

Alumno: \_\_\_\_\_

Alumno: \_\_\_\_\_

Grupo: \_\_\_\_\_

## Sesión XI

# Procesador segmentado. Dependencias y riesgos.

El objetivo de esta sesión es estudiar las dependencias y comprender los riesgos que estas pueden generar en su ejecución en un procesador segmentado.

1. Carga y ejecuta (usando la configuración segmentada con latencia 2 en la unidad de suma) el siguiente código:

```
# programa segmentado.s
#
# segmento de texto
.text
.globl main

main:

la $s0,array
la $t0,count
lw $s4,0($t0)

loop:
    lwc1 $f0,0($s0)
    add.s $f4,$f4,$f0
    swc1 $f4,0($s0)
    addi $s0,$s0,4
    addi $s4,$s4,-1
    bne $s4,$0,loop
    nop

addi $v0,$0,10
syscall

.data:
array: .float 1,1,1,1
count: .word 4
# fin
```

Contesta a las siguientes cuestiones:

- Rellena la siguiente tabla con los datos para la ejecución de este código en las tres configuraciones (suponer que la frecuencia de reloj es 2.0 GHz para el procesador segmentado y multiciclo y 400 MHz para el monociclo):

Configuración	ciclos totales	CPI	Tiempo de ejecución	Rendimiento
monociclo				
Multiciclo				
Segmentado				

- ¿Cuál es el más rápido? ¿Cuánto?
- Dibuja el diagrama multiciclo que ves en el simulador en el ciclo 10. ¿Por qué la instrucción `add.s` se queda bloqueada en la etapa ID durante un ciclo?
- Dibuja el diagrama multiciclo que ves en el simulador en el ciclo 16. ¿Por qué la instrucción `beq` se queda bloqueada en la etapa ID?
- Identifica los riesgos existentes en el lazo loop y enuméralos indicando de que tipo son. Indica además cuantas burbujas se producen en cada caso.
- Intenta solucionar los riesgos reordenando las instrucciones del código. Indica como sería el código resultante y cual es la ganancia que obtienes (en rendimiento).