

# Organización de Computadoras

CURSO 2024

TURNO RECURSANTES

CLASE 5 – CICLO DE INSTRUCCIÓN

# Resumen de clase 5

- Ciclo de Instrucción simple
- Detalle del ciclo de instrucción simple
- Esquema básico de la CPU
- Análisis de la ejecución de una instrucción a nivel de programación
- Ciclo de instrucción detallado
- Ejemplo de análisis de ejecución de una instrucción

# Ciclo de Instrucción

## Ejecución de programas

- Un programa está compuesto de instrucciones.
- Las instrucciones están almacenadas en memoria.
- La CPU necesita traer ordenadamente las instrucciones desde la memoria hasta la CPU.
- Para resolver esa acción, la CPU procesa las instrucciones mediante una secuencia ordenada que consiste en:
  - Captar (leer) una instrucción
  - Resolver (ejecutar) la acción indicada por la instrucción

# Ciclo de Instrucción

- En base a esas 2 acciones fundamentales, podemos descomponer (modelizar) el procesamiento de instrucciones en dos etapas:
  - Búsqueda de la instrucción: lectura desde memoria
  - Ejecución de la instrucción: procesamiento del dato
- La búsqueda de la instrucción es una operación típica de todas las instrucciones, aunque no necesariamente idéntica (es decir, puede cambiar de una instrucción a otra)
- La ejecución es típicamente distinta para cada instrucción, incluso puede implicar varias operaciones.

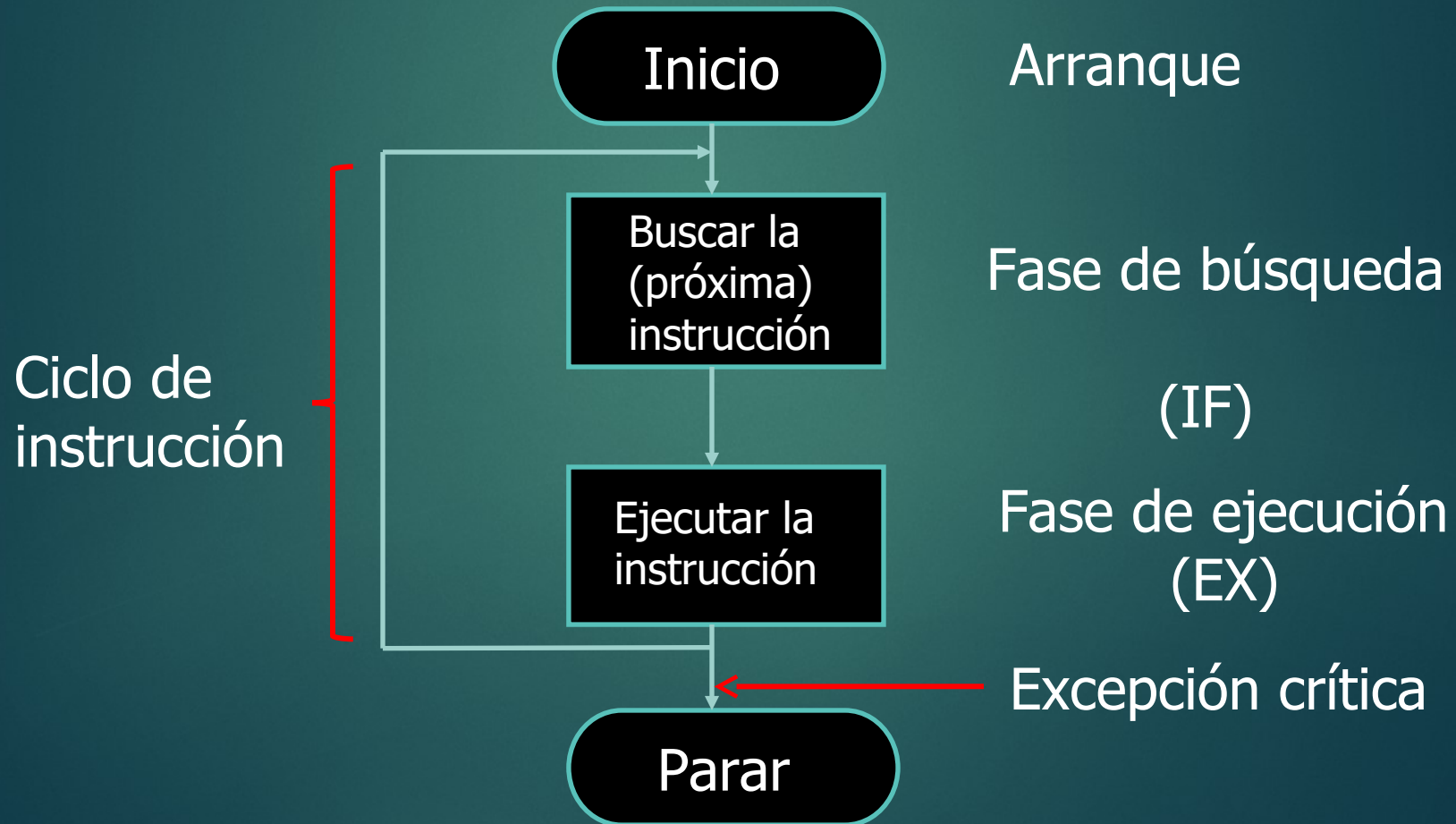


# Ciclo de Instrucción simple

- El proceso requerido para resolver una instrucción se llama ciclo de instrucción.
- En principio se puede considerar que el ciclo de instrucción consta de 2 fases:
  - Fase de búsqueda de la instrucción
  - Fase de ejecución de la instrucción
- El proceso de ejecución de instrucciones es ininterrumpido.
- Solo se detiene en casos excepcionales, por ejemplo:
  - cuando se desenergiza la máquina
  - hay un error muy grave
  - instrucción especiales

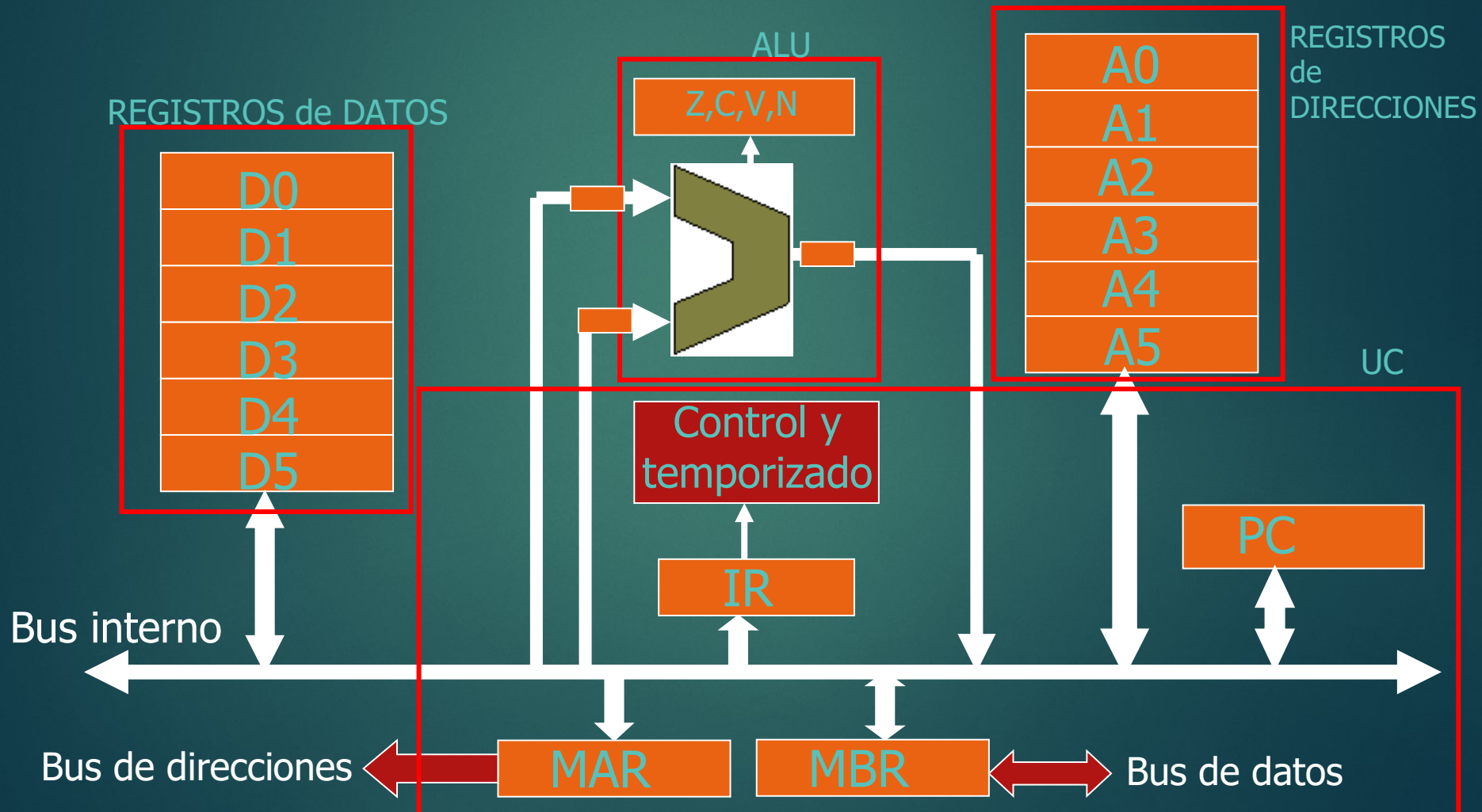
# Ciclo de Instrucción simple (modelo secuencial)

- El ciclo de instrucción puede interpretarse como un lazo de ejecución interna de la CPU, infinito, del tipo:



# CPU

Recordemos que internamente la CPU consta de las siguientes unidades funcionales interconectadas entre sí de la siguiente manera.



# Detalle del ciclo de instrucción simple

8

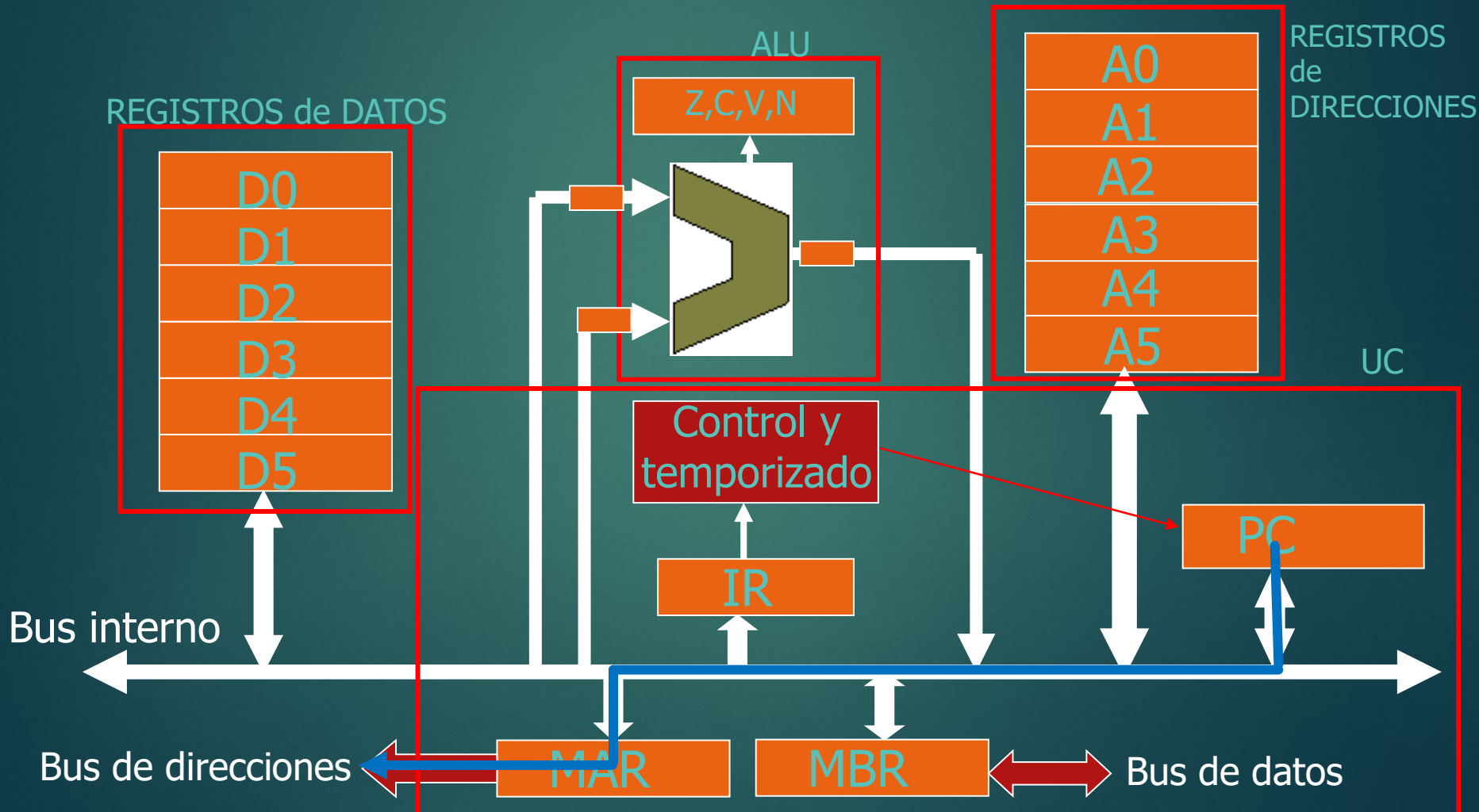
## Fase de búsqueda: el Contador de Programa PC

- Cada ciclo de instrucción comienza con la CPU buscando una instrucción (“la próxima instrucción”) en memoria.
- El proceso de buscar la instrucción se resuelve mediante el uso de un registro dentro de la CPU llamado Contador de programa (PC).
- El PC contiene siempre la dirección de memoria de la próxima instrucción a ejecutar.
- La CPU usa el PC para acceder a la memoria y leer la instrucción.

An orange rectangular box with the letters "PC" in white, representing the Program Counter register.



# Búsqueda de la instrucción



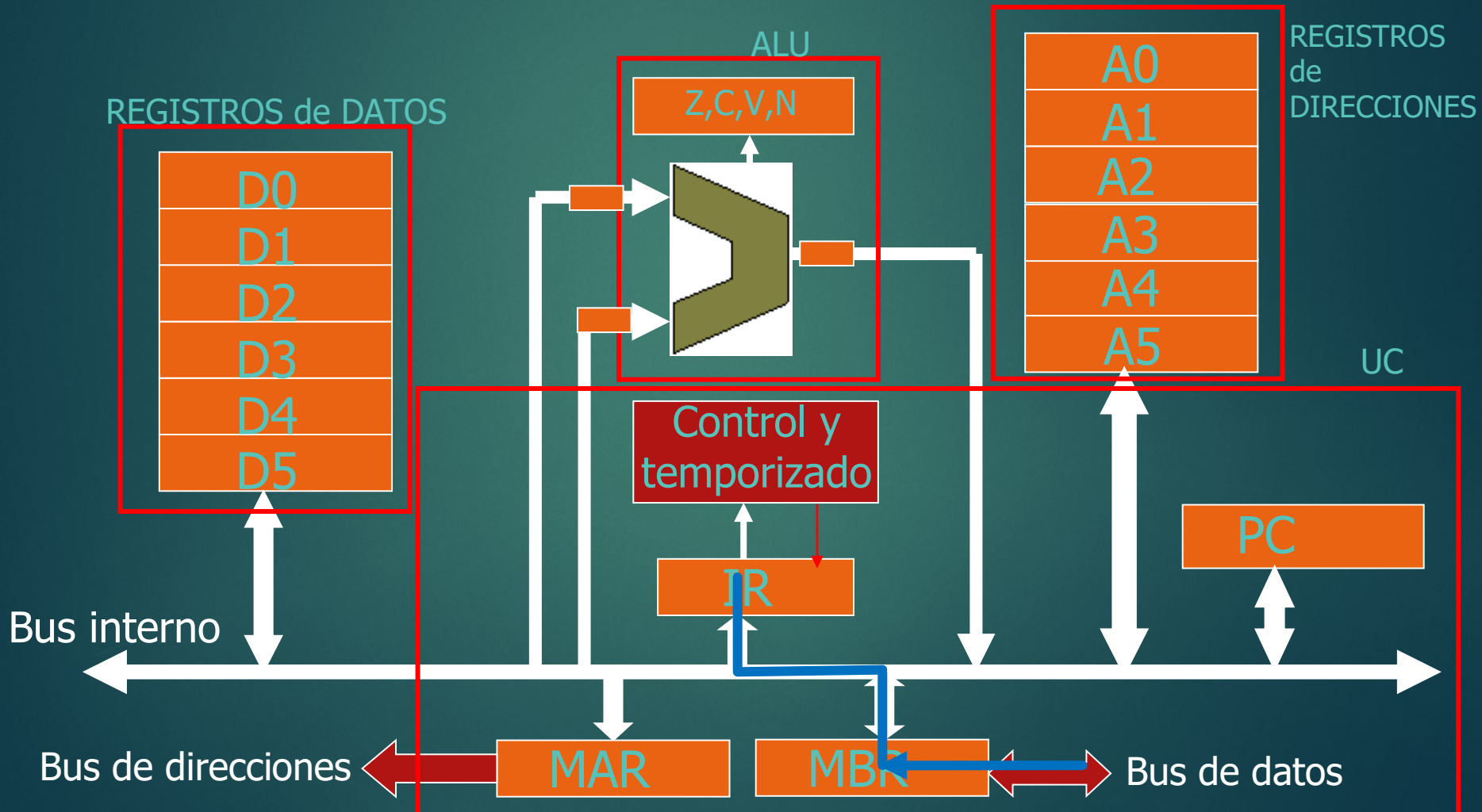
# Detalle del ciclo de instrucción simple

10

## Fase de búsqueda: captura de la instrucción

- La dirección de memoria, contenida en el PC, con la que lee la instrucción, se transfiere a través del registro de direcciones a memoria MAR al bus de direcciones.
- La instrucción leída desde memoria se copia temporalmente en el registro de datos a memoria MBR.
- La instrucción contenida temporalmente en el MBR es transferida a un registro especial interno, donde queda almacenada hasta completar la instrucción.
- Este registro se denomina Registro de Instrucción (IR).

# Captura de la instrucción



# Detalle del ciclo de instrucción simple

12

## Fase de búsqueda: el Registro de Instrucción IR

➤ El registro especial donde queda almacenada la instrucción buscada (leída de la memoria) se llama Registro de Instrucción (IR).



IR

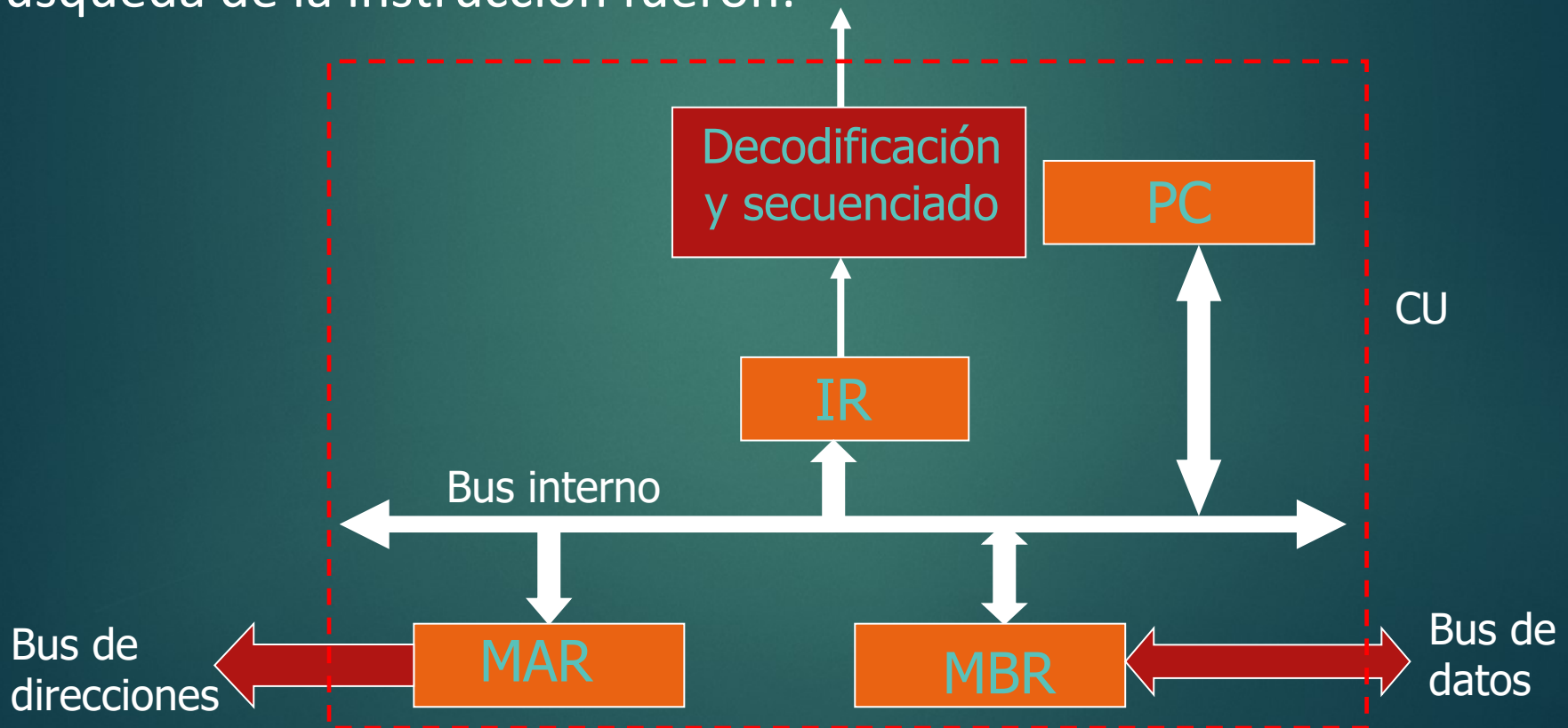
➤ La instrucción está en la forma de un código binario que contiene toda la información requerida por la CPU para poder resolverla.

➤ La Unidad de Control (CU) decodifica (“interpreta”) los bits de la instrucción y lleva a cabo el resto de las acciones requeridas por la instrucción.

# Detalle del ciclo de instrucción simple

## Fase de búsqueda: la Unidad de Control (CU)

➤ Los elementos principales de la CU que intervinieron en la fase de búsqueda de la instrucción fueron:





# Detalle del ciclo de instrucción simple

14

## Fase de ejecución:

- Luego de completada la fase de búsqueda de la instrucción, la CU procede a ejecutarla (resolverla).
- Las acciones para completar la instrucción dependen fuertemente del tipo de instrucción a ejecutar.
- Algunas de las acciones más importantes (y básicas) son:
  - Movimientos de datos:
    - Dentro de la CPU
    - Entre la CPU y Memoria
  - Procesamiento de datos (aritméticas, lógicas, etc.)
  - Entrada/salida

# Detalle del ciclo de instrucción simple

15

## Fase de ejecución: los Registros

- Consideremos una operación básica de suma de 2 operandos.
- Los operandos pueden estar en registros dentro de la CPU o en la memoria.
  - Si están dentro de la CPU, los datos se acceden a través del bus interno de la CPU.
  - Si están en la memoria, se requiere acceder a la memoria a través del bus externo de la CPU.

# Detalle del ciclo de instrucción

## simple

16

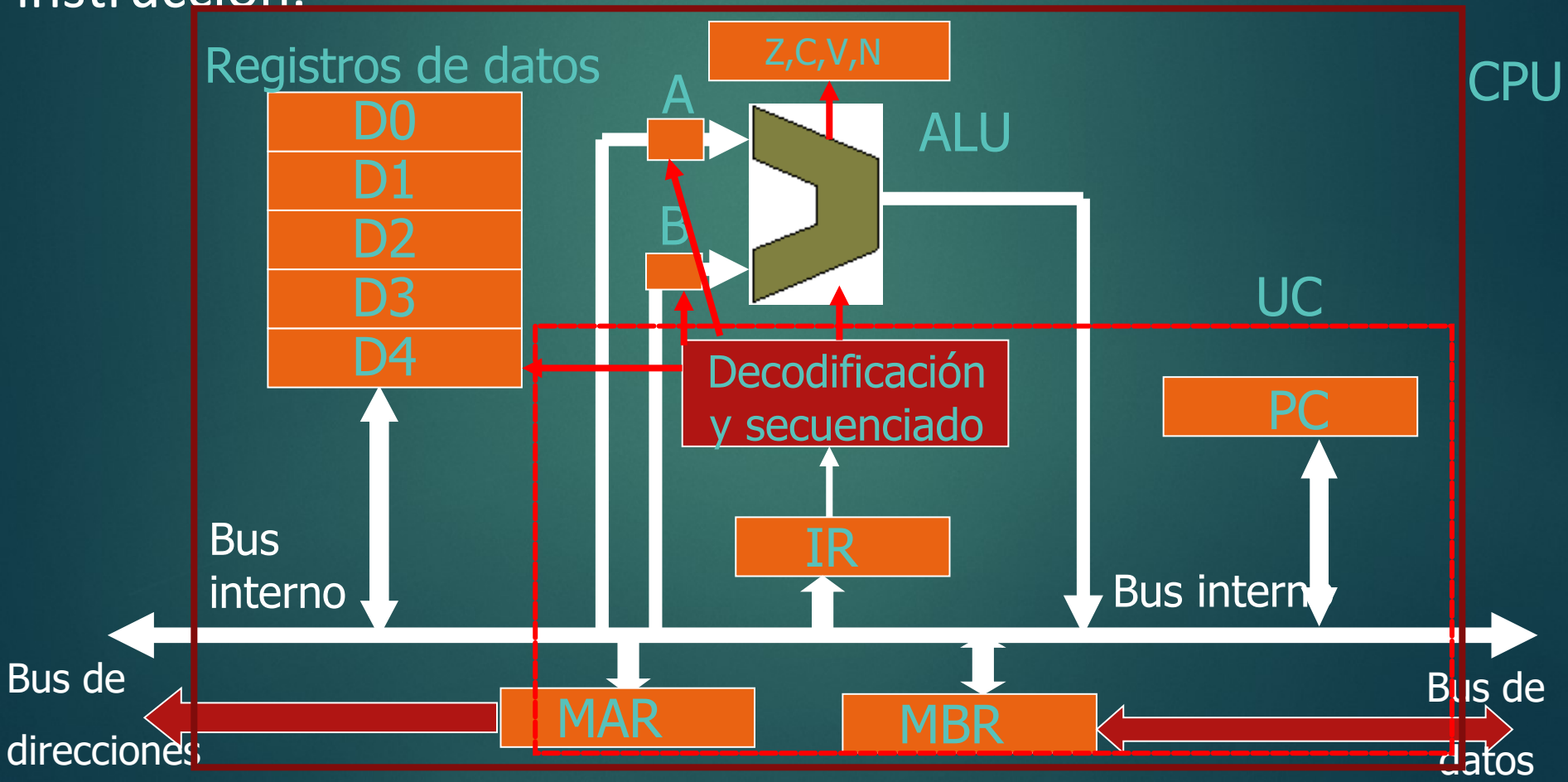
### Fase de ejecución: la ALU

- Como ya se vio previamente, hay una Unidad de cálculo (ALU) encargada de resolver operaciones aritméticas y lógicas.
- Típicamente la ALU dispone de 2 entradas de datos, 1 salida de resultado, n bits de control que “seleccionan” la operación a ejecutar, y registros binarios auxiliares para almacenar información referida al status de la ALU y al resultado de la operación (las “banderas de estado”).
- Comúnmente las entradas de datos provienen de registros (internos) de la CPU. A veces provienen de la memoria.
- Los registros pueden ser referenciados por el programador, como fuente ó destino (ó ambos) en una instrucción.

# Fase de ejecución con la ALU

## Fase de ejecución en la ALU

La UC controla los registros de datos y la ALU para ejecutar la instrucción.





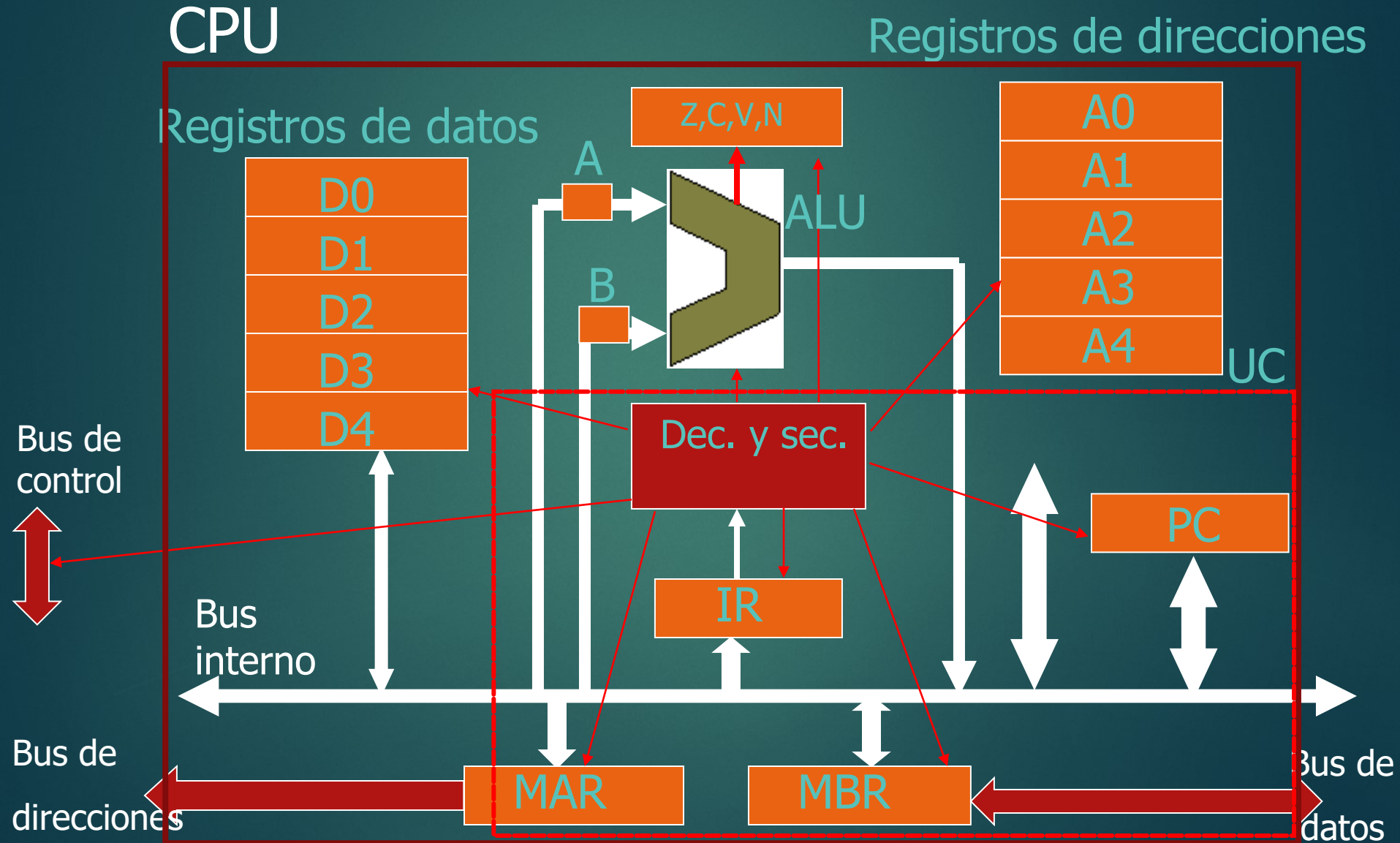
# Fase de ejecución con la ALU

18

- La Unidad de decodificación y secuenciado selecciona:
  - Los registros con los datos
  - Controla la ALU y sus registros asociados
  - El registro a donde se guarda el resultado.
- En el caso anterior se consideró el uso de los Registros de datos. Sin embargo, como se ha visto anteriormente, típicamente existen 2 categorías de registros:
  - Registros que manipulan datos: se los puede identificar como D0, D1, etc.
  - Registros que manipulan direcciones: se los puede identificar como A0, A1, etc.
  - La misma operación podría haber involucrado los registros de direcciones.



# Fase de ejecución con la ALU

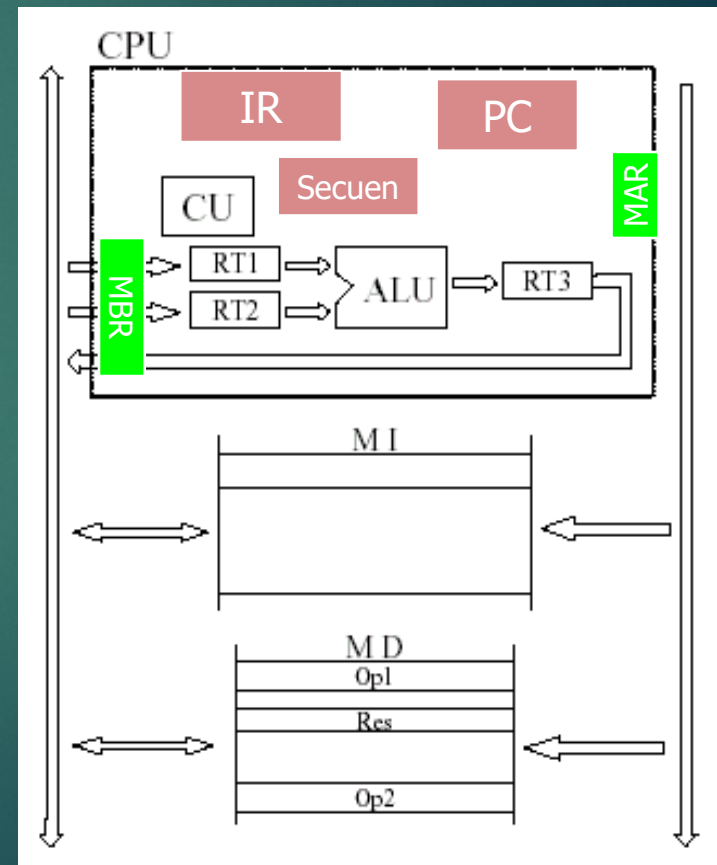


# Ejecución de una instrucción con datos en memoria

- A continuación, vamos a analizar el proceso de ejecución de una instrucción con los datos en memoria.
- Para el análisis siguiente, se considera una CPU con:
  - Registro de instrucción IR.
  - Contador de Programa PC.
  - Registros a memoria MBR y MAR.
  - ALU con 2 entradas de datos, con registros temporales en cada una de ellas, y 1 salida de resultado con registro temporal.
  - Un secuenciador que lleva la secuencia de acciones de una instrucción.
  - Una memoria de instrucciones MI y una memoria de datos MD.
  - Las memorias se acceden a través de los buses de datos y direcciones.

# Ejecución de una instrucción con datos en memoria

Consideremos una instrucción de suma 2 operandos almacenados en la memoria de datos, y el resultado se guarda también en la memoria de datos. El modelo de sistema a usar es:

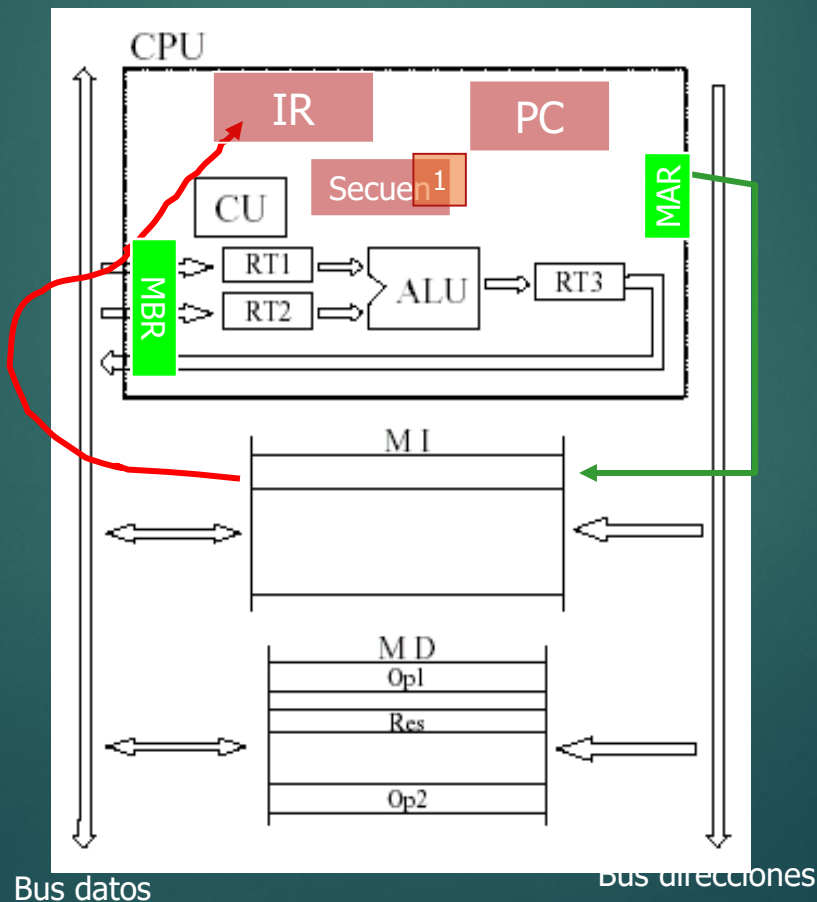


Bus datos

Bus direcciones

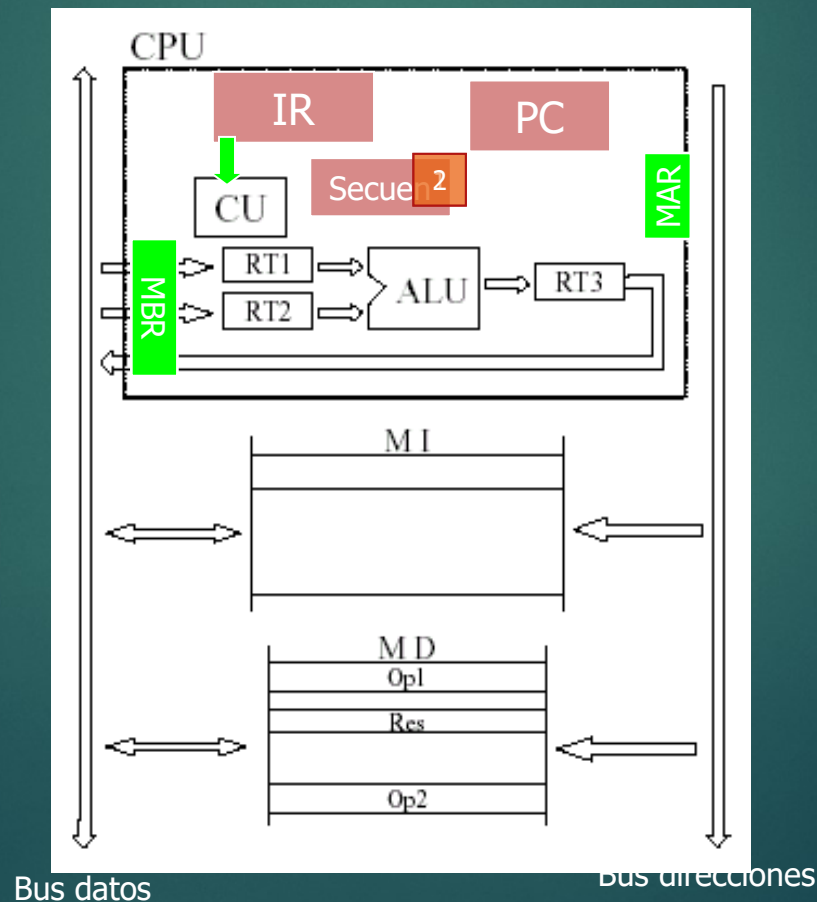
# Ejecución de una instrucción con datos en memoria

## Paso 1: buscar la instrucción



# Ejecución de una instrucción con datos en memoria

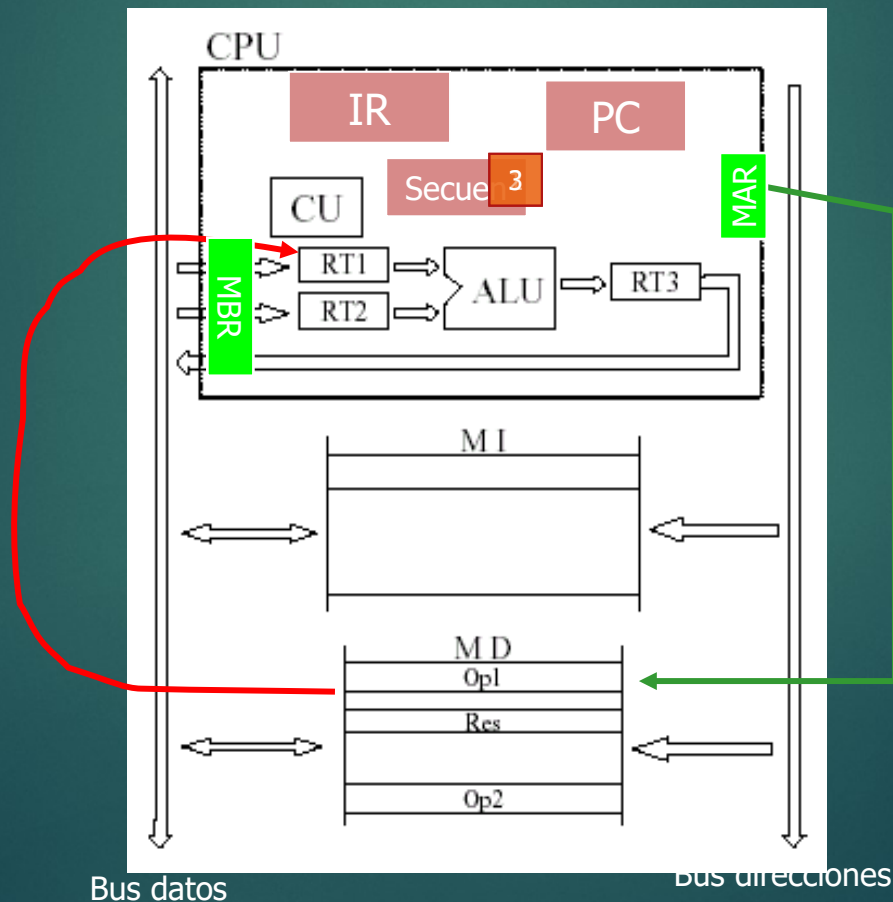
## Paso 2: decodificación





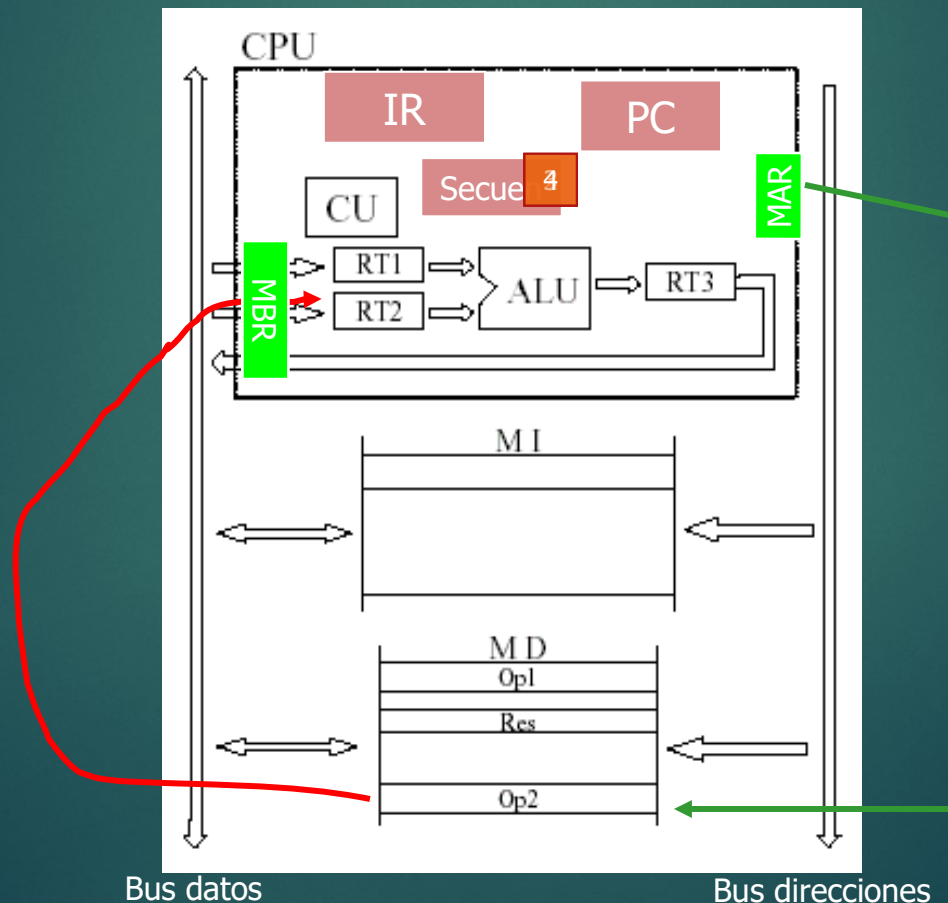
# Ejecución de una instrucción con datos en memoria

## Paso 3: búsqueda del operando 1 (Op1)



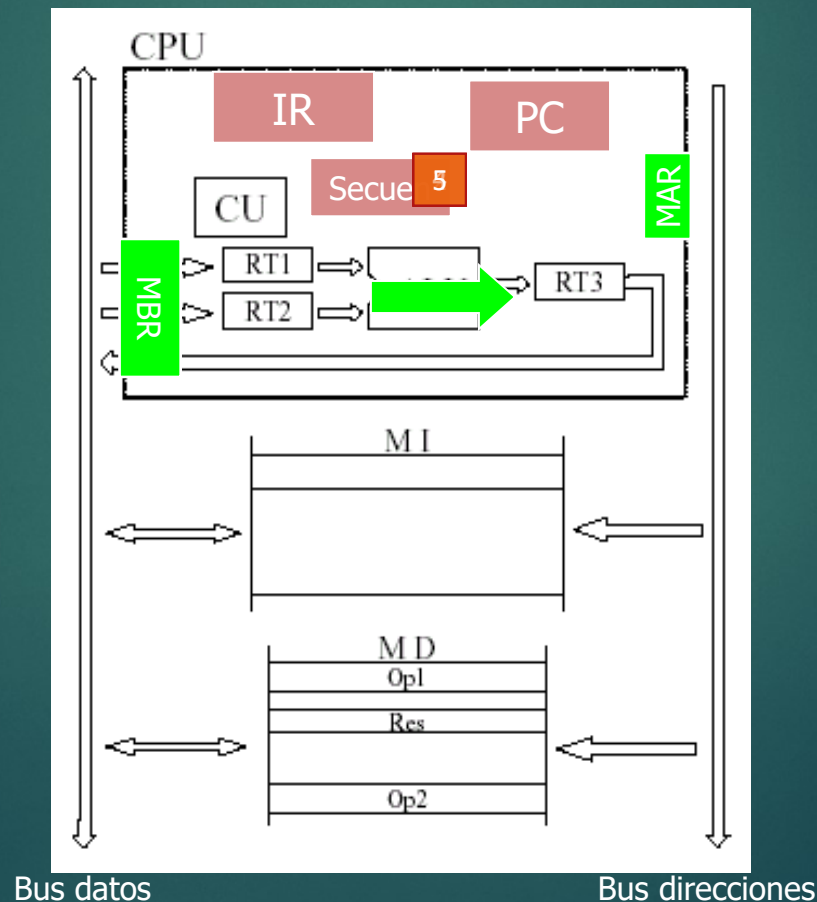
# Ejecución de una instrucción con datos en memoria

## Paso 4: búsqueda del operando 2 (Op2)



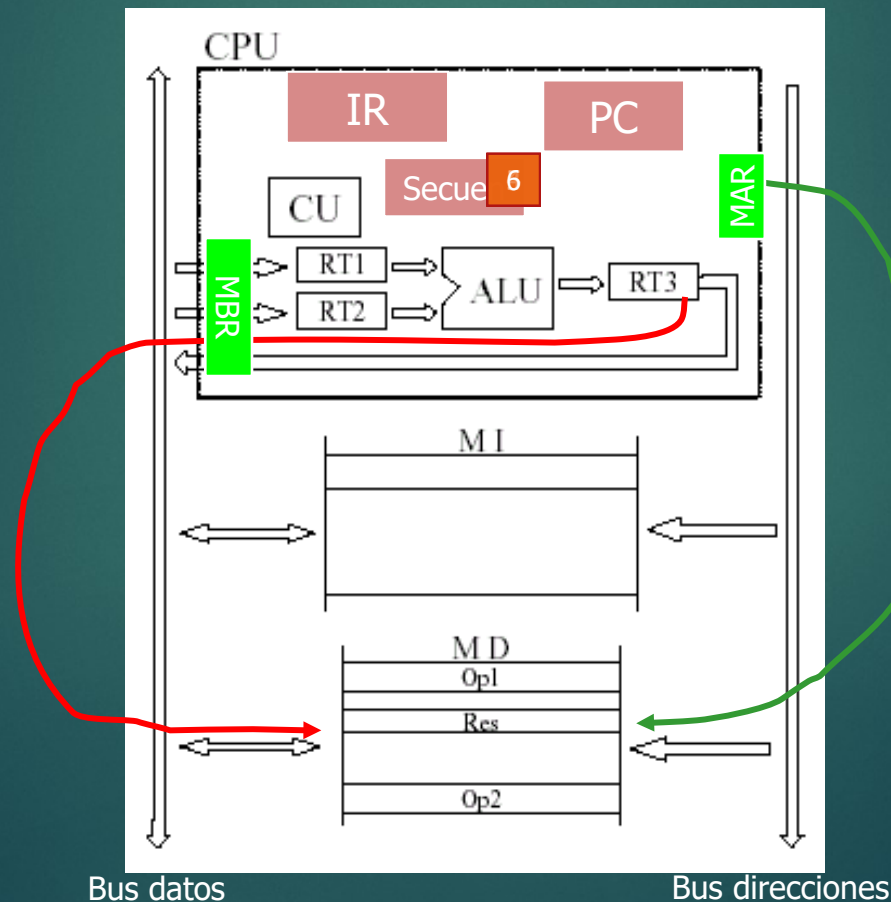
# Ejecución de una instrucción con datos en memoria

## Paso 5: ejecución de la suma



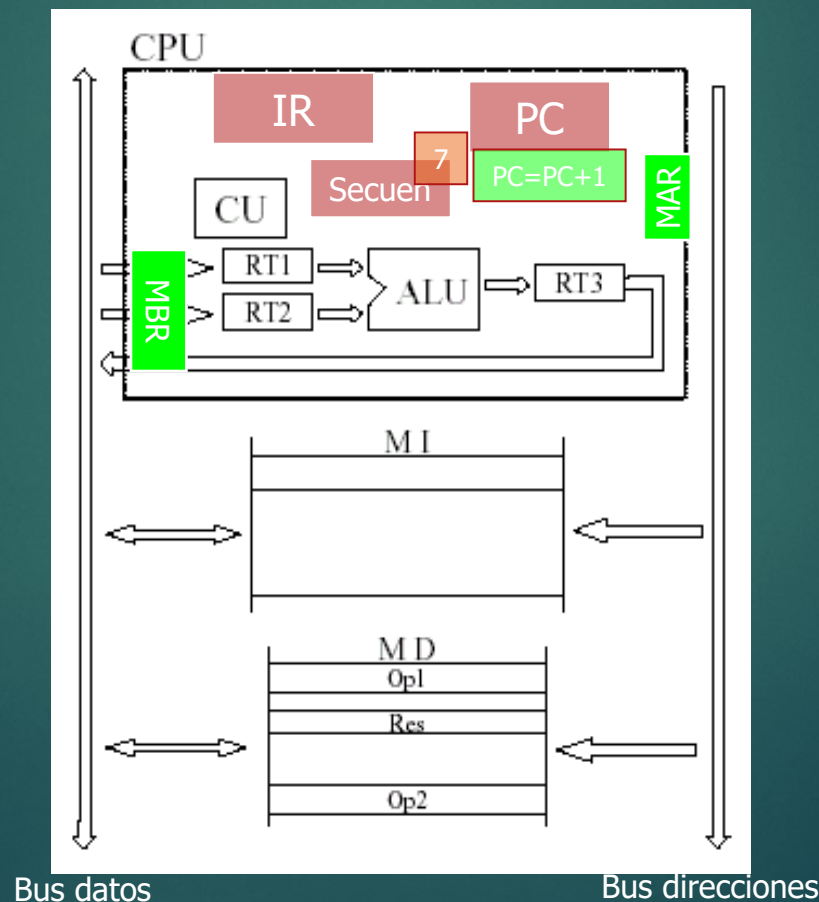
# Ejecución de una instrucción con datos en memoria

## Paso 6: almacenamiento del resultado (Res)



# Ejecución de una instrucción con datos en memoria

Paso 7: incremento del PC para apuntar a la próxima instrucción a ejecutar.





# Análisis de ejecución de una instrucción a nivel de programación

Para tener una mejor comprensión del ciclo de instrucción, vamos a analizar el proceso de ejecución de una secuencia de 3 instrucciones, ahora a nivel de programa.

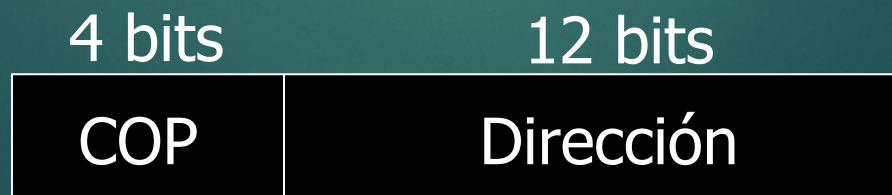
Para el análisis siguiente se considera un procesador con 3 registros internos:

- Contador de Programa PC
- Registro de Instrucciones IR
- Registro de datos D

 PC D IR

# Análisis de ejecución de una instrucción a nivel de programación

- Las instrucciones y los datos residen en memoria (común).
- La palabra de memoria tiene un tamaño de 16 bits.
- Para las palabras que contienen instrucciones, los 4 bits más significativos indican la operación a realizar, y los siguientes 12 bits indican una dirección de memoria.
- Observación: los 4 bits se conocen como “Código de operación” (COP).
- El formato binario de la instrucción es de la forma:



# Análisis de ejecución de una instrucción a nivel de programación

- La CPU puede ejecutar (“interpretar”) 3 instrucciones.
- Por ejemplo, los 4 bits del COP se pueden codificar, para identificar las 3 instrucciones, de la siguiente manera:
  - $0001_2=1$  cargar D desde la memoria
  - $0010_2=2$  almacenar D en memoria
  - $0101_2=5$  sumar D con un dato en memoria
- La 3 instrucciones van a tener el siguiente formato:

|                        | COP  | Dir            |
|------------------------|------|----------------|
| cargar D desde memoria | 0001 | XXXXXXXXXXXXXX |
| almacenar D en memoria | 0010 | XXXXXXXXXXXXXX |
| sumar D con memoria    | 0101 | XXXXXXXXXXXXXX |

# Análisis de ejecución de una instrucción a nivel de programación

- Vamos a considerar el siguiente programa, compuesto por 3 instrucciones (I1, I2, e I3):
  - I1: Cargar en el registro D el contenido de la posición de memoria  $940_{16}$
  - I2: Sumar el contenido de la posición de memoria  $941_{16}$  al registro D y guardar el resultado en D
  - I3: Almacenar el valor del registro D en la posición de memoria  $941_{16}$
- I1 está cargada en la dirección de memoria  $300_{16}$
- I2 está cargada en la dirección de memoria  $301_{16}$
- I3 está cargada en la dirección de memoria  $302_{16}$



# Análisis de ejecución de una instrucción a nivel de programación

El programa en binario quedaría:

| Inst. | Memoria |                |     | Dirección |
|-------|---------|----------------|-----|-----------|
| I1    | 0001    | 1001 0100 0000 | 300 |           |
| I2    | 0101    | 1001 0100 0001 | 301 |           |
| I3    | 0010    | 1001 0100 0001 | 302 |           |

Y en hexadecimal:

|    |                    |     |
|----|--------------------|-----|
| I1 | 1940 <sub>16</sub> | 300 |
| I2 | 5941 <sub>16</sub> | 301 |
| I3 | 2941 <sub>16</sub> | 302 |



# Análisis de ejecución de una instrucción a nivel de programación

En la dirección de memoria 940 está almacenado el número  $0003_{16}$ , y en la 941 está el número  $0002_{16}$ .

| Memoria             | Dirección |
|---------------------|-----------|
| 0000 0000 0000 0011 | 940       |
| 0000 0000 0000 0010 | 941       |

Y en hexadecimal:

|             |     |
|-------------|-----|
| $0003_{16}$ | 940 |
| $0002_{16}$ | 941 |

# Análisis de ejecución de una instrucción a nivel de programación

35

- En el comienzo del análisis se considera que el contador de programa (PC) contiene la dirección  $300_{16}$ .
- Para simplificar el análisis, no se consideran los registros temporales MAR y MBR.

# Análisis de ejecución

## Instrucción 1 – Paso 1

### Estado inicial

Instrucciones Memoria

|     |      |
|-----|------|
| 300 | 1940 |
| 301 | 5941 |
| 302 | 2941 |

Operandos

|     |      |
|-----|------|
| 940 | 0003 |
| 941 | 0002 |

Registros de la CPU

|      |    |
|------|----|
| 300  | PC |
| xxxx | D  |
| yyyy | IR |

# Análisis de ejecución

## Instrucción 1 – Paso 1

### I1 – PASO 1: Fase de búsqueda de la instrucción

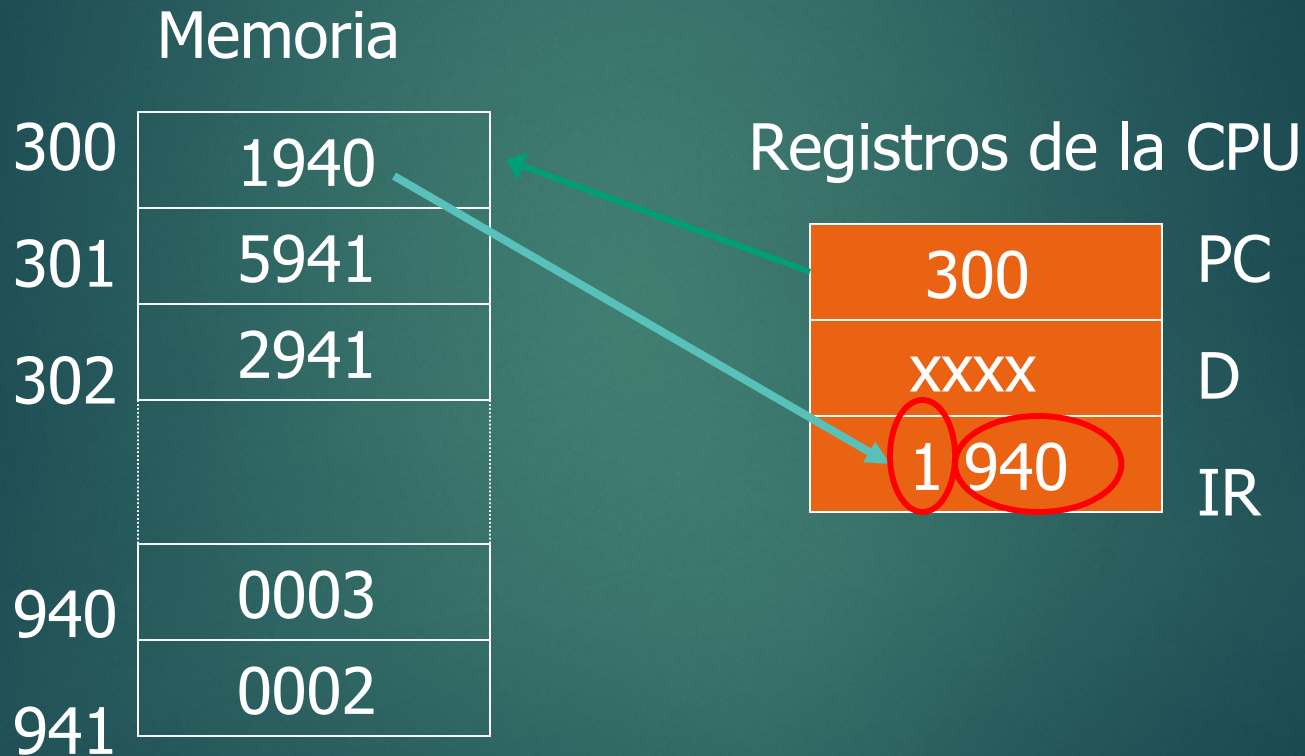
- La CPU manda al bus de direcciones el valor del PC.
- La memoria le devuelve su contenido (“lectura de instrucción”).
- La instrucción se almacena en el IR (se completa la fase de búsqueda de la instrucción).

Nota: Para este ejemplo, los primeros 4 bits en IR indican que el registro D se cargará con un dato proveniente de la dirección especificada en los restantes 12 bits de la instrucción. En este caso tal dirección es  $940_{16}$ .

# Análisis de ejecución

## Instrucción 1 – Paso 1

### I1 – PASO 1: Fase de búsqueda de la instrucción





# Análisis de ejecución

## Instrucción 1 – Paso 2

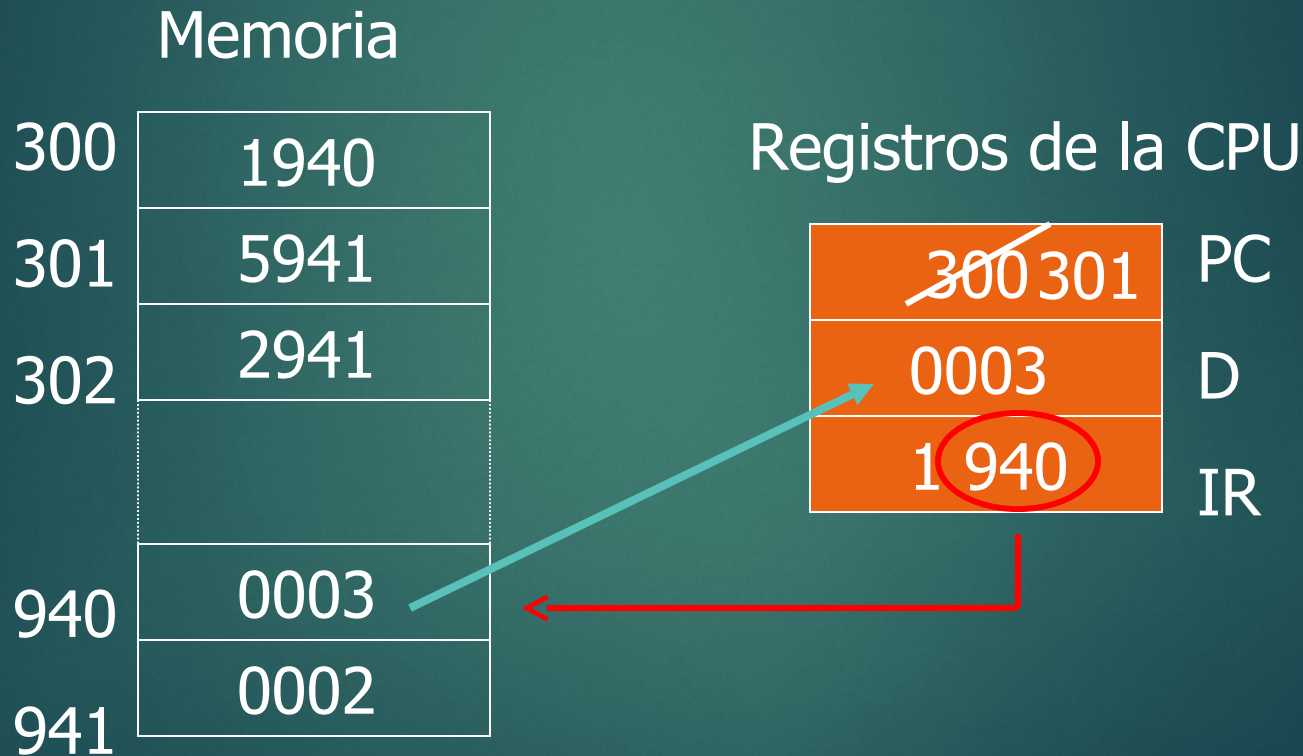
### I1 – PASO 2: Fase de ejecución de la instrucción

- Con la parte inferior del contenido del IR (940) se busca el primer operando en memoria.
- El contenido de la posición de memoria 940 se almacena en D. Dado que la posición de memoria 940 contiene 0003, se transfiere ese valor (0003) al registro D.
- El PC se incrementa en 1 (en algún momento) para contener la dirección de la próxima instrucción (301).

# Análisis de ejecución

## Instrucción 1 – Paso 2

### I1 – PASO 2: Fase de ejecución de la instrucción



# Análisis de ejecución

## Instrucción 2 – Paso 1

### I2 – PASO 1: Fase de búsqueda de la instrucción

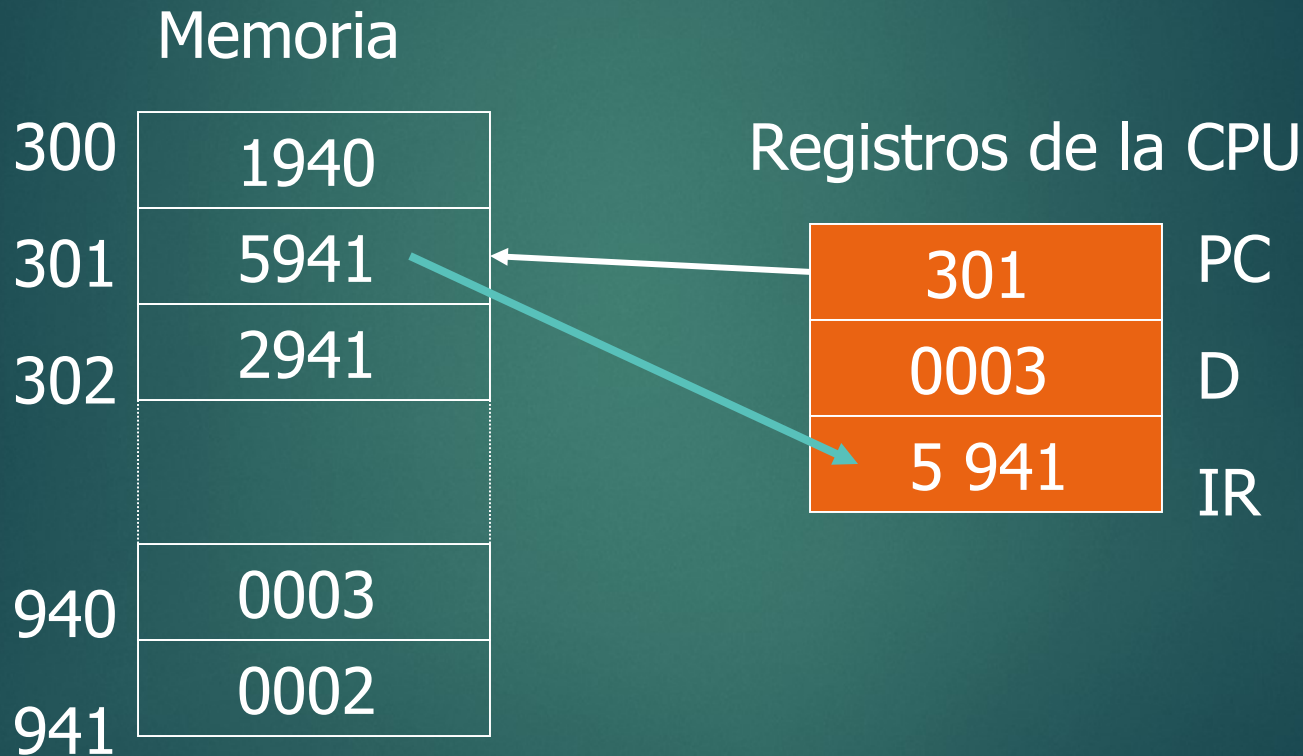
- Se repite la misma secuencia de búsqueda de la instrucción de la I1, solo que ahora el PC contiene 301, que corresponde a la dirección de memoria de la I2.
- La CPU manda al bus de direcciones el valor del PC.
- La memoria le devuelve su contenido.
- La instrucción se almacena en el IR.

Nota: Los primeros 4 bits ( $5_{16}$ ) en IR indican que se debe sumar el contenido de una dirección de memoria especificada (en este caso la dirección es  $941_{16}$ ) con el contenido del registro D y almacenar el resultado en el registro D.

# Análisis de ejecución

## Instrucción 2 – Paso 1

### I2 – PASO 1: Fase de búsqueda de la instrucción





# Análisis de ejecución

## Instrucción 2 – Paso 2

### I2 – PASO 2: Fase de ejecución de la instrucción

- Con la parte inferior del contenido del IR (941) se busca el segundo operando en memoria.
- El contenido de la posición de memoria 941 (0002) se suma al contenido del registro D (0003)
- El resultado:

$$0003 + 0002 = 0005$$

se guarda en el registro D.

- El PC se incrementa en 1 (en algún momento) para contener la dirección de la próxima instrucción (302).



# Análisis de ejecución

## Instrucción 2 – Paso 2

### I2 – PASO 2: Fase de ejecución de la instrucción



# Análisis de ejecución

## Instrucción 3 – Paso 1

### I3 – PASO 1: Fase de búsqueda de la instrucción

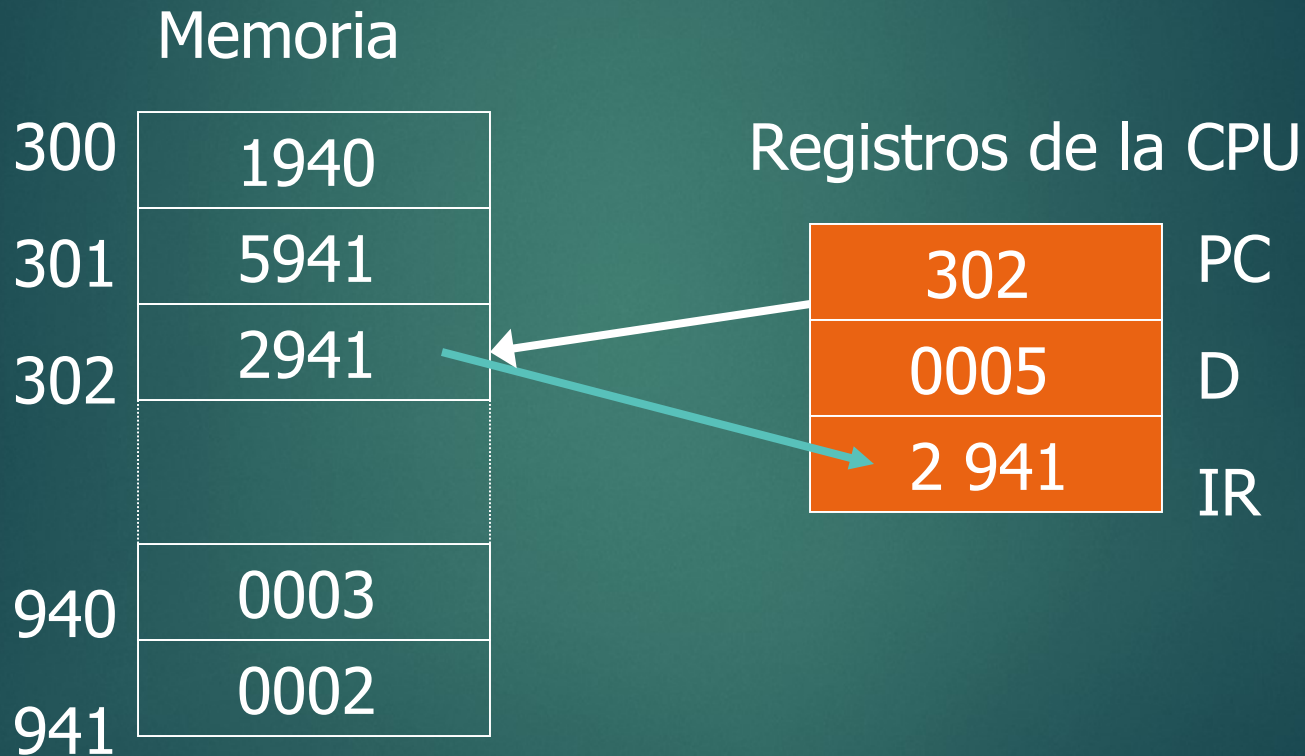
- Se repite la misma secuencia de búsqueda de las instrucciones I1 e I2, solo que ahora el PC contiene 302, que corresponde a la dirección de memoria de la I3.
- La CPU manda al bus de direcciones el valor del PC.
- La memoria le devuelve su contenido.
- La instrucción se almacena en el IR.

Nota: Los primeros 4 bits ( $2_{16}$ ) en IR indican que se debe almacenar el contenido del registro D en la dirección de memoria especificada en el resto de la instrucción (en este caso la dirección es  $941_{16}$ ).

# Análisis de ejecución

## Instrucción 3 – Paso 1

### I3 – PASO 1: Fase de búsqueda de la instrucción



# Análisis de ejecución

## Instrucción 3 – Paso 2

### I3 – PASO 2: Fase de ejecución de la instrucción

- El contenido del registro D se almacena en la posición de memoria contenida en la parte inferior del IR (941).
- Dado que D contiene 0005, en la posición de memoria 941 quedará almacenado ese valor (0005).
- El PC se incrementa en 1 (en algún momento) para contener la dirección de la próxima instrucción (303).

# Análisis de ejecución

## Instrucción 3 – Paso 2

### I3 – PASO 2: Fase de ejecución de la instrucción





# Ciclo de instrucción detallado

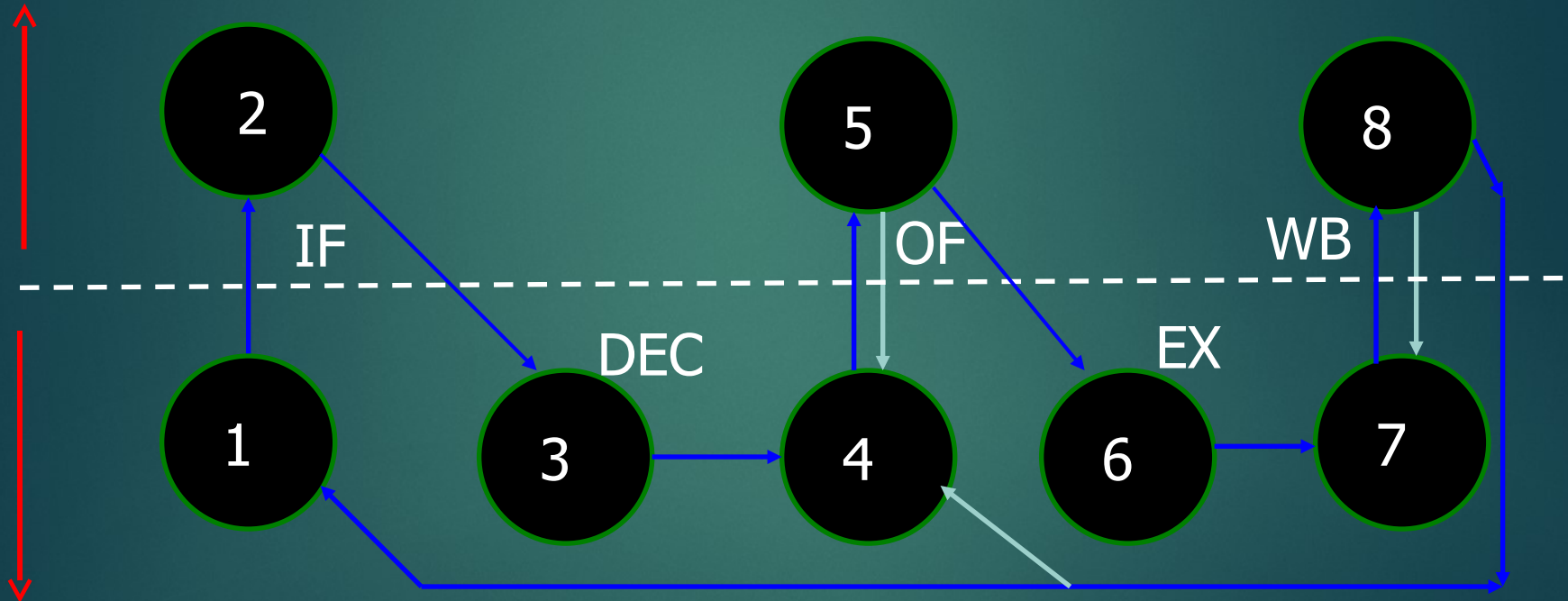
49

- El ciclo de instrucción simple contiene únicamente las 2 acciones básicas del proceso de ejecución de una instrucción:
  - Búsqueda de la instrucción
  - Ejecución de la instrucción
- Sin embargo, un modelo más detallado debería incluir otras acciones requeridas para la ejecución de la instrucción.
- Este modelo de ciclo de instrucción más detallado se muestra a continuación.

# Ciclo de instrucción detallado

## Modelo de ciclo de instrucción detallado

Acceso de CPU a memoria ó E/S



Operación interna de la CPU

# Ciclo de instrucción detallado

51

- El diagrama anterior corresponde a un modelo de ciclo de instrucción más detallado.
- Este diagrama representa un diagrama de estados de una máquina, donde:
  - Los círculos numerados del 1 al 8 son los posibles estados en los que puede estar la máquina, y
  - las flechas son transiciones entre estados.
- Los estados de la parte inferior (1,3,4,6,7) implican acciones dentro de la CPU.
- Los estados de la parte superior (2,5,8) representan acciones fuera de la CPU (típicamente memoria).

# Descripción del ciclo de instrucción detallado

## Fase de búsqueda de la instrucción - Estados 1 y 2

Los estados 1 y 2 se conocen normalmente como Fase de búsqueda de la instrucción (en inglés IF, Instruction Fetch).

- Estado 1 - cálculo de la dirección de la próxima instrucción: El procesador “calcula” la dirección donde reside la próxima instrucción a ejecutar.
- Estado 2 – búsqueda de la instrucción: el PC se envía al bus de direcciones para leer la memoria. La CPU recibe la instrucción leída por el bus de datos.



# Descripción del ciclo de instrucción detallado

## Fase de decodificación de la instrucción - Estado 3

El estado 3 se conoce normalmente como Fase de decodificación de la instrucción (en inglés DEC, Decode).

➤ Estado 3 – decodificación: La instrucción se carga en el IR. La Unidad de Control (CU) interpreta (decodifica) los bits de la instrucción para continuar el proceso de ejecución



# Descripción del ciclo de instrucción detallado

## Fase de búsqueda de los operandos - Estados 4 y 5

Los estados 4 y 5 se conocen normalmente como Fase de búsqueda de los operandos (en inglés OF, Operand fetch). Algunas instrucciones pueden requerir 1 operando, y otras 2 (o más). El o los operandos pueden estar en registros o en memoria.

En el diagrama de estados, los estados 4 y 5 corresponden al caso de operandos en memoria.

- Estado 4 – cálculo de la dirección o direcciones de los operandos.
- Estado 5 - Búsqueda del o de los operandos, en memoria y almacenamiento temporal en la CPU.

# Descripción del ciclo de instrucción detallado

55

## Fase de ejecución - Estado 6

El estado 6 se conoce normalmente como Fase de ejecución propiamente dicha (en inglés EX, Execute).

➤ Estado 6 – ejecución propiamente dicha: con los operandos en la CPU, se ejecuta la operación indicada en la instrucción.

# Descripción del ciclo de instrucción detallado

## Fase de almacenamiento de los resultados - Estados 7 y 8

Los estados 7 y 8 se conocen normalmente como Fase de almacenamiento de los resultados (en inglés WB, Write back). Algunas instrucciones pueden guardar 1 resultado y otras más. El o los operandos se pueden guardar en registros o memoria.

En el diagrama de estados, los estados 7 y 8 corresponden al caso de resultados en memoria.

- Estado 7 – cálculo de la dirección o direcciones de los resultados.
- Estado 8 – almacenamiento del o de los resultados, en memoria.

# Ejemplo: análisis de ejecución de la instrucción SUMAR (ADD)

- Vamos a aplicar el modelo del ciclo de instrucción detallado para analizar el comportamiento de la máquina en la ejecución de una instrucción de suma (nemónicamente conocida como ADD). La instrucción ADD suma 2 operandos y produce 1 resultado.
- Típicamente, una instrucción como la de ADD contiene varios “campos” (es decir, conjuntos de bits asociados) que informan de:
  - El tipo de operación (o código de operación)
  - Referencia a operando 1 (lugar donde reside operando 1)
  - Referencia a operando 2 (lugar donde reside operando 2)
  - Referencia a resultado (lugar donde se guarda el resultado): a veces se referencia, a veces no.



# Ejemplo: análisis de ejecución de la instrucción SUMAR (ADD)

- Una posible forma que tendría la instrucción (“el formato de instrucción”) típicamente podría ser el siguiente:



ref. 1er operando

ref. 2do operando

Código de operación

- Se ha supuesto que uno de los operandos está en un registro, y el otro en memoria.
- Por razones que se explicarán más adelante, no hay referencia al lugar donde se guarda el resultado (se dice que está implícita esa referencia).



# Ejemplo: análisis de ejecución de la instrucción SUMAR (ADD)

## Fase de búsqueda de la instrucción (IF)

- Estado 1: La CPU determina la dirección de la próxima instrucción a iniciar, en este caso ADD. Esta dirección deberá estar en el PC, obtenida como:
  - Opción 1: un incremento de la dirección anterior (ejecución secuencial de instrucciones consecutivas)
  - Opción 2: mediante un cálculo a partir del valor corriente del PC y/o de la instrucción previa.

# Ejemplo: análisis de ejecución de la instrucción SUMAR (ADD)

## Fase de búsqueda de la instrucción (IF)

- Estado 2: La CPU busca en memoria la instrucción que está iniciando, es decir ADD.
  - Para ello, la CPU transfiere el PC al registro interno MAR, y desde este registro, a través del bus de direcciones, se envía la dirección completa donde reside la instrucción que está comenzando.
  - Se leen una ó más palabras de memoria (la instrucción puede ocupar más de una palabra en memoria). A través del bus de datos y el MBR, se carga la instrucción en el IR.

Nota: es muy común que después de buscar la instrucción, la CPU automáticamente incremente el PC para apuntar a la dirección de memoria siguiente (próxima instrucción).

# Ejemplo: análisis de ejecución de la instrucción SUMAR (ADD)

## Fase de decodificación de la instrucción (DEC)

- Estado 3: La CPU decodifica la instrucción corriente almacenada en el IR.
  - La instrucción ADD queda temporalmente almacenada en el IR.
  - La CU decodifica la instrucción y determina a partir de su información:
    - Tipo de operación
    - Donde buscar los operandos
    - Donde guardar el resultado
  - La instrucción es auto-contenida, en ella está toda la información requerida para resolverla.

# Ejemplo: análisis de ejecución de la instrucción SUMAR (ADD)

## Fase de búsqueda de los operandos (OF)

- Estado 4: La CPU determina donde están los operandos.
  - Un operando está en un registro interno. No tiene que calcular dirección de memoria, sólo obtener el número de registro desde el campo “referencia a operando 1” actualmente almacenado en el IR.
  - El segundo operando está en memoria, por lo que debe obtener la dirección de memoria donde reside. Esta dirección la calcula a partir de la información en el campo “referencia operando 2” actualmente almacenado en el IR.



# Ejemplo: análisis de ejecución de la instrucción SUMAR (ADD)

## Fase de búsqueda de los operandos (OF)

- Estado 5: La CPU busca el segundo operando en memoria.
  - La CPU transfiere al registro interno MAR el valor obtenido de la dirección de memoria donde reside el segundo operando, y desde este registro, a través del bus de direcciones, se envía la dirección completa donde reside el segundo operando.
  - Se leen una ó más palabras de memoria (el operando puede ocupar más de una palabra en memoria). A través del bus de datos y el MBR se guarda en un registro temporal dentro de la CPU.



# Ejemplo: análisis de ejecución de la instrucción SUMAR (ADD)

## Fase de ejecución propiamente dicha (EX)

- Estado 6: La CPU ejecuta la instrucción de suma ADD.
  - La CPU tiene los 2 operandos disponibles.
  - Los transfiere a la ALU como entradas de datos, y le indica a la ALU ejecutar la operación de suma.
  - La ALU ejecuta la suma y el resultado se guarda provisoriamente en un registro temporal “invisible” al programador.

# Ejemplo: análisis de ejecución de la instrucción SUMAR (ADD)

## Fase de almacenamiento del resultado (WB)

- Estado 7: La CPU calcula donde se debe guardar el resultado.
  - El cálculo para determinar donde se guarda el resultado depende del lugar donde se almacena:
    - Algún registro de la CPU: no hay cálculo, es el número de registro referenciado en la instrucción.
    - Alguna posición de memoria: se calcula con la información presente en la instrucción.
  - En este caso en el que la instrucción no tiene referencia sobre el lugar donde se debe guardar el resultado, la CPU asume que se guardará en un lugar predeterminado (referencia implícita).

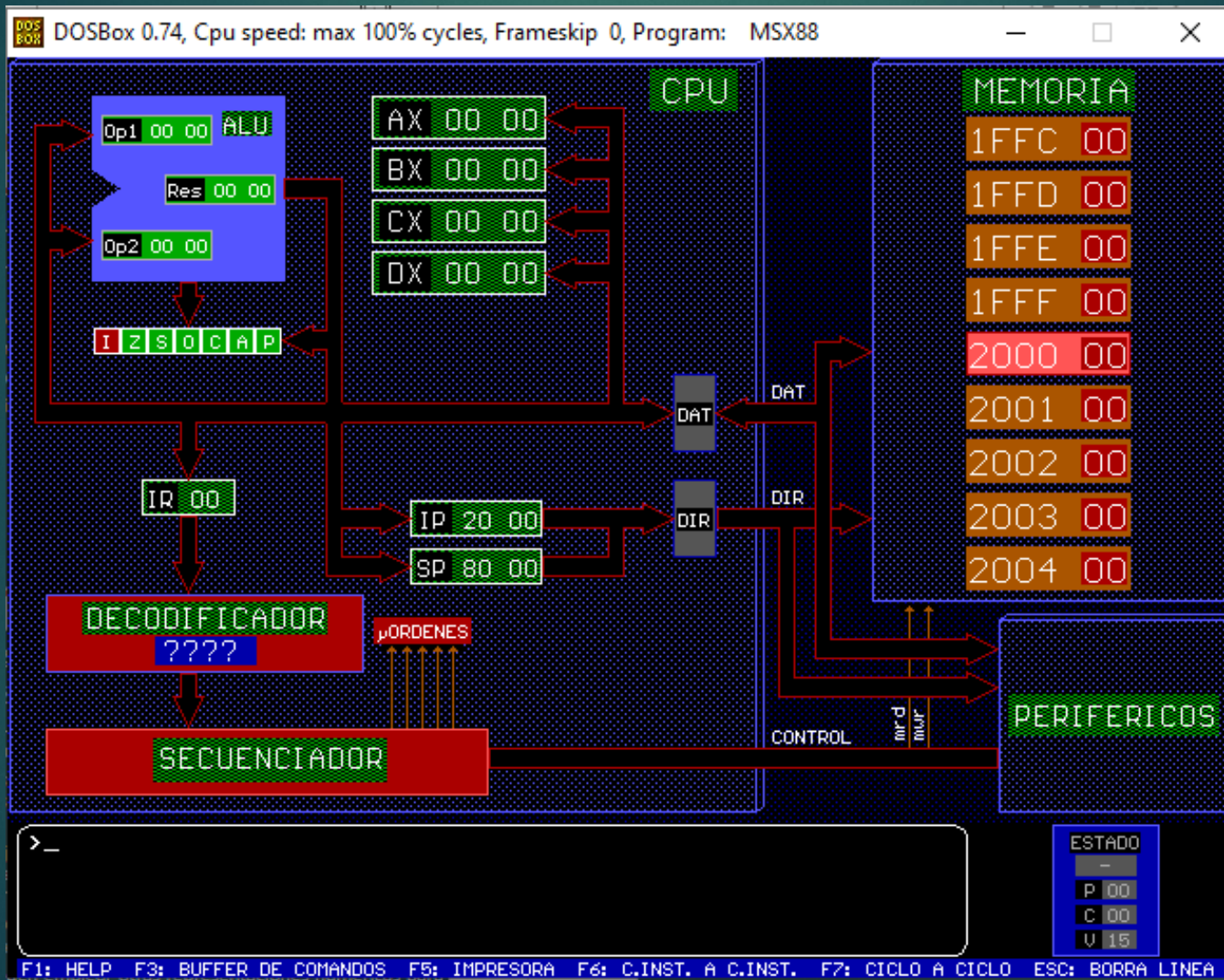
# Ejemplo: análisis de ejecución de la instrucción SUMAR (ADD)

## Fase de almacenamiento del resultado (WB)

- Estado 8: La CPU almacena el resultado.
  - Una vez determinado el lugar donde se almacenará, el resultado es transferido al destino final: registro o posición de memoria.
  - La última acción de la instrucción corriente es saltar al estado 1, con lo que comienza un nuevo ciclo de instrucción.

# Otros ejemplos

- En el simulador MSX88:





# Otros ejemplos

- Ejecutar aplicación MSX88.exe
- Almacenar en memoria los siguientes números hexadecimales, a partir de la dirección de memoria  $2000_{16}$ :  
e 2000 b8002081c000e0
- Ejecutar el programa con la opción de ejecución ciclo a ciclo mediante la tecla <F6>, y analizar la respuesta de la CPU y memoria durante la ejecución del mismo. Sacar conclusiones y compararlas con el análisis desarrollado en esta clase.



# Referencias

- Stallings. 5ta Ed.
  - Capítulo 3
  - Capítulo 11
- Lenguaje Assembly
  - Apunte 4 de cátedra
  - Simulador MSX88
  - Descargas en página web de cátedra
  - <http://weblidi.info.unlp.edu.ar/catedras/organiza/>