

### Ejercicio Tema 3

En un procesador VLIW cuyas instrucciones pueden codificar dos operaciones (dos campos o slots en cada instrucción VLIW), todas las operaciones pueden predicarse. Para establecer los valores de los predicados se usan instrucciones de comparación (cmp) con el formato (p) p1, [p2] cmp.cnd x,y donde cnd es la condición que se comprueba entre x e y. Si la condición es verdadera entonces p1=1 y p2=0 y si es falsa p1=0 y p2=1. La operación solo se ejecuta si el predicado p=1. Indica cómo se escribiría el siguiente código en lenguaje ensamblador sin ninguna operación de salto y con el mínimo número de instrucciones VLIW teniendo en cuenta que las operaciones de comparación solo pueden aparecer en el primer campo o slot de la instrucción VLIW.

```
for i=1 to 2 do
  if ((X[i]/8)==4) then
    X[i]=16*X[i];
  else
    if(X[0]<4) then X[i]=4;
  finIf
finFor
```

### Solución:

#### Iteración 1:

p1 cmp.eq r0,r0;	p1=1
p2 cmp.ne r0,r0;	p2=0
p3 cmp.ne r0,r0;	p3=0
p4 cmp.ne r0,r0;	p4=0

  

(p1) lw r1, X(r0);	r1 = X[0]
(p1) lw r2, #4;	r2 = 4
(p1) lw r3, X+4(r0);	r3 = X[1]
(p1) sllr r4, r3, #3;	r4 = X[1]/8 (desplazamiento a la derecha para dividir)
(p1) p2, p3 cmp.eq r4, r2;	
(p2) slli r4, r3, #4;	r4 = 16*X[1] (desplazamiento a la izquierda para multiplicar)
(p2) sw X+4(r0), r4;	
(p3) p4 cmp.lt r1, r2;	
(p4) sw X+4(r0), r2;	X[1]=2

### Iteración 2:

p5 cmp.eq r0,r0

p6 cmp.ne r0,r0

p7 cmp.ne r0,r0

(p5) lw r5, X+8(r0);            r5 = X[2]

(p5) sllr r6, r5, #3;            r6 = X[2]/8

(p5) p6, p7 cmp.eq r6, r2;

(p6) slli r6, r5, #4;            r6 = 16\*X[2]

(p6) sw X+8(r0), r6;

(p7) p8 cmp.lt r1, r2

(p8) sw X+8(r0), r2

Colocamos ahora las instrucciones VLIW en los SLOTS

<u>SLOT 1</u>	<u>SLOT 2</u>
p1 cmp.eq r0,r0	
p2 cmp.ne r0,r0	(p1) lw r1, X(r0)
p3 cmp.ne r0,r0	(p1) lw r2, #4
p4 cmp.ne r0,r0	(p1) lw r3, X+4(r0)
p5 cmp.eq r0,r0	(p1) sllr r4, r3, #3
(p1) p2, p3 cmp.eq r4, r2	(p5) lw r5, X+8(r0)
p6 cmp.ne r0,r0	(p2) slli r4, r3, #4
p7 cmp.ne r0,r0	(p2) sw X+4(r0), r4
(p3) p4 cmp.lt r1, r2	(p5) sllr r6, r5, #3
p8 cmp.ne r0,r0	(p4) sw X+4(r0), r2
(p5) p6, p7 cmp.eq r6, r2	
(p7) p8 cmp.lt r1, r2	(p6) slli r6, r5, #4
(p8) sw X+8(r0), r2	(p6) sw X+8(r0), r6