

## Pregunta 5:

1. Desarrolla un programa que, dados dos vectores  $a$  y  $b$  de números en CF de doble precisión, calcule lo siguiente:  $C(i) = a(i) \cdot cte + b(i)$ . La  $cte$  es un número real en doble precisión. Reserva el espacio necesario para el vector  $C$ . Se recomienda esquematizar el cauce SOLO al lado de las paradas para comprender qué sucede.
2. Incluye el código antes de la optimización con dibujos del cauce según se sugiere, identifica paradas y obtén razonadamente  $N^\circ$  ciclos
3. Incluye código optimizado eliminando paradas y usando HR intentando un  $CPI=1$  si es posible, explicando si no, por qué. Obten  $n^\circ$  ciclos y Ganancia

### Contenido

|               |   |
|---------------|---|
| 1. Y 2.....   | 2 |
| Código: ..... | 2 |
| Flujo:.....   | 3 |
| 3.....        | 4 |
| Código: ..... | 4 |
| Flujo:.....   | 6 |

## 1. Y 2.

### Código:

```
.data 64
a: .double 165.8585849383, 45.67, 52.78, 99.0008, 85.2222224
b: .double 168.2324094432, 98.67, 67.34, 85.45, 55.3457886
K: .double 65.98
c: .space 45
```

```
.text 2000
; r1 = indice
; r2 = cantidad

; f0 = a(i)
; f2 = b(i)
; f6 = K
; f8 = a(i) por K+b(i)
; Algoritmo: c(i) = a(i) por K + b(i)
```

ini:

```
xor r1, r1, r1
addi r2, r0, #5 ; len(a)
ld f6, K(r0)
```

bucle:

```
ld f0, a(r1)
ld f2, b(r1)

add f8, f6, f2 ; RAW 1 parada
multd f8, f0, f8 ; RAW 1 parada, WAW 2 paradas

sd c(r1), f8 ; RAW 4 paradas, REstructural 1 parada

addi r1, r1, #8 ; REstructural 4 paradas

subi r2, r2, #1 ; REstructural 2 paradas

bnez r2, bucle ; RAW 1 parada
nop ; REstructural 1 parada
```

trap #6

```
; Num ciclos = 3c + 5 * (2c + (1cadd + 1p) + (1cmul + 3p) + (1c + 5p) + (1c + 4p) + (1c + 2p) + (1c + 1p) + (1 HR + 1p) ) + 1c = 134c
; Num ciclos = 83c
```

```

1  .data 64
2  a: .double 165.8585849383, 45.67, 52.78, 99.0008, 85.2222224
3  b: .double 168.2324094432, 98.67, 67.34, 85.45, 55.3457886
4  K: .double 65.98
5  c: .space 45
6
7  .text 2000
8  ; r1 = indice
9  ; r2 = cantidad
10
11 ; f0 = a(i)
12 ; f2 = b(i)
13 ; f6 = K
14 ; f8 = a(i) por K+b(i)
15 ; Algoritmo: c(i) = a(i) por K + b(i)
16
17 ini:
18     xor r1, r1, r1
19     addi r2, r0, #5 ; len(a)
20     ld f6, K(r0)
21 bucle:
22     ld f0, a(r1)
23     ld f2, b(r1)
24
25     addd f8, f6, f2 ; RAW 1 parada
26     multd f8, f0, f8 ; RAW 1 parada, WAW 2 paradas
27
28     sd c(r1), f8 ; RAW 4 paradas, REstructural 1 parada
29
30     addi r1, r1, #8 ; REstructural 4 paradas
31
32     subi r2, r2, #1 ; REstructural 2 paradas
33
34     bnez r2, bucle ; RAW 1 parada
35     nop ; REstructural 1 parada
36
37     trap #6
38
39 ; Num ciclos = 3c + 5 * (2c + (1cadd + 1p) + (1cmul + 3p) + (1c + 5p) + (1c + 4p) + (1c + 2p) + (1c + 1p) + (1 HR + 1p) ) + 1c = 134c
40 ; Num ciclos = 83c

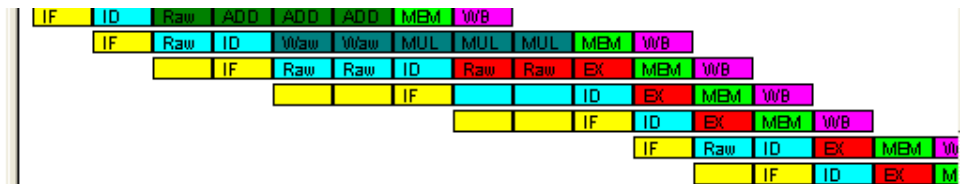
```

## Flujo:

```

addd f8, f6, f2
multd f8, f0, f8
sd c(r1), f8
addi r1, r1, #8
subi r2, r2, #1
bnez r2, bucle
nop

```



Como se puede observar se producen muchas paradas debidas a riesgos RAW, WAW y Riesgos Estructurales.

3.

Código:

.data 64

a: .double 165.8585849383, 45.67, 52.78, 99.0008, 85.2222224

b: .double 168.2324094432, 98.67, 67.34, 85.45, 55.3457886

K: .double 65.98

c: .space 45

.text 2000

; r1 = indice

; r2 = cantidad

; f0 = a(i)

; f2 = b(i)

; f6 = K

; f8 = a(i) por K+b(i)

; Algoritmo:  $c(i) = a(i) \text{ por } K + b(i)$

ini:

xor r1, r1, r1

addi r2, r0, #5 ; len(a)

ld f6, K(r0)

bucle:

ld f2, b(r1)

ld f0, a(r1)

add f8, f6, f2

multd f8, f0, f8 ; WAW 2 paradas

subi r2, r2, #1 ; REstructural 1 parada

sd c(r1), f8 ; REstructural 1 parada, RAW 2 paradas

bnez r2, bucle ; REstructural 3 paradas

addi r1, r1, #8 ; REstructural 2 paradas

trap #6

; Num ciclos = 63c

; Ganancia = 83/63 = 32%

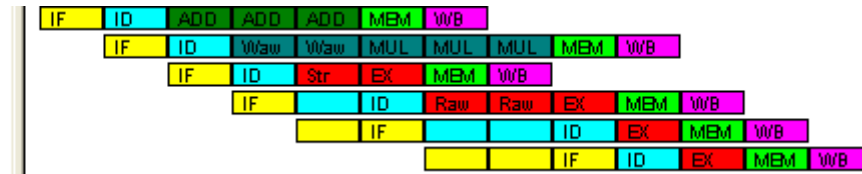
```
1  .data 64
2  a: .double 165.8585849383, 45.67, 52.78, 99.0008, 85.2222224
3  b: .double 168.2324094432, 98.67, 67.34, 85.45, 55.3457886
4  K: .double 65.98
5  c: .space 45
6
7  .text 2000
8  ; r1 = indice
9  ; r2 = cantidad
10
11  ; f0 = a(i)
12  ; f2 = b(i)
13  ; f6 = K
14  ; f8 = a(i) por K+b(i)
15  ; Algoritmo: c(i) = a(i) por K + b(i)
16
17  ini:
18  xor r1, r1, r1
19  addi r2, r0, #5 ; len(a)
20  ld f6, K(r0)
21  bucle:
22  ld f2, b(r1)
23  ld f0, a(r1)
24
25  addd f8, f6, f2
26
27  multd f8, f0, f8 ; WAW 2 paradas
28  subi r2, r2, #1 ; REstructural 1 parada
29  sd c(r1), f8 ; REstructural 1 parada, RAW 2 paradas
30
31  bnez r2, bucle ; REstructural 3 paradas
32  addi r1, r1, #8 ; REstructural 2 paradas
33
34  trap #6
35
36  ; Num ciclos = 63c
37  ; Ganancia = 83/63 = 32%
```

Flujo:

```

addd f8, f6, f2
mulld f8, f0, f8
subi r2, r2, #1
sd c(r1), f8
bnez r2, bucle
addi r1, r1, #8

```



Se ha optimizado el código quitando principalmente los riesgos Raw y algunos estructurales. Para optimizarlo más se podrían meter instrucciones entre medias de estas, de por ejemplo otro algoritmo, y así al juntarse evitan los riesgos WAW y RAW.