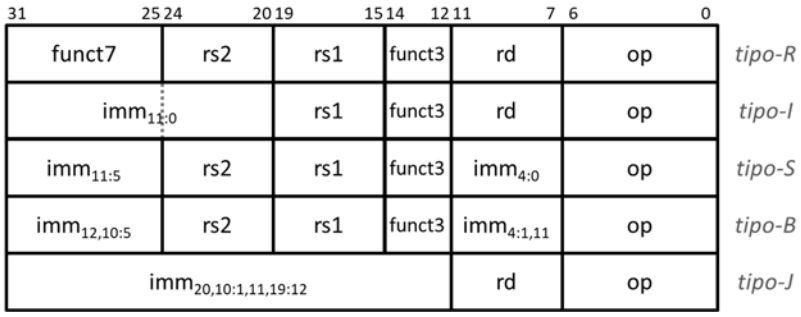
Facultad de Informática
Universidad Complutense de Madrid

FUNDAMENTOS DE COMPUTADORES II
RISC-V DE ARQUITECTURA REDUCIDA

`lw` ***rd, imm$_{12b}$(rs1)***	rd ← Mem[rs1 + sExt(imm)]	tipo-I
`sw` ***rs2, imm$_{12b}$(rs1)***	Mem[rs1 + sExt(imm$_{12b}$)] ← rs2	tipo-S
`add` ***rd, rs1, rs2***	rd ← rs1 + rs2	tipo-R
`sub` ***rd, rs1, rs2***	rd ← rs1 – rs2	tipo-R
`and` ***rd, rs1, rs2***	rd ← rs1 & rs2	tipo-R
`or` ***rd, rs1, rs2***	rd ← rs1 | rs2	tipo-R
`slt` ***rd, rs1, rs2***	rd ← *if* (rs1 <$_S$ rs2) *then* (1) *else* (0)	tipo-R
`addi` ***rd, rs1, imm$_{12b}$***	rd ← rs1 + sExt(imm),	tipo-I
`andi` ***rd, rs1, imm$_{12b}$***	rd ← rs1 & sExt(imm)	tipo-I
`ori` ***rd, rs1, imm$_{12b}$***	rd ← rs1 | sExt(imm)	tipo-I
`slti` ***rd, rs1, imm$_{12b}$***	rd ← *if* (rs1 <$_S$ sExt(imm)) *then* (1) *else* (0)	tipo-I
`beq` ***rs1, rs2, imm$_{13b}$***	PC ← *if* (rs1 = rs2) *then* (PC + sExt(imm$_{12:1}$ << 1)) *else* (PC+4)	tipo-B
`jal` **rd, imm$_{21b}$**	PC ← PC + sExt(imm$_{20:1}$ << 1), rd ← PC+4	tipo-J



Instrucción	Tipo	funct7 bits 31:25	funct3 bits 14:12	op bits 6:0
lw	I	–	010	0000011
sw	S	–	010	0100011
add	R	0000000	000	0110011
sub	R	0100000	000	
slt	R	0000000	010	
or	R	0000000	110	
and	R	0000000	111	
addi	I	–	000	0010011
slti	I	–	010	
ori	I	–	110	
andi	I	–	111	
beq	B	–	000	1100011
jal	J	–	–	1101111