

### **EJERCICIO TEMA 3 AIC**

Tenemos un procesador VLIW en el que todas las instrucciones deben predicarse. Para establecer el valor de un predicado se utiliza la instrucción `cmp` con el siguiente formato:

$$p_i, p_j \text{ cmp. ('lt' | 'le' | 'gt' | 'ge' | 'eq' | 'ne') } x, y$$

Donde  $p_i$  y  $p_j$  son los predicados a los que se les dará valor y  $x$  e  $y$  son las variables de la comparación. Si la comparación es cierta,  $p_i = 1$  y  $p_j = 0$ , y viceversa en caso contrario.

Dado el siguiente fragmento de código, construye el código vliw para que el código se ejecute sin saltos. El procesador tiene dos slots, siendo el segundo exclusivo para operaciones de asignación de predicados.

```
for (int i = 0; i < 4; i++)
{
    if (x[i] < y[i])
    {
        z[i] = x[i];
    }
    else
    {
        z[i] = y[i]
    }
}
```

- Desenrollamos el bucle para eliminar los saltos derivados de él.
- Suponemos que el predicado  $p_0$  tiene el valor fijo 1.

INSTRUCCIONES PREDICADAS	
[ 1] (p0) ld r0, 0(x)	[11] (p0) ld r0, 2(x)
[ 2] (p0) ld r1, 0(y)	[12] (p0) ld r1, 2(y)
[ 3] (p0) p1,p2 cmp.lt r0,r1	[13] (p0) p1,p2 cmp.lt r0,r1
[ 4] (p1) st 0(z), r0	[14] (p1) st 2(z), r0
[ 5] (p2) st 0(z), r1	[15] (p2) st 2(z), r1
[ 6] (p0) ld r2, 1(x)	[16] (p0) ld r2, 3(x)
[ 7] (p0) ld r3, 1(y)	[17] (p0) ld r3, 3(y)
[ 8] (p0) p3,p4 cmp.lt r2,r3	[18] (p0) p3,p4 cmp.lt r2,r3
[ 9] (p3) st 1(z), r2	[19] (p3) st 3(z), r2
[10] (p4) st 1(z), r3	[20] (p4) st 3(z), r3

SLOT A	SLOT B
[ 1] (p0) ld r0, 0(x)	NOOP
[ 2] (p0) ld r1, 0(y)	NOOP
[ 6] (p0) ld r2, 1(x)	[ 3] (p0) p1,p2 cmp.lt r0,r1
[ 4] (p1) st 0(z), r0	[ 5] (p2) st 0(z), r1
[ 7] (p0) ld r3, 1(y)	NOOP
[11] (p0) ld r0, 2(x)	[ 8] (p0) p3,p4 cmp.lt r2,r3
[ 9] (p3) st 1(z), r2	[10] (p4) st 1(z), r3
[12] (p0) ld r1, 2(y)	NOOP
[16] (p0) ld r2, 3(x)	[13] (p0) p1,p2 cmp.lt r0,r1
[14] (p1) st 2(z), r0	[15] (p2) st 2(z), r1
[17] (p0) ld r3, 3(y)	NOOP
NOOP	[18] (p0) p3,p4 cmp.lt r2,r3
[19] (p3) st 3(z), r2	[20] (p4) st 3(z), r3