

LAB. ARQUITECTURA DE COMPUTADORES: GUÍA DEL ALUMNO

Desenrollado de bucles

La técnica de desenrollado de bucles consiste en disminuir el número de iteraciones de un bucle con el propósito de “ahorrar instrucciones”, específicamente aquellas dedicadas al control del propio bucle, como actualización de índices y el propio salto. Para ello, el cómputo que se realiza en una iteración se repite una o más veces dentro del bucle, desmultiplicando el número de iteraciones necesarias. Por ejemplo, si en cada iteración se procesa un elemento de una serie de vectores, se pasa a procesar dos o mas elementos por iteración, disminuyendo el número de iteraciones necesarias para procesar la lista completa.

De este modo es posible disminuir el número de ciclos consumidos en la ejecución. Es frecuente que con el desenrollado sea más fácil reordenar instrucciones para conseguir minimizar paradas.

Modificar el hardware en DLXV3.1 para asignar a la suma y a la multiplicación una latencia total de 2 ciclos y comprobar que haya solo una unidad funcional de cada tipo.

Veamos esto mediante un ejemplo.. Puedes adaptarlo o/y partir del siguiente código: estúdialo, escríbelo y anota en un comentario las paradas etc. Luego simúlalo paso a paso. Comprueba los ciclos.

<pre> .data 100 cte: .double 3 n: .word 12 a: .double 1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12 b: .double 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 .text 0x1000 ini: lw r8,n(r0) ld f0,cte(r0) xor r1,r1,r1 loop: ld f2,a(r1) addd f4,f0,f2 sd b(r1),f4 addi r1,r1,#8 subi r8,r8,#1 bnez r8,loop nop trap #6 </pre>	<p>Hay 3 paradas (señala dónde al lado de las mismas), y no se ha usado el hueco de retardo. Y se emplean 124 ciclos.</p> <p>Reordenando y aprovechando el hueco de retardo se obtienen 76 ciclos, consiguiendo una acleración de $A = 124/76 = 1,63$</p> <pre> loop1: ld f2,a(r1) subi r8,r8,#1 addd f4,f0,f2 addi r1,r1,#8 bnez r8,loop 1 sd b-8(r1),f4 trap #6 </pre>
--	---

Este bucle se ejecuta 12 veces procesando una componente por vuelta.

Podemos procesar varias componentes por vuelta y disminuir en concordancia el número de vueltas. Un desenrollado de **factor 2** procesa dos componentes en cada vuelta, sabiendo que hemos de usar **una única instrucción de ajuste de índices** para ahorrar ciclos (de lo contrario no se justifica esta técnica de optimización).

Una recomendación: resulta más fácil hacer primero una versión no optimizada agrupando las instrucciones de carga, las de proceso y las ajuste de índices y contadores, y luego, identificar paradas y reordenar. Ambos pasos se muestran a continuación:

<pre> ini2: ld f0,cte(r0) xor r1,r1,r1 lw r8,n(r0) loop2: ld f2,a(r1) ld f6,a+8(r1) addd f4,f0,f2 addd f8,f0,f6 sd b(r1),f4 sd b+8(r1),f8 addi r1,r1,#16 subi r8,r8,#2 bnez r8,loop2 nop trap #6 </pre>	<p>Se consigue ejecutar sin paradas en 58 ciclos reordenando, con una Aceleración $A = 2,14$ y respecto al optimizado sin desenrollar 1,3</p> <pre> loop3: ld f2,a(r1) ld f6,a+8(r1) addd f4,f0,f2 subi r8,r8,#2 addd f8,f0,f6 sd b(r1),f4 sd b+8(r1),f8 bnez r8,loop3 addi r1,r1,#16 trap #6 </pre>
--	---

Es muy importante darse cuenta que el ahorro está en que se ejecutan menos saltos y menos instrucciones de ajuste de índices: esto lo organiza el compilador, los ingenieros de computación hacen estas cosas.

De un modo similar, podemos hacer un desenrollado de **factor 4**

<pre> ini4: ld f0,cte(r0) xor r1,r1,r1 lw r8,n(r0) loop4: ld f2,a(r1) ld f6,a+8(r1) ld f10,a+16(r1) ld f14,a+24(r1) addd f4,f0,f2 addd f8,f0,f6 addd f12,f0,f10 addd f16,f0,f14 sd b(r1),f4 sd b+8(r1),f8 sd b+16(r1),f12 sd b+24(r1),f16 addi r1,r1,#32 subi r8,r8,#4 bnez r8,loop4 nop trap #6 </pre>	<p>Se consigue ejecutar sin paradas en 49 ciclos reordenándolo, con una Aceleración $A = \underline{\hspace{1cm}}$ y respecto al optimizado sin desenrollar $A = \underline{\hspace{1cm}}$</p> <p>¡SUERTE!</p>
--	--

Ejercicios

Todos con las opciones de adelantamiento de operandos y salto retardado.

1. Para el programa del Producto Escalar de dos vectores de la práctica en coma flotante y para $N=32$ elementos, realiza una versión de desenrollado de factor 2 y optimízala para minimizar paradas. A continuación haz un desenrollado de factor 4 y optimízalo. Calcula las ganancias.
2. Utiliza el programa de $C(i) = a(i) \cdot cte + b(i)$ diseñado en la sesión anterior y haz lo mismo.
3. Altera la latencia de la multiplicación a 3 y/o 4 ciclos e intenta reordenar eliminar las paradas.