



Module 7 - Problems:

Pipelined processor design

Introduction to computers II

José Manuel Mendías Cuadros

Dpto. Arquitectura de Computadores y Automática
Universidad Complutense de Madrid





Problem 1a

- **Multicycle** (correct result in 12 cycles):
 - $t1 = 22+5 = 27$
 - $t3 = 27+22 = 49$
 - $t4 = 27+15 = 42$

	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

Instructions 2 and 3 read $t1$ from the RF after instruction 1 updates $t1$ in the RF



Problem 1b

- **Pipelined without hazard management** (incorrect result in 7 cycles):
 - $t1 = 22+5 = 27$
 - $t3 = 11+22 = 33$
 - $t4 = 11+15 = 26$

	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		

Data hazard in t1:

Instructions 2 and 3 read t1 from the RF before instruction 1 updates t1 in the RF



Problem 1c

- **Pipelined without hazard management + NOP**
(correct result in 10 cycles):
 - $t1 = 22+5 = 27$
 - $t3 = 27+22 = 49$
 - $t4 = 27+15 = 42$

	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

Instructions 5 and 6 read `t1` from the RF after
instruction 1 updates `t1` in the RF



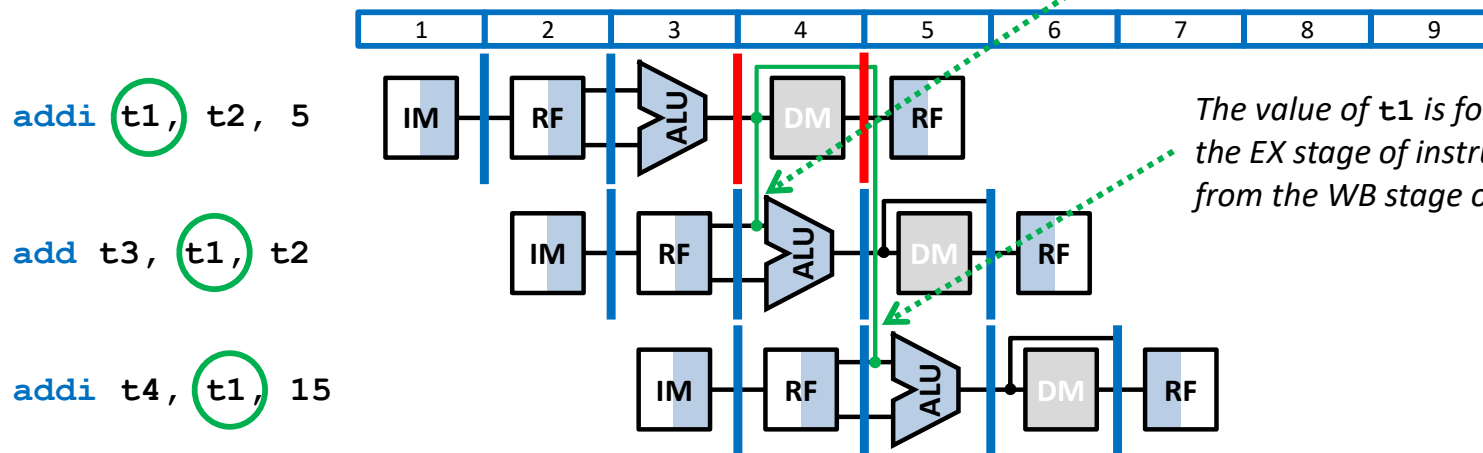
Problem 1d

■ **Pipelined** (correct result in 7 cycles):

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

The value of $t1$ is forwarded to the EX stage of instruction 2 from the MEM stage of instruction 1

The value of $t1$ is forwarded to the EX stage of instruction 3 from the WB stage of instruction 1



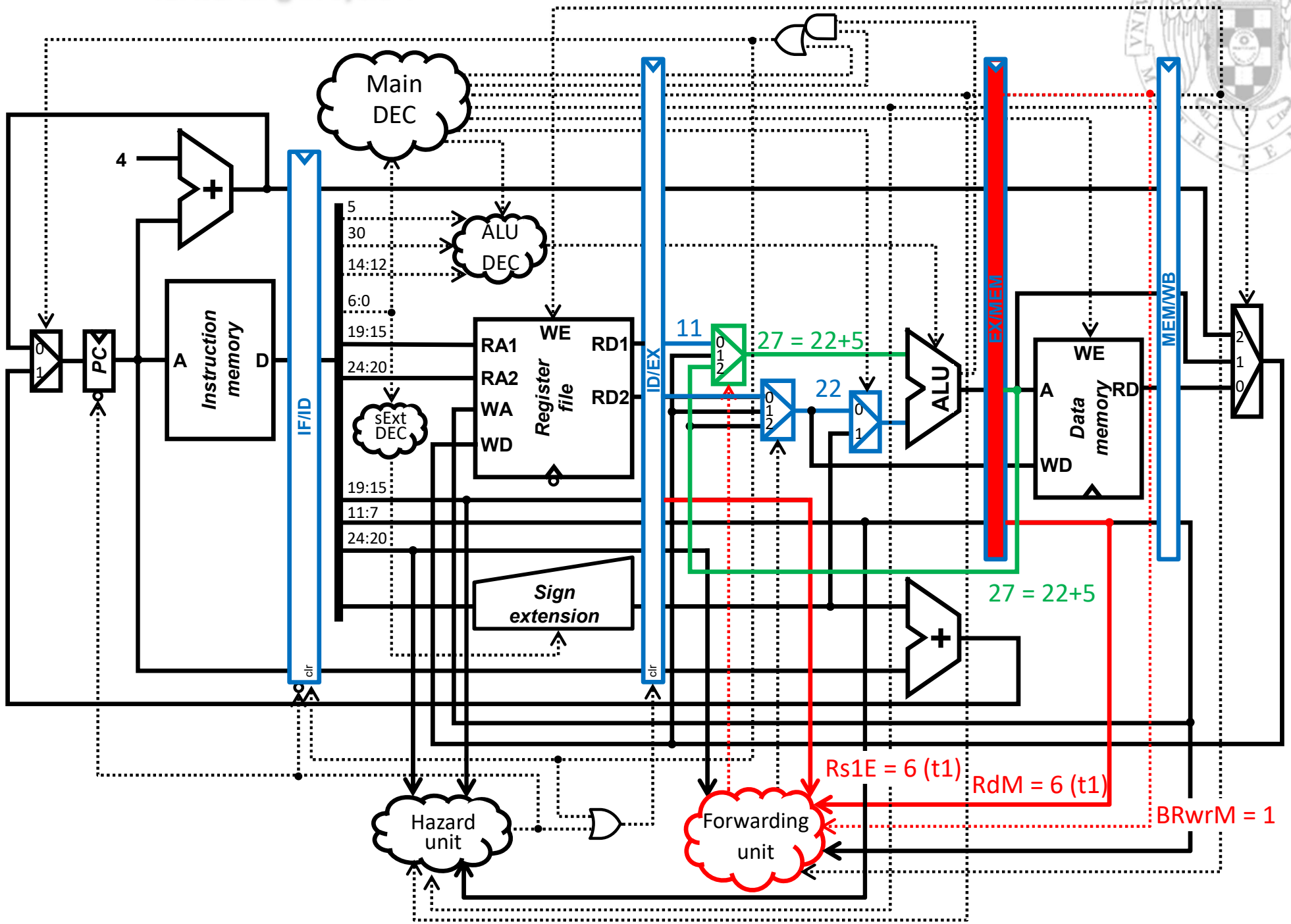
	1	2	3	4	5	6	7	8	9
<code>addi t1, t2, 5</code>	IF	ID	EX	M	WB				
<code>add t3, t1, t2</code>		IF	ID	EX	M	WB			
<code>addi t4, t1, 15</code>			IF	ID	EX	M	WB		

Problem 1d

forwarding in cycle 4

add t3,t1,t2

addi t1,t2,5



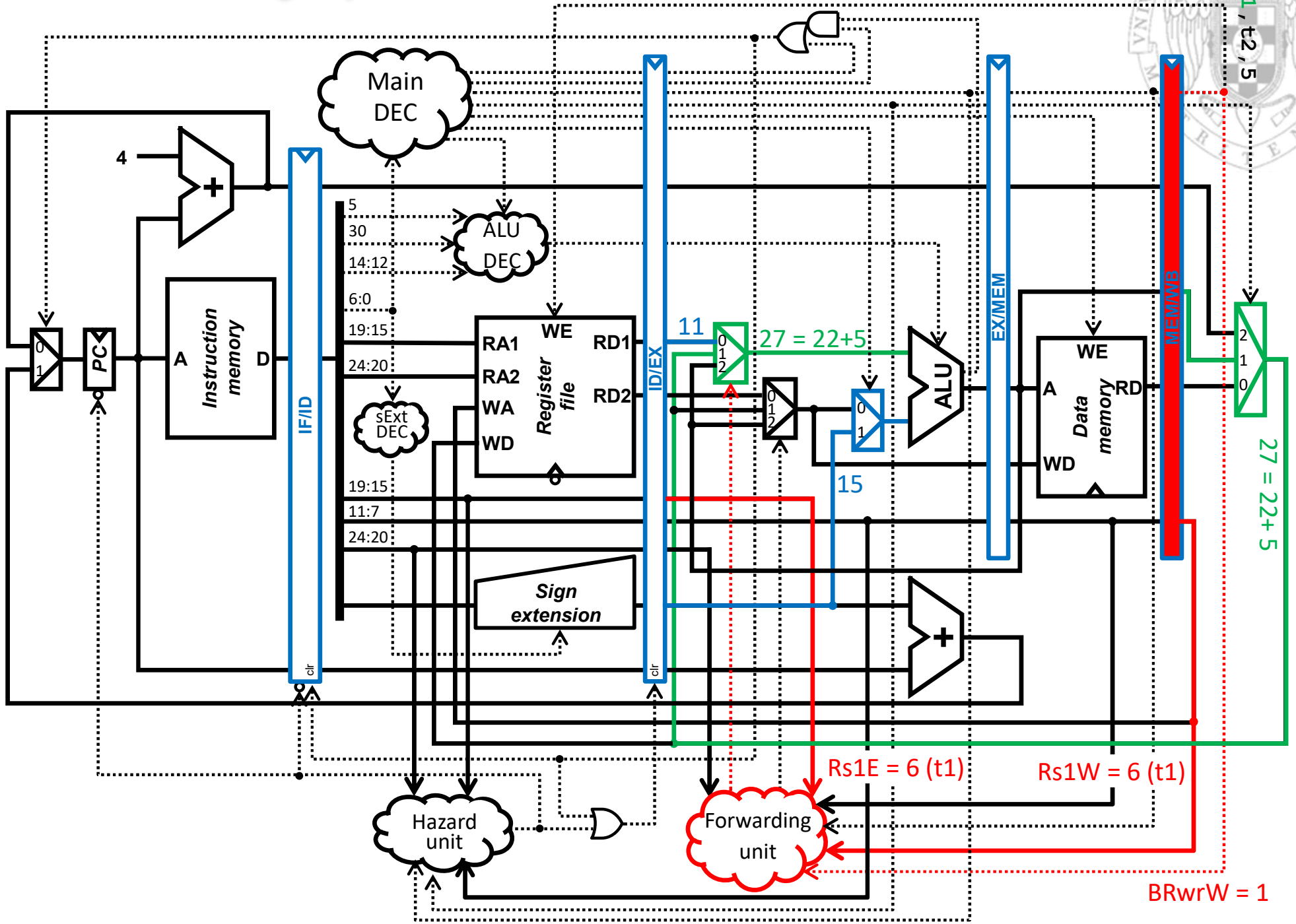
Problem 1d

forwarding in cycle 5

`addi t4,t1,15`

`add t3,t1,t2`

`addi t1,t2,5`





Problem 2a-2b

- **Multicycle** (correct result in 16 cycles):
 - $t1 = 22+5 = 27$
 - $t3 = 27+22 = 49$
 - $t4 = 27+15 = 42$
 - $t5 = 27+27 = 54$
- **Pipelined without hazard management** (incorrect result in 8 cycles):
 - $t1 = 22+5 = 27$
 - $t3 = 11+22 = 33$
 - $t4 = 11+15 = 26$
 - $t5 = 11+11 = 22$

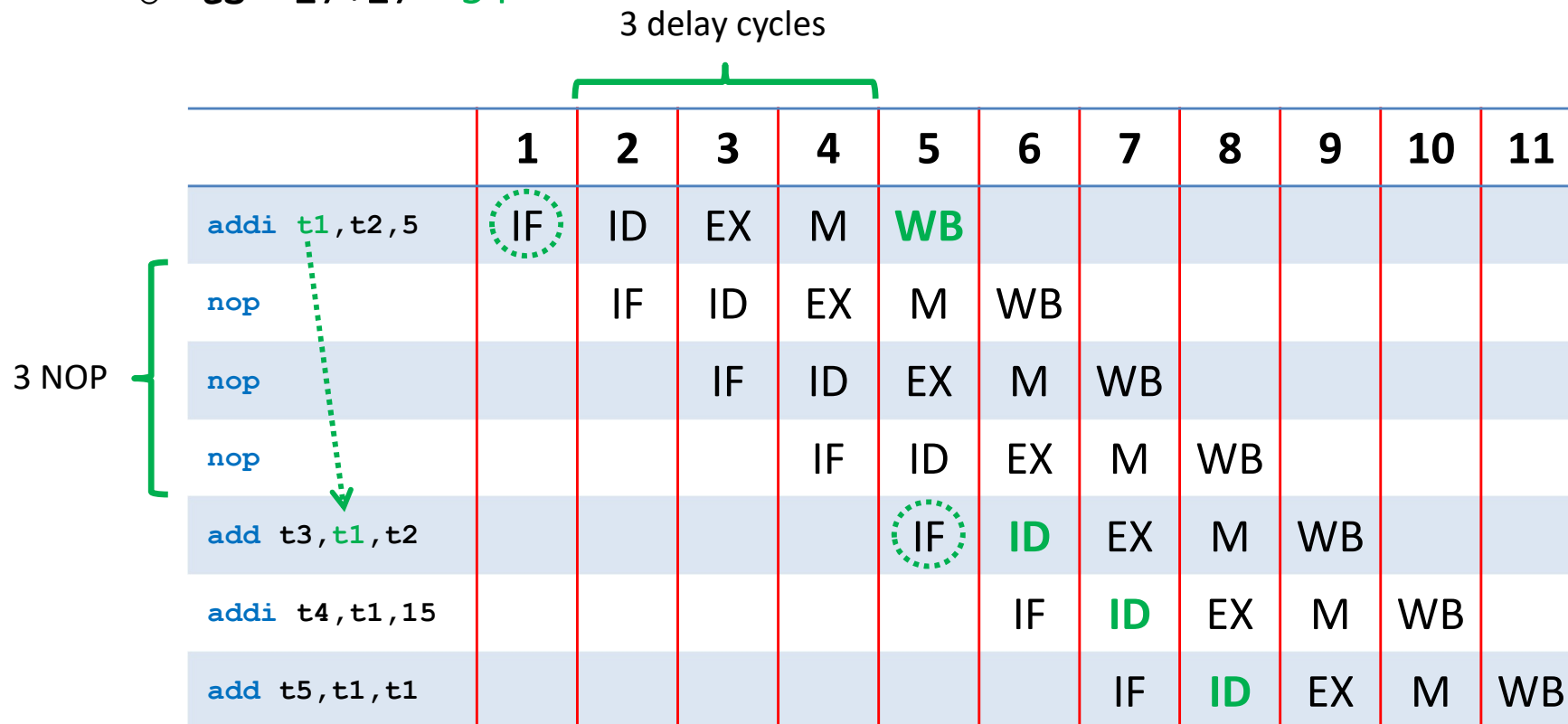
	1	2	3	4	5	6	7	8
<code>addi t1,t2,5</code>	IF	ID	EX	M	WB			
<code>add t3,t1,t2</code>		IF	ID	EX	M	WB		
<code>addi t4,t1,15</code>			IF	ID	EX	M	WB	
<code>add t5,t1,t1</code>				IF	ID	EX	M	WB



Problem 2c

- **Pipelined without hazard management + NOP**
(correct result in 11 cycles):

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





Problem 2d

- Pipelined without hazard management + RF write in the middle of the cycle (incorrect result in 8 cycles):
 - $t1 = 22+5 = 27$
 - $t3 = 11+22 = 33$
 - $t4 = 11+15 = 26$
 - $t5 = 27+27 = 54$

	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



Problem 2e

- Pipelined without hazard management + RF write in the middle of the cycle + NOP (correct result in 10 cycles):
 - $t1 = 22+5 = 27$
 - $t3 = 27+22 = 49$
 - $t4 = 27+15 = 42$
 - $t5 = 27+27 = 54$

2 delay cycles

	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

2 NOP



Problem 2f

■ **Pipelined** (correct result in 8 cycles):

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

	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



Problem 3

	IF	ID	EX	M	WB
1	<code>xor s1,s2,s3</code>				
2	<code>addi s0,s3,-4</code>	<code>xor s1,s2,s3</code>			
3	<code>lw s3,16(s7)</code>	<code>addi s0,s3,-4</code>	<code>xor s1,s2,s3</code>		
4					
5					

	1	2	3	4	5	6	7	8	9
<code>xor s1,s2,s3</code>	IF	ID	EX						
<code>addi s0,s3,-4</code>		IF	ID						
<code>lw s3,16(s7)</code>			IF						



Problem 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					



Problem 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



Problem 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





Problem 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



Problem 5a

- Pipelined without hazard management

	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



Problem 5b

- Pipelined without hazard management + RF write in the middle of the cycle

	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



Problem 5c

- Pipelined

	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 5-stage pipeline (IF, ID, EX, M, WB) over 8 clock cycles. The instructions are:

- `addi t1,t2,5`
- `add t3,t1,t2`
- `addi t4,t1,15`
- `add t5,t3,t2`

Green arrows indicate data hazards and forwarding:

- `add t3,t1,t2` depends on `t1` from `addi t1,t2,5`. A hazard is detected at the start of its EX stage (cycle 4), and the value of `t1` is forwarded from the M stage of the first instruction.
- `addi t4,t1,15` depends on `t1` from `addi t1,t2,5`. A hazard is detected at the start of its EX stage (cycle 5), and the value of `t1` is forwarded from the M stage of the first instruction.
- `add t5,t3,t2` depends on `t3` from `add t3,t1,t2`. A hazard is detected at the start of its EX stage (cycle 6), and the value of `t3` is forwarded from the M stage of the second instruction.



Problem 6a

- Pipelined without hazard management

	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



Problem 6b

- Pipelined without hazard management + RF write in the middle of the cycle

	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	^{x7} 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	^{x6} 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	^{x6} ID	EX	M	WB



Problem 6c

- Pipelined

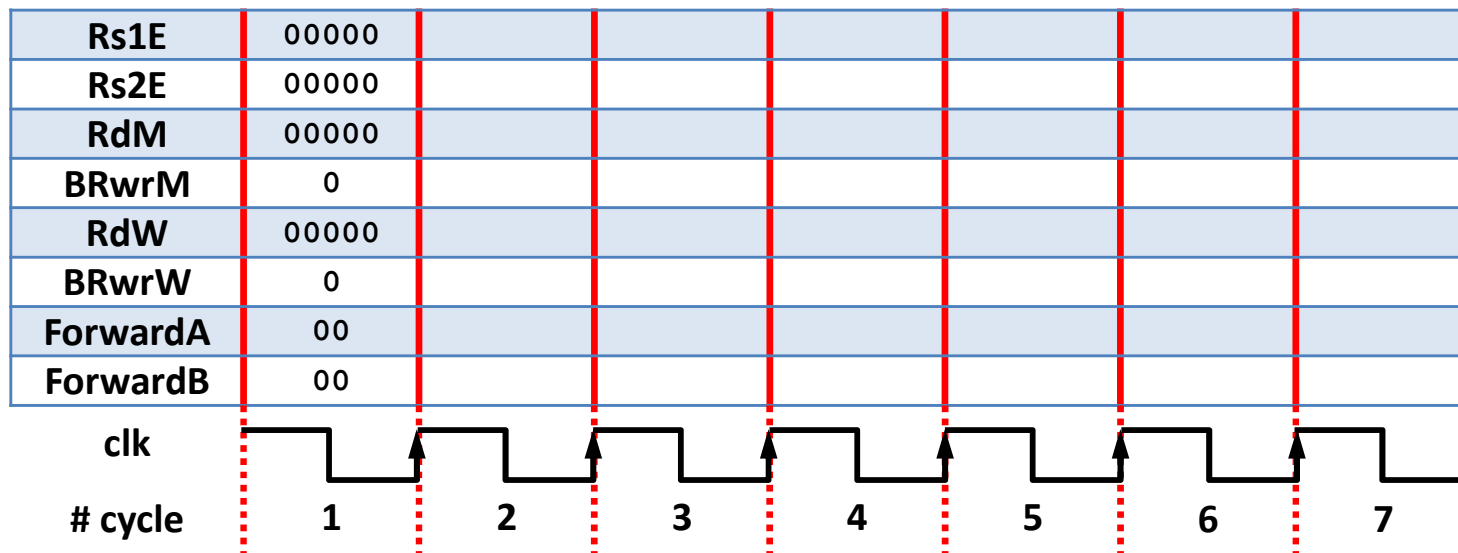
	1	2	3	4	5	6	7	8	9
<code>add x7, x5, x8</code>	IF	ID	EX	M	WB				
<code>lw x6, 8(x7)</code>		IF	ID	EX	M	WB			
<code>lw x5, 0(x5)</code>			IF	ID	EX	M	WB		
<code>or x6, x7, x6</code>				IF	ID	EX	M	WB	
<code>sw x6, 0(x7)</code>					IF	ID	EX	M	WB

Diagram illustrating the pipeline execution of five instructions: `add x7, x5, x8`, `lw x6, 8(x7)`, `lw x5, 0(x5)`, `or x6, x7, x6`, and `sw x6, 0(x7)`. The pipeline stages are IF (Instruction Fetch), ID (Instruction Decode), EX (Execute), M (Memory Access), and WB (Write Back). The instructions are pipelined, and the diagram shows the progression of each instruction through the stages over 9 clock cycles. Data hazards are indicated by green arrows: `lw x6, 8(x7)` depends on `add x7, x5, x8` (x7), and `or x6, x7, x6` and `sw x6, 0(x7)` depend on `lw x6, 8(x7)` (x6).



Problem 6c

	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>							





Problem 6c

	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							
# cycle	1	2	3	4	5	6	7



Problem 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							
# cycle	1	2	3	4	5	6	7



Problem 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	garbage			
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	garbage			
clk							
# cycle	1	2	3	4	5	6	7



Problem 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	garbage	garbage		
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	garbage	garbage		
clk							
# cycle	1	2	3	4	5	6	7



Problem 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	garbage	garbage	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	garbage	garbage	01	
clk							
# cycle	1	2	3	4	5	6	7



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

clk

cycle

1 2 3 4 5 6 7



Problem 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						



Problem 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						



Problem 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> ^{STOP}	<code>add t5,t2,t1</code> ^{STOP}	<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	IDs						
<code>addi t0,t0,1</code>			IFs						



Problem 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					

Annotations: Red 'STOP' signs and red 'X' marks over the add and addi instructions in cycle 3. Green arrows point from the addi instruction in cycle 3 to the addi instruction in cycle 4. A red cloud is in the EX stage of cycle 4.

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



Problem 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 data flow from the addi instruction in row 4 to the add instruction in row 5, and from the add instruction in row 5 to the lw instruction in row 6. Red clouds indicate stalls in the EX stage.

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

Note: Green arrows show data flow from the M stage of the first row to the WB stage of the second row, and from the EX stage of the second row to the EX stage of the third row.



Problem 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					

Diagram annotations: Green arrows show data flow from the EX stage of one instruction to the ID stage of the next. A green line with arrows shows the forwarding path from the EX stage of the second instruction to the EX stage of the third instruction. Red clouds indicate stalls in the EX stage.

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

Diagram annotations: A green arrow labeled 't1' points from the WB stage of the first instruction to the EX stage of the second instruction.



Problem 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		sub t3,t5,t0	addi t0,t0,1	add t5,t2,t1	Cloud
7			sub t3,t5,t0	addi t0,t0,1	add t5,t2,t1

Note: Red clouds indicate stalls. Green arrows show data flow: t1 from EX3 to M4, t5 from EX5 to EX6, 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	IDs	ID	EX	M	WB		
addi t0,t0,1			IFs	IF	ID	EX	M		
sub t3,t5,t0					IF	ID	EX		

Note: Green arrows show data flow: t1 from M4 to EX5, t5 from EX6 to EX7, and t0 from EX7 to EX8.



Problem 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 showing data flow and stalls:

- Green arrows show data flow: t1 from EX3 to M4, t1 from EX4 to M5, t5 and t0 from EX6 to EX7.
- Red clouds indicate stalls in the EX stage for instructions 4, 5, and 6.
- Red STOP signs are on instructions 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	IDs	ID	EX	M	WB		
addi t0,t0,1			IFs	IF	ID	EX	M	WB	
sub t3,t5,t0					IF	ID	EX	M	WB

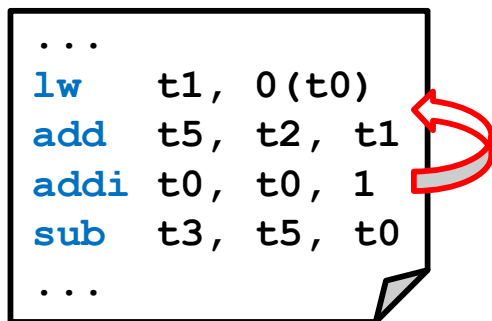
Diagram showing pipeline stages and data flow:

- Green arrows show data flow: t1 from M4 to EX5, t5 and t0 from EX6 to EX7.
- Red boxes highlight the stalls in the ID stage for instructions 2 and 3.



Problem 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 EX stage of one instruction to the EX stage of a subsequent instruction. Labels t1, t0, and t5 indicate the specific data values being forwarded.



Problem 8a-8b

- Pipelined without hazard management

	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

- Pipelined without hazard management + RF write in the middle of the cycle

	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' stage of the first instruction to the 'ID' stage of the third instruction, indicating a data hazard.



Problem 8c-8d

- Pipelined with partial hazard management

	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 a pipeline with partial hazard management. The pipeline stages are 1 (IF), 2 (ID), 3 (EX), 4 (M), 5 (WB), 6, 7, and 8. The instructions are: `lw t1,0(t0)`, `nop`, `add t5,t2,t1`, and `add t2,t1,14`. The `add t5,t2,t1` instruction is stalled in the EX stage for two cycles (stages 4 and 5) because it depends on the value of `t1` written by the `lw t1,0(t0)` instruction. The `add t2,t1,14` instruction is stalled in the ID stage for two cycles (stages 4 and 5) because it also depends on the value of `t1`. The `nop` instruction is stalled in the EX stage for two cycles (stages 4 and 5) because it depends on the value of `t1`. The `add t5,t2,t1` instruction is stalled in the EX stage for two cycles (stages 4 and 5) because it depends on the value of `t1`. The `add t2,t1,14` instruction is stalled in the ID stage for two cycles (stages 4 and 5) because it depends on the value of `t1`.

- Pipelined

	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	IDs	ID	EX	M	WB	
<code>add t2,t1,14</code>			IFs	IF	ID	EX	M	WB

Diagram illustrating a pipeline with full hazard management. The pipeline stages are 1 (IF), 2 (ID), 3 (EX), 4 (M), 5 (WB), 6, 7, and 8. The instructions are: `lw t1,0(t0)`, `add t5,t2,t1`, and `add t2,t1,14`. The `add t5,t2,t1` instruction is stalled in the ID stage for two cycles (stages 3 and 4) because it depends on the value of `t1` written by the `lw t1,0(t0)` instruction. The `add t2,t1,14` instruction is stalled in the IF stage for two cycles (stages 3 and 4) because it also depends on the value of `t1`. The `add t5,t2,t1` instruction is stalled in the EX stage for two cycles (stages 4 and 5) because it depends on the value of `t1`. The `add t2,t1,14` instruction is stalled in the ID stage for two cycles (stages 4 and 5) because it depends on the value of `t1`.



Problem 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	IDs	ID	EX	M	WB	
add t2,t1,14			IFs	IF	ID	EX	M	WB



Problem 9

15/01/24 version

	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	IDs	ID	EX	M	WB				
add s3,s2,s5				IFs	IF	IDs	ID	EX	M	WB		
or s4,s3,t4						IFs	IF	ID	EX	M	WB	
and s2,s3,s4								IF	ID	EX	M	WB



Problem 10a

	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> STOP	<code>sw s3,20(s0)</code> STOP	<code>lw s3,16(s0)</code>	<code>addi s0,s1,-4</code>	<code>addi s1,zero,52</code>
6	<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	IDs	ID	EX	M	WB		
<code>xor s2,s0,s3</code>					IFs	IF	ID	EX	M	WB	
<code>or s2,s2,s3</code>							IF	ID	EX	M	WB



Problem 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	



Problem 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	IDs	ID	EX	M	WB						
<code>add s2,s3,s6</code>			IFs	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	IDs	ID	EX	M	WB	
<code>or s2,s2,s5</code>								IFs	IF	ID	EX	M	WB



Problem 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	IDs	ID	EX	M	WB						
<code>add s2,s3,s6</code>			IFs	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	



Problem 12a

s0 ≠ 0

- **Multicycle** (correct result in 19 cycles):
 - 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
beq zero,s0,L1	IF	ID	EX																
addi t1,t1,-1				IF	ID	EX	WB												
addi t2,t2,-1								IF	ID	EX	WB								
L1:addi t3,t1,1												IF	ID	EX	WB				
addi t4,t2,1																IF	ID	EX	WB



Problem 12b

s0 ≠ 0

- **Pipelined without control hazard management**
(correct result in 9 cycles):
 - 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

Note: Green arrows in the original image show data forwarding from the EX stage of the first `addi` to the EX stage of the second `addi`, and from the EX stage of the second `addi` to the EX stage of the `L1: addi`.

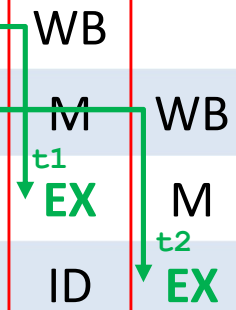


Problem 12c

s0 ≠ 0

- Pipelined without control hazard management + NOP (correct result in 11 cycles):

	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





Problem 12d

s0 ≠ 0

- Pipelined with control hazard management through pipeline stalling (correct result in 11 cycles):

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

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



Problem 12d

s0 ≠ 0

- Pipelined with control hazard management through pipeline stalling (correct result in 11 cycles):

	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									



Problem 12d

`s0 ≠ 0`

- Pipelined with control hazard management through pipeline stalling (correct result in 11 cycles):

	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>		IFs									



Problem 12d

$s0 \neq 0$

- Pipelined with control hazard management through pipeline stalling (correct result in 11 cycles):

	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					
5					
6					
7					

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



Problem 12d

$s0 \neq 0$

- Pipelined with control hazard management through pipeline stalling (correct result in 11 cycles):

	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 branch		
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>		IFs	IFs								



Problem 12d

s0 ≠ 0

- Pipelined with control hazard management through pipeline stalling (correct result in 11 cycles):

	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 ^{STOP}			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		IFs	IFs	IF							



Problem 12d

$s0 \neq 0$

- Pipelined with control hazard management through pipeline stalling (correct result in 11 cycles):

	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		IFs	IFs	IF	ID						
addi t2,t2,-1					IF						



Problem 12d

s0 ≠ 0

- Pipelined with control hazard management through pipeline stalling (correct result in 11 cycles):

	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		IFs	IFs	IF	ID	EX					
addi t2,t2,-1					IF	ID					
L1:addi t3,t1,1						IF					



Problem 12d

s0 ≠ 0

- Pipelined with control hazard management through pipeline stalling (correct result in 11 cycles):

	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		IFs	IFs	IF	ID	EX	M				
addi t2,t2,-1					IF	ID	EX				
L1:addi t3,t1,1						IF	ID				
addi t4,t2,1							IF				



Problem 12d

s0 ≠ 0

- Pipelined with control hazard management through pipeline stalling (correct result in 11 cycles):

	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		IFs	IFs	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



Problem 12e

s0 ≠ 0

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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								



Problem 12e

s0 ≠ 0

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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							



Problem 12e

`s0 ≠ 0`

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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						



Problem 12e

$s0 \neq 0$

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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 branch: right prediction		
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						



Problem 12e

`s0 ≠ 0`

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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 branch: right prediction		
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					



Problem 12e

`s0 ≠ 0`

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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				



Problem 12e

s0 ≠ 0

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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			



Problem 12e

s0 ≠ 0

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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		



Problem 12e

$s0 \neq 0$

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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



Problem 12f

`s0 = 0`

- **Multicycle** (correct result in 11 cycles):

- `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

The instruction to execute after `beq` is fetched from memory in the cycle after deciding whether taking the branch or not



Problem 12g

s0 = 0

- Pipelined without control hazard management (incorrect result in 9 cycles):

- 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
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

There is a control hazard:

Some instructions are fetched from memory before the beq instruction decides whether taking the branch or not



Problem 12h

`s0 = 0`

- Pipelined without control hazard management + NOP
(correct result in 9 cycles):

2 delay cycles

	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



Problem 12i

$s0 = 0$

- Pipelined with control hazard management through pipeline stalling (correct result in 9 cycles).

	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								



Problem 12i

`s0 = 0`

- Pipelined with control hazard management through pipeline stalling (correct result in 9 cycles).

	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							



Problem 12i

`s0 = 0`

- Pipelined with control hazard management through pipeline stalling (correct result in 9 cycles).

	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					
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>		IFs							



Problem 12i

$s0 = 0$

- Pipelined with control hazard management through pipeline stalling (correct result in 9 cycles).

	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					
5					
6					
7					

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



Problem 12i

$s0 = 0$

- Pipelined with control hazard management through pipeline stalling (correct result in 9 cycles).

	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>	branch		
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>		IFs	IFs						



Problem 12i

`s0 = 0`

- Pipelined with control hazard management through pipeline stalling (correct result in 9 cycles).

	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>		IFs	IFs						
<code>L1:addi t3,t1,1</code>				IF					



Problem 12i

$s0 = 0$

- Pipelined with control hazard management through pipeline stalling (correct result in 9 cycles).

	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		IFs	IFs	IF					
L1:addi t3,t1,1				IF	ID				
addi t4,t2,1					IF				



Problem 12i

s0 = 0

- Pipelined with control hazard management through pipeline stalling (correct result in 9 cycles).

	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		IFs	IFs	IF					
L1:addi t3,t1,1				IF	ID	EX			
addi t4,t2,1					IF	ID			



Problem 12i

s0 = 0

- Pipelined with control hazard management through pipeline stalling (correct result in 9 cycles).

	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		IFs	IFs	IF					
L1:addi t3,t1,1				IF	ID	EX	M		
addi t4,t2,1					IF	ID	EX		



Problem 12i

$s0 = 0$

- Pipelined with control hazard management through pipeline stalling (correct result in 9 cycles).

	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		IFs	IFs	IF					
L1:addi t3,t1,1				IF	ID	EX	M	WB	
addi t4,t2,1					IF	ID	EX	M	WB



Problem 12j

$s0 = 0$

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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								



Problem 12j

`s0 = 0`

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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							



Problem 12j

$s0 = 0$

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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						



Problem 12j

`s0 = 0`

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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>	branch: wrong prediction		
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						



Problem 12j

$s0 = 0$

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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>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	EX					
<code>L1:addi t3,t1,1</code>				IF					



Problem 12j

`s0 = 0`

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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>beq zero,s0,L1</code>	
5	<code>addi t4,t2,1</code>	<code>L1:addi t3,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						
<code>addi t2,t2,-1</code>			IF						
<code>L1:addi t3,t1,1</code>				IF	ID				
<code>addi t4,t2,1</code>					IF				



Problem 12j

s0 = 0

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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			



Problem 12j

s0 = 0

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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		



Problem 12j

$s0 = 0$

- Pipelined (with not-taken branch prediction)
(correct result in 9 cycles):

	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



Problem 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					



Problem 13

	IF	ID	EX	M	WB
1	<code>jal x0,L1</code>				
2	<code>addi t1,x0,5</code>	<code>jal x0,L1</code>	branch: wrong prediction		
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					



Problem 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	branch: wrong prediction jal x0,L1		
4	L1:sw t4,0(t3)			jal x0,L1	
5					
6					
7					
8					

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



Problem 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			



Problem 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		



Problem 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	



Problem 13

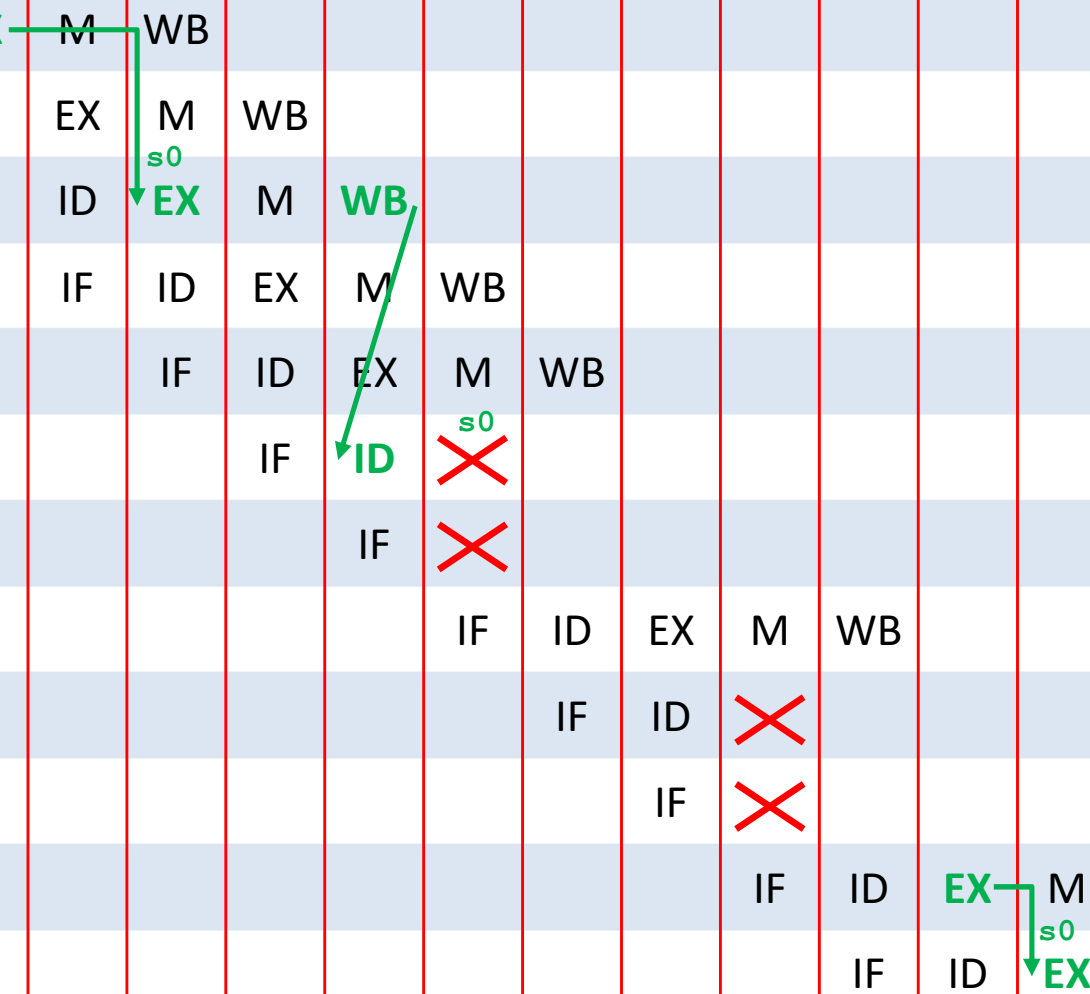
	IF	ID	EX	M	WB
1	jal x0,L1				
2	addi t1,x0,5	jal x0,L1			
3	add t3,t1,t2	addi t1,x0,5	jal x0,L1		
4	L1:sw t4,0(t3)			jal x0,L1	
5		L1:sw t4,0(t3)			jal x0,L1
6			L1:sw t4,0(t3)		
7				L1:sw t4,0(t3)	
8					L1:sw t4,0(t3)

	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



Problem 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



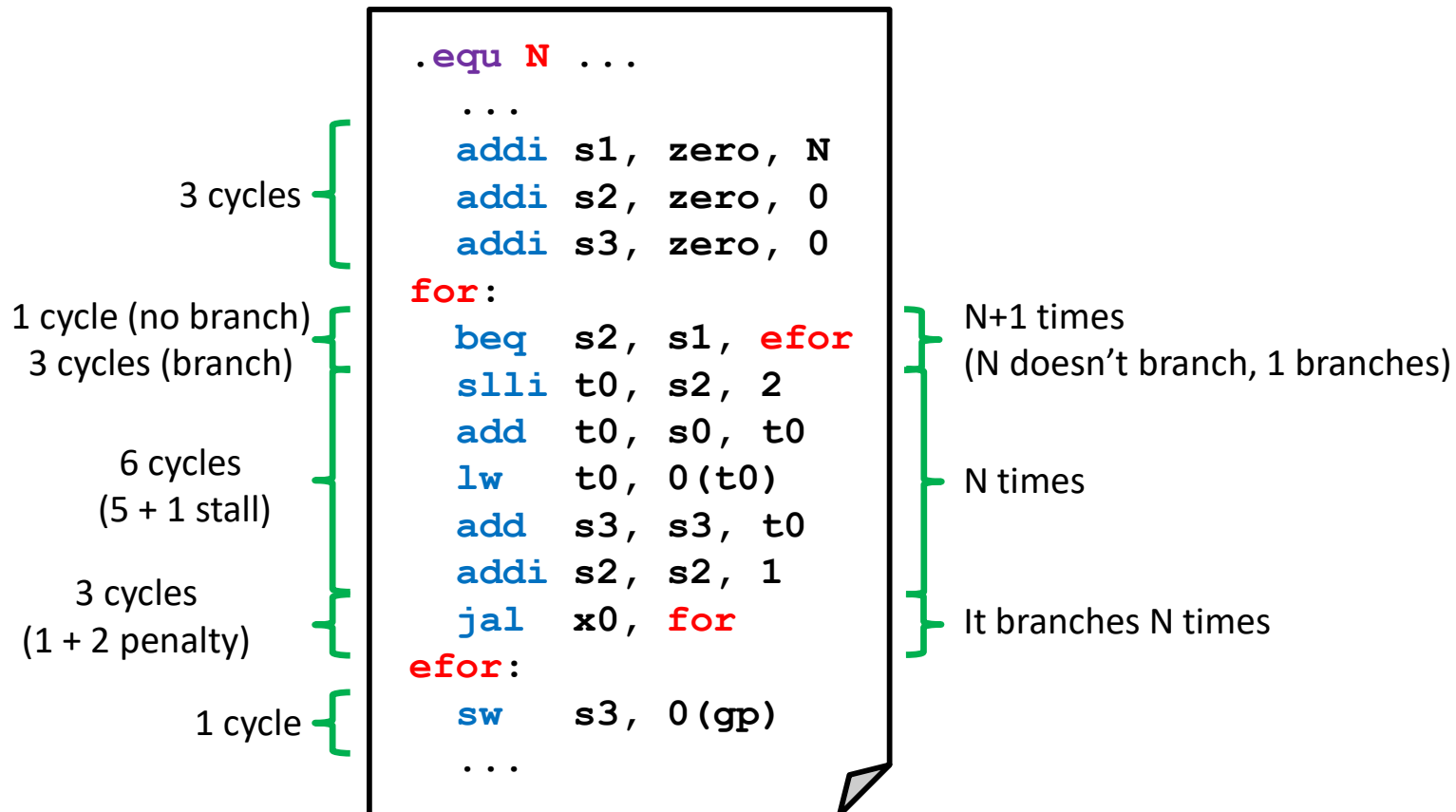


Problem 15

	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	IDs	ID	EX	M	WB												
<code>add s0,s0,t0</code>				IFs	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	IDs	ID	EX	M	WB	
<code>sw s1,0(gp)</code>															IFs	IF	ID	EX	M	WB



Problem 16



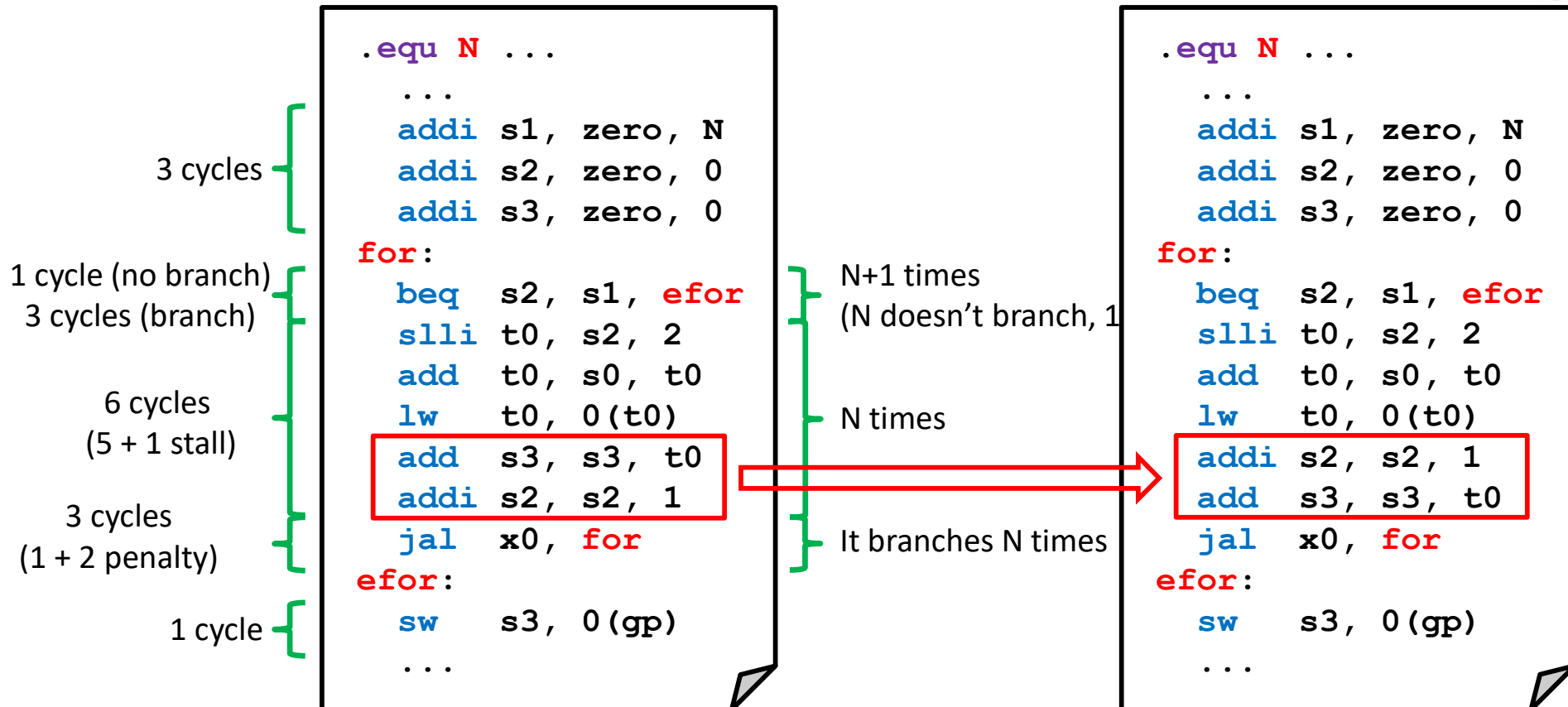
$$3 + (7 \times N + 1) + 1 = 7 \times N + 5 \text{ executed instructions}$$

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

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



Problem 16



$3 + (7 \times N + 1) + 1 = 7 \times N + 5$ executed instructions
 $3 + (1 \times N + 3 \times 1) + 6 \times N + 3 \times N + 1 = 10 \times N + 7$ cycles

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

$9 \times N + 7$ cycles

$CPI \approx 9/7 = 1.29$



Problem 17

- The program **executes 500 instructions**, distributed as follows:
 - 20% of the instructions are **lw**
 - 50% followed by an arithmetic instruction that reads the loaded register
 - 15% of the instructions are **sw**
 - 25% of the instructions are **beq**
 - 70% are taken branches
 - 5% of the instructions are **jal**
 - 35% of the instructions are **arithmetic-logic**

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

In a different way:

$$\begin{aligned}CPI &= 1 + \textit{penalty} \\ &= 1 + (0.2 \cdot 0.5 \cdot 1 + 0.25 \cdot 0.7 \cdot 2 + 0.05 \cdot 2) = \mathbf{1.55}\end{aligned}$$

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



Problem 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	IDs	ID	EX	M	WB						
<code>sw s3,0(s2)</code>					IFs	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



Problem 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	IFs	IFs	ID ^{s1}	EX	M	WB														
<code>sw s3,0(s2)</code>					IFs	IFs	IF	IFs	IFs	ID ^{s3}	EX	M	WB											
<code>add s2,s3,s6</code>							IFs	IFs	IF	ID	EX	M	WB											
<code>and s1,s1,s2</code>										IF	IFs	IFs	ID ^{s2}	EX	M	WB								
<code>beq s4,zero,L1</code>											IFs	IFs	IF	ID	EX	M	WB							
<code>addi s2,s2,1</code>														IFs	IFs	IF	ID	EX	M	WB				
<code>or s3,s2,s4</code>																	IF	IFs	IFs	ID ^{s2}	EX	M	WB	



Problem 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	M	WB											
<code>addi s1,s1,-1</code>				IF	ID	EX	M	WB										
<code>beq s1,zero,L1</code>					IF	ID	EX	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	M	WB				
<code>beq s1,zero,L1</code>											IF	ID	EX	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

1st iteration (s1=0) branch: miss

2nd iteration (s1=-1) no branch: hit

About *Creative Commons*



■ CC license (*Creative Commons*)

- This license enables reusers to distribute, remix, adapt, and build upon the material in any medium or format for noncommercial purposes only, and only so long as attribution is given to the creator. If you remix, adapt, or build upon the material, you must license the modified material under identical terms:



Attribution:

Credit must be given to the creator.



Non commercial:

Only noncommercial uses of the work are permitted.



Share alike:

Adaptations must be shared under the same terms.

More information: <https://creativecommons.org/licenses/by-nc-sa/4.0/>