

*Prácticas de Arquitecturas Avanzadas de Procesadores*

---

# Práctica 2

Curso 2021 / 2022

---



---

# Segmentación

---

- ❖ Segmentación: Técnica implementar el paralelismo a nivel de instrucción en un único procesador.
- ❖ Objetivo: a tener ocupadas con instrucciones todas las partes del procesador  
Para ello: Dividir las instrucciones en una serie de pasos secuenciales que efectuarán distintas unidades
- ❖ No disminuye el tiempo de ejecución pero sí la productividad



# Segmentación

Clock cycle  
0    1    2    3    4    5    6    7    8

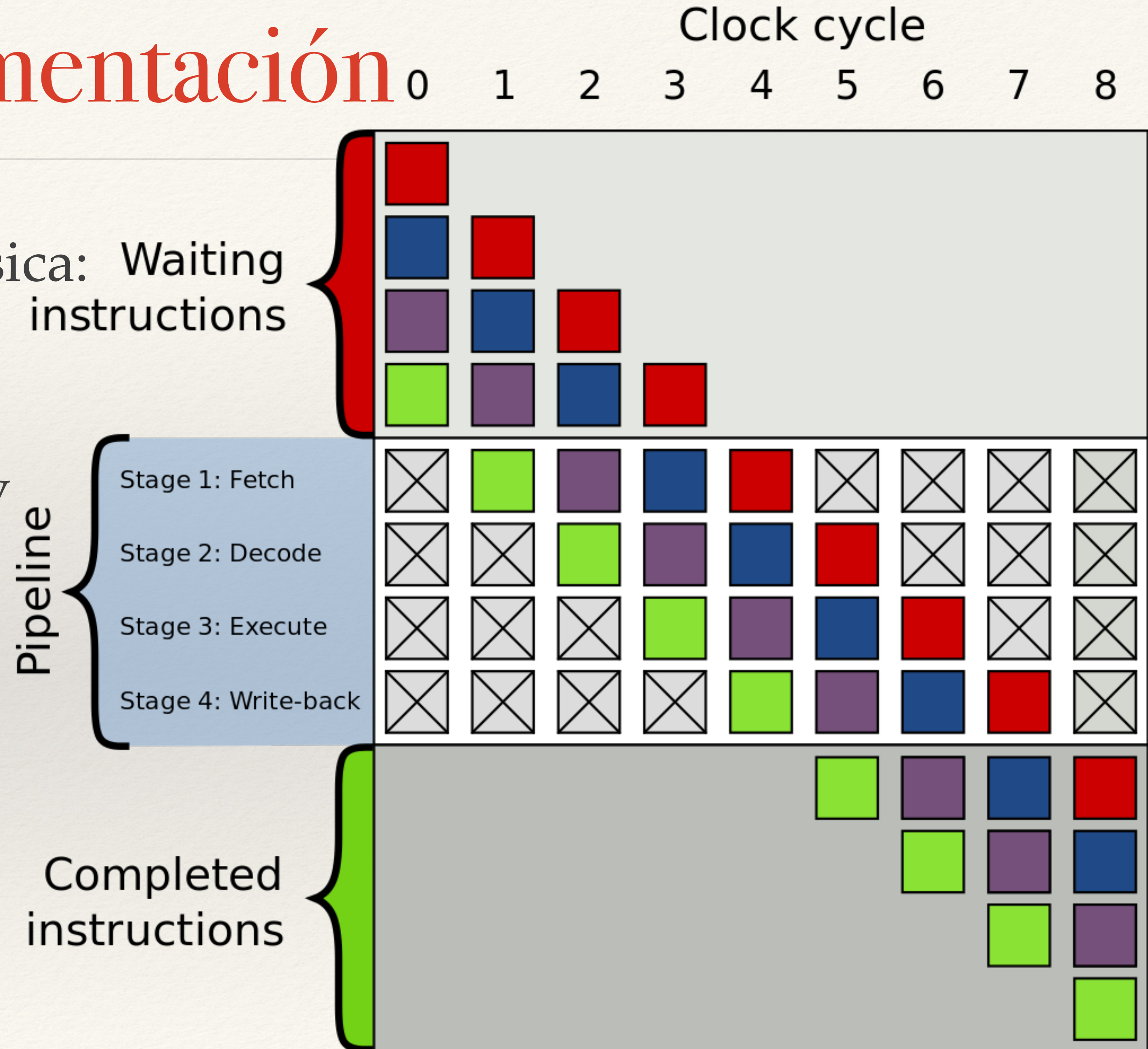
- ❖ Pipeline: Segmentación RISC Clásica: Waiting instructions

  - ❖ Lectura de instrucción
  - ❖ Decodificación de instrucción .y
  - ❖ Ejecución
  - ❖ lectura de registro
  - ❖ Acceso a memoria
  - ❖ Escritura en el registro

Pipeline

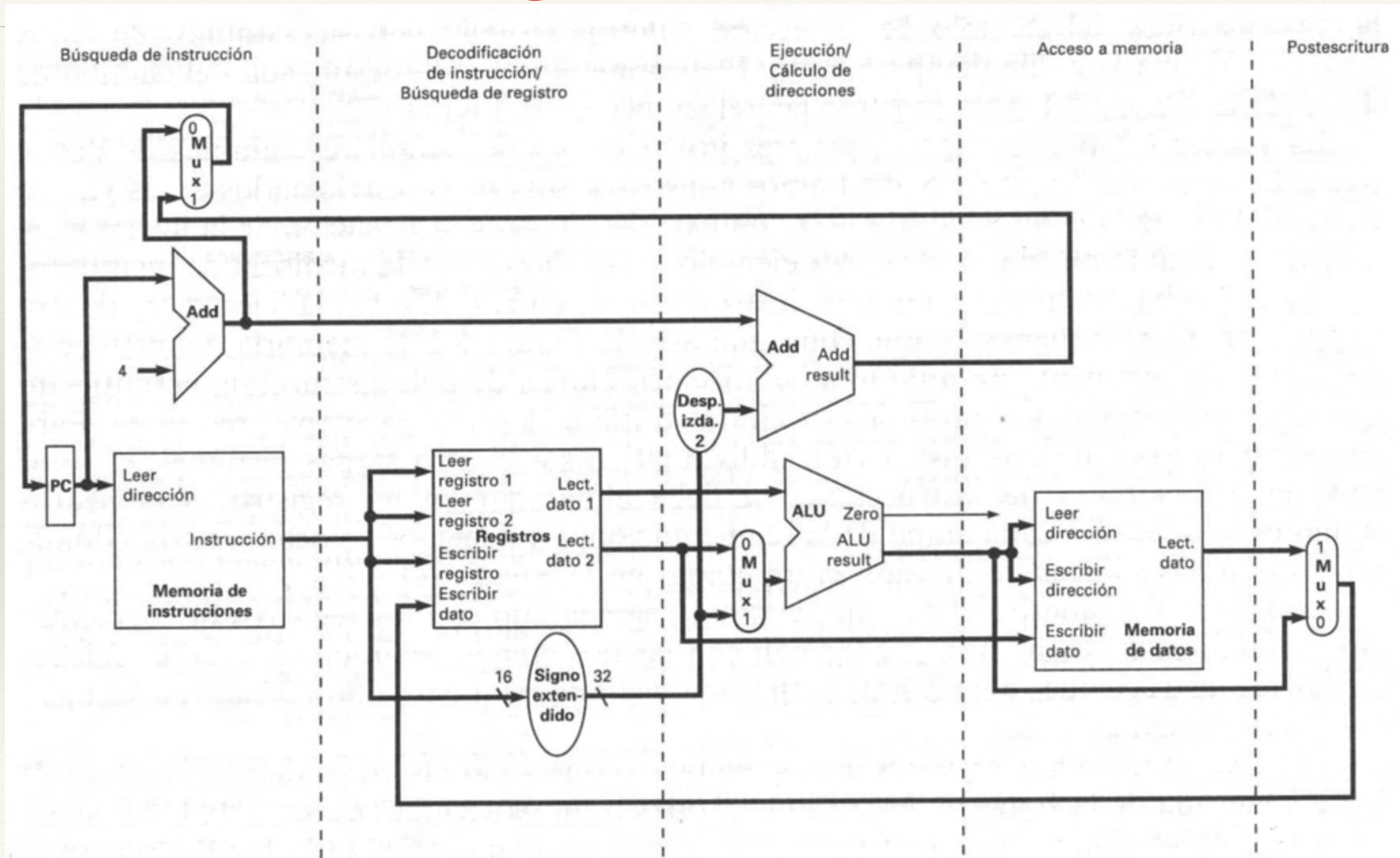
  - Stage 1: Fetch
  - Stage 2: Decode
  - Stage 3: Execute
  - Stage 4: Writeback

Completed instructions





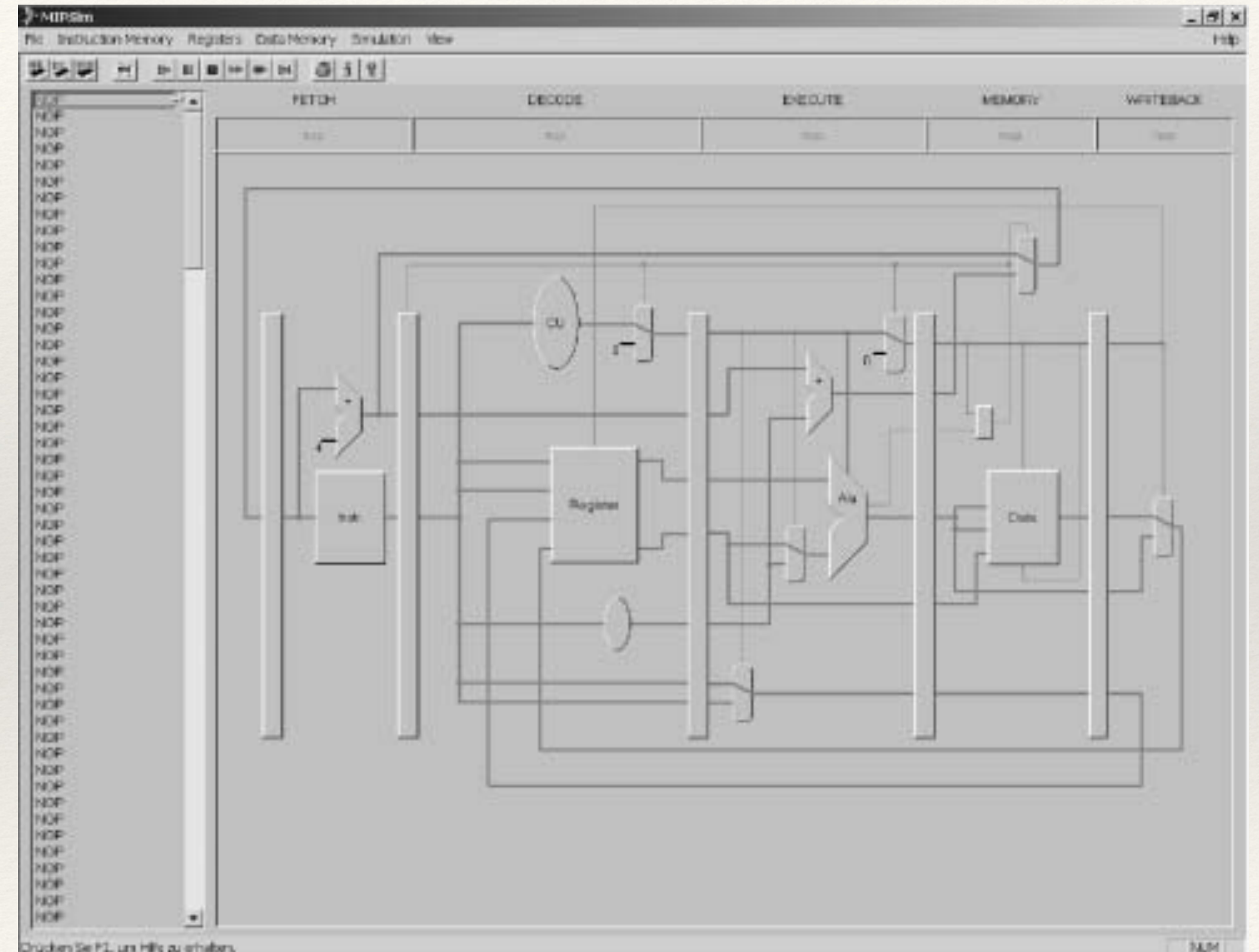
# Segmentación





# Herramienta MIPSIM

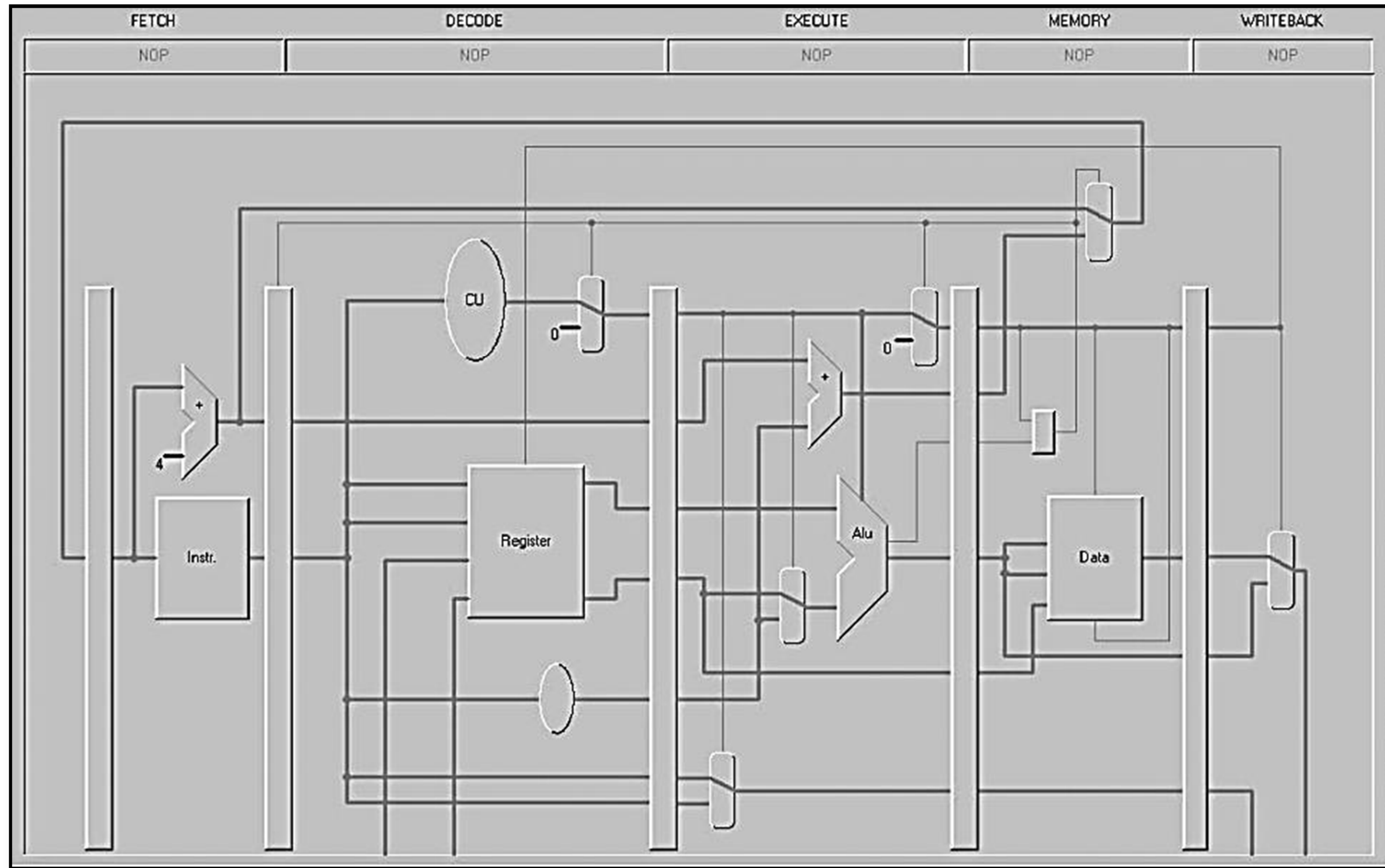
- ❖ IPSIM: Simulador de MIPS  
SEGMENTADO
- ❖ [https://www.uco.es/dptos/iec/arquitectura/?SOFTWARE\\_DOCENTE](https://www.uco.es/dptos/iec/arquitectura/?SOFTWARE_DOCENTE)
- ❖





# Estructura y camino de datos

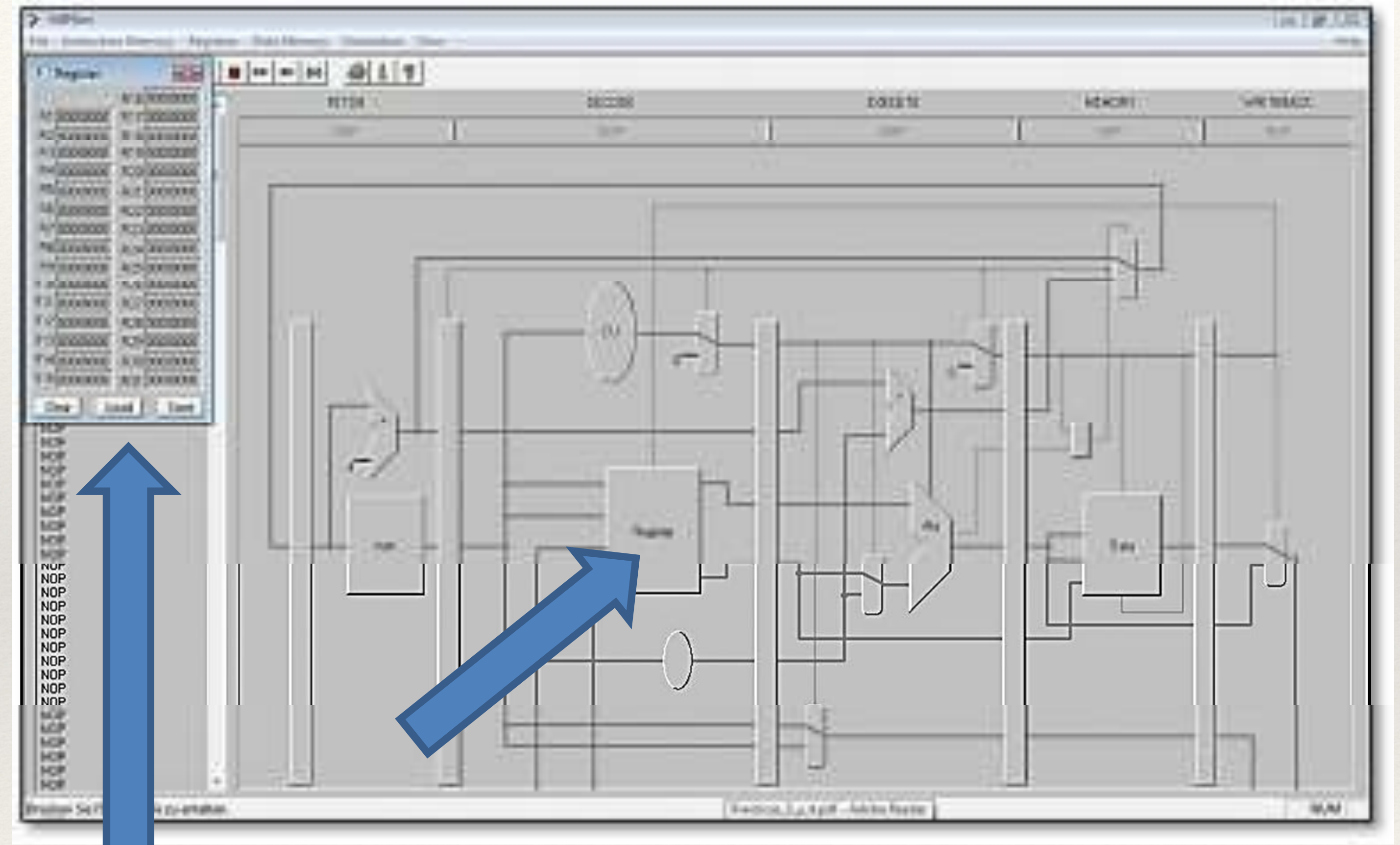
- ❖ Estructura de la maquina en la que realizamos la simulación.
- ❖ Separación de cada una de las etapas mediante una serie de registros que almacenan y separan los datos entre las etapas.
  - ❖ A estos registros se les denomina, registros de segmentación.
  - ❖ Para ver el contenido de los registros, de las instrucciones o de los datos pinchar sobre la unidad
  - ❖ Se puede ver el contenido de los buses que unen las distintas unidades.





# Banco de registros

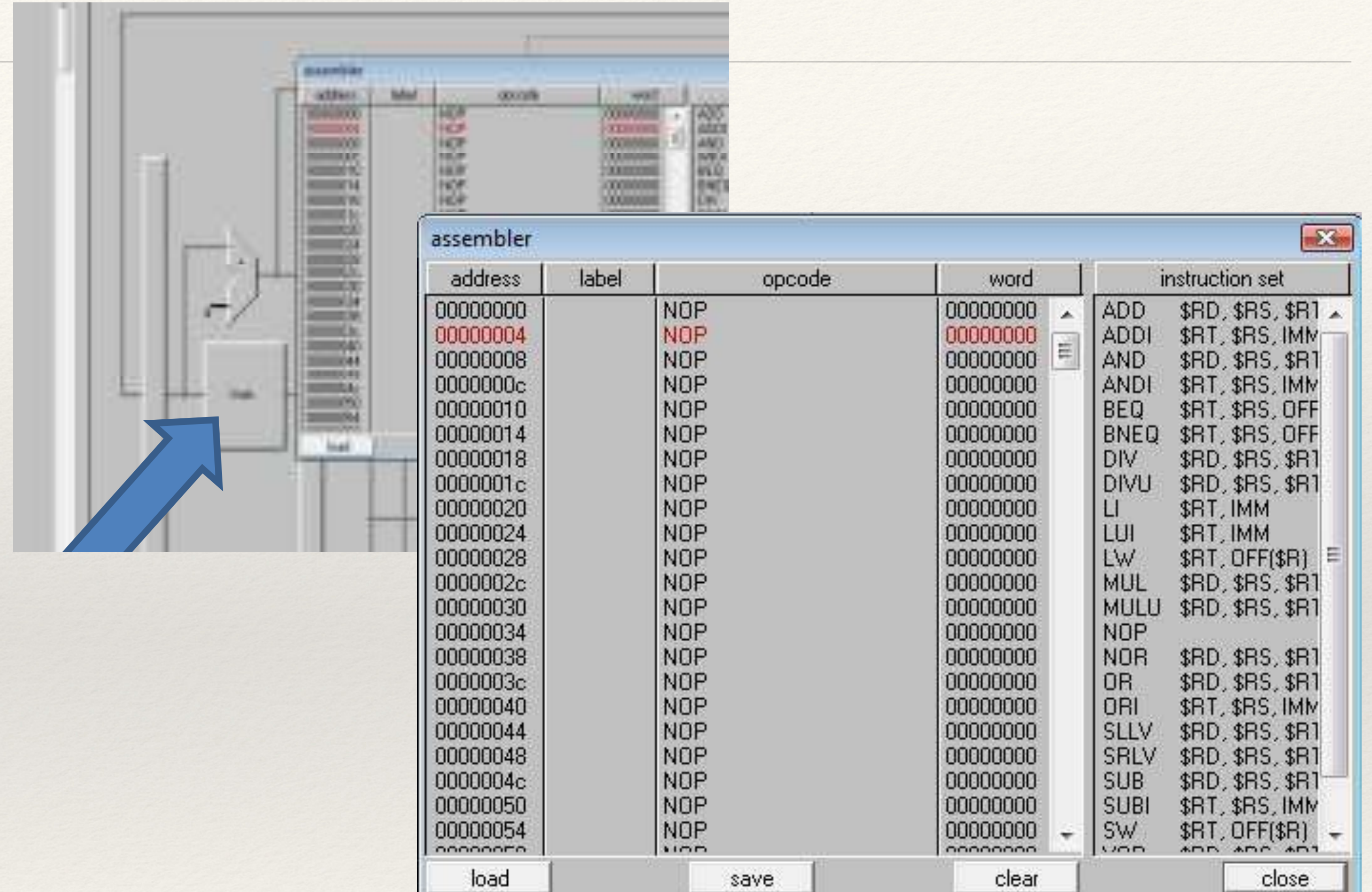
- ❖ Ver los valores de los registros pueden ser modificados por el usuario,
- ❖ Es posible cargar o salvar una configuración del banco de registros  
Fichero .mr





# Memoria de instrucciones

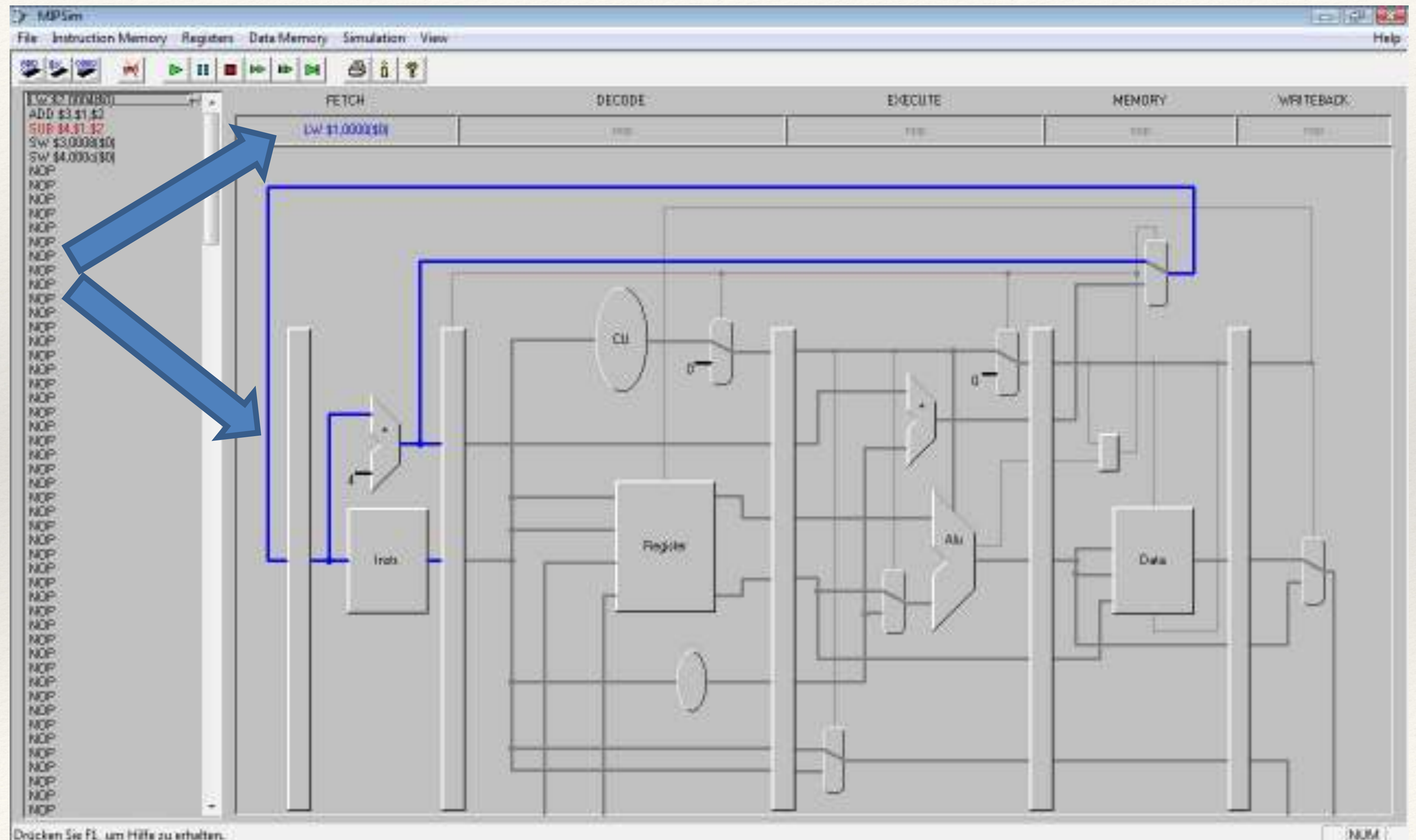
- ❖ So bye unidad “instr” se despliega el menú de memoria de instrucciones
- ❖ Sepuede cambiar el contenido de los campos anteriores,
- ❖ Cuando se introduce una nueva instrucción en lenguaje ensamblador, el simulador comprueba la sintaxis
- ❖ Cuatro columnas:
  - ❖ Address: Dirección de memoria de la instrucción.
  - ❖ Label: Etiqueta empleada en las instrucciones de salto.
  - ❖ pcode: La instrucción en lenguaje ensamblador.
  - ❖ Word: La instrucción en lenguaje máquina.
- ❖ Parte inferior:
  - ❖ Load: Para abrir un fichero con un programa (\*.mp).
  - ❖ Save: Guardar el programa actual.
  - ❖ Clear: Pone a cero la memoria de instrucciones,





# Ejemplo

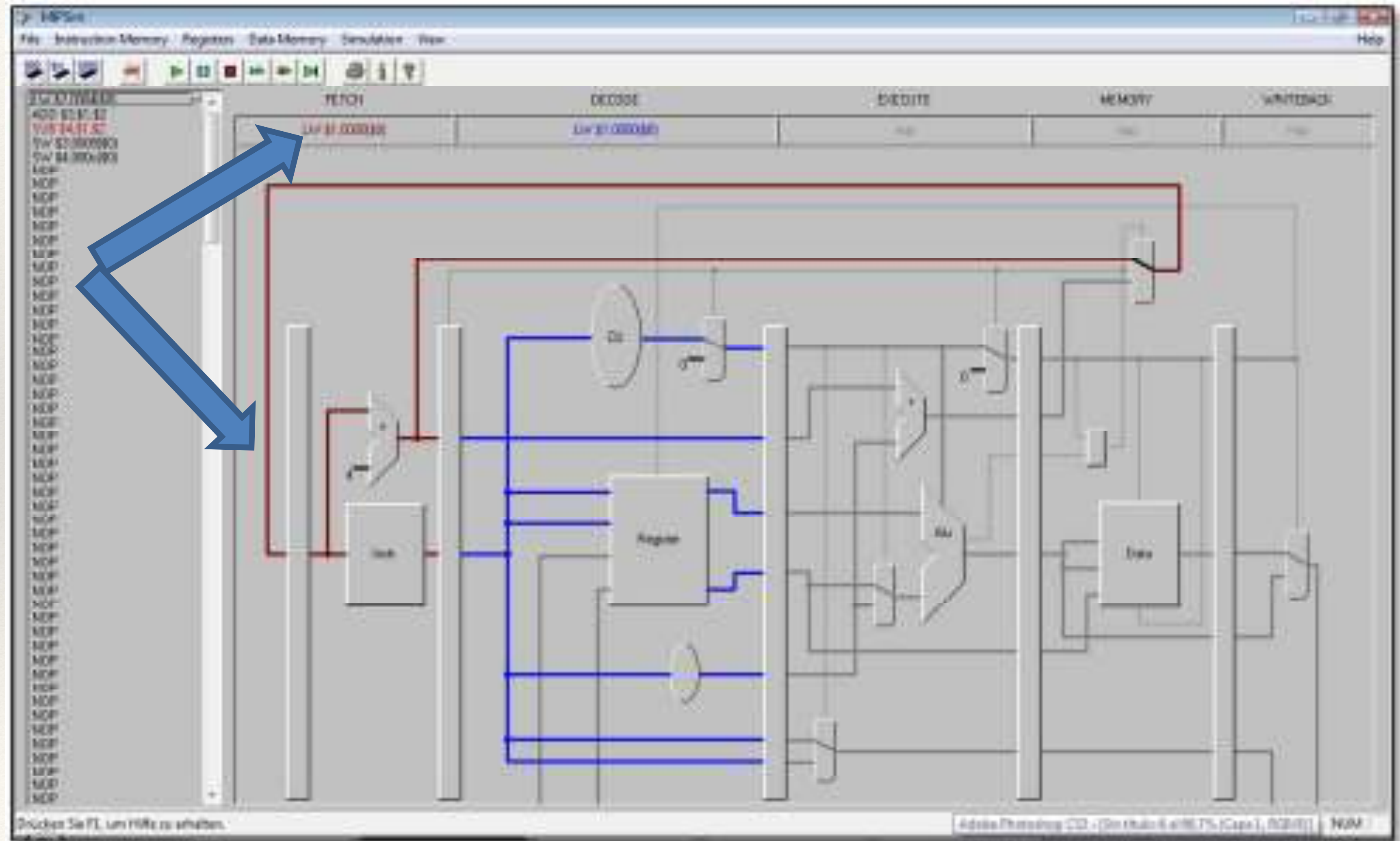
lw \$1,0(\$0)  
lw \$2,4(\$0)  
add \$3,\$1,\$2  
sub \$4,\$1,\$2  
sw \$3,8(\$0)  
sw \$4,c(\$0)





# Ejemplo

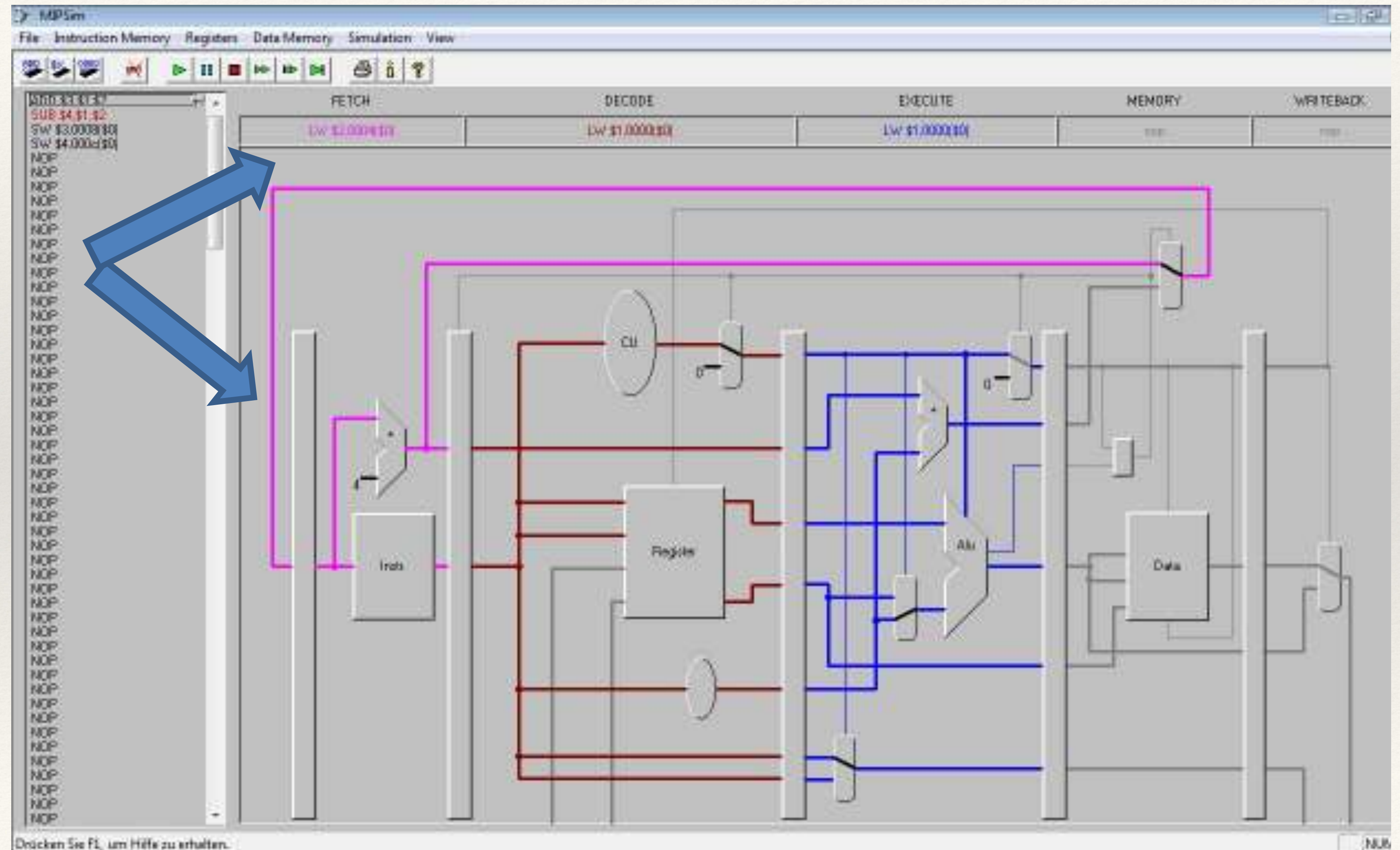
lw \$1,0(\$0)  
lw \$2,4(\$0)  
add \$3,\$1,\$2  
sub \$4,\$1,\$2  
sw \$3,8(\$0)  
sw \$4,c(\$0)





# Ejemplo

lw \$1,0(\$0)  
lw \$2,4(\$0)  
add \$3,\$1,\$2  
sub \$4,\$1,\$2  
sw \$3,8(\$0)  
sw \$4,c(\$0)





# Ejemplo

```
lw $1,0($0)
lw $2,4($0)
add $3,$1,$2
sub $4,$1,$2
sw $3,8($0)
sw $4,c($0)
```

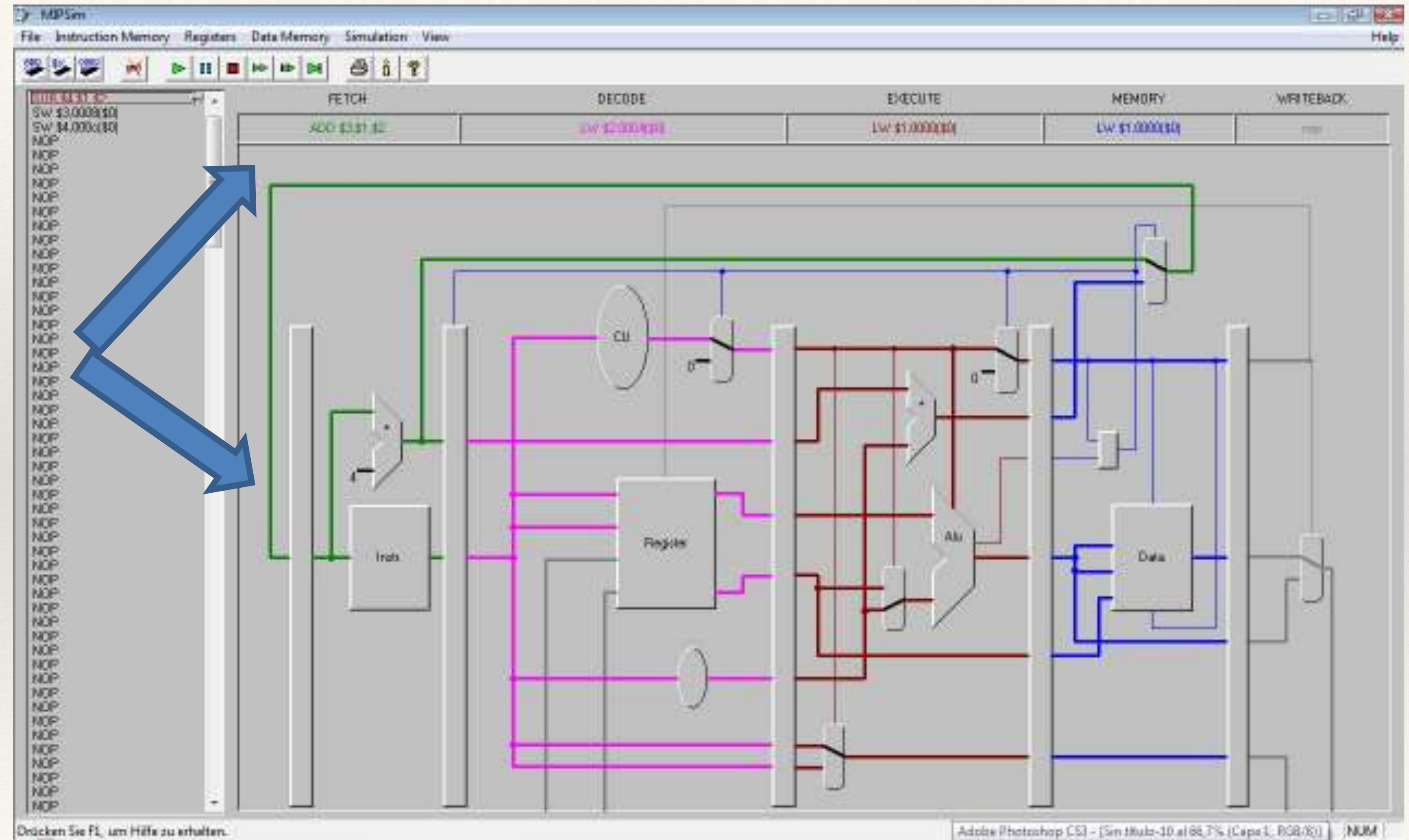


Figura 21 Etapa MEMORY



---

# Tareas a realizar

---

- ❖ Instalar la herramienta
- ❖ Ejecutar los ejemplos
- ❖ Realizar los ejercicios que se indican
- ❖ 2 sesiones ( el 7/12 no hay prácticas)
- ❖ Fecha de entrega: 14/12