

## Problema

Considere el siguiente bucle:

```
for (i=1; i<=10; i++) {  
    b[i]=a[i]*c;  
    c=c+1;  
    if (c>10) then goto etiqueta;  
}  
etiqueta:.....
```

Calcule la penalización debida a los saltos, en función del valor inicial de  $c$  (número entero), considerando que el procesador utiliza:

- a) Predicción fija (siempre se considera que se va a producir el salto)
- b) Predicción estática (si es desplazamiento es negativo, es decir, hacia atrás, se considera que se produce el salto; y si el desplazamiento es positivo, se predice como no efectivo).
- c) Predicción dinámica con 1 bit de historial para cada salto (1=saltar; 0=no saltar; valor inicial=1).

La penalización por saltos incorrectamente predichos es de 4 ciclos y para los saltos correctamente predichos, la penalización es 0.

## Solución

Los primero que hay que visualizar es que el código esconde dos saltos: el propio del bucle, determinado por la variable  $i$ , y el determinado por la variable  $c$ . Si se escribe en forma de pseudocódigo ensamblador se puede apreciar con claridad las dos instrucciones de salto:

```
        c=?;  
        i=1;  
inicio:  .....  
        c=c+1;  
        if (c>10) then goto etiqueta;    // Salto 1  
        i=i+1;  
        if (i<11) then goto inicio;      // Salto 2  
etiqueta: .....
```

El siguiente paso es analizar el valor de  $c$  que puede provocar la realización del salto 1. Dado que el bucle se ejecuta 10 veces, si inicialmente  $c$  es menor o igual que 0 entonces el salto 1 nunca se llega a ejecutar ya que su valor nunca llegará a ser 11 en las 10 iteraciones del bucle. A continuación, se realiza el análisis de los posibles casos.

### Penalización por predicción fija ( $c \leq 0$ ):

Salto 1: 40 ciclos. Dado que se predice que siempre se ejecuta y no sucede, hay 4 ciclos de penalización multiplicados por las 10 iteraciones del bucle.

Salto 2: 4 ciclos. Es efectivo las 9 primeras veces, con lo que solo hay discordancia en la predicción de la última iteración.

### **Penalización por predicción fija ( $c > 0$ ):**

Salto 1:  $(10-c)*4$  ciclos. En el momento en que  $c$  vale 11 el salto se hace efectivo y eso depende del valor inicial de  $c$  y del máximo de iteraciones posibles.

Salto 2: 0 ciclos. Nunca se produce discordancia dado que el salto, las veces que se ejecuta, siempre es efectivo.

### **Penalización por predicción estática ( $c \leq 0$ ):**

Salto 1: 0 ciclos. El salto siempre se predice como no efectivo dado que el desplazamiento es hacia delante y como nunca se produce, no hay discordancia en la predicción.

Salto 2: 4 ciclos. El salto siempre se predice como efectivo dado que el desplazamiento es hacia atrás y solo hay discordancia una vez que es en la última iteración.

### **Penalización por predicción estática ( $c > 0$ ):**

Salto 1: 4 ciclos. El salto siempre se predice como no efectivo dado que el desplazamiento es hacia delante y como solo se produce una vez, solo hay discordancia una vez.

Salto 2: 0 ciclos. El salto siempre se predice como efectivo dado que el desplazamiento es hacia atrás y como no se llega al número máximo de iteraciones al hacerse efectivo antes el salto 1, no hay discordancia.

### **Penalización por predicción dinámica ( $c \leq 0$ ):**

Salto 1: 4 ciclos. El contador inicialmente está a 1 (predicción de que se realiza) pero como el salto nunca es efectivo, la primera vez hay discordancia pero al decrementar el contador a 0 (al no ser efectivo el salto) ya siempre hay concordancia entre la efectividad del salto y la predicción.

Salto 2: 4 ciclos. El contador inicialmente está a 1 y como el salto es efectivo todas las veces salvo la última, siempre hay concordancia a excepción de la última iteración que el salto no se produce pero el contador estaba a 1.

### **Penalización por predicción dinámica ( $c > 0$ ):**

Salto 1: Surgen dos situaciones según el valor de  $c$ .

- c)  $10 \leq c$ . 0 ciclos. La primera vez el salto ya es efectivo ya que  $c$  pasa a ser mayor de 10 y hay concordancia con la predicción dado que el contador ya está a 1.
- ci)  $0 < c < 10$ . 8 ciclos. La primera vez el salto no es efectivo pero la predicción indica que sí por lo que hay discordancia y surgen 4 ciclos de penalización y dado que el salto real no ha sido efectivo, el contador pasa a 0. Con la salvedad de que  $c$  pase a valor 11, siempre hay concordancia entre salto y predicción (ambos no efectivos). Cuando  $c$  pasa a valer 11 entonces surge la discordancia: el salto real se produce pero la predicción es que no (contador a 0) por lo que aparecen 4 ciclos de penalización adicionales.

Salto 2: 0 ciclos. El contador inicialmente está a 1 y como el salto es efectivo todas las veces que se ejecuta (no se llega al final del bucle), siempre hay concordancia entre efectividad real y predicción.