

Diseño del procesador MIPS R2000

Aula Virtual → IS09

Sergio Barrachina Mir

Área de Arquitectura y Tecnología de Computadores

Dpt. de Ingeniería y Ciencia de los Computadores

Universidad Jaume I

Índice

1. Objetivos	4
2. Contenido y bibliografía	5
3. Introducción	6
3.1. Metodología de sincronización	11
4. Construcción de la ruta de datos	14
4.1. Búsqueda de instrucciones e incremento del PC	15
4.2. Instrucciones aritmético-lógicas (tipo R)	17
4.3. Instrucciones de carga-almacenamiento	19
4.4. Instrucción de salto condicional	21
5. Rutas monociclo y multiciclo	23

6. Ruta de datos monociclo	24
7. Unidad de control monociclo	34
7.1. El control de la ALU	35
7.2. Formatos de instrucción	40
7.3. Ruta de datos con señales de control	44
7.4. Ruta de datos monociclo con control	46
7.5. Añadir una nueva instrucción	56
7.6. Desventajas de la realización monociclo	59

1 Objetivos

- Dada una instrucción, conocer qué elementos y conexiones son necesarias para su ejecución.
- Poder construir la ruta de datos completa, tanto monociclo como multiciclo.
- Ser capaz de modificar la ruta de datos de un procesador monociclo o multiciclo para ampliar su juego de instrucciones.
- Conocer las ventajas y desventajas, en implementación y rendimiento, de un procesador multiciclo frente a otro monociclo.

2 Contenido y bibliografía

➤ Contenido:

- Introducción al diseño de un procesador.
- Construcción de la ruta de datos monociclo.
- Diseño de la unidad de control monociclo.
- Construcción de la ruta de datos multiciclo.
- Diseño de la unidad de control multiciclo.

➤ Bibliografía:

- *Estructura y Diseño de Computadores: interficie circuitería/programación* (2000). David A. Patterson y John L. Hennessy. Capítulo 5.

3 Introducción

► En este capítulo se va a construir:

⇒ la ruta de datos y

⇒ la unidad de control

para dos realizaciones del siguiente subconjunto del repertorio de instrucciones del MIPS:

⇒ Acceso a memoria: «**lw**» y «**sw**».

⇒ Aritmético-lógicas: «**add**», «**sub**», «**and**», «**or**» y «**slt**».

⇒ Salto: «**beq**» y, más adelante, «**j**».

3 Introducción (II)

- Gran parte de lo necesario para construir las realizaciones de las anteriores instrucciones es común a todas ellas.
- Para cada instrucción, los dos primeros pasos son idénticos:
 1. Enviar el contenido del PC a la memoria, cargar la instrucción de memoria e incrementar el PC.

3 Introducción (II)

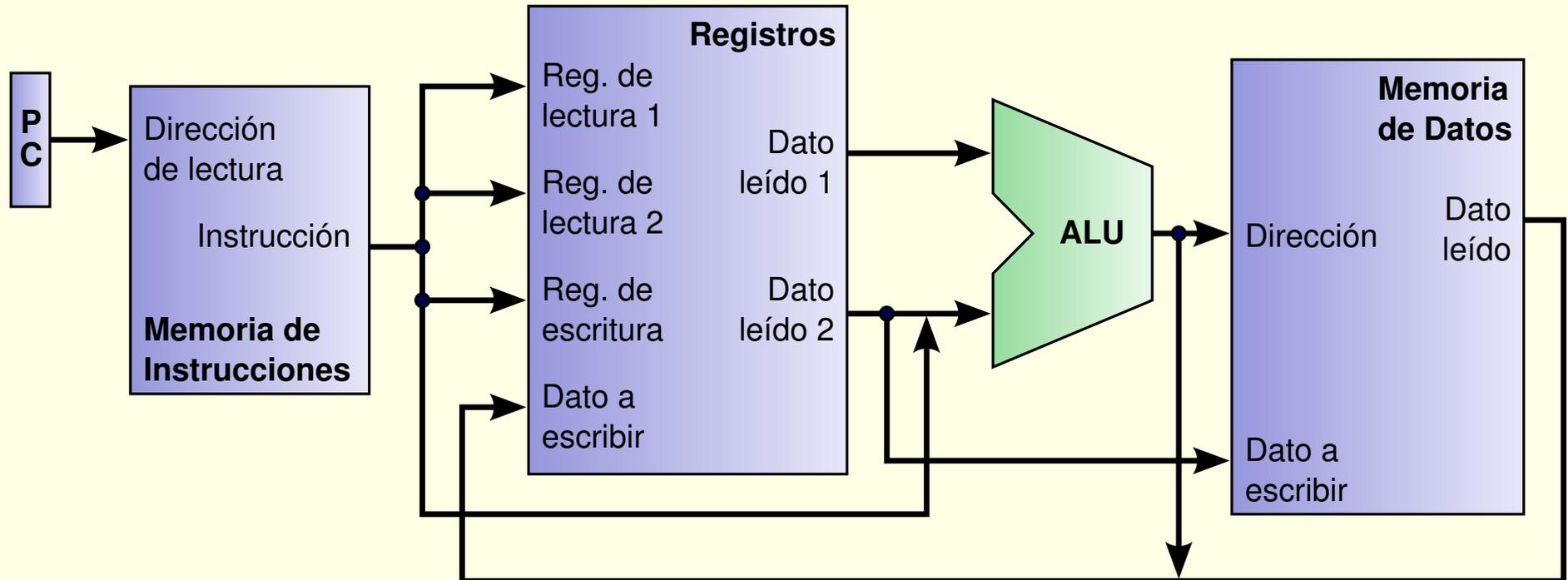
- Gran parte de lo necesario para construir las realizaciones de las anteriores instrucciones es común a todas ellas.
- Para cada instrucción, los dos primeros pasos son idénticos:
 1. Enviar el contenido del PC a la memoria, cargar la instrucción de memoria e incrementar el PC.
 2. Leer uno o dos registros, dependiendo de la instrucción.

3 Introducción (II)

- Gran parte de lo necesario para construir las realizaciones de las anteriores instrucciones es común a todas ellas.
- Para cada instrucción, los dos primeros pasos son idénticos:
 1. Enviar el contenido del PC a la memoria, cargar la instrucción de memoria e incrementar el PC.
 2. Leer uno o dos registros, dependiendo de la instrucción.
- Tras estos dos pasos, depende de la instrucción.
- Aún así, para las instrucciones del mismo tipo, las acciones generalmente son las mismas.

3 Introducción (III)

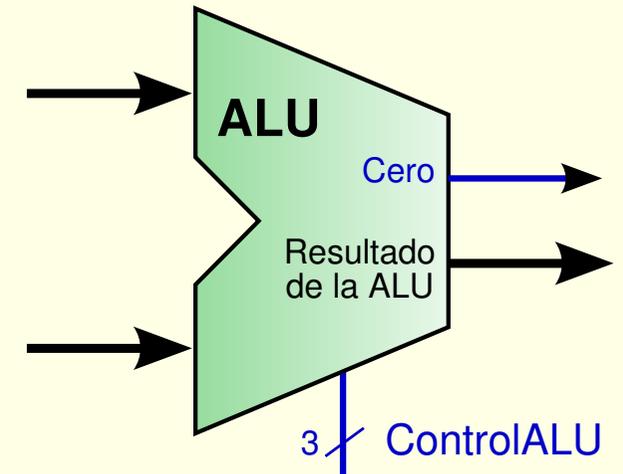
- Esquema de alto nivel de una realización del MIPS:



- Este esquema se irá detallando a lo largo del tema.

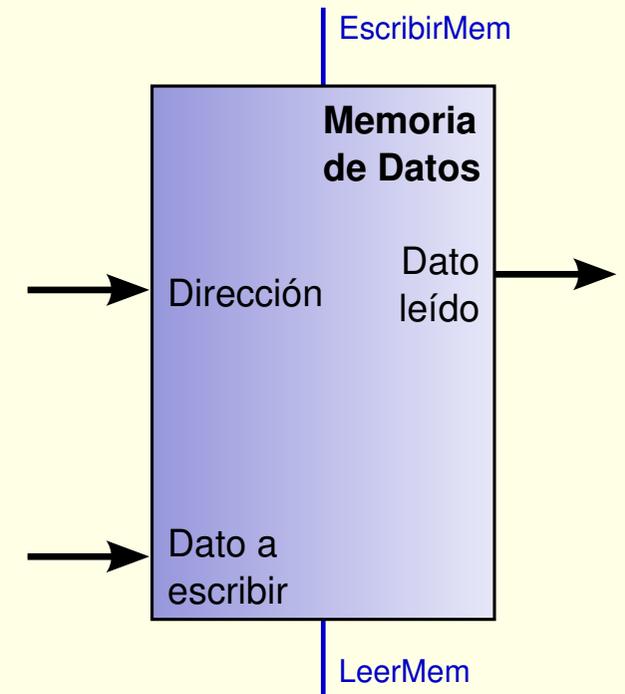
3 Introducción (IV)

- ▶ Las unidades funcionales que se utilizan son de dos tipos:
 - ⇒ Combinacionales.
 - ⇒ Secuenciales.
- ▶ En las unidades combinacionales:
 - ⇒ La salida depende exclusivamente del valor de las entradas.



3 Introducción (V)

- ▶ Las unidades secuenciales:
 - ⇒ Son aquellas cuya salida depende no sólo de las entradas, sino también del estado que almacenan.
 - ⇒ Son las que definen el estado del computador: si se guarda su contenido, se apaga, se enciende y se restaura → como si no se hubiera apagado.
 - ⇒ Las unidades combinatoriales deben partir o llegar a una unidad secuencial.

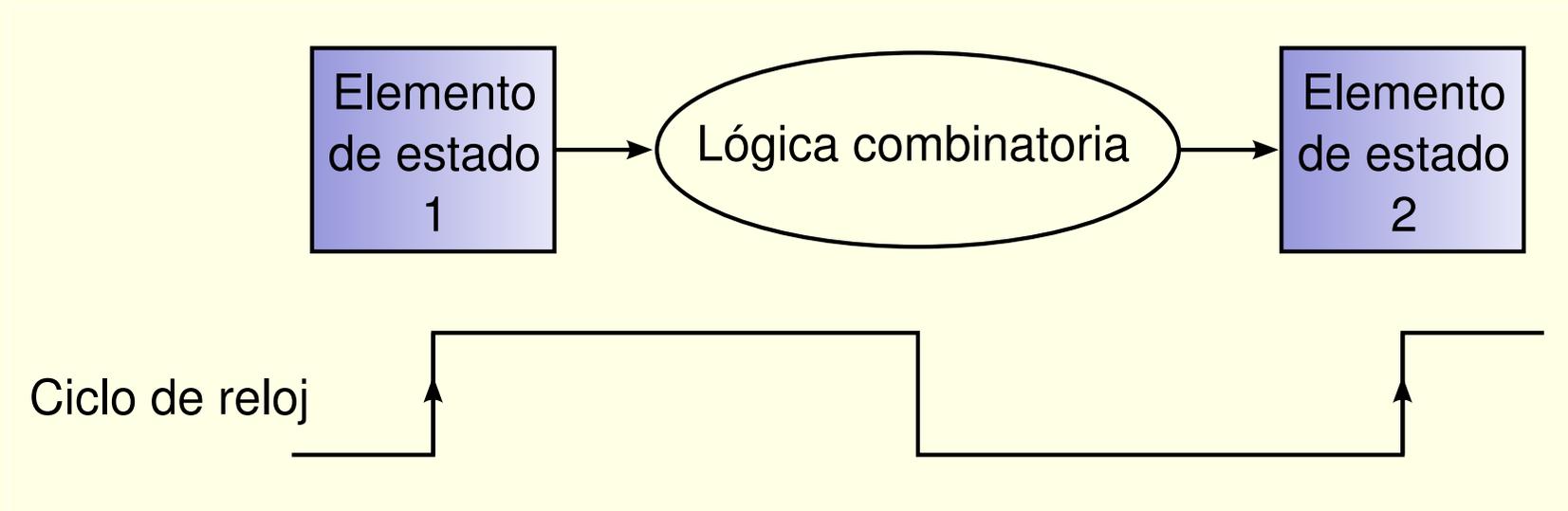


3.1 Metodología de sincronización

- Las unidades secuenciales síncronas poseen una entrada adicional: el reloj.
- Dicha entrada controla cuándo se debe escribir en los elementos de estado.
- La escritura puede provocarse por:
 - ⇒ Nivel: nivel alto o bajo.
 - ⇒ Flanco: flanco ascendente o descendente.
- La metodología generalmente empleada es la de flanco.
- En la realización propuesta: **flanco ascendente**.

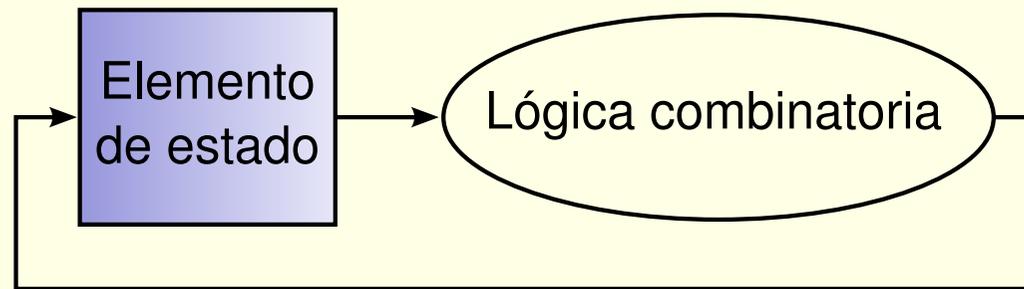
3.1 Metodología de sincronización (II)

- Escritura controlada por flanco ascendente:



3.1 Metodología de sincronización (II)

- La activación por flanco permite leer y escribir de un elemento de estado en el mismo ciclo de reloj (sin que se produzca una condición de carrera):



4 Construcción de la ruta de datos

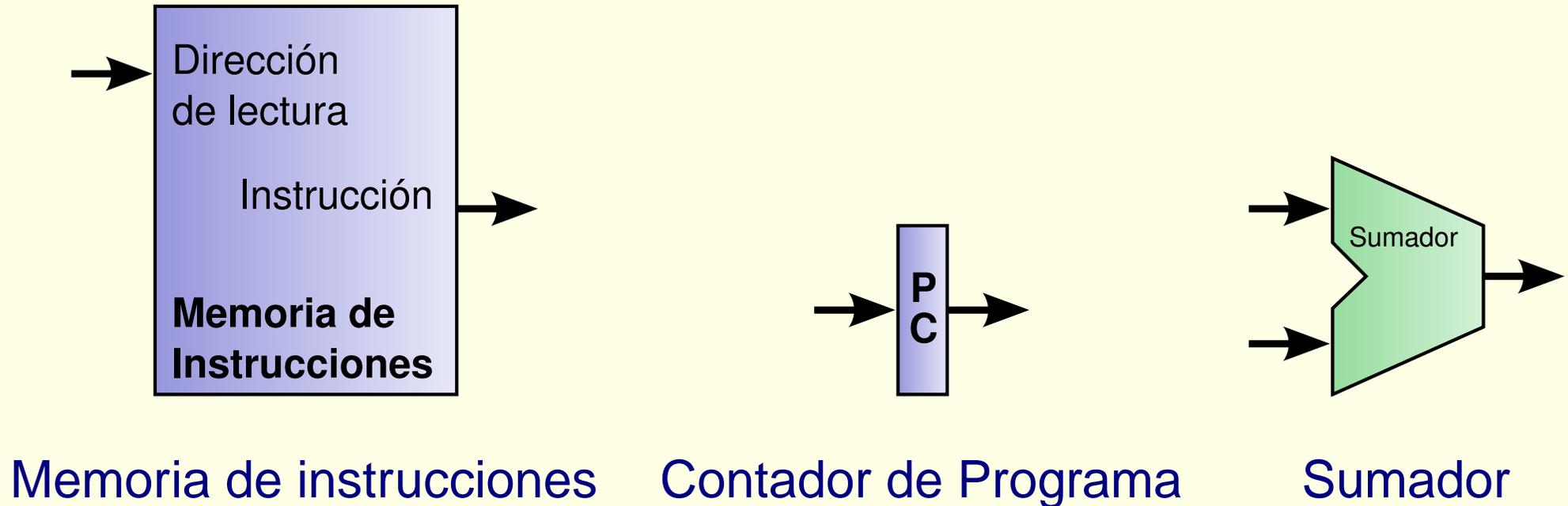
- Forma de construir la ruta de datos:
 - Examinar los elementos requeridos para la ejecución de las distintas instrucciones.
 - Construir las diferentes partes de la ruta de datos.
 - Determinar las señales de control necesarias.
 - Finalmente, unir las distintas partes para generar la ruta de datos:
 - ➡ Las señales de control seleccionaran los diferentes caminos de la ruta de datos.

4.1 Búsqueda de instrucciones e incremento del PC

➤ Elementos necesarios:

4.1 Búsqueda de instrucciones e incremento del PC

➤ Elementos necesarios:



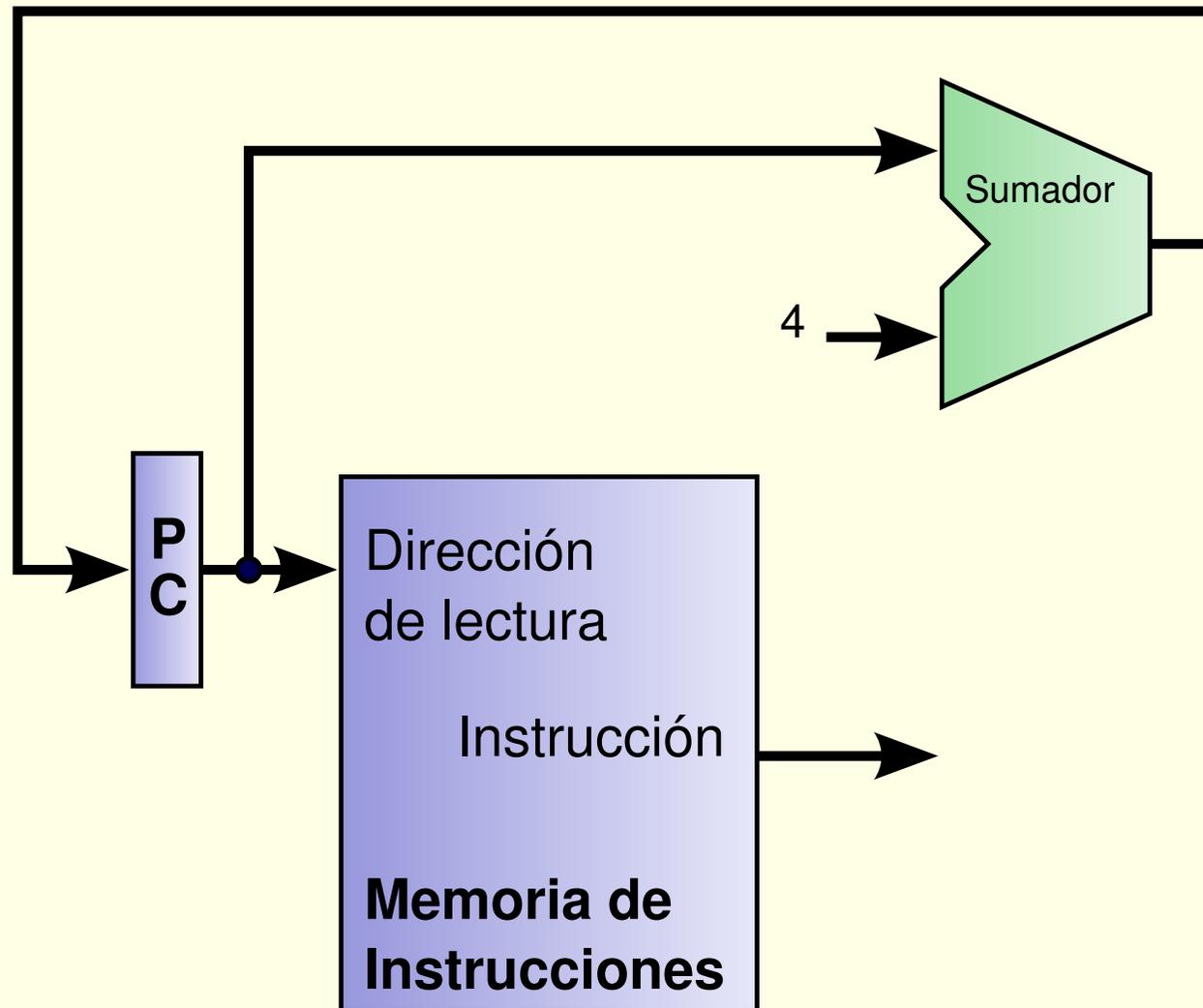
Memoria de instrucciones

Contador de Programa

Sumador

4.1 Búsqueda de instrucciones e incremento del PC (II)

► Parte de la ruta de datos:

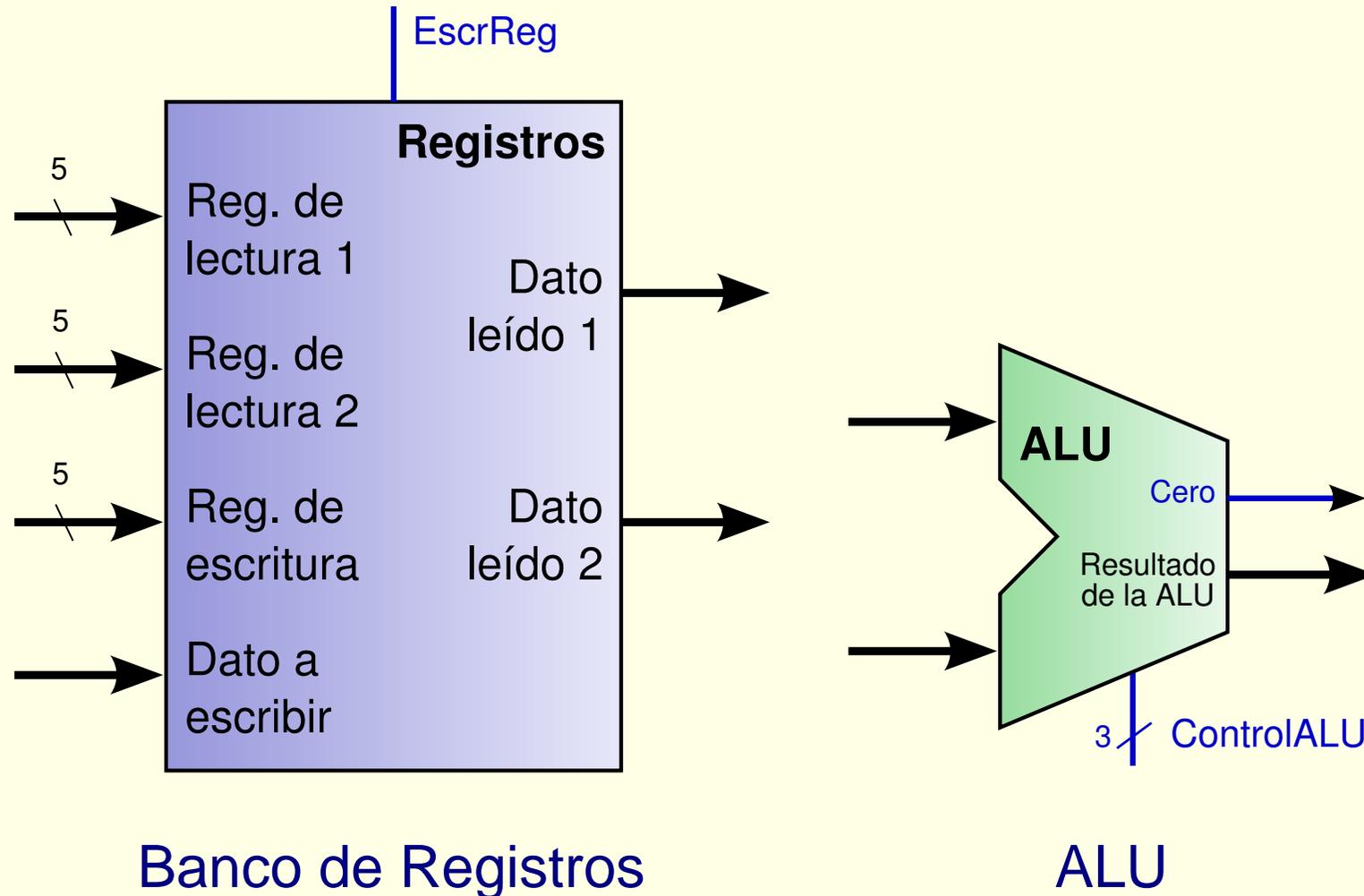


4.2 Instrucciones aritmético-lógicas (tipo R)

- ▶ Elementos necesarios para instrucciones tipo «**add** \$t1,\$t2,\$t3»

4.2 Instrucciones aritmético-lógicas (tipo R)

- Elementos necesarios para instrucciones tipo «**add** \$t1,\$t2,\$t3»

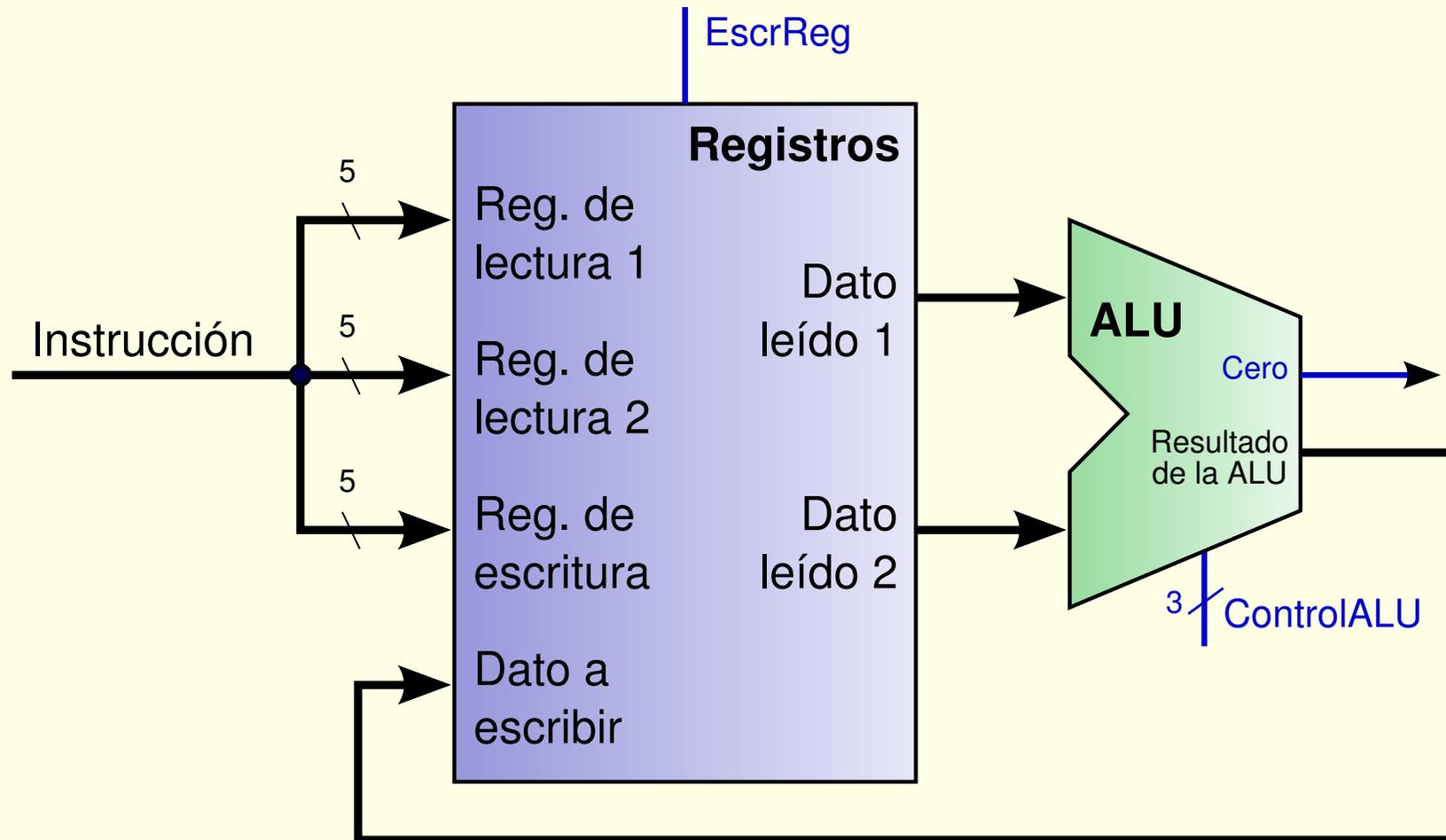


Banco de Registros

ALU

4.2 Instrucciones aritmético-lógicas (tipo R) (II)

- ▶ Parte de la ruta de datos para las instrucciones aritmético-lógicas:

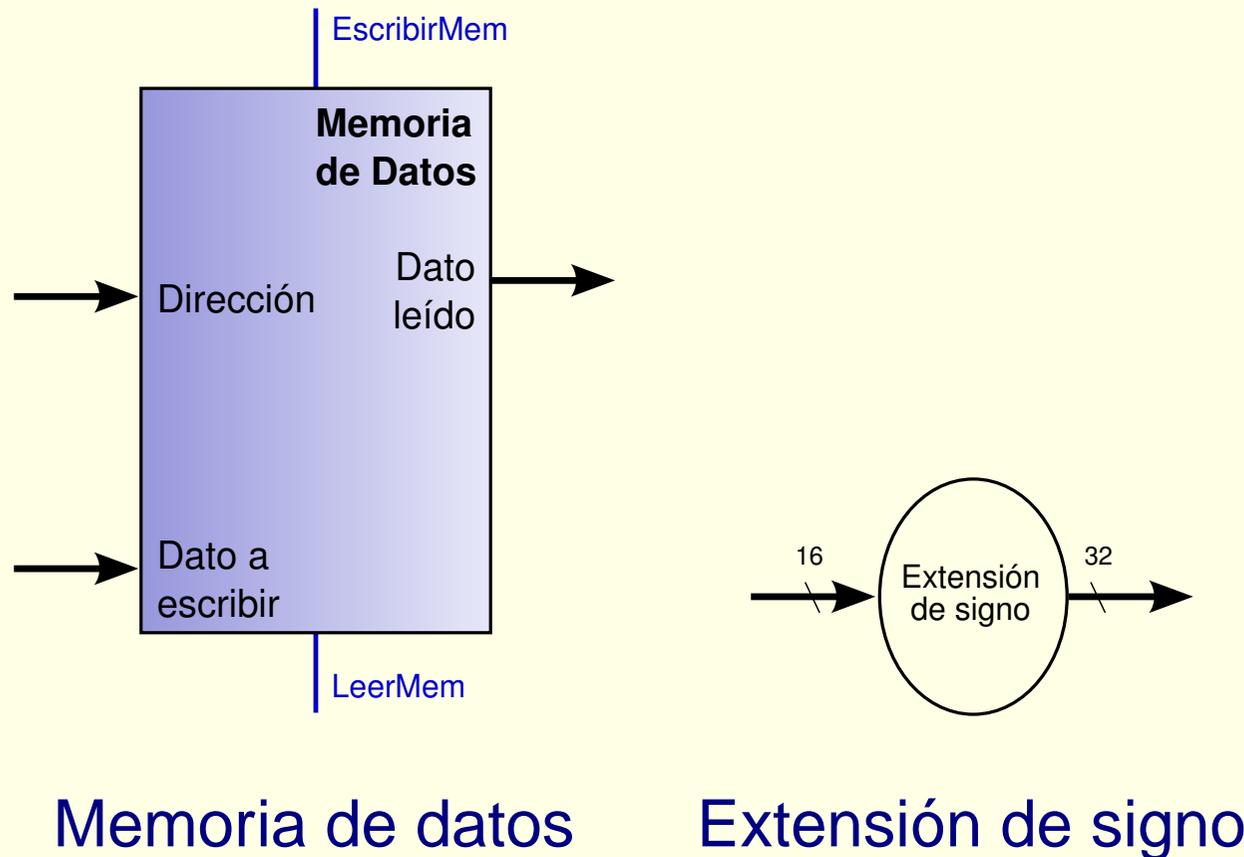


4.3 Instrucciones de carga-almacenamiento

- ▶ Elementos necesarios para instrucciones tipo «**lw** \$t1, desp(\$t2)» o «**sw** \$t0, desp(\$t4)». Banco de registros, ALU y:

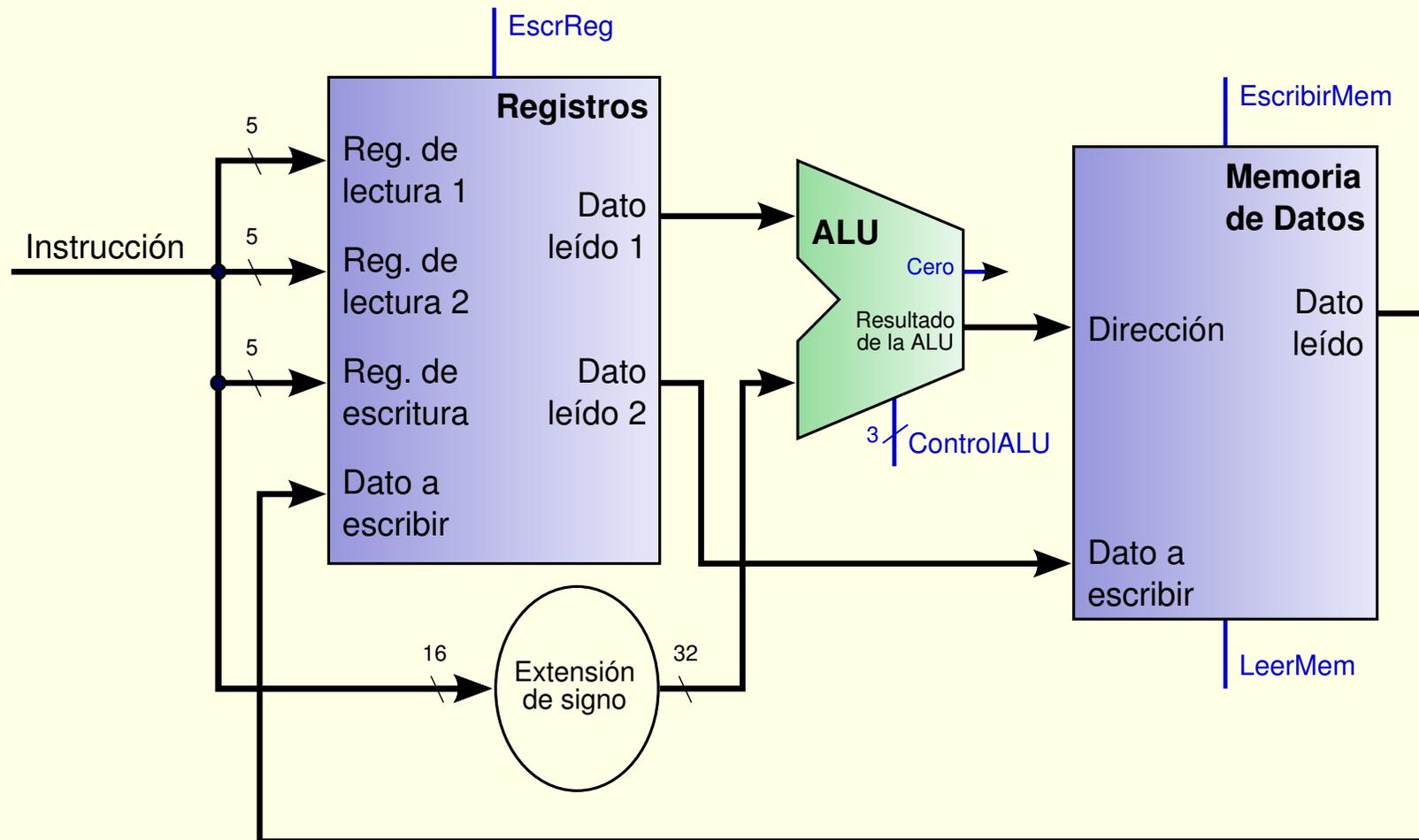
4.3 Instrucciones de carga-almacenamiento

- ▶ Elementos necesarios para instrucciones tipo «**lw** \$t1, desp(\$t2)» o «**sw** \$t0, desp(\$t4)». Banco de registros, ALU y:



4.3 Instrucciones de carga-almacenamiento (II)

- ▶ Parte de la ruta de datos para las instrucciones de carga-almacenamiento:



4.4 Instrucción de salto condicional

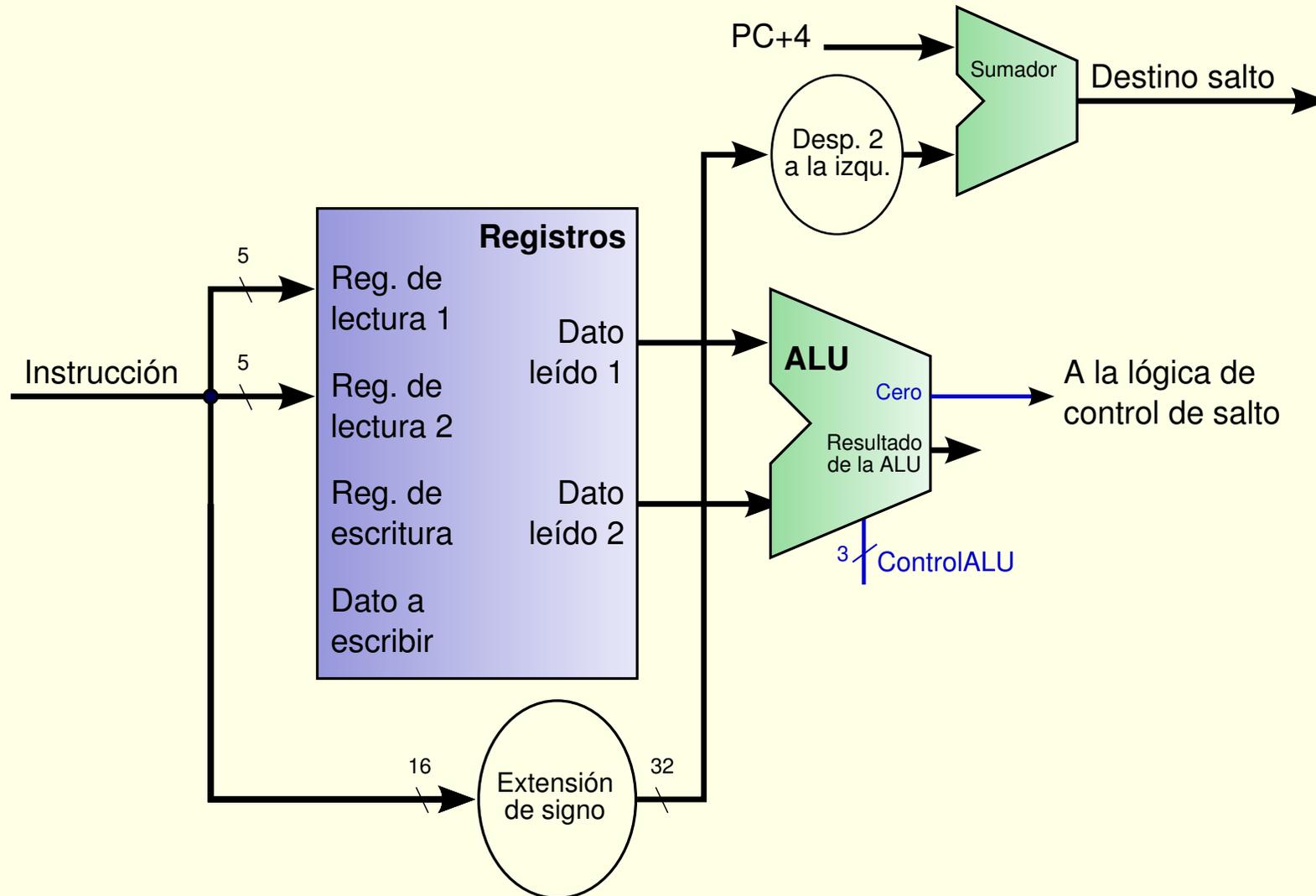
- ▶ Elementos necesarios para instrucciones tipo «**beq** \$t1,\$t2,bucle»

4.4 Instrucción de salto condicional

- Elementos necesarios para instrucciones tipo «**beq** \$t1,\$t2,bucle»
 - ⇒ Banco de registros.
 - ⇒ ALU.
 - ⇒ Extensión del signo.
 - ⇒ Desplazamiento 2 bits a la izquierda.
 - ⇒ Sumador.

4.4 Instrucción de salto condicional (II)

- ▶ Parte de la ruta de datos para la instrucción «**beq**»:



5 Rutas monociclo y multiciclo

- Para completar el procesador basta con:
 - ⇒ Combinar las rutas de datos vistas en una sólo y
 - ⇒ añadir las señales y unidad de control necesarias.
- En este tema se verán dos realizaciones posibles:
 - ⇒ Monociclo: todas las instrucciones se ejecutan en un único ciclo largo de instrucción.
 - ⇒ Multiciclo: utiliza múltiples ciclos de instrucción (más cortos) para cada instrucción.

6 Ruta de datos monociclo

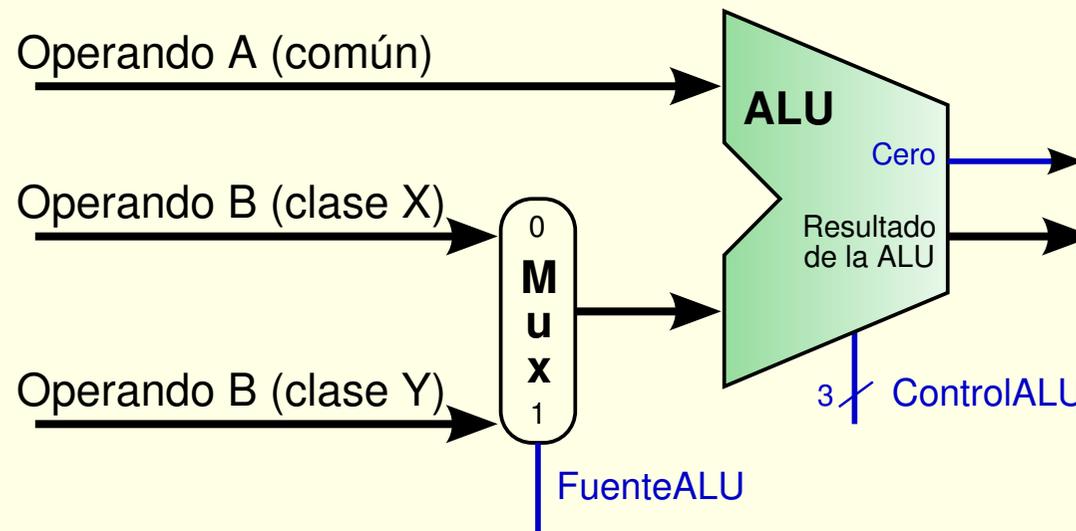
- Ésta es la realización más sencilla posible de un procesador.
- Se construirá uniendo las partes vistas anteriormente y añadiendo las señales de control necesarias.
- Recordatorio: las instrucciones que se implementarán son:
 - «**add**», «**sub**», «**and**», «**or**», «**slt**»,
 - «**lw**», «**sw**»,
 - «**beq**», y posteriormente, «**j**».

6 Ruta de datos monociclo (II)

- Todas las instrucciones se han de ejecutar en un sólo ciclo
 - Ningún elemento puede utilizarse más de una vez por instrucción
 - Cualquier recurso que se requiera más de una vez **por instrucción** deberá estar duplicado.
- Por ejemplo: la memoria debe estar separada en memoria de instrucciones y memoria de datos.

6 Ruta de datos monociclo (III)

- Por otro lado, podrán compartirse los elementos comunes a distintas instrucciones. Cuando sea necesario, deberán proporcionarse:
 - ⇒ múltiples entradas,
 - ⇒ una señal de control para seleccionar la entrada adecuada.
- Esta selección se realiza normalmente mediante un **multiplexor**.



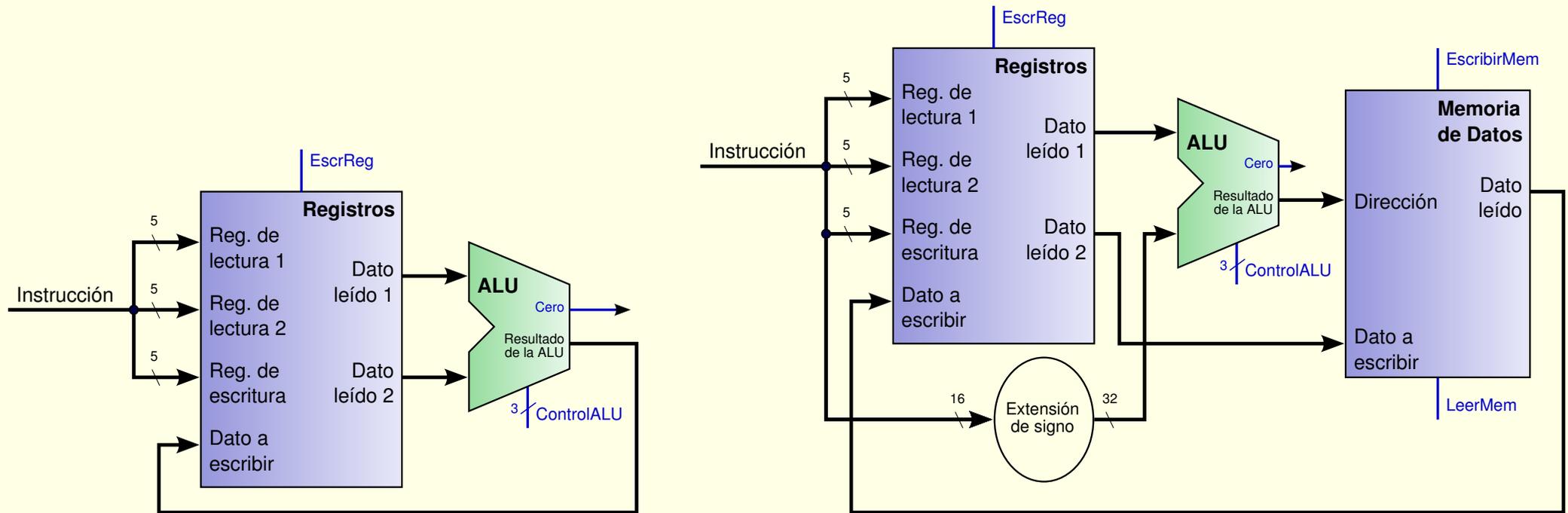
6 Ruta de datos monociclo (IV)

- ▶ Para la construcción de la ruta de dato vamos a:
 1. Combinar las partes de la ruta de datos correspondiente a las instrucción aritmético-lógicas (tipo R) y las de carga-almacenamiento.
 2. Añadir la parte correspondiente a la búsqueda de la instrucción e incremento del PC.
 3. Combinar dicha ruta de datos con la parte correspondiente a la instrucción «**beq**».

6 Ruta de datos monociclo (V)

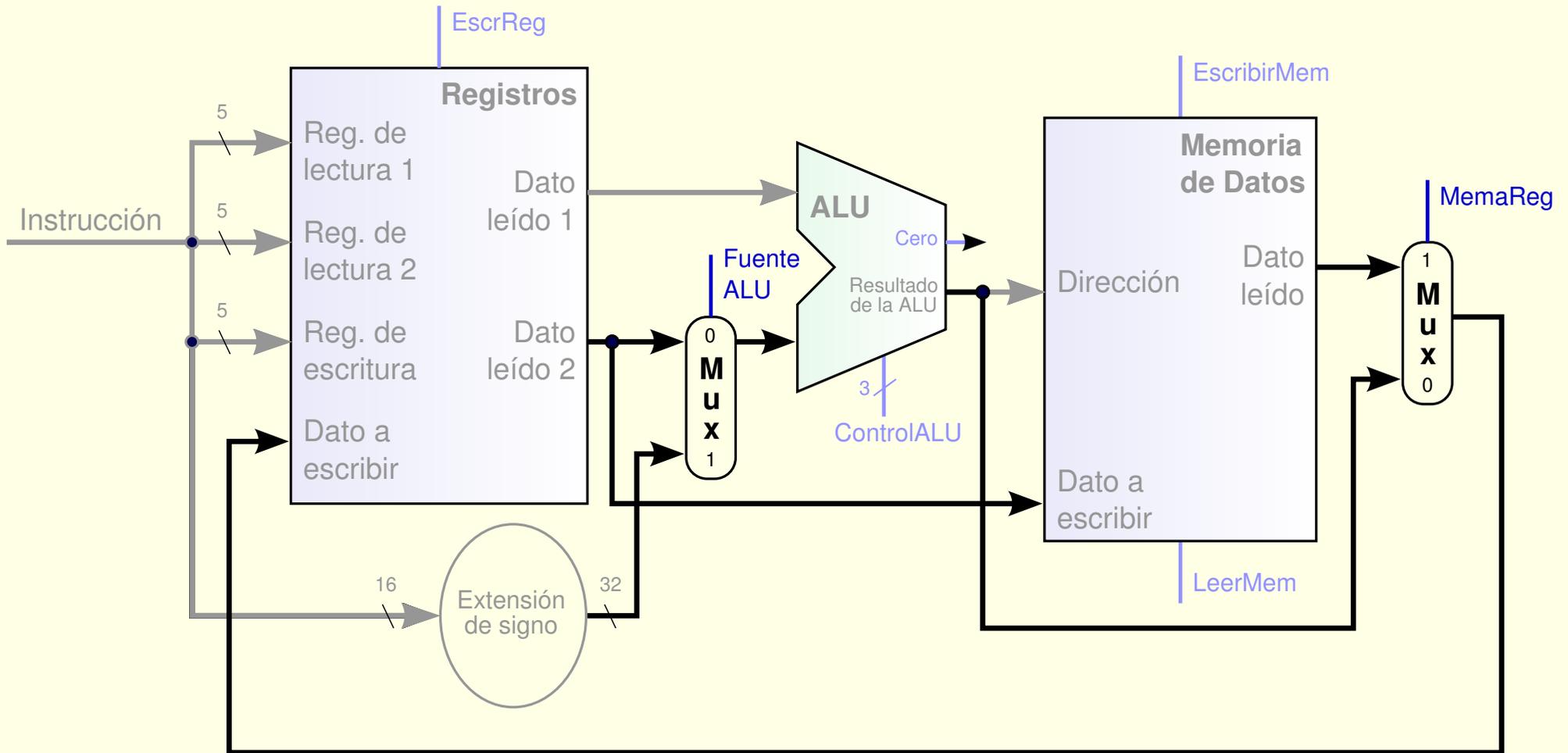
► Ejercicio 1:

Combinar las rutas de datos de las instrucciones aritmético-lógicas y las de carga-almacenamiento.



6 Ruta de datos monociclo (VI)

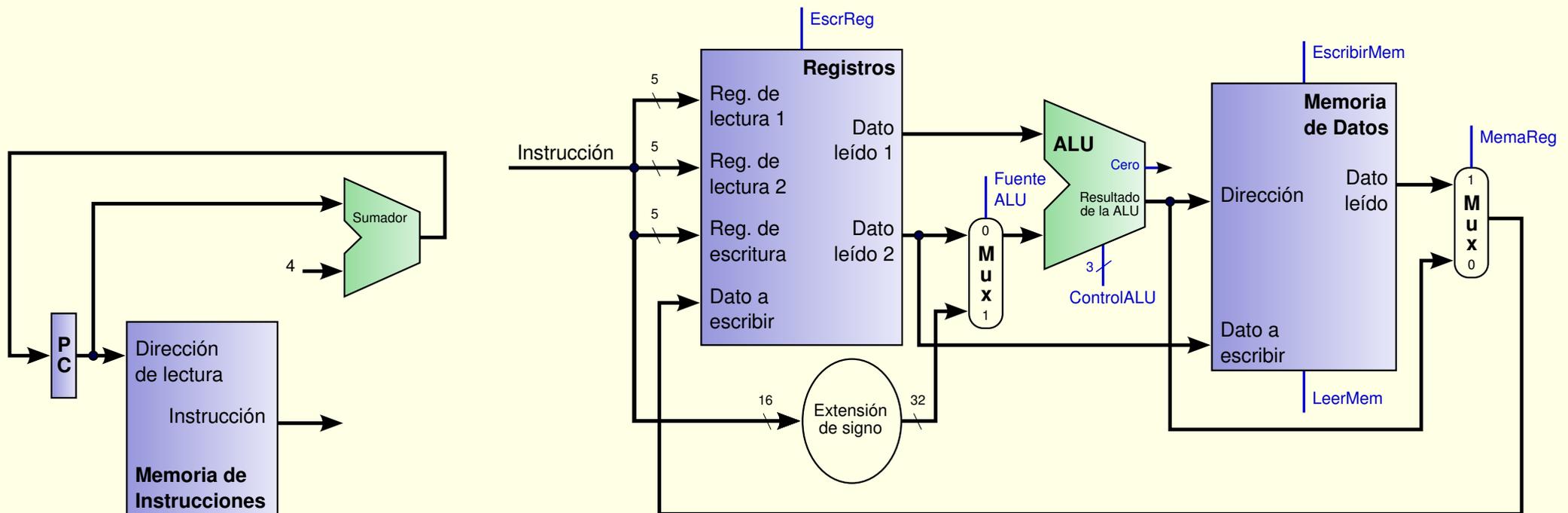
► Solución:



6 Ruta de datos monociclo (VII)

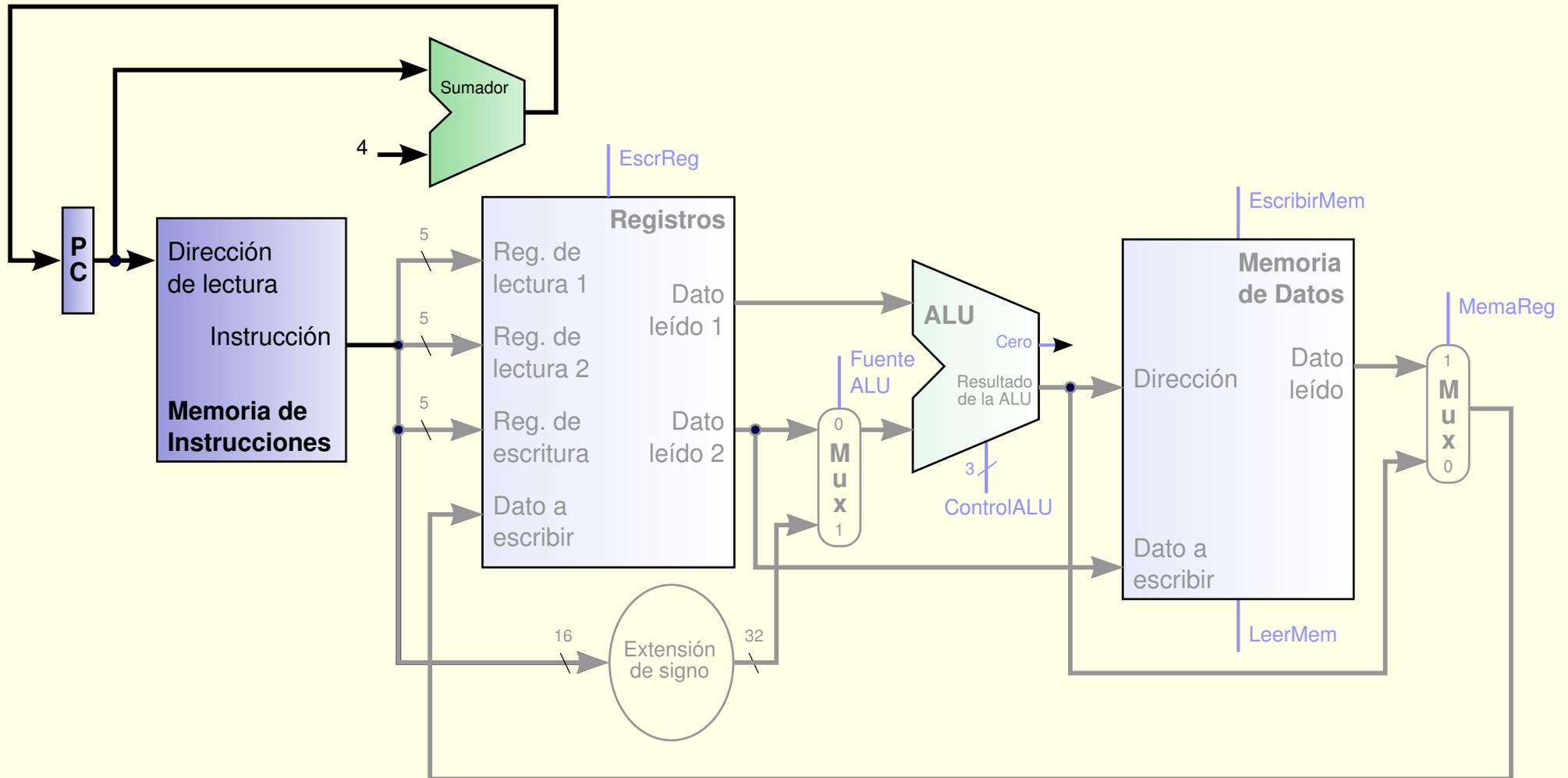
➤ Ejercicio 2:

Añadir la parte correspondiente a la búsqueda de la instrucción e incremento del PC a la ruta de datos obtenida:



6 Ruta de datos monociclo (VIII)

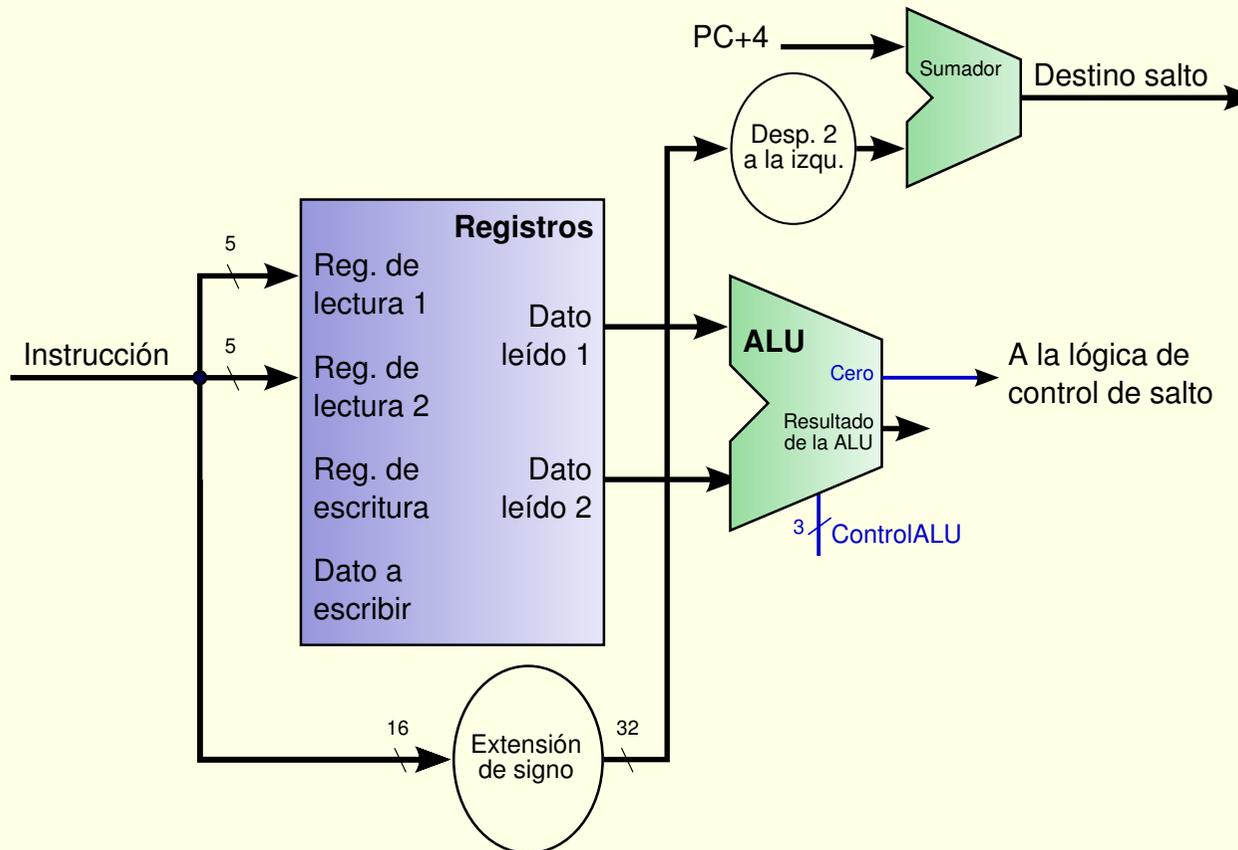
► Solución:



6 Ruta de datos monociclo (IX)

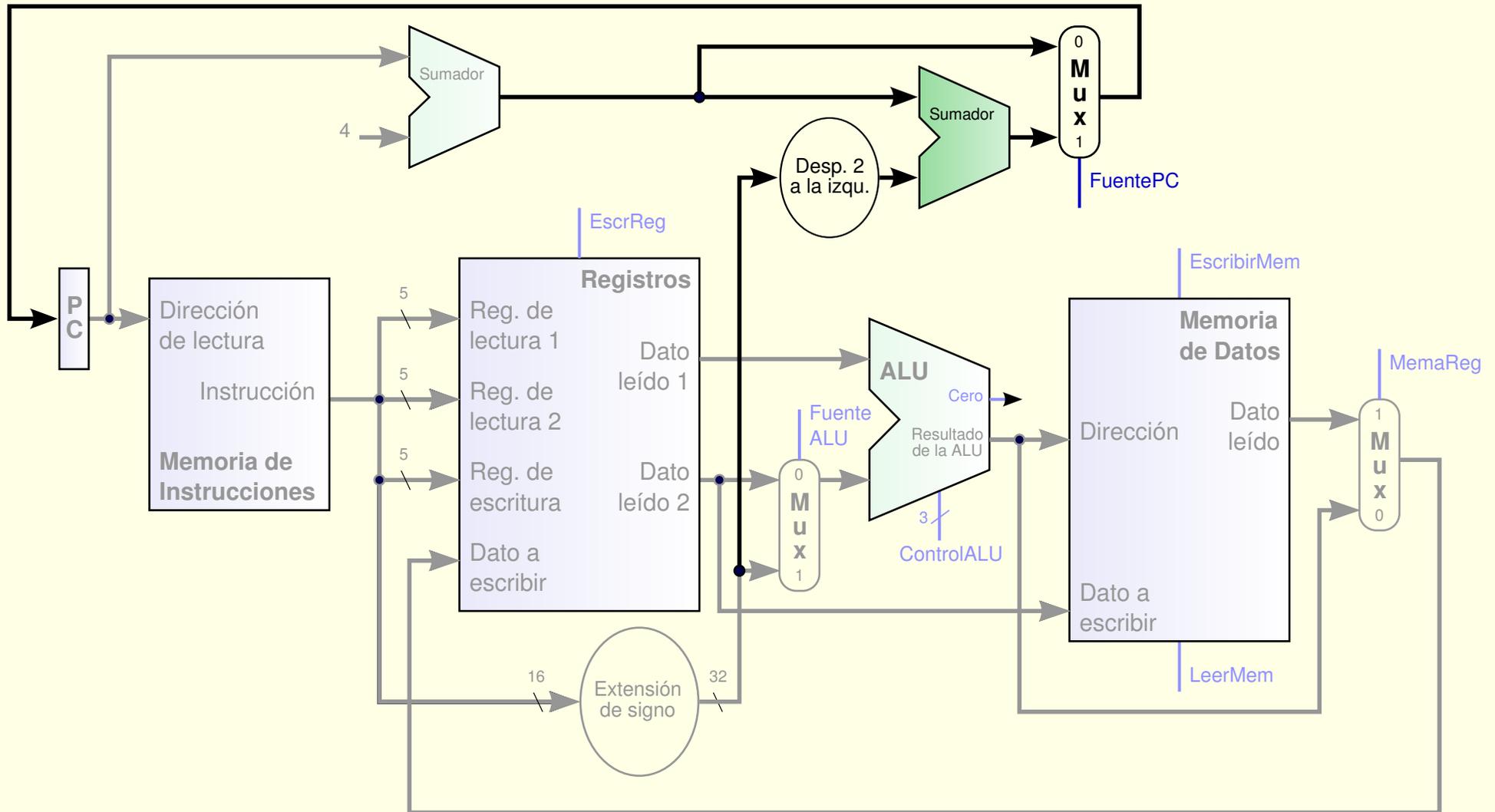
➤ Ejercicio 3:

Combinar la actual ruta de datos con la parte correspondiente a la instrucción «**beq**».



6 Ruta de datos monociclo (X)

► Solución:



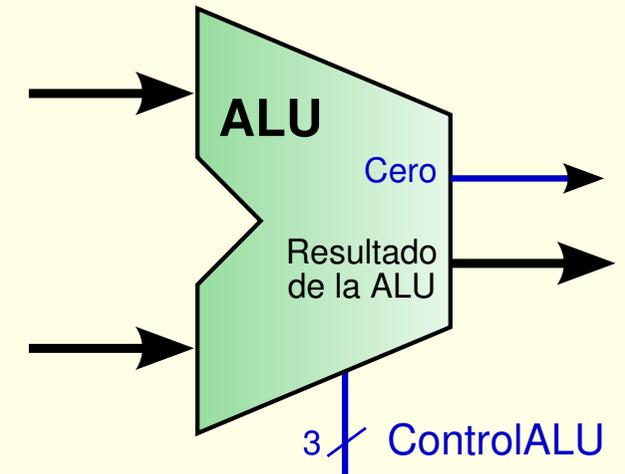
7 Unidad de control monociclo

- La unidad de control es quien determina el valor de las diferentes señales de control en un instante dado.
- Antes de diseñar la unidad de control es necesario disponer de todas las señales de control. Para ello, veremos a continuación:
 - ⇒ El control de la ALU.
 - ⇒ Qué información proporcionan las instrucciones. ¿Qué bits se utilizan para ello?
 - ⇒ Ruta de datos con todas las señales de control.
- Finalmente, se procederá al diseño de la Unidad de Control.

7.1 El control de la ALU

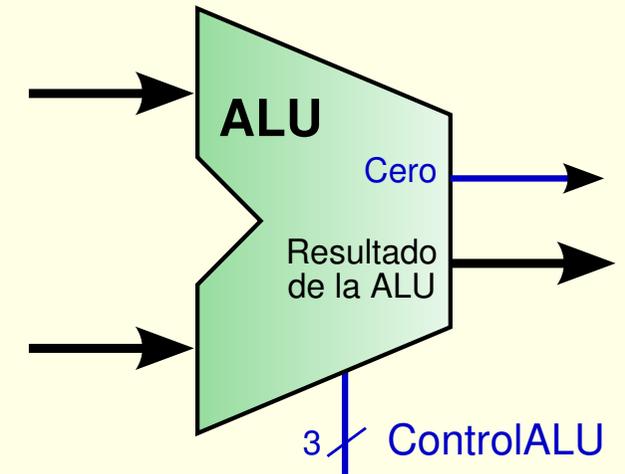
- Supongamos que la señal ControlALU determina la operación de la ALU de acuerdo con la siguiente tabla de verdad:

ControlALU	Función
000	AND
001	OR
010	suma
110	resta
111	activar si menor que



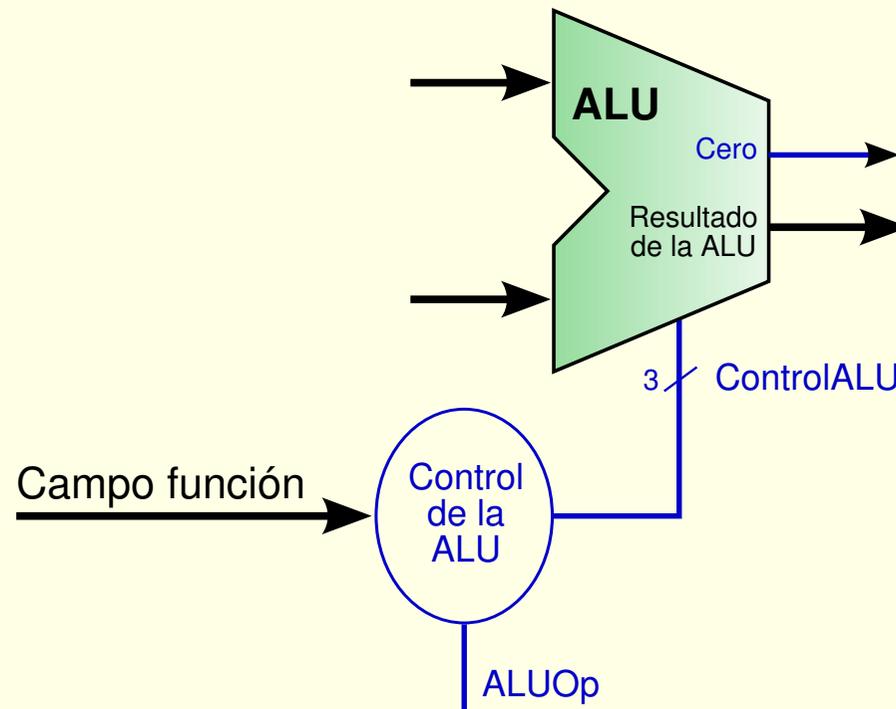
7.1 El control de la ALU (II)

- Sería posible que la Unidad de Control determinara el valor correcto de la señal ControlALU en función de la instrucción en curso.
- Sin embargo, para simplificar el diseño de la Unidad de Control, se pueden crear elementos de control especializados.
- Así, para generar la señal ControlALU se va a utilizar un elemento que llamaremos:
 - ⇒ Control de la ALU.



7.1 El control de la ALU (III)

- El Control de la ALU dispondrá de 2 entradas:
 - Señal de control ALUOp (generada por la Unidad de Control).
 - Bits del campo función de la instrucción.
- y 1 salida: señal ControlALU.



7.1 El control de la ALU (IV)

► Funcionamiento del Control de la ALU

Código operación	ALUOp	Campo de función	Operación	Acción deseada	ControlALU
LW	00	XXXXXX	<i>cargar palabra</i>	suma	010
SW	00	XXXXXX	<i>almacenar palabra</i>	suma	010
Branch Equal	01	XXXXXX	<i>saltar si igual</i>	resta	110
R-type	10	100000	<i>suma</i>	suma	010
R-type	10	100010	<i>resta</i>	resta	110
R-type	10	100100	<i>and</i>	and	000
R-type	10	100101	<i>or</i>	or	001
R-type	10	101010	<i>poner si menor</i>	slt	111

7.1 El control de la ALU (V)

➤ Tabla de verdad del Control de la ALU

ALUOp	Campo de función	ControlALU
00	XXXXXX	010
01	XXXXXX	110
1X	XX0000	010
1X	XX0010	110
1X	XX0100	000
1X	XX0101	001
1X	XX1010	111

7.2 Formatos de instrucción

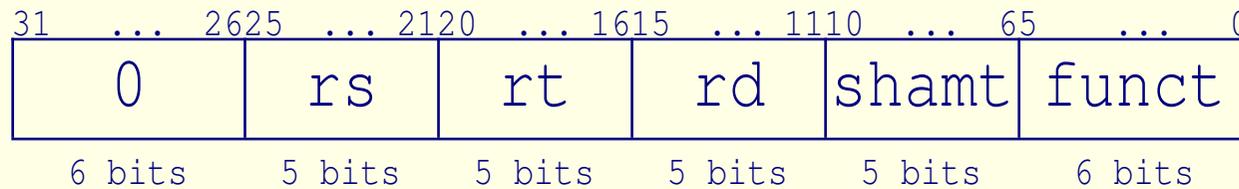
RECORDATORIO

- Antes de diseñar la unidad de control es necesario disponer de todas las señales de control. Para ello, veremos a continuación:
 - ⇒ El control de la ALU.
 - ⇒ **Qué información proporcionan las instrucciones. ¿Qué bits se utilizan para ello?**
 - ⇒ Ruta de datos con todas las señales de control.

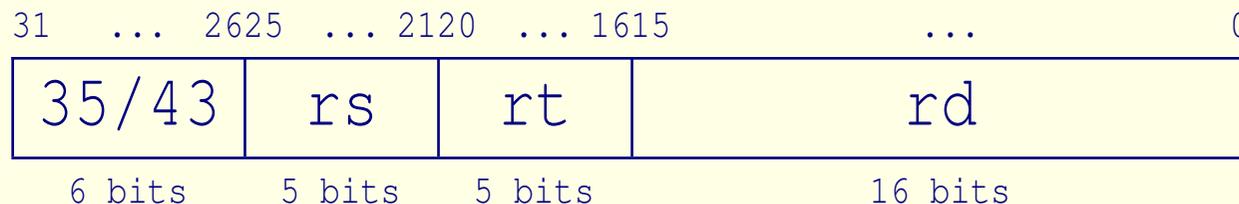
7.2 Formatos de instrucción (II)

- Las instrucciones que queremos implementar pertenecen a uno de los siguientes formatos (con el código de operación indicado):

⇒ Formato R («**add**», «**sub**», «**and**», etc.):



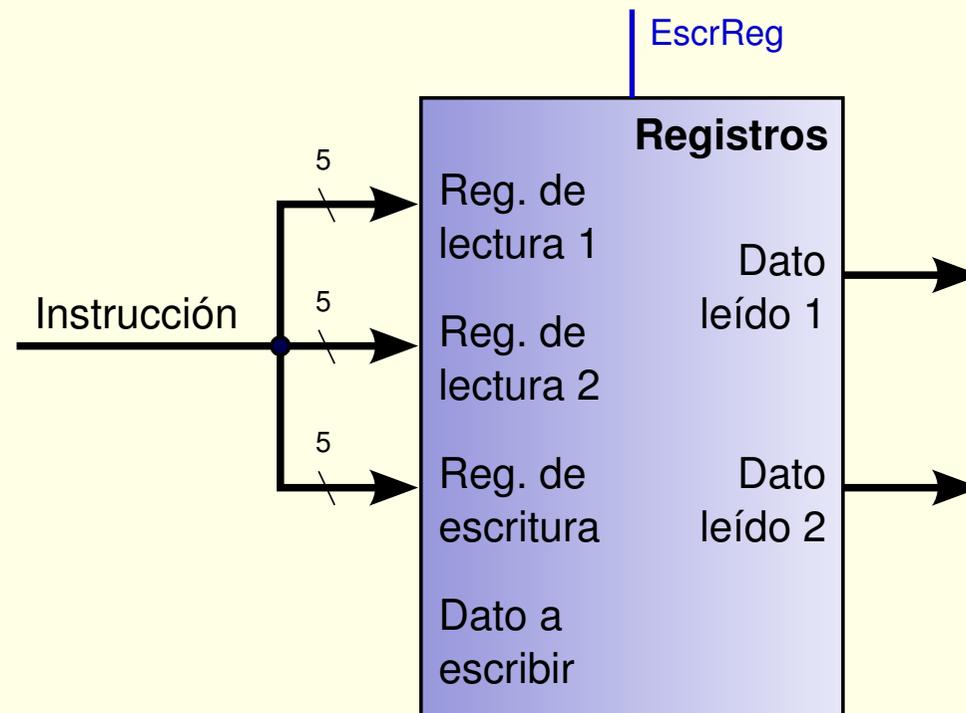
⇒ Formato I («**lw**», «**sw**» y «**beq**»):



7.2 Formatos de instrucción (III)

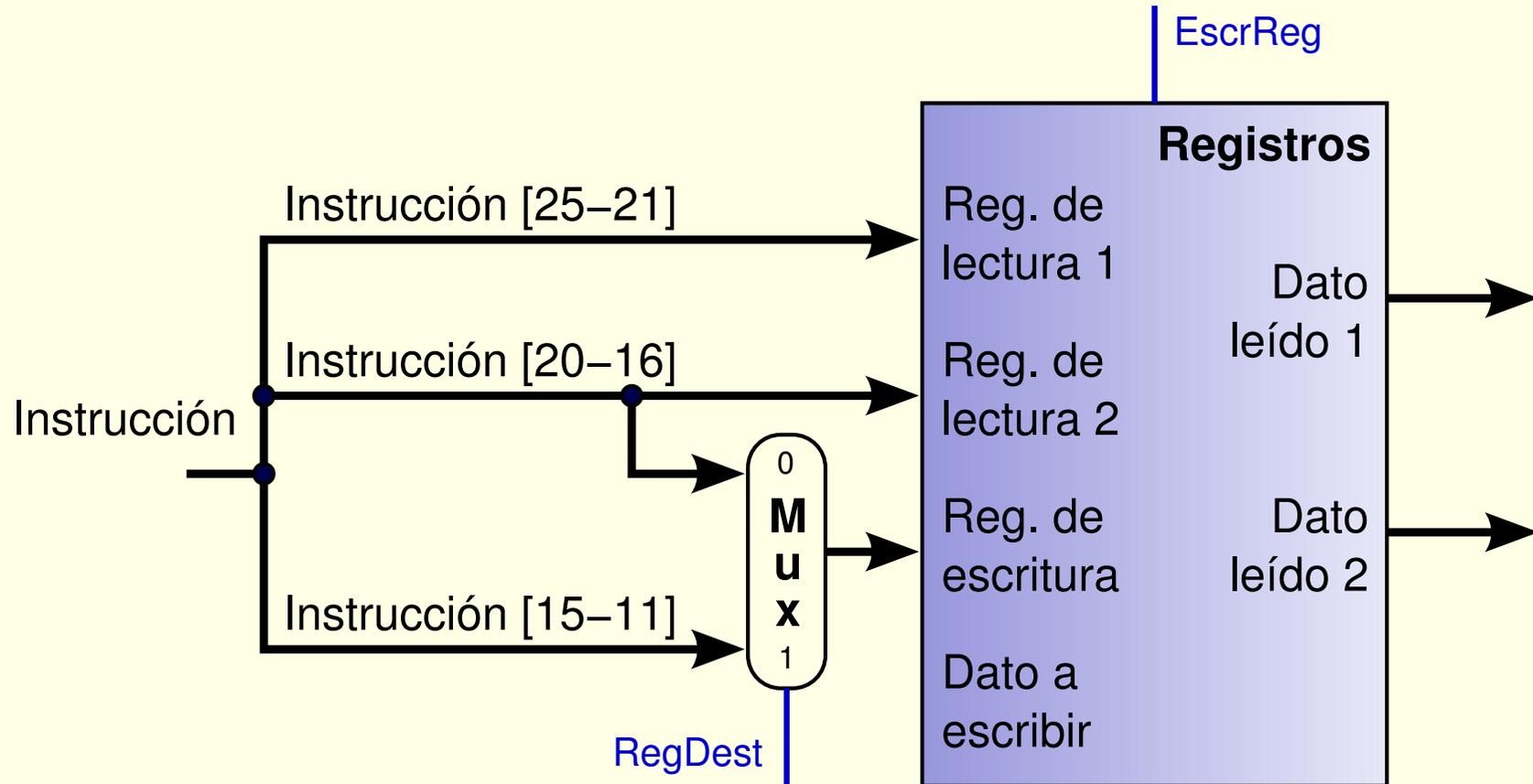
► Ejercicio 4:

Modifica la siguiente parte de la ruta de datos de forma que se especifique qué bits de la instrucción van a qué entrada del banco de registros:



7.2 Formatos de instrucción (IV)

► Solución:

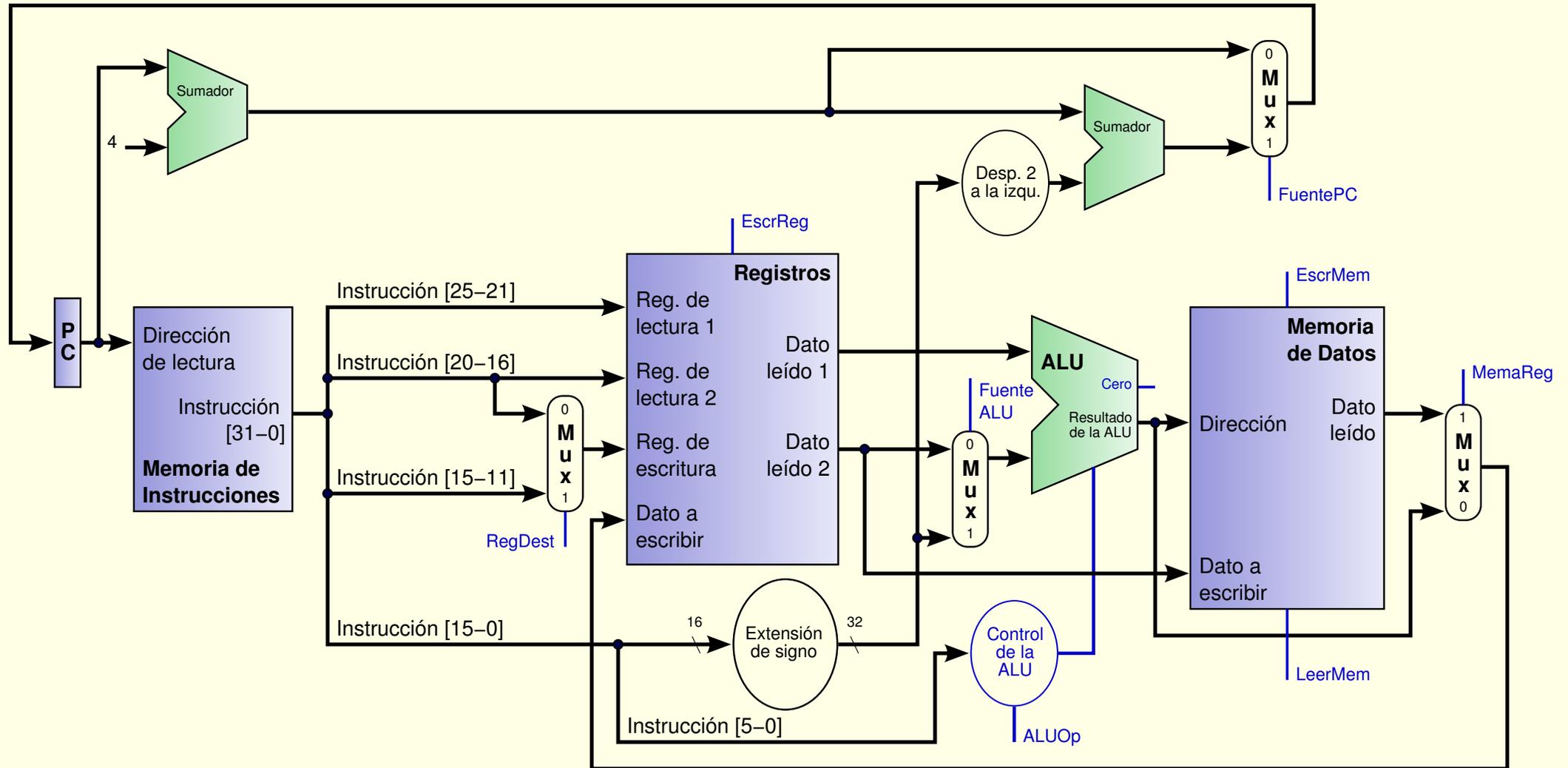


7.3 Ruta de datos con señales de control

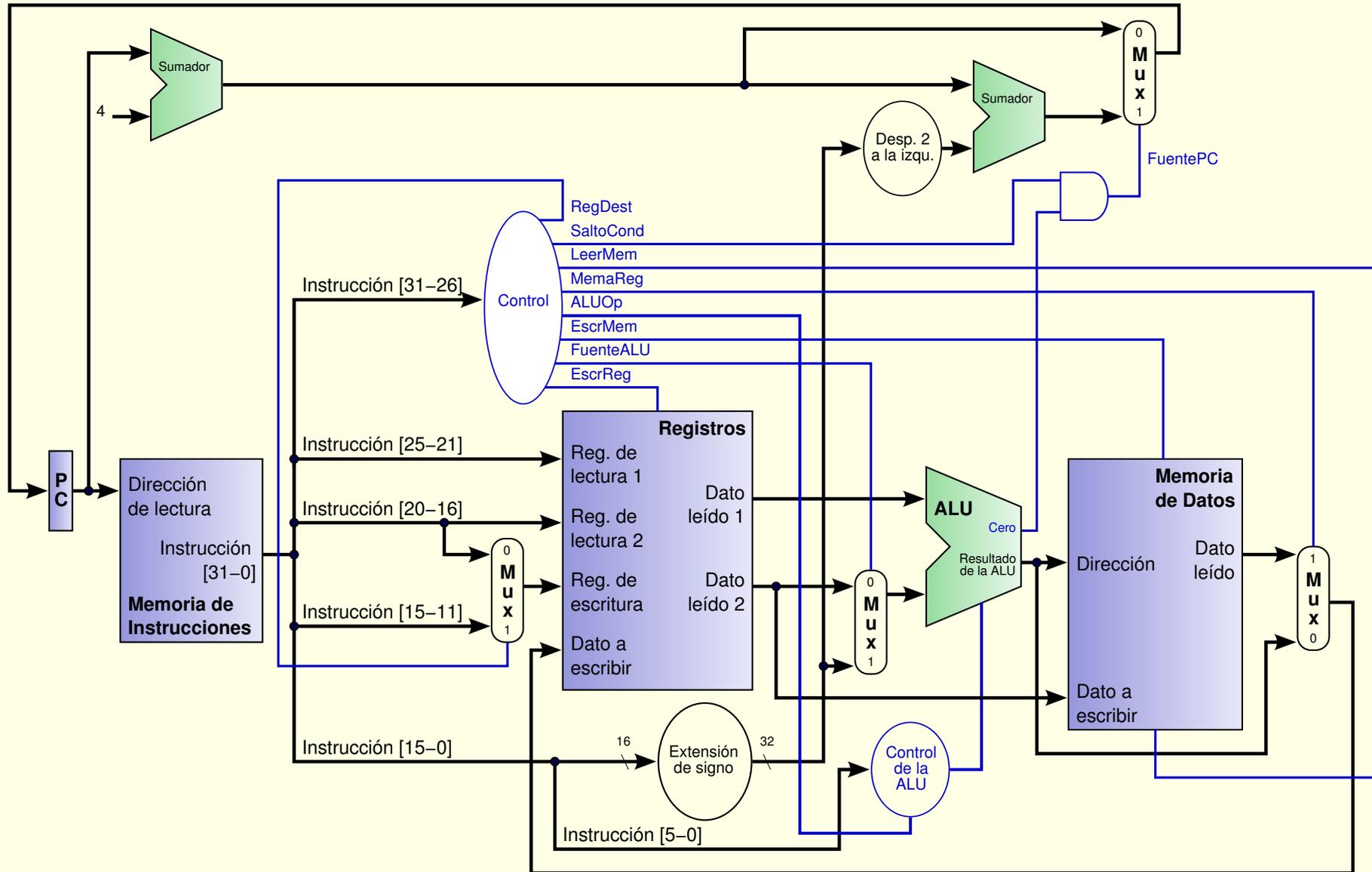
RECORDATORIO

- Antes de diseñar la unidad de control es necesario disponer de todas las señales de control. Para ello, veremos a continuación:
 - ⇒ El control de la ALU.
 - ⇒ Qué información proporcionan las instrucciones. ¿Qué bits se utilizan para ello?
 - ⇒ **Ruta de datos con todas las señales de control.**

7.3 Ruta de datos con señales de control (II)



7.4 Ruta de datos monociclo con control



7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R								1	0
«lw»								0	0
«sw»								0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1							1	0
«lw»								0	0
«sw»								0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1	0						1	0
«lw»								0	0
«sw»								0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1	0	0					1	0
«lw»								0	0
«sw»								0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1	0	0	1				1	0
«lw»								0	0
«sw»								0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1	0	0	1	0			1	0
«lw»								0	0
«sw»								0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1	0	0	1	0	0		1	0
«lw»								0	0
«sw»								0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1	0	0	1	0	0	0	1	0
«lw»								0	0
«sw»								0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1	0	0	1	0	0	0	1	0
«lw»	0							0	0
«sw»								0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1	0	0	1	0	0	0	1	0
«lw»	0	1						0	0
«sw»								0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1	0	0	1	0	0	0	1	0
«lw»	0	1	1					0	0
«sw»								0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1	0	0	1	0	0	0	1	0
«lw»	0	1	1	1				0	0
«sw»								0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1	0	0	1	0	0	0	1	0
«lw»	0	1	1	1	1	0	0	0	0
«sw»	0	0	0	0	0	1	0	0	0
«beq»	0	0	0	0	0	0	1	0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1	0	0	1	0	0	0	1	0
«lw»	0	1	1	1	1	0	0	0	0
«sw»	0	0	0	0	0	1	0	0	0
«beq»	0	0	0	0	0	0	1	0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
Formato R	1	0	0	1	0	0	0	1	0
«lw»	0	1	1	1	1	0	0	0	0
«sw»								0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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							0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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						0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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					
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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	0	0	0	0
«beq»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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»								0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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						0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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	0	0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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	0	0	1

7.4 Ruta de datos monociclo con control (II)

- Figura anterior → significado señales de control.
- El valor de las señales de control queda determinado por el código de operación de la instrucción en curso:

Instrucción	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	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

7.4 Ruta de datos monociclo con control (III)

- Tabla de verdad de la Unidad de Control monociclo:

Entradas						Salidas								
CO ₅	CO ₄	CO ₃	CO ₂	CO ₁	CO ₀	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
0	0	0	0	0	0	1	0	0	1	0	0	0	1	0
1	0	0	0	1	1	0	1	1	1	1	0	0	0	0
1	0	1	0	1	1	X	1	X	0	0	1	0	0	0
0	0	0	1	0	0	X	0	X	0	0	0	1	0	1

- Tabla de verdad →

7.4 Ruta de datos monociclo con control (III)

- Tabla de verdad de la Unidad de Control monociclo:

Entradas						Salidas								
CO ₅	CO ₄	CO ₃	CO ₂	CO ₁	CO ₀	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
0	0	0	0	0	0	1	0	0	1	0	0	0	1	0
1	0	0	0	1	1	0	1	1	1	1	0	0	0	0
1	0	1	0	1	1	X	1	X	0	0	1	0	0	0
0	0	0	1	0	0	X	0	X	0	0	0	1	0	1

- Tabla de verdad → Simplificación →

7.4 Ruta de datos monociclo con control (II)

- ▶ Tabla de verdad de la Unidad de Control monociclo:

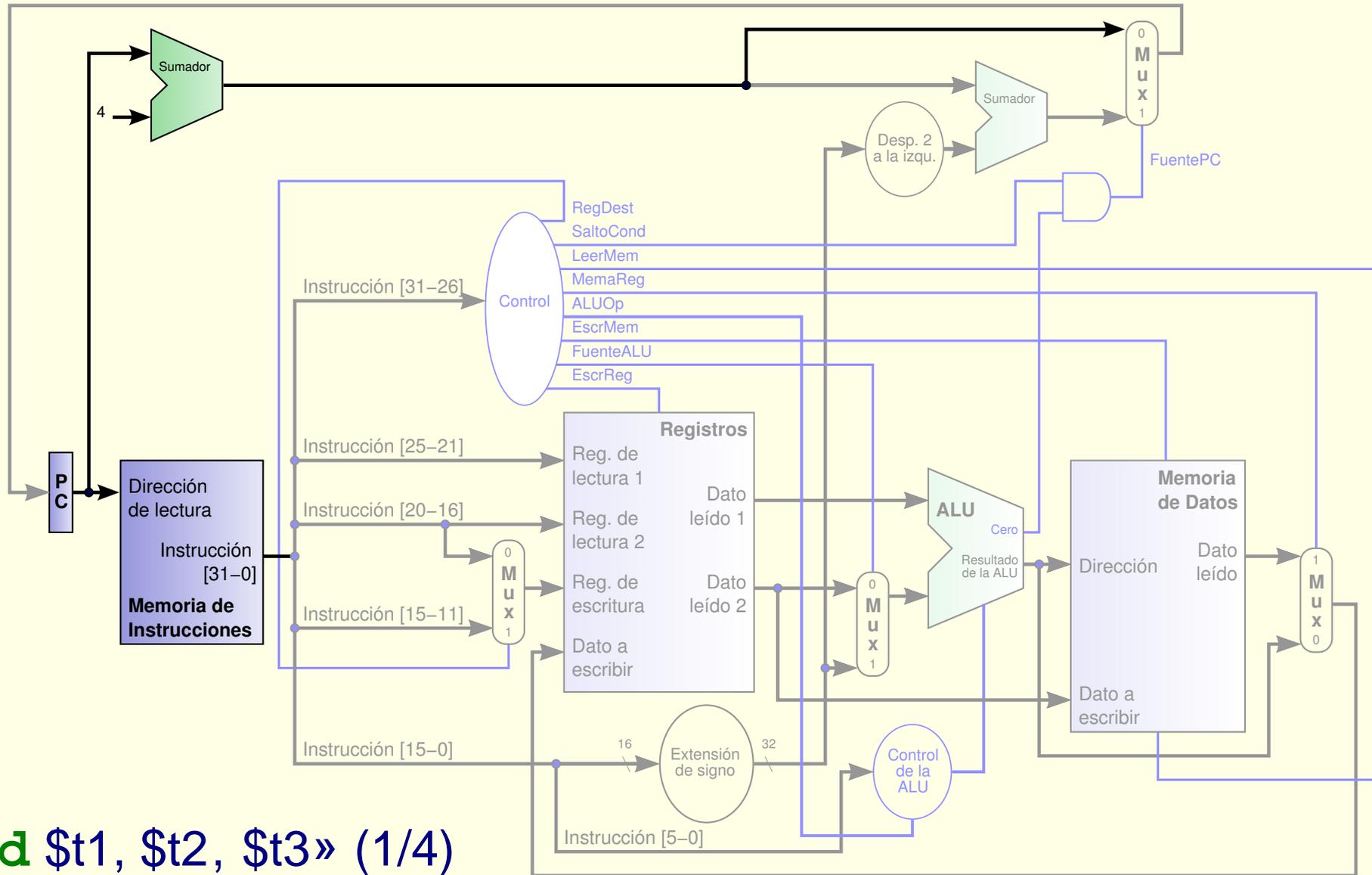
Entradas						Salidas								
CO ₅	CO ₄	CO ₃	CO ₂	CO ₁	CO ₀	RegDest	FuenteALU	MemaReg	EscrReg	LeerMem	EscrMem	SaltoCond	ALUOp1	ALUOp0
0	0	0	0	0	0	1	0	0	1	0	0	0	1	0
1	0	0	0	1	1	0	1	1	1	1	0	0	0	0
1	0	1	0	1	1	X	1	X	0	0	1	0	0	0
0	0	0	1	0	0	X	0	X	0	0	0	1	0	1

- ▶ Tabla de verdad → Simplificación → **¡Circuito Combinacional!**

7.4 Ruta de datos monociclo con control (IV)

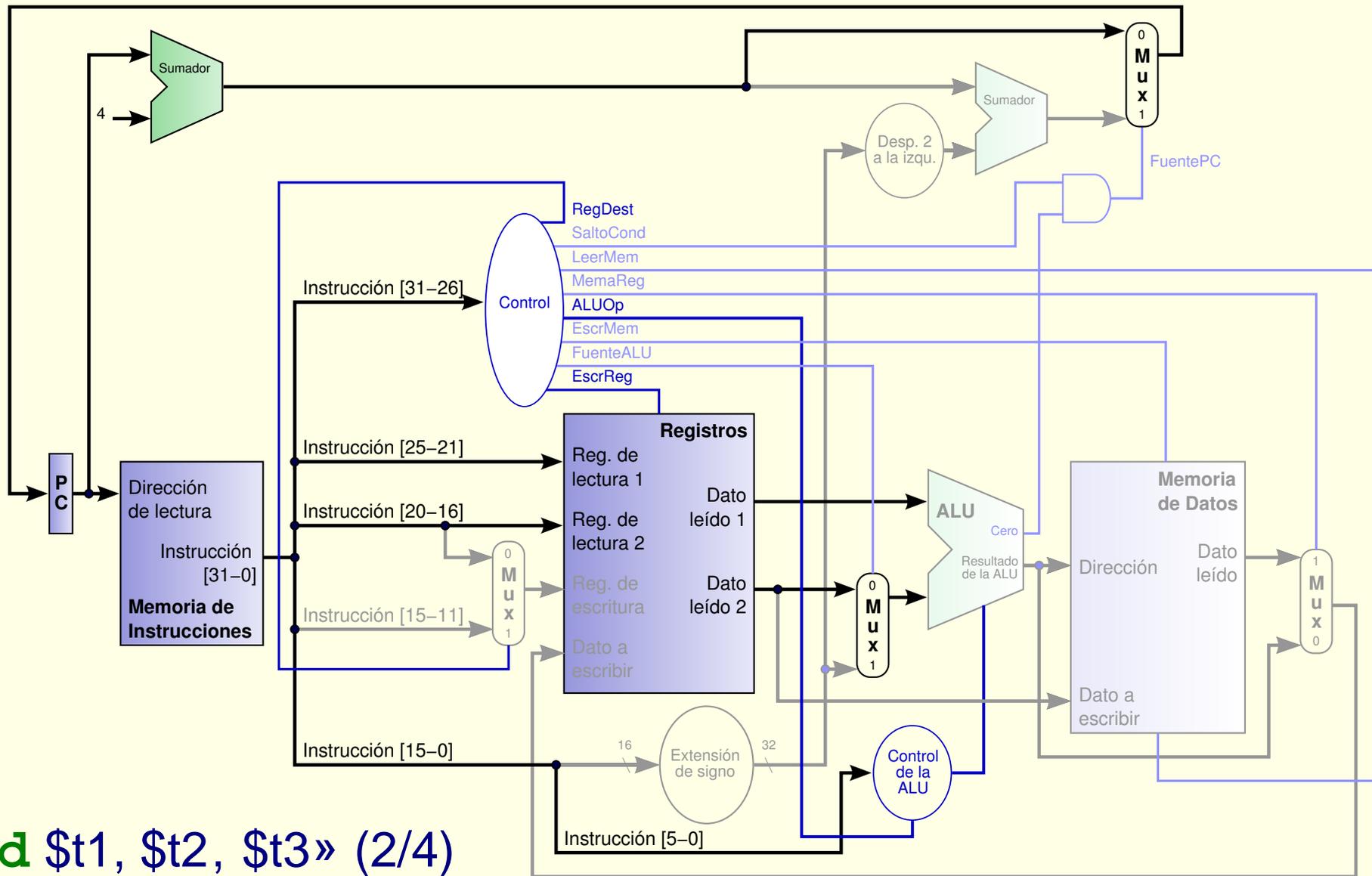
- A continuación se va mostrar el flujo a través de la ruta de datos de las siguientes instrucciones:
 - ⇒ «**add** \$t1, \$t2, \$t3»
 - ⇒ «**lw** \$t1, desp1(\$t2)»
 - ⇒ «**beq** \$t1, \$t2, desp»
- Se resaltarán en la ruta de datos:
 - ⇒ Elementos activos de la ruta de datos.
 - ⇒ Señales de control activadas.
(Si una señal de control está a cero, no se resaltará)

7.4 Ruta de datos monociclo con control (V)



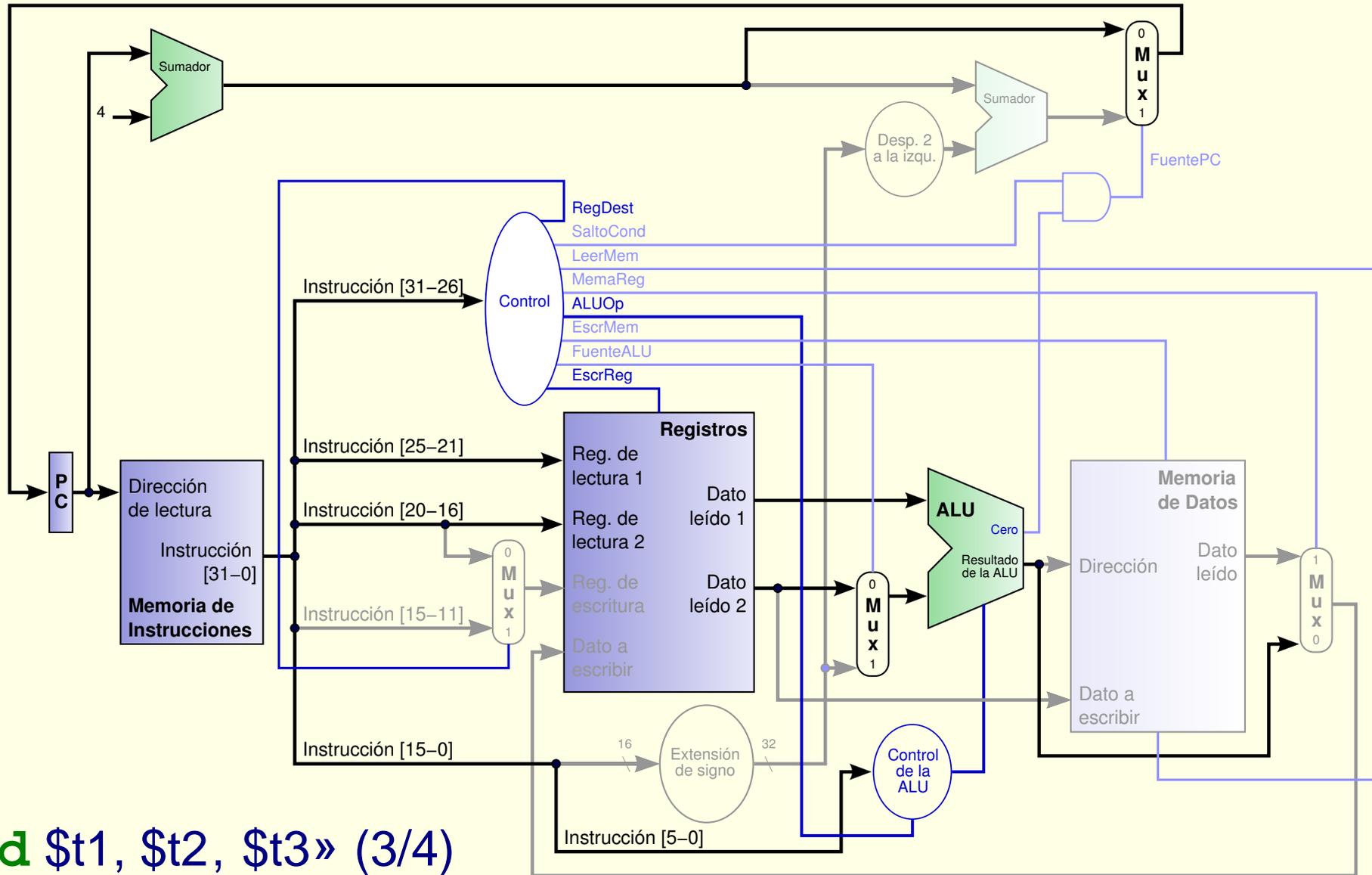
«add \$t1, \$t2, \$t3» (1/4)

7.4 Ruta de datos monociclo con control (VI)



«add \$t1, \$t2, \$t3» (2/4)

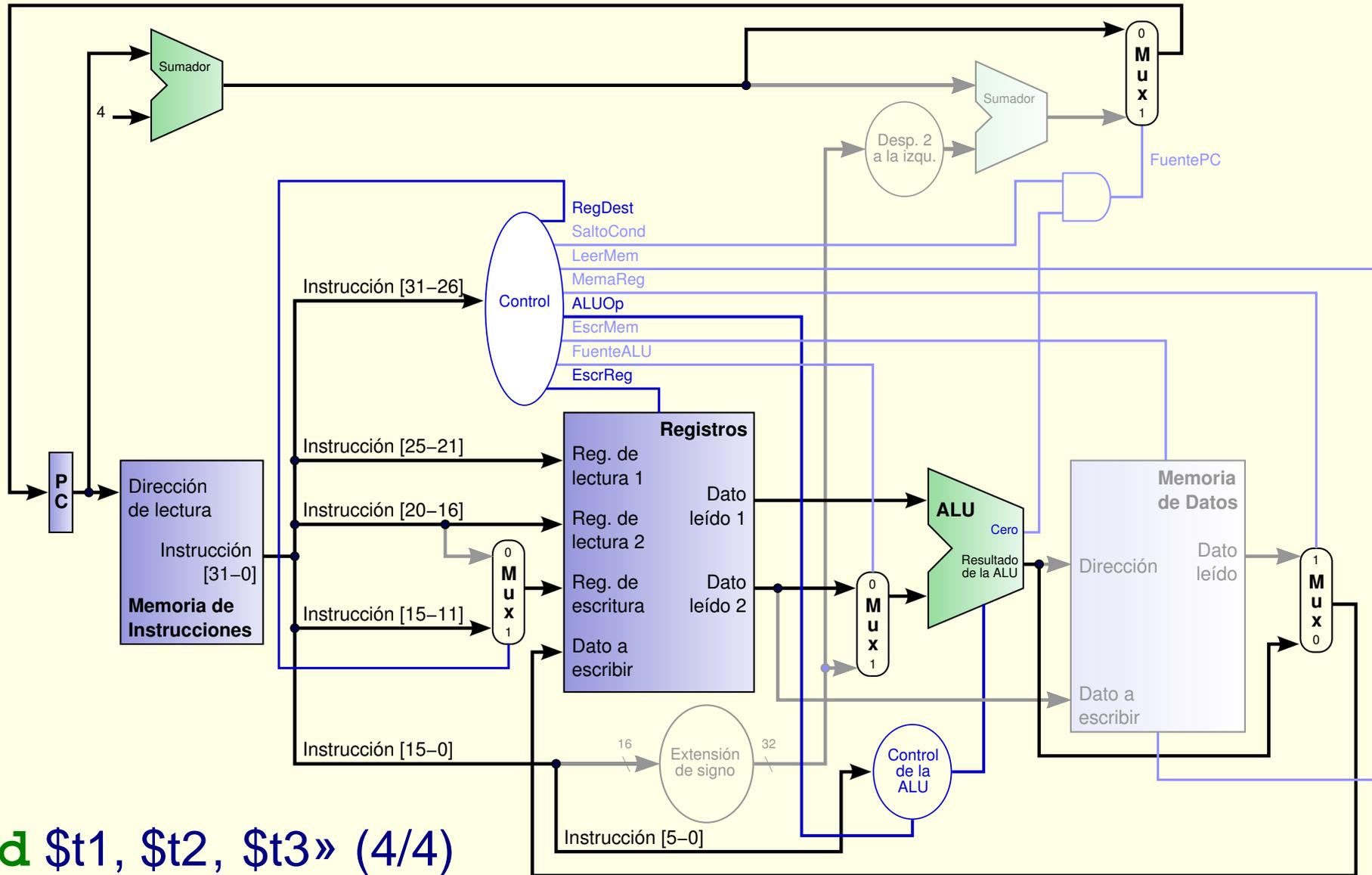
7.4 Ruta de datos monociclo con control (VII)



«add \$t1, \$t2, \$t3» (3/4)

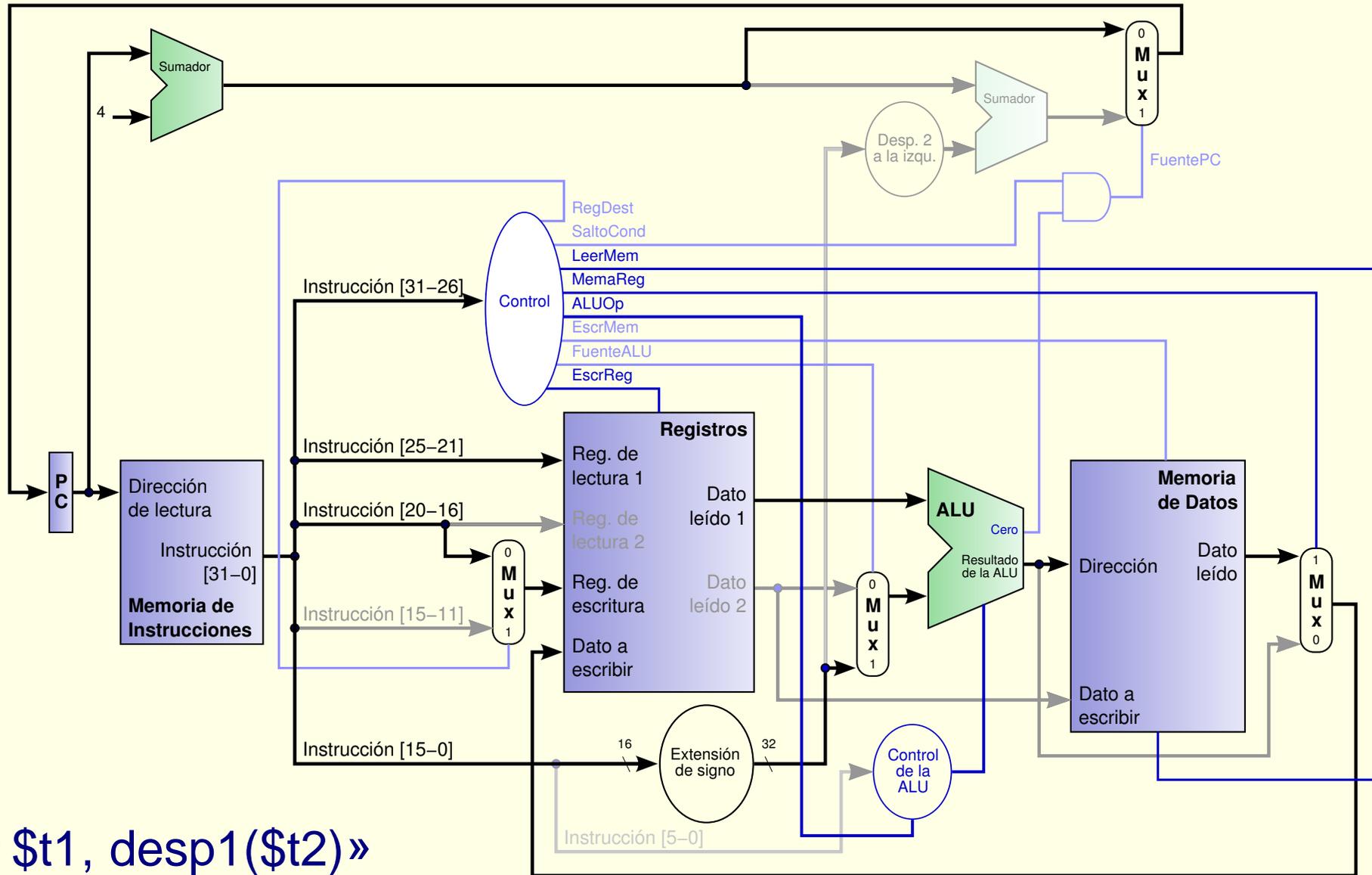


7.4 Ruta de datos monociclo con control (VIII)



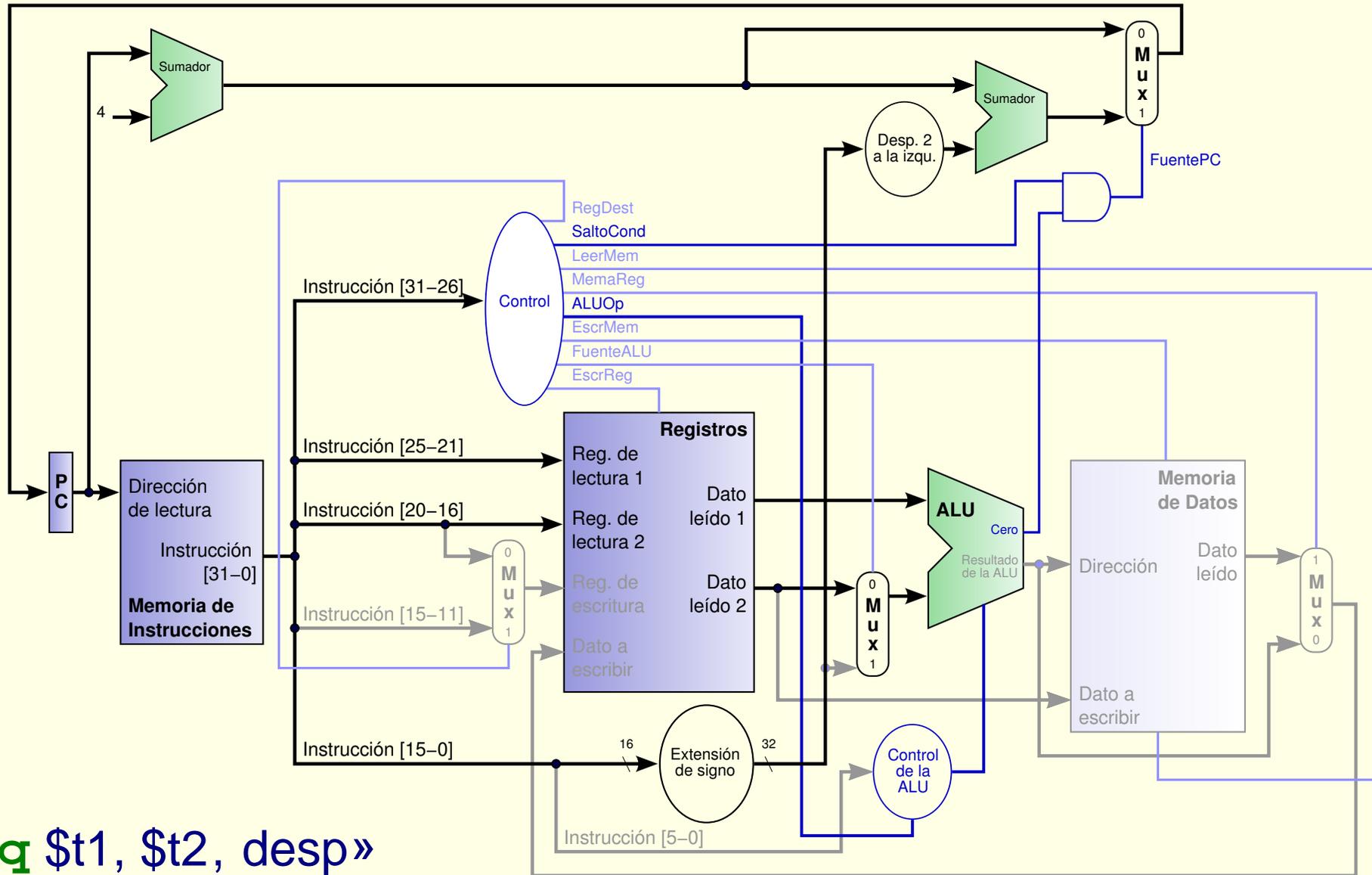
«add \$t1, \$t2, \$t3» (4/4)

7.4 Ruta de datos monociclo con control (IX)



«lw \$t1, desp1(\$t2)»

7.4 Ruta de datos monociclo con control (X)



«**beq** \$t1, \$t2, desp»

7.5 Añadir una nueva instrucción

➤ Ejercicio 5:

Amplía la ruta de datos de tal forma que sea capaz de ejecutar instrucciones de salto incondicional («j»).

➤ Recordatorio:

La dirección de salto de una instrucción «j dest» se obtienen concatenando:

- ➡ Los 4 bits de mayor peso de «PC+4».
- ➡ Los 26 bits de menor peso de la instrucción «j».
- ➡ Dos bits a 0: (00_2) .

7.5 Añadir una nueva instrucción (II)

► Solución:

Los elementos que se deben añadir son:

- ⇒ Circuitería necesaria para generar la dirección de salto.

7.5 Añadir una nueva instrucción (II)

► Solución:

Los elementos que se deben añadir son:

- ⇒ Circuitería necesaria para generar la dirección de salto.
- ⇒ Multiplexor para seleccionar:
 - ⇒ Dirección de salto incondicional o
 - ⇒ PC+4 o dirección de salto condicional.

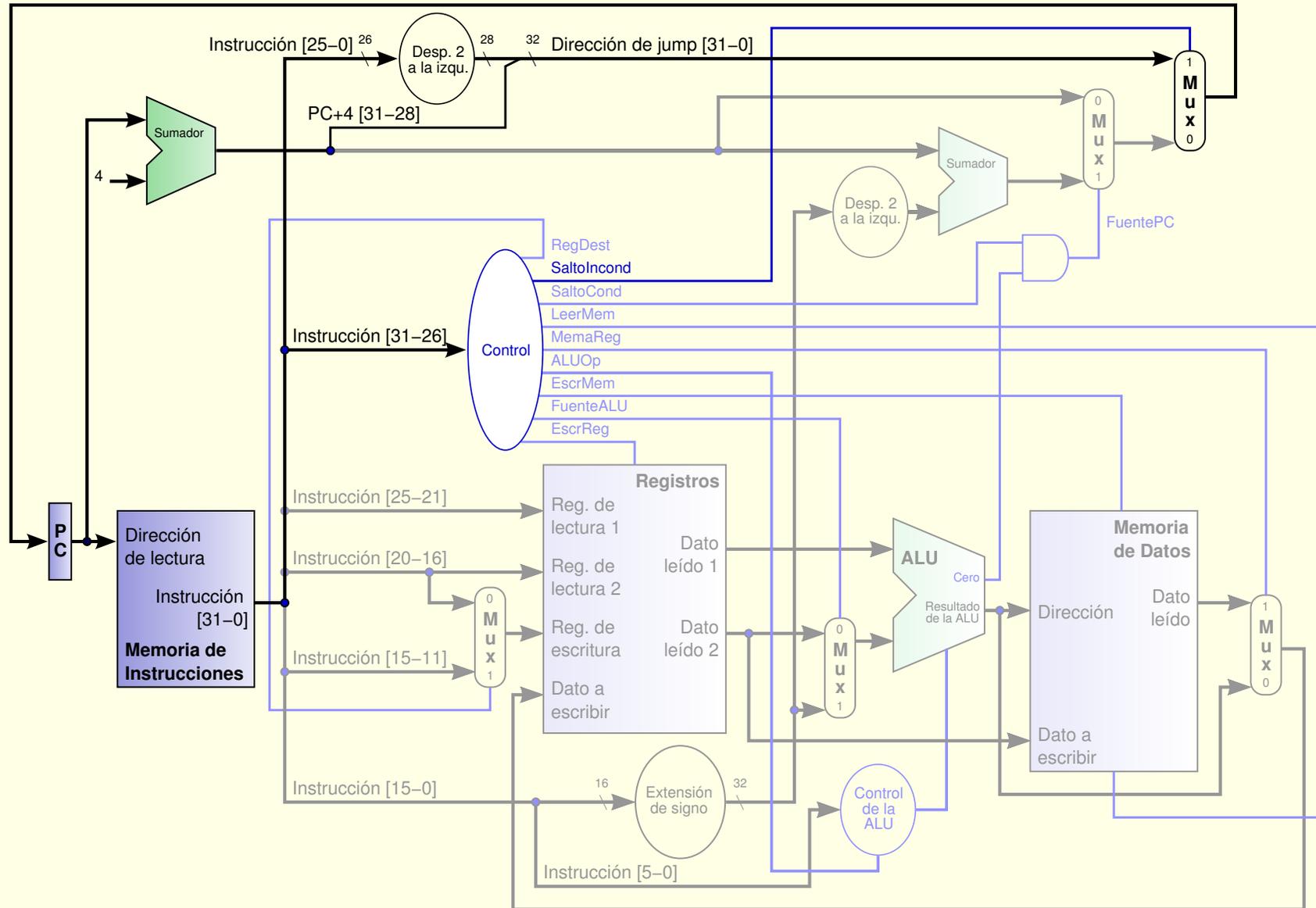
7.5 Añadir una nueva instrucción (II)

► Solución:

Los elementos que se deben añadir son:

- ⇒ Circuitería necesaria para generar la dirección de salto.
- ⇒ Multiplexor para seleccionar:
 - ⇒ Dirección de salto incondicional o
 - ⇒ PC+4 o dirección de salto condicional.
- ⇒ Nueva señal de control, SaltoIncond, que:
 - ⇒ se active cuando la instrucción sea una «j» (CO=2)
 - ⇒ y controle el anterior multiplexor.

7.5 Añadir una nueva instrucción (III)



7.6 Desventajas de la realización monociclo

- Es posible realizar un procesador utilizando esta técnica.
- Sin embargo, no se construyen debido a que es ineficiente:
 - ⇒ El tiempo de ciclo viene determinado por la instrucción más lenta
→ no es posible reducir el tiempo de las instrucciones rápidas.
 - ⇒ Las unidades funcionales sólo pueden utilizarse una vez por ciclo
→ encarece el coste de la realización.

7.6 Desventajas de la realización monociclo

- Es posible realizar un procesador utilizando esta técnica.
- Sin embargo, no se construyen debido a que es ineficiente:
 - ⇒ El tiempo de ciclo viene determinado por la instrucción más lenta
→ no es posible reducir el tiempo de las instrucciones rápidas.
 - ⇒ Las unidades funcionales sólo pueden utilizarse una vez por ciclo
→ encarece el coste de la realización.
- Realizaciones alternativas:
 - ⇒ Ruta de datos Multiciclo.
 - ⇒ Ruta de datos Segmentada.

Fin

Copyright © 2006 Sergio Barrachina Mir

Área de Arquitectura y Tecnología de Computadores
Dpt. de Ingeniería y Ciencia de los Computadores
Universidad Jaume I

Realizada con `ujislides` © 2002-5 Sergio Barrachina (barrachi@icc.uji.es)