

Arquitectura de los Computadores. Primera convocatoria 2017

4. (3 puntos) Considera que el segmento de código que aparece a continuación se ejecuta sobre una máquina segmentada MIPS estándar de 5 etapas. Supongamos que el valor inicial del registro R23 es mucho mayor que el valor inicial del registro R20 y que todas las referencias de memoria se encuentran en las cachés y TLBs.

```

LOOP: lw R10, 0(R20)
      lw R11, 100(R20)
      subu R10, R10, R11
      addiu R20, R20, 4
      sw R10, 200(R20),
      subu R5, R23, R20
      bnez R5, LOOP
    
```

- a) (0,5 puntos) Dibuja el diagrama de temporización para la ejecución de la secuencia de instrucciones que se muestran en la tabla. Considera que el salto se resuelve en la etapa ID y que además la técnica del adelantamiento o forwarding no está implementada en esta máquina.

Solución:

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
Lw R10 0(R20)	IF	ID	EX	M	WB																			
Lw R11, 0(R20)		IF	ID	EX	M	WB																		
subu R10, R10, R11			IF	ID	P	ID	EX	M	WB															
addiu R20, R20, #4						IF	ID	X	M	WB														
sw R10, 200(R20)							IF	ID	P	ID	EX	M	WB											
subu R5, R23, R20										IF	ID	EX	M	WB										
bnez R5, LOOP											IF	P	P	ID	EX	M	WB							
lw R10, 0(R20)														P	IF	ID	EX	M	WB					

- b) (0,4 puntos) Obtén el número total de ciclos necesarios para completar 2 iteraciones del bucle y calcula el CPI medio obtenido en completar esta ejecución. Muestra cómo has obtenido los valores.

Solución:

Desde la primera búsqueda del primer lw hasta la segunda búsqueda del primer lw han pasado 14 ciclos. Como el bucle itera dos veces serán 28 ciclos. A estos hay que sumar unos pocos ciclos extra para vaciar el pipeline, en concreto 3. En total, el número de ciclos necesarios para completar 2 iteraciones del bucle es de 31 ciclos.

El número de instrucciones ejecutadas en las dos iteraciones son $2 \times 7 = 14$ instrucciones.

Por tanto podemos calcular el CPI medio en completar la ejecución de 2 iteraciones del bucle como:

$$CPI_{medio} = \frac{\text{Número de ciclos}}{\text{Número de instrucciones}} = \frac{31}{14} = 2,21$$

- c) **(0,5 puntos)** Supongamos ahora que se implementa la técnica de *forwarding* en la máquina anterior. Con esta nueva condición rellenar el diagrama de temporización siguiente:

Solución:

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
Lw R10 0(R20)	IF	ID	EX	M	WB																			
Lw R11, 0(R20)		IF	ID	EX	M	WB																		
subu R10, R10, R11			IF	ID	ID	EX	M	WB																
addiu R20, R20, #4					IF	ID	X	M	WB															
sw R10, 200(R20)						IF	ID	EX	M	WB														
subu R5, R23, R20							IF	ID	EX	M	WB													
bnez R5, LOOP								IF	P	ID	EX	M	WB											
lw R10, 0(R20)										P	IF	ID	EX	M	WB									

- d) **(0,4 puntos)** Obtén el número total de ciclos necesarios para completar 2 iteraciones del bucle. ¿Cuál es ahora el CPI medio obtenido? ¿Cuál es la ganancia obtenida con esta alternativa?

Solución:

Desde la primera búsqueda del primer lw hasta la segunda búsqueda del primer lw han pasado 10 ciclos. Como el bucle itera dos veces serán 20 ciclos. A estos hay que sumar unos pocos ciclos extra para vaciar el pipeline, en concreto 3. En total, el número de ciclos necesarios para completar 2 iteraciones del bucle es de 23 ciclos.

El número de instrucciones ejecutadas en las dos iteraciones no ha cambiado, $2 \times 7 = 14$ instrucciones. Por tanto podemos calcular el CPI medio en completar la ejecución como:

$$CPI_{medio} = \frac{\text{Número de ciclos}}{\text{Número de instrucciones}} = \frac{23}{14} = 1,64$$

La ganancia obtenida con esta implementación se puede calcular como:

$$Ganancia = \frac{RI_a}{RI_c} = \frac{CPI_a}{CPI_c} = \frac{31}{23} = \frac{2,21}{1,64} = 1,35$$

Es decir, al aplicar la técnica de forwarding el pipeline es un 35% más rápido.

- e) **(0,4 puntos)** Supongamos ahora que el procesador implementa saltos retardados (*delay slot*). ¿Podrías reorganizar el código para evitar cualquier parada, aprovechar el delay slot y conseguir mayor ganancia? Razona la respuesta. Muestra en cualquier caso como rellenarías el delay slot.

Solución:

Efectivamente se puede mejorar la ejecución del programa aprovechando el *delay slot* y rellenándolo con una instrucción que se ejecute siempre dentro del bucle y no tenga ningún problema de dependencia con la instrucción *bnez*. Además se puede reorganizar el código para evitar la dependencia del registro R5 con la instrucción *bnez*. Los cambios se muestran a continuación:

Numeramos las instrucciones del código original:

```

LOOP:  lw R10, 0(R20)      ;1
        lw R11, 100(R20)   ;2
        subu R10, R10, R11 ;3
        addiu R20, R20, 4   ;4
        sw R10, 200(R20),   ;5
        subu R5, R23, R20    ;6
        bnez R5, LOOP       ;7

```

Código reorganizado relleno el delay slot:

```
LOOP: lw R10, 0(R20)      ;1
      lw R11, 100(R20)    ;2
      addiu R20, R20, 4    ;4
      subu R10, R10, R11   ;3
      subu R5, R23, R20    ;6
      bnez R5, LOOP       ;7
      sw R10, 200(R20),   ;5
```

Código reorganizado relleno el delay slot y evitando cualquier parada:

```
LOOP: lw R10, 0(R20)      ;1
      lw R11, 100(R20)    ;2
      addiu R20, R20, 4    ;4
      subu R5, R23, R20    ;6
      subu R10, R10, R11   ;3
      bnez R5, LOOP       ;7
      sw R10, 200(R20),   ;5
```

- f) **(0,3 puntos)** ¿Cuál sería en este caso el número de ciclos que tardaría en ejecutarse 2 iteraciones del bucle? ¿Cuál sería la ganancia conseguida para este código respecto a la versión original si la hubiere?

Solución:

Como no hay ninguna parada, desde la primera búsqueda del primer lw hasta la segunda búsqueda del primer lw han pasado 7 ciclos. Como el bucle itera dos veces serán 14 ciclos. A estos hay que sumar unos pocos ciclos extra para vaciar el pipeline, en concreto 3. En total, el número de ciclos necesarios para completar 2 iteraciones del bucle es de 17 ciclos.

El número de instrucciones ejecutadas en las dos iteraciones no ha cambiado, $2 \times 7 = 14$ instrucciones. Por tanto, podemos calcular el CPI medio en completar la ejecución como:

$$CPI_{medio} = \frac{\text{Número de ciclos}}{\text{Número de instrucciones}} = \frac{17}{14} = 1,21$$

La ganancia obtenida reorganizando el código y relleno el delay slot respecto a la versión original será:

$$\text{Ganancia} = \frac{RI_a)}{RI_e)} = \frac{CPI_a)}{CPI_e)} = \frac{31}{17} = \frac{2,21}{1,21} = 1,82$$

Es decir, el código ahora se ejecuta un 82% más rápido que la versión original del apartado a).

- g) **(0,5 puntos)** Se está estudiando dos posibilidades para mejorar la segmentación en los saltos en la máquina anterior en la que el salto se resuelve en la etapa ID. La primera posibilidad es utilizar saltos retardados, en este caso en una ejecución típica del 25% de las instrucciones de salto se consigue rellenar el 50% de los delay slot con instrucciones útiles. **Calcula** la ganancia que se consigue al implementar esta técnica. La segunda posibilidad es suponer que los saltos son no efectivos. En este caso en una ejecución típica el 25% de los saltos en un bucle son no efectivos y el 80% de las instrucciones son saltos dentro de un bucle. **Calcula** la ganancia que se consigue al implementar esta técnica. Por simplicidad suponer que las instrucciones que no son de salto su CPI es 1. ¿Qué optimización es la mejor opción?

Solución:

Para calcular la mejora vamos a comparar los CPI del cauce en cada opción que se podrán calcular como:

$$CPI_{segmentación} = CPI_{ideal} + \text{ciclos reloj detención por instrucción}$$

- Mejora introducida con la inclusión de saltos retardos.

Calculemos primero cual es el CPI original, es decir, sin ninguna mejora introducida. En este caso para cada instrucción de salto se debe incluir una parada de un ciclo de reloj, como el 25% de las instrucciones son saltos el CPI se puede calcular como:

$$CPI_{sin\ delay\ slot} = CPI_{ideal} + \text{ciclos reloj detención por instrucción} = 1 + 0,25 \times 1 = 1,25$$

Calculemos ahora cual es el CPI con la mejora introducida al incluir los delay slot. En este caso el CPI será:

$$CPI_{con\ delay\ slot} = CPI_{ideal} + \text{ciclos reloj detención por instrucción} = 1 + 0,5 \times 0,25 \times 1 = 1,125$$

La ganancia conseguida al implementar la técnica de los saltos retardados es:

Nombre: _____

$$Ganancia = \frac{CPI_{sin\ delay\ slot}}{CPI_{con\ delay\ slot}} = \frac{1,25}{1,125} = 1,11$$

- Mejora introducida al suponer que los saltos son no efectivos.

Calculemos primero cual es el CPI original, es decir, sin ninguna mejora introducida. En este caso de nuevo para cada instrucción de salto el CPI se puede calcular como:

$$CPI_{sin\ mejora} = CPI_{ideal} + ciclos\ reloj\ detención\ por\ instrucción = 1 + 0,8 \times 1 = 1,8$$

Calculemos ahora el CPI con la mejora introducida al suponer que los saltos son no efectivos:

$$CPI_{saltos\ no\ tomados} = CPI_{ideal} + ciclos\ reloj\ detención\ por\ instrucción = 1 + 0,8 \times 0,75 \times 1 = 1,6$$

La ganancia conseguida al implementar la técnica de suponer saltos no efectivos es:

$$Ganancia = \frac{CPI_{sin\ mejora}}{CPI_{saltos\ no\ tomados}} = \frac{1,8}{1,6} = 1,125$$

La mejor opción será la que obtenga mayor ganancia, en este caso se consigue con la introducción de la mejora al suponer que los saltos son no efectivos.