



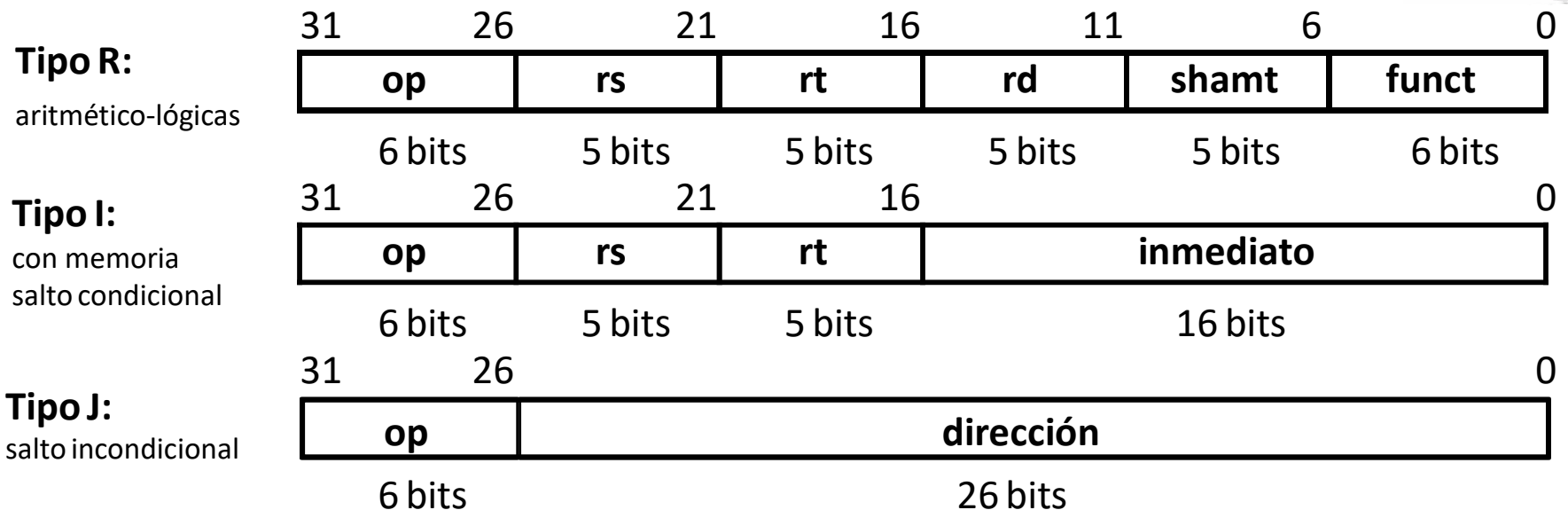
Práctica 6

MIPS multicycle

Repertorio de instrucciones del MIPS



Formato de las instrucciones



El significado de los campos es:

- **op**: identificador de instrucción
- **rs, rt, rd**: identificadores de los registros fuentes y destino
- **shamt**: cantidad a desplazar (en operaciones de desplazamiento)
- **funct**: selecciona la operación aritmética a realizar
- **inmediato**: es el valor del desplazamiento en direccionamiento a registro-base
- **dirección**: dirección destino del salto

Repertorio de instrucciones del MIPS



■ Instrucciones de referencia a memoria:

- lw rt, inmed(rs) $rt \leftarrow \text{Memoria}(rs + \text{SignExt}(\text{inmed})), PC \leftarrow PC + 4$
- sw rt, inmed(rs) $\text{Memoria}(rs + \text{SignExt}(\text{inmed})) \leftarrow rt$

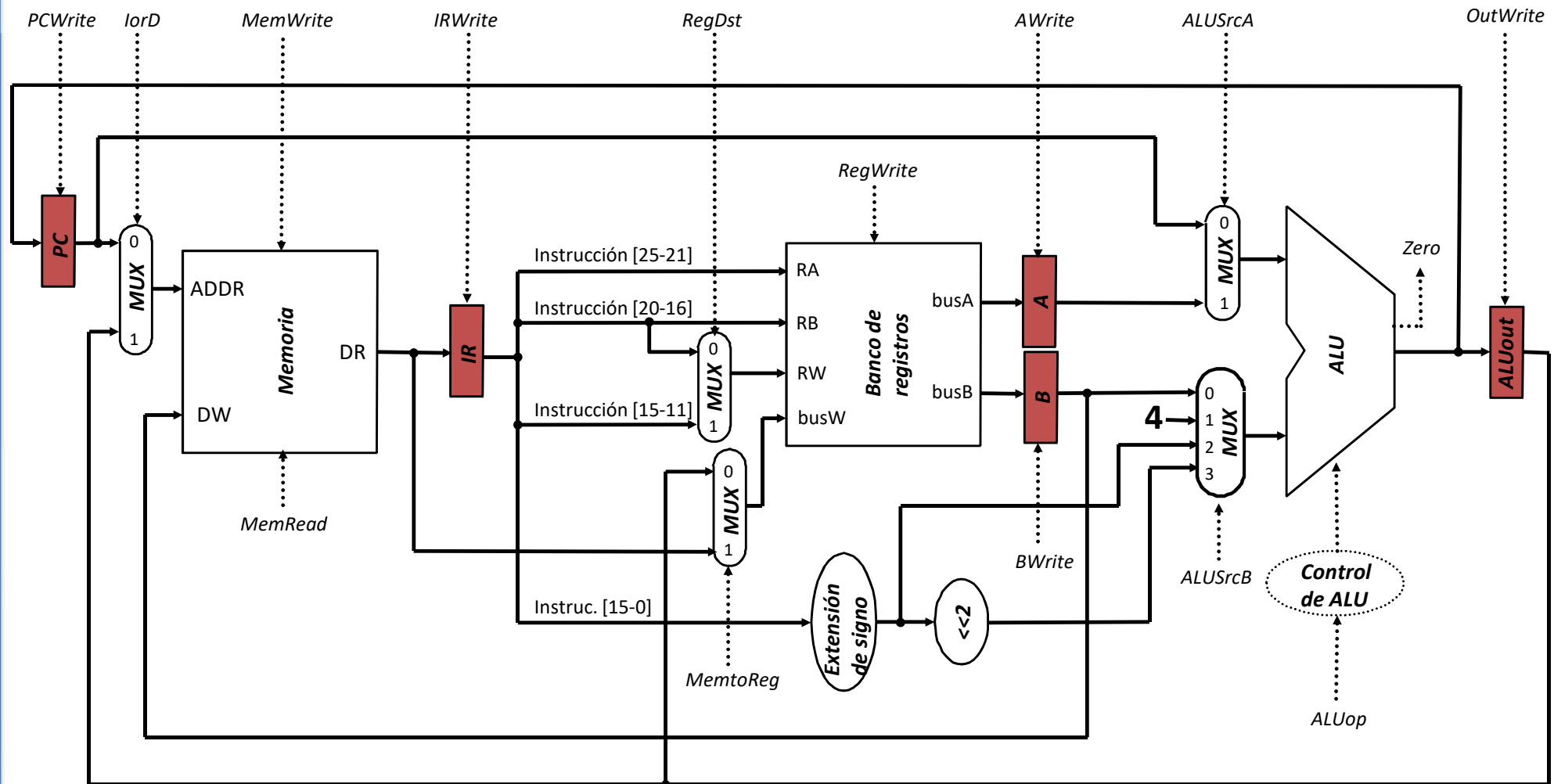
■ Instrucciones aritmético-lógicas con operandos en registros:

- add rd, rs, rt $rd \leftarrow rs + rt, PC \leftarrow PC + 4$
- sub rd, rs, rt $rd \leftarrow rs - rt, PC \leftarrow PC + 4$
- and rd, rs, rt $rd \leftarrow rs \text{ and } rt, PC \leftarrow PC + 4$
- or rd, rs, rt $rd \leftarrow rs \text{ or } rt, PC \leftarrow PC + 4$
- nor rd, rs, rt $rd \leftarrow rs \text{ nor } rt, PC \leftarrow PC + 4$
- xor rd, rs, rt $rd \leftarrow rs \text{ xor } rt, PC \leftarrow PC + 4$

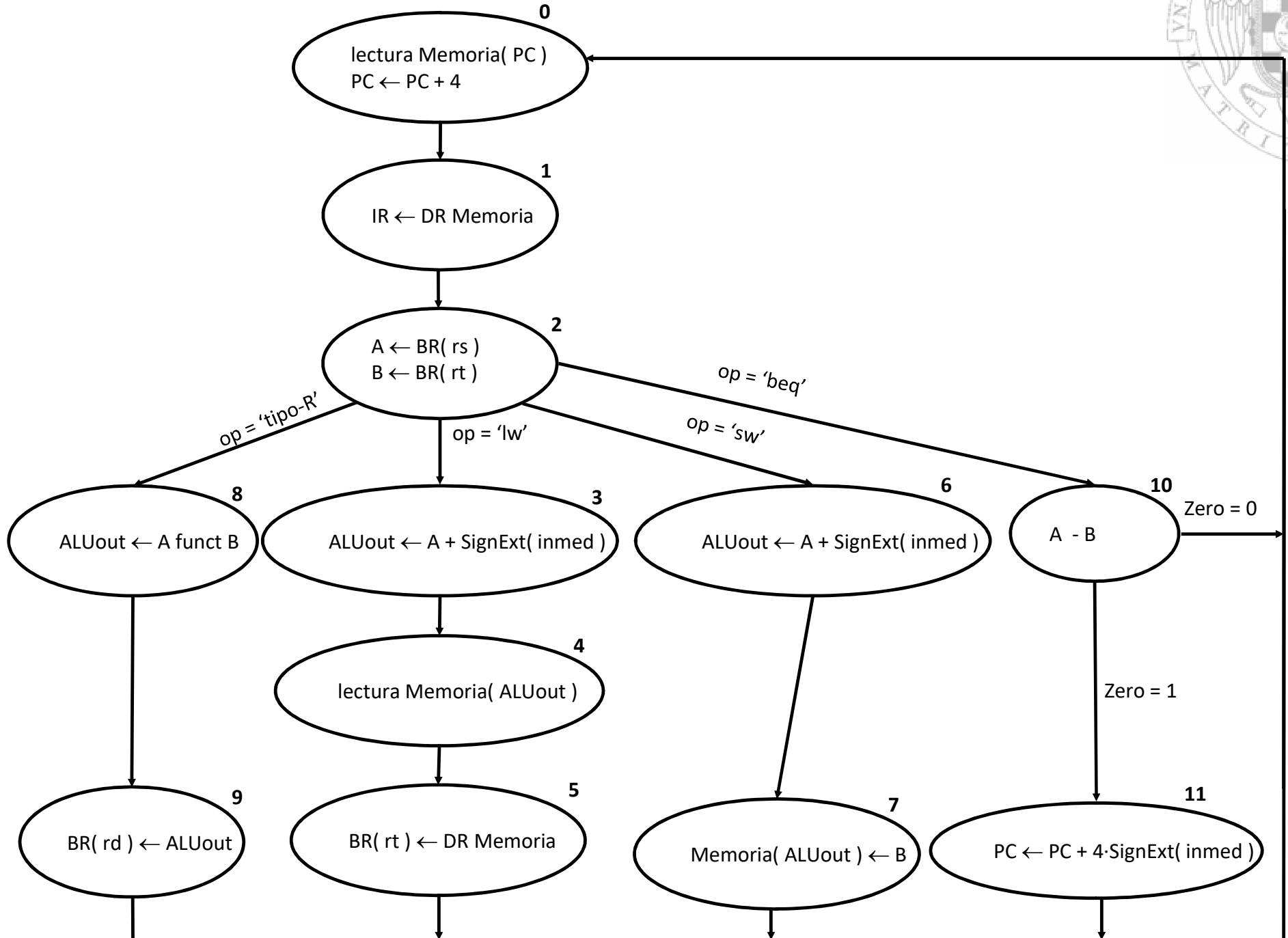
■ Instrucciones de salto condicional:

- beq rs, rt, inmed si $(rs = rt)$ entonces $(PC \leftarrow PC + 4 + 4 * \text{SignExt}(\text{inmed}))$
 en otro caso $PC \leftarrow PC + 4$

Diseño de la ruta de datos multiciclo



Diseño del controlador multiciclo



Diseño del controlador multiciclo



Estado actual	op	Zero	Estado siguiente	IRWrite	PCWrite	AWrite	BWrite	ALUSrcA	ALUSrcB	ALUOp	OutWrite	MemWrite	MemRead	lorD	MDRWrite	MemtoReg	RegDest	RegWrite
0000	XXXXXX	X	0001	0	1			0	01	00 (add)		0	1	0				0
0001	XXXXXX	X	0010	1	0							0	0	0				0
0010	100011 (lw)	X	0011	0	0	1	1					0	0					0
0010	101011 (sw)	X	0110															
0010	000000 (tipo-R)	X	1000															
0010	000100 (beq)	X	1010															
0011	XXXXXX	X	0100	0	0			1	10	00 (add)	1	0	0					0
0100	XXXXXX	X	0101	0	0							0	1	1	1			0
0101	XXXXXX	X	0000	0	0							0	0			1	0	1
0110	XXXXXX	X	0111	0	0		0	1	10	00 (add)	1	0	0					0
0111	XXXXXX	X	0000	0	0							1	0	1				0
1000	XXXXXX	X	1001	0	0			1	00	10 (funct)	1	0	0					0
1001	XXXXXX	X	0000	0	0							0	0			0	1	1
1010	XXXXXX	0	0000	0	0			1	00	01 (sub)		0	0					0
1010	XXXXXX	1	1011															
1011	XXXXXX	X	0000	0	1			0	11	00 (add)		0	0					0