

El procesador: camino de datos y control

Sergio Barrachina Mir

Departamento de Ingeniería y Ciencia de los Computadores
Universidad Jaume I



Contenido I

- 1 Introducción
- 2 Camino de datos monociclo
- 3 Unidad de control monociclo



Introducción

Se quiere diseñar un procesador capaz de ejecutar las siguientes instrucciones:

- «lw» y «sw» (Acceso a memoria).
- «add», «sub», «and», «or» y «slt» (Aritmético-lógicas).
- «beq» y «j» (Salto).

Bibliografía:

- Capítulo 4 de *Estructura y Diseño de Computadores: la interfaz hardware/software* (2011).

David A. Patterson y John L. Hennessy.



Introducción (II)

Gran parte de lo necesario para construir un procesador es común a todas las instrucciones.

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.

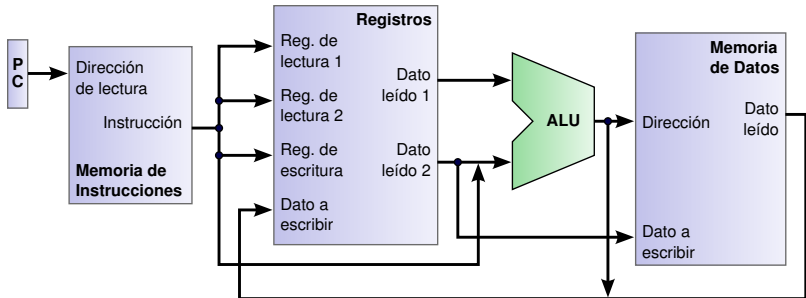
Los siguientes pasos dependen de cuál sea la instrucción.

Aún así, para instrucciones del mismo tipo, las acciones serán generalmente las mismas.



Introducción (III)

Esquema de alto nivel de una realización MIPS:



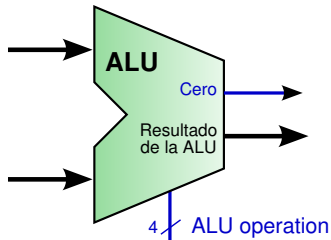
Introducción (IV)

Las unidades funcionales que se utilizan son de dos tipos:

- Combinacionales.
- Secuenciales.

Las unidades combinacionales:

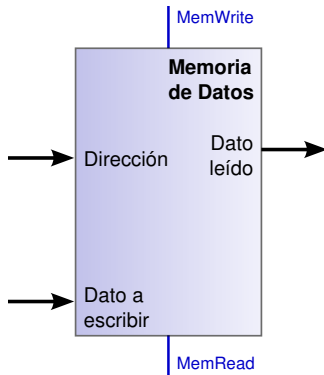
- Son aquellas en las que la salida depende exclusivamente del valor de las entradas.



Introducción (v)

Las unidades secuenciales:

- Son aquellas cuya salida depende no solo de las entradas, sino también del estado que almacenan.
- Definen el estado del computador: si se guarda su contenido, se apaga, se enciende y se restaura → como si no se hubiera apagado.



Construcción del camino de datos

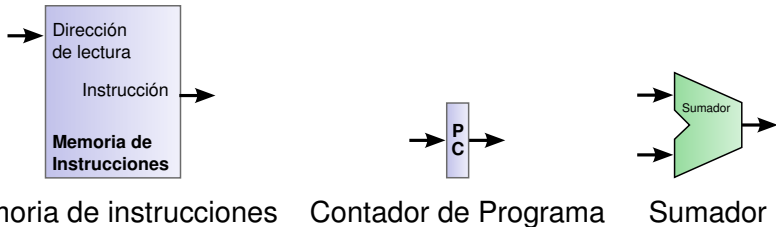
Estrategia para construir el camino de datos:

- Examinar los elementos requeridos para la ejecución de cada una de las instrucciones.
- Construir un camino de datos para cada instrucción.
- Determinar las señales de control necesarias para cada una de ellas.
- Finalmente, unir las distintas partes para generar el camino de datos completo:
 - Qué rutas del camino de datos se utilizan en cada instrucción vendrá marcado por las correspondientes señales de control.



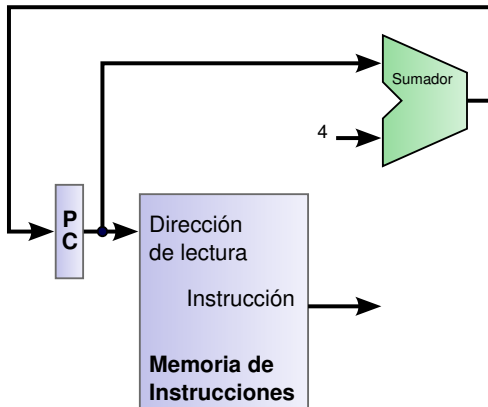
Búsqueda de la instrucción e incremento del PC

Elementos necesarios:



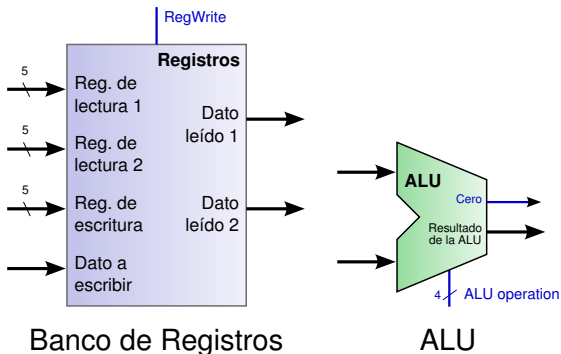
Búsqueda de la instrucción e incremento del PC (II)

Parte del camino de datos:



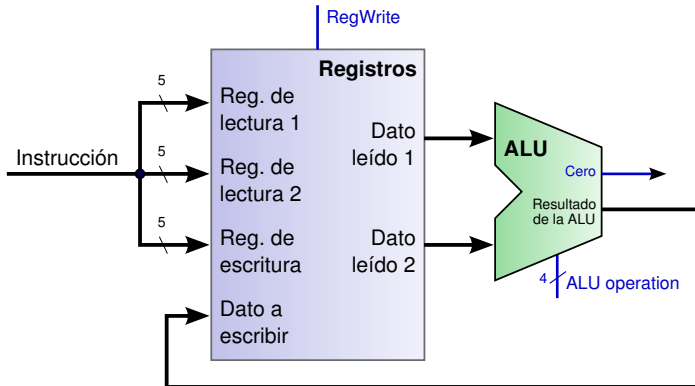
Instrucciones aritmético-lógicas (tipo R)

Elementos necesarios:



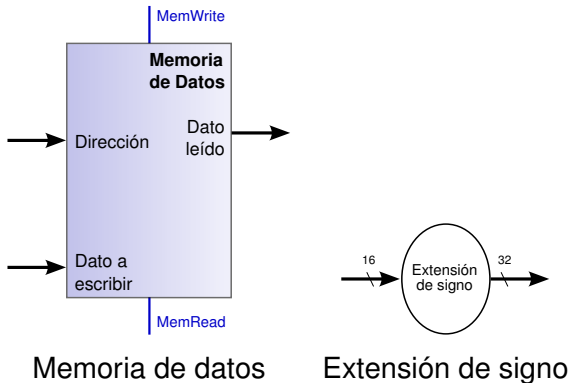
Instrucciones aritmético-lógicas (tipo R) (II)

Parte del camino de datos:



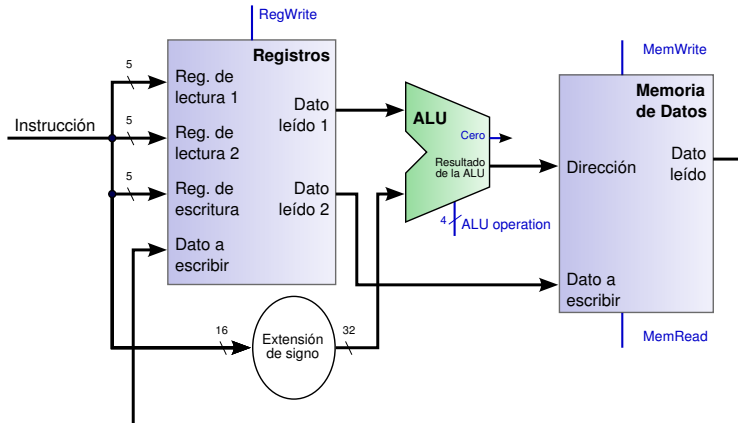
Instrucciones de carga-almacenamiento

Elementos necesarios: Banco de registros, ALU y



Instrucciones de carga-almacenamiento (II)

Parte del camino de datos:



Instrucción de salto condicional

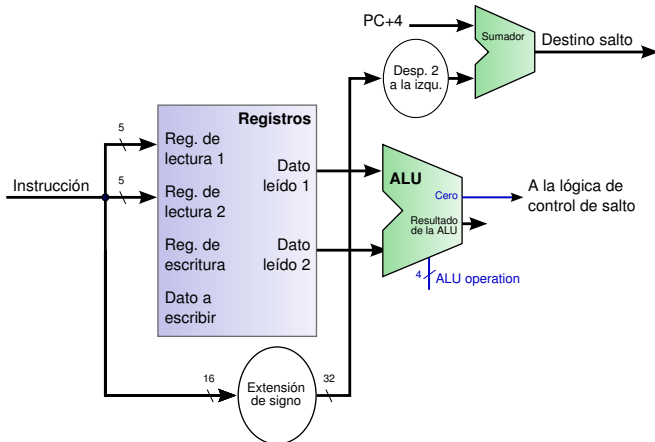
Elementos necesarios:

- Banco de registros.
- ALU.
- Extensión del signo.
- Desplazamiento 2 bits a la izquierda.
- Sumador.



Instrucción de salto condicional (II)

Parte del camino de datos:



Procesador

Para completar el diseño de un procesador basta con:

- Combinar las partes vistas en un único camino de datos y
- añadir las señales de control necesarias (así como la unidad de control).



Camino de datos monociclo

El camino de datos monociclo:

- Es la realización más sencilla posible de un procesador.
- Se construye uniendo las partes vistas anteriormente y añadiendo las señales de control necesarias.

Recordatorio

Las instrucciones que se van a implementar son:

- «**add**», «**sub**», «**and**», «**or**», «**slt**»,
- «**lw**», «**sw**»,
- «**beq**», y posteriormente, «**j**».



Camino de datos monociclo (II)

Todas las instrucciones se ejecutan en un solo ciclo:

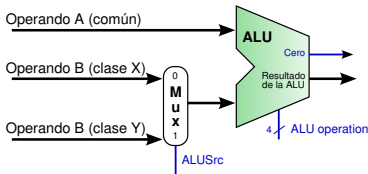
- Ningún elemento puede utilizarse más de una vez por instrucción.
- Cualquier recurso que se requiera más de una vez **por instrucción** deberá estar duplicado.
- Ejemplo: no puede haber una memoria, debe estar separada en memoria de instrucciones y memoria de datos.



Camino 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.



Camino de datos monociclo (IV)

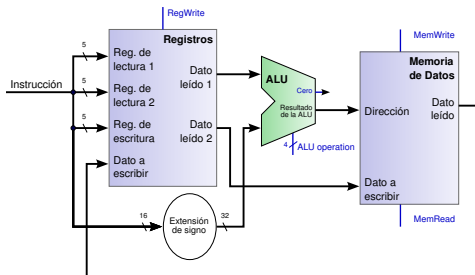
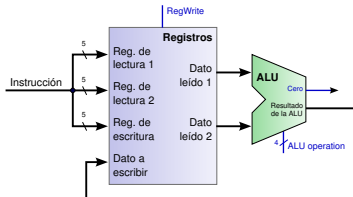
Para la construcción del camino de datos completo se va a:

- 1 Combinar el camino de datos de las instrucciones aritmético-lógicas (tipo R) y el camino de datos de las de carga-almacenamiento.
- 2 Añadir el camino de datos correspondiente a la búsqueda de la instrucción e incremento del PC.
- 3 Combinar el camino de datos resultante con el camino de datos de la instrucción «**beq**».



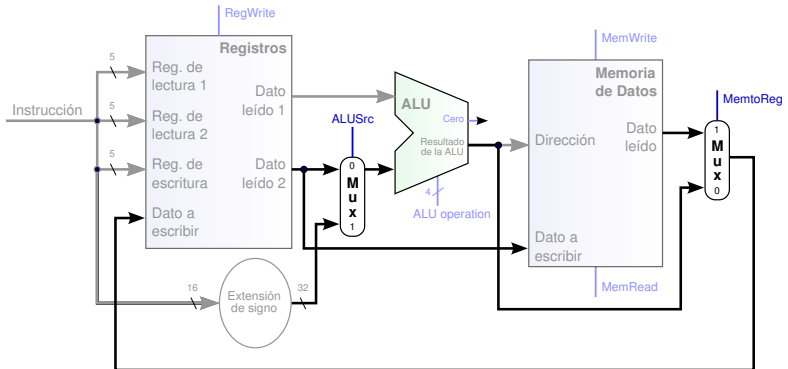
Camino de datos monociclo (v)

Ejercicio 1: Combinar los caminos de datos de las instrucciones aritmético-lógicas y de las de carga-almacenamiento.



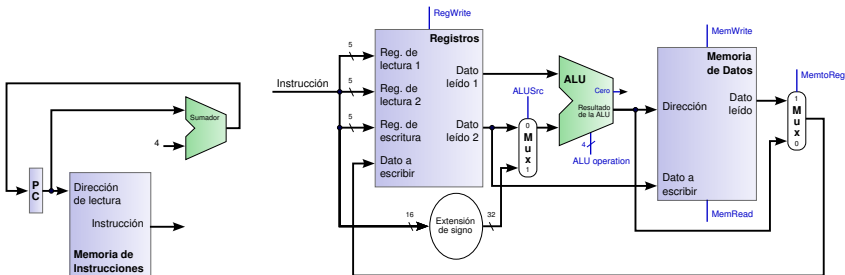
Camino de datos monociclo (VI)

Solución:



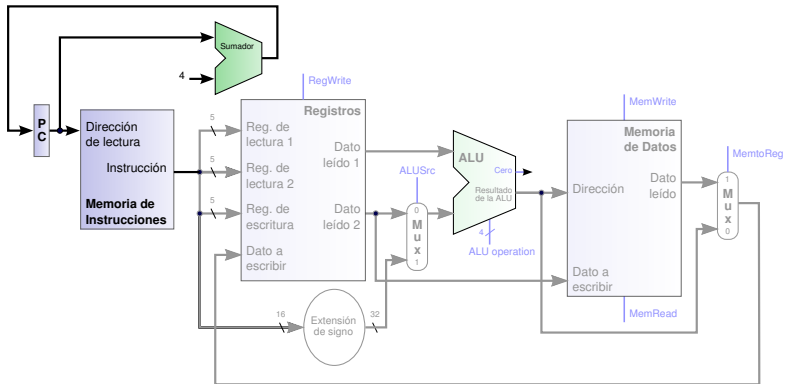
Camino de datos monociclo (VII)

Ejercicio 2: Añadir el camino de datos de búsqueda de la instrucción e incremento del PC al camino de datos anterior:



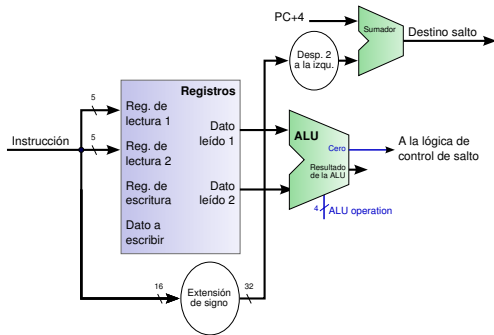
Camino de datos monociclo (VIII)

Solución:



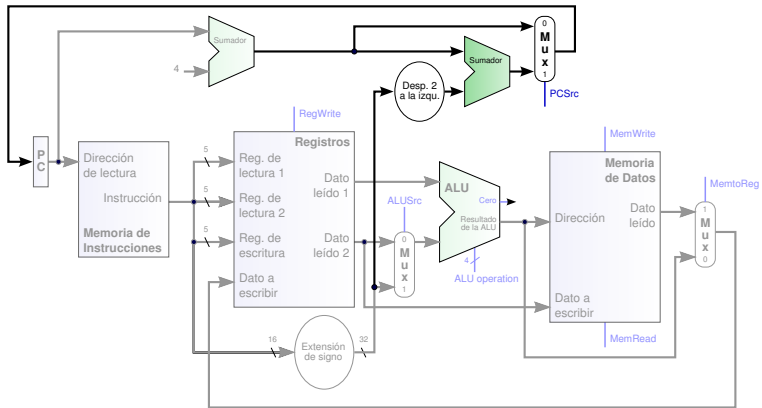
Camino de datos monociclo (IX)

Ejercicio 3: Combinar el anterior camino de datos con el correspondiente a la instrucción «**beq**».



Camino de datos monociclo (x)

Solución:



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. Por ello, se verá a continuación:

- **El control de la ALU.**
- Qué información proporcionan las instrucciones. Qué bits se utilizan.
- Camino de datos con todas las señales de control.

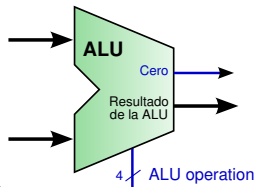
Finalmente, se procederá al diseño de la Unidad de Control.



Control de la ALU

La señal ALU operation determina la operación de la ALU de acuerdo con la siguiente tabla de verdad:

ALU operation	Función
0000	AND
0001	OR
0010	sumar
0110	restar
0111	iniciar si menor que
1100	NOR



Control de la ALU (II)

La Unidad de Control podría determinar el valor correcto de la señal ALU operation en función de la instrucción en curso.

Sin embargo, se pueden crear elementos de control especializados para simplificar el diseño de la Unidad de Control.

Así, para generar la señal ALU operation se utiliza un elemento llamado:

- Control de la ALU.

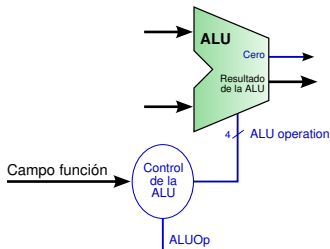


Control de la ALU (III)

El Control de la ALU dispone de 2 entradas:

- Señal de control ALUOp (de la Unidad de Control).
- Bits del campo función de la instrucción.

y 1 salida: señal ALU operation.



Control de la ALU (IV)

Funcionamiento del Control de la ALU

Código operación	ALUOp	Campo de función	Operación	Acción deseada	ALU operation
LW/SW	00	xxxxxx	<i>cargar/almacenar</i>	suma	0010
Branch Equal	01	xxxxxx	<i>saltar si igual</i>	resta	0110
R-type	10	100000	<i>suma</i>	suma	0010
R-type	10	100010	<i>resta</i>	resta	0110
R-type	10	100100	<i>and</i>	and	0000
R-type	10	100101	<i>or</i>	or	0001
R-type	10	101010	<i>poner si menor</i>	slt	0111



Control de la ALU (v)

Tabla de verdad del Control de la ALU

ALUOp	Campo de función	ALU operation
00	xxxxxx	0010
01	xxxxxx	0110
1x	xx0000	0010
1x	xx0010	0110
1x	xx0100	0000
1x	xx0101	0001
1x	xx1010	0111



Formatos de instrucción

RECORDATORIO

Antes de diseñar la unidad de control es necesario disponer de todas las señales de control. Por ello, se verá a continuación:

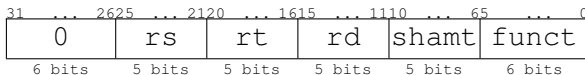
- El control de la ALU.
- **Qué información proporcionan las instrucciones. Qué bits se utilizan.**
- Camino de datos con todas las señales de control.



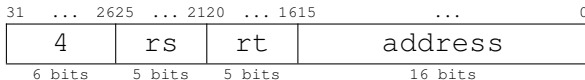
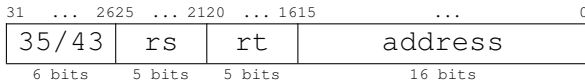
Formatos de instrucción (III)

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

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

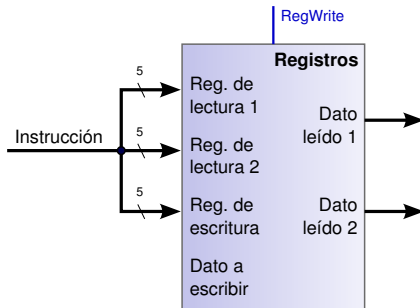


- Formato I («**lw**», «**sw**» y «**beq**»):



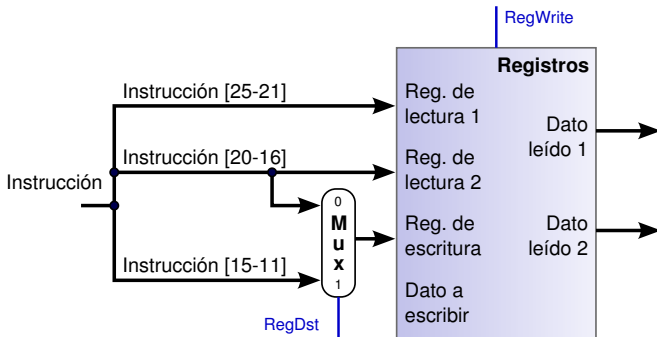
Formatos de instrucción (IV)

Ejercicio 4: Modifica la siguiente parte de el camino de datos de forma que se especifique qué bits de la instrucción van a qué entrada del banco de registros:



Formatos de instrucción (v)

Solución:



Camino de datos con señales de control

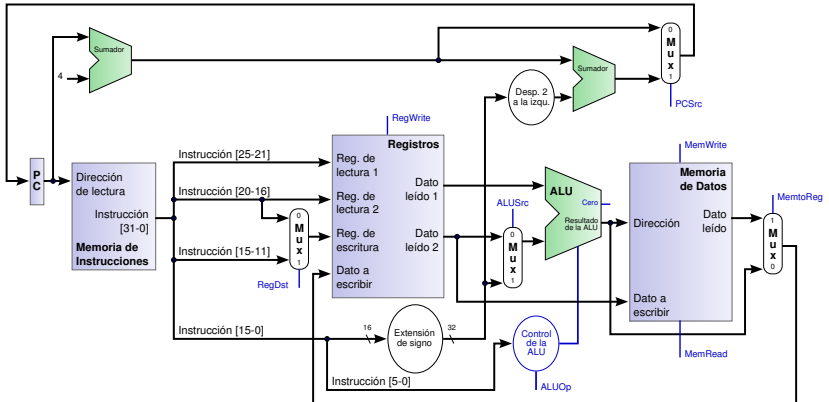
RECORDATORIO

Antes de diseñar la unidad de control es necesario disponer de todas las señales de control. Por ello, se verá a continuación:

- El control de la ALU.
- Qué información proporcionan las instrucciones. Qué bits se utilizan.
- **Camino de datos con todas las señales de control.**



Camino de datos con señales de control (II)



Camino de datos monociclo con control

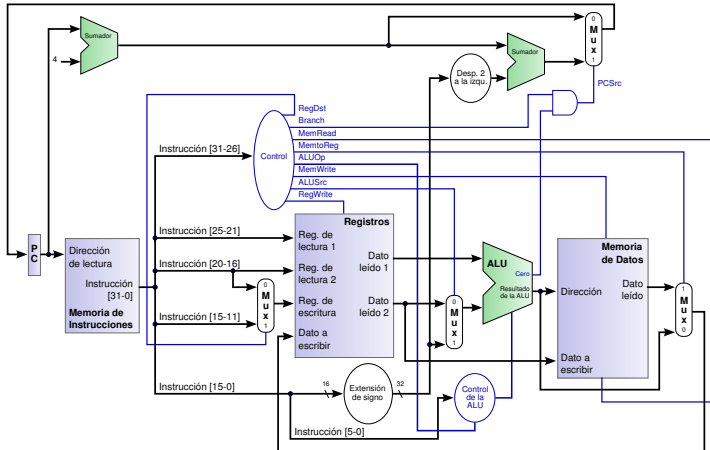


Tabla de verdad de la Unidad de control

El valor de las señales de control está determinado por el código de operación de la instrucción en curso:

Instrucción	RegDst	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch	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



Tabla de verdad de la Unidad de control (III)

Entradas						Salidas								
						RegDst	ALUSrc	MemtoReg	RegWrite	MemRead	MemWrite	Branch	ALUOp1	ALUOp0
CO ₅	CO ₄	CO ₃	CO ₂	CO ₁	CO ₀									
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!**



Ejecución de instrucciones

A continuación se muestra el flujo a través del camino de datos de las siguientes instrucciones:

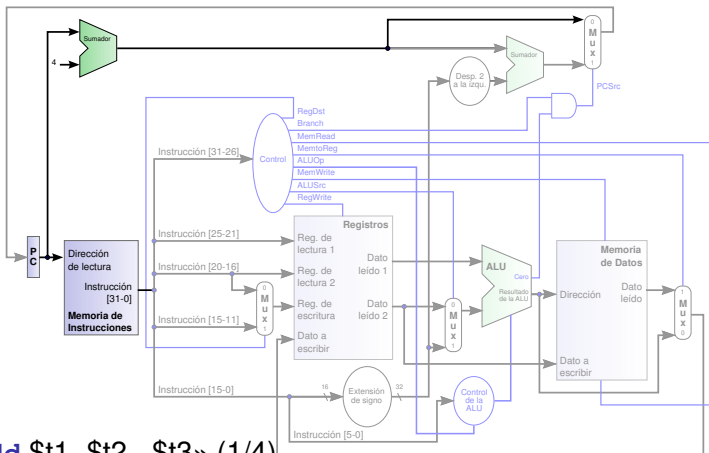
- «**add** \$t1, \$t2, \$t3»
- «**lw** \$t1, desp1(\$t2)»
- «**beq** \$t1, \$t2, desp»

En el camino de datos se resaltarán:

- Los elementos activos.
 - Las señales de control activadas.
- (No se resaltan las señales de control que estén a cero.)



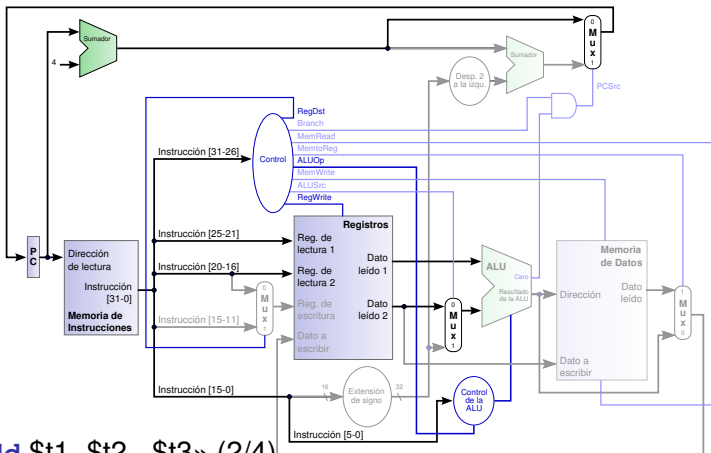
Ejecución de instrucciones (II)



«**add** \$t1, \$t2, \$t3» (1/4)



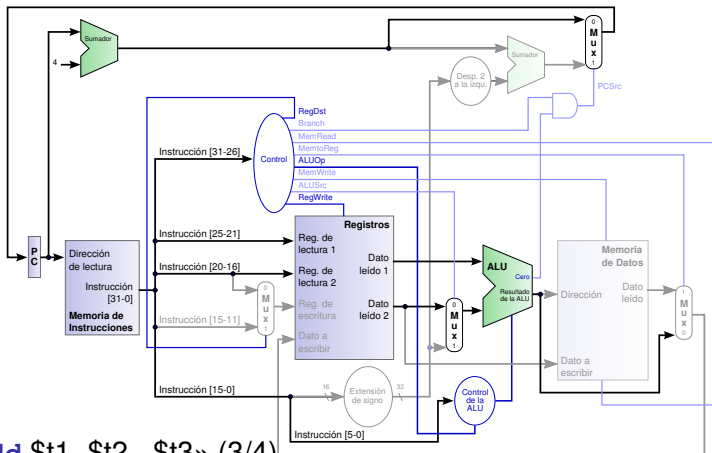
Ejecución de instrucciones (III)



«add \$t1, \$t2, \$t3» (2/4)



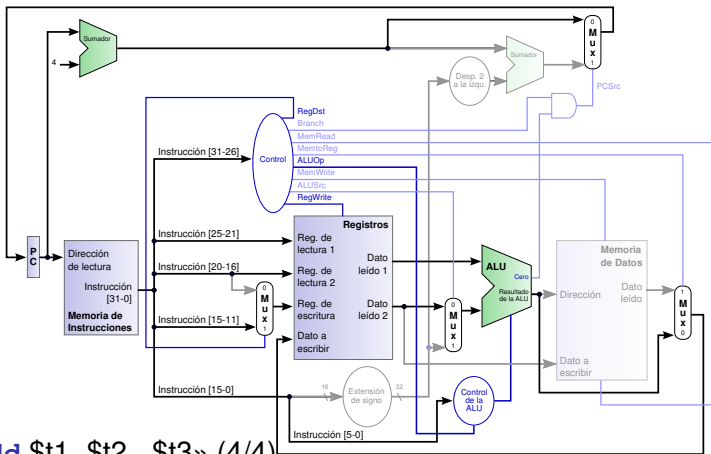
Ejecución de instrucciones (IV)



«add \$t1, \$t2, \$t3» (3/4)



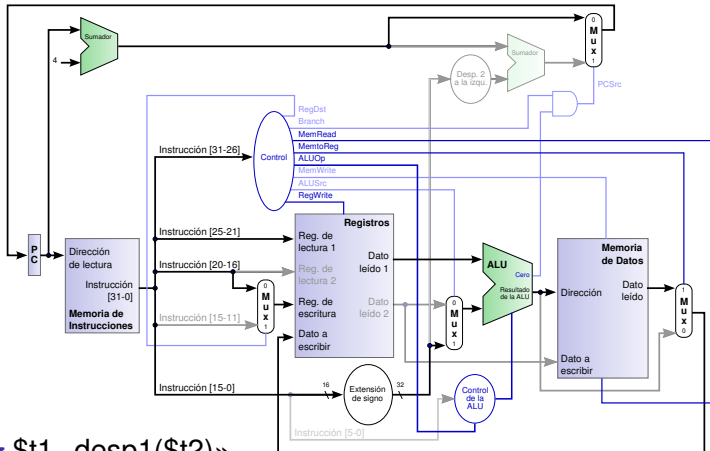
Ejecución de instrucciones (v)



«**add** \$t1, \$t2, \$t3» (4/4)



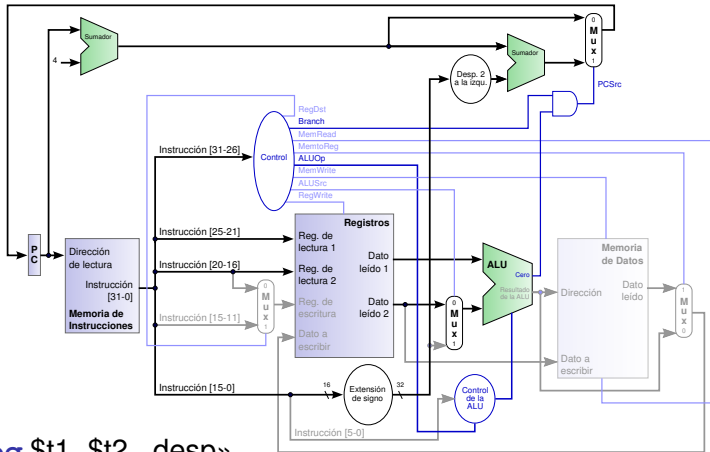
Ejecución de instrucciones (VI)



«lw \$t1, desp1(\$t2)»



Ejecución de instrucciones (VII)



«**beq** \$t1, \$t2, desp»



Añadir una nueva instrucción

Ejercicio 5: Amplía el camino 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: (00₂).



Añadir una nueva instrucción (II)

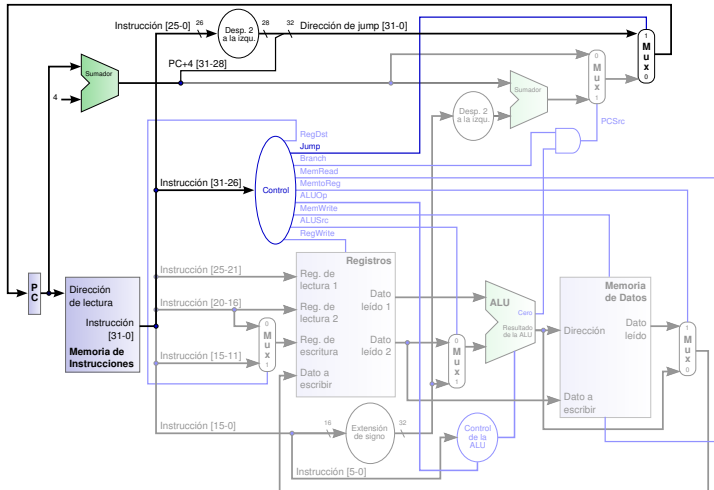
Solución:

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, Jump, que:
 - se active cuando la instrucción sea una «j» (CO=2)
 - y controle el anterior multiplexor.



Añadir una nueva instrucción (III)



Desventajas de la realización monociclo

Es posible realizar un procesador utilizando esta técnica.

Sin embargo, no se utiliza 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 solo pueden utilizarse una vez por ciclo → encarece el coste de la realización.

Realizaciones alternativas:

- Camino de datos Multiciclo.
- Camino de datos Segmentado.

