



Problemas Tema 7:

Diseño segmentado del procesador

Fundamentos de computadores II

José Manuel Mendías Cuadros

Dpto. Arquitectura de Computadores y Automática

Universidad Complutense de Madrid





Problema 1a

- **Multiciclo** (resultado correcto en 12 ciclos):
 - $t1 = 22+5 = 27$
 - $t3 = 27+22 = 49$
 - $t4 = 27+15 = 42$

	1	2	3	4	5	6	7	8	9	10	11	12
<code>addi t1,t2,5</code>	IF	ID	EX	WB								
<code>add t3,t1,t2</code>					IF	ID	EX	WB				
<code>addi t4,t1,15</code>									IF	ID	EX	WB

Las instrucciones 2 y 3 leen $t1$ del BR en ciclos posteriores al ciclo en que la instrucción 1 actualiza $t1$ en el BR



Problema 1b

- **Segmentado sin gestión de conflictos** (resultado incorrecto en 7 ciclos):
 - $t1 = 22+5 = 27$
 - $t3 = 11+22 = 33$
 - $t4 = 11+15 = 26$

	1	2	3	4	5	6	7	8	9
<code>addi t1,t2,5</code>	IF	ID	EX	M	WB				
<code>add t3,t1,t2</code>		IF	ID	EX	M	WB			
<code>addi t4,t1,15</code>			IF	ID	EX	M	WB		

Existe conflicto de datos por $t1$:

Las instrucciones 2 y 3 leen $t1$ del BR en ciclos anteriores o iguales al ciclo en que la instrucción 1 actualiza $t1$ en el BR



Problema 1c

- Segmentado sin gestión de conflictos + Inserción de NOP (resultado correcto en 10 ciclos):
 - $t1 = 22+5 = 27$
 - $t3 = 27+22 = 49$
 - $t4 = 27+15 = 42$

	1	2	3	4	5	6	7	8	9	10
<code>addi t1, t2, 5</code>	IF	ID	EX	M	WB					
<code>nop</code>		IF	ID	EX	M	WB				
<code>nop</code>			IF	ID	EX	M	WB			
<code>nop</code>				IF	ID	EX	M	WB		
<code>add t3, t1, t2</code>					IF	ID	EX	M	WB	
<code>addi t4, t1, 15</code>						IF	ID	EX	M	WB

Las instrucciones 5 y 6 leen $t1$ del BR en ciclos posteriores al ciclo en que la instrucción 1 actualiza $t1$ en el BR



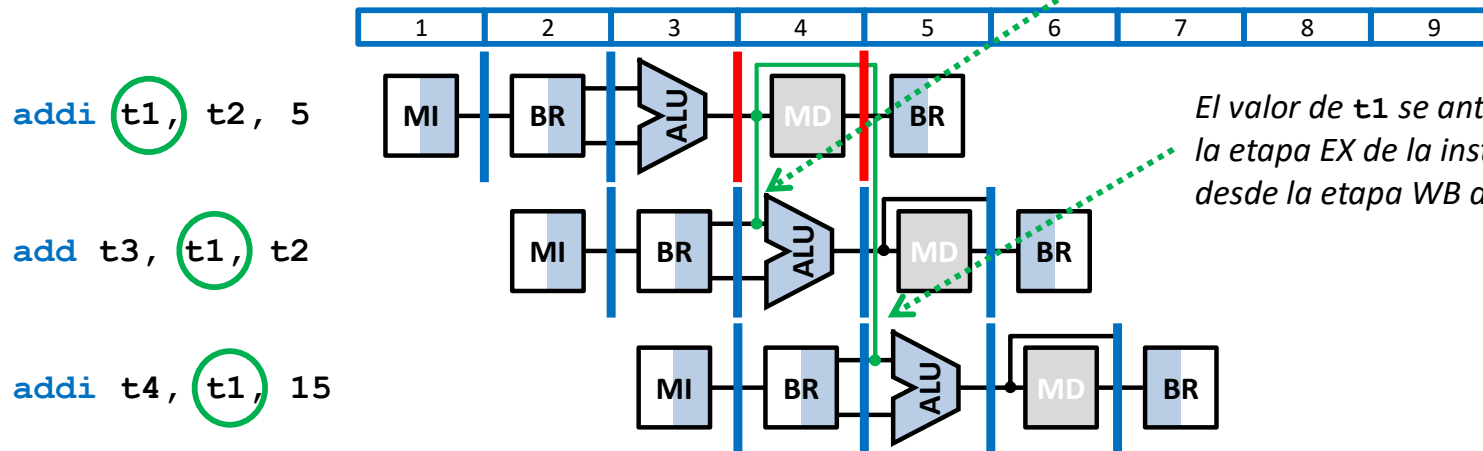
Problema 1d

■ **Segmentado** (resultado correcto en 7 ciclos):

- $t1 = 22+5 = 27$
- $t3 = (22+5)+22 = 49$
- $t4 = (22+5)+15 = 42$

El valor de t1 se anticipa a la etapa EX de la instrucción 2 desde la etapa MEM de la instrucción 1

El valor de t1 se anticipa a la etapa EX de la instrucción 3 desde la etapa WB de la instrucción 1



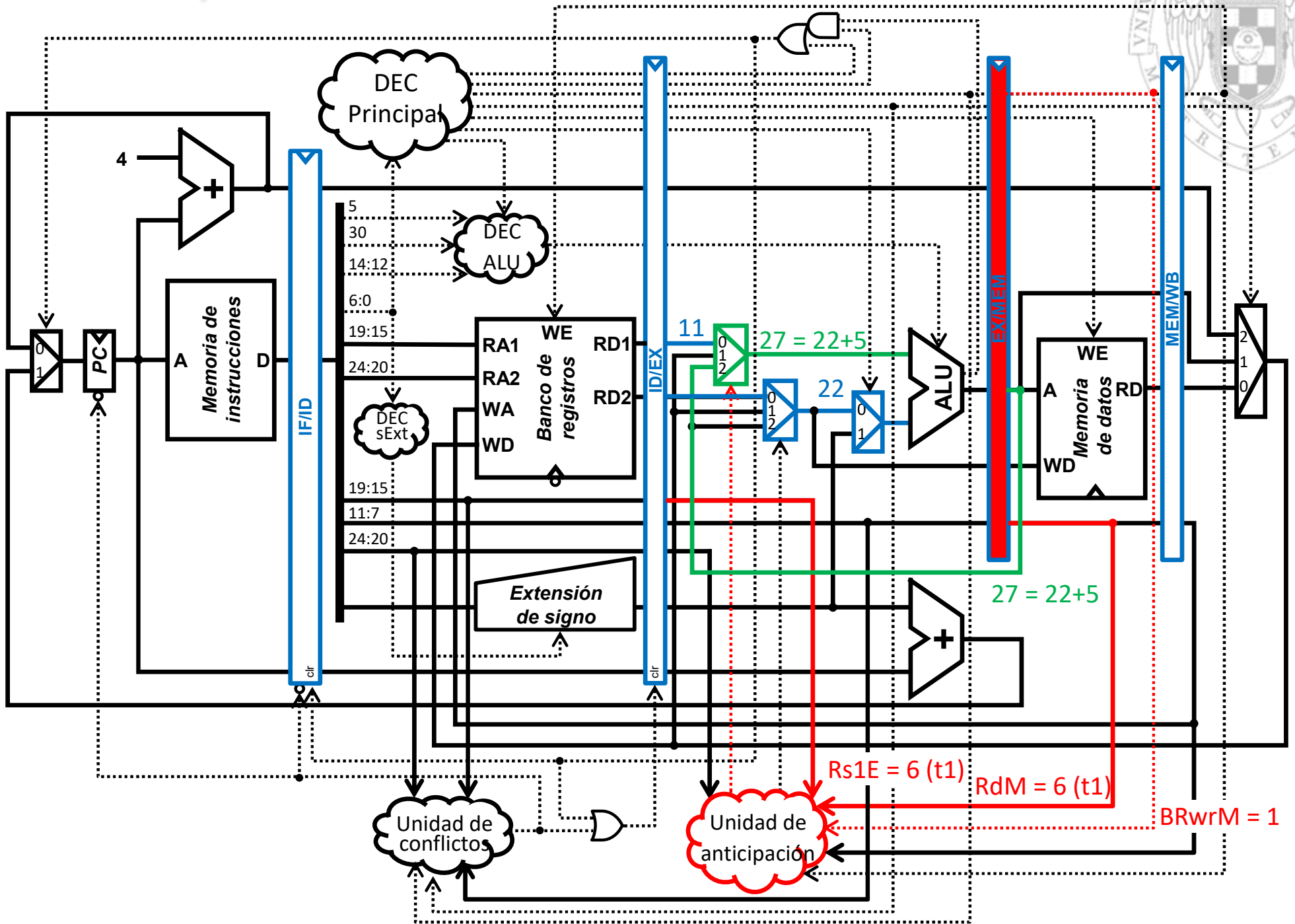
	1	2	3	4	5	6	7	8	9
<code>addi t1, t2, 5</code>	IF	ID	EX	M	WB				
<code>add t3, t1, t2</code>		IF	ID	EX	M	WB			
<code>addi t4, t1, 15</code>			IF	ID	EX	M	WB		

Problema 1d

anticipación en el ciclo 4

add t3,t1,t2

addi t1,t2,5



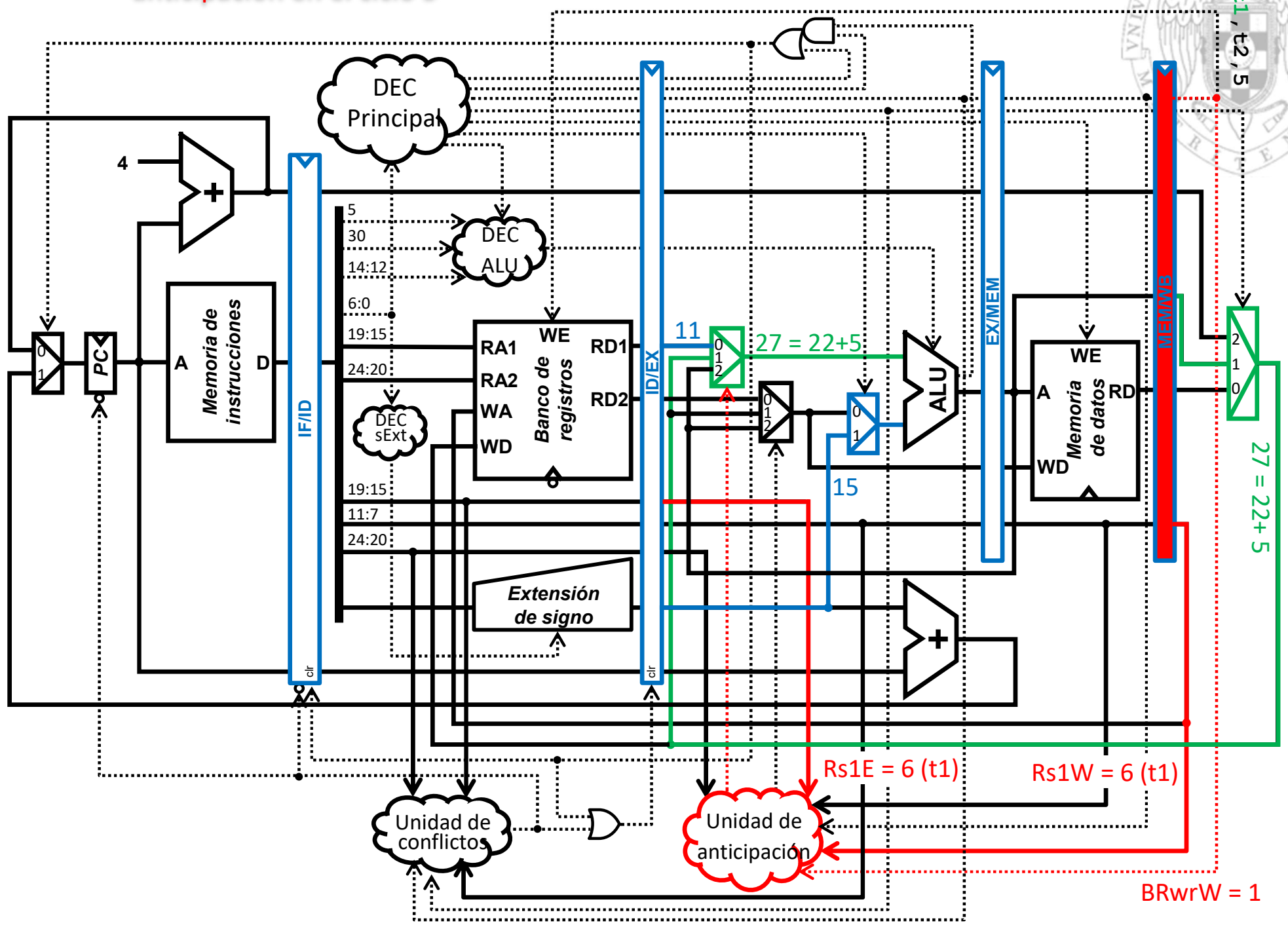
Problema 1d

anticipación en el ciclo 5

addi t4,t1,15

add t3,t1,t2

addi t1,t2,5





Problema 2a-2b

- **Multiciclo** (resultado correcto en 16 ciclos):
 - $t1 = 22+5 = 27$
 - $t3 = 27+22 = 49$
 - $t4 = 27+15 = 42$
 - $t5 = 27+27 = 54$
- **Segmentado sin gestión de conflictos** (resultado incorrecto en 8 ciclos):
 - $t1 = 22+5 = 27$
 - $t3 = 11+22 = 33$
 - $t4 = 11+15 = 26$
 - $t5 = 11+11 = 22$

	1	2	3	4	5	6	7	8
<code>addi t1,t2,5</code>	IF	ID	EX	M	WB			
<code>add t3,t1,t2</code>		IF	ID	EX	M	WB		
<code>addi t4,t1,15</code>			IF	ID	EX	M	WB	
<code>add t5,t1,t1</code>				IF	ID	EX	M	WB



Problema 2c

- Segmentado sin gestión de conflictos + Inserción de NOP (resultado correcto en 11 ciclos):

- $t3 = 27 + 22 = 49$
- $t4 = 27 + 15 = 42$
- $t5 = 27 + 27 = 54$

3 ciclos de retraso

	1	2	3	4	5	6	7	8	9	10	11
<code>addi t1, t2, 5</code>	IF	ID	EX	M	WB						
<code>nop</code>		IF	ID	EX	M	WB					
<code>nop</code>			IF	ID	EX	M	WB				
<code>nop</code>				IF	ID	EX	M	WB			
<code>add t3, t1, t2</code>					IF	ID	EX	M	WB		
<code>addi t4, t1, 15</code>						IF	ID	EX	M	WB	
<code>add t5, t1, t1</code>							IF	ID	EX	M	WB

3 NOP



Problema 2d

- Segmentado sin gestión de conflictos + Escritura de BR a mitad de ciclo (resultado incorrecto en 8 ciclos):
 - $t1 = 22+5 = 27$
 - $t3 = 11+22 = 33$
 - $t4 = 11+15 = 26$
 - $t5 = 27+27 = 54$

	1	2	3	4	5	6	7	8
<code>addi t1,t2,5</code>	IF	ID	EX	M	WB			
<code>add t3,t1,t2</code>		IF	ID	EX	M	WB		
<code>addi t4,t1,15</code>			IF	ID	EX	M	WB	
<code>add t5,t1,t1</code>				IF	^{t1} ID	EX	M	WB



Problema 2e

- Segmentado sin gestión de conflictos + Escritura de BR a mitad de ciclo + Inserción de NOP en el programa (resultado correcto en 10 ciclos):

- $t1 = 22 + 5 = 27$
- $t3 = 27 + 22 = 49$
- $t4 = 27 + 15 = 42$
- $t5 = 27 + 27 = 54$

2 ciclos de retraso

	1	2	3	4	5	6	7	8	9	10
<code>addi t1, t2, 5</code>	IF	ID	EX	M	WB					
<code>nop</code>		IF	ID	EX	M	WB				
<code>nop</code>			IF	ID	EX	M	WB			
<code>add t3, t1, t2</code>				IF	ID	EX	M	WB		
<code>addi t4, t1, 15</code>					IF	ID	EX	M	WB	
<code>add t5, t1, t1</code>						IF	ID	EX	M	WB

Diagram annotations: A green bracket above cycles 3-4 is labeled "2 ciclos de retraso". A green arrow points from the "WB" stage of the first instruction to the "IF" stage of the third instruction, labeled "t1". A green bracket on the left side of the first three rows is labeled "2 NOP".



Problema 2f

■ **Segmentado** (resultado correcto en 8 ciclos):

- $t1 = 22+5 = 27$
- $t3 = (22+5)+22 = 49$
- $t4 = (22+5)+15 = 42$
- $t5 = 27+27 = 54$

	1	2	3	4	5	6	7	8
<code>addi t1,t2,5</code>	IF	ID	EX	M	WB			
<code>add t3,t1,t2</code>		IF	ID	EX	M	WB		
<code>addi t4,t1,15</code>			IF	ID	EX	M	WB	
<code>add t5,t1,t1</code>				IF	ID	EX	M	WB

Diagram illustrating the pipeline stages for four instructions. The instructions are: `addi t1,t2,5`, `add t3,t1,t2`, `addi t4,t1,15`, and `add t5,t1,t1`. The pipeline stages are: IF (Instruction Fetch), ID (Instruction Decode), EX (Execute), M (Memory Access), and WB (Write Back). The diagram shows the progression of each instruction through the pipeline stages over 8 clock cycles. Green arrows labeled 't1' indicate data forwarding from the EX stage of the first instruction to the EX stages of the second, third, and fourth instructions.



Problema 3

	IF	ID	EX	M	WB
1	xor s1,s2,s3				
2	addi s0,s3,-4	xor s1,s2,s3			
3	lw s3,16(s7)	addi s0,s3,-4	xor s1,s2,s3		
4	sw s4,20(s1)	lw s3,16(s7)	addi s0,s3,-4	xor s1,s2,s3	
5					

	1	2	3	4	5	6	7	8	9
xor s1,s2,s3	IF	ID	EX	M					
addi s0,s3,-4		IF	ID	EX					
lw s3,16(s7)			IF	ID					
sw s4,20(s1)				IF					



Problema 3

	IF	ID	EX	M	WB
1	xor s1,s2,s3				
2	addi s0,s3,-4	xor s1,s2,s3			
3	lw s3,16(s7)	addi s0,s3,-4	xor s1,s2,s3		
4	sw s4,20(s1)	lw s3,16(s7)	addi s0,s3,-4	xor s1,s2,s3	
5	or t2,s0,s1	sw s4,20(s1)	lw s3,16(s7)	addi s0,s3,-4	xor s1,s2,s3

	1	2	3	4	5	6	7	8	9
xor s1,s2,s3	IF	ID	EX	M	WB				
addi s0,s3,-4		IF	ID	EX	M				
lw s3,16(s7)			IF	ID	EX				
sw s4,20(s1)				IF	ID				
or t2,s0,s1					IF				

s1
ID



Problema 3

	IF	ID	EX	M	WB
1	xor s1,s2,s3				
2	addi s0,s3,-4	xor s1,s2,s3			
3	lw s3,16(s7)	addi s0,s3,-4	xor s1,s2,s3		
4	sw s4,20(s1)	lw s3,16(s7)	addi s0,s3,-4	xor s1,s2,s3	
5	or t2,s0,s1	sw s4,20(s1)	lw s3,16(s7)	addi s0,s3,-4	xor s1,s2,s3

	1	2	3	4	5	6	7	8	9
xor s1,s2,s3	IF	ID	EX	M	WB				
addi s0,s3,-4		IF	ID	EX	M	WB			
lw s3,16(s7)			IF	ID	EX	M	WB		
sw s4,20(s1)				IF	ID	EX	M	WB	
or t2,s0,s1					IF	ID	EX	M	WB





Problema 4

	IF	ID	EX	M	WB
1	<code>addi s1,zero,11</code>				
2	<code>lw s2,25(s0)</code>	<code>addi s1,zero,11</code>			
3	<code>add s3,s3,s4</code>	<code>lw s2,25(s0)</code>	<code>addi s1,zero,11</code>		
4	<code>or s4,s1,s2</code>	<code>add s3,s3,s4</code>	<code>lw s2,25(s0)</code>	<code>addi s1,zero,11</code>	
5	<code>lw s5,16(s2)</code>	<code>or s4,s1,s2</code>	<code>add s3,s3,s4</code>	<code>lw s2,25(s0)</code>	<code>addi s1,zero,11</code>
6		<code>lw s5,16(s2)</code>	<code>or s4,s1,s2</code>	<code>add s3,s3,s4</code>	<code>lw s2,25(s0)</code>
7			<code>lw s5,16(s2)</code>	<code>or s4,s1,s2</code>	<code>add s3,s3,s4</code>

	1	2	3	4	5	6	7	8	9
<code>addi s1,zero,11</code>	IF	ID	EX	M	WB				
<code>lw s2,25(s0)</code>		IF	ID	EX	M	WB			
<code>add s3,s3,s4</code>			IF	ID	EX	M	WB		
<code>or s4,s1,s2</code>				IF	ID	EX	M	WB	
<code>lw s5,16(s2)</code>					IF	ID	EX	M	WB



Problema 5a

- Segmentado sin gestión de conflictos.

	1	2	3	4	5	6	7	8	9	10	11	12	13
<code>addi t1,t2,5</code>	IF	ID	EX	M	WB								
<code>nop</code>		IF	ID	EX	M	WB							
<code>nop</code>			IF	ID	EX	M	WB						
<code>nop</code>				IF	ID	EX	M	WB					
<code>add t3,t1,t2</code>					IF	ID	EX	M	WB				
<code>addi t4,t1,15</code>						IF	ID	EX	M	WB			
<code>nop</code>							IF	ID	EX	M	WB		
<code>nop</code>								IF	ID	EX	M	WB	
<code>add t5,t3,t2</code>									IF	ID	EX	M	WB



Problema 5b

- Segmentado sin gestión de conflictos + Escritura de BR a mitad de ciclo.

	1	2	3	4	5	6	7	8	9	10	11
<code>addi t1,t2,5</code>	IF	ID	EX	M	WB						
<code>nop</code>		IF	ID	EX	M	WB					
<code>nop</code>			IF	ID	EX	M	WB				
<code>add t3,t1,t2</code>				IF	^{t1} ID	EX	M	WB			
<code>addi t4,t1,15</code>					IF	ID	EX	M	WB		
<code>nop</code>						IF	ID	EX	M	WB	
<code>add t5,t3,t2</code>							IF	^{t3} ID	EX	M	WB



Problema 5c

- Segmentado.

	1	2	3	4	5	6	7	8
<code>addi t1,t2,5</code>	IF	ID	EX	M	WB			
<code>add t3,t1,t2</code>		IF	ID	EX	M	WB		
<code>addi t4,t1,15</code>			IF	ID	EX	M	WB	
<code>add t5,t3,t2</code>				IF	ID	EX	M	WB

Diagram illustrating the execution of four instructions in a segmented pipeline. The instructions are: `addi t1,t2,5`, `add t3,t1,t2`, `addi t4,t1,15`, and `add t5,t3,t2`. The pipeline stages are IF (Instruction Fetch), ID (Instruction Decode), EX (Execute), M (Memory Access), and WB (Write Back). The diagram shows the progression of each instruction through the pipeline stages over 8 clock cycles. Green arrows indicate data dependencies: `add t3,t1,t2` depends on the result of `addi t1,t2,5` (labeled `t1`), and `add t5,t3,t2` depends on the result of `add t3,t1,t2` (labeled `t3`). The `addi t4,t1,15` instruction does not depend on the results of the previous instructions.



Problema 6a

- Segmentado sin gestión de conflictos.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
add x7,x5,x8	IF	ID	EX	M	WB												
nop		IF	ID	EX	M	WB											
nop			IF	ID	EX	M	WB										
nop				IF	ID	EX	M	WB									
lw x6,8(x7)					IF	ID	EX	M	WB								
lw x5,0(x5)						IF	ID	EX	M	WB							
nop							IF	ID	EX	M	WB						
nop								IF	ID	EX	M	WB					
or x6,x7,x6									IF	ID	EX	M	WB				
nop										IF	ID	EX	M	WB			
nop											IF	ID	EX	M	WB		
nop												IF	ID	EX	M	WB	
sw x6,0(x7)													IF	ID	EX	M	WB



Problema 6b

- Segmentado sin gestión de conflictos + Escritura de BR a mitad de ciclo.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
<code>add x7, x5, x8</code>	IF	ID	EX	M	WB									
<code>nop</code>		IF	ID	EX	M	WB								
<code>nop</code>			IF	ID	EX	M	WB							
<code>lw x6, 8(x7)</code>				IF	ID	EX	M	WB						
<code>lw x5, 0(x5)</code>					IF	ID	EX	M	WB					
<code>nop</code>						IF	ID	EX	M	WB				
<code>or x6, x7, x6</code>							IF	ID	EX	M	WB			
<code>nop</code>								IF	ID	EX	M	WB		
<code>nop</code>									IF	ID	EX	M	WB	
<code>sw x6, 0(x7)</code>										IF	ID	EX	M	WB



Problema 6c

- Segmentado.

	1	2	3	4	5	6	7	8	9
<code>add x7, x5, x8</code>	IF	ID	EX	M	WB				
<code>lw x6, 8(x7)</code>		IF	ID	EX	M	WB			
<code>lw x5, 0(x5)</code>			IF	ID	EX	M	WB		
<code>or x6, x7, x6</code>				IF	ID	EX	M	WB	
<code>sw x6, 0(x7)</code>					IF	ID	EX	M	WB



Problema 6c

versión 15/01/24

Problemas Tema 7:
Diseño segmentado del procesador

FC-2

26

	1	2	3	4	5	6	7
<code>add x7, x5, x8</code>	IF						
<code>lw x6, 8(x7)</code>							
<code>lw x5, 0(x5)</code>							
<code>or x6, x7, x6</code>							
<code>sw x6, 0(x7)</code>							

Rs1E	00000						
Rs2E	00000						
RdM	00000						
BRwrM	0						
RdW	00000						
BRwrW	0						
ForwardA	00						
ForwardB	00						
clk							
# ciclo	1	2	3	4	5	6	7



Problema 6c

versión 15/01/24

Problemas Tema 7:
Diseño segmentado del procesador

FC-2

27

	1	2	3	4	5	6	7
<code>add x7, x5, x8</code>	IF	ID					
<code>lw x6, 8(x7)</code>		IF					
<code>lw x5, 0(x5)</code>							
<code>or x6, x7, x6</code>							
<code>sw x6, 0(x7)</code>							

Rs1E	00000	00000					
Rs2E	00000	00000					
RdM	00000	00000					
BRwrM	0	0					
RdW	00000	00000					
BRwrW	0	0					
ForwardA	00	00					
ForwardB	00	00					
clk							
# ciclo	1	2	3	4	5	6	7



Problema 6c

	1	2	3	4	5	6	7
<code>add x7, x5, x8</code>	IF	ID	EX				
<code>lw x6, 8(x7)</code>		IF	ID				
<code>lw x5, 0(x5)</code>			IF				
<code>or x6, x7, x6</code>							
<code>sw x6, 0(x7)</code>							

Rs1E	00000	00000	00101				
Rs2E	00000	00000	01000				
RdM	00000	00000	00000				
BRwrM	0	0	0				
RdW	00000	00000	00000				
BRwrW	0	0	0				
ForwardA	00	00	00				
ForwardB	00	00	00				
clk							
# ciclo	1	2	3	4	5	6	7



Problema 6c

	1	2	3	4	5	6	7
<code>add x7, x5, x8</code>	IF	ID	EX	M			
<code>lw x6, 8(x7)</code>		IF	ID	EX			
<code>lw x5, 0(x5)</code>			IF	ID			
<code>or x6, x7, x6</code>				IF			
<code>sw x6, 0(x7)</code>							

Rs1E	00000	00000	00101	00111			
Rs2E	00000	00000	01000	basura			
RdM	00000	00000	00000	00111			
BRwrM	0	0	0	1			
RdW	00000	00000	00000	00000			
BRwrW	0	0	0	0			
ForwardA	00	00	00	10			
ForwardB	00	00	00	basura			

clk

ciclo

1 2 3 4 5 6 7



Problema 6c

	1	2	3	4	5	6	7
<code>add x7, x5, x8</code>	IF	ID	EX	M	WB		
<code>lw x6, 8(x7)</code>		IF	ID	EX	M		
<code>lw x5, 0(x5)</code>			IF	ID	EX		
<code>or x6, x7, x6</code>				IF	ID		
<code>sw x6, 0(x7)</code>					IF		

Rs1E	00000	00000	00101	00111	00101		
Rs2E	00000	00000	01000	basura	basura		
RdM	00000	00000	00000	00111	00110		
BRwrM	0	0	0	1	1		
RdW	00000	00000	00000	00000	00111		
BRwrW	0	0	0	0	1		
ForwardA	00	00	00	10	00		
ForwardB	00	00	00	basura	basura		

clk

ciclo

1 2 3 4 5 6 7



Problema 6c

	1	2	3	4	5	6	7
<code>add x7, x5, x8</code>	IF	ID	EX	M	WB		
<code>lw x6, 8(x7)</code>		IF	ID	EX	M	WB	
<code>lw x5, 0(x5)</code>			IF	ID	EX	M	
<code>or x6, x7, x6</code>				IF	ID	EX	
<code>sw x6, 0(x7)</code>					IF	ID	

Rs1E	00000	00000	00101	00111	00101	00111	
Rs2E	00000	00000	01000	basura	basura	00110	
RdM	00000	00000	00000	00111	00110	00101	
BRwrM	0	0	0	1	1	1	
RdW	00000	00000	00000	00000	00111	00110	
BRwrW	0	0	0	0	1	1	
ForwardA	00	00	00	10	00	00	
ForwardB	00	00	00	basura	basura	01	
clk							
# ciclo	1	2	3	4	5	6	7



Problema 6c

	1	2	3	4	5	6	7
<code>add x7, x5, x8</code>	IF	ID	EX	M	WB		
<code>lw x6, 8(x7)</code>		IF	ID	EX	M	WB	
<code>lw x5, 0(x5)</code>			IF	ID	EX	M	WB
<code>or x6, x7, x6</code>				IF	ID	EX	M
<code>sw x6, 0(x7)</code>					IF	ID	EX

Note: A green arrow points from the 'M' stage of the 'or' instruction to the 'EX' stage of the 'sw' instruction, with the label 'x6' next to it.

Rs1E	00000	00000	00101	00111	00101	00111	00111	00111
Rs2E	00000	00000	01000	basura	basura	00110	00110	00110
RdM	00000	00000	00000	00111	00110	00101	00110	00110
BRwrM	0	0	0	1	1	1	1	1
RdW	00000	00000	00000	00000	00111	00110	00110	00101
BRwrW	0	0	0	0	1	1	1	1
ForwardA	00	00	00	10	00	00	00	00
ForwardB	00	00	00	basura	basura	01	10	10

clk

ciclo

1 2 3 4 5 6 7



Problema 7a

	IF	ID	EX	M	WB
1	<code>lw t1,0(t0)</code>				
2	<code>add t5,t2,t1</code>	<code>lw t1,0(t0)</code>			
3	<code>addi t0,t0,1</code>	<code>add t5,t2,t1</code>	<code>lw t1,0(t0)</code>		
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>lw t1,0(t0)</code>	IF	ID	EX						
<code>add t5,t2,t1</code>		IF	ID						
<code>addi t0,t0,1</code>			IF						





Problema 7a

	IF	ID	EX	M	WB
1	<code>lw t1,0(t0)</code>				
2	<code>add t5,t2,t1</code>	<code>lw t1,0(t0)</code>			
3	<code>addi t0,t0,1</code>	<code>add t5,t2,t1</code>	<code>lw t1,0(t0)</code>		
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>lw t1,0(t0)</code>	IF	ID	EX						
<code>add t5,t2,t1</code>		IF	ID						
<code>addi t0,t0,1</code>			IF						



Problema 7a

	IF	ID	EX	M	WB
1	<code>lw t1,0(t0)</code>				
2	<code>add t5,t2,t1</code>	<code>lw t1,0(t0)</code>			
3	<code>addi t0,t0,1</code> 	<code>add t5,t2,t1</code> 	<code>lw t1,0(t0)</code>		
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>lw t1,0(t0)</code>	IF	ID	EX						
<code>add t5,t2,t1</code>		IF	IDp						
<code>addi t0,t0,1</code>			IFp						



Problema 7a

	IF	ID	EX	M	WB
1	lw t1,0(t0)				
2	add t5,t2,t1	lw t1,0(t0)			
3	addi t0,t0,1	add t5,t2,t1	lw t1,0(t0)		
4	addi t0,t0,1	add t5,t2,t1		lw t1,0(t0)	
5					
6					
7					

	1	2	3	4	5	6	7	8	9
lw t1,0(t0)	IF	ID	EX	M					
add t5,t2,t1		IF	IDp	ID					
addi t0,t0,1			IFp	IF					



Problema 7a

	IF	ID	EX	M	WB
1	lw t1,0(t0)				
2	add t5,t2,t1	lw t1,0(t0)			
3	addi t0,t0,1 STOP	add t5,t2,t1 STOP	lw t1,0(t0)		
4	addi t0,t0,1	add t5,t2,t1	Cloud	lw t1,0(t0)	
5	sub t3,t5,t0	addi t0,t0,1	add t5,t2,t1	Cloud	lw t1,0(t0)
6					
7					

Note: Green arrows show the forwarding path from the EX stage of the second 'add t5,t2,t1' to the EX stage of the first 'add t5,t2,t1'. Red clouds indicate stalls in the EX stage of the second 'add t5,t2,t1' and the first 'lw t1,0(t0)'.

	1	2	3	4	5	6	7	8	9
lw t1,0(t0)	IF	ID	EX	M	WB				
add t5,t2,t1		IF	IDp	ID	EX				
addi t0,t0,1			IFp	IF	ID				
sub t3,t5,t0					IF				

Note: Green arrows show the forwarding path from the WB stage of the first 'lw t1,0(t0)' to the EX stage of the first 'add t5,t2,t1'.



Problema 7a

	IF	ID	EX	M	WB
1	lw t1,0(t0)				
2	add t5,t2,t1	lw t1,0(t0)			
3	addi t0,t0,1 STOP	add t5,t2,t1 STOP	lw t1,0(t0)		
4	addi t0,t0,1	add t5,t2,t1	☁	lw t1,0(t0)	
5	sub t3,t5,t0	addi t0,t0,1	add t5,t2,t1	☁	lw t1,0(t0)
6		sub t3,t5,t0	addi t0,t0,1	add t5,t2,t1	☁
7					

Note: Red clouds indicate stalls. Green arrows show the flow of the lw t1,0(t0) instruction from row 4 to row 5, and from row 5 to row 6. A red 'STOP' sign is present above the addi instructions in rows 3 and 4.

	1	2	3	4	5	6	7	8	9
lw t1,0(t0)	IF	ID	EX	M	WB				
add t5,t2,t1		IF	IDp	ID	EX	M			
addi t0,t0,1			IFp	IF	ID	EX			
sub t3,t5,t0					IF	ID			

Note: Red boxes highlight the first 7 columns. Green arrows show the flow of the lw t1,0(t0) instruction from row 4 to row 5, and from row 5 to row 6.



Problema 7a

	IF	ID	EX	M	WB
1	lw t1,0(t0)				
2	add t5,t2,t1	lw t1,0(t0)			
3	addi t0,t0,1 STOP	add t5,t2,t1 STOP	lw t1,0(t0)		
4	addi t0,t0,1	add t5,t2,t1		lw t1,0(t0)	
5	sub t3,t5,t0	addi t0,t0,1	add t5,t2,t1		lw t1,0(t0)
6		sub t3,t5,t0	addi t0,t0,1	add t5,t2,t1	
7			sub t3,t5,t0	addi t0,t0,1	add t5,t2,t1

Diagram illustrating pipeline stalls and forwarding. Red clouds indicate stalls. Green arrows show forwarding paths: t1 from EX3 to EX4, t1 from EX4 to EX5, t5 from EX6 to EX7, and t0 from EX6 to EX7.

	1	2	3	4	5	6	7	8	9
lw t1,0(t0)	IF	ID	EX	M	WB				
add t5,t2,t1		IF	IDp	ID	EX	M	WB		
addi t0,t0,1			IFp	IF	ID	EX	M		
sub t3,t5,t0					IF	ID	EX		

Diagram illustrating the pipeline stages for each instruction, including stalls (IDp, IFp) and forwarding (t1, t5, t0).



Problema 7a

	IF	ID	EX	M	WB
1	lw t1,0(t0)				
2	add t5,t2,t1	lw t1,0(t0)			
3	addi t0,t0,1	add t5,t2,t1	lw t1,0(t0)		
4	addi t0,t0,1	add t5,t2,t1		lw t1,0(t0)	
5	sub t3,t5,t0	addi t0,t0,1	add t5,t2,t1		lw t1,0(t0)
6		sub t3,t5,t0	addi t0,t0,1	add t5,t2,t1	
7			sub t3,t5,t0	addi t0,t0,1	add t5,t2,t1

Diagram illustrating data forwarding. Green arrows show the flow of data from the MEM stage of one instruction to the EX stage of a subsequent instruction. For example, the value of t1 from the MEM stage of instruction 4 is forwarded to the EX stage of instruction 5. Similarly, the value of t5 from the MEM stage of instruction 5 is forwarded to the EX stage of instruction 7. Red clouds indicate stalls in the EX stage of instructions 4, 5, and 6.

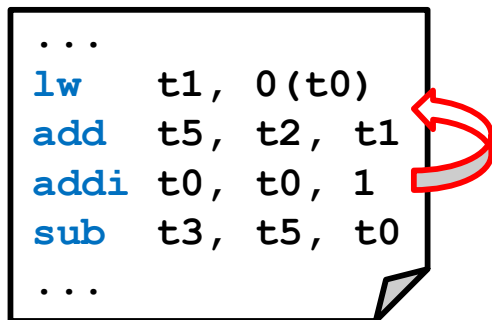
	1	2	3	4	5	6	7	8	9
lw t1,0(t0)	IF	ID	EX	M	WB				
add t5,t2,t1		IF	IDp	ID	EX	M	WB		
addi t0,t0,1			IFp	IF	ID	EX	M	WB	
sub t3,t5,t0					IF	ID	EX	M	WB

Diagram illustrating the pipeline stages for each instruction. Green arrows show the flow of data from the MEM stage of one instruction to the EX stage of a subsequent instruction. For example, the value of t1 from the MEM stage of instruction 1 is forwarded to the EX stage of instruction 2. Similarly, the value of t5 from the MEM stage of instruction 2 is forwarded to the EX stage of instruction 3, and the value of t0 from the MEM stage of instruction 3 is forwarded to the EX stage of instruction 4. Red clouds indicate stalls in the EX stage of instructions 4, 5, and 6.



Problema 7b

	IF	ID	EX	M	WB
1	lw t1,0(t0)				
2	addi t0,t0,1	lw t1,0(t0)			
3	add t5,t2,t1	addi t0,t0,1	lw t1,0(t0)		
4	sub t3,t5,t0	add t5,t2,t1	addi t0,t0,1	lw t1,0(t0)	
5		sub t3,t5,t0	add t5,t2,t1	addi t0,t0,1	lw t1,0(t0)
6			sub t3,t5,t0	add t5,t2,t1	addi t0,t0,1
7				sub t3,t5,t0	add t5,t2,t1



	1	2	3	4	5	6	7	8
lw t1,0(t0)	IF	ID	EX	M	WB			
addi t0,t0,1		IF	ID	EX	M	WB		
add t5,t2,t1			IF	ID	EX	M	WB	
sub t3,t5,t0				IF	ID	EX	M	WB

Diagram illustrating data hazards and forwarding. Green arrows show the flow of data from the MEM stage of the first instruction to the EX stage of the second instruction (labeled t1), from the MEM stage of the second instruction to the EX stage of the third instruction (labeled t0), and from the MEM stage of the third instruction to the EX stage of the fourth instruction (labeled t5).



Problema 8a-8b

- Segmentado sin gestión de conflictos.

	1	2	3	4	5	6	7	8	9	10
lw t1,0(t0)	IF	ID	EX	M	WB					
nop		IF	ID	EX	M	WB				
nop			IF	ID	EX	M	WB			
nop				IF	ID	EX	M	WB		
add t5,t2,t1					IF	ID	EX	M	WB	
add t2,t1,14						IF	ID	EX	M	WB

- Segmentado sin gestión de conflictos + Escritura de BR a mitad de ciclo.

	1	2	3	4	5	6	7	8	9
lw t1,0(t0)	IF	ID	EX	M	WB				
nop		IF	ID	EX	M	WB			
nop			IF	ID	EX	M	WB		
add t5,t2,t1				IF	ID	EX	M	WB	
add t2,t1,14					IF	ID	EX	M	WB

Note: A green arrow points from the 'WB' in cycle 5 of the first row to the 'ID' in cycle 5 of the fourth row, indicating a write-back to the branch register.



Problema 8c-8d

- Segmentado con gestión parcial de conflictos.

	1	2	3	4	5	6	7	8
<code>lw t1,0(t0)</code>	IF	ID	EX	M	WB			
<code>nop</code>		IF	ID	EX	M	WB		
<code>add t5,t2,t1</code>			IF	ID	EX	M	WB	
<code>add t2,t1,14</code>				IF	ID	EX	M	WB

Diagram illustrating partial conflict management. Green arrows show data dependencies: from the MEM stage of the first instruction to the EX stage of the second, and from the EX stage of the second to the ID stage of the third. The third instruction's ID stage is labeled with 't1'.

- Segmentado.

	1	2	3	4	5	6	7	8
<code>lw t1,0(t0)</code>	IF	ID	EX	M	WB			
<code>add t5,t2,t1</code>		IF	IDp	ID	EX	M	WB	
<code>add t2,t1,14</code>			IFp	IF	ID	EX	M	WB

Diagram illustrating full segmentation. Red text indicates stalls: 'IDp' (Instruction Delay) for the second instruction in cycle 3 and 'IFp' (Instruction Fetch) for the third instruction in cycle 3. Green arrows show data dependencies as in the previous diagram.



Problema 8e

	IF	ID	EX	M	WB
1	lw t1,0(t0)				
2	add t5,t2,t1	lw t1,0(t0)			
3	add t2,t1,14 STOP	add t5,t2,t1 STOP	lw t1,0(t0)		
4	add t2,t1,14	add t5,t2,t1	☁	lw t1,0(t0)	
5		add t2,t1,14	add t5,t2,t1	☁	lw t1,0(t0)
6			add t2,t1,14	add t5,t2,t1	☁
7				add t2,t1,14	add t5,t2,t1

	1	2	3	4	5	6	7	8
lw t1,0(t0)	IF	ID	EX	M	WB			
add t5,t2,t1		IF	IDp	ID	EX	M	WB	
add t2,t1,14			IFp	IF	ID	EX	M	WB



Problema 9

versión 15/01/24

	IF	ID	EX	M	WB
1	addi s1,zero,11				
2	lw s2,25(s1)	addi s1,zero,11			
3	lw s5,16(s2)	lw s2,25(s1)	addi s1,zero,11		
4	add s3,s2,s5 STOP	lw s5,16(s2) STOP	lw s2,25(s1)	addi s1,zero,11	
5	add s3,s2,s5	lw s5,16(s2)	Cloud	lw s2,25(s1)	addi s1,zero,11
6	or s4,s3,t4 STOP	add s3,s2,s5 STOP	lw s5,16(s2)	Cloud	lw s2,25(s1)
7	or s4,s3,t4	add s3,s2,s5	Cloud	lw s5,16(s2)	Cloud
8	and s2,s3,s4	or s4,s3,t4	add s3,s2,s5	Cloud	lw s5,16(s2)
9		and s2,s3,s4	or s4,s3,t4	add s3,s2,s5	Cloud
10			and s2,s3,s4	or s4,s3,t4	add s3,s2,s5

	1	2	3	4	5	6	7	8	9	10	11	12
addi s1,zero,11	IF	ID	EX	M	WB							
lw s2,25(s1)		IF	ID	EX	M	WB						
lw s5,16(s2)			IF	IDp	ID	EX	M	WB				
add s3,s2,s5				IFp	IF	IDp	ID	EX	M	WB		
or s4,s3,t4						IFp	IF	ID	EX	M	WB	
and s2,s3,s4								IF	ID	EX	M	WB



Problema 10a

	IF	ID	EX	M	WB
1	addi s1,zero,52				
2	addi s0,s1,-4	addi s1,zero,52			
3	lw s3,16(s0)	addi s0,s1,-4	addi s1,zero,52		
4	sw s3,20(s0)	lw s3,16(s0)	addi s0,s1,-4	addi s1,zero,52	
5	xor s2,s0,s3 STOP	sw s3,20(s0) STOP	lw s3,16(s0)	addi s0,s1,-4	addi s1,zero,52
6	xor s2,s0,s3	sw s3,20(s0)	☁	lw s3,16(s0)	addi s0,s1,-4
7	or s2,s2,s3	xor s2,s0,s3	sw s3,20(s0)	☁	lw s3,16(s0)

	1	2	3	4	5	6	7	8	9	10	11
addi s1,zero,52	IF	ID	EX	M	WB						
addi s0,s1,-4		IF	ID	EX	M	WB					
lw s3,16(s0)			IF	ID	EX	M	WB				
sw s3,20(s0)				IF	IDp	ID	EX	M	WB		
xor s2,s0,s3					IFp	IF	ID	EX	M	WB	
or s2,s2,s3							IF	ID	EX	M	WB



Problema 10b

	IF	ID	EX	M	WB
1	<code>addi s1,zero,52</code>				
2	<code>addi s0,s1,-4</code>	<code>addi s1,zero,52</code>			
3	<code>lw s3,16(s0)</code>	<code>addi s0,s1,-4</code>	<code>addi s1,zero,52</code>		
4	<code>sw s3,20(s0)</code>	<code>lw s3,16(s0)</code>	<code>addi s0,s1,-4</code>	<code>addi s1,zero,52</code>	
5	<code>xor s2,s0,s3</code>	<code>sw s3,20(s0)</code>	<code>lw s3,16(s0)</code>	<code>addi s0,s1,-4</code>	<code>addi s1,zero,52</code>
6	<code>or s2,s2,s3</code>	<code>xor s2,s0,s3</code>	<code>sw s3,20(s0)</code>	<code>lw s3,16(s0)</code>	<code>addi s0,s1,-4</code>
7		<code>or s2,s2,s3</code>	<code>xor s2,s0,s3</code>	<code>sw s3,20(s0)</code>	<code>lw s3,16(s0)</code>

	1	2	3	4	5	6	7	8	9	10	11
<code>addi s1,zero,52</code>	IF	ID	EX	M	WB						
<code>addi s0,s1,-4</code>		IF	ID	EX	M	WB					
<code>lw s3,16(s0)</code>			IF	ID	EX	M	WB				
<code>sw s3,20(s0)</code>				IF	ID	EX	M	WB			
<code>xor s2,s0,s3</code>					IF	ID	EX	M	WB		
<code>or s2,s2,s3</code>						IF	ID	EX	M	WB	



Problema 11a

	1	2	3	4	5	6	7	8	9	10	11	12	13
<code>lw s3,0(s4)</code>	IF	ID	EX	M	WB								
<code>add s5,s4,s3</code>		IF	IDp	ID	EX	M	WB						
<code>add s2,s3,s6</code>			IFp	IF	ID	EX	M	WB					
<code>and s1,s1,s2</code>					IF	ID	EX	M	WB				
<code>lw s5,0(t3)</code>						IF	ID	EX	M	WB			
<code>sw s5,0(t3)</code>							IF	IDp	ID	EX	M	WB	
<code>or s2,s2,s5</code>								IFp	IF	ID	EX	M	WB



Problema 11b

	1	2	3	4	5	6	7	8	9	10	11	12	13
<code>lw s3,0(s4)</code>	IF	ID	EX	M	WB								
<code>add s5,s4,s3</code>		IF	IDp	ID	EX	M	WB						
<code>add s2,s3,s6</code>			IFp	IF	ID	EX	M	WB					
<code>and s1,s1,s2</code>					IF	ID	EX	M	WB				
<code>lw s5,0(t3)</code>						IF	ID	EX	M	WB			
<code>sw s5,0(t3)</code>							IF	ID	EX	M	WB		
<code>or s2,s2,s5</code>								IF	ID	EX	M	WB	



Problema 12a

$s0 \neq 0$

- **Multiciclo** (resultado correcto en 19 ciclos):
 - $t1 = 11-1 = 10$
 - $t2 = 22-1 = 21$
 - $t3 = 10+1 = 11$
 - $t4 = 21+1 = 22$

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
<code>beq zero,s0,L1</code>	IF	ID	EX																
<code>addi t1,t1,-1</code>				IF	ID	EX	WB												
<code>addi t2,t2,-1</code>								IF	ID	EX	WB								
<code>L1:addi t3,t1,1</code>												IF	ID	EX	WB				
<code>addi t4,t2,1</code>																IF	ID	EX	WB



Problema 12b

$s0 \neq 0$

- Segmentado sin gestión de conflictos de control (resultado correcto en 9 ciclos):
 - $t1 = 11 - 1 = 10$
 - $t2 = 22 - 1 = 21$
 - $t3 = (11 - 1) + 1 = 11$
 - $t4 = (22 - 1) + 1 = 22$

	1	2	3	4	5	6	7	8	9
<code>beq zero, s0, L1</code>	IF	ID	EX	M	WB				
<code>addi t1, t1, -1</code>		IF	ID	EX	M	WB			
<code>addi t2, t2, -1</code>			IF	ID	EX	M	WB		
<code>L1: addi t3, t1, 1</code>				IF	ID	EX	M	WB	
<code>addi t4, t2, 1</code>					IF	ID	EX	M	WB



Problema 12c

$s0 \neq 0$

- Segmentado sin gestión de conflictos de control + Inserción de NOP (resultado correcto en 11 ciclos):

	1	2	3	4	5	6	7	8	9	10	11
<code>beq zero,s0,L1</code>	IF	ID	EX	M	WB						
<code>nop</code>		IF	ID	EX	M	WB					
<code>nop</code>			IF	ID	EX	M	WB				
<code>addi t1,t1,-1</code>				IF	ID	EX	M	WB			
<code>addi t2,t2,-1</code>					IF	ID	EX	M	WB		
<code>L1:addi t3,t1,1</code>						IF	ID	EX	M	WB	
<code>addi t4,t2,1</code>							IF	ID	EX	M	WB



Problema 12d

$s0 \neq 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 11 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero, s0, L1</code>				
2					
3					
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9	10	11
<code>beq zero, s0, L1</code>	IF										



Problema 12d

$s0 \neq 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 11 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>			
3					
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9	10	11
<code>beq zero,s0,L1</code>	IF	ID									
<code>addi t1,t1,-1</code>		IF									



Problema 12d

$s0 \neq 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 11 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code> <small>STOP</small>	<code>beq zero,s0,L1</code>			
3					
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9	10	11
<code>beq zero,s0,L1</code>	IF	ID									
<code>addi t1,t1,-1</code>		IFp									



Problema 12d

s0 ≠ 0

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 11 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code> STOP	<code>beq zero,s0,L1</code>			
3	<code>addi t1,t1,-1</code>		<code>beq zero,s0,L1</code>		
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9	10	11
<code>beq zero,s0,L1</code>	IF	ID	EX								
<code>addi t1,t1,-1</code>		IFp	IF								



Problema 12d

$s0 \neq 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 11 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>	no saltar		
3	<code>addi t1,t1,-1</code>		<code>beq zero,s0,L1</code>		
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9	10	11
<code>beq zero,s0,L1</code>	IF	ID	EX								
<code>addi t1,t1,-1</code>		IFp	IFp								



Problema 12d

$s0 \neq 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 11 ciclos):

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1 ^{STOP}	beq zero,s0,L1			
3	addi t1,t1,-1 ^{STOP}		beq zero,s0,L1		
4	addi t1,t1,-1			beq zero,s0,L1	
5					
6					
7					

	1	2	3	4	5	6	7	8	9	10	11
beq zero,s0,L1	IF	ID	EX	M							
addi t1,t1,-1		IFp	IFp	IF							



Problema 12d

$s0 \neq 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 11 ciclos):

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1 ^{STOP}	beq zero,s0,L1			
3	addi t1,t1,-1 ^{STOP}		beq zero,s0,L1		
4	addi t1,t1,-1			beq zero,s0,L1	
5	addi t2,t2,-1	addi t1,t1,-1			beq zero,s0,L1
6					
7					

	1	2	3	4	5	6	7	8	9	10	11
beq zero,s0,L1	IF	ID	EX	M	WB						
addi t1,t1,-1		IFp	IFp	IF	ID						
addi t2,t2,-1					IF						



Problema 12d

$s0 \neq 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 11 ciclos):

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1 ^{STOP}	beq zero,s0,L1			
3	addi t1,t1,-1 ^{STOP}		beq zero,s0,L1		
4	addi t1,t1,-1			beq zero,s0,L1	
5	addi t2,t2,-1	addi t1,t1,-1			beq zero,s0,L1
6	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1		
7					

	1	2	3	4	5	6	7	8	9	10	11
beq zero,s0,L1	IF	ID	EX	M	WB						
addi t1,t1,-1		IFp	IFp	IF	ID	EX					
addi t2,t2,-1					IF	ID					
L1:addi t3,t1,1						IF					



Problema 12d

$s0 \neq 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 11 ciclos):

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1 ^{STOP}	beq zero,s0,L1			
3	addi t1,t1,-1 ^{STOP}		beq zero,s0,L1		
4	addi t1,t1,-1			beq zero,s0,L1	
5	addi t2,t2,-1	addi t1,t1,-1			beq zero,s0,L1
6	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1		
7	addi t4,t2,1	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1	

	1	2	3	4	5	6	7	8	9	10	11
beq zero,s0,L1	IF	ID	EX	M	WB						
addi t1,t1,-1		IFp	IFp	IF	ID	EX	M				
addi t2,t2,-1					IF	ID	EX				
L1:addi t3,t1,1						IF	ID				
addi t4,t2,1							IF				



Problema 12d

$s0 \neq 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 11 ciclos):

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1	beq zero,s0,L1			
3	addi t1,t1,-1		beq zero,s0,L1		
4	addi t1,t1,-1			beq zero,s0,L1	
5	addi t2,t2,-1	addi t1,t1,-1			beq zero,s0,L1
6	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1		
7	addi t4,t2,1	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1	

	1	2	3	4	5	6	7	8	9	10	11
beq zero,s0,L1	IF	ID	EX	M	WB						
addi t1,t1,-1		IFp	IFp	IF	ID	EX	M	WB			
addi t2,t2,-1					IF	ID	EX	M	WB		
L1:addi t3,t1,1						IF	ID	EX	M	WB	
addi t4,t2,1							IF	ID	EX	M	WB



Problema 12e

$s0 \neq 0$

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2					
3					
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF								



Problema 12e

`s0 ≠ 0`

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>			
3					
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID							
<code>addi t1,t1,-1</code>		IF							



Problema 12e

$s0 \neq 0$

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>			
3	<code>addi t2,t2,-1</code>	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>		
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID	EX						
<code>addi t1,t1,-1</code>		IF	ID						
<code>addi t2,t2,-1</code>			IF						



Problema 12e

`s0 ≠ 0`

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>	no saltar: predicción correcta		
3	<code>addi t2,t2,-1</code>	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>		
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID	EX						
<code>addi t1,t1,-1</code>		IF	ID						
<code>addi t2,t2,-1</code>			IF						



Problema 12e

`s0 ≠ 0`

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>	no saltar: predicción correcta		
3	<code>addi t2,t2,-1</code>	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>		
4	<code>L1:addi t3,t1,1</code>	<code>addi t2,t2,-1</code>	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>	
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID	EX	M					
<code>addi t1,t1,-1</code>		IF	ID	EX					
<code>addi t2,t2,-1</code>			IF	ID					
<code>L1:addi t3,t1,1</code>				IF					



Problema 12e

$s0 \neq 0$

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>			
3	<code>addi t2,t2,-1</code>	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>		
4	<code>L1:addi t3,t1,1</code>	<code>addi t2,t2,-1</code>	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>	
5	<code>addi t4,t2,1</code>	<code>L1:addi t3,t1,1</code>	<code>addi t2,t2,-1</code>	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID	EX	M	WB				
<code>addi t1,t1,-1</code>		IF	ID	EX	M				
<code>addi t2,t2,-1</code>			IF	ID	EX				
<code>L1:addi t3,t1,1</code>				IF	ID				
<code>addi t4,t2,1</code>					IF				



Problema 12e

s0 ≠ 0

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1	beq zero,s0,L1			
3	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1		
4	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1	
5	addi t4,t2,1	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1
6		addi t4,t2,1	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1
7					

	1	2	3	4	5	6	7	8	9
beq zero,s0,L1	IF	ID	EX	M	WB				
addi t1,t1,-1		IF	ID	EX	M	WB			
addi t2,t2,-1			IF	ID	EX	M			
L1:addi t3,t1,1				IF	ID	EX			
addi t4,t2,1					IF	ID			



Problema 12e

$s0 \neq 0$

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1	beq zero,s0,L1			
3	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1		
4	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1	
5	addi t4,t2,1	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1
6		addi t4,t2,1	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1
7			addi t4,t2,1	L1:addi t3,t1,1	addi t2,t2,-1

	1	2	3	4	5	6	7	8	9
beq zero,s0,L1	IF	ID	EX	M	WB				
addi t1,t1,-1		IF	ID	EX	M	WB			
addi t2,t2,-1			IF	ID	EX	M	WB		
L1:addi t3,t1,1				IF	ID	EX	M		
addi t4,t2,1					IF	ID	EX		



Problema 12e

$s0 \neq 0$

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1	beq zero,s0,L1			
3	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1		
4	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1	
5	addi t4,t2,1	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1
6		addi t4,t2,1	L1:addi t3,t1,1	addi t2,t2,-1	addi t1,t1,-1
7			addi t4,t2,1	L1:addi t3,t1,1	addi t2,t2,-1

	1	2	3	4	5	6	7	8	9
beq zero,s0,L1	IF	ID	EX	M	WB				
addi t1,t1,-1		IF	ID	EX	M	WB			
addi t2,t2,-1			IF	ID	EX	M	WB		
L1:addi t3,t1,1				IF	ID	EX	M	WB	
addi t4,t2,1					IF	ID	EX	M	WB



Problema 12f

`s0 = 0`

- **Multiciclo** (resultado correcto en 11 ciclos):
 - `t1 = 11`
 - `t2 = 22`
 - `t3 = 11+1 = 12`
 - `t4 = 22+1 = 23`

	1	2	3	4	5	6	7	8	9	10	11
<code>beq zero, s0, L1</code>	IF	ID	EX								
<code>L1: addi t3, t1, 1</code>				IF	ID	EX	WB				
<code>addi t4, t2, 1</code>								IF	ID	EX	WB

La instrucción a ejecutar tras `beq` se lee de memoria en un ciclo posterior al ciclo en que decide si debe saltarse o no



Problema 12g

s0 = 0

- Segmentado sin gestión de conflictos de control (resultado incorrecto en 9 ciclos):

- t1 = 11-1 = 10
- t2 = 22-1 = 21
- t3 = (11-1)+1 = 11
- t4 = (22-1)+1 = 22

	1	2	3	4	5	6	7	8	9
<code>beq zero, s0, L1</code>	IF	ID	EX	M	WB				
<code>addi t1, t1, -1</code>		IF	ID	EX	M	WB			
<code>addi t2, t2, -1</code>			IF	ID	EX	M	WB		
<code>L1: addi t3, t1, 1</code>				IF	ID	EX	M	WB	
<code>addi t4, t2, 1</code>					IF	ID	EX	M	WB

Existe conflicto de control:

Se leen instrucciones de memoria en ciclos anteriores o iguales al ciclo en que la instrucción `beq` decide si debe saltarse o no



Problema 12h

$s0 = 0$

- Segmentado sin gestión de conflictos de control + Inserción de NOP (resultado correcto en 9 ciclos):

2 ciclos de retraso

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID	EX	M	WB				
<code>nop</code>		IF	ID	EX	M	WB			
<code>nop</code>			IF	ID	EX	M	WB		
<code>L1:addi t3,t1,1</code>				IF	ID	EX	M	WB	
<code>addi t4,t2,1</code>					IF	ID	EX	M	WB

2 NOP



Problema 12i

$s0 = 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 9 ciclos).

	IF	ID	EX	M	WB
1	<code>beq zero, s0, L1</code>				
2					
3					
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero, s0, L1</code>	IF								



Problema 12i

$s0 = 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 9 ciclos).

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>			
3					
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID							
<code>addi t1,t1,-1</code>		IF							



Problema 12i

$s0 = 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 9 ciclos).

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code> <small>STOP</small>	<code>beq zero,s0,L1</code>			
3					
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID							
<code>addi t1,t1,-1</code>		IFp							



Problema 12i

$s0 = 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 9 ciclos).

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>			
3	<code>addi t1,t1,-1</code>		<code>beq zero,s0,L1</code>		
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID	EX						
<code>addi t1,t1,-1</code>		IFp	IF						



Problema 12i

$s0 = 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 9 ciclos).

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code> <small>STOP</small>	<code>beq zero,s0,L1</code>	salto		
3	<code>addi t1,t1,-1</code> <small>STOP</small>		<code>beq zero,s0,L1</code>		
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID	EX						
<code>addi t1,t1,-1</code>		IFp	IFp						



Problema 12i

$s0 = 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 9 ciclos).

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code> <small>STOP</small>	<code>beq zero,s0,L1</code>			
3	<code>addi t1,t1,-1</code> <small>STOP</small>		<code>beq zero,s0,L1</code>		
4	<code>L1:addi t3,t1,1</code>			<code>beq zero,s0,L1</code>	
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID	EX	M					
<code>addi t1,t1,-1</code>		IFp	IFp	IF					
<code>L1:addi t3,t1,1</code>				IF					



Problema 12i

$s0 = 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 9 ciclos).

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1 ^{STOP}	beq zero,s0,L1			
3	addi t1,t1,-1 ^{STOP}		beq zero,s0,L1		
4	L1:addi t3,t1,1			beq zero,s0,L1	
5	addi t4,t2,1	L1:addi t3,t1,1			beq zero,s0,L1
6					
7					

	1	2	3	4	5	6	7	8	9
beq zero,s0,L1	IF	ID	EX	M	WB				
addi t1,t1,-1		IFp	IFp	IF					
L1:addi t3,t1,1				IF	ID				
addi t4,t2,1					IF				



Problema 12i

$s0 = 0$

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 9 ciclos).

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1 ^{STOP}	beq zero,s0,L1			
3	addi t1,t1,-1 ^{STOP}		beq zero,s0,L1		
4	L1:addi t3,t1,1			beq zero,s0,L1	
5	addi t4,t2,1	L1:addi t3,t1,1			beq zero,s0,L1
6		addi t4,t2,1	L1:addi t3,t1,1		
7					

	1	2	3	4	5	6	7	8	9
beq zero,s0,L1	IF	ID	EX	M	WB				
addi t1,t1,-1		IFp	IFp	IF					
L1:addi t3,t1,1				IF	ID	EX			
addi t4,t2,1					IF	ID			



Problema 12i

s0 = 0

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 9 ciclos).

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1 ^{STOP}	beq zero,s0,L1			
3	addi t1,t1,-1 ^{STOP}		beq zero,s0,L1		
4	L1:addi t3,t1,1			beq zero,s0,L1	
5	addi t4,t2,1	L1:addi t3,t1,1			beq zero,s0,L1
6		addi t4,t2,1	L1:addi t3,t1,1		
7			addi t4,t2,1	L1:addi t3,t1,1	

	1	2	3	4	5	6	7	8	9
beq zero,s0,L1	IF	ID	EX	M	WB				
addi t1,t1,-1		IFp	IFp	IF					
L1:addi t3,t1,1				IF	ID	EX	M		
addi t4,t2,1					IF	ID	EX		



Problema 12i

s0 = 0

- Segmentado con gestión de conflictos de control por parada (resultado correcto en 9 ciclos).

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1 ^{STOP}	beq zero,s0,L1			
3	addi t1,t1,-1 ^{STOP}		beq zero,s0,L1		
4	L1:addi t3,t1,1			beq zero,s0,L1	
5	addi t4,t2,1	L1:addi t3,t1,1			beq zero,s0,L1
6		addi t4,t2,1	L1:addi t3,t1,1		
7			addi t4,t2,1	L1:addi t3,t1,1	

	1	2	3	4	5	6	7	8	9
beq zero,s0,L1	IF	ID	EX	M	WB				
addi t1,t1,-1		IFp	IFp	IF					
L1:addi t3,t1,1				IF	ID	EX	M	WB	
addi t4,t2,1					IF	ID	EX	M	WB



Problema 12j

$s0 = 0$

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero, s0, L1</code>				
2					
3					
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero, s0, L1</code>	IF								



Problema 12j

$s0 = 0$

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>			
3					
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID							
<code>addi t1,t1,-1</code>		IF							



Problema 12j

$s0 = 0$

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>			
3	<code>addi t2,t2,-1</code>	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>		
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID	EX						
<code>addi t1,t1,-1</code>		IF	ID						
<code>addi t2,t2,-1</code>			IF						



Problema 12j

$s0 = 0$

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	<code>beq zero,s0,L1</code>				
2	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>	salto: predicción fallida		
3	<code>addi t2,t2,-1</code>	<code>addi t1,t1,-1</code>	<code>beq zero,s0,L1</code>		
4					
5					
6					
7					

	1	2	3	4	5	6	7	8	9
<code>beq zero,s0,L1</code>	IF	ID	EX						
<code>addi t1,t1,-1</code>		IF	ID						
<code>addi t2,t2,-1</code>			IF						



Problema 12j

$s0 = 0$

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1	beq zero,s0,L1			
3	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1		
4	L1:addi t3,t1,1			beq zero,s0,L1	
5					
6					
7					

	1	2	3	4	5	6	7	8	9
beq zero,s0,L1	IF	ID	EX	M					
addi t1,t1,-1		IF	ID	EX					
addi t2,t2,-1			IF	EX					
L1:addi t3,t1,1				IF					



Problema 12j

$s0 = 0$

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1	beq zero,s0,L1			
3	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1		
4	L1:addi t3,t1,1			beq zero,s0,L1	
5	addi t4,t2,1	L1:addi t3,t1,1			beq zero,s0,L1
6					
7					

	1	2	3	4	5	6	7	8	9
beq zero,s0,L1	IF	ID	EX	M	WB				
addi t1,t1,-1		IF	ID	EX					
addi t2,t2,-1			IF	EX					
L1:addi t3,t1,1				IF	ID				
addi t4,t2,1					IF				



Problema 12j

s0 = 0

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1	beq zero,s0,L1			
3	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1		
4	L1:addi t3,t1,1			beq zero,s0,L1	
5	addi t4,t2,1	L1:addi t3,t1,1			beq zero,s0,L1
6		addi t4,t2,1	L1:addi t3,t1,1		
7					

	1	2	3	4	5	6	7	8	9
beq zero,s0,L1	IF	ID	EX	M	WB				
addi t1,t1,-1		IF	ID						
addi t2,t2,-1			IF						
L1:addi t3,t1,1				IF	ID	EX			
addi t4,t2,1					IF	ID			



Problema 12j

$s0 = 0$

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1	beq zero,s0,L1			
3	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1		
4	L1:addi t3,t1,1			beq zero,s0,L1	
5	addi t4,t2,1	L1:addi t3,t1,1			beq zero,s0,L1
6		addi t4,t2,1	L1:addi t3,t1,1		
7			addi t4,t2,1	L1:addi t3,t1,1	

	1	2	3	4	5	6	7	8	9
beq zero,s0,L1	IF	ID	EX	M	WB				
addi t1,t1,-1		IF	ID						
addi t2,t2,-1			IF						
L1:addi t3,t1,1				IF	ID	EX	M		
addi t4,t2,1					IF	ID	EX		



Problema 12j

$s0 = 0$

- Segmentado (con predicción de salto no tomado)
(resultado correcto en 9 ciclos):

	IF	ID	EX	M	WB
1	beq zero,s0,L1				
2	addi t1,t1,-1	beq zero,s0,L1			
3	addi t2,t2,-1	addi t1,t1,-1	beq zero,s0,L1		
4	L1:addi t3,t1,1			beq zero,s0,L1	
5	addi t4,t2,1	L1:addi t3,t1,1			beq zero,s0,L1
6		addi t4,t2,1	L1:addi t3,t1,1		
7			addi t4,t2,1	L1:addi t3,t1,1	

	1	2	3	4	5	6	7	8	9
beq zero,s0,L1	IF	ID	EX	M	WB				
addi t1,t1,-1		IF	ID						
addi t2,t2,-1			IF						
L1:addi t3,t1,1				IF	ID	EX	M	WB	
addi t4,t2,1					IF	ID	EX	M	WB



Problema 13

	IF	ID	EX	M	WB
1	<code>jal x0,L1</code>				
2	<code>addi t1,x0,5</code>	<code>jal x0,L1</code>			
3	<code>add t3,t1,t2</code>	<code>addi t1,x0,5</code>	<code>jal x0,L1</code>		
4					
5					
6					
7					
8					

	1	2	3	4	5	6	7	8
<code>jal x0,L1</code>	IF	ID	EX					
<code>addi t1,x0,5</code>		IF	ID					
<code>add t3,t1,t2</code>			IF					



Problema 13

	IF	ID	EX	M	WB
1	<code>jal x0,L1</code>				
2	<code>addi t1,x0,5</code>	<code>jal x0,L1</code>	salto: predicción fallida		
3	<code>add t3,t1,t2</code>	<code>addi t1,x0,5</code>	<code>jal x0,L1</code>		
4					
5					
6					
7					
8					

	1	2	3	4	5	6	7	8
<code>jal x0,L1</code>	IF	ID	EX					
<code>addi t1,x0,5</code>		IF	ID					
<code>add t3,t1,t2</code>			IF					



Problema 13

	IF	ID	EX	M	WB
1	jal x0,L1				
2	addi t1,x0,5	jal x0,L1			
3	add t3,t1,t2	addi t1,x0,5	jal x0,L1		
4	L1:sw t4,0(t3)			jal x0,L1	
5		L1:sw t4,0(t3)			jal x0,L1
6					
7					
8					

	1	2	3	4	5	6	7	8
jal x0,L1	IF	ID	EX	M	WB			
addi t1,x0,5		IF	ID					
add t3,t1,t2			IF					
L1:sw t4,0(t3)				IF	ID			



Problema 13

	IF	ID	EX	M	WB
1	jal x0,L1				
2	addi t1,x0,5	jal x0,L1			
3	add t3,t1,t2	addi t1,x0,5	jal x0,L1		
4	L1:sw t4,0(t3)			jal x0,L1	
5		L1:sw t4,0(t3)			jal x0,L1
6			L1:sw t4,0(t3)		
7					
8					

	1	2	3	4	5	6	7	8
jal x0,L1	IF	ID	EX	M	WB			
addi t1,x0,5		IF	ID					
add t3,t1,t2			IF					
L1:sw t4,0(t3)				IF	ID	EX		



Problema 13

	IF	ID	EX	M	WB
1	jal x0,L1				
2	addi t1,x0,5	jal x0,L1			
3	add t3,t1,t2	addi t1,x0,5	jal x0,L1		
4	L1:sw t4,0(t3)			jal x0,L1	
5		L1:sw t4,0(t3)			jal x0,L1
6			L1:sw t4,0(t3)		
7				L1:sw t4,0(t3)	
8					

	1	2	3	4	5	6	7	8
jal x0,L1	IF	ID	EX	M	WB			
addi t1,x0,5		IF	ID					
add t3,t1,t2			IF					
L1:sw t4,0(t3)				IF	ID	EX	M	



Problema 13

	IF	ID	EX	M	WB
1	jal x0,L1				
2	addi t1,x0,5	jal x0,L1			
3	add t3,t1,t2	addi t1,x0,5	jal x0,L1		
4	L1:sw t4,0(t3)			jal x0,L1	
5		L1:sw t4,0(t3)			jal x0,L1
6			L1:sw t4,0(t3)		
7				L1:sw t4,0(t3)	
8					L1:sw t4,0(t3)

	1	2	3	4	5	6	7	8
jal x0,L1	IF	ID	EX	M	WB			
addi t1,x0,5		IF	ID					
add t3,t1,t2			IF					
L1:sw t4,0(t3)				IF	ID	EX	M	WB



Problema 14

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
<code>addi s0,x0,0</code>	IF	ID	EX	M	WB											
<code>L1:beq t0,x0,L2</code>		IF	ID	EX	M	WB										
<code>add s0,s0,t0</code>			IF	ID	EX	M	WB									
<code>addi t0,t0,-1</code>				IF	ID	EX	M	WB								
<code>jal x0, L1</code>					IF	ID	EX	M	WB							
<code>L2:add s0,s0,s0</code>						IF	ID	EX								
<code>sw s0,0(gp)</code>							IF	EX								
<code>L1:beq t0,x0,L2</code>								IF	ID	EX	M	WB				
<code>add s0,s0,t0</code>									IF	ID	EX					
<code>addi t0,t0,-1</code>										IF	EX					
<code>L2:add s0,s0,s0</code>											IF	ID	EX	M	WB	
<code>sw s0,0(gp)</code>												IF	ID	EX	M	WB

→

s0

→

s0

s0



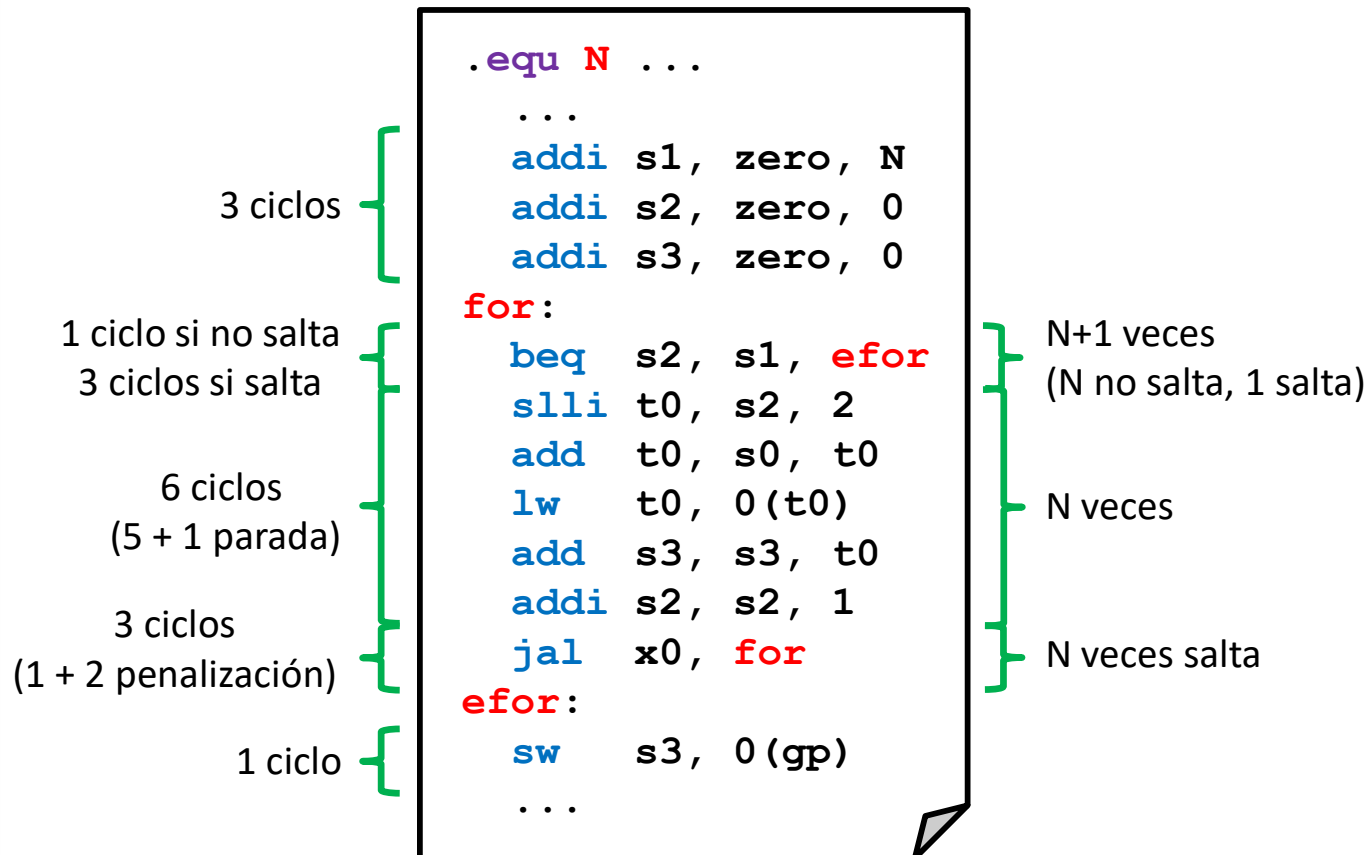
Problema 15

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
<code>addi s0,x0,0</code>	IF	ID	EX	M	WB															
<code>lw t0,4(gp)</code>		IF	ID	EX	M	WB														
<code>L1:beq t0,x0,L2</code>			IF	IDp	ID	EX	M	WB												
<code>add s0,s0,t0</code>				IFp	IF	ID	EX	M	WB											
<code>addi t0,t0,-1</code>						IF	ID	EX	M	WB										
<code>jal x0,L1</code>							IF	ID	EX	M	WB									
<code>L2:lw s1,0(gp)</code>								IF	ID	×										
<code>add s1,s0,s1</code>									IF	×										
<code>L1:beq t0,x0,L2</code>										IF	ID	EX	M	WB						
<code>add s0,s0,t0</code>											IF	ID	×							
<code>addi t0,t0,-1</code>												IF	×							
<code>L2:lw s1,0(gp)</code>													IF	ID	EX	M	WB			
<code>add s1,s0,s1</code>														IF	IDp	ID	EX	M	WB	
<code>sw s1,0(gp)</code>															IFp	IF	ID	EX	M	WB





Problema 16



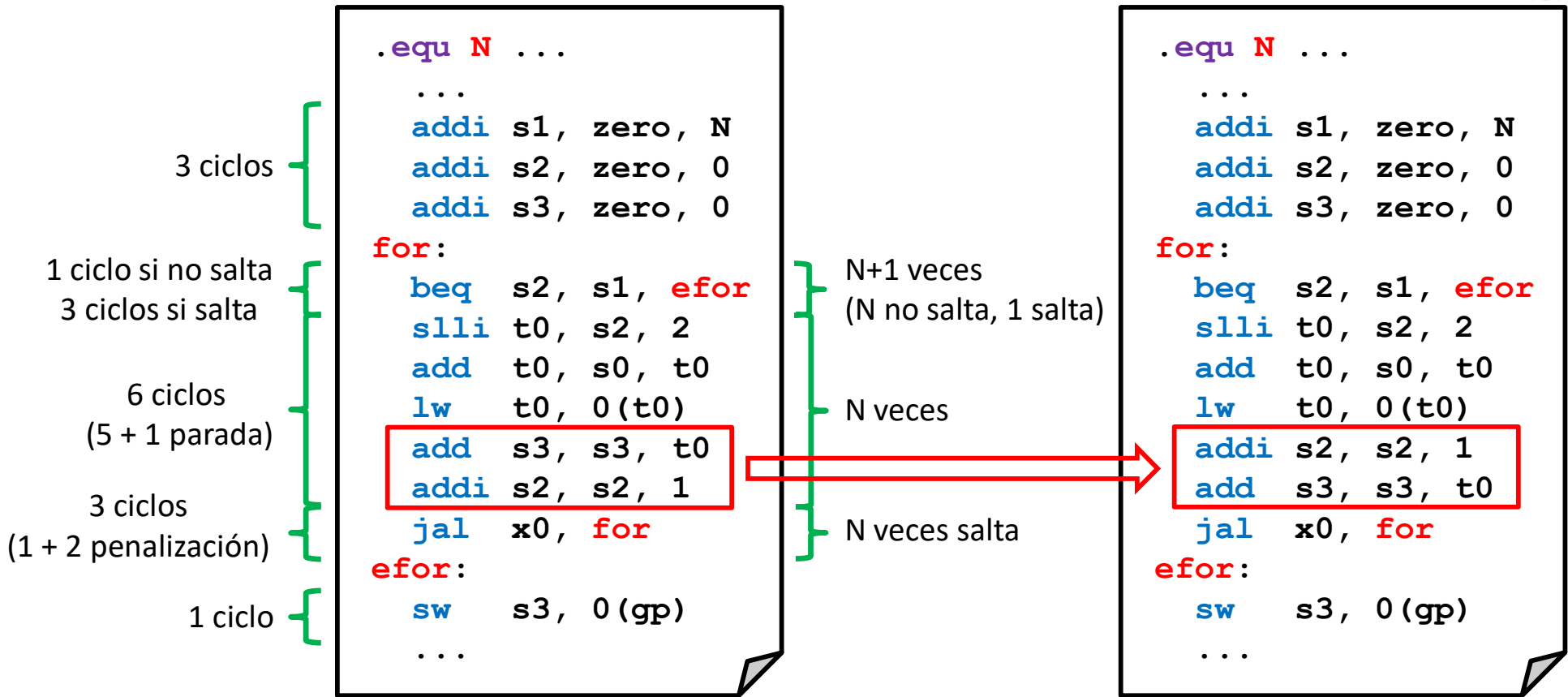
$$3 + (7 \times N + 1) + 1 = 7 \times N + 5 \text{ instrucciones ejecutadas}$$

$$3 + (1 \times N + 3 \times 1) + 6 \times N + 3 \times N + 1 = 10 \times N + 7 \text{ ciclos}$$

$$\text{CPI} = 10 \times N + 7 / 7 \times N + 5 \approx 10/7 = 1.43$$



Problema 16



$3 + (7 \times N + 1) + 1 = 7 \times N + 5$ instrucciones ejecutadas

$3 + (1 \times N + 3 \times 1) + 6 \times N + 3 \times N + 1 = 10 \times N + 7$ ciclos

$CPI = 10 \times N + 7 / 7 \times N + 5 \approx 10/7 = 1.43$

$9 \times N + 7$ ciclos

$CPI \approx 9/7 = 1.29$



Problema 17

- El programa que ejecuta 500 instrucciones tal que:
 - 20% de las instrucciones son de tipo `lw`
 - 50% seguidas de una instrucción aritmética que lee el registro cargado
 - 15% de las instrucciones son de tipo `sw`
 - 25% de las instrucciones son de tipo `beq`
 - 70% de los casos se salta
 - 5% de las instrucciones son de tipo `jal`
 - 35% de las instrucciones son aritmético-lógicas

$$CPI = 0.2 \cdot (0.5 \cdot 1 + 0.5 \cdot 2) + 0.15 \cdot 1 + 0.25 \cdot (0.7 \cdot 3 + 0.3 \cdot 1) + 0.05 \cdot 3 + 0.35 \cdot 1 = \mathbf{1.55}$$

Alternativamente:

$$CPI = 1 + \text{penalización} \\ = 1 + (0.2 \cdot 0.5 \cdot 1 + 0.25 \cdot 0.7 \cdot 2 + 0.05 \cdot 2) = \mathbf{1.55}$$

$$t_{ejec} = \frac{500 \cdot 1.55}{1.5 \cdot 10^9} = \mathbf{516.7 \text{ ns}}$$



Problema 18a

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
<code>addi s4,zero,2</code>	IF	ID	EX	M	WB										
<code>L1:lw s0,0(s2)</code>		IF	ID	EX	M	WB									
<code>lw s1,0(s2)</code>			IF	ID	EX	M	WB								
<code>add s3,s0,s1</code>				IF	IDp	ID	EX	M	WB						
<code>sw s3,0(s2)</code>					IFp	IF	ID	EX	M	WB					
<code>add s2,s3,s6</code>							IF	ID	EX	M	WB				
<code>and s1,s1,s2</code>								IF	ID	EX	M	WB			
<code>beq s4,zero,L1</code>									IF	ID	EX	M	WB		
<code>addi s2,s2,1</code>										IF	ID	EX	M	WB	
<code>or s3,s2,s4</code>											IF	ID	EX	M	WB



Problema 18b

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
<code>addi s4,zero,2</code>	IF	ID	EX	M	WB																			
<code>L1:lw s0,0(s2)</code>		IF	ID	EX	M	WB																		
<code>lw s1,0(s2)</code>			IF	ID	EX	M	WB																	
<code>add s3,s0,s1</code>				IF	IDp	IDp	ID	EX	M	WB														
<code>sw s3,0(s2)</code>					IFp	IFp	IF	IDp	IDp	ID	EX	M	WB											
<code>add s2,s3,s6</code>							IFp	IFp	IF	ID	EX	M	WB											
<code>and s1,s1,s2</code>										IF	IDp	IDp	ID	EX	M	WB								
<code>beq s4,zero,L1</code>											IFp	IFp	IF	ID	EX	M	WB							
<code>addi s2,s2,1</code>														IFp	IFp	IF	ID	EX	M	WB				
<code>or s3,s2,s4</code>																	IF	IDp	IDp	ID	EX	M	WB	



Problema 19

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
<code>addi s1,zero,1</code>	IF	ID	EX	M	WB													
L1: <code>sub s5,s5,s2</code>		IF	ID	EX	M	WB												
<code>addi s2,s2,1</code>			IF	ID	EX _{s1}	M	WB											
<code>addi s1,s1,-1</code>				IF	ID _{s1}	EX _{s1}	M	WB										
<code>beq s1,zero,L1</code>					IF	ID _{s2}	EX _{s2}	M	WB									
<code>and s5,s2,s3</code>						IF	ID	×										
<code>or s3,s3,s4</code>							IF	×										
L1: <code>sub s5,s5,s2</code>								IF	ID	EX	M	WB						
<code>addi s2,s2,1</code>									IF	ID	EX	M	WB					
<code>addi s1,s1,-1</code>										IF	ID	EX _{s1}	M	WB				
<code>beq s1,zero,L1</code>											IF	ID	EX _{s2}	M	WB			
<code>and s5,s2,s3</code>												IF	ID	EX	M	WB		
<code>or s3,s3,s4</code>													IF	ID	EX	M	WB	
<code>andi s4,s3,s2</code>														IF	ID	EX	M	WB

1ª iteración (s1=0) salta: fallo

2ª iteración (s1=-1) no salta: acierto

Acerca de *Creative Commons*



■ Licencia CC (**Creative Commons**)

- Ofrece algunos derechos a terceras personas bajo ciertas condiciones. Este documento tiene establecidas las siguientes:



Reconocimiento (*Attribution*):

En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.



No comercial (*Non commercial*):

La explotación de la obra queda limitada a usos no comerciales.



Compartir igual (*Share alike*):

La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Más información: <https://creativecommons.org/licenses/by-nc-sa/4.0/>