

# Organización de computadoras

2023

Agregado de instrucciones al Risc-V (didáctico)

## Agregado de instrucciones tipo-I (addi)

Las instrucciones tipo-I son similares a las de tipo-R.

Solo que la ALU trabaja sobre un registro y el campo inmediato.

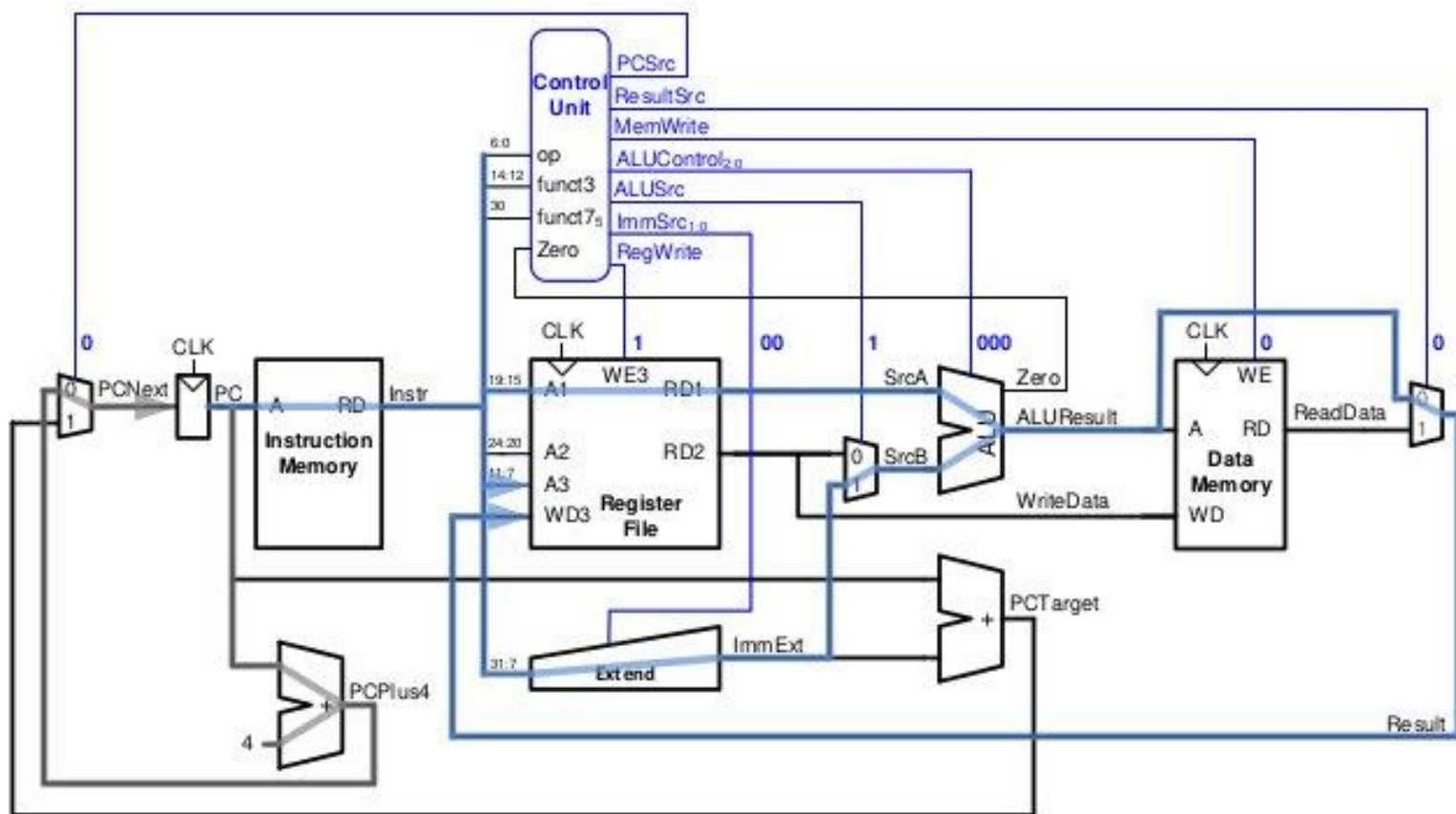
Debemos cambiar ALUSrc para seleccionar el campo inmediato.

Y la señal InmSrc para elegir el inmediato correcto

## Tabla Control agregando addi

op	Instruct.	RegWrite	ImmSrc	ALUSrc	MemWrite	ResultSrc	Branch	ALUOp
3	lw	1	00	1	0	1	0	00
35	sw	0	01	1	1	X	0	00
51	R-type	1	XX	0	0	0	0	10
99	beq	0	10	0	0	X	1	01
19	I-type	1	00	1	0	0	0	10

# Camino de datos - addi



**Descanso !**

# Agregado de instrucciones tipo-j (jal)

Similar a beq, Pero el salto siempre se ejecuta

PCSrc debe ser 1

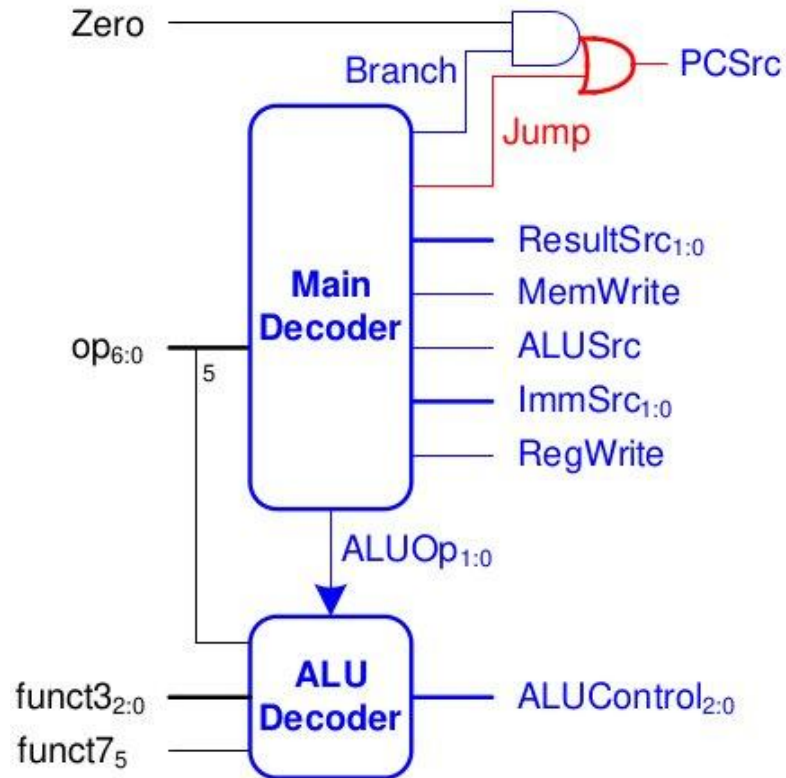
El formato inmediato es diferente

Necesita un nuevo ImmSrc de 11

Y jal debe calcular PC+4 y almacenarlo en rd

Tomar PC+4 del sumador a través de ResultMux

# Control - jal





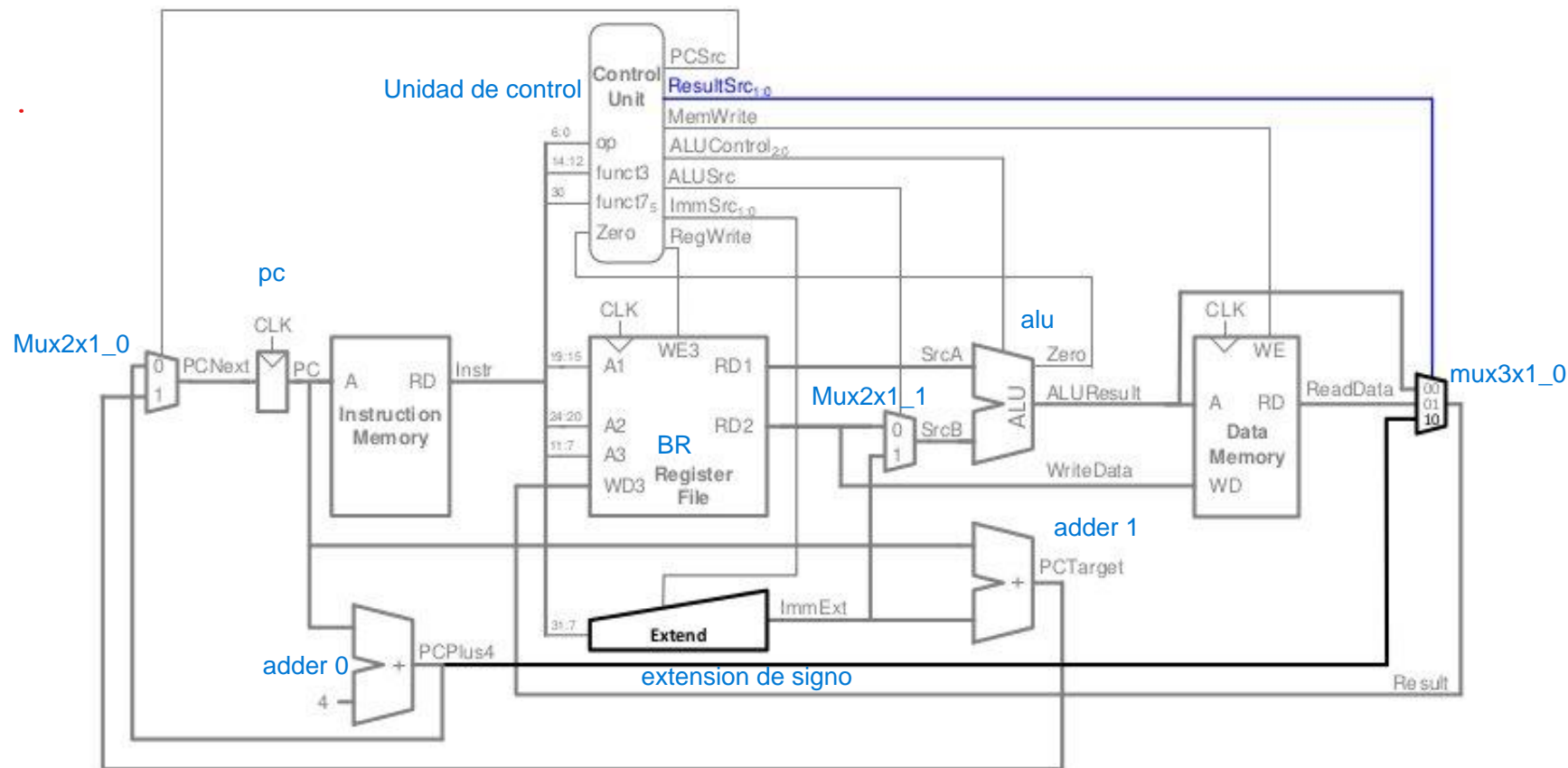
# Campo immediato - jal

ImmSrc <sub>1:0</sub>	ImmExt	Instruction Type
00	{{20{instr[31]}}, instr[31:20]}	I-Type
01	{{20{instr[31]}}, instr[31:25], instr[11:7]}	S-Type
10	{{19{instr[31]}}, instr[31], instr[7], instr[30:25], instr[11:8], 1'b0}	B-Type
<b>11</b>	<b>{{12{instr[31]}}, instr[19:12], instr[20], instr[30:21], 1'b0}</b>	<b>J-Type</b>

## Tabla Control agregando jal

op	Instruct.	RegWrite	ImmSrc	ALUSrc	MemWrite	ResultSrc	Branch	ALUOp	Jump
3	lw	1	00	1	0	01	0	00	0
35	sw	0	01	1	1	XX	0	00	0
51	R-type	1	XX	0	0	00	0	10	0
99	beq	0	10	0	0	XX	1	01	0
19	I-type	1	00	1	0	00	0	10	0
111	jal	1	11	X	0	10	0	XX	1

# Camino de datos - jal



**that's all !**