

## Problema

Suponga un procesador VLIW que puede codificar tres operaciones en cada instrucción con las restricciones que indica la siguiente tabla:

Unidad funcional	Latencia	Op1	Op2	Op3	Operaciones realizadas por la unidad
ALU entera	1	X	X	X	Comparación/suma/resta entera, operaciones lógicas
ALU FP	4			X	Aritmética de coma flotante
Memoria	3	X	X		Cargas y almacenamientos
Salto	3			X	Salto condicionales e incondicionales

Muestre la ejecución del siguiente código en el procesador VLIW indicado previamente:

```
void sum (int c[], int a[], int b[], int n) {
    int i;
    for (i=0; i<n; i++)
        if (a[i] = 0) then
            c[i]:=a[i]+b[i];
        else
            c[i]:=a[i]+1;
}
```

en los siguientes supuestos:

- Sin utilizar instrucciones con predicado, ni procesamiento especulativo.
- Utilizando instrucciones con predicado. El procesador puede utilizar predicados con cualquier instrucción, considerándose las instrucciones de manipulación de predicados como instrucciones de comparación a efectos de latencias.
- Utilizando procesamiento especulativo considerando que la rama `then` es la más probable.

## Solución

- Secuencia de ejecución de las instrucciones VLIW sin predicados.

Ciclos	Operación 1	Operación 2	Operación 3
1	Inicio: LD R1,0 (Ra)	LD R2,0 (Rb)	
2	SUBI Ra,Ra,#4	SUBI Rb,Rb,#4	
3			
4			BNEZ R1,else
5			
6			
7	then: ADD R3,R1,R2		JMP fin
8			
9			
10	else: ADDI R3,R1,#1		
11	fin: SD 0 (Rc),R3	SUBI Rc,Rc,#4	BNEZ Ra,inicio
12			
13			

b) Secuencia de ejecución de las instrucciones VLIW con predicados.

Ciclos	Operación 1	Operación 2	Operación 3
1	Inicio: LD R1,0 (Ra)	LD R2,0 (Rb)	
2	SUBI Ra,Ra,#4	SUBI Rb,Rb,#4	
3			
4	PRE_EQ p1,p2,R1,#0		
5	ADD R3,R1,R2 (p1)	ADDI R3,R1,#1 (p2)	
6	SD 0 (Rc),R3	SUBI Rc,Rc,#4	BNEZ Ra,inicio
7			
8			

c) Secuencia de ejecución de las instrucciones VLIW con ejecución especulativa considerando que la rama then del if es la más probable.

Ciclos	Operación 1	Operación 2	Operación 3
1	Inicio: LD R1,0 (Ra)	LD R2,0 (Rb)	
2	SUBI Ra,Ra,#4	SUBI Rb,Rb,#4	
3			
4	ADD R3,R1,R2		BNEZ R1,else
5			
6			
7			JMP fin
8			
9			
10	else: ADDI R3,R1,#1		
11	Fin: SD 0 (Rc),R3	SUBI Rc,Rc,#4	BNEZ Ra,inicio
12			
13			