

1. Sea un computador superescalar capaz de decodificar 2inst/c, emitir 2inst/c, escribir 2inst/c en los registros correspondientes y retirar 2 inst/c. Dispone de buffer de reorden con número de entradas suficiente, la emisión (ejecución) puede ser desordenada y la finalización ordenada. El siguiente fragmento de programa aplica un filtro lineal sobre un vector A y almacena el resultado en un vector B:

```
for (i = 1; i < 100; i = i + 1) //99 iteraciones !!
{
  b[i] = coef*a[i-1] + a[i] ;
}
```

El compilador lo traduce al siguiente código:

```

; r1 almacena la dirección de a
; r2 almacena la dirección de b
addi r3,r1,#800 ; condicion de final
addi r1,r1,#8 ; inicialización de los indices
addi r2,r2,#8 ;
ld f0,coef ; cargar coeficiente
loop: ld f2,-8(r1) ; cargar a[i-1]
      ld f4,0(r1) ; cargar a[i]
      muld f8,f2,f0 ; a[i-1]*coef
      addd f4,f8,f4 ; a[i-1]*coef + a[i]
      sd 0(r2),f4 ; almacenar b[i]
      addi r1,r1,#8 ; incrementar indices
      addi r2,r2,#8
      slt r4,r1,r3
      bnez r4,loop
```

Se dispone de las siguientes unidades segmentadas: 2 FP mul/div (4c), 2 FP add (2c), 2 ALU int (1) y 2 load/store (2). Se dispone de un predictor de saltos estático que predice como tomados los saltos hacia atrás y como no tomados los saltos hacia adelante.

a) Planificar las instrucciones en la tabla 1

Tabla 1.

inst	IF	ID/ISS	EX	ROB	WB	renombrado en el ROB

- b) Desenrollar el bucle para obtener 3 copias del cuerpo (haciendo especial hincapié en **eliminar los cálculos redundantes**). Planificarlo asumiendo las latencias anteriores Nota: Las instrucciones de salto no pueden lanzarse a ejecución junto con una instrucción posterior a este.
- c) Calcular la ganancia de velocidad de b) respecto de a)

**a) (0,75) planificación sin errores 0,75 /errores menores 0,6 /errores mayores 0**

**b) (0,5) Desenrollado correcto 0,3 / planificación 0,2**

Se realizan 3 copias y se agrupan las instrucciones para aprovechar el paralelismo (0,1)

Se eliminan cargas redundantes, se ajustan los offset (0,1)

Se modifica la condición de finalización del bucle o los incrementos de punteros (0,1)

**c) (0,25) Correcto 0,25**

a) Como no se dice nada se supone que la ventana de instrucciones tiene el tamaño suficientemente

a.1) En esta solución se tiene en cuenta que no hay adelantamientos ; los operandos se leen del ROB

inst	IF	ID/ISS	EX	RB	WB	comentario
addi r3,r1,#800	1	2	3 alui1	4	5	
addi r1,r1,#8	1	2	3 alui2	(4)	5	
addi r2,r2,#8	2	3	4 alui1	5	6	
ld f0,coef	2	3	4-5 ld1	6	7	
<b>loop:</b> ld f2,#-8(r1)	3	(4)	5-6 ld2	(7)	8	
ld f4,0(r1)	3	4-5	6-7 ld1	8	9	Ld ocupadas
muld f8,f2,f0	4	5 (7)	8-11	(12)	13	
addd f4,f8,f4	4	5 (12)	13-14	(15)	16	
sd 0(r2),f4	5	6 (15)	16-17	18	19	
addi r1,r1,#8	5	6	7 alui1	(8)	19	
addi r2,r2,#8	6	7	8 alui1	9	20	Se renombra r2
slt r4,r1,r3	6	7 (8)	9 alui1	(10)	20	
bnez r4,loop	7	8 (10)	11	12	21	Acierta la predicc.
sig	7	8-11	-			Se anula
<b>loop:</b> ld f2,#-8(r1)	8	9	10-11 ld1	12	21	
ld f4,0(r1)	8	9	10-12 ld2	12	22	

a.2) En esta solución se utilizan adelantamientos

inst	IF	ID/ISS	EX	RB	WB	comentario
addi r3,r1,#800	1	2	3 alui1	4	5	
addi r1,r1,#8	1	2	3 alui2	(4)	5	
addi r2,r2,#8	2	3	4 alui1	5	6	
ld f0,coef	2	3	4-5 ld1	6	7	
<b>loop:</b> ld f2,#-8(r1)	3	4	(5)-6 ld2	(7)	8	
ld f4,0(r1)	3	4-5	6-7 ld1	8	9	Ld ocupadas
muld f8,f2,f0	4	5-6	(7)-10	(11)	12	
addd f4,f8,f4	4	5-10	(11)-12	(13)	14	
sd 0(r2),f4	5	6-12	(13)-14	15	16	
addi r1,r1,#8	5	6	7 alui1	(8)	16	
addi r2,r2,#8	6	7	8 alui1	9	17	Se renombra r2
slt r4,r1,r3	6	7	(8) alui2	(9)	17	
bnez r4,loop	7	8	(9)	10	18	Acierta la predicc.
sig	7	8--	--	--	--	Se anula
<b>loop:</b> ld f2,#-8(r1)	8	9	10-11 ld1	12	20	
ld f4,0(r1)	8	9	10-12 ld2	12	20	

a.3) En esta solución, ventana de instrucciones de 2 entradas y alineada (adelantamientos)

inst	IF	ID/ISS	EX	RB	WB	comentario
addi r3,r1,#800	1	2	3 alui1	4	5	
addi r1,r1,#8	1	2	3 alui2	4	5	
addi r2,r2,#8	2	3	4 alui1	5	6	
ld f0,coef	2	3	4-5 ld1	6	7	
<b>loop:</b> ld f2,#-8(r1)	3	4	5-6 ld2	7	8	
ld f4,0(r1)	3	4-5	6-7 ld1	8	9	Ld ocupadas
muld f8,f2,f0	4-5	6	7-10	11	12	Ventana alineada
addd f4,f8,f4	4-5	6-10	11-12	13	14	Ventana alineada
sd 0(r2),f4	5-10	11-12	13-14	15	16	
addi r1,r1,#8	5-10	11	12 alui1	13	16	
addi r2,r2,#8	6-12	13	14 alui1	15	17	Se renombra r2 en RB
slt r4,r1,r3	6-12	13	14 alui2	15	17	
bnez r4,loop	7-13	14	15	16	18	Acierta la predicc.
sig	7-13	14--	--	--	--	Se anula
<b>loop:</b> ld f2,#-8(r1)	8					
ld f4,0(r1)	8					

b) Desenrollamos el bucle 2 veces.

```

addi r3,r1,#800 ; condición modificada(800-3*8)
addi r1,r1,#8 ; inicialización de los índices
addi r2,r2,#8 ;
ld f0,coef ; cargar coeficiente
loop: ld f2,-8(r1) ; cargar a[i-1]
ld f4,0(r1) ; cargar a[i]
ld f5,8(r1); cargar a[i+1]
ld f6,16(r1); cargar a[i+2]
muld f8,f2,f0 ; a[i-1]*coef
muld f9,f4,f0 ; a[i]*coef
muld f10,f5,f0 ; a[i+1]*coef
addd f4,f8,f4 ; a[i-1]*coef + a[i]
addd f5,f9,f5 ; a[i]*coef + a[i+1]
addd f6,f10,f6 ; a[i+1]*coef + a[i+2]
sd 0(r2),f4 ; almacenar b[i]
sd 8(r2),f5 ; almacenar b[i+1]
sd 16(r2),f6 ; almacenar b[i+2]

addi r1,r1,#24 ; incrementar índices
addi r2,r2,#24
slt r4,r1,r3
bnez r4,loop

```

inst	IF	ID/ISS	EX	RB	WB	comentario
addi r3,r1,#800	1	2	3 alui1	4	5	alui1
addi r1,r1,#8	1	2	3 alui2	(4)	5	alui2
addi r2,r2,#8	2	3	4 alui1	5	6	alui1
ld f0,coef	2	3	4-5 ld1	6	7	ld1
<b>loop:</b> ld f2,-8(r1)	3	4	5-6 ld2	(7)	8	ld2
ld f4,0(r1)	3	4-5	6-7 ld1	(8)	9	Ld1
ld f5,8(r1)	4	5-6	7-8 ld2	9	10	ld2
ld f6,16(r1)	4	5-7	8-9 ld1	10	11	Ld1
muld f8,f2,f0	5	6	7-10 mul1	(11)	12	Mul1
muld f9,f4,f0	5	6-7	8-11 mul2	(12)	13	Mul2
muld f10,f5,f0	6	7-10	11-14 mul1	(15)	16	Mul ocupadas mul1
addd f4,f8,f4	6	7-10	11-12 sum1	(13)	16	sum1
addd f5,f9,f5	7	8-11	12-13 sum2	(14)	17	Sum2
addd f6,f10,f6	7	8-14	15-16 sum1	(17)	18	sum1
sd 0(r2),f4	8	9-12	13-14 sd1	15	18	Sd1
sd 8(r2),f5	8	9-13	14-15 sd2	16	19	Sd2
sd 16(r2),f6	9	10-16	17-18 sd1	19	20	Sd1
addi r1,r1,#24	9	10	11 alui1	(12)	20	Alu1
addi r2,r2,#24	10	11	12 alui2	13	21	Alu2
slt r4,r1,r3	10	11	12 alui1	(14)	21	Alu1 / solo 2 en ROB
bnez r4,loop	11	11-13	14	15	22	Acierta la predicc.
--	11	--				
loop	12					