Ejercicio 1. Orden en el qué se emiten las instrucciones. Tabla de ciclos



						<u> Ciclos</u>	<u> </u>				-		-	
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
lw r1 , 0x1ac														
lw r2, 0xc1f														
add r3, r0, r0														
mul r4, r2, r1														
add r3, r3, r4														
add r5, r0, 0x1ac														
add r 6, r 0, 0xc1f														
sub r 5, r 5, #4														
sub r6, r6, #4														
sw (r5), r3														
sw (r6), r4														
Unidados: 1 LD /2 cir	-l-s\ 1	CT /1	داماء)	2 4 D D	/CLID /	1 \ 1 \ 1	111 /41		-			L D	2c	

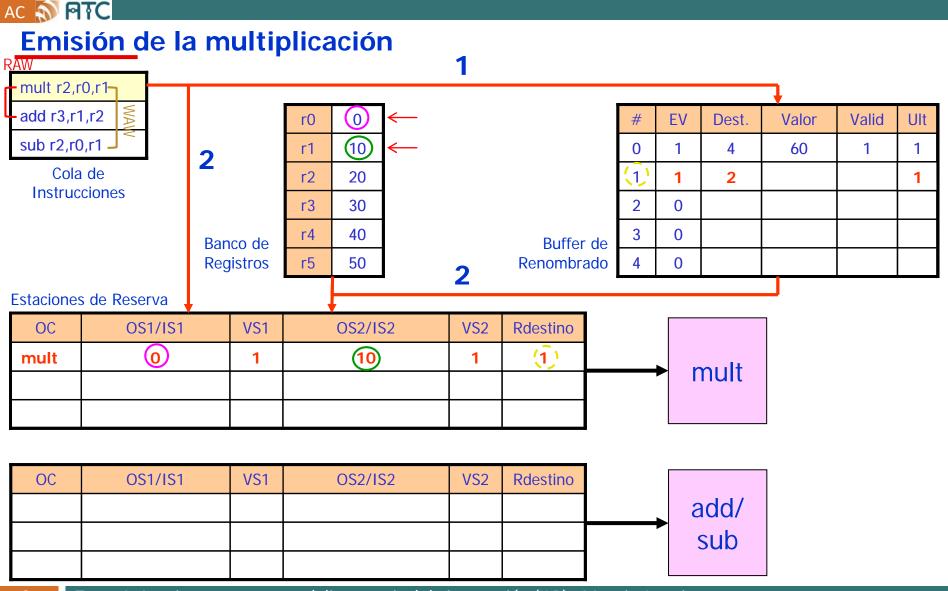
IF

ID

WB

- Unidades: 1 LD (2 ciclos), 1 ST (1 ciclo), 3 ADD/SUB (1), 1 MUL (4).
- > Emisión ordenada.
- > Suponemos unidades funcionales **no segmentadas**. Suponemos válido el contenido de un registro la primera vez que se usa en el código.
- > WAW y WAR resueltos por **renombrado** (los colores representan renombrado y destacan los RAW)
- Las instrucciones decodificadas, que estarán en la ventana de instrucciones, se pueden emitir si tienen los operandos disponibles y hay unidad disponible para ejecutarlas.

Bits de Válido (VS1 y VS2) en la Ventana de instr./estaciones de reserva



Ejercicio 1a. Captación (*Instruction Fetch*–IF)

AC N PTC														
					Ci	clos								
OK INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1 , 0x1ac	IF													
1 lw <u>r2</u> , 0xc1f	IF													
11 add r3 , r 0, r 0	IF													
00 mul r4, r2, r1	IF													
00 add r3 , r3 , r4		IF												
11 add r5 , r0, 0x1ac		IF												
11 add r 6, r 0, 0xc1f		IF												
01 sub r5 , r5 , #4		IF												
01 sub r6, r6, #4			IF											
00 sw (r5), r3			IF											
00 sw (r6), r4			IF											

- > Las instrucciones se captan y decodifican en el orden del programa
- ➤ Ciclo 1: al terminar este ciclo estarán captadas las 4 primeras instrucciones. En *gris* se muestra lo que aún no ha ocurrido.
- > Primera columna: bits OK o *válido* (VS1 y VS2) de la ventana de instrucciones para los dos operandos de entrada. En *gris* se muestran los bits de las instrucciones que aún no están en la ventana de instrucciones.

Ejercicio 1a. Decodificación (*Instruction Decode* –ID)

AC N PTC														
					Ci	clos								
OK INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1 , 0x1ac	IF	ID												
1 lw r2, 0xc1f	IF	ID												
11 add r3, r0, r0	IF	ID												
00 mul r4, r2, r1	IF	ID												
00 add r3 , r3 , r4		IF	ID											
11 add r5 , r 0, 0x1ac		IF	ID											
11 add r 6, r 0, 0xc1f		IF	ID											
01 sub r5 , r5 , #4		IF	ID											
01 sub r6, r6, #4			IF	ID										
00 sw (r5), r3			IF	ID		_								
00 sw (r6), r4			IF	ID										

- Las instrucciones se captan y decodifican en el orden del programa
- Ciclo 2: al terminar este ciclo se habrán decodificado las 4 primeras instrucciones (estarán entonces en la ventana de instrucciones) y captado las 4 siguientes.

Ejercicio 1a. Decodificación (*Instruction Decode* –ID)

AC N PTC														
					Ci	clos								
OK INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1, 0x1ac	IF	ID												
1 lw <u>r2</u> , 0xc1f	IF	ID												
11 add r 3, r 0, r 0	IF	ID												
00 mul r4, r2, r1														
00 add r3, r3, r4														
11 add r5, r0, 0x1ac														
11 add r 6, r0, 0xc1f		IF	ID											
01 sub r5 , r5 , #4		IF	ID											
01 sub r6, r6, #4			IF	ID										
00 sw (r5), r3			IF	ID										
00 sw (r6), r4			IF	ID										
Ciclo 3:			·									L D	2c	
 Las 4 primeras instr 	uccior	ies ya	están (en la v	entan	a de ir	struc.	ပ		ر ا	့	S T A D S	1c 1c	VA/D

- Se podrían emitir las tres instrucciones con operandos disponibles si hubiera 2 unidades de LD, pero sólo hay 1 (dependencia estructural).
- Se podría emitir también la tercera instrucción si la emisión fuese desordenada.

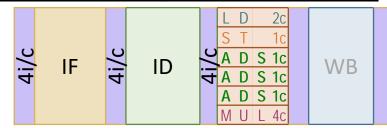
• Se puede emitir a ejecución en este ciclo sólo la primera

instrucción

Ejercicio 1a. Ejecución (Execution - Ex)

AC MIC														
					Ci	clos								
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1, 0x1ac	IF	ID	EX	EX										
1 lw r2, 0xc1f	IF	ID												
11 add r3, r0, r0	IF	ID	EX											
00 mul r4, r2, r1	IF	ID												
00 add r3, r3, r4		IF	ID											
11 add r5, r0, 0x1ac		IF	ID	EX										
11 add r6, r0, 0xc1f		IF	ID	EX										
01 sub r 5, r 5, #4		IF	ID											
01 sub r6, r6, #4			IF	ID										
00 sw (r5), r3			IF	ID										
00 sw (r6), r4			IF	ID										

- Ciclo 3: Con emisión desordenada se podría haber emitido también add r3, r0, r0
- Ciclo 4: Con emisión desordenada se podrían emitir add r5, r0, 0x1ac y add r6, r0, 0xc1f

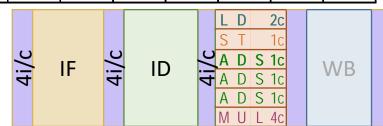


AC MIC														
					Ci	clos								
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1, 0x1ac	IF	ID	EX	EX										
1 lw r2, 0xc1f	IF	ID												
11 add r3, r0, r0	IF	ID												
01 mul r4, r2, r1	IF	ID												
00 add r3, r3, r4		IF	ID											
11 add r 5, r 0, 0x1ac		IF	ID											
11 add r 6, r 0, 0xc1f		IF	ID											
01 sub r 5, r 5, #4		IF	ID											
01 sub r6, r6, #4			IF	ID										
00 sw (r5), r3			IF	ID										
00 sw (r6), r4			IF	ID										
Cido Fu														

Ciclo 5:

Ya está disponible el resultado de lw rl, 0x1ac, que es el 2º operando de mul r4, r2, rl

• Se pueden emitir lw r2, 0xclf y add r3, r0, r0 (ambas tienen los operandos disponibles y unidad disponible)



AC N PTC														
					Ci	clos								
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1, 0x1ac	IF	ID	EX	EX										
1 lw <u>r2</u> , 0xc1f	IF	ID			EX	EX								
11 add r3, r0, r0	IF	ID			EX									
01 mul r4, r2, r1	IF	ID												
10 add r3, r3, r4		IF	ID											
11 add r5, r0, 0x1ac		IF	ID											
11 add r 6, r 0, 0xc1f		IF	ID											
01 sub r5 , r5 , #4		IF	ID											
01 sub r6, r6, #4			IF	ID										
00 sw (r5), r3			IF	ID										
00 sw (r6), r4			IF	ID										

- > Ciclo 6:
 - Ya está disponible el resultado de add r3, r0, r0, que es el 1er operando de add r3, r3, r4
 - No se puede emitir ninguna instrucción (las que hay después, en el orden del programa, a la última emitida no tiene los dos operandos disponibles)
- EX rojo: si la emisión hubiera sido desordenada se hubiera emitido antes. EX verde: si no hubiera habido dependencia estructural se hubiera emitido antes.

AC MAC SALE														
					Ci	clos								
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1, 0x1ac	IF	ID	EX	EX										
1 lw r2, 0xc1f	IF	ID			EX	EX								
11 add r3, r0, r0	IF	ID			EX									
11 mul r4, r2, r1	IF	ID												
10 add r3 , r3 , r4		IF	ID											
11 add r5, r0, 0x1ac		IF	ID											
11 add r6, r0, 0xc1f		IF	ID											
01 sub r 5, r 5, #4		IF	ID											
01 sub r6, r6, #4			IF	ID			•							
00 sw (r 5), r 3			IF	ID										
00 sw (r6), r4			IF	ID										
Ciclo 7:													2c	

- Ya está disponible el resultado de lw r2, 0xclf, que es el 1er operando de mul r4, r2, r1 (ya tiene todos los operandos
- Se puede emitir mul r4, r2, r1 (tiene todos los operandos disponibles y unidad disponible)

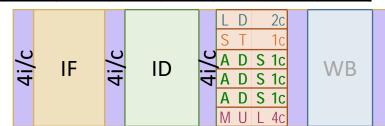
AC	≫ MIC														
						Ci	clos		·	·					
	INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	lw r1, 0x1ac	IF	ID	EX	EX	1									
1	lw r2, 0xc1f	IF	ID			EX	EX								
11	add r3 , r0, r0	IF	ID			EX									
11	mul r4, r2, r1	IF	ID					EX	EX	EX	EX				
10	add r3, r3, r4		IF	ID											
11	add r5 , r0, 0x1ac		IF	ID											
11	add r 6, r0, 0xc1f		IF	ID											
01	sub r 5, r 5, #4		IF	ID											
01	sub r6, r6, #4			IF	ID										
00	sw (r 5), r 3			IF	ID										
00	sw (r6), r4			IF	ID										

- ➤ Unidades: 1 LD (2 ciclos), 1 ST (1 ciclo), 3 ADD/SUB (1), 1 MUL (4). Emisión ordenada.
- Las flechas notan dependencias RAW (ver colores de los registros)
- Ciclos 8-10: No hay instrucciones que se puedan emitir porque la siguiente instrucción, en el orden del programa, a la última emitida necesita que acabe la multiplicación para tener todos sus operandos.

AC M PIC														
					Ci	clos		ii.		ii.			ii.	
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1, 0x1ac	IF	ID	EX	EX-	1									
1 lw r2, 0xc1f	IF	ID			EX	EX								
11 add r3, r0, r0	IF	ID			EX									
11 mul r4, r2, r1	IF	ID					EX	EX	EX	EX				
11 add r3, r3, r4		IF	ID											
11 add r5, r0, 0x1ac		IF	ID											
11 add r6, r0, 0xc1f		IF	ID											
01 sub r 5, r 5, #4		IF	ID											
01 sub r6, r 6, #4			IF	ID										
00 sw (r5), r3	_		IF	ID		_			_	_			_	_
01 sw (r6), r4			IF	ID										

➤ Ciclo 11:

- Ya está disponible el resultado de mul r4, r2, r1, que es el 2º operando add r3, r3, r4
- Se pueden emitir add r3, r3, r4, add r5, r0, 0x1ac y add r6, r0, 0xc1f (tiene todos los operandos disponibles y, como hay 3 unidades ADD/SUB, también tienen unidades disponibles)



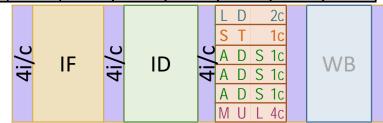
AC M PTC														
					Ci	clos								
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1, 0x1ac	IF	ID	EX	EX-	1									
1 lw r2, 0xc1f	IF	ID			EX	EX								
11 add r3, r0, r0	IF	ID			EX-									
11 mul r4, r2, r1	IF	ID					EX	EX	EX	EX	<u> </u>			
11 add r3 , r3 , r4		IF	ID							4	EX			
11 add r5, r0, 0x1ac		IF	ID								EX			
11 add r 6, r 0, 0xc1f		IF	ID								EX			
11 sub r5 , r5 , #4		IF	ID											
11 sub r6, r6, #4			IF	ID										
0 1 sw (r 5), r 3			IF	ID										
01 sw (r6), r4			IF	ID										
Ciclo 12:												L D	2c	
 Ya están disponible r5, r0, 0x1ac y ac 				e add	r3, r3	3, r4 , a	ıdd	4i/c	IF	41/c	4i/c	A D S	1c 1c 1c	WB
 Se pueden emitir s 	sub <mark>r5</mark>	, r5, #	‡4 y sı	ıb r6,	<mark>r6</mark> ,#4	1						A D S M U L	4c	

EX rojo: si la emisión hubiera sido desordenada se hubiera emitido antes. EX verde: si no hubiera habido dependencia estructural se hubiera emitido antes.

(tiene todos los operandos disponibles y unidades disponibles porque hay 3 unidades ADD/SUB

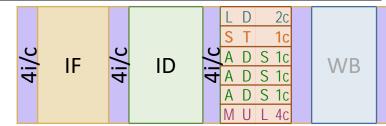
AC MAC														
					Ci	clos								
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1, 0x1ac	IF	ID	EX	EX-	1									
1 lw <u>r2</u> , 0xc1f	IF	ID			EX	EX								
11 add r3, r0, r0	IF	ID			EX-									
11 mul r4, r2, r1	IF	ID				, /	EX	EX	EX	EX				
11 add r3, r3, r4		IF	ID							2	EX			
11 add r5, r0, 0x1ac		IF	ID								EX			
11 add r 6, r 0, 0xc1f		IF	ID								EX			
11 sub r 5, r 5, #4		IF	ID									EX		
11 sub r6, r6, #4			IF	ID								EX		
11 sw (r5), r3			IF	ID										
11 sw (r6), r4			IF	ID										
Ciclo 13:													2c	
., ., ., .						- 11.4	ı					ST	1c	

- Ya están disponibles los resultados de sub r5, r5, #4 y sub r6, r6, #4 (sw (r5), r3 y sw (r6), r4 tienen todos los operandos disponibles).
- De las dos instrucciones con operandos disponibles se puede emitir sólo la primera, sw (r5), r3 porque hay sólo 1 unidad ST.



AC MIC														
					Ci	clos								
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1, 0x1ac	IF	ID	EX	EX-	1									
1 lw r2, 0xc1f	IF	ID			EX	EX								
11 add r3, r0, r0	IF	ID			EX-									
11 mul r4, r2, r1	IF	ID				, ,	EX	EX	EX	EX				
11 add r3, r3, r4		IF	ID							2	EX.			
11 add r5, r0, 0x1ac		IF	ID								EX	\		
11 add r 6, r 0, 0xc1f		IF	ID								EX			
11 sub r 5, r 5, #4		IF	ID									EX		
11 sub r6, r6, #4			IF	ID								EX	44	
11 sw (r 5), r 3			IF	ID									EX	
11 sw (r6), r4			IF	ID										
C: al a 4.4.														

- ➤ Ciclo 14:
 - Ya está disponible el resultado de sw (r5), r3.
 - Se puede emitir sw (r6), r4 (porque ya tiene unidad disponible).



AC NATC														
					Ci	clos								
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1, 0x1ac	IF	ID	EX	EX-	1									
1 lw <u>r2</u> , 0xc1f	IF	ID			EX	EX								
11 add r3, r0, r0	IF	ID			EX-									
11 mul r4, r2, r1	IF	ID				,	EX	EX	EX	EX-	1			
11 add r3, r3, r4		IF	ID								EX			
11 add r5, r0, 0x1ac		IF	ID								EX	\		
11 add r 6, r 0, 0xc1f		IF	ID								EX			
11 sub r 5, r 5, #4		IF	ID									EX		ı
11 sub r6, r6, #4			IF	ID								EX	NA .	1
11 sw (r5), r3			IF	ID	_	_	_	_	_	_			ΕX	X X
11 sw (r6), r4			IF	ID										EX
	1 1.						• •			,				

- EX rojo: si la emisión hubiera sido desordenada se hubiera emitido antes. EX en verde: si no hubiera habido dependencia estructural se hubiera emitido antes.
- Las instrucciones están ejecutándose desde el ciclo 3 al ciclo 14.
- Las instrucciones comienzan a procesarse en el ciclo 1, terminarán de procesarse cuando todas hayan modificado los registros de la arquitectura (no se muestra la finalización del procesamiento).

Ejercicio 1b. Captación (IF) y Decodificación (ID)

AC M M C														
					Ci	clos								
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1, 0x1ac	IF	ID												
1 lw <u>r2</u> , 0xc1f	IF	ID												
11 add r3 , r0 , r0	IF	ID												
00 mul r4, r2, r1	IF	ID												
00 add r3 , r3 , r4		IF	ID											
11 add r5, r0, 0x1ac		IF	ID											
11 add r 6, r 0, 0xc1f		IF	ID											
01 sub r5 , r5 , #4		IF	ID											
01 sub r6, r6, #4			IF	ID										
00 sw (r5), r3			IF	ID										
00 sw (r6), r4			IF	ID										

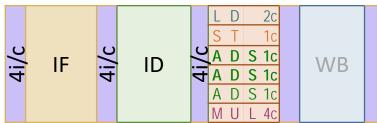
- Unidades: 1 LD (2 ciclos), 1 ST (1 ciclo), 3 ADD/SUB (1), 1 MUL (4).
- > Emisión desordenada.
- Suponemos unidades funcionales no segmentadas. Suponemos válido el contenido de un registro la primera vez que se usa en el código
- | L D 2c | S T 1c | A D S 1c | A D S 1c | M U L 4c | WB |
- > WAW y WAR resueltos por renombrado (los colores representan renombrado y destacan los RAW)
- > Las instrucciones decodificadas, que estarán en la ventana de instrucciones, se pueden emitir si tienen los *operandos disponibles* y hay *unidad disponible para ejecutarlas*.

Ejercicio 1b. Captación (IF) y Decodificación (ID)

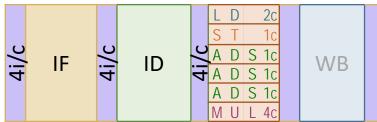
AC MATC														
					Ci	clos								
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$\frac{1}{1}$ lw $\frac{1}{1}$, 0x1ac	IF	ID	EX	EX										
1 lw r2, 0xc1f	IF	ID												
11 add r3, r0, r0	IF	ID	EX											
00 mul r4, r2, r1	IF	ID												
10 add r3, r3, r4		IF	ID											
11 add r5, r0, 0x1ac		IF	ID											
11 add r6, r0, 0xc1f		IF	ID											
01 sub r 5, r 5, #4		IF	ID											
01 sub r6, r6, #4			IF	ID										
00 sw (r5), r3			IF	ID										
00 sw (r6), r4			IF	ID										
												L D	2c	

Ejercicio 1b. Ejecución (Execution - Ex)

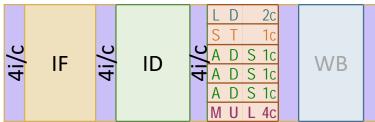
AC MAC														
					Ci	iclos								
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1, 0x1ac	IF	ID	EX	EX										
1 l w r2, 0xc1f	IF	ID												
11 add r3, r0, r0	IF	ID	EX											
01 mul r4, r2, r1	IF	ID												
10 add r3, r3, r4		IF	ID											
11 add r5, r0, 0x1ac		IF	ID	EX										
11 add r 6, r0, 0xc1f		IF	ID	EX										
11 sub r5 , r5 , #4		IF	ID											
11 sub r6, r6, #4			IF	ID										
00 sw (r5), r3			IF	ID										
00 sw (r6), r4			IF	ID										
•														



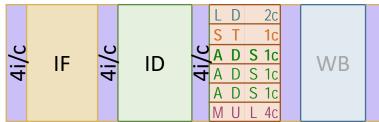
AC MATC												
				Cic	los							
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12
1 lw $r1$, $0x1ac$	IF	ID	EX	EX								
1 lw r2, 0xc1f	IF	ID			EX	EX						
11 add r3, r0, r0	IF	ID	EX									
01 mul r4, r2, r1	IF	ID										
10 add r3 , r3 , r4		IF	ID									
11 add $r5$, $r0$, $0x1ac$		IF	ID	EX								
11 add r 6, r0, 0xc1f		IF	ID	EX	M							
11 sub r 5, r 5, #4		IF	ID		EX							
11 sub r6, r6, #4			IF	ID	EX							
1 0 sw (r 5), r 3			IF	ID								
10 sw (r6), r4			IF	ID								



AC S PTC												
				Cic	los							
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12
1 lw $r1$, $0x1ac$	IF	ID	EX	EX								
1 lw r2, 0xc1f	IF	ID			EX	EX						
11 add r3, r0, r0	IF	ID	EX									
11 mul r4, r2, r1	IF	ID										
10 add r3 , r3 , r4		IF	ID									
11 add $r5$, $r0$, $0x1ac$		IF	ID	EX								
11 add 16 , 10 , 0 xc1f		IF	ID	EX	N							
11 sub r 5, r 5, #4		IF	ID		EX							
11 sub r6, r6, #4			IF	ID	EX							
10 sw (r 5), r 3			IF	ID		·						
10 sw (r6), r4			IF	ID								



AC MATC												
				Cic	os							
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12
1 lw r1, 0x1ac	IF	ID	EX	EX-	1							
1 lw r2, 0xc1f	IF	ID			EX	EX						
11 add r3, r0, r0	IF	ID	EX									
11 mul r4, r2, r1	IF	ID					EX	EX	EX	EX		
11 add r3, r3, r4		IF	ID									
11 add $r5$, $r0$, $0x1ac$		IF	ID	EX								
11 add r 6, r0, 0xc1f		IF	ID	EX								
11 sub r 5, r 5, #4		IF	ID		EX							
11 sub r6, r6, #4			IF	ID	EX							
10 sw (r 5), r 3			IF	ID								
1 <mark>1</mark> sw (r6), r4			IF	ID								
· · · · · · · · · · · · · · · · · · ·												



AC N PIC												
				Cicl	os							
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12
1 lw $r1$, $0x1ac$	IF	ID	EX	EX-								
1 lw r2, 0xc1f	IF	ID			EX	EX						
11 add r3, r0, r0	IF	ID	EX-	1								
11 mul r4, r2, r1	IF	ID					EX	EX	EX	EX		
11 add r3, r3, r4		IF	ID							-+	EX	
11 add r5 , r0, 0x1ac		IF	ID	EX								
11 add r 6, r0, 0xc1f		IF	ID	EX								
11 sub r 5, r 5, #4		IF	ID		EX							
11 sub r6, r6, #4			IF	ID	EX.							
11 sw (r5), r3			IF	ID				- '				
11 sw (r6), r4			IF	ID							EX	
										L D	2c	

AC N PTC				0: 1								
			,	<u>Cicl</u>	<u>os </u>							
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12
1 lw r1, 0x1ac	IF	ID	EX	EX-	1							
1 lw r2, 0xc1f	IF	ID			EX	EX						
11 add r3, r0, r0	IF	ID	EX-	1								
11 mul r4, r2, r1	IF	ID					EX	EX	EX	EX		
11 add r3 , r3 , r4		IF	ID							-+	EX	
11 add r5, r0, 0x1ac		IF	ID	EX								
11 add r 6, r 0, 0 xc 1 f		IF	ID	EX								
11 sub r5, r5, #4		IF	ID		EX-	 -						
11 sub r6, r6, #4			IF	ID	EX.			<u>'</u> '	1			•
11 sw (r5), r3			IF	ID			! !	, ,				EX
11 sw (r6), r4			IF	ID						<i> </i>	EX	

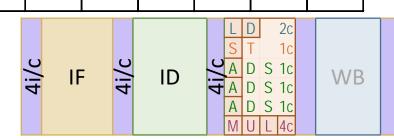
- Las instrucciones están ejecutándose desde el ciclo 3 al ciclo 12 (en lugar del 14).
- ¿Qué diferencia habría si las unidades estuvieran segmentadas?
- ¿Cuánto tiempo tardaría el procesamiento de esta secuencia de instrucciones si se usara ROB y la etapa WB pudiera retirar del ROB 4 instrucciones por ciclo?

Ejercicio 1b. Ejecución suponiendo unidades segmentadas

AC	◎ PIC												
					Ci	<u>clos</u>							
	INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12
1	lw r1, 0x1ac	IF	ID	EX	EX-	1							
1	lw r2, 0xc1f	IF	ID		EX	EX							
11	add r3, r0, r0	IF	ID	EX-	1								
11	mul r4, r2, r1	IF	ID				EX	EX	EX	EX			
11	add r3, r3, r4		IF	ID			1				EX		
11	add r5, r0, 0x1ac		IF	ID	EX								
11	add <u>r6</u> , r0, 0xc1f		IF	ID	EX								
11	sub r5 , r5 , #4		IF	ID		EX-	,						
11	sub r6, r6, #4			IF	ID	EX -	, ,					1	
11	sw (r 5), r 3			IF	ID				1	,	4	EX	
11	sw (r6), r4			IF	ID						EX		
		_		_									

¿Qué diferencia habría si las unidades estuvieran segmentadas?

Las instrucciones estarían ejecutándose desde el ciclo 3 al ciclo 11 (en lugar del 12).

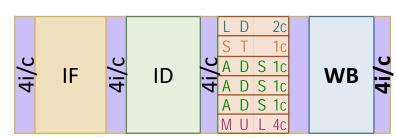


Ejercicio 1b. Ejecución incluyendo la última etapa de WB

PTC													
					Ciclo:	S							
TRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13
r1 , 0x1ac	IF	ID	EX	EX-	WB								
r2 , 0xc1f	IF	ID			EX	EX	WB						
r 3, r 0, r 0	IF	ID	EX-	1			WB						
r4, r2, r1	IF	ID			1		EX	EX	EX	EX	WB		
r3, r3, r4		IF	ID							-+	EX	WB	
r5, r0, 0x1ac		IF	ID	EX								WB	
r6, r0, 0xc1f		IF	ID	EX								WB	
r5, r5, #4		IF	ID		EX-	1						WB	
r6, r 6, #4			IF	ID	EX.							1	WB
(r5), r3			IF	ID								EX	WB
(r6), r4			IF	ID						/ /	EX		WB
	TRUCCIÓN r1, 0x1ac r2, 0xc1f r3, r0, r0 r4, r2, r1 r3, r3, r4 r5, r0, 0x1ac r6, r0, 0xc1f r5, r5, #4 r6, r6, #4 (r5), r3	TRUCCIÓN 1 1, 0x1ac IF 1, 0x1ac IF 12, 0xc1f IF 13, r0, r0 IF 14, r2, r1 IF 15, r3, r4 15, r0, 0x1ac 16, r0, 0xc1f 175, r5, #4 176, r6, #4 175, r3	STRUCCIÓN 1 2 r1, 0x1ac IF ID r2, 0xc1f IF ID r3, r0, r0 IF ID r4, r2, r1 IF ID r3, r3, r4 IF r5, r0, 0x1ac IF r6, r0, 0xc1f IF r5, r5, #4 IF r6, r6, #4 (r5), r3	STRUCCIÓN 1 2 3 r1, 0x1ac IF ID EX r2, 0xc1f IF ID EX r3, r0, r0 IF ID EX r4, r2, r1 IF ID r3, r3, r4 IF ID r5, r0, 0x1ac IF ID r6, r0, 0xc1f IF ID r5, r5, #4 IF ID r6, r6, #4 IF IF (r5), r3 IF	STRUCCIÓN 1 2 3 4 r1, 0x1ac IF ID EX EX- r2, 0xc1f IF ID EX- - r3, r0, r0 IF ID EX- - r4, r2, r1 IF ID IF ID r5, r0, 0x1ac IF ID EX r6, r0, 0xc1f IF ID EX r5, r5, #4 IF ID IF ID (r5), r3 IF ID ID IF ID	Ciclos STRUCCIÓN 1 2 3 4 5 r1, 0x1ac IF ID EX EX- r2, 0xc1f IF ID EX r3, r0, r0 IF ID EX r4, r2, r1 IF ID IF r3, r3, r4 IF ID EX r5, r0, 0x1ac IF ID EX r6, r0, 0xc1f IF ID EX r5, r5, #4 IF ID EX r6, r6, #4 IF ID EX (r5), r3 IF ID ID	TRUCCIÓN 1 2 3 4 5 6 r1, 0x1ac IF ID EX EX WB r2, 0xc1f IF ID EX EX r3, r0, r0 IF ID EX EX r4, r2, r1 IF ID IF ID IF ID EX IF ID IF IF ID IF <	TRUCCIÓN 1 2 3 4 5 6 7 r1, 0x1ac IF ID EX EX WB r2, 0xc1f IF ID EX EX WB r3, r0, r0 IF ID EX WB r4, r2, r1 IF ID EX EX r3, r3, r4 IF ID EX EX r5, r0, 0x1ac IF ID EX EX r5, r5, #4 IF ID EX ID EX r6, r6, #4 IF ID EX ID ID	TRUCCIÓN 1 2 3 4 5 6 7 8 r1, 0x1ac IF ID EX EX WB	Ciclos STRUCCIÓN 1 2 3 4 5 6 7 8 9 r1, 0x1ac IF ID EX EX WB	CTRUCCIÓN 1 2 3 4 5 6 7 8 9 10 r1, 0x1ac IF ID EX EX WB	CIRUCCIÓN 1 2 3 4 5 6 7 8 9 10 11 r1, 0x1ac IF ID EX EX WB .	CTRUCCIÓN 1 2 3 4 5 6 7 8 9 10 11 12 r1, 0x1ac IF ID EX EX WB

¿Cuánto tiempo tardaría el procesamiento de esta secuencia de instrucciones si se usara ROB y la etapa WB pudiera retirar del ROB 4 instrucciones por ciclo?

13 ciclos (suponemos que el almacenamiento en el ROB no consume ningún ciclo)

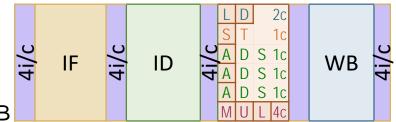


Ejercicio 1b. Ejecución con unidades segmentadas y la última etapa de WB

AC N PTC												
				Ci	<u>clos</u>							
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12
1 lw r1 , 0x1ac	IF	ID	EX	EX-	WB							
1 lw r2, 0xc1f	IF	ID		EX	EX	WB						
11 add r3, r0, r0	IF	ID	EX-	1		WB						
11 mul r4, r2, r1	IF	ID				EX	EX	EX	EX	WB		
11 add r3, r3, r4		IF	ID							EX	WB	
11 add r5, r0, 0x1ac		IF	ID	EX							WB	
11 add r6, r0, 0xc1f		IF	ID	EX							WB	
11 sub r 5, r 5, #4		IF	ID		EX-	1					WB	
11 sub r6, r6, #4			IF	ID	EX -	, ,		- '			1	WB
11 sw (r5), r3			IF	ID						7	EX	WB
11 sw (r6), r4			IF	ID						EX		WB
: Cuánto tiompo tardarí	0 0 0 0			<u> </u>	oto							

¿Cuánto tiempo tardaría el procesamiento de esta secuencia de instrucciones si las unidades estuvieran segmentadas y se usara ROB y la etapa WB pudiera retirar del ROB 4 instrucciones por ciclo?

12 ciclos (suponemos que el almacenamiento en el ROB no consume ningún ciclo)



Ejercicio 1c. Con ER distribuidas Captación (IF) y Decodificación (ID)

AC S PTC														
	Ciclos TRUCCIÓN 1 2 3 4 5 6 7 8 9 10 11 12 13 14 r1, 0x1ac IF ID IF ID IF ID IF ID IF IF ID IF IF													
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1 lw r1, 0x1ac	IF	ID												
1 lw r2, 0xc1f	IF	ID												
11 add r3, r0, r0	IF	ID												
00 mul r4, r2, r1	IF	ID												
00 add r3, r3, r4		IF	ID											
11 add r5, r0, 0x1ac		IF	ID											
11 add r 6, r 0, 0xc1f		IF	ID											
01 sub r 5, r 5, #4		IF	ID											
01 sub r6, r6, #4			IF	ID										
00 sw (r5), r3			IF	ID										
00 sw (r6), r4			IF	ID										
> Unidades: 1 LD (2 cicl	os), 1 §	ST (1 ci	clo), 3	ADD/S	SUB (1)	, 1 MU	L (4).					L D	2c	

- Unidades: 1 LD (2 ciclos), 1 ST (1 ciclo), 3 ADD/SUB (1), 1 MUL (4).
- Estación de reserva con 3 líneas para cada UF (en la 1ª columna se nota con colores las ER). Envío desde ER ordenada.
- Suponemos unidades funcionales segmentadas. Suponemos válido el contenido de un registro la primera vez que se usan
- > WAW y WAR resueltos por **renombrado** (los colores en registros representan renombrado y destacan los RAW)

ID

Las instrucciones decodificadas, que estarán en las ventanas de instrucciones, se pueden emitir si tienen los operandos disponibles y hay unidad disponible para ejecutarlas.

Ejercicio 1c. Ejecución suponiendo unidades segmentadas

AC MAC												
<u>Ciclos</u>												
INSTRUCCIÓN	1	2	3	4	5	6	7	8	9	10	11	12
1 lw r1, 0x1ac	IF	ID	EX	EX-	1							
1 lw r2, 0xc1f	IF	ID		EX	EX							
11 add r3, r0, r0	IF	ID	EX-	1								
11 mul r4, r2, r1	IF	ID				EX	EX	EX	EX	_		
11 add r3, r3, r4		IF	ID							EX		
11 add r5, r0, 0x1ac		IF	ID	EX								
11 add r6, r0, 0xc1f		IF	ID	EX								
11 sub r5, r5, #4		IF	ID		EX-	1						
11 sub r6, r6, #4			IF	ID	EX -	,	,	,	,			
11 sw (r5), r3			IF	ID						77	EX	
11 sw (r6), r4			IF	ID								EX
0 . 1												

Se han repartido las instrucciones add y sub entre las 3 ER usando turno rotatorio. Con este reparto, la ejecución de todos las instrucciones, excepto la última, coincide con la del caso con estación centralizada 1b. Se pueden enviar a ejecutar a la vez instrucciones en distintas ER. El uso de las estaciones de reserva con envío ordenado hace que la 2ª instrucción sw no pueda

| L D 2c | S T 1c | A D S 1c | A D S 1c | M U L 4c | WB

ejecutarse antes que la primera.