

Ejercicio 1. Para el fragmento de código siguiente:

```

1. lw r1,0x1ac ; r1 ← M(0x1ac)
2. lw r2,0xc1f ; r2 ← M(0xc1f)
3. add r3,r0,r0 ; r3 ← r0+r0
4. mul r4,r2,r1 ; r4 ← r2*r1
5. add r3,r3,r4 ; r3 ← r3+r4
6. add r5,r0,0x1ac ; r5 ← r0+0x1ac
7. add r6,r0,0xc1f ; r6 ← r0+0xc1f
8. sub r5,r5,#4 ; r5 ← r5 - 4
9. sub r6,r6,#4 ; r6 ← r6 - 4
10. sw (r5),r3 ; M(r5) ← r3
11. sw (r6),r4 ; M(r6) ← r4

```

y suponiendo que se pueden captar, decodificar, y emitir cuatro instrucciones por ciclo, indique el orden en que se emitirán las instrucciones para cada uno de los siguientes casos:

- Una ventana de instrucciones centralizada con emisión ordenada
- Una ventana de instrucciones centralizada con emisión desordenada
- Una estación de reserva de tres líneas para cada unidad funcional, con envío ordenado.

Nota: considere que hay una unidad funcional para la carga (2 ciclos), otra para el almacenamiento (1 ciclo), tres para la suma/resta (1 ciclo), y una para la multiplicación (4 ciclos). También puede considerar que, en la práctica, no hay límite para el número de instrucciones que pueden almacenarse en la ventana de instrucciones o en el buffer de instrucciones.



a)

INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
lw r1, 0x1ac	IF	ID	EX	EX										
lw r2, 0xc1f	IF	ID			EX	EX								
add r3, r0, r0	IF	ID			EX									
mul r4, r2, r1	IF	ID					EX	EX	EX	EX				
add r3, r3, r4		IF	ID								EX			
add r5, r0, 0x1ac		IF	ID								EX			
add r6, r0, 0xc1f		IF	ID								EX			
sub r5, r5, #4		IF	ID								EX			
sub r6, r6, #4			IF	ID							EX			
sw (r5), r3			IF	ID							EX			
sw (r6), r4			IF	ID							EX			

b)

INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
lw r1, 0x1ac	F	O	EX	EX										
lw r2, 0xc1f	F	O			EX	EX								
add r3, r0, r0	F	O	EX											
mul r4, r2, r1	F	D					EX	EX	EX	EX				
add r3, r3, r4		F	D								EX			
add r5, r0, 0x1ac		F	D	EX										
add r6, r0, 0xc1f		F	D	EX										
sub r5, r5, #4		F	D		EX									
sub r6, r6, #4			F	D	EX									
sw (r5), r3			F	D							EX			
sw (r6), r4			F	D							EX			

c)

INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
lw r1, 0x1ac	F	D	EX	EX										
lw r2, 0xc1f	F	D	.		EX	EX								
add r3, r0, r0	F	D	EX											
mul r4, r2, r1	F	D					EX	EX	EX	EX				
add r3, r3, r4		F	D								EX			
add r5, r0, 0x1ac		F	D	EX										
add r6, r0, 0xc1f		F	D	EX										
sub r5, r5, #4		F	D		EX									
sub r6, r6, #4			F	D	EX									
sw (r5), r3			F	D							EX			
sw (r6), r4			F	D							EX			

Operación	Ciclo(orden)	Estado	Flush	Máscara
lw	2	$x^2 f(4)$		
addi	2	f		
addi	2	f		
lw	2	$x^2 x^2 f(6)$		
lw	3	$x^2 x^2 f(8)$		
mult	3	$x^2 x^2 f(11)$		
addi	3	$x^2 x^2 f(12)$		
addi	3	$x^2 x^2 f(13)$		
sw	4	$x^2 x^2 f(7)$		
sw	4	$x^2 x^2 f(13)$		
sub	4	$x^2 f(4)$		

Ejercicio 5. En un procesador superescalar con renombramiento de registros se utiliza un buffer de renombramiento para implementar el mismo. Indique como evolucionarían los registros de renombramiento al realizar el renombramiento para las instrucciones:

1. mul r2,r0,r1 ; $r2 \leftarrow r0 * r1$
 2. add r3,r1,r2 ; $r3 \leftarrow r1 + r2$
 3. sub r2,r0,r1 ; $r2 \leftarrow r0 - r1$
 4. add r3,r3,r2 ; $r3 \leftarrow r3 + r2$

Vamos a señalar las **WDR** y las **DWD** ya que en estos casos se recomienda. Mostremos el código resultante:

$$\begin{aligned}rr_2 &= rr_0 \times rr_1 \\rr_3 &= rr_1 + rr_2 \\rr_4 &= rr_0 + rr_1 \\rr_5 &= rr_3 + rr_4\end{aligned}$$

#	Entrada válida	Registo destino	Valor	Valor Válido	Bit de Último
0	1	r0	[r0]	1	1
1	1	r1	[r1]	1	1
2	01	r2	$r0 \times r1$	1	No
3	011	r3	$r2 + r0$	1	0x0
4	011	r2	$r0 + r1$	1	1
5	011	r3	$(r1+r2) + r0 + r1$	1	1
6	0				
7	0				
8	0				

deser el último estopase
segundo lído (squelego) luego
esto pasaría a O ese
deja como está?

Ejercicio 6. Considere el bucle:

```
i=1;  
do  
{  
    b[i]=a[i]*c;  
    c=c+1;  
    if (c>10) then goto etiqueta; // 1  
    i=i+1;  
} while (i<=10); // 2  
etiqueta:.....
```

Supongo $c=0$.

Indique cuál es la penalización efectiva debida a los saltos, en función del valor inicial de c (número entero), considerando que el procesador utiliza:

a) Predicción fija (siempre se considera que se va a producir el salto)

	$c=1$	$c=2$	$c=3$	$c=4$	$c=5$	$c=6$	$c=7$	$c=8$	$c=9$	$c=10$						
	1 ^a	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	7 ^a	8 ^a	9 ^a	10 ^a	11 ^a	12 ^a	13 ^a	14 ^a	15 ^a	...
Ejec. sc1	N	N	N	N	N	N	N	N	N	N	S					
Ejec. sc2	N	N	N	N	N	N	N	N	N	S						
Hist. sc1																
Hist. sc2																
Pred. sc1	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	
Pred. sc2	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	
	4									4						
	4									4						

Ciclos perdidos.

$$c = 8 \sum_{i=0}^9 i = 8 \cdot \frac{9(8)}{2} = 32 \cdot 9 = 288 \text{ cm}^3$$

b) Predicción estática (si el desplazamiento es negativo se toma y si es positivo no)

	$C=1$	$C=2$	$C=3$	$C=4$	$C=5$	$C=6$	$C=7$	$C=8$	$C=9$	$C=10$	$C=11$				
	1 ^a	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	7 ^a	8 ^a	9 ^a	10 ^a	11 ^a				
Ejec. sc1	N	N	N	N	N	N	N	N	N	N	(S)				
Ejec. sc2	S	S	S	S	S	S	S	S	S	S	N				
Hist. sc1															
Hist. sc2															
Pred. sc1	N	N	N	N	N	N	N	N	N	N	N				
Pred. sc2	S	S	S	S	S	S	S	S	S	S	S				
P_1	0	0	0	0	0	0	0	0	0	0	14				
P_2	0	0	0	0	0	0	0	0	0	0	* No se cumplió				

c) Predicción dinámica con un bit (1= Saltar; 0 = No Saltar; Inicialmente está a 1)

c) Predicción dinámica con un bit (1 = Salir, 0 = No Salir, inicialmente está a 0).
 i=2 i=3 i=4 i=5 i=6 i=7 i=8 i=9 i=10 i=11
 C=1 C=2 C=3 C=4 C=5 C=6 C=7 C=8 C=9 C=10 C=11

	1 ^a	2 ^a	3 ^a	4 ^a	5 ^a	6 ^a	7 ^a	8 ^a	9 ^a	10 ^a	11 ^a	12 ^a	13 ^a	14 ^a	15 ^a	...
Ejec. sc1	22	22	22	22	22	22	22	22	22	22	5					
Ejec. sc2																
Hist. sc1	-1	0	0	0	0	0	0	0	0	0	0	1				
Hist. sc2	-1	0	0	0	0	0	0	0	0	0	-1					
Pred. sc1	00	022	022	022	022	022	022	022	022	022	022	5				
Pred. sc2	00	022	022	022	022	022	022	022	022	022	022					

$$\begin{array}{ccccccccc} P_1 & \boxed{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ P_2 & \boxed{4} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \quad \boxed{14} + \rightarrow 12 \text{ ciclos.}$$

Ejercicio 8. En un programa, una instrucción de salto condicional (a una dirección de salto anterior) dada tiene el siguiente comportamiento en una ejecución de dicho programa:

SSNNNNSNSNSNSSSSSN

donde S indica que se produce el salto y N que no. Indique la penalización efectiva que se introduce si se utiliza:

a) Predicción fija (siempre se considera que se no se va a producir el salto)

b) Predicción estática (si el desplazamiento es negativo se toma y si es positivo no)

c) Predicción dinámica con dos bits, inicialmente en el estado (11).

d) Predicción dinámica con tres bits, inicialmente en el estado (111).

