

Problema 3.1

a)

Desarrollo de 4 iteraciones y renombramiento de registros

```
foo: LD      F2, 0(R1)
      MULTD   F4, F2, F0
      LD      F6, 0(R2)
      ADDD    F6, F4, F6
      SD      0(R2), F6

      LD      F8, 8(R1)      ; Renombrado: F(i+6) ← F(i)
      MULTD   F10, F8, F0
      LD      F12, 8(R2)
      ADDD    F12, F10, F12
      SD      8(R2), F12

      LD      F14, 16(R1)   ; Renombrado: F(i+12) ← F(i)
      MULTD   F16, F14, F0
      LD      F18, 16(R2)
      ADDD    F18, F16, F18
      SD      16(R2), F18

      LD      F20, 24(R1)   ; Renombrado: F(i+18) ← F(i)
      MULTD   F22, F20, F0
      LD      F24, 24(R2)
      ADDD    F24, F22, F24
      SD      24(R2), F24

      ADDI    R1, R1, #32
      ADDI    R2, R2, #32
      SGTI    R3, R1, done
      BEQZ    R3, foo
```

Reordenar para que sean independientes 2 a dos (suponemos suficientes estaciones de reserva)

```
foo: LD      F2, 0(R1)
      LD      F8, 8(R1)

      MULTD   F4, F2, F0
      MULTD   F10, F8, F0

      LD      F6, 0(R2)
      LD      F12, 8(R2)

      ADDD    F6, F4, F6
      ADDD    F12, F10, F12

      SD      0(R2), F6
      SD      8(R2), F12

      LD      F14, 16(R1)
      LD      F20, 24(R1)

      MULTD   F16, F14, F0
      MULTD   F22, F20, F0

      LD      F18, 16(R2)
      LD      F24, 24(R2)

      ADDD    F18, F16, F18
      ADDD    F24, F22, F24

      SD      16(R2), F18
      ADDI    R1, R1, #32

      SD      24(R2), F24
      SGTI    R3, R1, done

      ADDI    R2, R2, #32
      BEQZ    R3, foo
```

Podemos suponer que hay predicción de saltos.

Si hay suficientes recursos de planificación → no se para el pipe

CPI = 0,5 → (IPC=2)

b)

			Issue	Exec	Mem	Write
foo:	LD	F2, 0(R1)	1	-		
	MULTD	F4, F2, F0	2	-		
	LD	F6, 0(R2)	2	-		
	ADDD	F6, F4, F6	3	-		
	SD	0(R2), F6	4	-		
	ADDI	R1, R1, #8	4			
	ADDI	R2, R2, #8	5	-		
	SGTI	R3, R1, done	5	-		
	BEQZ	R3, foo	6	-		

Suponiendo suficientes estaciones de reserva, tan solo es relevante cuando se hace el "issue" → 6 ciclos por iteración (CPI = 6/9)

c)

$T_a = 25 \text{ iter} \times 24 \text{ instr/iter} \times 0,5 \text{ ciclos/instr} = 300 \text{ ciclos}$

$T_b = 100 \text{ iter} \times 9 \text{ instr/iter} \times 6/9 \text{ ciclos/instr} = 600 \text{ ciclos}$

$\text{Speedup} = T_b / T_a = 2$

Es decir, se obtiene una mejora de rendimiento del 100%

Problema 3.2

	Issue	Exec	Write	Comm
LD F2, 0(R1)	1	2	3	4
MULTD F4, F2, F0	2	4-10 ^{\$}	11	12
LD F6, 0(R2)	2	3	4	13
ADDD F6, F4, F6	3	12-16 ^{\$}	17	18
SD 0(R2), F6	4		17	19
ADDI R1, R1, #8	4	5	6	20
ADDI R2, R2, #8	5	6	7	21
SGTI R3, R1, done	5	7 ^{\$}	8	22
BEQZ R3, foo	6	9 ^{\$}	10	23
LD F2, 0(R1)	7	8	9	24
MULTD F4, F2, F0	8	10-16 ^{\$}	18 ⁺	25
LD F6, 0(R2)	8	9	12 ⁺	26
ADDD F6, F4, F6	9	19-23 ^{\$}	24	27
SD 0(R2), F6	10		24	28
ADDI R1, R1, #8	10	11	13 ⁺	29
ADDI R2, R2, #8	11	12	14 ⁺	30
SGTI R3, R1, done	11	14 ^{\$}	15	31
BEQZ R3, foo	12	16 ^{\$}	19 ⁺	32

^{\$}: Dependencia LDE

⁺: Uso de CDB

Problema 3.3

a) Las ER disponibles nunca se agotan en la primera iteración

	ISSUE	EJECUCIÓN	WRITE
ADDI R1,R0,#DIR	1	2	3
LD F0, 0(R7)	2	3-4	5
LD F4, 0(R1)	3	5-6 ^{ESTRUC}	7
DIVD F8,F4,F0	4	7-14 ^{LDE}	15
SUBI R1, R1, #4	5	6 -7 ^{BCD}	8
LD F2, 0(R1)	6	8-9 ^{LDE}	10
MULD F8,F2,F8	7	15-18 ^{LDE}	19
SD 0(R1), F8	8	19-20 ^{LDE}	
SUBI R3,R3,#8	9	10	11
SGTI R5,R3, #1000	10	11	12
BEQZ R5,LOOP	11	12	
NOP	12		

b) En esta parte, dado que en el enunciado no hay límites para las ER, se supone que existen suficientes.

	ISSUE	EJECUCIÓN	WRITE
ADDI R1,R0,#DIR	1	2	3
LD F0, 0(R7)	1	2-3	4
LD F4, 0(R1)	2 ^{LDE}	4-5 ^{ESTRUCTU}	6
DIVD F8,F4,F0	3	6-13 ^{LDE}	14
SUBI R1, R1, #4	3	4	5
LD F2, 0(R1)	4 ^{LDE}	6-7 ^{ESTRUC}	8
MULD F8,F2,F8	5 ^{LDE}	14-17 ^{LDE}	18
SD 0(R1), F8	6	18-19 ^{LDE}	
SUBI R3,R3, #8	6	7	8
SGTI R5,R3, #1000	7 ^{LDE}	8	9
BEQZ R5,LOOP	8	9	
NOP	8		