

Ejercicios II **(Ejercicios tema 2)**

Arquitectura de Computadores

Dora Blanco Heras
Área de Arquitectura de Computadores
CITIUS
Universidad de Santiago de Compostela

Ejercicio 1(resuelto en clase de teoría)

- Considere un procesador tipo MIPS con arquitectura segmentada (pipeline) que tiene dos bancos de registros separados uno para números enteros y otro para números en coma flotante. El banco de registros enteros dispone de 32 registros.
- El banco de registros de coma flotante dispone de 16 registros de doble precisión (\$f0, \$f2, . . . , \$f31).
- Se dispone de suficiente ancho de banda de captación y decodificación como para que no se generen detenciones por esta causa y que se puede iniciar la ejecución de una instrucción en cada ciclo, excepto en los casos de detenciones debidas a dependencias de datos.
- La instrucción bnez usa bifurcación retrasada con una ranura de retraso (delay slot).

Ejercicio (cont.)

- En esta máquina se desea ejecutar el siguiente código:

```
Bucle: ldc1 $f0, 0($t0)
      ldc1 $f2, 0($t1)
      add.d $f4, $f0, $f2
      mul.d $f4, $f4, $f6
      sdc1 $f4, ($t2)
      addi $t0, $t0, 8
      addi $t1, $t1, 8
      subi $t3, $t3, 1
      bnez $t3, bucle
      addi $t2, $t2, 8
```

- Inicialmente los valores de los registros son:

```
$t0: 0x00100000
$t1: 0x00140000
$t2: 0x00180000
$t3: 0x00000100
```

Ejercicio (cont.)

➤ Se pide:

- Enumere las dependencias de datos RAW que hay en el código anterior.
- Indique todas las detenciones que se producen al ejecutar una iteración del código anterior, e indique el número total de ciclos por iteración.
- Intente planificar el bucle para reducir el número de detenciones.
- Desenrolle el bucle de manera que en cada iteración se procesen cuatro posiciones de los arrays y determine el speedup conseguido. Utilice nombres de registros reales (\$f0, \$f2, . . . , \$f30).
- La siguiente tabla muestra las latencias adicionales de algunas categorías de instrucciones, en caso de que haya dependencia de datos. Si no hay dependencia de datos la latencia no es aplicable.

Cuadro 1: Latencias adicionales por instrucción

Instrucción	Latencia adicional	Operación
ldc1	+2	Carga un valor de 64 bits en un registro de coma flotante.
sdc1	+2	Almacena un valor de 64 bits en memoria principal.
add.d	+4	Suma registros de coma flotante de doble precisión.
mul.d	+6	Multiplica registros de coma flotante de doble precisión.
addi	+0	Suma un valor a un registro entero.
subi	+0	Resta un valor a un registro entero.
bnez	+1	Salta si el valor de un registro no es cero.

> Detenciones:

```
Bucle: ldc1 $f0, 0($t0)      #11
       ldc1 $f2, 0($t1)      #12
       <stall> x 2
       add.d $f4, $f0, $f2    #13
       <stall> x 4
       mul.d $f4, $f4, $f6    #14
       <stall> x 6
       sdc1 $f4, ($t2)        #15
       addi $t0, $t0, 8       #16
       addi $t1, $t1, 8       #17
       subi $t3, $t3, 1       #18
       bnez $t3, bucle        #19
       addi $t2, $t2, 8       #110
```

> Planificación del bucle:

```
Bucle: ldc1 $f0, ($t0)      #12
       ldc1 $f2, ($t1)      #11
       addi $t0, $t0, 8      #16
       addi $t1, $t1, 8      #17
       add.d $f4, $f0, $f2    #13
       subi $t3, $t3, 1      #18
       <stall> x 3
       mul.d $f4, $f4, $f6    #14
       <stall> x 6
       sdc1 $f4, ($t2)        #15
       bnez $t3, bucle        #19
       addi $t2, $t2, 8       #110
```

Al usar salto retardado con 1 ranura de retraso, no se produce parada al ejecutar bnez

Dependencias RAW:

```
$f0: 11 -> 13
$f2: 12 -> 13
$f4: 13 -> 14
$f4: 14 -> 15
$t3: 18 -> 19
```

¿Cuántos ciclos se requieren para ejecutar cada uno de los códigos?

22 y 19 ciclos respectivamente

Ejercicio (cont)

```
Bucle: ldc1 $f0, ($t0)
      ldc1 $f2, ($t1)
      ldc1 $f8, 8($t0)
      ldc1 $f10, 8($t1)
      ldc1 $f14, 16($t0)
      ldc1 $f16, 16($t1)
      ldc1 $f20, 24($t0)
      ldc1 $f22, 24($t1)
      add.d $f4, $f0, $f2
      add.d $f12, $f8, $f10
      add.d $f18, $f14, $f16
      add.d $f24, $f20, $f22
      <stall>
      mul.d $f4, $f4, $f6
      mul.d $f12, $f12, $f6
      mul.d $f18, $f18, $f6
      mul.d $f24, $f24, $f6
      addi $t0, 32
      addi $t1, 32
      subi $t3, 4
      sdc1 $f4, ($t2)
      sdc1 $f12, 8($t2)
      sdc1 $f18, 16($t2)
      sdc1 $f24, 24($t2)
      bnez $t3, bucle
      addi $t2, 32 → 1 ranura de retraso (slot) para el salto
```

Bucle desenrollado con factor 4:

¿Cuántos ciclos se requieren para las 4 iteraciones? 26 ciclos

¿Cuántos por iteración?
 $26/4 = 6,5$

Ejercicio 2 (será resuelto por el profesor en la sesión de prácticas dedicada al taller)

Sea el siguiente fragmento de código:

```
Bucle: lw $f0, 0( $r1 )  
      lw $f2, 0( $r2 )  
      mul.f $f4, $f0, $f2  
      add.d $f6, $f6, $f4  
      addi $r1, $r1, 4  
      addi $r2, $r2, 4  
      sub $r3, $r3, 1  
      bnez $r3, bucle
```

1. Haga una lista con todas las posibles dependencias de datos RAW y WAR. Para cada dependencia debe indicar, registro, instrucción de origen, instrucción de destino y tipo de dependencia.

Ejercicio 2 (cont)

2. Elabore un diagrama de tiempos para una arquitectura MIPS con un pipeline en 5 etapas, con las siguientes consideraciones:
 - No hay hardware de *forwarding* (adelantamiento).
 - Todas las instrucciones tardan 1 único ciclo en ejecutar EX si no hay paradas.
 - La arquitectura permite que una instrucción escriba en un registro y otra instrucción lea ese mismo registro en el mismo ciclo sin problemas.
 - Las bifurcaciones (saltos) se tratan vaciando el pipeline.
 - Las referencias a memoria requieren un ciclo de reloj.
 - La dirección efectiva de salto se calcula en la etapa ID pero no se sabe si se produce salto hasta la etapa EX.
3. Determine cuantos ciclos se necesitan para ejecutar N iteraciones.
4. Elabore un diagrama de tiempos para una arquitectura MIPS con un pipeline en 5 etapas como el del apartado 2 pero con las siguientes consideraciones:
 - Hay hardware completo de *forwarding*.
 - Asuma que las bifurcaciones se tratan prediciendo todos los saltos como tomados.
5. Determine cuantos ciclos se necesitan para ejecutar N iteraciones del bucle en las condiciones del apartado 4 y compare con el resultado del apartado 3.

Ejercicio 3

Consideremos el siguiente código donde cada instrucción tiene como coste asociado **un ciclo** además de los indicados en la tabla. Supongamos además que la máquina en la que se ejecuta el código es capaz de **emitir una instrucción por ciclo**, salvo las esperas debidas a detenciones y que es un procesador con un único camino de datos (**un único pipeline**).

Debido a **riesgos estructurales**, las **detenciones** de una instrucción (stalls) se realizan **siempre** y son las indicadas en la tabla que figura en la siguiente página bajo el nombre "latencia", independientemente de que **haya o no dependencia de datos con las instrucciones siguientes**. Inicialmente se tiene que $R1=0$, $R2=24$, $R3=16$.

Responder a las siguientes cuestiones:

1. **Calcular el número de ciclos** necesarios para ejecutar una iteración del bucle exterior y tres del bucle interior.
2. **Desenrollar tres iteraciones del bucle interior**, no desenrollar el bucle exterior y volver a realizar el cálculo anterior.
3. **Comparar** los resultados de ambos apartados y justificar la respuesta.

Ejercicio 3 (cont)

```
Loop1: LD F2 , 0(R1)
      ADDD F2, F0, F2
Loop2: LD F4 , 0 ( R3)
      MULTD F4 , F0 , F4
      DIVD F10 , F4 , F0
      ADDD F12 , F10 , F4
      ADDI R1 , R1 , #8
      SUB R18 , R2 , R1
      BNZ R18 , Loop2
      SD F2 , 0 ( R3)
      ADDI R3 , R3 , #8
      SUB R20 , R2 , R3
      BNZ R20 , Loop1
```

Instrucción	Latencia
LD	3
SD	1
ADD	2
MULTD	4
DIVD	10
ADDI	0
SUB	0
BNZ	1