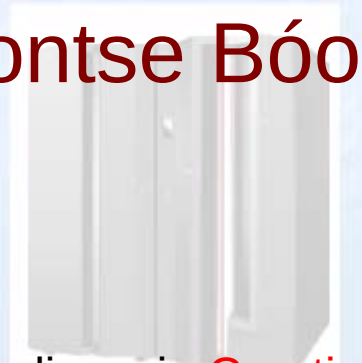


Sección de procesamiento: Sección de control





Montse Bóo Cepeda

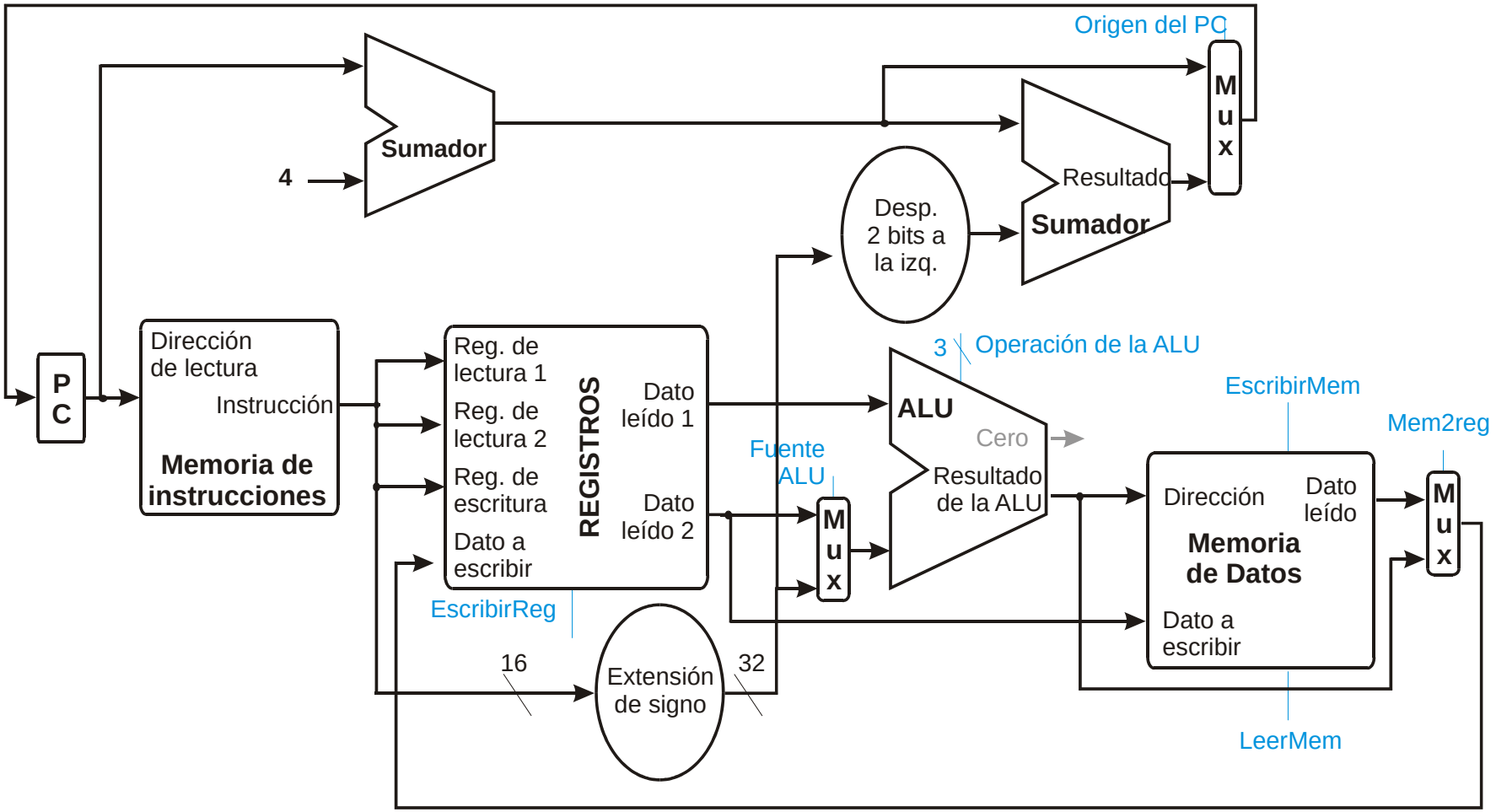
Estructura del curso

1. Evolución y caracterización de los computadores.
2. Arquitectura del MIPS: Introducción.
3. Tipo de datos.
4. El repertorio de instrucciones.
5. Aritmética del computador.
6. El camino de datos.
7. Sección de control.
8. El camino de datos multiciclo.
9. Sección de control multiciclo.
10. Entrada/Salida (I/O).

Esquema de contenidos

1. Diseño del control de la ALU
2. Diseño de la unidad de control principal
3. Extensión a otras instrucciones

Camino de datos



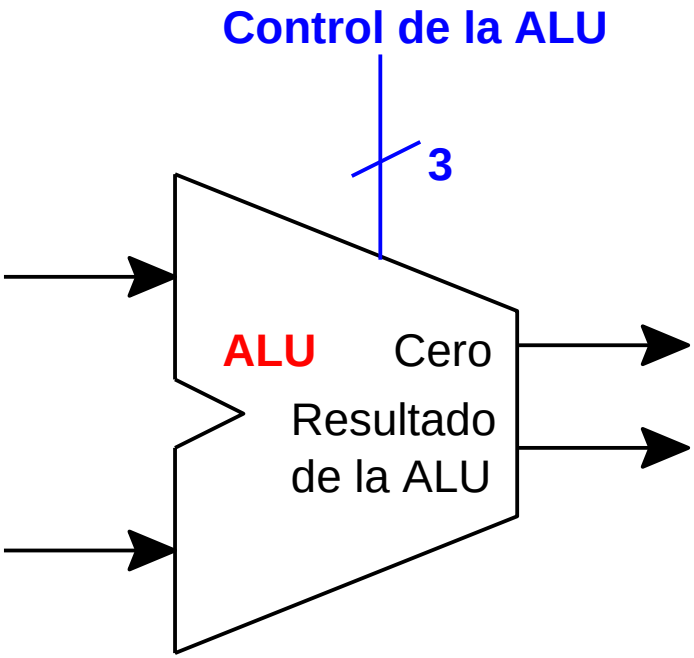
Unidad de control

A partir de las señales de entrada se deben generar:

- Control de la ALU
- Una señal de escritura para cada elemento de estado
- El control de la selección de cada multiplexor

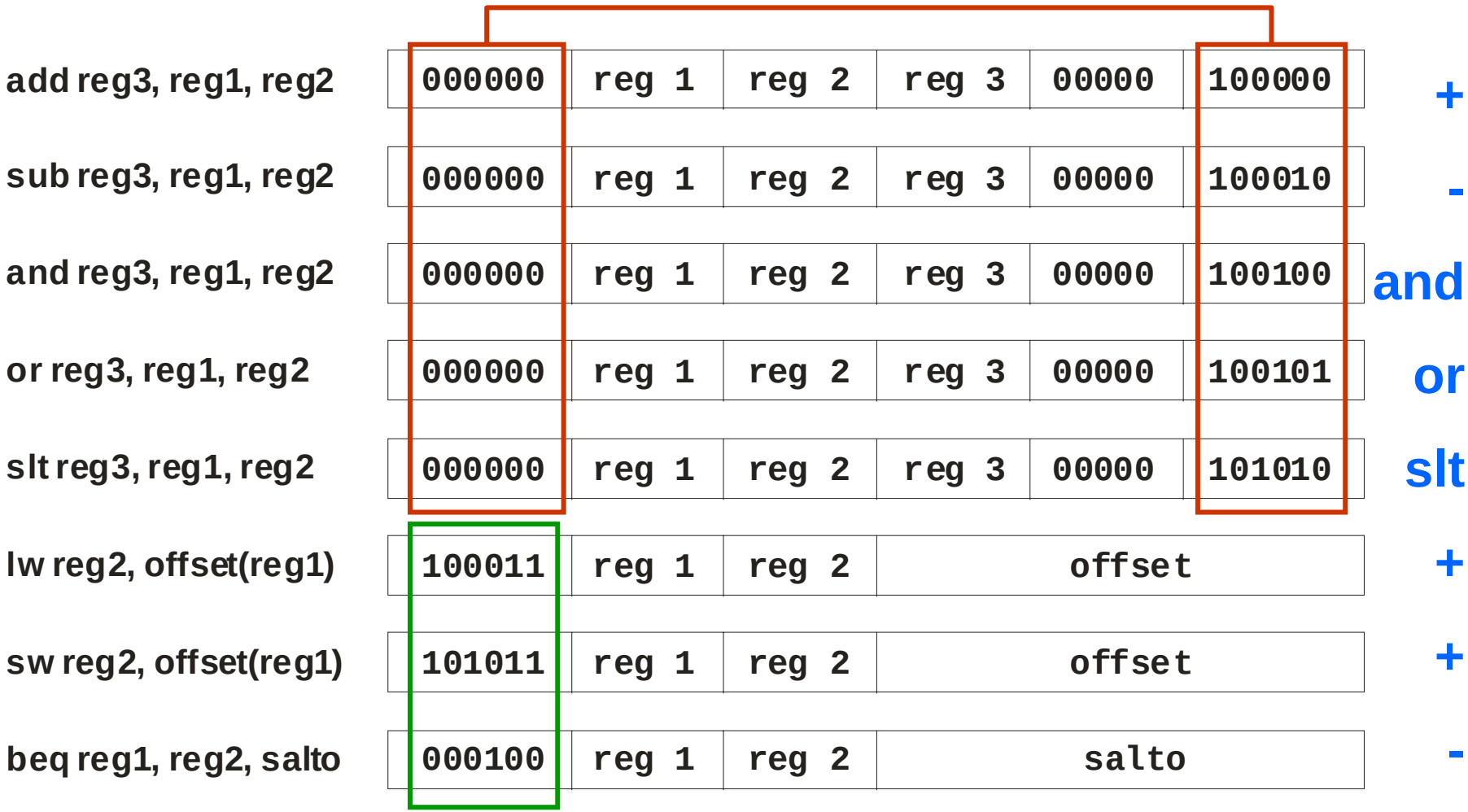
Control de la ALU

Dependiendo de la instrucción a ejecutar la ALU debe realizar cinco operaciones



Líneas de control	Función
000	AND
001	OR
010	Suma
110	Resta
111	Set on less than

Control de la ALU: Instrucciones



Control de la ALU: Instrucciones

- Código con 12 bits:
  - Código de operación: 6 bits
  - Código de función: 6 bits
- Por Karnaugh: tablas de 64 x 64 celdas
- Además hace falta una tabla por cada bit de la señal de control
  - Total: 3 funciones de 12 bits

Control de la ALU: Código de operación

Reducción número bits del código de operación (ALUOp):

Códigos de operación (*abcdef*):

- Aritméticas: 000000 => 10
- Load Word: 100011 => 00
- Store Word: 101011 => 00
- Branch equal: 000100 => 01

Control de la ALU: Código de función

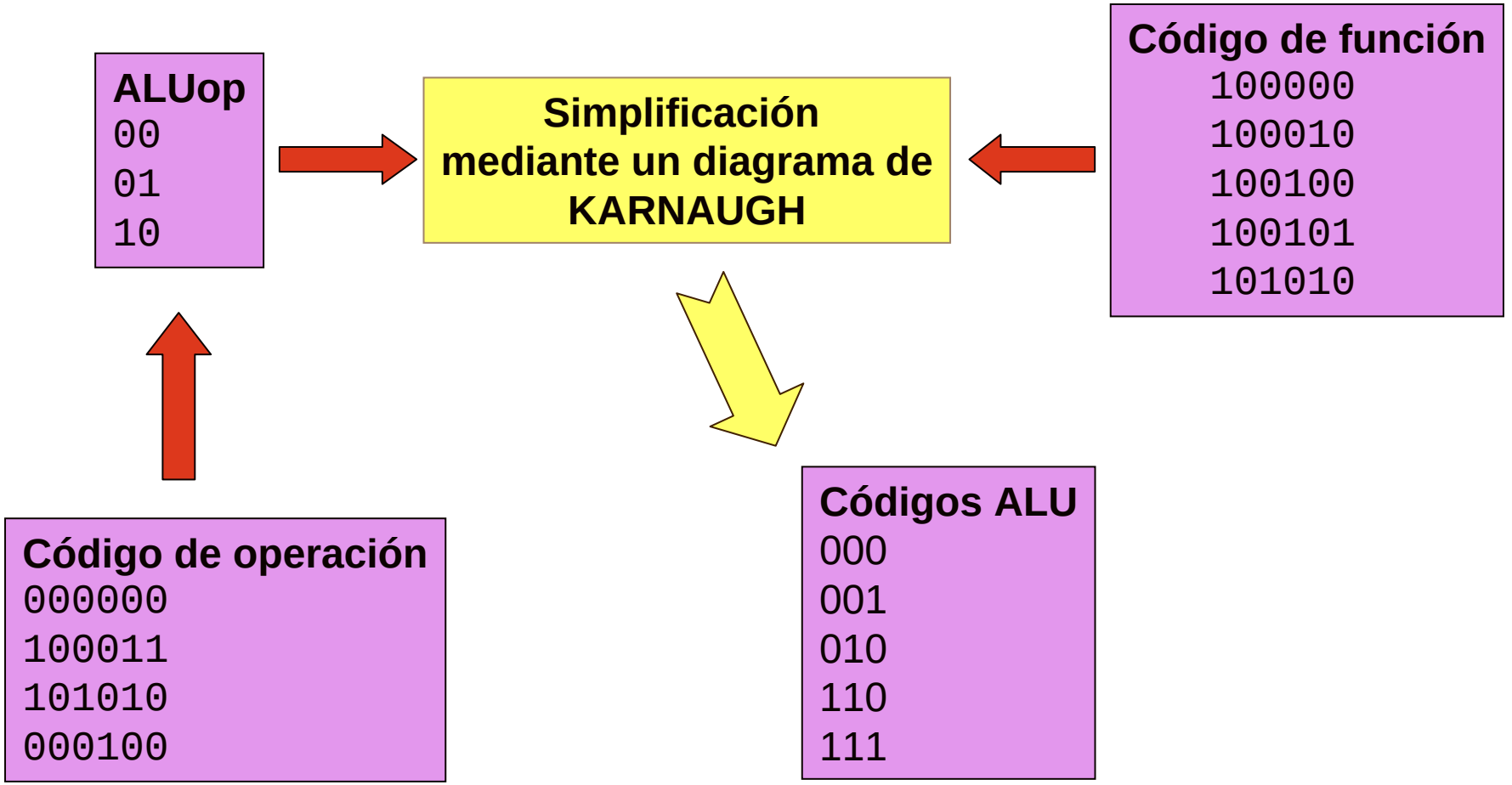
Operación	ALUOp	Campo de función	Acción de la ALU	Control ALU
Load word	00	XXXXXX	Suma	010
Store word	00	XXXXXX	Suma	010
Branch equal	01	XXXXXX	Resta	110
Suma	10	100000	Suma	010
Resta	10	100010	Resta	110
AND	10	100100	And	000
OR	10	100101	Or	001
Activar si menor que	10	101010	Activar si menor que	111

Control de la ALU: Tabla de verdad

ALUOp		Campo de la función						
ALUOp1	ALUOp2	F5	F4	F3	F2	F1	F0	Operación
0	0	X	X	X	X	X	X	010
X	1	X	X	X	X	X	X	110
1	X	X	X	0	0	0	0	010
1	X	X	X	0	0	1	0	110
1	X	X	X	0	1	0	0	000
1	X	X	X	0	1	0	1	001
1	X	X	X	1	0	1	0	111

Control de la ALU

GENERACIÓN DE LAS SEÑALES DE CONTROL



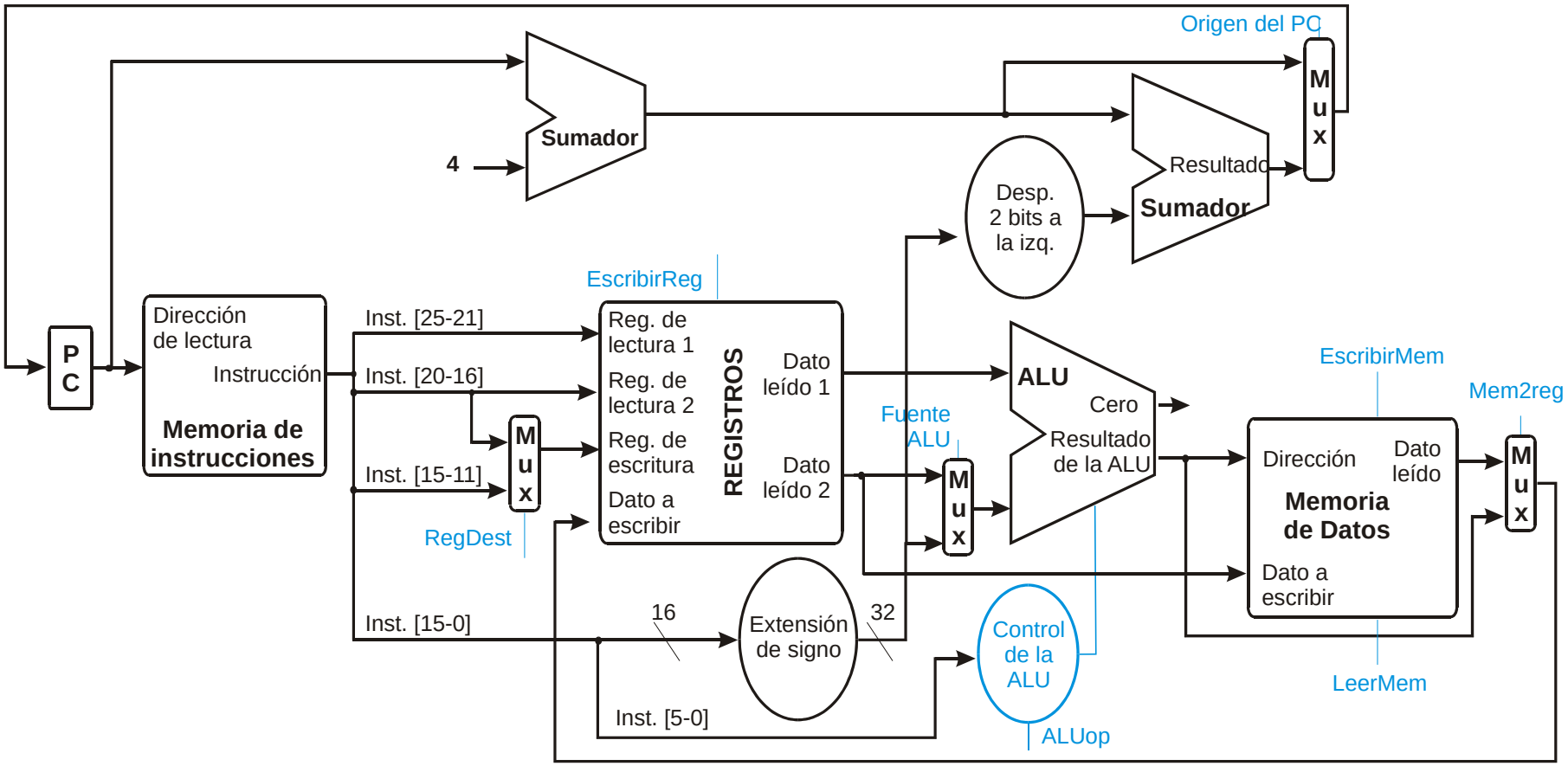
Diseño de la unidad de control principal

Se han de identificar:

- Campos de las instrucciones
- Líneas de control necesarias para la construcción del camino de datos.

add reg3, reg1, reg2	000000	reg 1	reg 2	reg 3	00000	100000
sub reg3, reg1, reg2	000000	reg 1	reg 2	reg 3	00000	100010
and reg3, reg1, reg2	000000	reg 1	reg 2	reg 3	00000	100100
or reg3, reg1, reg2	000000	reg 1	reg 2	reg 3	00000	100101
slt reg3, reg1, reg2	000000	reg 1	reg 2	reg 3	00000	101010
lw reg2, offset(reg1)	100011	reg 1	reg 2	offset		
sw reg2, offset(reg1)	101011	reg 1	reg 2	offset		
beq reg1, reg2, salto	000100	reg 1	reg 2	salto		

Camino de datos: Con todos los multiplexores necesarios

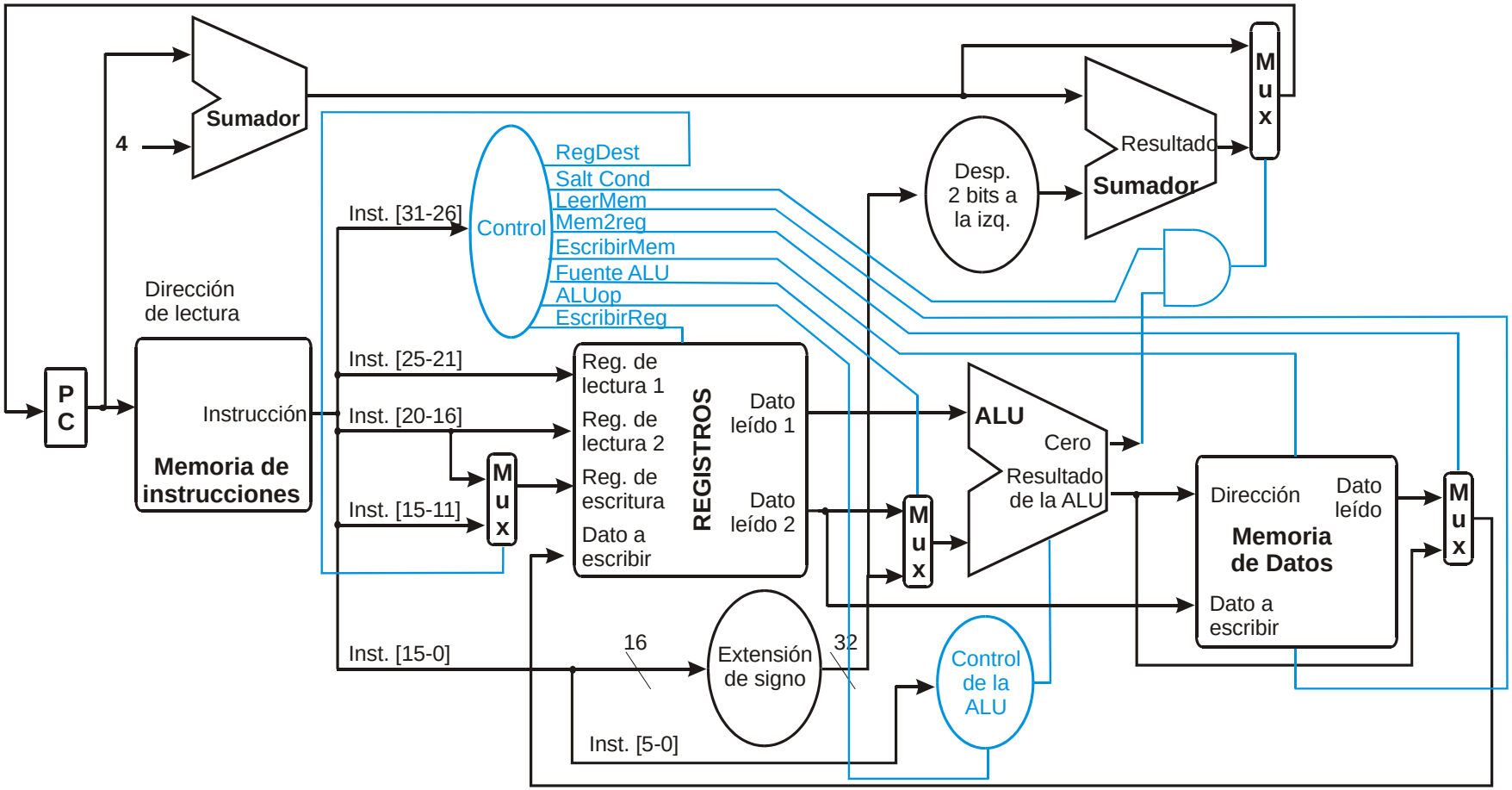


Efecto de las señales de control

Señal de control	**Efecto cuando está activa (1)**		**Efecto cuando no está activa (0)**	
	operación	**selecciona**	**operación**	**selecciona**
RegDest	aritméticas	bits 15-11	lw	bits 20-16
SaltoCond	beq	ALU_cero	otras	0
LeerMem	lw	leer memoria	otras	no leer
EscrMem	sw	escribir memoria	otras	no escribir
Mem2Reg	lw	desde memoria	aritméticas	desde ALU
FuenteALU	lw, sw	valor inmediato	otras	registro
EscrReg	lw y aritméticas	escribir registro	otras	no escribir
ALUop (bit 1)	aritméticas		otras	
ALUop (bit 0)	beq		otras	

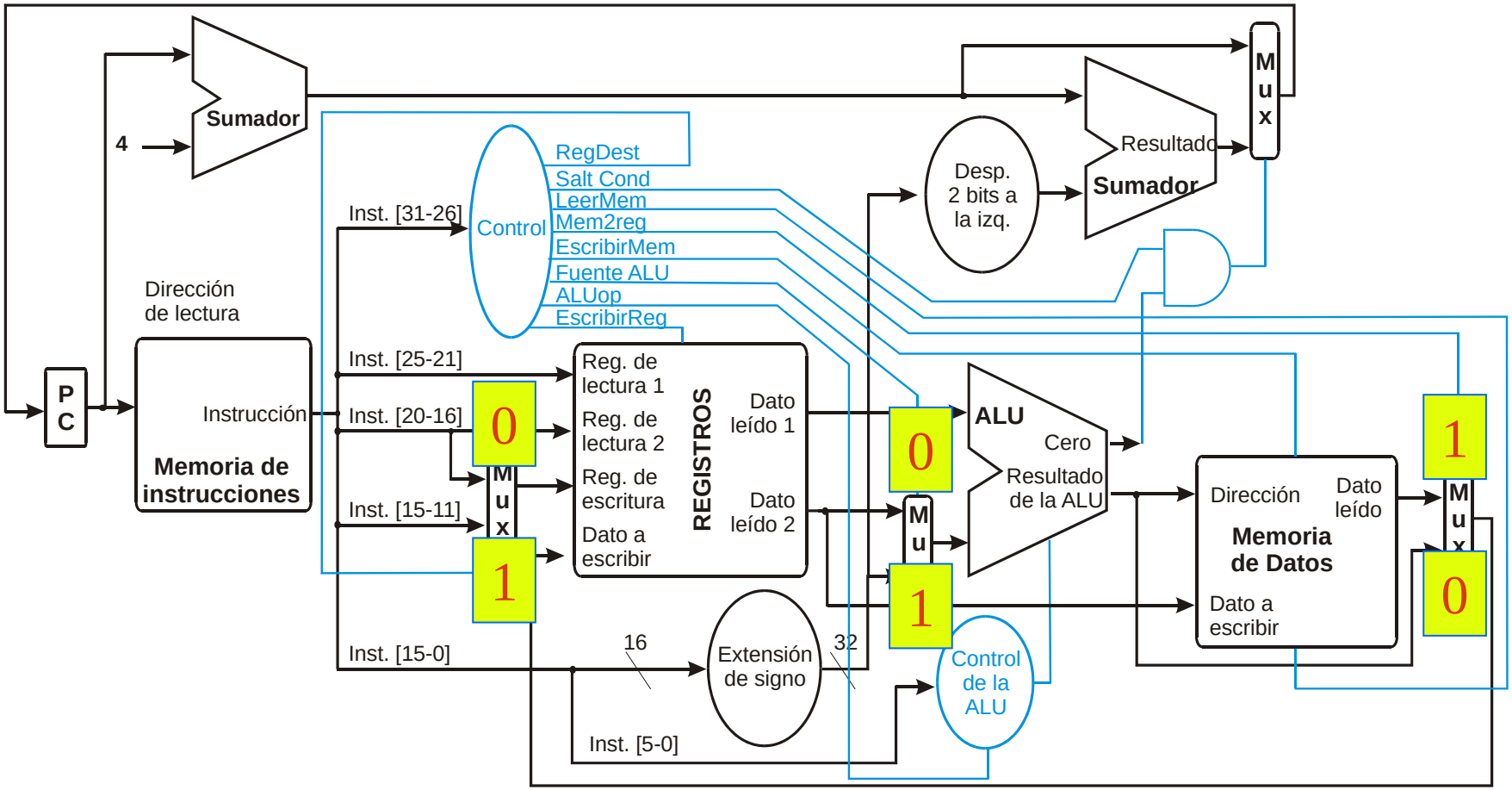
Unidad de control principal

Efecto de cada señal de control



Unidad de control principal

Efecto de cada señal de control



Activación líneas de control

La activación de las líneas está determinada por el código de operación de las instrucciones:

Instrucción	RegDest	Fuente ALU	Mem2 Reg	Reg Write	Leer Mem	Escr Mem	Salto Cond	ALUOp1	ALUOp0
Formato R	1	0	0	1	0	0	0	1	0
lw	0	1	1	1	1	0	0	0	0
sw	X	1	X	0	0	1	0	0	0
beq	X	0	X	0	0	0	1	0	1

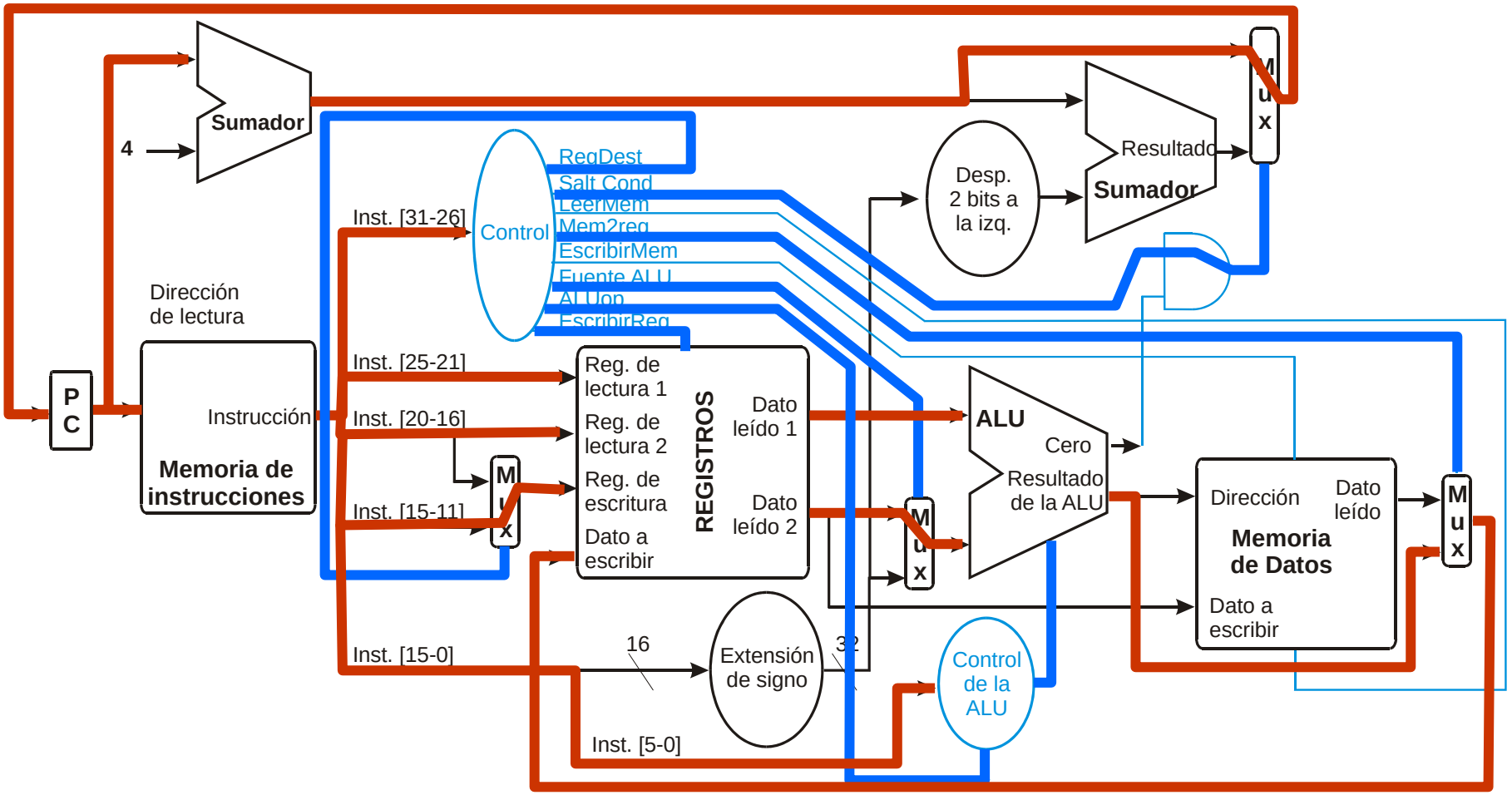
Ejemplo: Instrucción tipo R

add $t1, $t2, $t3

Hay cuatro etapas en la ejecución de una instrucción tipo R:

1. Se carga la instrucción de la memoria de instrucciones y se incrementa el PC
2. Se leen los registros $t2 y $t3 del banco de registros. Adicionalmente la unidad de control principal se encarga de calcular la activación de las señales de control.
3. La ALU se encarga de realizar la operación adecuada con los datos procedentes del banco de registros, utilizando para ello el campo de función (bits 5-0).
4. El resultado calculado en la ALU se escribe en el banco de registros utilizando los bits 15-11 de la instrucción para seleccionar el registro destino ($t1).

Instrucciones aritmético-lógicas



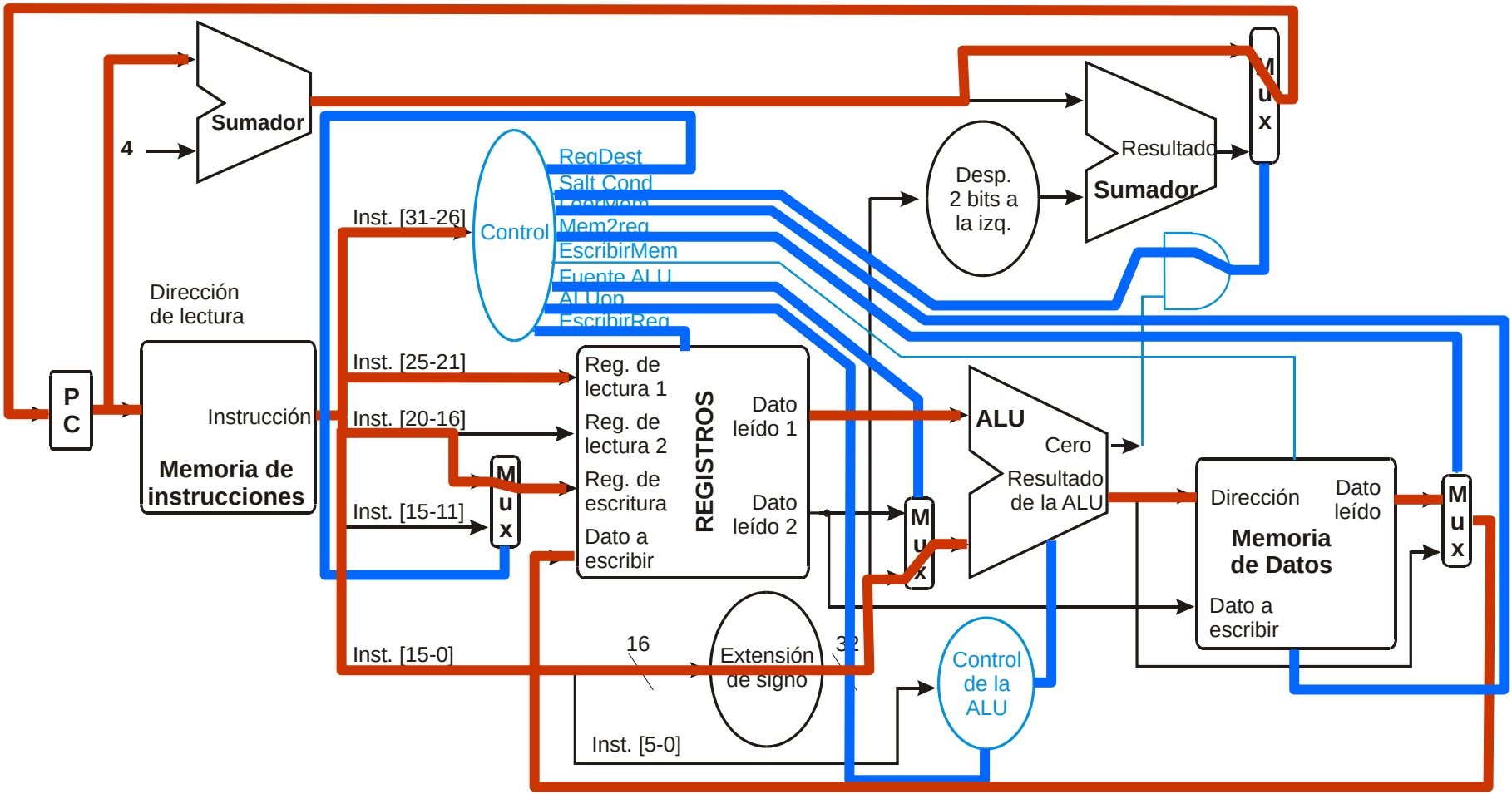
Ejemplo: Instrucción de carga

lw $t1, despl ($t2)

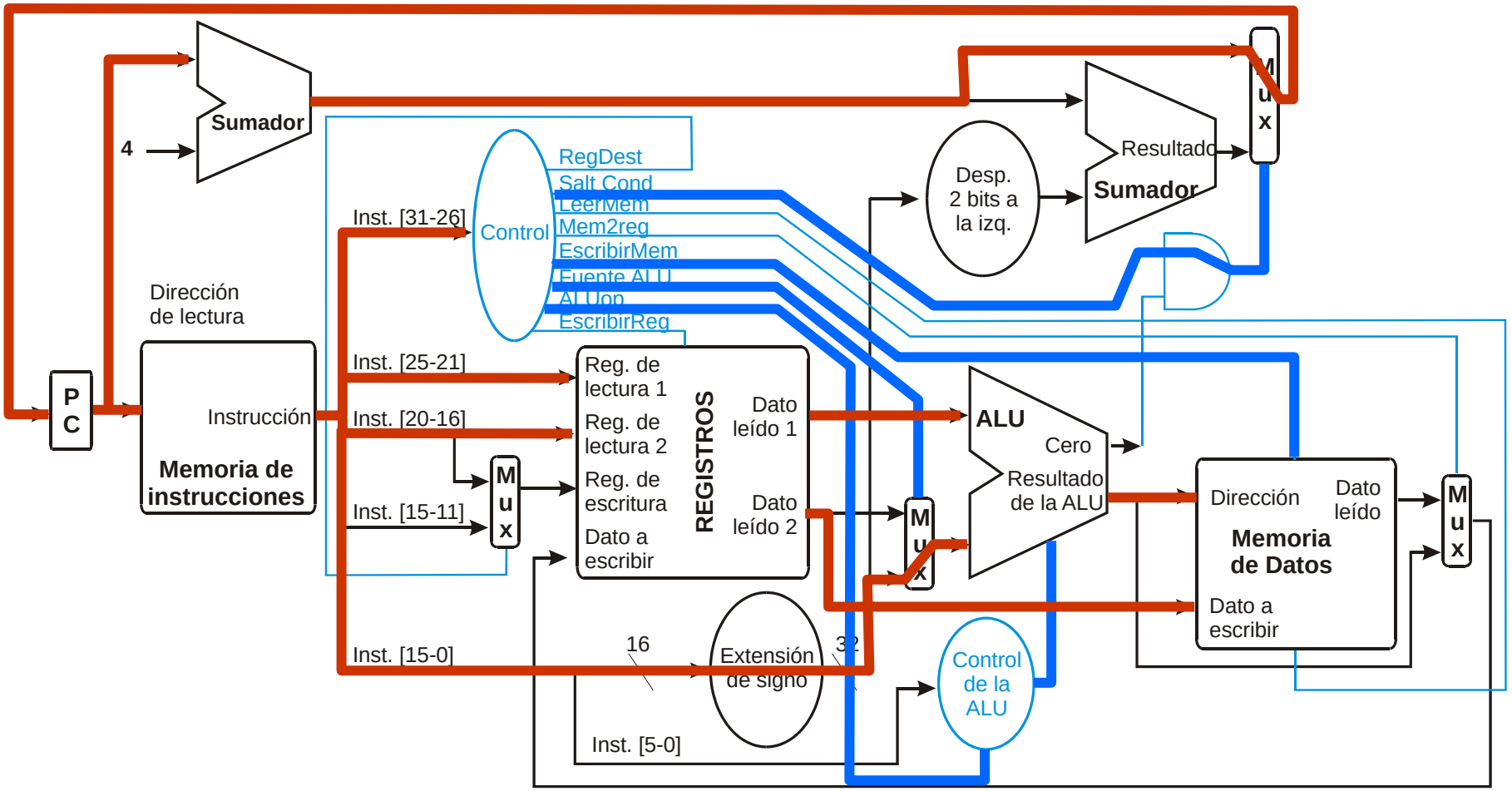
Hay cinco etapas en la ejecución de una instrucción de carga:

1. Se carga la instrucción de la memoria de instrucciones y se incrementa el PC
2. Se lee el valor del registro $t2 del banco de registros
3. La ALU calcula la suma del valor leído del banco de registros y los 16 bits de menor peso de la instrucción, con el signo extendido.
4. Dicha suma se utiliza como dirección para acceder a la memoria de datos
5. El dato procedente de la memoria se escribe en el banco de registros. El registro destino viene determinado por los bits 20-16 de la instrucción ($t1).

Instrucción lw



Instrucción sw



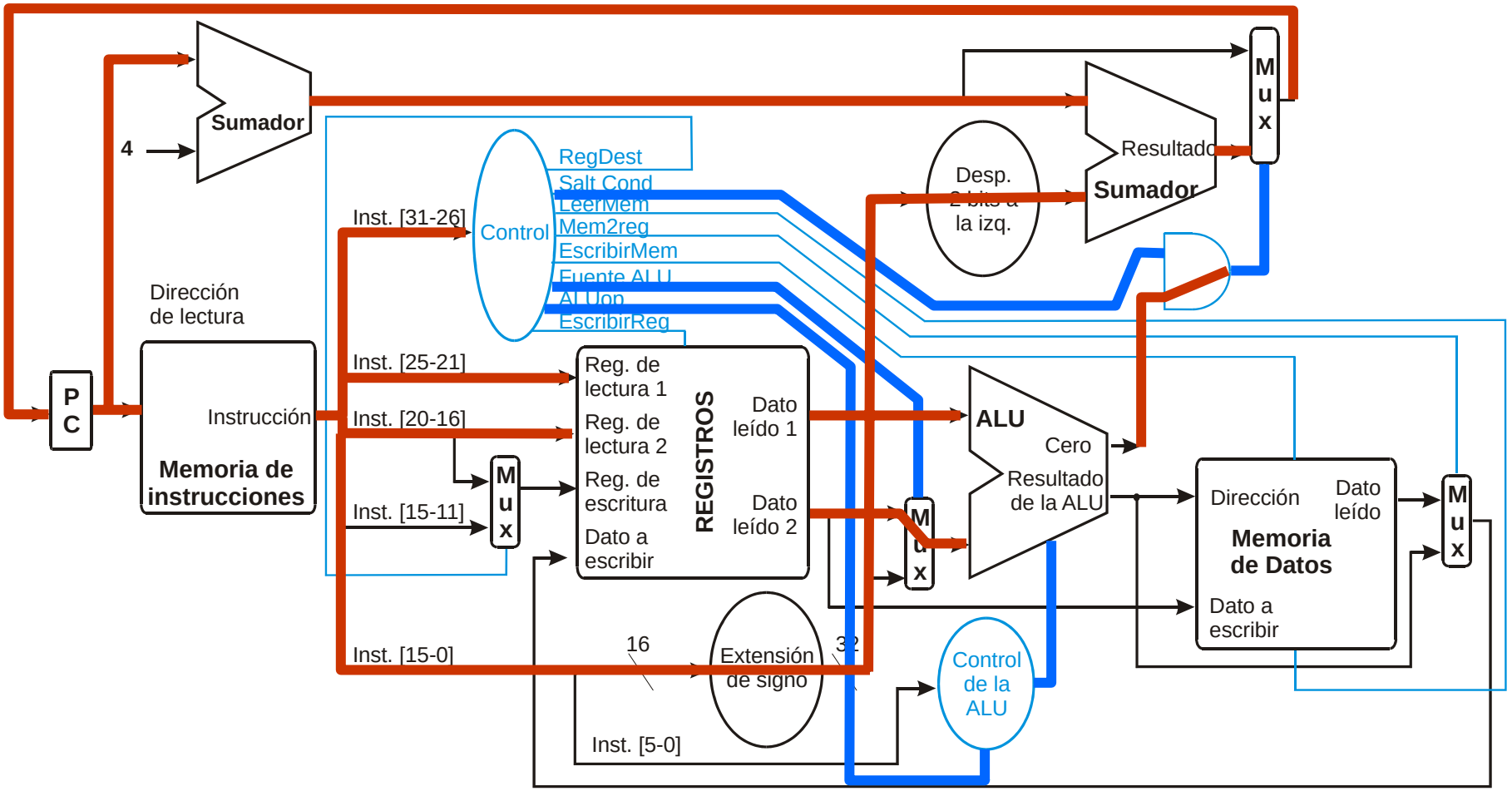
Ejemplo: Instrucción de salta si igual

beq $t1, $t2, desp

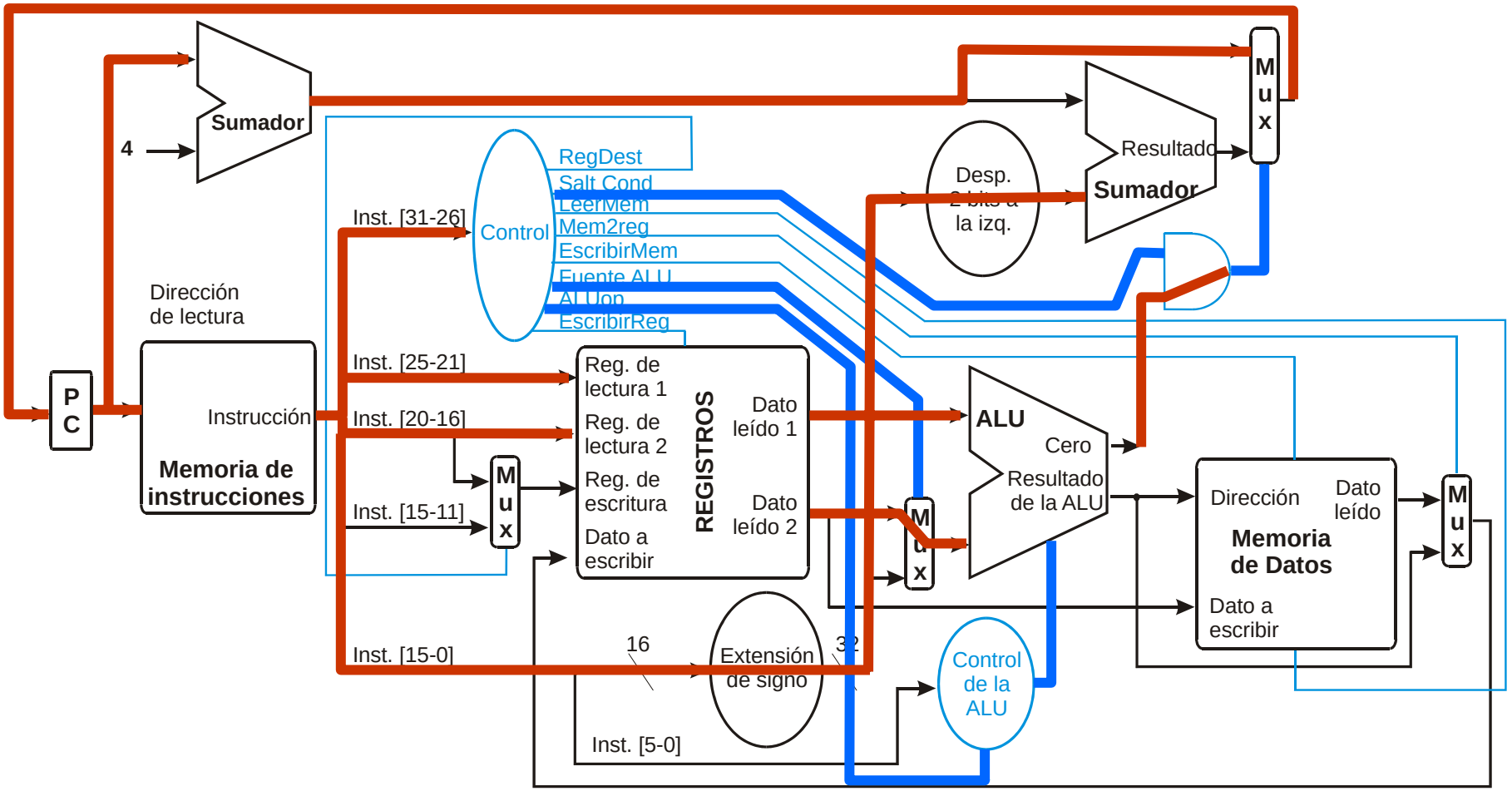
Hay cuatro etapas:

1. Se carga la instrucción de la memoria de instrucciones y se incrementa el PC
2. Se leen los registros $t1 y $t2 del banco de registros.
3. La ALU realiza una resta de los operandos leídos del banco de registros. Se suma el valor de PC+4 a los 16 bits de menor peso (con el signo extendido) de la instrucción desplazados 2 bits. El resultado es la dirección destino del salto.
4. Se utiliza la señal de CERO de la ALU para decidir qué valor se almacena en el PC.

Instrucción beq (con éxito)



Instrucción beq (sin éxito)



Añadir instrucciones

Para añadir instrucciones puede ser necesario:

- Añadir hardware
- Modificar el control
- Añadir multiplexores para compartir
  - Hardware
  - Datos

Añadir instrucciones: salto incondicional