

## BOLETÍN DE PROBLEMAS

### TEMA 4. EL PARALELISMO INTERNO EN LOS SISTEMAS COMPUTADORES

Etapa	Instrucción ALU	Carga/Almacenamiento	Salto/Bifurcación
IF	$IR \leftarrow M[PC]$ $PC \leftarrow PC+4$	$IR \leftarrow M[PC]$ $PC \leftarrow PC+4$	$IR \leftarrow M[PC]$ $PC \leftarrow PC+4$
ID	$A \leftarrow Rs1; B \leftarrow Rs2$ $PC1 \leftarrow PC$ $IR1 \leftarrow IR$	$A \leftarrow Rs1; B \leftarrow Rs2$ $PC1 \leftarrow PC$ $IR1 \leftarrow IR$	$A \leftarrow Rs1; B \leftarrow Rs2$ $PC1 \leftarrow PC$ $IR1 \leftarrow IR$
EX	$ALU_{output} \leftarrow A \text{ op } (B \text{ or } (IR1_{16})^{16} \text{##} IR1_{16..31})$	$DMAR \leftarrow A + (IR1_{16})^{16} \text{##} IR1_{16..31}$ $SMDR \leftarrow B$	$ALU_{output} \leftarrow PC1 + (IR1_{16})^{16} \text{##} IR1_{16..31}$ } Salto $Cond \leftarrow (Rs1 \text{ op } 0);$  $ALU_{output} \leftarrow PC1 + (IR1_{26})^6 \text{##} IR1_{6..31})$ } Bifurc.
MEM	$ALU_{output1} \leftarrow ALU_{output}$	$LMDR \leftarrow M[DMAR]$ ó $M[DMAR] \leftarrow SMDR$	If (cond): $PC \leftarrow ALU_{output}$
WB	$Rd \leftarrow ALU_{output1}$	$Rd \leftarrow LMDR$	

**Ejercicio 1.** El siguiente fragmento de código se ejecuta en un DLX con segmentación:

```
SUB R1,R2,R3
ADD R4,R5,R6
SUB R5,R4,R8
ADD R7,R2,R3
ADD R9,R7,R3
LW R1,10(R6)
ADD R3,R1,R4
SUB R6,R7,R8
```

- Indicar el estado de cada instrucción durante los ciclos de ejecución del código, suponiendo DLX segmentado con memoria caché partida y sin bypass. Calcular el número de ciclos necesarios para ejecutar este código. Indicar si existen bloqueos en la cadena y a qué se deben.
- Repetir el primer apartado para una versión segmentada del DLX que incluye adelantamiento (bypass) generalizado. Indicar en qué casos se utiliza el adelantamiento o bypass.
- ¿Es posible, para la segmentación propuesta en el apartado b), mejorar el rendimiento del conjunto de instrucciones referido para la versión de DLX que incluye adelantamiento generalizado?. En caso positivo indicar detalladamente cómo. Para la solución propuesta, indicar en qué casos se utiliza el adelantamiento o bypass.

## SOLUCIÓN EJERCICIO 1

a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SUB R1, R2, R3	IF	ID	EX	MEM	WB													
ADD R4, R5, R6		IF	ID	EX	MEM	WB												
SUB R5, R4, R8			IF	Det	Det	ID	EX	MEM	WB									
ADD R7, R2, R3						IF	ID	EX	MEM	WB								
ADD R9, R7, R3							IF	Det	Det	ID	EX	MEM	WB					
LW R1, 10(R6)										IF	ID	EX	MEM	WB				
ADD R3, R1, R4											IF	Det	Det	ID	EX	MEM	WB	
SUB R6, R7, R8														IF	ID	EX	MEM	WB

Número de ciclos: 18.

Ciclos de bloqueo: 6.

**Bloqueo1:** Riesgo de Datos RAW. Debido al registro R4.**Bloqueo2:** Riesgo de Datos RAW. Debido al registro R7.**Bloqueo3:** Riesgo de Datos RAW. Debido al registro R1.

b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SUB R1, R2, R3	IF	ID	EX	MEM	WB													
ADD R4, R5, R6		IF	ID	EX	MEM	WB												
SUB R5, R4, R8			IF	ID	EX	MEM	WB											
ADD R7, R2, R3				IF	ID	EX	MEM	WB										
ADD R9, R7, R3					IF	ID	EX	MEM	WB									
LW R1, 10(R6)						IF	ID	EX	MEM	WB								
ADD R3, R1, R4							IF	ID	Det	EX	MEM	WB						
SUB R6, R7, R8								IF	Det	ID	EX	MEM	WB					

Número de ciclos: 13.

Ciclos de bloqueo: 1.

**Adelantamiento1:** Debido al registro R4. Indicado en rojo. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Adelantamiento2:** Debido al registro R7. Indicado en cyan. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Adelantamiento3:** Debido al registro R1. Indicado en verde. El registro de datos de memoria MDR se conecta directamente con la entrada de la ALU. Se produce un ciclo de bloqueo (**Bloqueo1**).

c) Sí se puede mejorar el rendimiento. Se puede eliminar el único bloqueo cambiando de posición la última instrucción.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SUB R1, R2, R3	IF	ID	EX	MEM	WB													
ADD R4, R5, R6		IF	ID	EX	MEM	WB												
SUB R5, R4, R8			IF	ID	EX	MEM	WB											
ADD R7, R2, R3				IF	ID	EX	MEM	WB										
ADD R9, R7, R3					IF	ID	EX	MEM	WB									
LW R1, 10(R6)						IF	ID	EX	MEM	WB								
SUB R6, R7, R8							IF	ID	EX	MEM	WB							
ADD R3, R1, R4								IF	ID	EX	MEM	WB						

Número de ciclos: **12**.

Ciclos de bloqueo: **0**.

**Adelantamiento1:** Debido al registro R4. Indicado en rojo. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Adelantamiento2:** Debido al registro R7. Indicado en cyan. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Adelantamiento3:** Debido al registro R1. Indicado en verde. El registro de datos de memoria MDR se conecta directamente con la entrada de la ALU. Ahora no se produce un ciclo de bloqueo.

**Ejercicio 2.** El siguiente fragmento de código se ejecuta en un DLX con segmentación:

```
SUB R3,R2,R2
ADD R3,R2,R6
SUB R5,R3,R8
ADD R7,R2,R5
LW R1,10(R6)
SW 10(R1),R3
ADD R3,R1,R4
SUB R6,R7,R8
```

- Indicar el estado de cada instrucción durante los ciclos de ejecución del código, suponiendo DLX segmentado con memoria caché partida y sin bypass. Calcular el número de ciclos necesarios para ejecutar este código. Indicar si existen bloqueos en la cadena y a qué se deben.
- Repetir el primer apartado para una versión segmentada del DLX que incluye adelantamiento (bypass) generalizado. Indicar en qué casos se utiliza el adelantamiento o bypass.
- ¿Es posible mejorar el rendimiento del conjunto de instrucciones referido para la versión de DLX con adelantamiento generalizado?. En caso positivo indicar detalladamente cómo. Indicar en qué casos se utiliza el adelantamiento o bypass.
- Repetir el ejercicio teniendo en cuenta que la caché de memoria principal utilizada está unificada, esto es, no existen cachés separadas de datos e instrucciones.

## SOLUCIÓN EJERCICIO 2

a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SUB R3,R2,R2	IF	ID	EX	MEM	WB													
ADD R3,R2,R6		IF	ID	EX	MEM	WB												
SUB R5,R3,R8			IF	Det	Det	ID	EX	MEM	WB									
ADD R7,R2,R5						IF	Det	Det	ID	EX	MEM	WB						
LW R1,10(R6)									IF	ID	EX	MEM	WB					
SW 10(R1),R3										IF	Det	Det	ID	EX	MEM	WB		
ADD R3,R1,R4													IF	ID	EX	MEM	WB	
SUB R6,R7,R8														IF	ID	EX	MEM	WB

Número de ciclos: **18**.Ciclos de bloqueo: **6**.**Bloqueo1:** Riesgo de Datos RAW. Debido al registro R3.**Bloqueo2:** Riesgo de Datos RAW. Debido al registro R5.**Bloqueo3:** Riesgo de Datos RAW. Debido al registro R1.

b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SUB R3,R2,R2	IF	ID	EX	MEM	WB													
ADD R3,R2,R6		IF	ID	EX	MEM	WB												
SUB R5,R3,R8			IF	ID	EX	MEM	WB											
ADD R7,R2,R5				IF	ID	EX	MEM	WB										
LW R1,10(R6)					IF	ID	EX	MEM	WB									
SW 10(R1),R3						IF	ID	Det	EX	MEM	WB							
ADD R3,R1,R4							IF	Det	ID	EX	MEM	WB						
SUB R6,R7,R8									IF	ID	EX	MEM	WB					

Número de ciclos: **13**.Ciclos de bloqueo: **1**.

**Adelantamiento1:** Debido al registro R3. Indicado en rojo. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Adelantamiento2:** Debido al registro R5. Indicado en cyan. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Adelantamiento3:** Debido al registro R1. Indicado en verde. El registro de datos de memoria MDR se conecta directamente con la entrada de la ALU. Se produce un ciclo de bloqueo (**Bloqueo1**).



c) Sí se puede mejorar el rendimiento. Se puede eliminar el único bloqueo cambiando de posición la última instrucción.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SUB R3,R2,R2	IF	ID	EX	MEM	WB													
ADD R3,R2,R6		IF	ID	EX	MEM	WB												
SUB R5,R3,R8			IF	ID	EX	MEM	WB											
ADD R7,R2,R5				IF	ID	EX	MEM	WB										
LW R1,10(R6)					IF	ID	EX	MEM	WB									
SUB R6,R7,R8						IF	ID	EX	MEM	WB								
SW 10(R1),R3							IF	ID	EX	MEM	WB							
ADD R3,R1,R4								IF	ID	EX	MEM	WB						

Número de ciclos: **12**.

Ciclos de bloqueo: **0**.

**Adelantamiento1:** Debido al registro R3. Indicado en rojo. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Adelantamiento2:** Debido al registro R5. Indicado en cyan. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Adelantamiento3:** Debido al registro R1. Indicado en verde. El registro de datos de memoria MDR se conecta directamente con la entrada de la ALU. Ahora no se produce un ciclo de bloqueo.

- d) Repetir el ejercicio anterior teniendo en cuenta que la caché de memoria principal utilizada está unificada, esto es, no existen cachés separadas de datos e instrucciones.

## SOLUCIÓN EJERCICIO 2-apartado d)

### a) Memoria unificada sin bypass

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SUB R3, R2, R2	IF	ID	EX	MEM	WB													
ADD R3, R2, R6		IF	ID	EX	MEM	WB												
SUB R5, R3, R8			IF	Det	Det	ID	EX	MEM	WB									
ADD R7, R2, R5						IF	Det	Det	ID	EX	MEM	WB						
LW R1, 10(R6)									IF	ID	EX	MEM	WB					
SW 10(R1), R3										IF	Det	Det	ID	EX	MEM	WB		
ADD R3, R1, R4													IF	ID	EX	MEM	WB	
SUB R6, R7, R8														IF	ID	EX	MEM	WB

Número de ciclos: **18**.

Ciclos de bloqueo: **6**.

**Bloqueo1:** Riesgo de Datos RAW. Debido al registro R3.

**Bloqueo2:** Riesgo de Datos RAW. Debido al registro R5.

**Bloqueo3:** Riesgo de Datos RAW. Debido al registro R1.

## b) Memoria unificada con bypass

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SUB R3,R2,R2	IF	ID	EX	MEM	WB													
ADD R3,R2,R6		IF	ID	EX	MEM	WB												
SUB R5,R3,R8			IF	ID	EX	MEM	WB											
ADD R7,R2,R5				IF	ID	EX	MEM	WB										
LW R1,10(R6)					IF	ID	EX	MEM	WB									
SW 10(R1),R3						IF	ID	Det1	EX	MEM	WB							
ADD R3,R1,R4							IF	Det2	ID	EX	MEM	WB						
SUB R6,R7,R8									IF	ID	EX	MEM	WB					

Número de ciclos: **13**.

Ciclos de bloqueo: **2**.

**Adelantamiento1:** Debido al registro R3. Indicado en rojo. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Adelantamiento2:** Debido al registro R5. Indicado en azul. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Adelantamiento3:** Debido al registro R1. Indicado en verde. El registro de datos de memoria MDR se conecta directamente con la entrada de la ALU. Se produce un ciclo de bloqueo (**Bloqueo1**).

**Ejercicio 3.** El siguiente fragmento de código se ejecuta en un DLX con segmentación:

```
SUB R5,R3,R2
ADD R6,R3,R1
SW 10(R6),R2
SUB R6,R7,R8
ADD R7,R2,R5
LW R1,5(R7)
AND R4,R1,R2
SW 4(R4),R8
```

- Indicar el estado de cada instrucción durante los ciclos de ejecución del código, suponiendo DLX segmentado con memoria caché partida y sin bypass. Calcular el número de ciclos necesarios para ejecutar este código. Indicar si existen bloqueos en la cadena y a qué se deben.
- Repetir el primer apartado para una versión segmentada del DLX que incluye adelantamiento (bypass) generalizado. Indicar en qué casos se utiliza el adelantamiento o bypass.
- ¿Es posible mejorar el rendimiento del conjunto de instrucciones referido para la versión de DLX con adelantamiento generalizado?. En caso positivo indicar detalladamente cómo. Indicar en qué casos se utiliza el adelantamiento o bypass.

## SOLUCIÓN EJERCICIO 3

a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SUB R5,R3,R2	IF	ID	EX	MEM	WB													
ADD R6,R3,R1		IF	ID	EX	MEM	WB												
SW 10(R6),R2			IF	Det1	Det1	ID	EX	MEM	WB									
SUB R6,R7,R8						IF	ID	EX	MEM	WB								
ADD R7,R2,R5							IF	ID	EX	MEM	WB							
LW R1,5(R7)								IF	Det2	Det2	ID	EX	MEM	WB				
AND R4,R1,R2											IF	Det3	Det3	ID	EX	MEM	WB	
SW 4(R4),R8														IF	Det4	Det4	ID	EX

	19	20
AND R4, R1, R2		
SW 4(R4), R8	MEM	WB

Número de ciclos: **20**.

Ciclos de bloqueo: **8**.

**Bloqueo1:** Riesgo de Datos RAW. Debido al registro R6.

**Bloqueo2:** Riesgo de Datos RAW. Debido al registro R7.

**Bloqueo3:** Riesgo de Datos RAW. Debido al registro R1.

**Bloqueo4:** Riesgo de Datos RAW. Debido al registro R4.

b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SUB R5, R3, R2	IF	ID	EX	MEM	WB													
ADD R6, R3, R1		IF	ID	EX	MEM	WB												
SW 10 (R6), R2			IF	ID	EX	MEM	WB											
SUB R6, R7, R8				IF	ID	EX	MEM	WB										
ADD R7, R2, R5					IF	ID	EX	MEM	WB									
LW R1, 5 (R7)						IF	ID	EX	MEM	WB								
AND R4, R1, R2							IF	ID	Det1	EX	MEM	WB						
SW 4 (R4), R8								IF	Det2	ID	EX	MEM	WB					

Número de ciclos: 13.

Ciclos de bloqueo: 1.

**Adelantamiento1:** Debido al registro R6. Indicado en verde. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Adelantamiento2:** Debido al registro R7. Indicado en cyan. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Bloqueo1:** Riesgo de Datos RAW. Debido al registro R1.

**Bloqueo2:** Debido al bloqueo 1.

**Adelantamiento3:** Debido al registro R1. Indicado en azul. La salida de la MEM se conecta a la entrada de la ALU. Se produce bloqueo (Bloqueo 1).

**Adelantamiento4:** Debido al registro R4. Indicado en rosa. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Ejercicio 4.** Repetir el Ejercicio 3 teniendo en cuenta que la caché de memoria principal utilizada está unificada, esto es, no existen cachés separadas de datos e instrucciones.

## SOLUCIÓN EJERCICIO 4

### a) Caché unificada y sin bypass

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SUB R5, R3, R2	IF	ID	EX	MEM	WB													
ADD R6, R3, R1		IF	ID	EX	MEM	WB												
SW 10 (R6), R2			IF	Det1	Det1	ID	EX	MEM	WB									
SUB R6, R7, R8						IF	ID	EX	MEM	WB								
ADD R7, R2, R5							IF	ID	EX	MEM	WB							
LW R1, 5 (R7)								Det2	IF	Det3	ID	EX	MEM	WB				
AND R4, R1, R2											IF	Det4	Det4	ID	EX	MEM	WB	
SW 4 (R4), R8														IF	Det5	Det5	ID	EX

	19	20
AND R4, R1, R2		
SW 4 (R4), R8	MEM	WB

Número de ciclos: **20**.

Ciclos de bloqueo: **8**.

**Bloqueo1:** Riesgo de Datos RAW. Debido al registro R6.

**Bloqueo2:** Riesgo estructural. No se puede acceder a memoria de forma simultánea.

**Bloqueo3:** Riesgo de Datos RAW. Debido al registro R7.

**Bloqueo4:** Riesgo de Datos RAW. Debido al registro R1.

**Bloqueo5:** Riesgo de Datos RAW. Debido al registro R4.

b) Caché unificada con bypass generalizado

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
SUB R5,R3,R2	IF	ID	EX	MEM	WB													
ADD R6,R3,R1		IF	ID	EX	MEM	WB												
SW 10(R6),R2			IF	ID	EX	MEM	WB											
SUB R6,R7,R8				IF	ID	EX	MEM	WB										
ADD R7,R2,R5					IF	ID	EX	MEM	WB									
LW R1,5(R7)						Det1	IF	ID	EX	MEM	WB							
AND R4,R1,R2								IF	ID	Det2	EX	MEM	WB					
SW 4(R4),R8									IF	Det3	ID	EX	MEM	WB				

Número de ciclos: **14**.

Ciclos de bloqueo: **2**.



**Adelantamiento1:** Debido al registro R6. Indicado en verde. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Bloqueo1:** Riesgo estructural. No se puede acceder a memoria de forma simultánea.

**Adelantamiento2:** Debido al registro R7. Indicado en cyan. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Bloqueo2:** Riesgo de Datos RAW. Debido al registro R1.

**Bloqueo3:** Debido al bloqueo 2.

**Adelantamiento3:** Debido al registro R1. Indicado en rosa. La salida de memoria se conecta a la entrada de la ALU. Se produce bloqueo.

**Adelantamiento4:** Debido al registro R4. Indicado en azul. La salida de la ALU se conecta a la entrada de la ALU. No se produce bloqueo.

**Ejercicio 7.** Optimizar el siguiente código para que se ejecute en el menor número de ciclos posible en un procesador DLX con caché partida y con adelantamiento generalizado. Indicar claramente la solución propuesta y la mejora conseguida.

```
LW R1, 10(R6)
SW 10(R1), R3
LW R4, 10(R6)
SW 10(R2), R4
LW R2, 10(R6)
SW 10(R2), R5
LW R3, 10(R6)
SW 10(R3), R3
```

## SOLUCIÓN EJERCICIO 7

Sin ninguna reestructuración de código:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
LW R1, 10(R6)	IF	ID	EX	MEM	WB													
SW 10(R1), R3		IF	ID	Det	EX	MEM	WB											
LW R4, 10(R6)			IF	Det	ID	EX	MEM	WB										
SW 10(R2), R4					IF	ID	EX	MEM	WB									
LW R2, 10(R6)						IF	ID	EX	MEM	WB								
SW 10(R2), R5							IF	ID	Det	EX	MEM	WB						
LW R3, 10(R6)								IF	Det	ID	EX	MEM	WB					
SW 10(R3), R3										IF	ID	Det	EX	MEM	WB			

Número de ciclos: **15**.

Ciclos de bloqueo: **3**.

**Adelantamiento1:** Debido al registro R1. Indicado en rojo. El registro de datos de memoria MDR se conecta directamente con la entrada de la ALU. Se produce un ciclo de bloqueo (**Bloqueo1**).

**Adelantamiento2:** Debido al registro R4. Indicado en cyan. El registro de datos de memoria MDR se conecta directamente al bus de datos de memoria. No se produce un ciclo de bloqueo.

**Adelantamiento3:** Debido al registro R2. Indicado en verde. El registro de datos de memoria MDR se conecta directamente con la entrada de la ALU. Se produce un ciclo de bloqueo (**Bloqueo 2**).

**Adelantamiento4:** Debido al registro R3. Indicado en gris. El registro de carga de datos de memoria LMDR se conecta directamente con la entrada de la ALU y al registro de almacenamiento de datos en memoria SMDR; adelantamiento MEM-ALU y MEM-MEM. Se produce un ciclo de bloqueo (**Bloqueo 3**).

Reestructurando el código:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
LW R4, 10(R6)	IF	ID	EX	MEM	WB													
LW R1, 10(R6)		IF	ID	EX	MEM	WB												
SW 10(R2), R4			IF	ID	EX	MEM	WB											
LW R2, 10(R6)				IF	ID	EX	MEM	WB										
SW 10(R1), R3					IF	ID	EX	MEM	WB									
LW R3, 10(R6)						IF	ID	EX	MEM	WB								
SW 10(R2), R5							IF	ID	EX	MEM	WB							
SW 10(R3), R3								IF	ID	EX	MEM	WB						

Número de ciclos: 12.

Ciclos de bloqueo: 0.

**Adelantamiento1:** Debido al registro R4. Indicado en rojo. El registro de datos de memoria MDR se conecta directamente con el bus de datos. No se produce un ciclo de bloqueo.

**Adelantamiento2:** Debido al registro R3. Indicado en verde. El registro de datos de memoria MDR se conecta directamente con la entrada de la ALU y al bus de datos de memoria. No se produce un ciclo de bloqueo.

**Ejercicio 8.** El siguiente fragmento de código se ejecuta en un procesador DLX con segmentación. En dicho procesador la etapa EX dura 3 ciclos de reloj. Las direcciones de acceso a datos en memoria son calculadas exclusivamente en el tercer ciclo EX. La unidad aritmético-lógica está segmentada.

```
MUL R5,R1,R8
ADD R7,R5,R3
SUB R6,R2,R3
LW R3,10(R6)
LW R2,5(R3)
ADD R4,R2,R1
ADD R9,R7,R3
SUB R6,R7,R4
ADD R8,R5,R1
```

- a) Indicar el estado de cada instrucción durante los ciclos de ejecución del código. Calcular el número de ciclos necesarios para ejecutar este código. Indicar si existen bloqueos en la cadena y a qué se deben.
- b) Repetir el primer apartado para una versión segmentada del DLX que incluye adelantamiento (bypass) generalizado. Indicar claramente los datos que se adelantan y entre qué etapas actúa el bypass.

## SOLUCIÓN EJERCICIO 8

a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
MUL R5,R1,R8	IF	ID	EX1	EX2	EX3	MEM	WB											
ADD R7,R5,R3		IF	Det	Det	Det	Det	ID	EX1	EX2	EX3	MEM	WB						
SUB R6,R2,R3							IF	ID	EX1	EX2	EX3	MEM	WB					
LW R3,10(R6)								IF	Det	Det	Det	Det	ID	EX1	EX2	EX3	MEM	WB
LW R2,5(R3)													IF	Det	Det	Det	Det	ID
ADD R4,R2,R1																		IF
ADD R9,R7,R3																		
SUB R6,R7,R4																		
ADD R8,R5,R1																		

	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
MUL R5, R1, R8																
ADD R7, R5, R3																
SUB R6, R2, R3																
LW R3, 10 (R6)																
LW R2, 5 (R3)	EX1	EX2	EX3	MEM	WB											
ADD R4, R2, R1	Det	Det	Det	Det	ID	EX1	EX2	EX3	MEM	WB						
ADD R9, R7, R3					IF	ID	EX1	EX2	EX3	MEM	WB					
SUB R6, R7, R4						IF	Det	Det	Det	ID	EX1	EX2	EX3	MEM	WB	
ADD R8, R5, R1										IF	ID	EX1	EX2	EX3	MEM	WB

Número de ciclos: **34**.

Ciclos de bloqueo: **19**.

**Bloqueo1:** Riesgo de Datos RAW. Debido al registro R5.

**Bloqueo2:** Riesgo de Datos RAW. Debido al registro R6.

**Bloqueo3:** Riesgo de Datos RAW. Debido al registro R3.

**Bloqueo4:** Riesgo de Datos RAW. Debido al registro R2.

**Bloqueo5:** Riesgo de Datos RAW. Debido al registro R4.

b)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
MUL R5, R1, R8	IF	ID	EX1	EX2	EX3	MEM	WB											
ADD R7, R5, R3		IF	ID	Det	Det	EX1	EX2	EX3	MEM	WB								
SUB R6, R2, R3			IF	Det	Det	ID	EX1	EX2	EX3	MEM	WB							
LW R3, 10 (R6)						IF	ID	EX1	EX2	EX3	MEM	WB						
LW R2, 5 (R3)							IF	ID	Det	EX1	EX2	EX3	MEM	WB				
ADD R4, R2, R1								IF	Det	ID	Det	Det	Det	EX1	EX2	EX3	MEM	WB
ADD R9, R7, R3										IF	Det	Det	Det	ID	EX1	EX2	EX3	MEM
SUB R6, R7, R4														IF	ID	Det	EX1	EX2
ADD R8, R5, R1															IF	Det	ID	EX1

	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
MUL R5, R1, R8																
ADD R7, R5, R3																
SUB R6, R2, R3																
LW R3, 10(R6)																
LW R2, 5(R3)																
ADD R4, R2, R1																
ADD R9, R7, R3	WB															
SUB R6, R7, R4	EX3	MEM	WB													
ADD R8, R5, R1	EX2	EX3	MEM	WB												

Número de ciclos: **22**.

Ciclos de bloqueo: **7**.

**Bloqueo1:** Riesgo de Datos RAW. Debido al registro R5.

**Adelantamiento1:** Debido al registro R5. En color rojo. La salida de la ALU se conecta a la entrada de la ALU.

**Adelantamiento2:** Debido al registro R6. En color amarillo. La salida de la ALU se conecta a la entrada de la ALU. En este caso, la dependencia de datos RAW debida al registro R6 no ha producido detención al hacer el bypass.

**Bloqueo2:** Riesgo de Datos RAW. Debido al registro R3.

**Adelantamiento3:** Debido al registro R3. En color verde. El registro de datos de memoria MDR se conecta directamente a la entrada de la ALU en la etapa EX3.

**Bloqueo3:** Riesgo de Datos RAW. Debido al registro R2.

**Adelantamiento4:** Debido al registro R2. En color cyan. El registro de datos de memoria MDR se conecta a la entrada de la ALU en la etapa EX1.

**Adelantamiento5:** Debido al registro R4. En color magenta. La salida de la ALU se conecta a la entrada de la ALU.

**Bloqueo4:** Riesgo de Datos RAW. Debido al registro R4.

**Ejercicio 9.** El siguiente fragmento de código se ejecuta en un procesador DLX con segmentación. En dicho procesador la etapa MEM dura 2 ciclos de reloj y la etapa EX dura 2 ciclos de reloj. Las direcciones de acceso a datos en memoria son calculadas exclusivamente en el segundo ciclo EX. La unidad aritmético-lógica está segmentada. El acceso a memoria se realiza mediante un único puerto de acceso que no permite la segmentación de varias operaciones de memoria.

```
ADD R2,R2,R3
ADD R2,R1,R6
SUB R5,R2,R8
LW  R1,10(R6)
LW  R5,5(R1)
ADD R7,R5,R3
ADD R9,R7,R3
SUB R6,R7,R8
```

- Indicar el estado de cada instrucción durante los ciclos de ejecución del código. Calcular el número de ciclos necesarios para ejecutar este código. Indicar si existen bloqueos en la cadena y a qué se deben.
- Repetir el primer apartado para una versión segmentada del DLX que incluye adelantamiento (bypass) generalizado. Indicar claramente los datos que se adelantan y entre qué etapas actúa el bypass.
- ¿Es posible mejorar el rendimiento del conjunto de instrucciones referido para la versión de DLX con adelantamiento?. En caso positivo indicar detalladamente cómo.

## SOLUCIÓN EJERCICIO 9:

a)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ADD R2, R2, R3	IF	ID	EX1	EX2	MEM	MEM	WB											
ADD R2, R1, R6		IF	ID	EX1	EX2	Det1	MEM	MEM	WB									
SUB R5, R2, R8			IF	Det2	Det2	Det2	Det2	Det2	ID	EX1	EX2	MEM	MEM	WB				
LW R1, 10 (R6)									IF	ID	EX1	EX2	Det3	MEM	MEM	WB		
LW R5, 5 (R1)										IF	Det4	Det4	Det4	Det4	Det4	ID	EX1	EX2
ADD R7, R5, R3																IF	Det5	Det5

	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
LW R5, 5 (R1)	MEM	MEM	WB												
ADD R7, R5, R3	Det5	Det5	ID	EX1	EX2	MEM	MEM	WB							
ADD R9, R7, R3			IF	Det6	Det6	Det6	Det6	ID	EX1	EX2	MEM	MEM	WB		
SUB R6, R7, R8								IF	ID	EX1	EX2	Det7	MEM	MEM	WB

Número de ciclos: 33.

**Bloqueo 1:** Det.1-La Memoria no permite segmentación.**Bloqueo 2:** Det.2- No existe adelantamiento de datos. Riesgo de Datos RAW. Debido al registro R2.



**Bloqueo 3:** Det.3-La Memoria no permite segmentación.

**Bloqueo 4:** Det.4- No existe adelantamiento de datos. Riesgo de Datos RAW. Debido al registro R1.

**Bloqueo 5:** Det.5- No existe adelantamiento de datos. Riesgo de Datos RAW. Debido al registro R5.

**Bloqueo 6:** Det.6- No existe adelantamiento de datos. Riesgo de Datos RAW. Debido al registro R7.

**Bloqueo 7:** Det.7- La Memoria no permite segmentación.

b) Adelantamiento generalizado

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ADD R2, R2, R3	IF	ID	EX1	EX2	MEM	MEM	WB											
ADD R2, R1, R6		IF	ID	EX1	EX2	Det1	MEM	MEM	WB									
SUB R5, R2, R8			IF	ID	Det2	Det3	EX1	EX2	MEM	MEM	WB							
LW R1, 10 (R6)				IF	Det4	Det4	ID	EX1	EX2	Det5	MEM	MEM	WB					
LW R5, 5 (R1)							IF	ID	EX1	Det6	Det7	Det7	EX2	MEM	MEM	WB		
ADD R7, R5, R3								IF	ID	Det8	Det8	Det8	Det9	Det9	Det9	EX1	EX2	MEM
ADD R9, R7, R3									IF	Dt.10	Dt.10	Dt.10	Dt.10	Dt.10	Dt.10	ID	Dt.11	EX1
SUB R6, R7, R8																IF	Dt.12	ID

	19	20	21	22	23	24
ADD R7, R5, R3	MEM	WB				
ADD R9, R7, R3	EX2	MEM	MEM	WB		
SUB R6, R7, R8	EX1	EX2	Det.13	MEM	MEM	WB

Número de ciclos: **24**.

**Bloqueo 1:** Det.1-La Memoria no permite segmentación.

**Bloqueo 2:** Det.2-Riesgo de Datos RAW. Hay que esperar que la ALU haya calculado  $R1+R6$  (nuevo valor de R2).

**Bloqueo 3:** Det.3-Derivado de la Det.1.

**Adelantamiento 1:** EX2-EX1-Debido al registro R2. Indicado en verde. La salida de la ALU se conecta a la entrada de la ALU.

**Bloqueo 4:** Det.4-Derivado de la Det.2 y Det. 3.

**Bloqueo 5:** Det.5-La Memoria no permite segmentación.

**Bloqueo 6:** Det.6-Derivado de la Det.5.

**Bloqueo 7:** Det.7- Riesgo de Datos RAW. Hay que esperar a que  $M(10+R6)$  esté en el registro de datos de memoria (nuevo valor de R1).

**Adelantamiento 2:** MEM-EX2-Debido al registro R1. Indicado en rosa. El registro de datos de memoria MDR se conecta a la entrada de la ALU.

**Bloqueo 8:** Det.8-Derivado de la Det.6 y Det. 7.

**Bloqueo 9:** Det.9- Riesgo de Datos RAW. Hay que esperar a que  $M(5+R1)$  esté en el registro de datos de memoria (nuevo valor de R5).

**Adelantamiento 3:** MEM-EX1-Debido al registro R1. Indicado en cyan. El registro de datos de memoria MDR se conecta a la entrada de la ALU.

**Bloqueo 10:** Det.10-Derivado de la Det.8 y Det.9.

**Bloqueo 11:** Det.11- Riesgo de Datos RAW. Hay que esperar a que la ALU haya calculado  $R5+R3$  (nuevo valor de R7).

**Adelantamiento 4:** EX2-EX1-Debido al registro R7. Indicado en gris. La salida de la ALU se conecta a la entrada de la ALU.

**Bloqueo 12:** Det.12-Derivado de la Det.11.

**Adelantamiento 4:** EX2-EX1-Debido al registro R7. Indicado en gris. La salida de la ALU se conecta a la entrada de la ALU, para la operación SUB R6, R7, R8.

**Bloqueo 13:** Det.13-La Memoria no permite segmentación.

**Ejercicio 10** El siguiente fragmento de código se ejecuta en un procesador tipo DLX con segmentación. La latencia de la unidad funcional sumador/restador flotante es de 2 ciclos, la del multiplicador de 3 ciclos y la del divisor de 5 ciclos. Indicar el estado de cada instrucción durante los ciclos de ejecución del código. Calcular el número de ciclos necesarios para ejecutar este código. Indicar si existen bloqueos en la cadena y a qué se deben. Se considera adelantamiento o bypass generalizado. La unidad aritmético-lógica no está segmentada.

```
FDIV F5,F2,F3
FMUL F2,F1,F6
FSUB F5,F2,F8
LF F1,10(R6)
LF F5,5(R1)
FADD F7,F5,F3
FADD F9,F7,F3
FSUB F6,F7,F8
```

### SOLUCIÓN EJERCICIO 10:

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
FDIV F5,F2,F3	IF	ID	EX	EX	EX	EX	EX	MEM	WB								
FMUL F2,F1,F6		IF	ID	EX	EX	EX	MEM	WB									
FSUB F3,F2,F8			IF	ID	Det1	Det1	EX	EX	MEM	WB							
LF F1,10(R6)				IF	Det1	Det1	ID	EX	Det2	MEM	WB						
LF F5,5(R1)					Det1	Det1	IF	ID	Det2	EX	MEM	WB					
FADD F7,F5,F3								IF	Det2	ID	Det3	EX	EX	MEM	WB		
FADD F9,F7,F3										IF	Det3	ID	Det4	EX	EX	MEM	WB
FSUB F6,F7,F8												IF	Det4	ID	Det5	EX	EX

	18	19	20	21	22	23	24
FSUB F6,F7,F8	MEM	WB					

Número de ciclos: **19**.

Ciclos de bloqueo: **6**.

**Bloqueo1:** Riesgo de datos RAW. Debido al registro F2. 2 ciclos de bloqueo.

**Bloqueo2:** Riesgo Estructural. Colisión en etapa MEM. 1 ciclo de bloqueo.

**Adelantamiento1:** Debido al registro F5. Indicado en rojo. El registro de carga de datos de memoria LMDR se conecta directamente con la entrada de la ALU.

**Bloqueo3:** Riesgo de datos RAW. Debido al registro F5. 1 ciclo de bloqueo.

**Adelantamiento2:** Debido al registro F7. Indicado en verde. La salida de la ALU se conecta directamente con la entrada de la ALU.

**Bloqueo4:** Riesgo de datos RAW. Debido al registro F7. 1 ciclo de bloqueo.

**Adelantamiento3:** Debido al registro F7. Indicado en verde. La salida de la ALU se conecta directamente con la entrada de la ALU.

**Bloqueo5:** Riesgo estructural. La unidad funcional no está segmentada. Las operaciones FADD y FSUB las debe realizar la misma unidad funcional repitiendo el bucle tantas veces como ciclos necesite. 1 ciclo de bloqueo.

**Ejercicio 11.** El siguiente fragmento de código se ejecuta en un procesador tipo DLX con segmentación. La latencia de la unidad funcional sumador/restador flotante es de 1 ciclo, la del multiplicador de 3 ciclos y la del divisor de 5 ciclos. Indicar el estado de cada instrucción durante los ciclos de ejecución del código. Calcular el número de ciclos necesarios para ejecutar este código. Indicar si existen bloqueos en la cadena y a qué se deben. Se considera adelantamiento o bypass generalizado. La unidad aritmético-lógica no está segmentada.

```
FDIV F5,F2,F3
FMUL F2,F1,F6
FSUB F5,F2,F8
LF F1,10(R6)
LF F5,5(R1)
FADD F7,F5,F3
FADD F9,F7,F3
FSUB F6,F7,F8
```

## SOLUCIÓN EJERCICIO 11

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
FDIV F5,F2,F3	IF	ID	EX	EX	EX	EX	EX	MEM	WB								
FMUL F2,F1,F6		IF	ID	EX	EX	EX	MEM	WB									
FSUB F5,F2,F8			IF	ID	Det1	Det1	EX	Det2	MEM	WB							
LF F1,10(R6)				IF	Det1	Det1	ID	Det2	EX	MEM	WB						
LF F5,5(R1)							IF	Det2	ID	EX	MEM	WB					
FADD F7,F5,F3									IF	ID	Det3	EX	MEM	WB			
FADD F9,F7,F3										IF	Det3	ID	EX	MEM	WB		
FSUB F6,F7,F8												IF	ID	EX	MEM	WB	

Número de ciclos: **16**.

Ciclos de bloqueo: **4**.

**Bloqueo1:** Riesgo de datos RAW. Debido al registro F2. 2 ciclos de bloqueo.

**Adelantamiento1:** Debido al registro F2. Indicado en cyan. La salida de la ALU se conecta directamente con la entrada de la ALU.

**Bloqueo2:** Riesgo de datos WAW. Debido al registro F5. 1 ciclo de bloqueo.

**Bloqueo3:** Riesgo de datos RAW. Debido al registro F5. 1 ciclo de bloqueo.

**Adelantamiento2:** Debido al registro F5. Indicado en rojo. El registro de carga de datos de memoria LMDR se conecta directamente con la entrada de la ALU.

**Adelantamiento3:** Debido al registro F7. Indicado en verde. La salida de la ALU se conecta directamente con la entrada de la ALU. No se produce ciclo de bloqueo.

**Adelantamiento4:** Debido al registro F7. Indicado en verde. La salida de la ALU se conecta directamente con la entrada de la ALU. No se produce ciclo de bloqueo.

**Ejercicio 12.** El siguiente fragmento de código se ejecuta en un procesador DLX con segmentación. La latencia de la unidad funcional sumador/restador flotante es de 2 ciclos de reloj, la del multiplicador es de 3 ciclos, la del divisor de 4 ciclos y la fase MEM tiene una duración de un ciclo de reloj. Considerando adelantamiento de la salida de la ALU a la entrada de la ALU exclusivamente, una memoria con un único puerto de acceso, y el resto de unidades funcionales de coma flotante segmentadas.

```
ADDI R1, R0, #10
LW R2, 2(R1)
LW R5, 8(R1)
LF F0, 10(R2)
LF F2, 20(R5)
FMULT F4, F0, F2
FDIV F8, F1, F3
FADD F8, F4, F5
FADD F10, F4, F8
SF 10(R2), F8
SF 20(R5), F10
```

a) Indicar el estado de cada instrucción durante los ciclos de ejecución del código. Calcular el número de ciclos necesarios para ejecutar este código. Indicar si existen bloqueos en la cadena y a qué se deben.

b) Repetir el primer apartado para una versión segmentada del DLX que incluya adelantamiento (bypass) generalizado. Indicar claramente los datos que se adelantan y entre qué etapas actúa el bypass.

c) ¿Es posible mejorar el rendimiento del conjunto de instrucciones referido para la versión de DLX inicial?. En caso positivo indicar detalladamente cómo. -> Modificado: Repetir apartado a) considerando la memoria con dos puertos de acceso.

d) ¿Es posible mejorar el rendimiento del conjunto de instrucciones referido para la versión de DLX con adelantamiento generalizado?. En caso positivo indicar detalladamente cómo.



## SOLUCIÓN EJERCICIO 12:

## a) Adelantamiento salida ALU-entrada ALU

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ADDI R1, R2, #10	IF	ID	EX	MEM	WB													
LW R2, 2(R1)		IF	ID	EX	MEM	WB												
LW R5, 8(R1)			IF	ID	EX	MEM	WB											
LF F0, 10(R2)				IF	Det.1	ID	EX	MEM	WB									
LF F2, 20(R5)						Det.2	IF	ID	EX	MEM	WB							
FMULT F4, F0, F2								Det.3	IF	Det.4	ID	EX1	EX2	EX3	MEM	WB		
FDIV F8, F1, F3											IF	ID	EX1	EX2	EX3	EX4	MEM	WB
FADD F8, F4, F5												IF	ID	Det.5	EX1	EX2	Det.6	MEM
FADD F10, F4, F8													IF	Det.5	ID	Det.7	Det.6	EX1
SF 10(R2), F8															IF	Det.7	Det.6	Det.8
SF 20(R5), F10																		

	19	20	21	22	23	24
FADD F8, F4, F5	WB					
FADD F10, F4, F8	EX2	MEM	WB			
SF 10(R2), F8	ID	EX	MEM	WB		
SF 20(R5), F10	IF	Det.12	ID	EX	MEM	WB

Número de ciclos: **24**.Ciclos de bloqueo: **9**.

**Adelantamiento 1:** EX-EX-Debido al registro R1. Indicado en verde. La salida de la ALU se conecta a la entrada de la ALU.

**Adelantamiento 2:** EX-EX-Riesgo RAW debido al registro R1. Indicado en verde. La salida de la ALU se conecta a la entrada de la ALU, para la operación `LW R5, 8(R1)`.

**Bloqueo 1:** Det.1- Riesgo RAW debido al registro R2. Indicado en azul. No existe adelantamiento MEMORIA-ALU, para la fase ID de `LF F0, 10(R2)`, hay que esperar que `LW R2, 2(R1)` escriba el contenido de R2.

**Bloqueo 2:** Det.2- Riegos estructural, memoria con un único puerto de acceso. No pueden solaparse las fases IF y MEM (estando la fase MEM asociada a una instrucción LW).

**Bloqueo 3:** Det.3- Riegos estructural, memoria con un único puerto de acceso. No pueden solaparse las fases IF y MEM (estando la fase MEM asociada a una instrucción LW).

**Bloqueo 4:** Det.4- Riesgo RAW debido al registro F2. Indicado en rojo. No existe adelantamiento MEMORIA-ALU, para la fase ID de `FMULT F4, F0, F2`, hay que esperar que `LF F2, 20(R5)` escriba el contenido de F2.

**Bloqueo 5:** Det.5- Riesgo RAW debido al registro F4. Indicado en rosa. Para la fase EX de `FADD F8, F4, F5` hay que esperar `FMULT F4, F0, F2` haya calculado el valor de  $F0 \cdot F2$  (nuevo valor de F4).

**Adelantamiento 3:** EX-EX-Debido al registro F4. Indicado en rosa. La salida de la ALU se conecta a la entrada de la ALU.

**Bloqueo 6:** Det.6- Riesgo WAW debido al registro F8. La instrucción FADD debe definir el valor de F8 después de que lo haga FDIV.

**Bloqueo 7:** Det.7- Riesgo de Datos RAW debido al registro F8. Indicado en gris. Hay que esperar a que la ALU haya calculado  $F4 + F5$  (nuevo valor de F8).

**Adelantamiento 4:** EX-EX-Debido al registro F8. Indicado en gris. La salida de la ALU se conecta a la entrada de la ALU.

**Bloqueo 8: Det.8-** Riesgo de Datos RAW. Debido al registro F8. No existe adelantamiento MEMORIA-ALU. La fase ID de SF 10(R2), F8 hay que retrasarla hasta que FADD F8, F4, F5 esté en la fase WB.

**Bloqueo 9: Det.9-** Riesgo de Datos RAW. Debido al registro F10. No existe adelantamiento MEMORIA-ALU. La fase ID de SF 20(R5), F10 hay que retrasarla hasta que FADD F10, F4, F8 esté en la fase WB.

b) Adelantamiento generalizado

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ADDI R1, R2, #10	IF	ID	EX	MEM	WB													
LW R2, 2(R1)		IF	ID	EX	MEM	WB												
LW R5, 8(R1)			IF	ID	EX	MEM	WB											
LF F0, 10(R2)				IF	ID	EX	MEM	WB										
LF F2, 20(R5)					Det.1	Det.1	Det.1	IF	ID	EX	MEM	WB						
FMULT F4, F0, F2									IF	ID	Det.2	EX1	EX2	EX3	MEM	WB		
FDIV F8, F1, F3										IF	Det.2	ID	EX1	EX2	EX3	EX4	MEM	WB
FADD F8, F4, F5												IF	ID	Det.3	EX1	EX2	Det.4	MEM
FADD F10, F4, F8													IF	Det.3	ID	Det.5	Det.4	EX1
SF 10(R2), F8															IF	Det.5	Det.4	ID
SF 20(R5), F10																		IF

	19	20	21	22	23
FADD F8, F4, F5	WB				
FADD F10, F4, F8	EX2	MEM	WB		
SF 10(R2), F8	EX	Det.6	MEM	WB	
SF 20(R5), F10	ID	Det.6	EX	MEM	WB

Número de ciclos: 22.

**Adelantamiento 1:** EX-EX-Debido al registro R1. Indicado en verde. La salida de la ALU se conecta a la entrada de la ALU.

**Adelantamiento 2:** MEM-EX-Debido al registro R2. Indicado en rosa. El registro de carga de datos de memoria LMDR se conecta a la entrada de la ALU.

**Bloqueo 1:** Det.1- Riesgo estructural. Memoria con un único puerto de acceso. No pueden solaparse la fase IF de cualquier instrucción con la fase MEM de instrucciones de carga/almacenamiento.

**Bloqueo 2:** Det.2- Hay que esperar a que  $M(20+R5)$  esté en el registro de almacenamiento de datos de memoria SMDR (nuevo valor de F2).

**Adelantamiento 3:** MEM-EX-Debido al registro F2. Indicado en gris. El registro de carga de datos de memoria LMDR se conecta a la entrada de la ALU.

**Bloqueo 3:** Det.3- Riesgo de Datos RAW. Hay que esperar a que la ALU haya calculado  $F0+F2$  (nuevo valor de F4).

**Adelantamiento 4:** EX-EX-Debido al registro F4. Indicado en azul. La salida de la ALU se conecta a la entrada de la ALU.

**Bloqueo 4:** Det.4- Riesgo WAW debido al registro F8. La instrucción FADD debe definir el valor de F8 después de que lo haga FDIV.

**Bloqueo 5:** Det.5- Riesgo de Datos RAW. Hay que esperar a que la ALU haya calculado  $F4+F5$  (nuevo valor de F8).

**Adelantamiento 5:** EX-EX-Debido al registro F8. Indicado en rojo. La salida de la ALU se conecta a la entrada de la ALU.

**Bloqueo 6:** Det.6 Riesgo estructural. No pueden existir dos fases idénticas asociadas a instrucciones diferentes.

**Adelantamiento 6:** EX-MEM-Debido al registro F8. Indicado en verde. La salida de la ALU se conecta a la entrada de la MEM.

**Adelantamiento 7:** EX-MEM-Debido al registro F10. Indicado en negro. La salida de la ALU se conecta a la entrada de la MEM.

c) Repetir el apartado a) considerando Memoria con dos puertos de acceso.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
ADDI R1, R2, #10	IF	ID	EX	MEM	WB													
LW R2, 2(R1)		IF	ID	EX	MEM	WB												
LW R5, 8(R1)			IF	ID	EX	MEM	WB											
LF F0, 10(R2)				IF	Det.1	ID	EX	MEM	WB									
LF F2, 20(R5)						IF	ID	EX	MEM	WB								
FMULT F4, F0, F2							IF	Det.2	Det.2	ID	EX1	EX2	EX3	MEM	WB			
FDIV F8, F1, F3										IF	ID	EX1	EX2	EX3	EX4	MEM	WB	
FADD F8, F4, F5											IF	ID	Det.3	EX1	EX2	Det.4	MEM	WB
FADD F10, F4, F8												IF	Det.3	ID	Det.5	Det.4	EX1	EX2
SF 10(R2), F8														IF	Det.5	Det.4	Det.6	ID
SF 20(R5), F10																		IF

	19	20	21	22	23
FADD F10, F4, F8	MEM	WB			
SF 10(R2), F8	EX	MEM	WB		
SF 20(R5), F10	Det.7	ID	EX	MEM	WB

Número de ciclos: 23.

**Adelantamiento 1:** EX-EX-Debido al registro R1. Indicado en verde. La salida de la ALU se conecta a la entrada de la ALU.

**Adelantamiento 2:** EX-EX-Debido al registro R1. Indicado en verde. La salida de la ALU se conecta a la entrada de la ALU, para la operación LW R5, 8(R1).

**Bloqueo 1:** Det.1-No existe adelantamiento MEMORIA-ALU, para la fase ID de `LF F0, 10(R2)`, hay que esperar que `LW R2, 2(R1)` escriba el contenido de R2.

**Bloqueo 2:** Det.2-No existe adelantamiento MEMORIA-ALU, para la fase ID de `FMULT F4, F0, F2` hay que esperar que `LF F0, 10(R2)` escriba el contenido de F0.

**Bloqueo 3:** Det.3-No existe adelantamiento MEMORIA-ALU, para la fase EX de `FADD F8, F4, F5` hay que esperar `FMULT F4, F0, F2` haya calculado el valor de  $F0 \cdot F2$  (nuevo valor de F4).

**Adelantamiento 2:** EX-EX-Debido al registro F4. Indicado en rosa. La salida de la ALU se conecta a la entrada de la ALU.

**Bloqueo 4:** Det.4- Riesgo WAW debido al registro F8. La instrucción FADD debe definir el valor de F8 después de que lo haga FDIV.

**Bloqueo 5:** Det.5- Riesgo de Datos RAW. Hay que esperar a que la ALU haya calculado  $F4 + F5$  (nuevo valor de F8).

**Adelantamiento 3:** EX-EX-Debido al registro F8. Indicado en gris. La salida de la ALU se conecta a la entrada de la ALU.

**Bloqueo 6:** Det.6- Riesgo de Datos RAW. Debido al registro F8. No existe adelantamiento MEMORIA-ALU. La fase ID de `SF 10(R2), F8` hay que retrasarla hasta que `FADD F8, F4, F5` esté en la fase WB.

**Bloqueo 7:** Det.7- Riesgo de Datos RAW. Debido al registro F10. No existe adelantamiento MEMORIA-ALU. La fase ID de `SF 20(R5), F10` hay que retrasarla hasta que `FADD F10, F4, F8` esté en la fase WB.