En un procesador VLIW con 2 slots todas las operaciones pueden predicarse para establecer los valores de los predicados se utilizan instrucciones de comparación (cmp) con el formato (p) p1, [p2] cmp.cnd x,y donde cnd es la condición. Si se cumple la condición p1=1 y p2=0, si no p1=0 y p2=1. La operación solo se ejecuta si p=1. Indica cómo se escribiría el siguiente código en lenguaje ensamblador desenrollando el bucle, teniendo en cuenta que las operaciones solo pueden aparecer en el primer slot y con el mínimo de instrucciones VLIW.

para i=1 hasta 2 hacer  
 si X[i]/8 == 4 entonces  
 X[i] = 16\*X[i]  
 sino:  
 si X[0] < 4 entonces  
 X[i] = 4  
 finsi  
 finsi  
finpara

**Iteración 1:**

|  |  |  |
| --- | --- | --- |
|  | p1 cmp.eq r0,r0; | p1=1 |
|  | p2 cmp.ne r0,r0; | p2=1 |
|  | p3 cmp.ne r0,r0; | p3=1 |
|  | p4 cmp.ne r0,r0; | p4=1 |
|  |  |  |
| (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 |
| (p1) | p2, p3 cmp.eq r4,r2; |  |
| (p2) | slli r4,r3,#4; | r4 = 16\*X[1] |
| (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; |  |
|  | p8 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 las instrucciones en los slots.

|  |  |
| --- | --- |
| **Slot 1** | **Slot 2** |
| 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; |