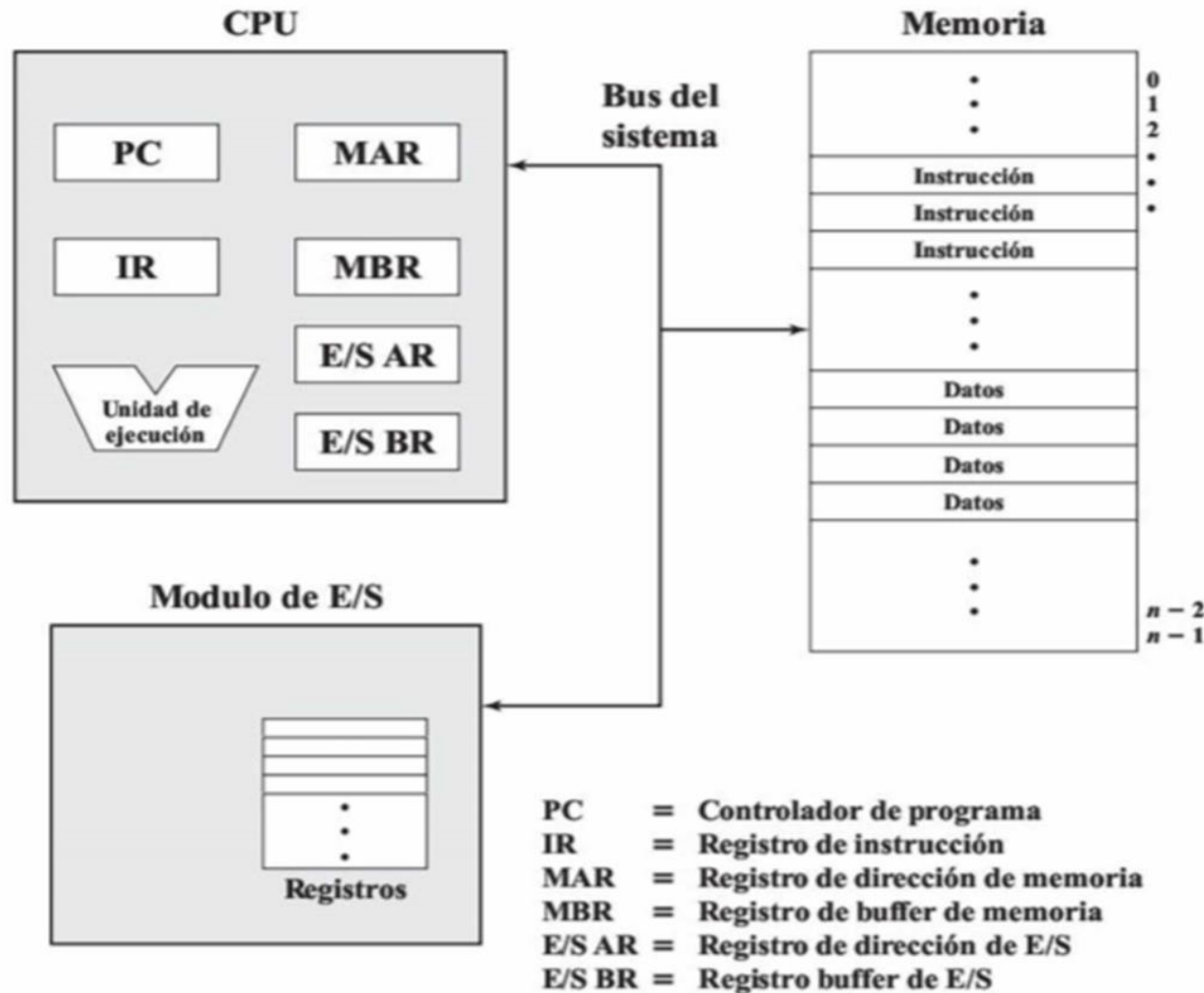


# Los Ciclos de Captación y Ejecución

---

UNIVERSIDAD DE SONSONATE

# Los Ciclos de Captación y Ejecución



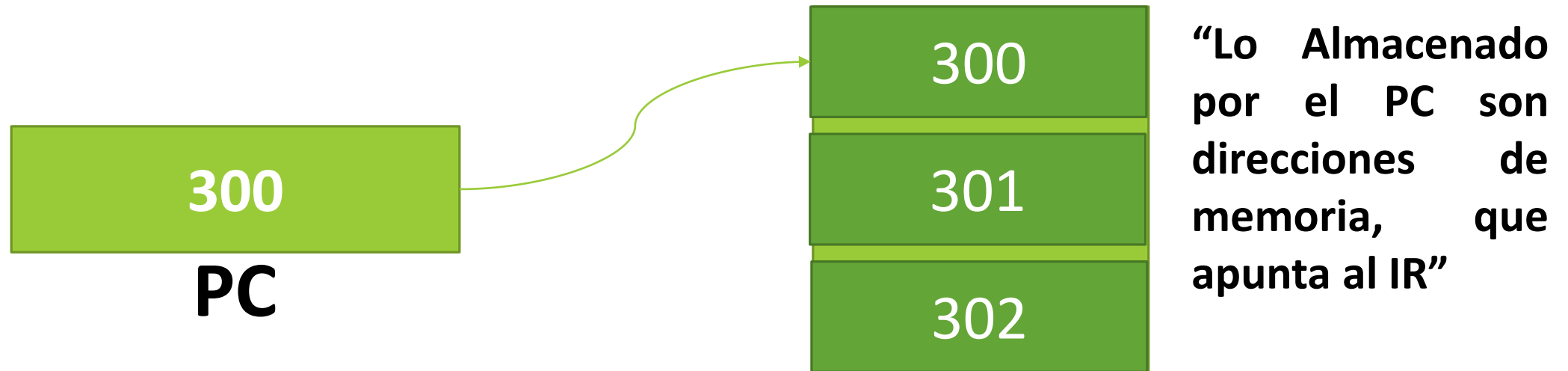
# Los Ciclos de Captación y Ejecución

---

Al comienzo de cada ciclo de instrucción, la CPU capta una instrucción de memoria. En una CPU típica, se utiliza un registro llamado contador de programa (**PC, Program Counter**) para seguir los flujos de la instrucción que debe ejecutar a continuación. A no ser que se indique otra cosa, la CPU siempre incrementa el **PC** de comprender cada instrucción, de forma que comprende la siguiente instrucción de la secuencia

# Los Ciclos de Captación y Ejecución

**Ejemplo:** Una computadora en que la instrucción ocupa una palabra de **memoria de 16 bits**, Se su pone que el contador de programa almacena el valor **300**. El CPU captara la próxima instrucción almacenada en la posición **301, 302, 303**. Esta secuencia es posible alterarla.



# Los Ciclos de Captación y Ejecución

---

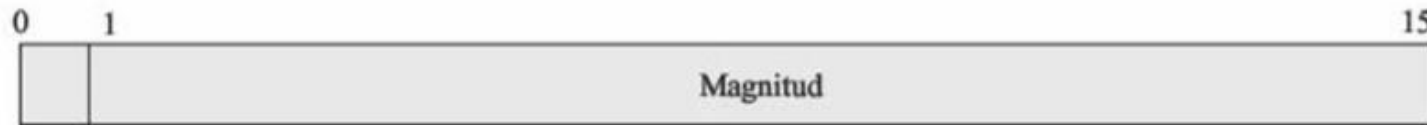
La instrucción captada se almacena en un registro de la CPU conocido como **registro de instrucción (IR, Instruction Register)**. *La instrucción se escribe utilizando un código binario que especifica la acción que debe realizar el CPU.* El CPU interpreta la instrucción y lleva a cabo la acción requerida. En general, esta puede ser de cuatro tipos:

- **Procesador-Memoria:** Deben transferirse datos desde la CPU a la memoria, o desde la memoria a la CPU.
- **Procesador-E/S:** Deben transferirse datos a o desde el exterior mediante transferencias entre la CPU y un módulo de E/S.
- **Procesamiento de Datos:** El CPU ha de realizar alguna operación aritmética o lógica con los datos.
- **Control:** Una instrucción puede especificar que la secuencia de ejecución se altere (como la instrucción de salto LAS). Por ejemplo, El CPU capta una instrucción de la posición 149 que especifica que la siguiente instrucción debe captarse de la posición 182. El CPU registrará este hecho poniendo en el contador de programa 182. Así, en el próximo ciclo de captación, la instrucción se cargará desde la posición 182 en lugar de desde la posición 150.

# Los Ciclos de Captación y Ejecución



(a) Formato de instrucción



(b) Formato de enteros

Contador de programa (PC) = Dirección de instrucción  
Registro de instrucción (IR) = Instrucción en ejecución  
Acumulador (AC) = Almacenamiento temporal

(c) Registros internos de la CPU

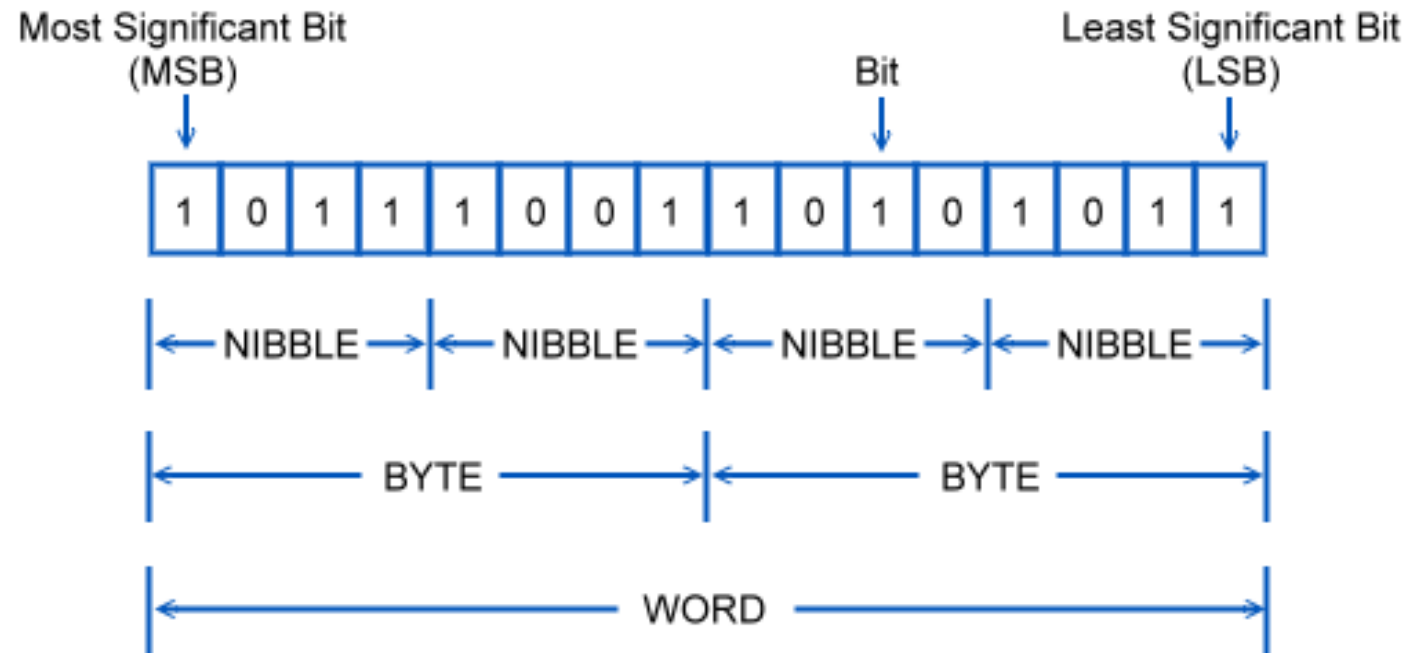
0001 = Cargar AC desde memoria  
0010 = Almacenar AC en memoria  
0101 = Sumar a AC un dato de memoria

(d) Lista parcial de Codops («códigos de operación»)

**Otro Ejemplo:** Considere un ejemplo sencillo donde utilizamos una “*maquina hipotética*” que incluye las características enumeradas a continuación:

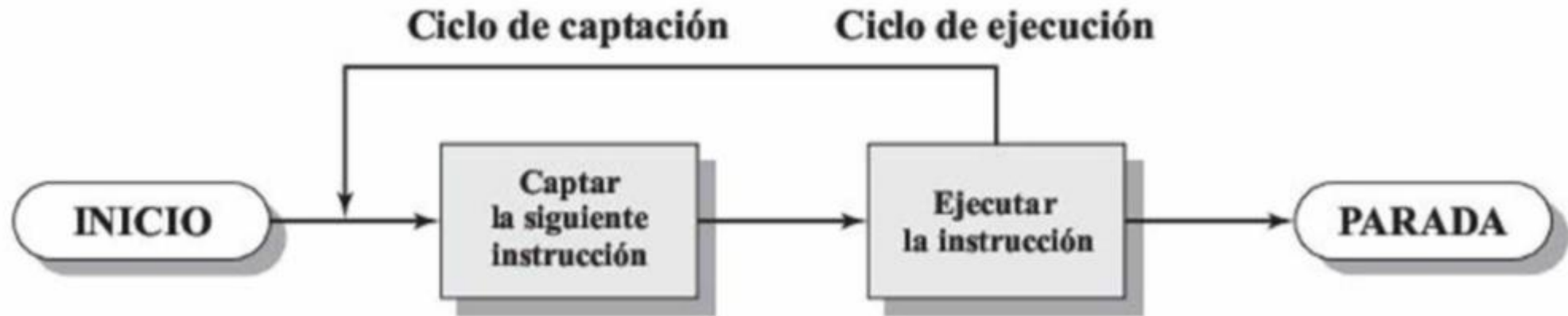
# Los Ciclos de Captación y Ejecución

El procesador posee un único registro de datos llamado **acumulador (AC)**. Tanto las instrucciones como los datos son de 16 bits. Así, es conveniente organizar la memoria utilizando posiciones de 16 bits, o **palabras**. El formato de instrucción indica que puede haber  **$2^4 = 16$  códigos de operación (codops) diferentes**, y se pueden direccionar directamente hasta  **$2^{12} = 4096$  (4K) palabras de memoria**.



# Los Ciclos de Captación y Ejecución

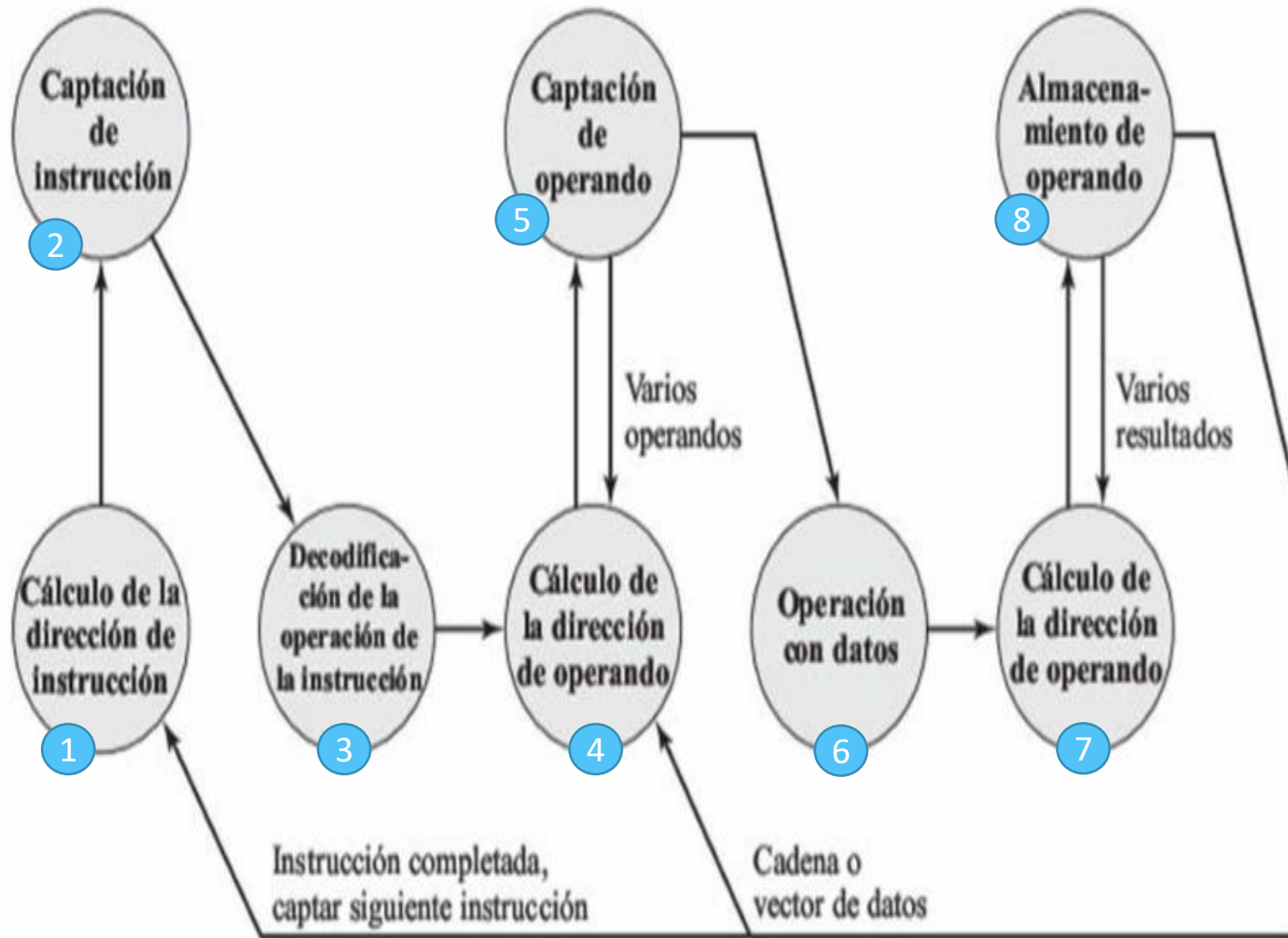
Si los **Ciclos de Instrucciones** fueran en particular solo para la *captación y ejecución de Instrucciones o de referenciar memoria* podríamos mantener nuestro esquema inicial intacto:





# Los Ciclos de Captación y Ejecución

Pero nuestras computadoras en vez de referencias solamente memoria, o una instrucción puede *especificar una operación de E/S.*



Para un ciclo de instrucción dado, algunos estados pueden no darse y otros pueden visitarse más de una vez. Los estados se describen a continuación:

# Los Ciclos de Captación y Ejecución

---

**1. Cálculo de la dirección de la instrucción (IAC, Instruction Address Calculation):** Determina la dirección de la siguiente instrucción a ejecutar. Normalmente, esto implica añadir un número fijo a la dirección de la instrucción previa.

***Por ejemplo,*** si las instrucciones tienen un tamaño de 16 bits y la memoria se organiza en palabras de 16 bits, se suma 1 a la dirección previa. En cambio, si la memoria se organiza en bytes (8 bits) direccionables individualmente, entonces hay que sumar 2 a la dirección previa.

# Los Ciclos de Captación y Ejecución

---

**2. Captación de instrucción (if, Instruction Fetch):** El CPU lee la instrucción desde su posición en memoria.

**3. Decodificación de la operación indicada en la instrucción (IOD, Instruction Operation Decoding):** analiza la instrucción para determinar el tipo de operación a realizar y el (los) operando(s) a utilizar.

**4. Cálculo de la dirección del operando (OAC, Operand Address Calculation):** Si la instrucción implica una referencia a un operando en memoria o disponible mediante E/S, determina la dirección del operando.

# Los Ciclos de Captación y Ejecución

---

- 5. Captación de operando (OF, Operand Fetch):** Capta el operando desde memoria o se lee desde el dispositivo de E/S.
- 6. Operación con los datos (DO, Data Operation):** Realiza la operación indicada en la instrucción.
- 7. Realizamos lo descrito en el PASO 4....**
- 8. Almacenamiento de operando (OS, Operand Store):** Escribe el resultado en memoria o lo saca a través de un dispositivo de E/S

# Los Ciclos de Captación y Ejecución

---

La pregunta que nos surge en estos momentos podría ser....

***¿Por qué es importante que conozcamos como funcionan los ciclos de captación y Ejecución?***



# Los Ciclos de Captación y Ejecución

---

Es importante ya que ahora sabemos que existe la posibilidad de que en los ciclos de instrucción, pueda existir la interacción de los elementos de E/S.

Y para ello es importante además que conozcamos el tema de:

## INTERRUPCIONES



# Practica con Logisim

Utilizando la solución de los ejercicios anteriores simule las compuertas lógicas encontradas.

