

ESTRUCTURA DE COMPUTADORES

PRÁCTICA Nº 3

DESENROLLADO DE BUCLES

Estas prácticas se basan en las desarrolladas por el Departamento de Informática Aplica de la Universidad Politécnica de Madrid (<http://www.dia.eui.upm.es>)

INTRODUCCIÓN

Este documento contiene el enunciado de la práctica de desenrollado de bucles sobre el simulador WinMIPS64.

OBJETIVO

El objetivo de la práctica es que el alumno aplique la técnica del desenrollado de bucles, conociendo sus ventajas y también sus limitaciones.

EL DESENROLLADO DE BUCLES

El desenrollado de bucles es una técnica que permite obtener un mayor rendimiento en el tiempo de ejecución de los programas. El mayor rendimiento se obtiene mediante el aprovechamiento del paralelismo que se puede obtener en los bucles de acceso a los elementos de un array.

Esta técnica, generalmente, la aplica el compilador realizando varias copias del código del bucle, además de utilizar algún registro adicional del procesador (renombrado de registros) para eliminar ciertos riesgos que puedan aparecer en el nuevo código.

El desenrollado no siempre es fácil aplicarlo debido a que en ciertas situaciones el número de veces que se ejecuta un bucle no es un valor estático sino que depende de una variable cuyo contenido no es conocido en tiempo de compilación. Otra dificultad añadida puede ser la no disponibilidad de registros para evitar ciertas dependencias de datos (RAW).

DESARROLLO DE LA PRÁCTICA

Para comprender mejor en qué consiste la técnica, partiremos del siguiente bucle de ejemplo:

```
FOR I := N DOWNT0 1 DO
  A[I] := A[I]+4;
END
```

Que en ensamblador quedaría como sigue:

```
LOOP:  daddi r1,r1,-4    ; Actualizar la variable índice
        lw  r10,0(r1)   ; Leer un elemento de un vector
        daddi r10,r10,4 ; Sumar 4 al elemento
        sw  r10,0(r1)   ; Escribir el nuevo valor
        bne r1,r0,LOOP  ; Fin de vector?
```

Existen tres dependencias de datos (RAW) que no permiten ninguna reordenación del código (por parte del compilador) para evitar las paradas que aparecerán en el cauce durante su ejecución.

El código completo podéis consultarlo en el fichero P3.s. Como podéis comprobar, en la declaración de datos (.data) para la variable array se han reservado 960 octetos correspondientes a los 240 elementos de tipo word32 (4 octetos).

Paso 1

Ejecutar el programa anterior en el simulador, activando los mecanismos de *Data Forwarding* y *Delay Slot* (menú *Configure*). Obsérvese que la instrucción que aparece a continuación de una de salto se ejecuta siempre, por eso en algunas partes del programa la instrucción NOP sucede a una de salto.

Obtener las informaciones siguientes:

RESULTADOS	
Ciclos	
Instrucciones	
CPI	
Riesgos RAW	
Tamaño del código	

Tabla 1

Paso 2

En una primera aproximación se va a desenrollar el bucle en **dos** copias, quedando de la forma siguiente:

```

LOOP:      daddi r1,r1,-8          ; Actualizar la variable índice
           lw  r10,4(r1)          ; Leer elemento vector
           daddi      r10,r10,4    ; Sumar 4 al elemento
           sw  r10,4(r1)          ; Escribir nuevo valor

           lw  r11,0(r1)          ; 2ª copia
           daddi      r11, r11, 4  ;
           sw  r11,0(r1)          ;

           bne r1,r0,LOOP        ; Fin de vector?

```

Con el desenrollado que se ha realizado el bucle necesitará ejecutarse únicamente la mitad de veces que el original.

Para poder evitar las dependencias de datos más adelante, se han empleado para cada copia registros distintos al original del bucle. También se han modificado los desplazamientos en las instrucciones de load y store para permitir el acceso a

los elementos anteriores al indicado por la variable índice del bucle (registro r1). Así mismo, la actualización de r1 se ha modificado, sustituyendo la constante -4 por -8 con el fin de reflejar el procesamiento de los dos elementos del array (cada elemento ocupa 4 octetos de memoria).

El código modificado se encuentra en el fichero p3-2.s

Ejecutar el nuevo programa obteniendo los nuevos resultados (Tabla 1)

Paso 3

Las dependencias de datos tipo RAW entre cada copia se mantienen (instrucciones lw, daddi y sw), para evitarlas puede reorganizarse el código de la forma siguiente:

```
LOOP:      daddi r1,r1,-8          ; Actualizar la variable índice

           lw      r10,4(r1)      ; Leer elemento vector
           lw      r11,0(r1)      ; 2ª copia

           daddi r10,r10, 4      ; Sumar 4 al elemento
           daddi r11, r11,4      ;

           sw      r10,4(r1)      ; Escribir nuevo valor
           sw      r11,0(r1)      ;

           bne r1,r0,LOOP        ; Fin de vector?
```

En el fichero de nombre p3.2opt.s podéis encontrar esta reorganización del código del Paso 2. Volved a ejecutar el programa y obtened los resultados de la Tabla 1.

Paso 4

Desenrollar el bucle LOOP en cuatro copias (fichero p3-4.s). Utilizar los registros **r12** y **r13** para las nuevas copias. Rellenar la tabla de resultados (Tabla 1) tras la ejecución

Paso 5

Reorganizar el código del paso 4 eliminando el mayor número de dependencias RAW. Guardar el resultado en el fichero p3-4opt.s y rellenar la tabla de resultados (Tabla 1).

Paso 6

Desenrollar el bucle LOOP en seis copias, utilizando los registros **r14** y **r15** para las nuevas copias (guardando el código en el fichero `p3-6.s`)

Paso 7

Reorganizar el código del paso 6 tratando de eliminar las dependencias RAW. Guardar el resultado en el fichero `p3-6opt.s`

Paso 8

Indicar la aceleración obtenida para cada versión del programa, con respecto al original (`p3.s`).

Versión de programa	Aceleración (SpeedUp)
P3-2.s	
P3-2opt.s	
P3-4.s	
P3-4opt.s	
P3-6.s	
P3-6opt.s	

HOJA DE RESPUESTAS

NOMBRE Y APELLIDOS

Paso 1

Ejecutar el programa anterior en el simulador obteniendo las informaciones siguientes:

RESULTADOS	
Ciclos	
Instrucciones	
CPI	
Riesgos RAW	
Tamaño del código	

Paso 2

Ejecutar el programa p3-2 . s

RESULTADOS	
Ciclos	
Instrucciones	
CPI	
Riesgos RAW	
Tamaño del código	

Paso 3

Ejecutar el programa `p3.2opt.s`

RESULTADOS	
Ciclos	
Instrucciones	
CPI	
Riesgos RAW	
Tamaño del código	

Paso 4

Ejecutar el programa `p3-4.s`

RESULTADOS	
Ciclos	
Instrucciones	
CPI	
Riesgos RAW	
Tamaño del código	

Paso 5

Ejecutar el programa p3-4opt.s

RESULTADOS	
Ciclos	
Instrucciones	
CPI	
Riesgos RAW	
Tamaño del código	

Paso 6

Ejecutar el programa p3-6.s

RESULTADOS	
Ciclos	
Instrucciones	
CPI	
Riesgos RAW	
Tamaño del código	

Paso 7

Ejecutar el programa `p3-6opt.s`

RESULTADOS	
Ciclos	
Instrucciones	
CPI	
Riesgos RAW	
Tamaño del código	

Paso 8

Indicar la aceleración obtenida para cada versión del programa, con respecto al original (`p3.s`).

Versión de programa	Aceleración (SpeedUp)
P3-2.s	
P3-2opt.s	
P3-4.s	
P3-4opt.s	
P3-6.s	
P3-6opt.s	