

Paso 1)

Forwarding (adelantamientos/cortocircuitos):

Inst. 1 (add r1,r0,x) → Forwarding a Inst. 2 (add r4,r1,#64) en ciclo 4 — 0 ciclos de parada
← Esta no cuenta, está fuera del bucle.

Inst. 6 (lw r12,0(r1)) → Forwarding a Inst. 7 (add r12,r10,r12) en ciclo 10 – 1 ciclo de parada

Inst. 8 (lw r14,0(r2)) → Forwarding a Inst. 9 (add r14,r12,r14) en ciclo 13 – 1 ciclo de parada

Inst. 9 (add r14,r12,r14) → Forwarding a Inst. 10 (sw 0(r3),r14) en ciclo 15 – 0 ciclos de parada

Inst. 14 (seq r5,r4,r1) → Forwarding a Inst. 15 (beqz r5,loop) en ciclo 19 – 3 ciclos de parada (salto)

Instrucciones totales = 10

Total ciclos de parada = 5

Ciclo final (WB de la instrucción 15): 21

Cálculos a mano

Intenta partirlo en “antes del bucle + bucle + después del bucle”

Instrucciones Totales = $5 + (16 * 10) + 1 = 166$

Ciclos de parada totales = $0 + (16 * 5) + 0 = 80$

Ciclos totales: Instrucciones totales + Ciclos de parada totales + 4 (4 es porque la primera instrucción tarda 4 ciclos extra más el suyo propio en salir del pipeline. → $80 + 166 + 4 = 250$.

Estadísticas del simulador

Instrucciones Totales = 166

Ciclos de parada totales = 80

Ciclos totales = 250

Coinciden los resultados → Sí

CPI = ciclos / instrucciones → $250 / 166 = 1.506$ CPI

Paso 2)

Primera iteración

Inst. ejecutadas: 10

Ciclos de parada: $8(\text{datos}) + 3(\text{control}) = 11$

¿Cuántos ciclos de parada se producen cada vez que se ejecuta la instrucción de salto en cada una de las iteraciones del bucle?

5 (2+3) ciclos de parada en las primeras 15 iteraciones

2 (2+0) ciclos de parada en la última iteración

Ciclos: 345 | Instrucciones: 166 | Ciclos de Parada: 177

CPI: $345/166 = 2.0783$

Paso 3)

Ciclos: 247

Instrucciones: 166

Ciclos de Parada: 77

CPI: 1.4879

Calculamos mejora en (%)

vs Paso 1 (solo con adelantamiento) $\rightarrow \text{CPI}_{\text{paso1}} / \text{CPI}_{\text{paso3}} = ((1,506/1,4879) - 1) \times 100$
 $= 1,209\%$

vs Paso 2 (solo con Pred Not-Taken) $\rightarrow \text{CPI}_{\text{paso2}} / \text{CPI}_{\text{paso3}} = ((2,078/1,4879) - 1) \times 100$
 $= 39,68\%$

Paso 4)

(Código más abajo).

Forwarding (adelantamientos/cortocircuitos):

~~Inst. 1 (add r1,r0,x) -> Forwarding a Inst. 2 (add r4,r1,#64) en ciclo 4 – 0 ciclos de parada~~
← Esta no cuenta, está antes del bucle.

Inst. 6 (lw r12,0(r1)) -> Forwarding a Inst. 8 (add r12,r10,r12) en ciclo 10 – 0 ciclos de parada

Inst. 7 (lw r14,0(r2)) -> Forwarding a Inst. 9 (add r14,r12,r14) en ciclo 11 – 0 ciclos de parada

Inst. 8 (add r12,r10,r12) -> Forwarding a Inst. 9 (add r14,r12,r14) en ciclo 11 – 0 ciclos de parada

Inst. 9 (add r14,r12,r14) -> Forwarding a Inst. 10 (sw 0(r3),r14) en ciclo 13 – 0 ciclos de parada

Inst. 14 (seq r5,r4,r1) -> Forwarding a Inst. 15 (beqz r5,loop) en ciclo 17 – 0 ciclos de parada (salto falla, +3)

Instrucciones totales = 10

Total ciclos de parada = 0(datos) + 3(fallo de predicción) = 3 ciclos

Estadísticas del simulador

Ciclos: 215

Instrucciones: 166

Ciclos de Parada: 45

CPI: 1.2951

Mejora respecto al paso anterior:

$$\text{CPI}_{\text{paso3}} / \text{CPI}_{\text{paso4}} = ((1.4879/1.2951) - 1) \times 100 = 14.88\%$$

Código:

```
; z = a + x + y
; Tamaño de los vectores: 16 palabras
; Vector x
```

```

.data
x: .word 0,1,2,3,4,5,6,7,8,9
   .word 10,11,12,13,14,15

; Vector y
y: .word 100,100,100,100,100,100,100,100
   .word 100,100,100,100,100,100,100,100

; Vector z
; 16 elementos son 64 bytes.
z: .space 64

; escalar a
a: .word -10

; El código
.text

start:
    add r1,r0,x
    add r4,r1,#64 ; 16*4
    add r2,r0,y
    add r3,r0,z
    lw r10,a(r0)

loop:
    lw r12,0(r1)
    lw r14,0(r2)
    add r12,r10,r12
    add r14,r12,r14
    sw 0(r3),r14
    add r1,r1,#4
    add r2,r2,#4
    add r3,r3,#4
    seq r5,r4,r1
    beqz r5,loop
    trap #0 ; Fin de programa

```