

## ESTRUCTURA DE COMPUTADORES

### Ejercicios Tema 2: Segmentación básica.

#### PROBLEMAS RESUELTOS

1. El programa BROT tarda 34 segundos en ejecutarse en un computador que dispone de un procesador con un CPI medio de 2,5 y un reloj a 800 MHz.

- a. Cuántas instrucciones máquina ejecuta este programa?

*El número de instrucciones  $I$  se calcula por medio de la ecuación que expresa el tiempo de ejecución  $T$  de un programa como el producto del número de instrucciones  $N$ , el número medio de ciclos por instrucción CPI y el periodo del reloj  $t_c$ . Por lo tanto, resolvemos la ecuación:*

$$34 = I \times 2,5 \times \frac{1}{800 \times 10^6}$$

*Luego el número de Instrucciones  $I = 10880 \times 10^6$*

- b. Si quisiéramos reducir el tiempo de ejecución por debajo de los 30 segundos, justifique cuál de las dos siguientes opciones sería la más apropiada: aumentar la frecuencia de reloj en 50 MHz o cambiar el procesador por otro compatible (con el mismo juego de instrucciones) que trabaja a la misma frecuencia con un CPI medio de 2,1.

*Si gastamos la expresión anterior, la primera opción conseguiría un tiempo de ejecución de:*

$$10880 \times 10^6 \times 2,5 \times \frac{1}{850 \times 10^6} = 32 \text{ s}$$

*Mientras que la segunda*

$$10880 \times 10^6 \times 2,1 \times \frac{1}{850 \times 10^6} = 28,56 \text{ s}$$

*Luego, esta última sería la opción que consigue un tiempo de ejecución por debajo de los 30 segundos*

2. Considera la siguiente secuencia de instrucciones,

etiqueta :	lw \$3, 100(\$4)	(1)
	or \$6, \$5, \$4	(2)
	add \$4, \$6, \$3	(3)
	beq \$5, \$4, etiqueta	(4)
	add \$1, \$4, \$8	(5)
	and \$6, \$3, \$5	(6)
	sw \$5, 100(\$6)	(7)

- a. Rellena la siguiente tabla para indicar los riesgos por dependencia de datos que se encuentran en este fragmento de código.

(En las casillas “Se escribe en” y “Se lee en” indica el número de instrucción, que se encuentra a la derecha de las instrucciones entre paréntesis, en la cual se escribe o lee el registro.)

	Registro	Se escribe en	Se lee en
Riesgo 1	\$3	(1)	(3)
Riesgo 2	\$6	(2)	(3)
Riesgo 3	\$4	(3)	(4)
Riesgo 4	\$4	(3)	(5)
Riesgo 5	\$6	(6)	(7)

- b. Realiza el diagrama **instrucciones/tiempo** considerando que la instrucción de salto **NO** realiza el salto.

*Empleo de ciclos de espera para resolver los conflictos de datos y el de control (latencia 2)*

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
etiqueta lw \$3, 100(\$4)	LI	DI	EX	M	ER											
or \$6, \$5, \$4		LI	DI	EX	M	ER										
add \$4, \$6, \$3			LI	•	•	DI	EX	M	ER							
beq \$5, \$4, etiqueta						LI	•	•	DI	EX	M	ER				
add \$1, \$4, \$8								•	•	LI	DI	EX	M	ER		
and \$6, \$3, \$5											LI	DI	EX	M	ER	
sw \$5, 100(\$6)												LI	•	•	DI	EX

3. Considera ahora que el programa anterior se ha ejecutado en la ruta de datos segmentada de cinco etapas (LI, DI, EX, M, ER) estudiada en clase y supón que las etapas tienen un retardo de 90, 40, 70, 120, 55 ns respectivamente y que el tiempo necesario para escribir en los registros de segmentación es de 5ns.

- a. ¿Cuál es la frecuencia de la señal de reloj a emplear en esta ruta de datos? Justifica la respuesta.

*La frecuencia de reloj es la inversa del periodo de reloj y éste ha de ser igual al retardo de la etapa más lenta más el retardo de los registros.*

*Luego,  $T = 40 + 10 = 50 \text{ ns}$ ,*

$$\text{Frecuencia} = \frac{1}{125 \text{ ns}} = 8 \text{ Mhz}$$

- b. ¿Cuál es la productividad máxima que se puede alcanzar con esta ruta de datos? Justifique la respuesta.

$$X_{\max} = \frac{1 \text{ instrucción}}{125 \text{ ns}} = 8 \text{ MIPS}$$

*La productividad máxima se produce cuando se ejecuta una instrucción cada ciclo de reloj*

- c. ¿Cuál es la productividad conseguida realmente al ejecutar el programa del **ejercicio 2**, utilizando para ello el resultado del apartado **1.b**? Justifique la respuesta.

$$X_{real} = \frac{7 \text{ instrucciones}}{19 \text{ ciclos} \times 125 \text{ ns}} = 2,95 \text{ MIPS}$$

*Se ejecutan 7 instrucciones en 19 ciclos. Cada ciclo dura 125 ns*

- d. ¿Cuál es la aceleración máxima que se puede alcanzar? Justifica la respuesta.

$$S_{max} = \frac{T_{ns}}{T_s} = \frac{375 \text{ ns}}{125 \text{ ns}} = 3$$

*La aceleración máxima es el ratio entre el tiempo de ciclo del procesador monociclo sin segmentar y el del procesador segmentado. Podría ser hasta 3 veces más rápido*

4. Se dispone de un procesador que ha sido segmentado en 5 etapas cuyos retardos son 40ns, 25ns, 20ns, 40ns y 25ns. Asumiendo que el retardo de los registros de segmentación es de 10ns, se pide:

- a. Calcular la frecuencia de reloj del procesador segmentado.

*La frecuencia de reloj es la inversa del periodo de reloj y éste ha de ser igual al retardo de la etapa más lenta más el retardo de los registros.*

$$\text{Luego, } T = 40 + 10 = 50 \text{ ns, Frecuencia} = \frac{1}{50} = 20 \text{ Mhz}$$

- b. Productividad máxima del procesador segmentado.

*La productividad máxima será ejecutar una instrucción por ciclo de reloj.*

$$\text{Por tanto, Productividad}_{max} = \frac{1}{50 \text{ ns}} = 20 \text{ MIPS}$$

- c. Calcular la aceleración (speedup) del procesador segmentado respecto al no segmentado ¿Cuál sería la aceleración ideal?

*La aceleración máxima es el ratio entre el tiempo de ciclo del procesador monociclo sin segmentar y el del procesador segmentado.*

$$S_{max} = \frac{T_{ns}}{T_s} = \frac{40 + 25 + 20 + 40 + 25 \text{ ns}}{50 \text{ ns}} = 3$$

*La aceleración ideal es 5. EL procesador segmentado podría ser hasta 5 veces más rápido que el no segmentado. 5 es el número de etapas (K=5)*

- d. ¿Cómo se podría incrementar la productividad máxima del procesador segmentado? ¿Qué factores pueden impedir que se alcance la productividad máxima?

La productividad se puede incrementar reduciendo el tiempo de ciclo y gestionando eficientemente los conflictos y riesgos que aparecen en el código. Los conflictos y riesgos son un impedimento para conseguir una alta productividad.

- e. Suponiendo que se dispone de un procesador superescalar de 4 vías cuyos cauces segmentados son similares al especificado en el enunciado, ¿qué tiempo se requeriría para ejecutar 8000 instrucciones en dicho procesador?

Al tener 4 vías podríamos ejecutar 4 instrucciones en paralelo. Por tanto el tiempo de ejecutar 8000 instrucciones sería similar al de ejecutar 2000. Si la existencia de conflictos y riesgos el tiempo para ejecutar 2000 instrucciones sería:

$$T = (n + K - 1)T = (2000 + 5 - 1)50 = 99,800\mu s$$

5. Los siguientes fragmentos de código contienen conflictos y/o riesgos cuando se ejecutan en un procesador segmentado como el visto en clase. Completa el cronograma tiempo-etapas e indica claramente donde se producen y aportando la solución adecuada.

- a.
- ```
add $1,$2,$3
or $3,$2,$1
```

|                 | 1  | 2  | 3  | 4 | 5  | 6  | 7 | 8 | 9 | 10 | Conflicto/Riesgo                                                       |
|-----------------|----|----|----|---|----|----|---|---|---|----|------------------------------------------------------------------------|
| add \$1,\$2,\$3 | BI | DI | EX | M | EX |    |   |   |   |    | La instrucción add escribe en el \$1 y la instrucción siguiente lo lee |
| or \$3,\$2,\$1  |    | BI | •  | • | DI | EX | M |   |   |    |                                                                        |

La solución consiste en introducir dos ciclos de espera entre ambas instrucciones. Otra posible solución hubiera sido insertar dos NOP entre ambas

- b.
- ```
lw $6,$0($7)
beq $7,$8,L
and $1,$2,$3
L: or $7,$6,$8
```

Considera que SI que se produce el salto y la latencia del mismo es 3.

	1	2	3	4	5	6	7	8	9	10	Conflicto/Riesgo
lw \$6,\$0(\$7)	BI	DI	EX	M	EX						Si se produce el salto existe una dependencia de datos en \$6 ya que lo escribe la instrucción load y lo lee la instrucción or. Sin embargo al tener que solucionar el riesgo de control generado por la instrucción beq mediante la inserción de 3 instrucciones NOP, esta dependencia desaparece. También se podrían haber insertado 3 ciclos de espera después de la instrucción beq
beq \$7,\$8,L		BI	DI	EX	M	EX					
NOP			BI	DI	EX	M	EX				
NOP				BI	DI	EX	M	EX			
NOP					BI	DI	EX	M	EX		
and \$1,\$2,\$3											
L:or \$7,\$6,\$8						BI	DI	EX	M	EX	

c.

```
lw $1,100($2)
and $2,$1,$0
```

	1	2	3	4	5	6	7	8	9	10	Conflicto/Riesgo
lw \$1,100(\$2)	BI	DI	EX	M	ER						La instrucción lw actualiza \$1 y la instrucción siguiente lo lee
and \$2,\$1,\$0		BI	•	•	DI	EX	M	ER			

*La solución consiste en introducir dos ciclos de espera entre ambas instrucciones. Otra posible solución hubiera sido insertar dos NOP entre ambas*

d.

```
sub $4,$3,$2
sw $3,100($2)
```

	1	2	3	4	5	6	7	8	9	10	Conflicto/Riesgo
sub \$4,\$3,\$2	BI	DI	EX	M	ER						No existe
sw \$3,100(\$2)		BI	DI	EX	M	ER					

6. Considera la siguiente secuencia de instrucciones,

```

                lw $1, 100($2)      (1)
                beq $1, $2, etiqueta (2)
                and $3, $1, $4      (3)
etiqueta:      lw $1, 50($3)       (4)
                sw $1, 100($2)      (5)
                sub $5, $1, $2      (6)
                lw $1, 10($5)       (7)
```

a. Rellena la siguiente tabla para indicar los riesgos por dependencia de datos que se encuentran en este fragmento de código.

	Registro	Se escribe en	Se lee en	
Riesgo 1	\$1	(1)	(2)	Genera conflicto
Riesgo 2	\$1	(1)	(3)	Se resolverá al resolver el anterior
Riesgo 3	\$3	(3)	(4)	Hay conflicto si no se salta
Riesgo 4	\$1	(4)	(5)	Genera conflicto
Riesgo 5	\$1	(4)	(6)	Se resolverá al resolver el anterior
Riesgo 4	\$5	(6)	(7)	Genera conflicto

b. Realiza el diagrama **instrucciones/tiempo** considerando que la instrucción de salto **SI** realiza el salto. En este caso asume la resolución de conflictos de control por medio de la técnica *predict not taken* y latencia de salto de tres ciclos. Los conflictos de datos se resolverán mediante inserción de ciclos de espera.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
lw \$1, 100(\$2)	BI	DI	EX	M	ER											
beq \$1, \$2, etiqueta		BI	•	•	DI	EX	M	ER								
and \$3, \$1, \$4					BI	DI	EX									
etiqueta lw \$1, 50(\$3)					BI	DI	•									
sw \$1, 100(\$2)																
etiqueta lw \$1, 50(\$3)								BI	DI	EX	M	ER				
sw \$1, 100(\$2)								BI	•	•	DI	EX	M	ER		

Nota. Aunque la latencia de salto es 3 la instrucción `sw` no llegaría a entrar al haberse detenido el avance de las instrucciones en el ciclo 7 debido a la dependencia de datos detectada (\$3)

7. Considerando la ruta de datos de cinco etapas vista en clase, y suponiendo que la duración de las etapas es  $LI=60ns$ ,  $DI=40ns$ ,  $EX=50ns$ ,  $M=70ns$ ,  $ER=45ns$ , y un retardo por los registros de segmentación de  $5ns$ , se pide:

- a. ¿Cuál es el tiempo de ciclo para este procesador?

*El tiempo de ciclo ha de ser igual al retardo de la etapa más lenta más el retardo de los registros, luego:*

$$T = 70 + 5 = 75 ns$$

- b. Suponiendo que se ejecuta este fragmento de código identifica los riesgos existentes.

```

(1)      sw    $2,0($3)
(2)      sw    $3,0($2)
(3)      beq   $2,$3,etiqueta
(4)      add   $4,$1,$2
(5)      lw    $5,100($4)
(6)      sw    $5,50($4)
etiqueta : (8)      or    $1,$5,$3

```

	Registro	Se escribe en	Se lee en
Riesgo 1	\$4	(4)	(5)
Riesgo 2	\$5	(5)	(6)

- c. Rellena el diagrama instrucciones/tiempo correspondiente a la ejecución del siguiente código, suponiendo que el salto NO tiene lugar y solucionando las dependencias de datos que se producen. Considera predicción de salto no efectivo para la resolución del riesgo de control.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
sw \$2,0(\$3)	BI	DI	EX	M	ER											
sw \$3,0(\$2)		BI	DI	EX	M	ER										
beq \$2,\$3, etiqueta			BI	DI	EX	M	ER									
add \$4,\$1,\$2				BI	DI	EX	M	ER								
lw \$5,100(\$4)					BI	•	•	DI	EX	M	ER					
sw \$5,50(\$4)								BI	•	•	DI	EX	M	ER		
etiqueta or \$1,\$5,\$3											BI	DI	EX	M	ER	

d. ¿Cuál es la productividad conseguida con la ejecución de dicho código?

$$Productividad = \frac{7}{15 \text{ ciclos} \times 75 \text{ ns}} = 6,222 \text{ MIPS}$$

e. ¿Cuál es la máxima productividad alcanzable en este procesador?

$$Productividad_{max} = \frac{1}{75 \text{ ns}} = 13,333 \text{ MIPS}$$

## PROBLEMAS PROPUESTOS

8. Se dispone de un sistema digital el cual se ha procedido a segmentar en 4 etapas secuenciales sin solapamiento entre ellas, y cuyos retardos son 33ns, 35ns, 22ns, 30ns. Se pide:
  - a. Retardo máximo permitido para los registros de segmento a fin garantizar una productividad máxima de 20 MOPS. Justificar la respuesta.
  - b. En el caso anterior, calcúla la aceleración (speed-up) máxima que se podría alcanzar. Justificar la respuesta.
  - c. ¿Qué factores pueden impedir que se alcance la productividad máxima?
  - d. ¿Qué factores pueden impedir que se alcance la aceleración ideal?
  
9. Considéra la ruta de datos segmentada de cinco etapas (LI, DI, EX, M, ER) estudiada en clase y supón que las etapas tienen un retardo de 60, 45, 50, 80, 30 ns respectivamente y que el tiempo necesario para escribir en los registros de segmentación es de 5ns.
  - a. ¿Cuál es la productividad máxima que se puede alcanzar con esta ruta de datos segmentada?
  - b. ¿Cuál es la aceleración máxima conseguida respecto a la versión no segmentada de esta ruta de datos?

- c. ¿Cuál es la aceleración ideal que teóricamente se puede conseguir con un circuito segmentado en cinco etapas?
- d. ¿Qué dos condiciones debería cumplir nuestro circuito segmentado para poder alcanzar la aceleración ideal?

10. Considera ahora la siguiente secuencia de programa,

```
lw $1, 50($4)
beq $1, $2, etiqueta
or $3, $5, $6
sw $3, 50($4)
add $4, $4, $1
sw $4, 100($6)
etiqueta : and $3, $5, $6
sw $3, 100($4)
```

- a. Identifica claramente todos los riesgos que aparecen. Señala los riesgos por dependencias de datos en el código para clarificar la respuesta.
  - b. Realiza el diagrama **instrucciones/tiempo** considerando que la instrucción de salto **SI** realiza el salto, utiliza la predicción de salto no efectiva como técnica de resolución de este tipo de riesgo. Compara con la utilización de predicción de salto efectiva para ese mismo caso. ¿Qué conclusiones se obtienen?
11. Un cierto procesador se ha segmentado en 5 etapas, cuyos retardos son los siguientes: 25ns, 15ns, 30ns, 25ns y 25ns. Suponiendo que el retardo de los registros de segmentación es de 10ns, se pide:
- a. Frecuencia de reloj a la que trabajará el procesador segmentado y productividad máxima que podrá alcanzar.
  - b. Frecuencia de reloj a la que trabajaría el procesador NO-segmentado y productividad que alcanzaría.
  - c. Tiempo de retardo para el procesamiento de 10 instrucciones
  - d. Aceleración máxima que podrá alcanzar el circuito segmentado
  - e. ¿Cuál sería la aceleración ideal? ¿Qué se podría hacer para intentar aproximar la aceleración máxima a la aceleración ideal?
12. Considera la siguiente secuencia de programa del MIPS R2000, el cual se ejecuta en cierta ruta de datos que se halla segmentada en cinco etapas (LI, DI, EX, M, ER):

```
sw $1,50($3)
lw $2,50($3)
beq $2,$5,etiqueta
sub $5,$1,$4
etiqueta: and $6,$5,$1
sw $6,50($3)
```



Se pide:

- a. ¿Qué tipo de riesgos aparecen y dónde?
  - b. ¿Cómo pueden solucionarse?
  - c. Diagrama instrucciones tiempo indicando los mecanismos de solución empleados para resolver los conflictos de dependencias de datos. Considera que los saltos tienen latencia 3 y se resuelven mediante inserción de `nop`. Para el caso de la ejecución del fragmento dado supón que el salto SI se produce.
  - d. ¿Cuál ha sido en este caso el tiempo de ejecución del fragmento de código?
13. Considera la siguiente secuencia de programa en ensamblador de MIPS R2000, el cual se ejecuta en cierta ruta de datos que se halla segmentada en cinco etapas (LI, DI, EX, M, ER):

```
lw $2,0($4)
beq $2,$3, etiqueta
add $1,$2,$3
sub $4,$1,$3
lw $5,100($4)
sw $5,50($4)
...
etiqueta : ...
```

- a. Qué tipo de riesgos aparecen, dónde aparecen y cómo pueden solucionarse.
- b. Diagrama instrucciones/tiempo suponiendo que el salto NO tiene lugar y considerando que los conflictos de control se resuelven mediante inserción de `nop` y que la latencia de salto es 2.
- c. Diagrama instrucciones/tiempo suponiendo que el salto NO tiene lugar y considerando que los conflictos de control se resuelven mediante predicción de salto NO efectivo.
- d. Diagrama instrucciones/tiempo suponiendo que el salto NO tiene lugar y considerando que los conflictos de control se resuelven mediante predicción de salto efectivo.

