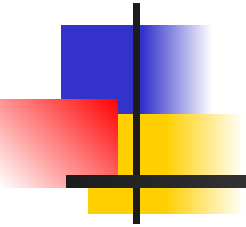


Organización de Computadoras



Clase 6



Tema de Clase

- Ciclo de Instrucción



Función de la computadora(1)

- Ejecutar programas
- El programa está compuesto de instrucciones almacenadas en memoria
- La CPU *procesa* las instrucciones
 - Debe traerlas desde memoria una por vez
 - Debe cumplir cada operación ordenada



Función de la computadora(2)

- Podemos descomponer el procesamiento de instrucciones en dos etapas:
 - Búsqueda: leer desde memoria.
 - ✓ Común a todas las instrucciones.
 - Ejecución: dependiendo de la instrucción puede implicar varias operaciones

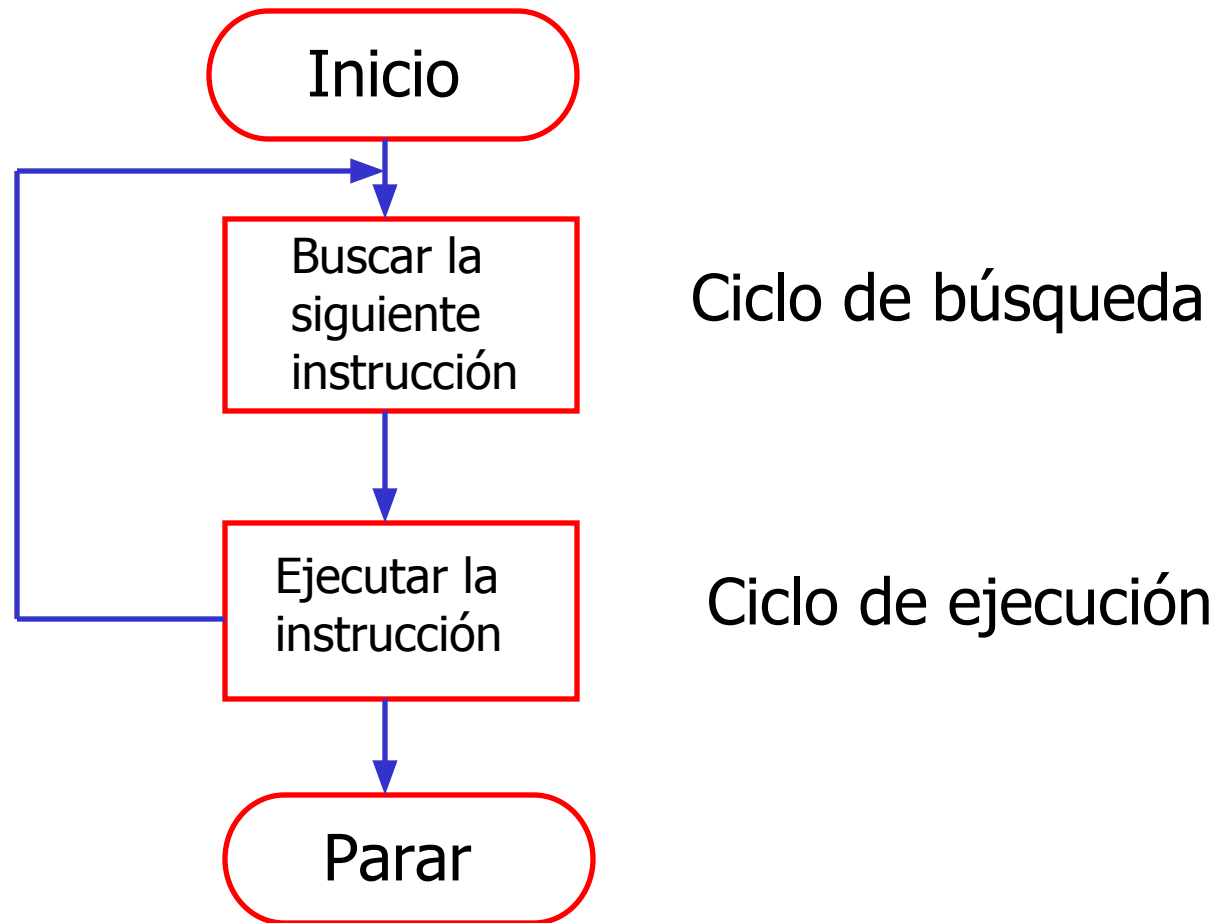


Función de la computadora(3)

- El procesamiento requerido para una sola instrucción se llama ciclo de instrucción.
- Dos pasos:
ciclo de búsqueda y ciclo de ejecución
- La ejecución del programa se interrumpe sólo si la máquina se apaga, hay un error ó una instrucción que interrumpa a la computadora.



Ciclos de búsqueda y ejecución(1)





Ciclos de búsqueda y ejecución(2)

- Al principio de cada ciclo, la CPU busca una instrucción en memoria.
- En la CPU hay un registro, llamado contador de programa (PC), que tiene la dirección de la próxima instrucción a buscar.
- La CPU, después de buscar cada instrucción, incrementa el valor contenido en PC; así podrá buscar la siguiente instrucción en secuencia.



Ciclos de búsqueda y ejecución(3)

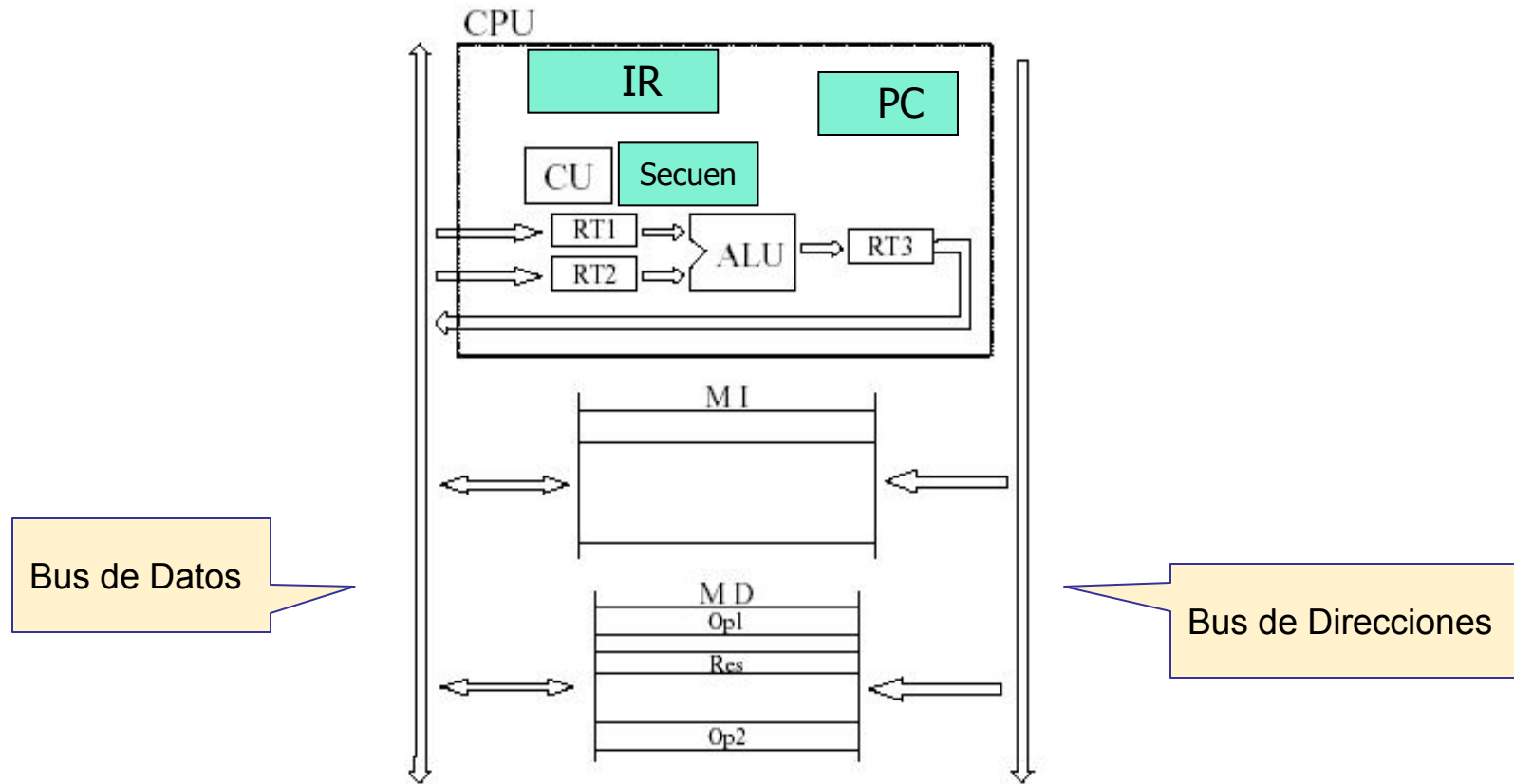
- La instrucción buscada se carga dentro de un registro de la CPU, llamado registro de instrucción (IR).
- La instrucción está en la forma de un código binario que especifica las acciones que tomará la CPU.
- La CPU interpreta cada instrucción y lleva a cabo las acciones requeridas.



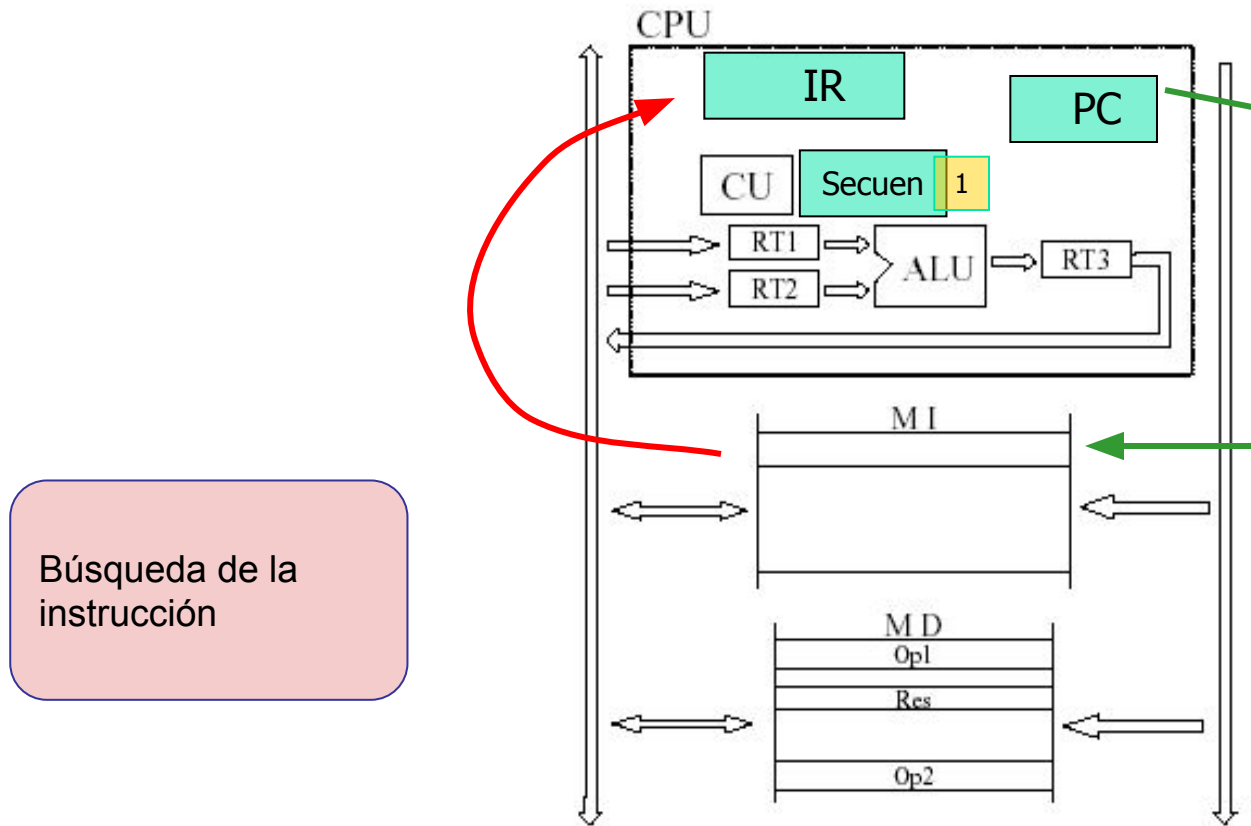
Ciclos de búsqueda y ejecución(4)

- En general las acciones caen en 4 tipos:
 - CPU – Memoria
 - datos pueden transferirse entre memoria y CPU.
 - CPU – E/S
 - datos pueden transferirse entre CPU y entrada/salida.
 - Procesamiento de datos
 - CPU efectúa operaciones aritméticas ó lógicas en datos.
 - Control
 - alterar la secuencia de ejecución de instrucciones.

Ejemplo de búsqueda y ejecución

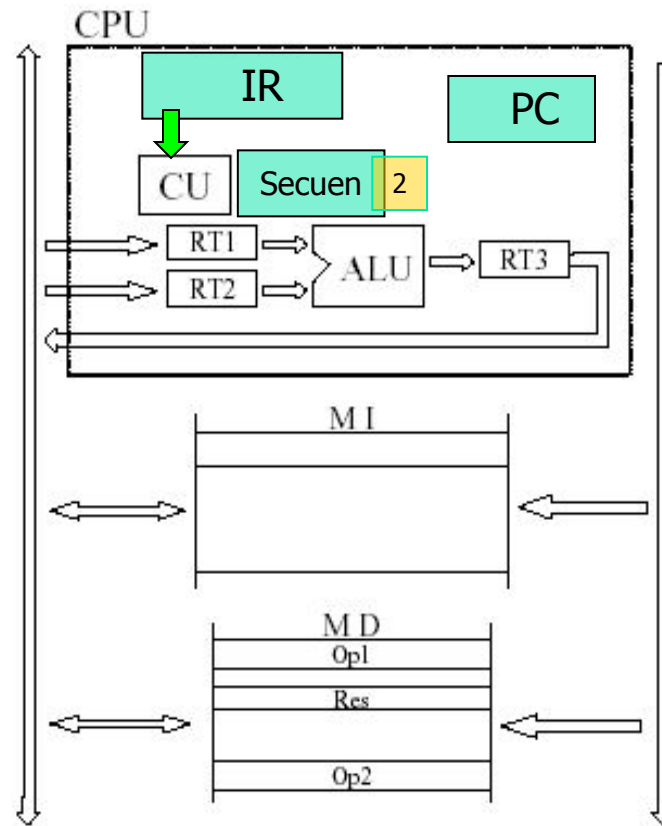


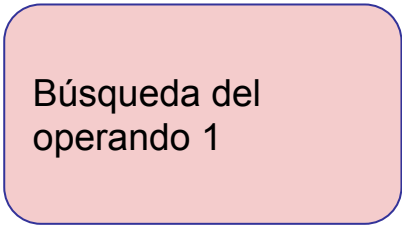
Ejemplo de búsqueda y ejecución (1)



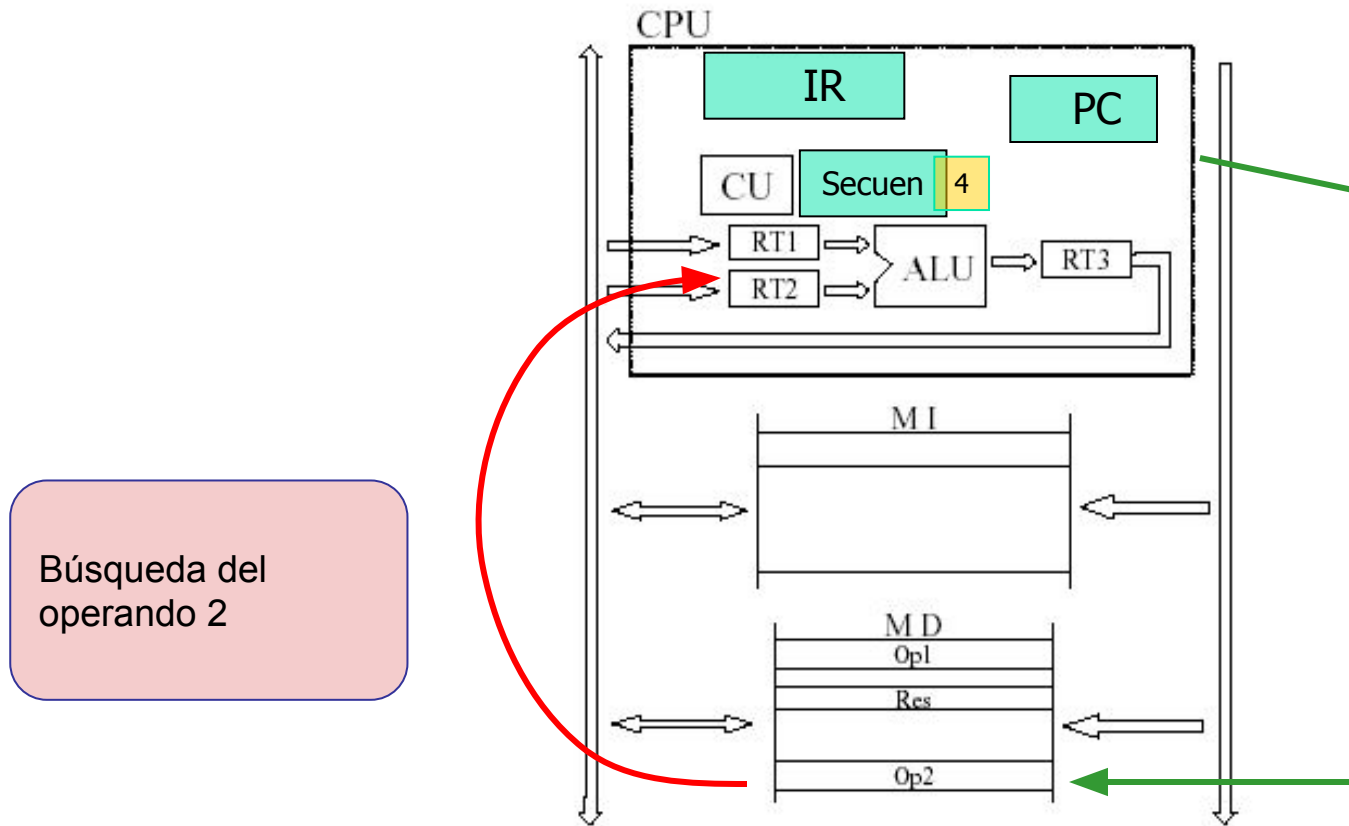
Ejemplo de búsqueda y ejecución (2)

Decodificación de la instrucción



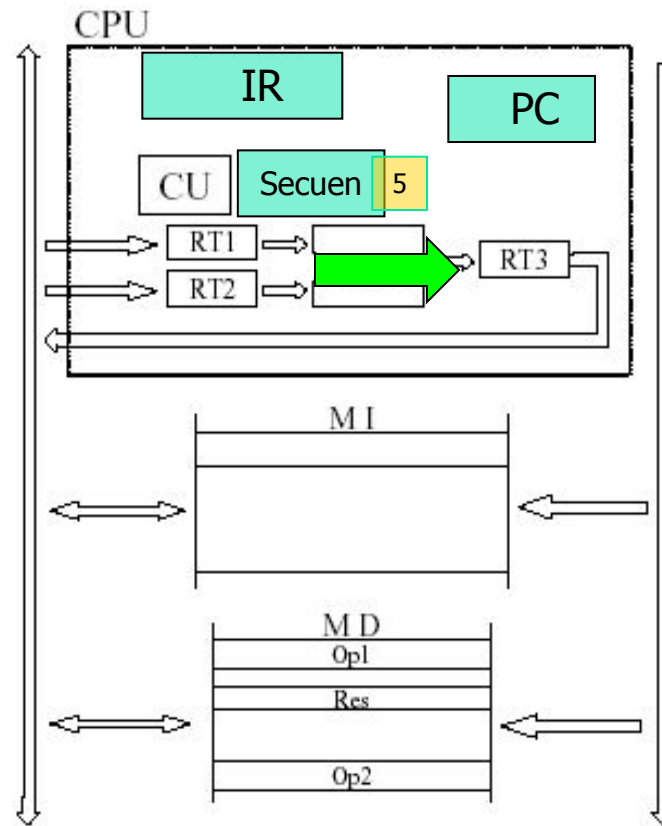


Ejemplo de búsqueda y ejecución (4)

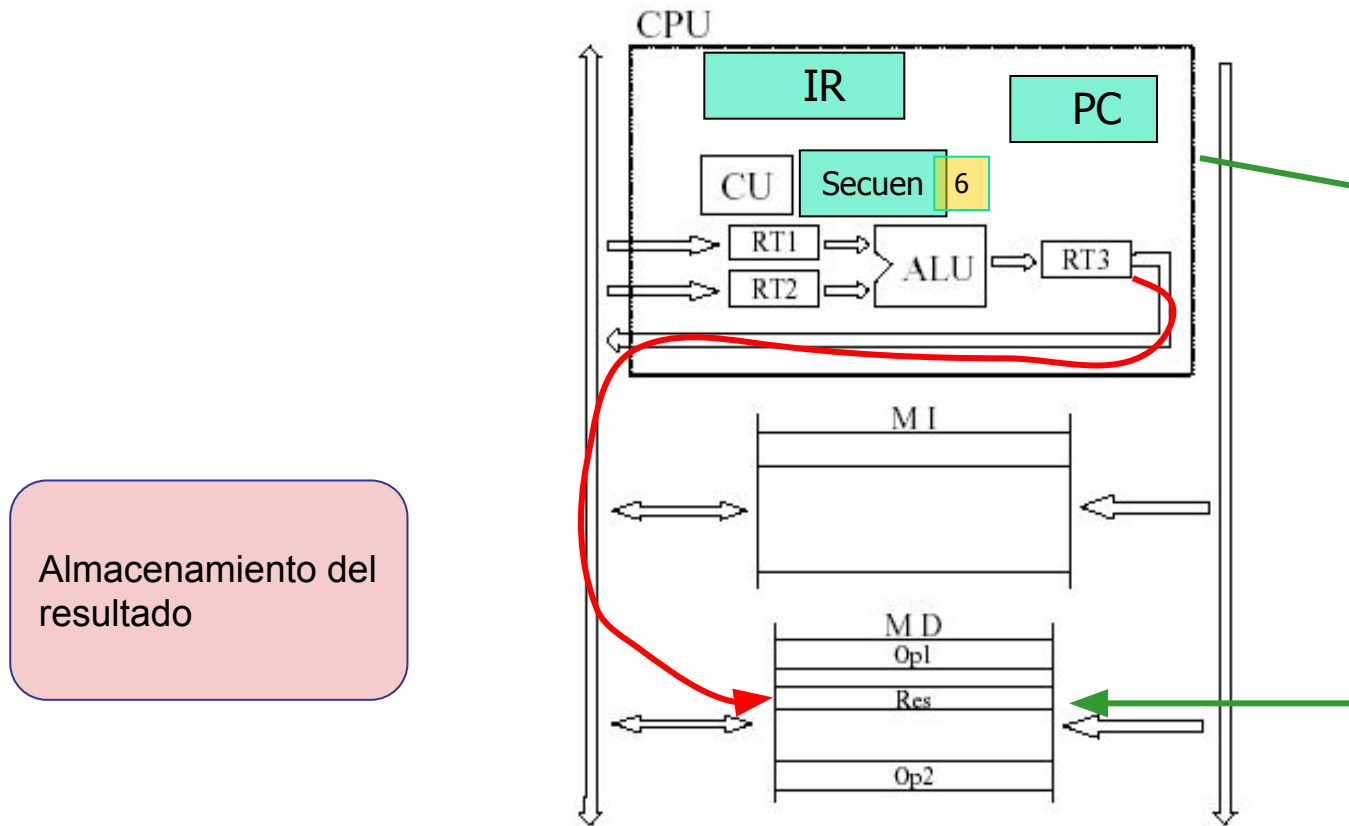


Ejemplo de búsqueda y ejecución (5)

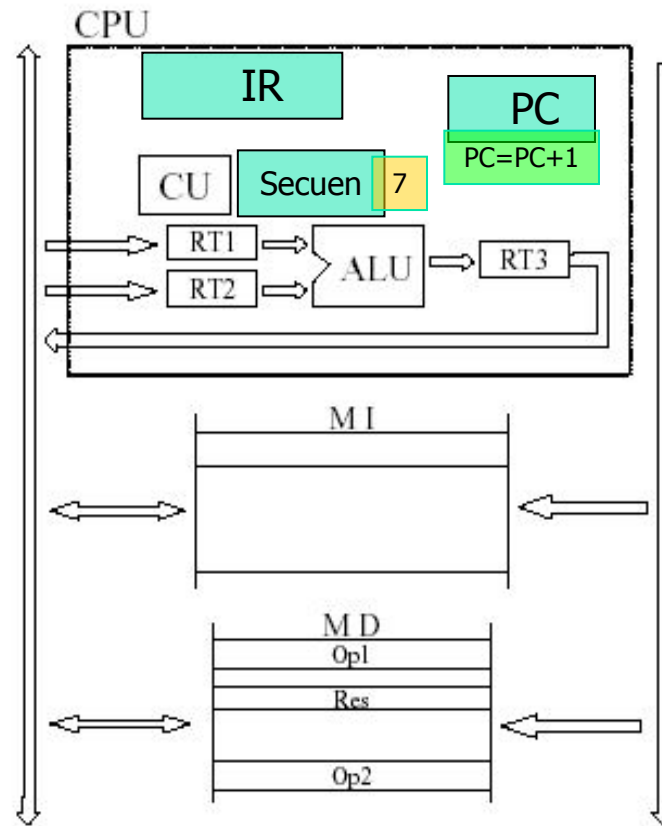
Cálculo del resultado
(operación con los
operandos 1 y 2)



Ejemplo de búsqueda y ejecución (6)

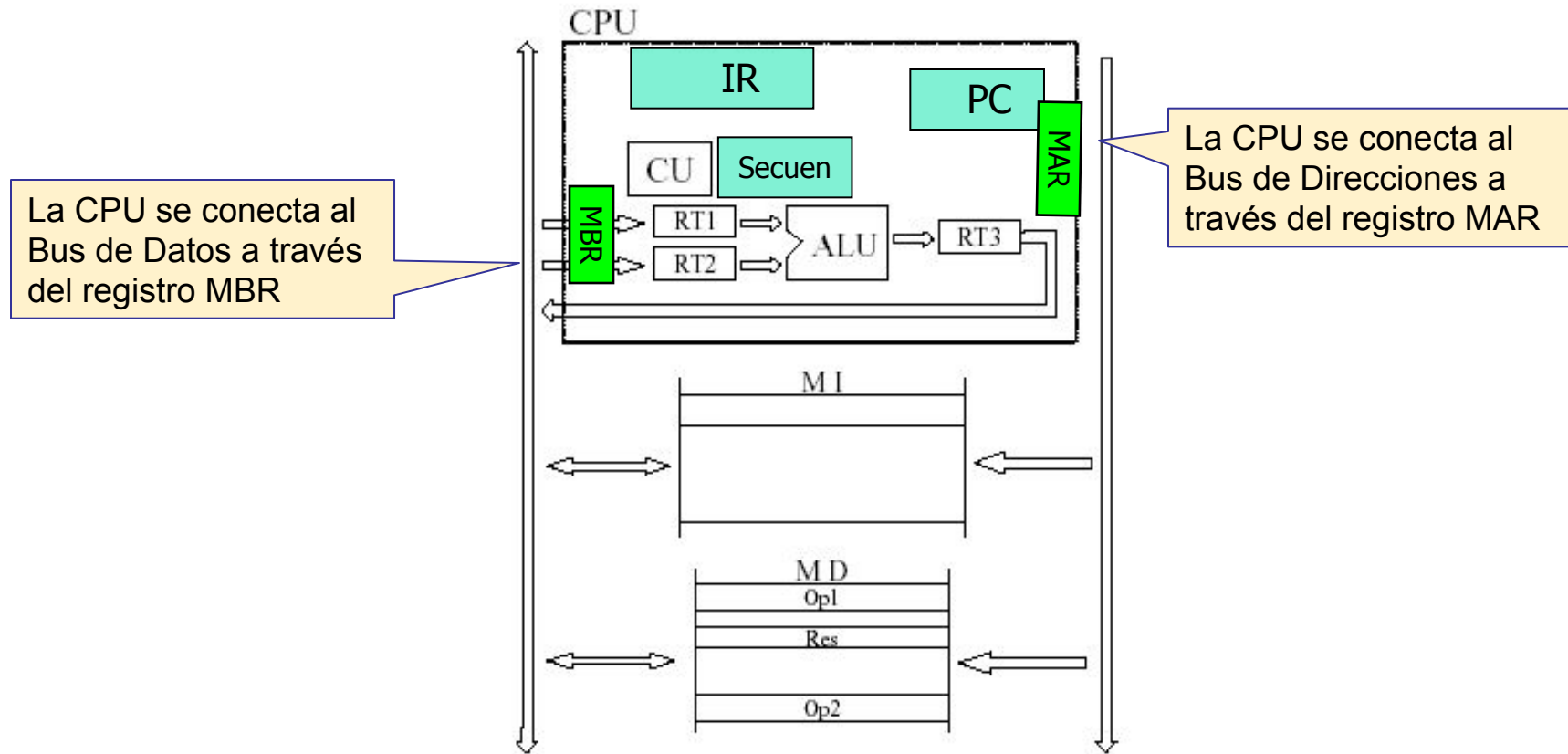


Ejemplo de búsqueda y ejecución (7)



Cálculo de la
dirección de la
próxima instrucción

Ejemplo de búsqueda y ejecución (8)





Ejemplo

- ❖ El siguiente ejemplo muestra la ejecución de un fragmento de programa que tiene tres instrucciones:
 - ❖ Cargar en el registro D el contenido de la posición de memoria 940_{16}
 - ❖ Sumar el contenido de la posición de memoria 941_{16} al registro D y guardar el resultado en D
 - ❖ Almacenar el valor del registro D en la posición de memoria 941_{16}



Ejemplo (cont)

- Consideremos que cada posición de memoria almacena 16 bits. Los primeros 4 bits indican la operación a realizar, los siguientes 12 bits indican una dirección de memoria.

- $0001_2 = 1 =$ cargar D desde la memoria
- $0010_2 = 2 =$ almacenar D en memoria
- $0101_2 = 5 =$ sumar D con un dato en memoria

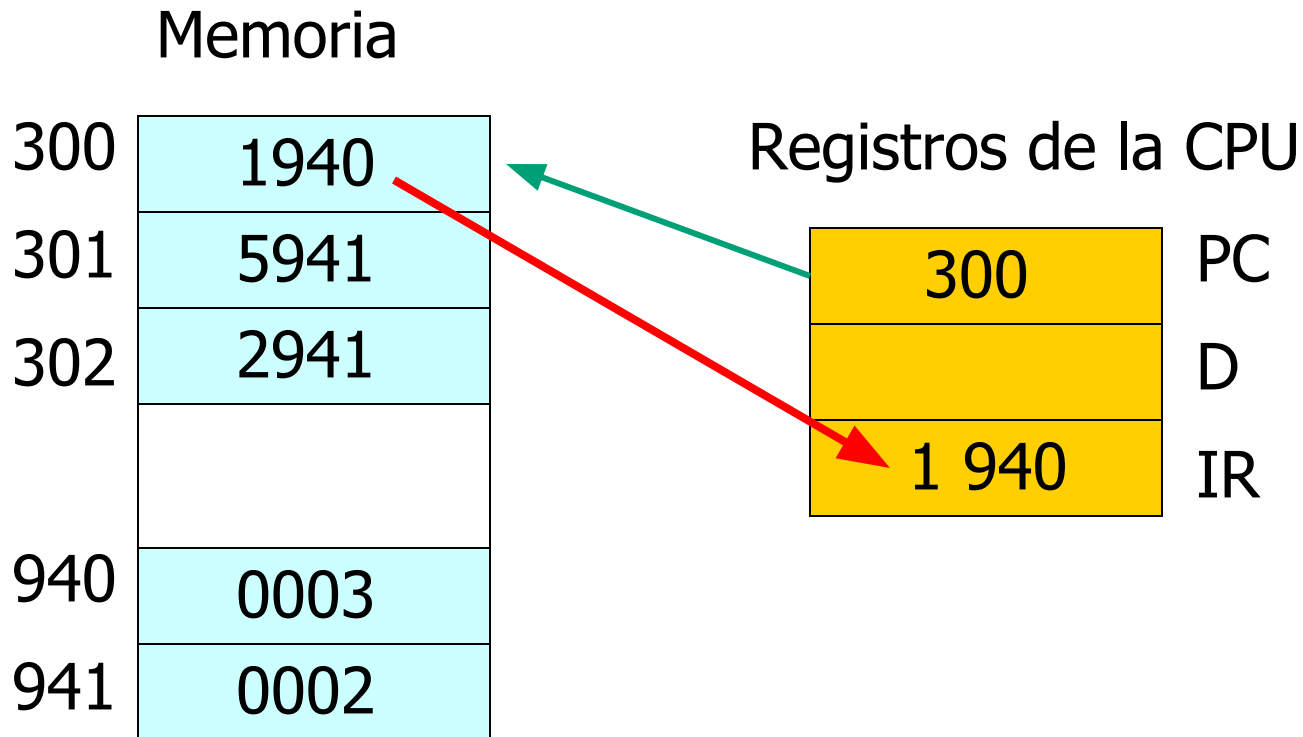


Ejemplo: paso 1

- ❖ El contador de programa (PC) contiene 300_{16} como la dirección de la primera instrucción. El contenido de esta dirección se carga en el registro de instrucción (IR).

Este proceso implica usar MAR y MBR. Para simplificar por ahora los ignoramos.

Ejemplo: paso 1



Paso 1



Ejemplo: paso 2

- ❖ 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} .

- $0001_2 = 1 =$ cargar D desde la memoria
- $0010_2 = 2 =$ almacenar D en memoria
- $0101_2 = 5 =$ sumar D con un dato en memoria

1 940

IR



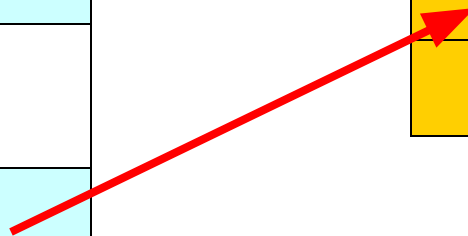
Ejemplo: paso 2

Memoria

300	1940
301	5941
302	2941
940	0003
941	0002

Registros de la CPU

300	PC
0003	D
1 940	IR



Paso 2

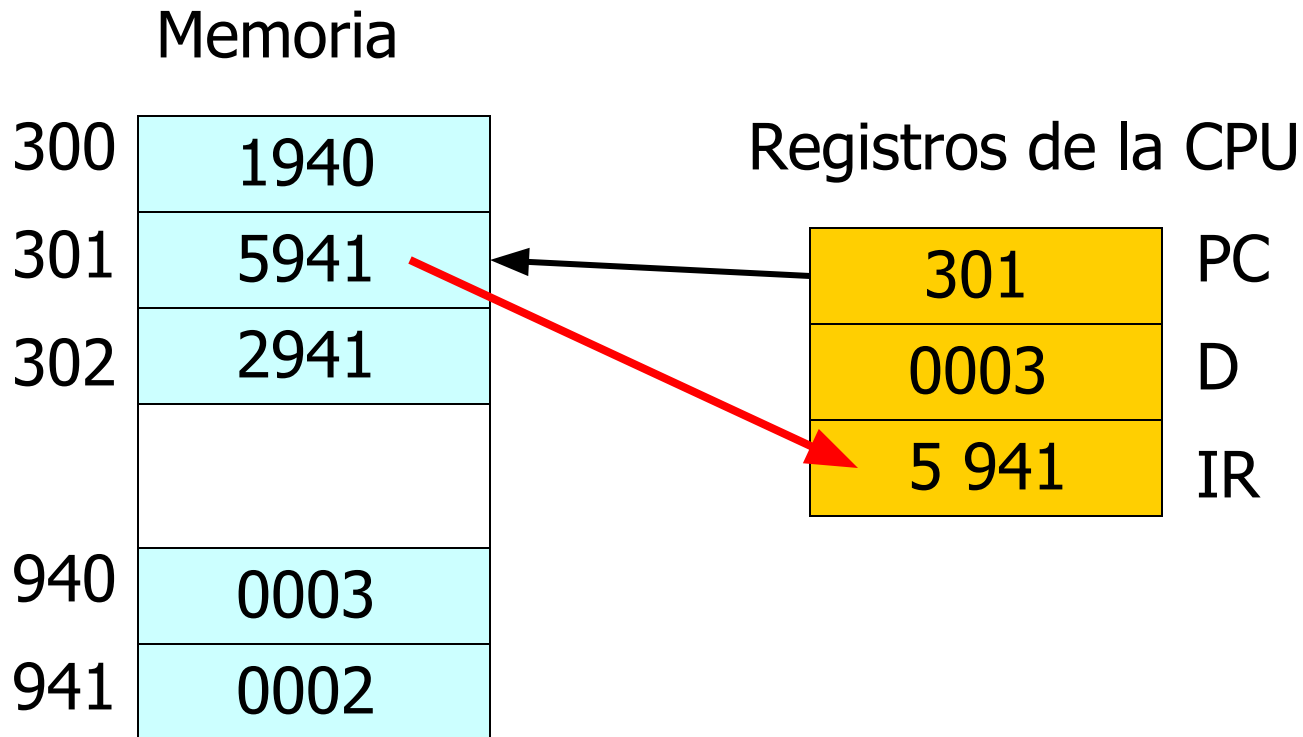


Ejemplo: paso 3

- ❖ Se incrementa el contador de programa y se busca la siguiente instrucción en la dirección 301_{16} .



Ejemplo: paso 3



Paso 3



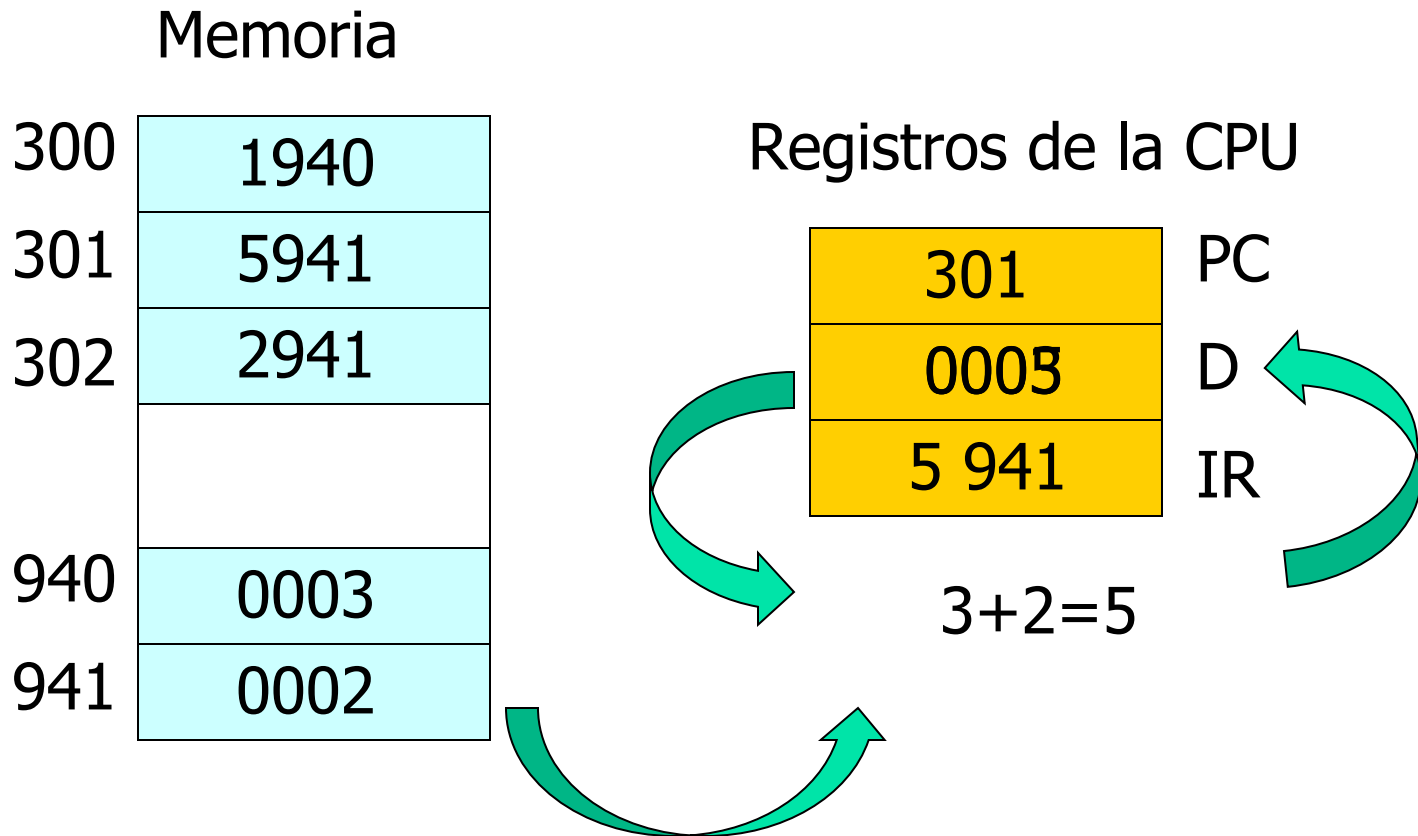
Ejemplo: paso 4

- ❖ El 5_{16} en IR indica 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.

- $0001_2 = 1 =$ cargar D desde la memoria
- $0010_2 = 2 =$ almacenar D en memoria
- $0101_2 = 5 =$ sumar D con un dato en memoria

5 941 IR

Ejemplo: paso 4



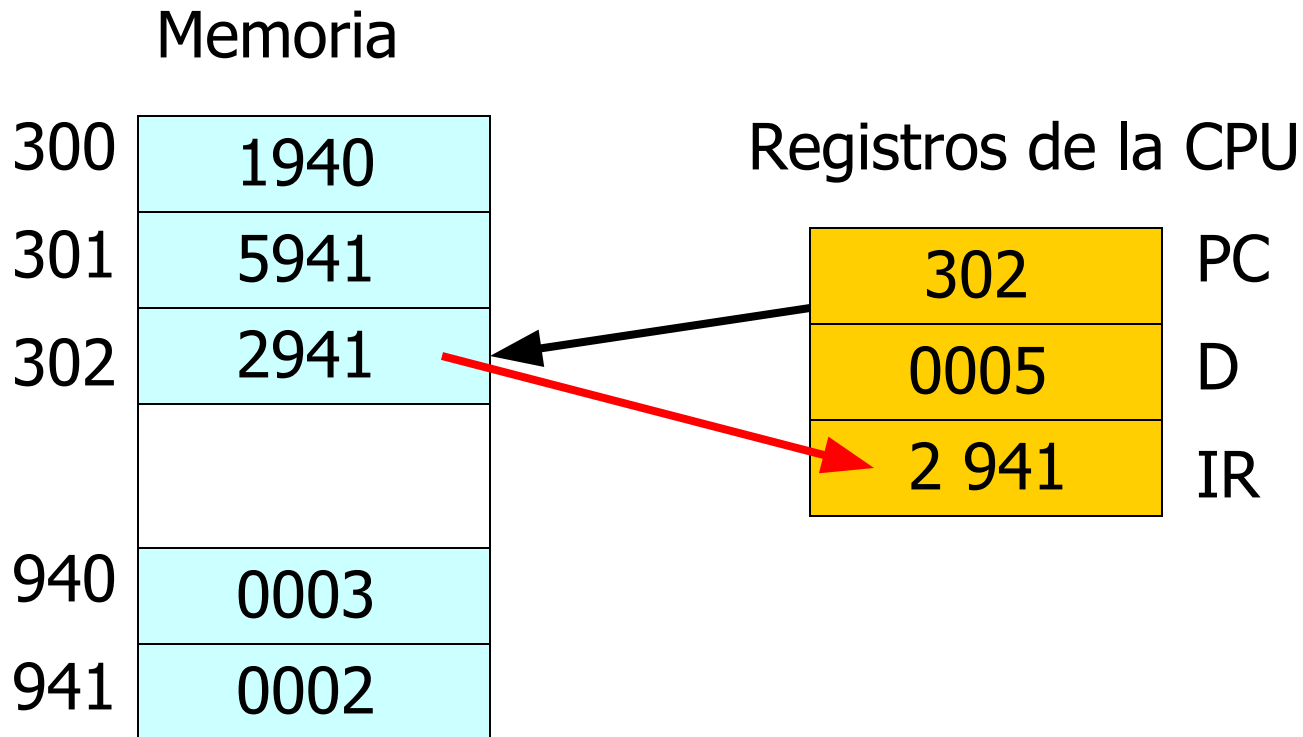
Paso 4



Ejemplo: paso 5

- ❖ Se incrementa el PC y se busca la siguiente instrucción en 302_{16}

Ejemplo: paso 5



Paso 5



Ejemplo: paso 6

- ❖ El 2_{16} en IR indica que el contenido del registro D se almacena en la dirección 941_{16} , que está especificada en los bits restantes de la instrucción.

- $0001_2 = 1 =$ cargar D desde la memoria
- $0010_2 = 2 =$ almacenar D en memoria
- $0101_2 = 5 =$ sumar D con un dato en memoria

2 941

IR



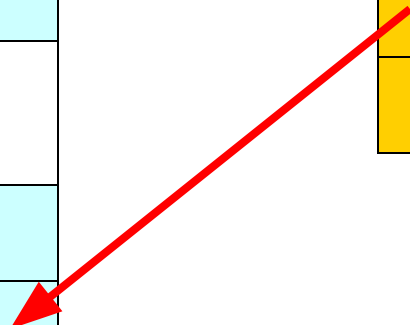
Ejemplo: paso 6

Memoria

300	1940
301	5941
302	2941
940	0003
941	0005

Registros de la CPU

302	PC
0005	D
2 941	IR



Paso 6



Ejemplo: paso 6

- En este ejemplo, se necesitaron 3 ciclos de instrucción, cada uno con un ciclo de búsqueda y un ciclo de ejecución.



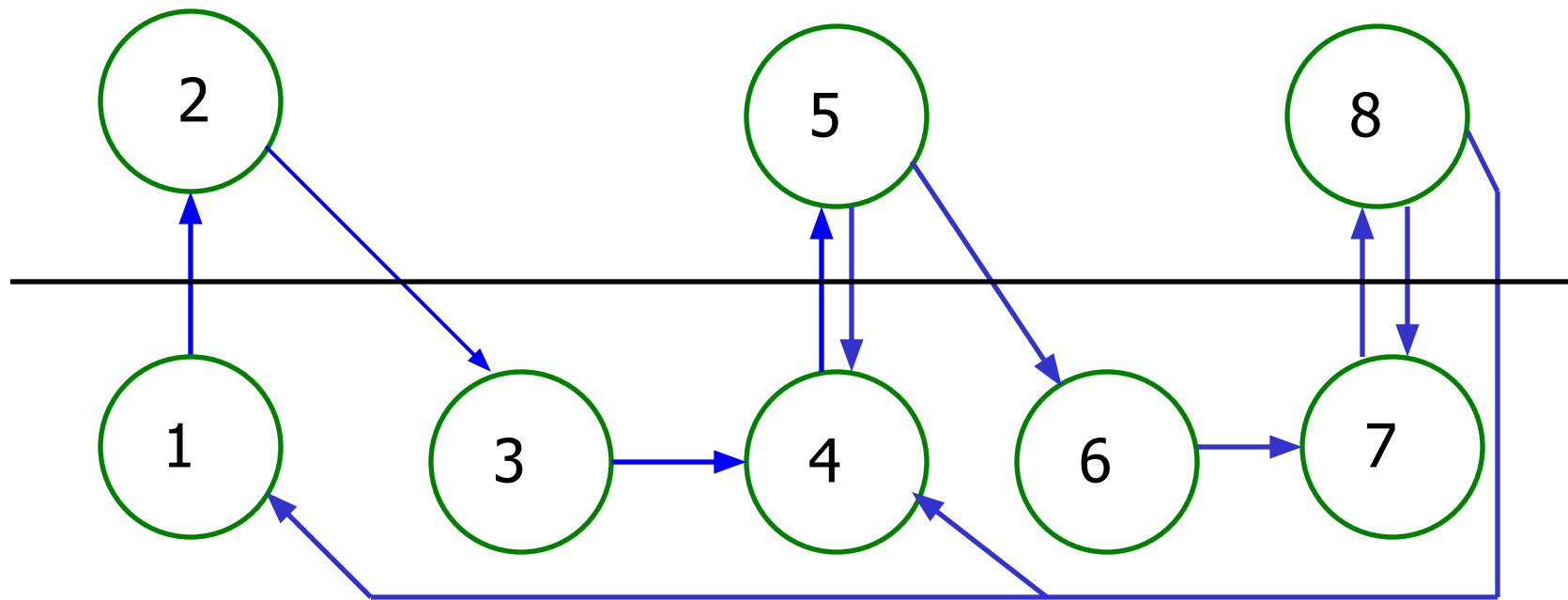
Ciclo de instrucción

- Con este ejemplo podemos ahora tener una visión más detallada del ciclo de instrucción básico. La figura siguiente está en forma de diagrama de estados. Para cualquier ciclo de instrucción dado, algunos estados pueden no estar y otros pueden repetirse.



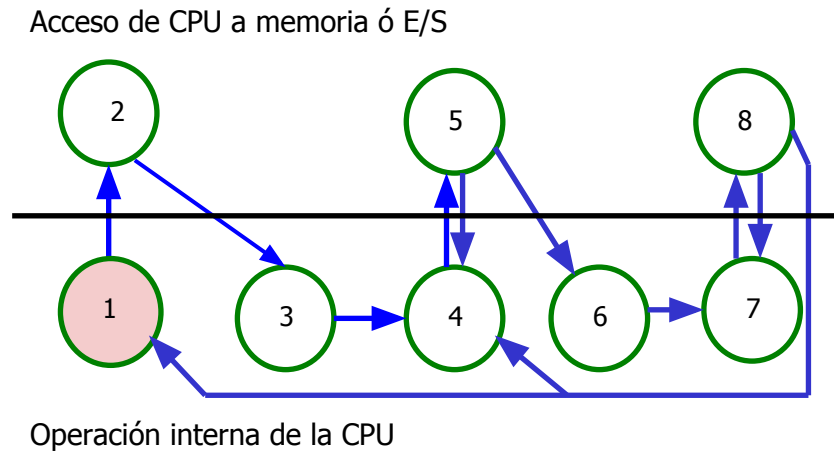
Diagrama de estados

Acceso de CPU a memoria ó E/S

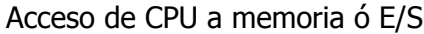


Operación interna de la CPU

Diagrama de estados (1)

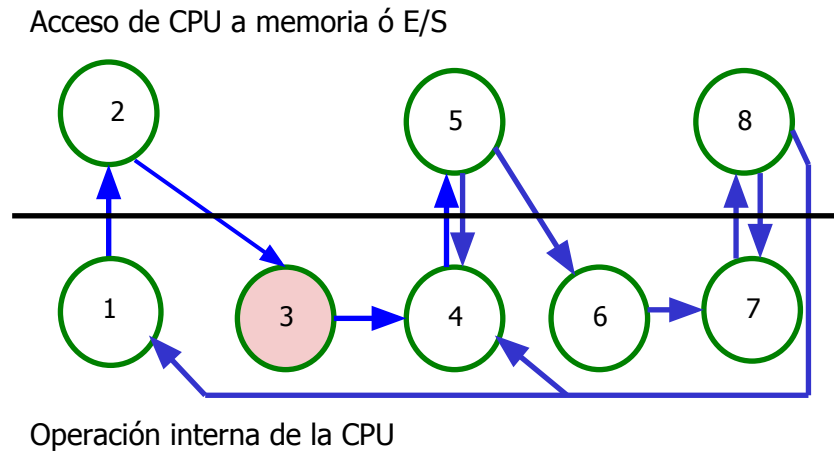


1. cálculo dirección instrucción: determina la dirección de la siguiente instrucción a ejecutarse



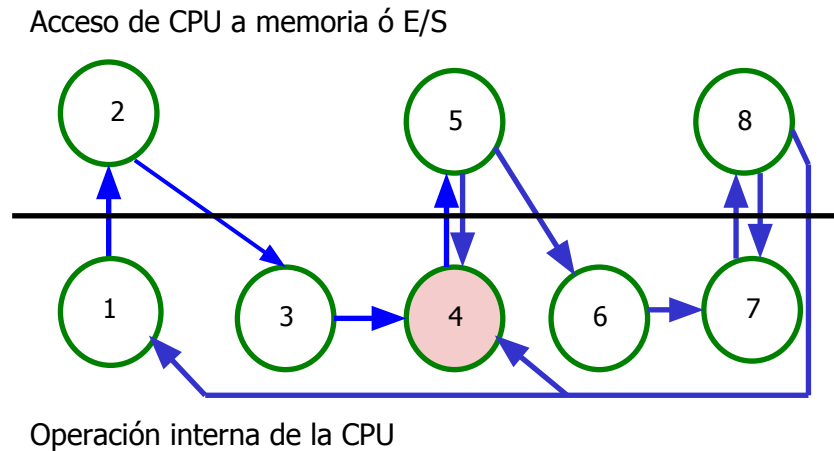
- ## Operación interna de la CPU

Diagrama de estados (3)

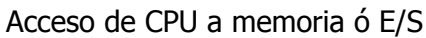


3. decodificación de la instrucción: analiza la instrucción para determinar el tipo de operación a realizar y los operandos que se usarán.

Diagrama de estados (4)

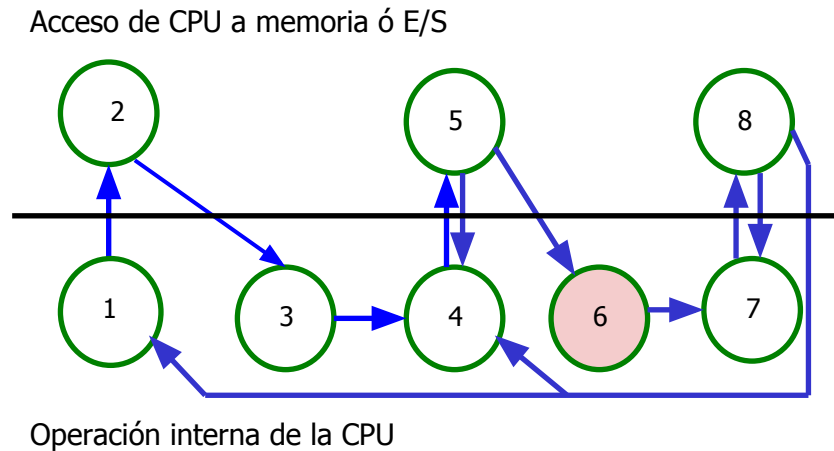


4. cálculo dirección operando: si la operación implica la referencia a un operando en la memoria ó e/s, entonces se determina la dirección.

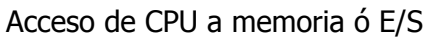


- ## Operación interna de la CPU

Diagrama de estados (6)



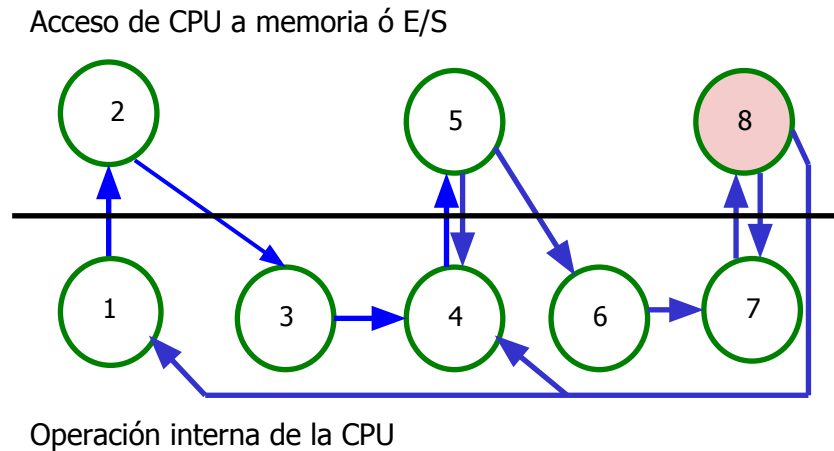
6. operación sobre los datos: ejecuta la instrucción.



Operación interna de la CPU

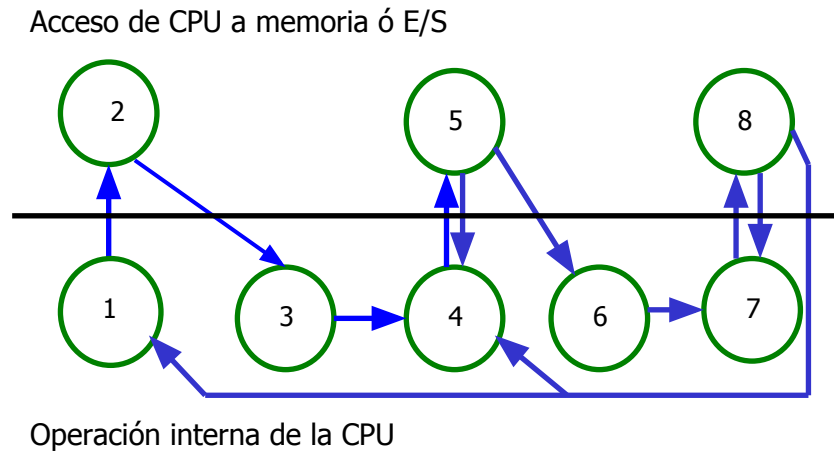
7. cálculo dirección resultado.

Diagrama de estados (8)



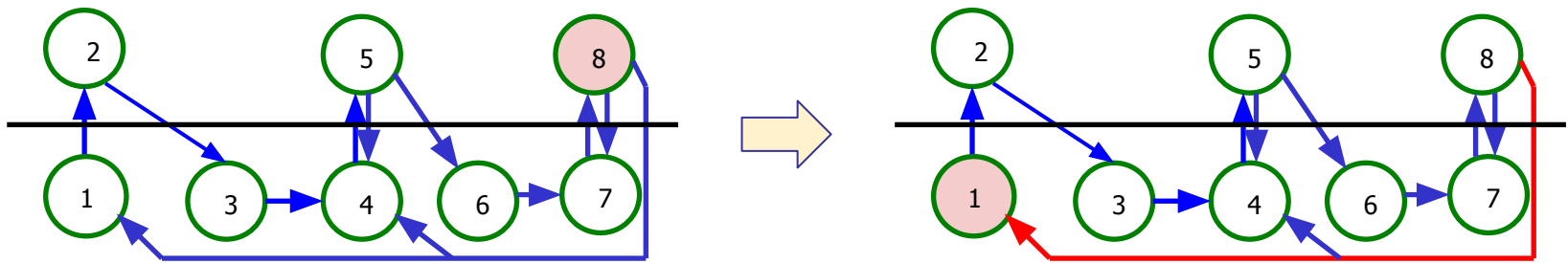
8. almacenamiento resultado.

Diagrama de estados (9)



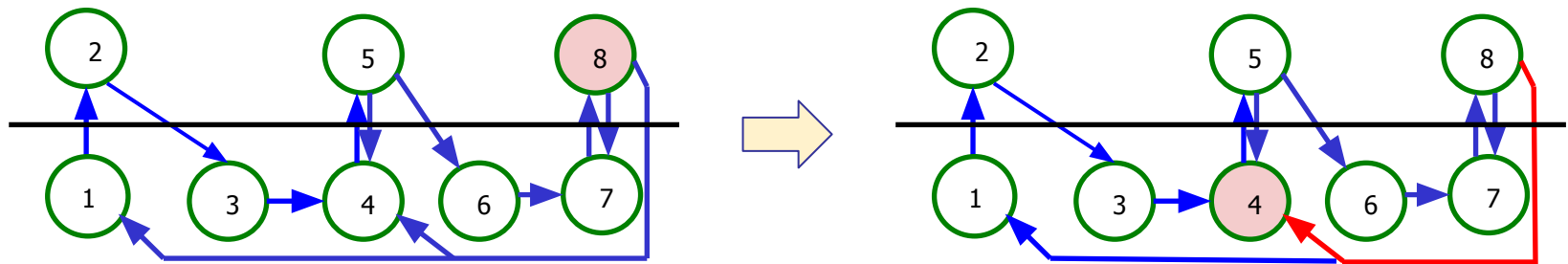
- Los estados en la parte superior implican un intercambio entre la cpu y la memoria ó e/s.
- Los estados en la parte inferior implican sólo operaciones internas en la cpu.

Diagrama de estados (10)



- Terminado el ciclo de instrucción, se da comienzo a uno nuevo.

Diagrama de estados (11)



En algunas máquinas hay instrucciones que procesan arreglos de operandos (vectores).

- Luego de almacenar una componente del vector resultado, se procesa la siguiente.



Diagrama de estados (12)

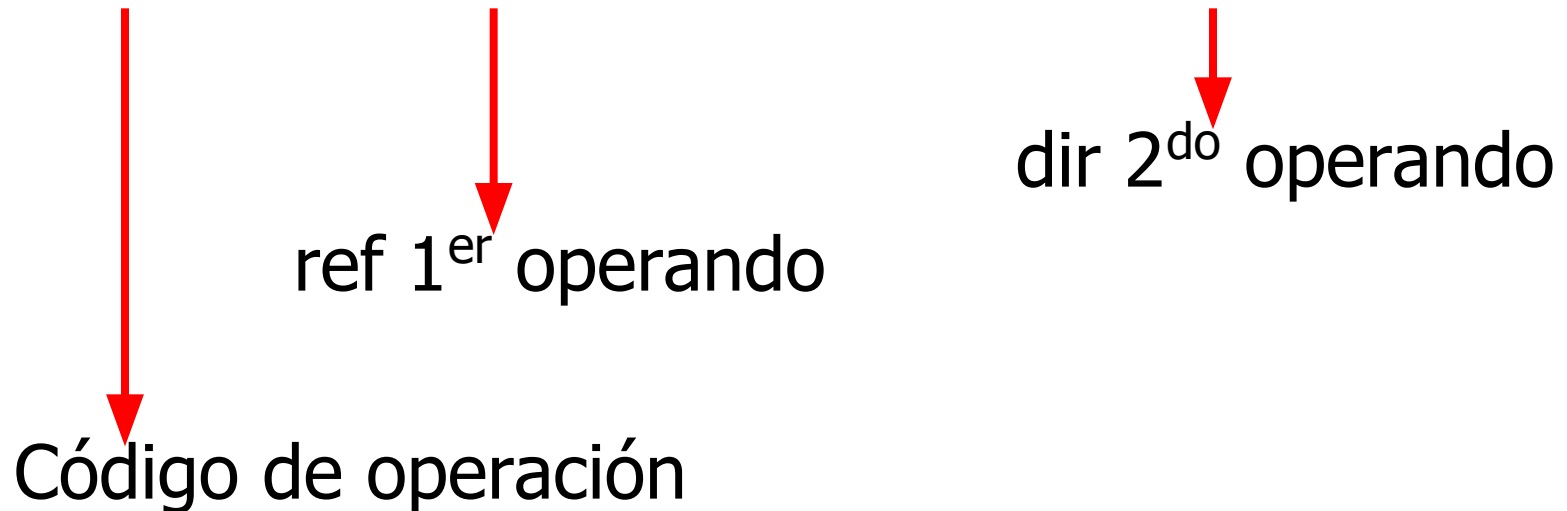
1. cálculo dirección instrucción
2. búsqueda instrucción
3. decodificación de la instrucción
4. cálculo dirección operando
5. búsqueda del operando
6. operación sobre los datos
7. cálculo dirección resultado.
8. almacenamiento resultado.



Ejemplo: ADD

- Supongamos la siguiente instrucción :

ADD referencia a registro, dirección de memoria





ADD: paso a paso (1)

- Buscar la instrucción en memoria.
- Incrementar el PC.
- Decodificar la instrucción.
- Si es necesario, buscar una constante en una dirección de memoria.
- Si es necesario, incrementar PC para que apunte más allá de la constante.



ADD: paso a paso (2)

- Si es necesario calcular la dirección del operando.
- Buscar uno de los operandos, desde memoria ó registro.
- Buscar el otro operando desde registro.
- Realizar la Suma.
- Almacenar el resultado.



Paso a paso (1)

- ❖ En el primer paso, la CPU busca la instrucción en memoria. Para esto copia el valor del PC al MAR y de ahí al bus de direcciones. La UC envía las señales necesarias para una operación de lectura. Se pueden leer uno ó más bytes. A través del bus de datos al MBR y luego al IR.



Paso a paso (2)

- ❖ Después de buscar la instrucción, la CPU debe incrementar el PC para apuntar a “/o *que sigue*”. Puede ser un dato, dirección ó la siguiente instrucción.



Paso a paso (3)

- ❖ El paso siguiente es decodificar la instrucción para saber que operación hacer (suma, resta, etc.). En este momento la CPU no sólo se entera de la operación, sino también dónde se encuentran los datos sobre los cuales operar.

La instrucción es *auto-contenida*, en ella “todo está dicho”.



Paso a paso (4)

- ❖ En el paso anterior la CPU determinó si tiene que ir a buscar un operando o constante a memoria, que ocupa una celda o más (byte/s) y lo hace en este momento.
- ❖ Si existió el paso anterior la CPU debe incrementar el PC en el valor adecuado de celdas (1, 2 o mas).



Paso a paso (5)

- ❖ En este paso si es necesario, calcula la dirección del operando.
- ❖ Buscar los operandos.
- ❖ Sumar, restar...
- ❖ Almacenar el resultado.



Ejemplos adicionales

Simulación de Instrucciones en MSX88

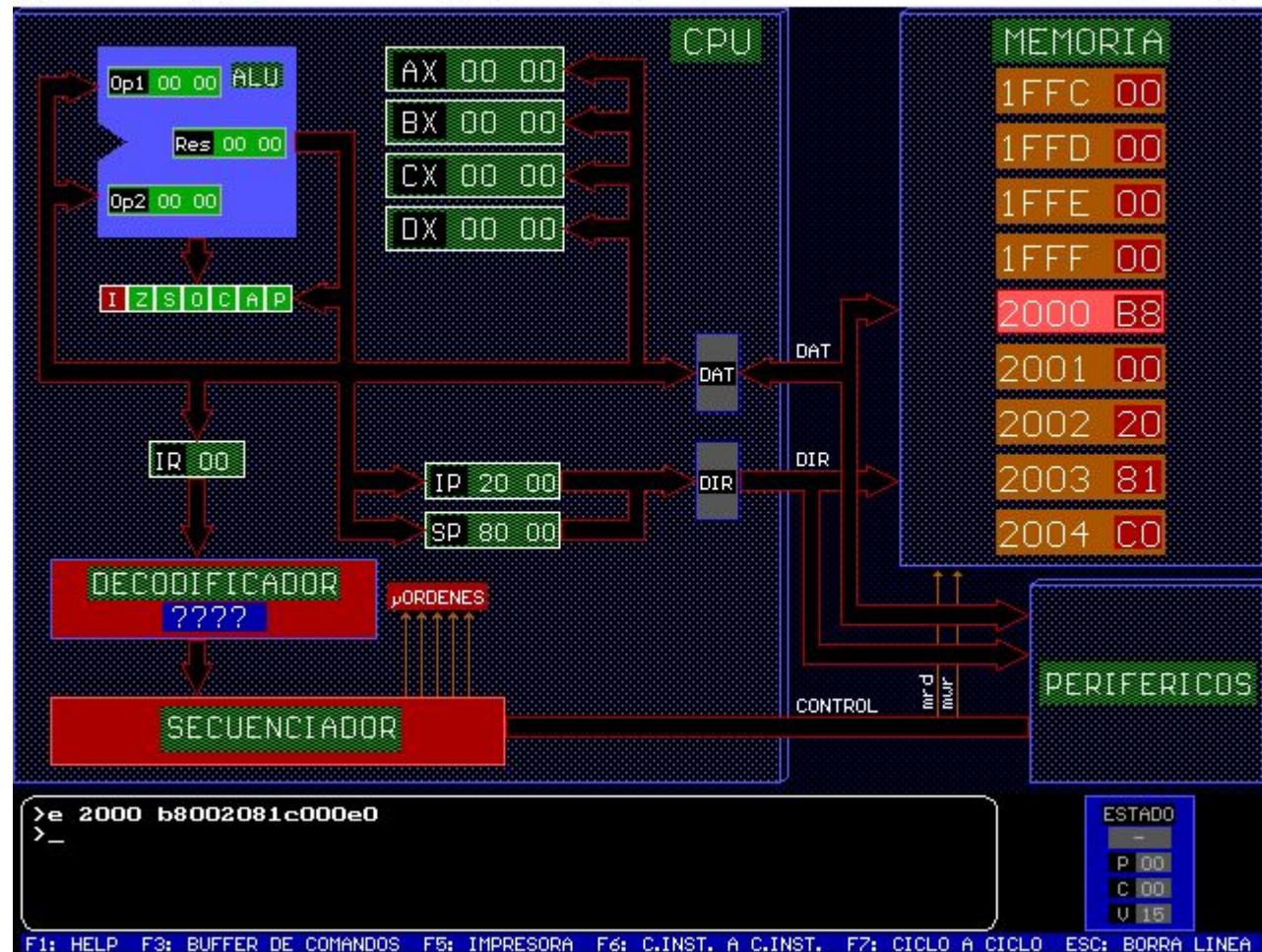
- Ejecutar aplicación MSX88.exe
- Almacenar en memoria:
 - e 2000 b8002081c000e0
- Analizar la ejecución ciclo de instrucción a ciclo de instrucción con <F6>

Ejemplos adicionales

DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: MSX88

Simulación de Instrucciones en MSX88

- Ejecutar aplicación MSX88.exe
- Almacenar en memoria: e 2000 b8002081c000e0
- Analizar la ejecución ciclo de instrucción a ciclo de instrucción con <F6>





mas información ...

Ciclo de instrucción:

- Capítulo 3 apartado 3.2.
- Capítulo 11 apartados 11.1. y 11.3.
 - Stallings, W., 5º Ed.
- Lenguaje Assembly
 - Apunte 4 de cátedra
- Simulador MSX88
 - En Descargas de página web de cátedra
<http://weblidi.info.unlp.edu.ar/catedras/organizacion/>