Ejercicio 05 – Multiplicación por 2

Programa de ejemplo

```
.data
cant:  .word 8
datos: .word 1, 2, 3, 4, 5, 6, 7, 8
res:   .word 0

.code
       dadd r1, r0, r0          ; R1 = desplazamiento (0)
       ld r2, cant(r0)          ; R2 = cantidad restante
loop:  ld r3, datos(r1)         ; R3 = dato de tabla
       daddi r2, r2, -1         ; Decrementar cantidad
       dsll r3, r3, 1           ; Multiplicar dato por 2 (rotar izq)
       sd r3, res(r1)           ; Agregar dato al arreglo de resultados
       daddi r1, r1, 8          ; Incrementar desplazamiento en 8
       bnez r2, loop            ; Repetir mientras queden datos
       nop
       halt
```

Inciso a: función del delay slot

Esta técnica reduce los atascos por dependencia de control ejecutando una instrucción siguiente en el ciclo de espera (en adelante, penalización por salto) para que se realice alguna operación en este "tiempo muerto".

Inciso b: instrucción nop

Es una operación nula. Sirve únicamente para realizar un ciclo de instrucción. Si en este programa no estuviese, al activar el delay slot se ejecutaría "halt", que es la instrucción siguiente al nop. Esto provocaría que se detenga la ejecución mientras se estaba ejecutando el primer salto, inutilizando al programa.

Inciso c: eficiencia

Activado el delay slot, se ejecutan 59 instrucciones en 63 ciclos (CPI = 1,068).
Sin esta técnica, se ejecutan 52 instrucciones en 63 ciclos (CPI = 1,212).

Ambos tardan lo mismo!	*NOP cuenta como instrucción*

Inciso d: instrucción útil

En lugar del nop, ejecutamos una instrucción del bucle no relacionada con el operando del salto (r2), por ejemplo el incremento de R1 (daddi). En este caso, con delay slot, se ejecutarían 51 instrucciones en 55 ciclos (CPI = 1,078).

8 instrucciones y ciclos menos!	*CPI aumenta (+0,01)*