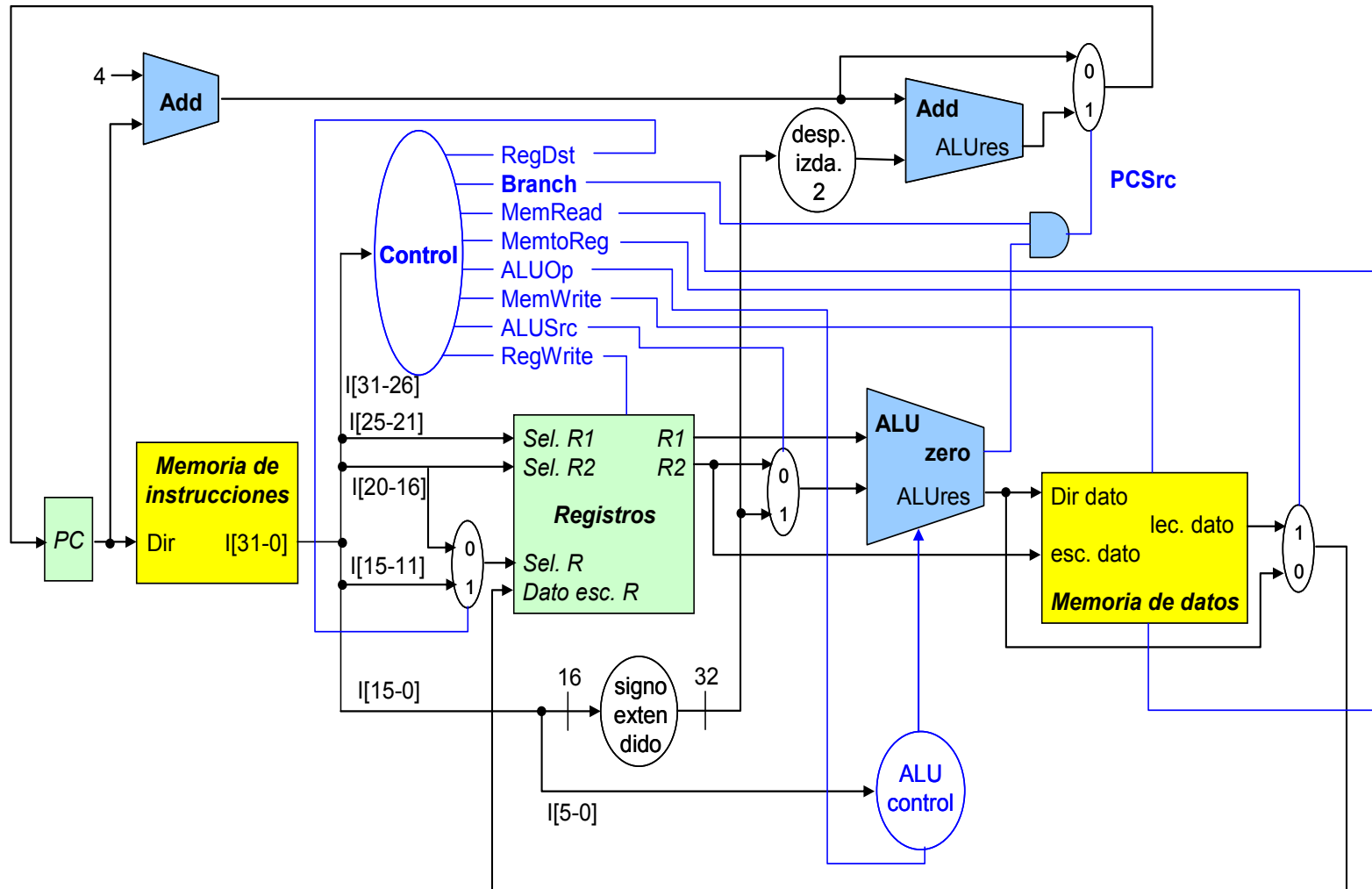
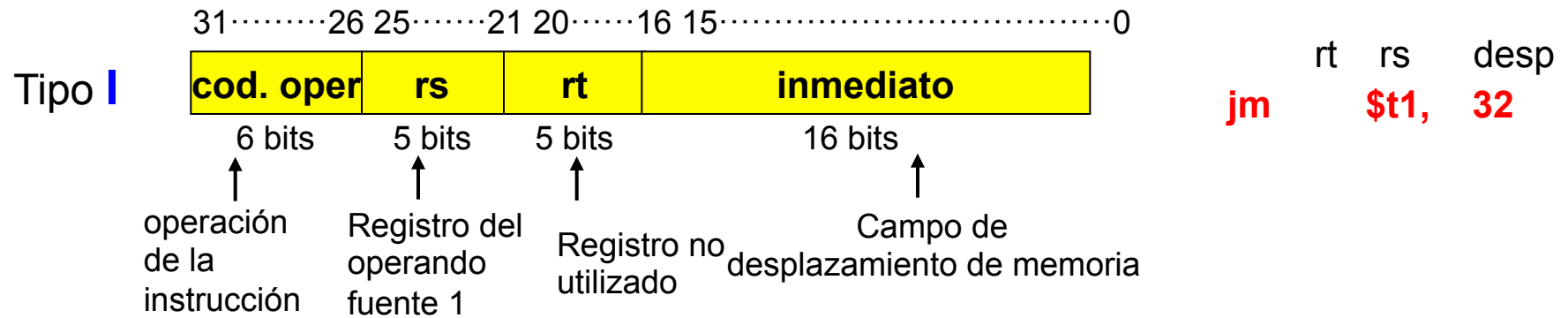


## EJERCICIO 5

Deseamos añadir al repertorio inicial de instrucciones de la CPU, modelo monociclo, la instrucción **jm**. Realizar las modificaciones necesarias en el datapath y en la unidad de control.

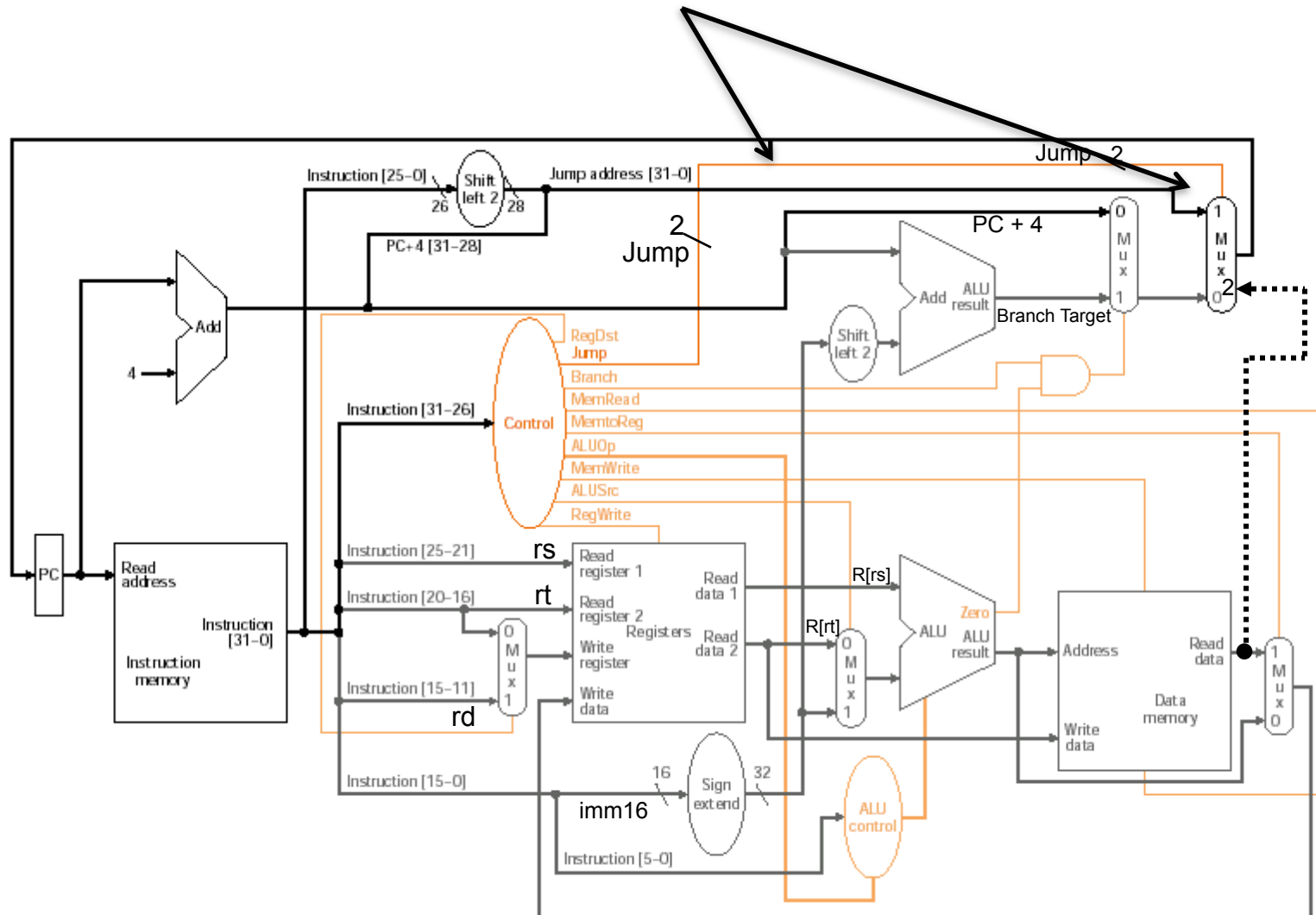




La instrucción **j**m carga una palabra desde una dirección de memoria dada (\$rs + offset), es similar a **lw** excepto porque la palabra cargada se coloca en el PC en lugar de en el registro \$rt

$$PC \leftarrow Mem[R[rs] + SignExt[inm16]]$$

Expandimos el multiplexor controlado por Jump para incluir la salida de la lectura de memoria como una nueva entrada 2. La señal de control Jump tiene ahora 2 bits.



La señal de control Jump tiene ahora 2 bits.

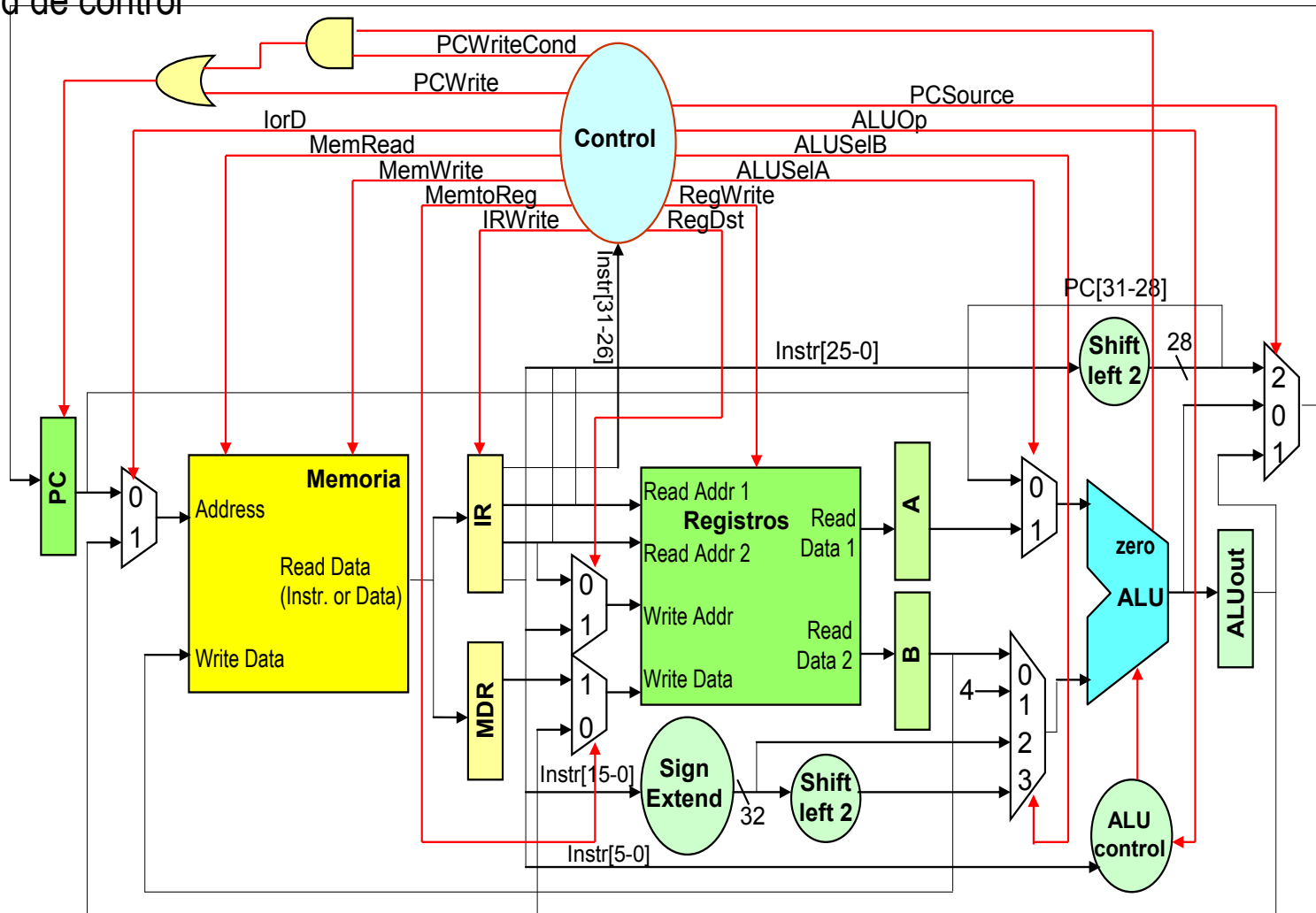
Instrucción	RegDst	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch	Jump	ALUOp1	ALUOp2
R	1	0	0	1	0	0	0	00	1	0
lw	0	1	1	1	1	0	0	00	0	0
sw	x	1	x	0	0	1	0	00	0	0
jmp	x	x	x	0	0	0	x	01	x	x
jm	x	1	x	0	1	0	x	10	0	0

R[rs]

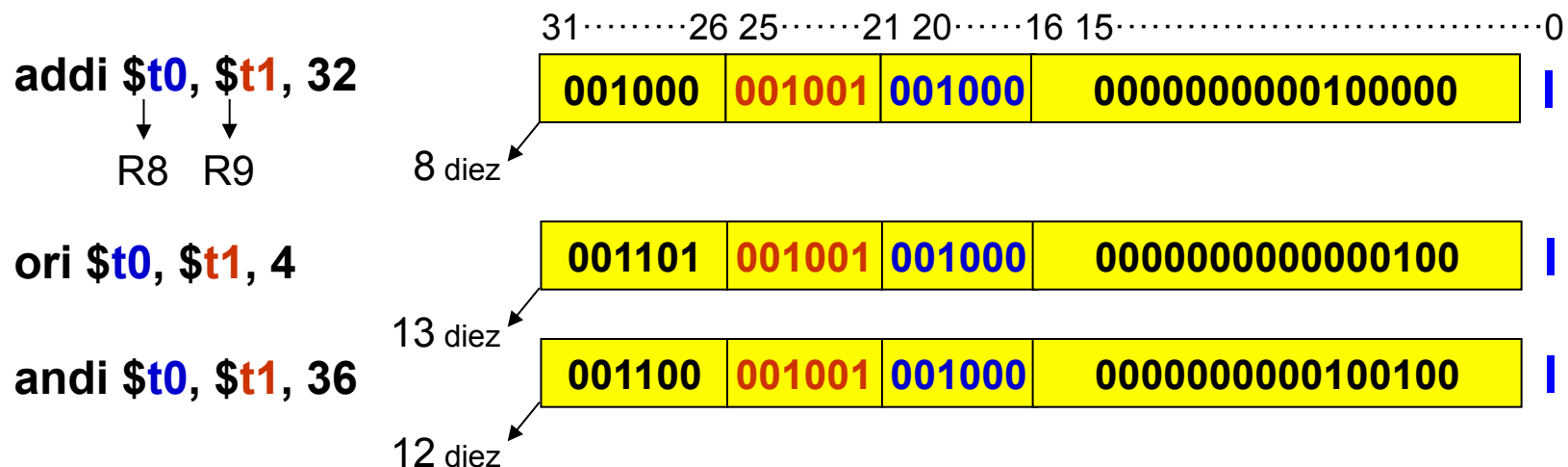
$PC \leftarrow Mem[R[rs] + SignExt(Inm16)]$

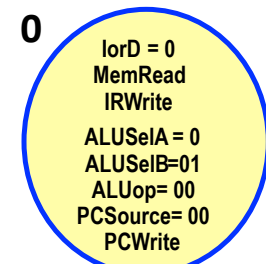
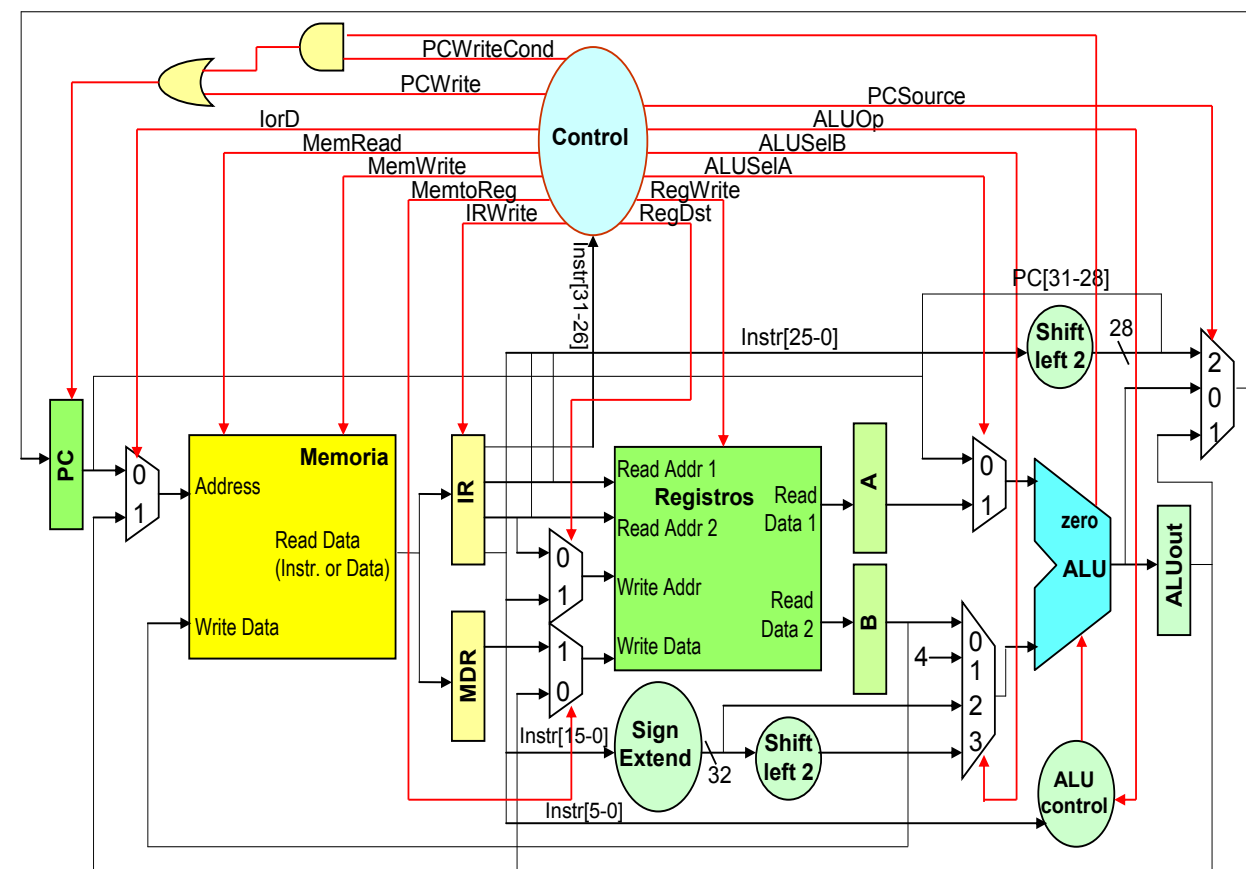
add

Deseamos añadir al repertorio inicial de instrucciones de la CPU, modelo multiciclo, la instrucción **addi**. Realizar las modificaciones necesarias en el datapath y en el autómata correspondiente a la unidad de control

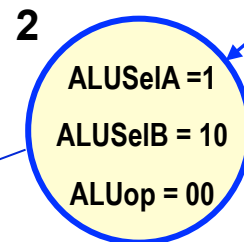


Instrucción:	Formato:
	<div> <div>op</div> <div>rs</div> <div>rt</div> <div>operando</div> <div> </div> </div>
<b>add immediate</b> (sumar inmediato)	<b>addi rt, rs, operando</b> Suma el inmediato con signo extendido al contenido del registro <b>rs</b> y coloca el resultado en el registro <b>rt</b>
<b>or immediate</b> (or inmediato)	<b>ori rt, rs, operando</b> Hace la operación lógica OR del inmediato con cero extendido y el contenido del registro <b>rs</b> y coloca el resultado en el registro <b>rt</b>
<b>and immediate</b> (and inmediato)	<b>andi rt, rs, operando</b> Hace la operación lógica AND del inmediato con cero extendido y el contenido del registro <b>rs</b> y coloca el resultado en el registro <b>rt</b>

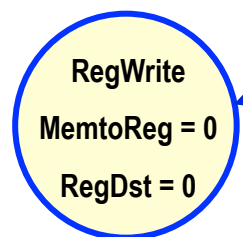




addi (lw, sw)



addi

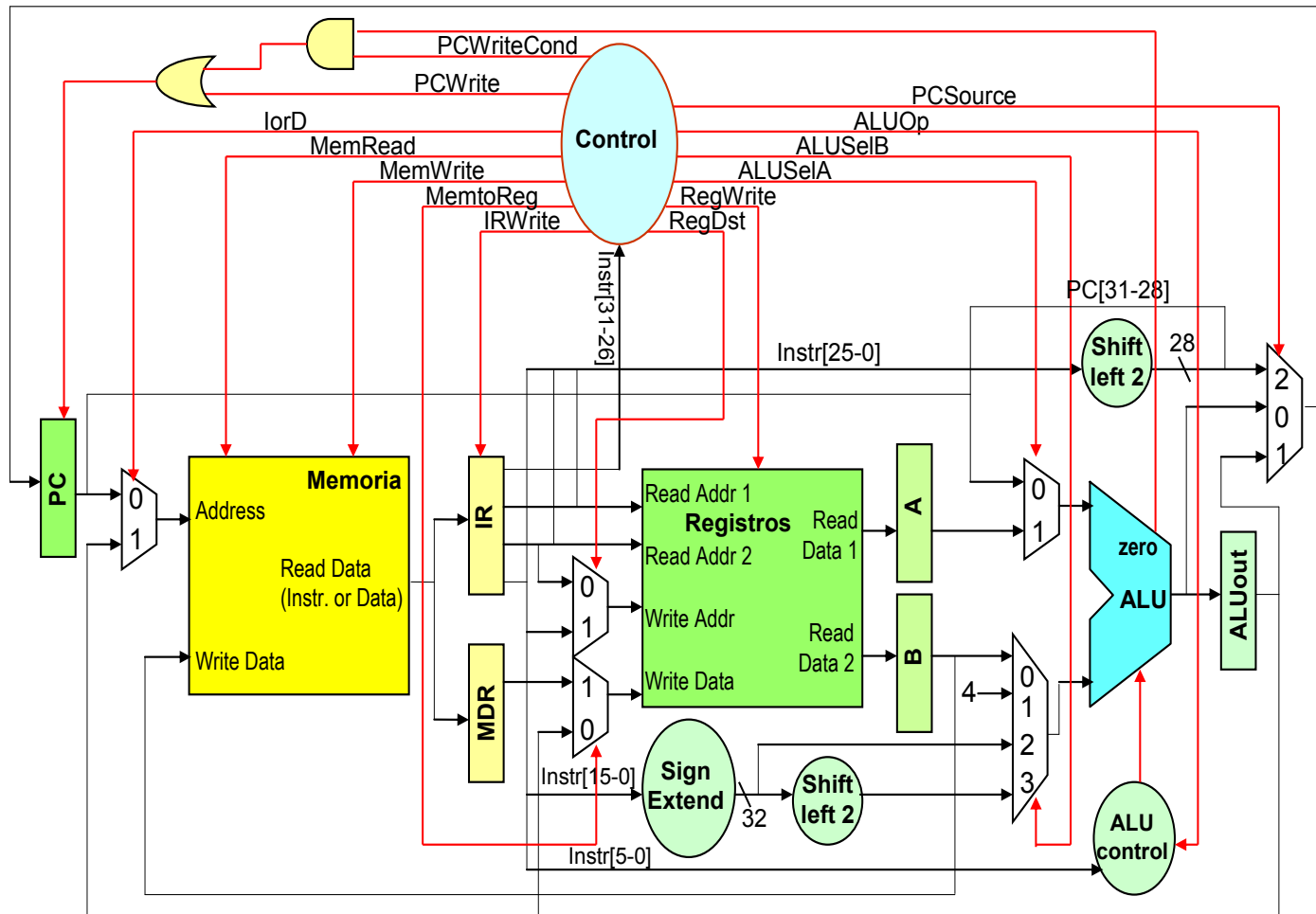


$ALUOut = A + \text{SignExt}(IR[15-0])$

$\text{Registros}(rt) = ALUOut$  Nuevo

## EJERCICIO 3

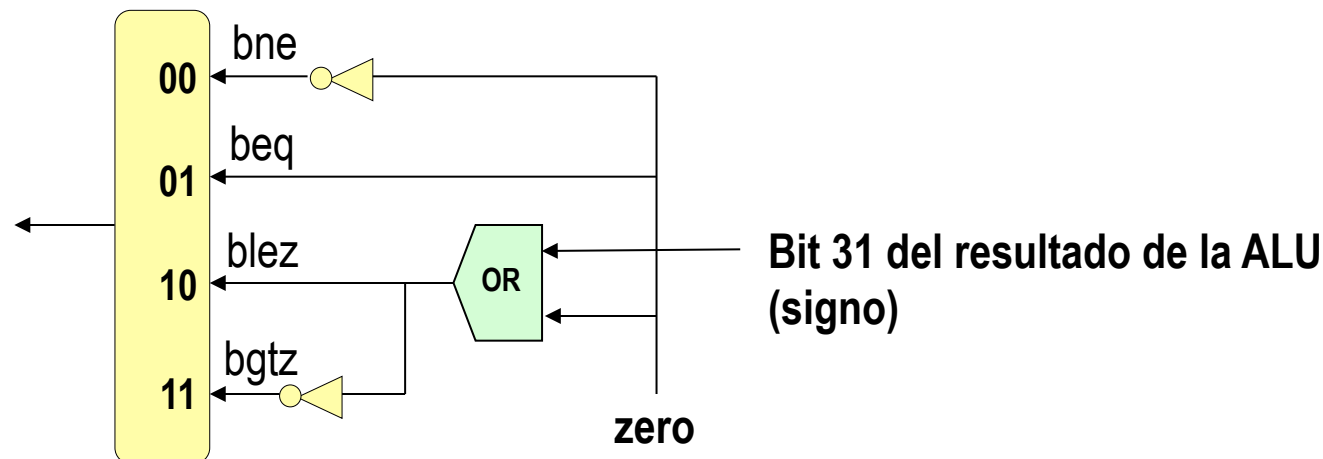
Deseamos añadir al repertorio inicial de instrucciones de la CPU, modelo multiciclo, las instrucciones **bne** (saltar si no igual), **blez** (saltar si menor o igual que 0) y **bgtz** (saltar si mayor que 0). Realizar las modificaciones necesarias en el datapath y en el autómata correspondiente a la unidad de control

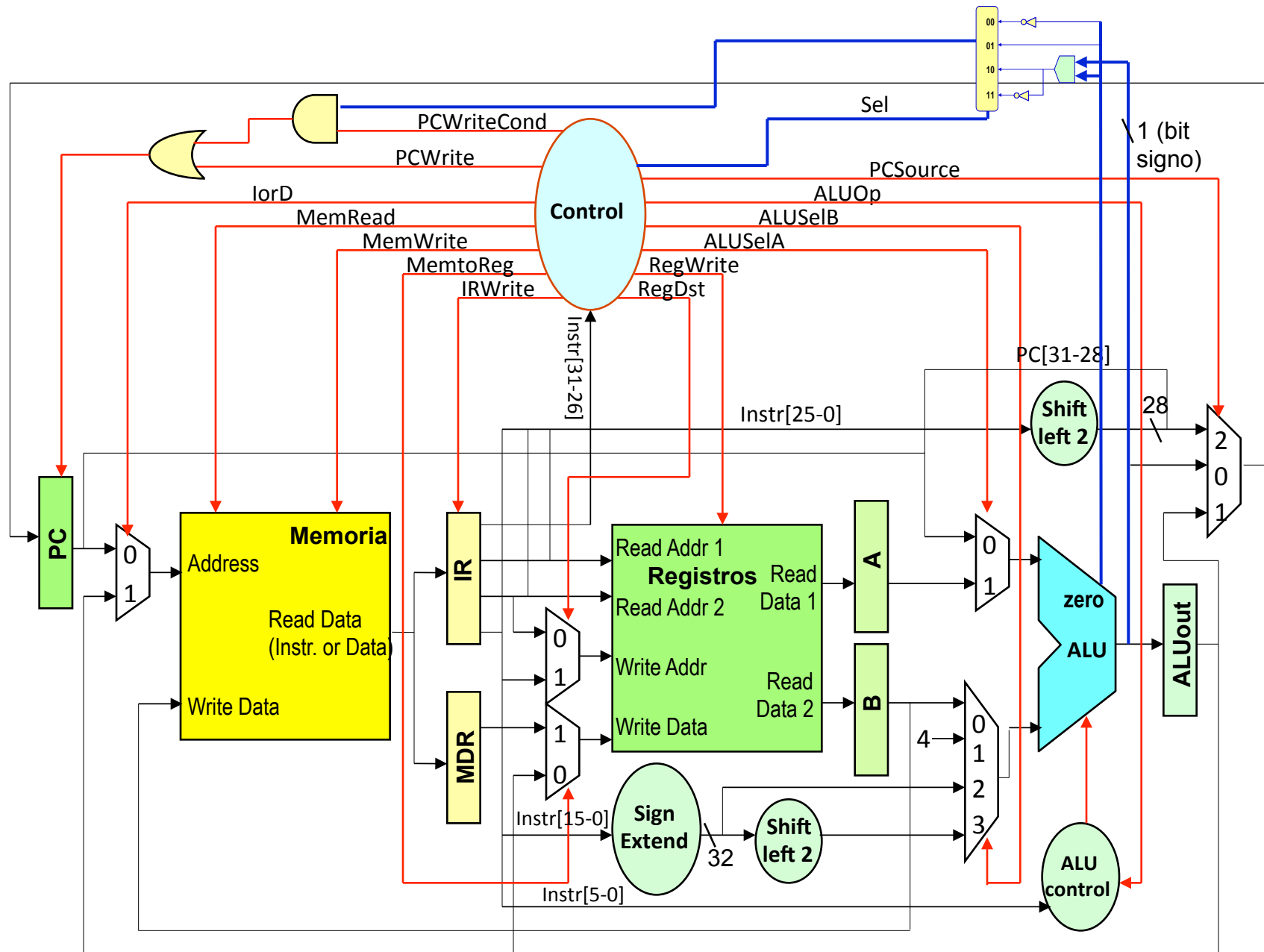


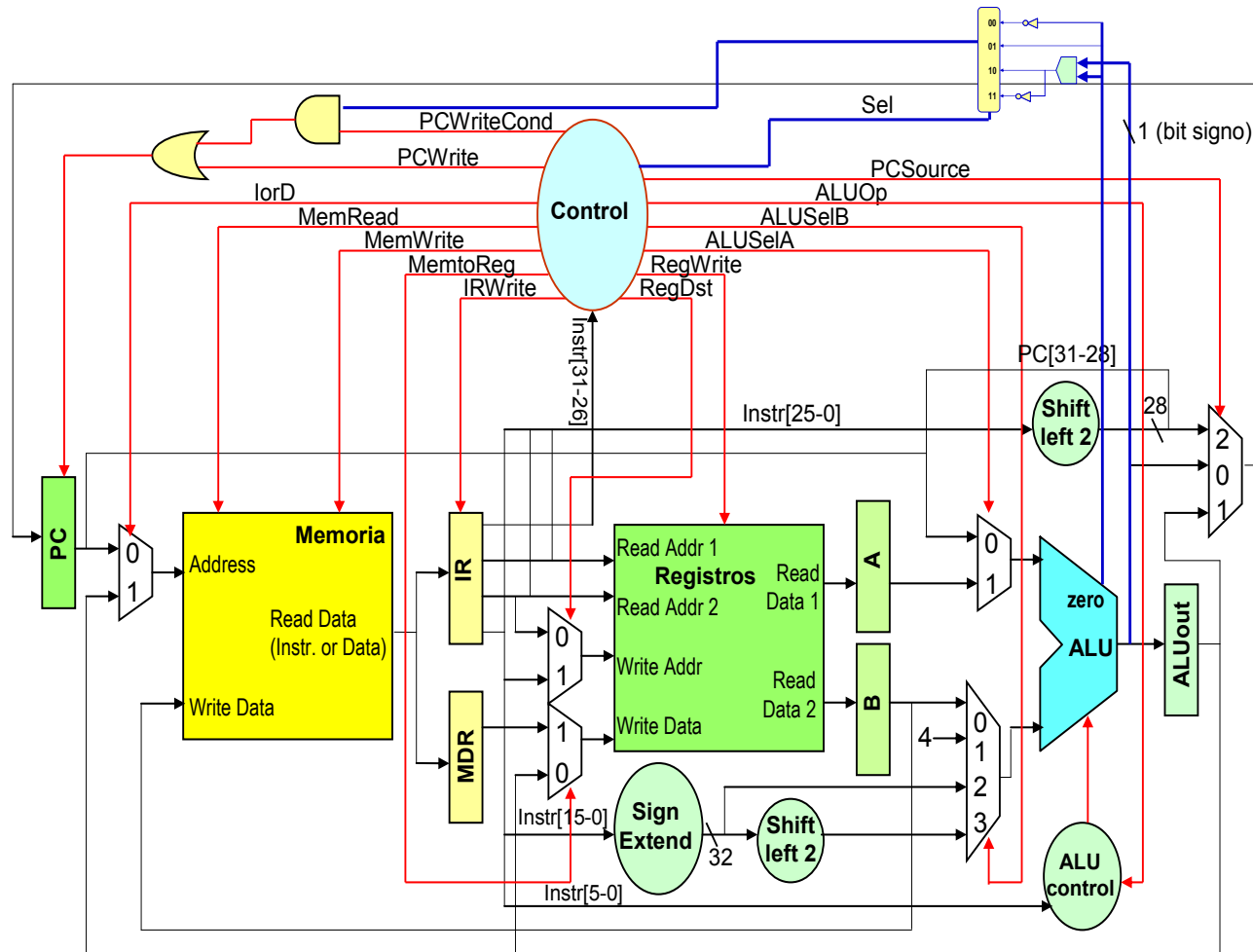


### EJERCICIO 3

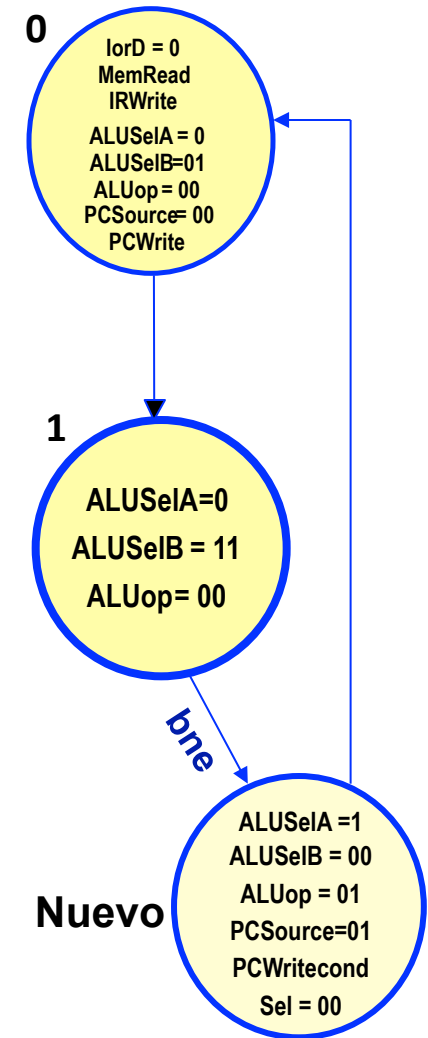
Deseamos añadir al repertorio inicial de instrucciones de la CPU, modelo multiciclo, las instrucciones **bne** (saltar si no igual), **blez** (saltar si menor o igual que 0) y **bgtz** (saltar si mayor que 0). Realizar las modificaciones necesarias en el datapath y en el autómata correspondiente a la unidad de control

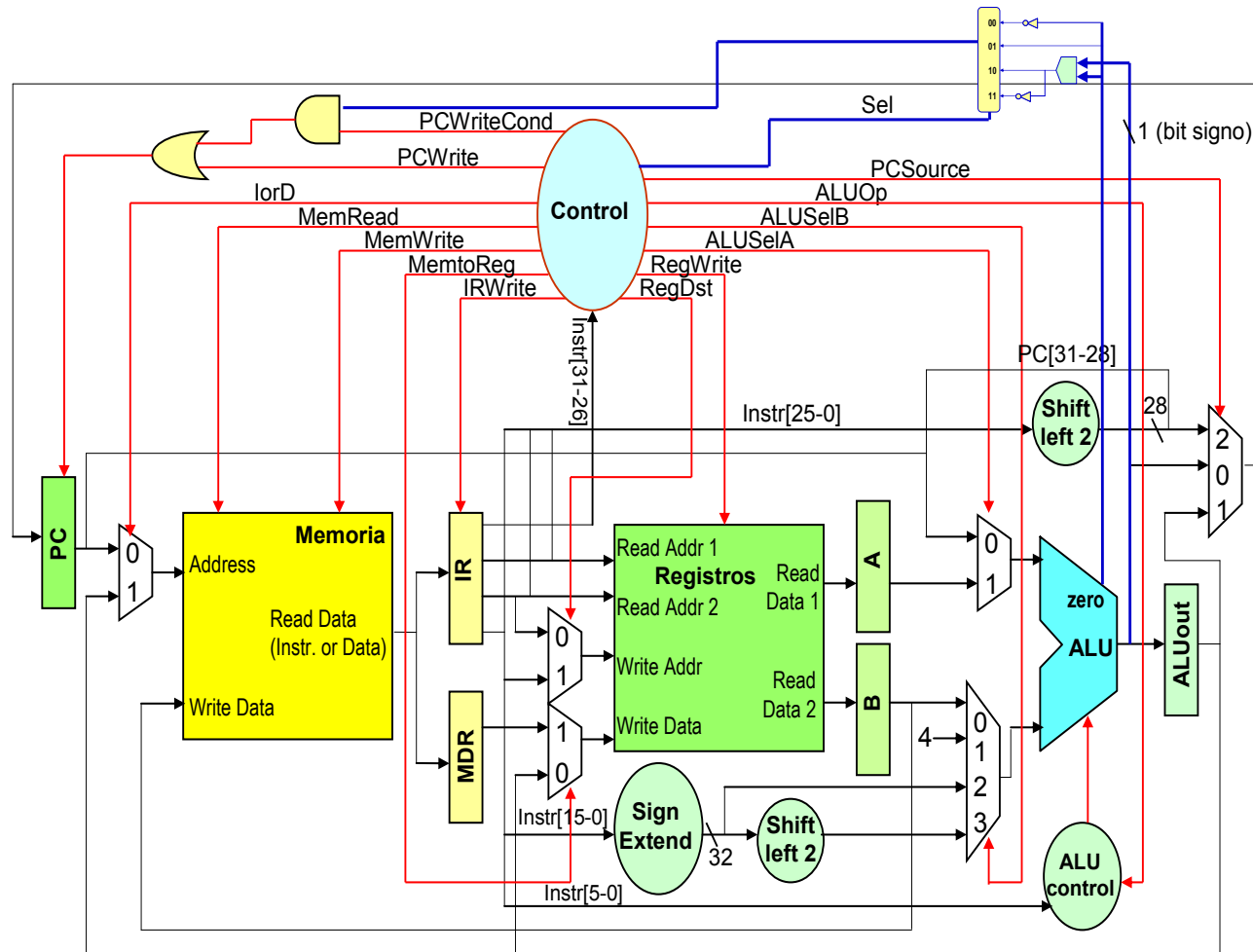




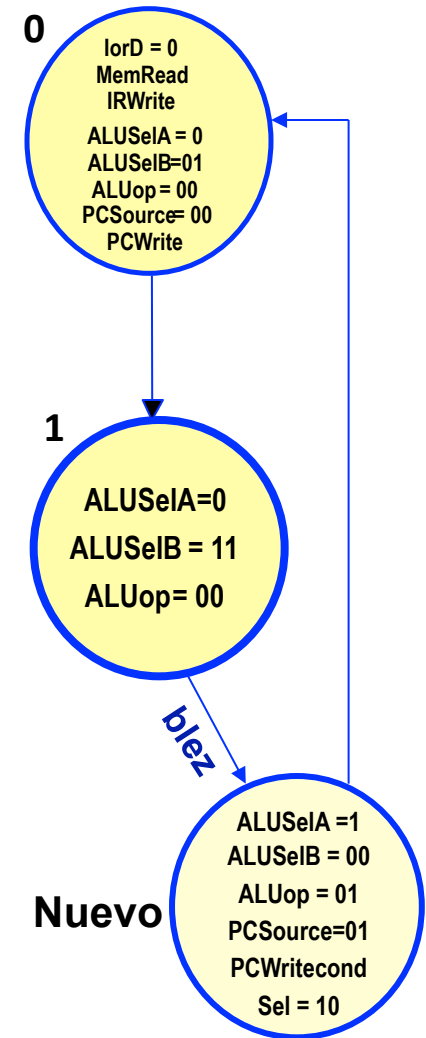


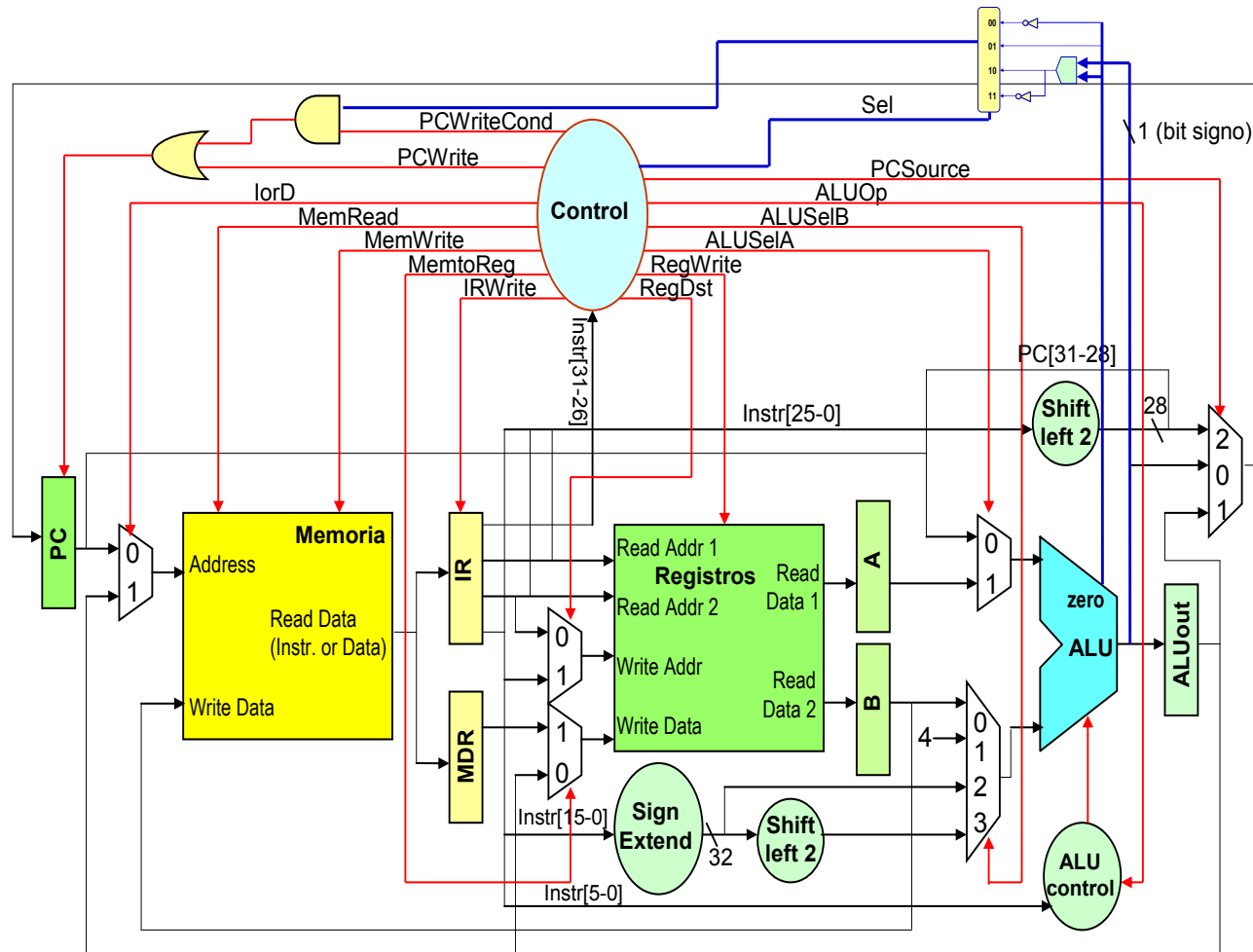
Si (A != B) PC = ALUout





Si  $(A - B \leq 0)$  PC = ALUout





Si  $(A - B > 0)$  PC = ALUout

