

Clase 22 - Paralelismo

Parte 1

Profesor: **IIC2343 - Arquitectura de Computadores**
- Felipe Valenzuela González
Correo:
frvalenzuela@alumni.uc.cl

Arquitectura de Computadores: Mejoras y extensiones

En lo que hemos visto hasta ahora

- Una máquina programable que ejecuta programas
- Interacción con dispositivos de entrada y salida

Lo que nos queda ver **mejoras**:

- Mejora los accesos a memoria ✓
- Mejora en lo que respecta a la acceso a memoria en más de un procesador ✓
- Mejora en lo que respecta a la ejecución de instrucciones en la CPU

Arquitectura de Computadores: Mejoras y extensiones

En lo que hemos visto hasta ahora

- Una máquina programable que ejecuta programas
- Interacción con dispositivos de entrada y salida

Lo que nos queda ver **mejoras**:

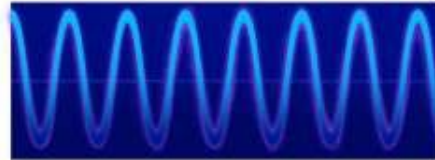
- Mejora los accesos a memoria ✓
- Mejora en lo que respecta a la acceso a memoria en más de un procesador ✓
- Mejora en lo que respecta a la ejecución de instrucciones en la CPU ?

¿Dudas?

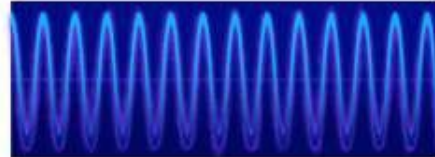
Mejorar rendimiento

- Aumentar la frecuencia de un computador, ¿Problemas?
- ¿Es posible utilizar otro esquema?

Intel® Core™ i9-13900K



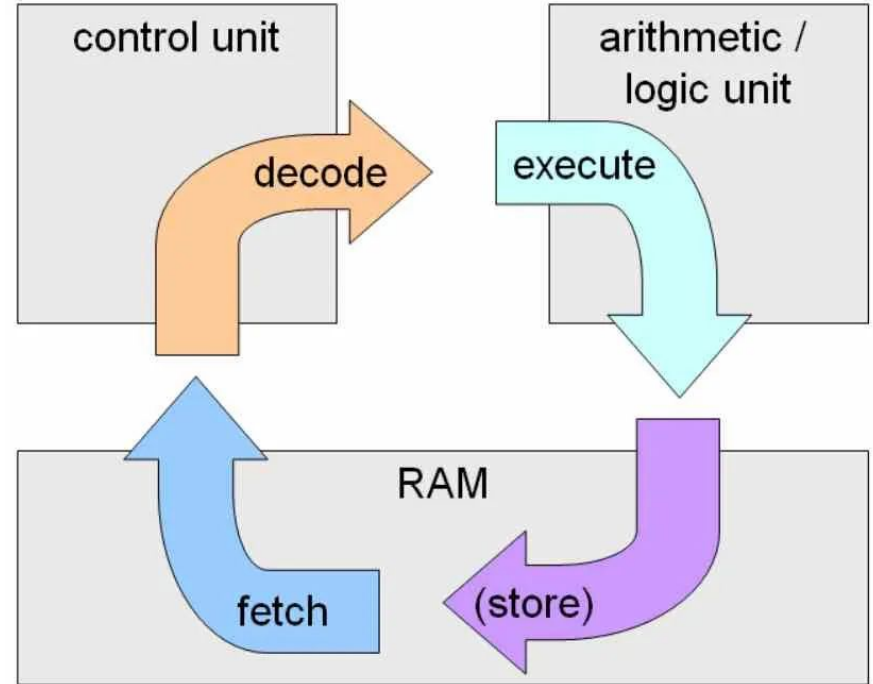
3.00 GHz
(Performance Core™ base frequency)



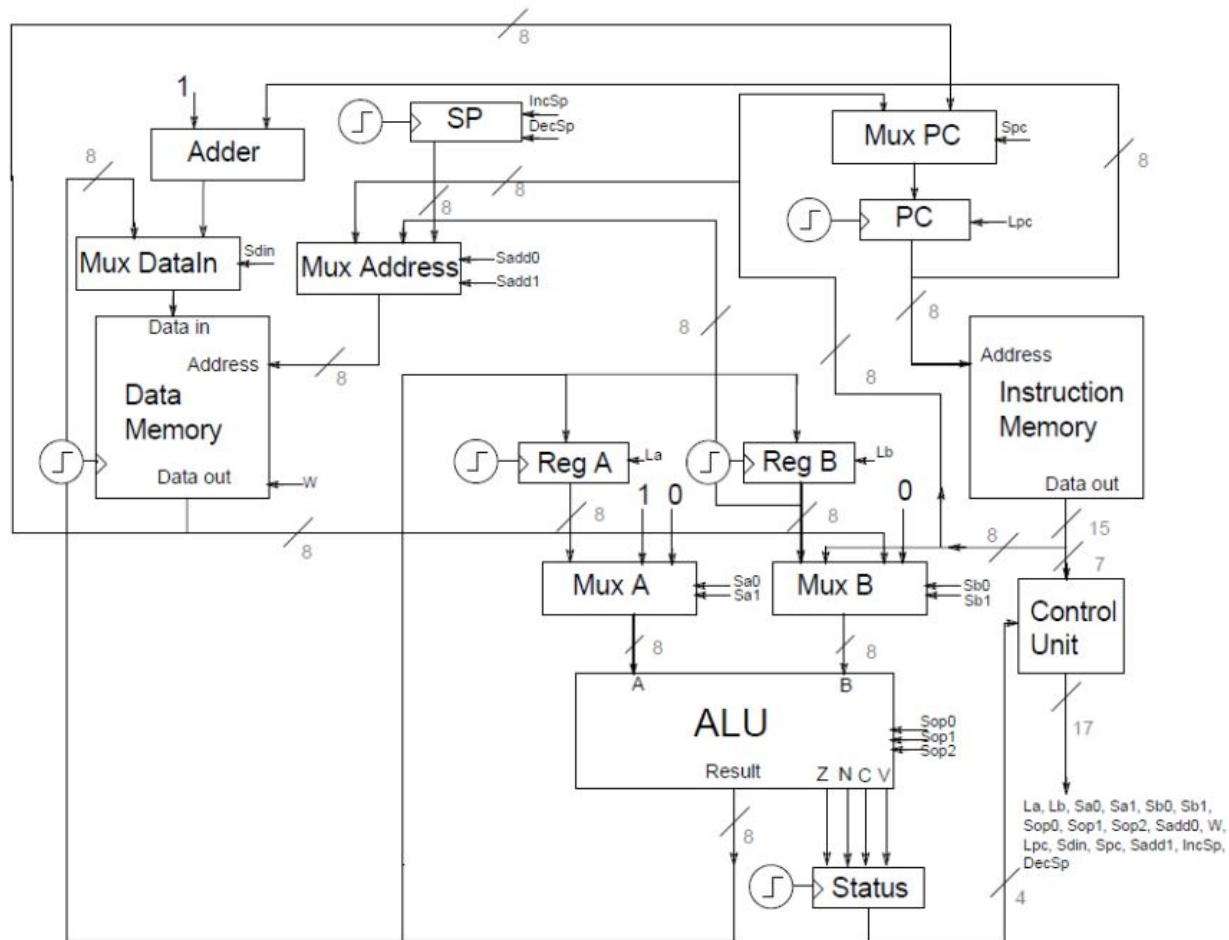
5.80 GHz
(Max Turbo frequency)

Flujo de instrucciones

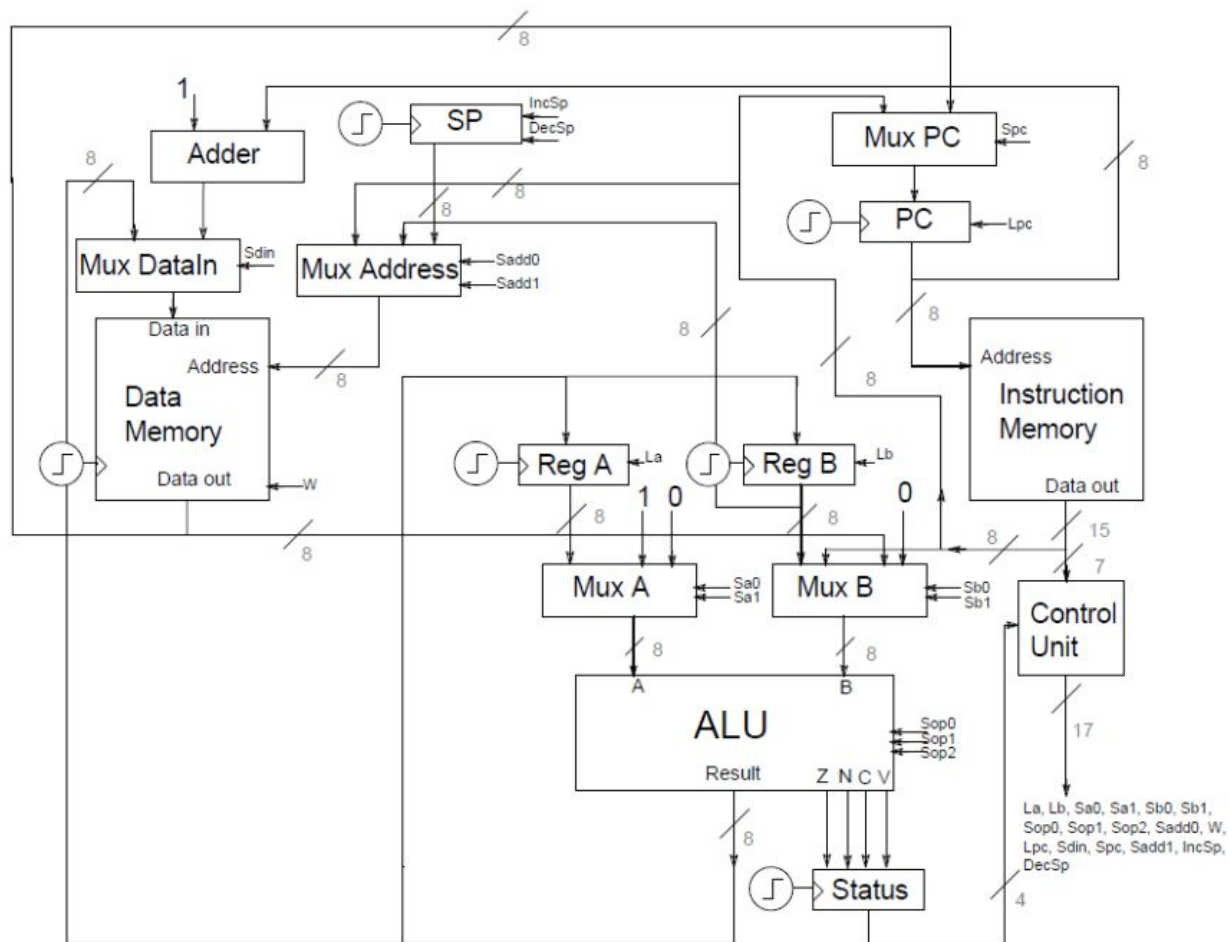
- Analizaremos el **proceso secuencial** de ejecución de las instrucciones
- Analizaremos el proceso secuencial de ejecución de las instrucciones



Recordar:

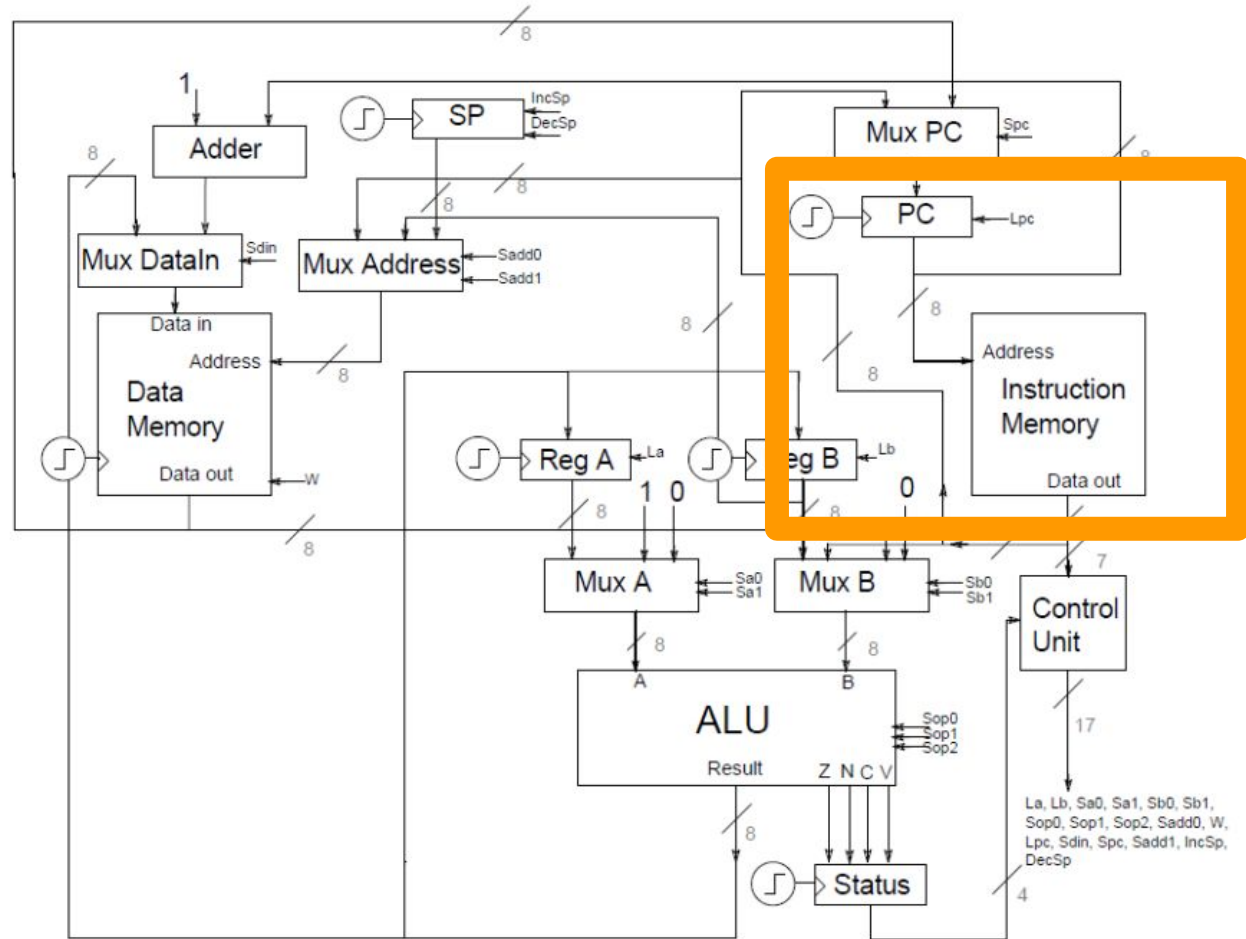


Secuencia: ADD A,B



Secuencia: ADD A,B

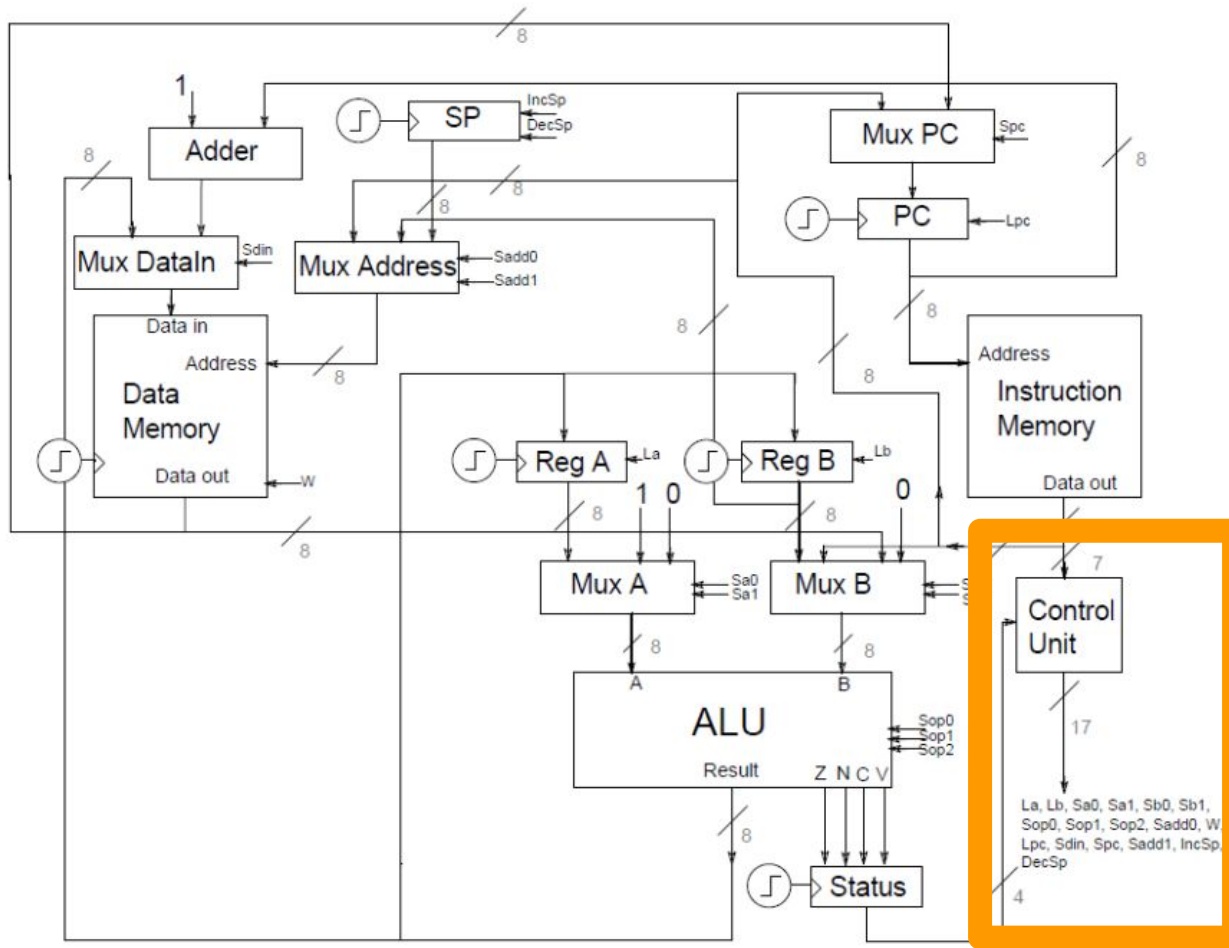
1- Lectura de
instrucción desde
memoria



Secuencia: ADD A,B

1- Lectura de instrucción desde memoria

2-Decodificar instrucción desde memoria

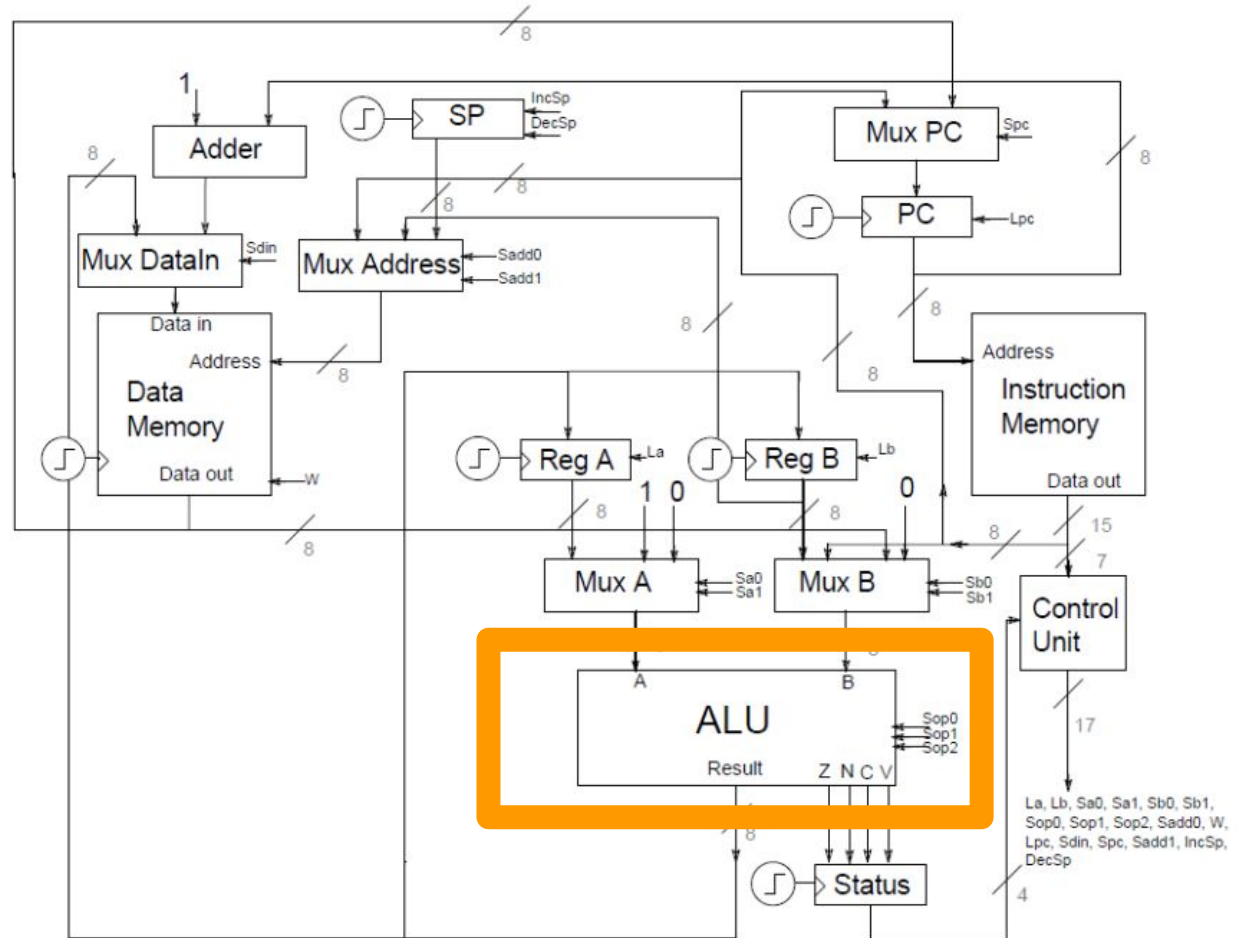


Secuencia: ADD A,B

1- Lectura de
instrucción desde
memoria

2-Decodificar
instrucción desde
memoria

3- Ejecutar en ALU



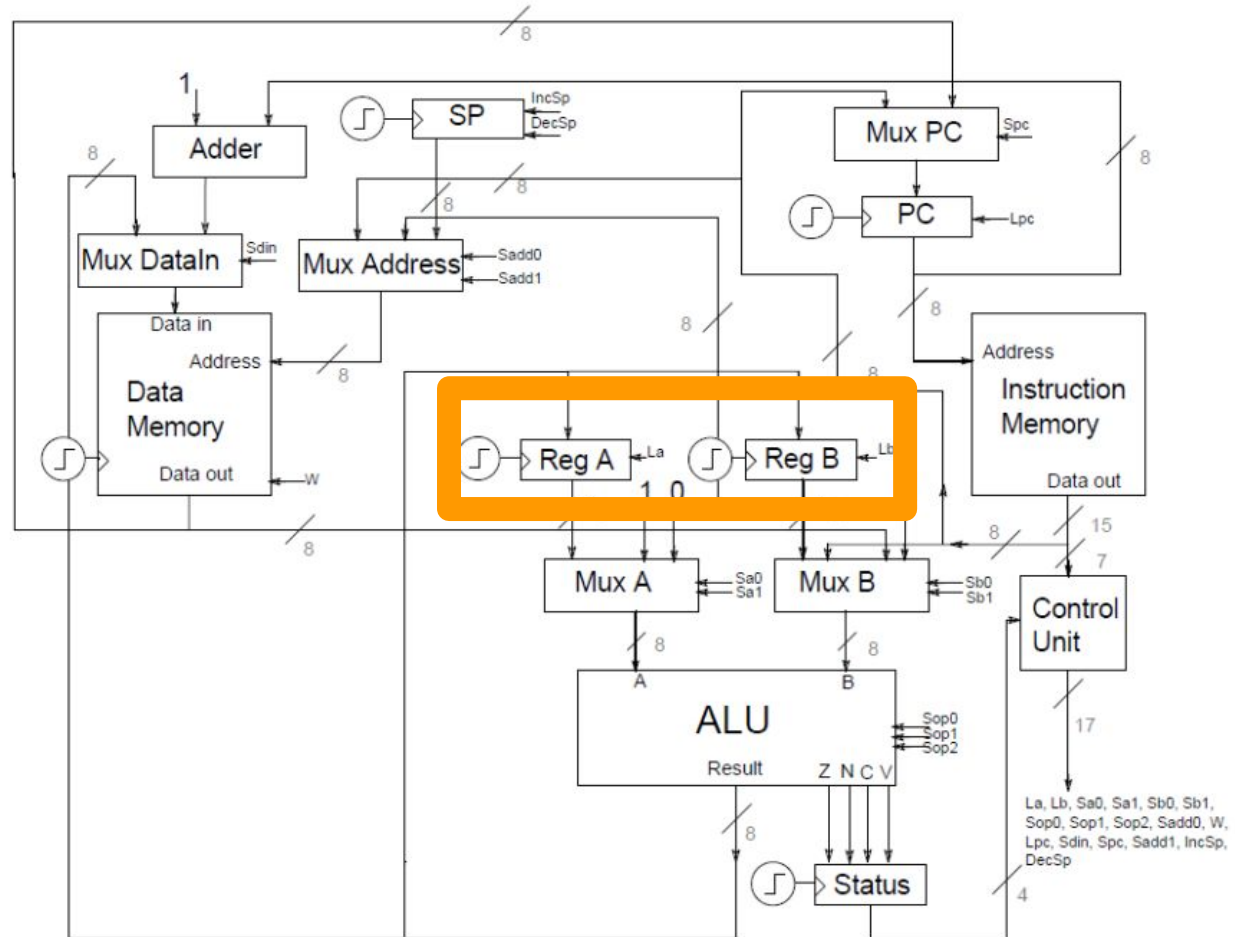
Secuencia: ADD A,B

1- Lectura de instrucción desde memoria

2- Decodificar instrucción desde memoria

3- Ejecutar en ALU

4- Escribir en Registro



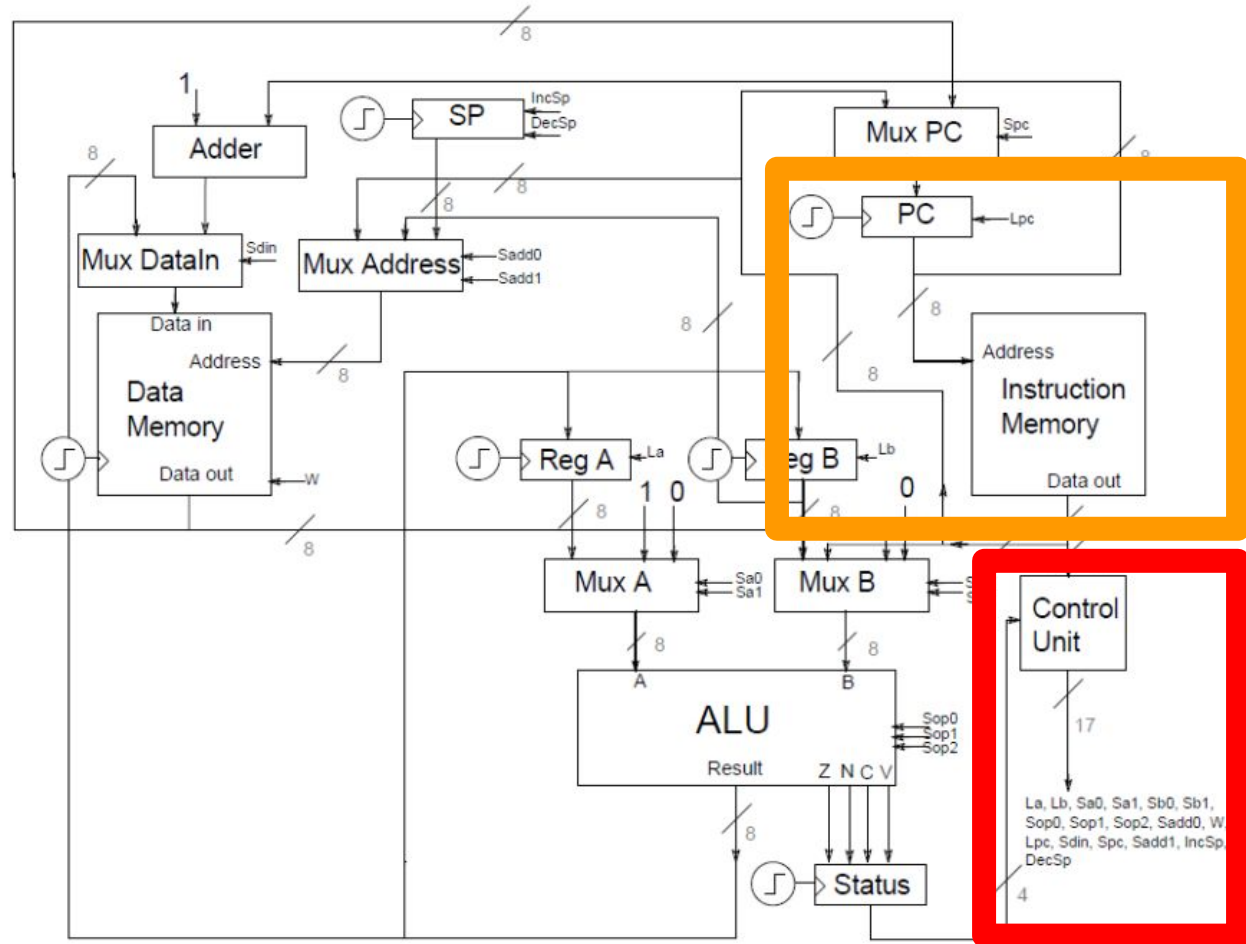
Flujo de la instrucción: **ADD A,B**

- 1- Lectura de instrucción desde memoria (**Fetch**)
- 2-Decodificar instr. en unidad de control (**Decode**)
- 3- Ejecutar en ALU (**Execute**)
- 4- Escribir resultado en registro (**Write Back**)

¿Dudas?

Secuencia: MOV A,(B)

- 1- Lectura de instrucción desde memoria (**Fetch**)
- 2- Decodificar instr. en unidad de control (**Decode**)

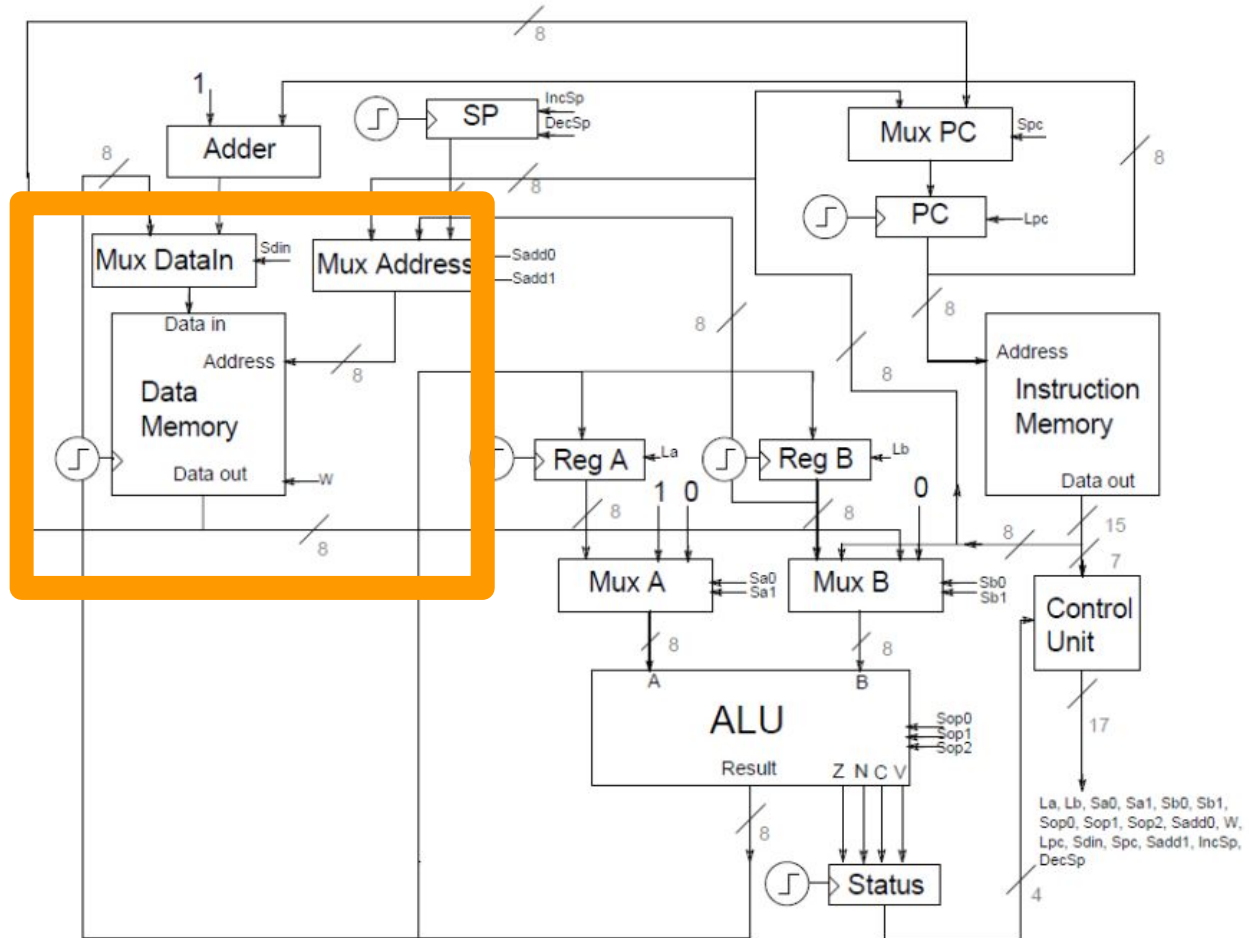


Secuencia: **MOV A,(B)**

1- Lectura de instrucción desde memoria (**Fetch**)

2- Decodificar instr. en unidad de control (**Decode**)

3- Acceder a memoria (**Mem**)



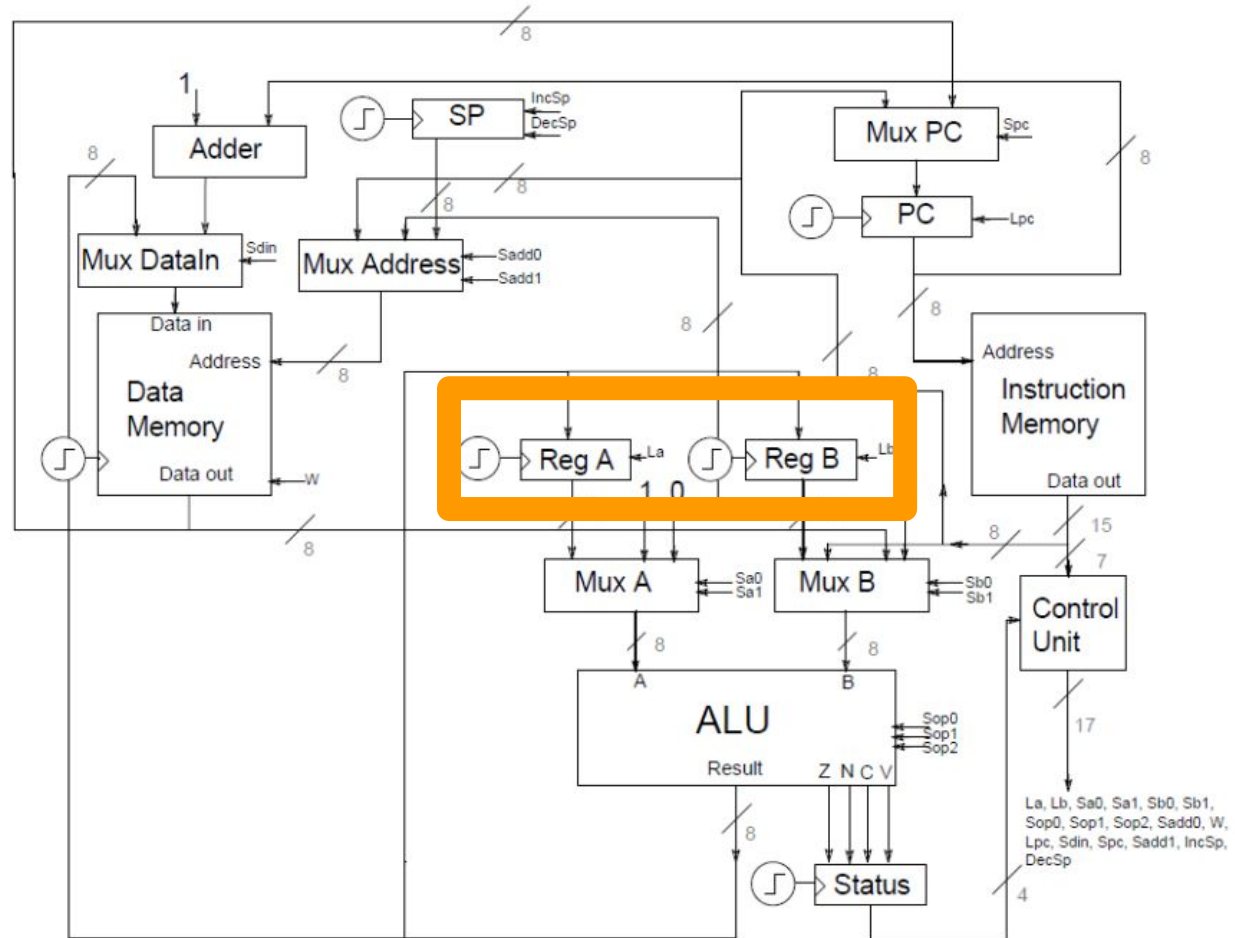
Secuencia: **MOV A,(B)**

1- Lectura de instrucción desde memoria (**Fetch**)

2- Decodificar inst en unidad de control (**Decode**)

3- Acceder a memoria (**Mem**)

4- Escribir resulta en registro (**Write Back**)



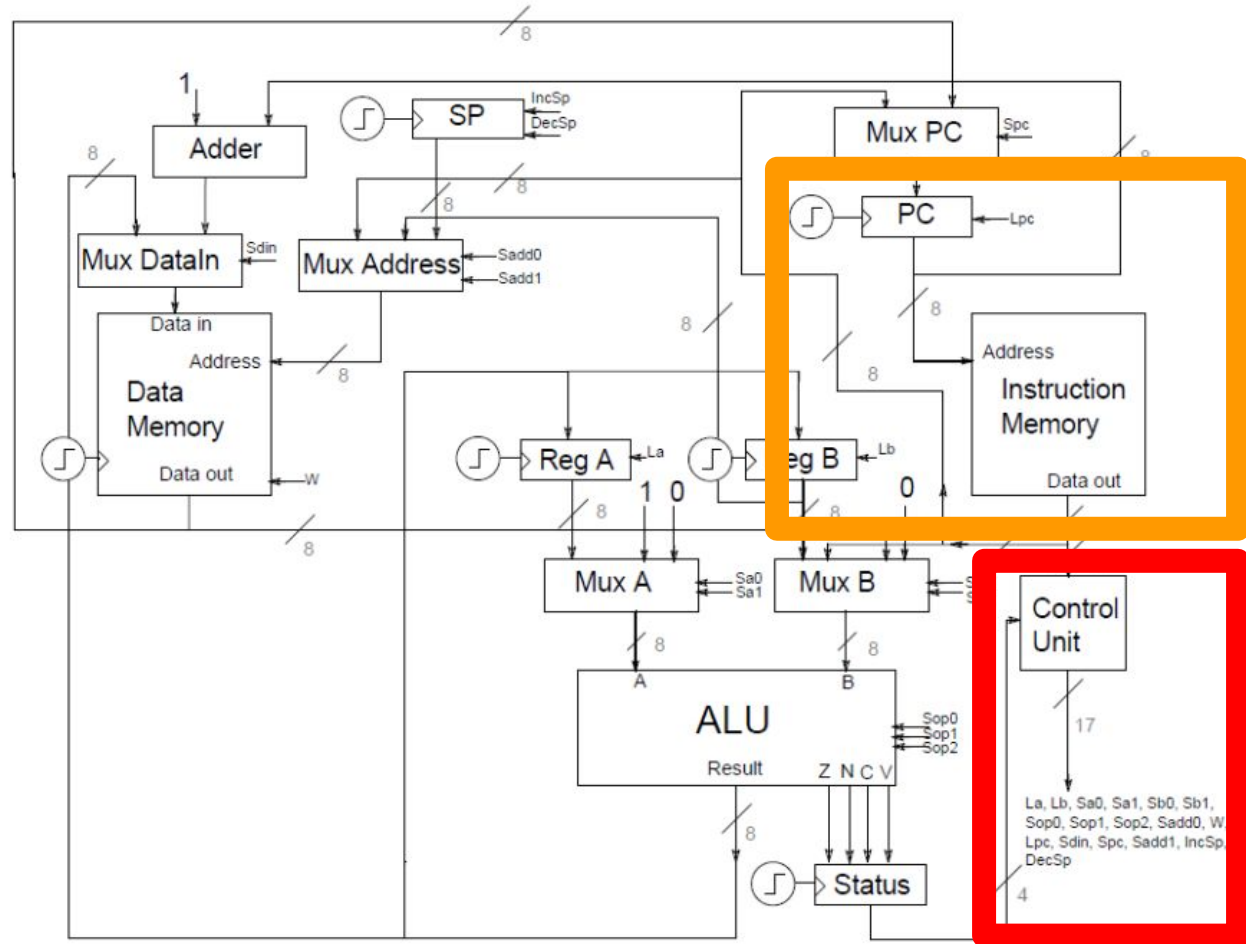
Flujo de la instrucción: **MOV A,(B)**

- 1- Lectura de instrucción desde memoria (**Fetch**)
- 2-Decodificar instr. en unidad de control (**Decode**)
- 3- Acceder a memoria (**Mem**)
- 4- Escribir resultado en registro (**Write Back**)

Secuencia: **MOV (B),A**

1- Lectura de
instrucción desde
memoria (**Fetch**)

2- Decodificar instr.
en unidad de
control (**Decode**)

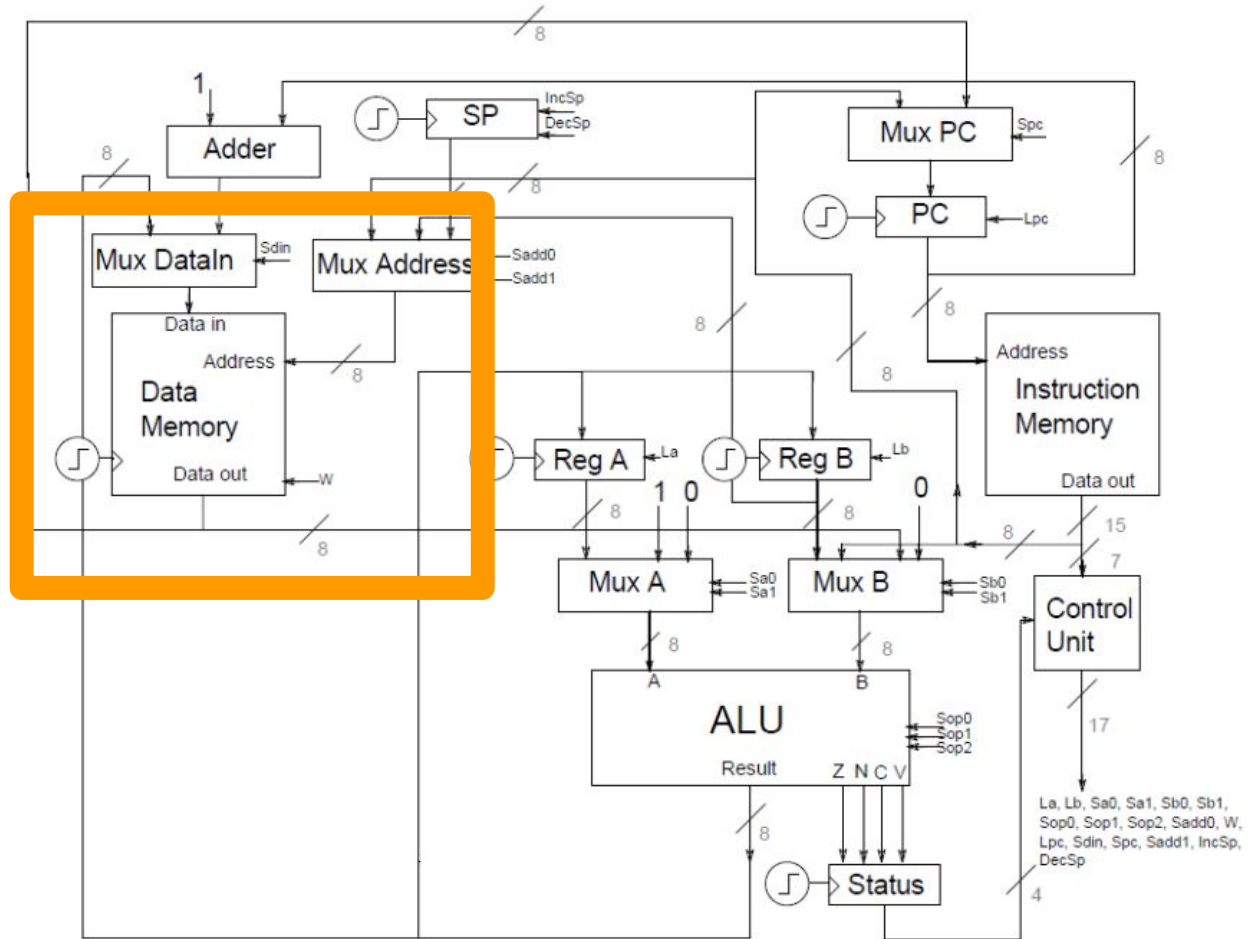


Secuencia: **MOV (B),A**

1- Lectura de instrucción desde memoria (**Fetch**)

2- Decodificar instr. en unidad de control (**Decode**)

3- Acceder a memoria (**Mem**)



Flujo de la instrucción: **MOV (B),A**

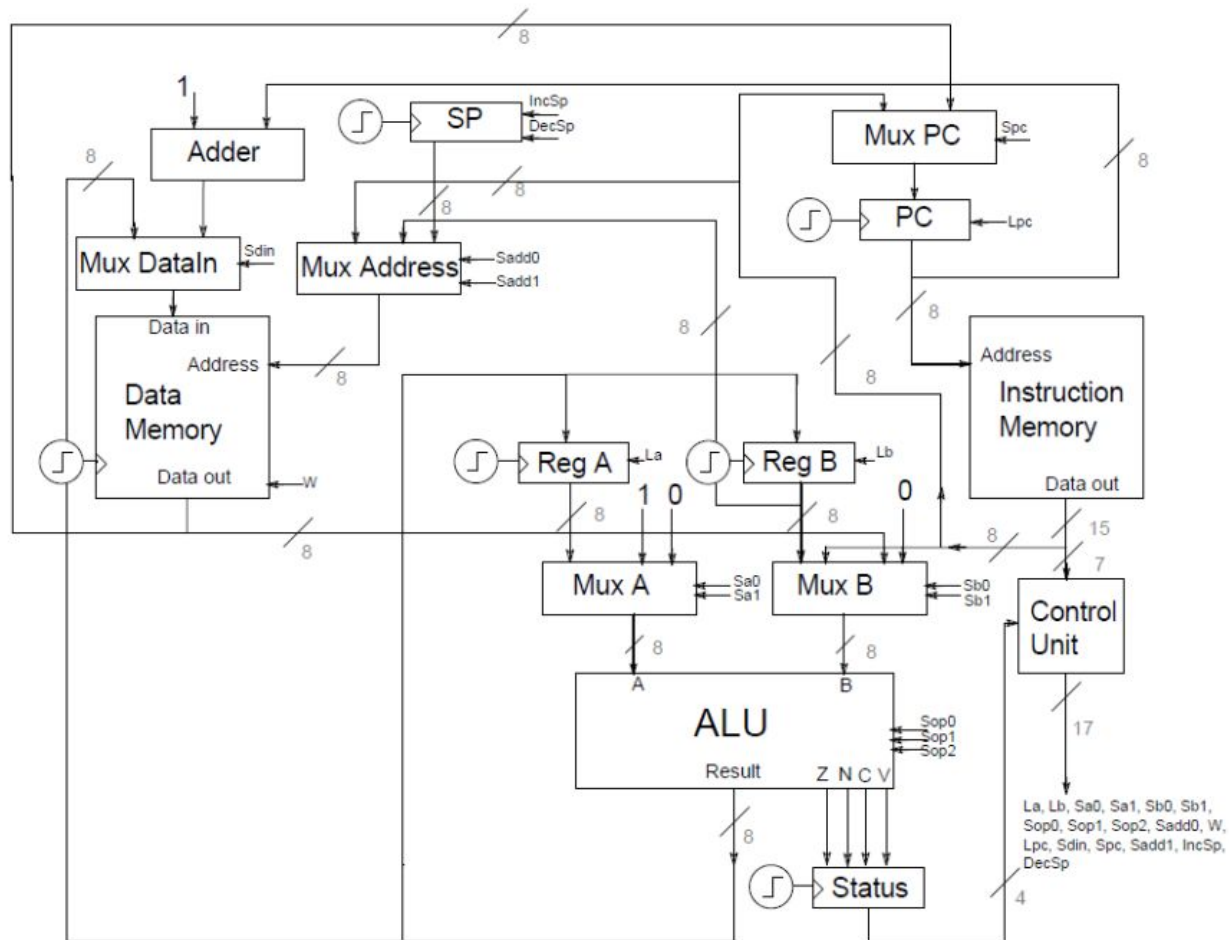
- 1- Lectura de instrucción desde memoria (**Fetch**)
- 2-Decodificar instr. en unidad de control (**Decode**)
- 3- Acceder a memoria (**Mem**)
- 4- Escribir resultado en registro (**Write Back**)

¿Dudas?

Flujo de instrucción: General

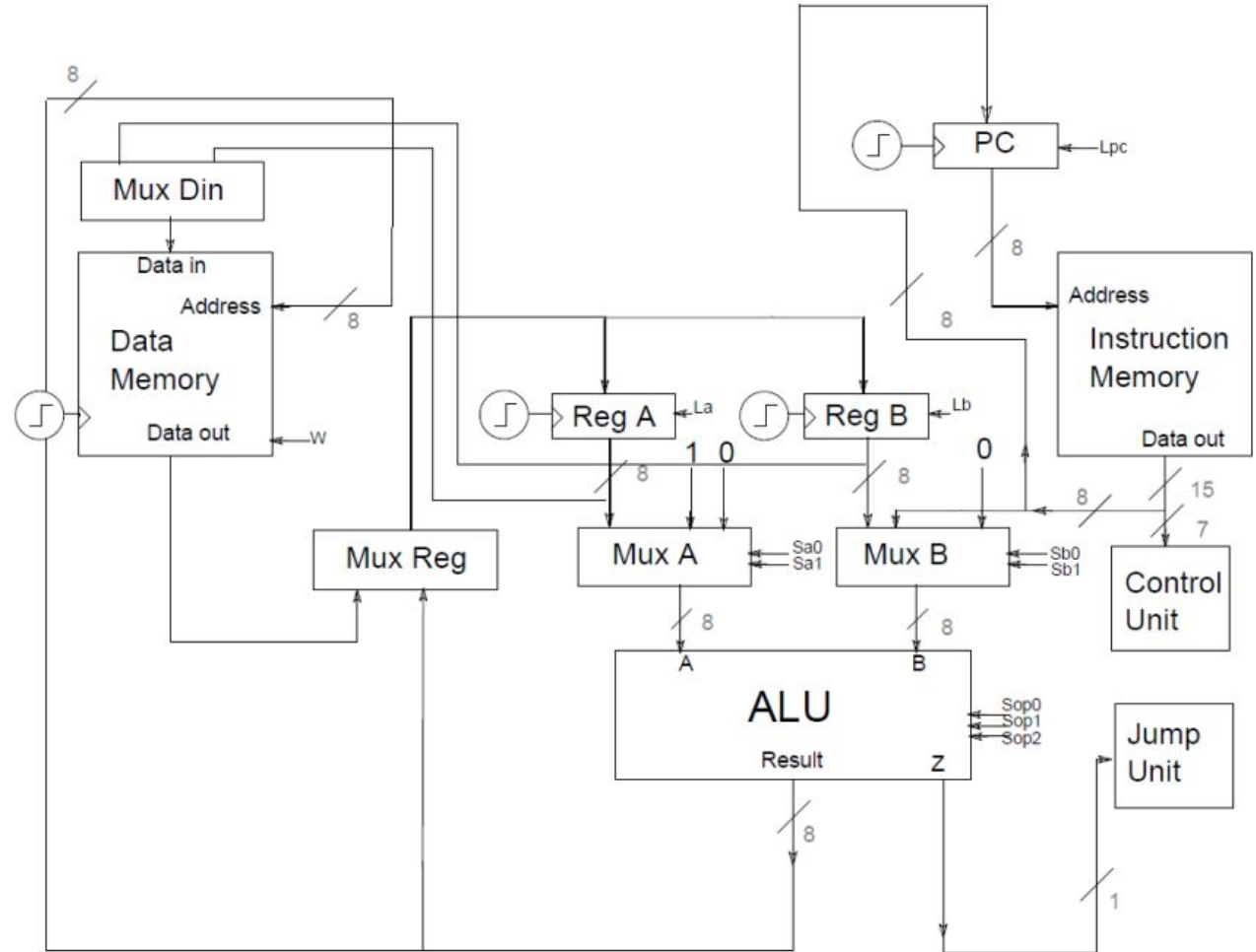
- 1- Lectura de instrucción desde memoria (**Fetch**)
- 2-Decodificar instr. en unidad de control (**Decode**)
- 3-Ejecutar en ALU (**Execute**)
- 4- Acceder a memoria (**Mem**)
- 5- Escribir resultado en registro (**Write Back**)

Recordar:



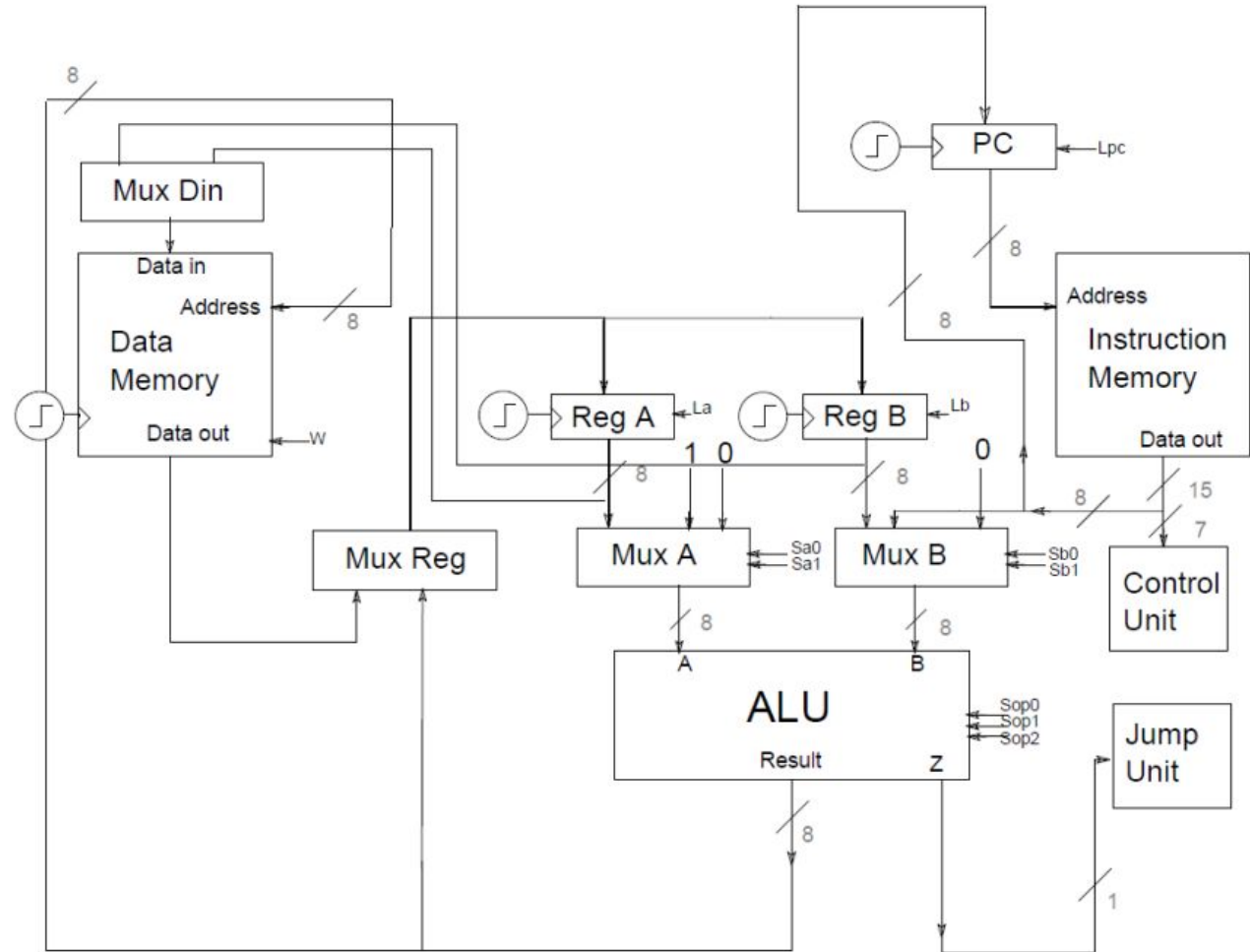
Nuevo Computador

Simplifiquemos la estructura para alinear **con el nuevo flujo de las instrucciones**



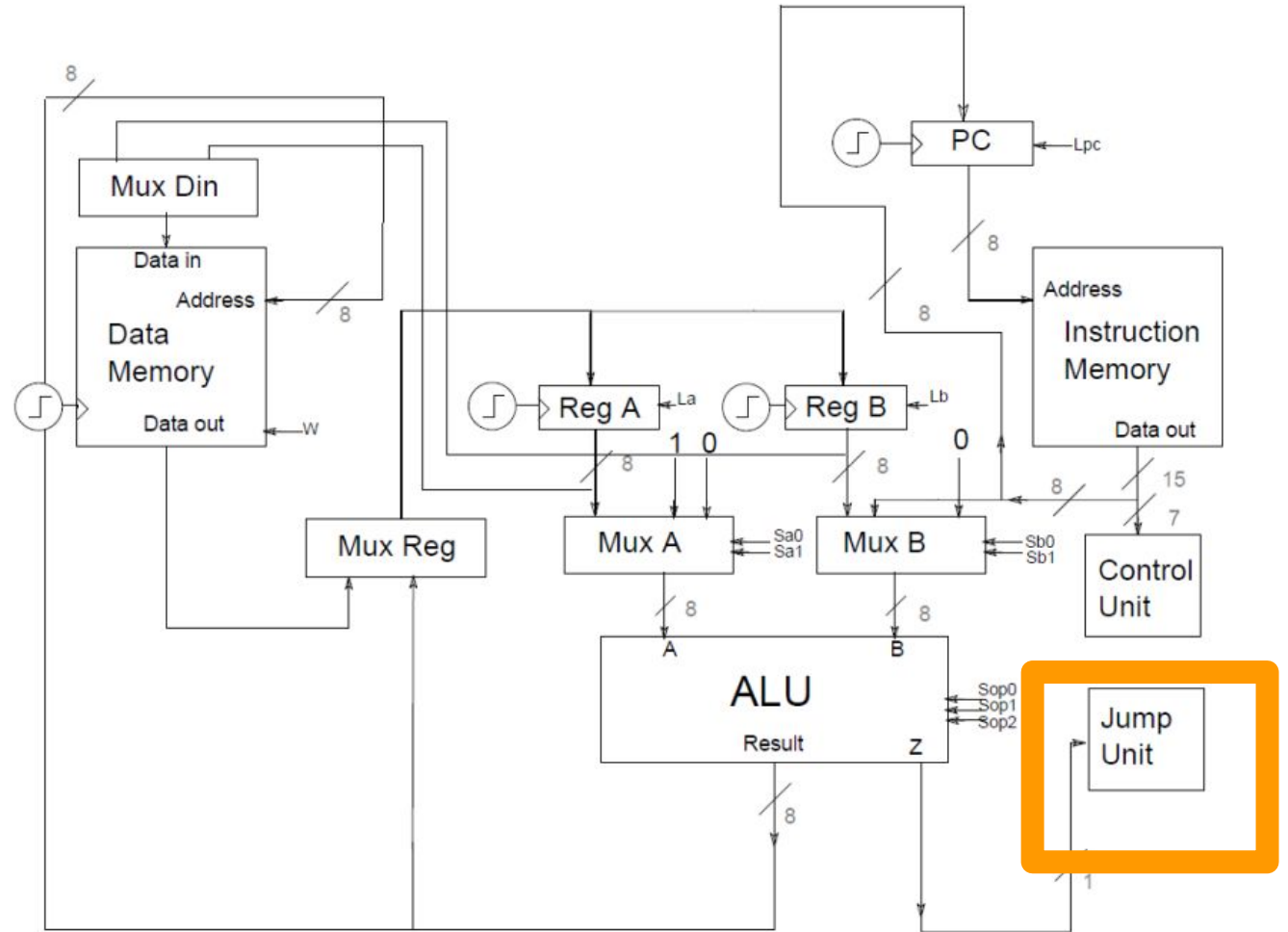
Nuevo Computador

- No hay stack ni subrutinas.
- Solo hay saltos condicionales con JEQ y JNE.
- Saltos ejecutan CMP A,B, por lo que no hay registro Status.



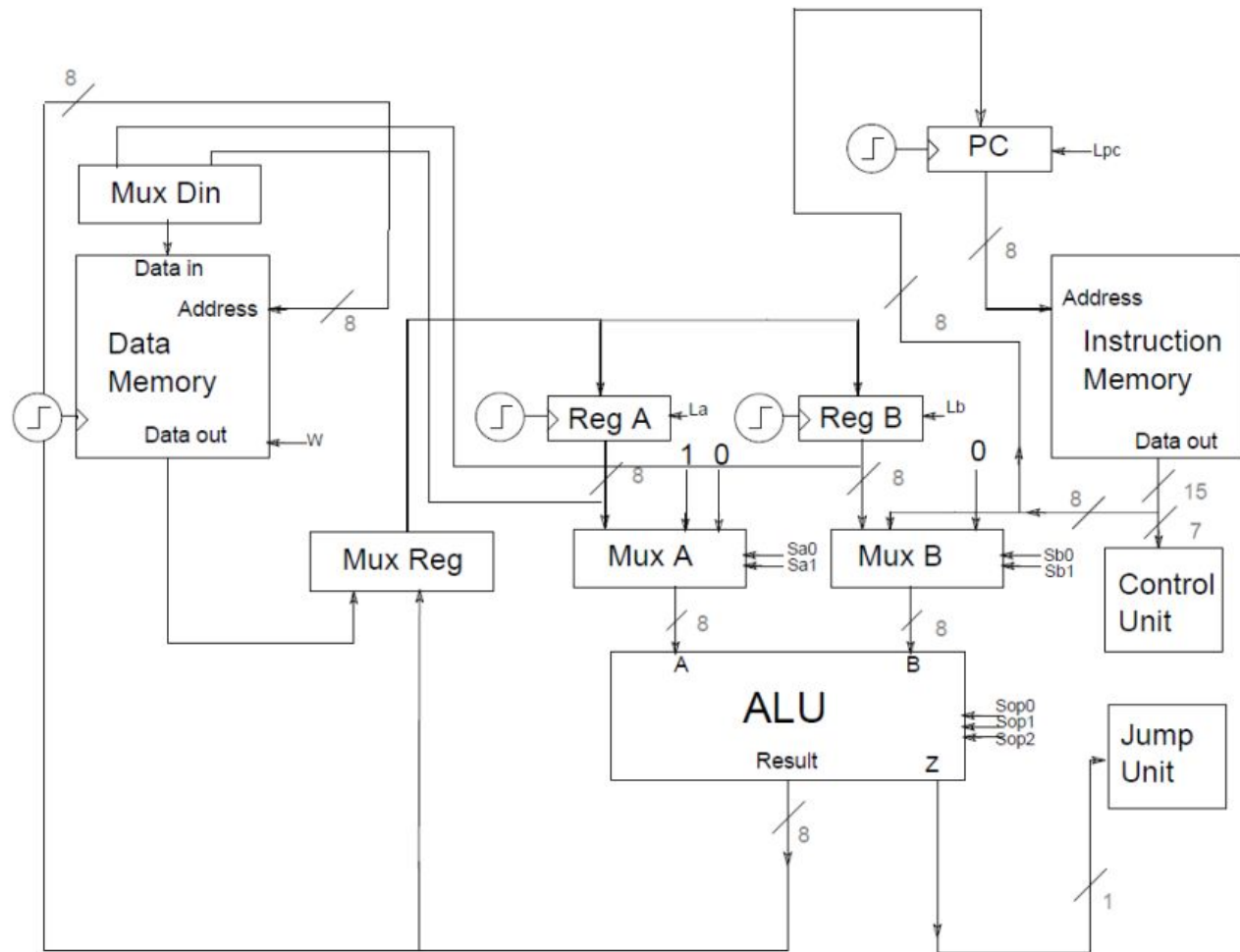
Nuevo Computador

- No hay conexión entre la salida de memoria y el MUX B
- La entrada de datos de la memoria solo viene de A y B
- La dirección de la memoria ahora viene de la ALU
- **Unidad de salto separada de la unidad de control**

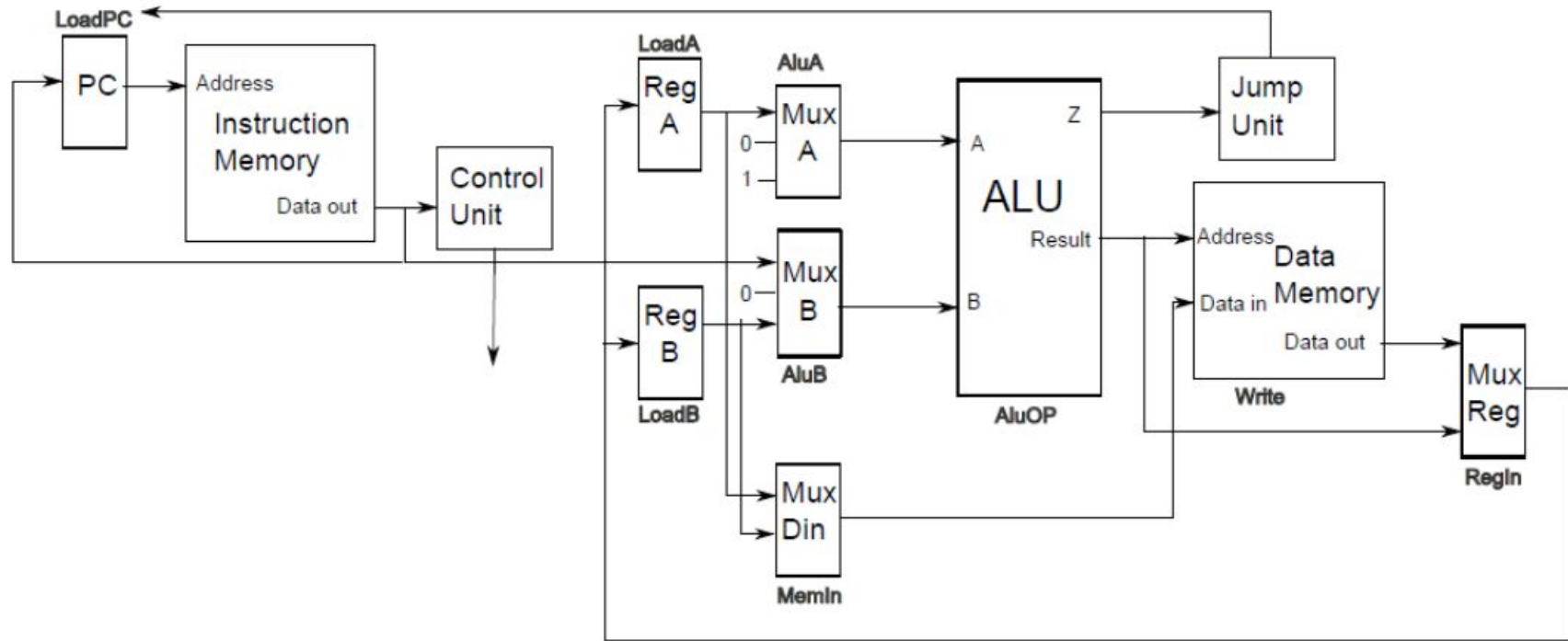


¿Dudas?

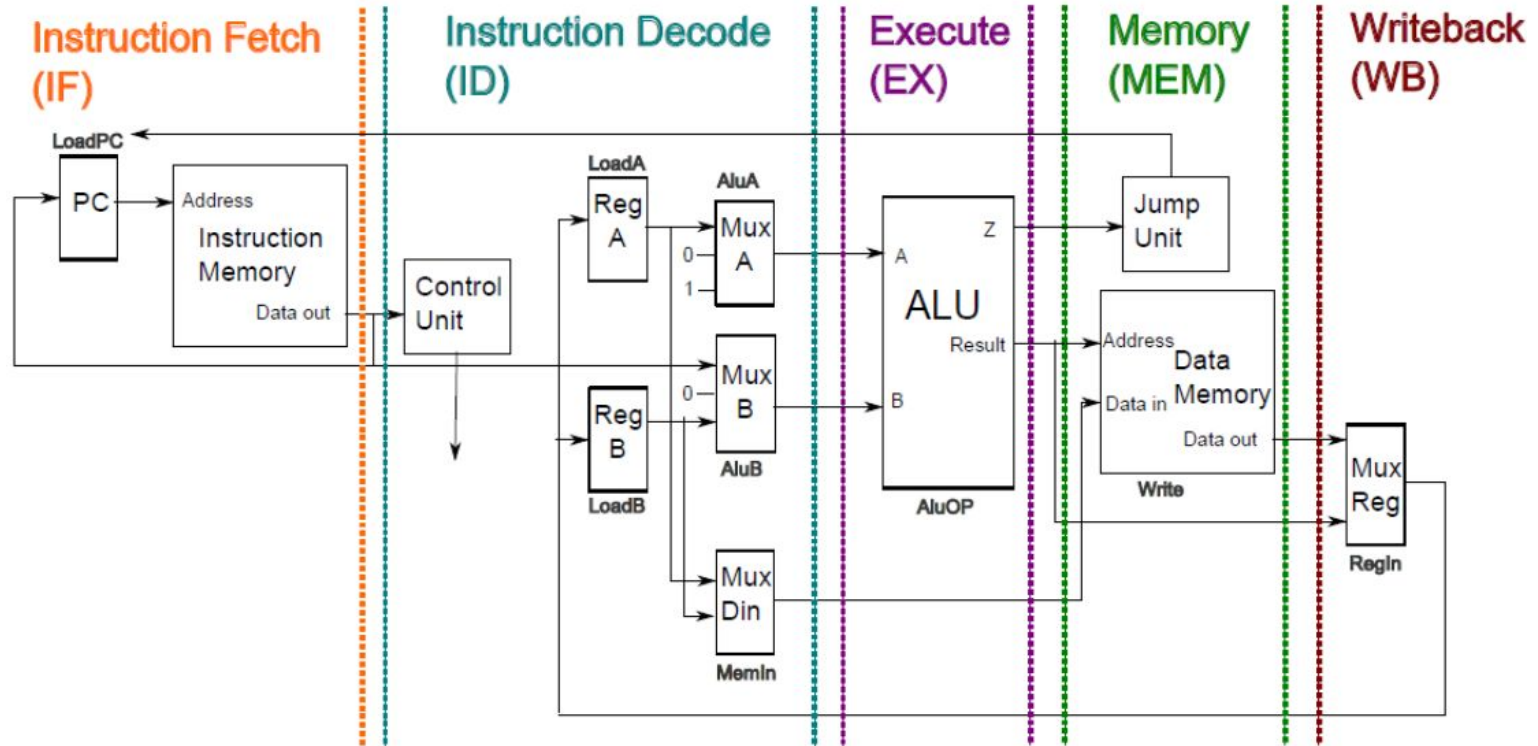
Nuevo Computador



Nuevo Computador Re-organizado



Nuevo Computador Re-organizado

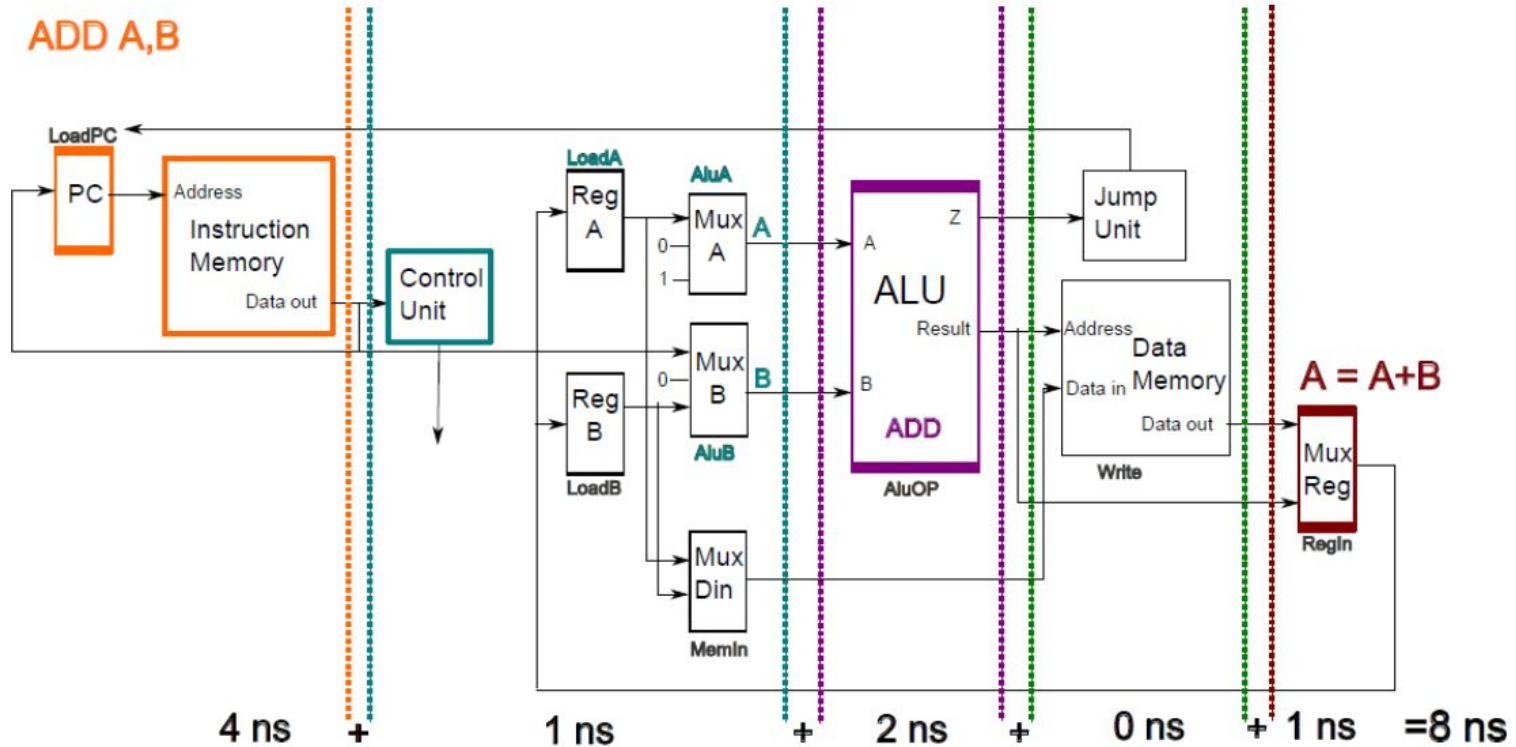


Pipeline

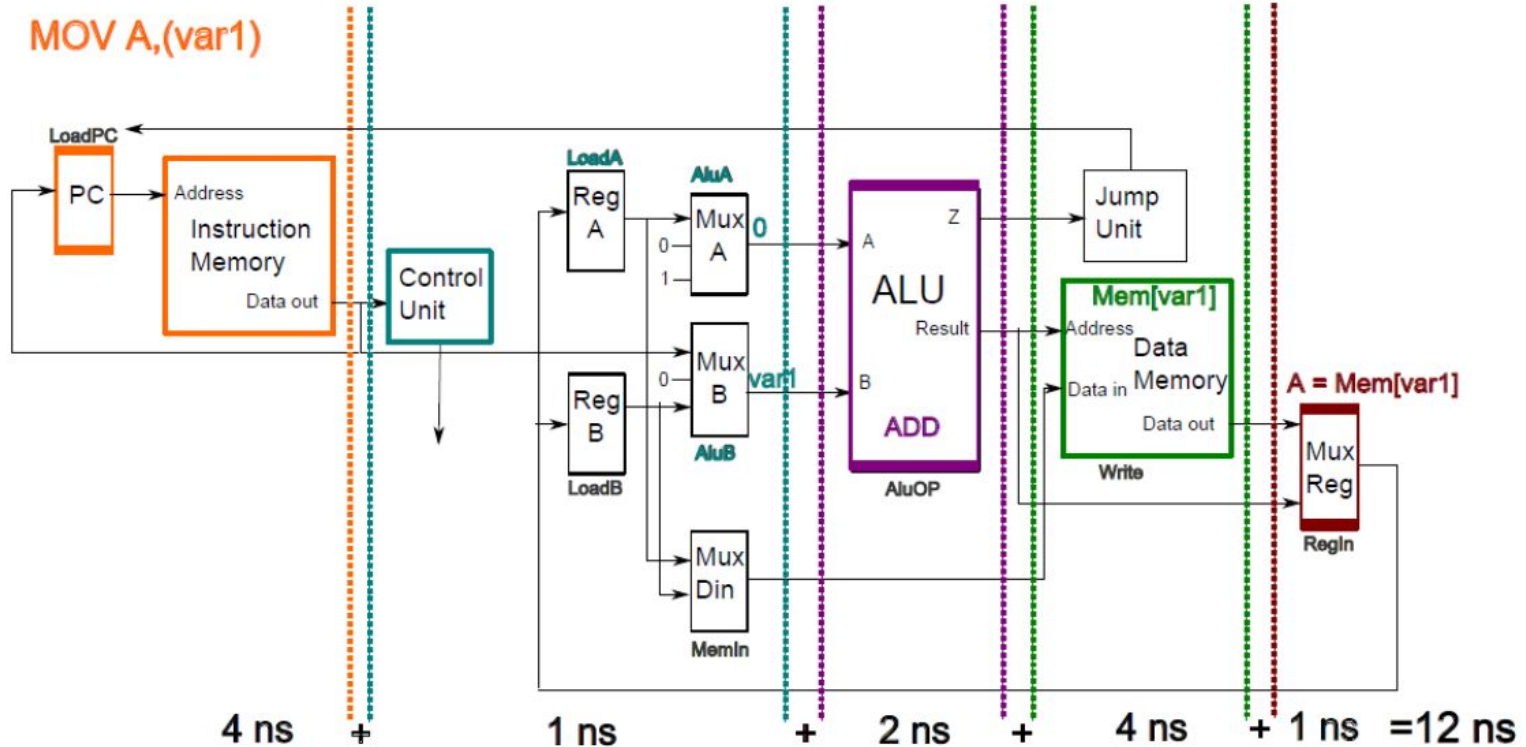
- Los procesos secuenciales con etapas independientes son conocidos como **pipeline**
- El diagrama con la nueva disposición del computador, nos permite ver el ciclo de una instrucción como un pipeline
- Esto permitiría procesar eventualmente múltiples instrucciones



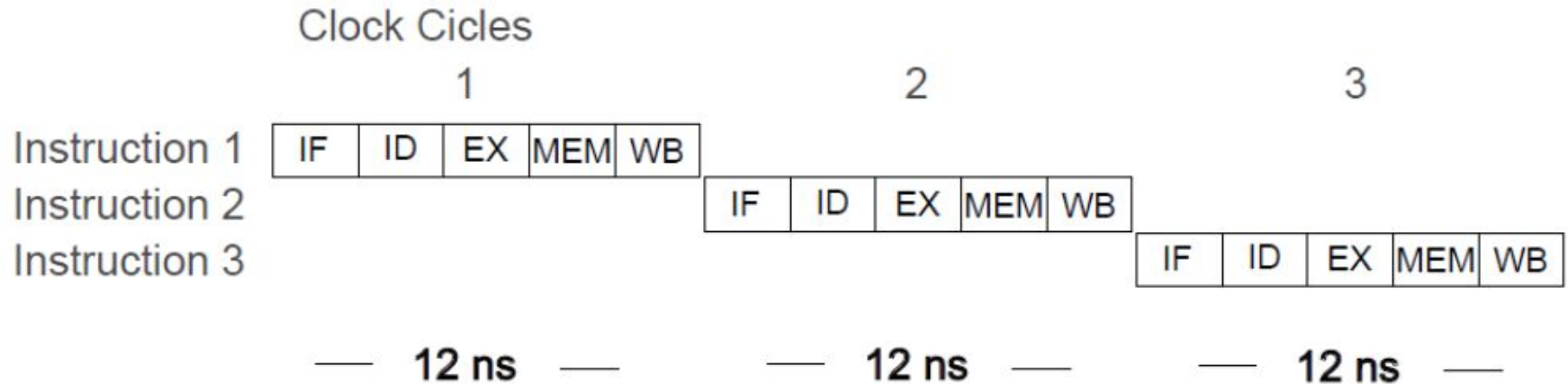
Nuevo Computador Re-organizado: Tiempo ADD A,B



Nuevo Computador Re-organizado: Tiempo MOV A,(var)



Nuevo Computador Re-organizado: Tiempo General

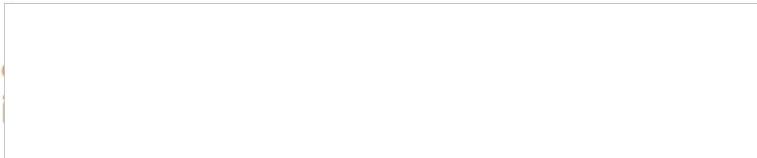


Total: 36 ns

Nuevo Computador Re-organizado: Tiempo Mejorado

	Cicles									
	1	2	3	4	5	6	7	8	9	10
Instruction 1	IF	ID	EX	MEM	WB					
Instruction 2		IF	ID	EX	MEM	WB				
Instruction 3			IF	ID	EX	MEM	WB			

4ns 4ns 4ns 4ns 4ns 4ns 4ns



Total: 28 ns

¿Dudas?

Clase 22 - Paralelismo

Parte 1

Profesor: **IIC2343 - Arquitectura de Computadores**
- Felipe Valenzuela González
Correo:
frvalenzuela@alumni.uc.cl