

AIC

Tema 3

Procesadores VLIW

Ejercicio

Tenemos un procesador VLIW donde pueden predicarse las instrucciones. Dispone de dos slots, donde las instrucciones de comparación únicamente pueden ejecutarse en el primero, y las instrucciones de carga/almacenamiento en el segundo. El resto de instrucciones pueden ejecutarse en ambos slots.

Las instrucciones de comparación que utilizan los predicados son del tipo:

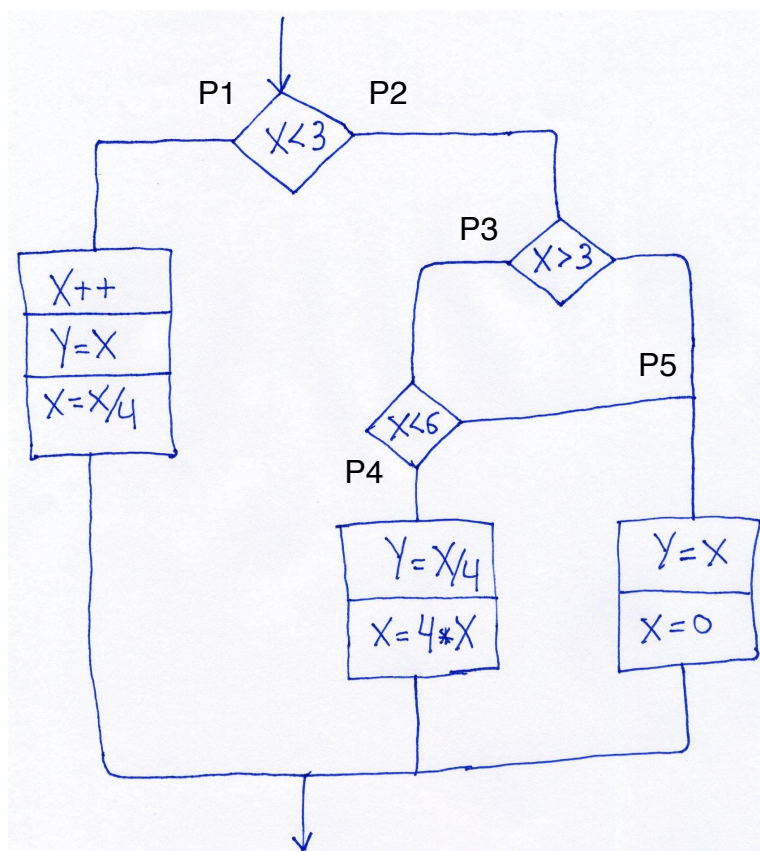
$P1, P2 \text{ cmp.}[gt,lt,eq,ne] \text{ rx, ry}$

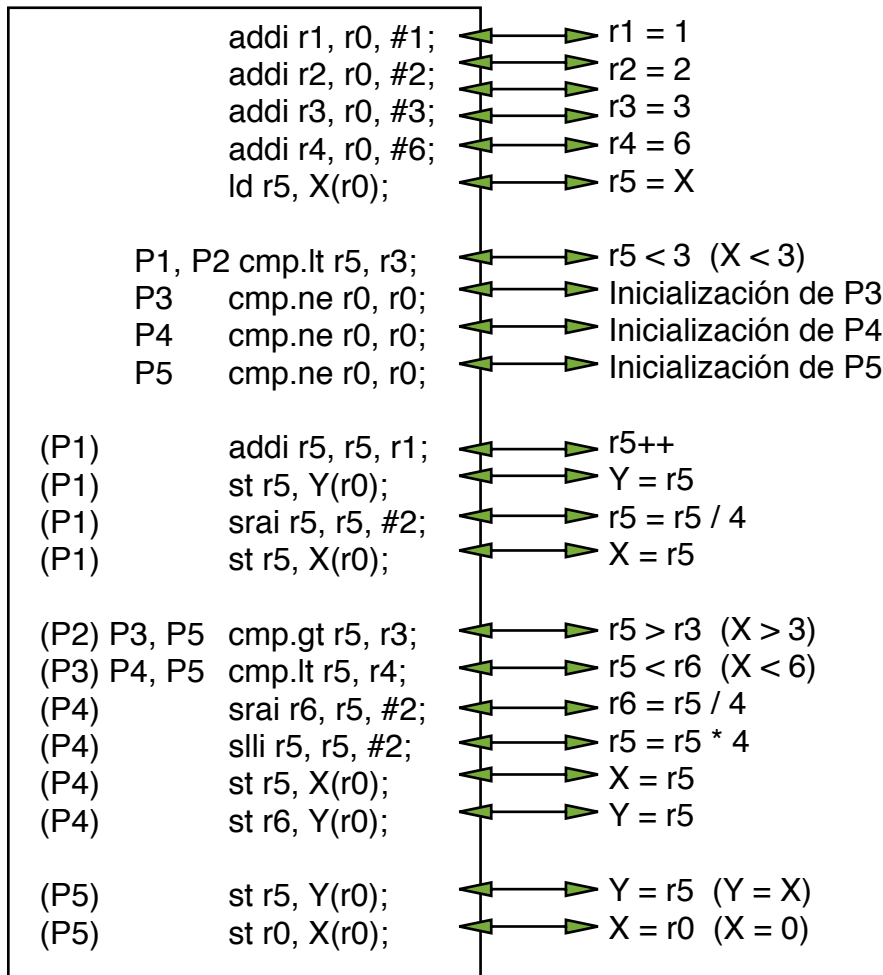
Dado el siguiente código:

```
if (X < 3)
{
    X++;
    Y = X;
    X = X/4;
}
else
{
    if (X > 3)
    {
        Y = X/4;
        X = 4*X;
    }
    else
    {
        Y = X;
        X = 0;
    }
}
```

Realiza el diagrama de flujo, el código en ensamblado y su distribución en los distintos slots.

Solución:





Slot 1	Slot 2
addi r3, r0, #3	ld r5, X(r0)
P1, P2 cmp.lt r5, r3	addi r1, r0, #1
P3 cmp.ne r0, r0	addi r2, r0, #2
P4 cmp.ne r0, r0	addi r4, r0, #6
P5 cmp.ne r0, r0	(P1) addi r5, r5, r1
(P1) srai r5, r5, #2	(P1) st r5, Y(r0)
(P2) P3, P5 cmp.gt r5, r3	(P1) st r5, X(r0)
(P3) P4, P5 cmp.lt r5, r4	(P4) srai r6, r5, #2
(P4) slli r5, r5, #2	(P4) st r6, Y(r0)
	(P4) st r5, X(r0)
	(P5) st r5, Y(r0)
	(P5) st r0, X(r0)

- No hay conflicto ya que "st" leerá el valor de "r5" antes de que "srai" cambie su valor.

- Adelantamos el almacenamiento de "r" ya que no hay dependencias y "r5" no podrá almacenarse hasta el siguiente ciclo.