

Ejercicio añadir instrucción

En las transparencias 23-29 hemos visto como se puede implementar las instrucciones de la tabla y en el proyecto, tu equipo ha generado una implementación del Z23.

type	Action	Instruction format															
ALI(rd, rs, imm)	$R[rd] \leftarrow R[rs] - func - imm$	opc	rd	rs	func				imm								
		0	0	1													
ALU(rd, rs)	$R[rd] \leftarrow R[rs] - func - R[rd]$	opc	rd	rs	func												
		0	1	0					x	x	x	x	x	x	x	x	
Load(rd, addr)	$R[rd] \leftarrow M(addr)$	opc	rd									addr					
		0	1	1		x	x	x	x	x	x						
Store(rd, addr)	$M(addr) \leftarrow R[rd]$	opc	rd									addr					
		0	0	0		x	x	x	x	x	x						
Instruction final		opc															
		1	1	1	x	x	x	x	x	x	x	x	x	x	x	x	

En el camino de datos vamos a introducir una instrucción de salto condicional:

type	Action	Instruction format													
BN(rd, rs, addr)	If (R[rd] ≠ R[rs]), PC← addr	opc	rd	rs						addr					
		1	0	0			x	x	x	x	x				

- Implementa el camino de datos y la unidad del control correspondiente en tu Z23 en logisim
- Carga el programa y genera una tabla de simulación que demuestre al menos . **PC, instrucción, R0, R1 y M(5)**
Sube el circuito, el programa.txt y (cuando puedes) una captura de la tabla de simulacion al taller Z23

Programa a ejecutar

PC	mnemotecnico	accion	Código Hex
0	Adi(r1,r0,1)	$R1 \leftarrow R0 + 1$	3201
1	Mov(r0,r1)	$R0 \leftarrow R1$	2800
2	Mulu(r0,r0)	$R0 \leftarrow R0 * R0$	4400
3	Muli(r1,r1,2)	$R1 \leftarrow 2 * R1$	3c02
4	Store(r1,5)	$M(5) \leftarrow R1$	1005
5	BN(r0,r1,1)		8801
6	LOAD(r0,5)	$R0 \leftarrow M(5)$	6005
7	final		f100