

# Exemple RISC-V

## 5-stage pipeline

## Codi font:

```
li x1,2000
li x2,50
li x8, 3
addi x3,x1,20
sub x5,x2,x8
andi x6,x3,0x3A
add x7,x1,x2
```

## Codi modificat:

0 : 7d000093 addi x1 x0 2000  
4 : 03200113 addi x2 x0 50  
8 : 00300413 addi x8 x0 3  
C : 01408193 addi x3 x1 20  
10: 408102b3 sub x5 x2 x8  
14: 03a1f313 andi x6 x3 58  
18: 002083b3 add x7 x1 x2

Fa aquesta traducció ja que  
x0 sempre contrindrà 0s

A partir d'aquí, cada **transparència** equival a un **cicle de rellotge**

		0											
1	<code>addi x1 x0 2000</code>	IF											
2	<code>addi x2 x0 50</code>												
3	<code>addi x8 x0 3</code>												
4	<code>addi x3 x1 20</code>												
5	<code>sub x5 x2 x8</code>												
6	<code>andi x6 x3 58</code>												
7	<code>add x7 x1 x2</code>												

Cicle 0: introduïm la primera instrucció a l'IF.

Si no tenim cap problema, haurem d'anar desplaçant aquesta instrucció pel pipeline a mesura que arriben els clocks. A més, anirem introduint les noves instruccions sempre que sigui possible.

A partir d'aquí, cada **transparència** equival a un **cicle de rellotge**

	0	1										
addi x1 x0 2000	IF	ID										
addi x2 x0 50		IF										
addi x8 x0 3												
addi x3 x1 20												
sub x5 x2 x8												
andi x6 x3 58												
add x7 x1 x2												

A partir d'aquí, cada **transparència** equival a un **cicle de rellotge**

	0	1	2									
addi x1 x0 2000	IF	ID	EX									
addi x2 x0 50		IF	ID									
addi x8 x0 3			IF									
addi x3 x1 20												
sub x5 x2 x8												
andi x6 x3 58												
add x7 x1 x2												

A partir d'aquí, cada **transparència** equival a un **cicle de rellotge**

	0	1	2	3								
addi x1 x0 2000	IF	ID	EX	M								
addi x2 x0 50		IF	ID	EX								
addi x8 x0 3			IF	ID								
addi x3 x1 20				IF								
sub x5 x2 x8												
andi x6 x3 58												
add x7 x1 x2												

A partir d'aquí, cada **transparència** equival a un **cicle de rellotge**

	0	1	2	3	4							
addi x1 x0 2000	IF	ID	EX	M	W							
addi x2 x0 50		IF	ID	EX	M							
addi x8 x0 3			IF	ID	EX							
addi x3 x1 20				IF	ID							
sub x5 x2 x8					IF							
andi x6 x3 58												
add x7 x1 x2												

Cicle 4: Finalitza la primera instrucció. Tenim el pipeline ple.

A partir d'aquí, cada **transparència** equival a un **cicle de rellotge**

	0	1	2	3	4	5						
addi x1 x0 2000	IF	ID	EX	M	W							
addi x2 x0 50		IF	ID	EX	M	W						
addi x8 x0 3			IF	ID	EX	M						
addi x3 x1 20				IF	ID	EX						
sub x5 x2 x8					IF	ID						
andi x6 x3 58						IF						
add x7 x1 x2												

Cicle 5: Finalitza la segona instrucció.



A partir d'aquí, cada **transparència** equival a un **cicle de rellotge**

	0	1	2	3	4	5	6					
addi x1 x0 2000	IF	ID	EX	M	W							
addi x2 x0 50		IF	ID	EX	M	W						
addi x8 x0 3			IF	ID	EX	M	W					
addi x3 x1 20				IF	ID	EX	M					
sub x5 x2 x8					IF	ID	EX					
andi x6 x3 58						IF	ID					
add x7 x1 x2							IF					

Cicle 6: Finalitza la tercera instrucció.

A partir d'aquí, cada **transparència** equival a un **cicle de rellotge**

	0	1	2	3	4	5	6	7				
addi x1 x0 2000	IF	ID	EX	M	W							
addi x2 x0 50		IF	ID	EX	M	W						
addi x8 x0 3			IF	ID	EX	M	W					
addi x3 x1 20				IF	ID	EX	M	W				
sub x5 x2 x8					IF	ID	EX	M				
andi x6 x3 58						IF	ID	EX				
add x7 x1 x2							IF	ID				

Cicle 7: Finalitza la quarta instrucció.

A partir d'aquí, cada **transparència** equival a un **cicle de rellotge**

	0	1	2	3	4	5	6	7	8			
addi x1 x0 2000	IF	ID	EX	M	W							
addi x2 x0 50		IF	ID	EX	M	W						
addi x8 x0 3			IF	ID	EX	M	W					
addi x3 x1 20				IF	ID	EX	M	W				
sub x5 x2 x8					IF	ID	EX	M	W			
andi x6 x3 58						IF	ID	EX	M			
add x7 x1 x2							IF	ID	EX			

Cicle 8: Finalitza la cinquena instrucció.

A partir d'aquí, cada **transparència** equival a un **cicle de rellotge**

	0	1	2	3	4	5	6	7	8	9		
<code>addi x1 x0 2000</code>	IF	ID	EX	M	W							
<code>addi x2 x0 50</code>		IF	ID	EX	M	W						
<code>addi x8 x0 3</code>			IF	ID	EX	M	W					
<code>addi x3 x1 20</code>				IF	ID	EX	M	W				
<code>sub x5 x2 x8</code>					IF	ID	EX	M	W			
<code>andi x6 x3 58</code>						IF	ID	EX	M	W		
<code>add x7 x1 x2</code>							IF	ID	EX	M		

Cicle 9: Finalitza la sisena instrucció.

A partir d'aquí, cada **transparència** equival a un **cicle de rellotge**

	0	1	2	3	4	5	6	7	8	9	10	
<code>addi x1 x0 2000</code>	IF	ID	EX	M	W							
<code>addi x2 x0 50</code>		IF	ID	EX	M	W						
<code>addi x8 x0 3</code>			IF	ID	EX	M	W					
<code>addi x3 x1 20</code>				IF	ID	EX	M	W				
<code>sub x5 x2 x8</code>					IF	ID	EX	M	W			
<code>andi x6 x3 58</code>						IF	ID	EX	M	W		
<code>add x7 x1 x2</code>							IF	ID	EX	M	W	

Cicle 10: Finalitza la última instrucció. El programa s'acaba.

Podeu jugar amb el simulador, provant diferents programes

**RIPES: <https://github.com/mortbopet/Ripes>**

The screenshot displays the Ripes simulator interface. The main window shows a block diagram of a 5-Stage RISC-V Processor w/o Hazard Detection. A configuration window titled 'Select Processor' is open, showing the selection of the '5-Stage Processor w/o hazard detection' option. The configuration window includes fields for Name, ISA (RV32IM), Layout (Standard), and a description. Below these fields is a 'Register initialization' section with a table for register values. The main window also shows a console area at the bottom left, a 'Registers' panel on the right, and an 'Execution info' panel at the bottom right.

addi x1 x0 2000

5-Stage RISC-V Processor w/o Hazard Detection

Select Processor

RISC-V

- Single Cycle Processor
- 5-Stage Processor
- 5-Stage Processor w/o hazard detection**
- 5-Stage Processor w/o forwarding or hazard detection

Name: 5-Stage Processor w/o hazard detection

ISA: RV32IM

Layout: Standard

Description: A 5-Stage in-order processor with forwarding but no hazard detection/elimination.

Register initialization

Register	Value
x2 (sp)	0x7ffffff0

OK Cancel

Registers

Name	Alias	Value
x0	zero	0x00000000
x1	ra	0x00000000
x2	sp	0x7ffffff0
x3	gp	0x10000000
x4	tp	0x00000000
x5	t0	0x00000000
x6	t1	0x00000000
x7	t2	0x00000000

Display type: Hex

Instruction memory

BP	Addr	Stage	Instruction
<input type="checkbox"/>	0x0	IF	addi x1 x0 2000
<input type="checkbox"/>	0x4		addi x2 x0 50
<input type="checkbox"/>	0x8		addi x8 x0 3
<input type="checkbox"/>	0xc		addi x3 x1 20
<input type="checkbox"/>	0x10		sub x5 x3 x8
<input type="checkbox"/>	0x14		andi x6 x3 58
<input type="checkbox"/>	0x18		add x7 x1 x2

Execution info

Cycles: 0

Instrs. retired: 0

CPI: 0

IPC: 0

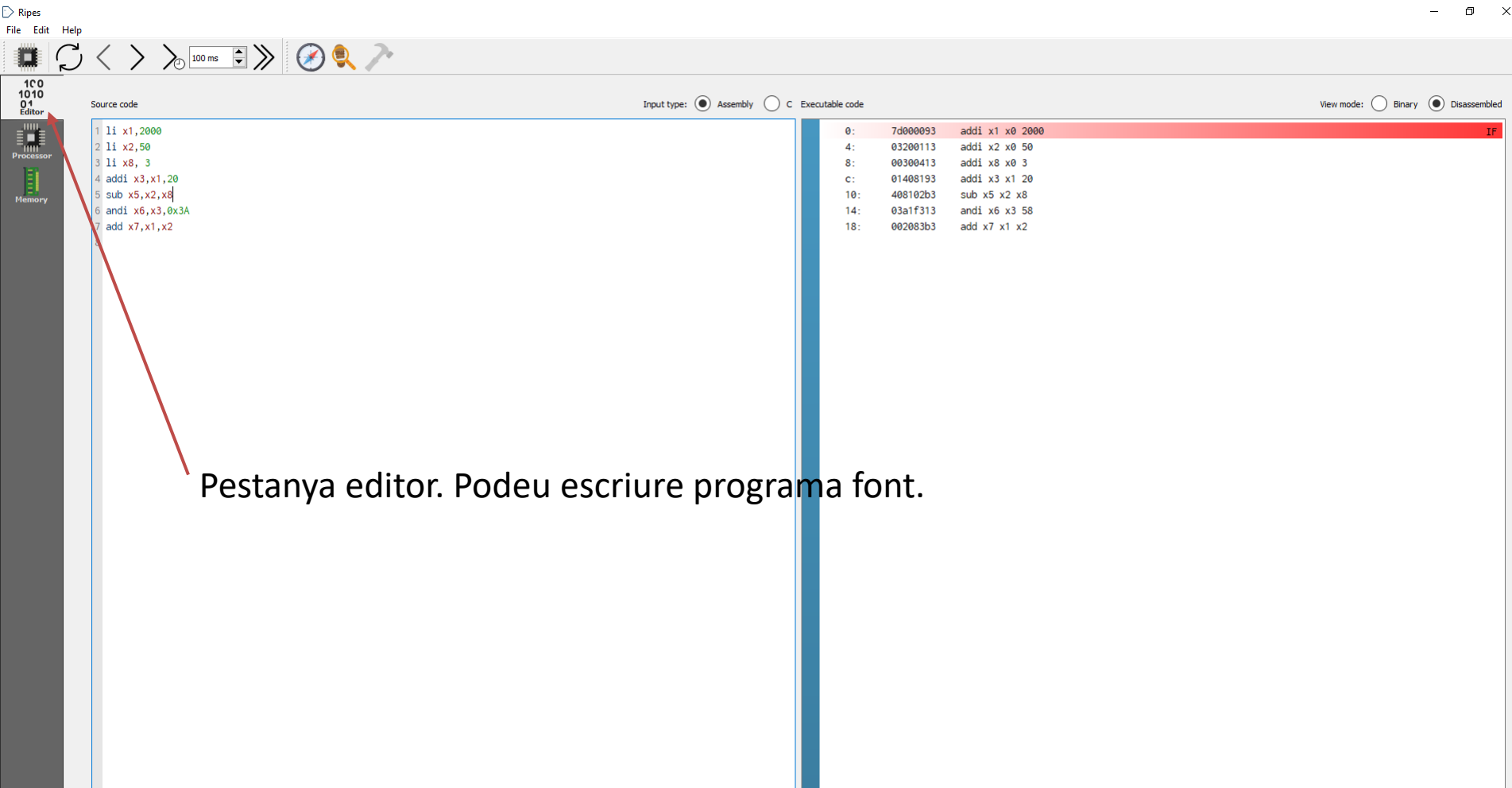
Clock rate: 0 Hz

Console

Ens permet triar entre diferents tipus de processador. El problema sempre ens dirà el HW que tenim

Podeu jugar amb el simulador, provant diferents programes

**RIPES: <https://github.com/mortbopet/Ripes>**



Pestanya editor. Podeu escriure programa font.

Executa  
pas a pas

Podeu jugar amb el simulador, provant diferents programes

RIPES: <https://github.com/mortbopet/Ripes>

Mostra valors variables

Pestanya processador. Mostra pipeline.

Registers

Name	Alias	Value
x0	zero	0x00000000
x1	ra	0x00000000
x2	sp	0x7fffffff0
x3	gp	0x10000000
x4	tp	0x00000000
x5	t0	0x00000000
x6	t1	0x00000000
x7	t2	0x00000000

Display type: Hex

Instruction memory

BP	Addr	Stage	Instruction
<input type="checkbox"/>	0x0	IF	addi x1 x0 2000
<input type="checkbox"/>	0x4		addi x2 x0 50
<input type="checkbox"/>	0x8		addi x8 x0 3
<input type="checkbox"/>	0xc		addi x3 x1 20
<input type="checkbox"/>	0x10		sub x5 x2 x8
<input type="checkbox"/>	0x14		andi x6 x3 58
<input type="checkbox"/>	0x18		add x7 x1 x2

Execution info

Cycles: 0

Instrs. retired: 0

CPI: 0

IPC: 0

Clock rate: 0 Hz



Podeu jugar amb el simulador, provant diferents programes

RIPES: <https://github.com/mortbopet/Ripes>

The screenshot displays the Ripes RISC-V simulator interface. The main window shows a block diagram of a 5-stage RISC-V processor with components like PC, Instr. memory, IFID, Decode, and MEM/WB. A red arrow points from the 'enable' button in the diagram to the 'Run' button in the top toolbar. A 'Stage table' window is open, showing the pipeline stages for several instructions. The 'Registers' window on the right lists registers x0 through x7 with their aliases and values. The 'Execution info' window at the bottom right shows statistics like Cycles, Instrs. retired, CPI, IPC, and Clock rate. The 'Console' window at the bottom left contains the text: 'Serveix per veure el diagrama multicycle del nostre programa. En aquest cas podem veure que no m'he equivocat en l'exemple 😊'.

	0	1	2	3	4	5	6	7	8	9	10
addi x1 x0 2000	IF	ID	EX	MEM	WB						
addi x2 x0 50		IF	ID	EX	MEM	WB					
addi x8 x0 3			IF	ID	EX	MEM	WB				
addi x3 x1 20				IF	ID	EX	MEM	WB			
sub x5 x2 x8					IF	ID	EX	MEM	WB		
andi x6 x3 58						IF	ID	EX	MEM	WB	
add x7 x1 x2							IF	ID	EX	MEM	WB

Name	Alias	Value
x0	zero	0x00000000
x1	ra	0x000007d0
x2	sp	0x00000032
x3	gp	0x000007e4
x4	tp	0x00000000
x5	t0	0x0000002f
x6	t1	0x00000020
x7	t2	0x00000802

Cycles:	11
Instrs. retired:	7
CPI:	1.57
IPC:	0.636
Clock rate:	1.26 Hz

Console

Serveix per veure el diagrama multicycle del nostre programa. En aquest cas podem veure que no m'he equivocat en l'exemple 😊