



Examen final

23 de mayo de 2025

Fundamentos de computadores II

*Dpto. Arquitectura de Computadores y Automática
Universidad Complutense de Madrid*





Ejercicios 1, 2 y 4

①

Instrucciones: $16 \times 4B = 64B$

Datos: $2 \times 4B = 8B$

Total: $64B + 8B = 72B$

```
sqrt:
    mv    t0, zero
    li    t1, 1
for:
    blt   a0, t1, efor
    sub   a0, a0, t1
    addi  t0, t0, 1
    addi  t1, t1, 2
    j     for
efor:
    mv    a0, t0
    ret

main:
    la    sp, _stack
    la    t0, num
    lw    a0, 0(t0)
    call  sqrt
    la    t0, raiz
    sw    a0, 0(t0)
    j     .
```

} 5+1 veces
(5 no salta, 1 salta)
} 5 veces

②

Al ser función hoja, todo es correcto

```
sqrt:
    addi  sp, sp, -8
    sw    s0, 0(sp)
    sw    s1, 4(sp)
    mv    s0, zero
    li    s1, 1
for:
    blt   a0, s1, efor
    sub   a0, a0, s1
    addi  s0, s0, 1
    addi  s1, s1, 2
    j     for
efor:
    mv    a0, s0
    lw    s0, 0(sp)
    lw    s1, 4(sp)
    addi  sp, sp, 8
    ret
```

④

Dir.	+0	+1	+2	+3
0x00000010	93	82	12	00

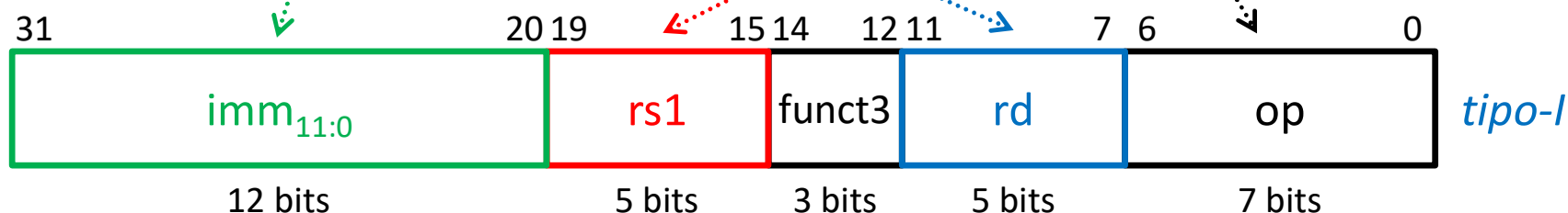
$$N = 4 + [2 + (5 \times 5 + 1) + 2] + 2 = 36$$

RISC-V: Little-Endian



Ejercicio 3

addi **rs1**, **rs2**, **imm**



0000 0000 0001 0010 1000 0010 1001 0011

0x00128293

000000000001

00101

000

00101

0010011

addi **x5**, **x5**, **1**

000000000010

00110

000

00110

0010011

addi **x6**, **x6**, **2**

0000 0000 0010 0011 0000 0011 0001 0011

0x00230313



Ejercicio 5

Tabla de verdad DEC principal (extracto)

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0000011 (<i>lw</i>)	0	0	1	1	00 ^(sumar)	0	00
0100011 (<i>sw</i>)	0	0	0	1	00 ^(sumar)	1	–
0010011 (tipo-I)	0	0	1	1	10 ^(operar)	0	01
0110011 (tipo-R)	0	0	1	0	10 ^(operar)	0	01
1100011 (<i>beq</i>)	1	0	0	0	01 ^(restar)	0	–
1101111 (<i>jal</i>)	0	1	1	–	–	0	10

MemWr = “1” $\Rightarrow$ *sw* a0, 0(t0)
que se ejecuta en el ciclo 36

BRwr = “1” $\Rightarrow$ *lw*, tipo-I, tipo-R o *jal*
ResSrc = “00” $\Rightarrow$ *lw* a0, 0(t0)
que se ejecuta en el ciclo 3

Jump = “1” $\Rightarrow$ *jal*
WA $\neq$ “00000” $\Rightarrow$ *rd* $\neq$ x0
call sqrt (jal ra, sqrt)
que se ejecuta en el ciclo 4

Tabla de verdad DEC ALU (extracto)

ALUop	op ₅	funct7 ₅	funct3	ALUctr
10 ^(operar)	1	1	000 ^(sub)	001 ^(A – B)

BRwr = “1” $\Rightarrow$ *lw*, tipo-I, tipo-R o *jal*
ResSrc = “01” $\Rightarrow$ tipo-I o tipo-R
ALUctr = “001” $\Rightarrow$ *sub* a0, a0, t1
que se ejecuta en los ciclos 8, 13, 18, 23 y 28

Ejercicio 6 y 10



⑥

ciclo	RT	ALUSrcA	ALUSrcB	ResSrc
1	$IR \leftarrow \text{Mem}[PC], PC \leftarrow PC+4, \text{OldPC} \leftarrow PC$	00	10	10
2	$A \leftarrow \text{BR}[rs1], B \leftarrow \text{BR}[rs2], \text{ALUout} \leftarrow \text{OldPC} + \text{sExt}(\text{imm})$	01	01	–
3	$PC \leftarrow A + \text{sExt}(\text{imm})$	10	01	10
4	$\text{BR}[rd] \leftarrow \text{OldPC} + 4$	01	10	10

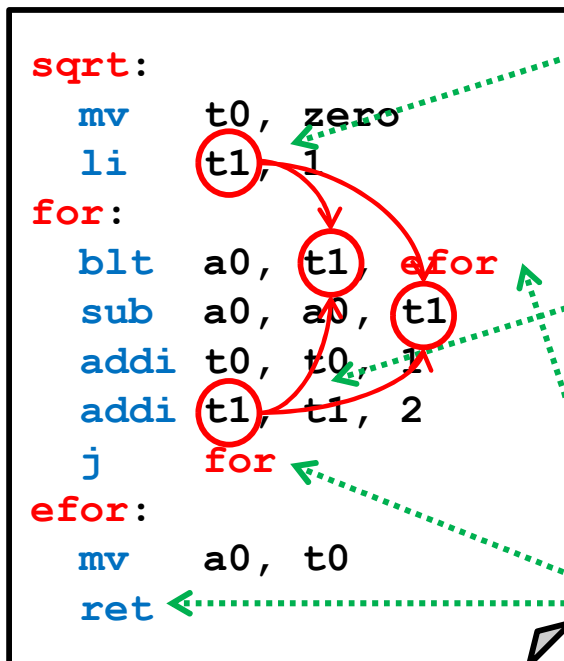
- No, es necesario hacer 4 sumas y solo hay una ALU.

⑩

- Excepción de dirección de instrucción desalineada, ya que la instrucción ret cargará el PC con un número impar y cuando haga el fetch de dicha dirección el procesador disparará la excepción por no ser múltiplo de 4.
- El último valor que en este programa toma el registro x1 es 11, luego esa es la dirección que tratará de hacer fetch y es la que se almacenará en mepc.
- En el ciclo siguiente al ciclo en que se ejecuta ret: $4+30+1 = 35$
- La RTE se comenzará a ejecutar en el ciclo siguiente al que la instrucción causante llegue a MEM: $i+4$



Ejercicio 7



Riesgos de datos
(primera iteración)

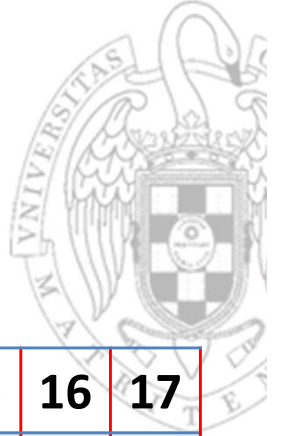
Riesgos de datos
(cada iteración)

Riesgos de control

Solución

```
sqrt:
    li    t1, 1
    mv    t0, zero
    nop
    nop
for:
    blt   a0, t1, efor
    nop
    nop
    sub   a0, a0, t1
    addi  t0, t0, 1
    addi  t1, t1, 2
    j     for
    nop
    nop
efor:
    mv    a0, t0
    ret
    nop
    nop
```

Ejercicio 8



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
mv t0, zero	IF	ID	EX	M	WB												
li t1, 1		IF	ID	EX	M	WB											
blt a0, t1, efor			IF	ID	EX	M	WB										
sub a0, a0, t1				IF	ID	EX	M	WB									
addi t0, t0, 1					IF	ID	EX	M	WB								
addi t1, t1, 2						IF	ID	EX	M	WB							
j for							IF	ID	EX	M	WB						
mv a0, t0								IF	ID	EX	M	WB					
ret									IF	ID	EX	M	WB				
blt a0, t1, efor										IF	ID	EX	M	WB			
sub a0, a0, t1											IF	ID	EX	M	WB		
addi t0, t0, 1												IF	ID	EX	M	WB	
mv a0, t0													IF	ID	EX	M	WB
ret														IF	ID	EX	M
la sp, _stack															IF	ID	EX
la t0, num																IF	EX
la t0, raiz																	IF



Ejercicio 9

```
sqrt:
    mv    t0, zero
    li    t1, 1
for:
    blt   a0, t1, efor
    sub   a0, a0, t1
    addi  t0, t0, 1
    addi  t1, t1, 2
    j     for
efor:
    mv    a0, t0
    ret
```

} 5+1 veces
(5 no salta, 1 salta)
} 5 veces

$$N = 2 + (5 \times 5 + 1) + 2 = 30$$

```
sqrt2:
    mv    t0, zero
    li    t1, 1
    blt   a0, t1, efor
for:
    sub   a0, a0, t1
    addi  t0, t0, 1
    addi  t1, t1, 2
    bge   a0, t1, for
efor:
    mv    a0, t0
    ret
```

} 5 veces
} 5 veces
(4 salta, 1 no salta)

$$N = 3 + 4 \times 5 + 2 = 25$$

monociclo	multiciclo	segmentado
1	4	1
1	4	1
1	3	1 o 3
1	4	1
1	4	1
1	4	1
1	4	3
1	4	1
1	4	3
1	4	1
1	4	1
1	3	1
1	4	1
1	4	1
1	4	1
1	3	1 o 3
1	4	1
1	4	3

ciclos por instrucción



Ejercicio 9

	sqrt(25)			sqrt2(25)		
	N	ciclos	CPI	N	ciclos	CPI
monociclo	30	30	1	25	25	1
multiciclo	30	114	3.80	25	94	3.76
segmentado	30	44	1.47	25	35	1.40

- En todas las microarquitecturas sqrt2 es más eficiente porque tarda un menor número de ciclos que sqrt.

	sqrt2(25)	
	ciclos	t_{ejec}
monociclo	25	$25 \cdot 5 \cdot t_{clk}$
multiciclo	94	$94 \cdot t_{clk}$
segmentado	35	$35 \cdot t_{clk}$

$$Speedup = \frac{t_{ejec}^{monociclo}}{t_{ejec}^{segmentado}} = \frac{125}{35} = 3.57$$

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/>