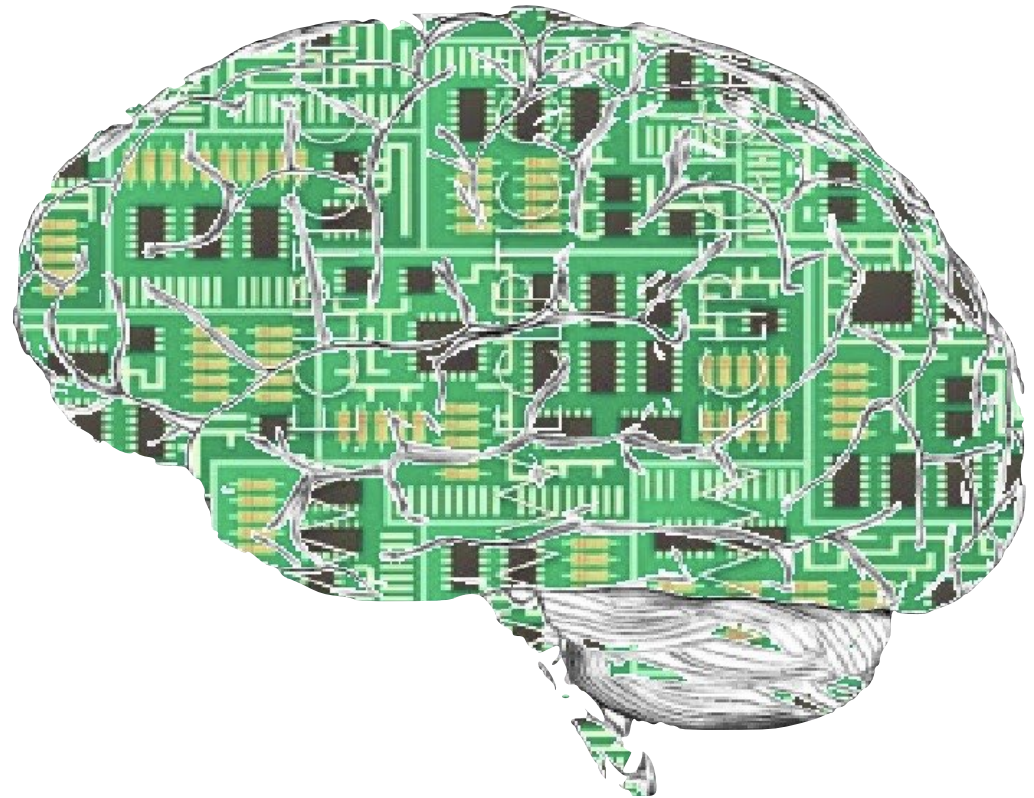


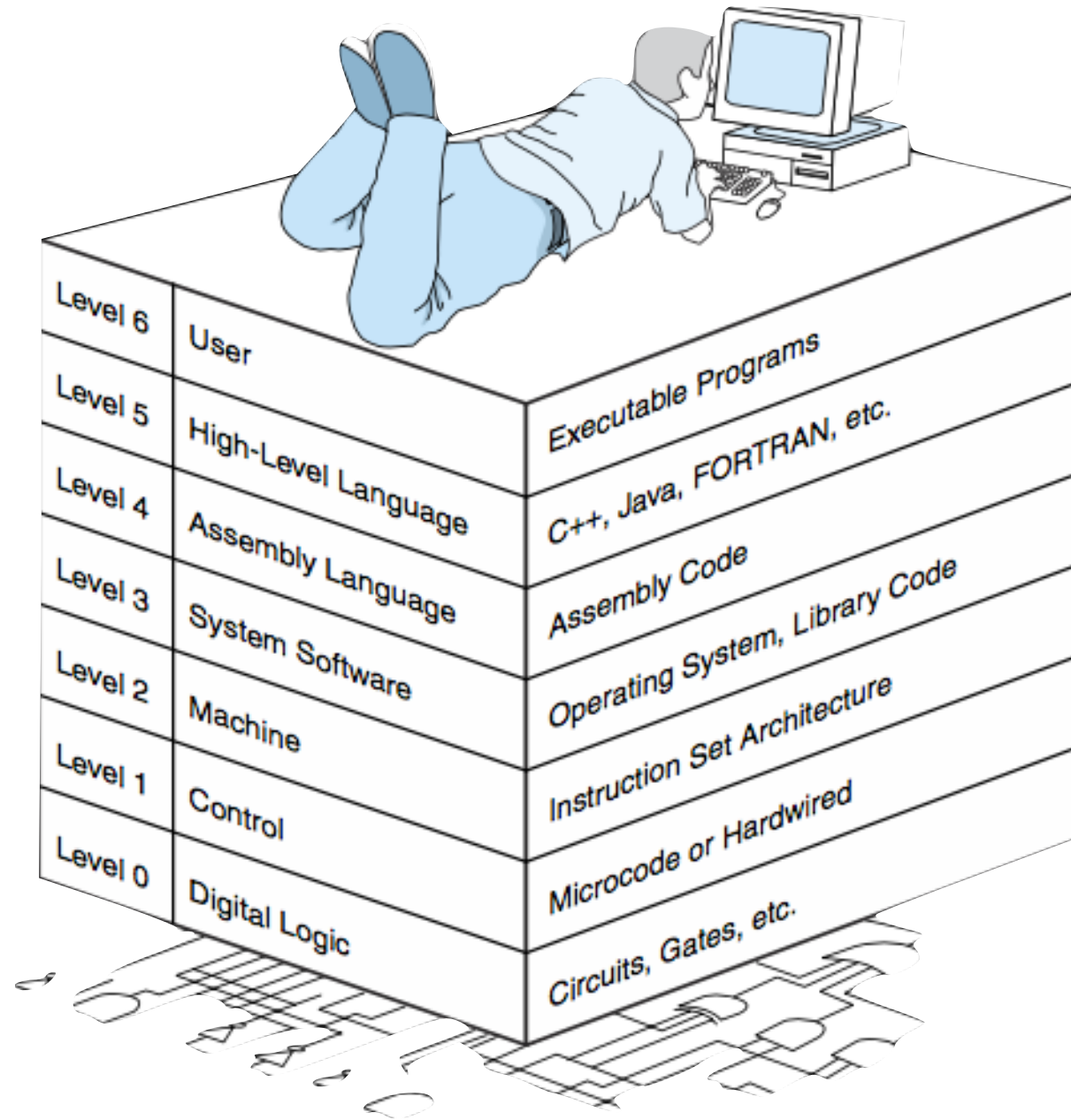
DEPARTAMENTO DE
COMPUTACIÓN
UNRC
2016

ORGANIZACIÓN DEL PROCESADOR



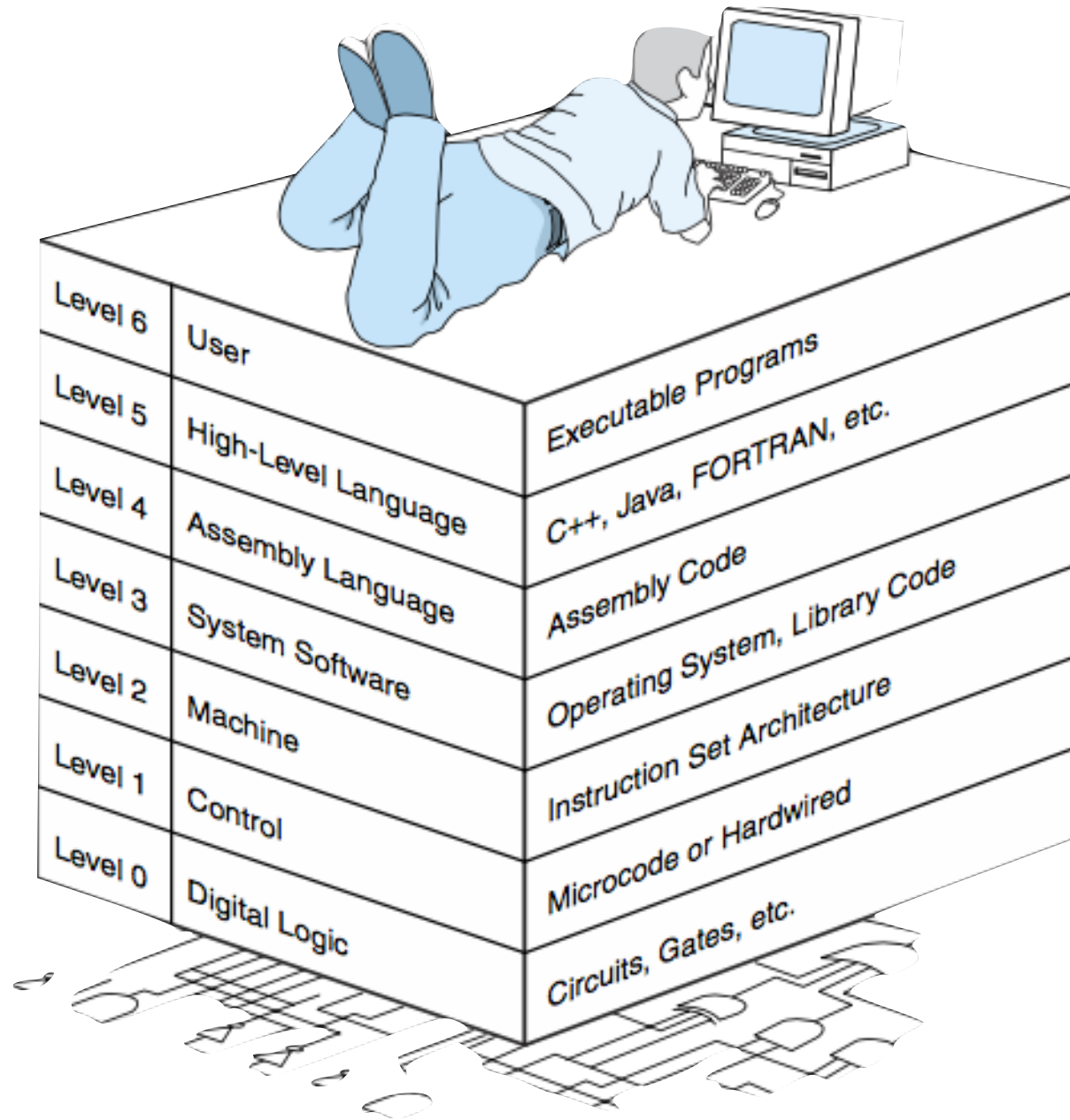
ORGANIZACIÓN DEL PROCESADOR

NIVELES ABSTRACTOS DE COMPUTADORAS MODERNAS



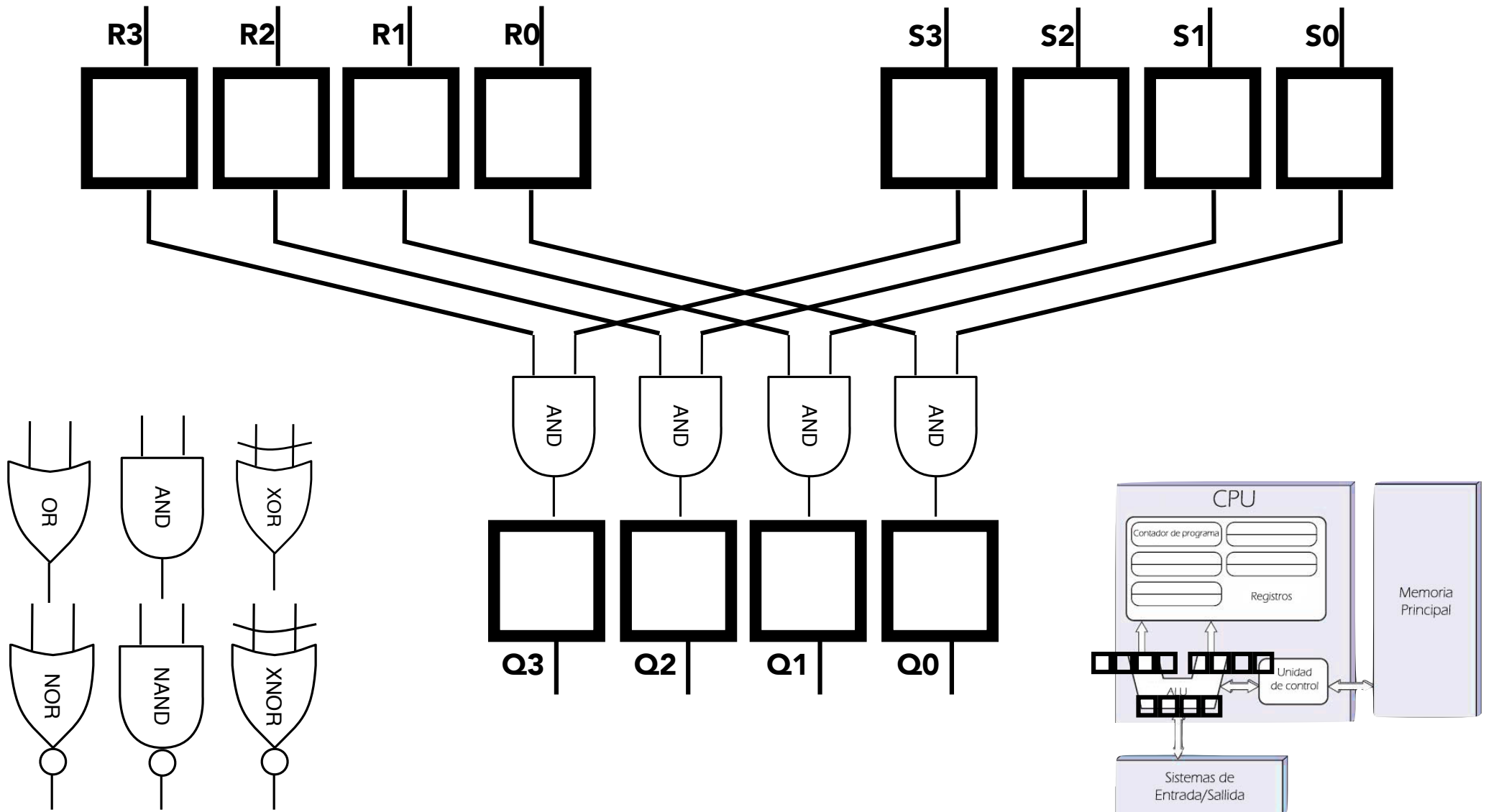
ORGANIZACIÓN DEL PROCESADOR

NIVELES ABSTRACTOS DE COMPUTADORAS MODERNAS



ORGANIZACIÓN DEL PROCESADOR

ALU Y REGISTROS



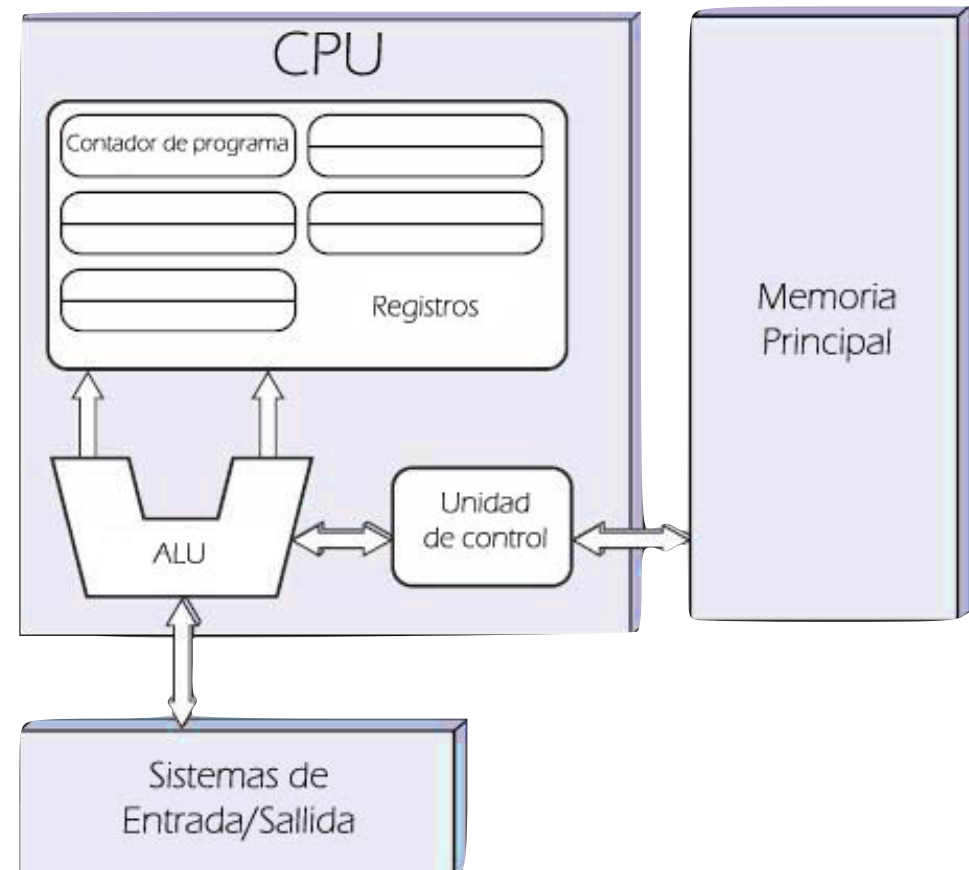
ORGANIZACIÓN DEL PROCESADOR

ASSEMBLER

Fetch: La Unidad de control obtiene de la memoria la próxima instrucción que indica el *contador de programa*

Decode: La Unidad de control obtiene de la memoria (si fuere necesario) y decodifica la instrucción para poder ser ejecutada por la ALU

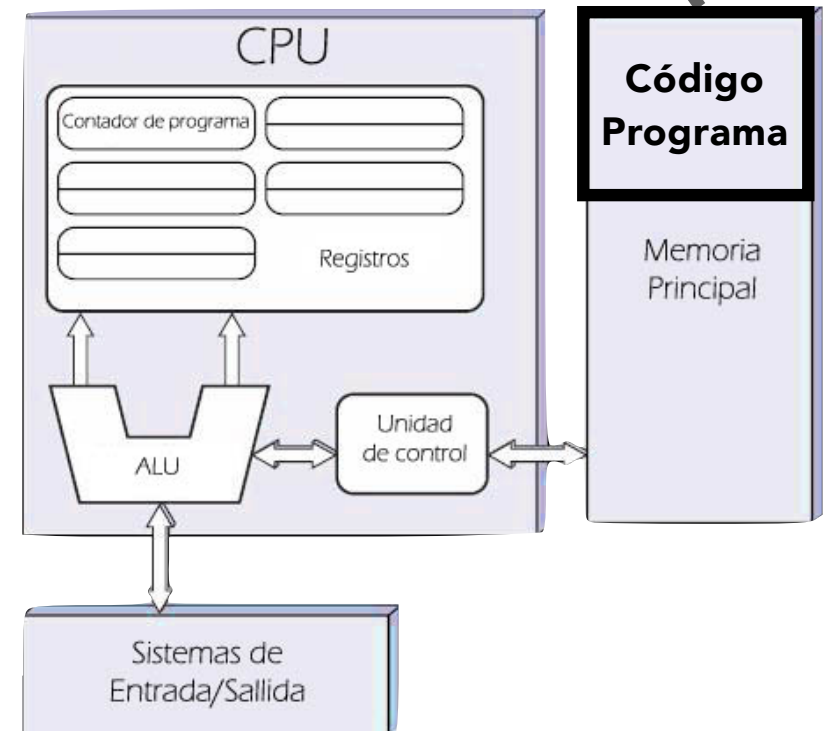
Execute: La ALU ejecuta (calcula) el resultado de la operación y lo almacena en un registro o memoria



ORGANIZACIÓN DEL PROCESADOR

QUE HAY DEBAJO DE ASSEMBLER

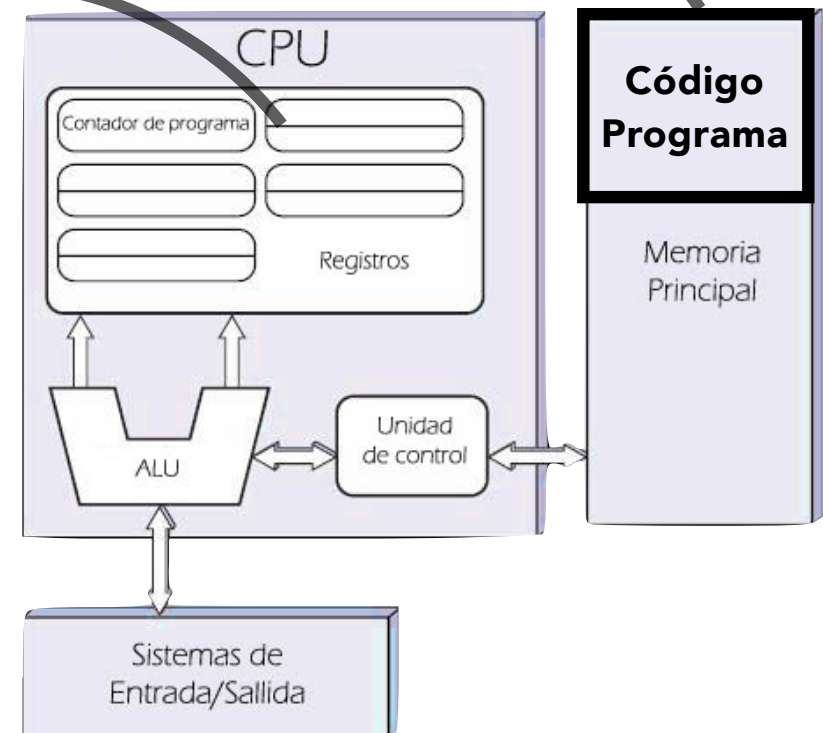
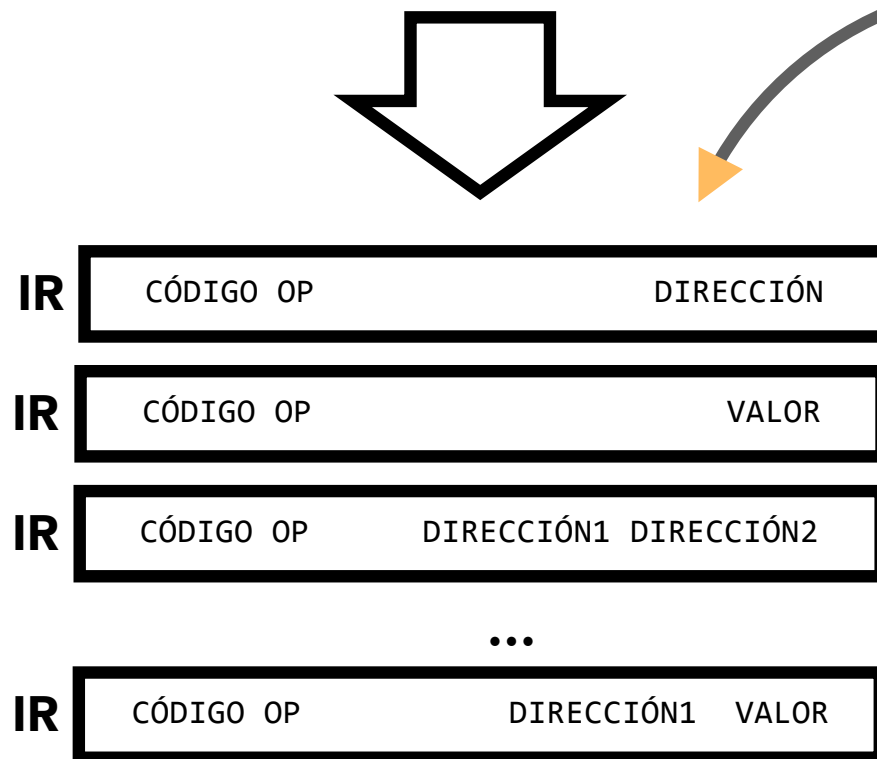
MOV **eax, [L1]**
ADD **ax, 10101b**
INC **ebx**



ORGANIZACIÓN DEL PROCESADOR

QUE HAY DEBAJO DE ASSEMBLER

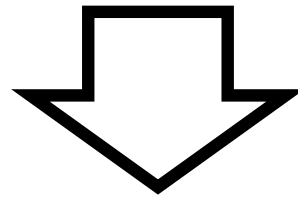
MOV **eax, [L1]**
ADD **ax, 10101b**
INC **ebx**



ORGANIZACIÓN DEL PROCESADOR

CONTROL POR HARDWARE VS. MICRO-PROGRAMADO

ADD eax, [L1]



- 1) Recuperar la instrucción de la memoria (al IR)
- 2) Decodificar cuál es la instrucción
- 3) Calcular los operandos (memoria efectiva)
- 4) Recuperar los operandos (en registros)
- 5) Ejecutar la instrucción
- 6) Guardar el resultado

ORGANIZACIÓN DEL PROCESADOR

MICROINSTRUCCIONES

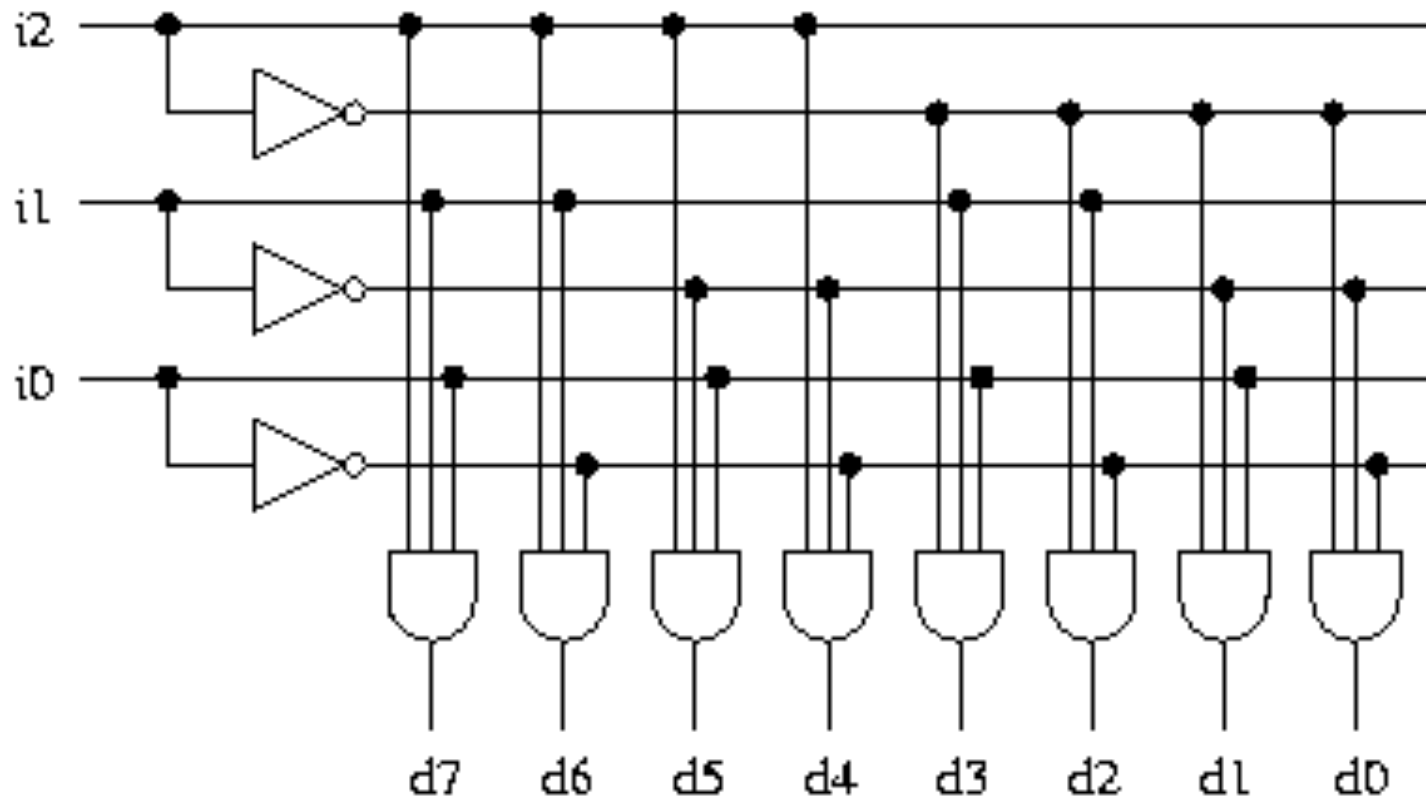
ADD eax, [L1]



- 1) Recuperar la instrucción de la memoria (al IR)**
 - Configurar la dirección (MAR) con el PC
 - Activar la Lectura de Memoria al IR
 - Incrementar el PC
- 2) Decodificar cuál es la instrucción**
 - Decodificar ADD
- 3) Calcular los operandos (memoria efectiva)**
 - Calcular la dirección L1
- 4) Recuperar los operandos (en registros)**
 - Guardar en el RegistroA de la ALU el contenido de EAX
 - Configurar la dirección (MAR) con L1 (cálculado en 3)
 - Activar la Lectura de Memoria al RegistroB de la ALU
- 5) Ejecutar la instrucción**
 - Activar la ALU con la Operación correspondiente
- 6) Guardar el resultado**
 - Transferir al EAX el valor alojado en el registro de salida de la ALU

ORGANIZACIÓN DEL PROCESADOR

EJEMPLO DE DECODIFICADOR 3 ENTRADAS 8 SALIDAS



ORGANIZACIÓN DEL PROCESADOR

CISC VS RISC (CARACTERIZACIÓN DE FLYNN)

CISC (Complex Instruction Set Computing)

- Tienen un conjunto de instrucciones que se caracteriza por ser muy amplio y permitir operaciones complejas.
- Permiten operandos situados en la memoria o en los registros internos.
- Permite reducir el costo total del sistema.
- Mejora la compactación de código.
- Facilita la depuración de errores.
- Ejemplos: **Intel** 8086, 8088, 80286, 80386, 80486

RISC (Reduced Instruction Set Computing)

- Instrucciones de tamaño fijo y presentadas en un reducido número de formatos
- Sólo las instrucciones de carga y almacenamiento acceden a la memoria de datos.

Maximizar paralelismo en la ejecución de instrucciones y reducir los accesos a memoria

Mayor cantidad de Registros

La CPU trabaja más rápido al utilizar menos ciclos de reloj para ejecutar instrucciones

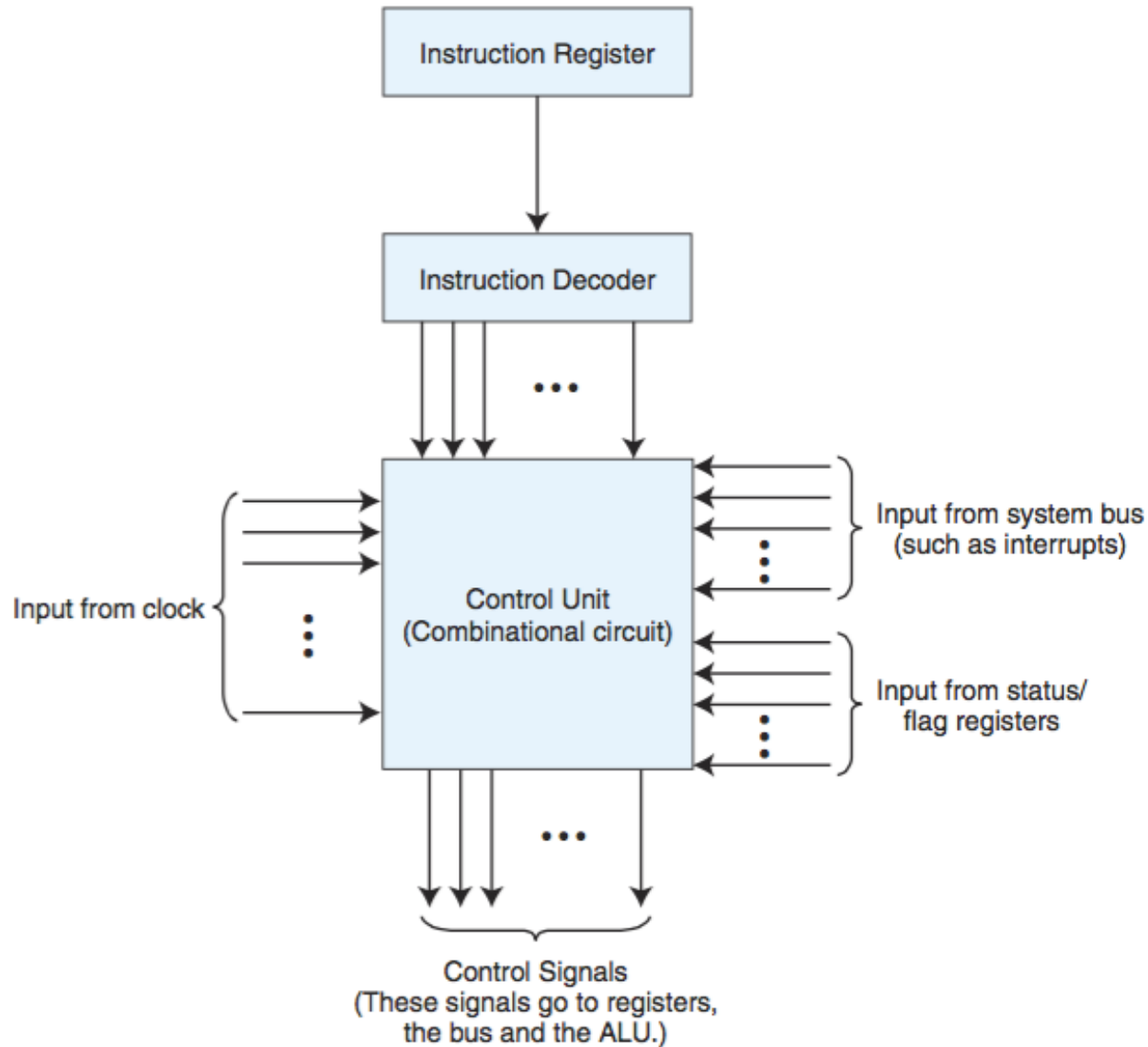
Cada instrucción puede ser ejecutada en un solo ciclo del CPU

Ejemplos: PowerPC, DEC Alpha, MIPS, ARM, SPARC

$$\text{Eficiencia} = \frac{\text{time}}{\text{program}} = \frac{\text{time}}{\text{cycle}} \times \frac{\text{cycles}}{\text{instruction}} \times \frac{\text{instructions}}{\text{program}}$$

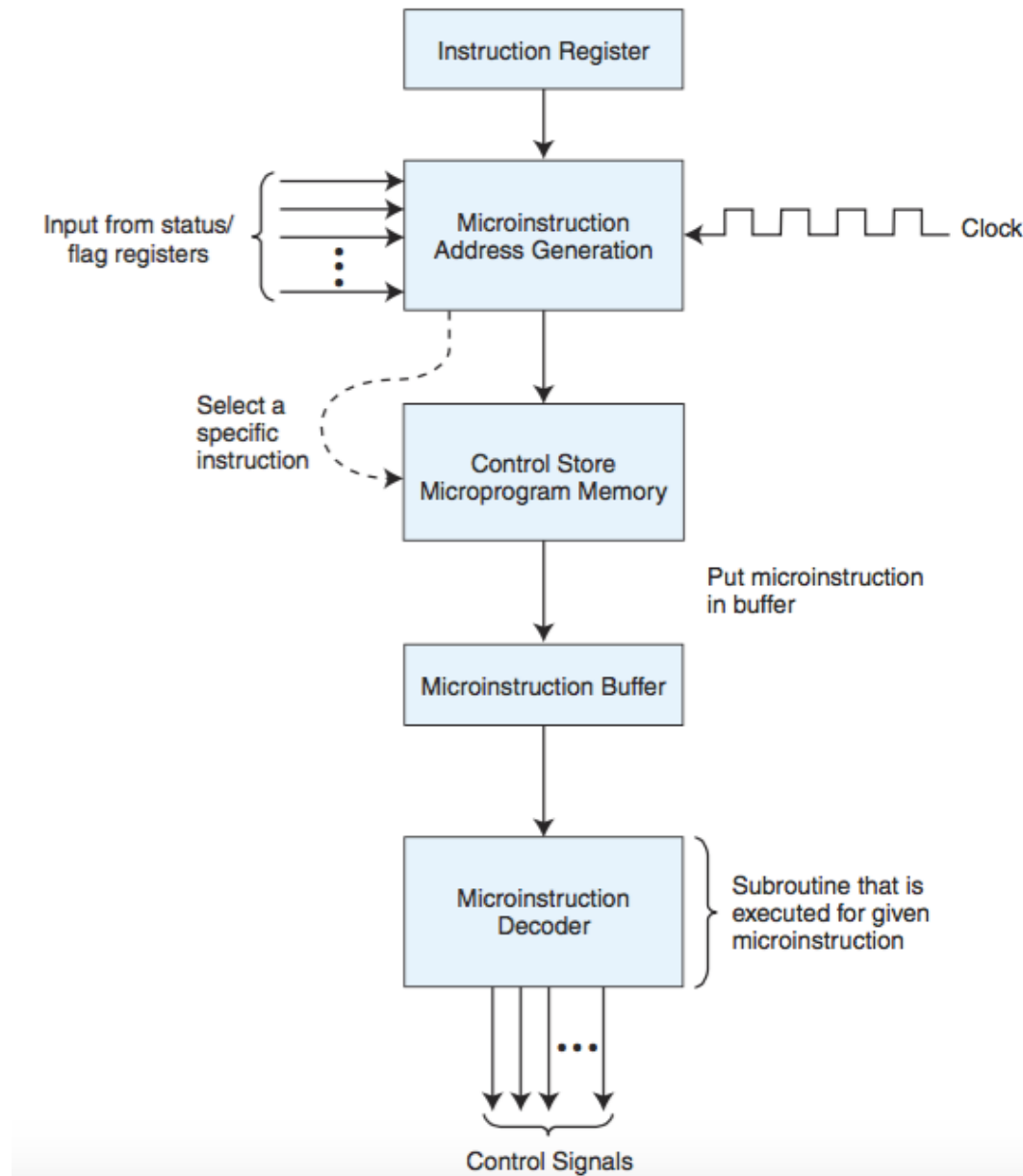
ORGANIZACIÓN DEL PROCESADOR

CONTROL POR HARDWARE



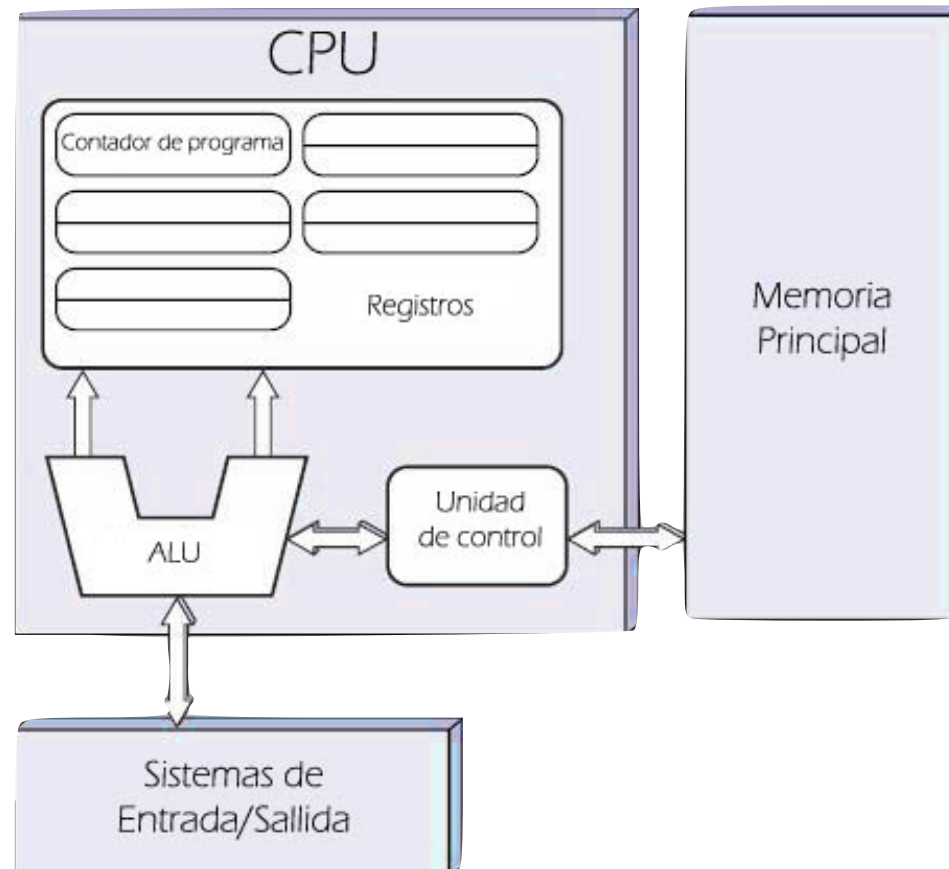
ORGANIZACIÓN DEL PROCESADOR

CONTROL MICROPROGRAMADO (WILKES)



ORGANIZACIÓN DEL PROCESADOR

PIPELINE



ORGANIZACIÓN DEL PROCESADOR

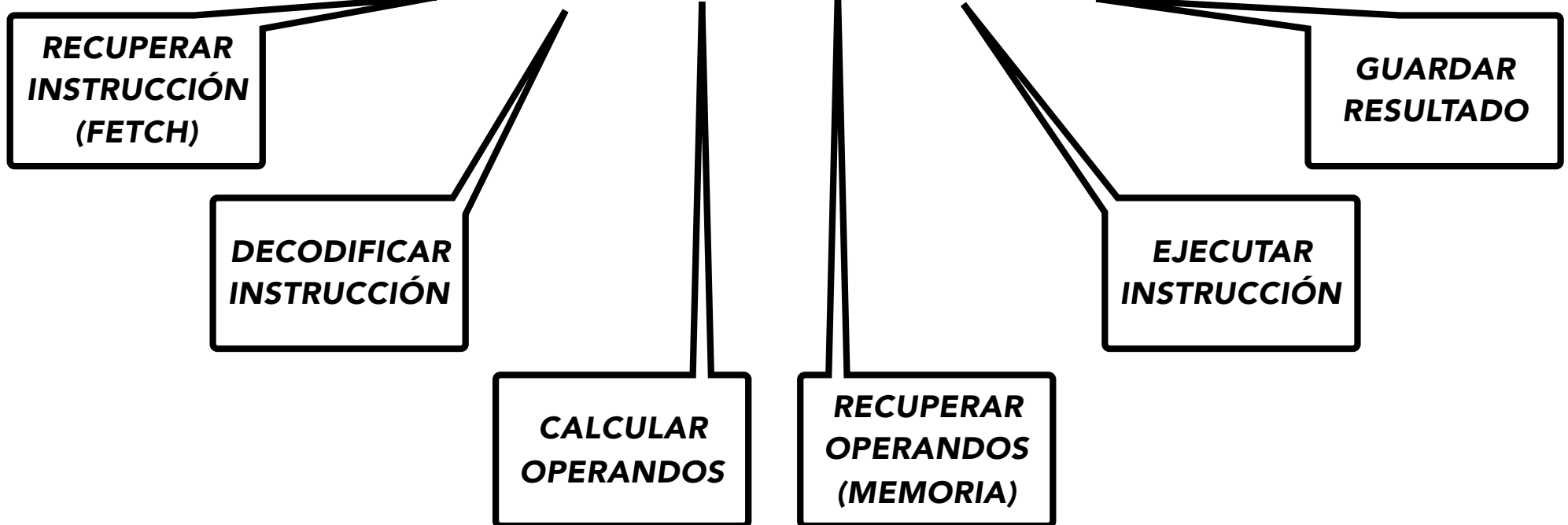
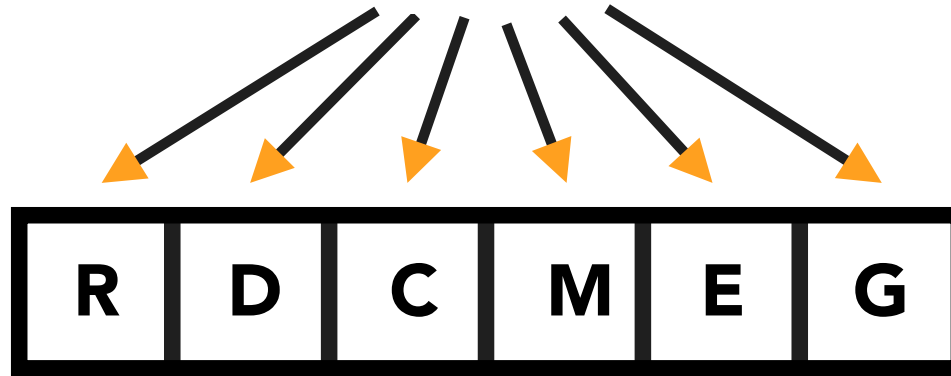
PIPELINE

ADD *eax*, [L1]

ORGANIZACIÓN DEL PROCESADOR

PIPELINE

ADD **eax, [L1]**



ORGANIZACIÓN DEL PROCESADOR

PIPELINE

...

ADD **eax, [L1]**

MOV **eax, [L1]**

INC **ebx**

...

ORGANIZACIÓN DEL PROCESADOR

PIPELINE

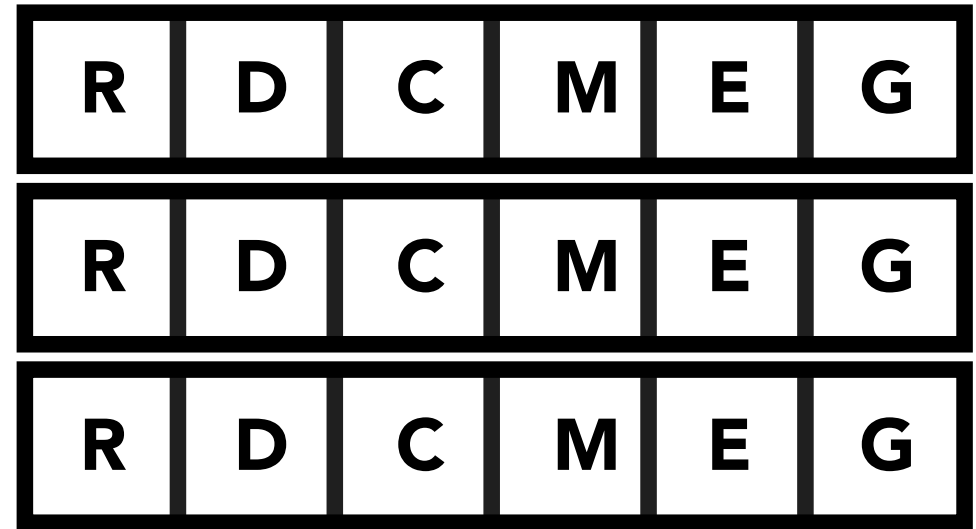
...

ADD **eax, [L1]**

MOV **eax, [L1]**

INC **ebx**

...



ORGANIZACIÓN DEL PROCESADOR

PIPELINE

...

ADD **eax, [L1]**

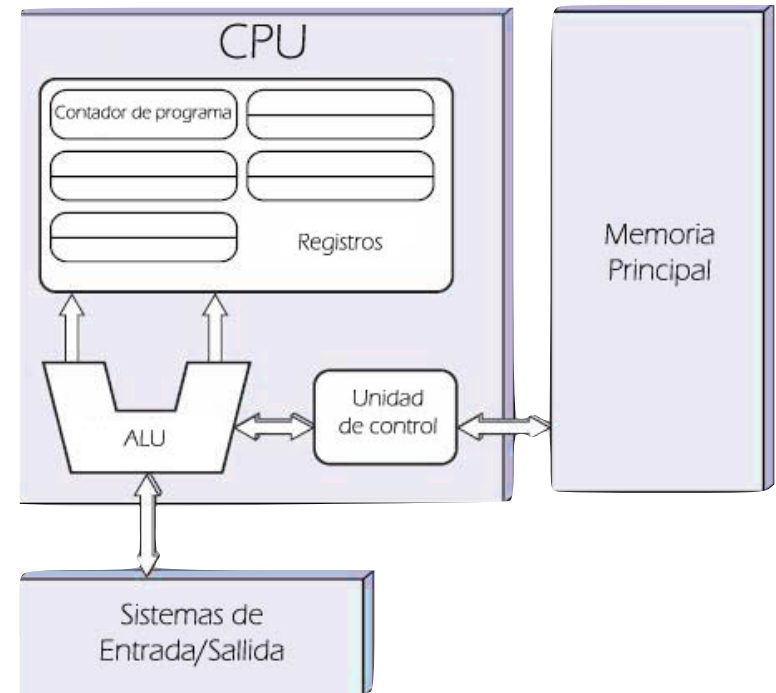
MOV **eax, [L1]**

INC **ebx**

...

ORGANIZACIÓN DEL PROCESADOR

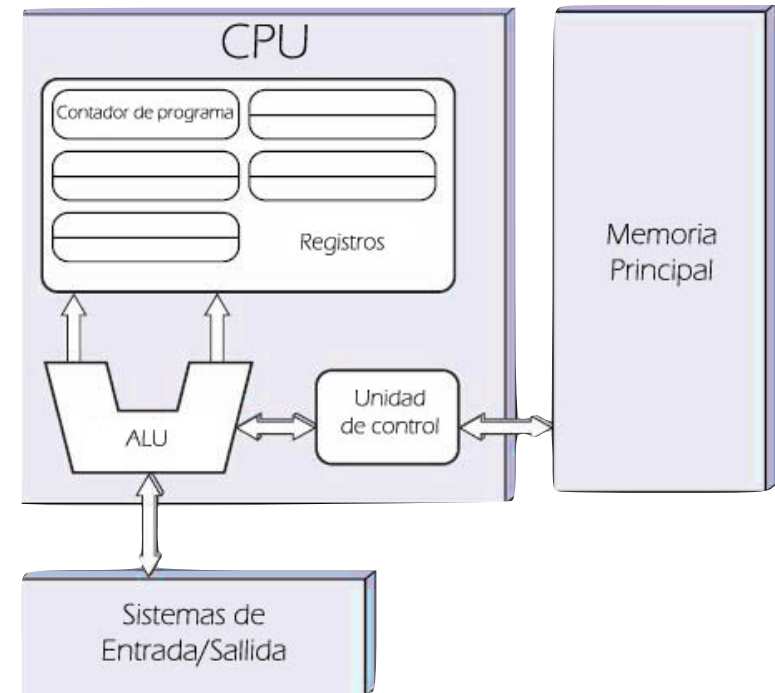
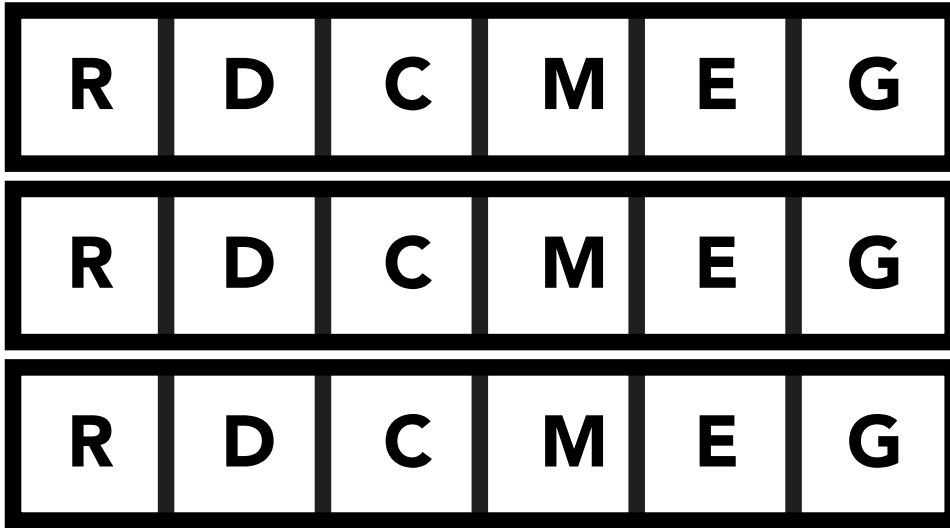
PIPELINE



Tiempo

ORGANIZACIÓN DEL PROCESADOR

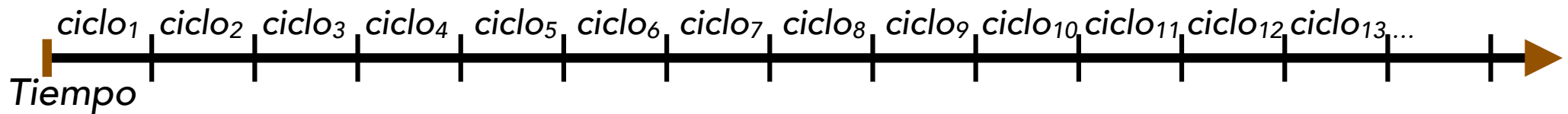
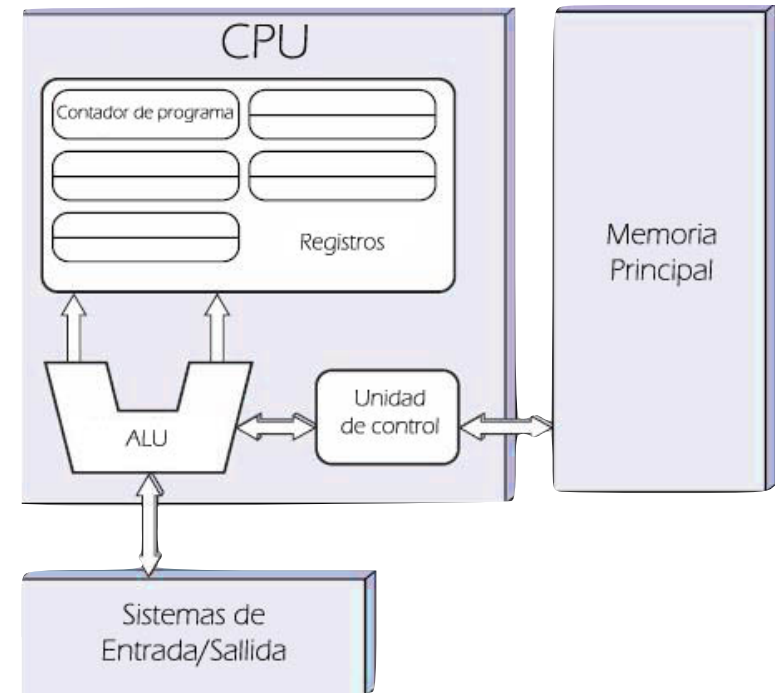
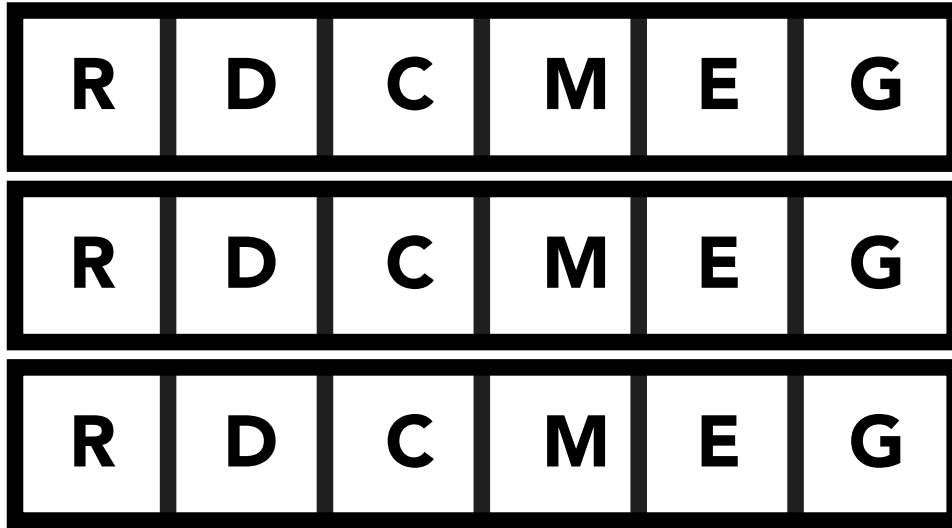
PIPELINE



Tiempo

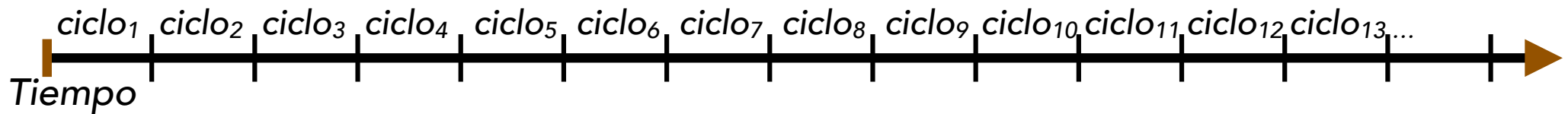
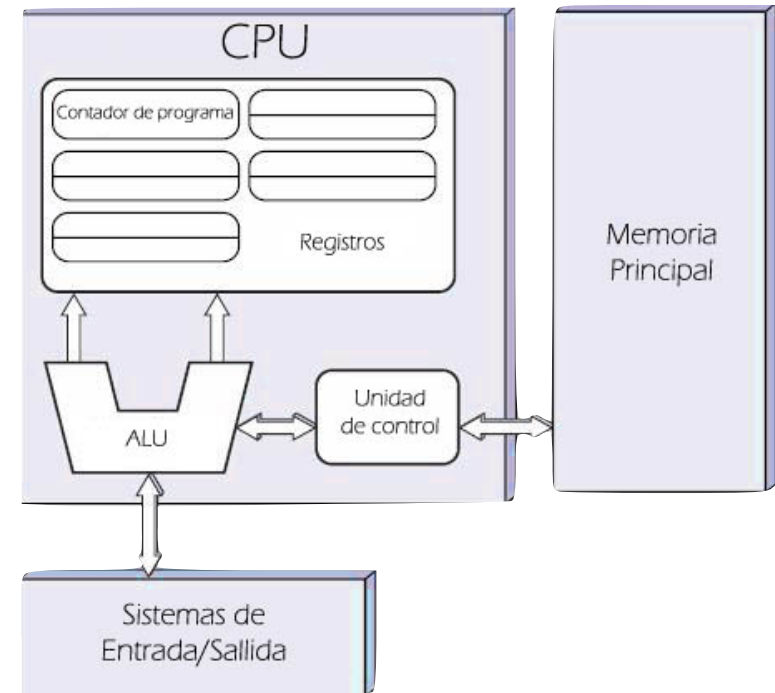
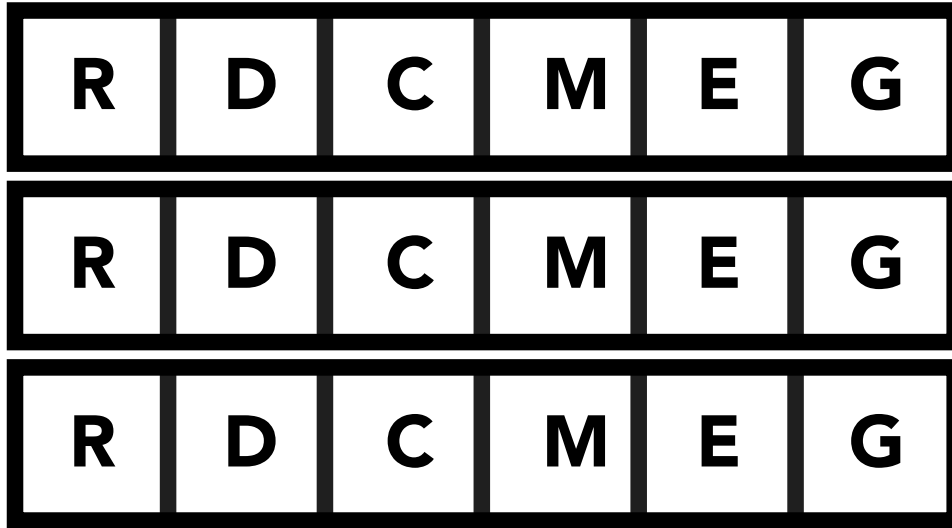
ORGANIZACIÓN DEL PROCESADOR

PIPELINE



ORGANIZACIÓN DEL PROCESADOR

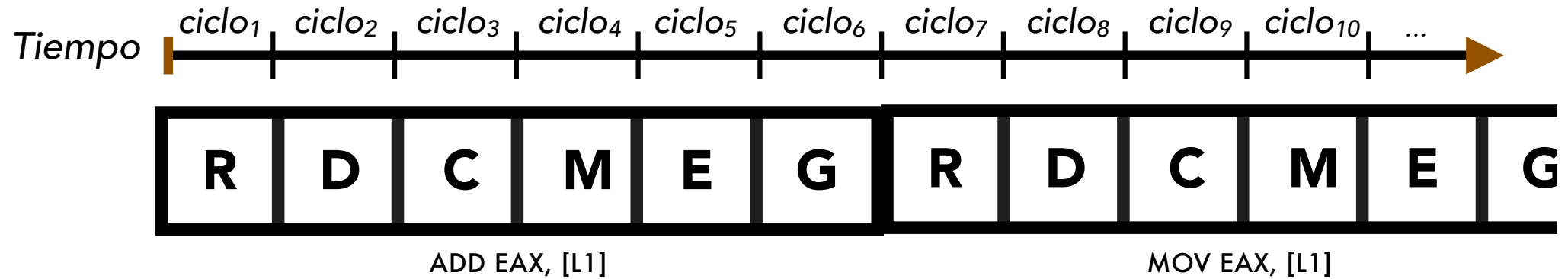
PIPELINE



Tiempo Total (

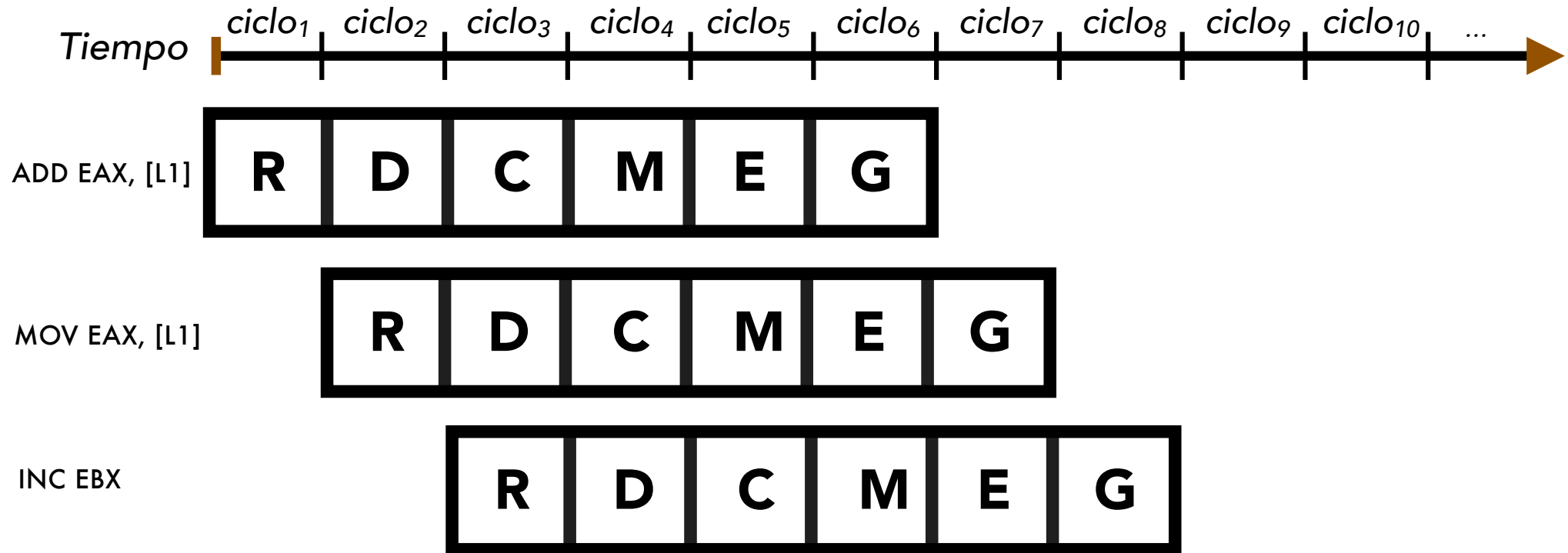
ORGANIZACIÓN DEL PROCESADOR

PIPELINE



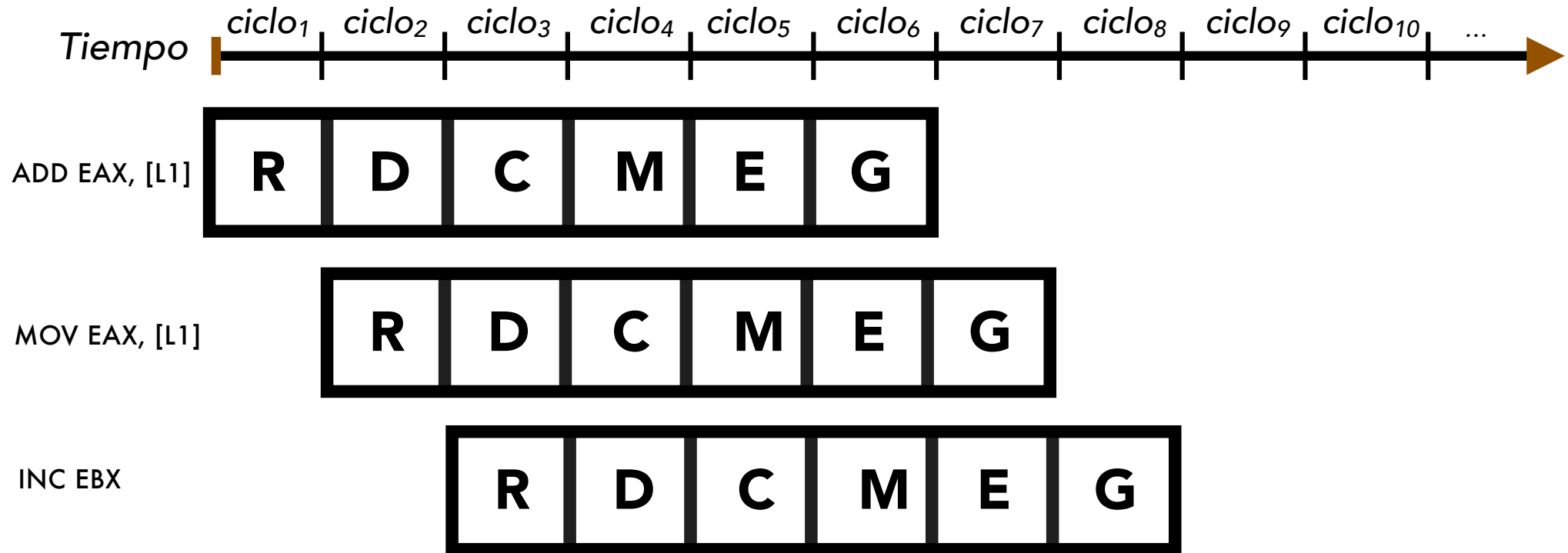
ORGANIZACIÓN DEL PROCESADOR

PIPELINE



ORGANIZACIÓN DEL PROCESADOR

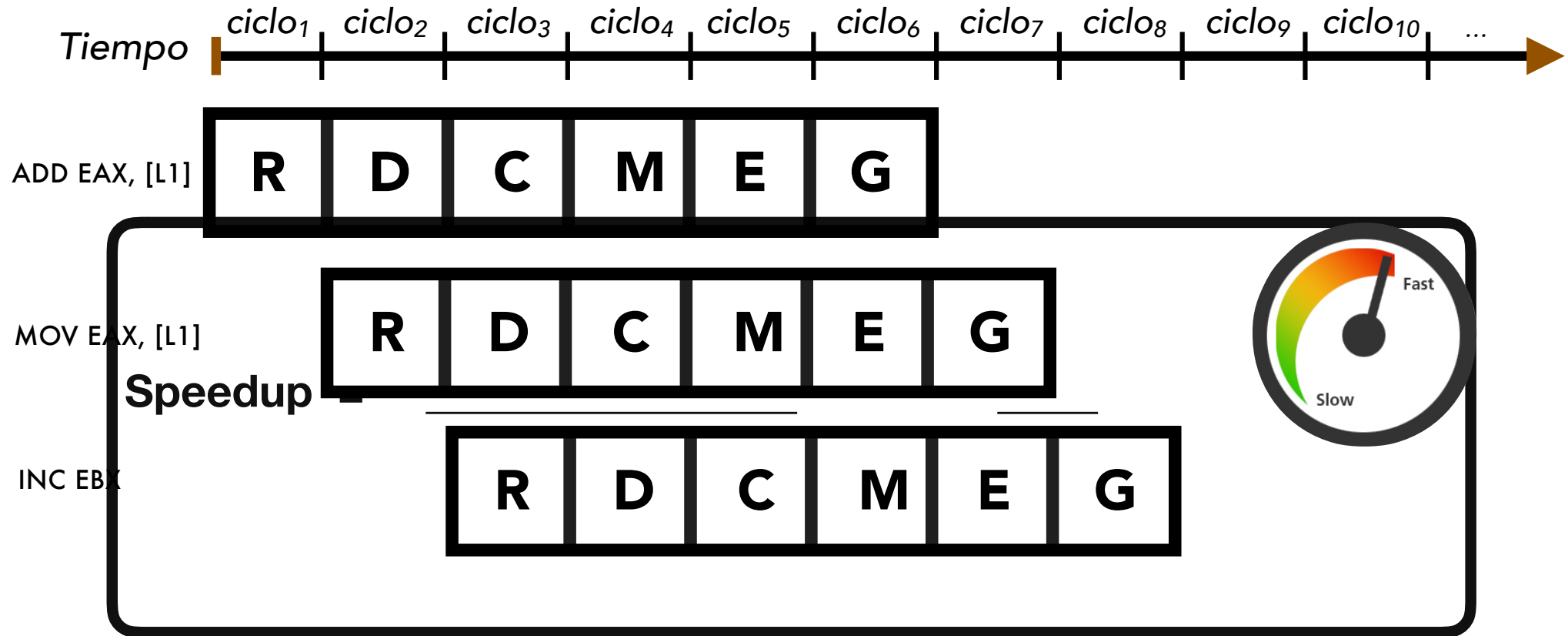
PIPELINE



Tiempo Total (

ORGANIZACIÓN DEL PROCESADOR

PIPELINE



Tiempo Total (

ORGANIZACIÓN DEL PROCESADOR

PIPELINE - CONFLICTOS

- **Conflictos con recursos:** Sucede cuando dos instrucciones necesitan utilizar el mismo recurso, en general la memoria.
- **Dependencia de Datos:** Sucede cuando una instrucción utiliza datos de la/s instrucciones (inmediatas) precedentes.
- **Salto incondicionales:** Cuando cambiamos el flujo de ejecución de las instrucciones, independientemente de contexto (siempre cambia).
- **Salto condicionales:** Cuando cambiamos el flujo de ejecución de las instrucciones dependiendo del contexto en el que se encuentra la ejecución del programa (estado).

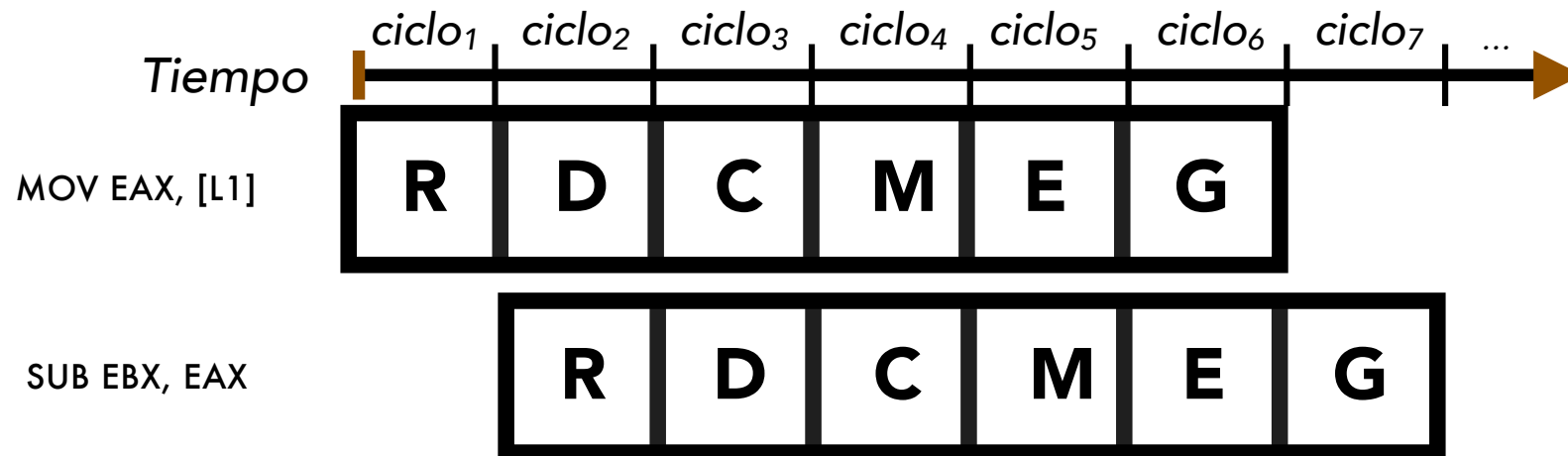
ORGANIZACIÓN DEL PROCESADOR

PIPELINE - CONFLICTOS - DEPENDENCIA DE DATOS

...

MOV **eax, [L1]**

SUB **ebx ,eax**



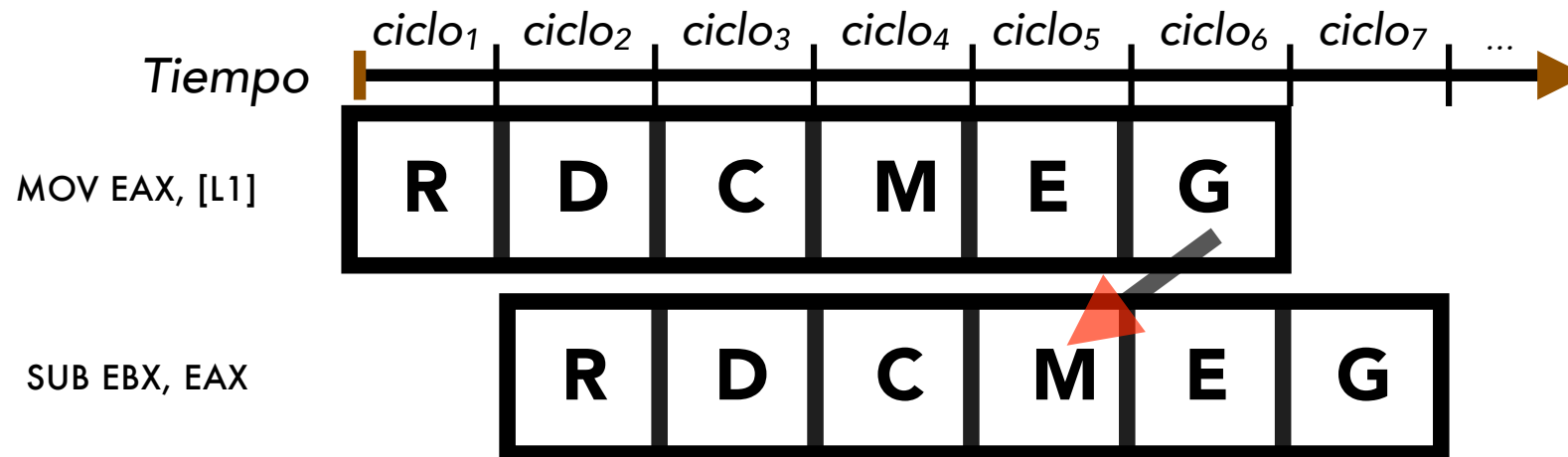
ORGANIZACIÓN DEL PROCESADOR

PIPELINE - CONFLICTOS - DEPENDENCIA DE DATOS

...

MOV **eax, [L1]**

SUB **ebx, eax**



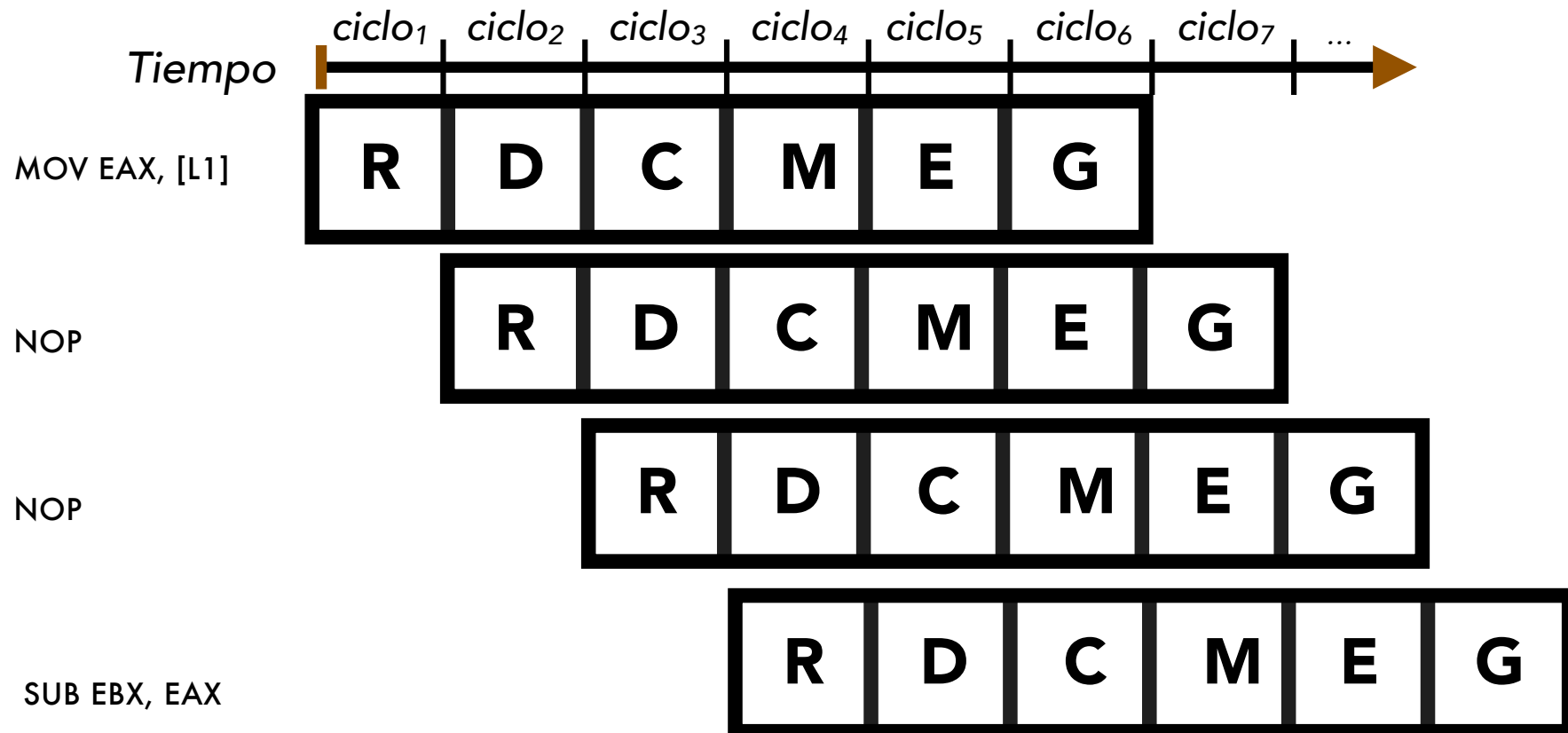
ORGANIZACIÓN DEL PROCESADOR

PIPELINE - CONFLICTOS - DEPENDENCIA DE DATOS

...

MOV **eax, [L1]**

SUB **ebx, eax**



ORGANIZACIÓN DEL PROCESADOR

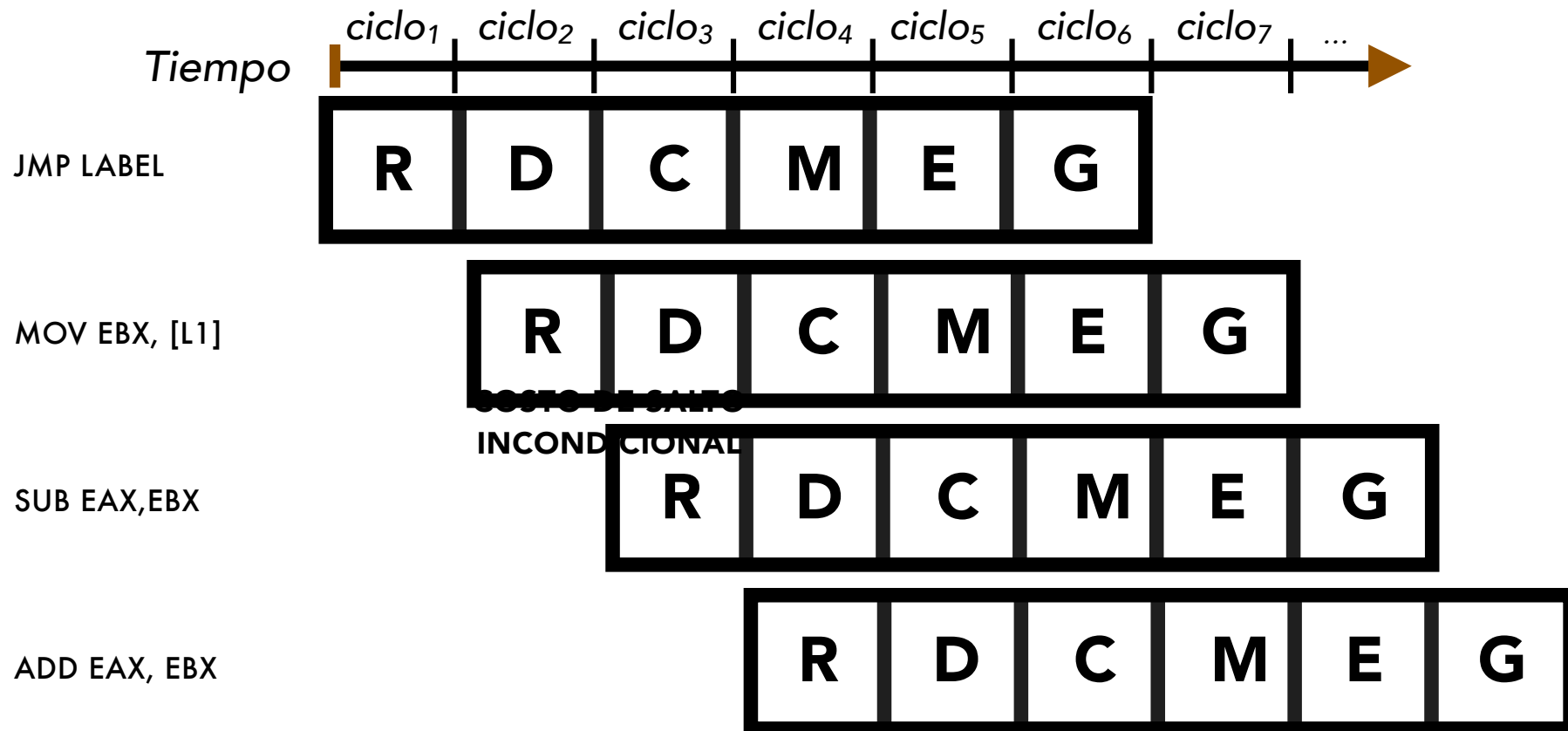
PIPELINE - CONFLICTOS - SALTOS INCONDICIONALES

JMP **label**

MOV **ebx, [L1]**

SUB **eax , ebx**

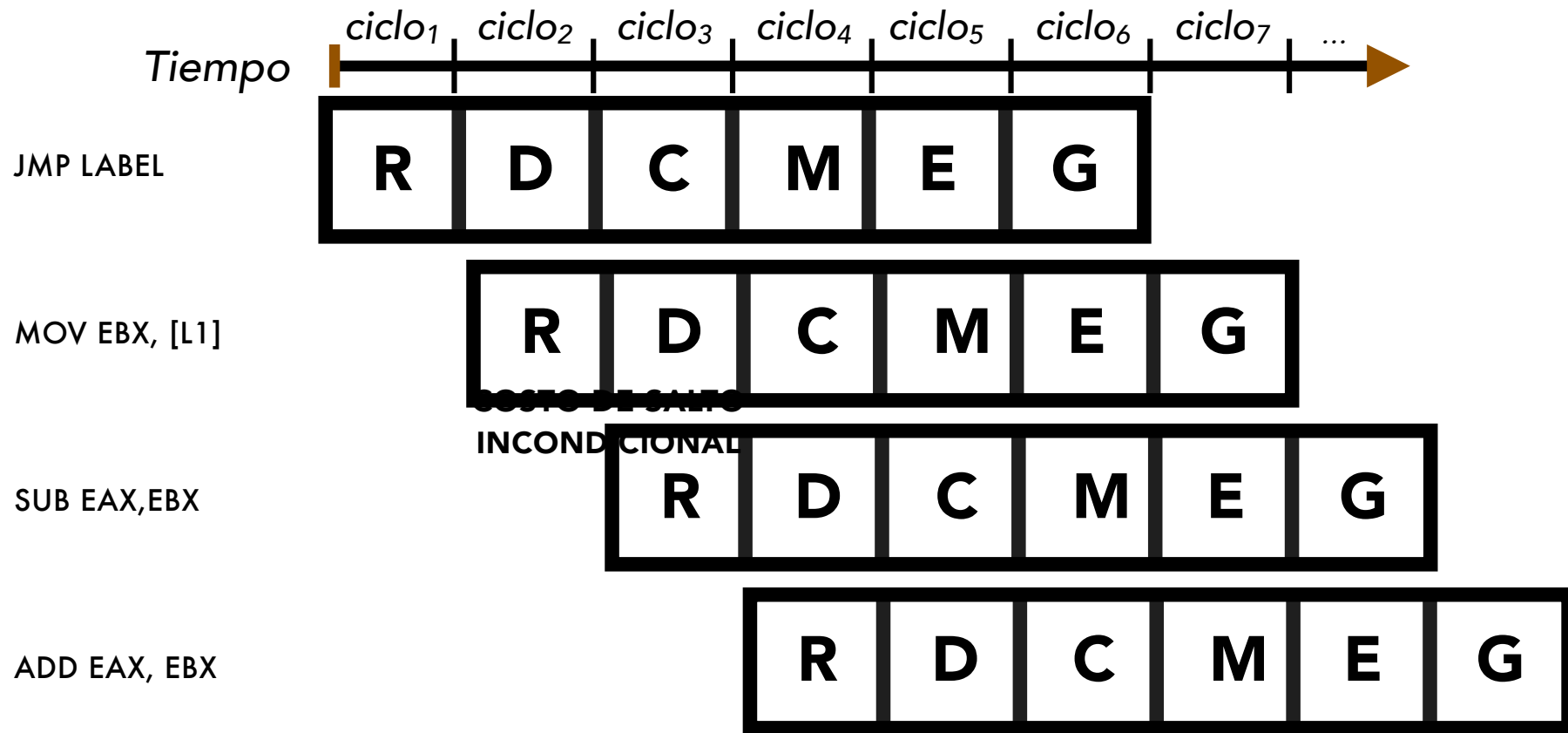
Label **ADD** **eax , ebx**



ORGANIZACIÓN DEL PROCESADOR

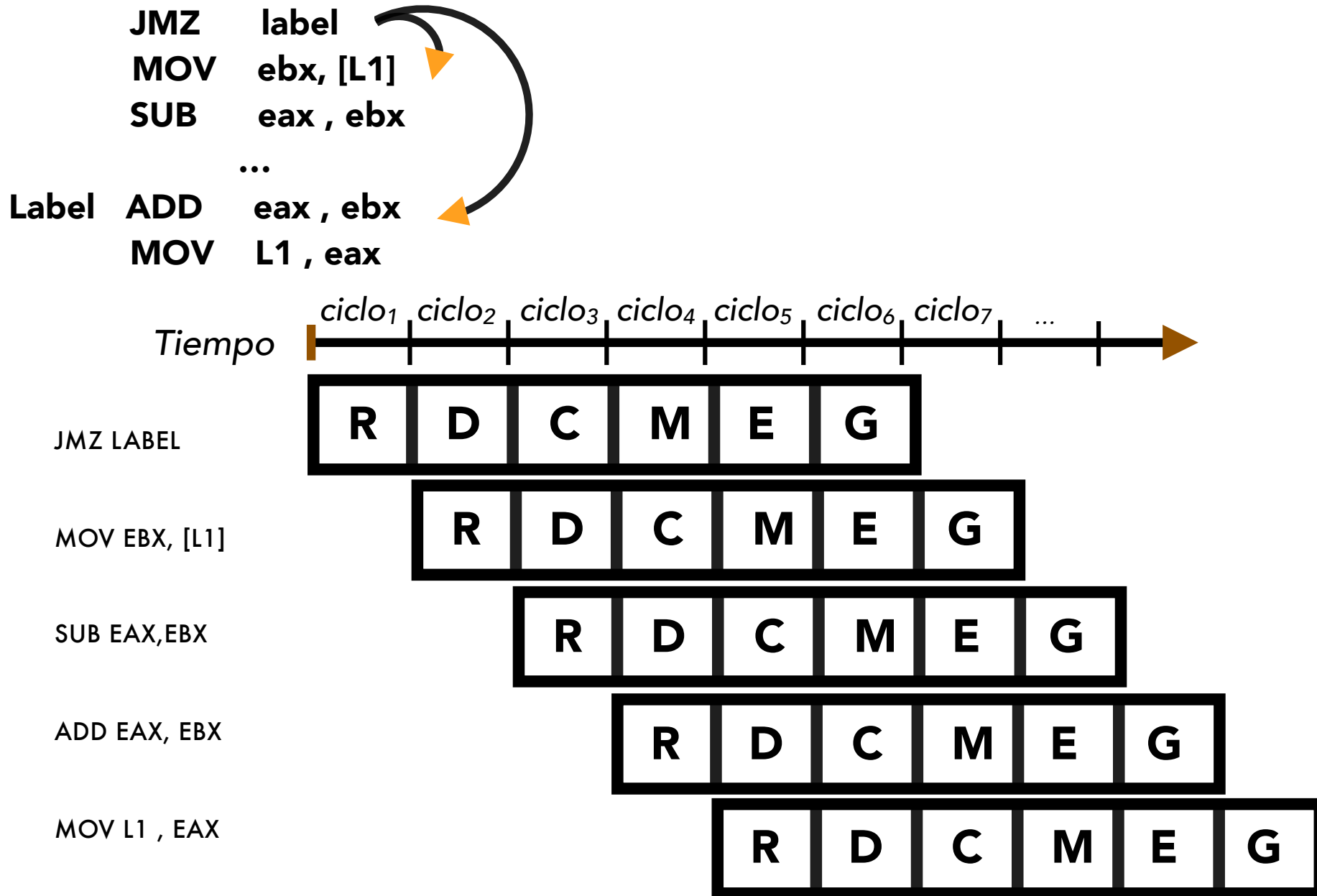
PIPELINE - CONFLICTOS - SALTOS INCONDICIONALES

JMP label
MOV ebx, [L1]
SUB eax, ebx
Label ADD eax, ebx



ORGANIZACIÓN DEL PROCESADOR

PIPELINE - CONFLICTOS - SALTOS CONDICIONALES



ORGANIZACIÓN DEL PROCESADOR

PIPELINE - CONFLICTOS - SALTOS CONDICIONALES

JMZ label
MOV ebx, [L1]
SUB eax, ebx

...

Label ADD eax, ebx,
MOV L1, eax

Predicción estática: Por ejemplo, asume que nunca va a saltar

ciclo₁ ciclo₂ ciclo₃ ciclo₄ ciclo₅ ciclo₆ ciclo₇ ...

Predicción dinámica: Toman información de ejecuciones

previas, por ejemplo, asume que la próxima vez va a tomar la misma decisión que la anterior.

JMZ LABEL
MOV EBX, [L1]

Proceso paralelo: en multiprocesadores, se procesan ambos branches en paralelo y con un flag se indica cuál fue tomado.

SUB EAX,EBX
ADD EAX,EBX

MOV L1, EAX

