

# Tema 3

---

EL PROCESADOR  
EJERCICIOS

# Ejercicio 1

- Considera el siguiente porcentaje de instrucciones ejecutadas:

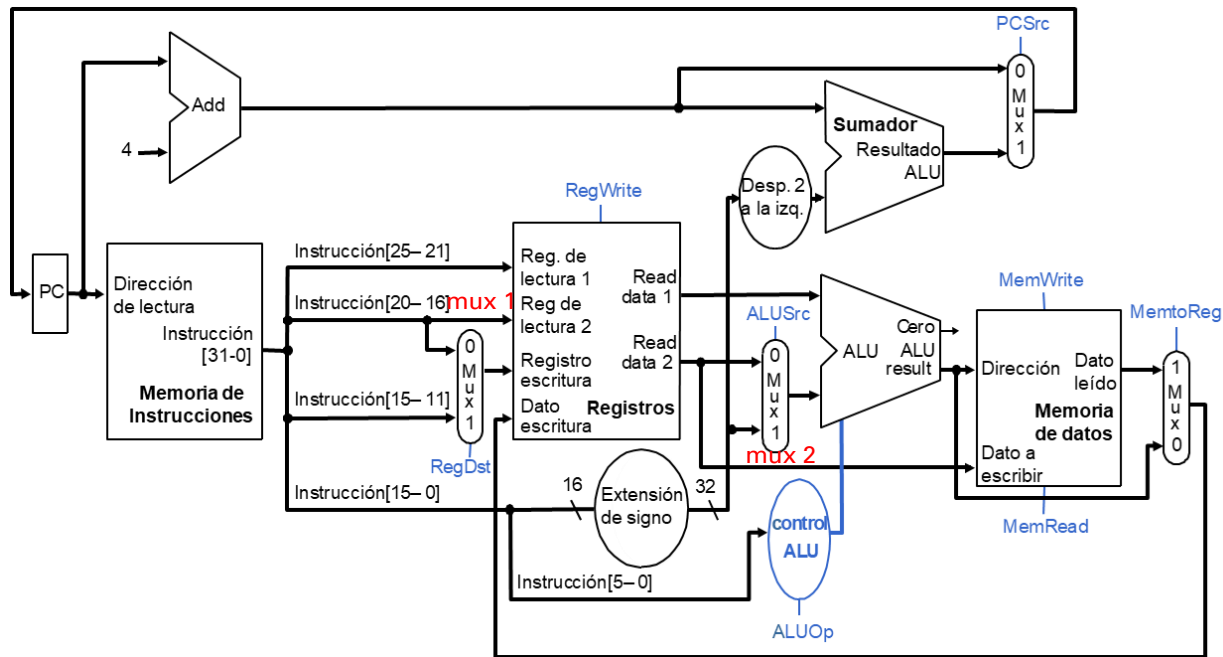
Tipo R	Tipo I (no lw o sw)	lw	sw	Branch	Jump
24%	28%	25%	10%	11%	2%

- ¿Qué fracción de todas las instrucciones ejecutadas usan la memoria de datos?
- ¿Qué fracción de todas las instrucciones ejecutadas usan la memoria de instrucciones?
- ¿Qué fracción de todas las instrucciones ejecutadas usan la extensión de signo?
- ¿Qué está haciendo la extensión de signo durante los ciclos en que su salida no es necesaria?

## Ejercicio 2

- Cuando se fabrican chips de silicio, los defectos en los materiales (por ejemplo, el silicio) y los errores de fabricación pueden dar lugar a circuitos defectuosos. Un defecto muy común es que uno de los cables de señal se "rompa" y registre siempre un 0 lógico. A menudo se denomina fallo "stuck-at-0" ("atascado en 0").
  - ¿Qué instrucciones no funcionan correctamente si la señal *MemtoReg* está atascada en 0?
  - ¿Qué instrucciones no funcionan correctamente si la señal *ALUSrc* está atascada en 0?

## Ejercicio 3



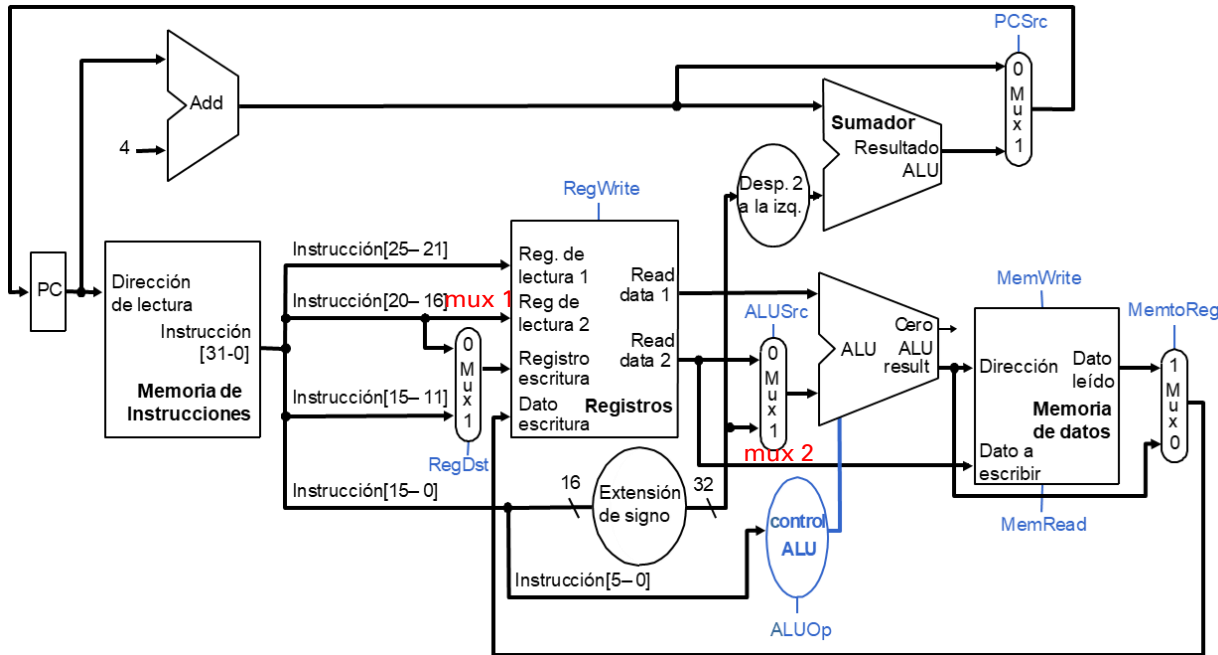
	BR	Contenido	
Núm Registro	Valor Inicial	Final ciclo 1	Final ciclo 2
\$0	0		
\$1	1		
\$2	4		
\$3	5		
\$4	-5		
\$5	4		
\$6	8		
\$7	1		
\$8	-4		
\$9	1		
\$17	9		
\$18	3		
\$31	-32		

	Valor inicial	Final ciclo1	Final ciclo2
PC	0x11000010		
Salida ALU		ciclo1	ciclo 2
Salida mux1		ciclo1	ciclo 2
Salida mux2			ciclo 2

Memoria Instrucciones	
Dirección	Contenido
.	.
.	.
.	.
0x11000010	0x8C430010
0x11000014	0x02324020
0x11000018	0x1023000C
.	.
.	.
.	.

Memoria de Datos			
Dirección	Valor inicial	Final ciclo 1	Final ciclo 2
.	.	.	.
.	.	.	.
.	.	.	.
0x00000010	0x00000007		
0x00000014	0x00000003		
0x00000018	0x00000004		
0x0000001C	0x1111811A		
0x00000020	0xFFFFFFFF		
.	.	.	.
.	.	.	.
.	.	.	.

# Ejercicio 3 –solución parcial



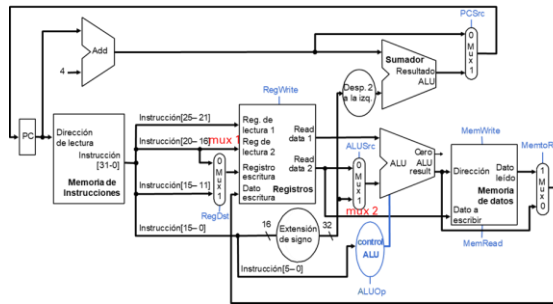
Núm Registro	BR Contenido		
	Valor Inicial	Final ciclo 1	Final ciclo 2
\$0	0		
\$1	1		
\$2	4		
\$3	5	3	
\$4	-5		
\$5	4		
\$6	8		
\$7	1		
\$8	-4		
\$9	1		
\$17	9		
\$18	3		
\$31	-32		

	Valor inicial	Final ciclo1	Final ciclo2
PC	0x11000010	0x11000014	
		ciclo1	ciclo 2
Salida ALU		0x00000014 = 20 <sub>10</sub>	
		ciclo1	ciclo 2
Salida mux1		3	
		ciclo1	ciclo 2
Salida mux2		0x00000010 = 16 <sub>10</sub>	

Memoria Instrucciones	
Dirección	Contenido
.	.
.	.
.	.
0x11000010	0x8C430010
0x11000014	0x02324020
0x11000018	0x1023000C
.	.
.	.

Memoria de Datos			
Dirección	Valor inicial	Final ciclo 1	Final ciclo 2
.	.	.	.
.	.	.	.
.	.	.	.
0x00000010	0x00000007		
0x00000014	0x00000003		
0x00000018	0x00000004		
0x0000001C	0x1111811A		
0x00000020	0xFFFFFFFF		
.	.	.	.
.	.	.	.
.	.	.	.

# Ejercicio 3 –solución parcial



## Explicación:

Se lee de la memoria de instrucciones la instrucción a ejecutar:

$M[0x11000010] = 0x8C430010$

Se decodifica la instrucción:

100011 00010 00011 000000000010000

Los 6 primeros bits corresponden al C.O. e indican que es una lw

Corresponde a: lw \$3, 16(\$2)

A la salida de mux 1 aparece el número del registro (3) donde se guardará el dato ( $\$rt = \$3$ )

A la salida de mux2 aparece la segunda entrada de la ALU que es el

Desplazamiento de la instrucción extendida a 32 bits =  $0x00000010 = 16_{10}$

A la salida de la ALU aparecerá la suma del contenido del registro  $\$rs = \$2$  que es 4 más el desplazamiento, es decir =  $0x00000014 = 20_{10}$

Se accede a la memoria de datos a esa dirección:  $M[0x00000014] = 3_{10}$

Y por último se guarda el dato leído de la memoria en el registro  $\$rt = \$3$

Núm Registro	BR Contenido		
	Valor Inicial	Final ciclo 1	Final ciclo 2
\$0	0		
\$1	1		
\$2	4		
\$3	5	3	
\$4	-5		
\$5	4		
\$6	8		
\$7	1		
\$8	-4		
\$9	1		
\$17	9		
\$18	3		
\$31	-32		

	Valor inicial	Final ciclo1	Final ciclo2
PC	0x11000010	0x11000014	
		ciclo1	ciclo 2
Salida ALU		0x00000014 =20 <sub>10</sub>	
		ciclo1	ciclo 2
Salida mux1		3	
		ciclo1	ciclo 2
Salida mux2		0x00000010 =16 <sub>10</sub>	
		ciclo1	ciclo 2

Memoria Instrucciones		Memoria de Datos			
Dirección	Contenido	Dirección	Valor inicial	Final ciclo 1	Final ciclo 2
.	.	.	.	.	.
0x11000010	0x8C430010	0x00000010	0x00000007		
0x11000014	0x02324020	0x00000014	0x00000003		
0x11000018	0x1023000C	0x00000018	0x00000004		
.	.	0x0000001C	0x1111811A		
.	.	0x00000020	0xFFFFFFFF		
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.

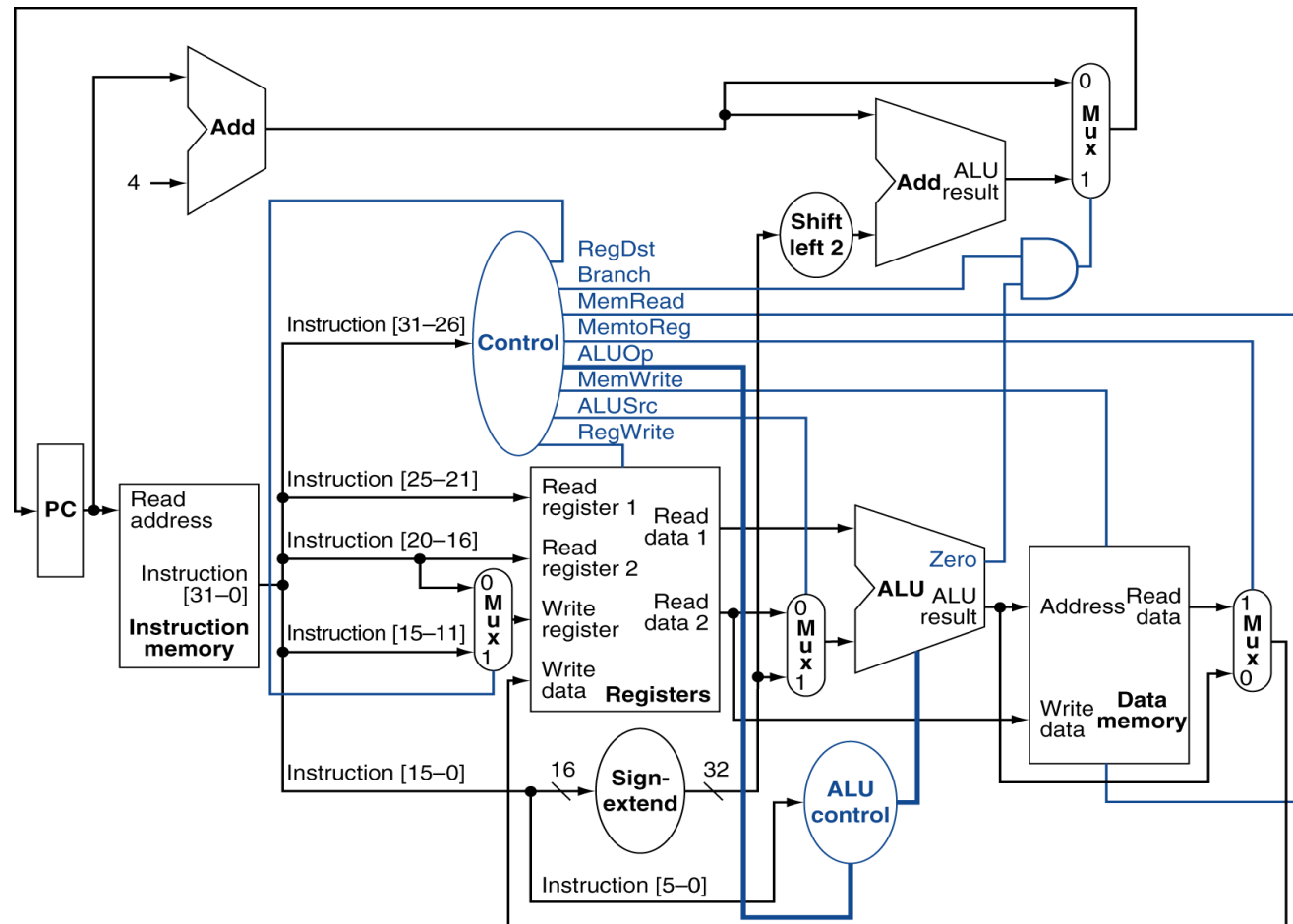
# Ejercicio 4

- Suponer que los bloques lógicos utilizados para implementar la ruta de datos tienen las siguientes latencias:

Memoria	Banco de Registros (solo lectura)	MUX	ALU	Sumador	Puerta simple	Lectura registros	Configuración de registro	Extensión de signo	Control
250ps	150ps	25ps	200ps	150ps	5ps	30ps	20ps	50ps	50ps

- “Lectura registros” es el tiempo que se necesita después del flanco de reloj para que el nuevo valor aparezca en la salida. Este valor se aplica solo al PC. “Configuración de registro” es la cantidad de tiempo que el dato de entrada a un registro debe estar estable antes del flanco de subida del reloj. El valor se aplica solo al PC y al banco de registros.
  - ¿Cuál es la latencia de las instrucciones tipo R? ( ¿es decir, qué duración debe de tener el ciclo de reloj para asegurar que estas instrucciones trabajen correctamente?)
  - ¿Cuál es la latencia de la instrucción lw?
  - ¿Cuál es la latencia de la instrucción sw?
  - ¿Cuál es la latencia de la instrucción beq?
  - ¿Cuál es la latencia de las instrucciones aritméticas, lógicas o de desplazamiento de tipo I?
  - ¿Cuál debe ser el periodo de reloj mínimo de esta CPU?

# Ejercicio 4



Memoria	Banco de Registros (solo lectura)	MUX	ALU	Sumador	Puerta simple	Lectura registros	Configuración de registro	Extensión de signo	Control
250ps	150ps	25ps	200ps	150ps	5ps	30ps	20ps	50ps	50ps



# Ejercicio 4 – solución parcial

- Suponer que los bloques lógicos utilizados para implementar la ruta de datos tienen las siguientes latencias:

Memoria	Banco de Registros (solo lectura)	MUX	ALU	Sumador	Puerta simple	Lectura registros	Configuración de registro	Extensión de signo	Control
250ps	150ps	25ps	200ps	150ps	5ps	30ps	20ps	50ps	50ps

El tiempo de control no tiene aplicación porque está en paralelo al acceso al banco de registros, lo mismo que el sumador, en paralelo a la ALU o el incremento del PC, en paralelo a la memoria, o la extensión de signo.

- ¿Cuál es la latencia de las instrucciones tipo R? ( ¿es decir, qué duración debe de tener el ciclo de reloj para asegurar que estas instrucciones trabajen correctamente?

$$\text{Tipo R} = 30 + 250 + 150 + 25 + 200 + 25 + 20 = 700\text{ps}$$

- ¿Cuál es la latencia de la instrucción lw?

$$\text{Lw} = 30 + 250 + 150 + 25 + 200 + 250 + 25 + 20 = 950\text{ps}$$

- .....

- ¿Cuál debe ser el periodo de reloj mínimo de esta CPU?

$$950\text{ps}$$

## Ejercicio 5

- Suponed que se pudiera construir una CPU donde la duración del ciclo de reloj fuera diferente para cada instrucción, ¿Cuál sería la mejora conseguida con esta nueva CPU respecto de la CPU del ejercicio 4 dados los siguientes porcentajes de instrucciones ejecutadas?

Tipo R/ Tipo I (no lw,sw)	lw	sw	beq
52%	25%	11%	12%

- Utilizad los resultados del ejercicio 4

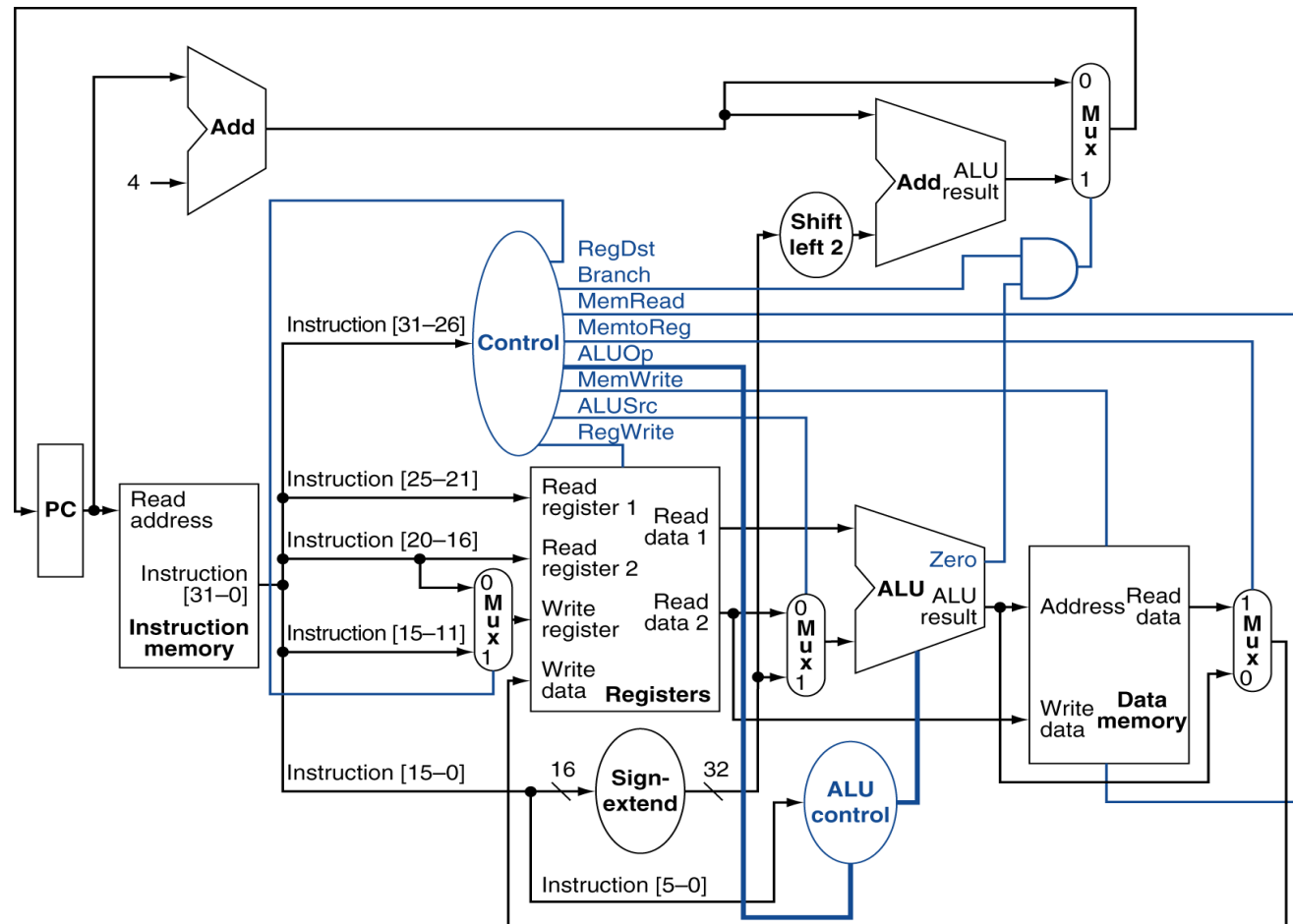
# Ejercicio 6

- Suponed que se quiere añadir una nueva instrucción al conjunto de instrucciones que se ejecutan en la ruta de datos monociclo estudiada. Dicha instrucción es: lwi rd, rs1, rs2 (“load con incremento”). Lo que hace esta instrucción es:

$$BR[rd] \leftarrow M[rs1 + rs2]$$

- ¿Se necesita añadir algún bloque funcional nuevo para que se pueda ejecutar la instrucción?
- ¿Se necesita modificar algún bloque funcional existente en la ruta de datos para que se pueda ejecutar la instrucción?
- ¿Se necesita modificar la ruta de datos para que se pueda ejecutar la instrucción?
- ¿Se necesita añadir alguna señal de control nueva para que se pueda ejecutar la instrucción?
- ¿Cuál sería el valor de las señales de control para que se ejecute la nueva instrucción?

# Ejercicio 6



$$BR[rd] \leftarrow M[rs1 + rs2]$$

# Ejercicio 6 solución

- lwi rd, rs1, rs2 (“load con incremento”):  $BR[rd] \leftarrow M[rs1 + rs2]$ 
  - ¿Se necesita añadir algún bloque funcional nuevo para que se pueda ejecutar la instrucción?

No se necesita añadir ni modificar nada. Solo la unidad de control sería necesario modificar para incluir la nueva instrucción

- ¿Se necesita modificar algún bloque funcional existente en la ruta de datos para que se pueda ejecutar la instrucción? **no**
- ¿Se necesita modificar la ruta de datos para que se pueda ejecutar la instrucción? **no**
- ¿Se necesita añadir alguna señal de control nueva para que se pueda ejecutar la instrucción? **no**
- ¿Cuál sería el valor de las señales de control para que se ejecute la nueva instrucción?

Señales de control (sin jump):  $ALUsrc=0$ ,  $ALUOp=00$ ,  $MemRead=1$ ,  $MemToReg=1$ ,  $RegDst=1$ ,  $RegWrite=1$

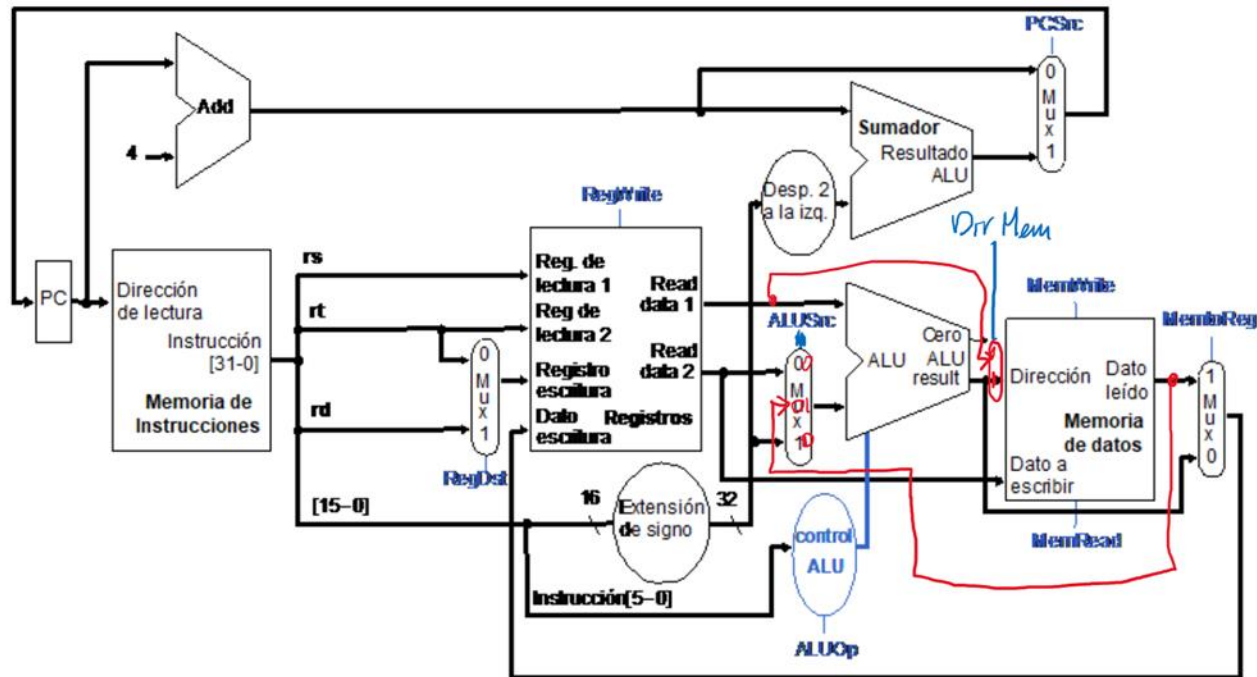
# Ejercicio 7

- Partimos de la ruta de datos monociclo estudiada que permite ejecutar instrucciones tipo-R, lw, sw y beq. Se está considerando la posibilidad de introducir nuevas instrucciones en esta versión monociclo que permita realizar operaciones aritmético-lógicas con datos que se encuentren en la memoria. Estas instrucciones utilizarán también el formato tipo R y su forma general será  $rd \leftarrow rs \text{ op } M[rs]$  donde op puede ser una suma, resta, and, or, slt. Por ejemplo, para la suma su forma general será:

Addm rs, rt, rd     $\#rd \leftarrow rs + M[rs]$

- Modifica la ruta de datos monociclo para que se puedan ejecutar estas nuevas instrucciones incluyendo las señales de control que sean necesarias
- Obtén el valor de las señales de control para que se ejecuten correctamente tanto las instrucciones tipo R originales como las nuevas.
- Supón que las memorias y la ALU tienen un retardo de 2ns y el Banco de Registros un retardo de 1ns. Calcula la duración del ciclo de reloj para nueva la ruta de datos que has modificado y compáralo con la duración del ciclo de reloj de la ruta de datos original.

# Ejercicio 7 – solución parcial



	RegDst	RegWrite	AluSrc	ALUOp	MemWrite	MemRead	MemToReg	PCSrc	<i>Dr Mem</i>			
Tipo-R Original	1	1	00	00	0	0	0	0	X			
Tipo_R Nuevas	1	1	01	00	0	1	X	0	0			

# Propuesta trabajo opcional

- Se propone la realización de un trabajo opcional sobre la temática: Computación Cuántica.
- Se podrá obtener hasta 1 punto extra de la nota en la parte teórica.
- Esta nota no podrá utilizarse para poder mediar con prácticas.
- El trabajo se deberá entregar por escrito con una extensión aproximada de 3500 palabras.
- Se tendrá la oportunidad de realizar una presentación en clase.
- Los contenidos mínimos incluirán el resumen de la visualización de diversos videos:
  - Una conferencia de Juan Ignacio Cirac sobre Computación Cuántica, por ejemplo:
    - [Conferencia de Juan Ignacio Cirac en la XX Semana de la Ciencia del Campus de Alcoy de la UPV](#)
    - [#FutureTalks sobre Quantum Computing con Ignacio Cirac](#)
    - O de más nivel: [Simulación cuántica: realizando el sueño de Feynman. Ignacio Cirac. Facultad de Ciencias Físicas UCM](#)
    - O cualquier otra conferencia sobre computación cuántica que consideréis, aquí os dejo algunos enlaces:
      - [ARQUITECTURA CÓSMICA IX: Ordenadores cuánticos: Promesas, hitos y retos](#)
      - [Foro Nacional de Computación Cuántica y sus implicaciones para la ingeniería industrial](#)
      - [Computación Cuántica: Estado de Desarrollo](#)
  - Y la visualización de al menos un par de vídeos del canal QuantumFracture:
    - [Todo lo que un Qubit puede Enseñarte sobre Física Cuántica](#)
    - [Cómo Mandar un Mensaje Secreto con Física Cuántica | Encriptación](#)