. Junio 2015

4. (3 puntos) Preguntas sobre segmentación:

a) **(0,4 puntos)** Considerad un procesador no segmentado con una ruta de datos de 5 etapas de ejecución que funciona a 2GHz, en el que las operaciones ALU y salto requieren cuatro ciclos de reloj y las de memoria cinco. Suponer que las frecuencias relativas para estas operaciones son 45%, 25% y 30% respectivamente. Se quiere segmentar la máquina con 5 etapas y debido al sesgo del reloj y a los registros de segmentación, segmentar el procesador alarga el periodo de reloj en un 10%. El procesador utiliza una cache unificada para datos e instrucciones con un único puerto para el acceso a la memoria lo que provoca un riesgo estructural entre las etapas IF y MEM y consiguiente parada de 1 ciclo de reloj. Suponed que las referencias a datos representan el 30% de les instrucciones ejecutadas y que el CPI ideal del procesador segmentado ignorando el riesgo estructural es 1. ¿Cuál es la ganancia que se puede conseguir con la segmentación?

Solución:

- Sin segmentación:

Obtenemos el periodo de reloj: $PeriodoReloj = \frac{1}{2GHz} = 0{,}5ns$

Calculamos el CPI medio:

$$CPI = \frac{\sum_{i=1}^{n}(CPI_i \cdot I_I)}{RI} = (0{,}45 + 0{,}25) \times 4 + 0{,}30 \times 5 = 4{,}3$$

- Con segmentación:

El periodo de reloj se alarga en un 10% , calculamos el nuevo periodo de reloj:

$$PeridoReloj = 0.5ns + (10\% \times 0.5) = 0.55ns$$

El 30% de las instrucciones accede a memoria durante la etapa MEM. Cada una de estas instrucciones resulta en un riesgo estructural con una parada de 1 ciclo. El nuevo CPI se puede calcular como:

$$CPI = CPI_{ideal} + Ciclos\ de\ reloj\ detención\ por\ instrucción = 1 + 0.3 \times 1 = 1.3$$

Calculamos ahora la ganancia que se puede conseguir con la segmentación:

$$Ganancia = \frac{CPI_{\text{sin }segmentación} \times CLK_{\text{sin }segmentación}}{CPI_{\text{con }segmentación} \times CLK_{\text{con }segmentación}} = \frac{0.5ns \times 4.3}{0.55 \times 1.3} = 3$$

El procesador segmentado es tres veces más rápido que el no segmentado.

b) **(0,4 puntos)** Suponer ahora que además se consideran las detenciones por dependencia de datos de 1 ciclo de reloj y que representan el 20% de las instrucciones ejecutadas y las paradas de control de 2 ciclos suponen el 5% de las instrucciones ejecutadas, ¿Cuál es ahora el nuevo CPI?. ¿En qué porcentaje se reduce la ganancia al considerar esta nueva situación?

Solución:

El 20% de las instrucciones ejecutadas tienen además riesgos por dependencia de datos y el 5% riesgos de control con paradas de 2 ciclos de reloj, el nuevo CPI se puede calcular como:

$$CPI = CPI_{ideal} + Ciclos\ de\ reloj\ detención\ por\ instrucción = 1 + 0.3 \times 1 + 0.2 \times 1 + 0.05 \times 2 = 1.6$$

Y la nueva ganancia como:

$$Ganancia = \frac{CPI_{\text{sin }segmentación} \times CLK_{\text{sin }segmentación}}{CPI_{\text{con }segmentación} \times CLK_{\text{con }segmentación}} = \frac{0.5ns \times 4.3}{0.55 \times 1.6} = 2.44$$

La ganancia ha bajado de 3 a 2.44:

$$\left.\begin{matrix} 3 - 100 \\ 2.44 - x \end{matrix}\right\} \quad Porcentaje\ Ganacia\ Nueva = \frac{2.44 \times 100}{3} = 81.33\%$$

Nombre:_____

Con esta nueva situación la ganancia ha pasado a ser del 81.33% de la original.

Ha habido una reducción del 18,67% $(Reducción = 100 - 81.33 = 18{,}67)$

Suponer que el siguiente código MIPS se ejecuta en la máquina segmentada anterior:

```
addi $3, $0, 100
add $4, $0, $0
Loop:   lw $5, 0($1)
        add $4, $4, $5
        lw $6, 0($2)
        sub $4, $4, $6
        addi $1, $1, 4
        addi $2, $2, 4
        addi $3, $3, -1
        bne $3, $0, Loop
```

c) **(1 punto)** Muestra el diagrama de temporización de una iteración del bucle suponiendo que no hay forwarding. Completa la tabla de temporización mostrando todas las todos los ciclos de parada. Suponed que los saltos paran la segmentación durante un ciclo de reloj.

Instrucción	Ciclos de reloj																								
	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
addi $3, $0, 100	IF	ID	EX	M	WB																					
add $4, $0, $0		IF	ID	EX	M	WB																				
lw $5, 0($1)			IF	ID	EX	M	WB																			
add $4, $4, $5				IF	--	--	ID	EX	M	WB																
lw $6, 0($2)							IF	ID	EX	M	WB															
sub $4, $4, $6								IF	--	--	ID	EX	M	WB												
addi $1, $1, 4											IF	ID	EX	M	WB											
addi $2, $2, 4												IF	ID	EX	M	WB										
addi $3, $3, -1													IF	ID	EX	M	WB									
bne $3, $0, Loop														IF	--	--	ID									
lw $5, 0($1)																	IF	IF	ID	EX	M	WB				
add $4, $4, $5																			IF	--	--	ID	EX	M	WB	

d) **(0,4 puntos)** De acuerdo con el diagrama de temporización del apartado anterior, calcula el número de ciclos de reloj y el CPI medio para ejecutar todas las iteraciones del bucle anterior.

Solución:

Hay 100 iteraciones del bucle (el registro $3 inicialmente tiene el valor $3=100).
Cada iteración requiere 15 ciclos como se observa en el cronograma anterior, (8 ciclos para empezar las 8 instrucciones dentro del cuerpo del bucle + 7 paradas).
Además hay 2 ciclos adicionales para comenzar las 2 primeras instrucciones antes del bucle.
Por tanto los ciclos totales necesarios para ejecutar el código serán:
$Ciclos\ Totales = 100 \times 15 + 2 = 1502\ ciclos$ (Aproximadamente 1500 ciclos)

Para el cálculo del número total de instrucciones ejecutadas hay que considerar que dentro del bucle se ejecutan 8 instrucciones, por tanto:
$Intrucciones\ Ejecutadas = 2 + 8 \times 100 = 802$

El CPI medio para ejecutar el código anterior será:

$$CPI_{medio} = \frac{Ciclos\ Totales}{Instrucciones\ Ejecutadas} = \frac{1502}{802} = 1.87$$

Nombre:_____

e) **(0,4 puntos)** Reordena las instrucciones del anterior bucle y rellena el delay slot del salto para que evitar el máximo posible de las paradas y escribe el código resultante.

Solución:

Una posible solución sería rellenar el delay slot con la instrucción sub $4, $4, $6 y mover las instrucciones lw $6, 0($2) y addi $3, $3, -1 para evitar dependencias de datos:

```
addi $3, $0, 100
addi $4, $0, $0
Loop:
        lw $5, 0($1)#
        lw $6, 0($2)        # Instrucción movida para evitar paradas load
        addi $3, $3, -1     # Instrucción movida para evitar dependencia de datos con bne
        add $4, $4, $5
        addi $1, $1, 4
        addi $2, $2, 4
        bne $3, $0, Loop
        sub $4, $4, $6      # Instrucción que rellena el delay slot
```

Hay que tener en cuenta que serían posibles otras soluciones.

f) **(0,4 puntos)** Calcula el número de ciclos y CPI medio para ejecutar todas las iteraciones del bucle con el nuevo código. ¿Cuál es la ganancia obtenida?

Solución

Hay 100 iteraciones y cada iteración requiere 8 ciclos ya que no hay paradas.
Hay dos ciclos adicionales por las primeras 2 instrucciones antes del bucle y hay que añadir 4 ciclos adicionales para terminar la instrucción *addi* en la última iteración.

Por tanto:

$$Ciclos\ Totales = 100 \times 8 + 6 = 806\ ciclos$$

El número total de instrucciones ejecutadas no cambia:

$$Intrucciones\ Ejecutadas = 2 + 8 \times 100 = 802$$

El CPI medio para ejecutar todas las iteraciones del bucle en este caso será:

$$CPI_{medio} = \frac{Ciclos\ Totales}{Instrucciones\ Ejecutadas} = \frac{806}{802} = 1{,}01$$

La ganancia obtenida al reorganizar el código y rellenar el delay slot es:

$$Ganancia = \frac{CPI_{apartado\ d)}}{CPI_{apartado\ f)}} = \frac{1.87}{1.01} = 1.85$$