

# Sistemas digitales

## Codificación de instrucciones Ciclo de Instrucción



2024 - Primer Cuatrimestre

Universidad de Buenos Aires - FCEyN - Departamento de Computación

# ¿Qué vamos a hacer hoy?

1

**Programa escrito  
en Assembler**

ADDI a0, a1, 0x002

SRLI a0,a0,3

**CODIFICAR**

0001100110000110011  
0000110011001110001  
10

Usaremos el  
manual de  
RISC-V

Código de alto nivel  
(Compilador)



Código de bajo nivel  
(Ensamblador)



Código objeto  
(Enlazador)



Código binario ejecutable

Código objeto en archivos  
Bibliotecas



2

Programa escrito  
en Assembler

CODIFICAR

0001100110000110011  
0000110011001110001  
10

CÓDIGO BINARIO

MÁQUINA  
(RISC-V)

Memoria

0x00

0x25

0x05

0x13

3

CICLO DE INSTRUCCIÓN

Programa

Memoria

0x00

0x25

0x05

0x13

MAQUINA DE RISC-V

PC

x0

x1

...

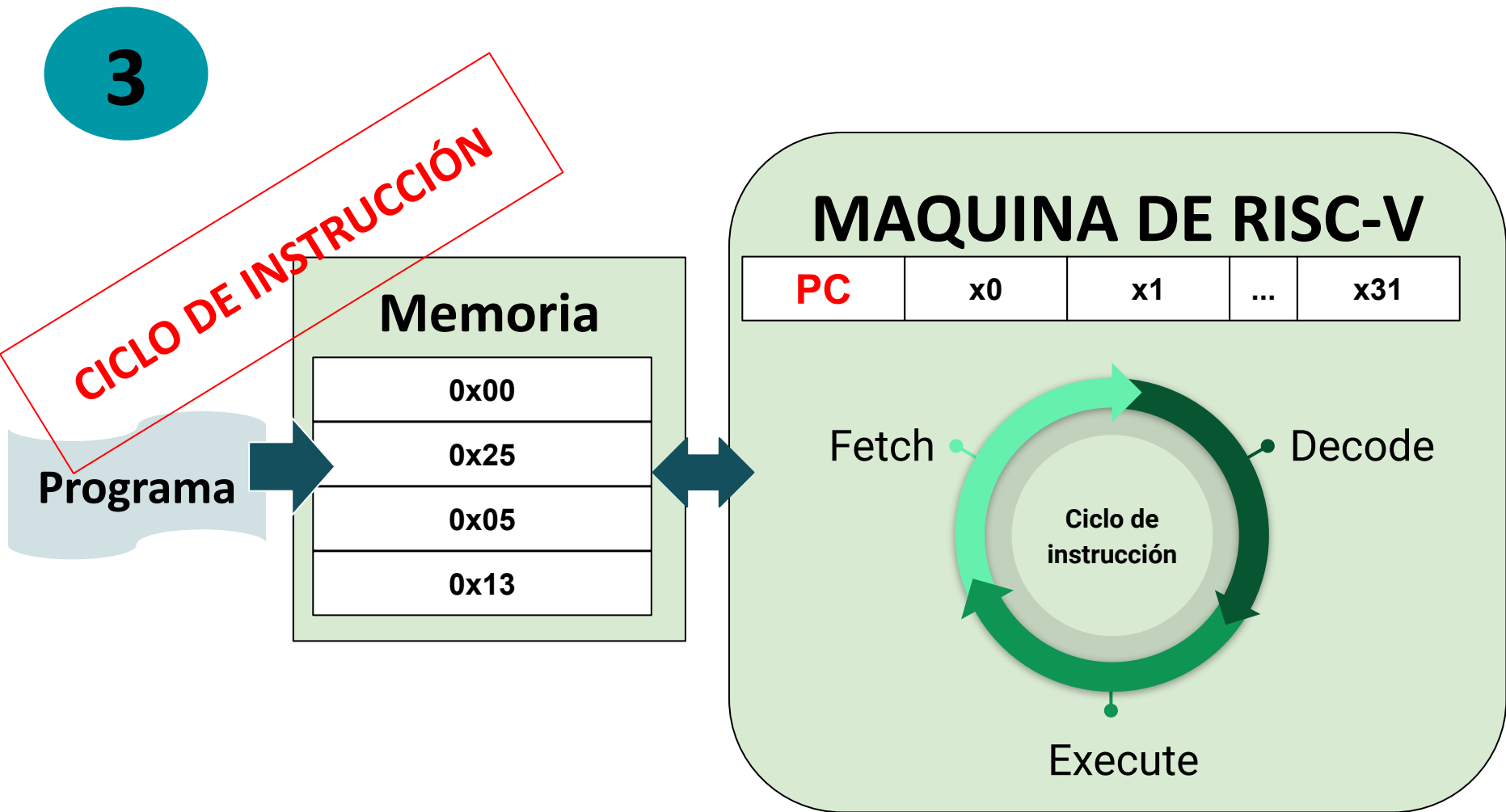
x31

Fetch

Decode

Ciclo de  
instrucción

Execute



# Codificación de programas en RISC-V

**ADDI a0, a0, 0x002**

Usamos el  
manual de  
RISC-V

**Código de  
Operación indica  
qué operación**

**Primer  
Operando  
DESTINO (d)**

**Segundo y  
Tercero  
Operandos  
FUENTES (f)**

# Codificación de programas en RISC-V

## Instrucción: ADDI (add immediate)

rd = registro destino  
rs1 = registro fuente 1

extensiones

efecto

**addi** rd, rs1, immediate

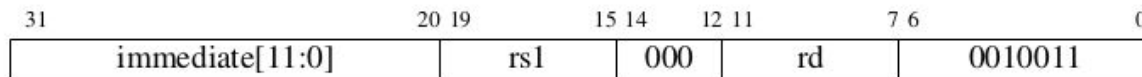
$x[rd] = x[rs1] + \text{sext}(\text{immediate})$

formato

*Add Immediate*: Tipo I, RV32I y RV64I.

Suma el *inmediato* sign-extended al registro  $x[rs1]$  y escribe el resultado en  $x[rd]$ . Overflow aritmético ignorado.

Formas comprimidas: **c.li** rd, imm; **c.addi** rd, imm; **c.addi16sp** imm; **c.addi4spn** rd, imm



# Codificación de programas en RISC-V

**ADDI a0, a0, 0x002**

**Código de  
Operación:  
0010011**

**Operando  
DESTINO:  
01010**

**Operandos  
FUENTES:  
01010  
0000 0000 0010**

imm[11:0]	rs1	000	rd	0010011	I addi
-----------	-----	-----	----	---------	--------

0000 0000 0010	01010	000	01010	0010011
----------------	-------	-----	-------	---------

**0000 0000 0010 0101 0000 0101 0001 0011 = 0x00250513**

# Codificación de programas en RISC-V

**Ejemplo de Seguimiento en la planilla de seguimiento**



# Para usar Ripes en los labos

En UBUNTU acceder

Usuario: Estudiante

contraseña: Estudiante

## En la terminal

```
$cd /media/libre
```

```
$/Ripes-v2.2.6-linux-x86_64.Appimage
```



FileEditViewHelp

100101001Editor

Processor

Cache

Memory

I/O

Source code

Input type: AssemblyExecutable code

View mode: BinaryDisassembled

100 ms

1

Console

Select Processor

RISC-V

32-bit

Single-cycle processor

5-stage processor w/o forwarding or hazard...

5-stage processor w/o hazard detection

5-stage processor w/o forwarding unit

5-stage processor

6-stage dual-issue processor

64-bit

Name:Single-cycle processor

ISA:RV32I

ISA Exts.☐ M☐ C

LayoutStandard

Description:A single cycle processor

Register initialization

x2 (sp)0x7ffffff0

x3 (gp)0x10000000

Cancel

OK

GPR

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
x8	s0	0x00000000
x9	s1	0x00000000
x10	a0	0x00000000
x11	a1	0x00000000
x12	a2	0x00000000
x13	a3	0x00000000
x14	a4	0x00000000
x15	a5	0x00000000
x16	a6	0x00000000
x17	a7	0x00000000
x18	s2	0x00000000
x19	s3	0x00000000
x20	s4	0x00000000
x21	s5	0x00000000
x22	s6	0x00000000
x23	s7	0x00000000
x24	s8	0x00000000
x25	s9	0x00000000
x26	s10	0x00000000

Display type:Hex