

# Arquitectura de Computadores

Tema: Ciclo de una Instrucción.

Profesor: Manuel Olivares Ávila

# Ciclo de una Instrucción.

## Modelo de Von Neumann

- Datos e instrucciones se almacenan en memoria principal como números binarios.
- A menos que se indique lo contrario, los datos e instrucciones se almacenarán usando el acuerdo *Little Endian Byte Order*, el byte menos significativo del dato se ubica en la dirección más baja.

Address	Data
300	0 1 0 0 0 0 0 0
301	0 0 0 1 1 0 0 1
302	0 1 0 0 0 0 0 1
303	0 1 0 1 1 0 0 1
304	0 1 0 0 0 0 0 1
305	0 0 1 0 1 0 0 1
...	...
...	...
...	...
940	0 0 0 0 0 0 1 1
941	0 0 0 0 0 0 0 0
942	0 0 0 0 0 0 1 0
943	0 0 0 0 0 0 0 0

# Ciclo de una Instrucción.

## Modelo de Von Neumann

- Suponiendo que una instrucción está formada por 2 bytes (ver selección de la derecha), el byte más significativo se asocia con la dirección 301

00011001

mientras que el byte menos significativo se asocia con la dirección de memoria 300

01000000

- La instrucción finalmente sería: 0001100101000000

Address

Data

300

0 1 0 0 0 0 0 0

301

0 0 0 1 1 0 0 1

302

0 1 0 0 0 0 0 1

303

0 1 0 1 1 0 0 1

304

0 1 0 0 0 0 0 1

305

0 0 1 0 1 0 0 1

...

...

...

...

...

...

940

0 0 0 0 0 0 1 1

941

0 0 0 0 0 0 0 0

942

0 0 0 0 0 0 1 0

943

0 0 0 0 0 0 0 0

# Ciclo de una Instrucción.

## Modelo de Von Neumann

- De la misma forma, si los datos se representan con 2 bytes, el valor correcto para el dato destacado en azul corresponde a:

000000000000000011

que escrito en hexadecimal quedaría:

0003

- Para abreviar la escritura de números binarios, se utiliza su equivalente en hexadecimal.

Address	Data
300	0 1 0 0 0 0 0 0
301	0 0 0 1 1 0 0 1
302	0 1 0 0 0 0 0 1
303	0 1 0 1 1 0 0 1
304	0 1 0 0 0 0 0 1
305	0 0 1 0 1 0 0 1
...	...
...	...
...	...
940	0 0 0 0 0 0 1 1
941	0 0 0 0 0 0 0 0
942	0 0 0 0 0 0 1 0
943	0 0 0 0 0 0 0 0

# Ciclo de una Instrucción.

## Instrucciones

Format	OpCode				Address											
Instruction	0	0	0	1	1	0	0	1	0	1	0	0	0	0	0	0

- Cuando se capta una instrucción desde memoria, se debe hacer calzar con uno o más formatos previamente definidos.
- En el ejemplo, los cuatro bits más significativos corresponden al código de operación, y los restantes hacen referencia a una dirección de memoria.
- A la acción de separar el código de operación y los parámetros de una instrucción se denomina “Decodificación de la instrucción”.

# Ciclo de una Instrucción.

## Instrucciones

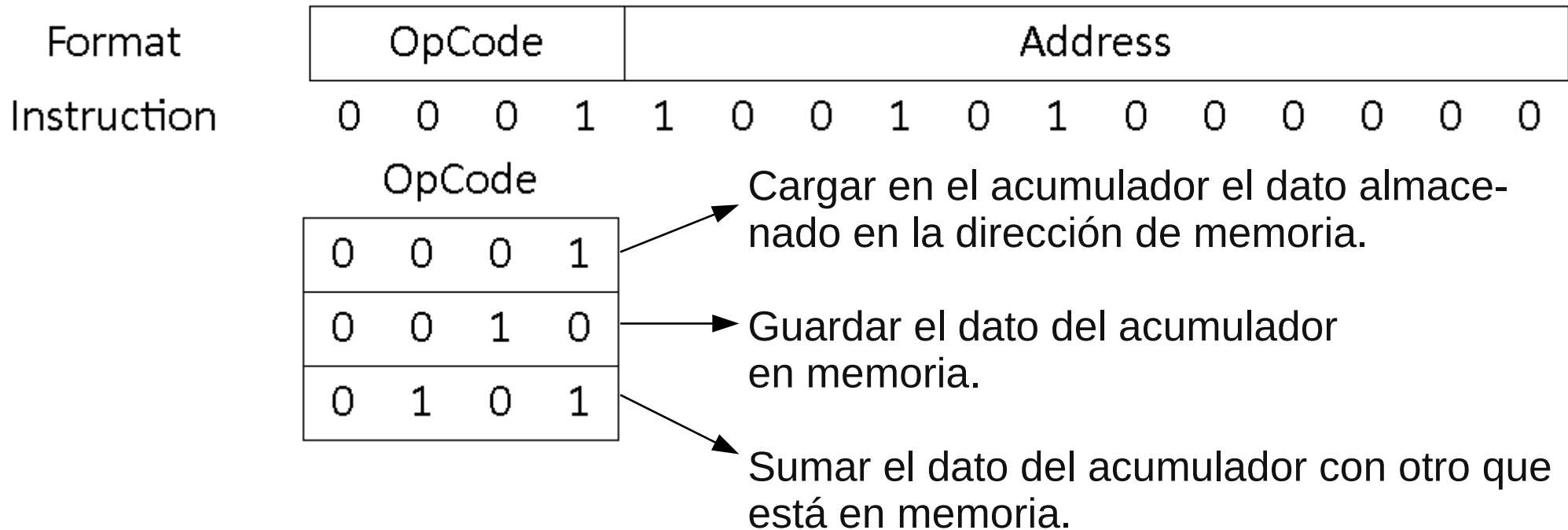
Format	OpCode	Address
Instruction	0 0 0 1	1 0 0 1 0 1 0 0 0 0 0 0

- En la etapa de diseño, se definen los códigos de operación admitidos para las diferentes funcionalidades que el computador realizará.
- Por ejemplo, se pueden definir los siguientes códigos de operación:

OpCode	
0 0 0 1	Cargar en el acumulador el dato almacenado en la dirección de memoria.
0 0 1 0	Guardar el dato del acumulador en memoria.
0 1 0 1	Sumar el dato del acumulador con otro que está en memoria.

# Ciclo de una Instrucción.

## Instrucciones



- En el ejemplo, se cargará en el acumulador el dato almacenado en la dirección de memoria 940.

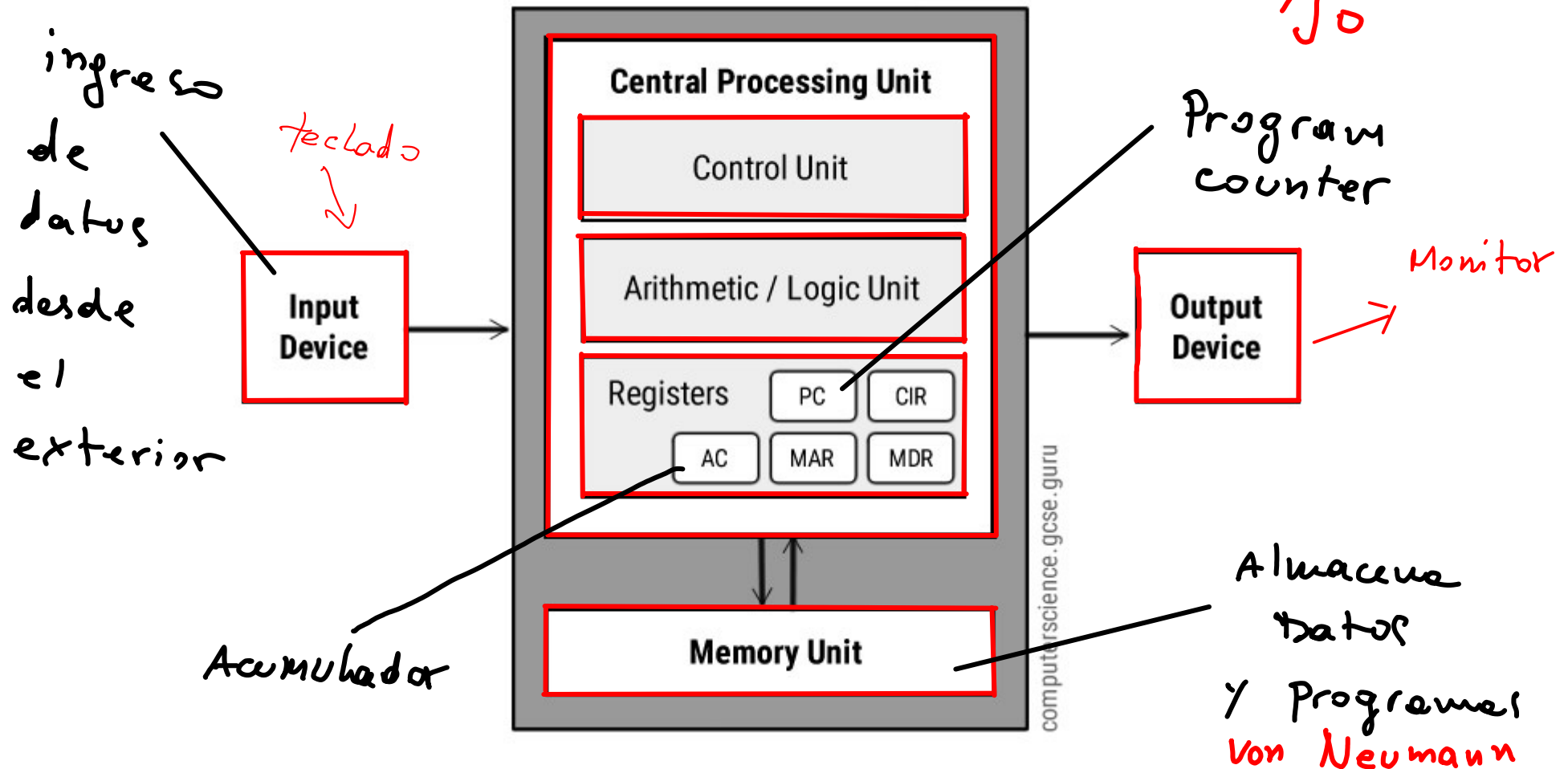
1) pedir edad al usuario

2) comparar el dato ingresado

3) imprimir mensaje

# Ciclo de una Instrucción.

## Modelo de Von Neumann



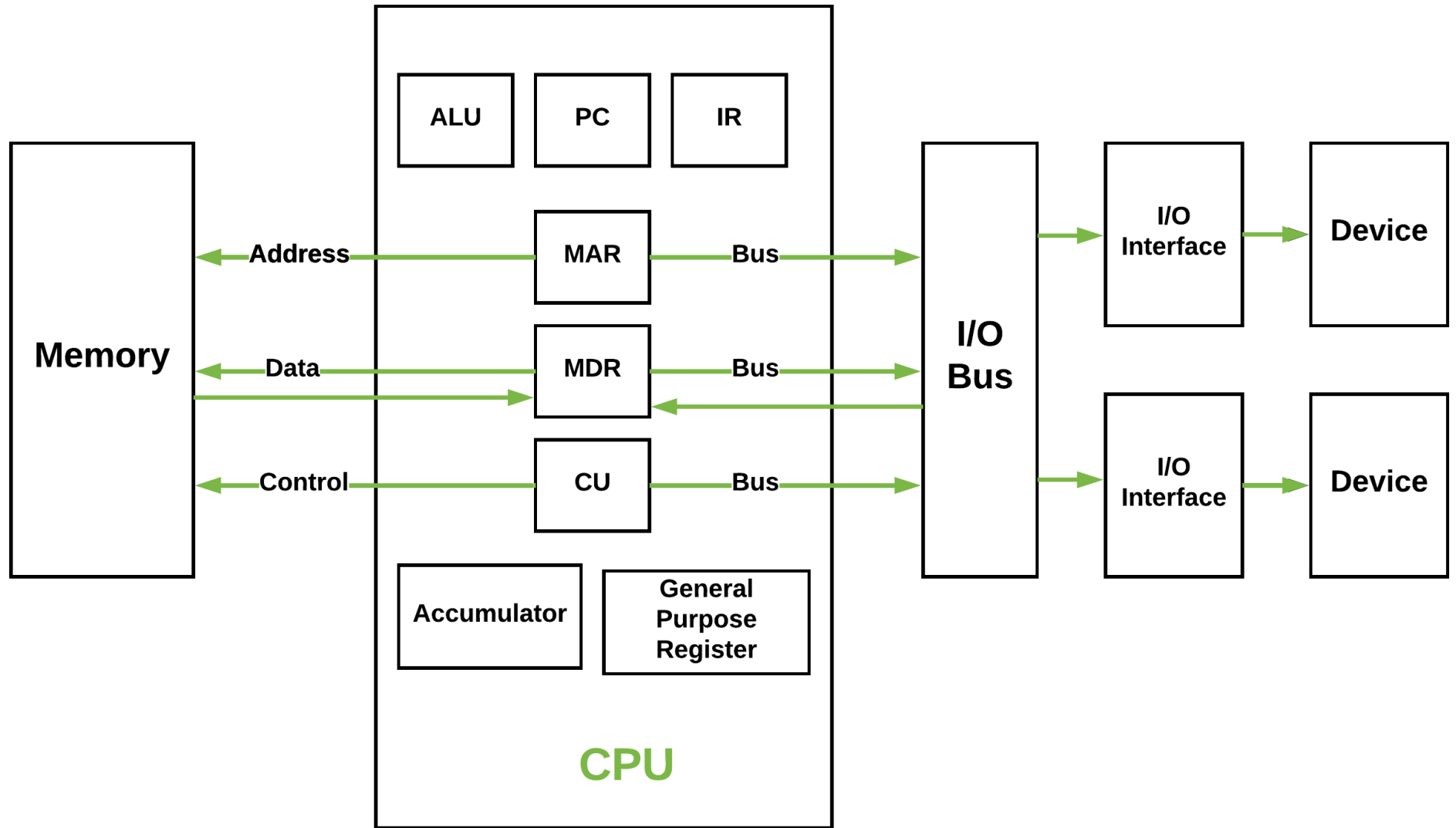
PC: Program Counter; CIR = IR: Instruction Register; AC = AX: Accumulator

MAR: Memory Address Register; MDR = MBR: Memory Data Register



# Ciclo de una Instrucción.

## Modelo de Von Neumann



# Ciclo de una Instrucción.

## Programa

- Un Programa es una secuencia de instrucciones que realizan un procesamiento para obtener un resultado.
- Por ejemplo, la zona destacada corresponde a un programa...
- El procesamiento de las instrucciones del programa se realizan mediante el “Ciclo de una Instrucción”
- Este ciclo se inicia cuando el computador se enciende y termina cuando se apaga.

Address

Data

300

0 1 0 0 0 0 0 0

301

0 0 0 1 1 0 0 1

302

0 1 0 0 0 0 0 1

303

0 1 0 1 1 0 0 1

304

0 1 0 0 0 0 0 1

305

0 0 1 0 1 0 0 1

...

...

...

...

...

940

0 0 0 0 0 0 1 1

941

0 0 0 0 0 0 0 0

942

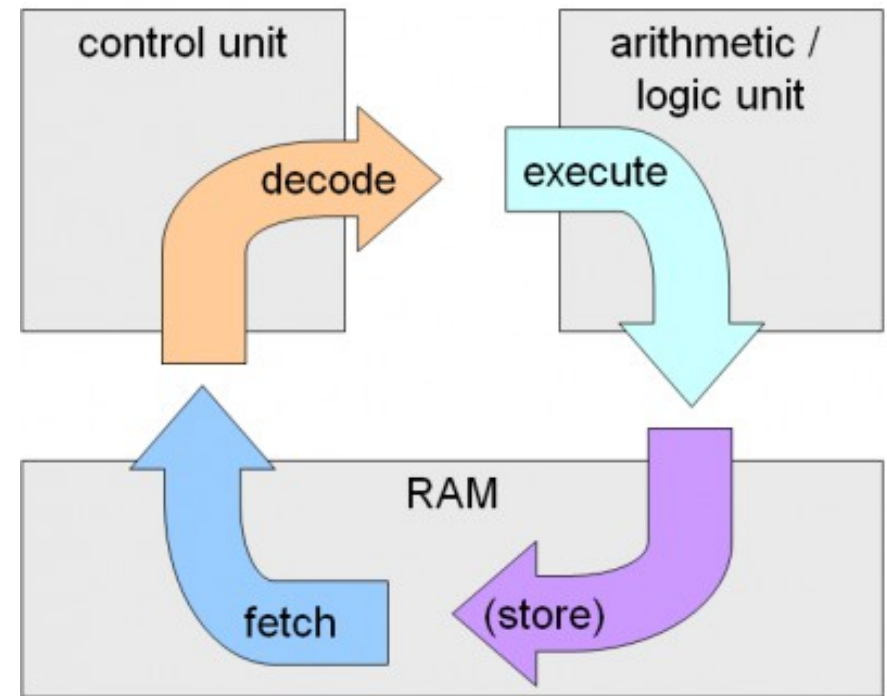
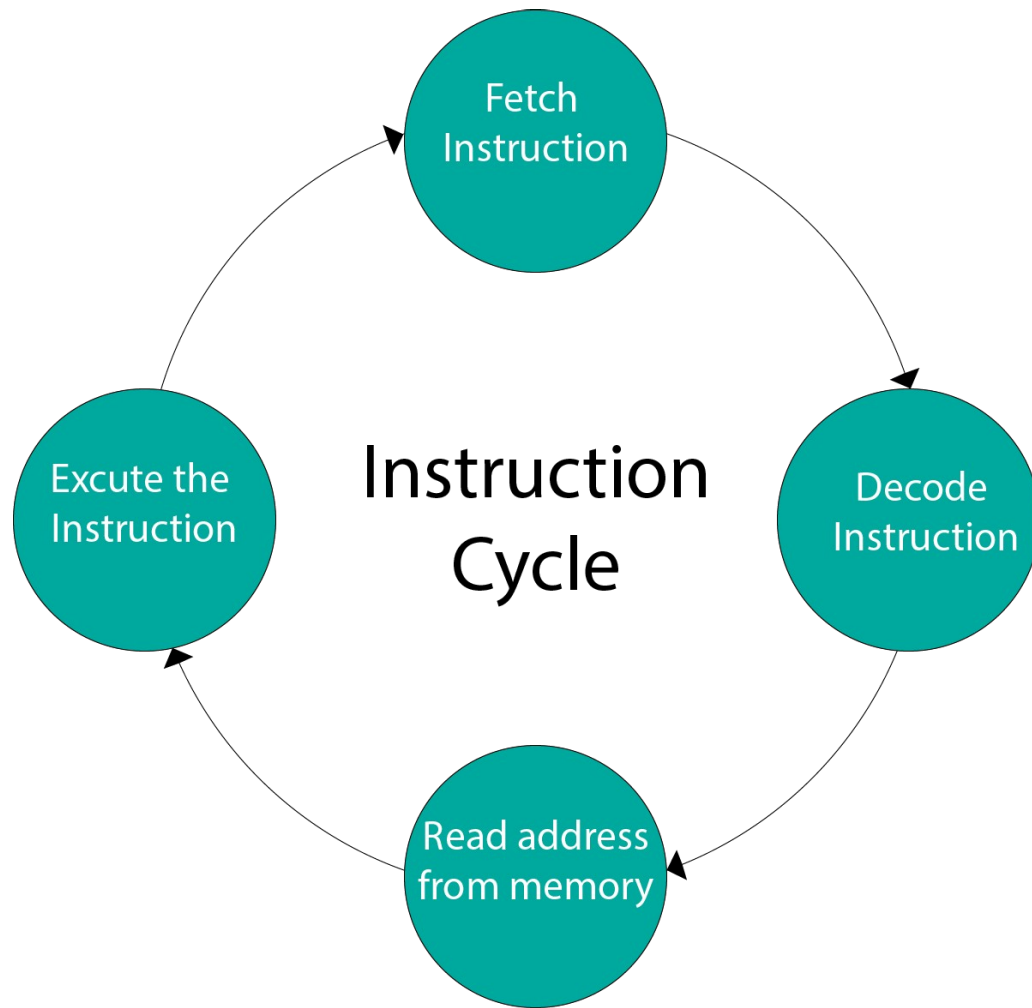
0 0 0 0 0 0 1 0

943

0 0 0 0 0 0 0 0

# Ciclo de una Instrucción.

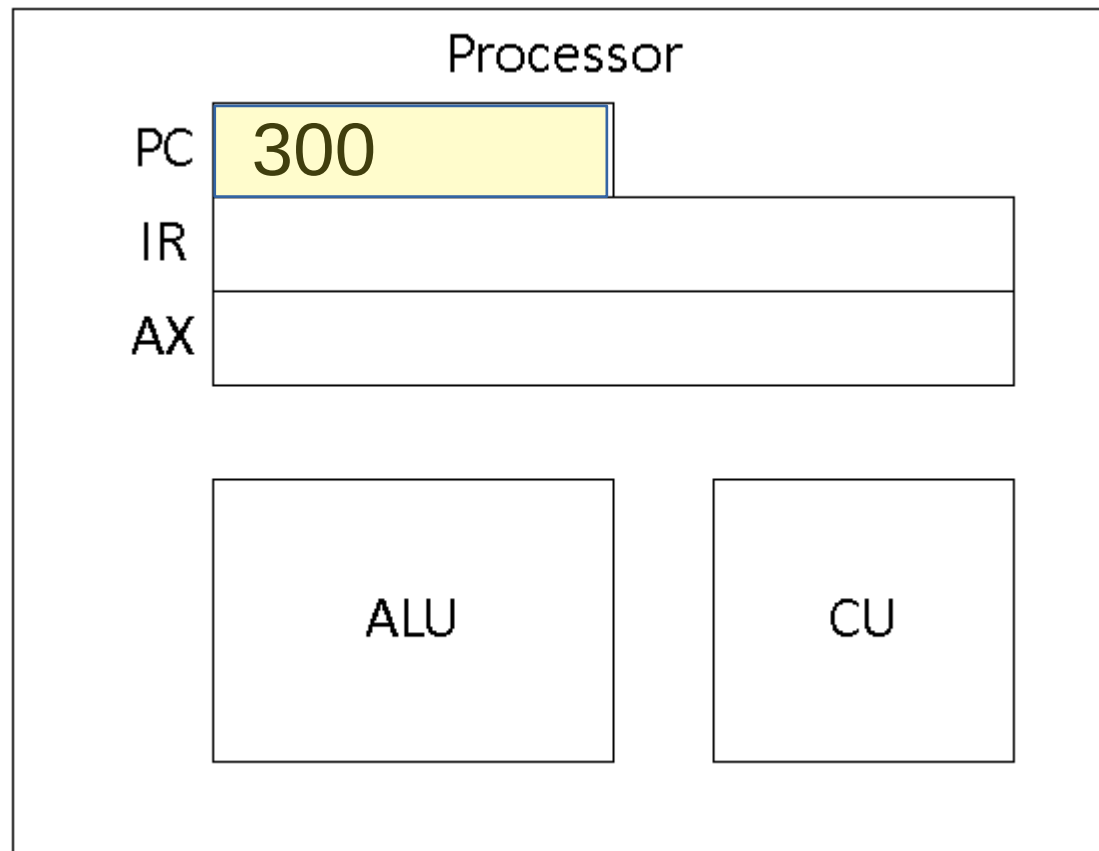
## Ciclo de una Instrucción



# Ciclo de una Instrucción.

## Ciclo de una Instrucción

INICIO



Address

Data

300	0	1	0	0	0	0	0	0
301	0	0	0	1	1	0	0	1
302	0	1	0	0	0	0	1	0
303	0	1	0	1	1	0	0	1
304	0	1	0	0	0	0	0	1
305	0	0	1	0	1	0	0	1

...

...

...

...

...

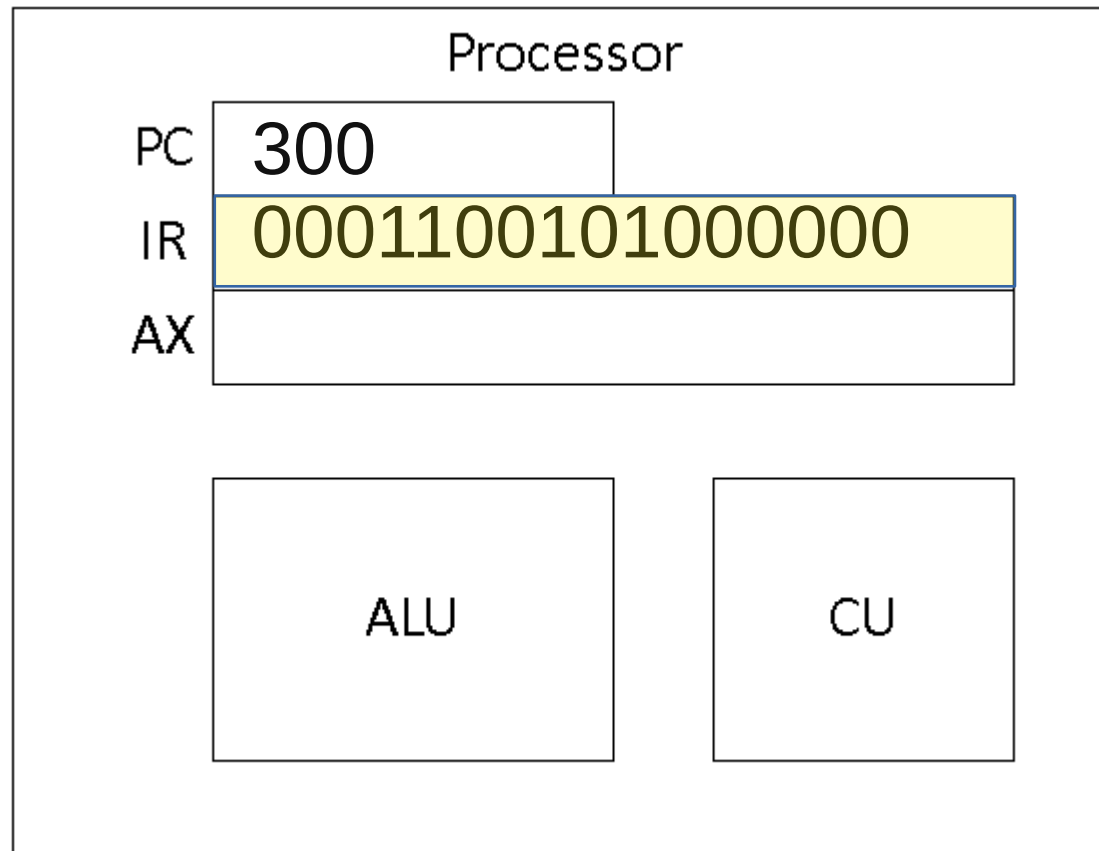
...

940	0	0	0	0	0	0	1	1
941	0	0	0	0	0	0	0	0
942	0	0	0	0	0	0	1	0
943	0	0	0	0	0	0	0	0

# Ciclo de una Instrucción.

## Ciclo de una Instrucción

### Fetch Instruction (Captación)

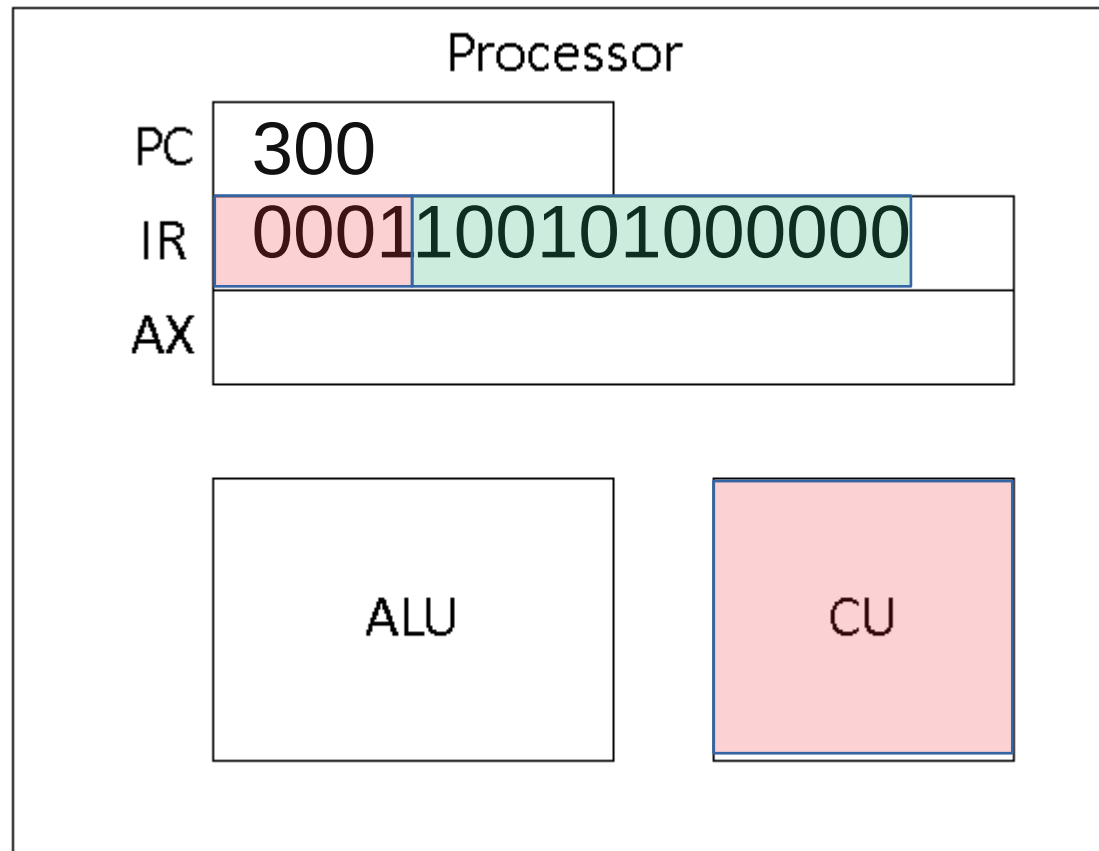


Address	Data
300	0 1 0 0 0 0 0 0
301	0 0 0 1 1 0 0 1
302	0 1 0 0 0 0 1 0
303	0 1 0 1 1 0 0 1
304	0 1 0 0 0 0 0 1
305	0 0 1 0 1 0 0 1
...	...
...	...
...	...
940	0 0 0 0 0 0 1 1
941	0 0 0 0 0 0 0 0
942	0 0 0 0 0 0 1 0
943	0 0 0 0 0 0 0 0

# Ciclo de una Instrucción.

## Ciclo de una Instrucción

### Decode Instruction (Decodificación)



0001 Cargar en el acumulador el dato almacenado en la dirección de memoria.

Address

Data

300

0 1 0 0 0 0 0 0

301

0 0 0 1 1 0 0 1

302

0 1 0 0 0 0 1 0

303

0 1 0 1 1 0 0 1

304

0 1 0 0 0 0 0 1

305

0 0 1 0 1 0 0 1

...

...

...

...

...

...

940

0 0 0 0 0 0 1 1

941

0 0 0 0 0 0 0 0

942

0 0 0 0 0 0 1 0

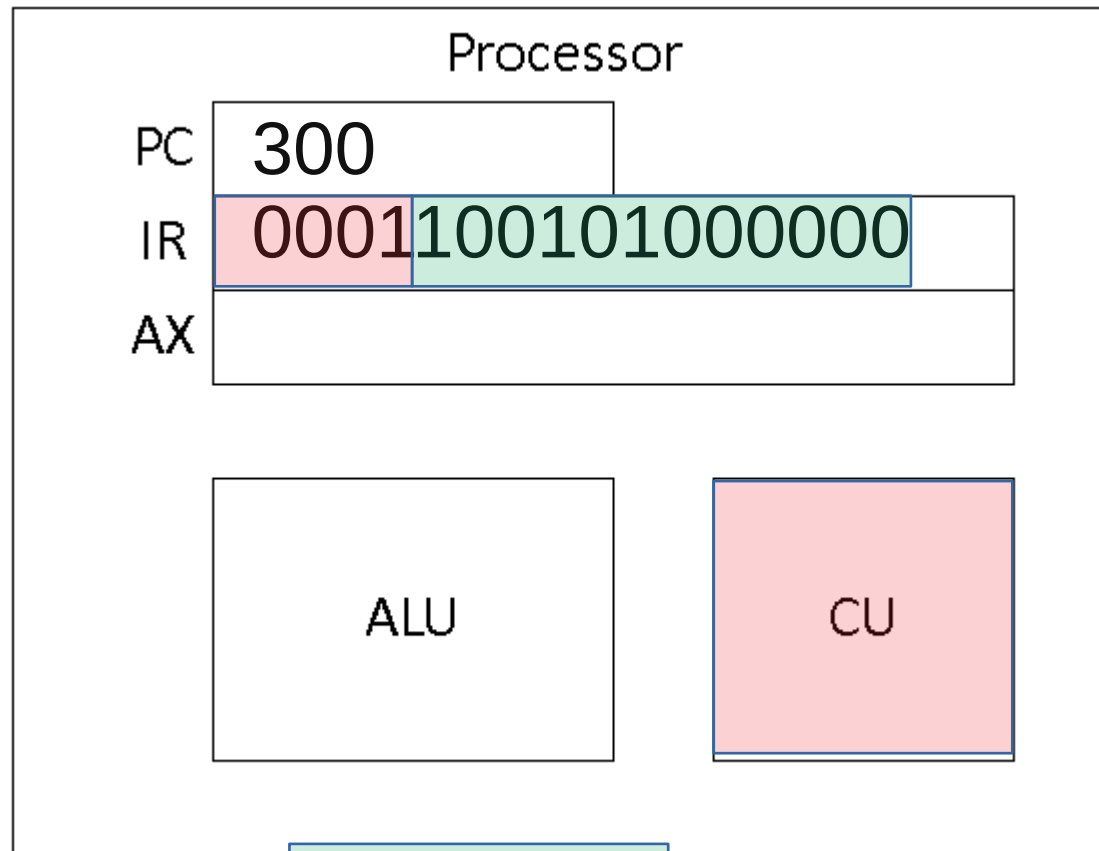
943

0 0 0 0 0 0 0 0

# Ciclo de una Instrucción.

## Ciclo de una Instrucción

### Read Data (Obtener dato)



MAR = 100101000000

MBR = 0000000000000011

Address

Data

300	0	1	0	0	0	0	0	0
301	0	0	0	1	1	0	0	1
302	0	1	0	0	0	0	1	0
303	0	1	0	1	1	0	0	1
304	0	1	0	0	0	0	0	1
305	0	0	1	0	1	0	0	1

...

...

...

...

...

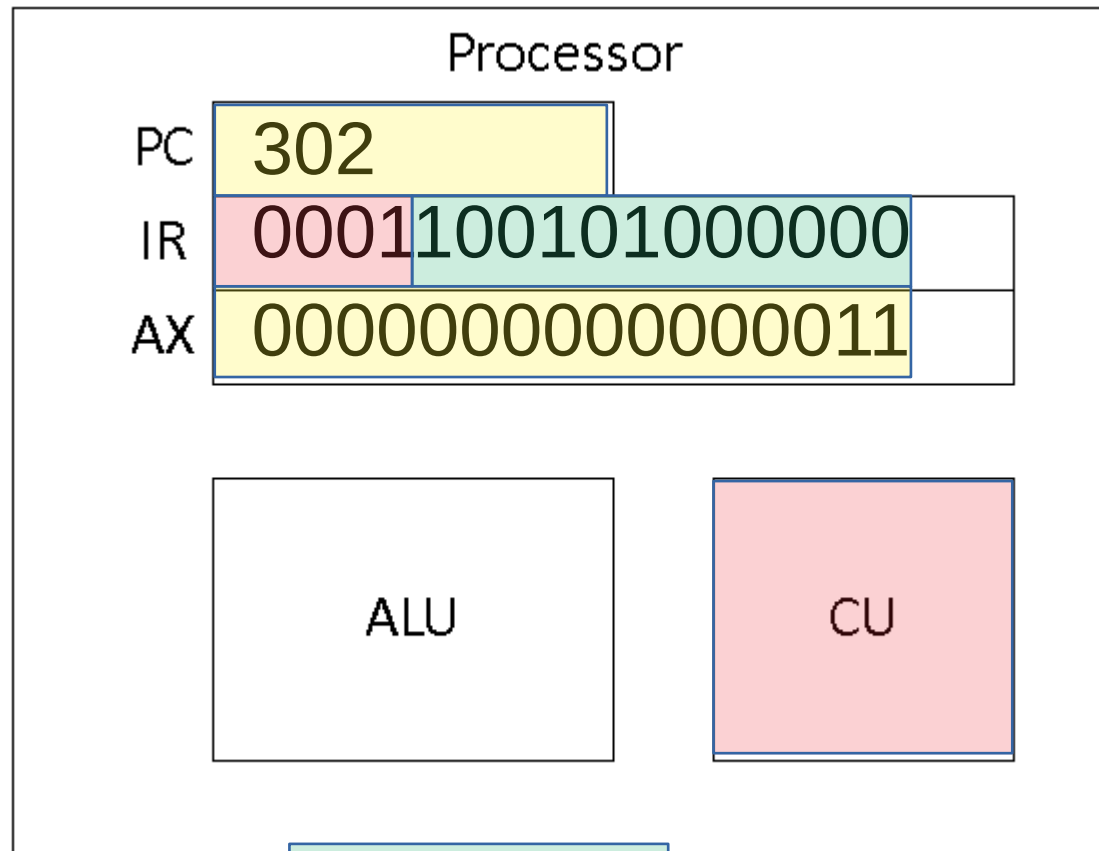
...

940	0	0	0	0	0	0	1	1
941	0	0	0	0	0	0	0	0
942	0	0	0	0	0	0	1	0
943	0	0	0	0	0	0	0	0

# Ciclo de una Instrucción.

## Ciclo de una Instrucción

### Execute Instruction (Ejecutar)



MAR = 100101000000

MBR = 000000000000000011

Address

Data

300	0	1	0	0	0	0	0	0
301	0	0	0	1	1	0	0	1
302	0	1	0	0	0	0	1	0
303	0	1	0	1	1	0	0	1
304	0	1	0	0	0	0	0	1
305	0	0	1	0	1	0	0	1

...

...

...

...

...

...

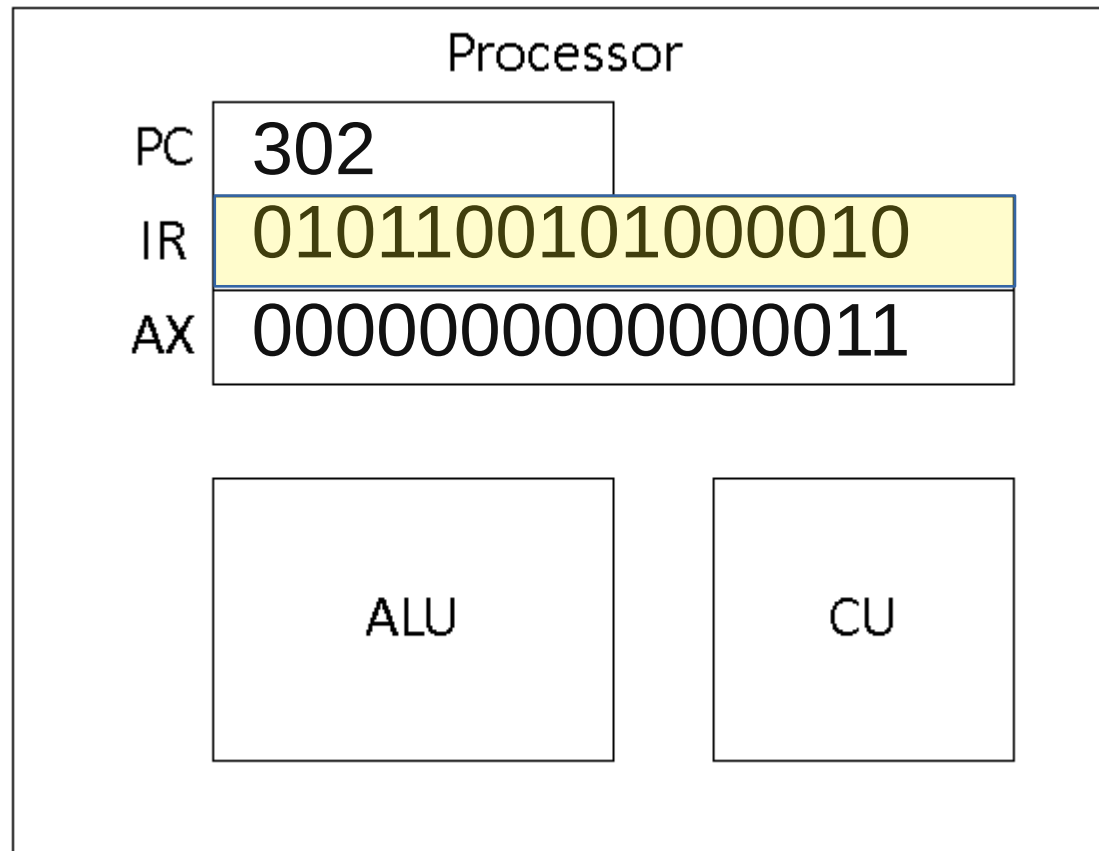
940	0	0	0	0	0	0	1	1
941	0	0	0	0	0	0	0	0
942	0	0	0	0	0	0	1	0
943	0	0	0	0	0	0	0	0



# Ciclo de una Instrucción.

## Ciclo de una Instrucción

### Fetch Instruction (Captación)

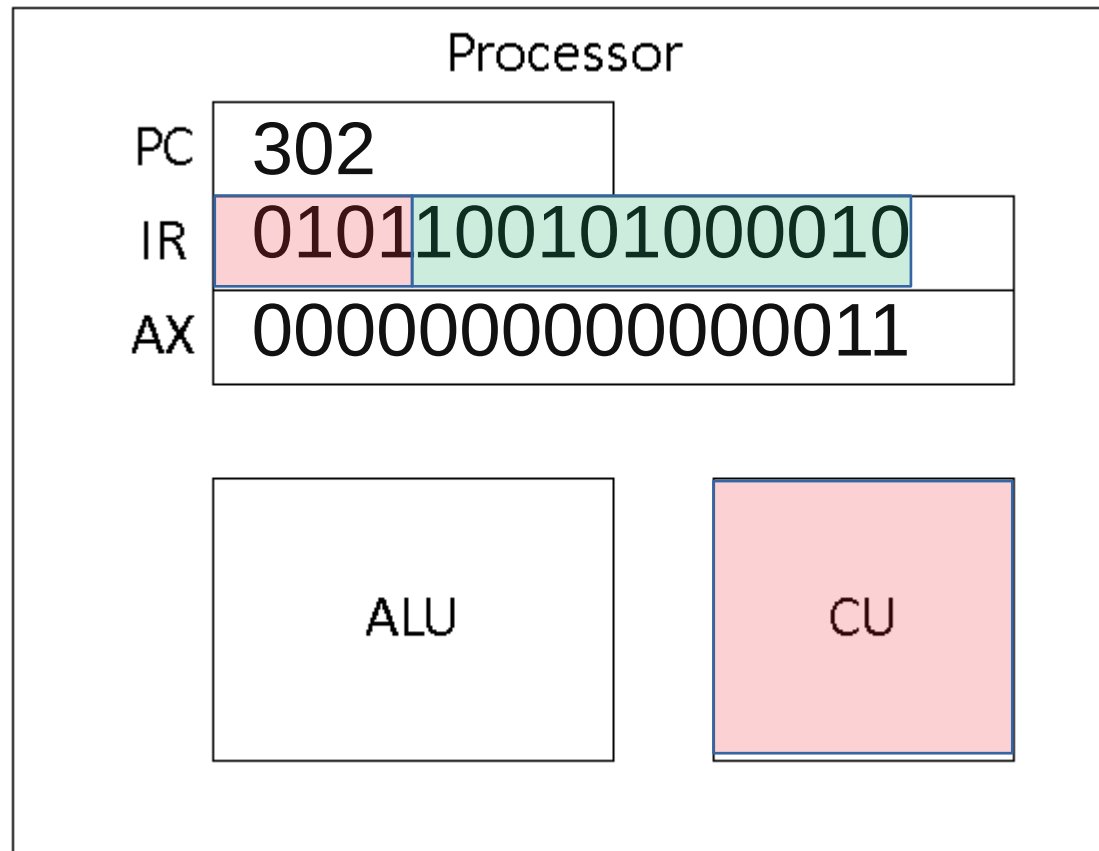


Address	Data
300	0 1 0 0 0 0 0 0
301	0 0 0 1 1 0 0 1
302	0 1 0 0 0 0 1 0
303	0 1 0 1 1 0 0 1
304	0 1 0 0 0 0 0 1
305	0 0 1 0 1 0 0 1
...	...
...	...
...	...
940	0 0 0 0 0 0 1 1
941	0 0 0 0 0 0 0 0
942	0 0 0 0 0 0 1 0
943	0 0 0 0 0 0 0 0

# Ciclo de una Instrucción.

## Ciclo de una Instrucción

### Decode Instruction (Decodifica



0101 Sumar el dato del acumulador con otro que está en memoria.

Address

Data

300	0	1	0	0	0	0	0	0
301	0	0	0	1	1	0	0	1
302	0	1	0	0	0	0	1	0
303	0	1	0	1	1	0	0	1
304	0	1	0	0	0	0	0	1
305	0	0	1	0	1	0	0	1

...

...

...

...

...

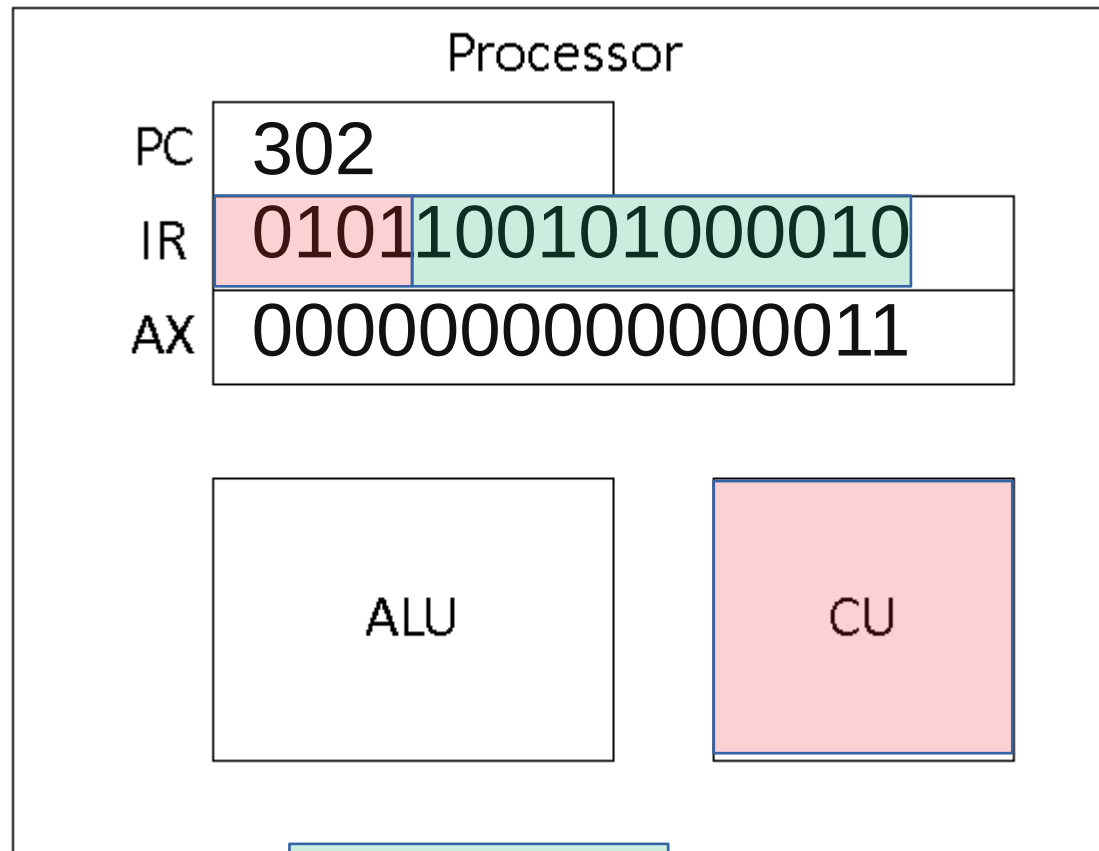
...

940	0	0	0	0	0	0	1	1
941	0	0	0	0	0	0	0	0
942	0	0	0	0	0	0	1	0
943	0	0	0	0	0	0	0	0

# Ciclo de una Instrucción.

## Ciclo de una Instrucción

### Read Data (Obtener dato)



MAR = 100101000010

MBR = 0000000000000010

Address

Data

300	0	1	0	0	0	0	0	0
301	0	0	0	1	1	0	0	1
302	0	1	0	0	0	0	1	0
303	0	1	0	1	1	0	0	1
304	0	1	0	0	0	0	0	1
305	0	0	1	0	1	0	0	1

...

...

...

...

...

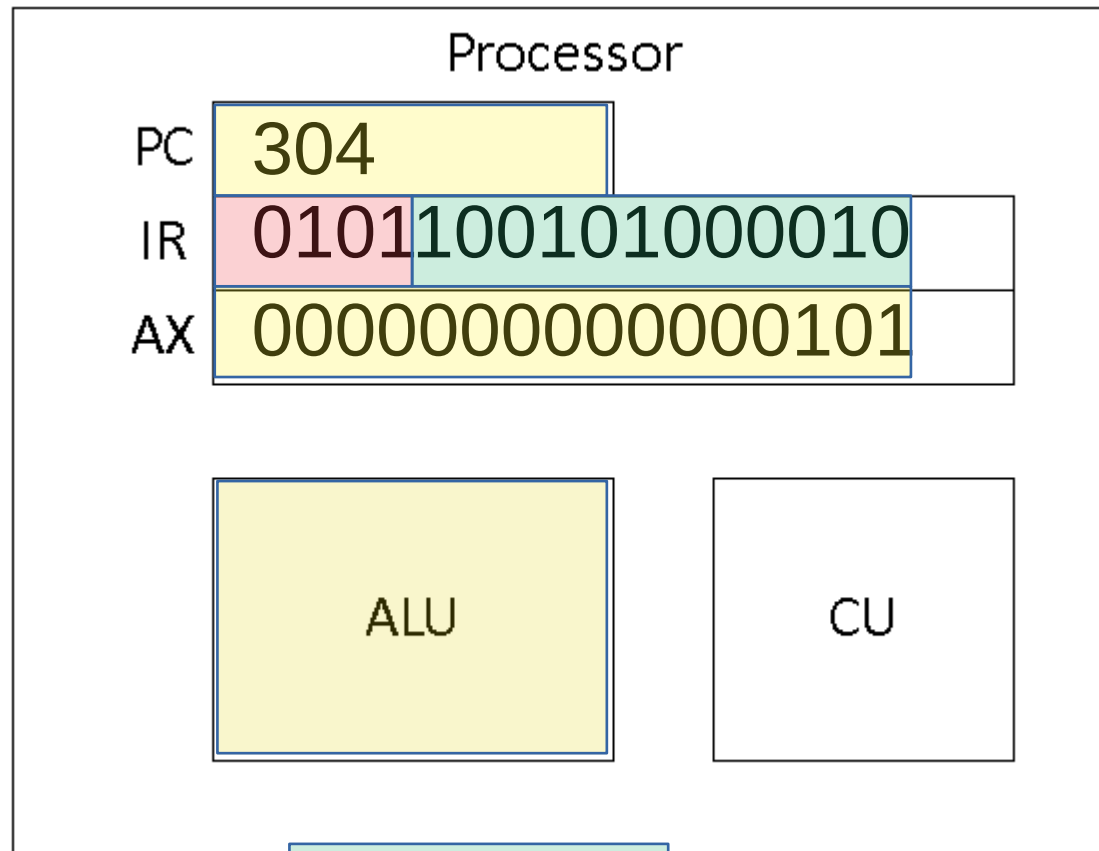
...

940	0	0	0	0	0	0	1	1
941	0	0	0	0	0	0	0	0
942	0	0	0	0	0	0	1	0
943	0	0	0	0	0	0	0	0

# Ciclo de una Instrucción.

## Ciclo de una Instrucción

### Execute Instruction (Ejecutar)



MAR = 100101000010

MBR = 0000000000000010

Address

Data

300	0	1	0	0	0	0	0	0
301	0	0	0	1	1	0	0	1
302	0	1	0	0	0	0	1	0
303	0	1	0	1	1	0	0	1
304	0	1	0	0	0	0	0	1
305	0	0	1	0	1	0	0	1

...

...

...

...

...

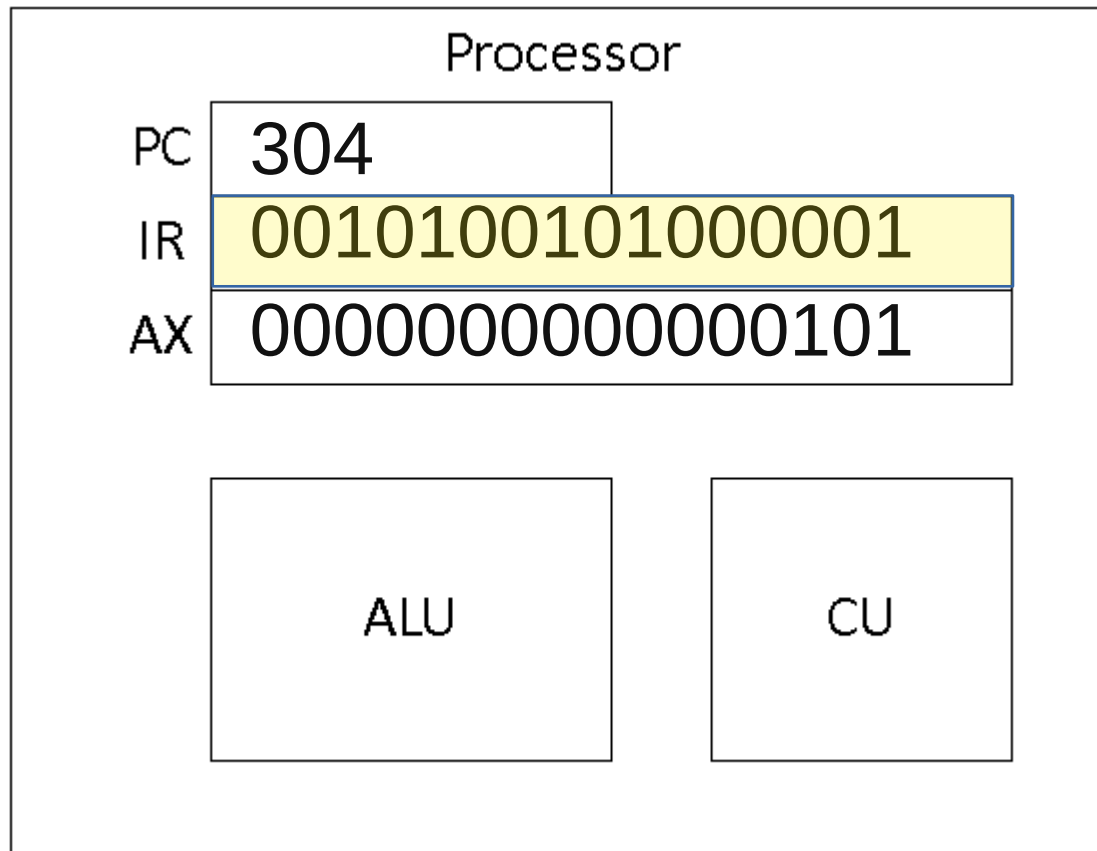
...

940	0	0	0	0	0	0	1	1
941	0	0	0	0	0	0	0	0
942	0	0	0	0	0	0	1	0
943	0	0	0	0	0	0	0	0

# Ciclo de una Instrucción.

## Ciclo de una Instrucción

### Fetch Instruction (Captación)

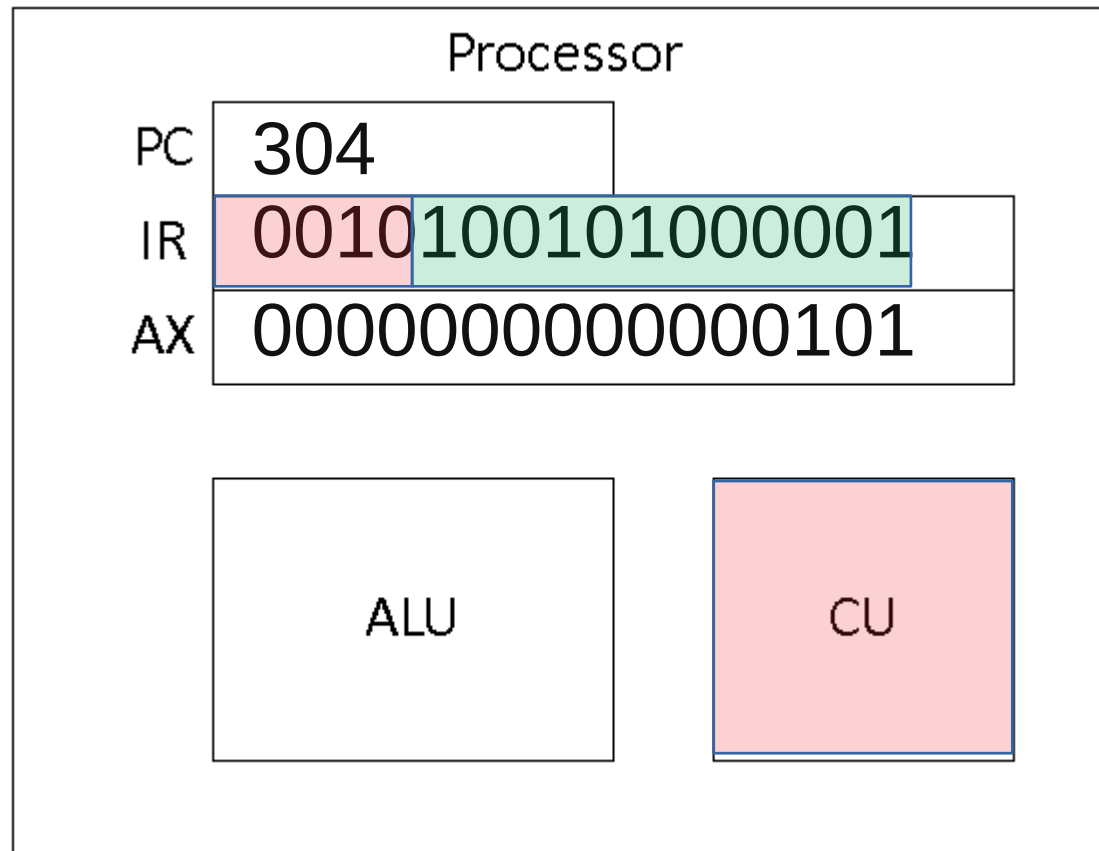


Address	Data
300	0 1 0 0 0 0 0 0
301	0 0 0 1 1 0 0 1
302	0 1 0 0 0 0 1 0
303	0 1 0 1 1 0 0 1
304	0 1 0 0 0 0 0 1
305	0 0 1 0 1 0 0 1
...	...
...	...
...	...
940	0 0 0 0 0 0 1 1
941	0 0 0 0 0 0 0 0
942	0 0 0 0 0 0 1 0
943	0 0 0 0 0 0 0 0

# Ciclo de una Instrucción.

## Ciclo de una Instrucción

### Decode Instruction (Decodifica



0010 Guardar el dato del acumulador en memoria.

Address

Data

300	0	1	0	0	0	0	0	0
301	0	0	0	1	1	0	0	1
302	0	1	0	0	0	0	1	0
303	0	1	0	1	1	0	0	1
304	0	1	0	0	0	0	0	1
305	0	0	1	0	1	0	0	1

...

...

...

...

...

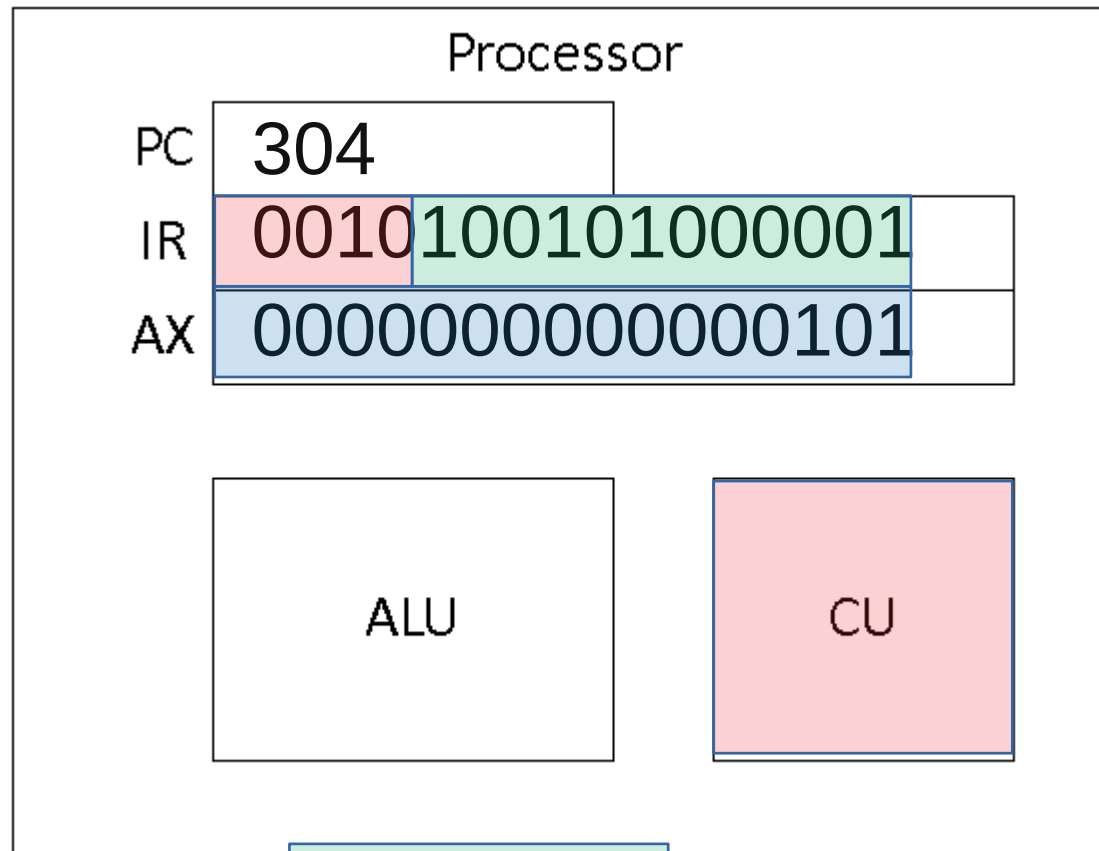
...

940	0	0	0	0	0	0	1	1
941	0	0	0	0	0	0	0	0
942	0	0	0	0	0	0	1	0
943	0	0	0	0	0	0	0	0

# Ciclo de una Instrucción.

## Ciclo de una Instrucción

### Read Data (Obtener dato)



MAR = 100101000001

MBR = 0000000000000000101

Address

Data

300	0	1	0	0	0	0	0	0
301	0	0	0	1	1	0	0	1
302	0	1	0	0	0	0	1	0
303	0	1	0	1	1	0	0	1
304	0	1	0	0	0	0	0	1
305	0	0	1	0	1	0	0	1

...

...

...

...

...

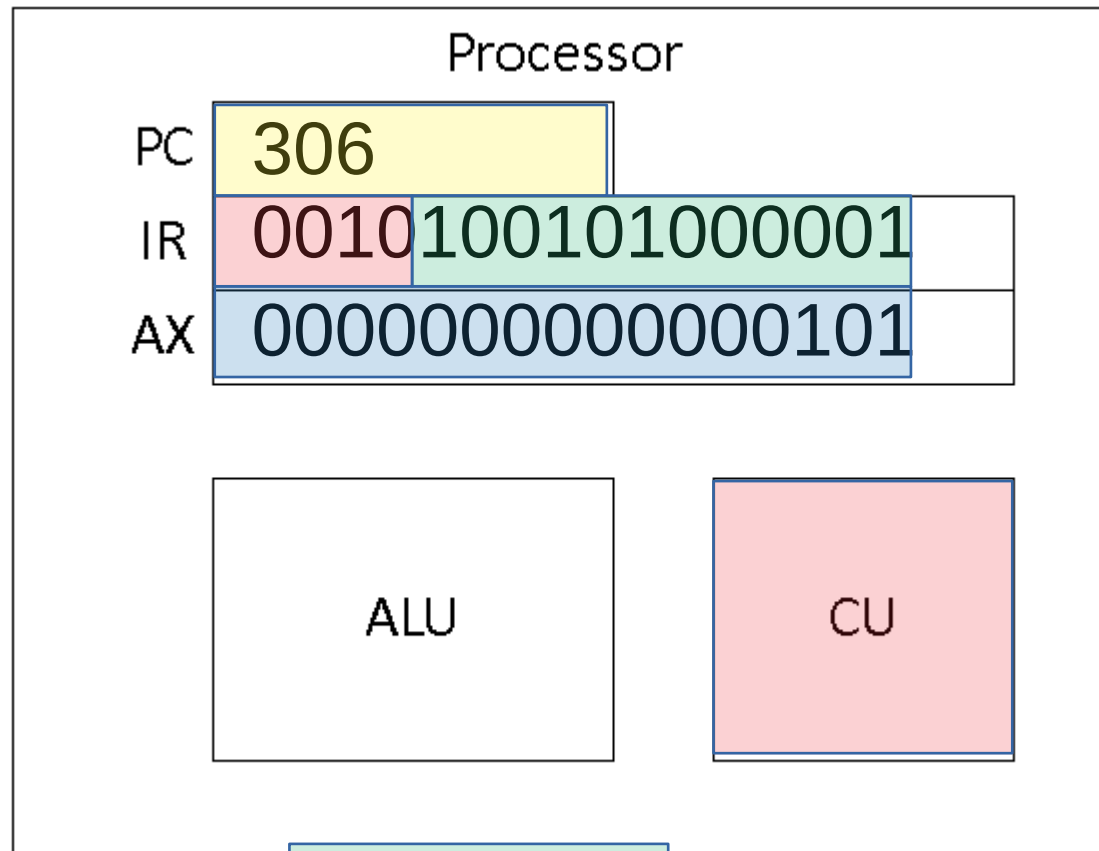
...

940	0	0	0	0	0	0	1	1
941	0	0	0	0	0	0	0	0
942	0	0	0	0	0	0	1	0
943	0	0	0	0	0	0	0	0

# Ciclo de una Instrucción.

## Ciclo de una Instrucción

### Execute Instruction (Ejecutar)



Address	Data
300	0 1 0 0 0 0 0 0
301	0 0 0 1 1 0 0 1
302	0 1 0 0 0 0 1 0
303	0 1 0 1 1 0 0 1
304	0 1 0 0 0 0 0 1
305	0 0 1 0 1 0 0 1
...	...
...	...
...	...
940	0 0 0 0 0 0 1 1
941	0 0 0 0 0 1 0 1
942	0 0 0 0 0 0 0 0
943	0 0 0 0 0 0 0 0