

Tecnología de Computadores

Departamento de Arquitectura de Computadores

Relación de Problemas del Tema 3.

1. (H.P. 4.12) Supongamos que las etapas del camino de datos del procesador visto en clase tienen las siguientes latencias:

IF	ID	EX	MEM	WB
200ps	150ps	120ps	190ps	140ps

Se pide:

- ¿Cuál es el ciclo de reloj si el procesador se decide implementar en una versión monociclo? ¿Y si fuera segmentado?
- ¿Cuál es la latencia de una instrucción `lw` en un procesador segmentado y en uno monociclo?
- Si se divide una etapa del camino de datos segmentado en dos nuevas etapas, cada una con una latencia mitad de la etapa original, ¿qué etapa se debería dividir y cuál sería el nuevo ciclo de reloj del procesador?

2. (H.P. 4.13) Sean las siguientes secuencias de instrucciones:

<p>a. <code>lw \$1, 40(\$6)</code> <code>add \$6, \$2, \$2</code> <code>sw \$6, 50(\$1)</code></p>	<p>b. <code>lw \$5, -16(\$5)</code> <code>sw \$5, -16(\$5)</code> <code>add \$5, \$5, \$5</code></p>
---	---

- Indica las dependencias y su tipo.
- Indica los riesgos y añade instrucciones `nop` para resolverlos.

3. (H.P. 4.15) Sean las nuevas instrucciones:

<code>bezi (rt), desp</code>	<code>opec r0 rt desp</code>	If Mem[rt]=0 then PC \leftarrow PC+4+desp
<code>swi rd, rs(rt)</code>	<code>opec rs rt rd xxx</code>	Mem[rs+rt] \leftarrow rd

- ¿Qué cambios hay que hacer en el camino de datos para añadir estas instrucciones a la ISA del MIPS?
- ¿Qué nuevas señales de control deben añadirse al diseño?
- ¿Qué tipo de riesgos se pueden producir con estas instrucciones?

4. (H.P. 4.16) Para cada una de las siguientes instrucciones:

a. `lw $1, 40($6)`
b. `add $5, $5, $5`

- a) Identifica qué se almacena en cada uno de los registros situados entre dos etapas del pipeline.
- b) ¿Qué registros necesitan leerse y cuáles se leen realmente?
- c) ¿Qué hace la instrucción en las etapas EX y MEM?

5. Sea el programa **suma1** dado por el siguiente código MIPS:

```

sub    $5, $0, $0
suma: lw    $10, 1000($20)
add    $5, $5, $10
addi   $20, $20, -4
bne    $20, $0, suma

```

Se pide:

- a) Describir brevemente la tarea realizada por **suma1**.
- b) Detectar las dependencias que afectan a **suma1** en el MIPS segmentado en 5 etapas y clasificarlas en dependencias de datos y dependencias de control.
- c) Gestión de dependencias por parte del compilador:
 - c.1 Suponer un MIPS segmentado sin ningún tipo de soporte hardware para solventar los problemas derivados de los riesgos de datos que conlleva la segmentación. Reescribir el código reordenándolo e insertando el el mínimo número de códigos de no-operación (nop) para producir una nueva versión, **suma2**, que pueda ejecutarse correctamente en este MIPS.
 - c.2 Evaluar la mejora obtenida hasta ahora con respecto al punto de partida. Para ello, comparar el tiempo de ejecución que ofrece **suma2** al ejecutarse sobre el MIPS anterior con respecto al de **suma1** sobre el MIPS sin segmentar de la implementación monociclo. Suponer un número N de iteraciones en ambos programas y que las latencias de las etapas del camino de datos del procesador son las vistas en clase:

IF	ID	EX	MEM	WB
40ps	20ps	40ps	40ps	20ps

6. Sea una arquitectura RISC con un conjunto de instrucciones similar al del procesador MIPS visto en clase. Su camino de datos presenta una segmentación en 3 etapas y una única memoria, común para instrucciones y datos. Esto hace que las instrucciones se ejecuten según se muestra a continuación:

INST	1	2	3	4	5
Inst1	IFD	REX	MEW		
Inst2		IFD	REX	MEW	
Inst3			IFD	REX	MEW

donde:

- IFD es la etapa de búsqueda y decodificación de instrucción.
- REX es la etapa de búsqueda de operandos (lectura del banco de registros), ejecución de operación ó resolución de condiciones de salto y cálculo de dirección para operando destino ó salto.
- MEW es la etapa de acceso a memoria para instrucciones de carga/almacenamiento, escritura de resultados en registros para instrucciones aritmético/lógicas y de carga y actualización del PC para instrucciones de salto.

Además no podrá leerse en un mismo ciclo un registro que va a ser escrito en dicho ciclo (es decir, no hay anticipación en el banco de registros).

Bajo estas condiciones, responder a las siguientes cuestiones:

- ¿Qué tipo de riesgos pueden darse? Poner ejemplos.
- ¿Qué ocurriría con las instrucciones de salto?
- Representar en un diagrama de ciclos la evolución del cauce para el siguiente trozo de código e identifica los riesgos que puedan darse.

```

sw $5, 3000($7)
or $5, $4, $5
sub $8, $4, $7
lw $4, 3000($8)
bne $7, $0, etiq
add $8, $8, $5
etiq: sw $5, 3000($8)

```

7. Sea el siguiente código MIPS, que a partir de ahora referenciaremos como `mi_prog`:

```

ori $2, $0, 1000h
loop: lw $1, 2800h($2)
sub $4, $1, $0
jal rotar
sw $7, 7800h($2)
sw $1, C800h($2)
subi $2, $2, 4
bne $2, $0, loop

rotar: add $10, $4, $4
mul $7, $10, 2
jr $31

```

Señalar las instrucciones que **pueden producir riesgos** en `mi_prog` para un MIPS segmentado con anticipación en el banco de registros. Responder en las siguientes tablas:

Riesgos de datos		Riesgos de control	
Instrucción(es) (dar mnemotécnicos)	Registro(s) involucrado(s)	Instrucción(es) (dar mnemotécnicos)	