

Exemple RISC-V

5-stage pipeline: **Stalls**

Mateix exemple però fent un canvi de registre (en vermell)

Codi font:

```
li x1, 2000
li x2, 50
li x8, 3
addi x3, x1, 20
sub x5, x3*, 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: 408182b3 sub x5 x3 x8
14: 03a1f313 andi x6 x3 58
18: 002083b3 add x7 x1 x2
```

Fa aquesta traducció ja que
x0 sempre contrindrà 0s

*És un exemple forçat per veure com queda el diagrama, el compilador, si pot, ho modificaria això.

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 x3 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 x3 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 x3 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 x3 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 x3 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 x3 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 x3 x8					IF	ID	-					
andi x6 x3 58						IF	-					
add x7 x1 x2												

Cicle 6: Finalitza la tercera instrucció.

Nota: “-” significa stall, pot ser una “S”, pot ser repetir l’estat on estem (en aquest cas ID). El que hem de veure és que no podem avançar fins que no tinguem la dada correcte i, per tant, esperem sense fer res fins a aconseguir-la

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 x3 x8					IF	ID	-	-				
andi x6 x3 58						IF	-	-				
add x7 x1 x2												

Cicle 7: Finalitza la quarta instrucció. Durant el primer mig cicle s'escriu x3 al conjunt de registres. Durant el segon mig cicle la instrucció sub el llegeix

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 x3 x8					IF	ID	-	-	EX			
andi x6 x3 58						IF	-	-	ID			
add x7 x1 x2									IF			

Cicle 8: Aconseguim introduir la última instrucció.

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

	0	1	2	3	4	5	6	7	8	9		
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 x3 x8					IF	ID	-	-	EX	M		
andi x6 x3 58						IF	-	-	ID	EX		
add x7 x1 x2									IF	ID		

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 x3 x8</code>					IF	ID	-	-	EX	M	W	
<code>andi x6 x3 58</code>						IF	-	-	ID	EX	M	
<code>add x7 x1 x2</code>									IF	ID	EX	

Cicle 10: 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	10	11
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 x3 x8					IF	ID	-	-	EX	M	W	
andi x6 x3 58						IF	-	-	ID	EX	M	W
add x7 x1 x2									IF	ID	EX	M

Cicle 11: Finalitza la sisena instrucció.

El programa acaba trigant **13 cicles** (emprant x2 a la resta trigava **11 cicles**)