

Arquitectura del Procesador II

Trabajo Práctico IV

Tratamiento de Saltos

Ejercicio 1: Para el siguiente fragmento de código:

```

                addi    $1, $0, 10
loop:          sw     $1, 0($2)
                addi    $2, $2, #4
                addi    $1, $1, -1
                bne    $1, $0, loop
                addi    $1, $0, 10


```

Muestre el diagrama de ciclos de reloj de esta secuencia de instrucciones para el pipeline del MIPS, donde la dirección efectiva del salto se resuelve en la etapa ID, suponiendo:

1. Que se utiliza la política: congelar/vaciar el pipeline.
2. Que se utiliza la política: Salto no tomado.
3. Que se utiliza la política: salto demorado (planifique el *slot* de forma adecuada).

Ejercicio 2: Supongamos la arquitectura básica del MIPS con pipeline de 5 etapas, sobre la cual se implementa una política de saltos demorado ([delayed branch](#)). Los saltos se resuelven en la etapa ID. Indique si las siguientes afirmaciones son ciertas o falsas:

a) La siguiente transformación es legal:

<pre> sub r1,r2,r3 beqz r1, Lab nop add r4,r1,r1 ... Lab: xor r4,r2,r2 ld r1,0(r4) </pre>		<pre> sub r1,r2,r3 beqz r1, Lab1 xor r4,r2,r2 add r4,r1,r1 ... Lab: xor r4,r2,r2 Lab1: ld r1,0(r4) </pre>
---	---	---

b) La siguiente transformación es legal :

<pre> add r6,r2,r2 sub r1,r2,r6 beqz r1, Lab nop add r5,r4,r6 ... </pre>	\Rightarrow	<pre> ... sub r1,r2,r6 beqz r1, Lab add r6,r2,r2 add r5,r4,r6 ... </pre>
--	---------------	--

c) Cuando el **delay slot** se rellena con una instrucción del destino del salto y el salto no se toma, entonces se produce una penalización de un ciclo de reloj.

d) La siguiente transformación es legal :

<pre> sub r1,r2,r3 beqz r1, Lab nop add r6,r1,r1 xor r8,r6,r5 ... Lab: sub r5,r6,r7 </pre>	\Rightarrow	<pre> sub r1,r2,r3 beqz r1, Lab add r6,r1,r1 xor r8,r6,r5 Lab: sub r5,r6,r7 </pre>
--	---------------	--

e) Si se cambia a una política de predicción de saltos como no tomado (predict-not-taken), entonces la instrucción que está a continuación del salto se lee igualmente de la memoria de instrucciones, pero su ejecución se cancela en caso de que el salto no se tome.