



Tema 5:

# Diseño monociclo del procesador

Fundamentos de computadores II

**José Manuel Mendías Cuadros**

*Dpto. Arquitectura de Computadores y Automática  
Universidad Complutense de Madrid*





# Contenidos



- ✓ Introducción.
- ✓ RISC-V de arquitectura reducida.
- ✓ Diseño de la ruta de datos.
- ✓ Diseño del controlador.
  
- ✓ Apéndice tecnológico.

Transparencias basadas en los libros:

- S.L. Harris and D. Harris. *Digital Design and Computer Architecture. RISC-V Edition.*
- D.A. Patterson and J.L. Hennessy. *Computer Organization and Design. RISC-V Edition.*



# Introducción

- El método más fiable para conocer el rendimiento de un computador es **midiendo el tiempo que tarda en ejecutar programas**.
  - El computador que los ejecute **más rápido** tendrá **mayor rendimiento**.
- Para una arquitectura dada, el tiempo de ejecución de un programa depende principalmente del:
  - **Número de instrucciones** que tenga el programa. 
  - **Número de ciclos** que tarde cada instrucción. 
  - **Tiempo de ciclo** (frecuencia de reloj).
- Comúnmente **número de ciclos y tiempo de ciclo tienen son inversos**:
  - **Procesador monociclo**: 1 ciclo por instrucción, tiempo de ciclo largo.
  - **Procesador multiciclo**: Varios ciclos por instrucción, tiempo de ciclo corto.



# Introducción

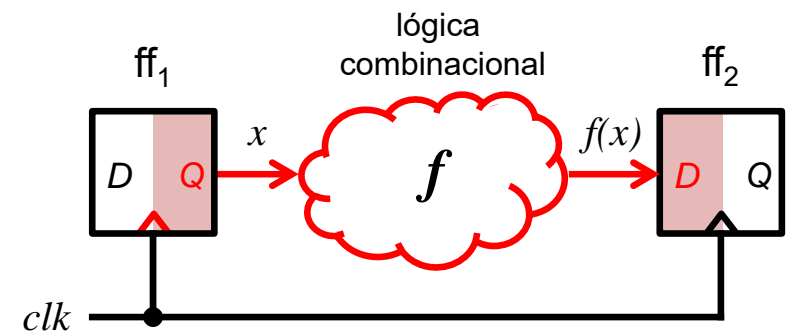
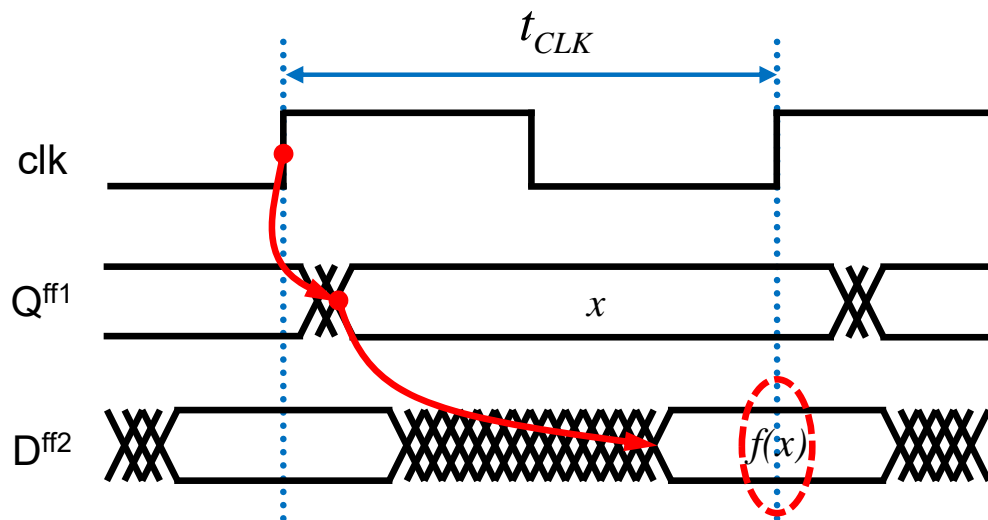
- Para **diseñar un procesador** se usan **técnicas de diseño algorítmico** en donde la **especificación** del circuito es su **arquitectura**.
- El procesador estará formado por 2 elementos:
  - **Ruta de datos**: realiza operaciones y almacena resultados.
    - Al menos, deberá incluir tantos **elementos de almacenamiento** como estén **definidos en la arquitectura** (sean visibles por el programador).
    - Deberá incluir los **elementos funcionales** que sean necesarios para **realizar todas las operaciones** del repertorio de instrucciones.
  - **Controlador**: secuencia la realización de las transferencias entre registros definidas para cada instrucción del repertorio.
- Según la **estrategia de diseño** elegida tendremos:
  - **Procesadores monociclo**: **todas las transferencias** de entre registros implicadas en una instrucción se realizan **en un único ciclo de reloj**.
  - **Procesadores multiciclo**: **las transferencias** entre registros implicadas en una instrucción **se reparten entre varios ciclos de reloj** consecutivos.



# Introducción



- Los procesadores se diseñarán según el **modelo de temporización síncrona por flanco** (de subida) de reloj global:
  - El reloj llega a todos los biestables del sistema y todos ellos **cambian de estado simultáneamente en el flanco (de subida) del reloj**.
  - Los nuevos **valores se propagan a través de las redes combinatoriales** hasta estabilizarse en las entradas de los biestables.
  - Se repite el proceso indefinidamente en cada ciclo de reloj.
  - El **tiempo de ciclo del reloj debe ser lo suficientemente largo** para que todos los sistemas combinatoriales alcancen su **régimen permanente**.





# RISC-V de arquitectura reducida

## Repertorio de instrucciones (i)

- Se diseñará una **microarquitectura** capaz de ejecutar un **subconjunto** del **repertorio de instrucciones del RISC-V32** que opere únicamente con datos de **32 bits**.
- **Instrucciones con acceso a memoria**

<code>lw rd, imm<sub>12b</sub>(rs1)</code>	$rd \leftarrow \text{Mem}[rs1 + \text{sExt}(\text{imm})]$	tipo-I
--	---	--------

<code>sw rs2, imm<sub>12b</sub>(rs1)</code>	$\text{Mem}[rs1 + \text{sExt}(\text{imm}_{12b})] \leftarrow rs2$	tipo-S
---	--	--------

- **Aritmético-lógicas con ambos operandos en registros**

<code>add rd, rs1, rs2</code>	$rd \leftarrow rs1 + rs2$	tipo-R
-------------------------------	---------------------------	--------

<code>sub rd, rs1, rs2</code>	$rd \leftarrow rs1 - rs2$	tipo-R
-------------------------------	---------------------------	--------

<code>and rd, rs1, rs2</code>	$rd \leftarrow rs1 \& rs2$	tipo-R
-------------------------------	----------------------------	--------

<code>or rd, rs1, rs2</code>	$rd \leftarrow rs1   rs2$	tipo-R
------------------------------	---------------------------	--------

<code>slt rd, rs1, rs2</code>	$rd \leftarrow \text{if}(rs1 <_s rs2) \text{ then } (1) \text{ else } (0)$	tipo-R
-------------------------------	--	--------

# RISC-V de arquitectura reducida

## Repertorio de instrucciones (ii)



### ■ Aritmético-lógicas con un operando inmediato

<code>addi rd, rs1, imm<sub>12b</sub></code>	$rd \leftarrow rs1 + sExt(imm)$ ,	tipo-I
<code>andi rd, rs1, imm<sub>12b</sub></code>	$rd \leftarrow rs1 \& sExt(imm)$	tipo-I
<code>ori rd, rs1, imm<sub>12b</sub></code>	$rd \leftarrow rs1   sExt(imm)$	tipo-I
<code>slti rd, rs1, imm<sub>12b</sub></code>	$rd \leftarrow if ( rs1 <_s sExt(imm) ) then ( 1 ) else ( 0 )$	tipo-I

### ■ Instrucciones de salto condicional

<code>beq rs1, rs2, imm<sub>13b</sub></code>	$PC \leftarrow if ( rs1 = rs2 )$ $then ( PC + sExt(imm_{12:1} \ll 1 ) ) else ( PC+4 )$	tipo-B
--	---	--------

### ■ Instrucciones de salto a función

<code>jal rd, imm<sub>21b</sub></code>	$PC \leftarrow PC + sExt(imm_{20:1} \ll 1), rd \leftarrow PC+4$	tipo-J
--	---	--------



# RISC-V de arquitectura reducida

## Formato de instrucción

- Los formatos de instrucción y codificación de campos serán los **mismos** que en la arquitectura completa.

31	25 24	20 19	15 14	12 11	7 6	0	
funct7	rs2	rs1	funct3	rd	op		<i>tipo-R</i>
imm <sub>11:0</sub>		rs1	funct3	rd	op		<i>tipo-I</i>
imm <sub>11:5</sub>	rs2	rs1	funct3	imm <sub>4:0</sub>	op		<i>tipo-S</i>
imm <sub>12,10:5</sub>	rs2	rs1	funct3	imm <sub>4:1,11</sub>	op		<i>tipo-B</i>
imm <sub>20,10:1,11,19:12</sub>				rd	op		<i>tipo-J</i>



# RISC-V de arquitectura reducida

## Formato de instrucción: codificación de campos



Instrucción	Tipo	funct7 bits 31:25	funct3 bits 14:12	op bits 6:0
lw	I	–	010	0000011
sw	S	–	010	0100011
add	R	0000000	000	0110011
sub	R	0100000	000	
slt	R	0000000	010	
or	R	0000000	110	
and	R	0000000	111	
addi	I	–	000	0010011
slti	I	–	010	
ori	I	–	110	
andi	I	–	111	
beq	B	–	000	1100011
jal	J	–	–	1101111



# Diseño de la ruta de datos

## Elementos de almacenamiento (i)

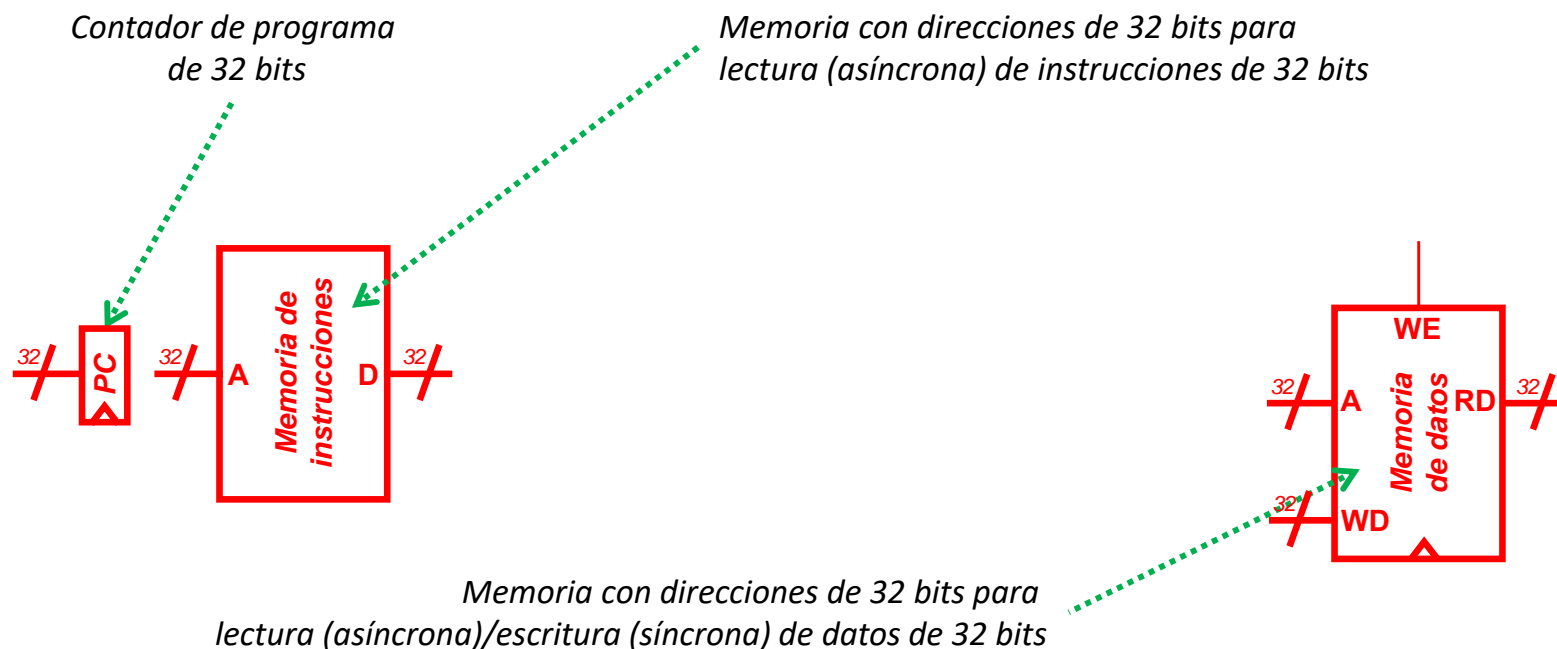
- Los **elementos de almacenamiento** serán los mismos que son visibles al programador: **Memoria**, **PC** y **32 registros** de propósito general.
- La **Memoria** tendrá un **comportamiento idealizado**:
  - Integrada en el procesador.
  - Direccionable por bytes, pero capaz de aceptar/ofrecer 4 bytes por acceso.
  - Tiempo de acceso inferior al tiempo de ciclo del procesador.
  - Dividida en dos porque las **instrucciones deben leerse** de memoria en el mismo **ciclo de reloj** en que se **leen/escriben los datos**.
    - **Memoria de instrucciones**: se comportará como una ROM combinacional.
    - **Memoria de datos**: se comportará como un gran Banco de Registros con doble puerto de datos para entrada y salida separada de los mismos.
- El **Contador de Programa** será un **array de biestables D**:
  - Al ejecutarse las instrucciones en un único ciclo, el PC **debe actualizarse en todos los ciclos** y no requiere tener una señal que controle su carga.



# Diseño de la ruta de datos

## Elementos de almacenamiento (ii)

- Los **elementos de almacenamiento** serán los mismos que son visibles al programador: **Memoria**, **PC** y **32 registros** de propósito general.

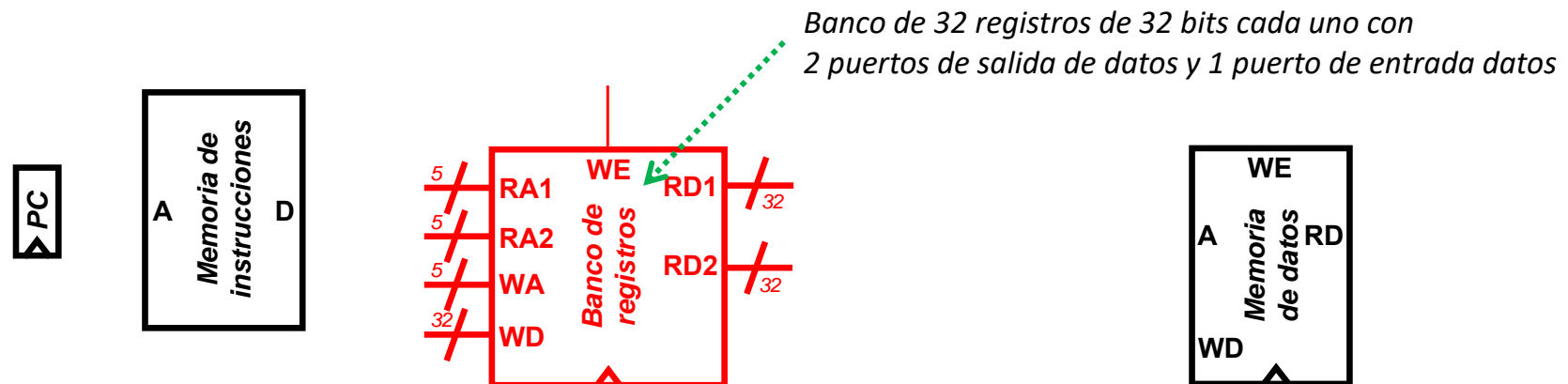




# Diseño de la ruta de datos

## Elementos de almacenamiento (iii)

- Los **elementos de almacenamiento** serán los mismos que son visibles al programador: **Memoria**, **PC** y **32 registros** de propósito general.



- Los 32 registros se organizan en un **Banco con 3 puertos**:
  - En un procesador monociclo, las instrucciones tipo-R en el **mismo ciclo de reloj** leen del Banco **2 registros** y escriben **1 registro**.

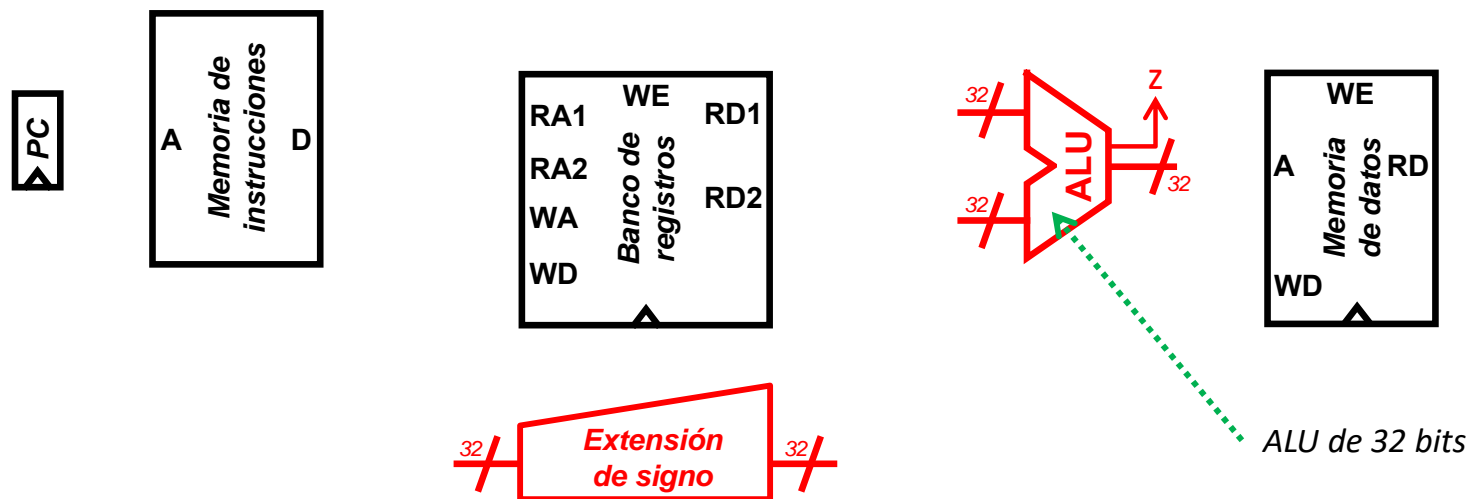
```
add rd, rs1, rs2    BR[ rd ] ← BR[ rs1 ] + BR[ rs2 ], PC ← PC+4
```



# Diseño de la ruta de datos

## Elementos funcionales (i)

- Dispondrá de una ALU y un Extensor de Signo combinacionales:
  - La ALU realizará **todas las operaciones aritmético-lógicas** del repertorio.
    - Con un **flag Z** para **realizar mediante resta la comparación** de igualdad en instrucciones `beq`
  - El **Extensor de Signo** construirá el **operando inmediato de 32 bits** a partir de los campos `imm` (de menor anchura) de las instrucciones.

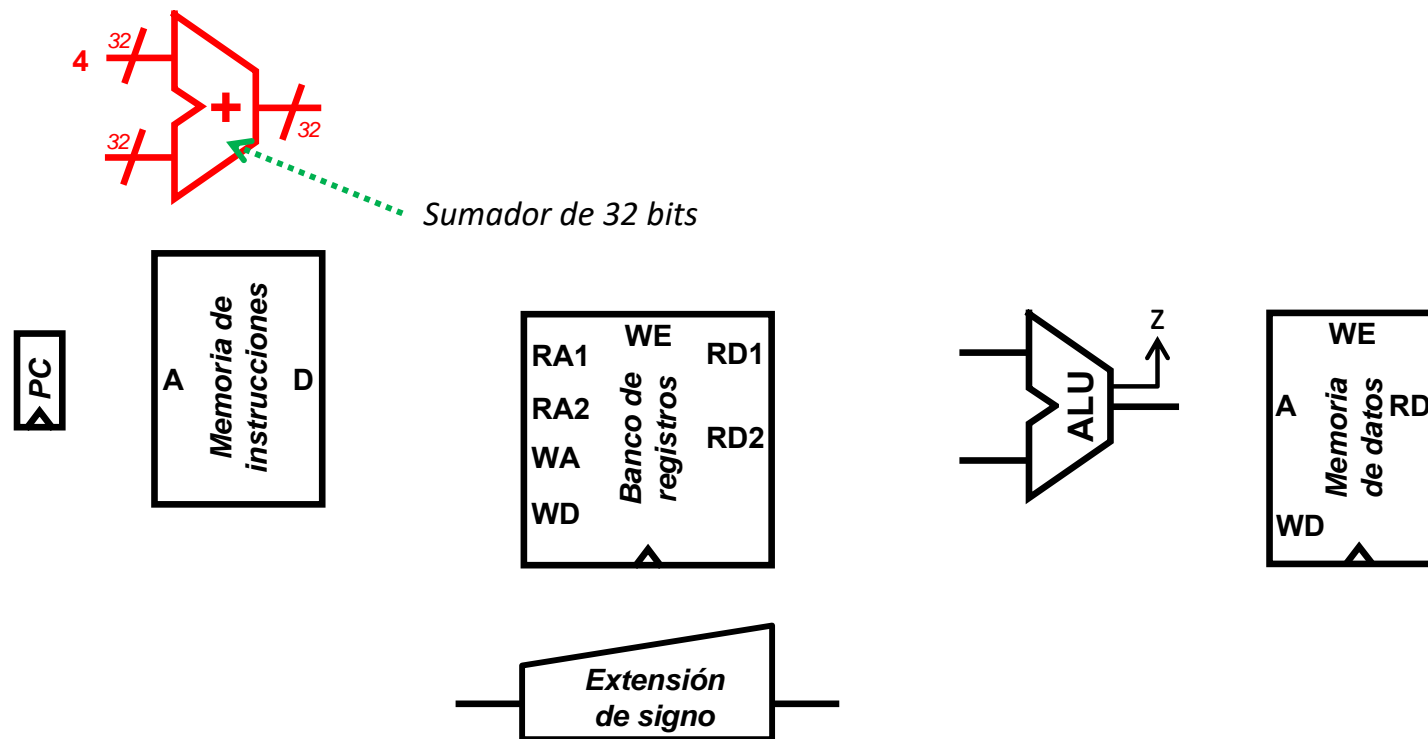




# Diseño de la ruta de datos

## Elementos funcionales (ii)

- Dispondrá de un **sumador adicional** para **incrementar el PC**
  - En un procesador monociclo, en el **mismo ciclo de reloj** en que **opera la ALU** se **actualiza el PC** con la dirección de la instrucción siguiente.



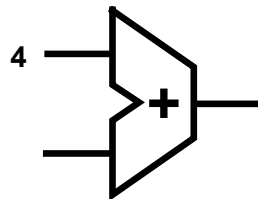
- Siempre **suma 4** porque la Memoria es direccionable por bytes y las instrucciones son de **32 bits** (4 bytes).



# Diseño de la ruta de datos

## Elementos funcionales (iii)

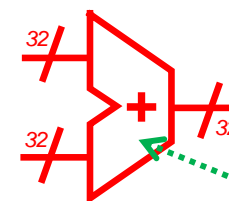
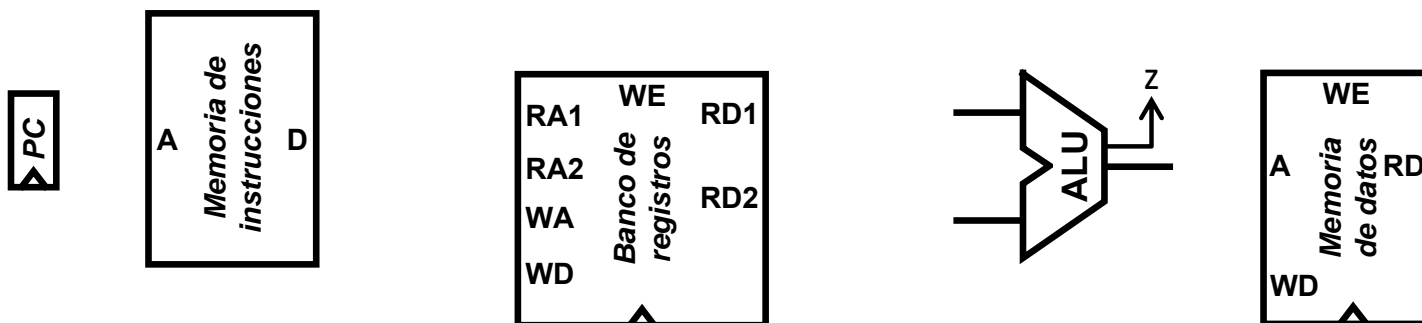
- Dispondrá de otro **sumador adicional** para **calcular direcciones de salto**
  - En un procesador monociclo, las instrucciones tipo **beq** en **mismo ciclo de reloj** operan en la ALU, efectúan **PC+4** y **calculan la dirección de salto**.



`beq rs1,rs2,imm`

`PC ← if ( BR[ rs1 ] = BR[ rs2 ] )`

`then ( PC + sExt(imm) ) else ( PC+4 )`



Sumador de 32 bits

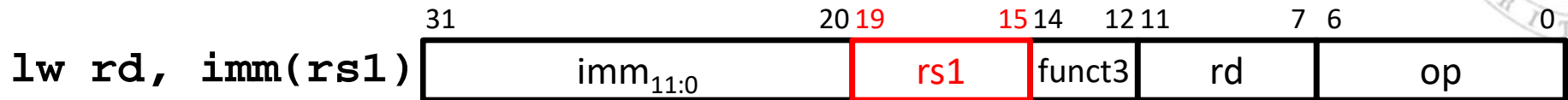




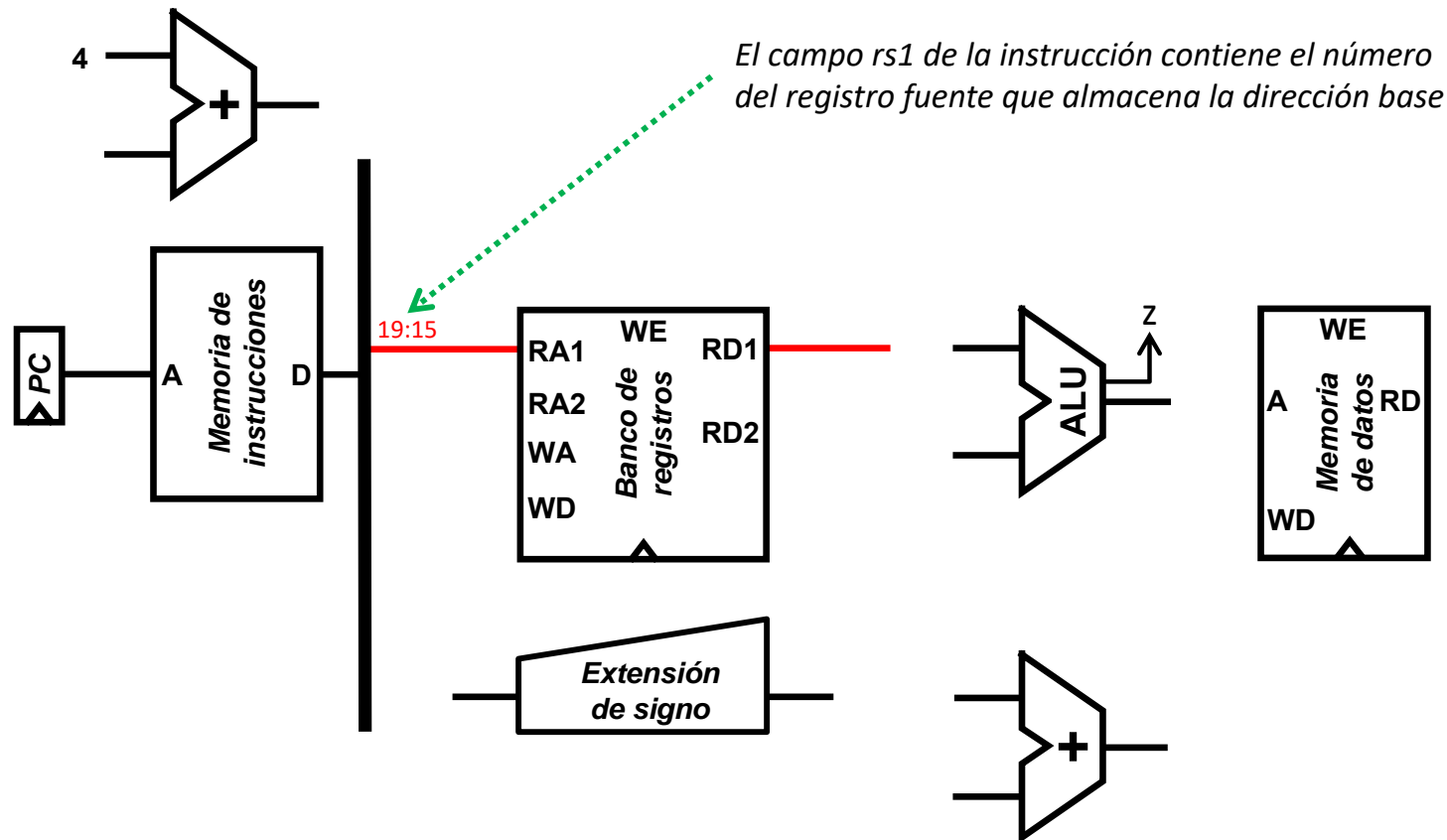


# Diseño de la ruta de datos

## Instrucción lw: lectura del registro base



$$BR[rd] \leftarrow Mem[BR[rs1] + sExt(imm)], PC \leftarrow PC+4$$



- La Memoria de Instrucciones y el Banco de Registros se conectan para leer la dirección base contenida en el registro rs1.

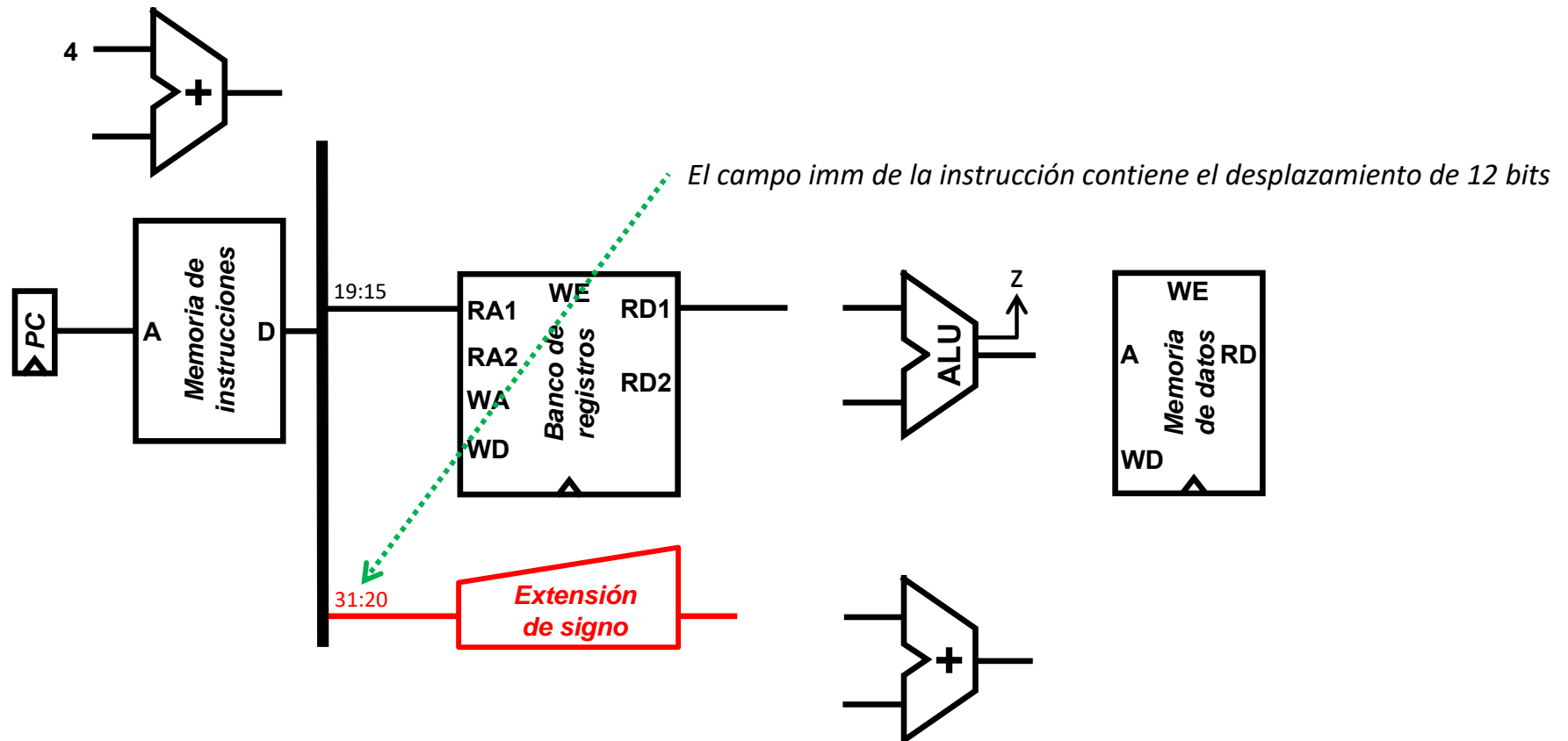


# Diseño de la ruta de datos

## Instrucción lw: cálculo del desplazamiento



$$BR[rd] \leftarrow Mem[BR[rs1] + sExt(imm)], PC \leftarrow PC+4$$

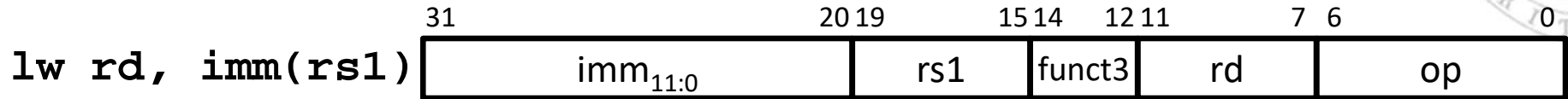


- La Memoria de Instrucciones y el Extensor de Signo se conectan para extender a 32 bits el desplazamiento de 12 bits indicado en la instrucción.

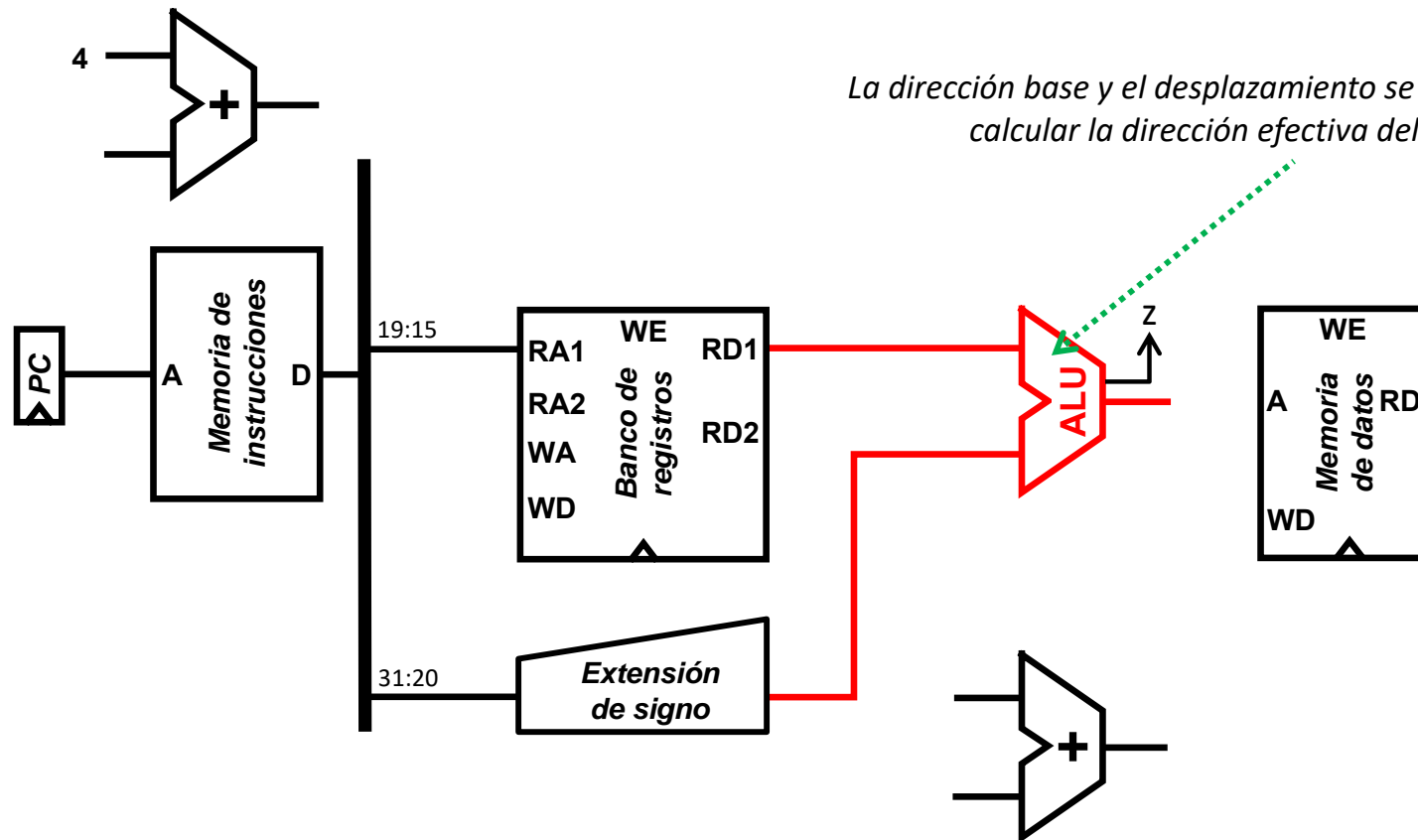


# Diseño de la ruta de datos

## Instrucción **lw**: cálculo de la dirección efectiva



$$BR[rd] \leftarrow Mem[BR[rs1] + sExt(imm)], PC \leftarrow PC+4$$



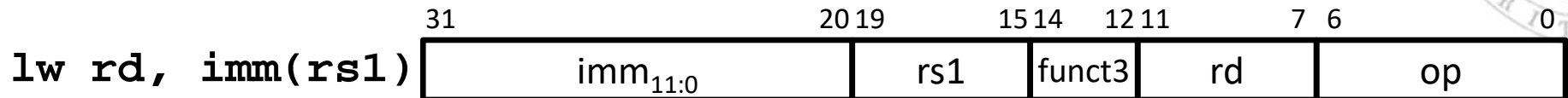
La dirección base y el desplazamiento se suman en la ALU para calcular la dirección efectiva del dato a cargar

- El Banco de Registros y la ALU se conectan para calcular la dirección efectiva

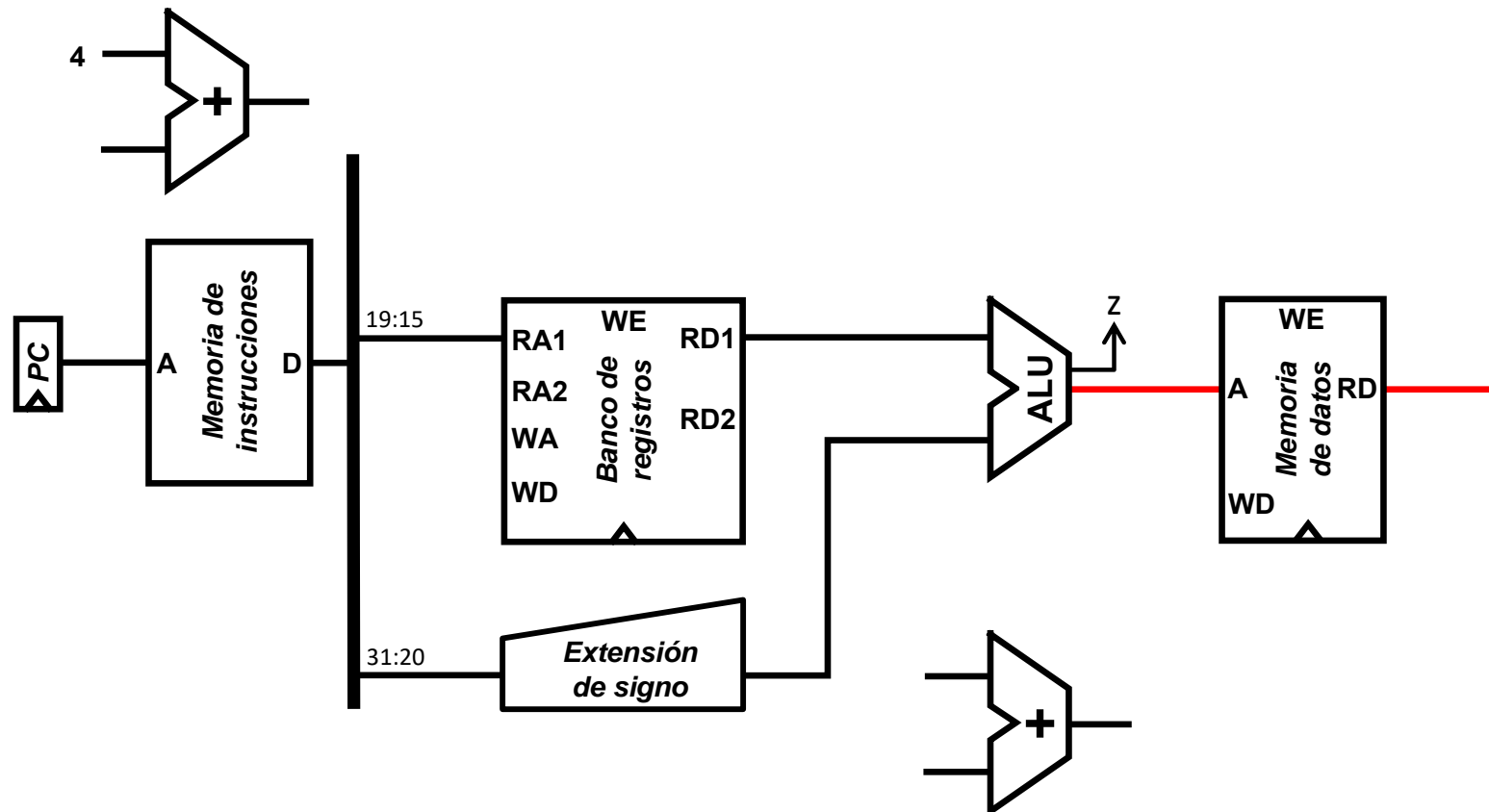


# Diseño de la ruta de datos

## Instrucción lw: lectura del operando



$$BR[rd] \leftarrow Mem[BR[rs1] + sExt(imm)], PC \leftarrow PC+4$$

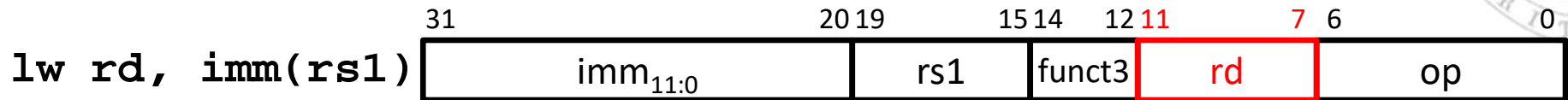


- La ALU y la Memoria de Datos se conectan para leer el dato a cargar

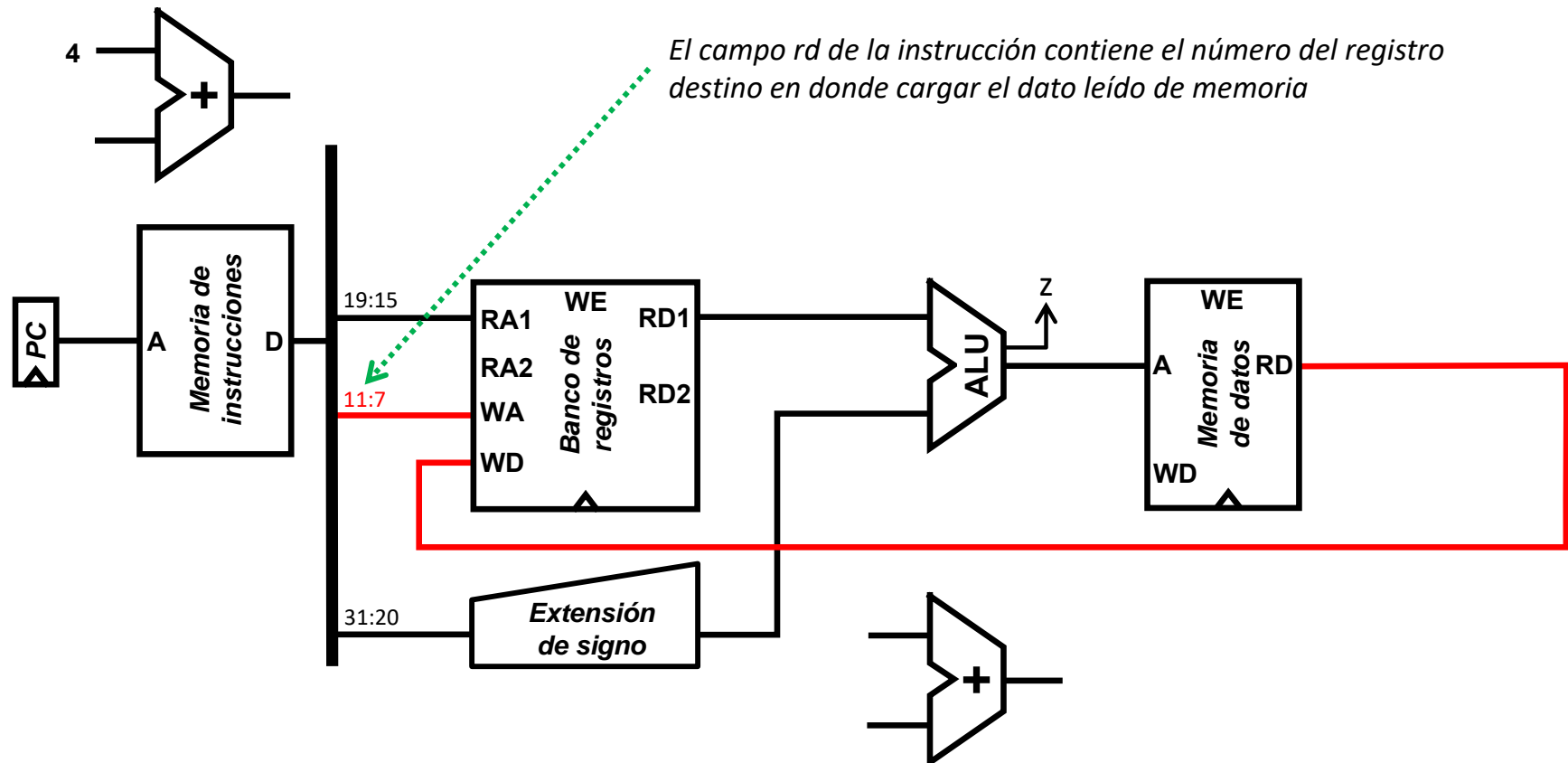


# Diseño de la ruta de datos

## Instrucción lw: lectura y carga del dato



$$BR[rd] \leftarrow Mem[BR[rs1] + sExt(imm)], PC \leftarrow PC+4$$

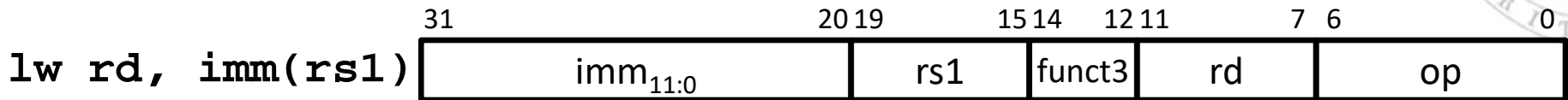


- La Memoria de Datos y el Banco de Registros se conectan para cargar en el registro rd el dato leído de memoria

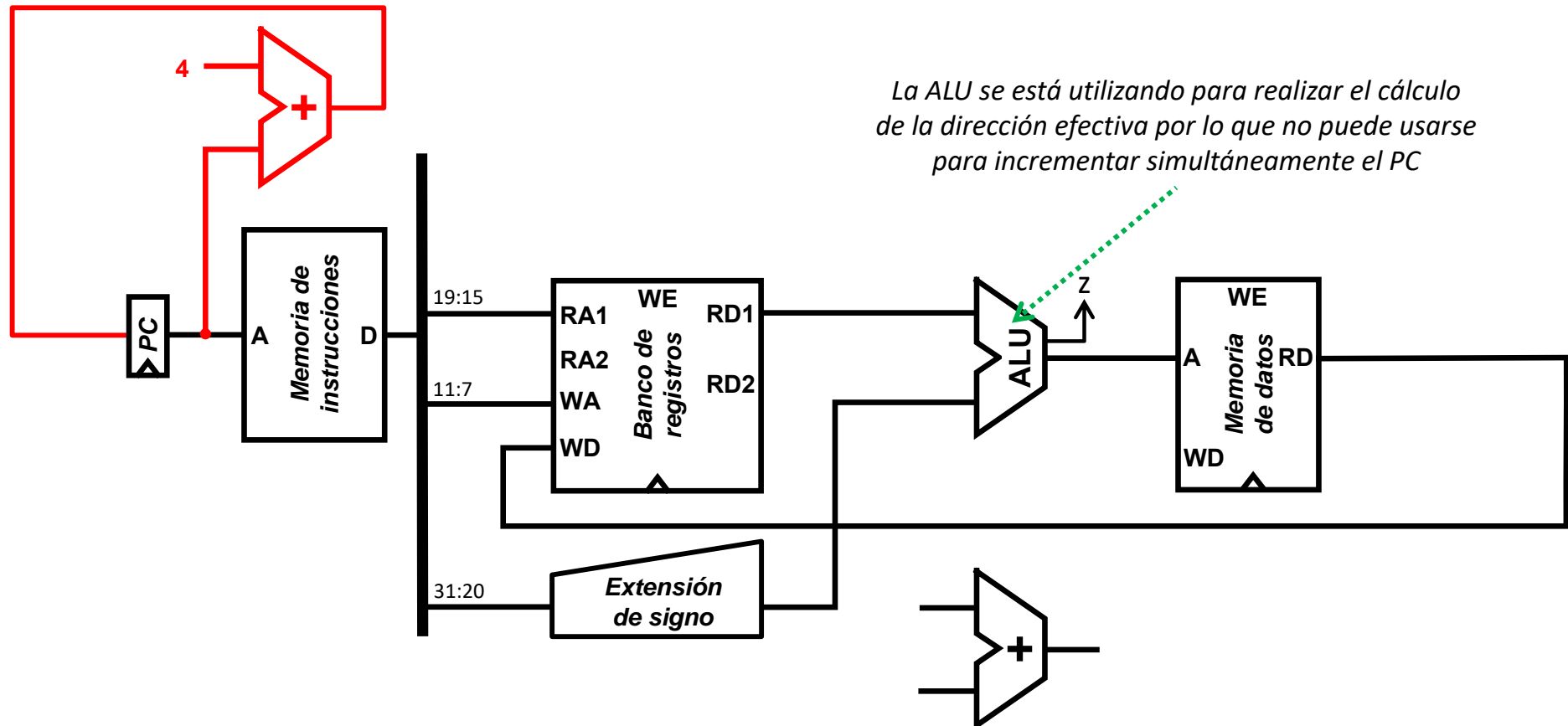


# Diseño de la ruta de datos

## Instrucción `lw`: incremento del PC



$$BR[rd] \leftarrow Mem[BR[rs1] + sExt(imm)], PC \leftarrow PC + 4$$



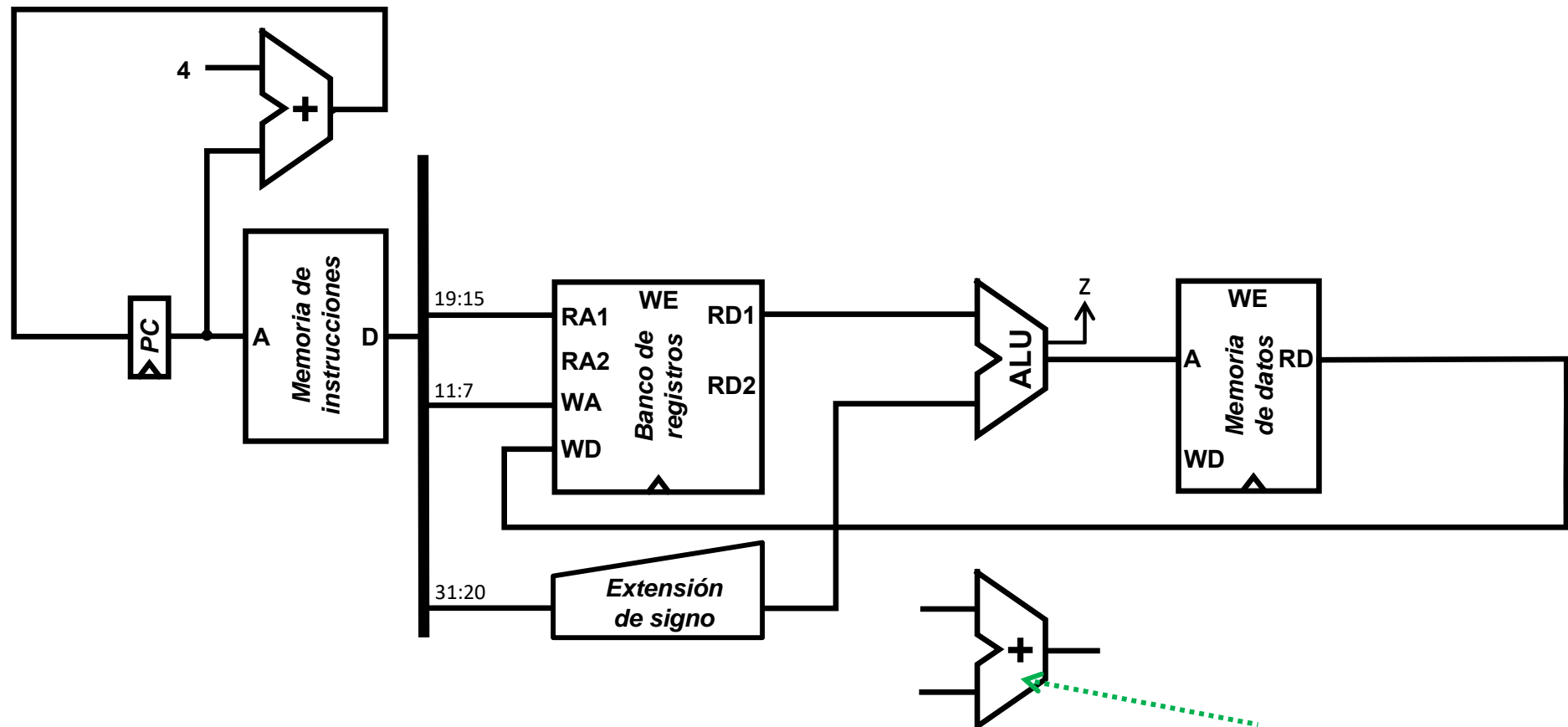
- El PC y el Incrementador se conectan para actualizar el PC con la dirección de la instrucción siguiente



# Diseño de la ruta de datos

## Ruta de datos para instrucciones **lw**

- Esta ruta de datos puede ejecutar **cualquier secuencia** de instrucciones **lw**, se continuará ampliando para poder ejecutar otras.



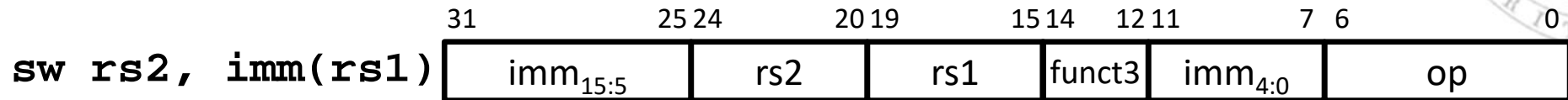
Este sumador solo se usa para calcular direcciones de salta en instrucciones **ja1/beq**



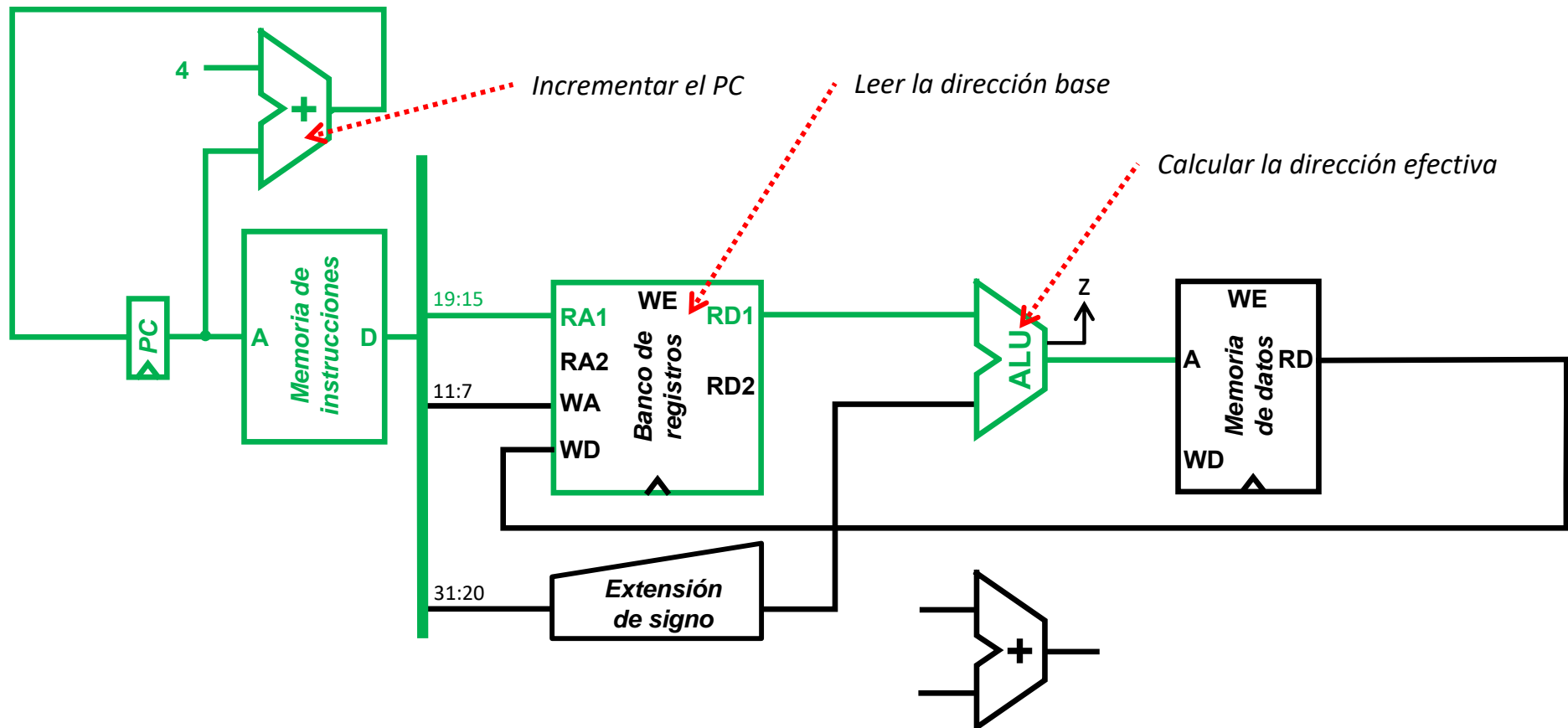
# Diseño de la ruta de datos

## Ruta de datos para instrucciones **sw**

versión 31/10/23



$$\text{Mem}[ \text{BR}[ \text{rs1} ] + \text{sExt}(\text{imm}) ] \leftarrow \text{BR}[ \text{rs2} ], \text{PC} \leftarrow \text{PC} + 4$$



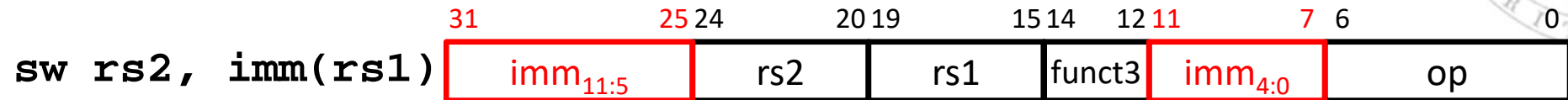
- Parte de esta ruta de datos **puede reutilizarse** para ejecutar la **instrucción sw**



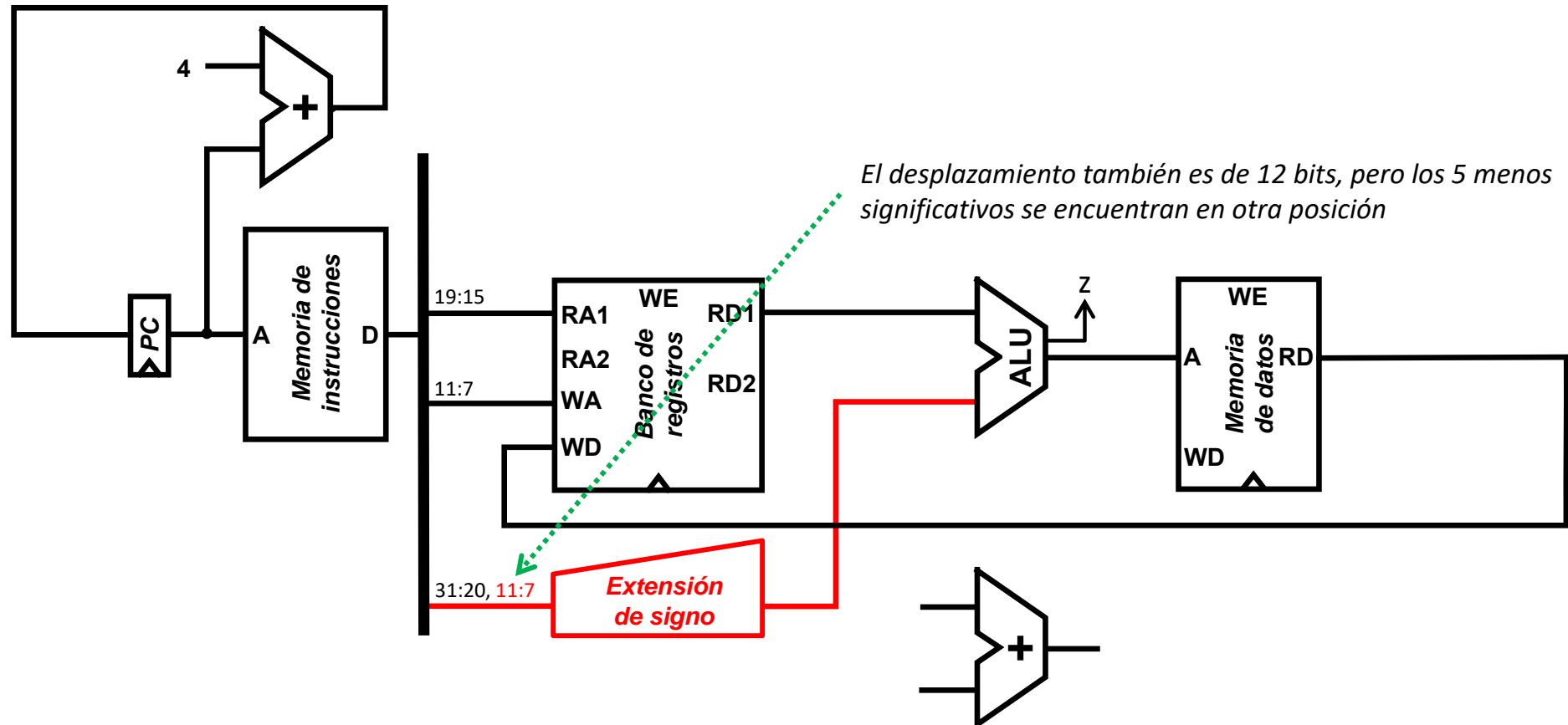


# Diseño de la ruta de datos

## Instrucción `sw`: cálculo de desplazamiento



$$\text{Mem}[ \text{BR}[ \text{rs1} ] + \text{sExt}(\text{imm}) ] \leftarrow \text{BR}[ \text{rs2} ], \text{PC} \leftarrow \text{PC} + 4$$



- La conexión entre la Memoria de Instrucciones y el Extensor de Signo se amplía porque el desplazamiento se encuentra en otra posición de la instrucción.



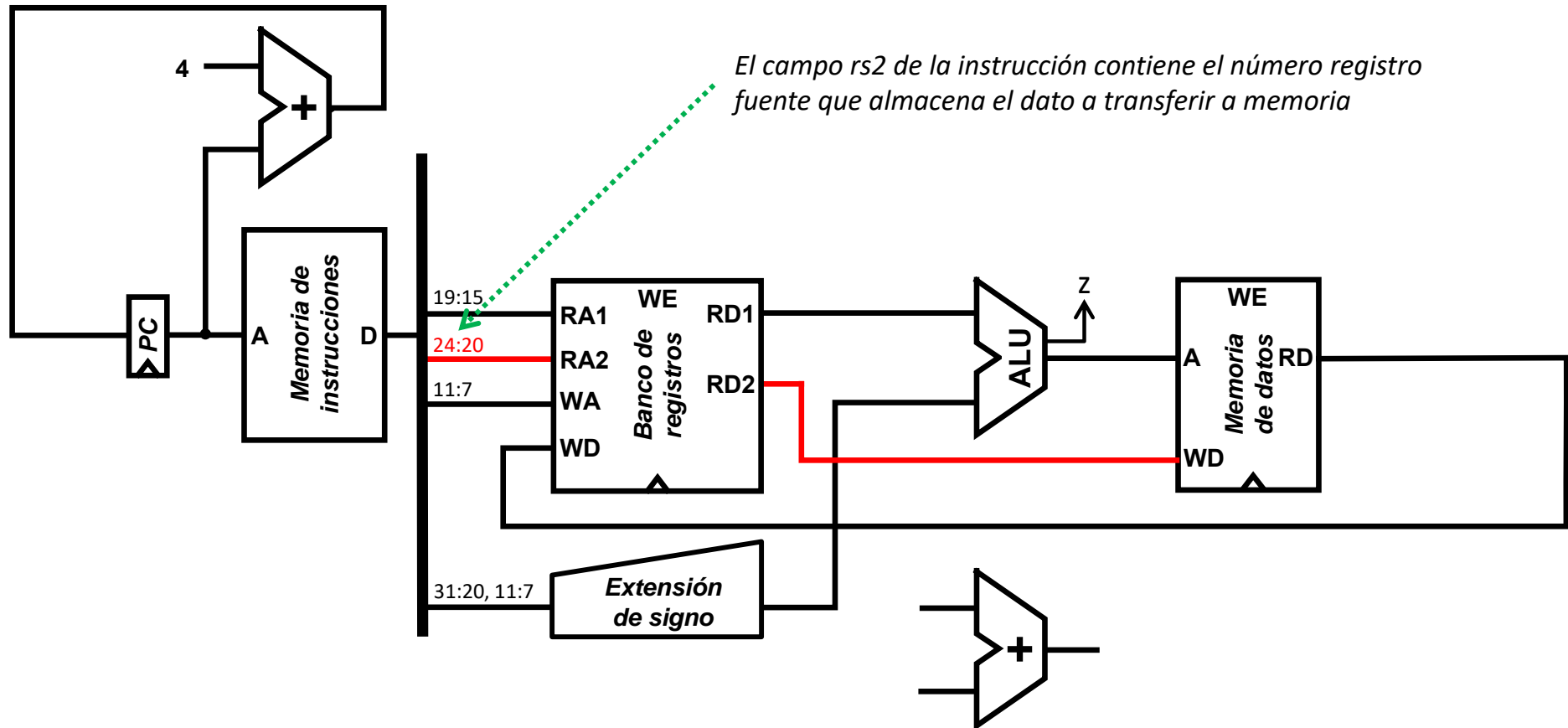
# Diseño de la ruta de datos

## Instrucción **sw**: lectura y almacenaje del dato



$$\text{Mem}[\text{BR}[\text{rs1}] + \text{sExt}(\text{imm})] \leftarrow \text{BR}[\text{rs2}], \text{PC} \leftarrow \text{PC} + 4$$

El campo rs2 de la instrucción contiene el número registro fuente que almacena el dato a transferir a memoria



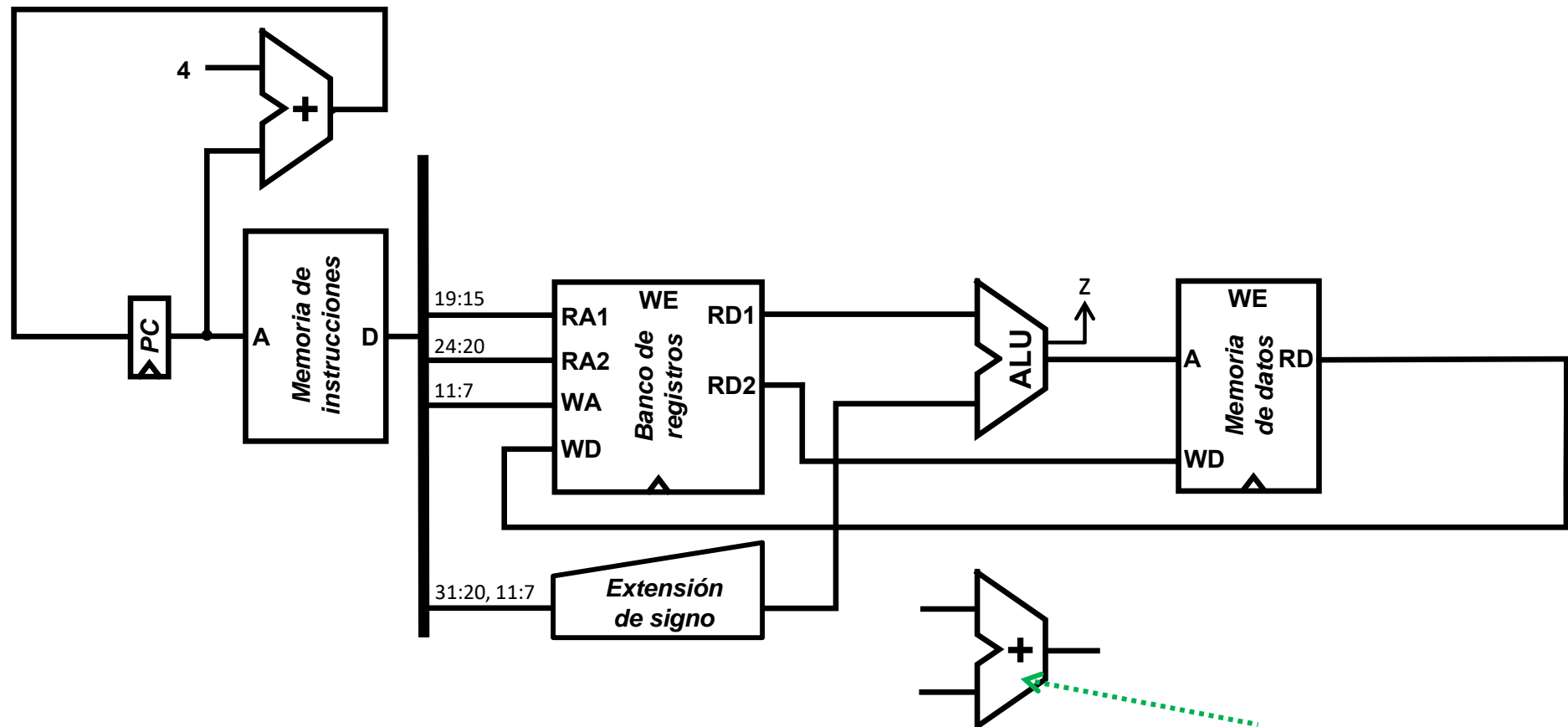
- El Banco de Registros y la Memoria de Datos se conectan para salvar en memoria el dato contenido en el registro rs2



# Diseño de la ruta de datos

## Ruta de datos para instrucciones **lw/sw**

- Esta ruta de datos puede ejecutar **cualquier secuencia** de instrucciones **lw y/o sw**, la seguiremos ampliando.



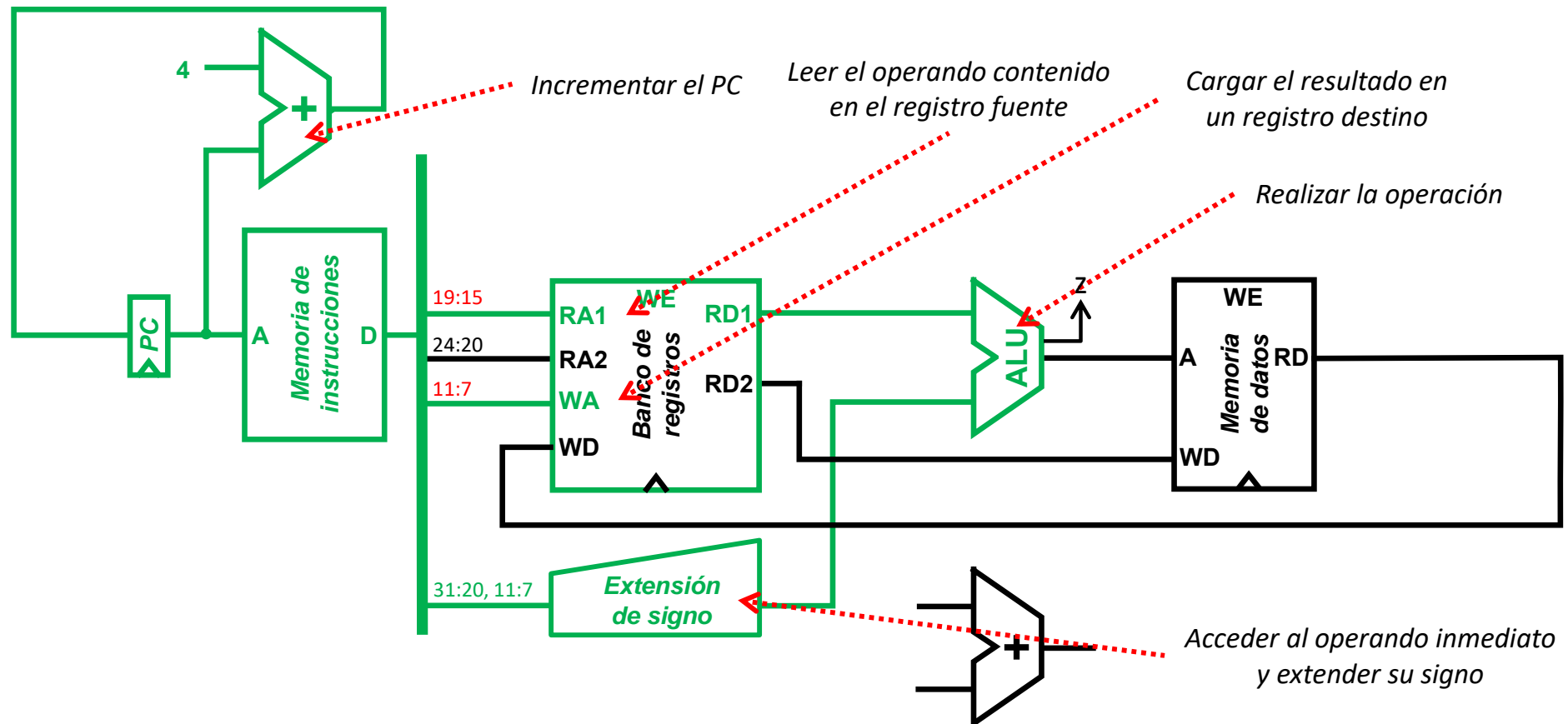


# Diseño de la ruta de datos

## Ruta de datos para instrucciones tipo **addi**



$$BR[rd] \leftarrow BR[rs1] \text{ op } sExt(imm), PC \leftarrow PC+4$$

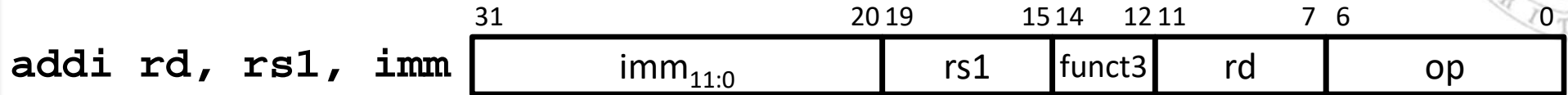


- Parte de esta ruta de datos **puede reutilizarse** para ejecutar las **instrucciones addi/andi/ori/slti**



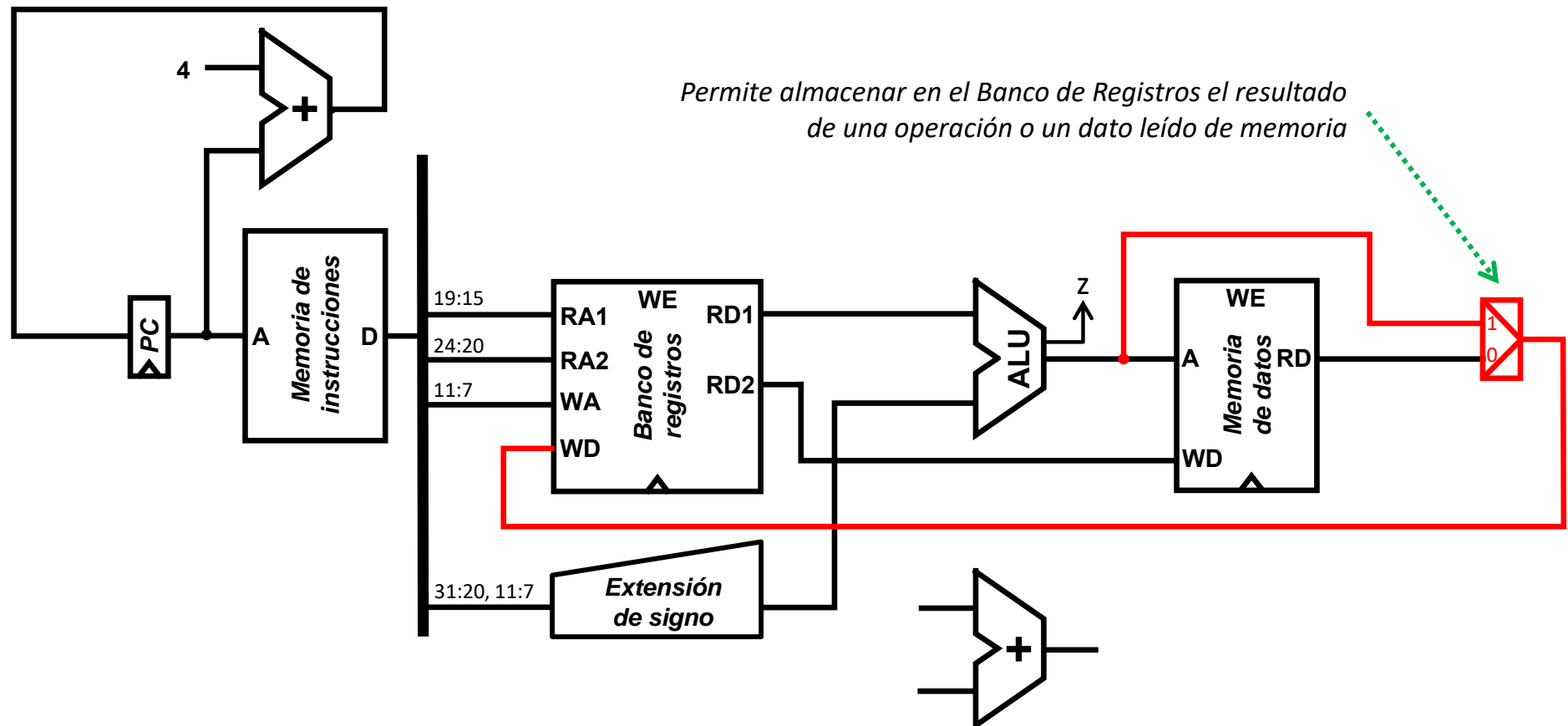
# Diseño de la ruta de datos

## Instrucciones tipo **addi**: almacenaje del resultado



$$BR[rd] \leftarrow BR[rs1] \text{ op } sExt(imm), PC \leftarrow PC+4$$

Permite almacenar en el Banco de Registros el resultado de una operación o un dato leído de memoria



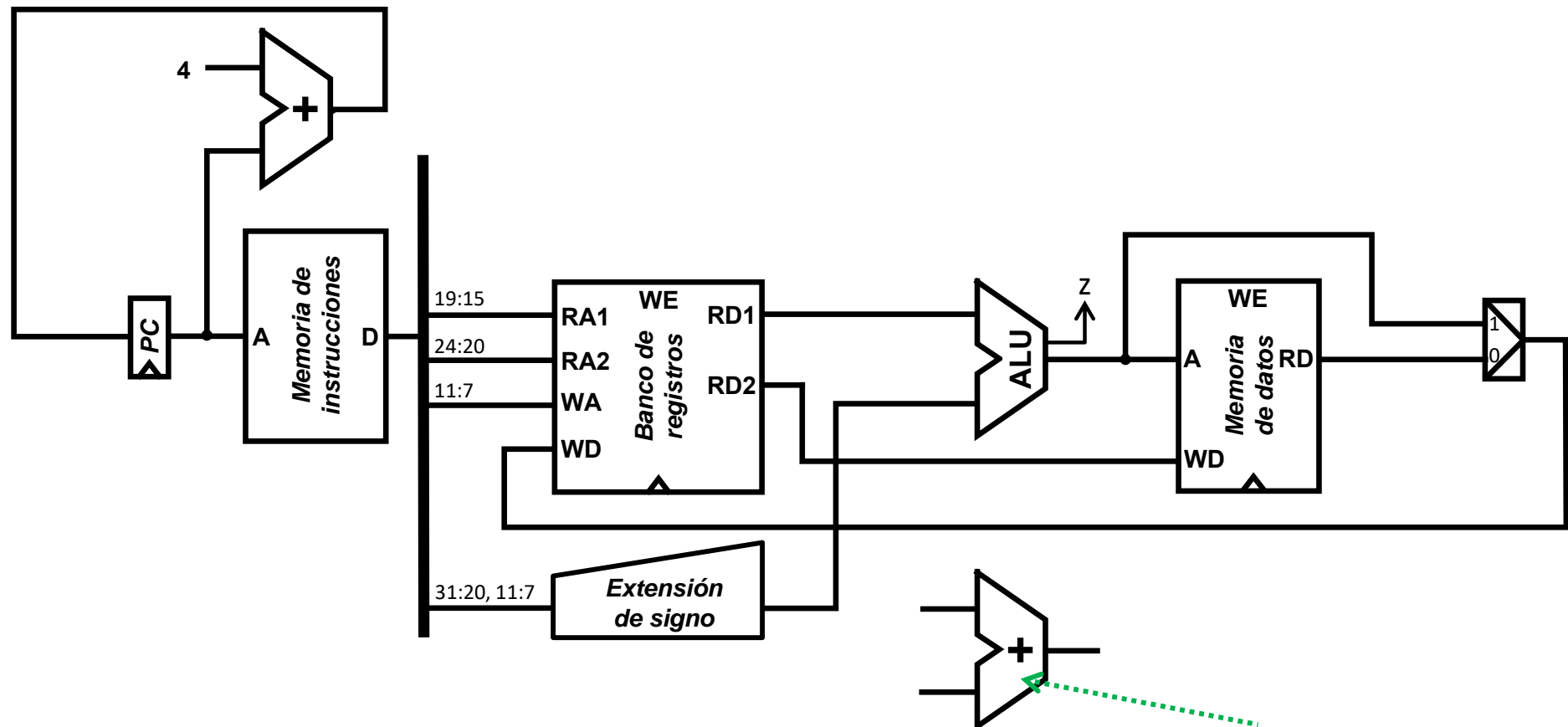
- Se **añade un multiplexor** para poder almacenar en el registro rd el resultado de la operación efectuada en la ALU



# Diseño de la ruta de datos

## Ruta de datos para instrucciones `lw/sw/addi`

- Esta ruta de datos puede ejecutar cualquier secuencia de instrucciones `lw`, `sw`, y/o aritmético-lógica con operando inmediato.



Este sumador solo se usa para calcular direcciones de salta en instrucciones `jal/beq`

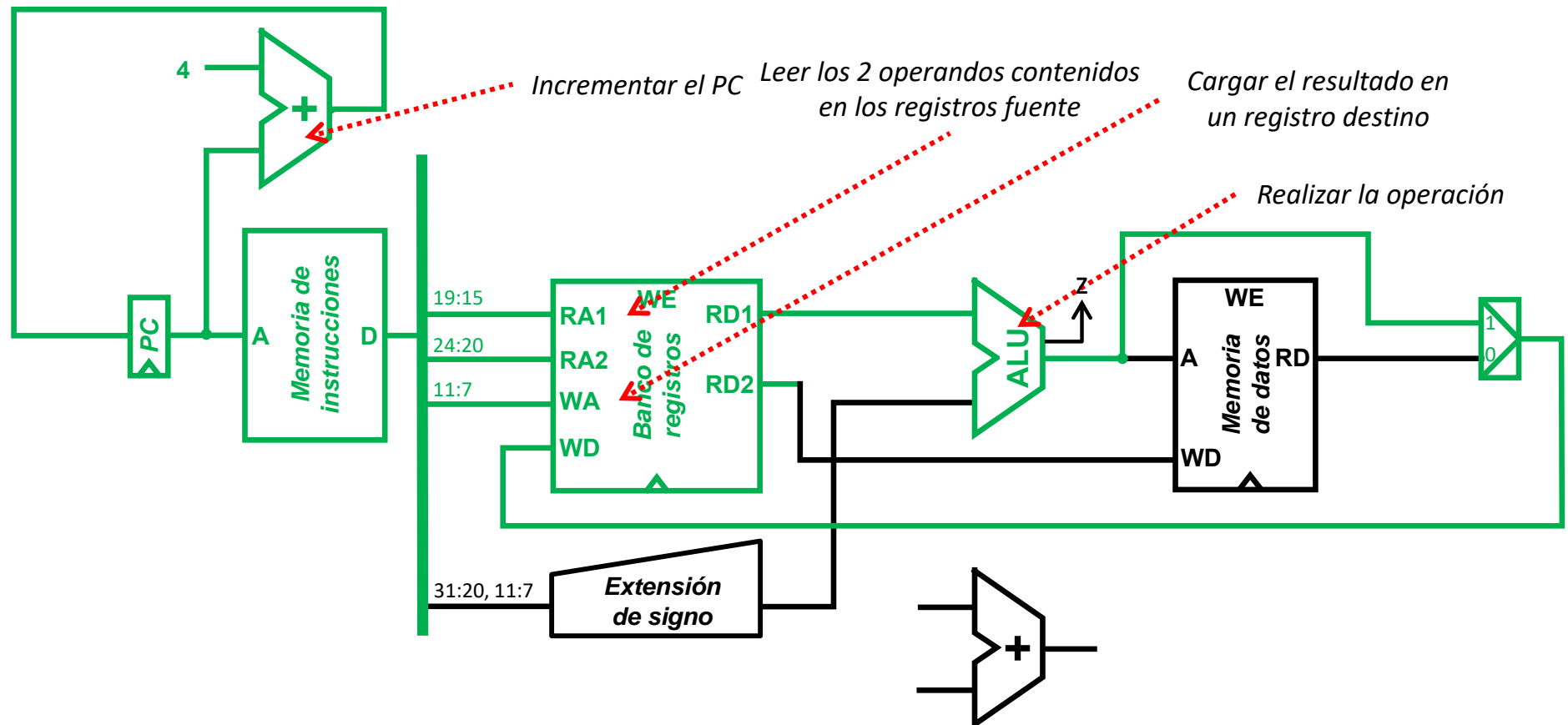


# Diseño de la ruta de datos

## Ruta de datos para instrucciones tipo **add**



$$BR[rd] \leftarrow BR[rs1] \text{ op } BR[rs2], PC \leftarrow PC+4$$

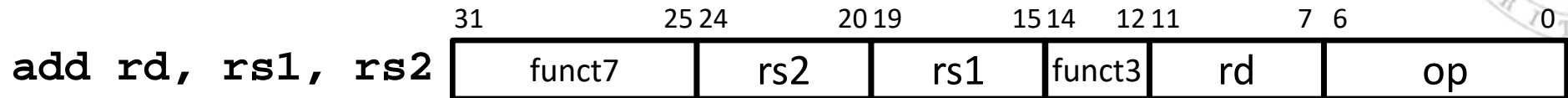


- Parte de esta ruta de datos **puede reutilizarse** para ejecutar las **instrucciones add/sub/and/or/slt**

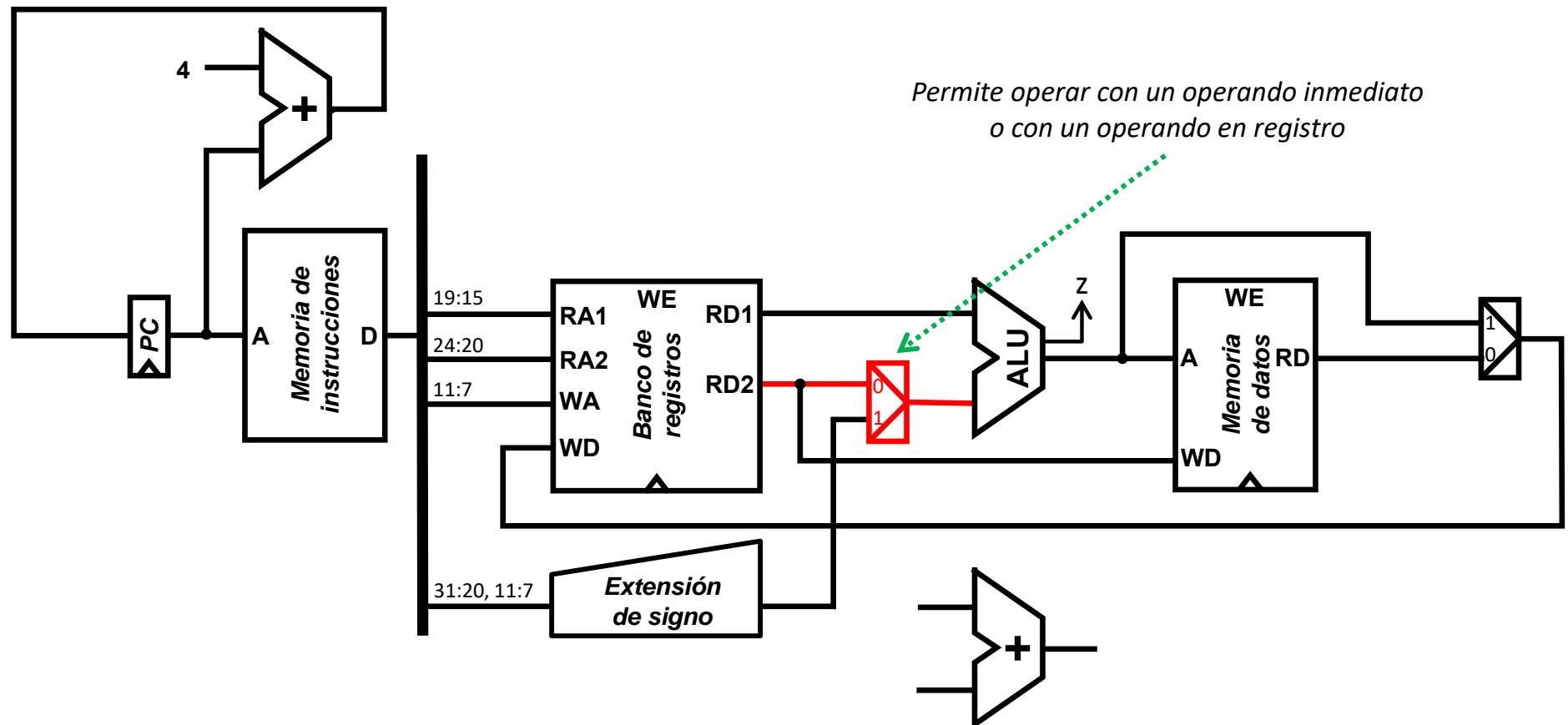


# Diseño de la ruta de datos

## Instrucciones tipo **add**: cálculo de la operación



$$BR[rd] \leftarrow BR[rs1] \text{ op } BR[rs2], PC \leftarrow PC+4$$



- Se **añade un multiplexor** para poder reutilizar la ALU para realizar operaciones aritmético lógicas con 2 registros

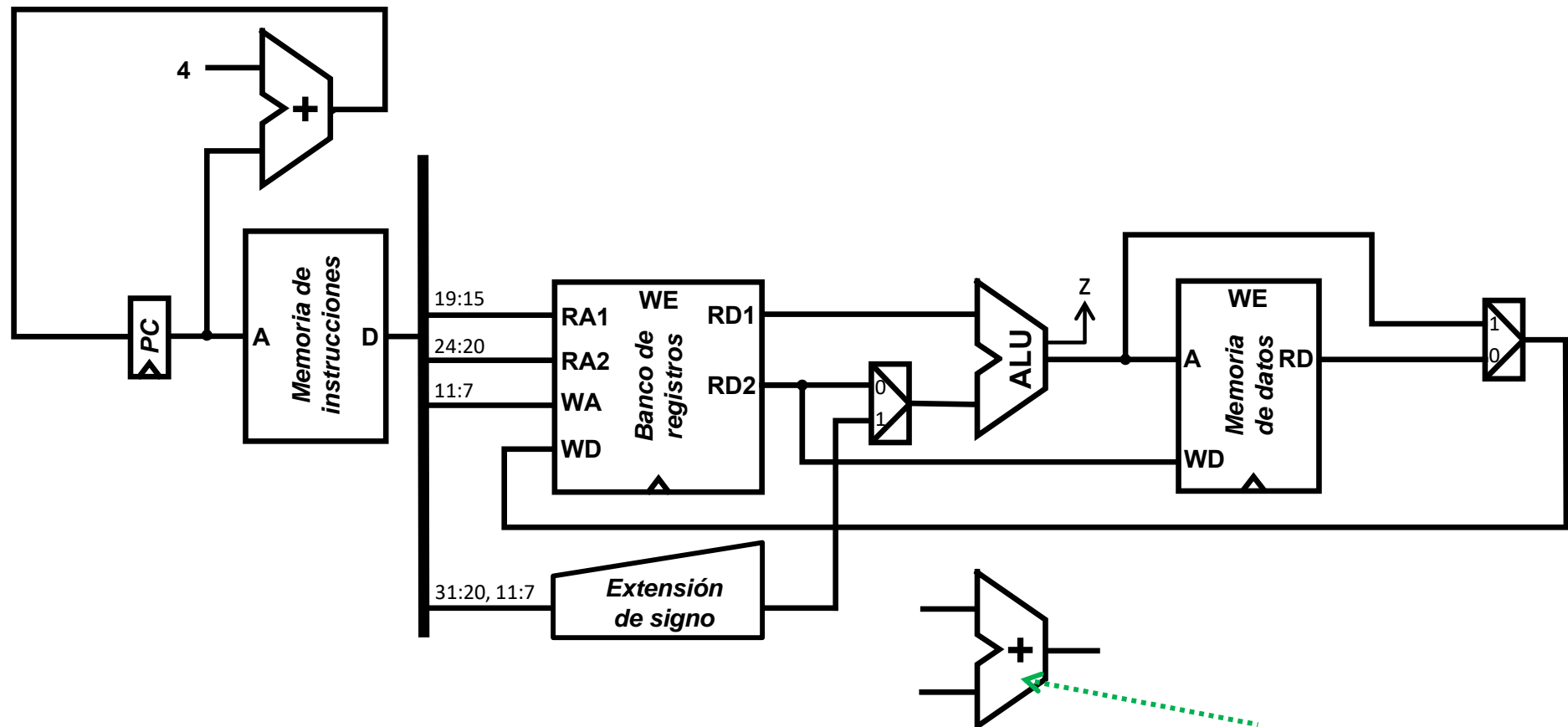




# Diseño de la ruta de datos

## Ruta de datos para instrucciones **lw/sw/addi/add**

- Esta ruta de datos puede ejecutar cualquier secuencia de instrucciones **lw**, **sw**, y/o aritmético-lógica.

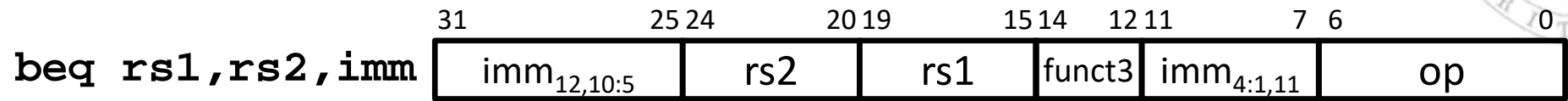


Este sumador solo se usa para calcular direcciones de salta en instrucciones **ja1/beq**

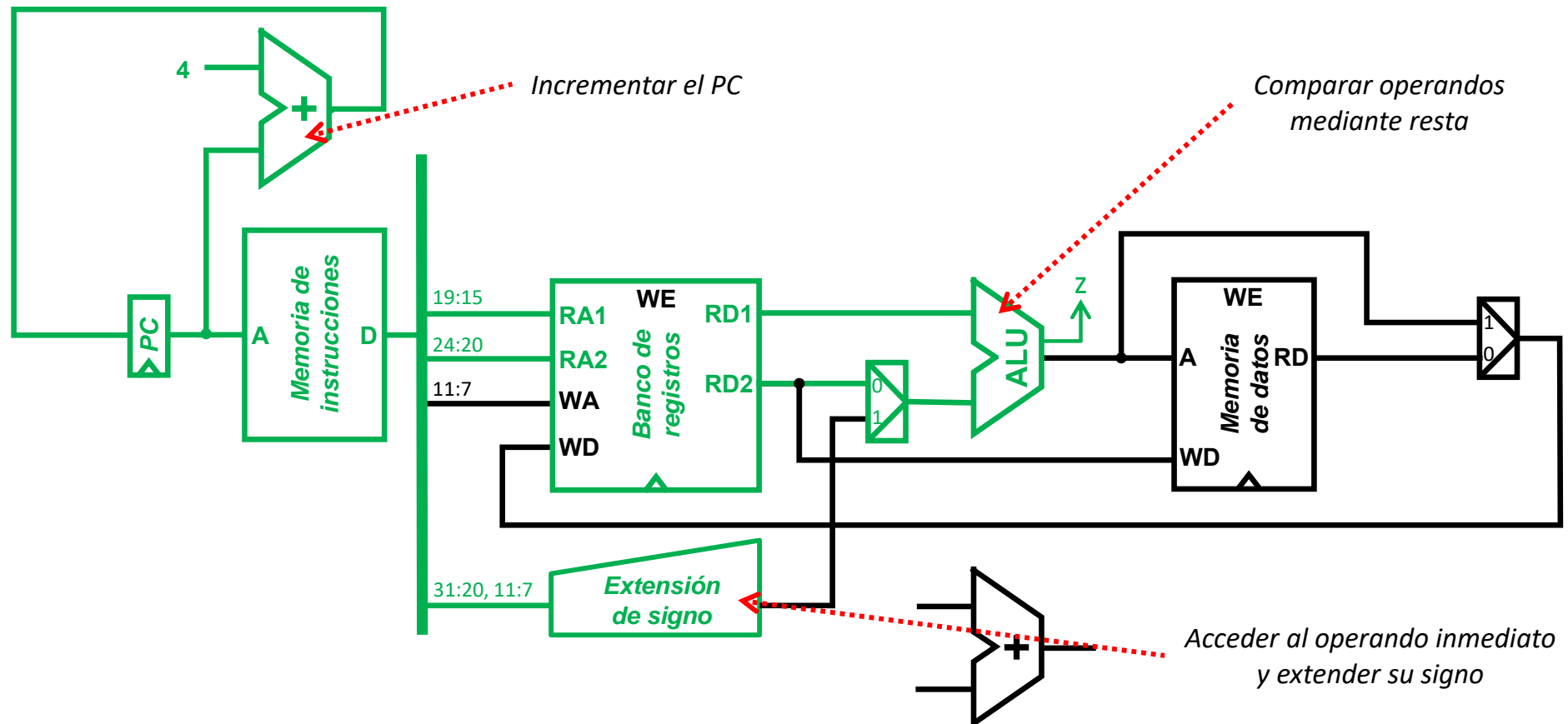


# Diseño de la ruta de datos

## Ruta de datos para instrucciones **beq**



$$PC \leftarrow \text{if} ( BR[ rs1 ] = BR[ rs2 ] ) \text{ then } ( PC + sExt(imm) ) \text{ else } ( PC+4 )$$

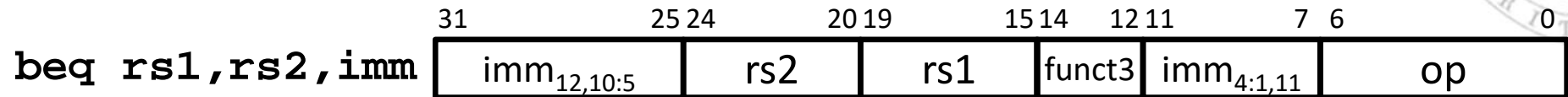


- Parte de esta ruta de datos **puede reutilizarse** para ejecutar la **instrucción beq**

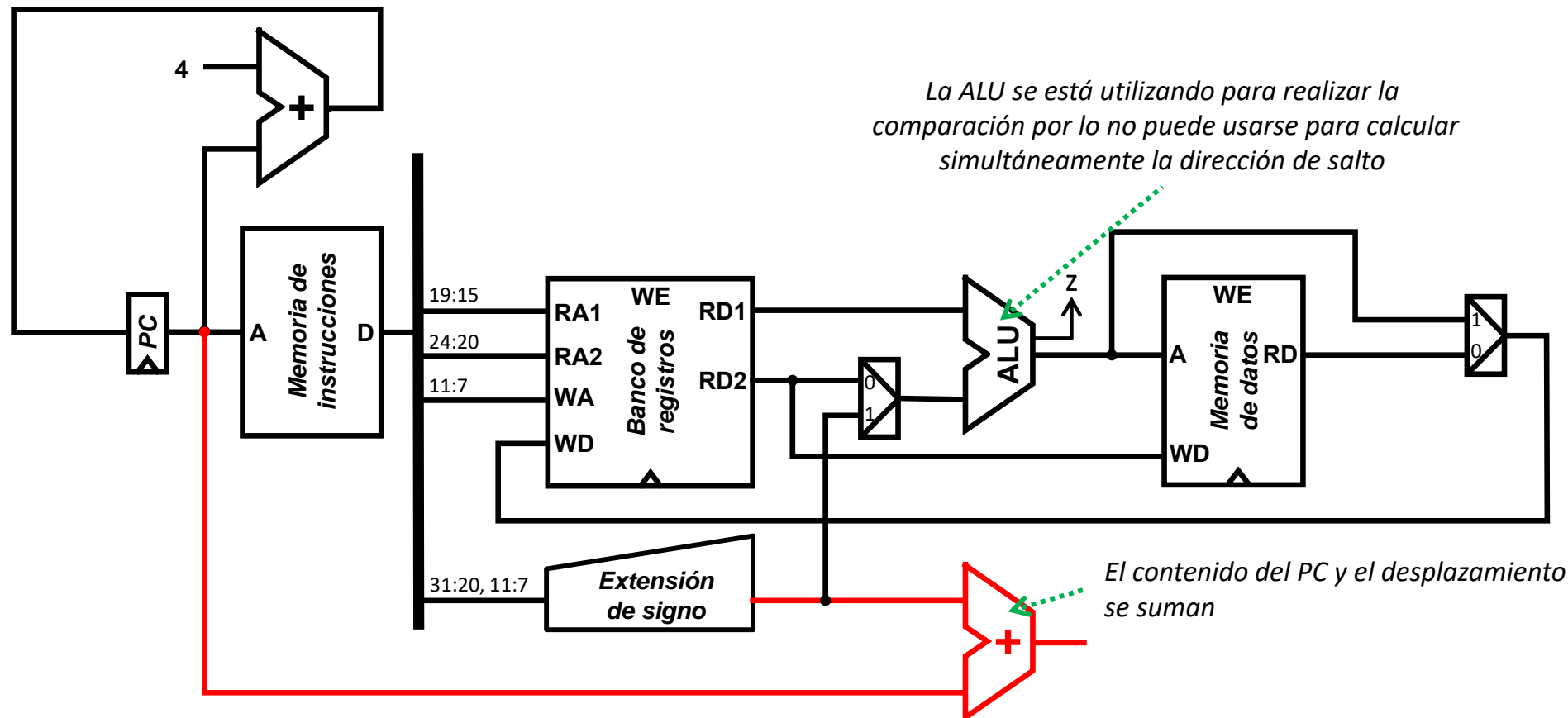


# Diseño de la ruta de datos

## Instrucción **beq**: cálculo de la dirección de salto



$$PC \leftarrow \text{if} ( BR[ rs1 ] = BR[ rs2 ] ) \text{ then } ( PC + sExt(imm) ) \text{ else } ( PC+4 )$$

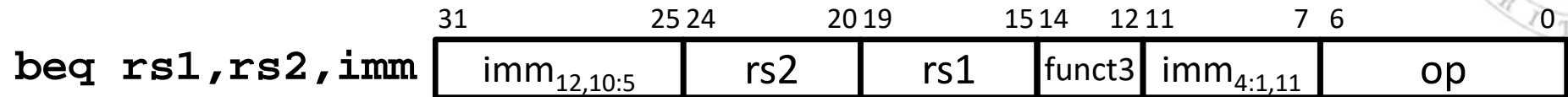


- El PC, el Extensor de Signo y el Sumador se conectan para calcular la dirección de salto.

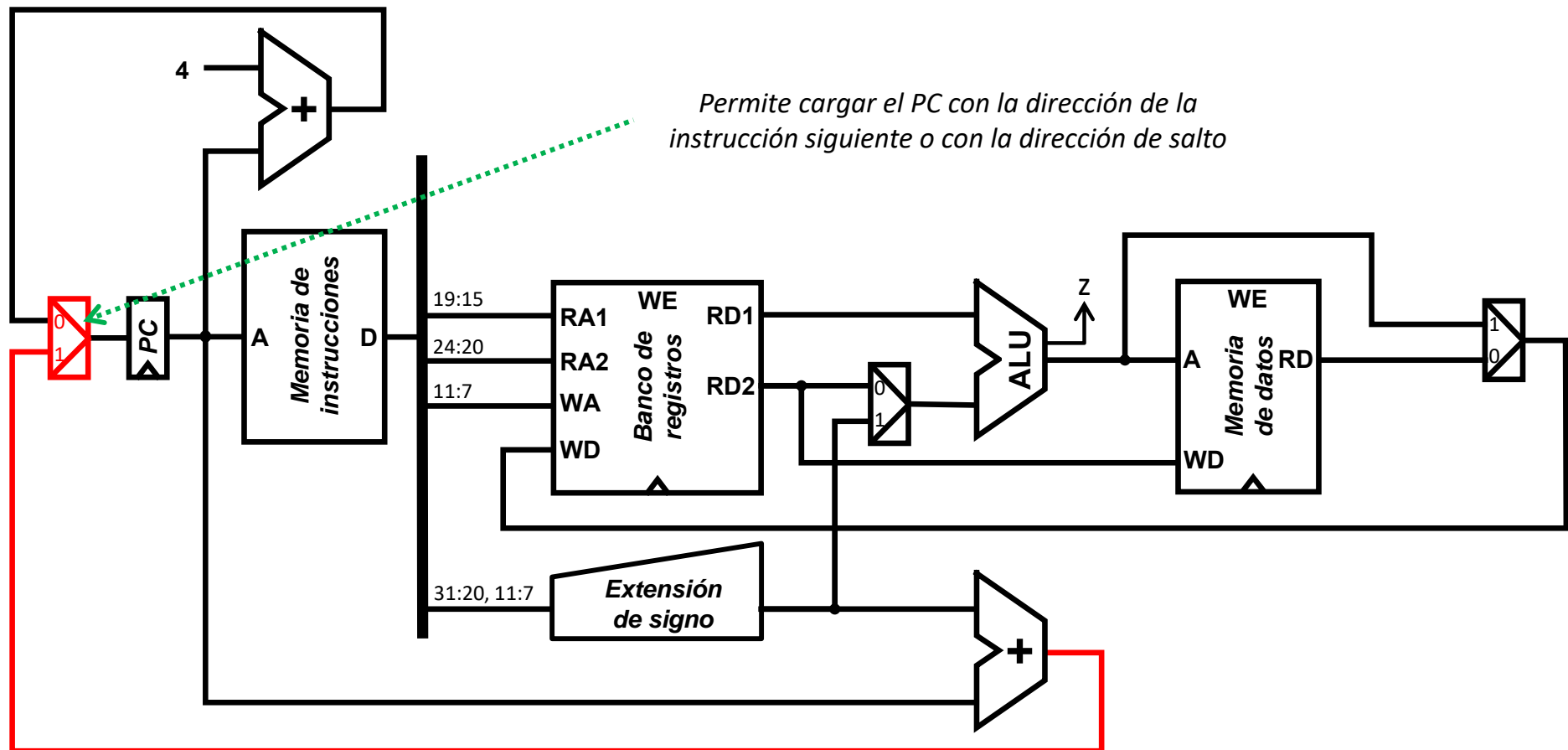


# Diseño de la ruta de datos

## Instrucción **beq**: actualización del PC



$$PC \leftarrow \text{if} ( BR[ rs1 ] = BR[ rs2 ] ) \text{ then } ( PC + sExt(imm) ) \text{ else } ( PC+4 )$$



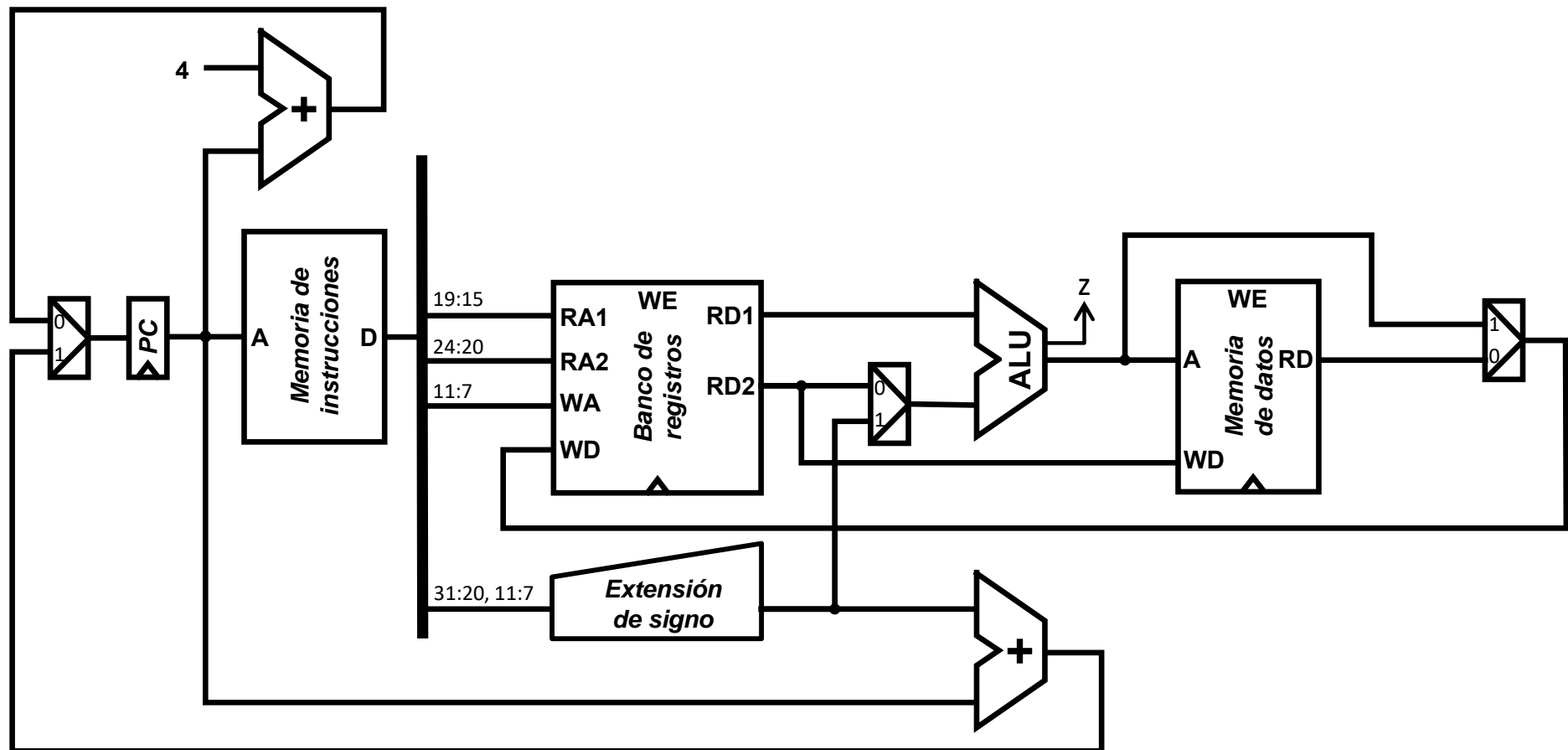
- Se **añade un multiplexor** para poder almacenar la dirección de salto en el PC.



# Diseño de la ruta de datos

## Ruta de datos para instrucciones **lw/sw/addi/add/beq**

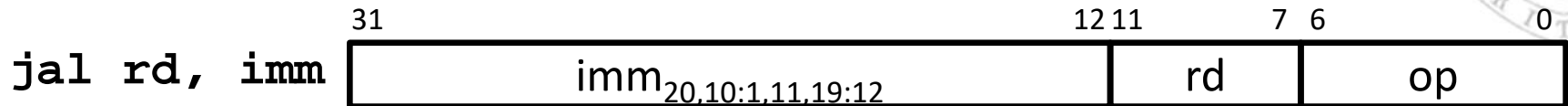
- Esta ruta de datos puede ejecutar **cualquier secuencia** de instrucciones **lw**, **sw**, **aritmético-lógica**. y/o de salto condicional.



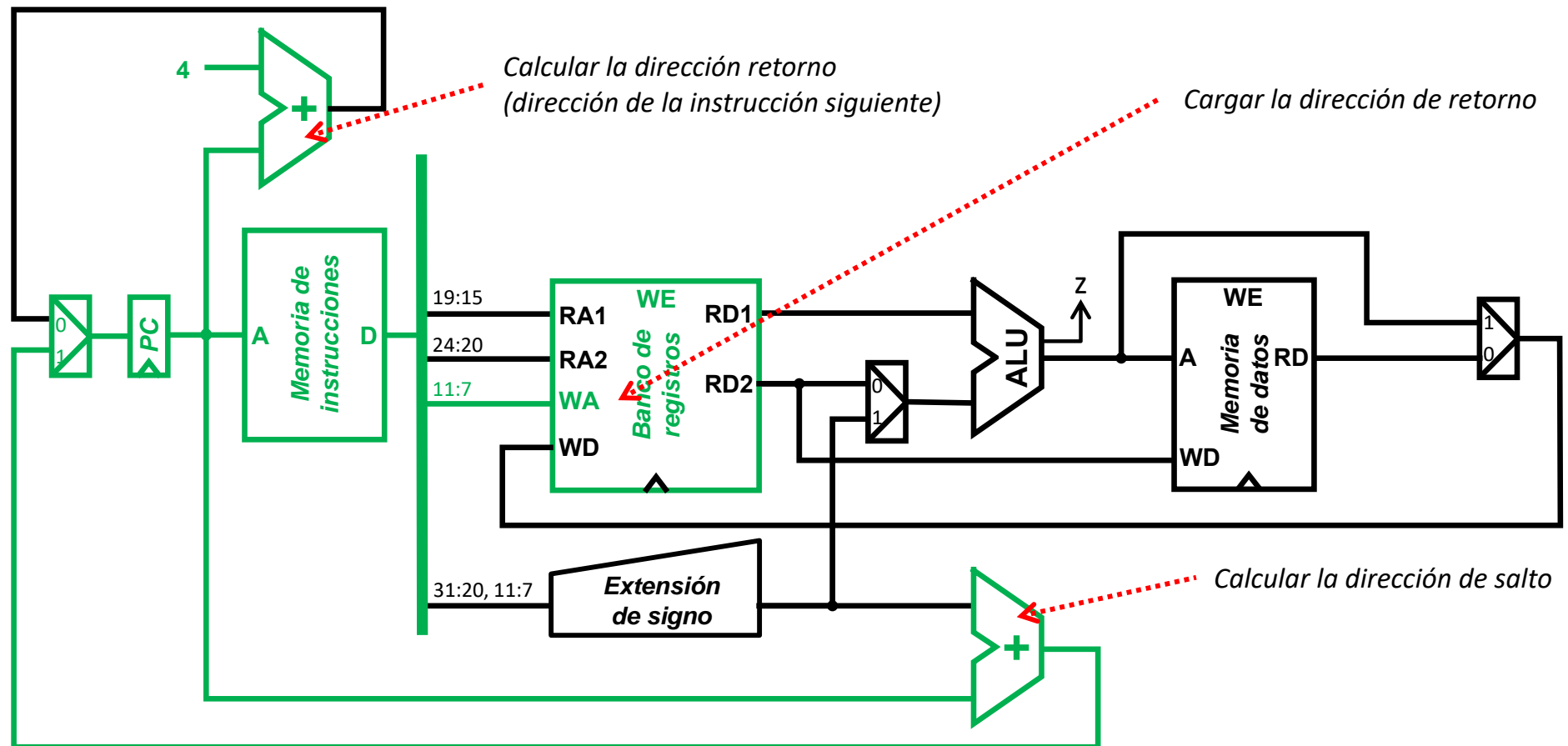


# Diseño de la ruta de datos

## Ruta de datos para instrucciones **jal**



$$PC \leftarrow PC + sExt(imm), BR[rd] \leftarrow PC+4$$

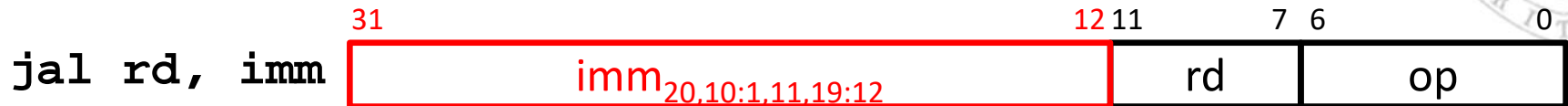


- Parte de esta ruta de datos **puede reutilizarse** para ejecutar la **instrucción jal**

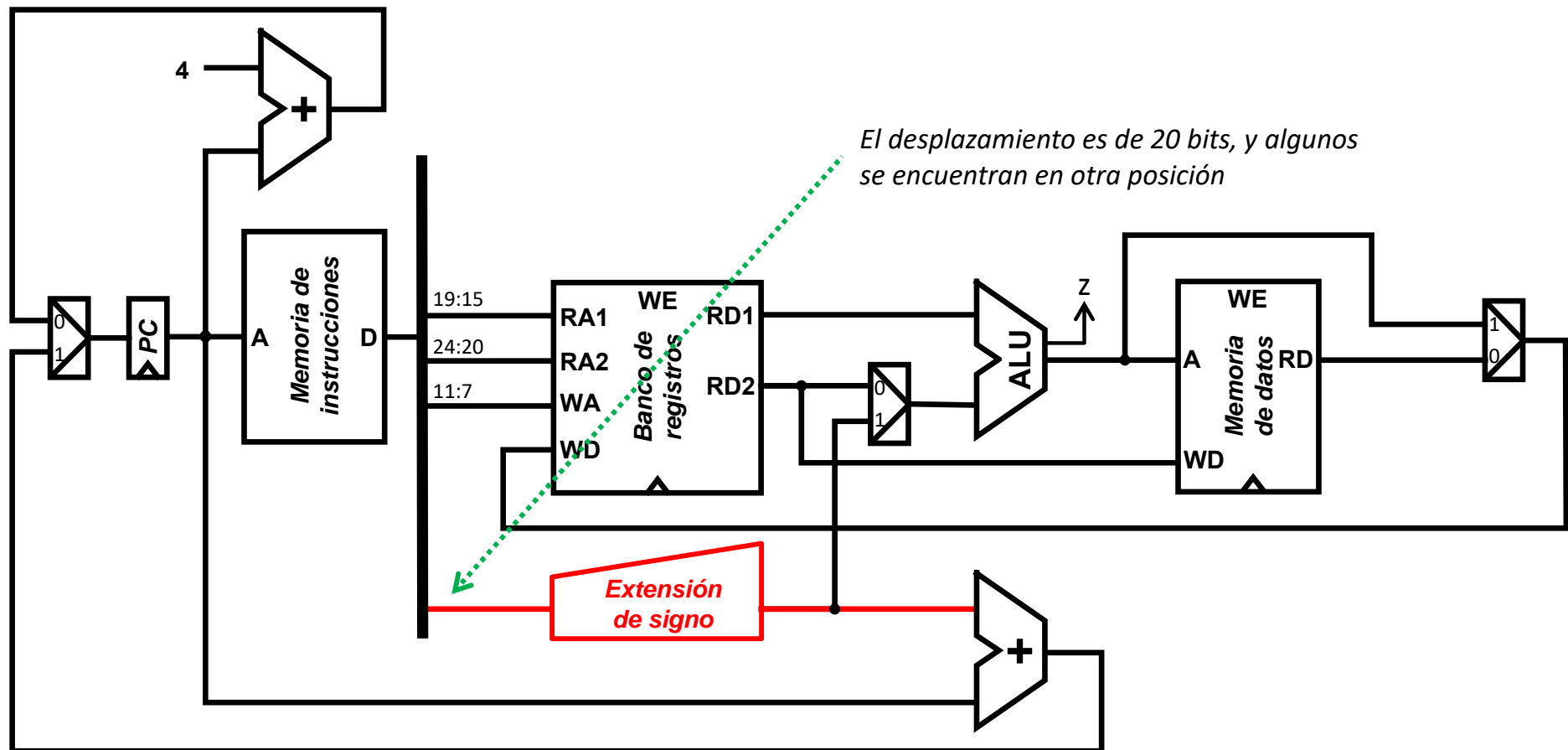


# Diseño de la ruta de datos

## Instrucción **jal**: cálculo de la dirección de salto



$$PC \leftarrow PC + sExt(imm), BR[rd] \leftarrow PC+4$$

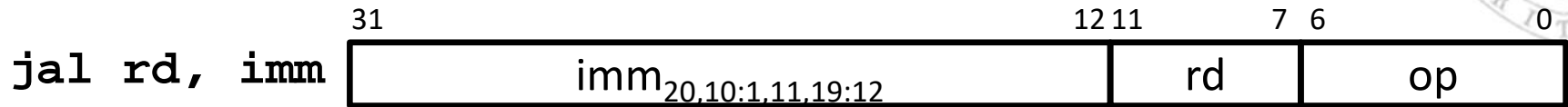


- La conexión entre la Memoria de Instrucciones y el Extensor de Signo se amplía.



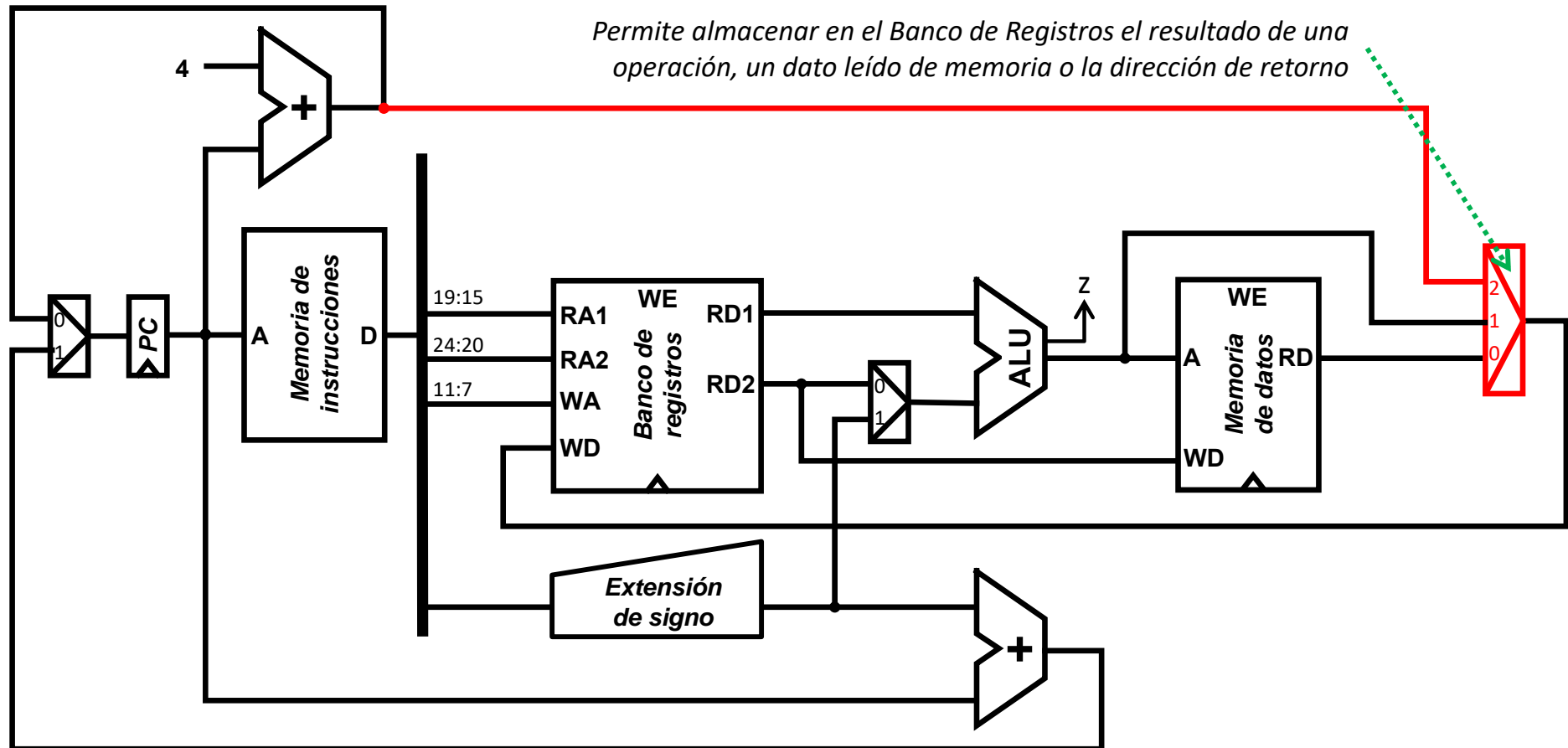
# Diseño de la ruta de datos

## Instrucción **jal**: salvado de la dirección de retorno



$$PC \leftarrow PC + sExt(imm), BR[rd] \leftarrow PC+4$$

Permite almacenar en el Banco de Registros el resultado de una operación, un dato leído de memoria o la dirección de retorno



- Se **amplía el multiplexor** para poder almacenar la dirección de retorno en rd.

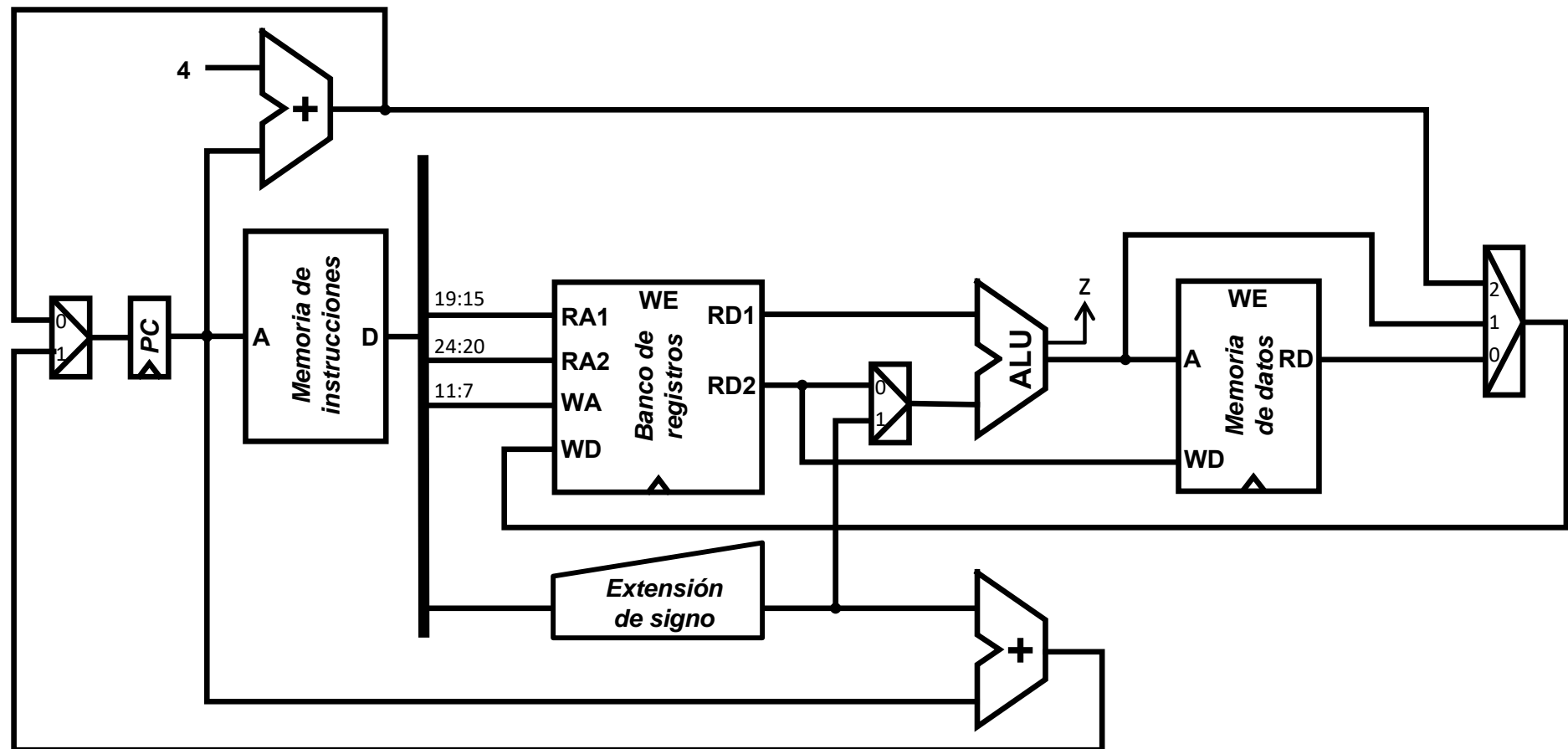




# Diseño de la ruta de datos

## Ruta de datos RISC-V reducido

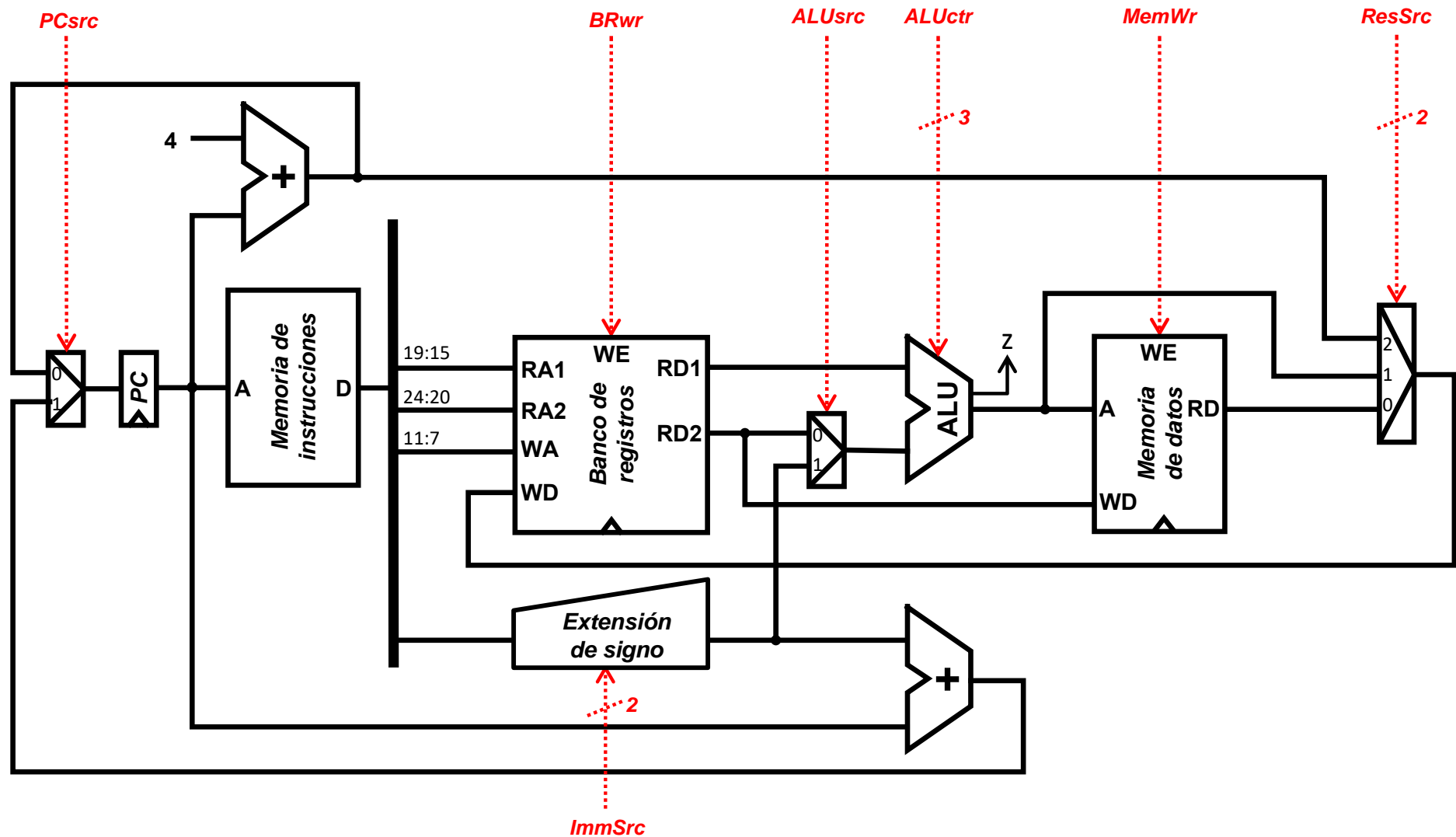
- Esta ruta de datos puede ejecutar cualquier secuencia de instrucciones del repertorio del RISC-V reducido.





# Diseño de la ruta de datos

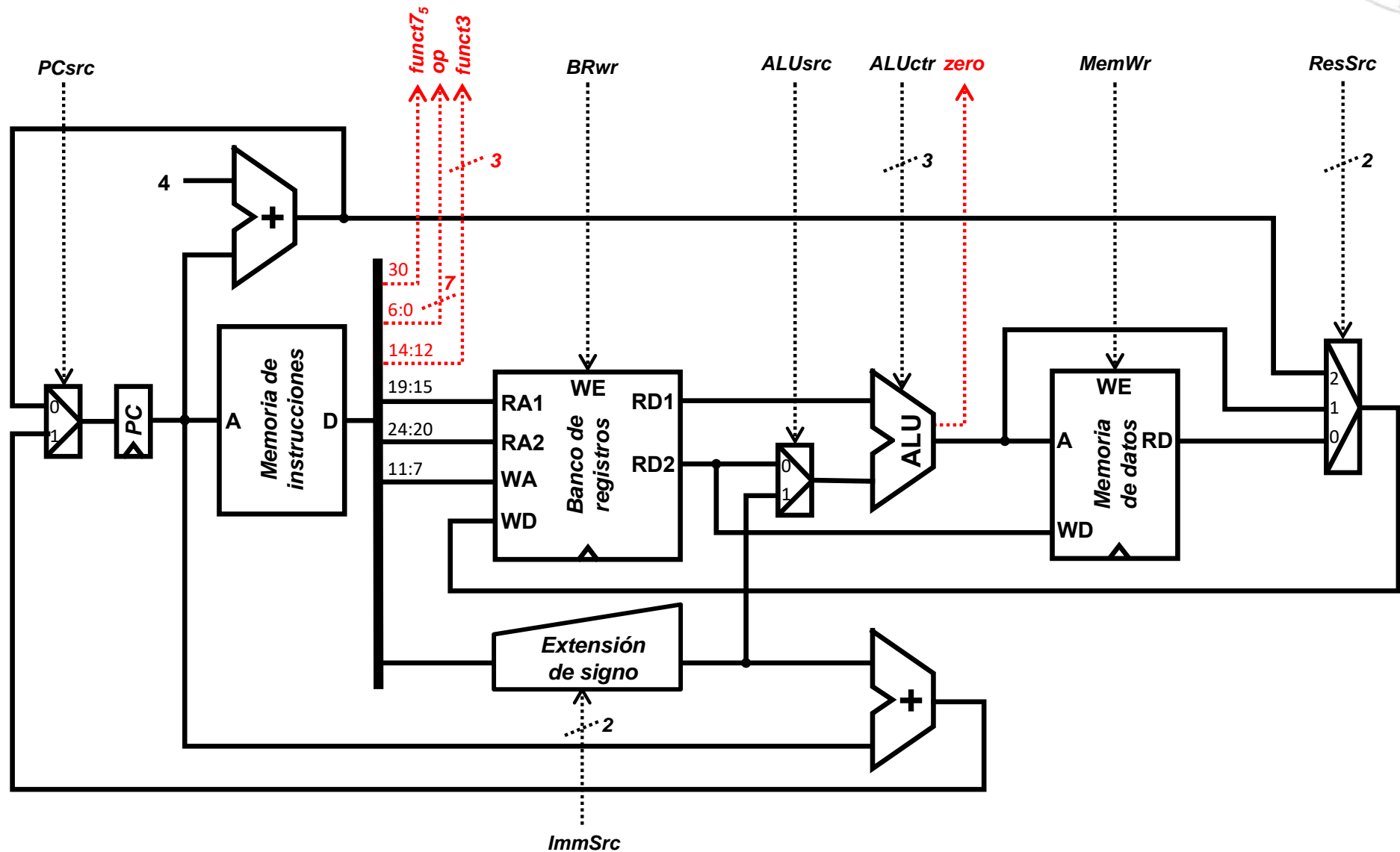
## Señales de control





# Diseño de la ruta de datos

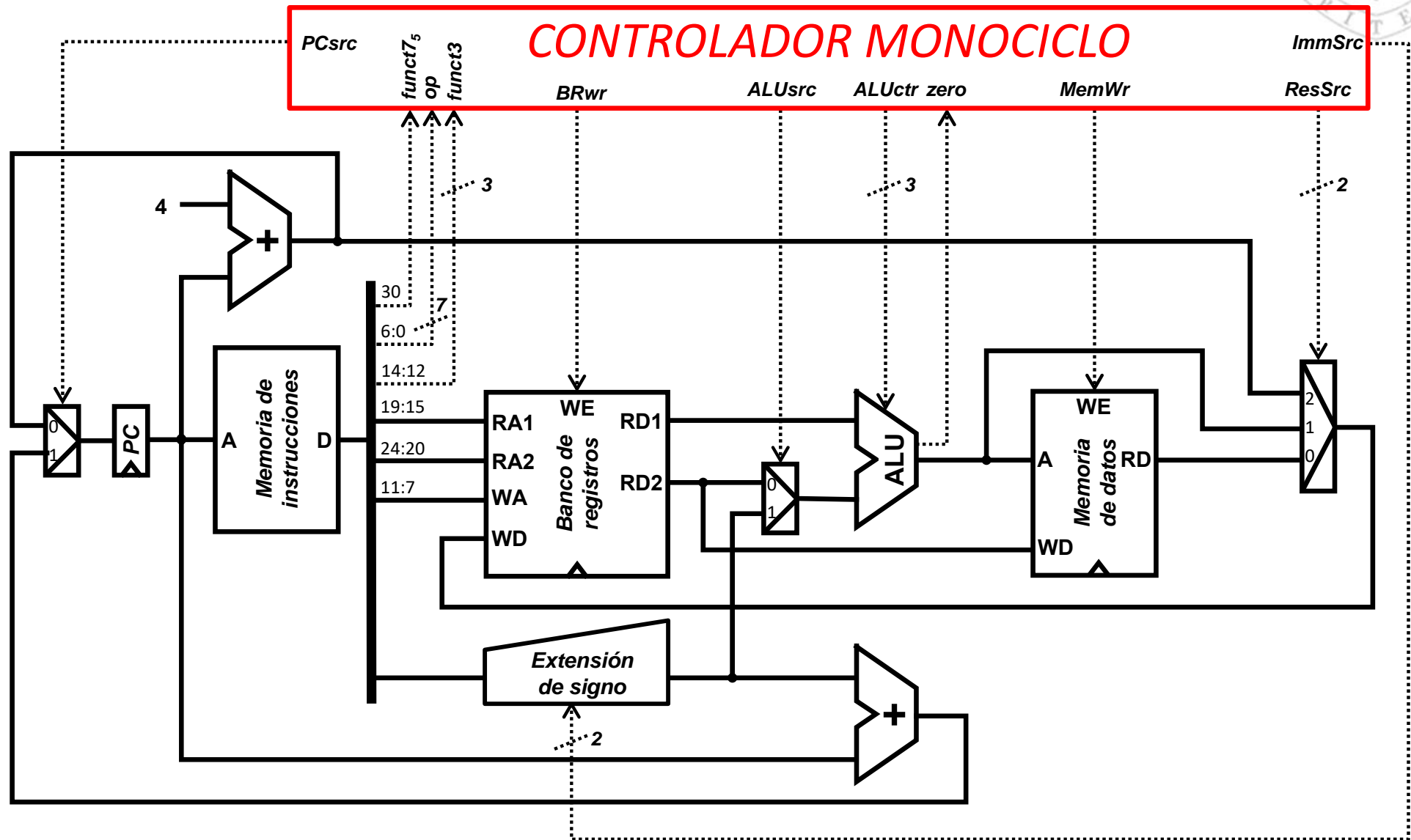
## Señales de estado





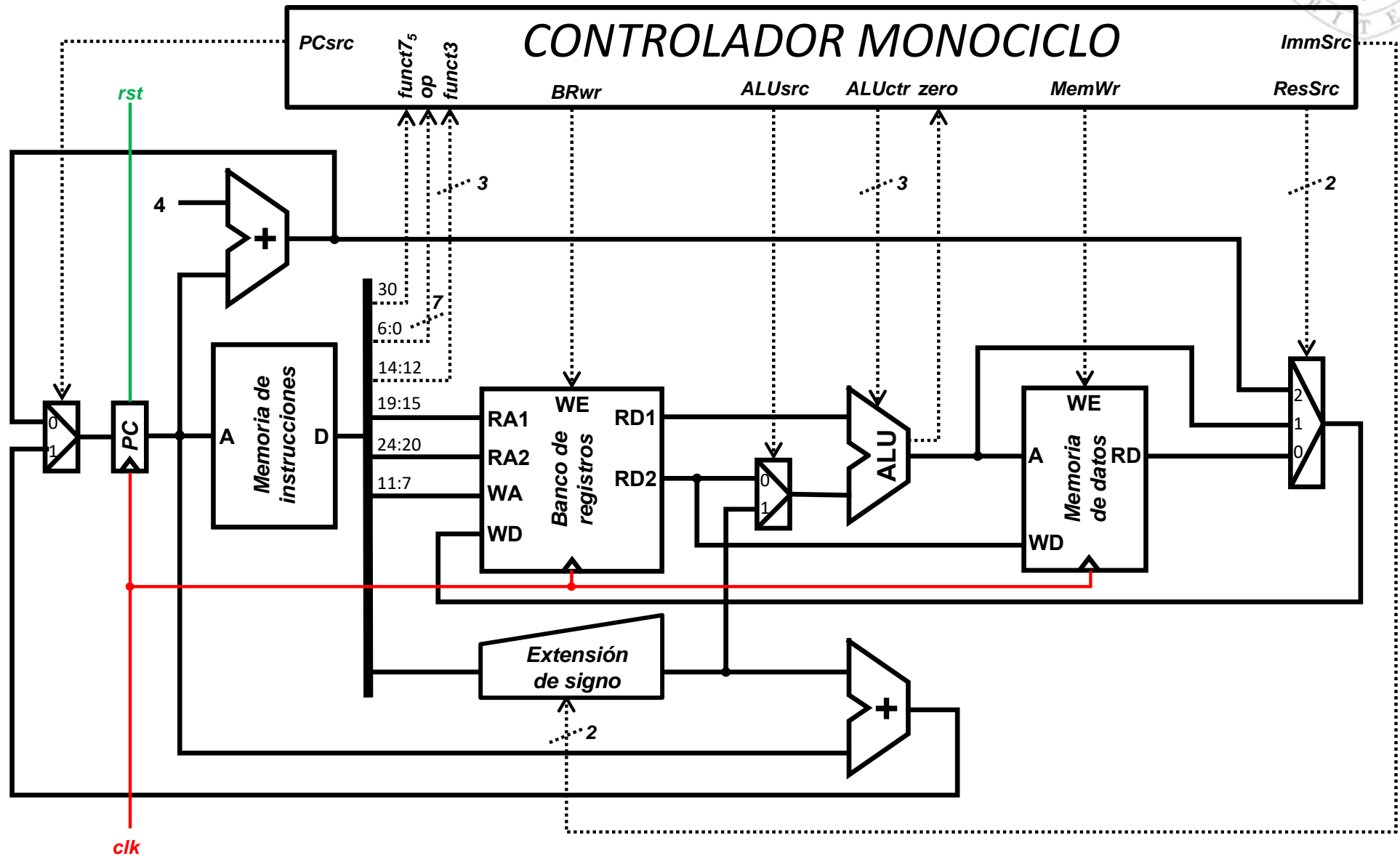
# Diseño de la ruta de datos

## Conexión con el controlador



# Diseño de la ruta de datos

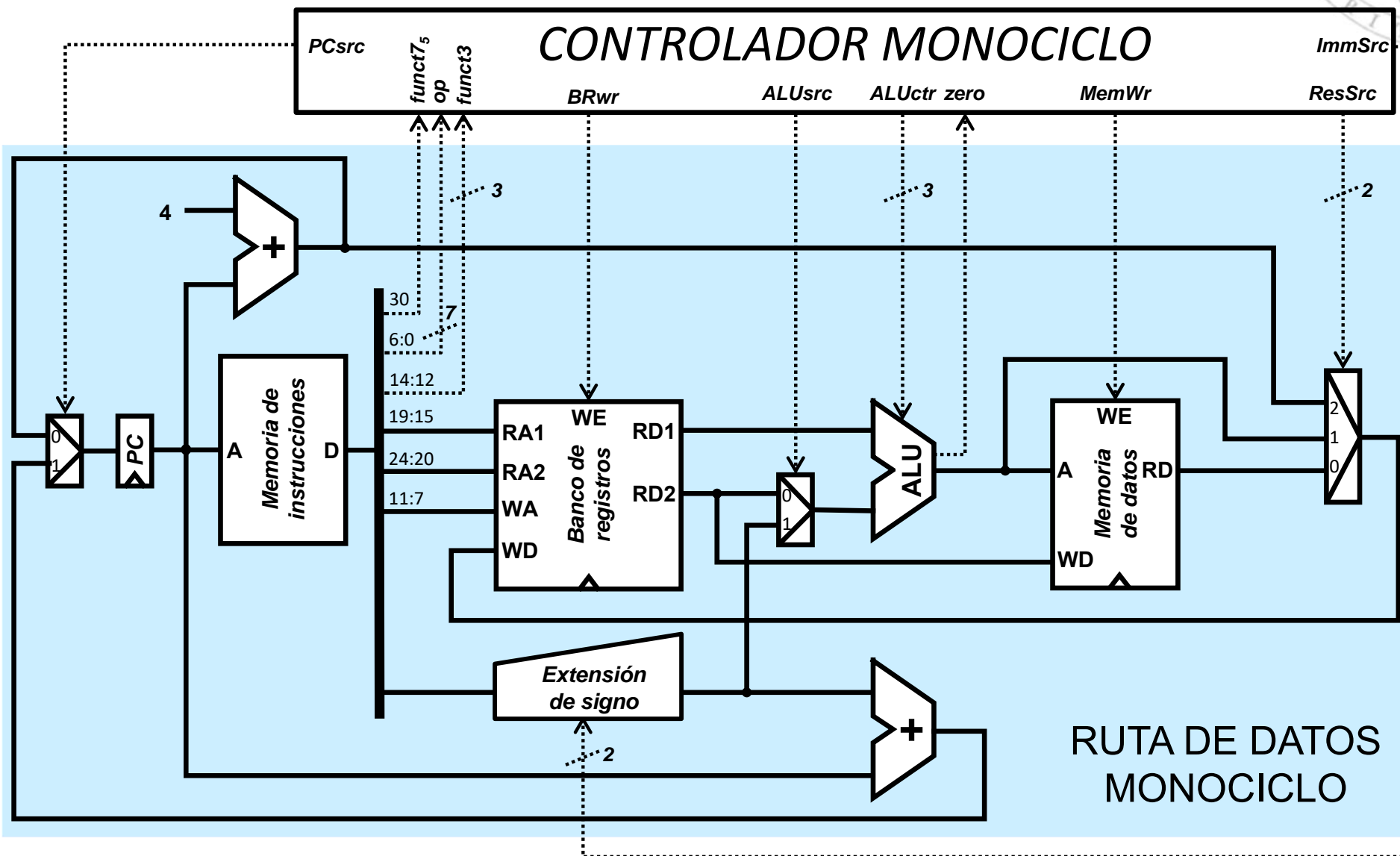
## Conexión del reloj y reset





# Diseño de la ruta de datos

## Estructura del sistema completo

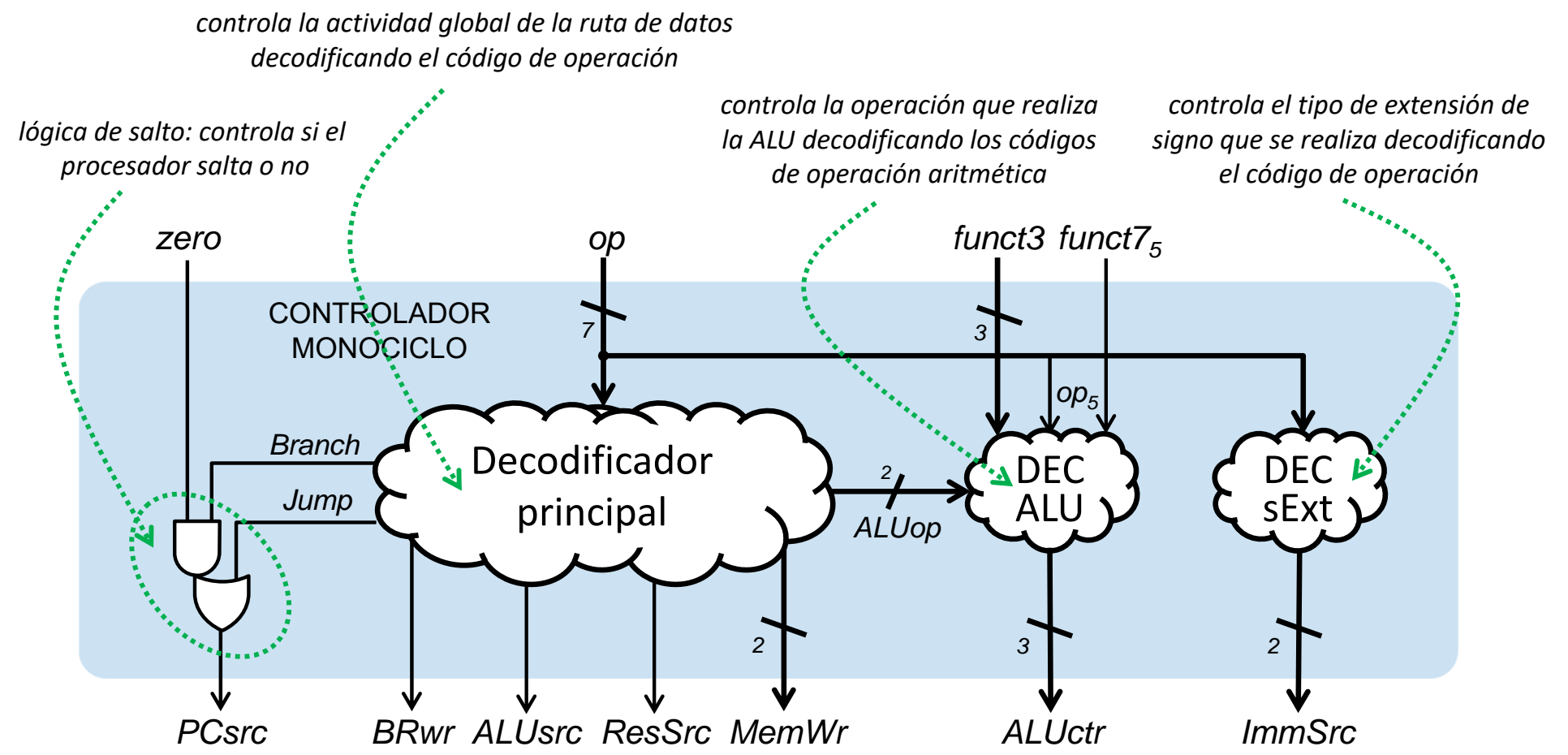




# Diseño del controlador

## Estructura del controlador (i)

- En el **procesador monociclo**, el controlador es un circuito **combinacional**:
  - Que para facilitar el diseño estará formado **por 4 subcircuitos**.





# Diseño del controlador

## Estructura del controlador (ii)

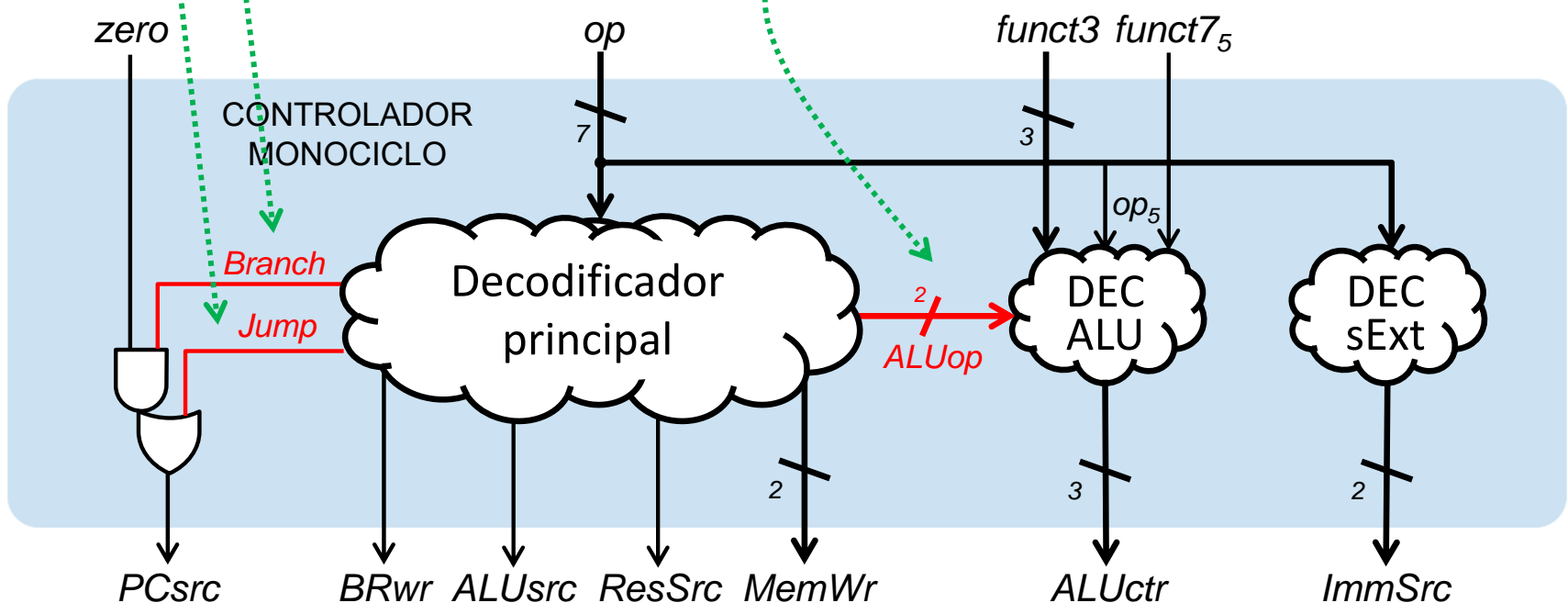
- En el **procesador monociclo**, el controlador es un circuito **combinacional**:
  - Que para facilitar el diseño estará formado **por 4 subcircuitos**:

*Jump*: indica si es una instrucción *ja1*

*Branch*: indica si es una instrucción *beq*

*ALUop*: Indica la operación que debe realizar la ALU:

- **00**: sumar (cálculo de una dirección efectiva de *lw/sw*)
- **01**: restar (comparación *beq*)
- **10**: operar según la instrucción (tipo I / tipo R)
- La instrucción *ja1* no utiliza la ALU

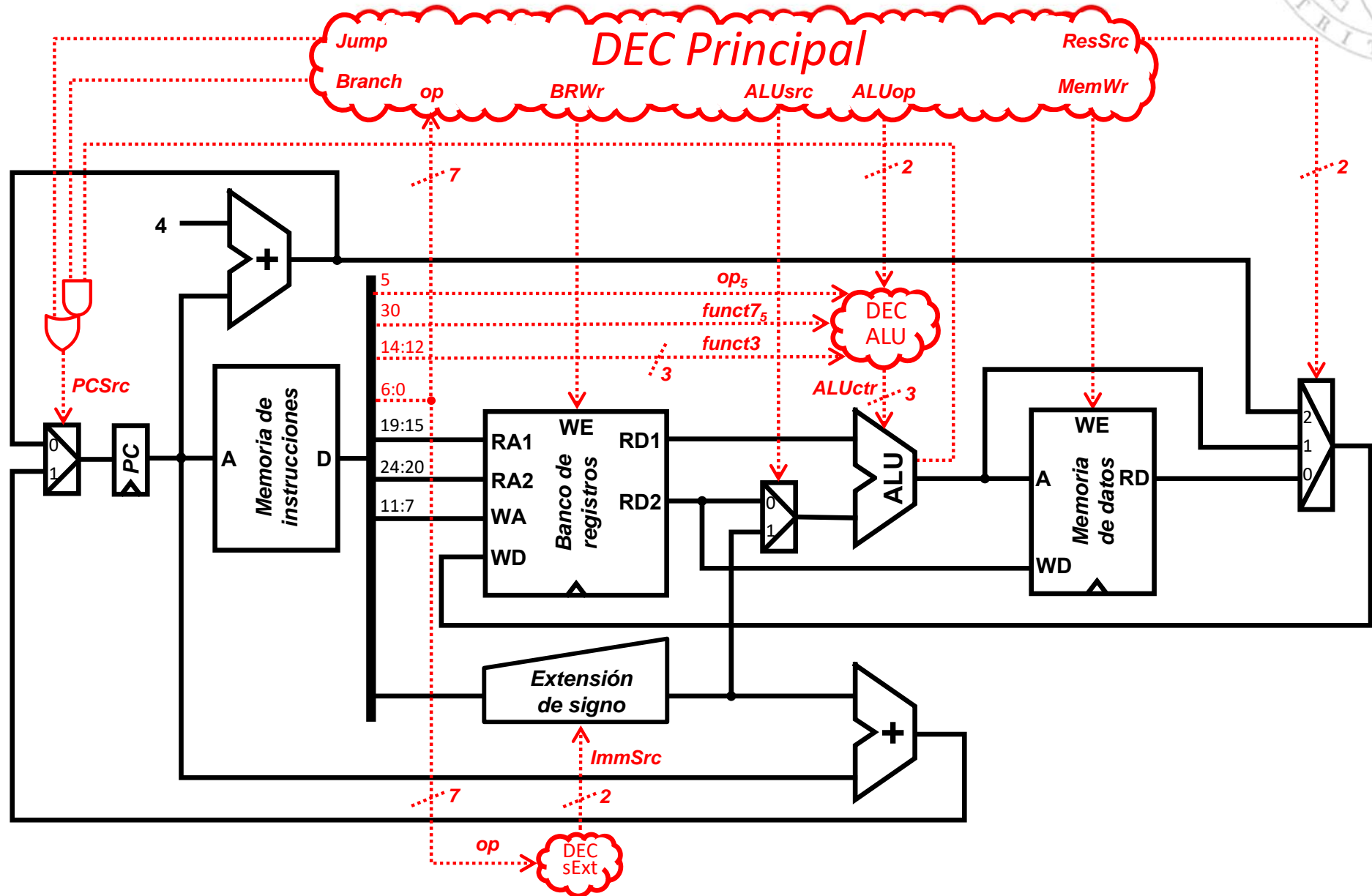






# Diseño del controlador

## Estructura del controlador (iii)

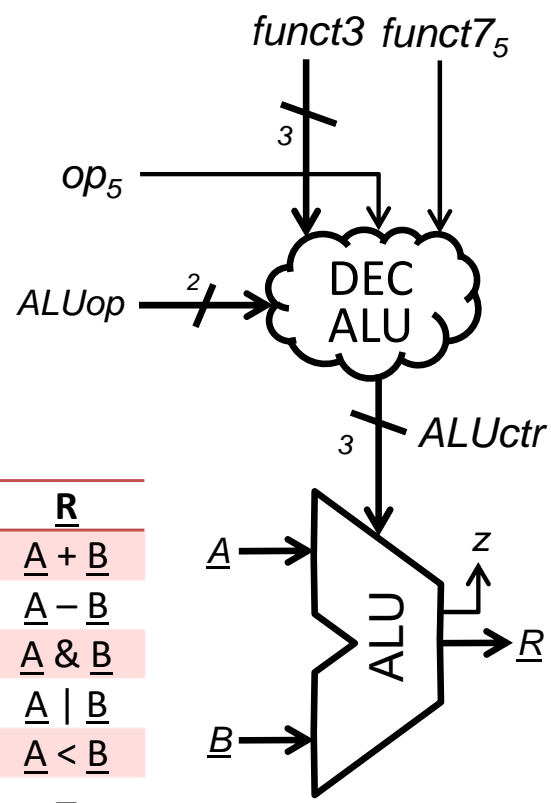




# Diseño del controlador

## Diseño del controlador: DEC local a la ALU

- Este subcircuito indica a la ALU el tipo de operación que deber realizar.
  - Adapta la codificación de la operación en la instrucción a la codificación de operaciones de la ALU.



ALUctr	R
000	$\underline{A} + \underline{B}$
001	$\underline{A} - \underline{B}$
010	$\underline{A} \& \underline{B}$
011	$\underline{A}   \underline{B}$
101	$\underline{A} < \underline{B}$
resto	-

Tabla de verdad

ALUop	op <sub>5</sub>	funct7 <sub>5</sub>	funct3	ALUctr
00 <sup>(sumar)</sup>	X	X	XXX	000 <sup>(A + B)</sup>
01 <sup>(restar)</sup>	X	X	XXX	001 <sup>(A - B)</sup>
10 <sup>(operar)</sup>	0	X	000 <sup>(addi)</sup>	000 <sup>(A + B)</sup>
10 <sup>(operar)</sup>	1	0	000 <sup>(add)</sup>	000 <sup>(A + B)</sup>
10 <sup>(operar)</sup>	1	1	000 <sup>(sub)</sup>	001 <sup>(A - B)</sup>
10 <sup>(operar)</sup>	X	X	010 <sup>(slt/slti)</sup>	101 <sup>(A &lt; B)</sup>
10 <sup>(operar)</sup>	X	X	110 <sup>(or/ori)</sup>	011 <sup>(A   B)</sup>
10 <sup>(operar)</sup>	X	X	111 <sup>(and/andi)</sup>	010 <sup>(A &amp; B)</sup>



# Diseño del controlador

## Diseño del controlador: DEC local a la ALU

- Este subcircuito indica a la ALU el tipo de operación que deber realizar.
  - Adapta la codificación de la operación en la instrucción a la codificación de operaciones de la ALU.

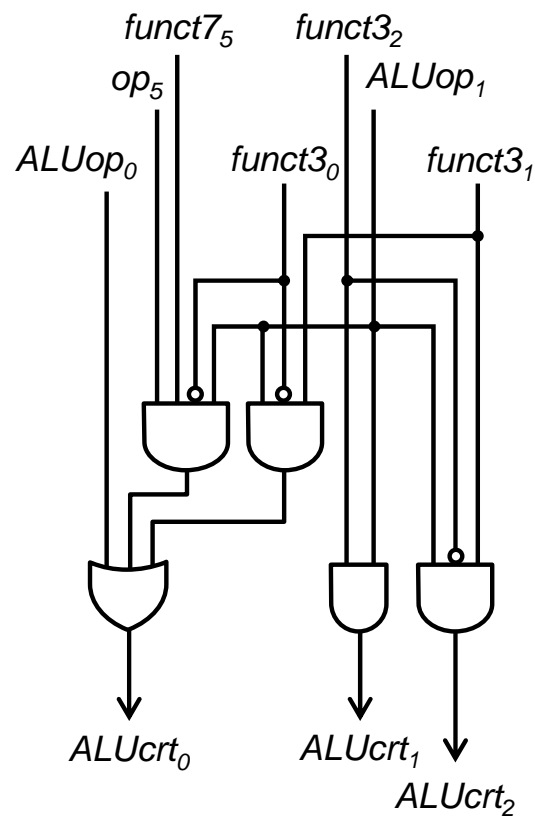


Tabla de verdad

ALUop	op <sub>5</sub>	funct7 <sub>5</sub>	funct3	ALUcrt
00 <sup>(sumar)</sup>	X	X	XXX	000 <sup>(A + B)</sup>
01 <sup>(restar)</sup>	X	X	XXX	001 <sup>(A - B)</sup>
10 <sup>(operar)</sup>	0	X	000 <sup>(addi)</sup>	000 <sup>(A + B)</sup>
10 <sup>(operar)</sup>	1	0	000 <sup>(add)</sup>	000 <sup>(A + B)</sup>
10 <sup>(operar)</sup>	1	1	000 <sup>(sub)</sup>	001 <sup>(A - B)</sup>
10 <sup>(operar)</sup>	X	X	010 <sup>(slt/slti)</sup>	101 <sup>(A &lt; B)</sup>
10 <sup>(operar)</sup>	X	X	110 <sup>(or/ori)</sup>	011 <sup>(A   B)</sup>
10 <sup>(operar)</sup>	X	X	111 <sup>(and/andi)</sup>	010 <sup>(A &amp; B)</sup>



# Diseño del controlador

## Diseño del controlador: DEC local al Extensor de Signo

- Esté subcircuito indica al Extensor de Signo el tipo de extensión que deber realizar.

ImmSrc	R
00	sExt( <u>X</u> ) tipo-I
01	sExt( <u>X</u> ) tipo-S
10	sExt( <u>X</u> ) tipo-B
11	sExt( <u>X</u> ) tipo-J

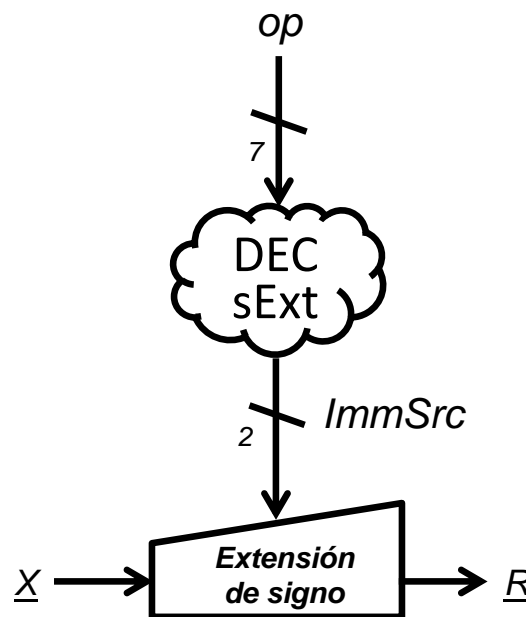


Tabla de verdad

Op	ImmSrc
0000011 <sup>(lw)</sup>	00 <sup>(tipo-I)</sup>
0100011 <sup>(sw)</sup>	01 <sup>(tipo-S)</sup>
0010011 <sup>(tipo-I)</sup>	00 <sup>(tipo-I)</sup>
0110011 <sup>(tipo-R)</sup>	—
1100011 <sup>(beq)</sup>	10 <sup>(tipo-B)</sup>
1101111 <sup>(jal)</sup>	11 <sup>(tipo-J)</sup>



# Diseño del controlador

## Diseño del controlador: DEC local al Extensor de Signo

- Esté subcircuito indica al Extensor de Signo el tipo de extensión que deber realizar.

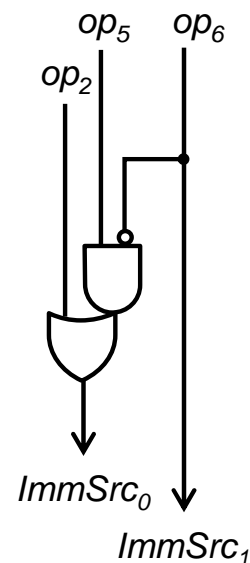


Tabla de verdad

Op	ImmSrc
0000011 <sup>(lw)</sup>	00 <sup>(tipo-I)</sup>
0100011 <sup>(sw)</sup>	01 <sup>(tipo-S)</sup>
0010011 <sup>(tipo-I)</sup>	00 <sup>(tipo-I)</sup>
0110011 <sup>(tipo-R)</sup>	—
1100011 <sup>(beq)</sup>	10 <sup>(tipo-B)</sup>
1101111 <sup>(jal)</sup>	11 <sup>(tipo-J)</sup>



# Diseño del controlador

## Diseño del controlador: DEC principal

- Este subcircuito gobierna el comportamiento general del procesador.

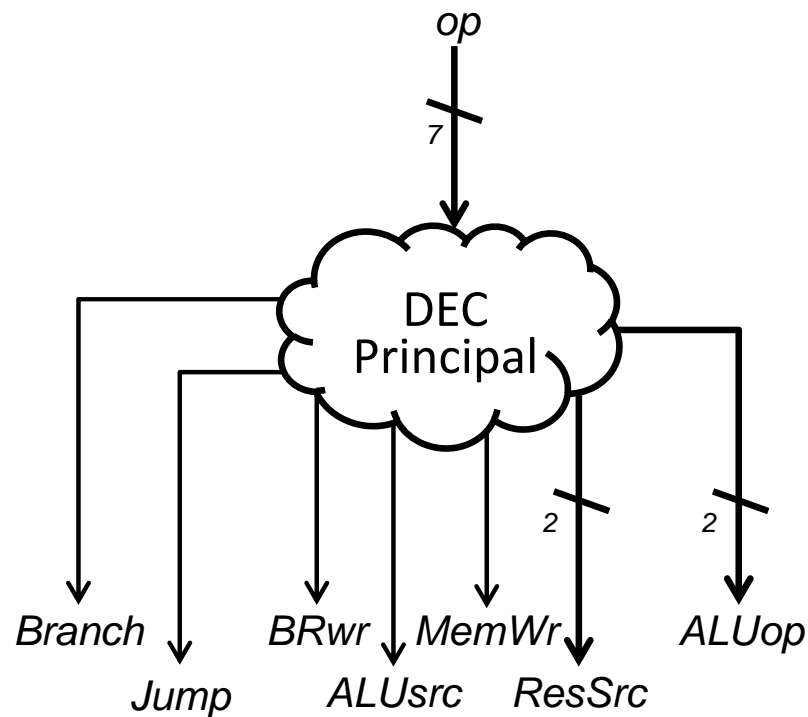


Tabla de verdad

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0000011 ( <i>lw</i> )							
0100011 ( <i>sw</i> )							
0010011 ( <i>tipo-l</i> )							
0110011 ( <i>tipo-R</i> )							
1100011 ( <i>beq</i> )							
1101111 ( <i>jal</i> )							



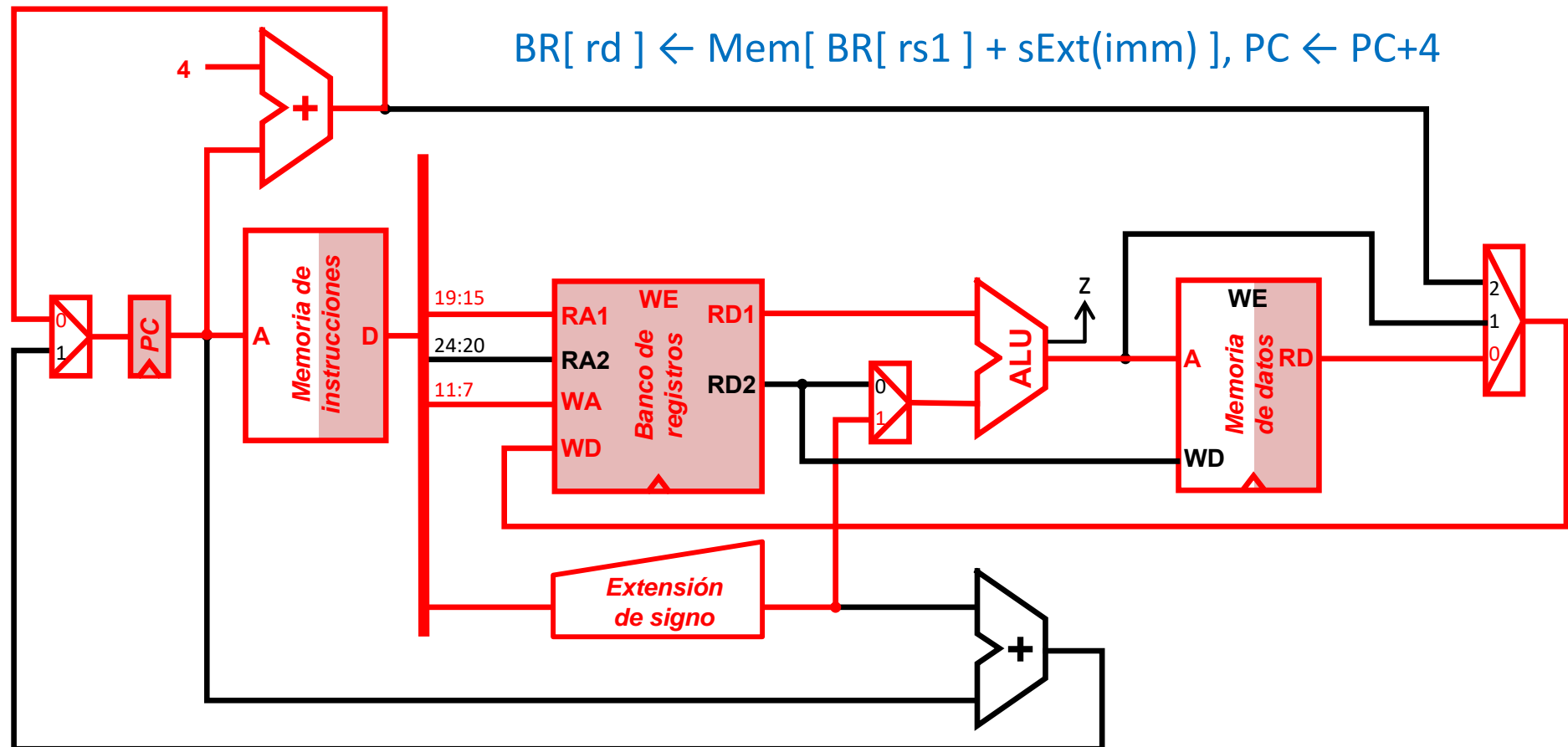
# Diseño del controlador

## Diseño del controlador: DEC principal

lw rd, imm(rs1)

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0000011 <sup>(lw)</sup>							

$BR[rd] \leftarrow Mem[BR[rs1] + sExt(imm)], PC \leftarrow PC+4$





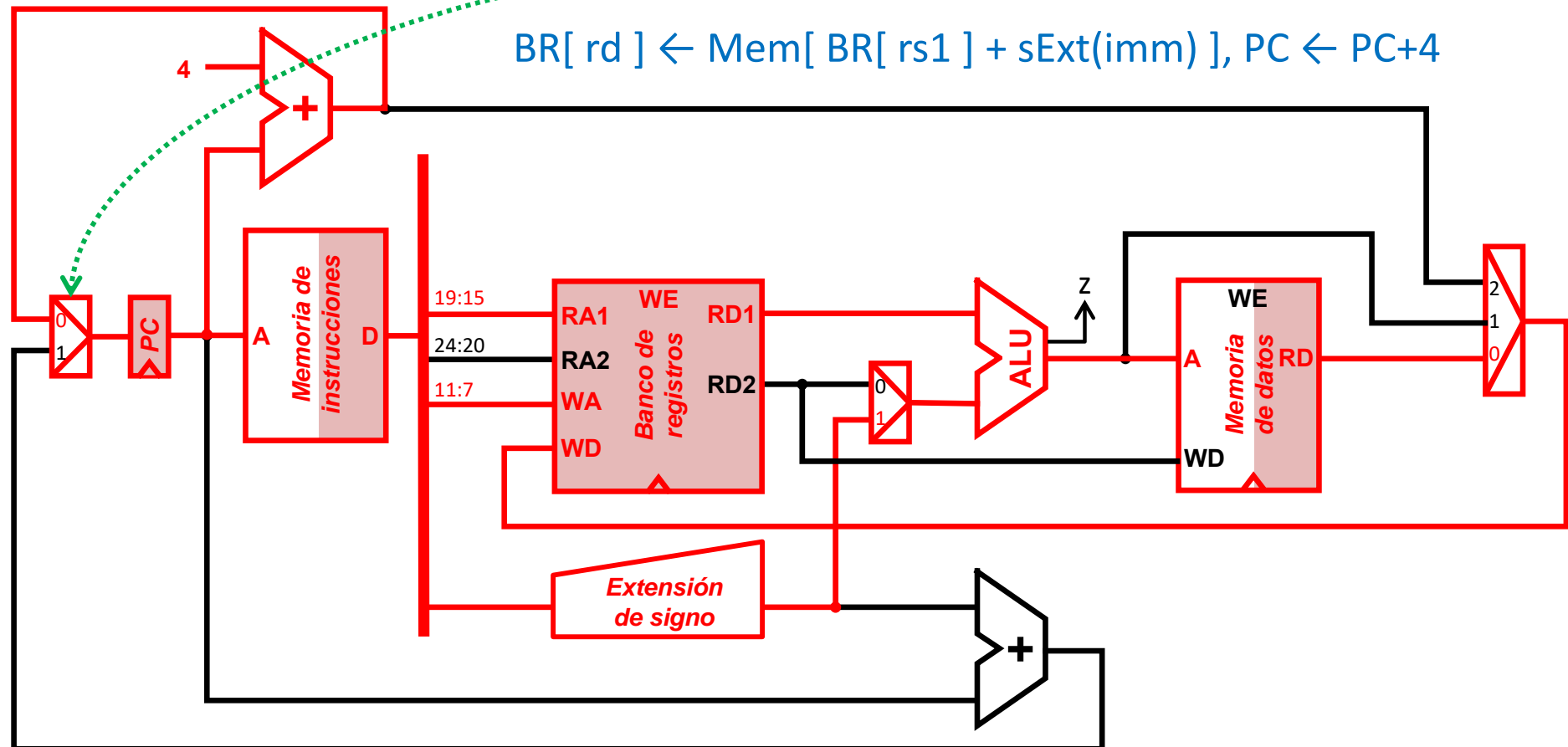
# Diseño del controlador

## Diseño del controlador: DEC principal

lw rd, imm(rs1)

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0000011 (lw)	0	0					

$BR[rd] \leftarrow Mem[BR[rs1] + sExt(imm)], PC \leftarrow PC + 4$







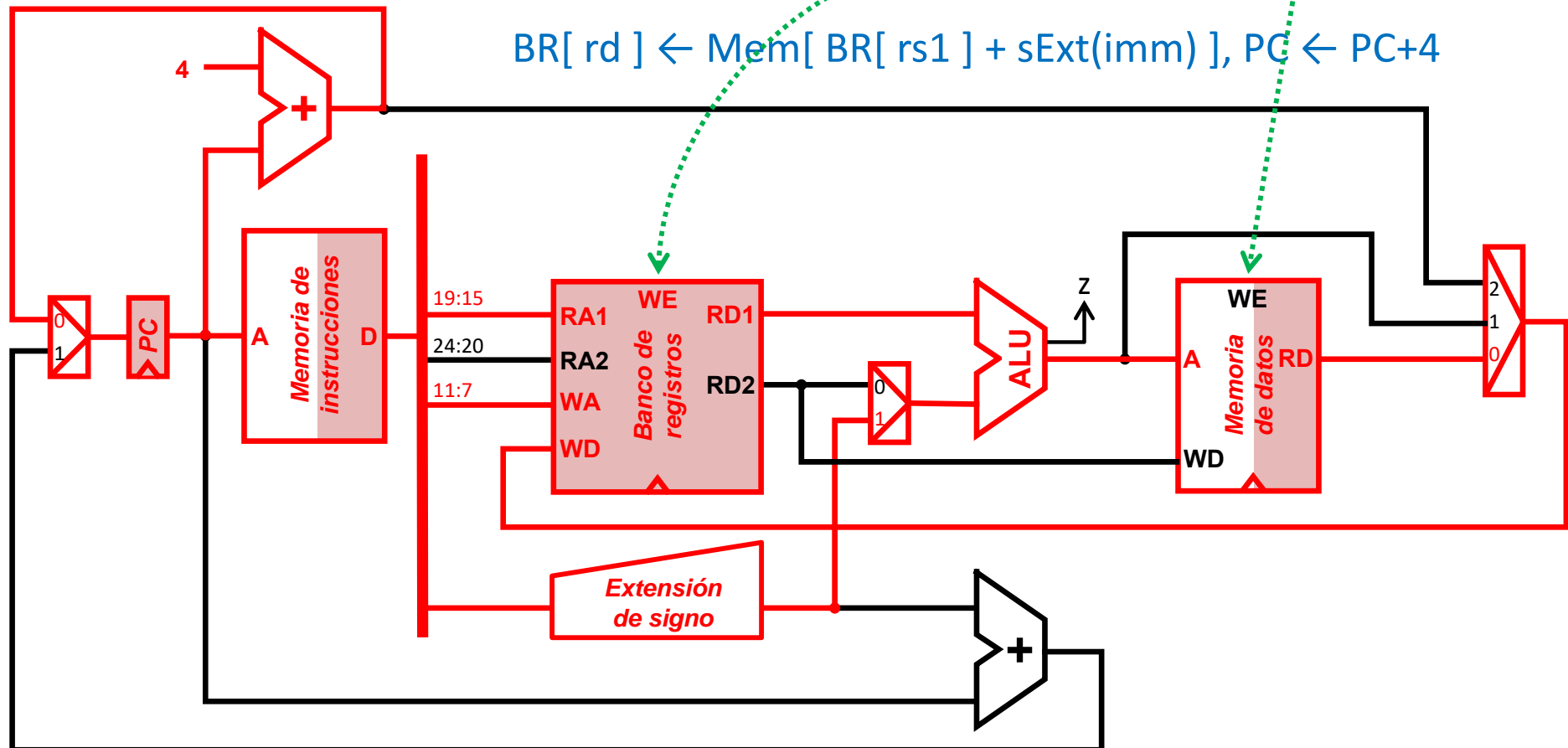
# Diseño del controlador

## Diseño del controlador: DEC principal

lw rd, imm(rs1)

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0000011 <sup>(lw)</sup>	0	0	1			0	

$BR[rd] \leftarrow Mem[BR[rs1] + sExt(imm)], PC \leftarrow PC + 4$





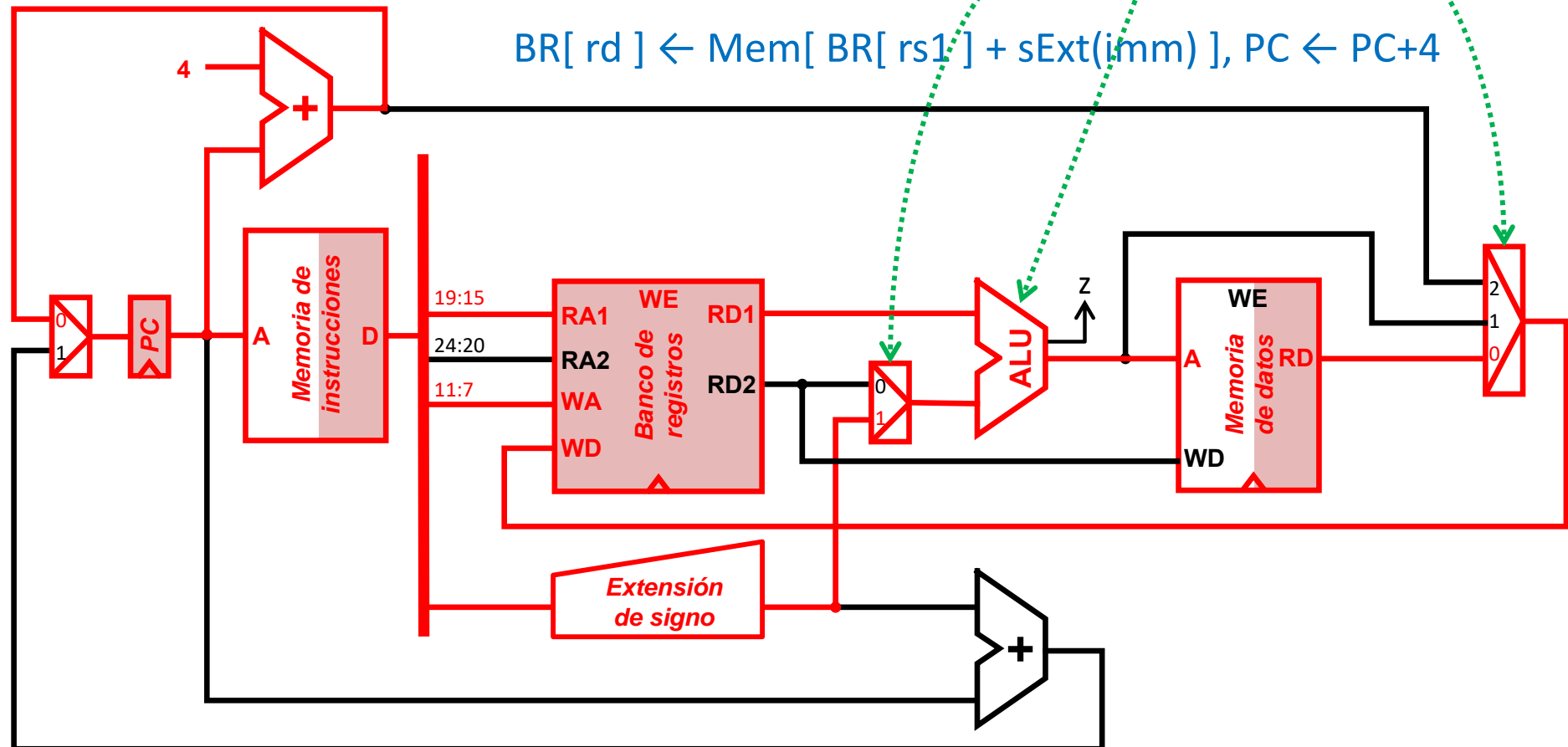
# Diseño del controlador

## Diseño del controlador: DEC principal

lw rd, imm(rs1)

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0000011 <sup>(lw)</sup>	0	0	1	1	00 <sup>(sumar)</sup>	0	00

$BR[rd] \leftarrow Mem[BR[rs1] + sExt(imm)], PC \leftarrow PC + 4$





# Diseño del controlador

## Diseño del controlador: DEC principal

- Este subcircuito gobierna el comportamiento general del procesador.

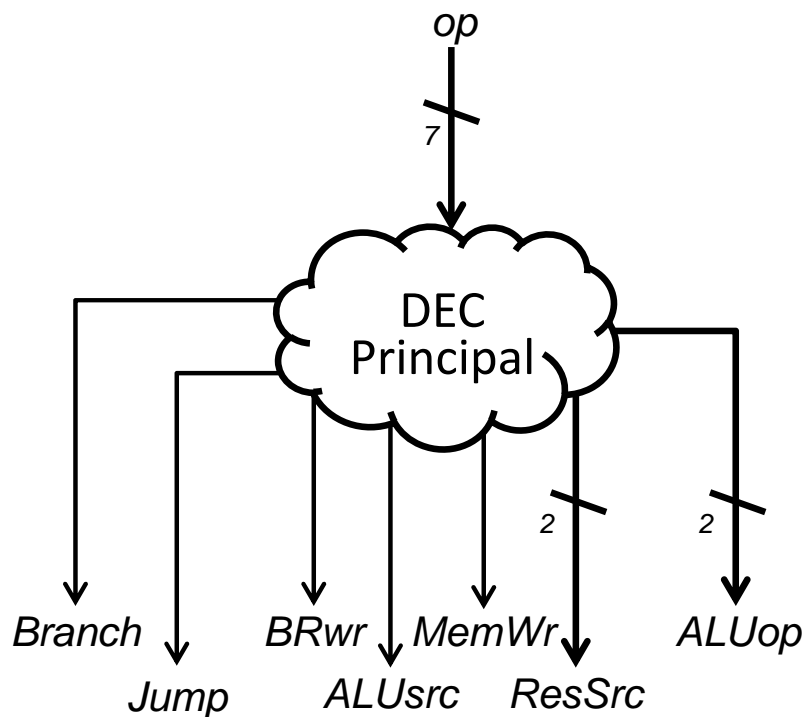


Tabla de verdad

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0000011 ( <i>lw</i> )	0	0	1	1	00 ( <i>sumar</i> )	0	00
0100011 ( <i>sw</i> )							
0010011 ( <i>tipo-l</i> )							
0110011 ( <i>tipo-R</i> )							
1100011 ( <i>beq</i> )							
1101111 ( <i>jal</i> )							



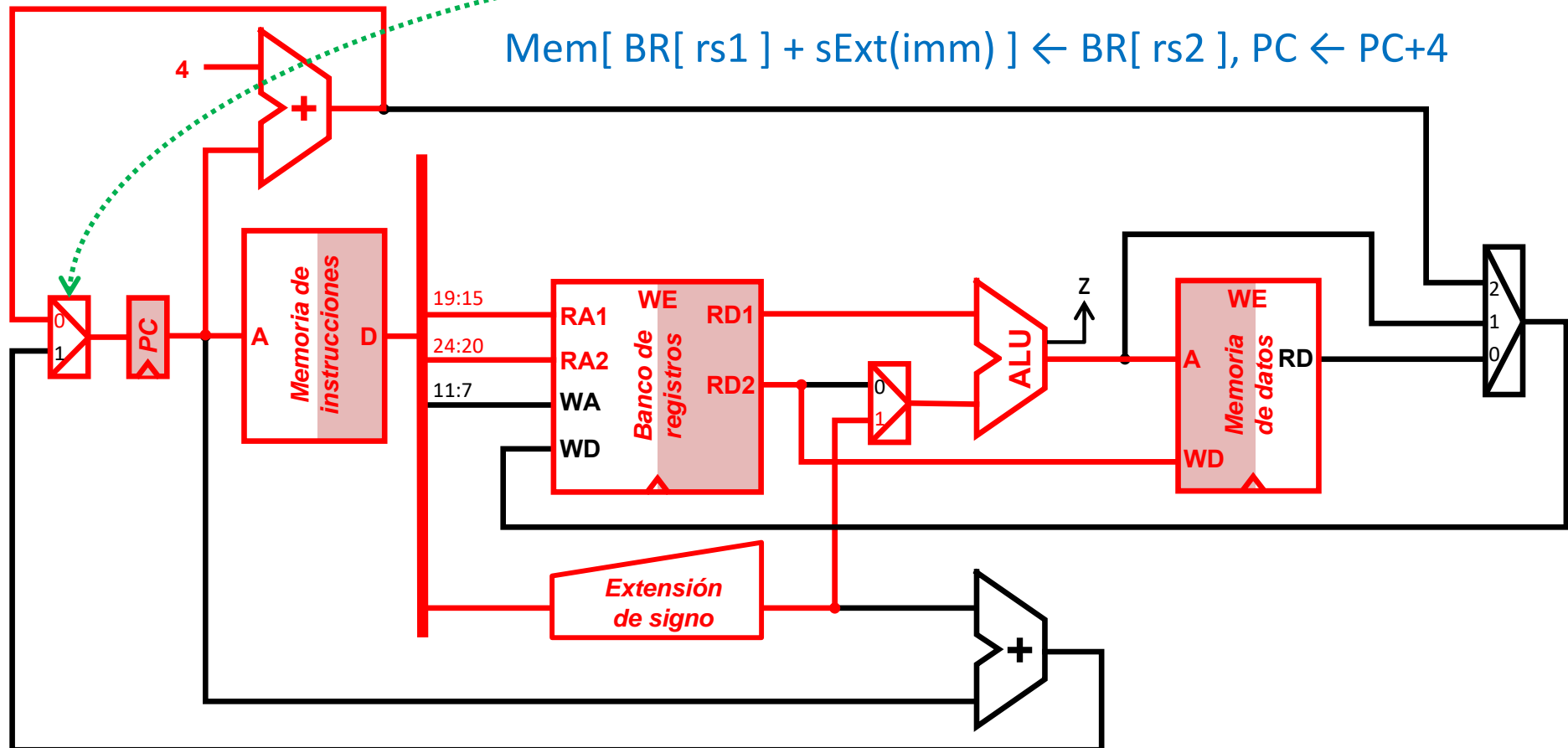
# Diseño del controlador

## Diseño del controlador: DEC principal

sw rs2, imm(rs1)

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0100011 <sup>(sw)</sup>	0	0					

$Mem[ BR[ rs1 ] + sExt(imm) ] \leftarrow BR[ rs2 ], PC \leftarrow PC+4$





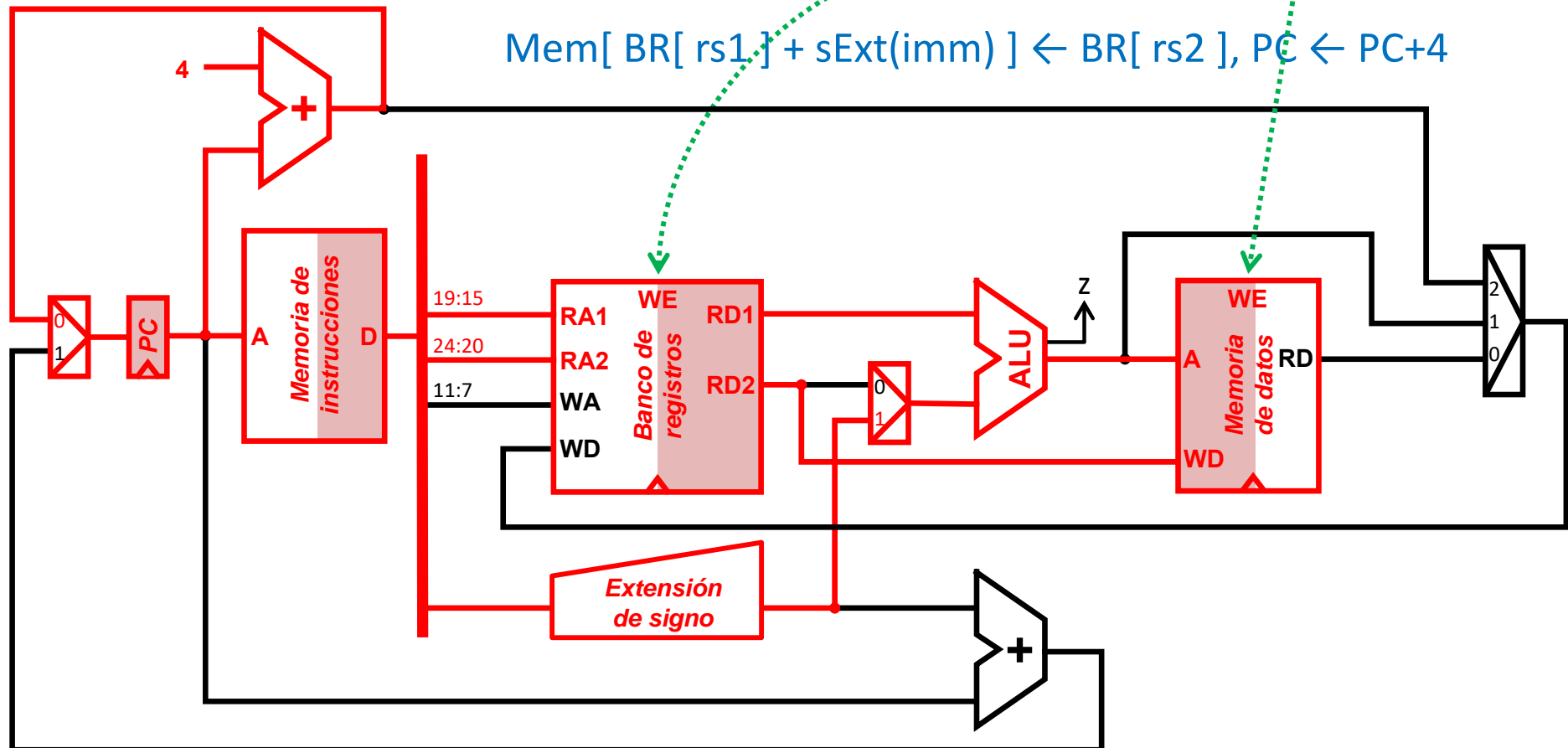
# Diseño del controlador

## Diseño del controlador: DEC principal

sw rs2, imm(rs1)

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0100011 (sw)	0	0	0			1	

$Mem[ BR[ rs1 ] + sExt(imm) ] \leftarrow BR[ rs2 ], PC \leftarrow PC+4$





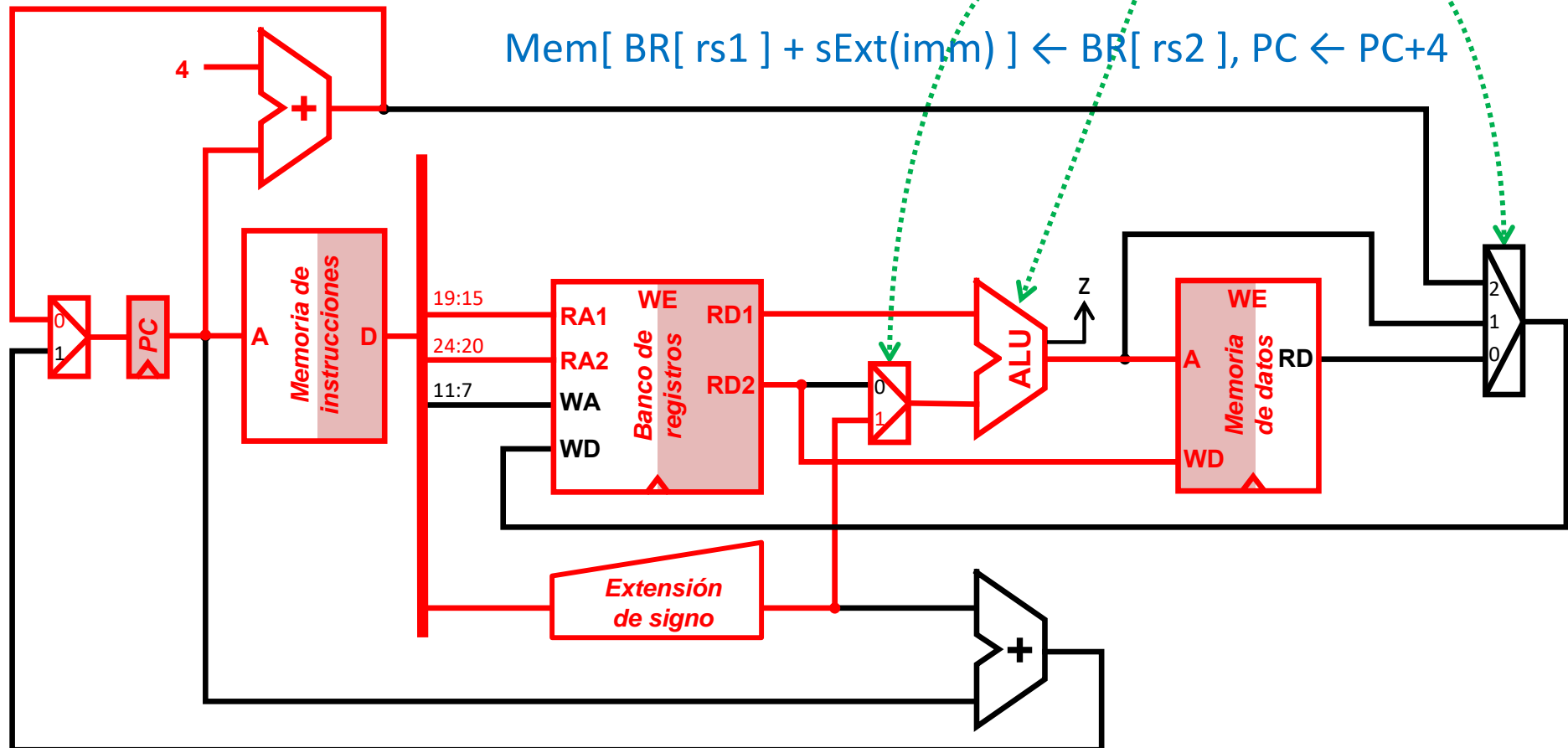
# Diseño del controlador

## Diseño del controlador: DEC principal

sw rs2, imm(rs1)

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0100011 <sup>(sw)</sup>	0	0	0	1	00 <sup>(sumar)</sup>	1	-

Mem[ BR[ rs1 ] + sExt(imm) ] ← BR[ rs2 ], PC ← PC+4





# Diseño del controlador

## Diseño del controlador: DEC principal

- Este subcircuito gobierna el comportamiento general del procesador.

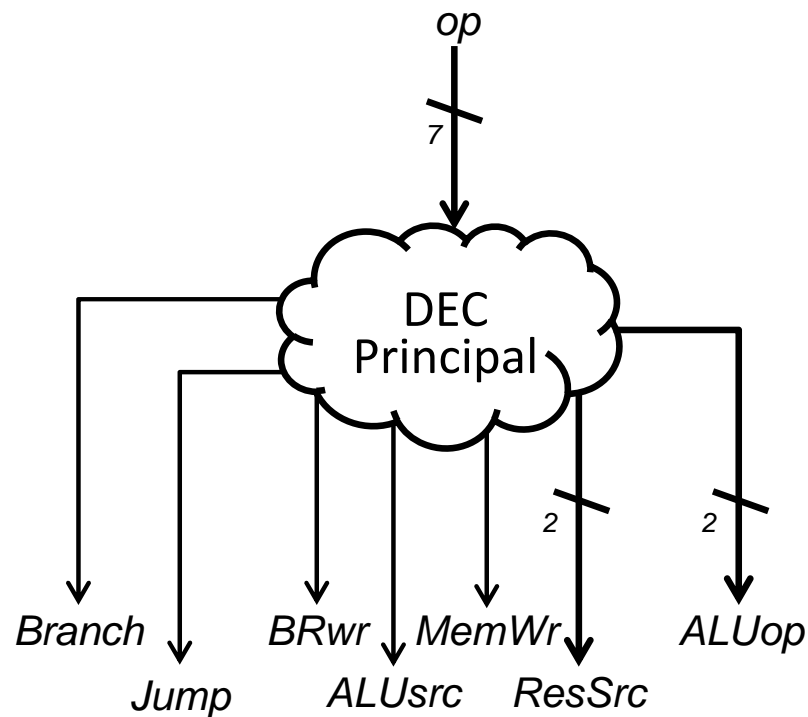


Tabla de verdad

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0000011 ( <i>lw</i> )	0	0	1	1	00 ( <i>sumar</i> )	0	00
0100011 ( <i>sw</i> )	0	0	0	1	00 ( <i>sumar</i> )	1	-
0010011 ( <i>tipo-I</i> )							
0110011 ( <i>tipo-R</i> )							
1100011 ( <i>beq</i> )							
1101111 ( <i>jal</i> )							



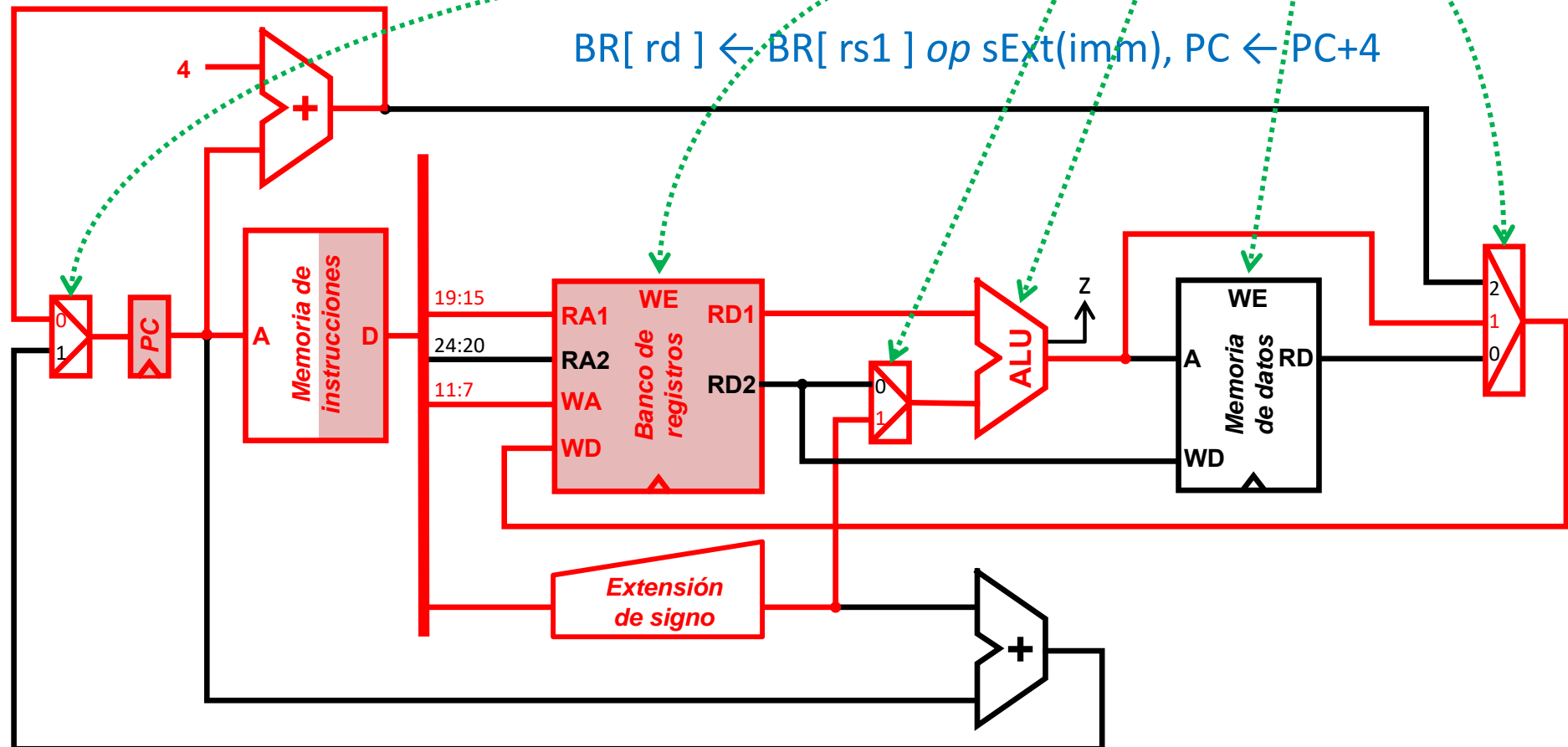
# Diseño del controlador

## Diseño del controlador: DEC principal

addi rd, rs1, imm

op	Branch	Jump	BRwr	ALUsrc	ALUOp	MemWr	ResSrc
0010011 (tipo-I)	0	0	1	1	10 (operar)	0	01

$$BR[rd] \leftarrow BR[rs1] \text{ op } sExt(imm), PC \leftarrow PC+4$$







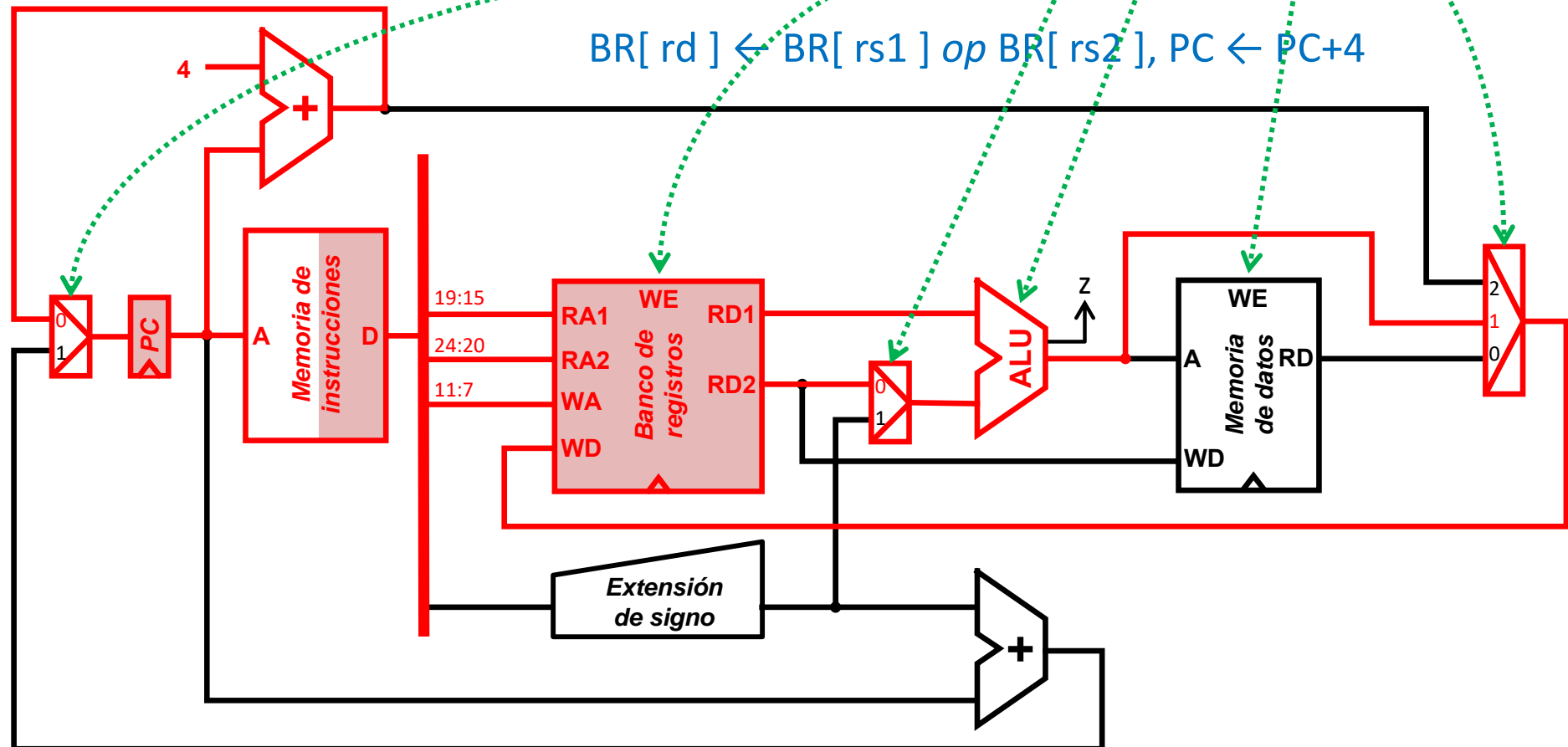
# Diseño del controlador

## Diseño del controlador: DEC principal

add rd, rs1, rs2

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0110011 (tipo-R)	0	0	1	0	10 (operar)	0	01

$BR[rd] \leftarrow BR[rs1] \text{ op } BR[rs2], PC \leftarrow PC+4$





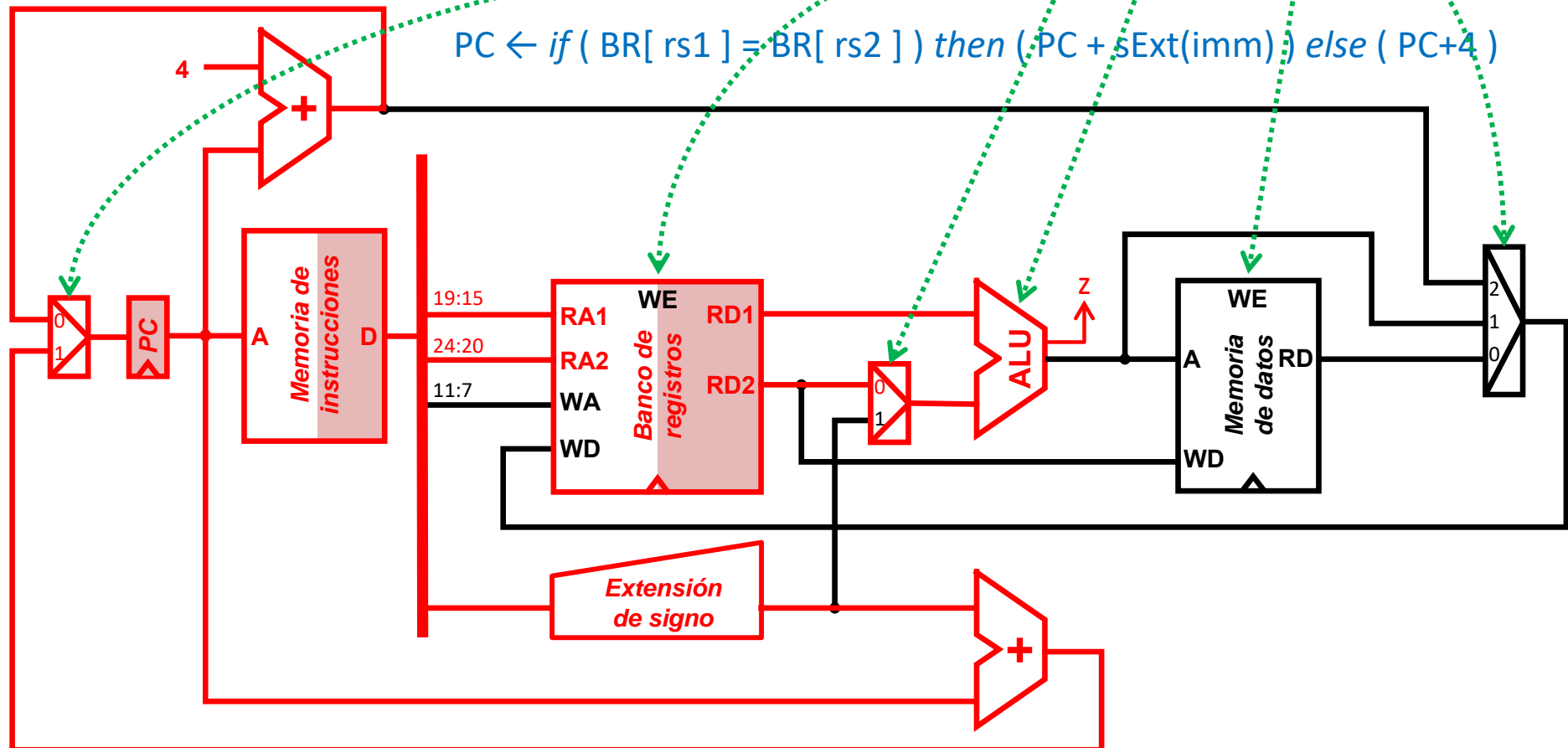
# Diseño del controlador

## Diseño del controlador: DEC principal

beq rs1,rs2,imm

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
1100011 (beq)	1	0	0	0	01 (restar)	0	-

$PC \leftarrow \text{if} ( BR[ rs1 ] = BR[ rs2 ] ) \text{ then } ( PC + sExt(imm) ) \text{ else } ( PC+4 )$





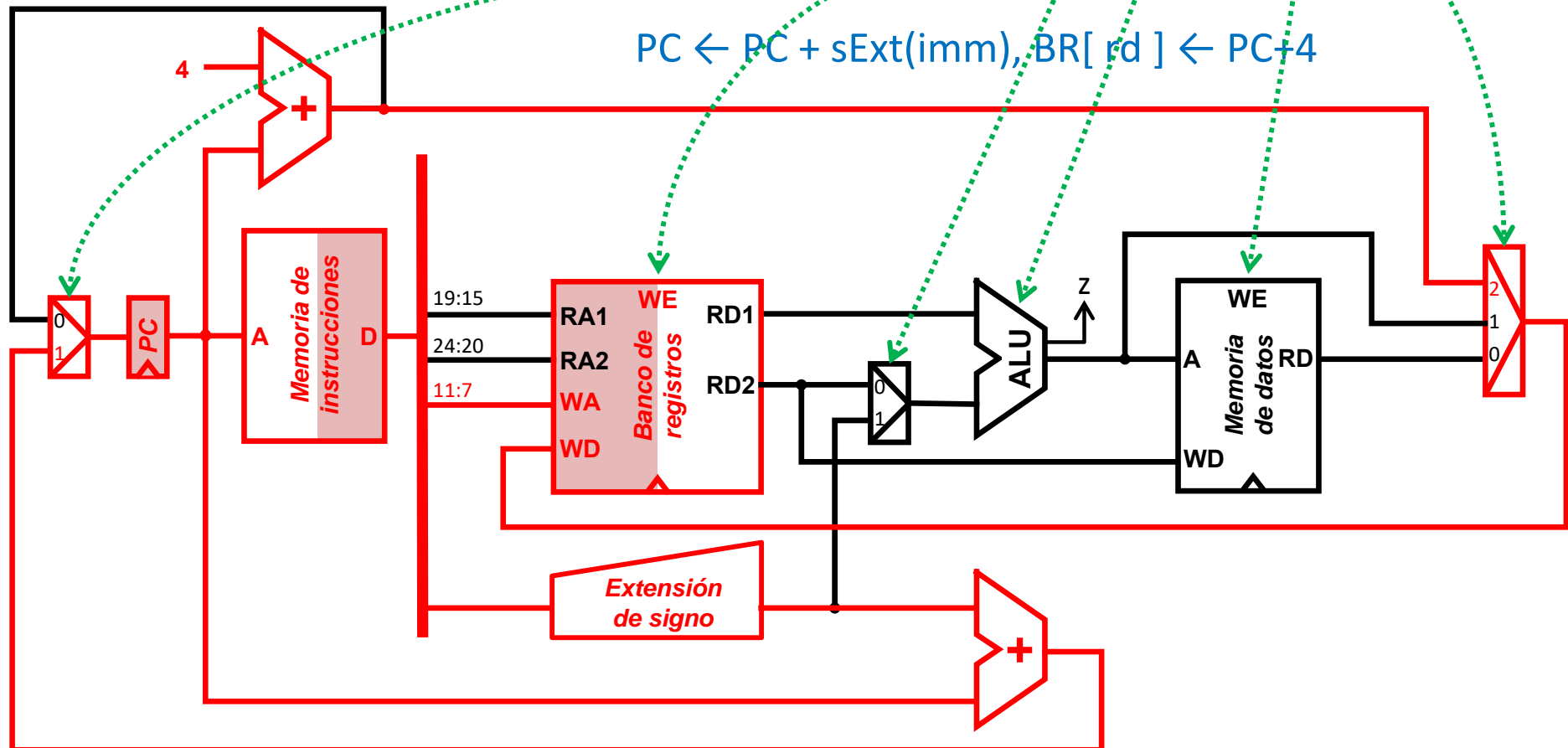
# Diseño del controlador

## Diseño del controlador: DEC principal

jal rd, imm

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
1101111 (jal)	0	1	1	-	-	0	10

$PC \leftarrow PC + sExt(imm), BR[rd] \leftarrow PC+4$





# Diseño del controlador

## Diseño del controlador: DEC principal

- Este subcircuito gobierna el comportamiento general del procesador.

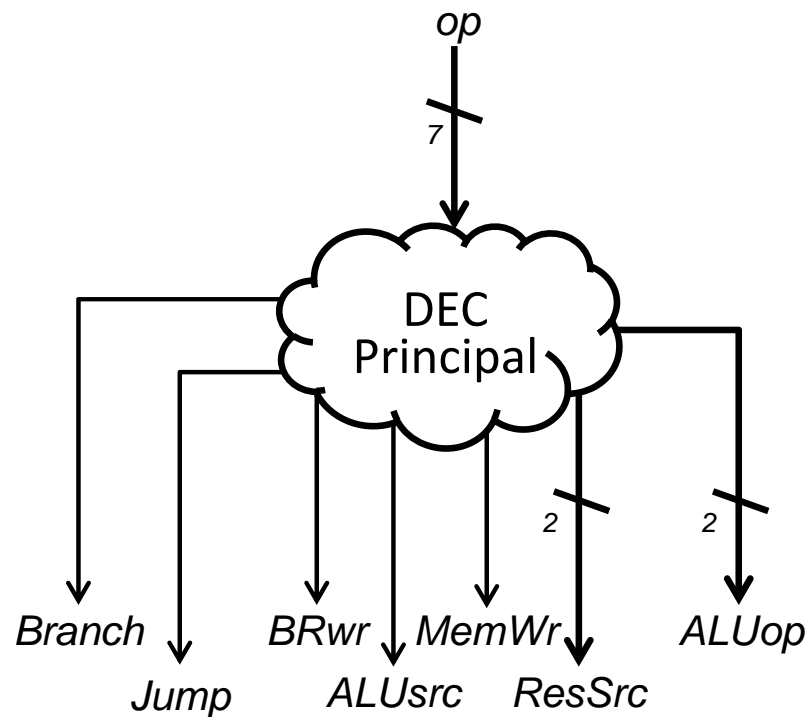


Tabla de verdad

op	Branch	Jump	BRwr	ALUsrc	ALUOp	MemWr	ResSrc
0000011 ( <sup>lw</sup> )	0	0	1	1	00 (sumar)	0	00
0100011 ( <sup>sw</sup> )	0	0	0	1	00 (sumar)	1	-
0010011 (tipo-I)	0	0	1	1	10 (operar)	0	01
0110011 (tipo-R)	0	0	1	0	10 (operar)	0	01
1100011 ( <sup>beq</sup> )	1	0	0	0	01 (restar)	0	-
1101111 ( <sup>jal</sup> )	0	1	1	-	-	0	10



# Diseño del controlador

## Diseño del controlador: DEC principal

- Este subcircuito gobierna el comportamiento general del procesador.

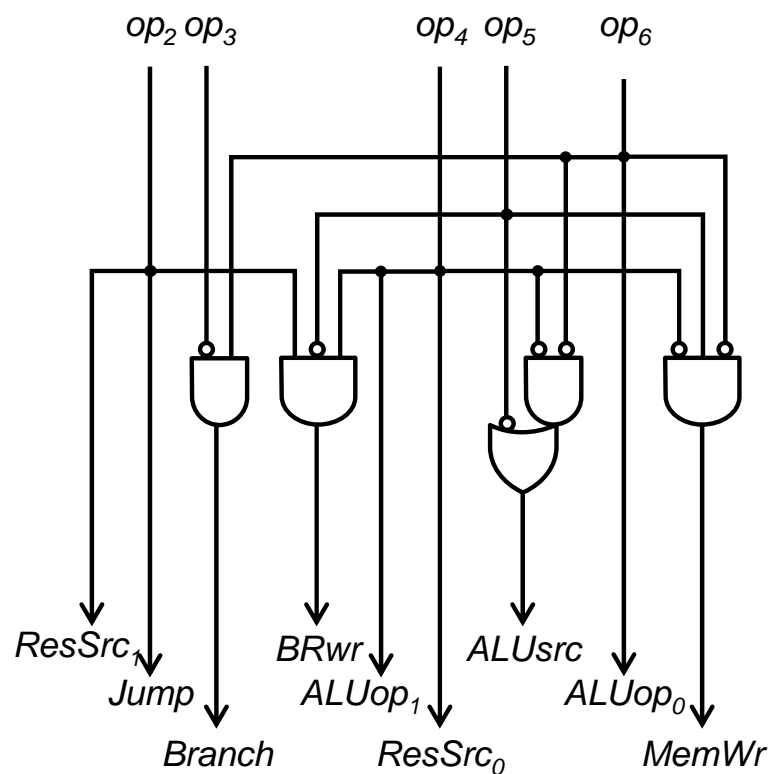


Tabla de verdad

op	Branch	Jump	BRwr	ALUsrc	ALUop	MemWr	ResSrc
0000011 ( <i>lw</i> )	0	0	1	1	00 (sumar)	0	00
0100011 ( <i>sw</i> )	0	0	0	1	00 (sumar)	1	-
0010011 (tipo-I)	0	0	1	1	10 (operar)	0	01
0110011 (tipo-R)	0	0	1	0	10 (operar)	0	01
1100011 ( <i>beq</i> )	1	0	0	0	01 (restar)	0	-
1101111 ( <i>jal</i> )	0	1	1	-	-	0	10



# Procesador monociclo

## Simulación (i)

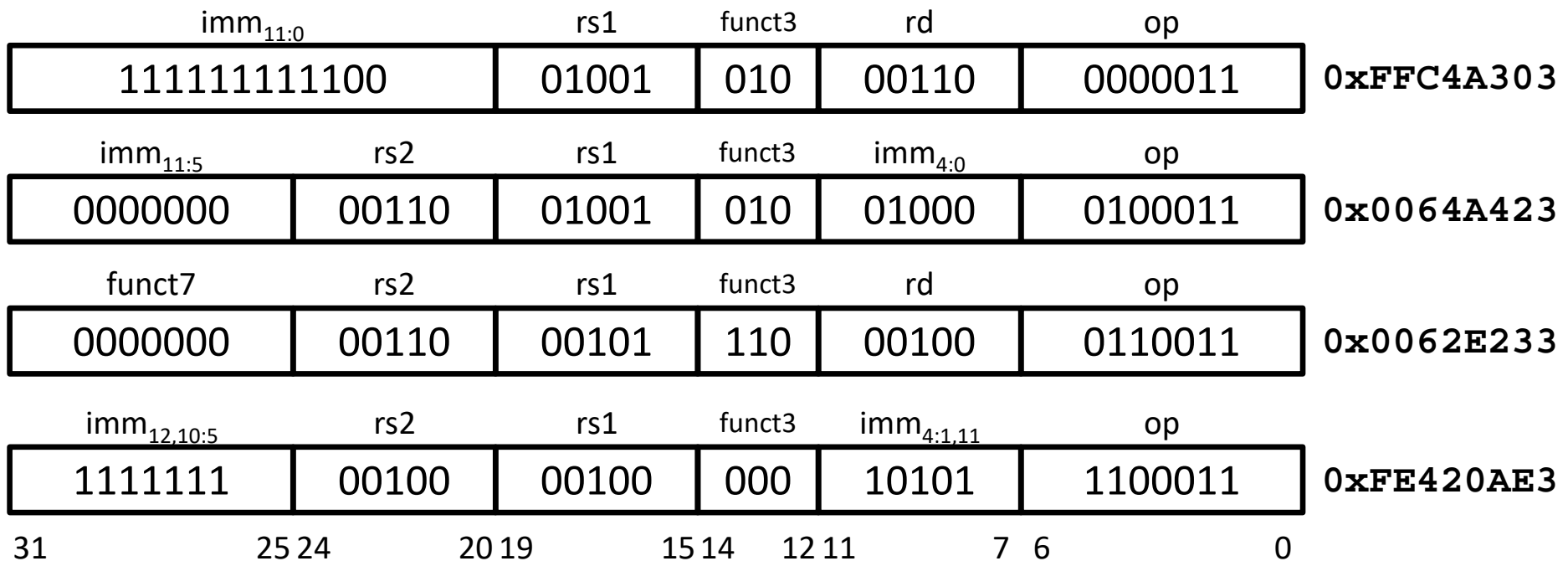
```

...
L7:
0x1000  lw  x6, -4(x9)
0x1004  sw  x6, 8(x9)
0x1008  or  x4, x5, x6
0x100C  beq x4, x4, L7
...
    
```

```

...
0xFFC4A303
0x0064A423
0x0062E233
0xFE420AE3
...
    
```

x5	6
x9	0x2004
Mem[0x2000]	10
PC	0x1000







# Procesador monociclo

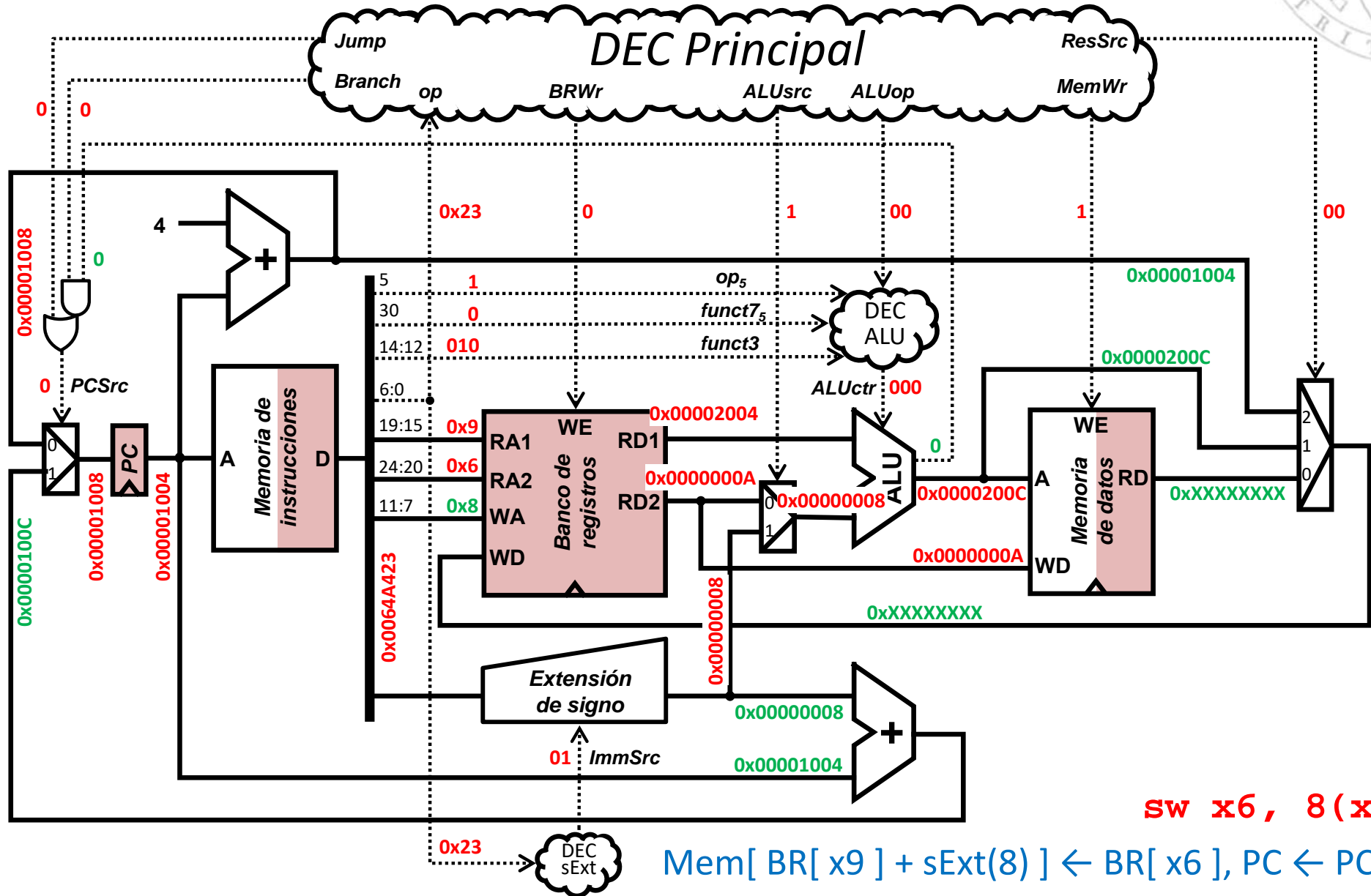
## Simulación: 2do. ciclo

versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

72



`sw x6, 8(x9)`

$Mem[ BR[ x9 ] + sExt(8) ] \leftarrow BR[ x6 ], PC \leftarrow PC+4$





# Procesador monociclo

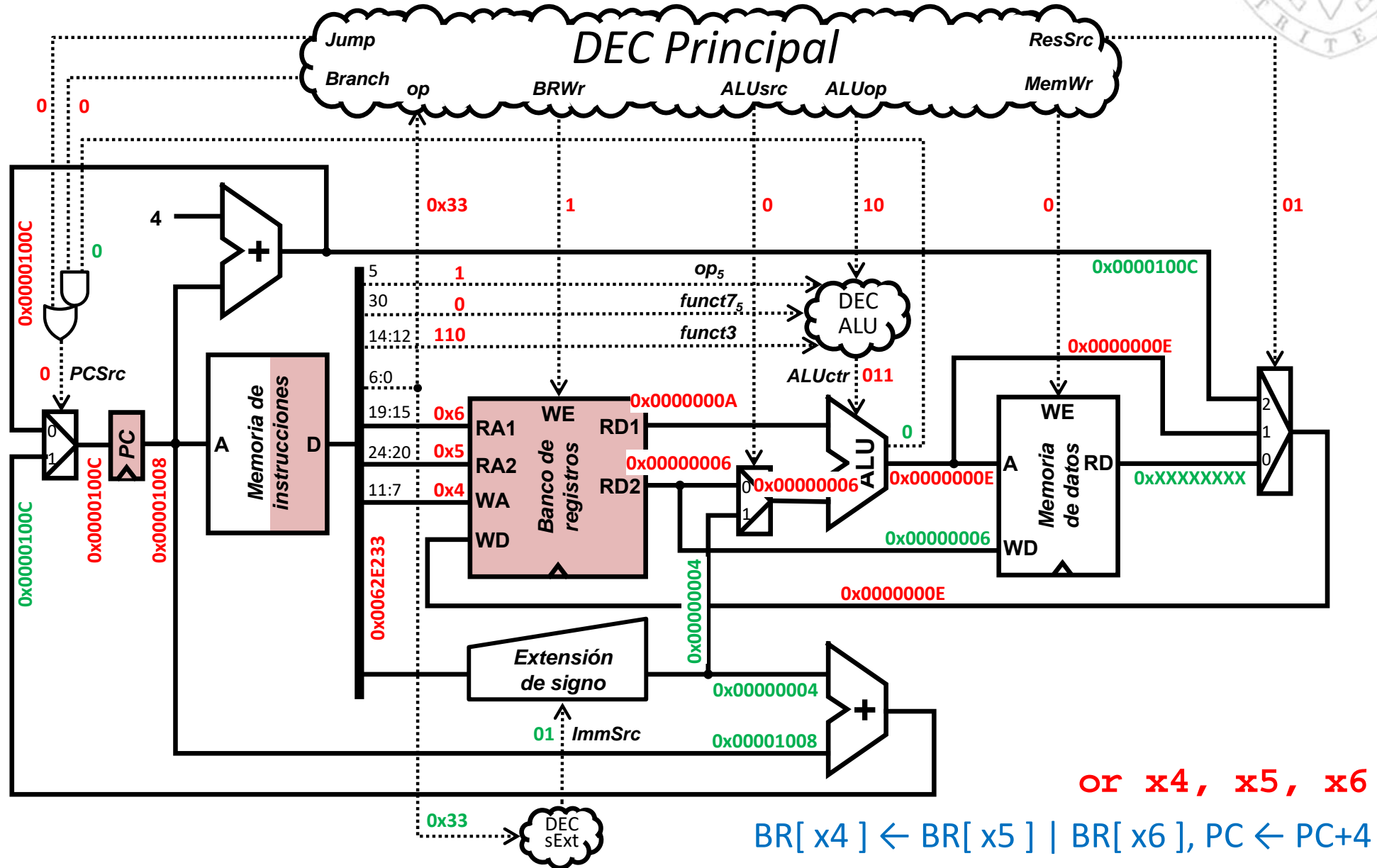
## Simulación: 3er. ciclo

versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

73



# Procesador monociclo

Simulación: 4o. ciclo

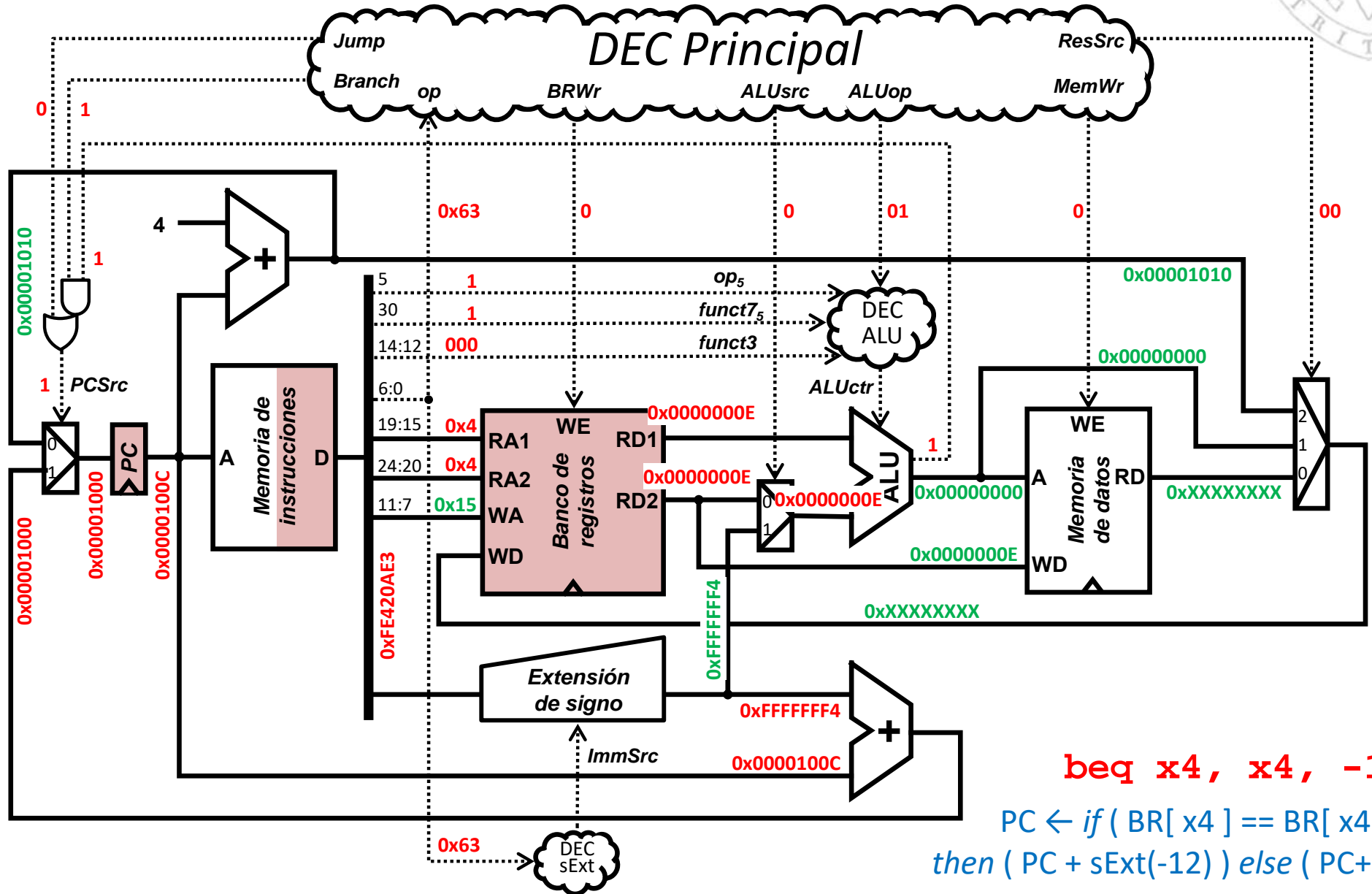


versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

74



**beq x4, x4, -12**

$PC \leftarrow \text{if} ( BR[ x4 ] == BR[ x4 ] )$   
 $\text{then} ( PC + \text{sExt}(-12) ) \text{ else} ( PC+4 )$



# Procesador monociclo

## Coste y tiempo de ciclo (CMOS 90 nm)

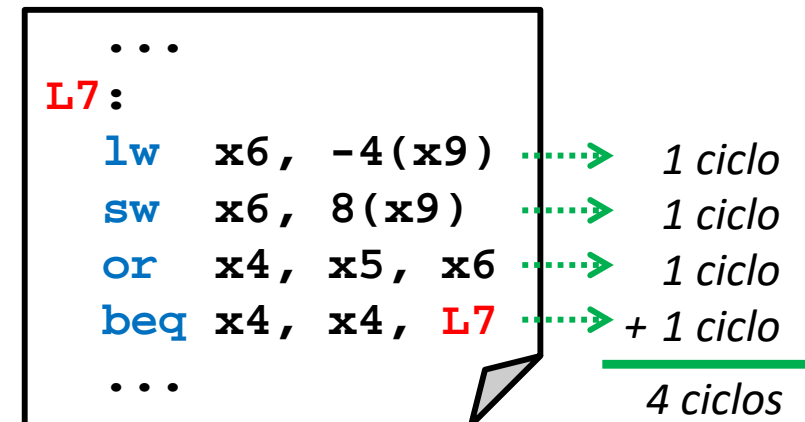
versión 31/10/23

$$area = 59181 \mu m^2$$

$$t_{clk} = 27.6 ns$$

$$f_{clk} = \frac{1}{t_{clk}} = \frac{1}{27.6 \cdot 10^{-9}s} = 36.2 MHz$$

transferencia entre registros	instr.	camino crítico
PC $\leftarrow$ PC+4	varias	9692 ps
BR[ rd ] $\leftarrow$ Mem[ BR[ rs1 ] + sExt(imm) ]	lw	<b>27616 ps</b>
Mem[ BR[ rs1 ] + sExt(imm) ] $\leftarrow$ BR[ rs2 ]	sw	26661 ps
BR[ rd ] $\leftarrow$ BR[ rs1 ] op sExt(imm)	tipo I	19116 ps
BR[ rd ] $\leftarrow$ BR[ rs1 ] op BR[ rs2 ]	tipo R	18928 ps
PC $\leftarrow$ if ( BR[ rs1 ] = BR[ rs2 ] ) then ( PC + sExt(imm) ) else ( PC+4 )	beq	18547 ps
BR[ rd ] $\leftarrow$ PC+4	jal	10073 ps
PC $\leftarrow$ PC + sExt(imm)		17033 ps
	<b>max.</b>	<b>27616 ps</b>



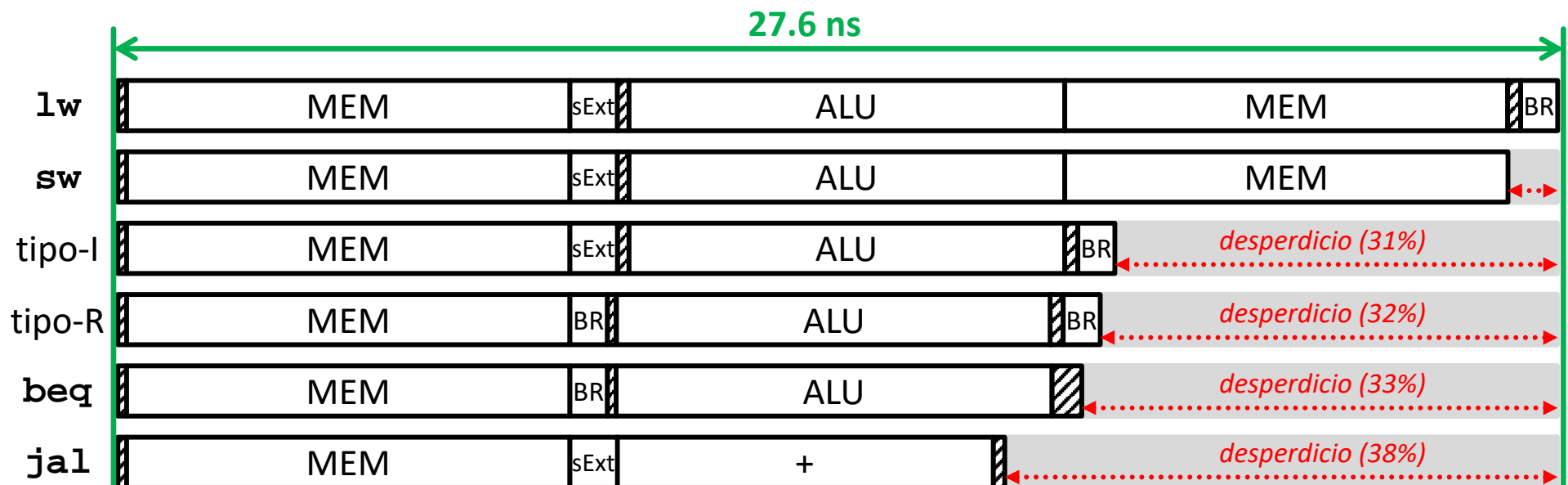
$$t_{ejec} = 4 \times 27.6 ns = 110.4 ns$$



# Procesador monociclo

## Conclusiones

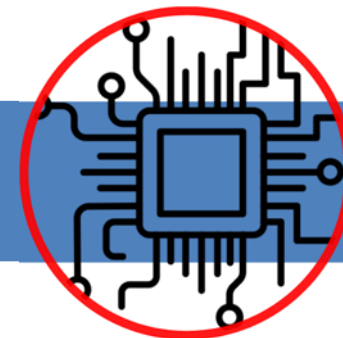
- La **implementación monociclo** tiene algunos **problemas**:
  - El **tiempo de ciclo** viene dado por la **instrucción más lenta**
    - Todas las instrucciones tardan lo mismo en ejecutarse con independencia de su complejidad: se **desperdicia tiempo** en la ejecución de instrucciones rápidas.
    - En **repertorios reales** existen algunas **instrucciones muy largas**: memorias lentas, operaciones aritméticas complejas, modos de direccionamiento complejos...
  - **No es posible reutilizar hardware**:
    - Requiere una ALU y 2 sumadores, memoria de instrucciones y datos separada...





- Diseño del Banco de Registros.
- Diseño de la Memoria.
- Diseño de la ALU.
- Diseño del Extensor de Signo.
- Cálculo del coste.
- Cálculo del tiempo de ciclo.

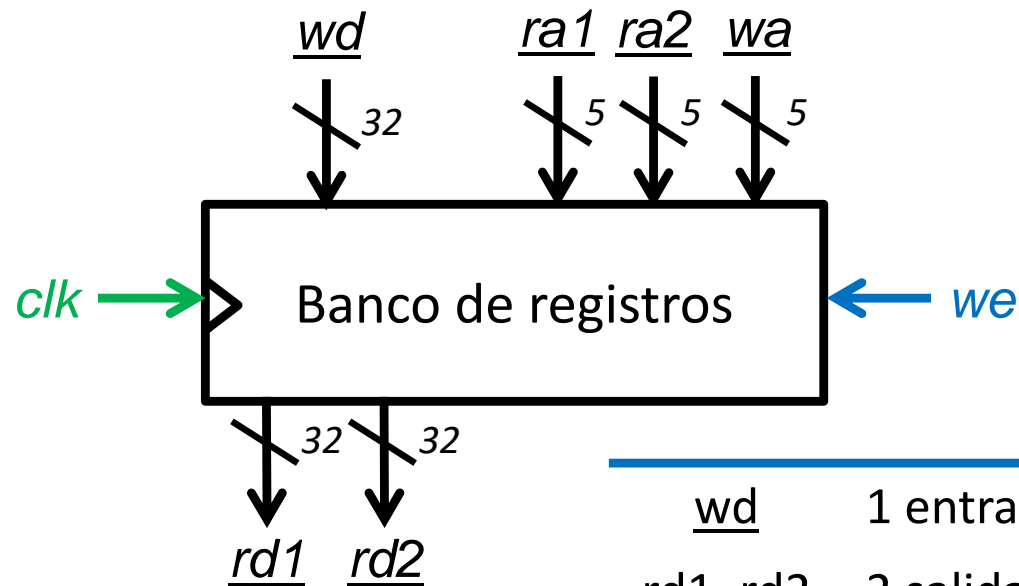
## Apéndice tecnológico





# Diseño del Banco de Registros

- El Banco de Registros contiene 32 registros de datos de 32 bits.



---

<u>wd</u>	1 entrada de datos de 32 bits
<u>rd1</u> , <u>rd2</u>	2 salidas de datos de 32 bits
<u>wa</u>	1 entrada de dirección de escritura de 5 bits
<u>ra1</u> , <u>ra2</u>	2 entradas de dirección de lectura de 5 bits
<u>we</u>	1 entrada de capacitación de escritura
<u>clk</u>	1 entrada de reloj

---



# Diseño del Banco de Registros



versión 31/10/23

tema 5:  
Diseño monociclo del procesador

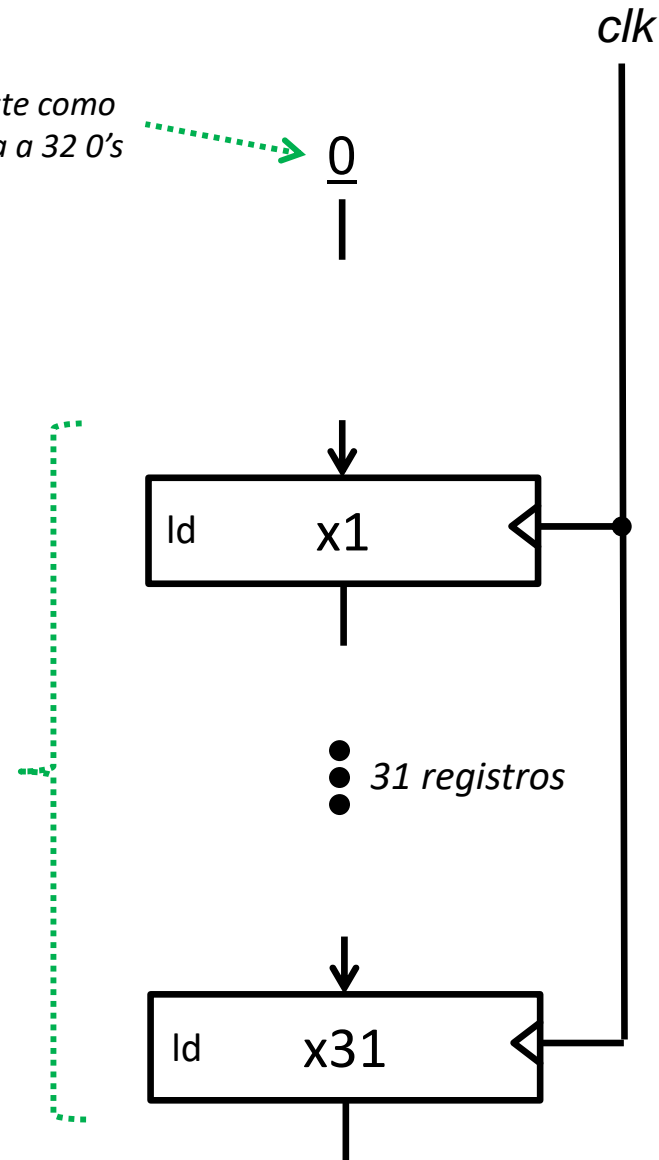
FC-2

79

El registro x0 (zero) no existe como tal es una conexión directa a 32 0's

0

Los restantes 31 registros (x0-x31) son registros convencionales





# Diseño del Banco de Registros

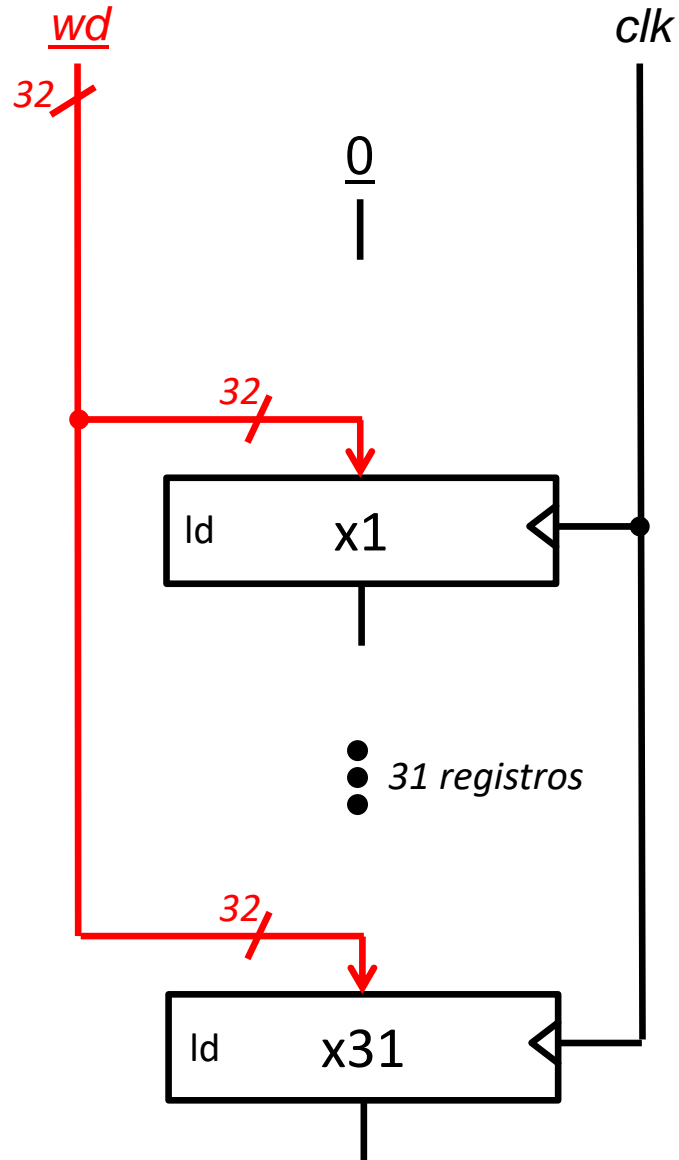


versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

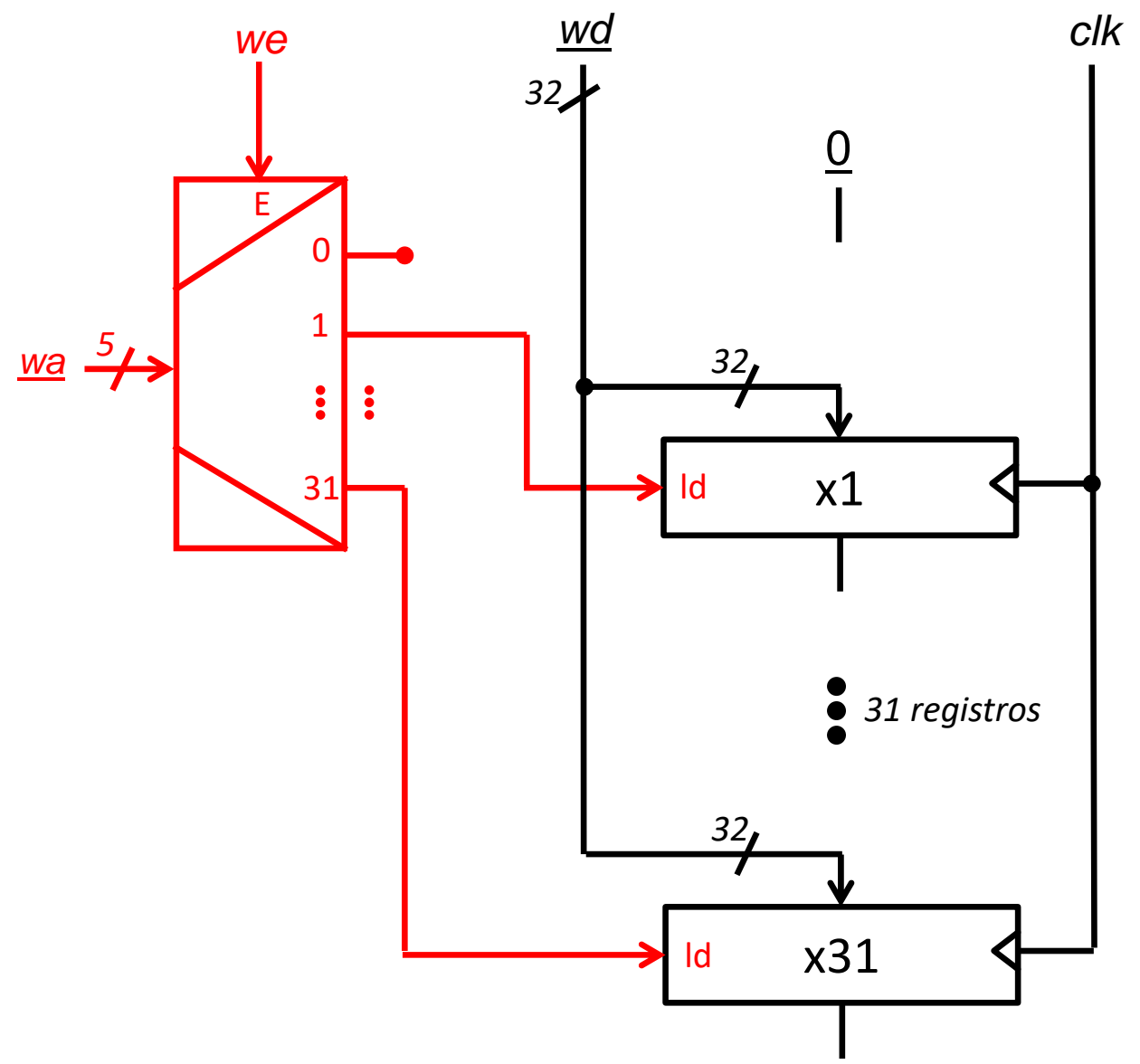
80





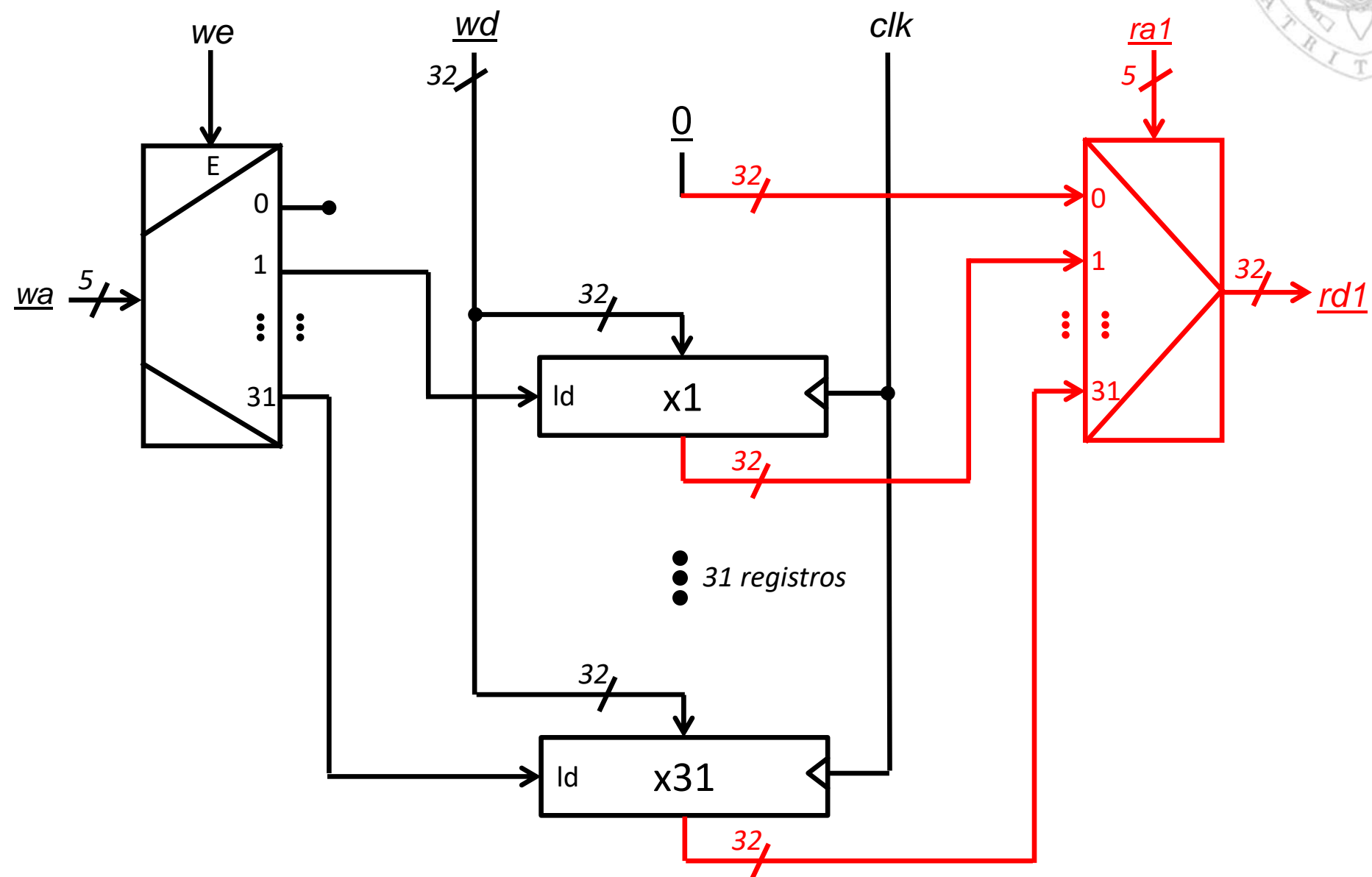


# Diseño del Banco de Registros





# Diseño del Banco de Registros





# Diseño del Banco de Registros

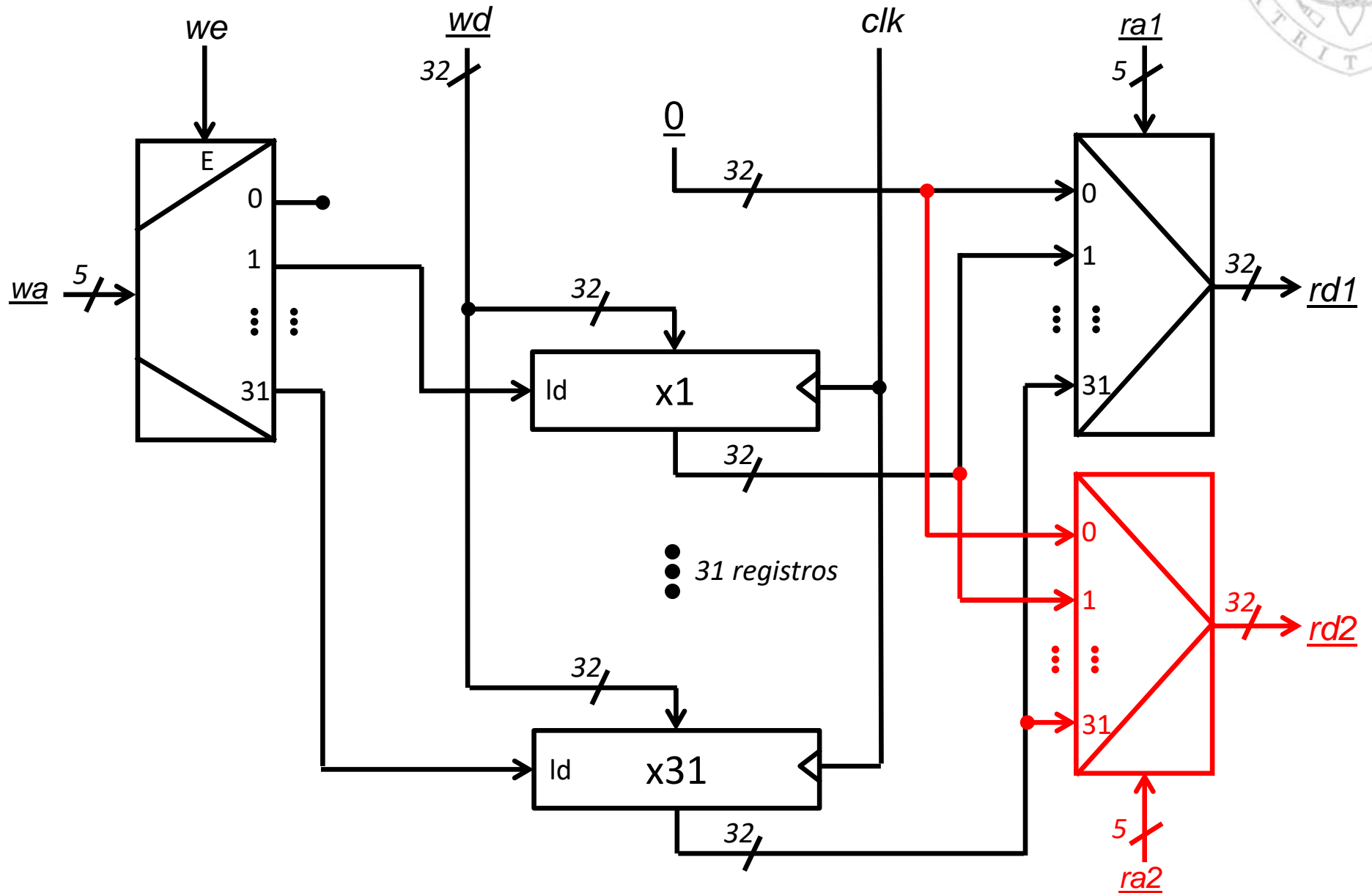


versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

83





# Diseño del Banco de Registros

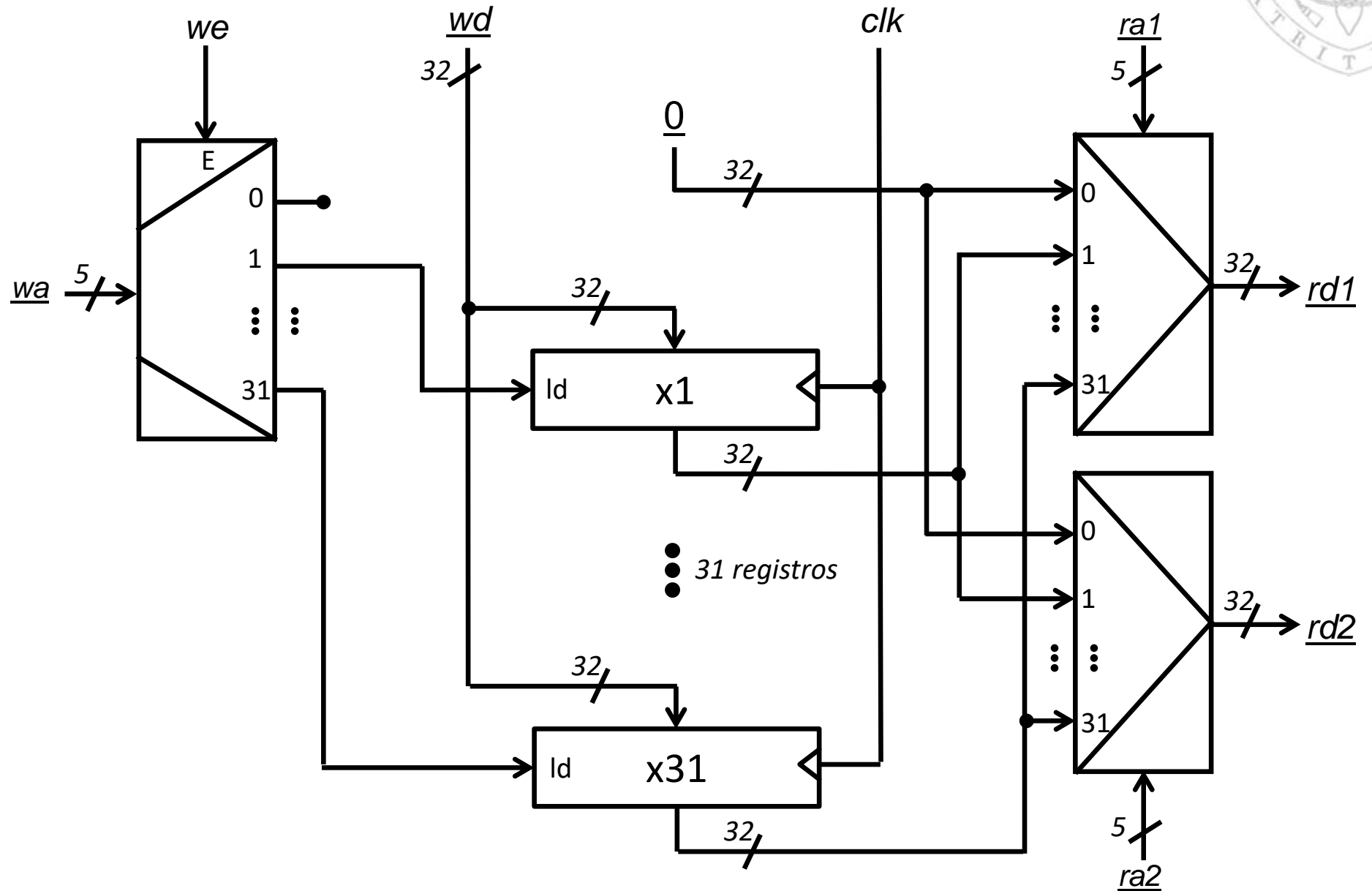


versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

84





# Diseño de la Memoria

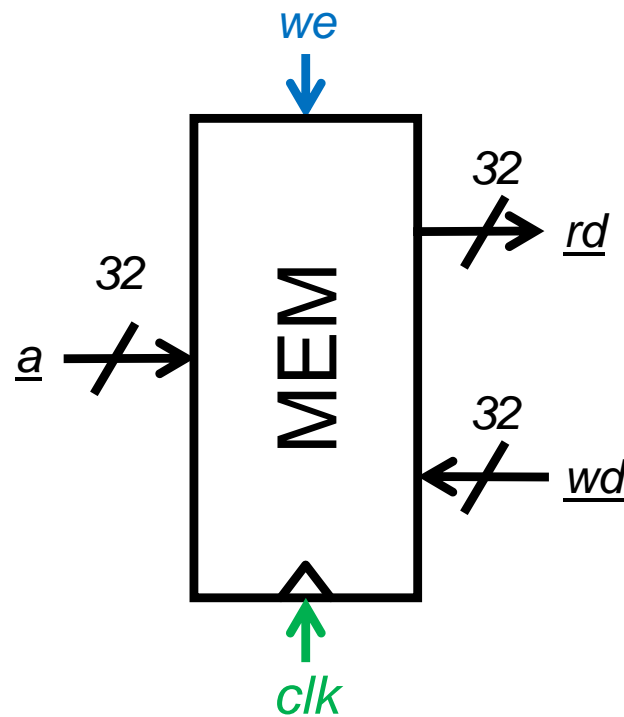


versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

85



RAM  $2^{32} \times 8 = 2^{30} \times 32$   
( $2^{30}$  palabras de 32 bits)

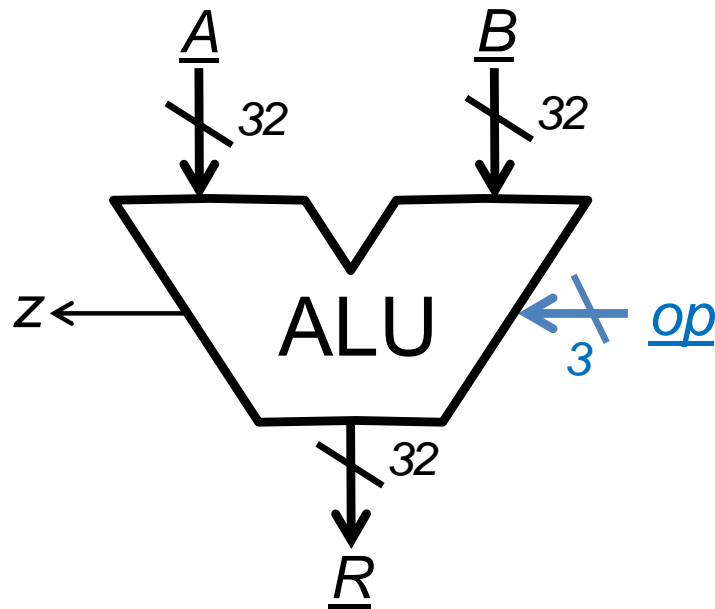
- 
- wd 1 entrada de datos de 32 bits
  - rd 1 salidas de datos de 32 bits
  - a 1 entrada de dirección de 32 bits
  - we 1 entrada de capacitación de escritura
  - clk 1 entrada de reloj
- 

*Memoria con doble puerto de datos,  
direccionable por bytes  
y con ordenamiento little-endian*





# Diseño de la ALU



---

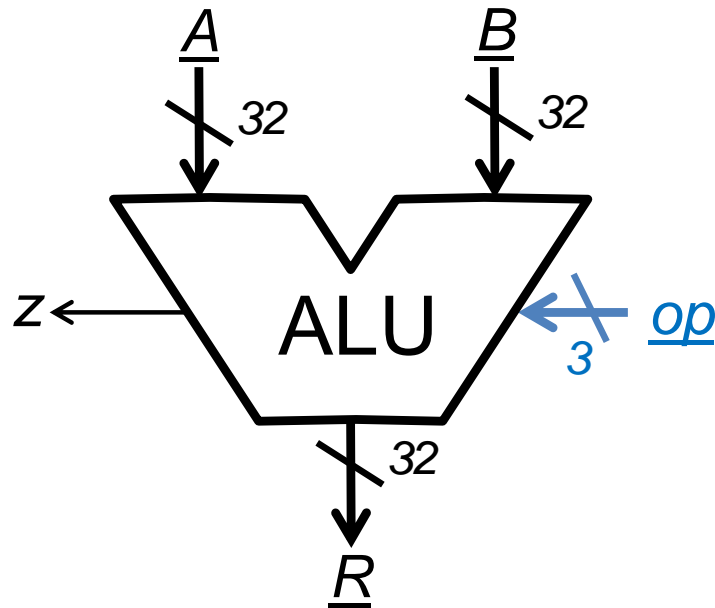
$\underline{A}, \underline{B}$	2 entradas de datos de 32 bits
$\underline{op}$	1 entrada de selección de operación
$\underline{R}$	1 salida de datos de 32 bits
$z$	1 salida de detección de cero

---

- La **ALU** es un módulo combinacional que realiza:
  - El **cálculo de direcciones efectivas** de memoria en instrucciones **lw/sw**.
  - Todas las **operaciones aritmético-lógicas** de instrucciones tipo I / tipo R.
  - La **comparación de operandos** en la instrucción **beq**.
  - En la **ruta de datos multiciclo**, también se usará para **incrementar el PC** y realizar el **cálculo de direcciones de salto** en instrucciones **beq/jal**



# Diseño de la ALU



- A, B 2 entradas de datos de 32 bits
- op 1 entrada de selección de operación
- R 1 salida de datos de 32 bits
- z 1 salida de detección de cero

## operaciones aritméticas

<u>op</u> <sub>2</sub>	<u>op</u> <sub>1</sub>	<u>op</u> <sub>0</sub>	<u>R</u>
0	0	0	<u>A</u> + <u>B</u>
0	0	1	<u>A</u> - <u>B</u>
1	0	0	-
1	0	1	if ( <u>A</u> < <u>B</u> ) then 1 else 0

## operaciones lógicas

<u>op</u> <sub>2</sub>	<u>op</u> <sub>1</sub>	<u>op</u> <sub>0</sub>	<u>R</u>
0	1	0	<u>A</u> & <u>B</u>
0	1	1	<u>A</u>   <u>B</u>
1	1	0	-
1	1	1	-

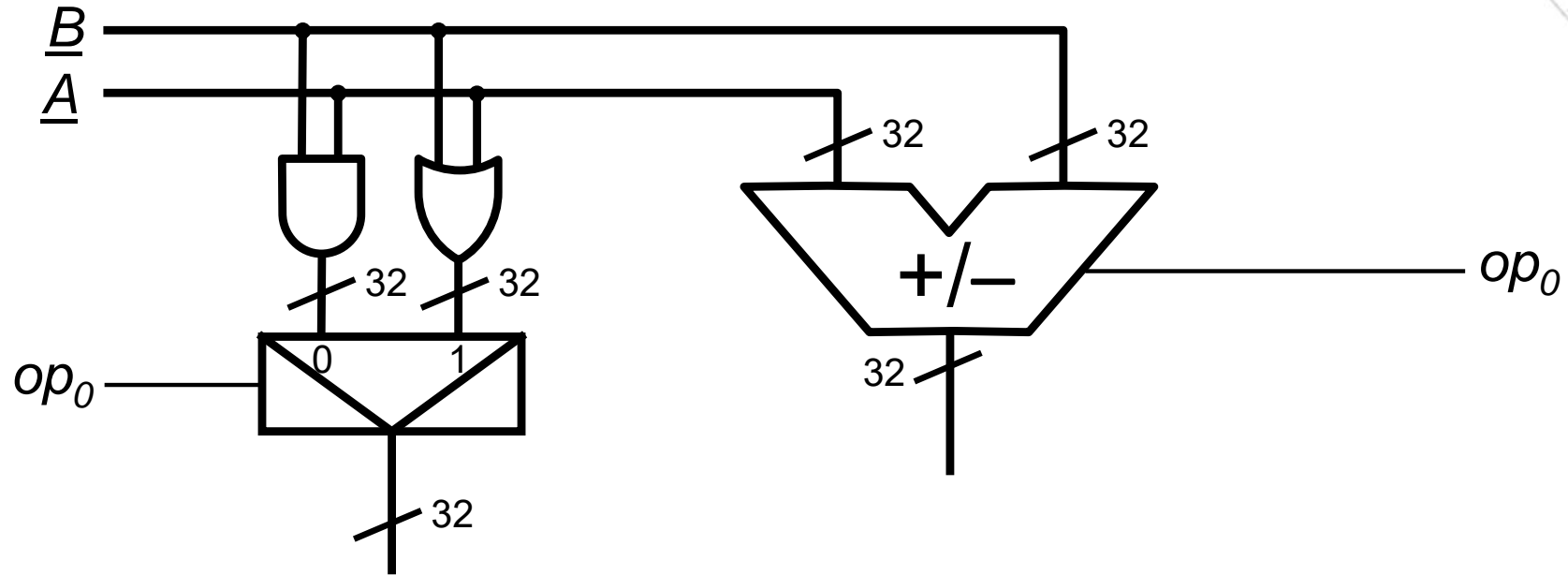




# Diseño de la ALU



versión 31/10/23



operaciones aritméticas

$op_2$	$op_1$	$op_0$	$\underline{R}$
0	0	0	$\underline{A} + \underline{B}$
0	0	1	$\underline{A} - \underline{B}$
1	0	0	—
1	0	1	if ( $\underline{A} < \underline{B}$ ) then 1 else 0

operaciones lógicas

$op_2$	$op_1$	$op_0$	$\underline{R}$
0	1	0	$\underline{A} \& \underline{B}$
0	1	1	$\underline{A}   \underline{B}$
1	1	0	—
1	1	1	—





# Diseño de la ALU

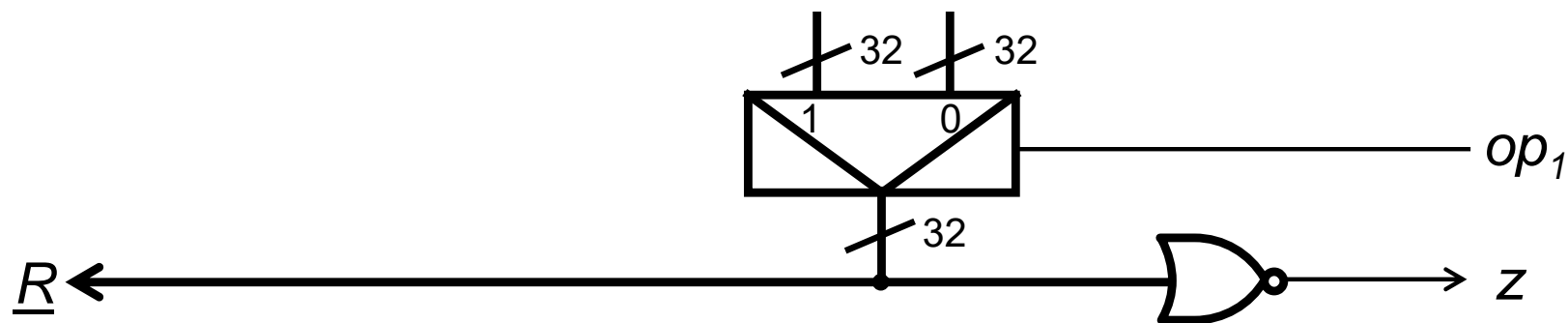


operaciones aritméticas

op <sub>2</sub>	op <sub>1</sub>	op <sub>0</sub>	<u>R</u>
0	0	0	$\underline{A} + \underline{B}$
0	0	1	$\underline{A} - \underline{B}$
1	0	0	-
1	0	1	if ( $\underline{A} < \underline{B}$ ) then 1 else 0

operaciones lógicas

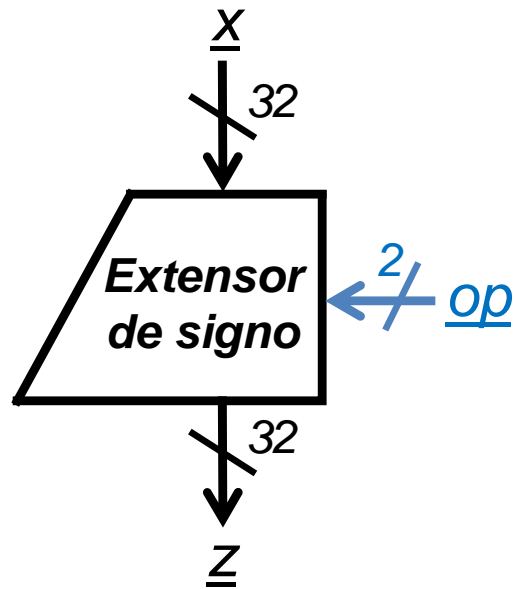
op <sub>2</sub>	op <sub>1</sub>	op <sub>0</sub>	<u>R</u>
0	1	0	$\underline{A} \& \underline{B}$
0	1	1	$\underline{A}   \underline{B}$
1	1	0	-
1	1	1	-







# Diseño del Extensor de Signo



---

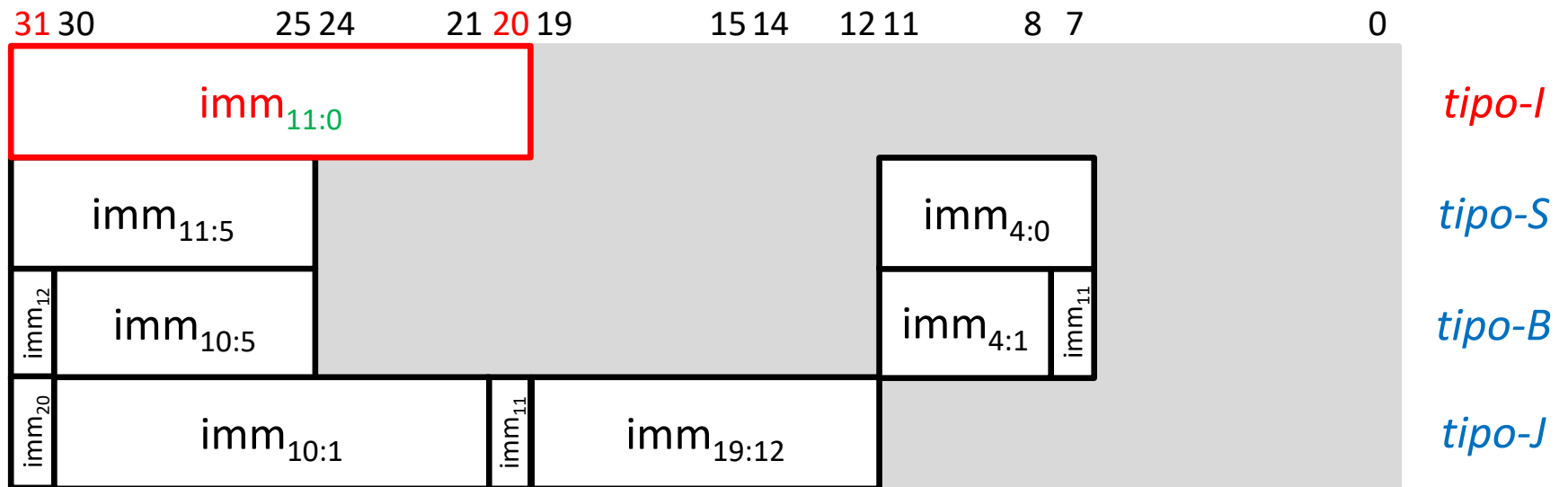
<u>x</u>	1 entrada de datos de 32 bits (instrucción)
<u>op</u>	1 entrada de selección de operación
<u>z</u>	1 salida de datos de 32 bits (operando inmediato)

---

- El **Extensor de Signo** es un módulo combinacional que **construye el operando inmediato** de 32 bits a partir de información contenida en los campos imm de la instrucción:
  - Para cada **tipo de instrucción** el **campo imm** tiene **diferente tamaño y posición** dentro de la instrucción.
  - Por ello, aparte de **extender el signo**, debe **reordenar bits y completar con 0** en el caso de direcciones de salto.



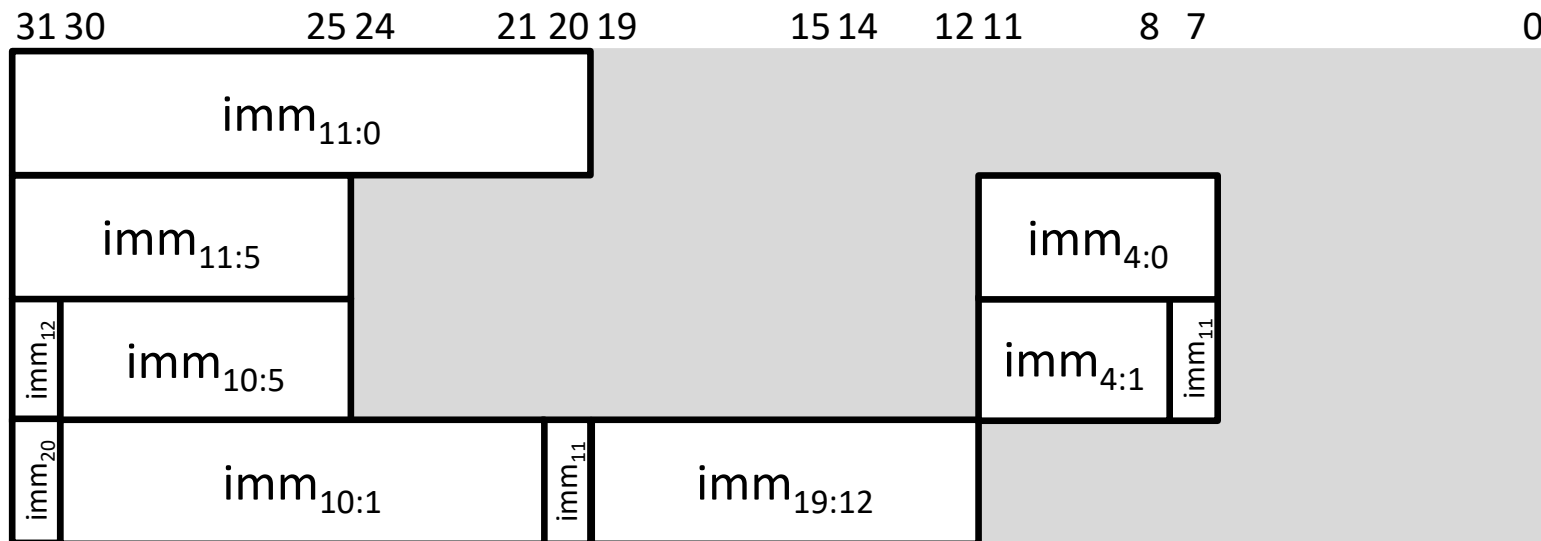
# Diseño del Extensor de Signo



<u>op</u>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>	
00																						X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>



# Diseño del Extensor de Signo



*tipo-I*

*tipo-S*

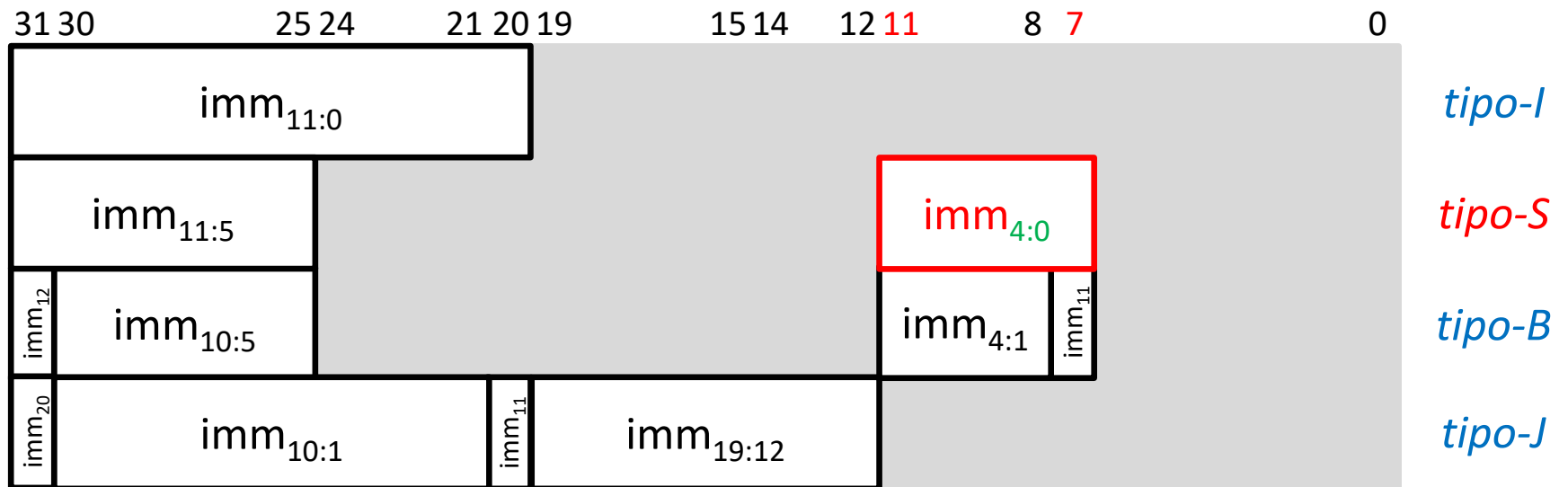
*tipo-B*

*tipo-J*

<u>op</u>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>	
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>



# Diseño del Extensor de Signo

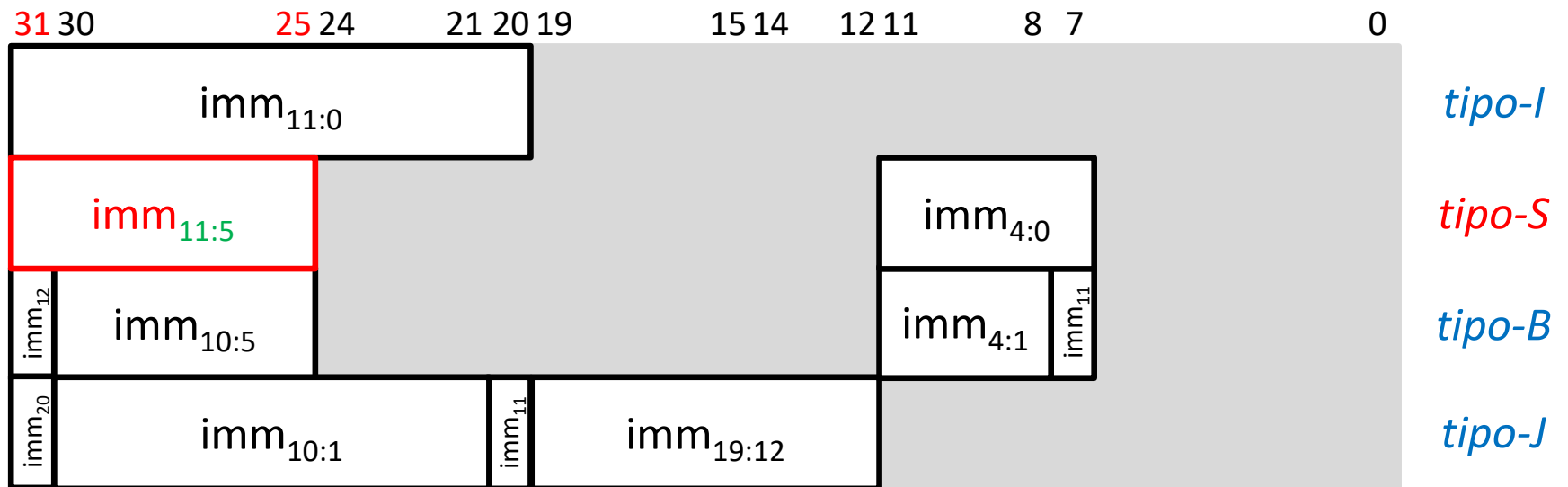


<u>op</u>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>	
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>	
01																																	





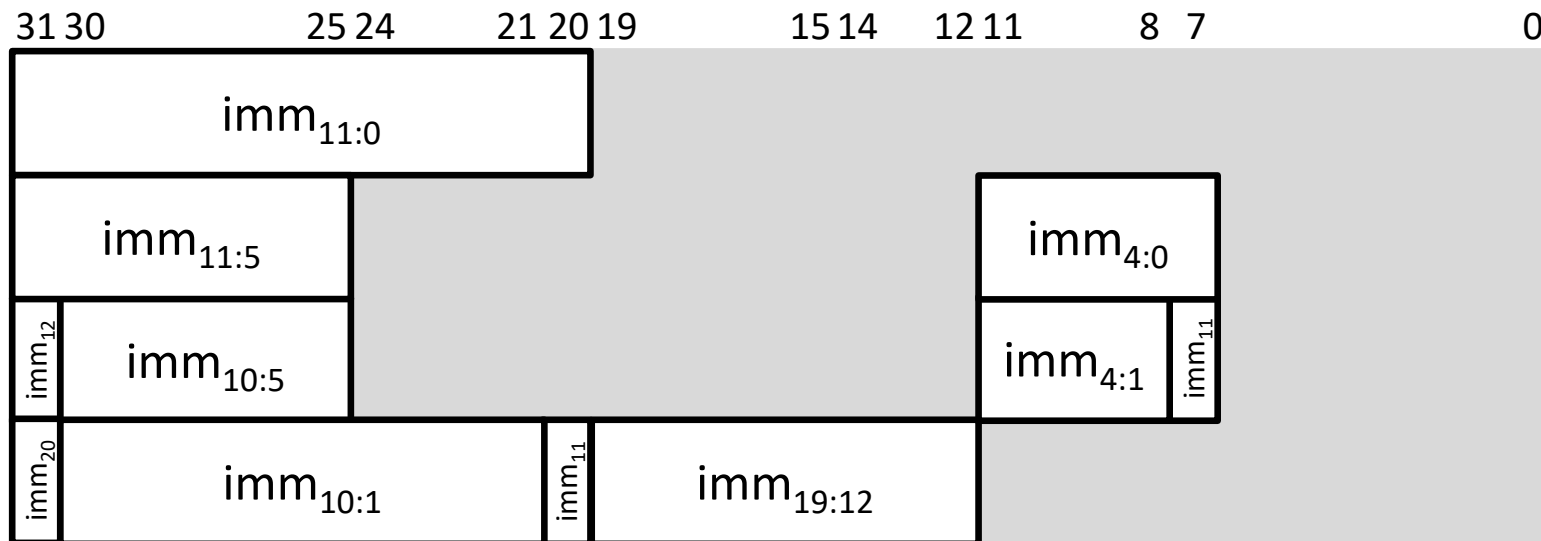
# Diseño del Extensor de Signo



<b>op</b>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>	
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>	
01																						X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>



# Diseño del Extensor de Signo



*tipo-I*

*tipo-S*

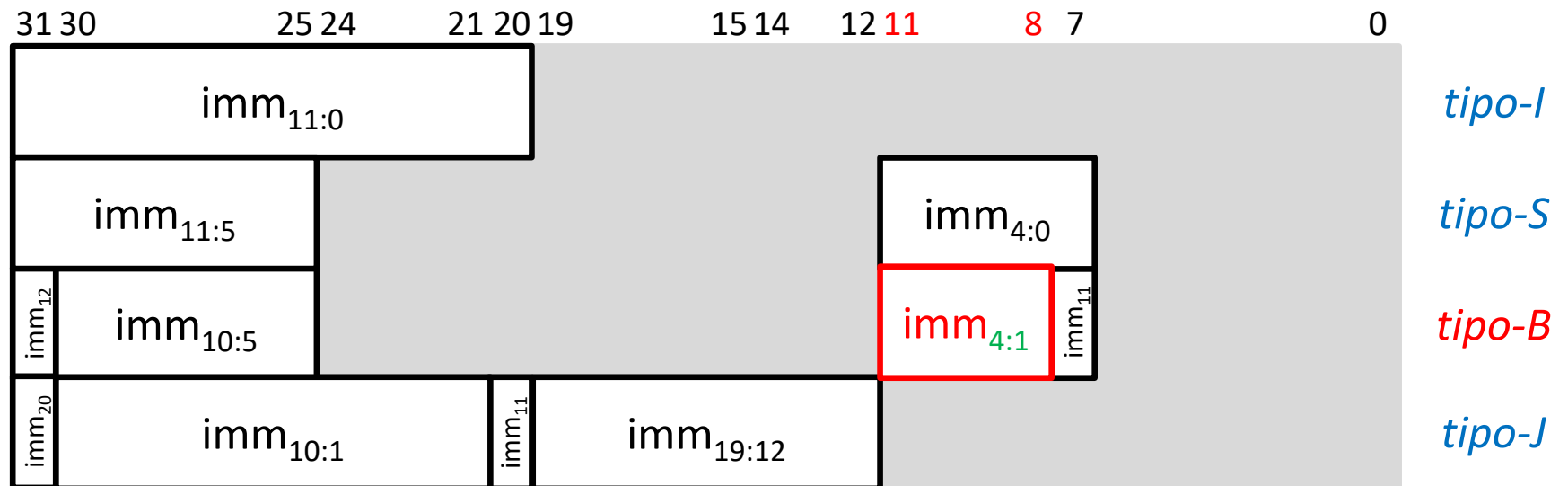
*tipo-B*

*tipo-J*

<b>op</b>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>



# Diseño del Extensor de Signo

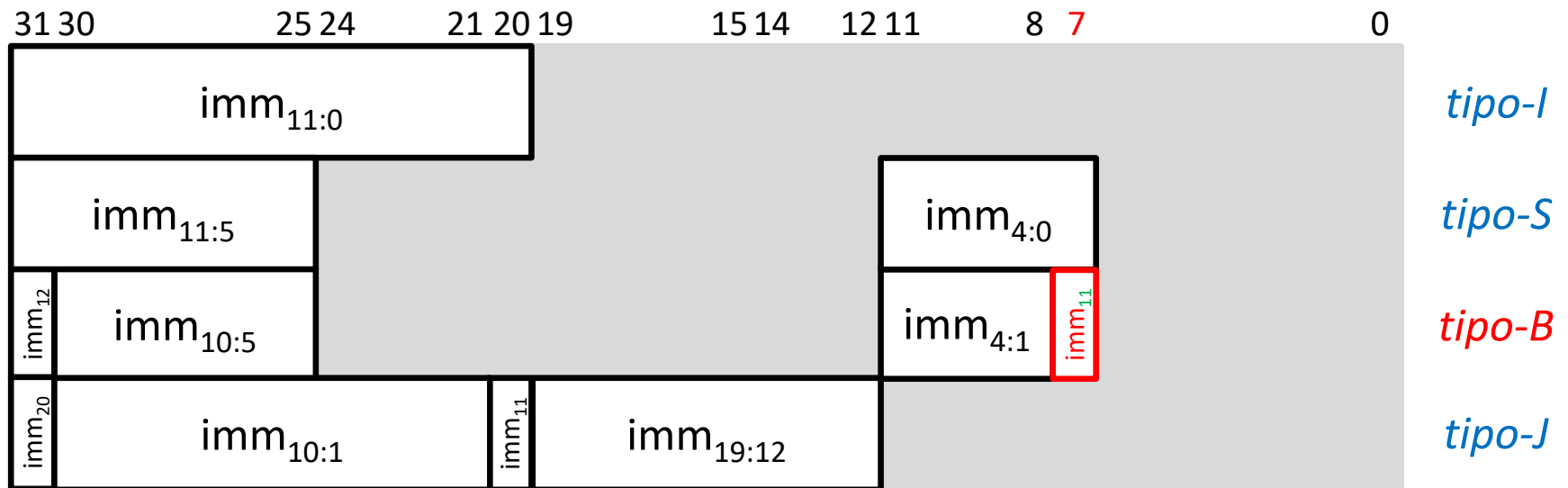


<b>op</b>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>				
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>				
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>				
10																																	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>





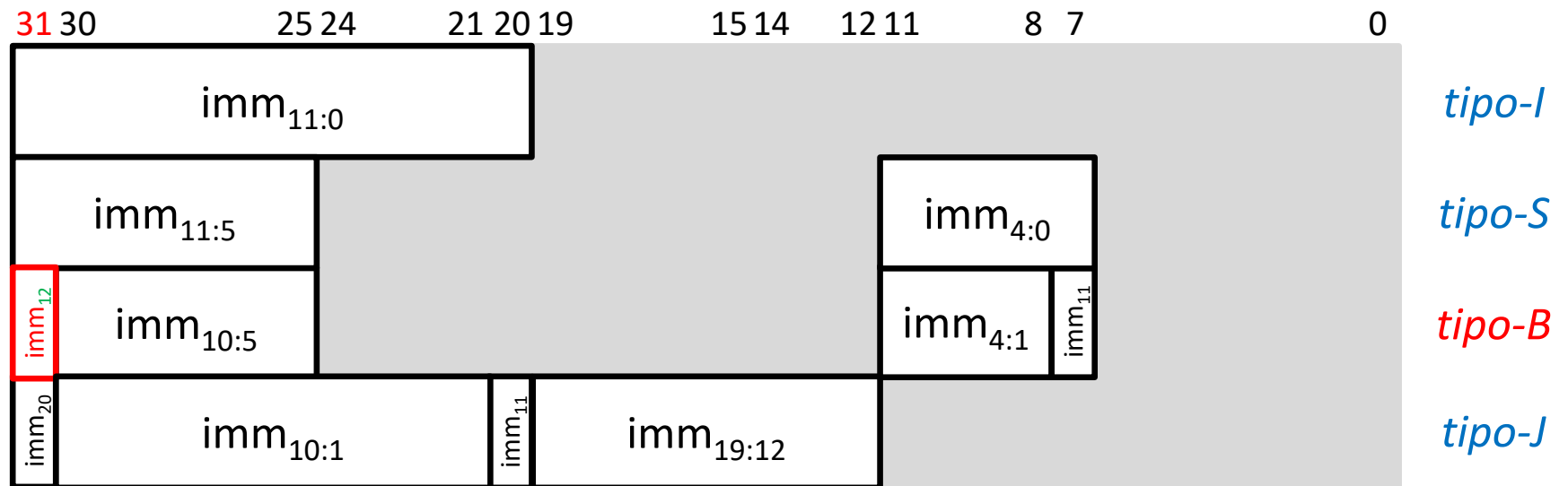
# Diseño del Extensor de Signo



<b>op</b>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>
10																						X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>



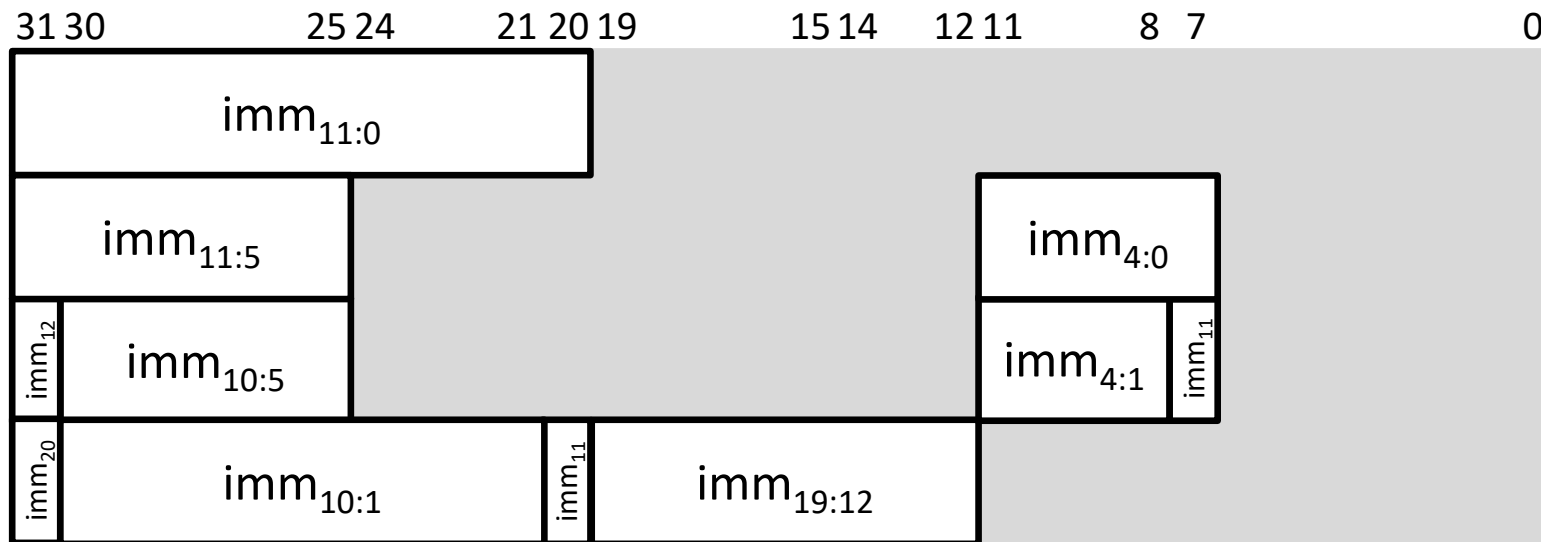
# Diseño del Extensor de Signo



<u>op</u>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>	
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>	
10																				X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	



# Diseño del Extensor de Signo



*tipo-I*

*tipo-S*

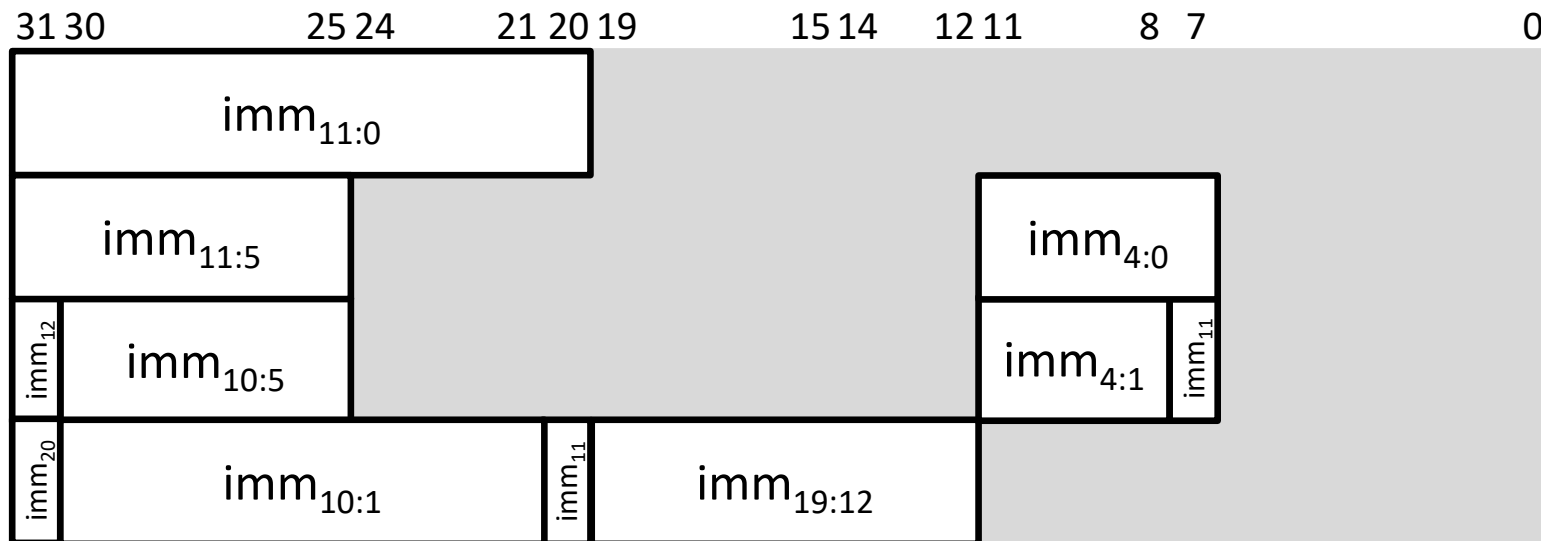
*tipo-B*

*tipo-J*

<b>op</b>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>	
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>	
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>	
10																					X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0



# Diseño del Extensor de Signo



*tipo-I*

*tipo-S*

*tipo-B*

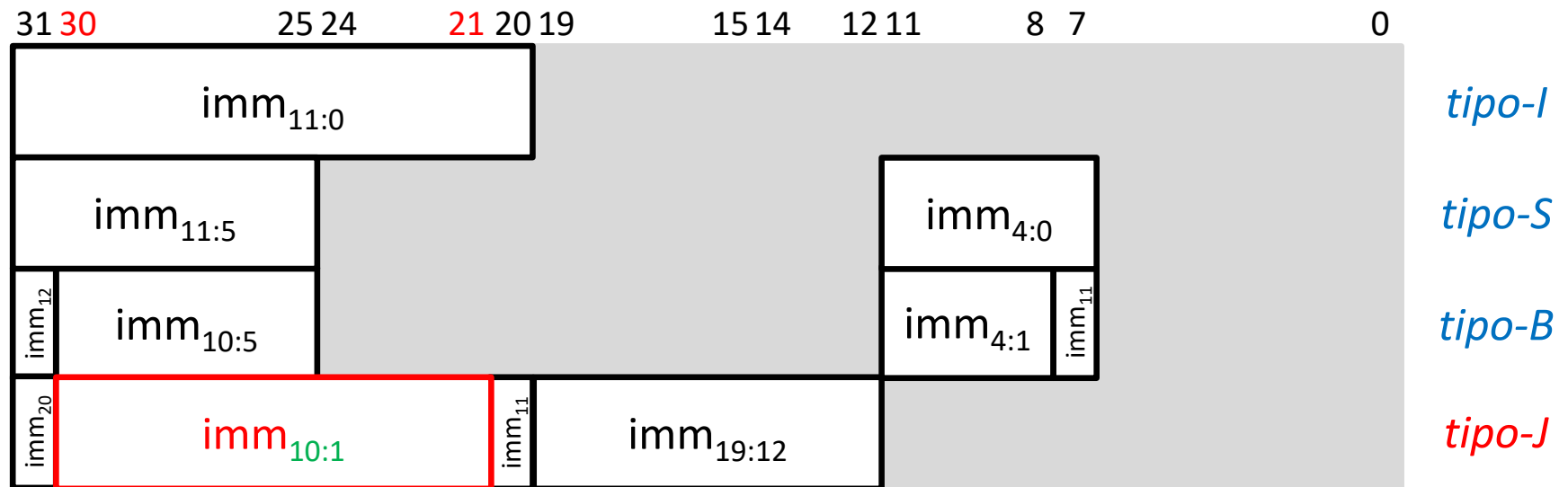
*tipo-J*

<b>op</b>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0





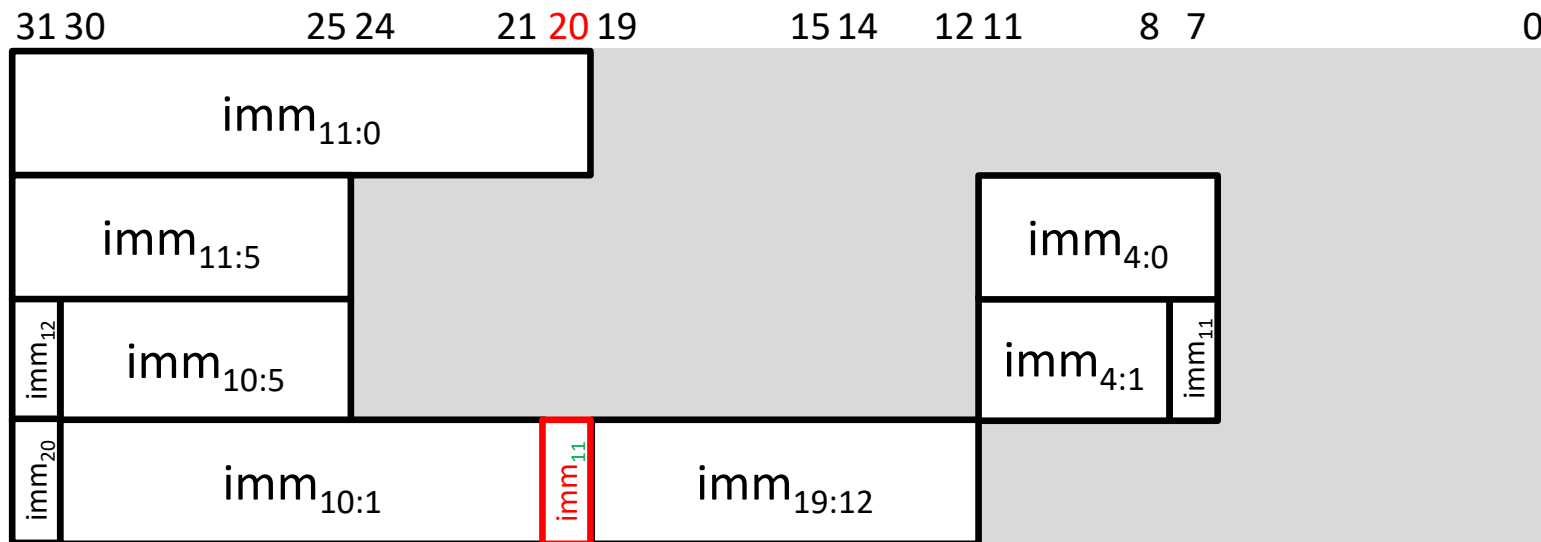
# Diseño del Extensor de Signo



<b>op</b>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>										
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>										
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>										
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0										
11																																	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>



# Diseño del Extensor de Signo



*tipo-I*

*tipo-S*

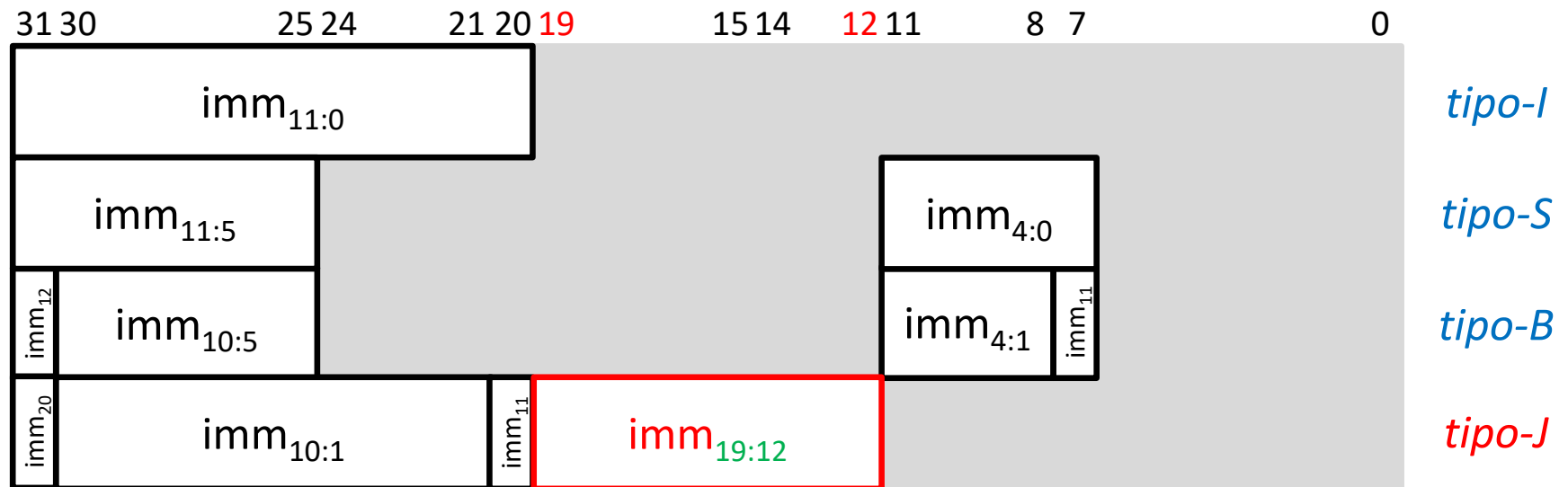
*tipo-B*

*tipo-J*

<b>op</b>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>											
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>											
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>											
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0											
11																																	X <sub>20</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>



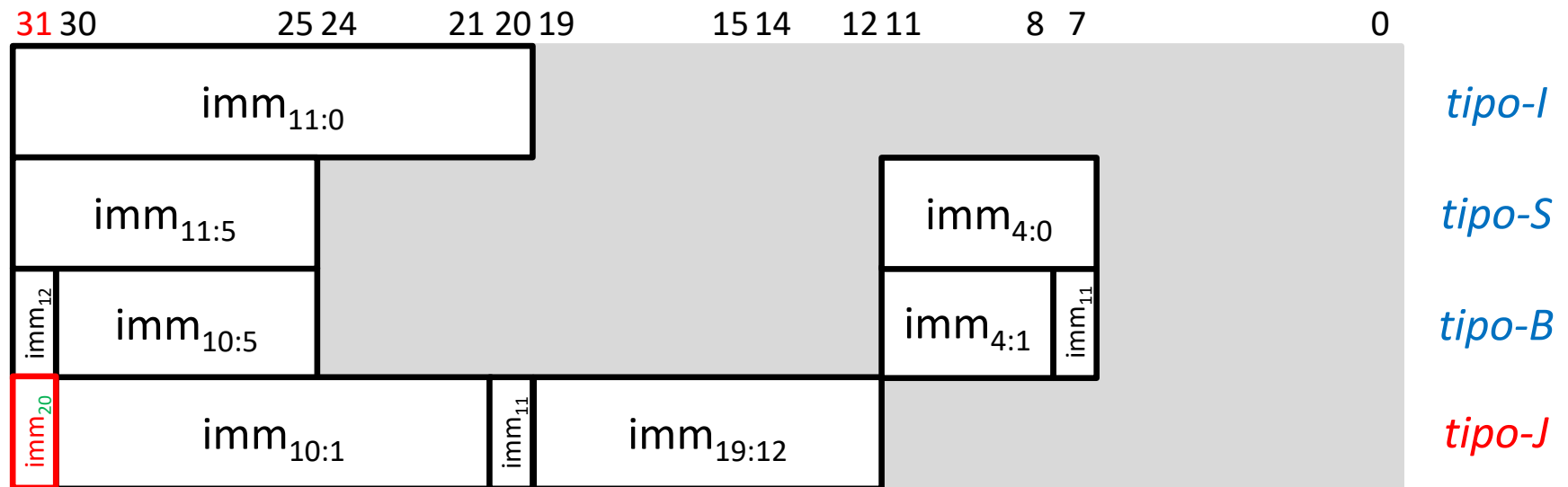
# Diseño del Extensor de Signo



op	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>	
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>	
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0	
11													X <sub>19</sub>	X <sub>18</sub>	X <sub>17</sub>	X <sub>16</sub>	X <sub>15</sub>	X <sub>14</sub>	X <sub>13</sub>	X <sub>12</sub>	X <sub>20</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	



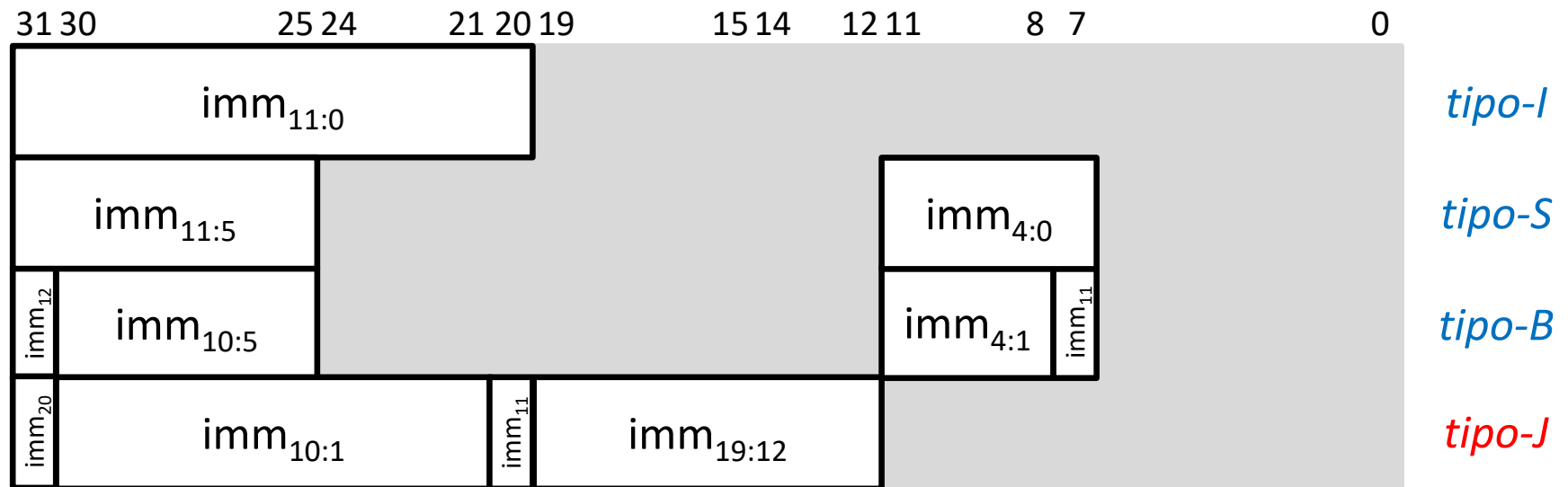
# Diseño del Extensor de Signo



<b>op</b>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0
11												X <sub>31</sub>	X <sub>19</sub>	X <sub>18</sub>	X <sub>17</sub>	X <sub>16</sub>	X <sub>15</sub>	X <sub>14</sub>	X <sub>13</sub>	X <sub>12</sub>	X <sub>20</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	



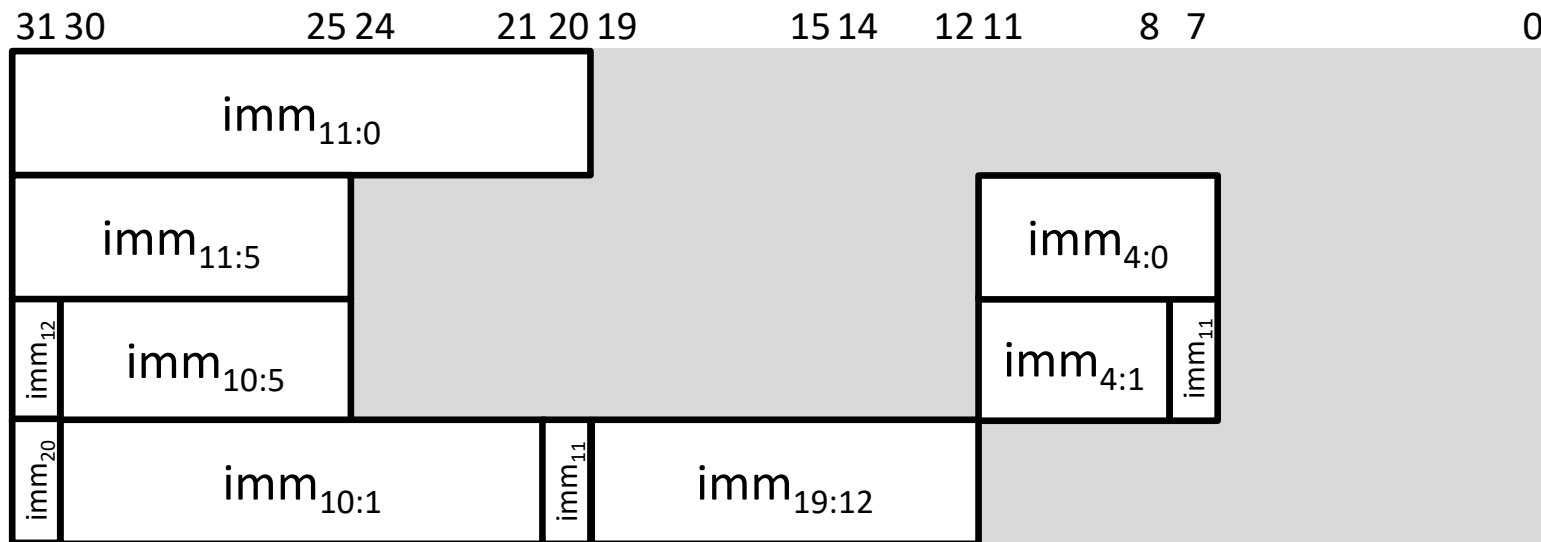
# Diseño del Extensor de Signo



<b>op</b>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>	
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>		
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>		
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0		
11													X <sub>31</sub>	X <sub>19</sub>	X <sub>18</sub>	X <sub>17</sub>	X <sub>16</sub>	X <sub>15</sub>	X <sub>14</sub>	X <sub>13</sub>	X <sub>12</sub>	X <sub>20</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	0



# Diseño del Extensor de Signo



*tipo-I*

*tipo-S*

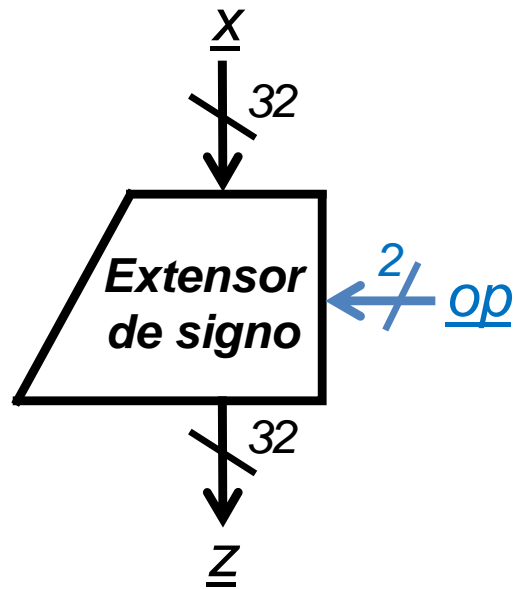
*tipo-B*

*tipo-J*

<b>op</b>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>	
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>	
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>	
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0	
11	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>19</sub>	X <sub>18</sub>	X <sub>17</sub>	X <sub>16</sub>	X <sub>15</sub>	X <sub>14</sub>	X <sub>13</sub>	X <sub>12</sub>	X <sub>20</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	0



# Diseño del Extensor de Signo

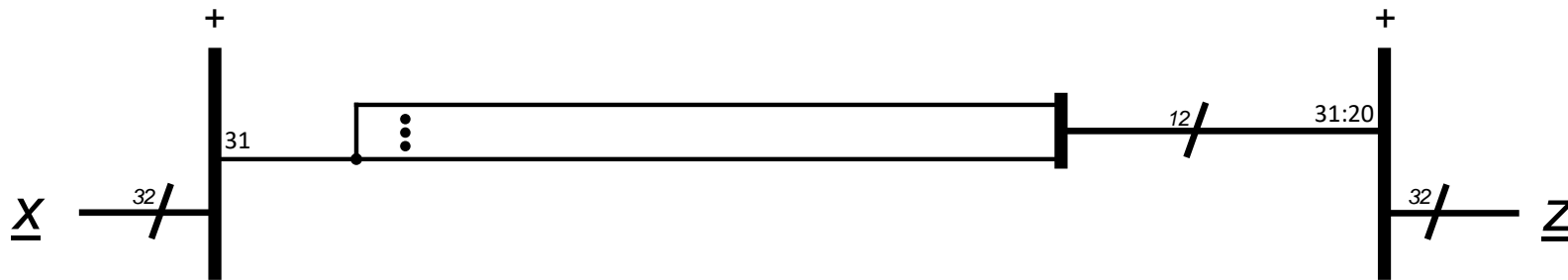


<u>x</u>	1 entrada de datos de 32 bits (instrucción)
<u>op</u>	1 entrada de selección de operación
<u>z</u>	1 salida de datos de 32 bits (operando inmediato)

<u>op</u>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>	
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>	
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0
11	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>20</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	0



# Diseño del Extensor de Signo

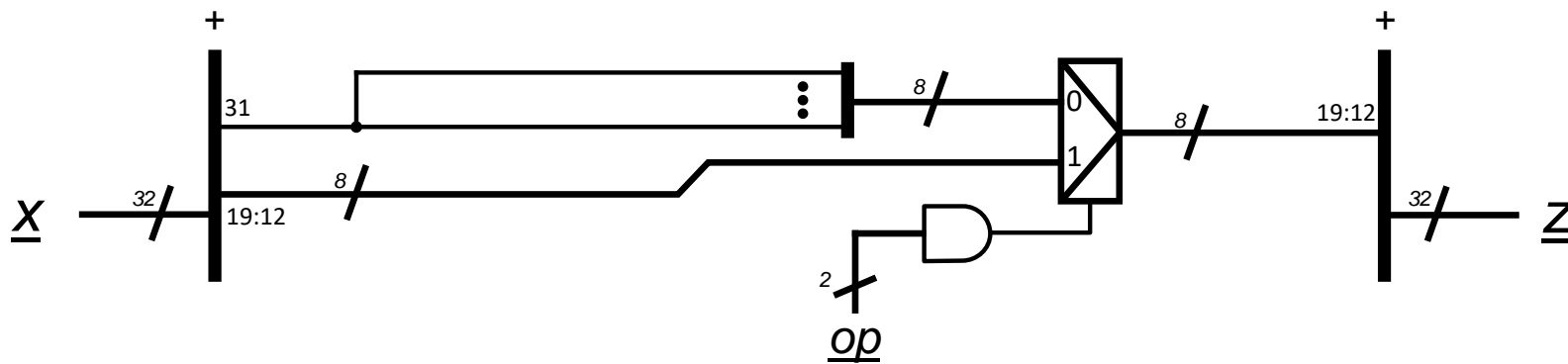


op	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0
11	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>19</sub>	X <sub>18</sub>	X <sub>17</sub>	X <sub>16</sub>	X <sub>15</sub>	X <sub>14</sub>	X <sub>13</sub>	X <sub>12</sub>	X <sub>20</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	0





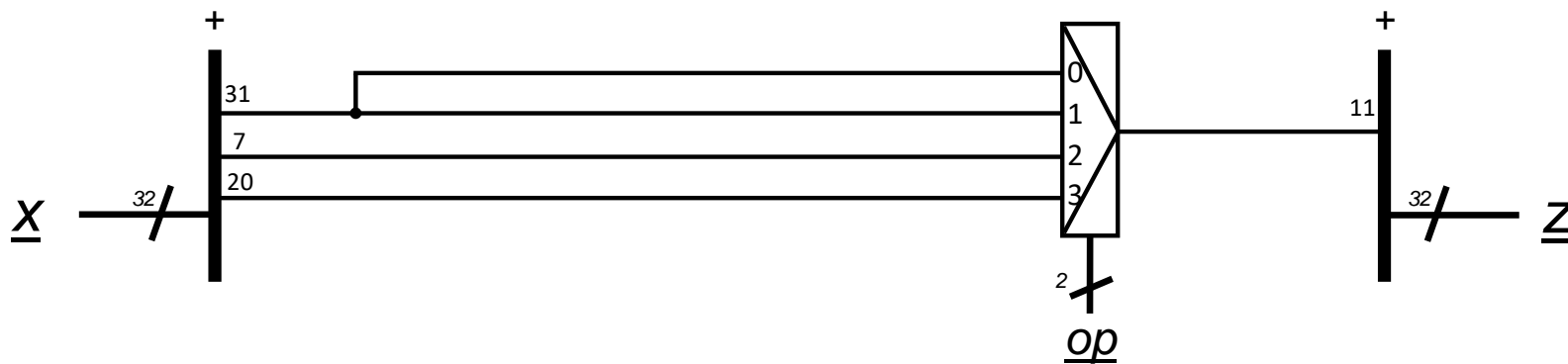
# Diseño del Extensor de Signo



<u>op</u>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0
11	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>19</sub>	X <sub>18</sub>	X <sub>17</sub>	X <sub>16</sub>	X <sub>15</sub>	X <sub>14</sub>	X <sub>13</sub>	X <sub>12</sub>	X <sub>20</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	0



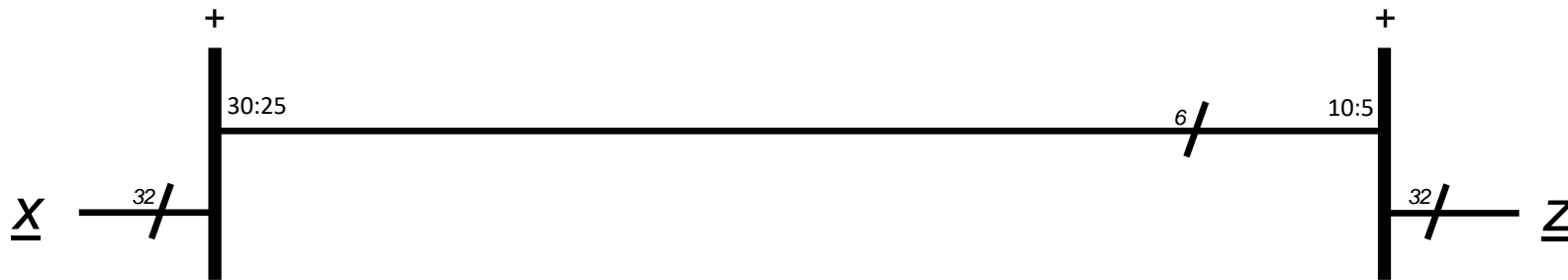
# Diseño del Extensor de Signo



<u>op</u>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>	
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0
11	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	0



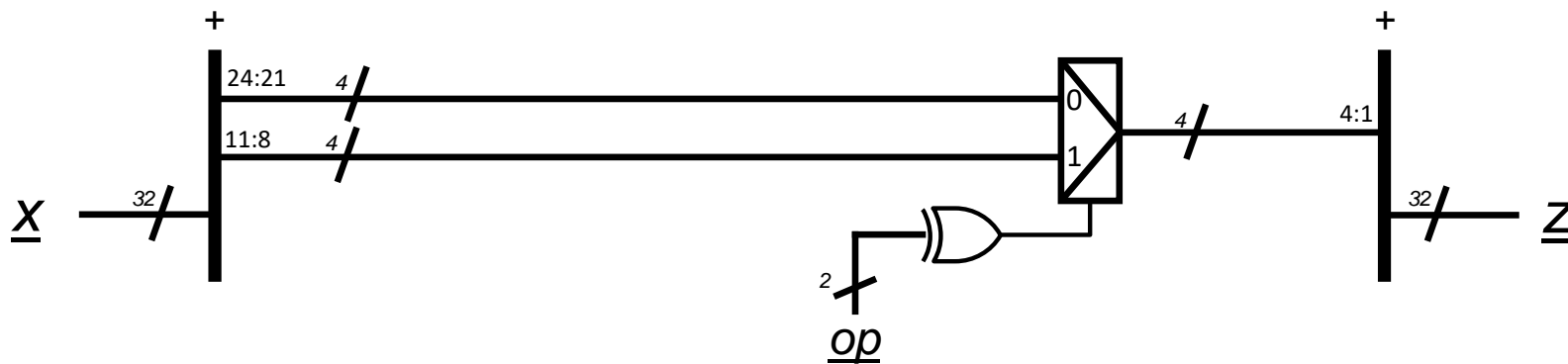
# Diseño del Extensor de Signo



op	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0
11	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>19</sub>	X <sub>18</sub>	X <sub>17</sub>	X <sub>16</sub>	X <sub>15</sub>	X <sub>14</sub>	X <sub>13</sub>	X <sub>12</sub>	X <sub>20</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	0



# Diseño del Extensor de Signo

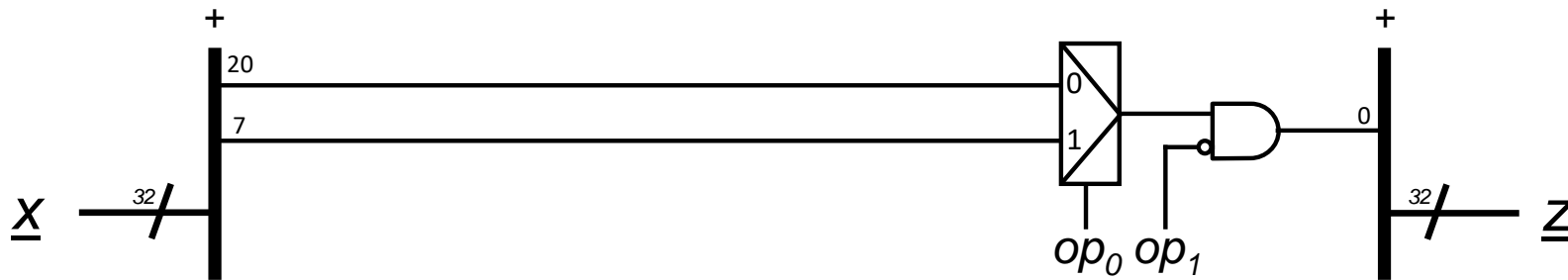


<u>op</u>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>	
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>	
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>	
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0
11	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>20</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	0



# Diseño del Extensor de Signo

versión 31/10/23

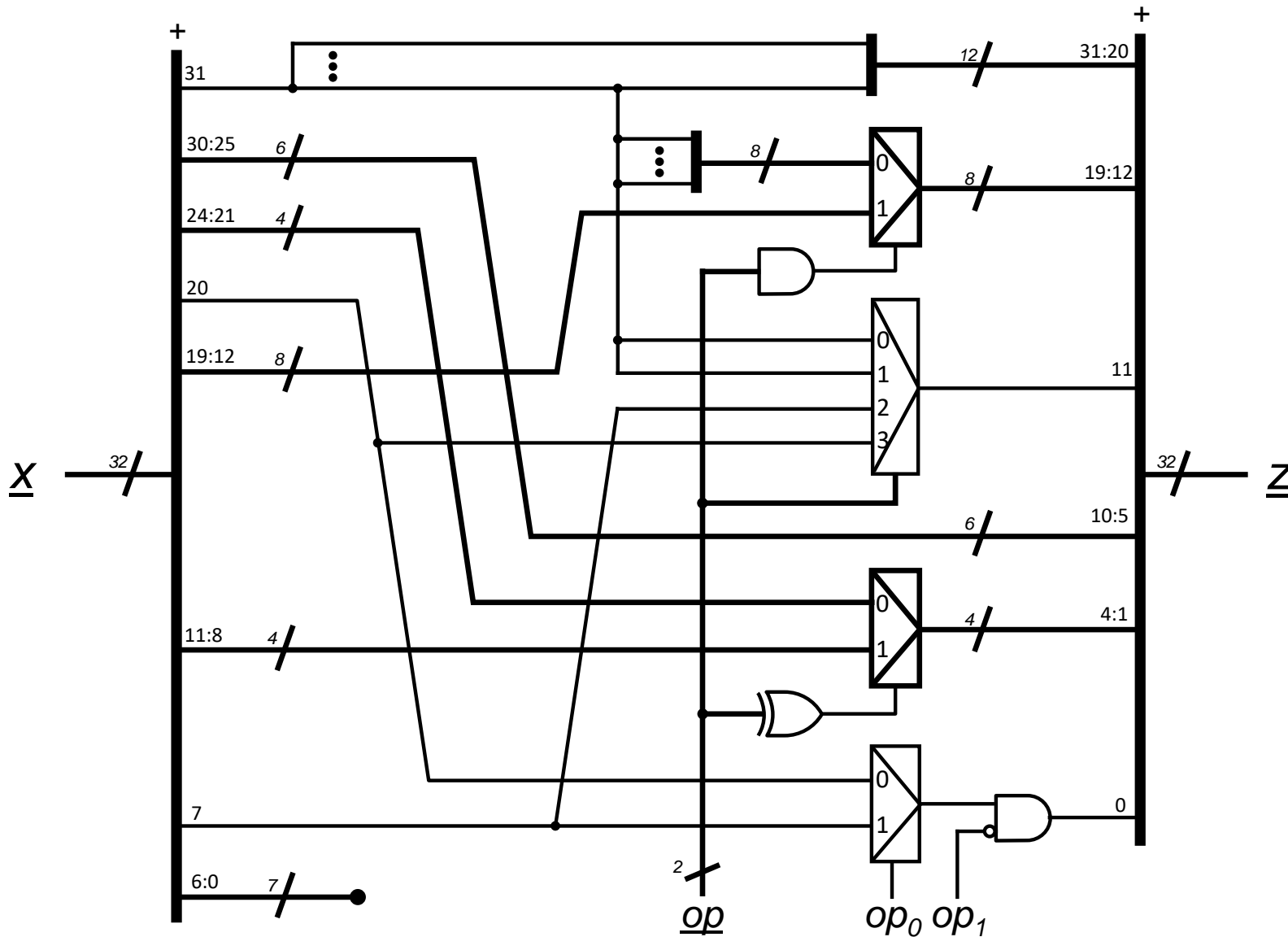


tema 5:  
Diseño monociclo del procesador

<u>op</u>	Z <sub>31</sub>	Z <sub>30</sub>	Z <sub>29</sub>	Z <sub>28</sub>	Z <sub>27</sub>	Z <sub>26</sub>	Z <sub>25</sub>	Z <sub>24</sub>	Z <sub>23</sub>	Z <sub>22</sub>	Z <sub>21</sub>	Z <sub>20</sub>	Z <sub>19</sub>	Z <sub>18</sub>	Z <sub>17</sub>	Z <sub>16</sub>	Z <sub>15</sub>	Z <sub>14</sub>	Z <sub>13</sub>	Z <sub>12</sub>	Z <sub>11</sub>	Z <sub>10</sub>	Z <sub>9</sub>	Z <sub>8</sub>	Z <sub>7</sub>	Z <sub>6</sub>	Z <sub>5</sub>	Z <sub>4</sub>	Z <sub>3</sub>	Z <sub>2</sub>	Z <sub>1</sub>	Z <sub>0</sub>
00	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	X <sub>20</sub>	
01	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	X <sub>7</sub>	
10	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>7</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>11</sub>	X <sub>10</sub>	X <sub>9</sub>	X <sub>8</sub>	0
11	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>31</sub>	X <sub>19</sub>	X <sub>18</sub>	X <sub>17</sub>	X <sub>16</sub>	X <sub>15</sub>	X <sub>14</sub>	X <sub>13</sub>	X <sub>12</sub>	X <sub>20</sub>	X <sub>30</sub>	X <sub>29</sub>	X <sub>28</sub>	X <sub>27</sub>	X <sub>26</sub>	X <sub>25</sub>	X <sub>24</sub>	X <sub>23</sub>	X <sub>22</sub>	X <sub>21</sub>	0



# Diseño del Extensor de Signo





# Cálculo del coste y tiempo de ciclo

- Para calcular el **coste del procesador** es la **suma del coste** de cada uno de los módulos que lo componen.
  - El **coste de cada módulo** se calcula sumando el **coste de sus celdas**.
- El **tiempo de ciclo del procesador** es el **máximo** de los **caminos críticos** de las **transferencias entre registros** que realiza el procesador.
  - El **camino crítico** de una transferencia entre registros es **el camino de datos de mayor retardo** de todos los implicados en dicha transferencia.
  - En el **procesador monociclo**, en cada ciclo se ejecuta una instrucción que implica **1 ó 2 transferencias entre registros**: la de actualización del PC y la operativa propia de cada instrucción.
- Para todos los cálculos se utilizará la **misma biblioteca** de celdas (CMOS 90nm) usada en **FC-1**



# Cálculo del coste y tiempo de ciclo

## CMOS 90 nm

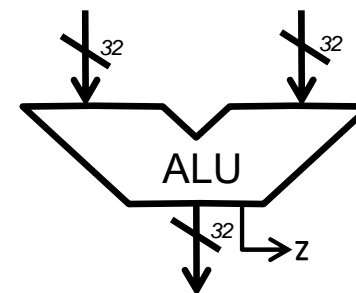
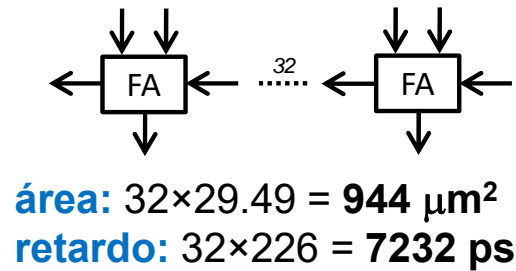
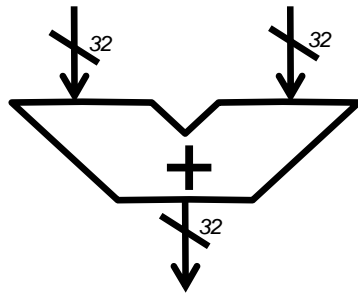


versión 31/10/23

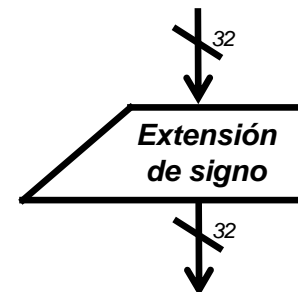
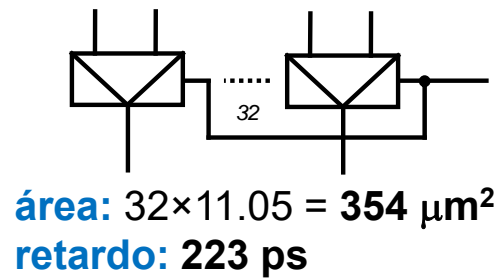
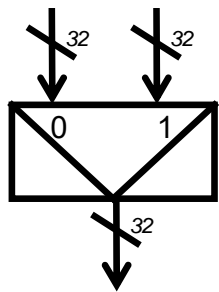
tema 5:  
Diseño monociclo del procesador

FC-2

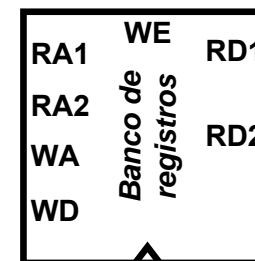
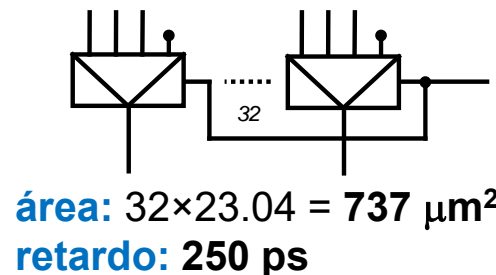
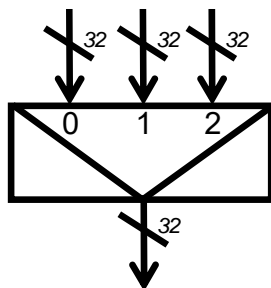
120



**área:**  $3052 \mu\text{m}^2$   
**retardo:**  $8360 \text{ ps}$



**área:**  $202 \mu\text{m}^2$   
**retardo:**  $460 \text{ ps}$



**área:**  $51405 \mu\text{m}^2$   
**retardo lectura:**  $723 \text{ ps}$   
**setup escritura:**  $705 \text{ ps}$   
(debido al DEC de dirección)





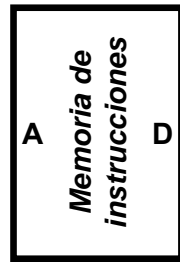
# Cálculo del coste y tiempo de ciclo

## CMOS 90 nm

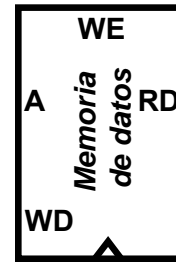


versión 31/10/23

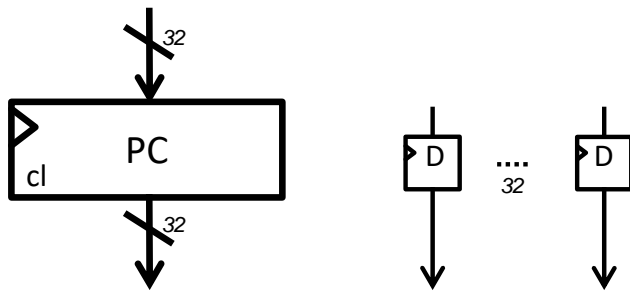
Comportamiento idealizado: retardo comparable al de la ALU  
(para que pueda leerse en un ciclo de reloj)



área: -  
tiempo de acceso: 8500 ps



área: -  
tiempo de acceso: 8500 ps



área:  $32 \times 32 \cdot 26 = 1032 \mu\text{m}^2$   
retardo CLK  $\rightarrow$  Q:  $1 \times 167 = 167 \text{ ps}$   
setup: 0 ps



área:  $56 \mu\text{m}^2$   
retardo: 490 ps



área:  $65 \mu\text{m}^2$   
retardo: 451 ps



área:  $21 \mu\text{m}^2$   
retardo: 451 ps



área:  $15 \mu\text{m}^2$   
retardo: 351 ps



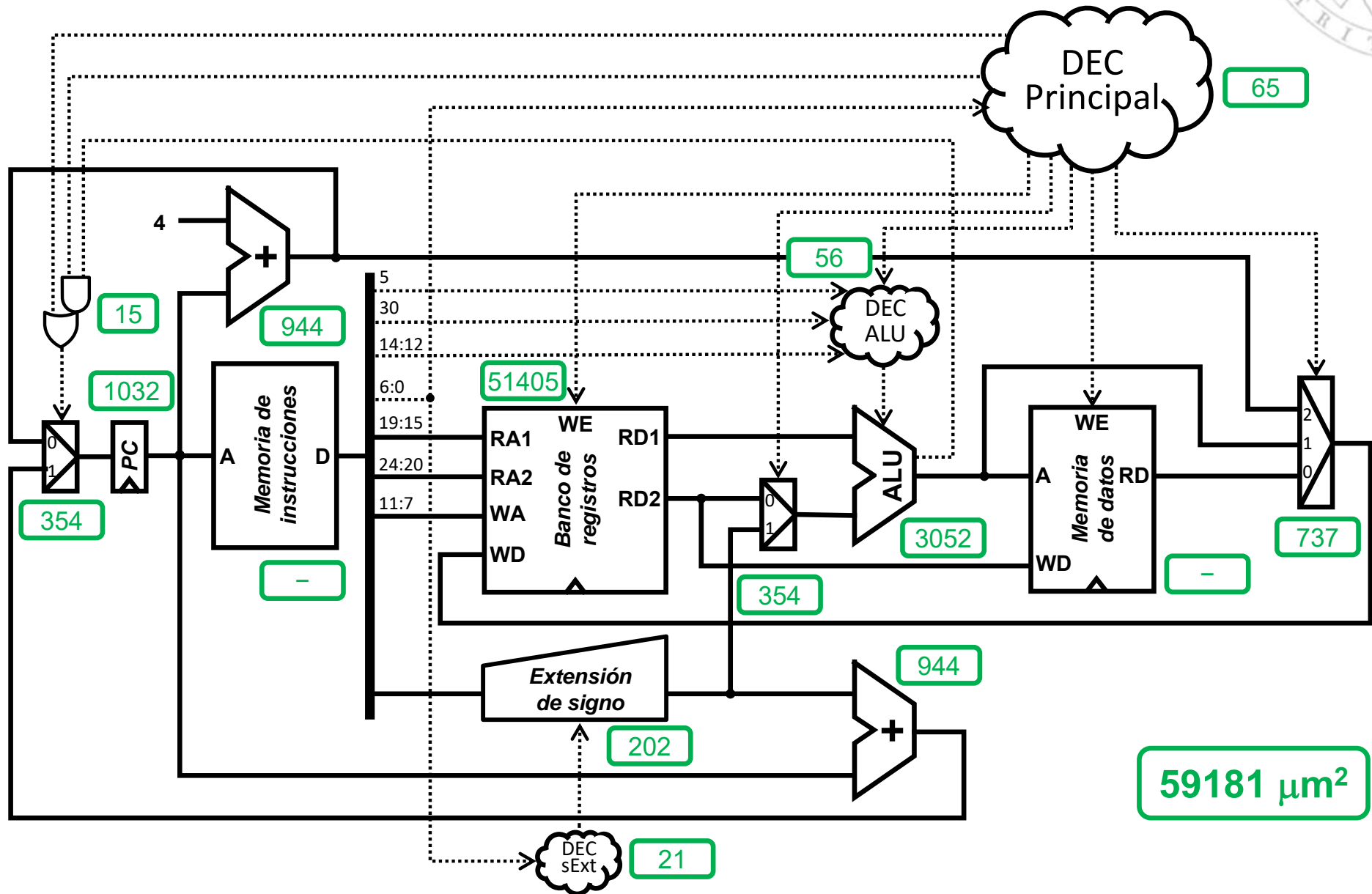
# Cálculo del coste

versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

122





# Cálculo del tiempo de ciclo

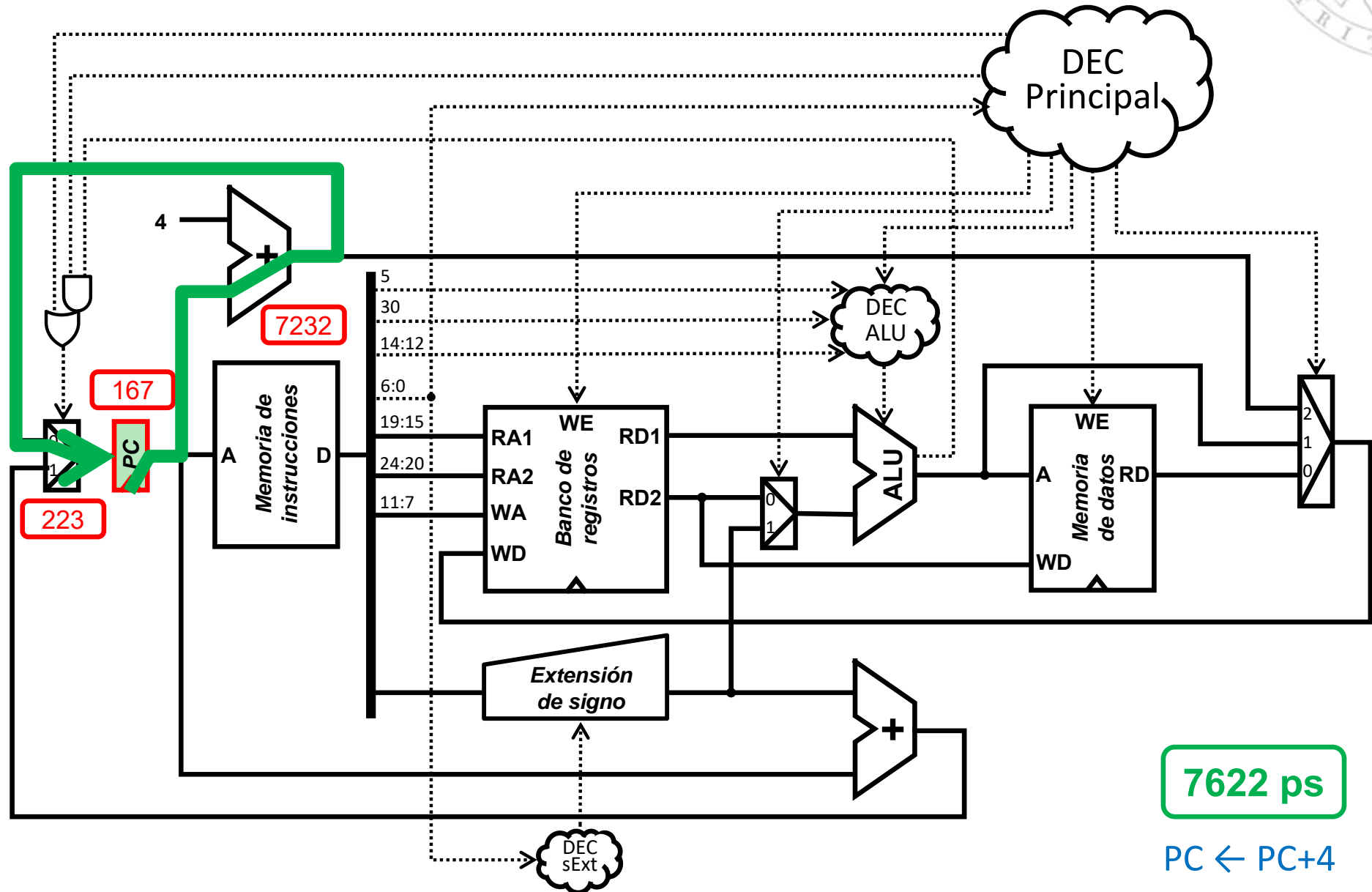
## Incremento del PC

versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

123





# Cálculo del tiempo de ciclo

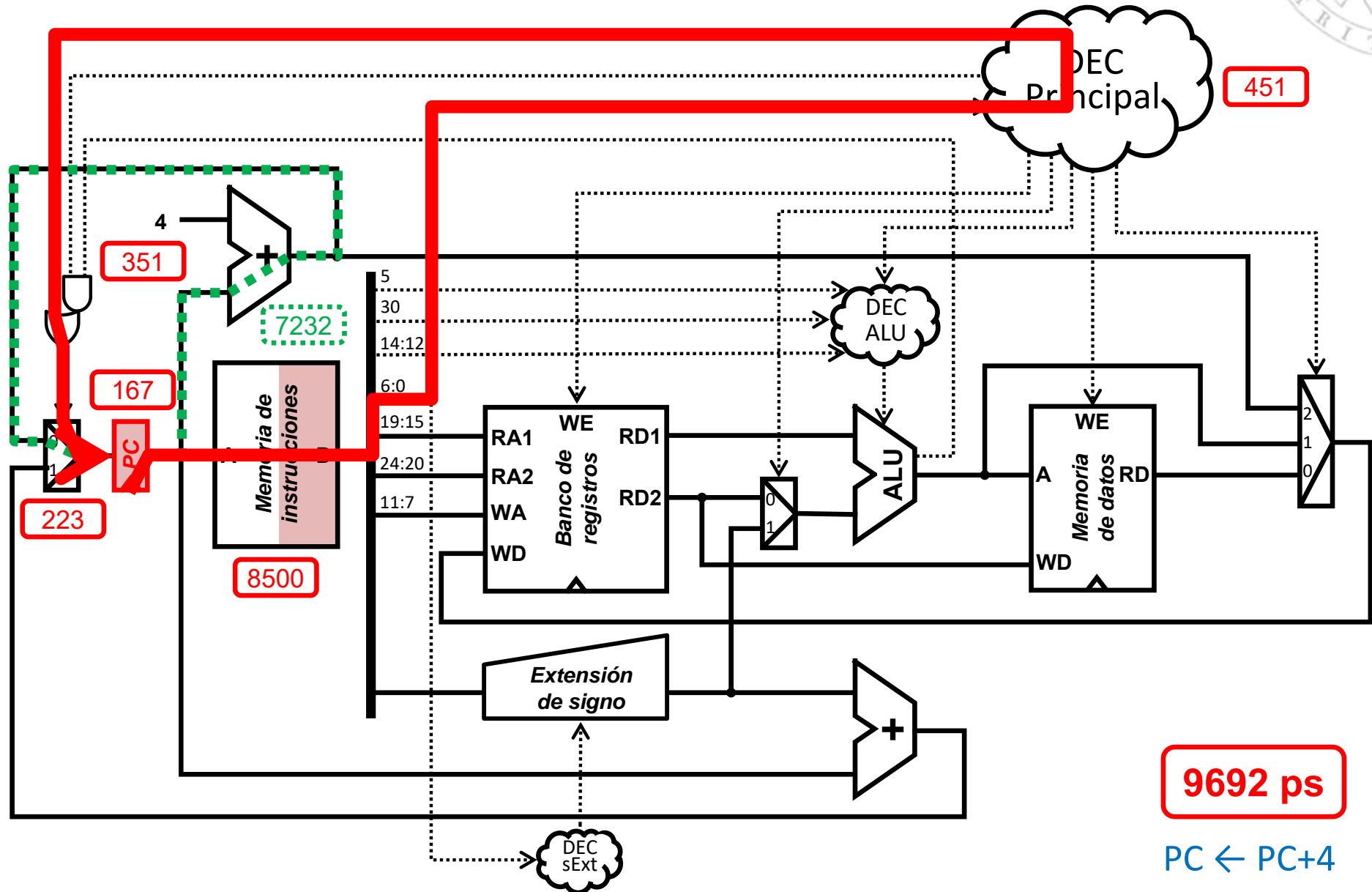
## Incremento del PC: camino crítico

versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

124



9692 ps

PC ← PC+4





# Cálculo del tiempo de ciclo

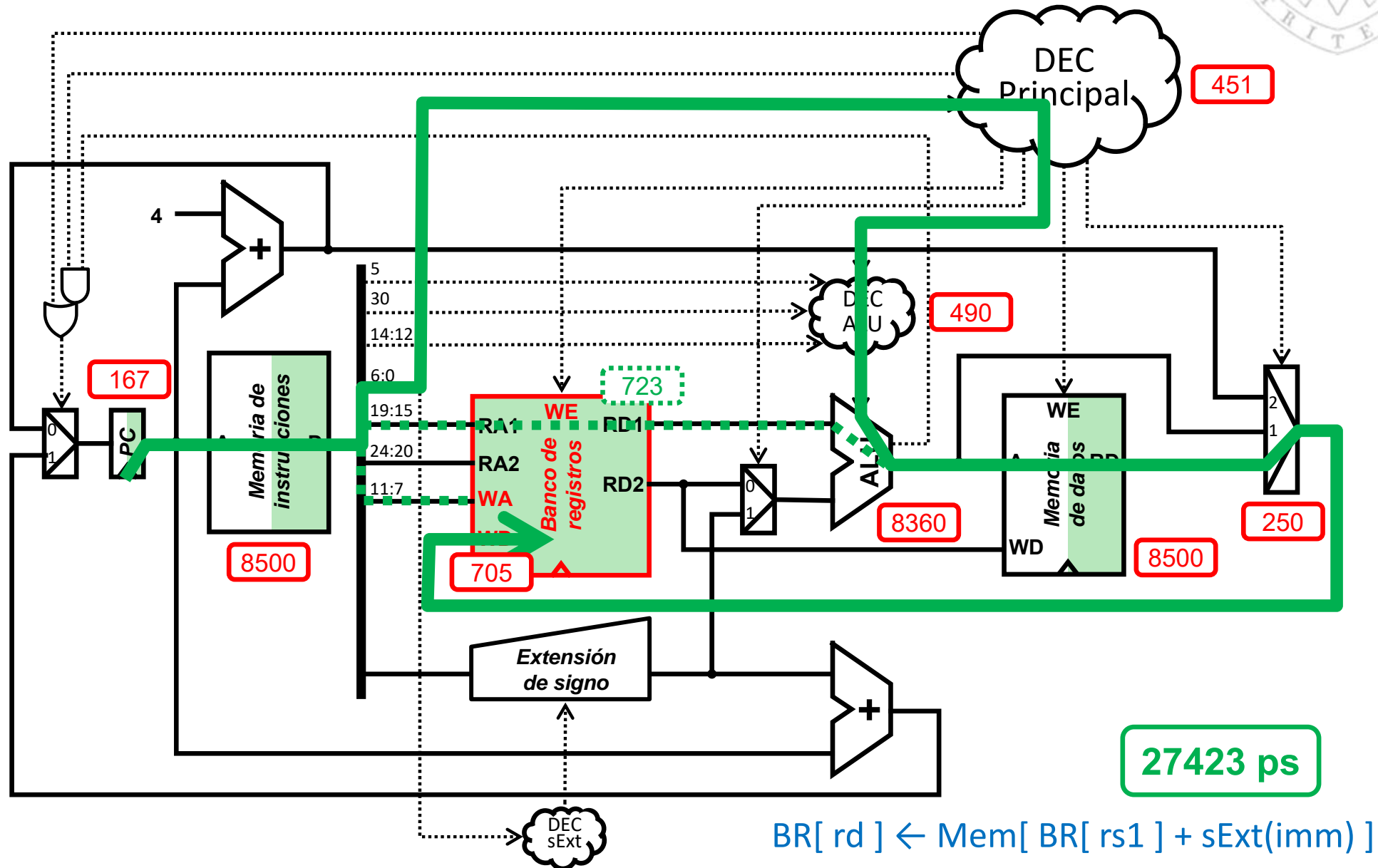
## Instrucción lw (ii)

versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

126





# Cálculo del tiempo de ciclo

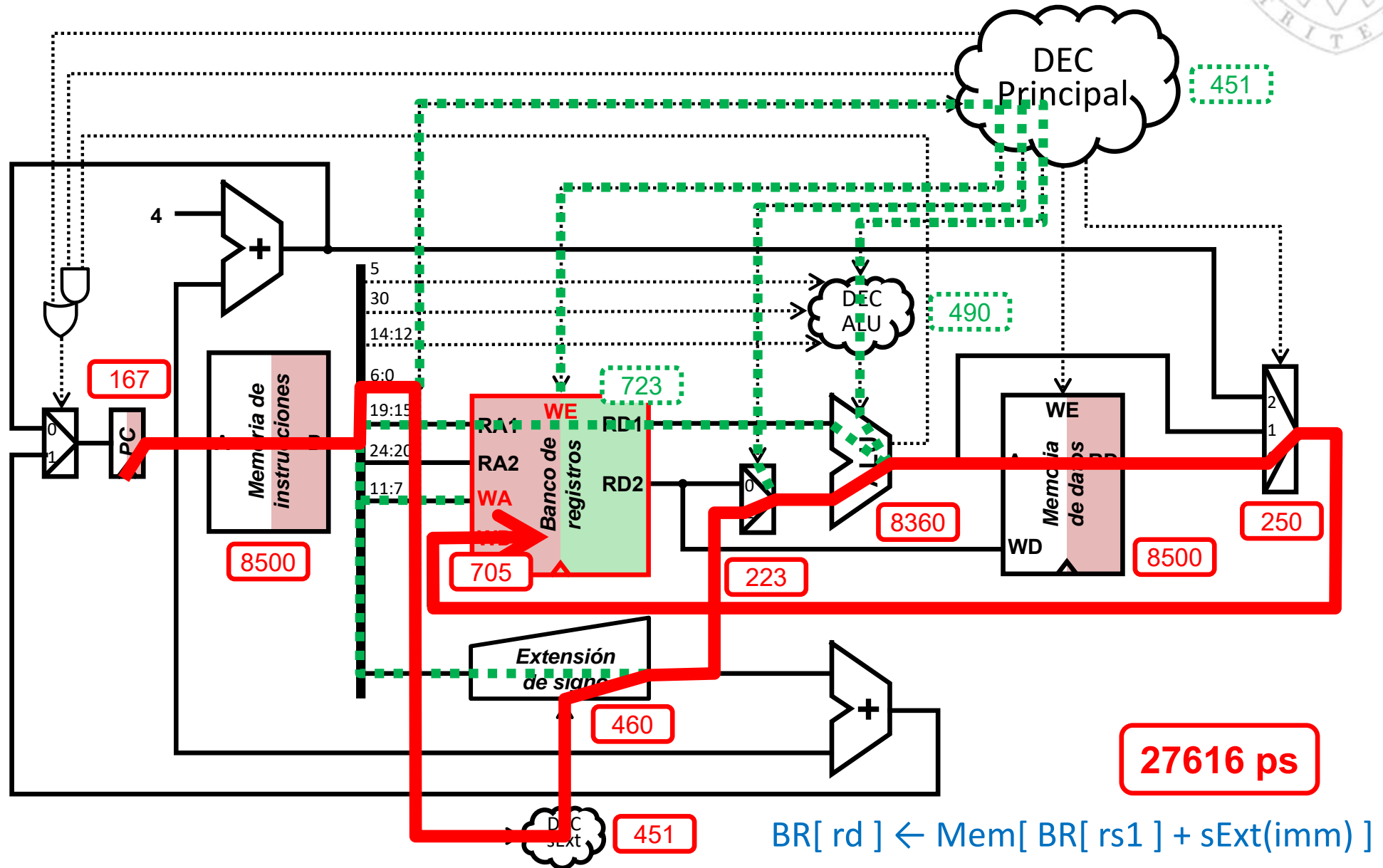
## Instrucción lw: camino crítico

versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

127



27616 ps

$$BR[rd] \leftarrow Mem[BR[rs1] + sExt(imm)]$$



# Cálculo del tiempo de ciclo

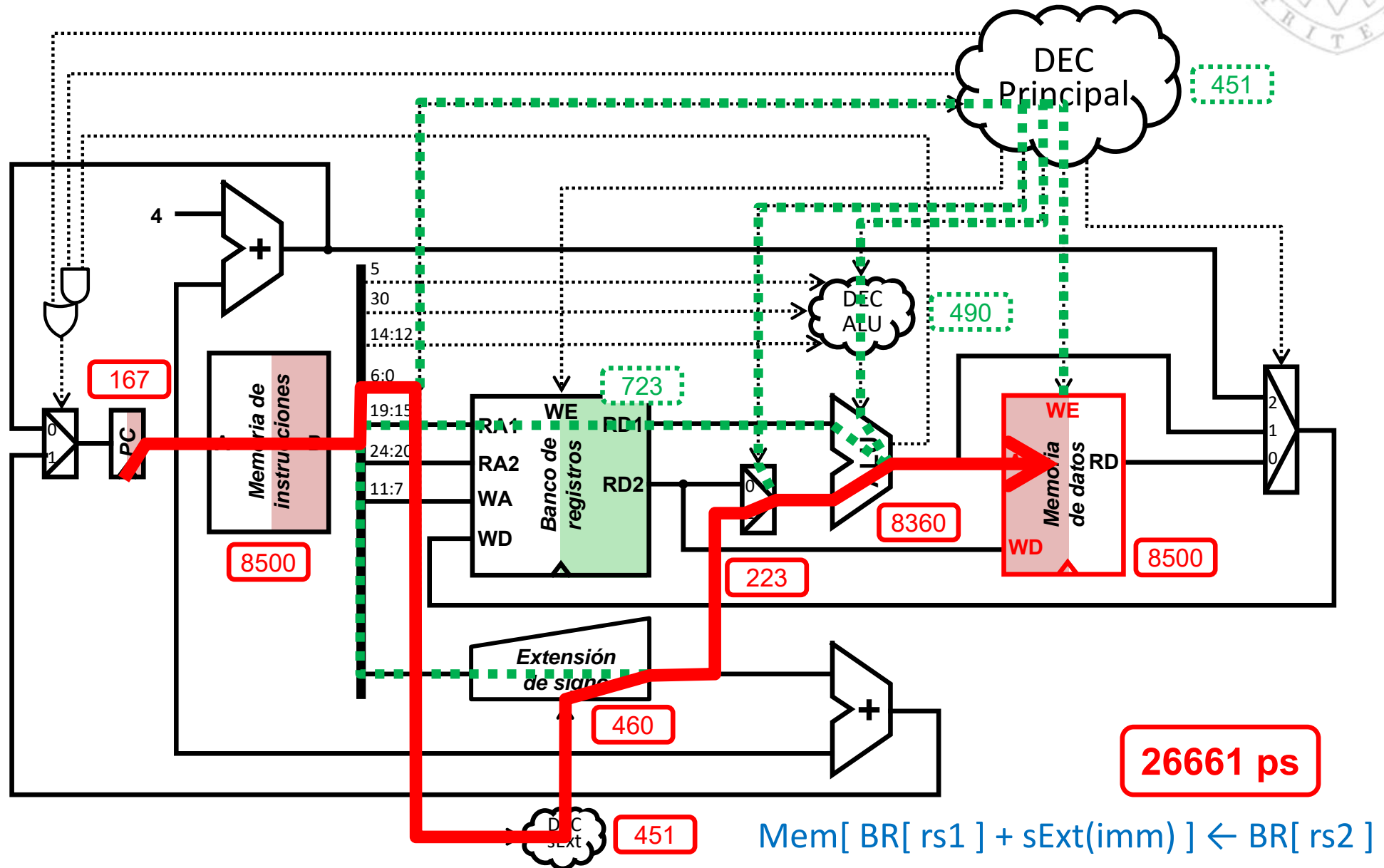
## Instrucción **sw**: camino crítico

versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

128







# Cálculo del tiempo de ciclo

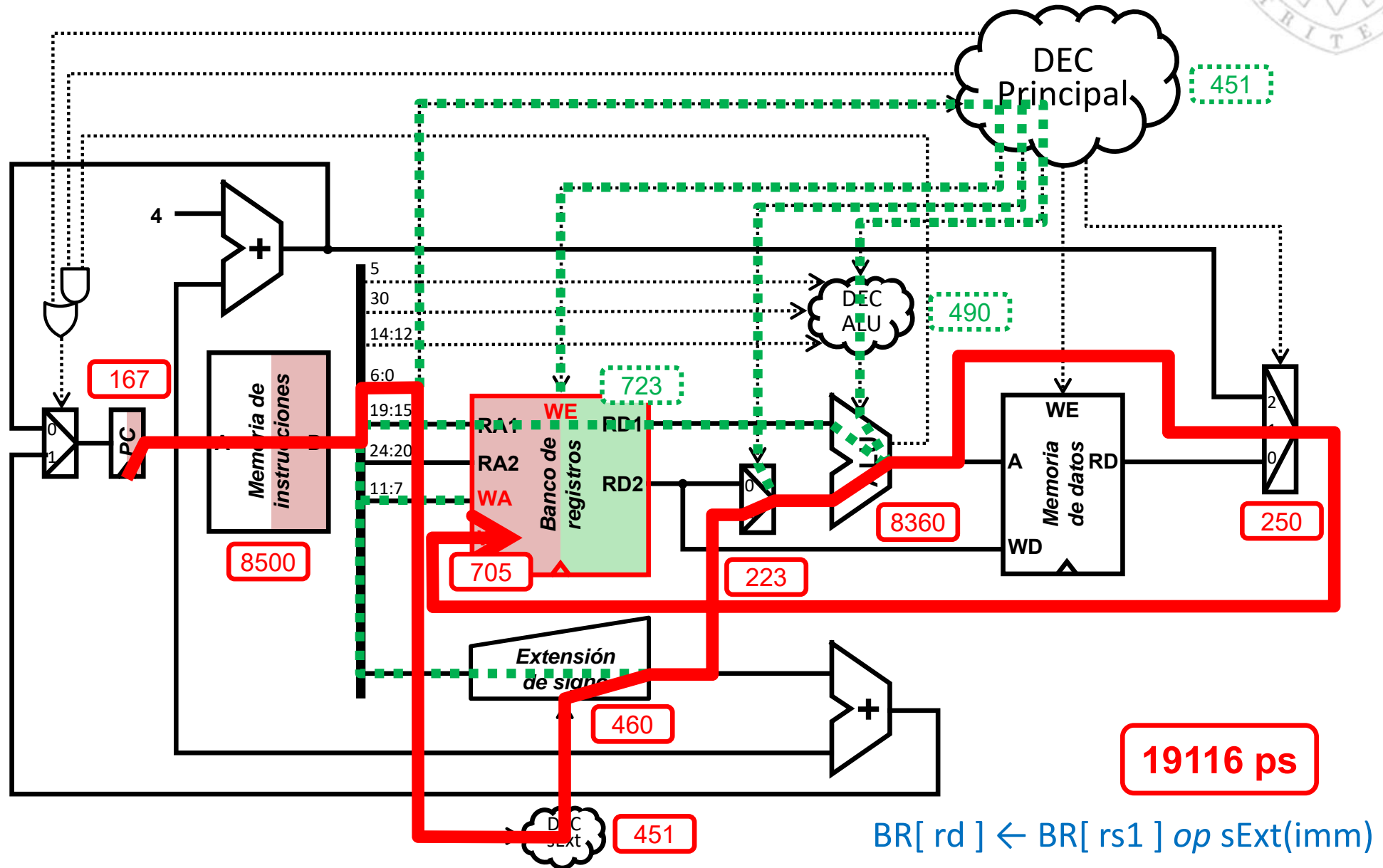
## Instrucciones tipo `addi`: camino crítico

versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

129





# Cálculo del tiempo de ciclo

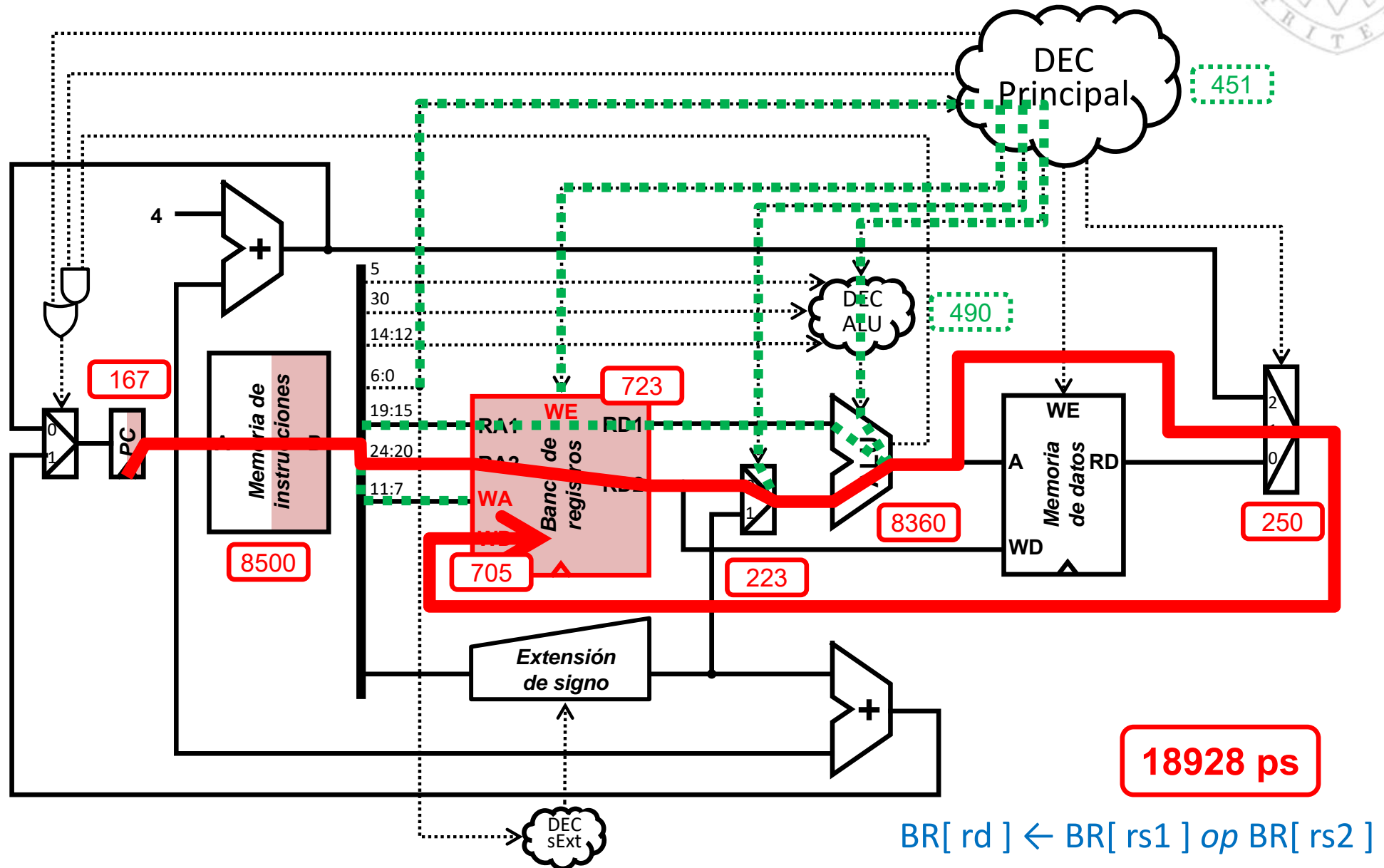
Instrucciones tipo **add**: camino crítico

versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

130





# Cálculo del tiempo de ciclo

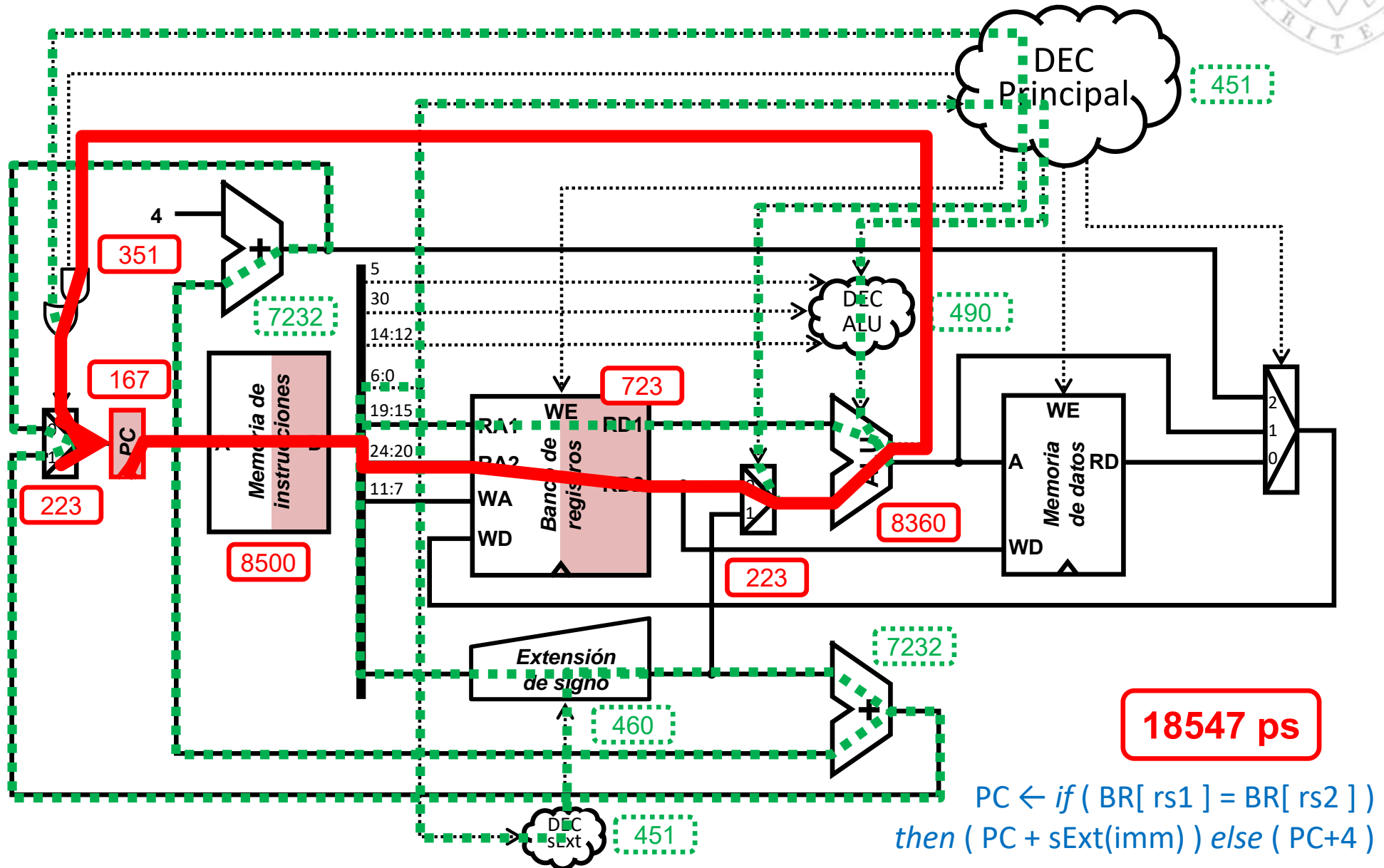
## Instrucción **beq**: camino crítico

versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

131





# Cálculo del tiempo de ciclo

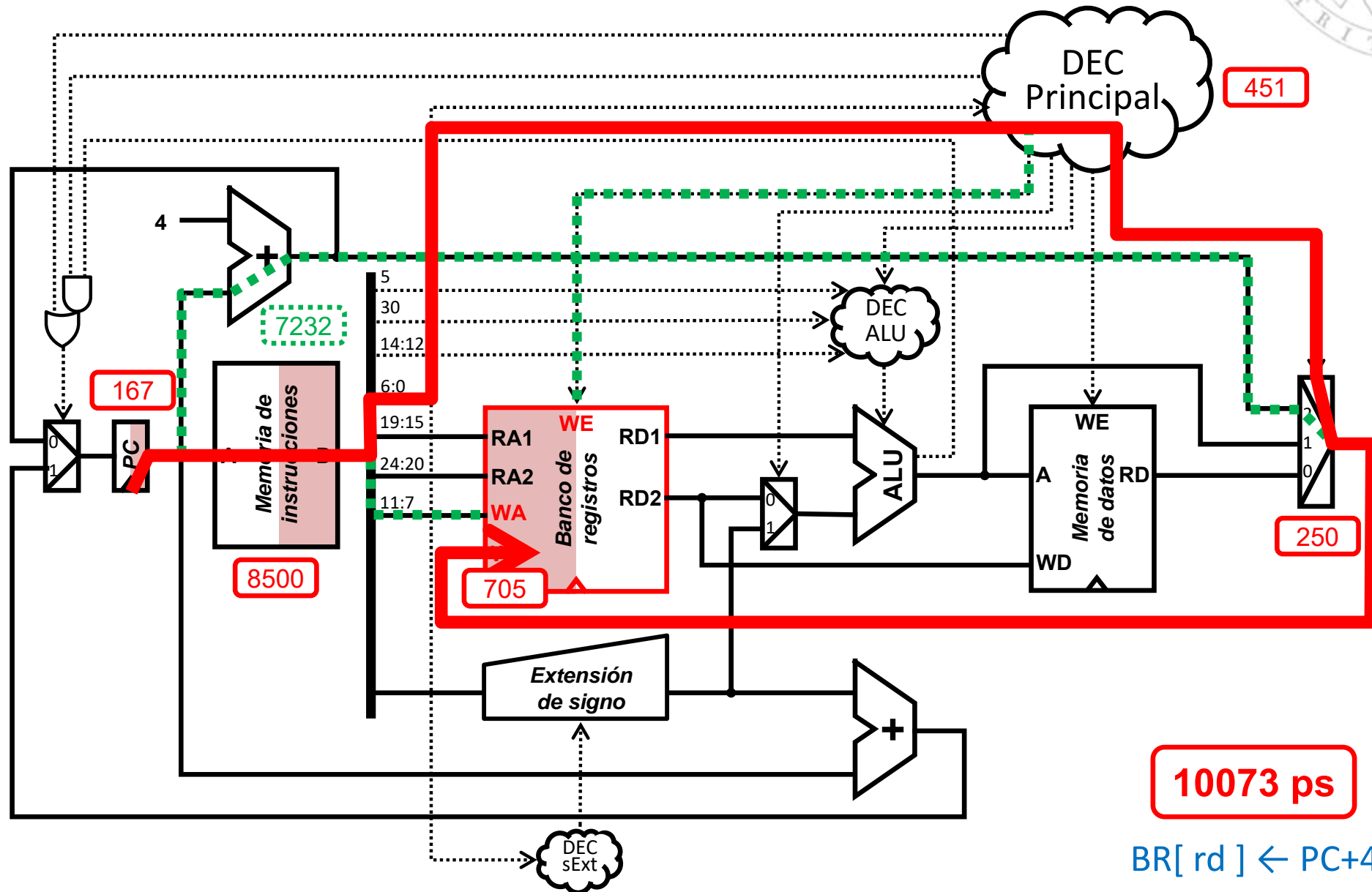
Instrucción **jal** (almacenaje dir. retorno): camino crítico

versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

132





# Cálculo del tiempo de ciclo

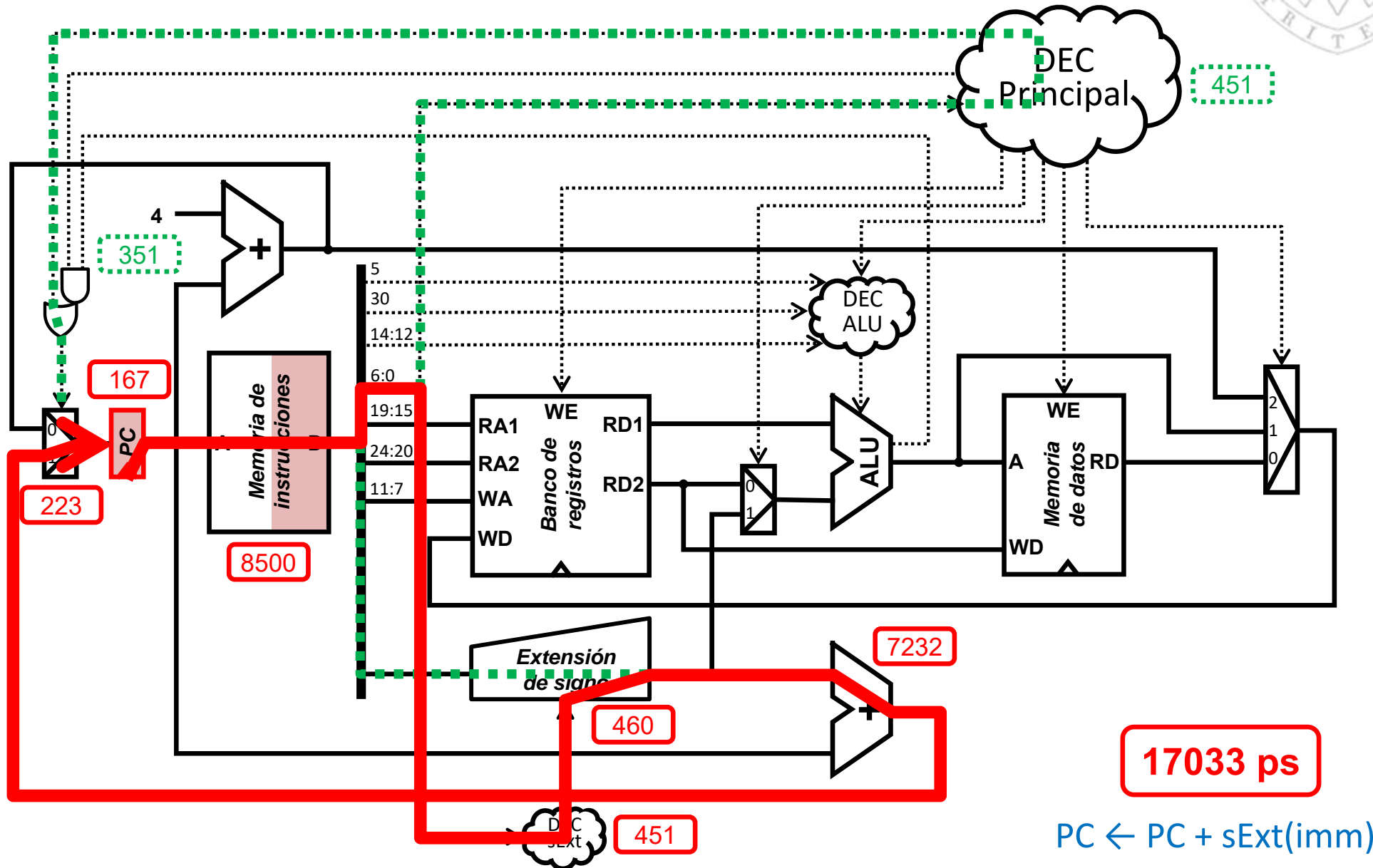
## Instrucción **ja1** (actualización PC): camino crítico

versión 31/10/23

tema 5:  
Diseño monociclo del procesador

FC-2

133



# Acerca de *Creative Commons*



## ■ Licencia CC (**Creative Commons**)

- Ofrece algunos derechos a terceras personas bajo ciertas condiciones. Este documento tiene establecidas las siguientes:



**Reconocimiento** (*Attribution*):

En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.



**No comercial** (*Non commercial*):

La explotación de la obra queda limitada a usos no comerciales.



**Compartir igual** (*Share alike*):

La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

**Más información:** <https://creativecommons.org/licenses/by-nc-sa/4.0/>