

ALCOBATON (90)

GF

Arquitectura de Computadoras  
Apellido: \_\_\_\_\_

Parcial 2 - Fecha 1 - 15/11/2024  
Nombre: \_\_\_\_\_

TEMA B  
DNI: \_\_\_\_\_

1. Indicar cuantos atascos se producen del tipo RAW y BTS ejecutando el programa. BTB y Delay Slot están deshabilitados (40 pts)

<b>.data</b>		<b>loop:</b>	<b>ld</b>	<b>\$t2, VEC1(\$t0)</b>
<b>VEC1:</b>	<b>.word 5, 10</b>		<b>daddi</b>	<b>\$t1, \$t1, -1</b>
<b>VEC2:</b>	<b>.word 0</b>		<b>sd</b>	<b>\$t2, VEC2(\$t0)</b>
<b>.code</b>			<b>daddi</b>	<b>\$t0, \$t0, 8</b>
	<b>daddi \$t0, \$0, 0</b>		<b>bnez</b>	<b>\$t1, loop</b>
	<b>daddi \$t1, \$0, 2</b>		<b>halt</b>	

Ejercicio	Puntos
1	40
2	15
3	5
4	30
Total	90

Con Forwarding deshabilitado: RAW: 3 (5 pts) BTS: 1 (5 pts) CPI: 21/13 (10 pts)  
Con Forwarding habilitado: RAW: 0 (5 pts) BTS: 1 (5 pts) CPI: 18/13 (10 pts)

2. El siguiente programa recorre un arreglo de números (TABLA1) y genera un segundo arreglo (TABLA2) con los números que sean impares. Además guarda en RES la cantidad de elementos guardados en el segundo arreglo. Completar instrucciones faltantes. (5 pts c/respuesta)

<b>.data</b>		<b>LOOP:</b>	<b>ld \$t4, TABLA1(\$t2)</b>
<b>TABLA1:</b>	<b>.word 11, 2, 6, 7, 4, 21, 10, 4</b>		<b>andi \$t5, \$t4, 1</b>
<b>CANT:</b>	<b>.word 8</b>		<b>beqz \$t5, FUERA</b>
<b>RES:</b>	<b>.word 0</b>		<b>daddi \$t1, \$t1, 1</b>
<b>TABLA2:</b>	<b>.word 0</b>		<b>sd \$t4, TABLA2(\$t3)</b>
<b>.code</b>			<b>daddi \$t3, \$t3, 8</b> <i>muevo posición en TABLA2</i>
	<b>ld \$t0, CANT(\$zero)</b> <i>carra contador</i>	<b>FUERA:</b>	<b>daddi \$t2, \$t2, 8</b>
	<b>daddi \$t1, \$zero, \$zero</b> <i>T=RES</i>		<b>daddi \$t0, \$t0, -1</b>
	<b>daddi \$t2, \$t0, 0</b> <i>✓</i>		<b>bnez \$t0, LOOP</b>
	<b>daddi \$t3, \$zero, \$zero</b>		<b>sd \$t1, RES(\$t0)</b> <i>✓</i>
			<b>halt</b>

3. Describa brevemente cómo se generan los atascos del tipo Read after Write (RAW). (5 pts)

Los atascos RAW se generan cuando una instrucción intenta acceder a un valor que no se ha terminado de escribir en una instrucción previa. Ej: un daddi \$t1, \$t1, 1 seguida de beqz \$t1, \$t2, algo.

4. Escribir un programa que use la información de 5 pares de números guardados en la variable pares, y dibuje en la pantalla gráfica 5 puntos naranjas de acuerdo a la diferencia en valor absoluto de los valores de cada par. (40 pts)

La variable pares contiene todos los valores de los pares, en la primera y segunda posición se guarda el primer par, en posición 3 y 4 el segundo, y así sucesivamente.

El procesamiento de los pares lo debe hacer la subrutina GRAFICAR que recibe por referencia el arreglo de pares y la cantidad. Para cada par de números leídos calcula la diferencia en valor absoluto y dibuja en la pantalla gráfica del simulador un punto naranja en la posición (diferencia, diferencia). El cálculo de la diferencia se realiza en la subrutina DIFERENCIA que recibe los dos números del par y devuelve la diferencia en valor absoluto. Recuerde que la diferencia en valor absoluto se obtiene restando el número mayor al menor.

En todo el ejercicio se debe usar la convención para nombrar a los registros.

```
.data
CONTROL: .word 0x10000
DATA: .word 0x10008

naranja: .byte 255, 128, 0, 0
pares: .word 10, 15, 24, 12, 30, 31, 42, 21, 33, 33
```

# data

```
CONTROL .word 0x10000
DATA .word 0x10008
NARANJA .byte 255, 128, 0, 0
PARES .word 10, 15, 24, 12, 30, 31, 42, 21, 33, 33
```

## code

```
ld $s0, CONTROL(0) ✓
ld $s1, DATA(0) ✓
ld $s2, NARANJA(0) ✓
sd $s2, 0($s1) # mando el color a Data
daddi $sp, $sp, 0x400 # inicio SP
daddi $a0, $zero, 5 # paso cont de pares
daddi $a1, $0, PARES # paso direccion de vector
jal graficar ✓
halt
```

diferencia: sll \$v0, \$a1, \$a2 ✓

bnez \$v0, menor ✓

dsub \$v0, \$a1, \$a2 ✓

jolver ✓

menor: dsub \$v0, \$a2, \$a1 ✓

wolver: jr \$ra ✓

## graficar:

```
daddi $sp, $sp, -24 ✓
sd $a0, $sp(0) ✓
sd $a1, $sp(4) ✓
sd $ra, $sp(8) ✓
```

# salvo a0, a1, \$ra

```
dadd $t0, $a0, $0 # T0 = contador de loops
```

```
dadd $t1, $a1, $0 # t1 = direccion de vector
```

```
daddi $t2, $0, 0 # posicion aux
```

lazo: ld \$a0, \$t1(\$t2) # 1º valor del par en a0

```
daddi $t2, $t2, 8 ✓
```

```
ld $a1, $t1($t2) # 2º valor en a1
```

```
daddi $t2, $t2, 8 ✓
```

jal diferencia ✓

```
sd $v0, 4($s1) # mando valor a eje X
```

```
sd $v0, 5($s1) # mando " " " " Y
```

```
daddi $t3, $0, 5 # deberia haberlo inicializado afuera
```

```
sd $t3, 0($s0) ✓
```

```
daddi $t0, $t0, -1 # contador
```

```
bnez $t0, lazo ✓
```

```
ld $ra, $sp(16) ✓
```

```
ld $a1, $sp(8) ✓
```

```
ld $a0, $sp(0) ✓
```

```
daddi $sp, $sp, 24 ✓
```

# rescato de la pila

```
jr $ra ✓
```