Grupo ARCOS

uc3m | Universidad Carlos III de Madrid

Tema 1
Introducción a los computadores

Estructura de Computadores
Grado en Ingeniería Informática

Contenidos

1. ¿Qué es un computador?
2. Concepto de estructura y arquitectura
3. Elementos constructivos de un computador
4. Computador Von Neumann
5. Instrucciones máquina y programación
6. Fases de ejecución de una instrucción
7. Parámetros característicos de un computador
8. Tipos de computadores
9. Evolución histórica

¿Qué aspecto tiene un computador?

¿   ?

¿Qué aspecto tiene un computador?





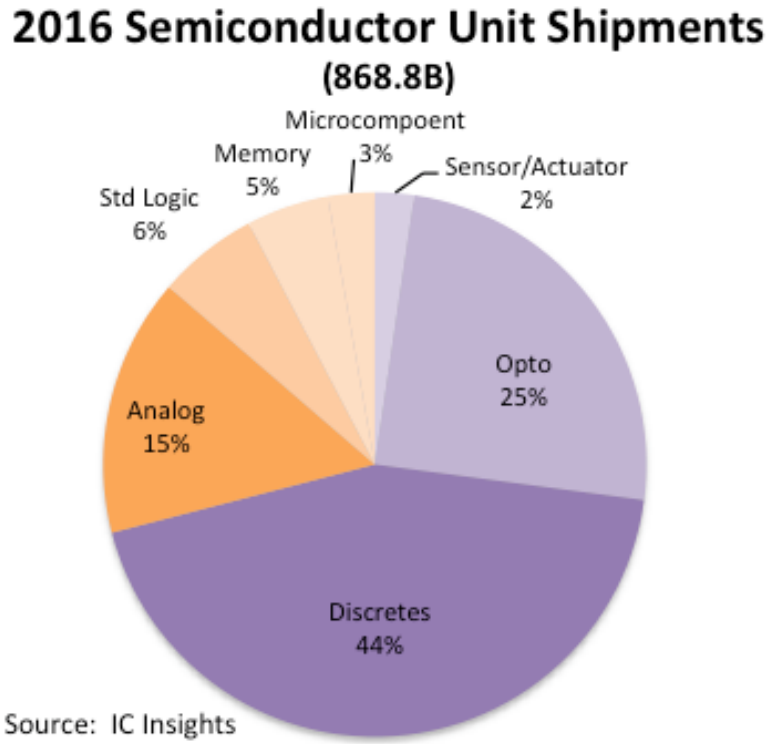






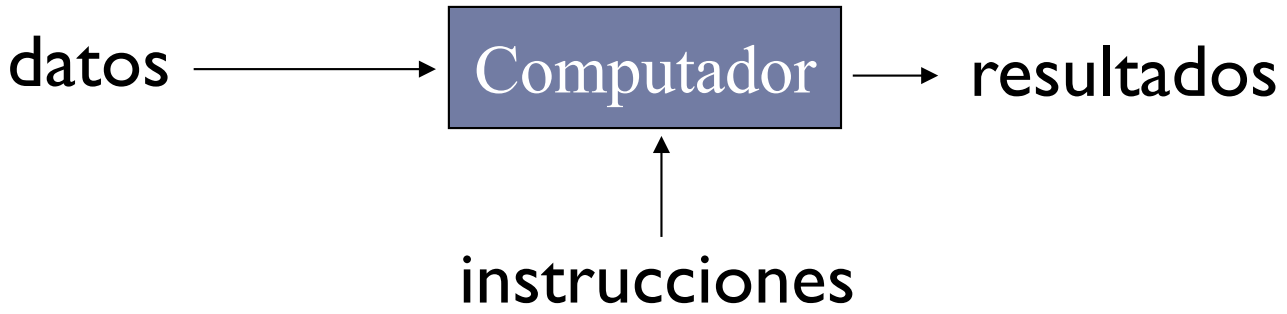


Industria de los semiconductores



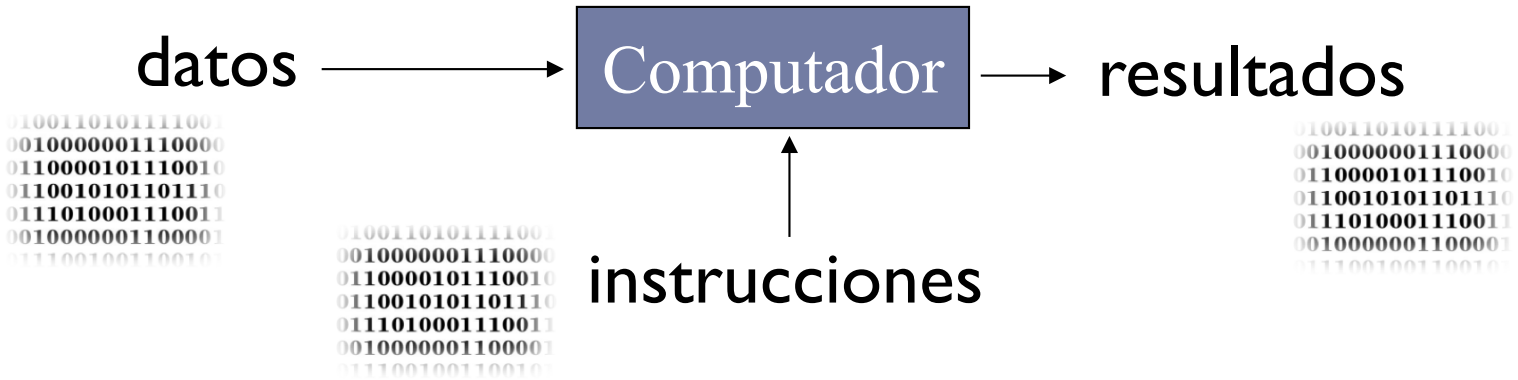
Procesadores:
3% de la industria

¿Qué es un computador?



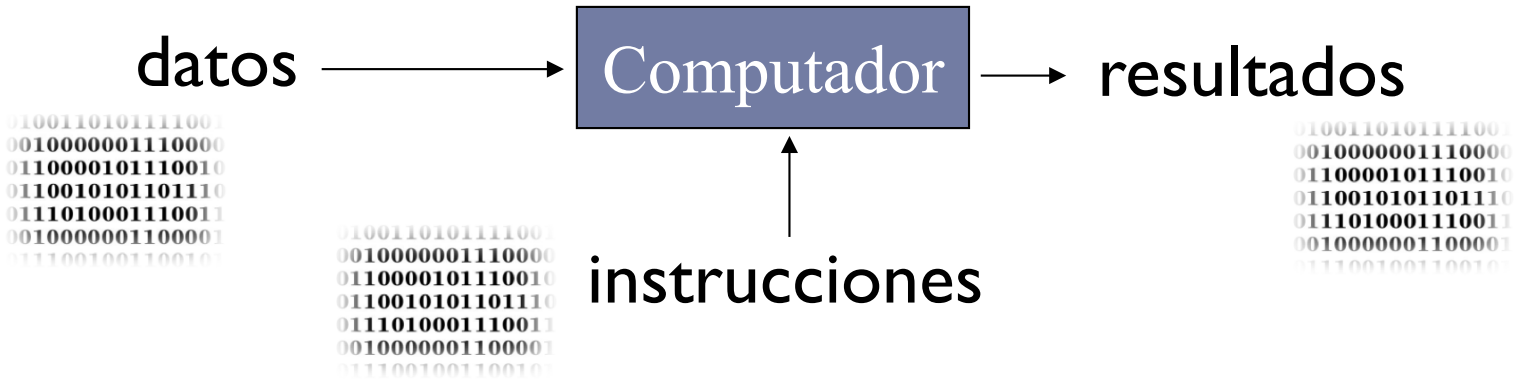
- **Computador: máquina destinada a procesar datos.**
  - Sobre ellos se aplican unas instrucciones obteniendo después unos resultados (datos/información)

¿Qué es un computador?



- **Computador: máquina destinada a procesar datos.**
  - Computador digital: datos e instrucciones en formato binario.

¿Qué es un computador?



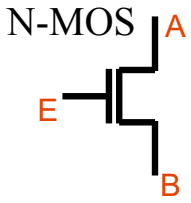
- **Computador: máquina destinada a procesar datos.**
  - Computador digital: datos e instrucciones en formato binario.
  - Matemáticamente se puede representar como:

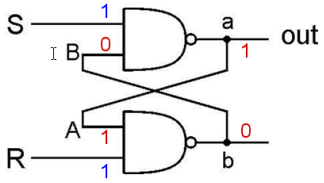
$$f : \{0,1\}^n \rightarrow \{0,1\}^m$$

Contenidos

1. ¿Qué es un computador?
2. Concepto de estructura y arquitectura
3. Elementos constructivos de un computador
4. Computador Von Neumann
5. Instrucciones máquina y programación
6. Fases de ejecución de una instrucción
7. Parámetros característicos de un computador
8. Tipos de computadores
9. Evolución histórica

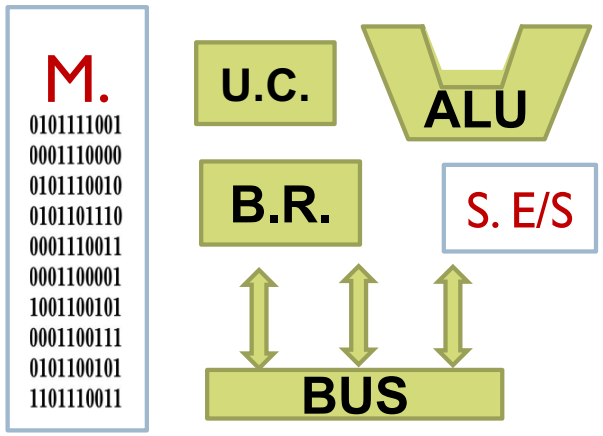
¿Qué aspectos hay que conocer en un computador?

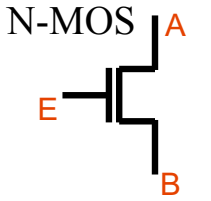


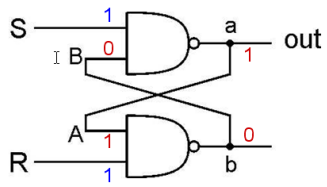


- **Tecnología:**
  - Cómo se construyen los componentes

¿Qué aspectos hay que conocer en un computador?



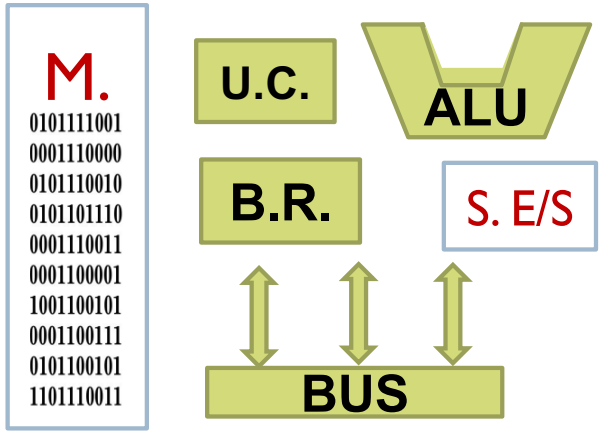




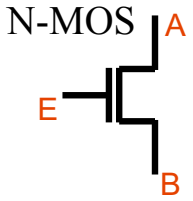
- **Estructura:**
  - Componentes y su organización
- **Tecnología:**
  - Cómo se construyen los componentes

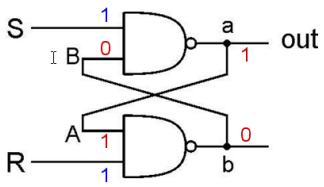
¿Qué aspectos hay que conocer en un computador?

- **Arquitectura:**
  - Atributos visibles para un programador



- **Estructura:**
  - Componentes y su organización





- **Tecnología:**
  - Cómo se construyen los componentes

Arquitectura de un computador

- Atributos visibles para un programador
  - Juego de instrucciones que ofrece la máquina (ISA, Instruction Set Architecture)
  - Tipo y formato de datos que es capaz de utilizar el computador
  - Número y tamaño de los registros
  - Técnicas y mecanismos de E/S
  - Técnicas de direccionamiento de la memoria

Ejercicio

- ¿Qué es un computador?
- ¿Qué aspecto tiene un computador?
- ¿Qué aspectos de un computador se han de conocer?

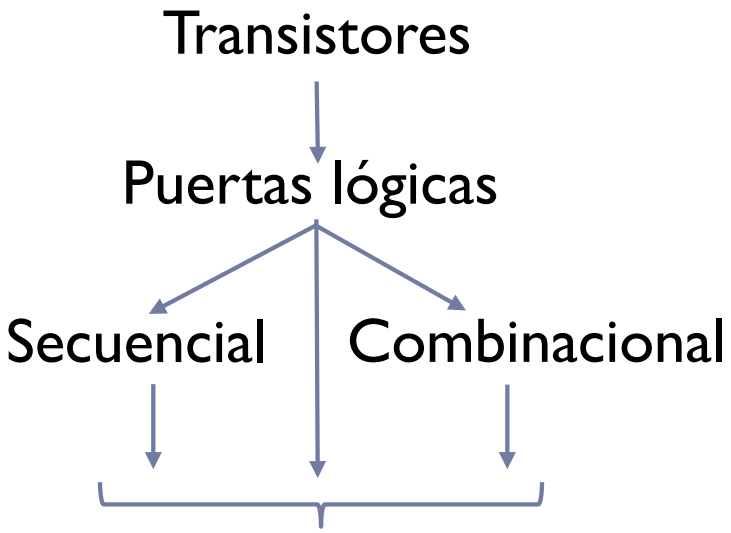
Contenidos

1. ¿Qué es un computador?
2. Concepto de estructura y arquitectura
3. Elementos constructivos de un computador
4. Computador Von Neumann
5. Instrucciones máquina y programación
6. Fases de ejecución de una instrucción
7. Parámetros característicos de un computador
8. Tipos de computadores
9. Evolución histórica

Repaso

- Sistema digital basado en: 0 y 1

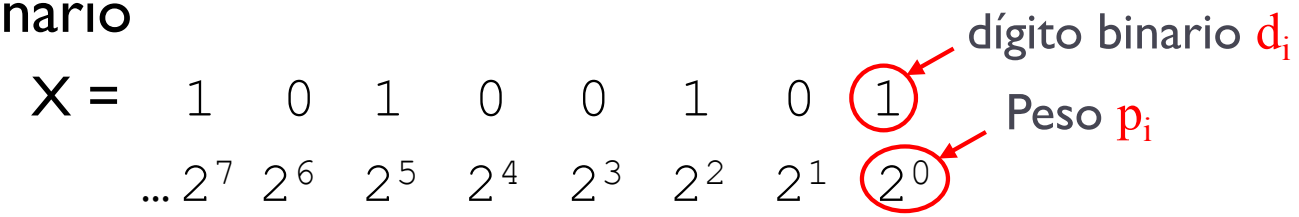
- Elementos constructivos: transistores, puertas lógicas, …:



  - Registros
  - …

Sistema binario

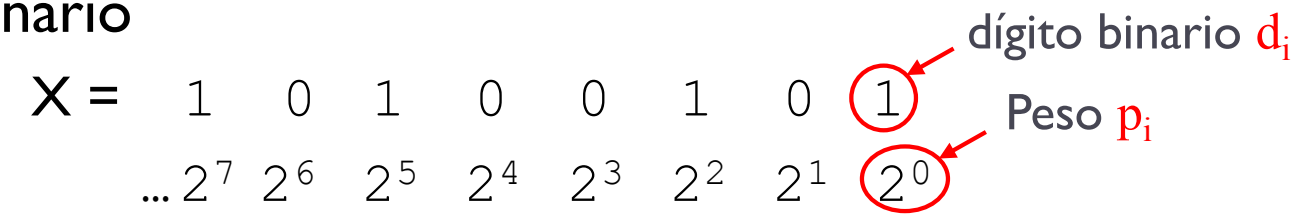
- Binario



  - Valor = $d_{31} \times 2^{31} + d_{30} \times 2^{30} + \ldots + d_1 \times 2^1 + d_0 \times 2^0$

Sistema binario

- Binario



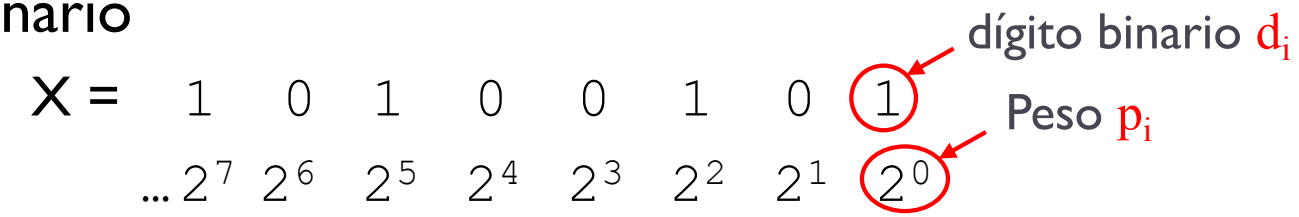
  - Valor = $d_{31} \times 2^{31} + d_{30} \times 2^{30} + \ldots + d_1 \times 2^1 + d_0 \times 2^0$

  - ¿Cuántos valores se pueden representar con n bits?
  - ¿Cuántos bits se necesitan para representar m 'valores'?

  - Con n bits, si los valores a representar son números y comienzo en el 0, ¿Cuál es el máximo valor representable?

Sistema binario

- Binario



  - Valor = $d_{31} \times 2^{31} + d_{30} \times 2^{30} + \ldots + d_1 \times 2^1 + d_0 \times 2^0$

  - ¿Cuántos valores se pueden representar con n bits? **2^n**
  - ¿Cuántos bits se necesitan para representar m 'valores'? **$Log_2(m)$ por exceso**
  - Con n bits, si los valores a representar son números y comienzo en el 0, ¿Cuál es el máximo valor representable? **2^n-1**

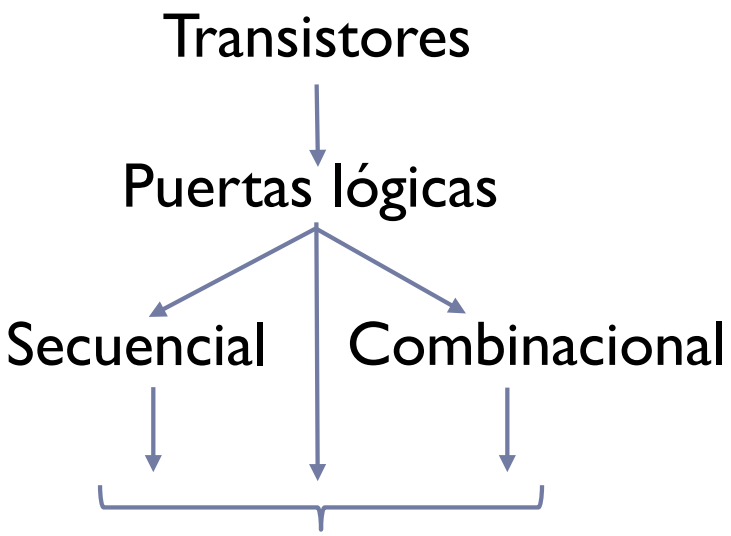
Ejercicio

- ¿Cuántos códigos distintos se pueden codificar con 8 bits?
- ¿Cuántos bits hacen falta para representar 512 códigos?

Repaso

- Sistema digital basado en: 0 y 1

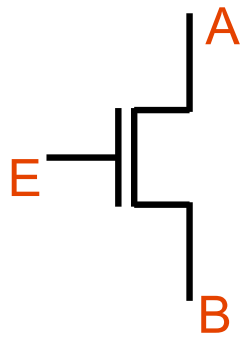
- Elementos constructivos: transistores, puertas lógicas, …:



  - Registros
  - …

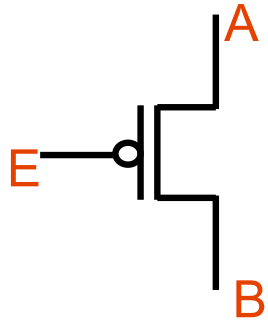
Transistor

N-MOS



E	Funcionamiento
1	Conecta A con B (circuito abierto)
0	No conecta A con B (circuito cerrado)

P-MOS

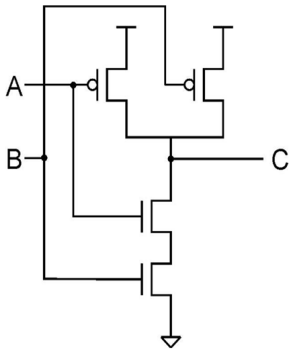


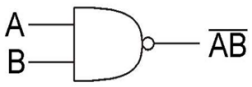
E	Funcionamiento
0	Conecta A con B (circuito abierto)
1	No conecta A con B (circuito cerrado)

- Un transistor actúa como un interruptor
- Los transistores tipo p y n son transistores de tipo MOSFET (Metal-Oxide-Semiconductor-Field-Effect Transistor)
- La combinación de transistores tipo p y n dan lugar a la familia CMOS

Puertas lógicas

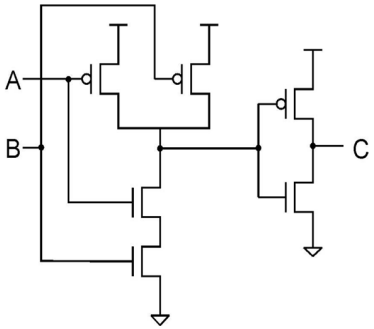
NAND

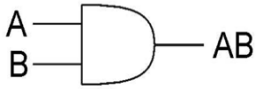




A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

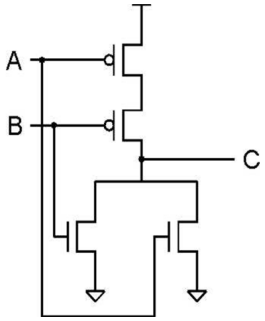
AND

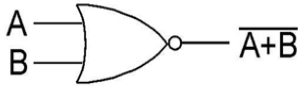




A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

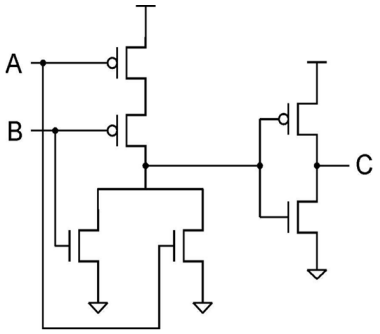
NOR

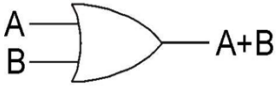




A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

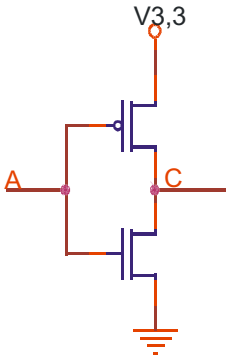
OR

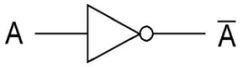




A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

NOT





A	C
1	0
0	1

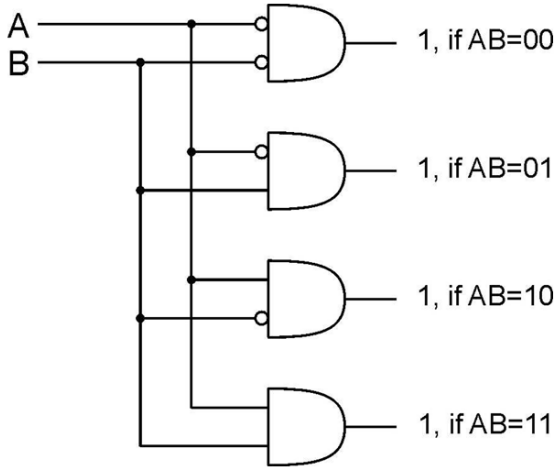
Circuitos combinacionales

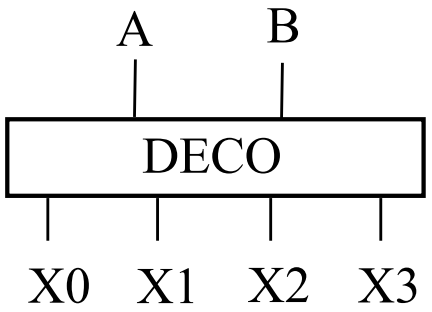
- La salida depende solo de los valores de entrada
- Ejemplos:
  - Decodifcadores
  - Multiplexores
  - Operadores aritméticos y lógicos

Decodificadores

- Transforma un valor codificado en la activación de una señal de salida
  - Los codificadores realizan el proceso inverso

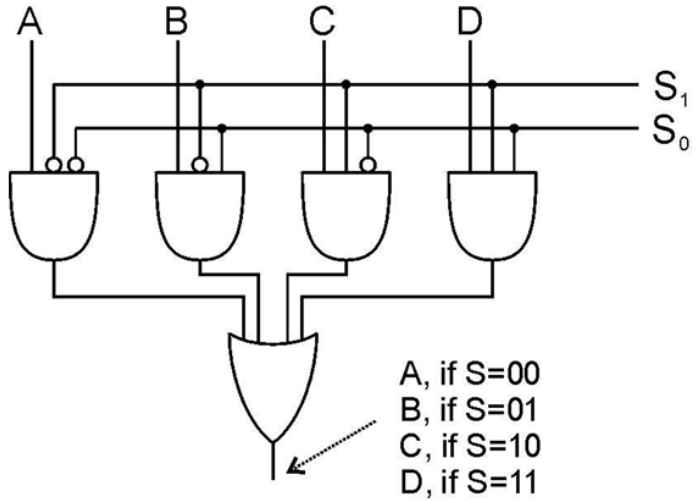
n entradas, 2^n salidas



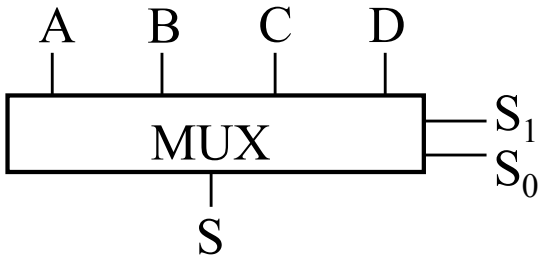


Multiplexores

- Selecciona una de las entradas y copia su valor a la salida
  - Los demultiplexores realizan el proceso inverso
- Con N entradas se necesitan $\log_2 N$ señales de control

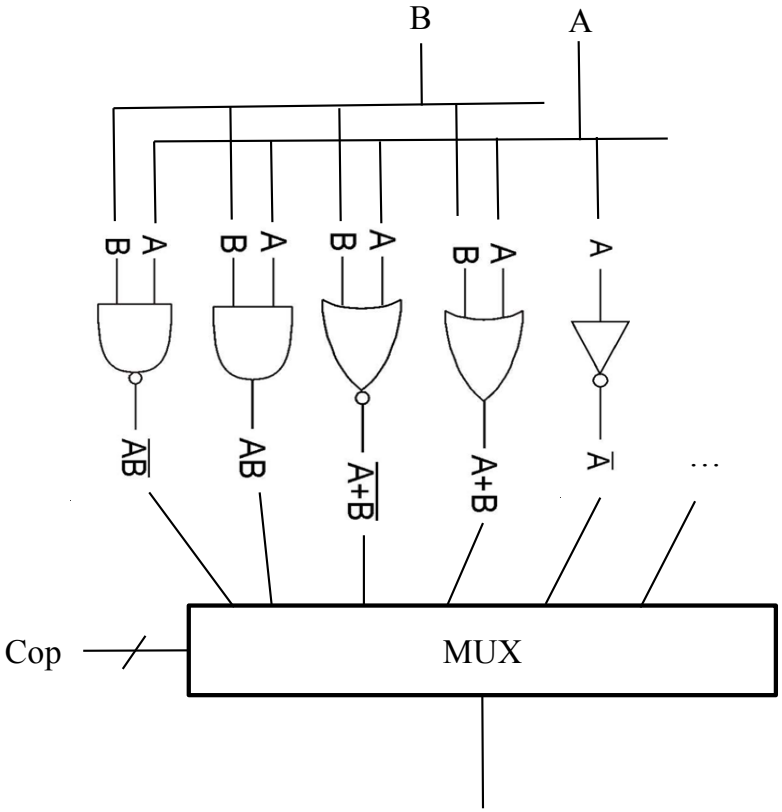


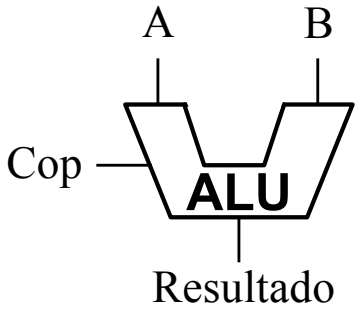
Selecciona con n bits entre 2^n entradas



ALU. Unidades aritmético-lógicas

- Realiza una operación aritmético-lógica



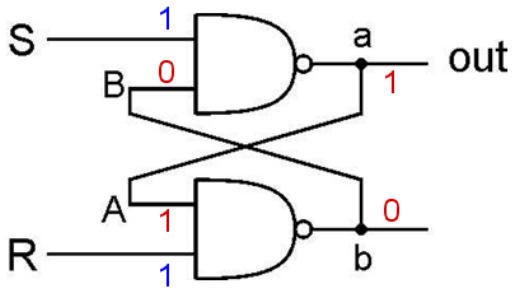


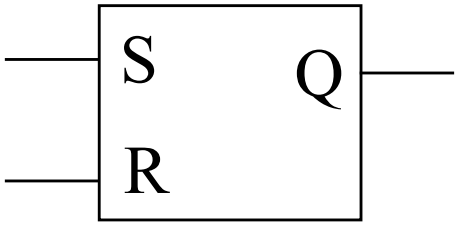
Circuitos secuenciales

- La salida depende de los valores de entrada y del estado actual
  - Necesitan almacenar estado

BIESTABLE R-S

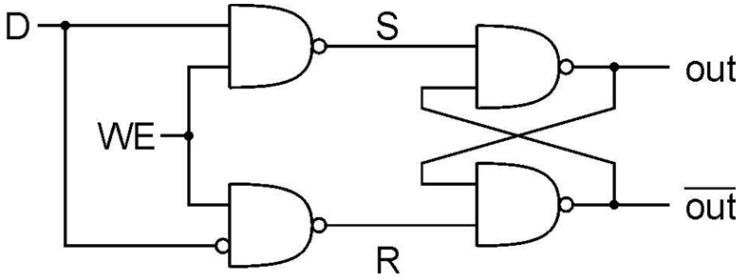
Almacena un bit

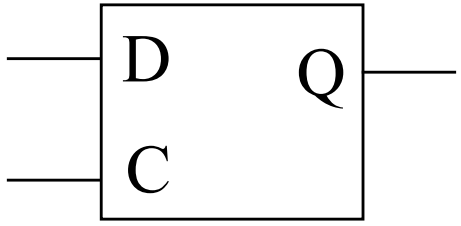




BIESTABLE D

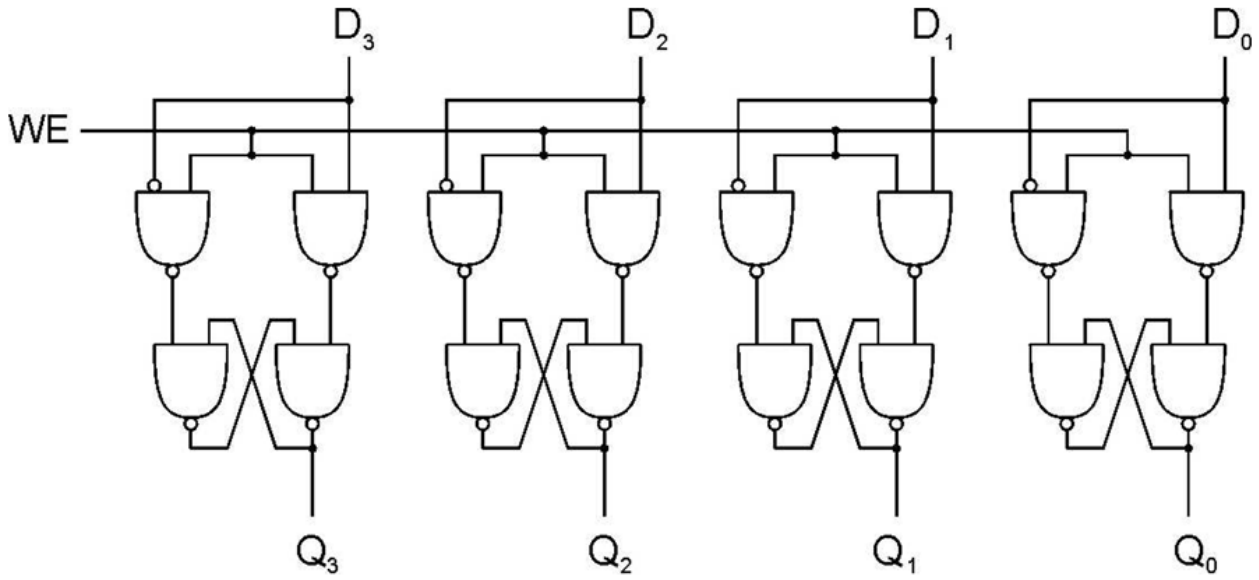
Almacena un bit

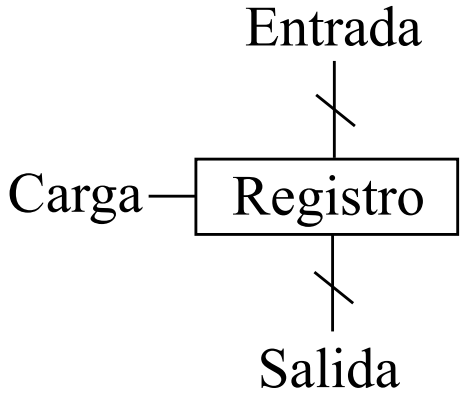




Registro

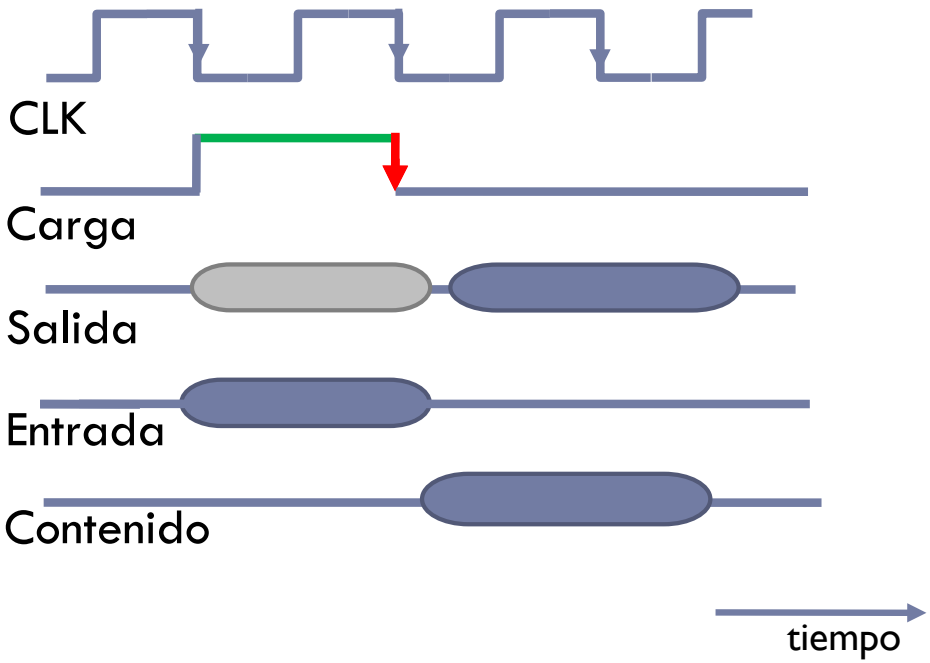
- Elemento que almacena n bits (a la vez)

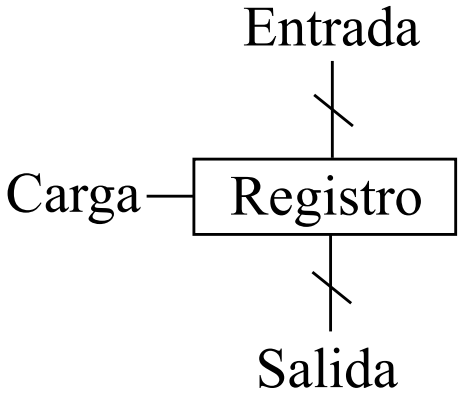




Registro

- Elemento que almacena n bits (a la vez)
  - Durante el nivel de Carga el registro tiene el valor antiguo
  - En el flanco de Carga se almacena el valor en la entrada

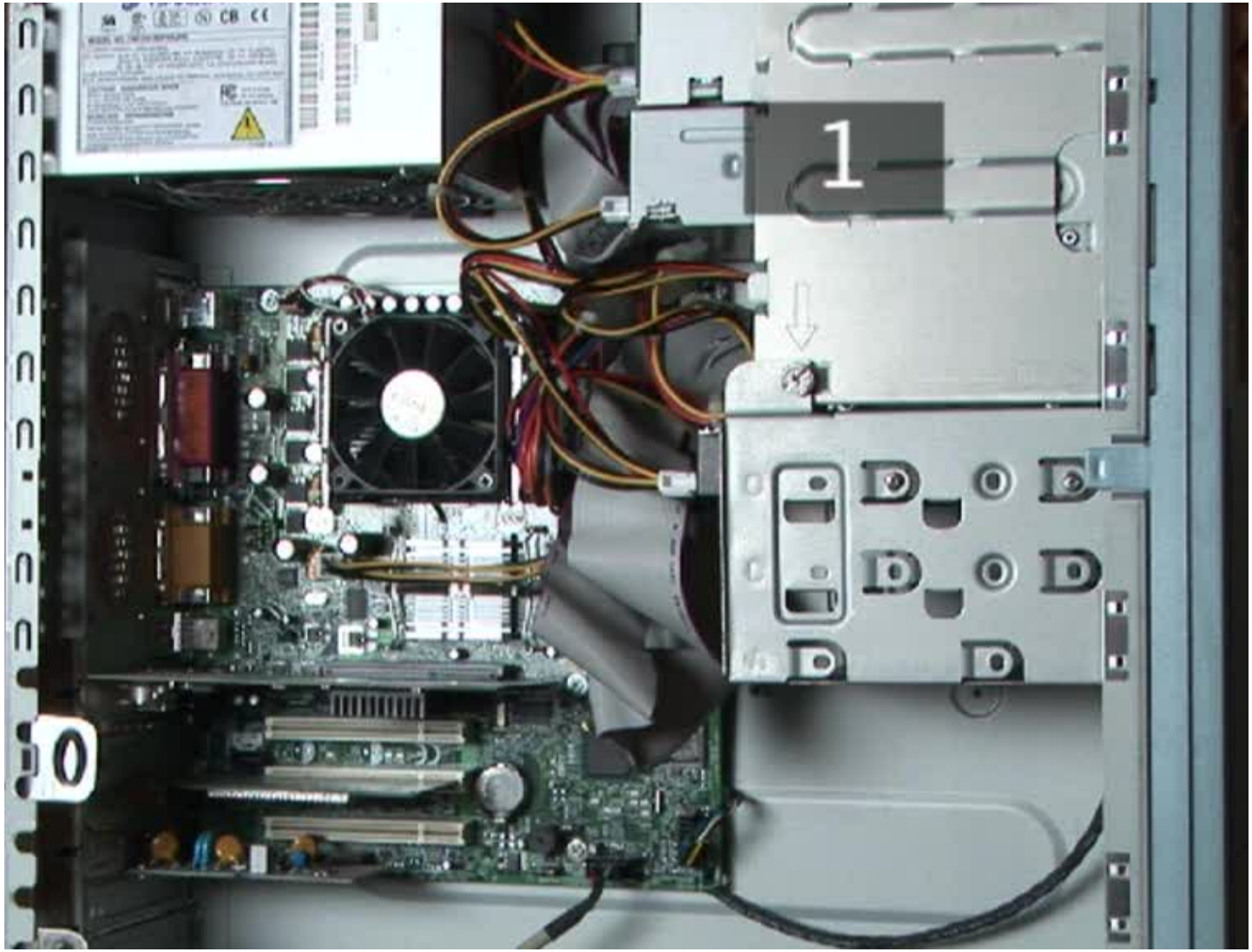




Contenidos

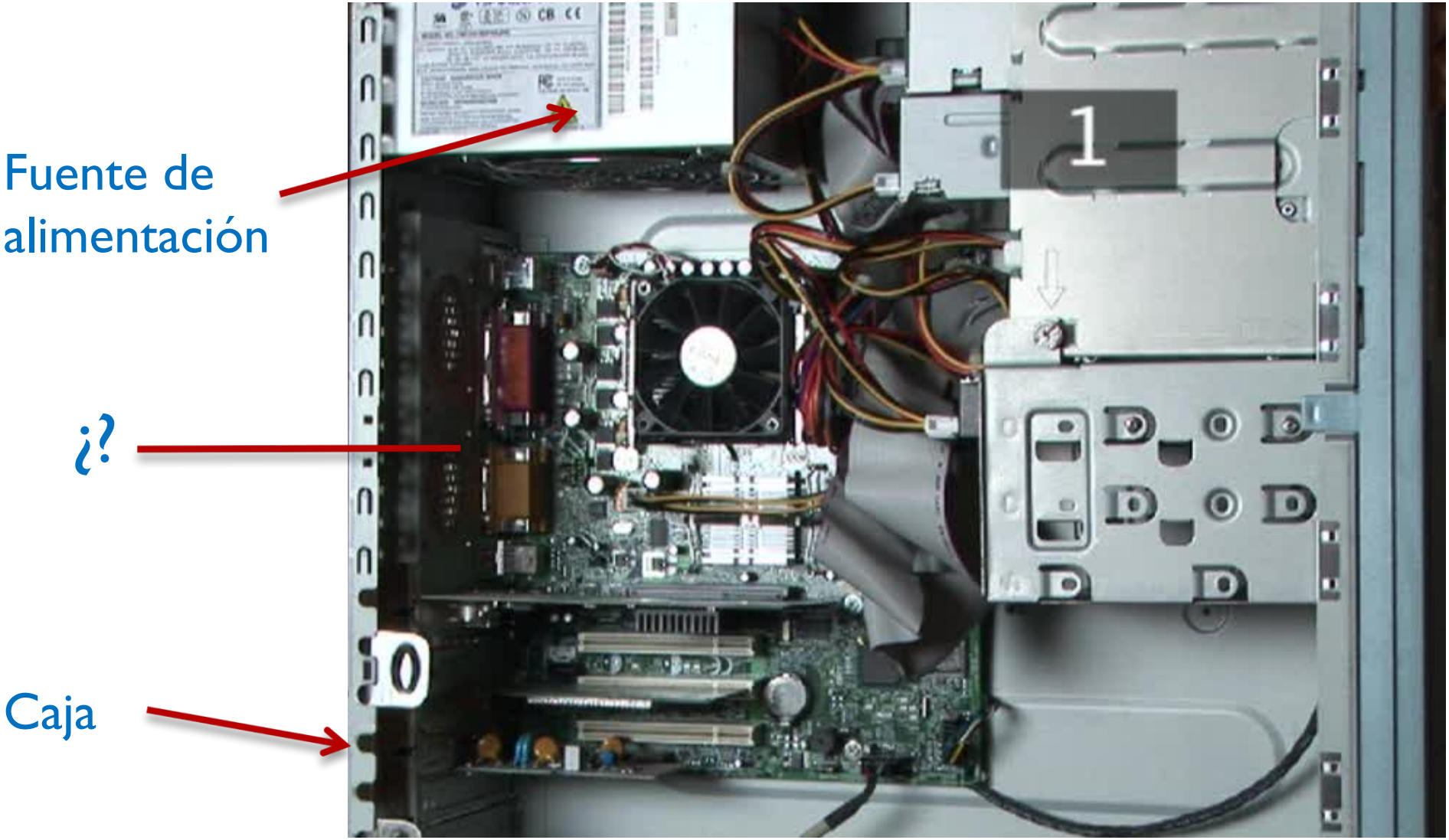
1. ¿Qué es un computador?
2. Concepto de estructura y arquitectura
3. Elementos constructivos de un computador
4. Computador Von Neumann
5. Instrucciones máquina y programación
6. Fases de ejecución de una instrucción
7. Parámetros característicos de un computador
8. Tipos de computadores
9. Evolución histórica

¿Podemos distinguir los componentes internos al abrir un ordenador personal?



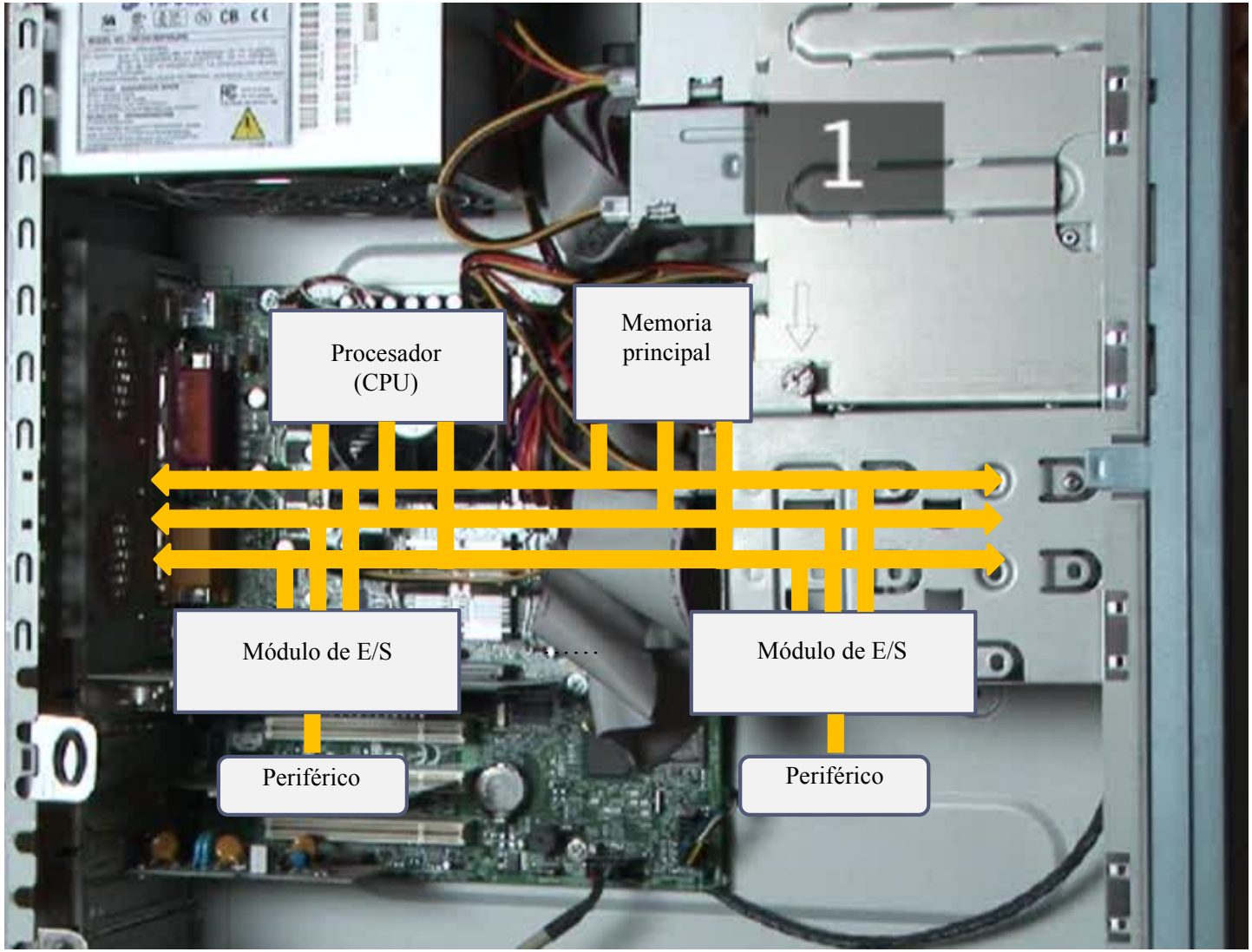
http://www.videojug.com/film/what-components-are-inside-my-computer

¿Podemos distinguir los componentes internos al abrir un ordenador personal?



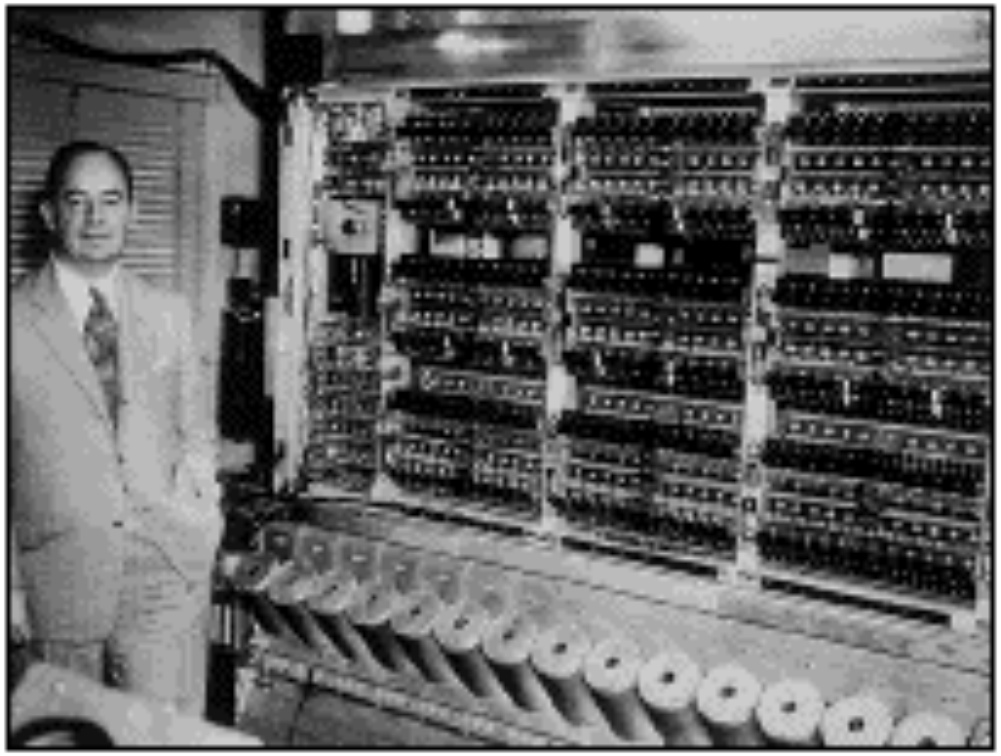
http://www.videojug.com/film/what-components-are-inside-my-computer

Modelo usado como base



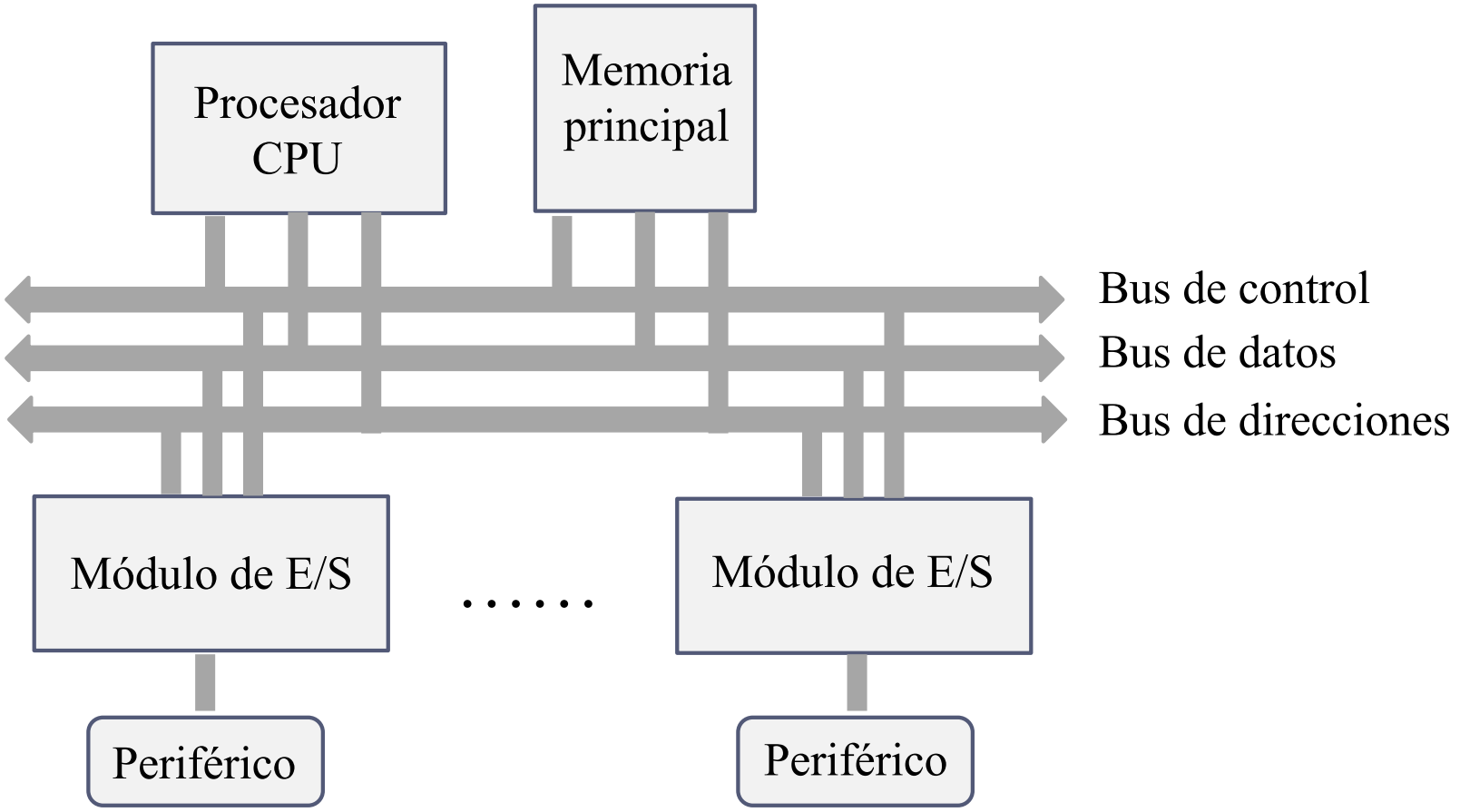
http://www.videojug.com/film/what-components-are-inside-my-computer

Computador Von Neumann

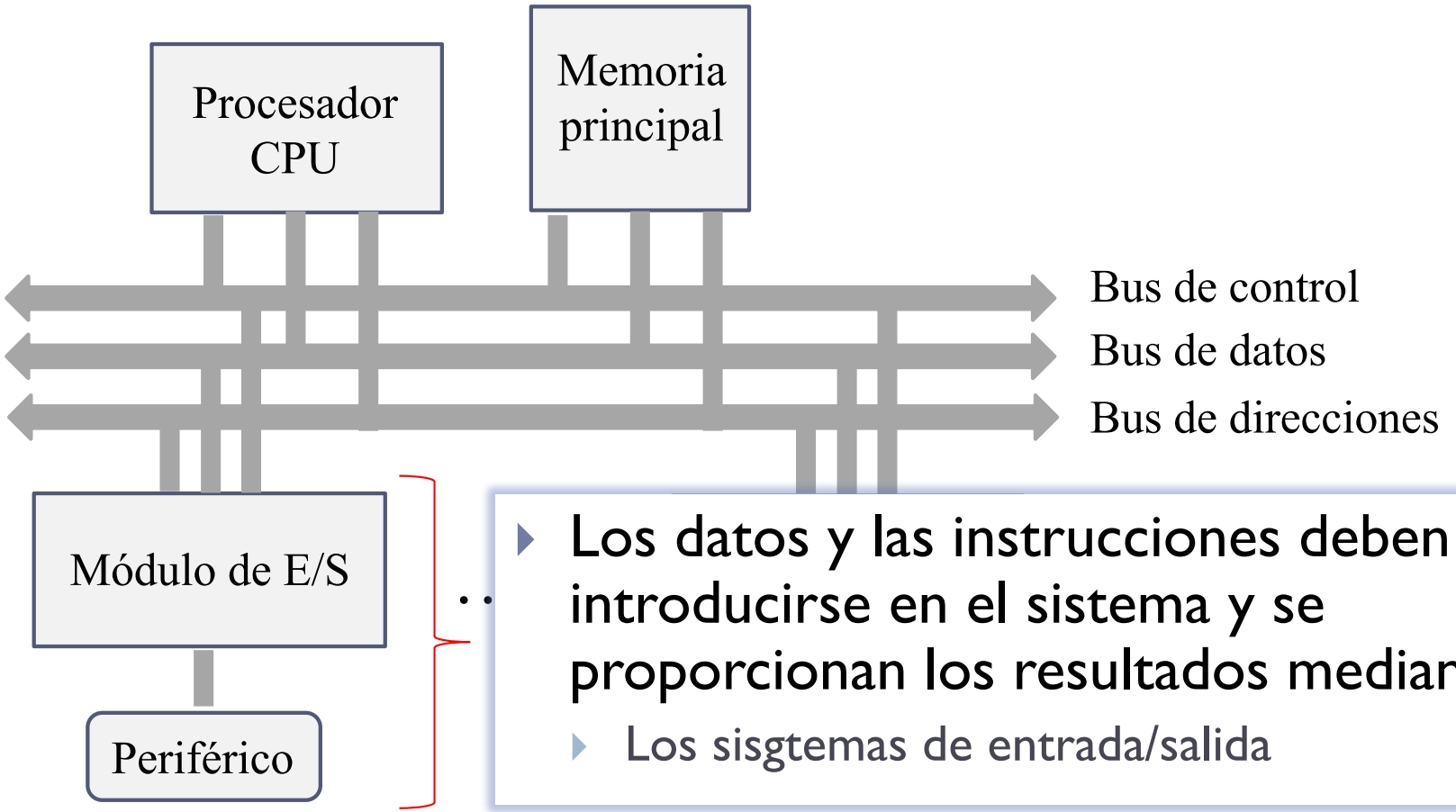


Máquina capaz de ejecutar una serie de instrucciones elementales (instrucciones máquina) que están almacenadas en memoria (son leídas y ejecutadas)

Arquitectura Von Neumann

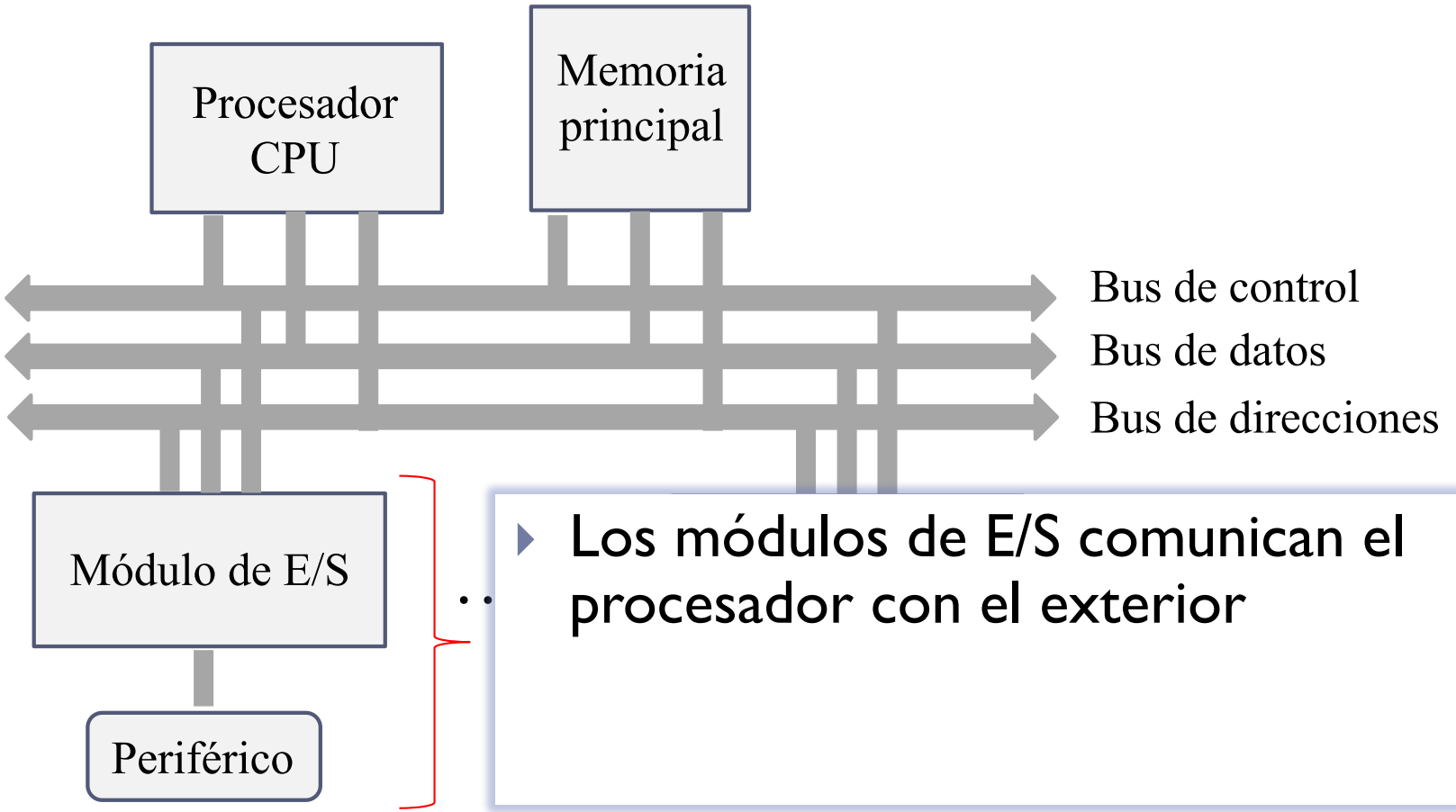


Arquitectura Von Neumann (1/4)



- Los datos y las instrucciones deben introducirse en el sistema y se proporcionan los resultados mediante:
  - Los sisgtemas de entrada/salida

Arquitectura Von Neumann (1/4)

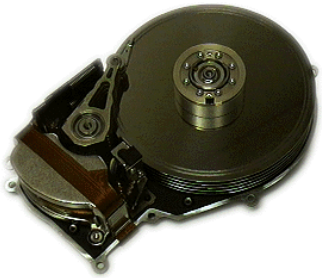


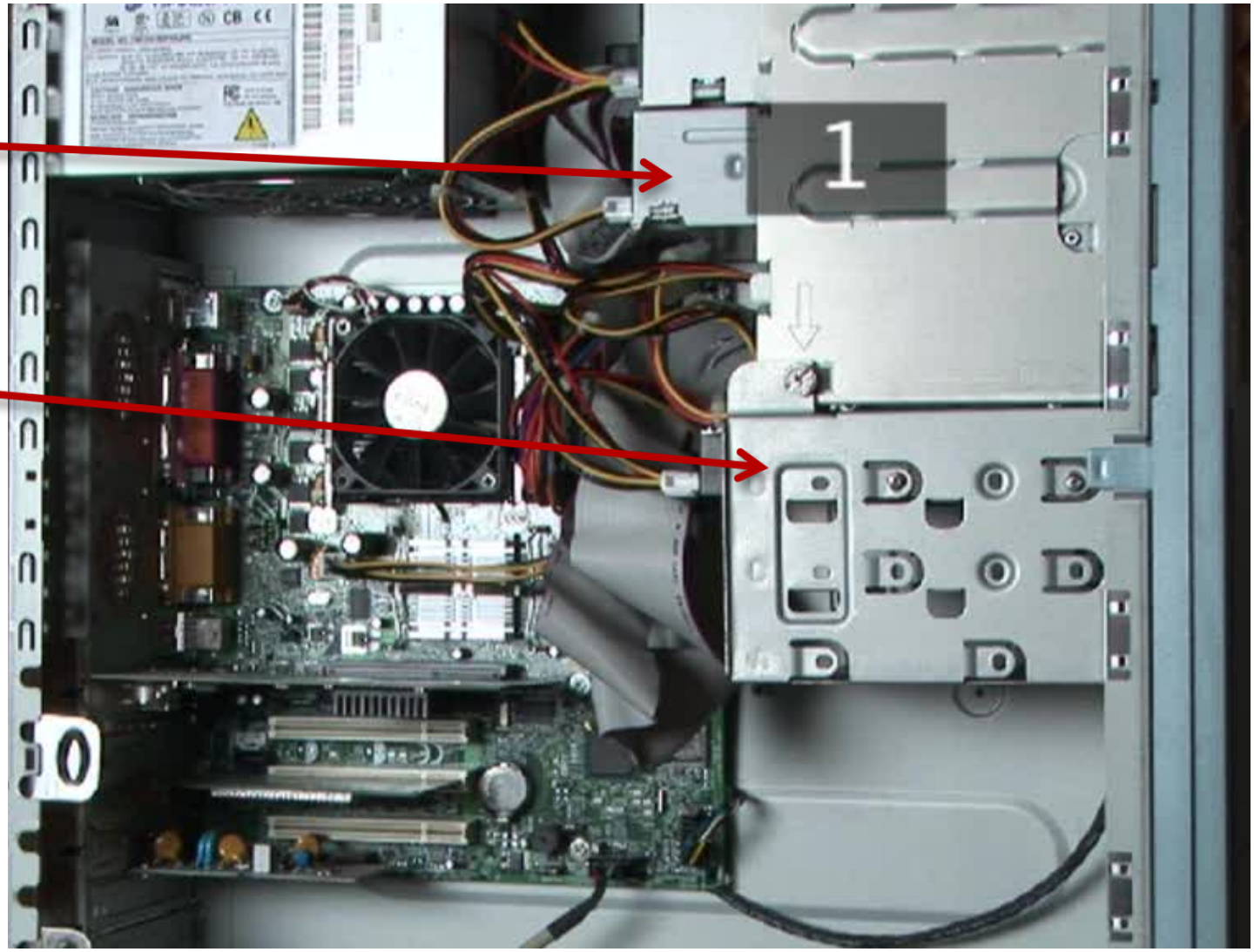
- Los módulos de E/S comunican el procesador con el exterior

Ejemplo de módulos + periféricos
almacenamiento

CD-ROM/
DVD-ROM/
BluRay/…

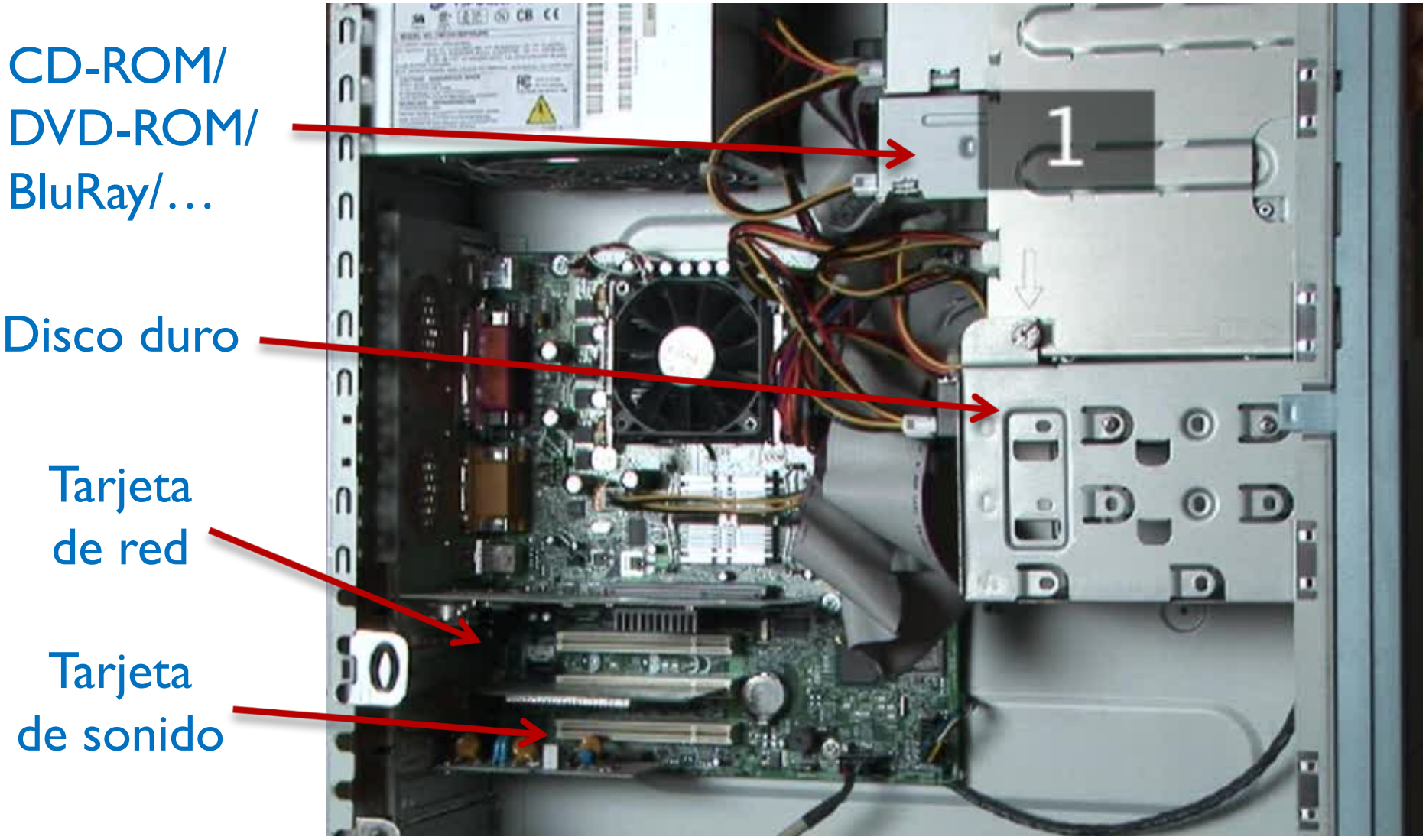
Disco duro





http://www.videojug.com/film/what-components-are-inside-my-computer

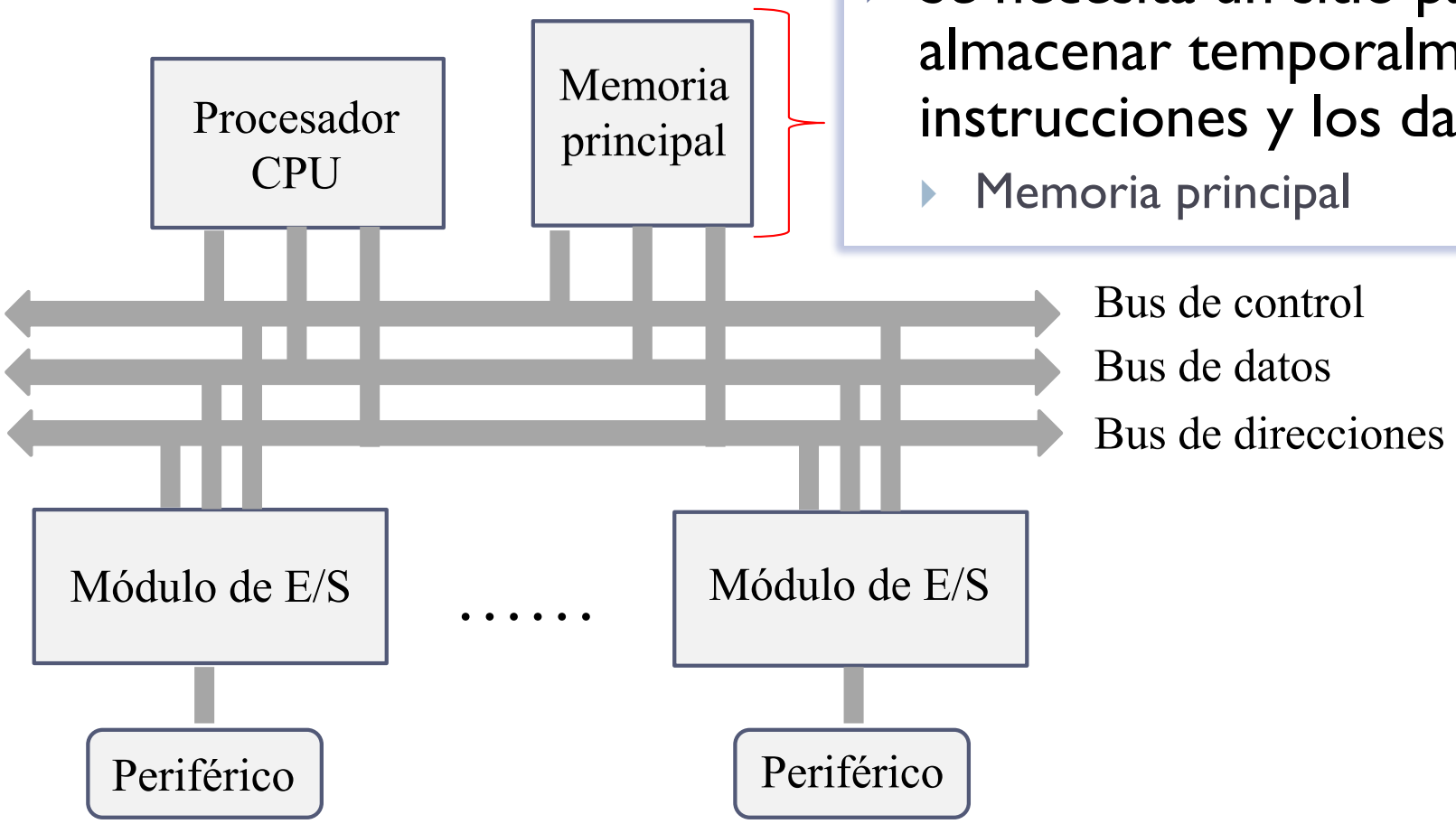
Ejemplo de módulos + periféricos
comunicación



http://www.videojug.com/film/what-components-are-inside-my-computer

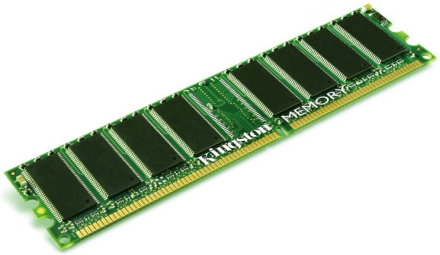
Arquitectura Von Neumann (2/4)

- Se necesita un sitio para almacenar temporalmente las instrucciones y los datos:
  - Memoria principal



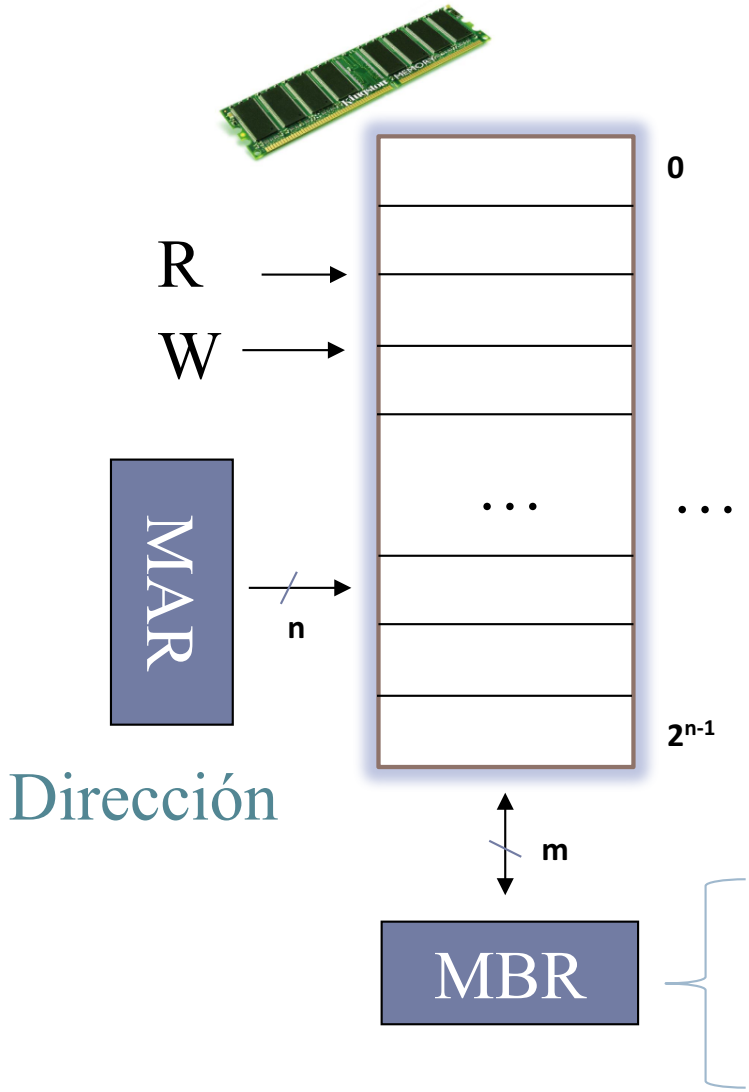
Ejemplo de memoria principal

Memoria principal



http://www.videojug.com/film/what-components-are-inside-my-computer

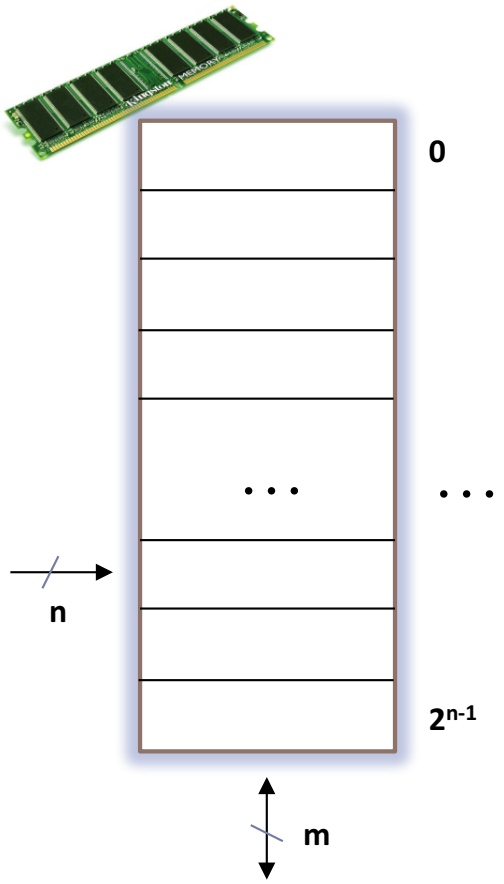
Elementos de la memoria principal



- Registro de direcciones (MAR, Memory Address Register)
- Registro de datos (MBR, Memory Buffer Register)
- Señales de control
  - R- Lectura (Read)
  - W- Escritura (Write)

Espacio de direcciones vs. tamaño de palabra

Espacio de direcciones:
Número de posiciones

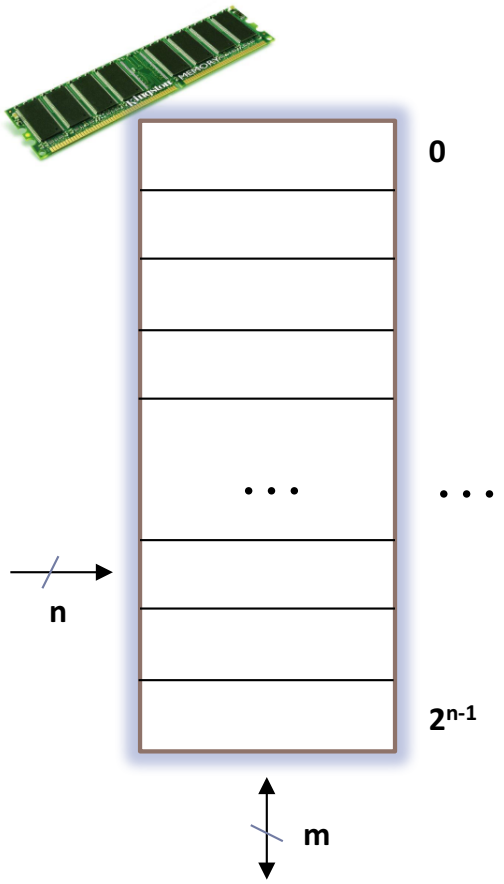


Tamaño de cada posición:
Número de bits por posición

Espacio de direcciones vs. tamaño de palabra

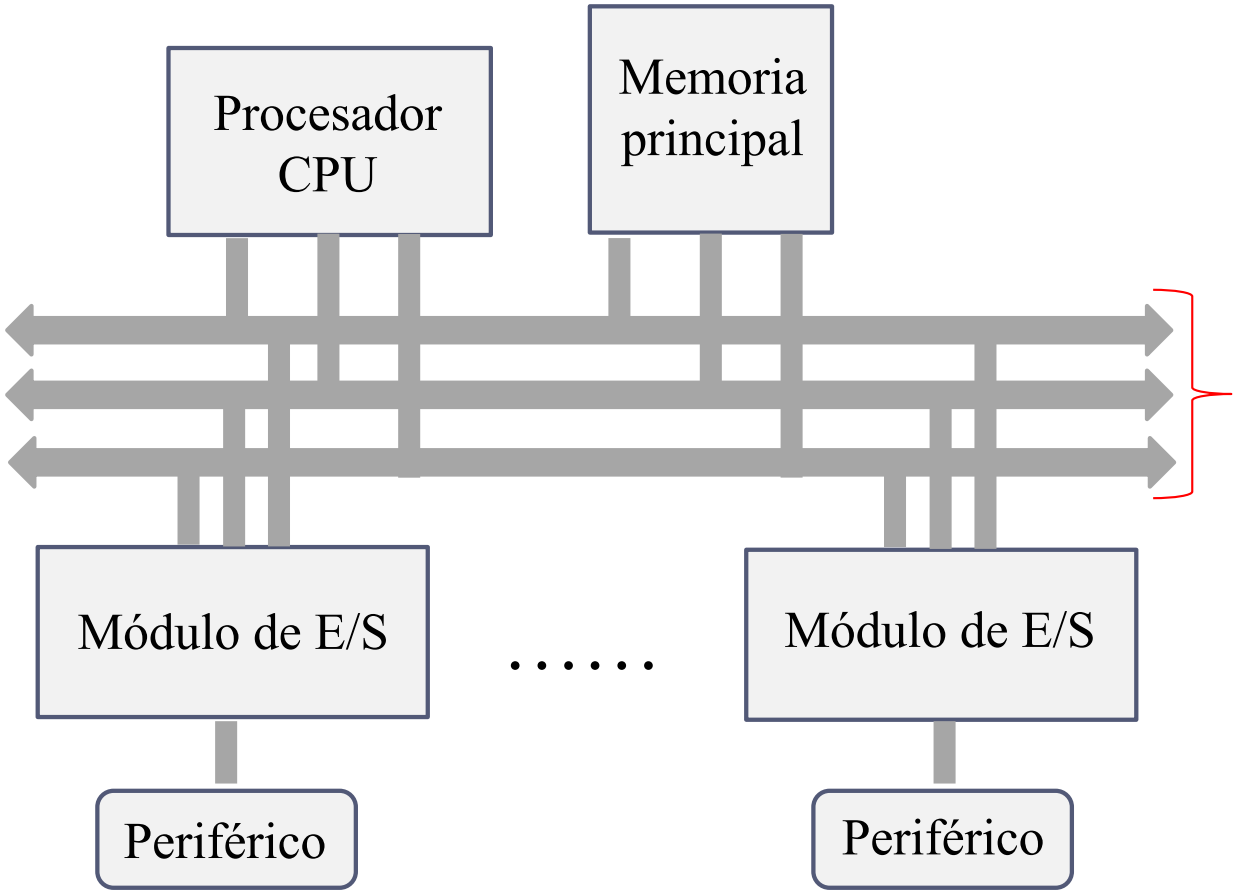
Espacio de direcciones:
Número de posiciones

2^n posiciones



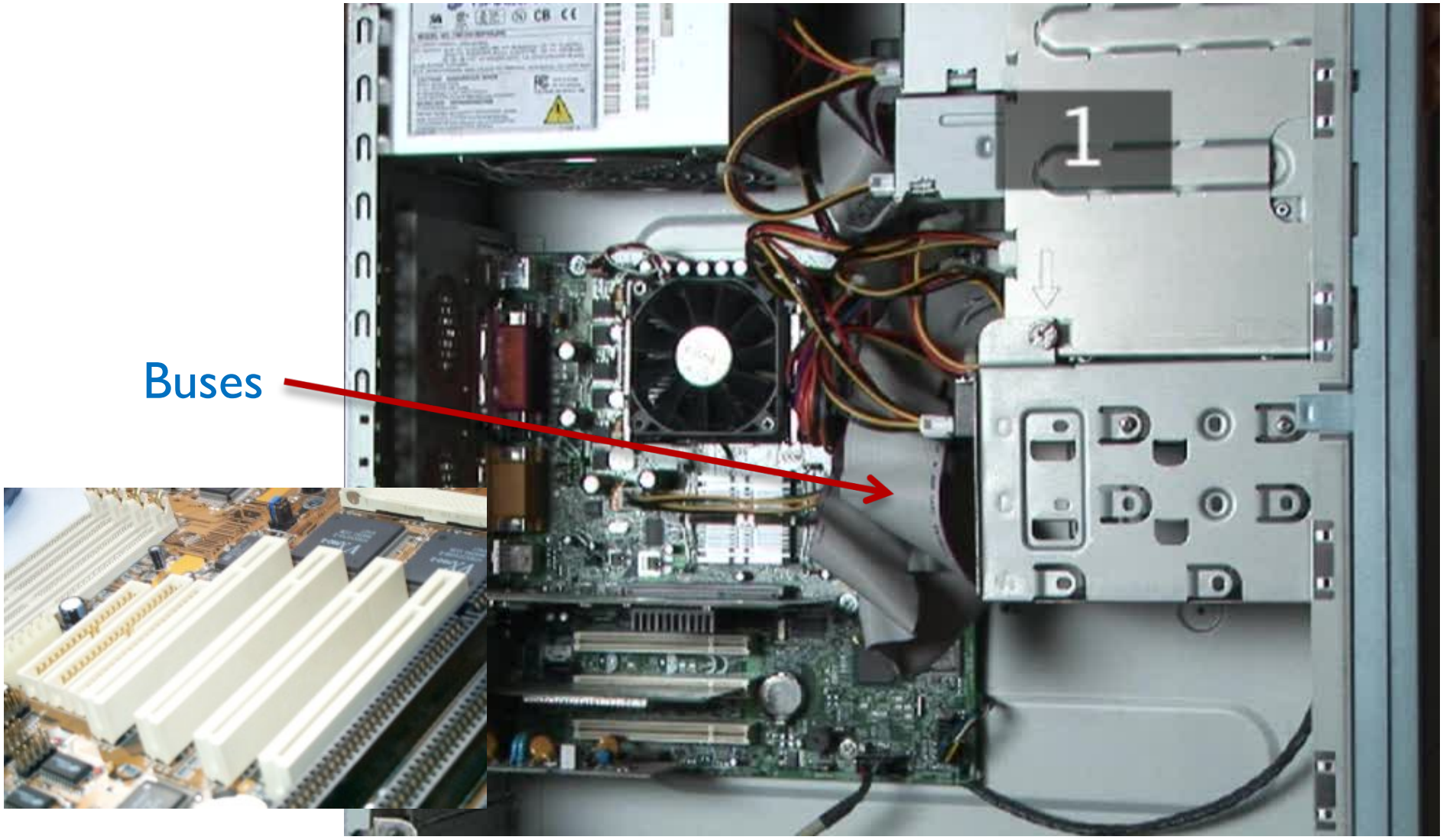
Tamaño de cada posición:
Número de bits por posición

Arquitectura Von Neumann (3/4)



- Es preciso comunicar las distintas partes del computador:
  - Buses

Ejemplo de buses

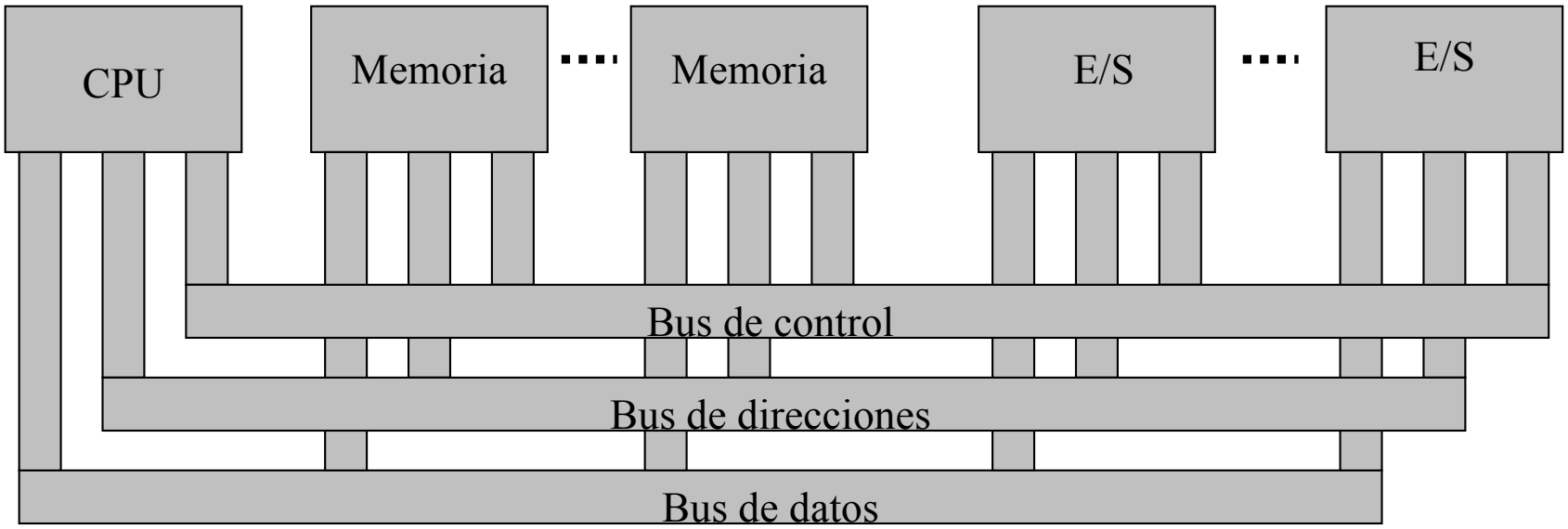


http://www.videojug.com/film/what-components-are-inside-my-computer

Buses

- Un bus es un camino de comunicación entre dos o más elementos (procesador, memoria, …) para la transmisión de información entre ellos.
- Un bus suele formarse por varias líneas de comunicación, cada una transmite un bit.
  - El ancho del bus representa el tamaño con el que trabaja el computador (ejemplo: bus de 32 bits)
- Tres tipos principales: datos, direcciones y control.

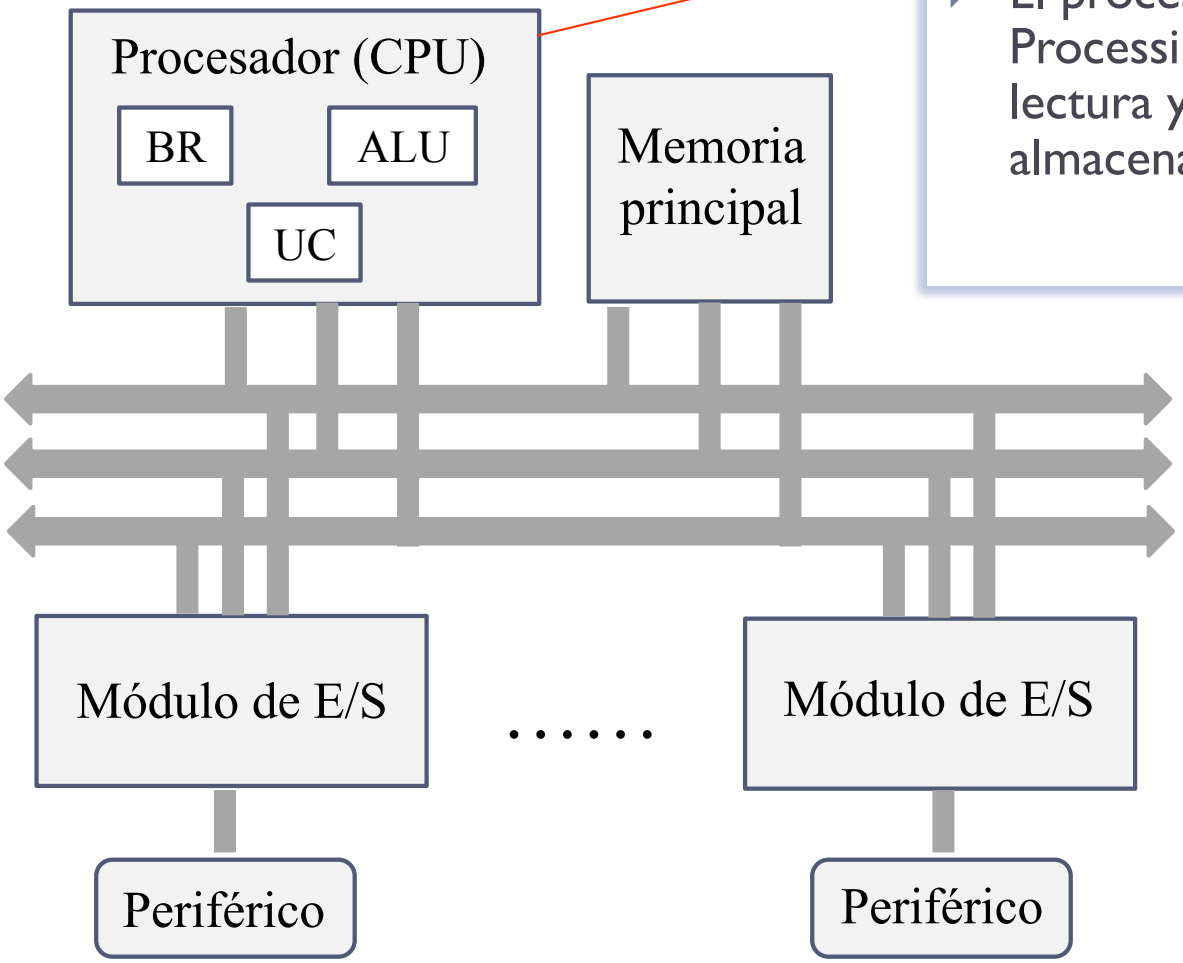
Esquema de interconexión de bus



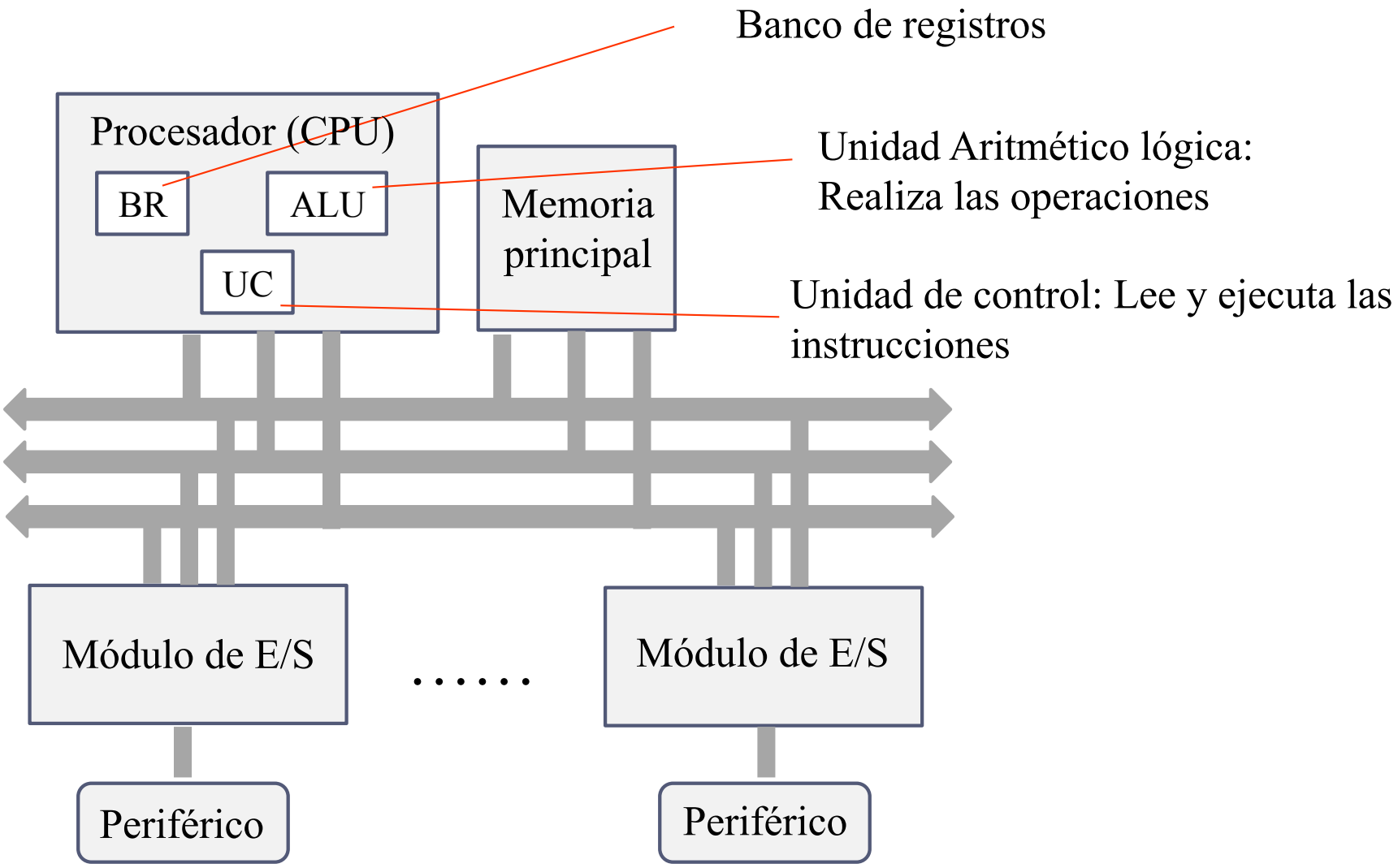
- Bus de control: señales de control y temporización
- Bus de direcciones: designa la fuente o destino de un dato
  - Su anchura determina la máxima capacidad de memoria del sistema
- Bus de datos: movimiento de datos entre componentes

Arquitectura Von Neumann (4/4)

- El procesador o CPU (Central Processing Unit) es el responsable de la lectura y ejecución de las instrucciones almacenadas en memoria principal.

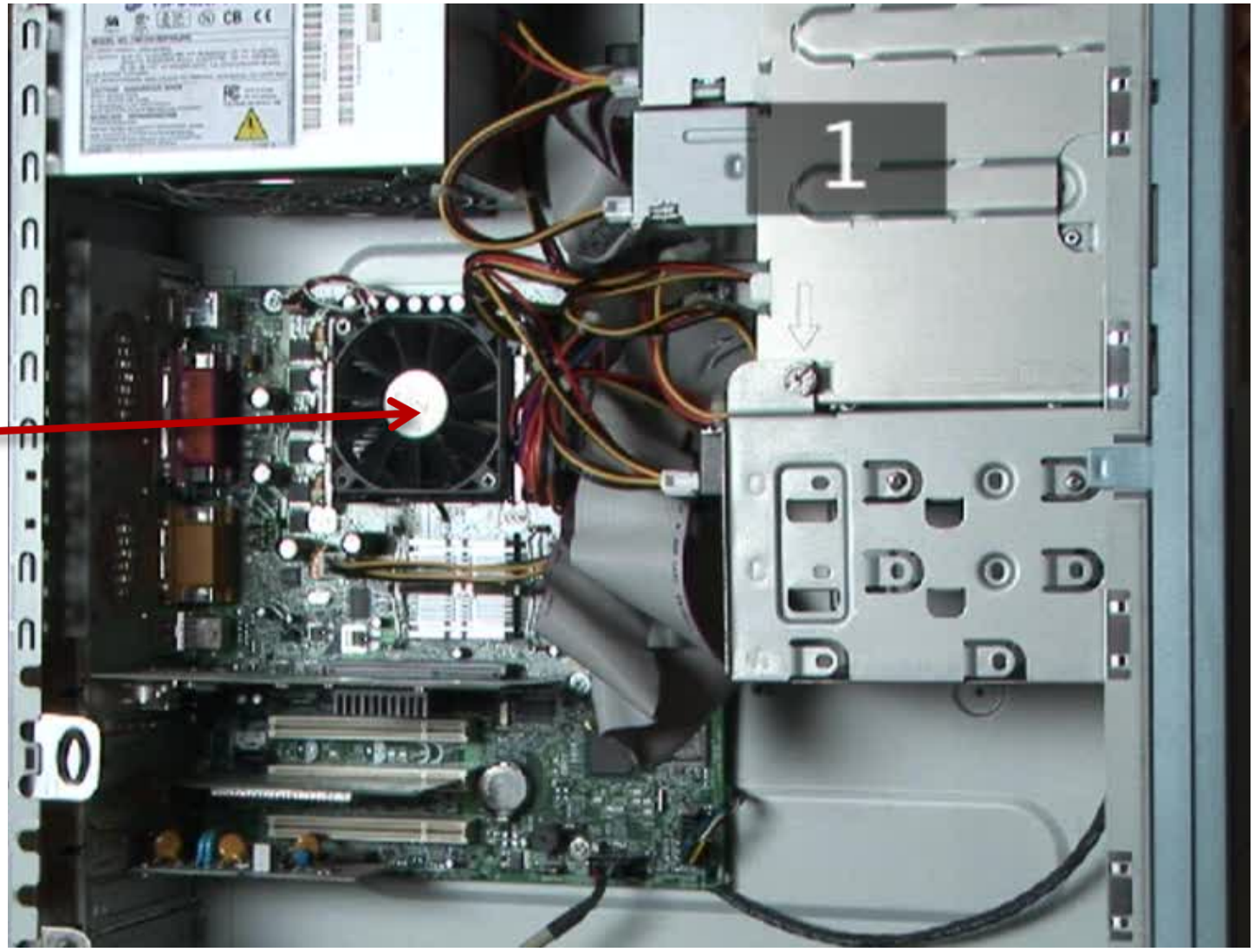


Arquitectura Von Neumann (4/4)



Ejemplo de CPU

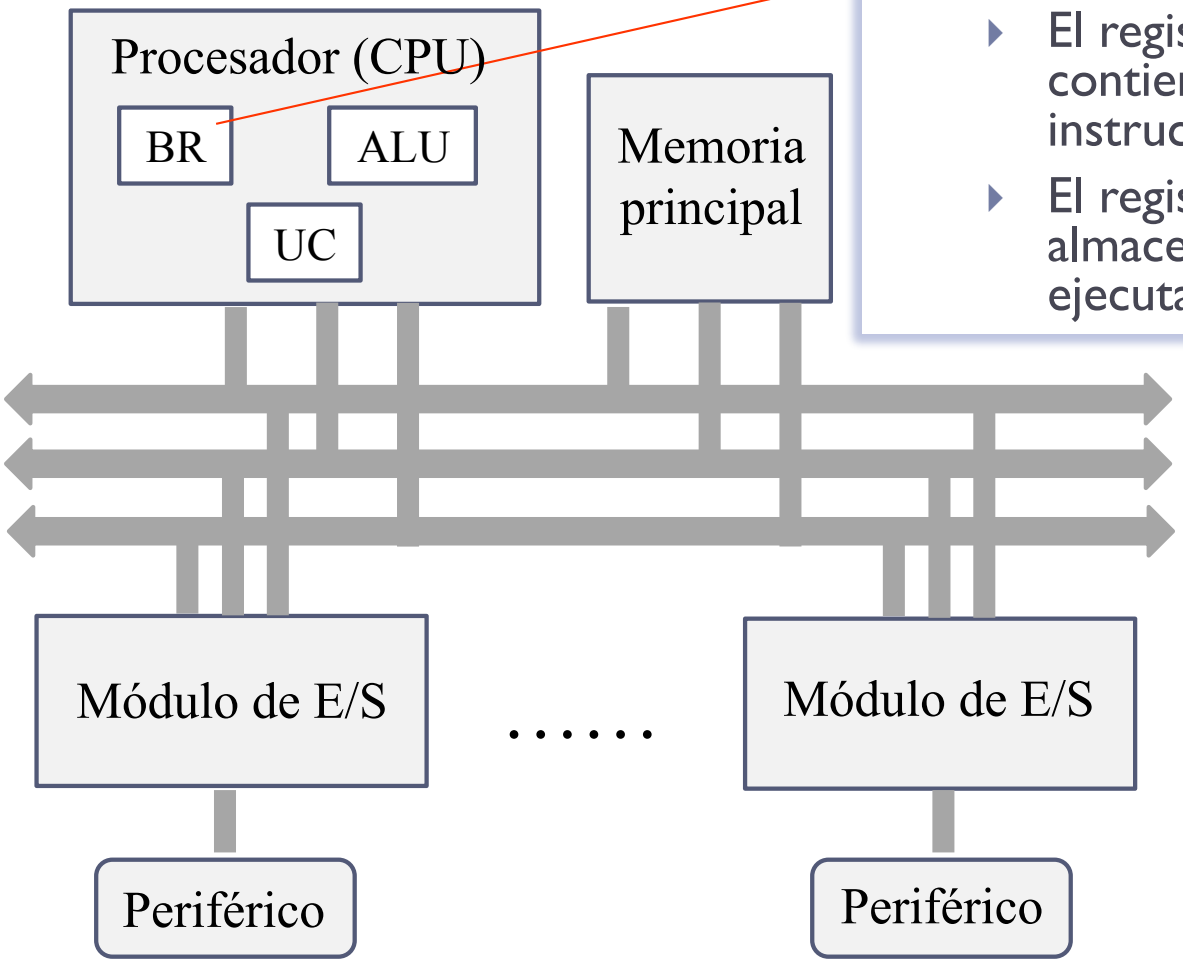
CPU



http://www.videojug.com/film/what-components-are-inside-my-computer

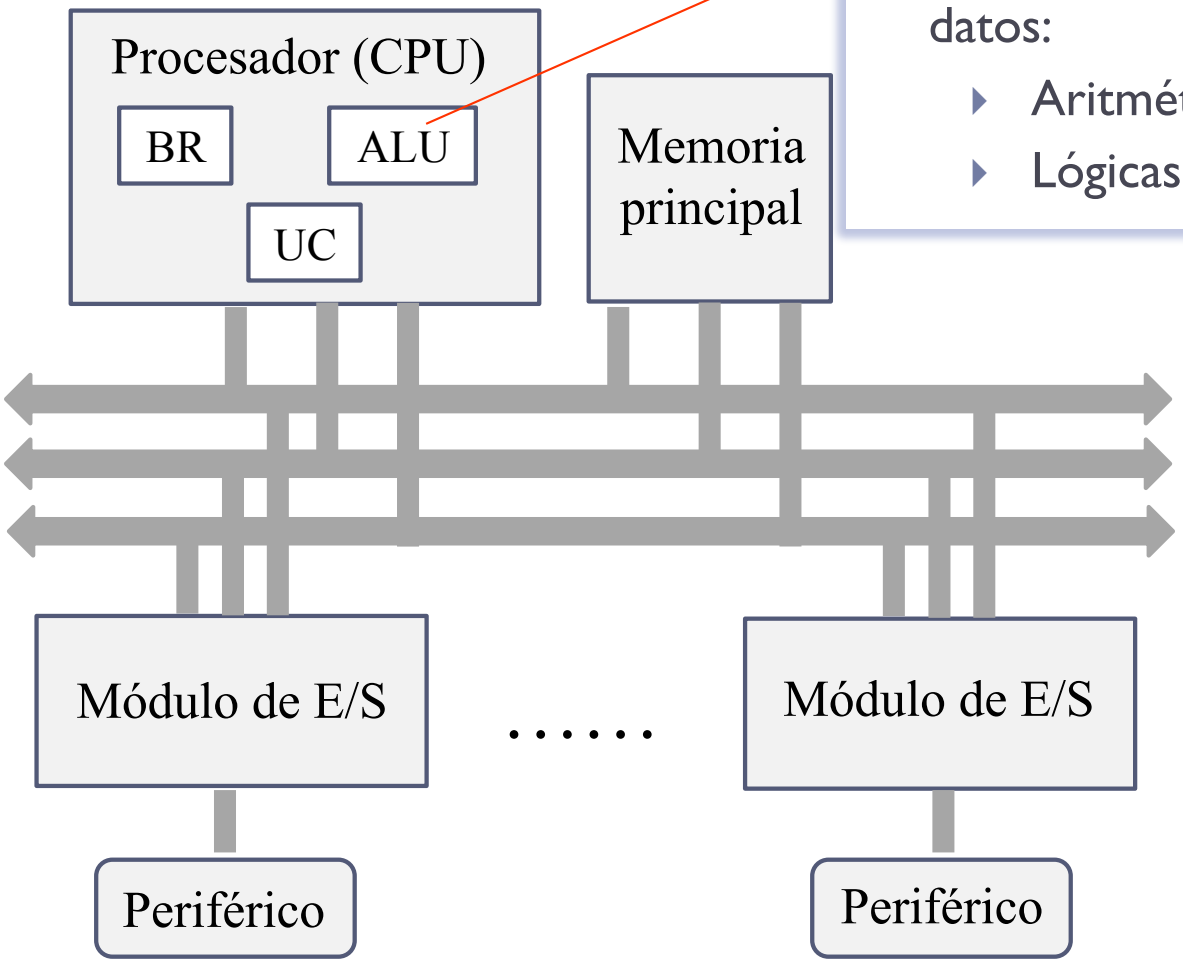
Procesador: registros

- Registros: almacenan una secuencia de bits.
- Dos registros especiales:
  - El registro PC (contador de programa) contiene la dirección de la siguiente instrucción a ejecutar.
  - El registro RI (registro de instrucción) almacena la instrucción que se está ejecutando

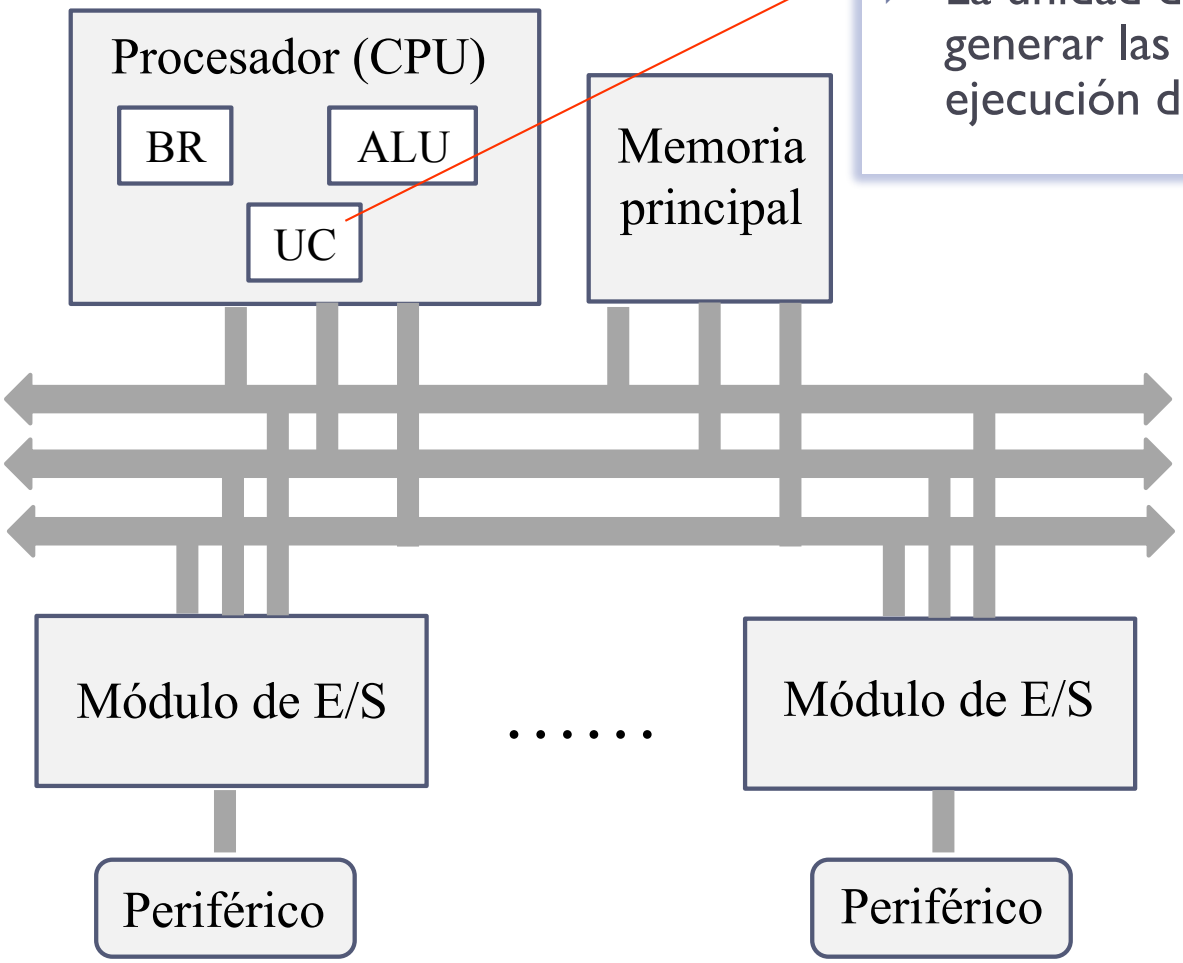


Procesador: Unidad aritmético lógica ALU

- Realiza operaciones elementales sobre los datos:
  - Aritméticas
  - Lógicas

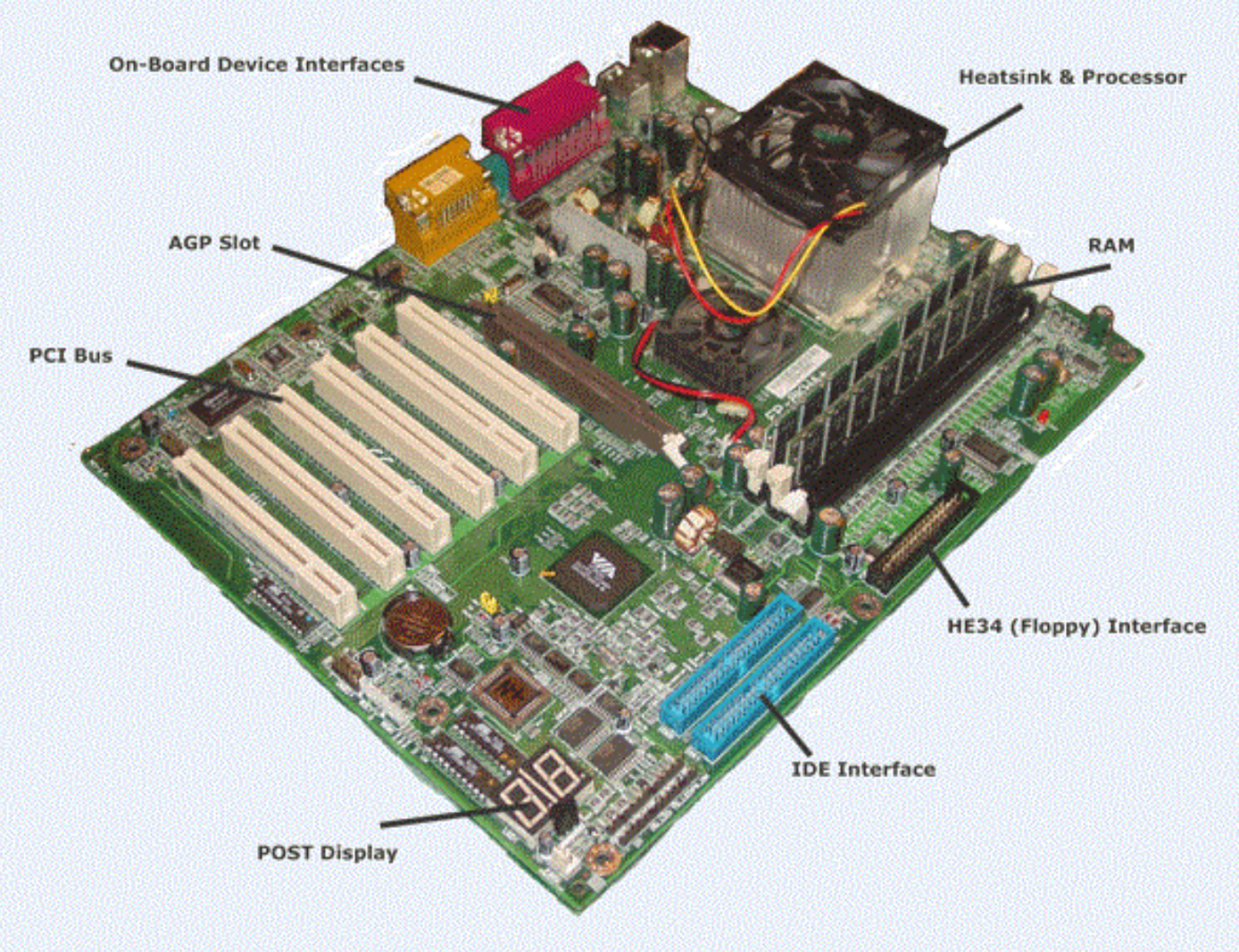


Procesador: Unidad de control, UC



- La unidad de control se encarga de generar las señales de control para la ejecución de instrucciones.

¿Podemos distinguir los componentes internos al abrir un ordenador personal?



http://electronicrepairing.blogspot.com.es/2013/08/cpu-ram-motherboard-troubleshooting.html

Contenidos

1. ¿Qué es un computador?
2. Concepto de estructura y arquitectura
3. Elementos constructivos de un computador
4. Computador Von Neumann
5. Instrucciones máquina y programación
6. Fases de ejecución de una instrucción
7. Parámetros característicos de un computador
8. Tipos de computadores
9. Evolución histórica

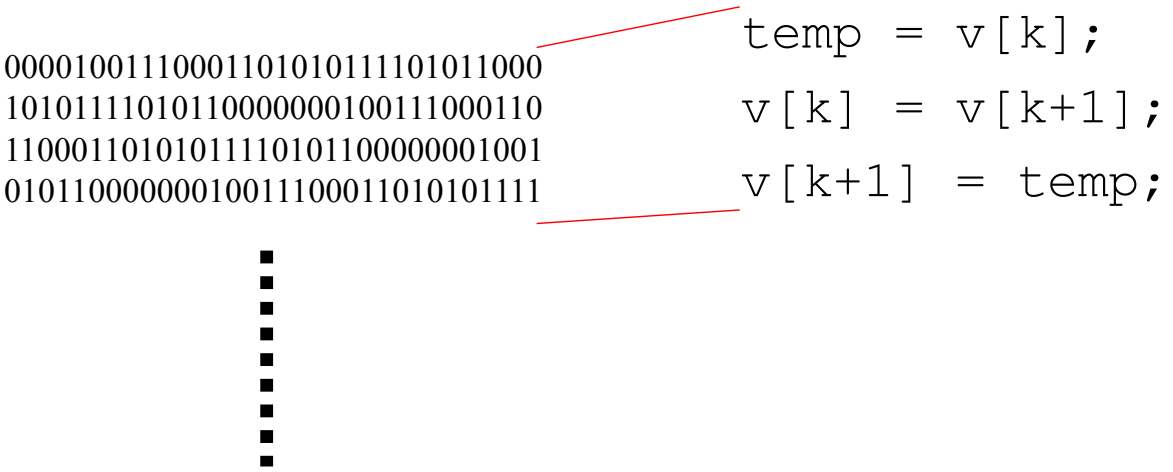
Programa

- Secuencia consecutiva de instrucciones máquina

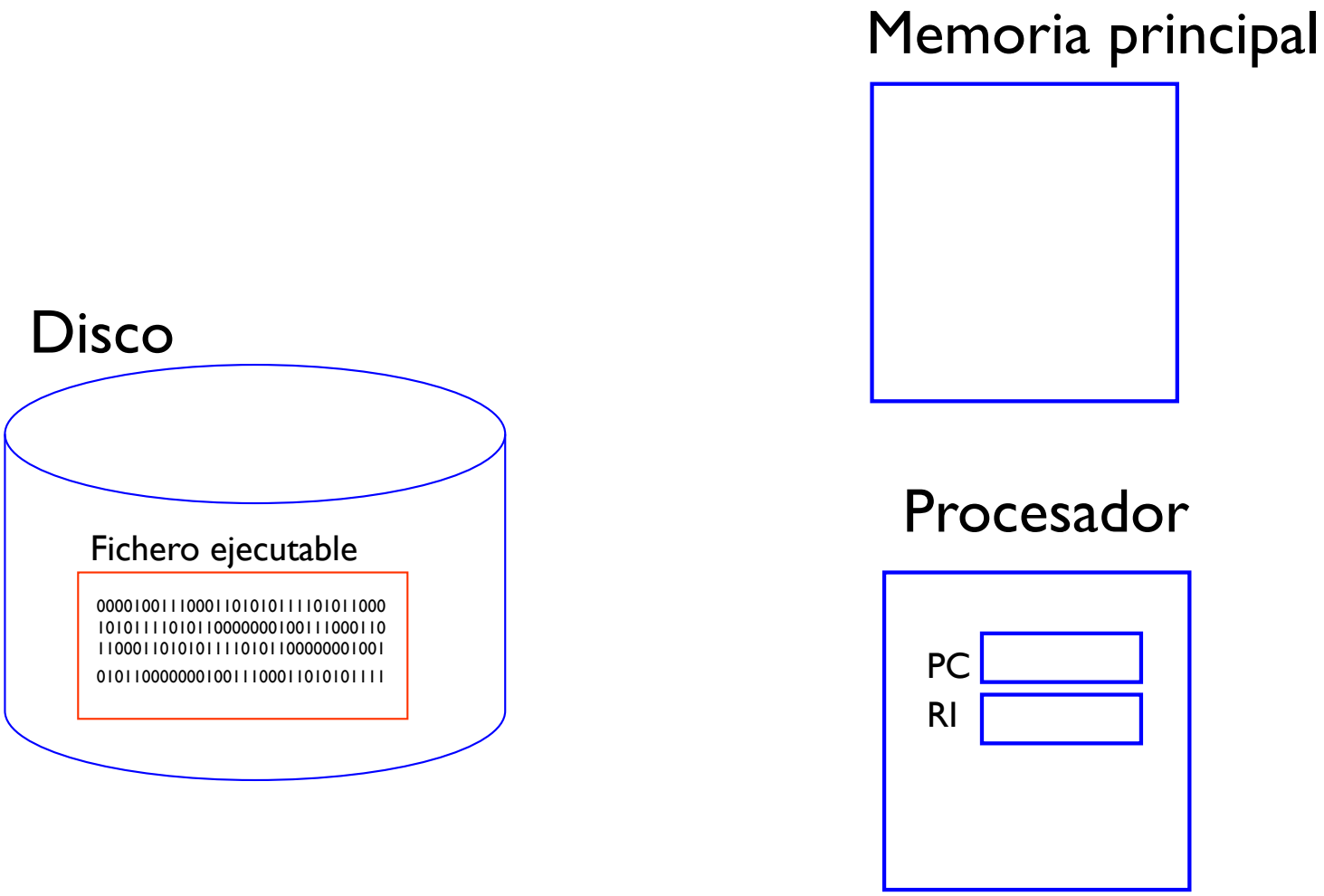
```
00001001110001101010111101011000
10101111010110000000100111000110
11000110101011110101100000001001
01011000000010011100011010101111
```

Programa

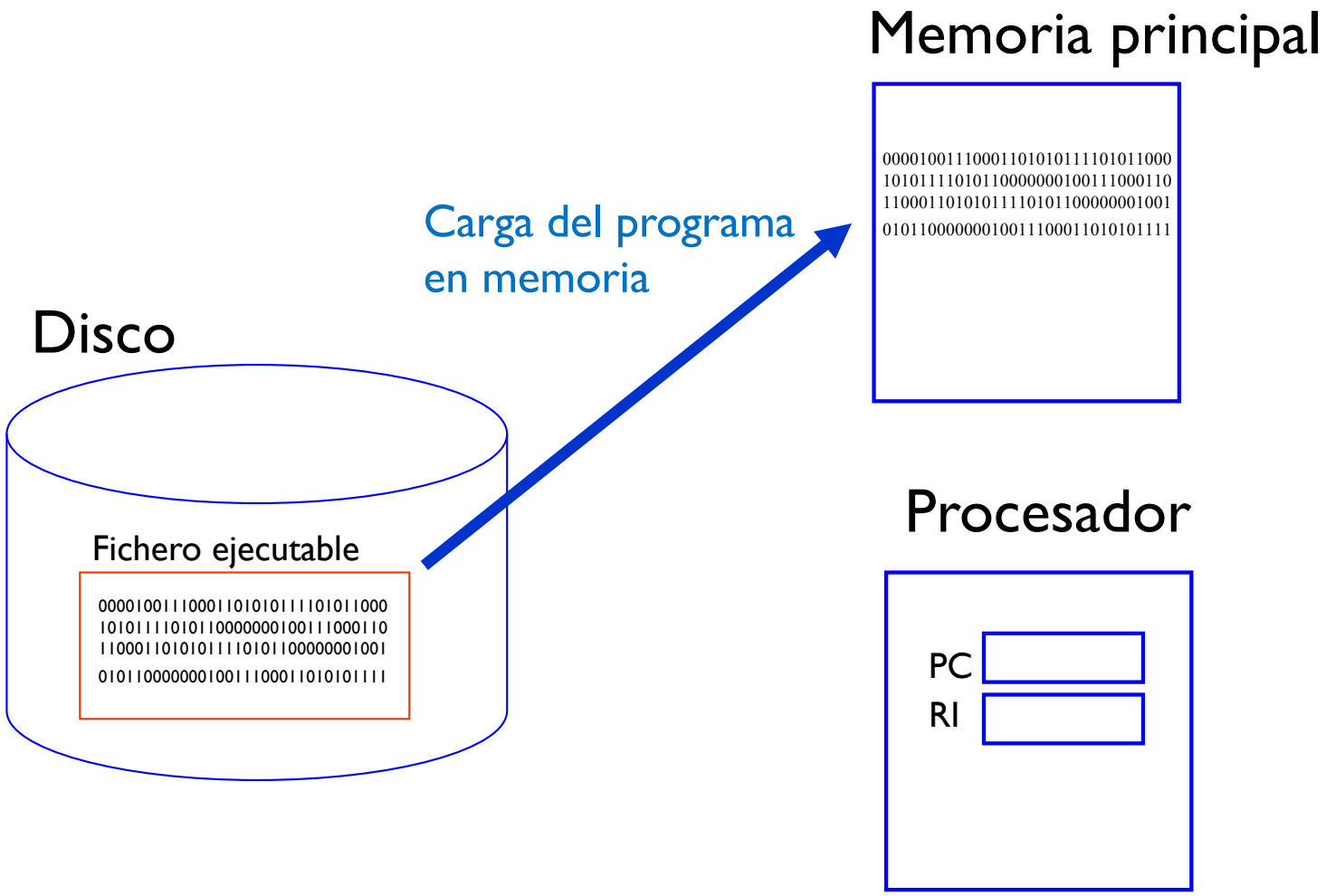
- Secuencia consecutiva de instrucciones máquina
- Instrucción máquina: operación elemental que puede ejecutar directamente un procesador
  - Codificación en binario



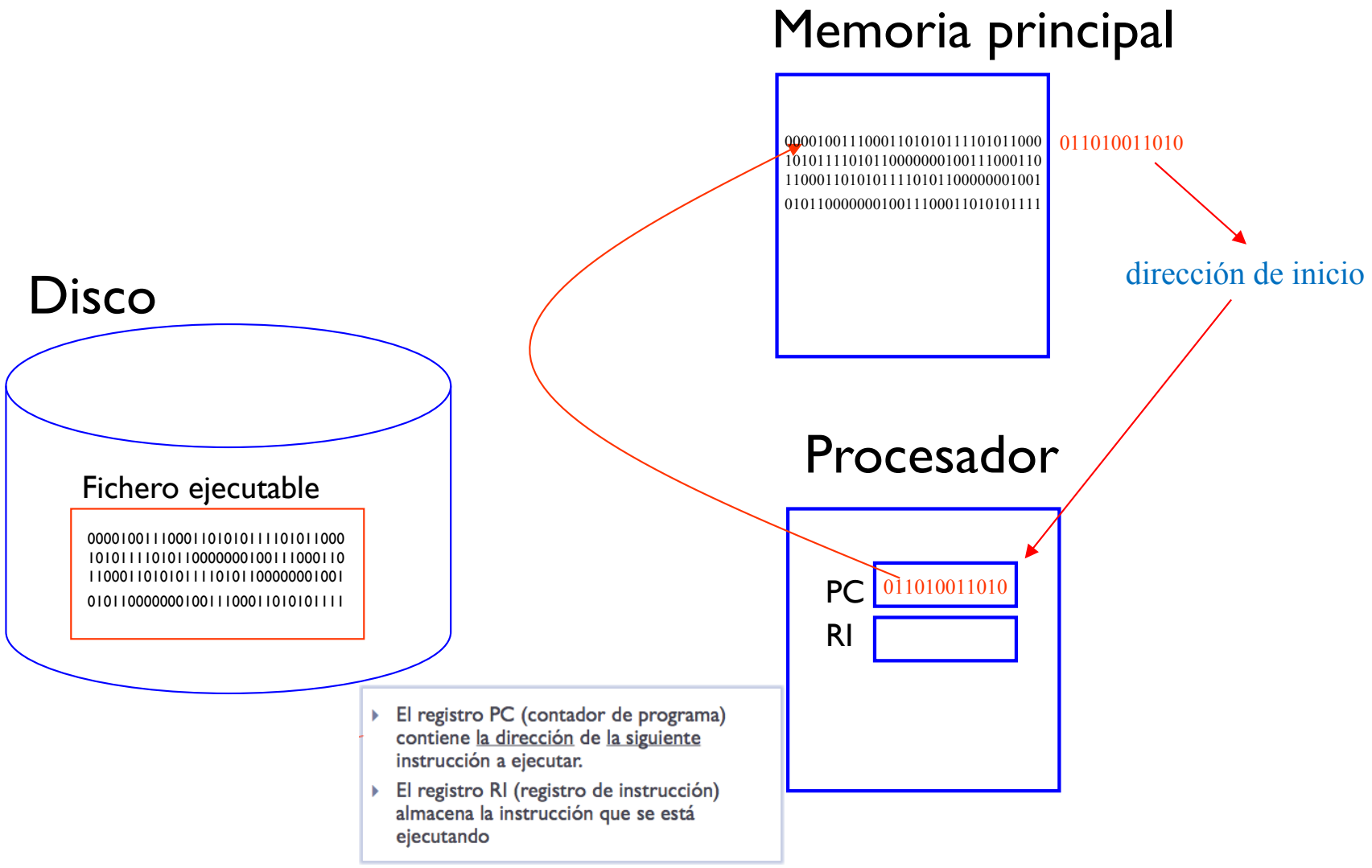
Ejecución de un programa



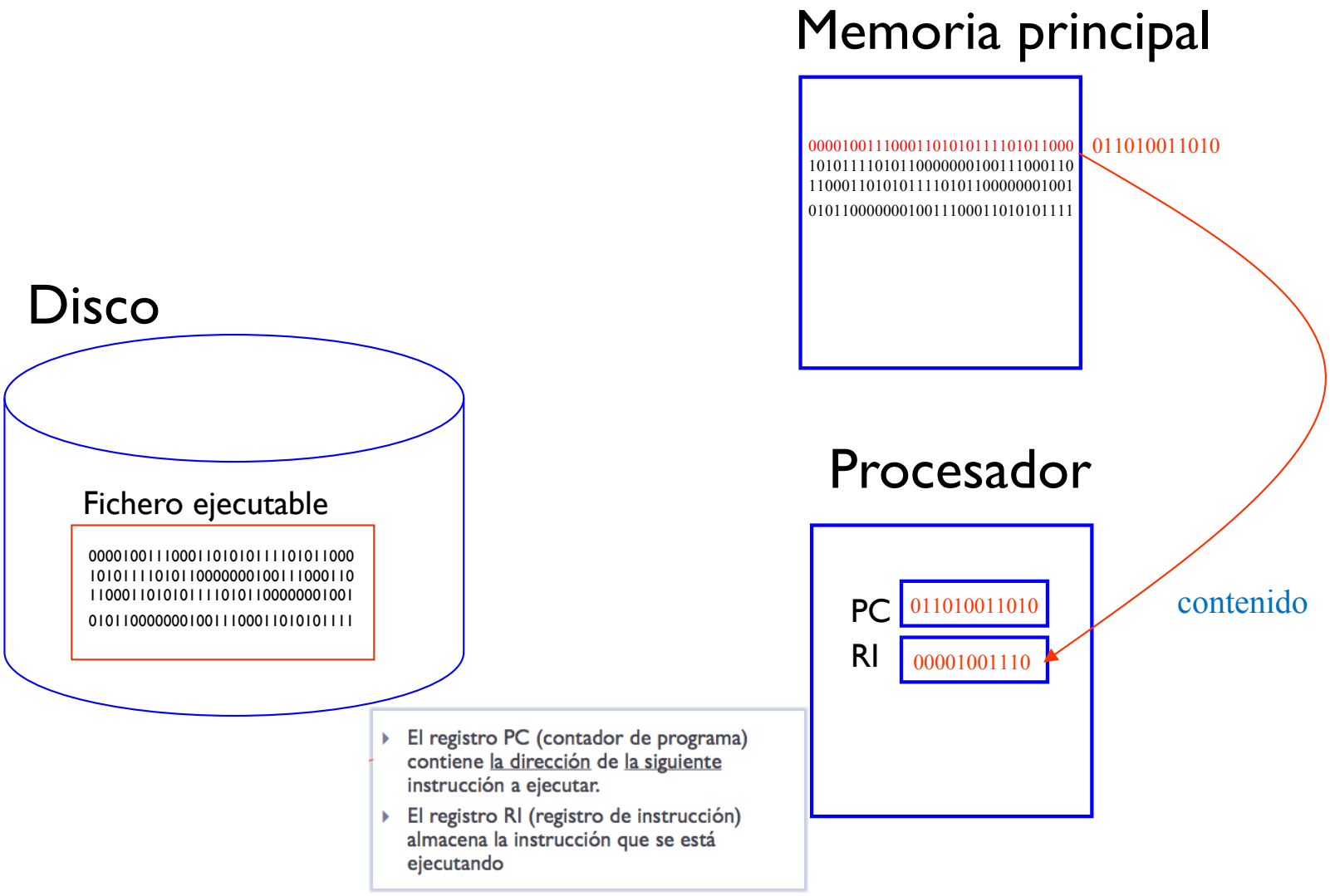
Ejecución de un programa



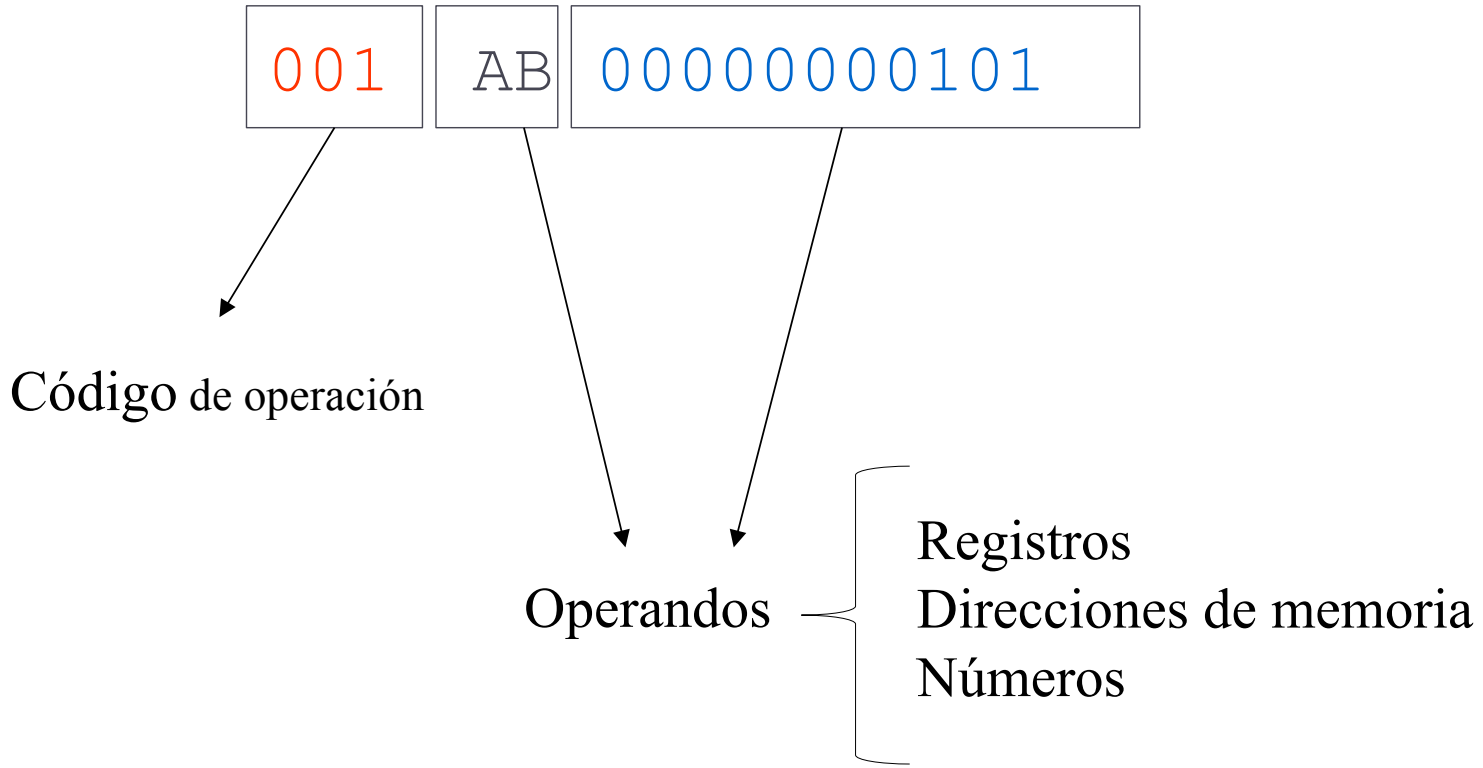
Ejecución de un programa



Ejecución de un programa

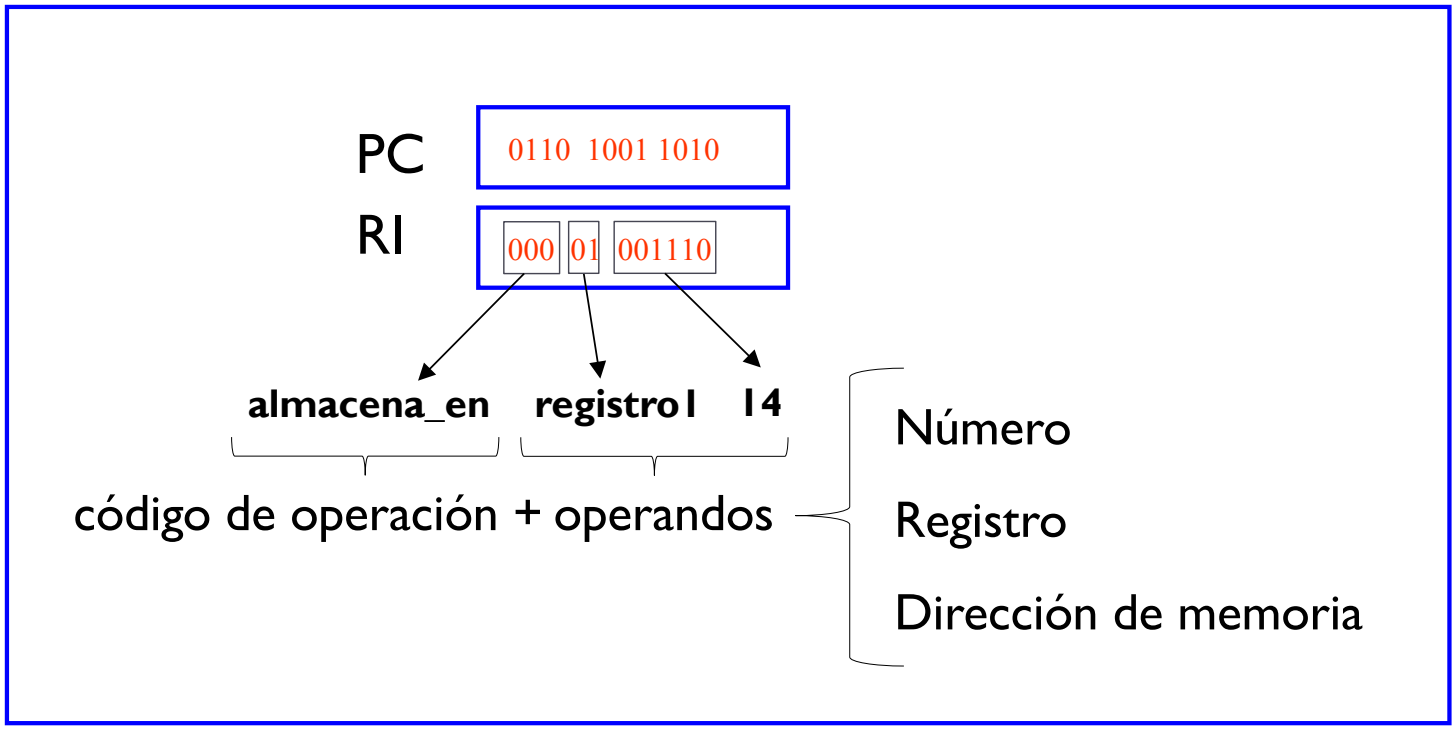


Formato de una instrucción máquina



Formato de instrucción

Procesador



Ejemplo de juego de instrucciones

- Conjunto de instrucciones con las siguientes características:
  - Tamaño de una posición de memoria: 16 bits
  - Tamaño de la instrucción: 16 bits
  - Código de operación: 3 bits
    - ¿Cuántas instrucciones diferentes puede tener este computador?
    - Número de registros de propósito general: 4
    - Identificadores simbólicos:
      - R0
      - R1
      - R2
      - R3
    - ¿Cuántos bits se necesitan para representar estos 4 registros?

Ejemplo de juego de instrucciones

- Conjunto de instrucciones con las siguientes características:
  - Tamaño de una posición de memoria: 16 bits
  - Tamaño de la instrucción: 16 bits
  - Código de operación: 3 bits
    - ¿Cuántas instrucciones diferentes puede tener este computador? 8
    - Número de registros de propósito general: 4 (2 bits)
    - Identificadores simbólicos:
      - R0 (00)
      - R1 (01)
      - R2 (10)
      - R3 (11)
    - ¿Cuántos bits se necesitan para representar estos 4 registros? 2

Ejemplo de juego de instrucciones

Instrucción	Descripción
000CCAABBXXXXXXX	Suma el registro AA con el BB y deja el resultado en CC
001AA00000000101	Almacena en el registro AA el valor 00000000101
010AA00000001001	Almacena en el registro AA el valor almacenado en la posición de memoria 00000001001
011AA00000001001	Almacena en la posición de memoria 00000001001 el contenido del registro AA
100000000001001	Se salta a ejecutar la instrucción almacenada en la posición de memoria 000000001001
101AABB000001001	Si el contenido del registro AA es igual al del registro BB se salta a ejecutar la instrucción almacenada en 000001001

Siendo A,B,C,D,E,F = 0 ó 1

Ejemplos

- Instrucción que almacena un 5 en el registro 00
- Instrucción que almacena un 7 en el registro 01
- Instrucción que suma el contenido del registro 00 y el registro 01 y deja el resultado en el registro 10
- Instrucción que almacena el resultado anterior en la posición de memoria 1027 (en decimal)

Ejemplos

Instrucción	Descripción
000CCAABBXXXXXXX	Suma el registro AA con el BB y deja el resultado en CC
001AA00000000101	Almacena en el registro AA el valor 00000000101
010AA00000001001	Almacena en el registro AA el valor almacenado en la posición de memoria 00000001001
011AA00000001001	Almacena en la posición de memoria 00000001001 el contenido del registro AA
1000000000001001	Se salta a ejecutar la instrucción almacenada en la posición de memoria 0000000001001
101AABB000001001	Si el contenido del registro AA es igual al del registro BB se salta a ejecutar la instrucción almacenada en 000001001

Siendo A,B,C,D,E,F = 0 ó 1

- Instrucción que almacena un 5 en el registro 00
- Instrucción que almacena un 7 en el registro 01
- Instrucción que suma el contenido del registro 00 y el registro 01 y deja el resultado en el registro 10
- Instrucción que almacena el resultado anterior en la posición de memoria 1027 (en decimal)

Ejemplos

Instrucción	Descripción
000CCAABBXXXXXXX	Suma el registro AA con el BB y deja el resultado en CC
001AA00000000101	Almacena en el registro AA el valor 00000000101
010AA00000001001	Almacena en el registro AA el valor almacenado en la posición de memoria 00000001001
011AA00000001001	Almacena en la posición de memoria 00000001001 el contenido del registro AA
1000000000001001	Se salta a ejecutar la instrucción almacenada en la posición de memoria 0000000001001
101AABB000001001	Si el contenido del registro AA es igual al del registro BB se salta a ejecutar la instrucción almacenada en 000001001

Siendo A,B,C,D,E,F = 0 ó 1

- Instrucción que almacena un 5 en el registro 00

  0010000000000101

- Instrucción que almacena un 7 en el registro 01

  0010100000000111

- Instrucción que suma el contenido del registro 00 y el registro 01 y deja el resultado en el registro 10

  000100001XXXXXXX

- Instrucción que almacena el resultado anterior en la posición de memoria 1027 (en decimal)

  0111010000000011

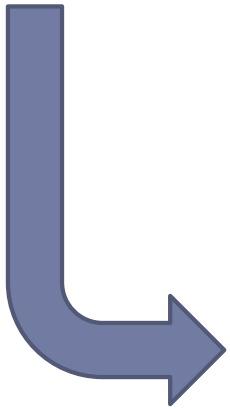
Ejemplo de programa cargado en memoria

Memoria principal

Dirección	Contenido
000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

Generación y carga de un programa

```
i=0;
s = 0;
while (i < 4)
{
   s = s + 1;
   i = i + 1;
}
```



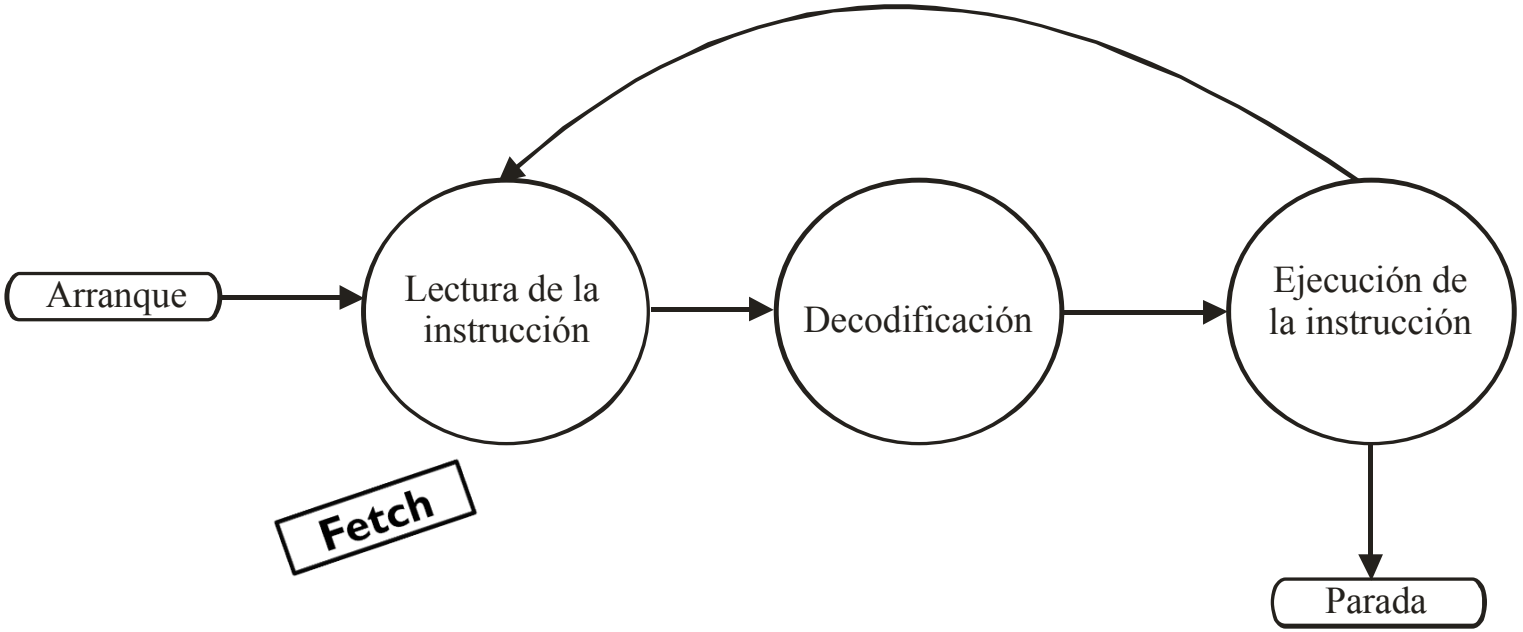
```
        li R0, 0
        li R1, 4
        li R2, 1
        li R3, 0
lazo:  beq R0, R1, fin
       add R3, R3, R2
       add R0, R0, R2
       b lazo
fin:   sw R3, 100000
```

000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

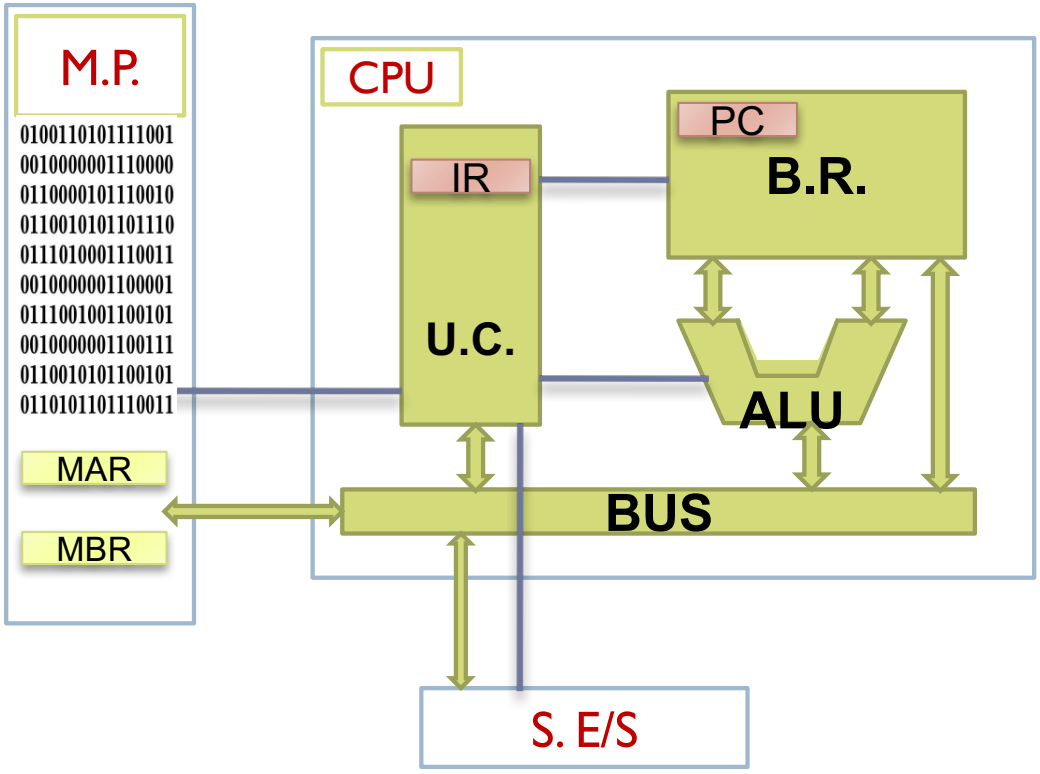
Contenidos

1. ¿Qué es un computador?
2. Concepto de estructura y arquitectura
3. Elementos constructivos de un computador
4. Computador Von Neumann
5. Instrucciones máquina y programación
6. Fases de ejecución de una instrucción
7. Parámetros característicos de un computador
8. Tipos de computadores
9. Evolución histórica

Fases de ejecución de una instrucción

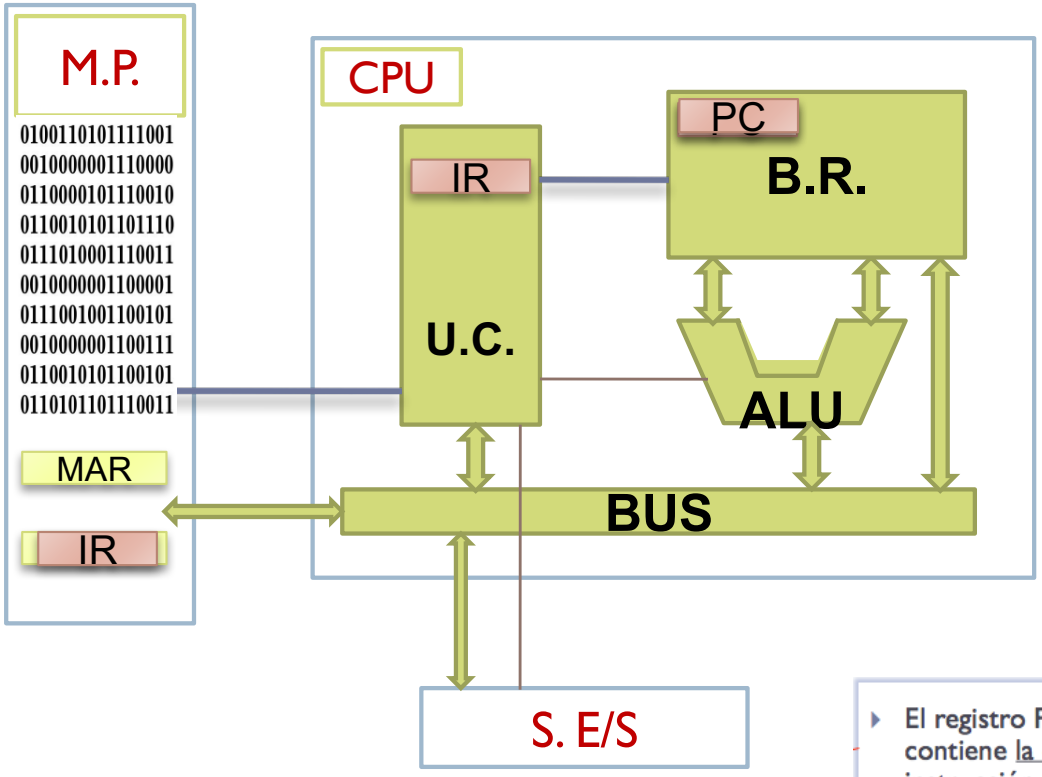


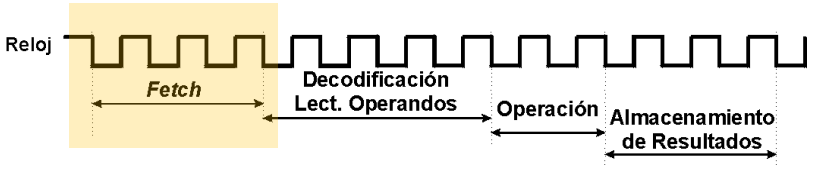
Fases de ejecución (1)



- Leer de memoria principal la instrucción apuntada por el PC
- Incrementar PC
- Decodificar instrucción
- Ejecutar la instrucción

Fases de ejecución (2)

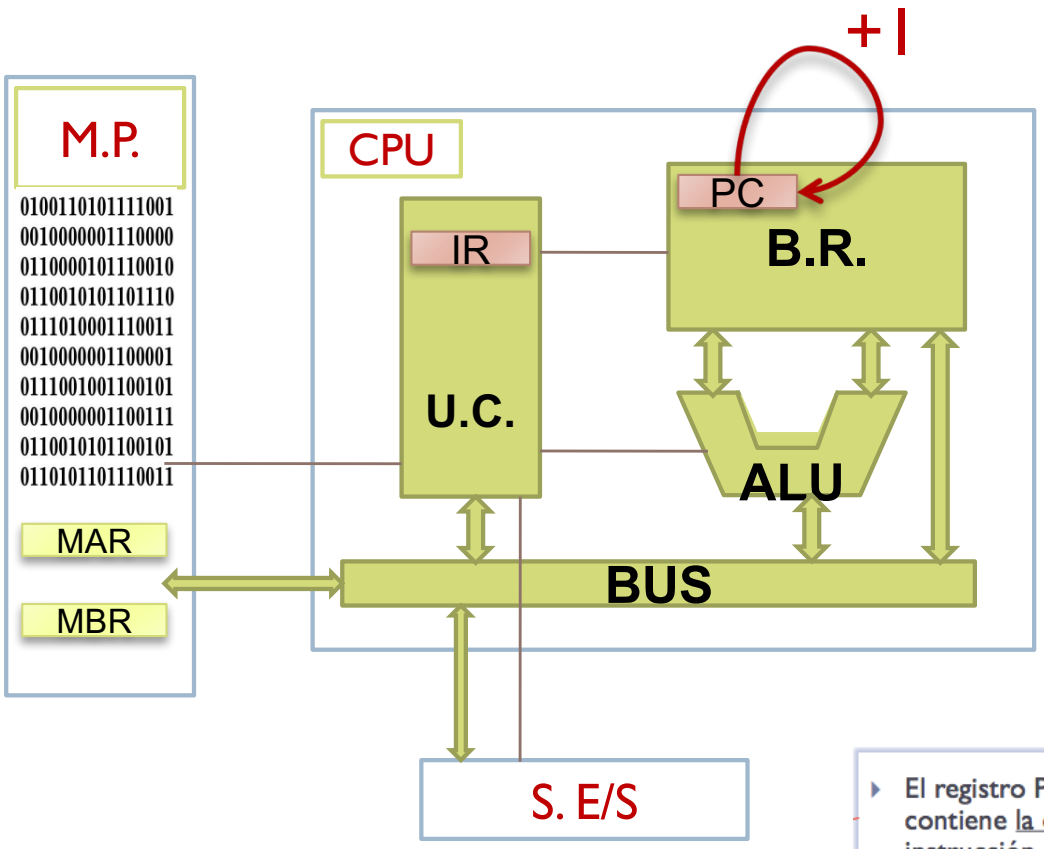


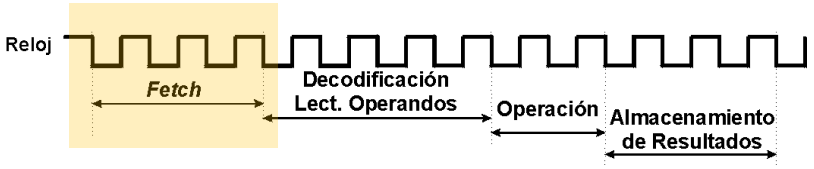


- Leer de memoria principal la instrucción apuntada por el PC
- Incrementar PC
- Decodificar instrucción
- Ejecutar la instrucción

- El registro PC (contador de programa) contiene la dirección de la siguiente instrucción a ejecutar.
- El registro RI (registro de instrucción) almacena la instrucción que se está ejecutando

Fases de ejecución (3)

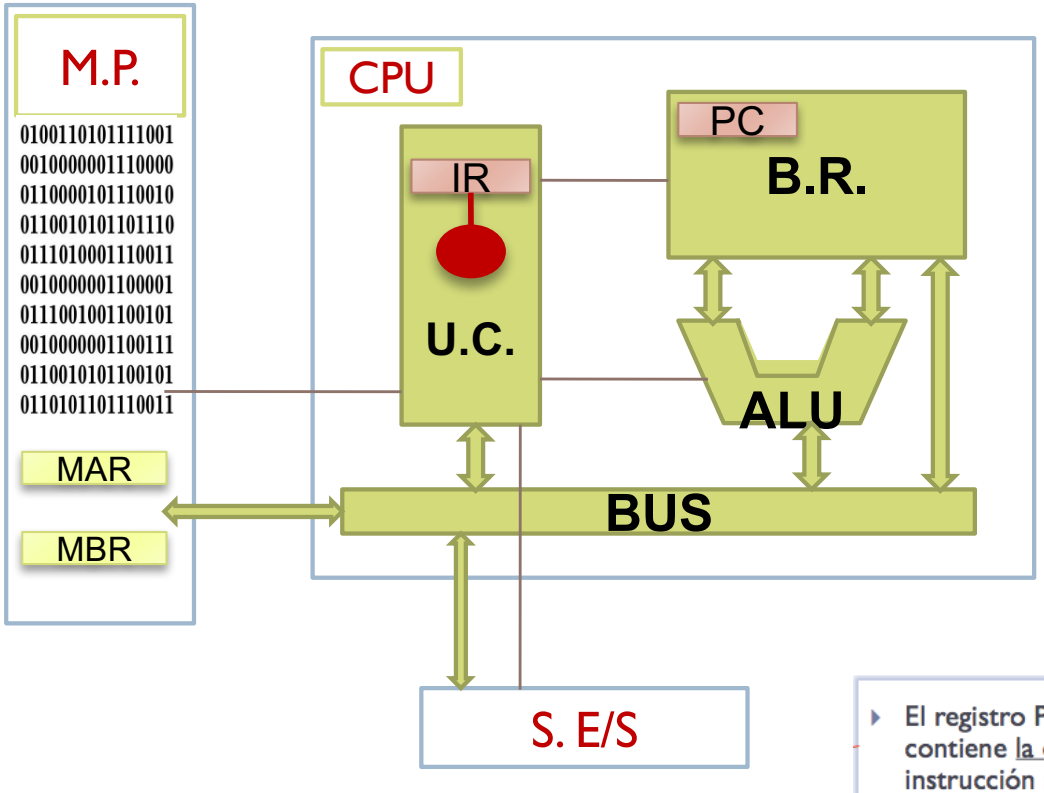


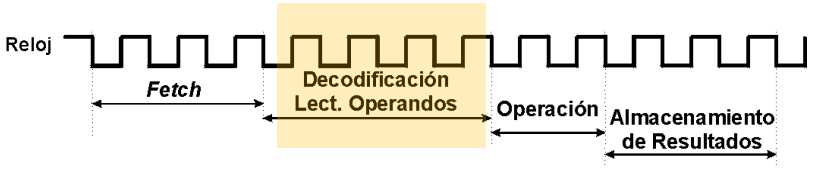


- Leer de memoria principal la instrucción apuntada por el PC
- Incrementar PC
- Decodificar instrucción
- Ejecutar la instrucción

- El registro PC (contador de programa) contiene la dirección de la siguiente instrucción a ejecutar.
- El registro RI (registro de instrucción) almacena la instrucción que se está ejecutando

Fases de ejecución (4)

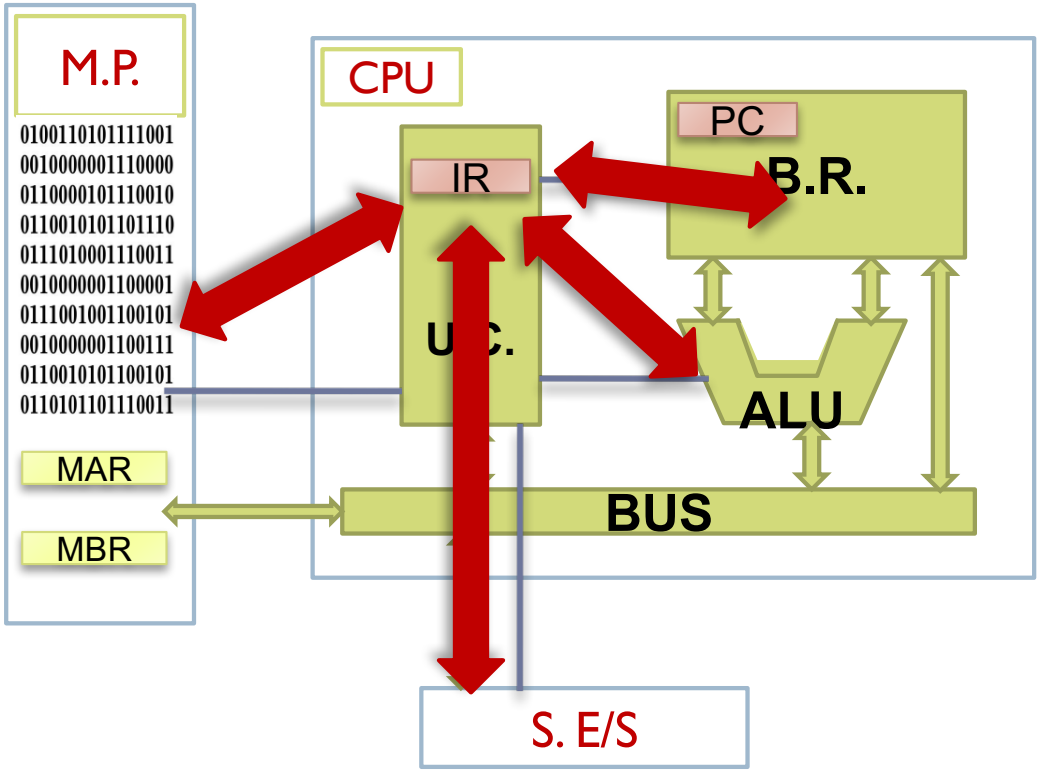


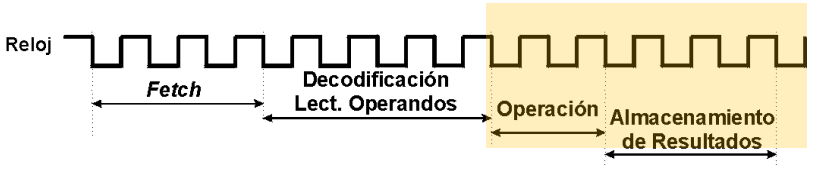


- Leer de memoria principal la instrucción apuntada por el PC
- Incrementar PC
- Decodificar instrucción
- Ejecutar la instrucción

- El registro PC (contador de programa) contiene la dirección de la siguiente instrucción a ejecutar.
- El registro RI (registro de instrucción) almacena la instrucción que se está ejecutando

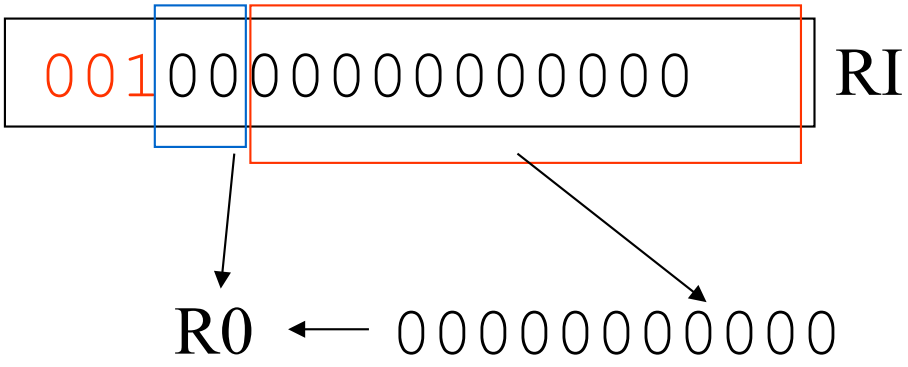
Fases de ejecución (5)





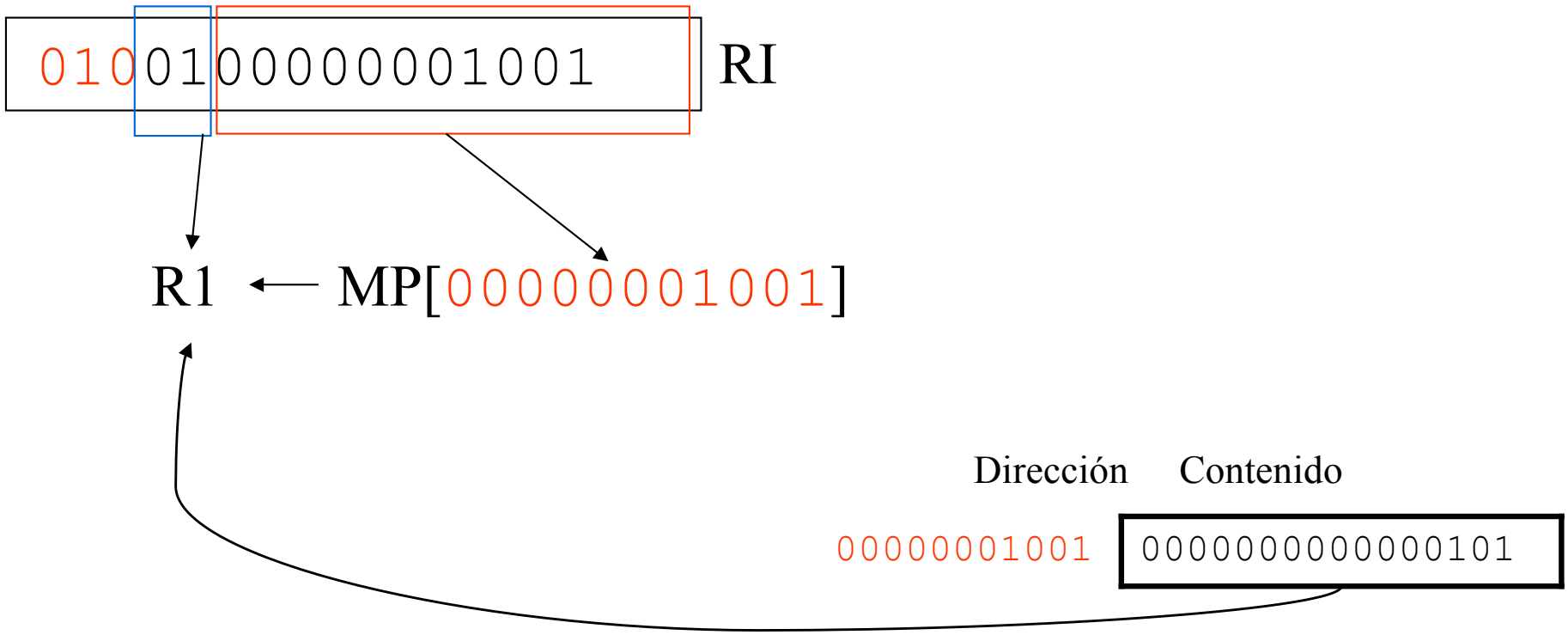
- Leer de memoria principal la instrucción apuntada por el PC
- Incrementar PC
- Decodificar instrucción
- Ejecutar la instrucción

Ejemplo
ejecución de instrucciones



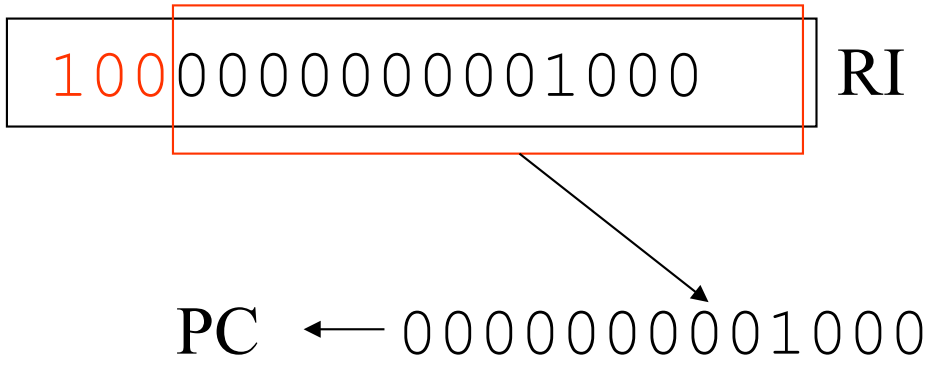
Se carga en R0 el valor 0

Ejemplo
ejecución de instrucciones



Se carga en R1 el contenido de la posición de memoria 00000001001

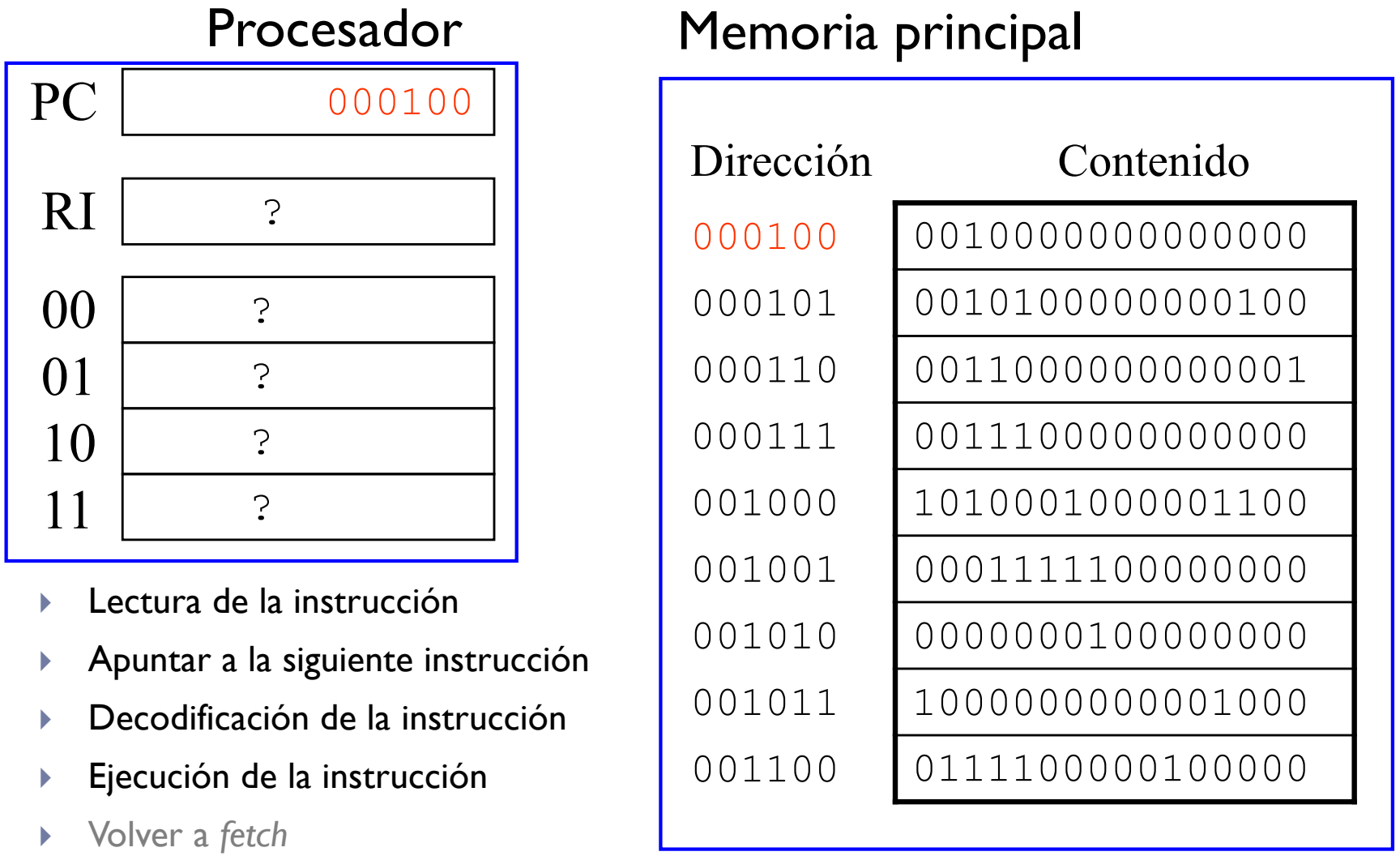
Ejemplo
ejecución de instrucciones



Se modifica el PC con la dirección 000000001000

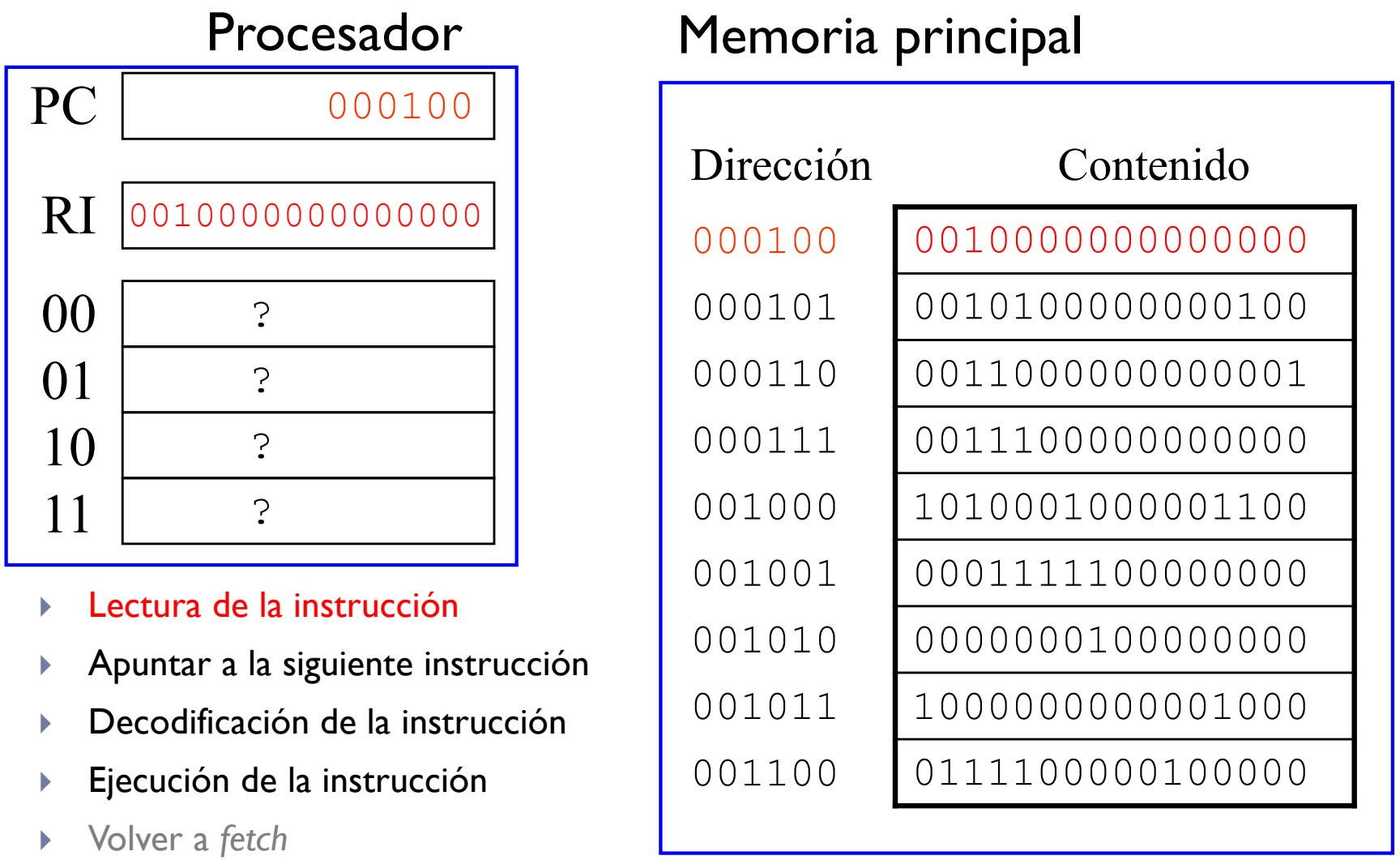
de forma que la siguiente instrucción a ejecutar es la que se encuentra en 000000001000

Ejemplo de ejecución de un programa



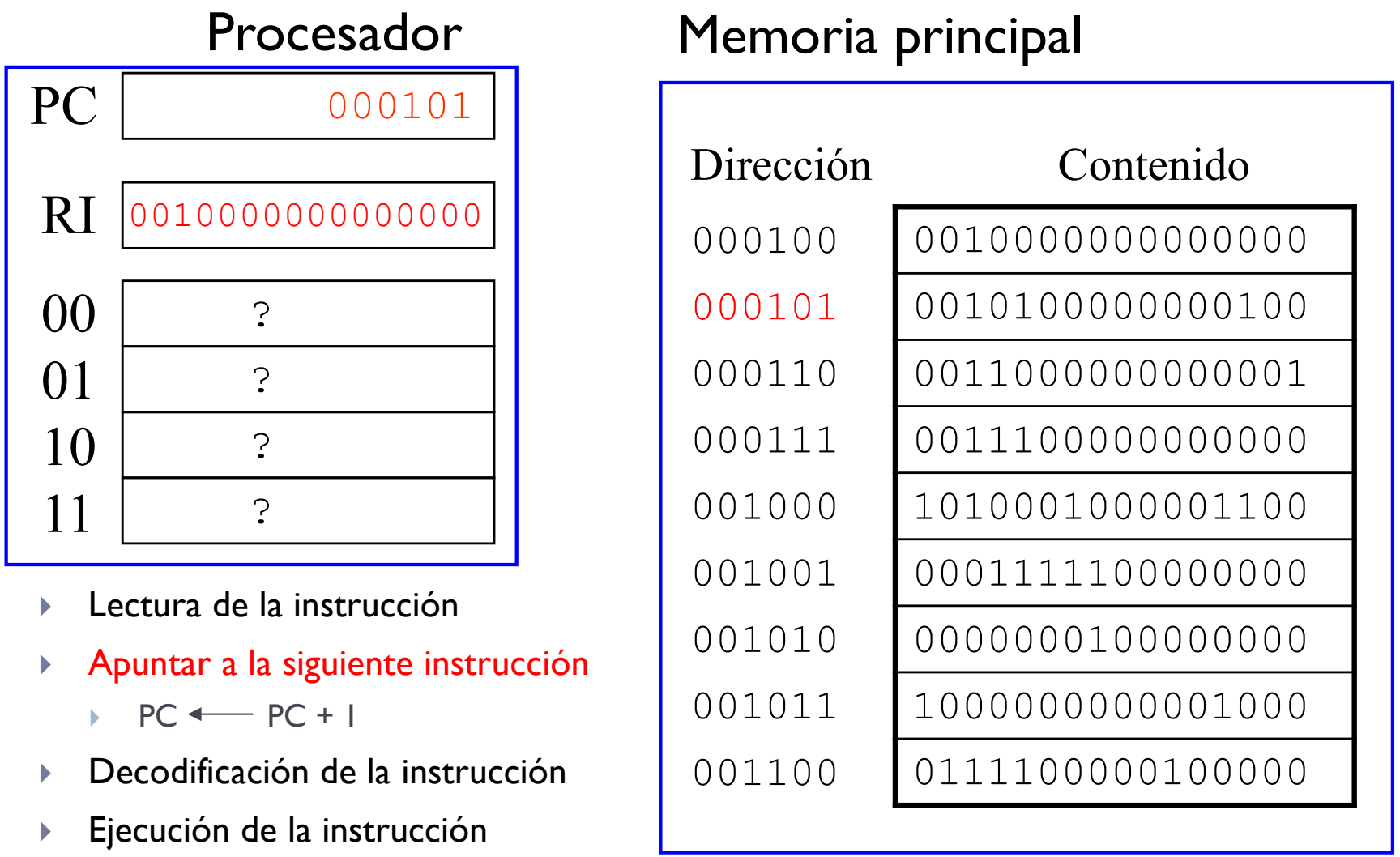
- Lectura de la instrucción
- Apuntar a la siguiente instrucción
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

Ejemplo de ejecución de un programa



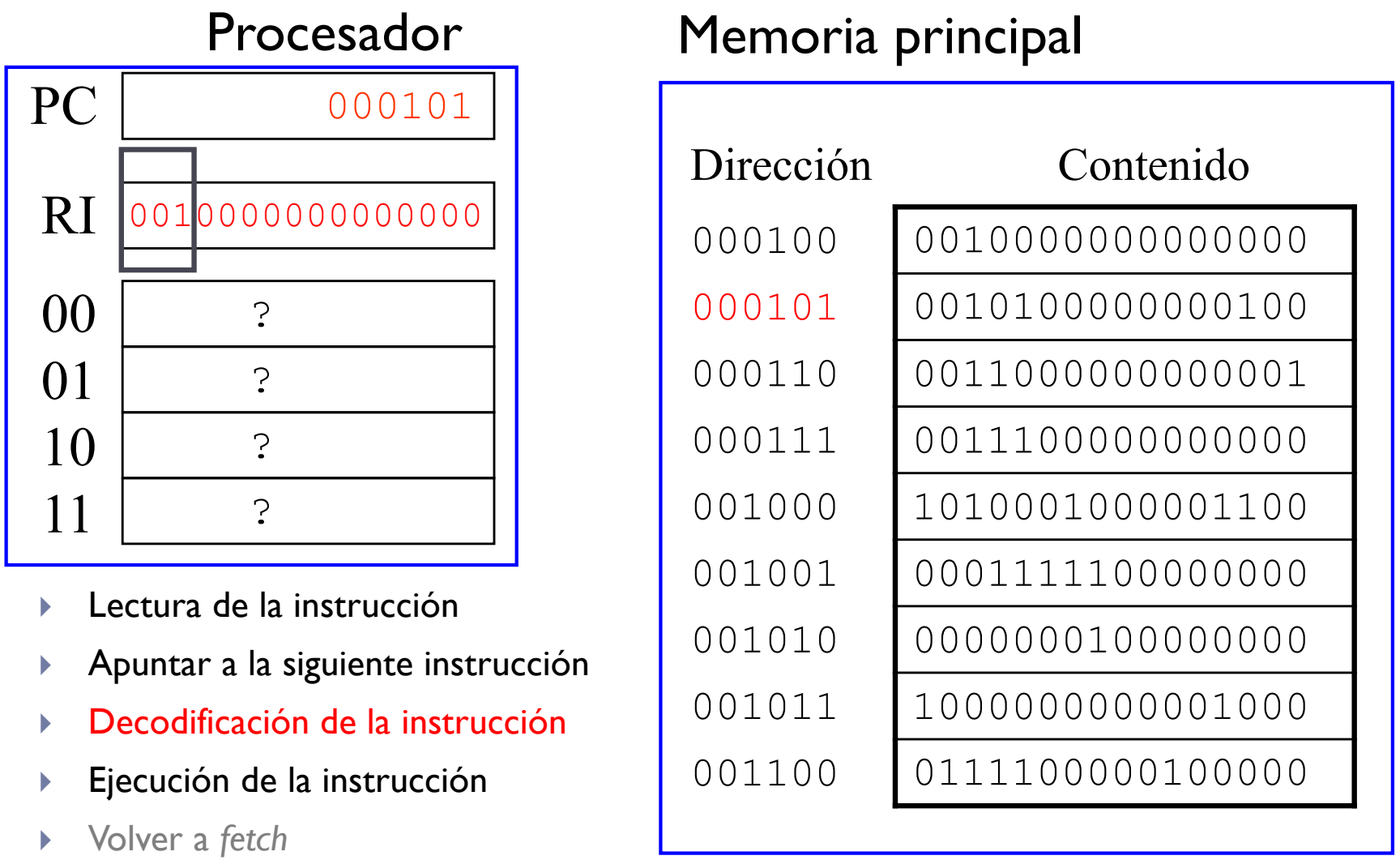
- Lectura de la instrucción
- Apuntar a la siguiente instrucción
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

Ejemplo de ejecución de un programa



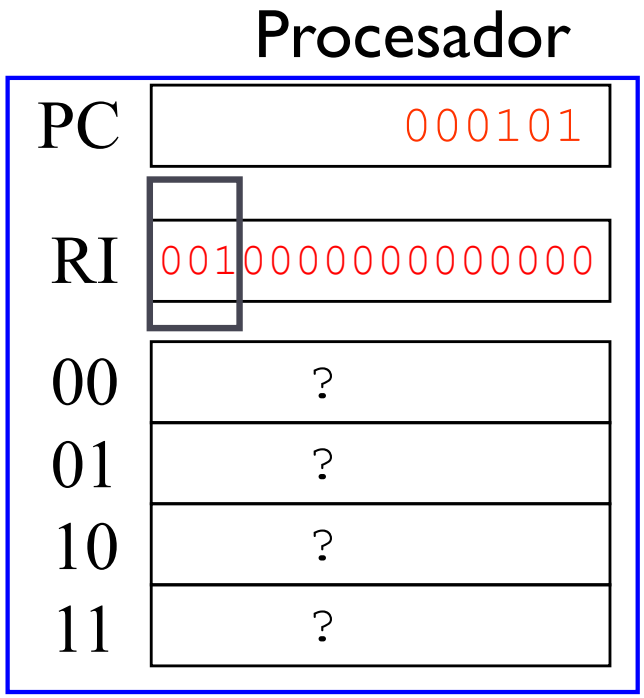
- Lectura de la instrucción
- Apuntar a la siguiente instrucción
  - PC ⟵ PC + 1
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

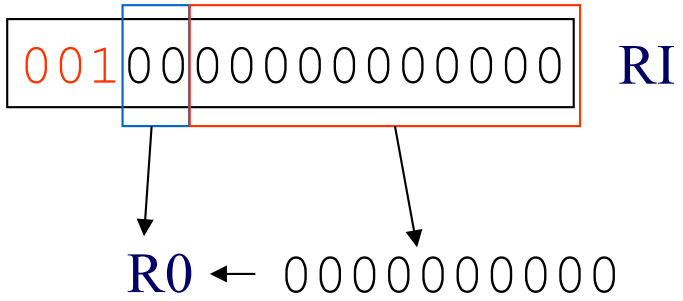
Ejemplo de ejecución de un programa



- Lectura de la instrucción
- Apuntar a la siguiente instrucción
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

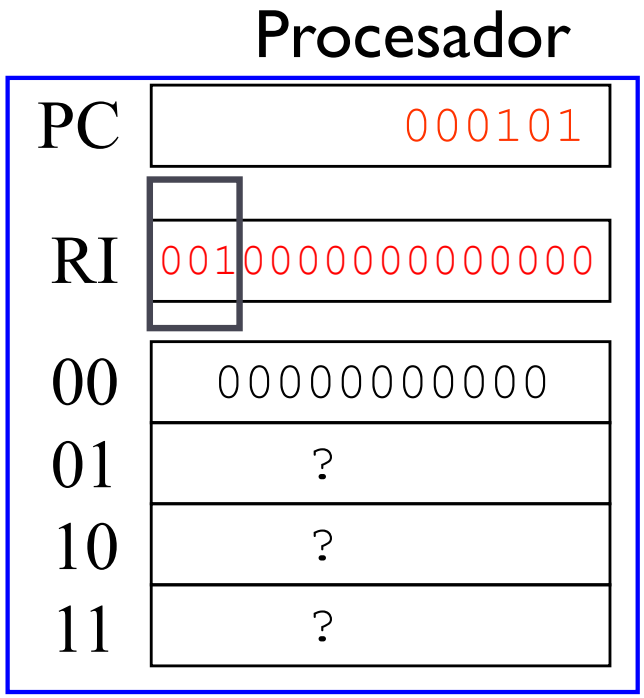
Ejemplo de ejecución de un programa

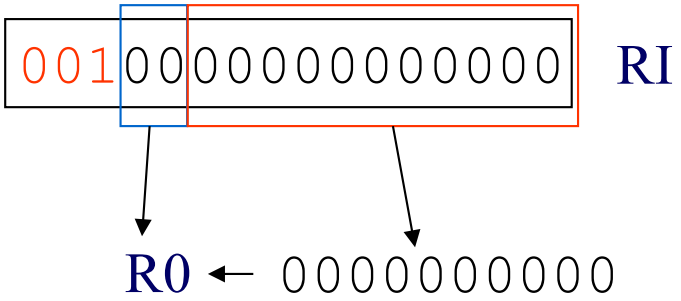




- Lectura de la instrucción
- Apuntar a la siguiente instrucción
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

Ejemplo de ejecución de un programa

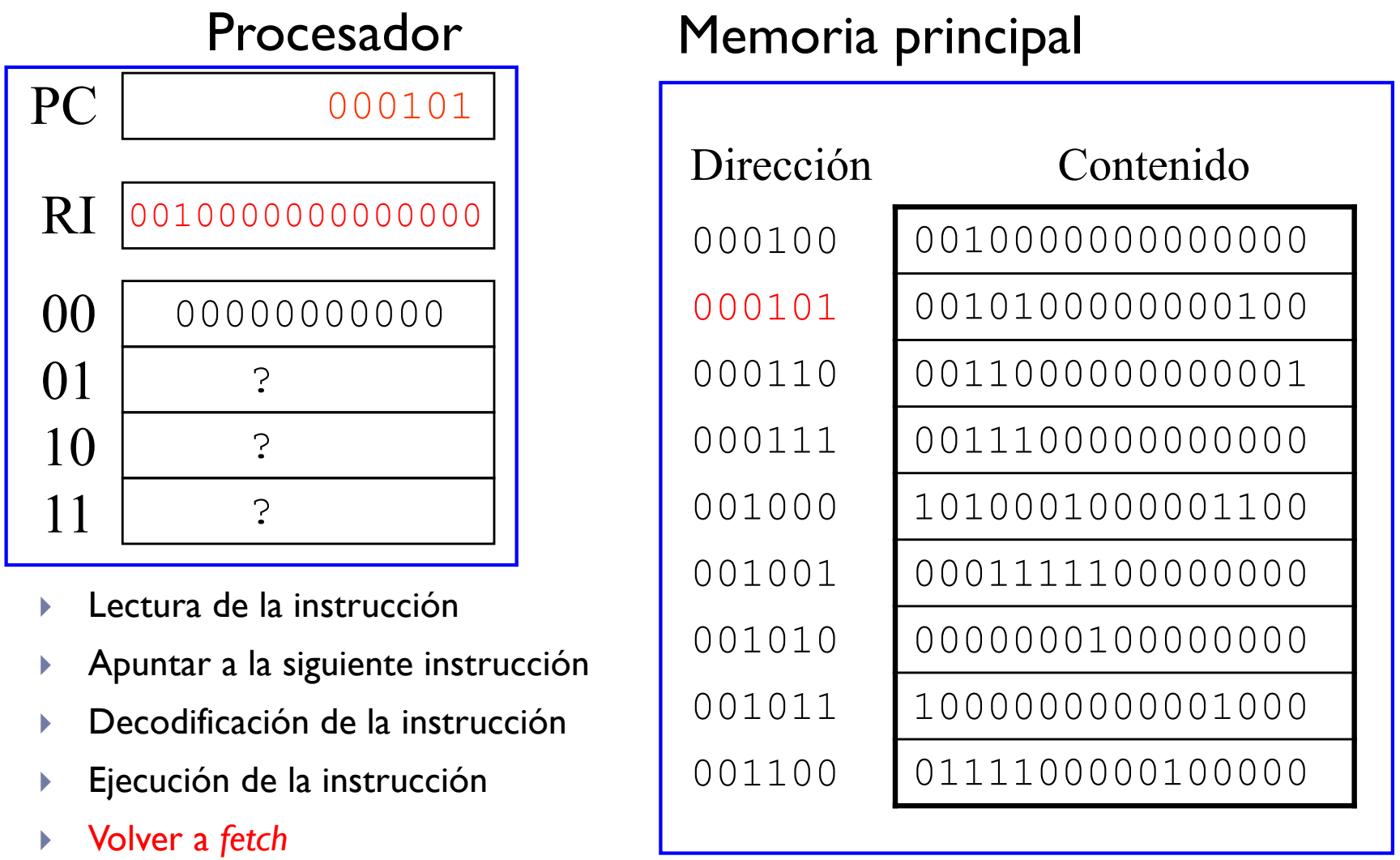




Se carga en R0 el valor 0

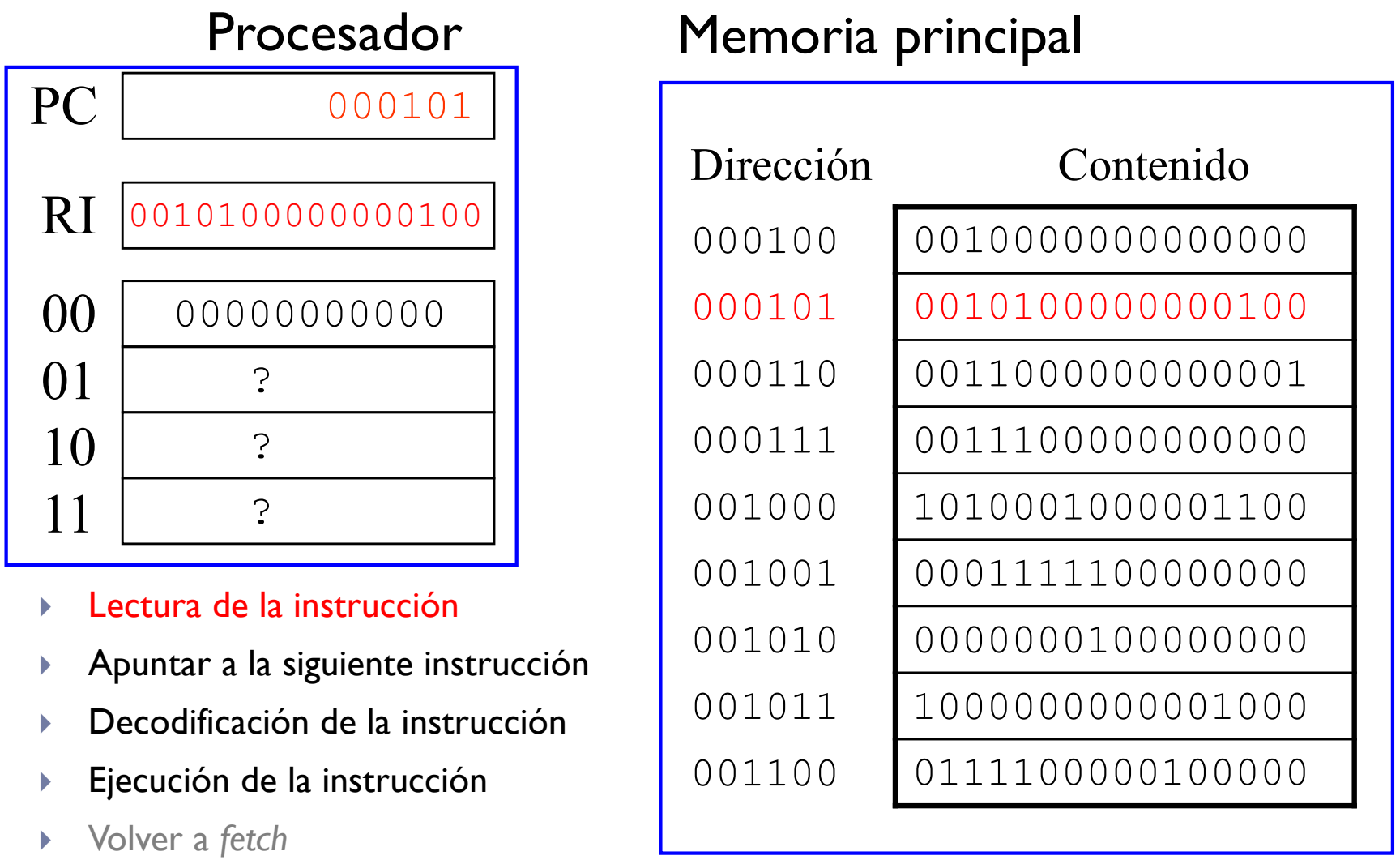
- Lectura de la instrucción
- Apuntar a la siguiente instrucción
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

Ejemplo de ejecución de un programa



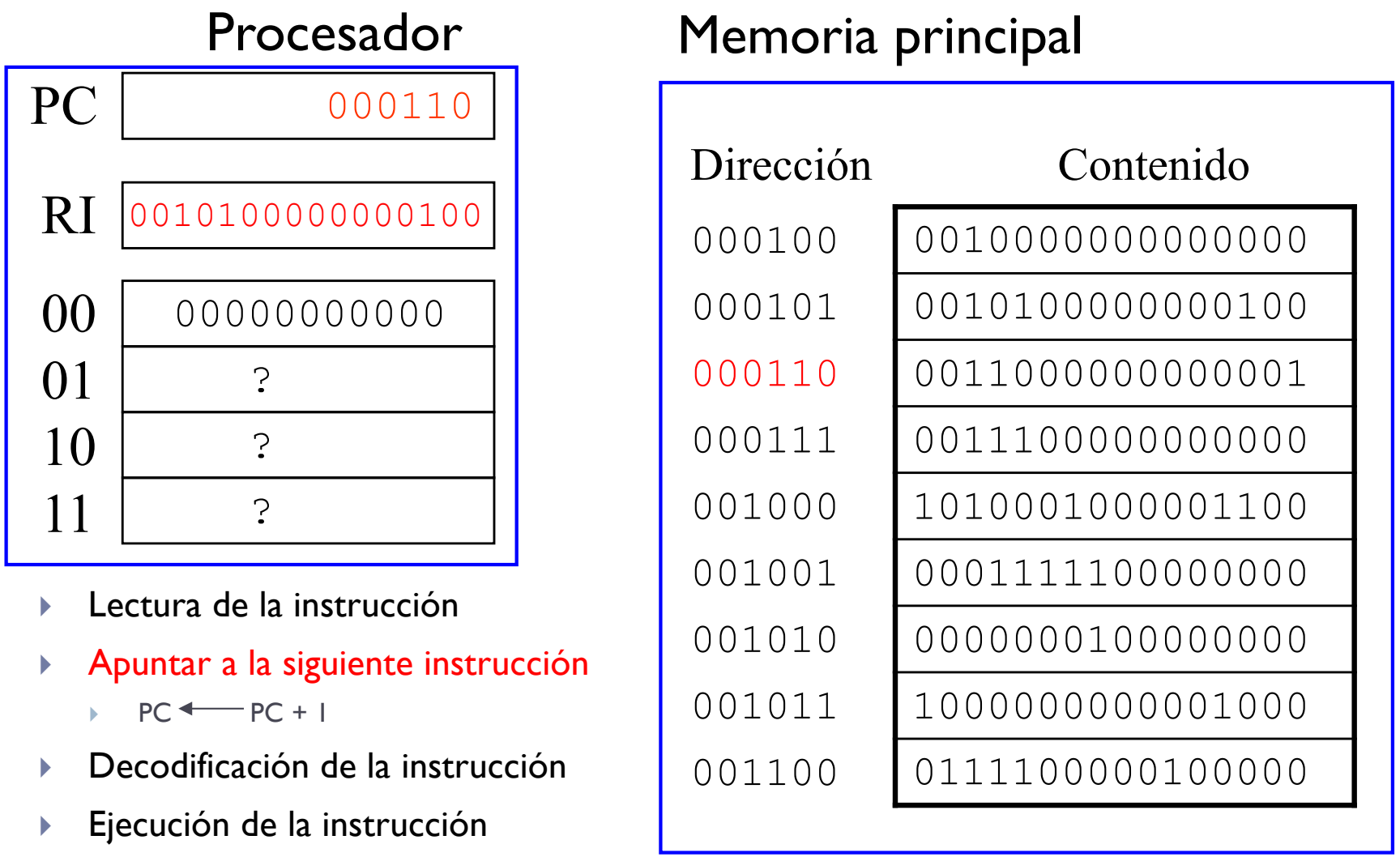
- Lectura de la instrucción
- Apuntar a la siguiente instrucción
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

Ejemplo de ejecución de un programa



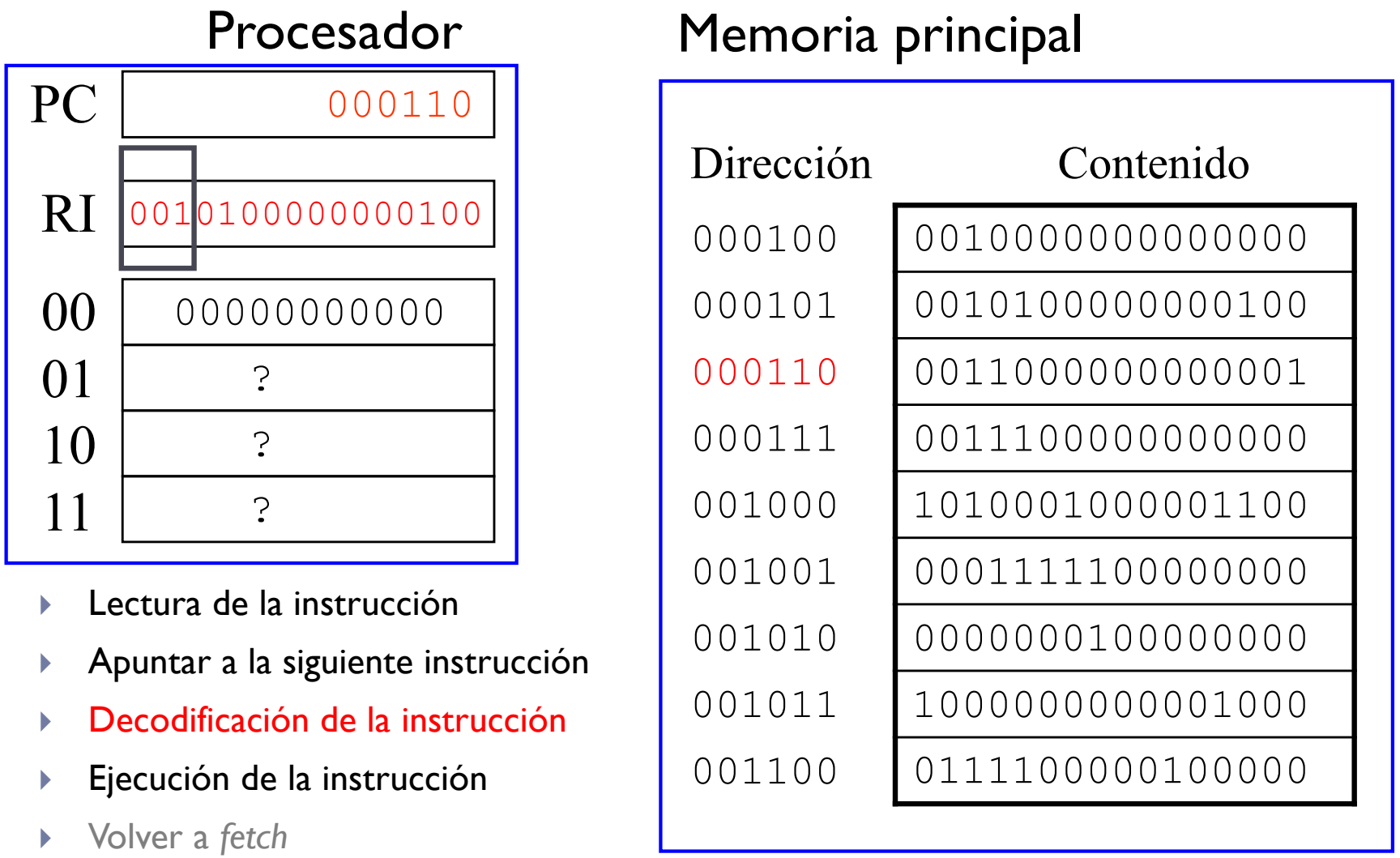
- Lectura de la instrucción
- Apuntar a la siguiente instrucción
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

Ejemplo de ejecución de un programa



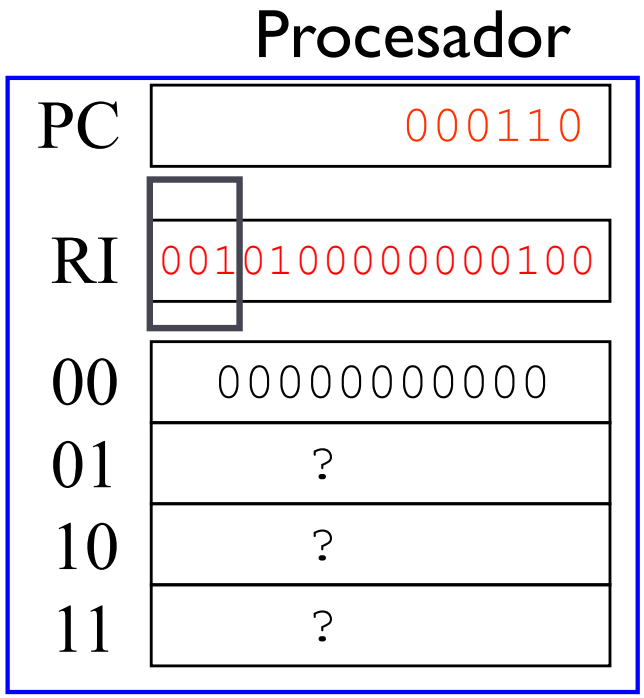
- Lectura de la instrucción
- Apuntar a la siguiente instrucción
  - PC ← PC + 1
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

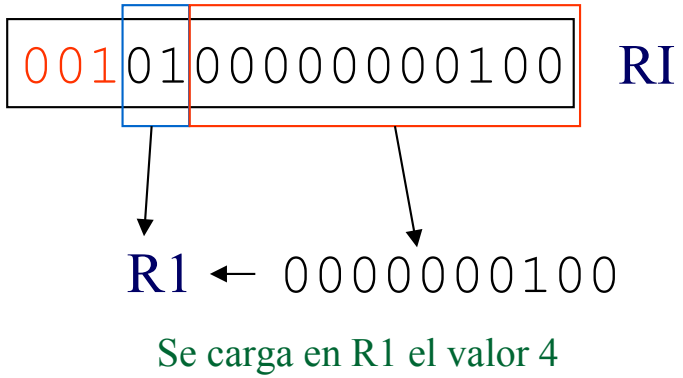
Ejemplo de ejecución de un programa



- Lectura de la instrucción
- Apuntar a la siguiente instrucción
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

Ejemplo de ejecución de un programa

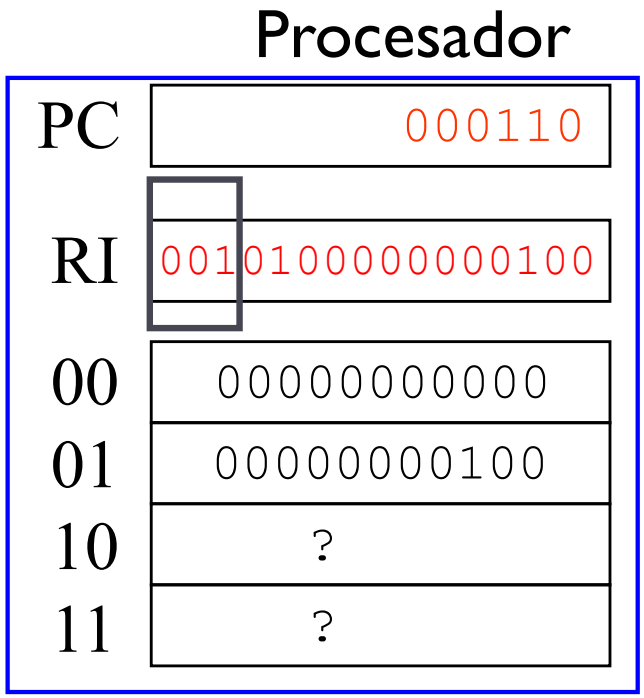


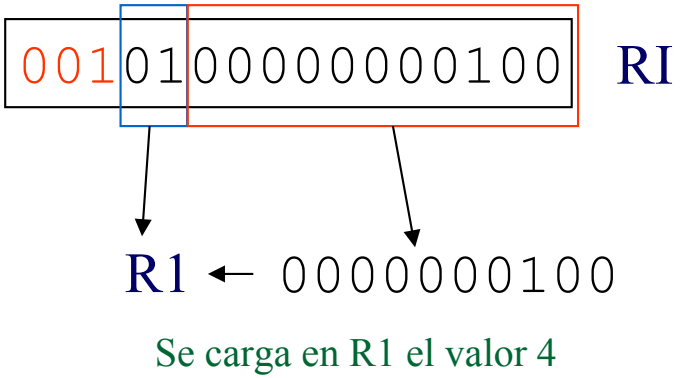


Se carga en R1 el valor 4

- Lectura de la instrucción
- Apuntar a la siguiente instrucción
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

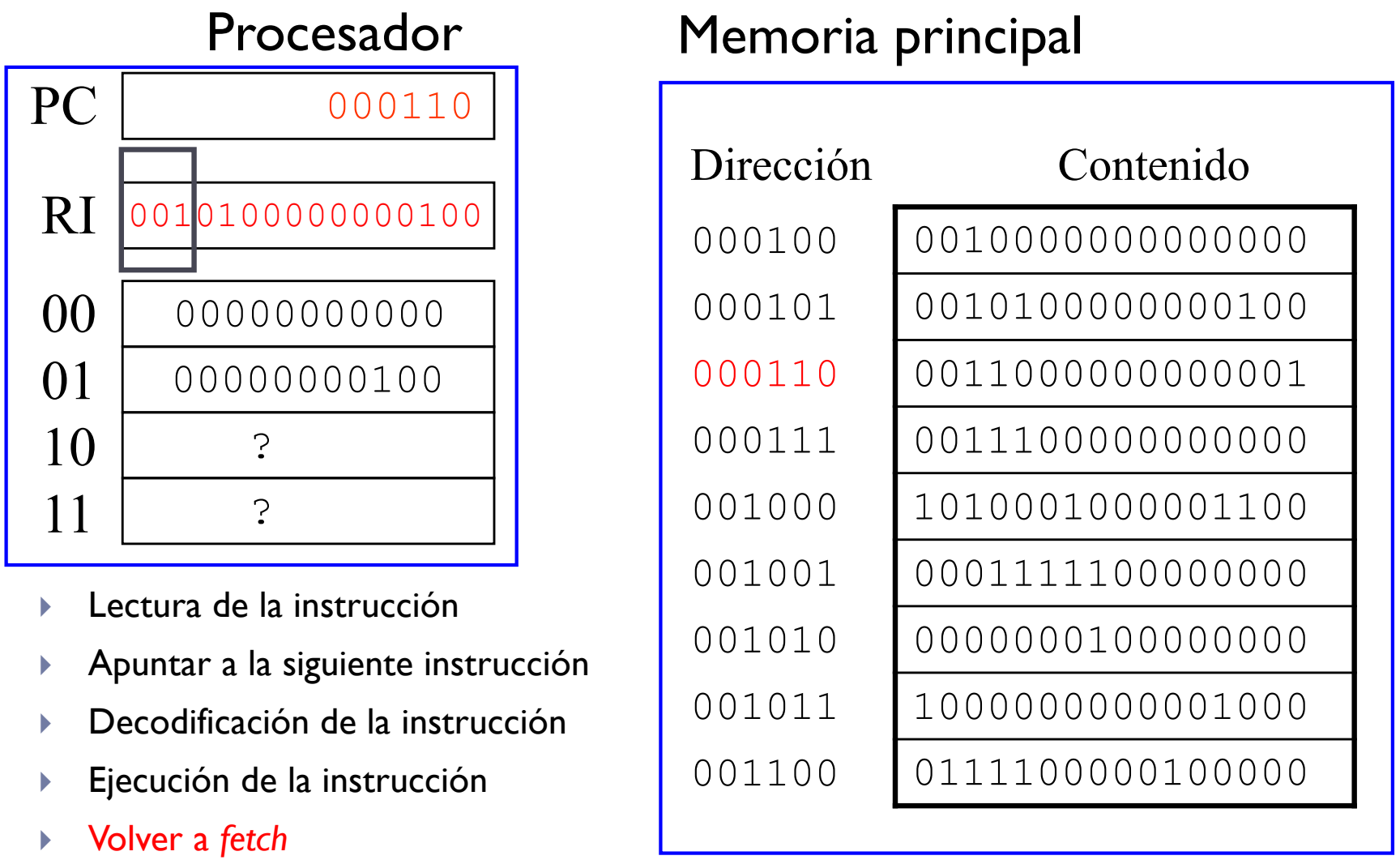
Ejemplo de ejecución de un programa





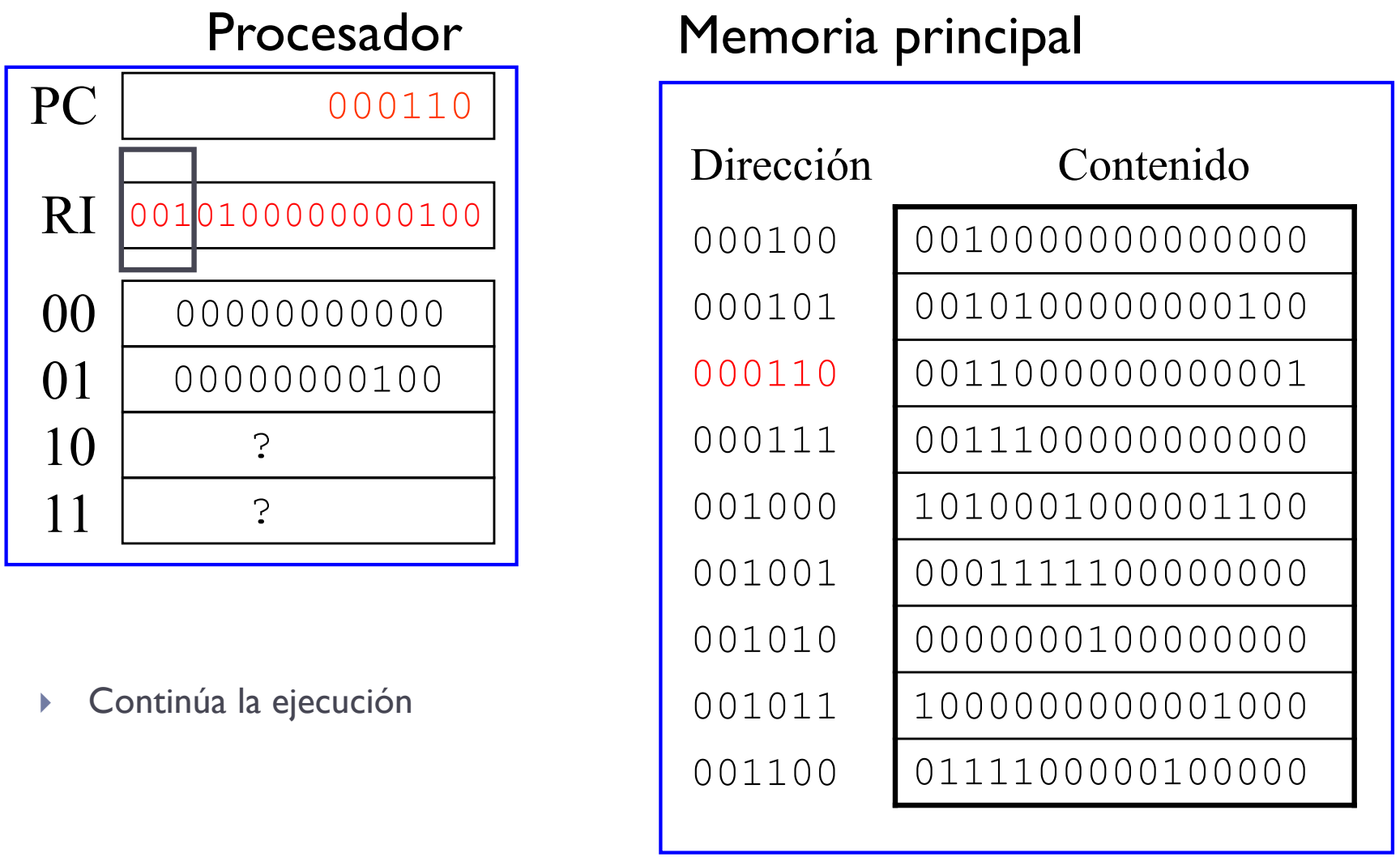
- Lectura de la instrucción
- Apuntar a la siguiente instrucción
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

Ejemplo de ejecución de un programa



- Lectura de la instrucción
- Apuntar a la siguiente instrucción
- Decodificación de la instrucción
- Ejecución de la instrucción
- Volver a *fetch*

Ejemplo de ejecución de un programa



- Continúa la ejecución

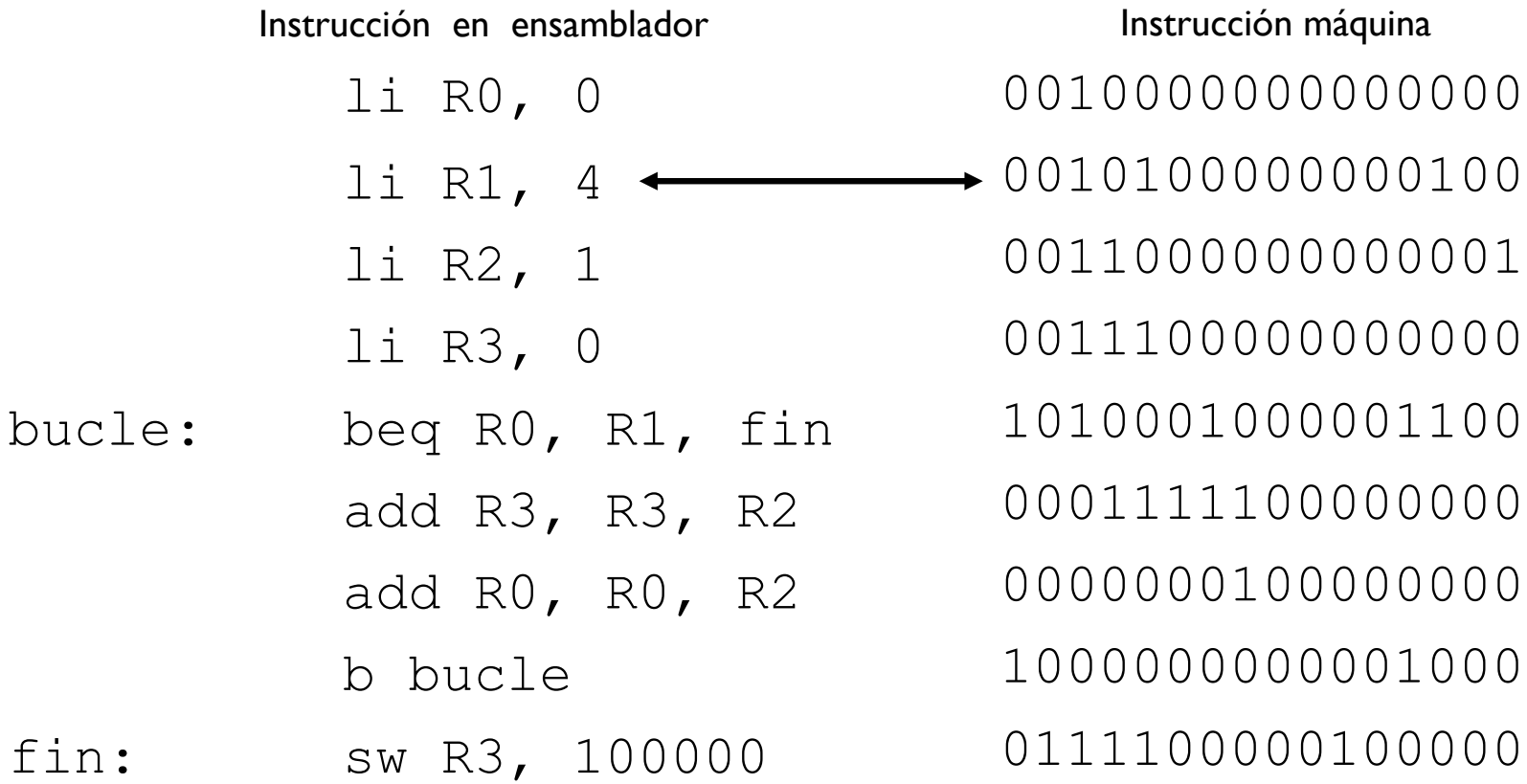
Algoritmo del programa anterior

```
i=0;
s = 0;
while (i < 4)
{
        s = s + 1;
        i = i + 1;
}
```

El programa almacena en la posición de memoria `00000100000` el valor: 1 + 1 + 1 + 1

Lenguaje ensamblador

- Utiliza códigos simbólicos y nemónicos para representar las instrucciones máquina que ejecuta un computador



Contenidos

1. ¿Qué es un computador?
2. Concepto de estructura y arquitectura
3. Elementos constructivos de un computador
4. Computador Von Neumann
5. Instrucciones máquina y programación
6. Fases de ejecución de una instrucción
7. Parámetros característicos de un computador
8. Tipos de computador
9. Evolución histórica

Parámetros característicos de un computador

- Respecto a su arquitectura
  - Ancho de palabra
- Almacenamiento
  - Tamaño
  - Unidades de almacenamiento
- Comunicaciones
  - Ancho de banda
  - Latencia
- Potencia del computador
  - MIPS
  - MFLOPS

Ancho de Palabra

- Número de bits manejados en paralelo en el interior del computador.
  - Influye en el tamaño de los registros (BR)
  - Por tanto, también en la ALU
    - No es lo mismo dos sumas de 32 bits que una sola de 64
  - Por tanto, también en el ancho de los buses
    - Un bus de direcciones de 32 bits 'solo' direcciona 4 GB
- Tamaños típicos ➔ 32 bits, 64 bits

Tamaños privilegiados

- Palabra
  - Información manejada en paralelo en el interior del procesador
  - Típicamente 32/64 bits
- Media palabra
- Doble palabra

- Octeto, carácter o byte
  - Representación de un carácter
  - Típicamente 8 bits

Ejercicio

- Considere un hipotético computador con un ancho de palabra de 20 bits con 60 registros que direcciona la memoria por bytes. Responda a las siguientes preguntas:

  a) ¿Cuántos bits se emplean para las direcciones de memoria?
  b) ¿Cuál es el tamaño de los registros?
  c) ¿Cuántos bits se almacenan en cada posición de memoria?
  d) ¿Cuántas posiciones de memoria se pueden direccionar? Exprese el resultado en KB. e) ¿Cuántos bits se necesitan para identificar a los registros?

Tamaño de la Memoria

- Tamaño de la memoria principal (RAM)
  - Capacidad habitual: 512MB – 4 GB
  - Se expresa en octetos o bytes
- Tamaño de la memoria auxiliar (Capacidad de almacenamiento de dispositivo de memoria secundaria)
  - Papel: pocos bytes
  - Diskette: 1,44 KB
  - CD-ROM: 600 MB
  - DVD: 4.7GB
  - Blu-ray: 50 GB
  - Disco duro: 10 GB – 2 TB

Unidades para tamaño

- Normalmente se expresa en octetos o bytes:

Nombre	Abr	Factor	SI
Kilo	K	$2^{10} = 1{,}024$	$10^{3} = 1{,}000$
Mega	M	$2^{20} = 1{,}048{,}576$	$10^{6} = 1{,}000{,}000$
Giga	G	$2^{30} = 1{,}073{,}741{,}824$	$10^{9} = 1{,}000{,}000{,}000$
Tera	T	$2^{40} = 1{,}099{,}511{,}627{,}776$	$10^{12} = 1{,}000{,}000{,}000{,}000$
Peta	P	$2^{50} = 1{,}125{,}899{,}906{,}842{,}624$	$10^{15} = 1{,}000{,}000{,}000{,}000{,}000$
Exa	E	$2^{60} = 1{,}152{,}921{,}504{,}606{,}846{,}976$	$10^{18} = 1{,}000{,}000{,}000{,}000{,}000{,}000$
Zetta	Z	$2^{70} = 1{,}180{,}591{,}620{,}717{,}411{,}303{,}424$	$10^{21} = 1{,}000{,}000{,}000{,}000{,}000{,}000{,}000$
Yotta	Y	$2^{80} = 1{,}208{,}925{,}819{,}614{,}629{,}174{,}706{,}176$	$10^{24} = 1{,}000{,}000{,}000{,}000{,}000{,}000{,}000{,}000$

Unidades para tamaño

- En comunicación se se utilizan potencias de 10
  - 1 Kb = 1000 bits
  - 1 KB = 1000 bytes
- En almacenamiento algunos fabricantes no utilizan potencias de dos, sino potencias de 10:
  - kilobyte 1 KB = 1.000 bytes 10^3 bytes
  - megabyte 1 MB = 1.000 KB 10^6 bytes
  - gigabyte 1 GB = 1.000 MB 10^9 bytes
  - terabyte 1 TB = 1.000 GB 10^{12} bytes
  - …..

Ejercicio

- ¿Cuántos bytes tiene un disco duro de 200 GB?

- ¿Cuántos bytes por segundo transmite mi ADSL de 20 Mb?

Ejercicio (solución)

- ¿Cuántos bytes tiene un disco duro de 200 GB?
  - 200 GB = 200 * 109bytes = 186.26 Gigabytes

- ¿Cuántos bytes por segundo transmite mi ADSL de 20 Mb?
  - B → Byte
  - b → bit.
  - 20 Mb = 20 * 10^6bits = 20 * 10^6 / 8 bytes = 2.38 Megabytes por segundo

Ancho de banda

- Varias interpretaciones:
  - Caudal de información que transmite un bus.
  - Caudal de información que transmite una unidad de E/S.
  - Caudal de información que puede procesar una unidad.
  - Número de bits transferidos por unidad de tiempo.
- Unidades:
  - Kb/s (Kilobits por segundo, no confundir con KB/s)
  - Mb/s (Megabits por segundo, no megabytes por segundo)

Latencia

- Varias interpretaciones:
  - Tiempo transcurrido en la emisión de una petición en un sistema de mensajería fiable.
  - Tiempo transcurrido entre la emisión de una petición y la realización de la acción asociada.
  - Tiempo transcurrido entre la emisión de una petición y la recepción de la respuesta.
- Unidades:
  - s (segundos)

Potencia de cómputo

- Medición de la potencia de cómputo.
- Factores que intervienen:
  - Juego de instrucciones
  - Reloj de la CPU (1 GHz vs 2 GHz vs 4 GHz…)
  - Número de 'cores' (quadcore vs dualcore vs…)
  - Ancho de palabra (32 bits vs 64 bits vs…)
- Formas típicas de expresar potencia de cómputo:
  - MIPS
  - MFLOPS
  - …

MIPS

- Millones de Instrucciones Por Segundo.
- Rango típico: 10-100 MIPS
- No todas las instrucciones tardan lo mismo en ejecutar
  ➔ Depende de qué instrucciones se ejecutan.
- No es fiable 100% como medida de rendimiento.

MFLOPS

- Millones de Operaciones en coma Flotante por Segundo.
- Potencia de cálculo científico.
- MFLOPS < MIPS (operación flotante más compleja que operación normal).
- Computadores vectoriales: MFLOPS > MIPS
- Ejemplo: Itanium 2 → 3,5 GFLOPS

Vectores por segundo

- Potencia de cálculo en la generación de gráficos.
- Aplicable a procesadores gráficos.
- Se pueden medir en:
  - Vectores 2D.
  - Vectores 3D.
- Ejemplo: ATI Radeon 8500 → 3 Millones.

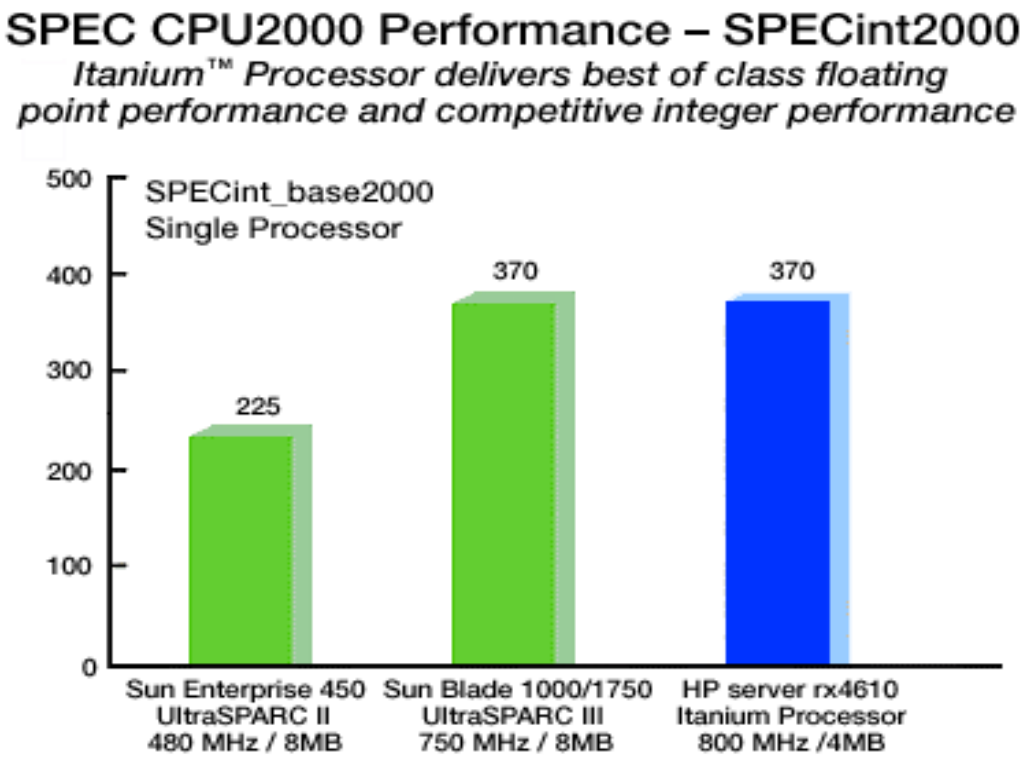
Tests sintéticos

- MIPS y MFLOPS no válidos para comparar distintas máquinas.
- Tests basados en ejecutar un mismo programa en distintas máquinas para compararlas.
- Miden efectividad Compilador + CPU

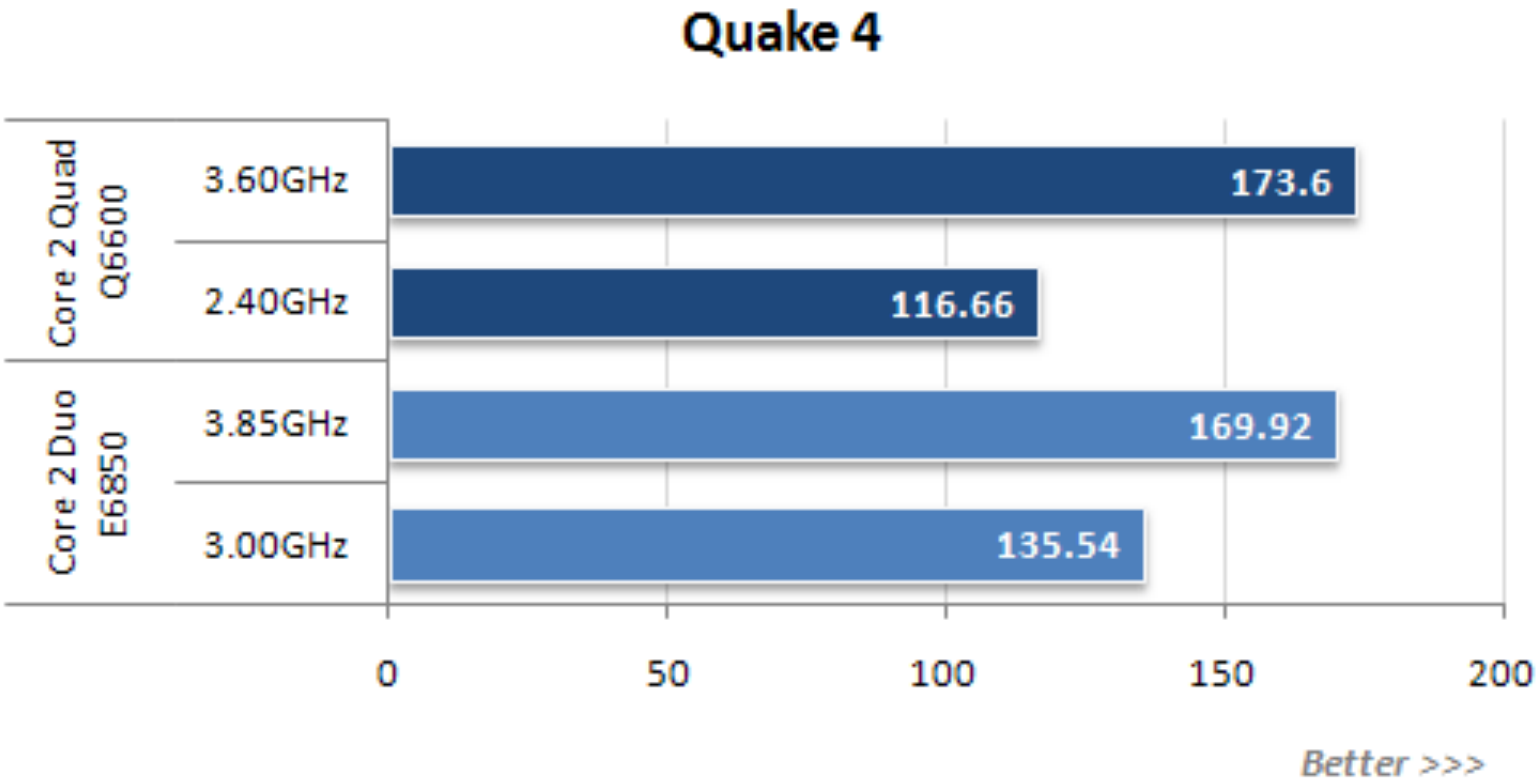
- Los test sintéticos estandarizados ("oficiales") buscan comparar la potencia de dos computadores.
- Es posible usar test sintéticos "no oficiales" para hacerse a la idea de la mejora con la carga de trabajo diaria

Tests sintéticos “oficiales”

- Tests más usados:
  - Linpack.
  - SPEC.

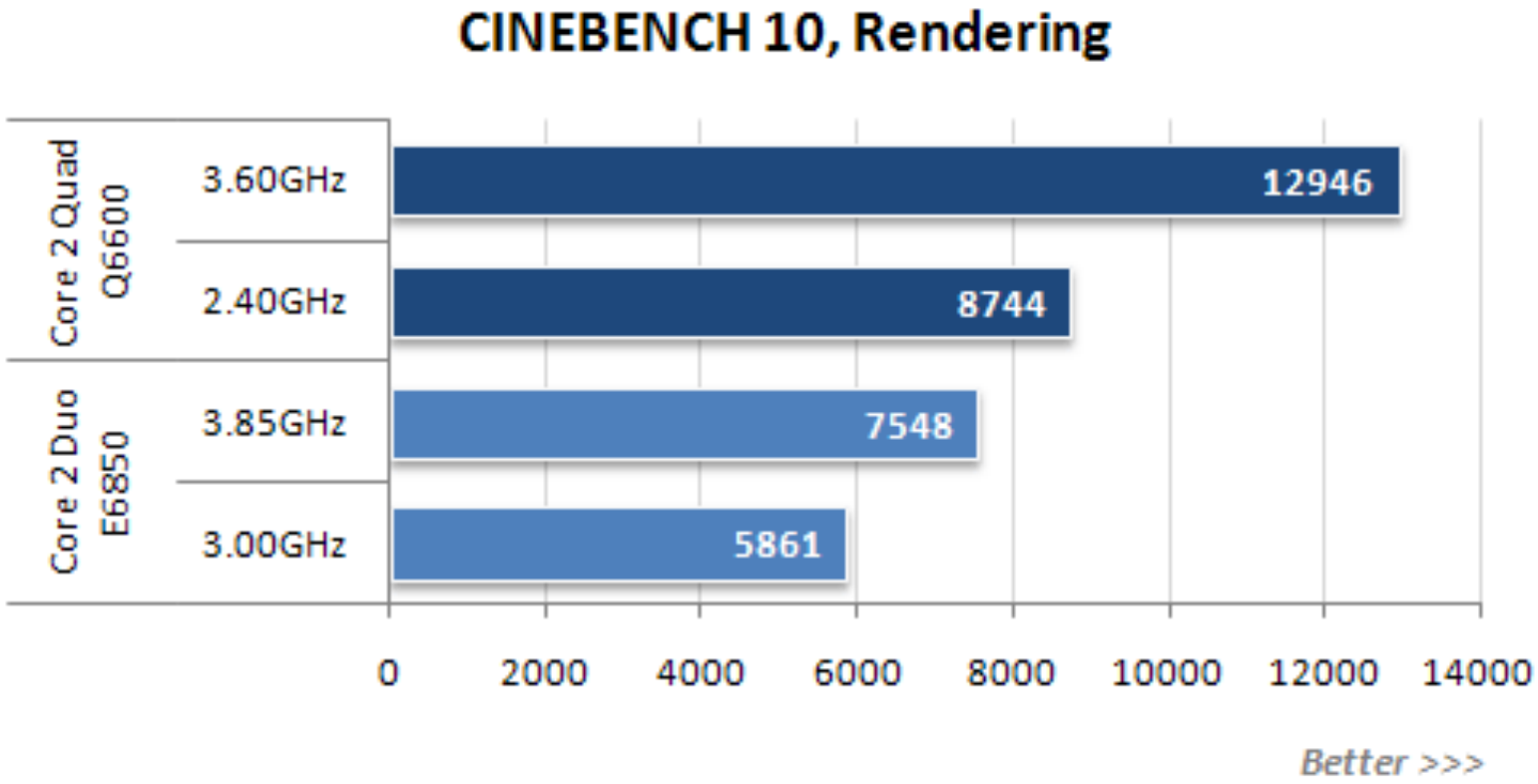


Tests sintéticos “no oficiales”



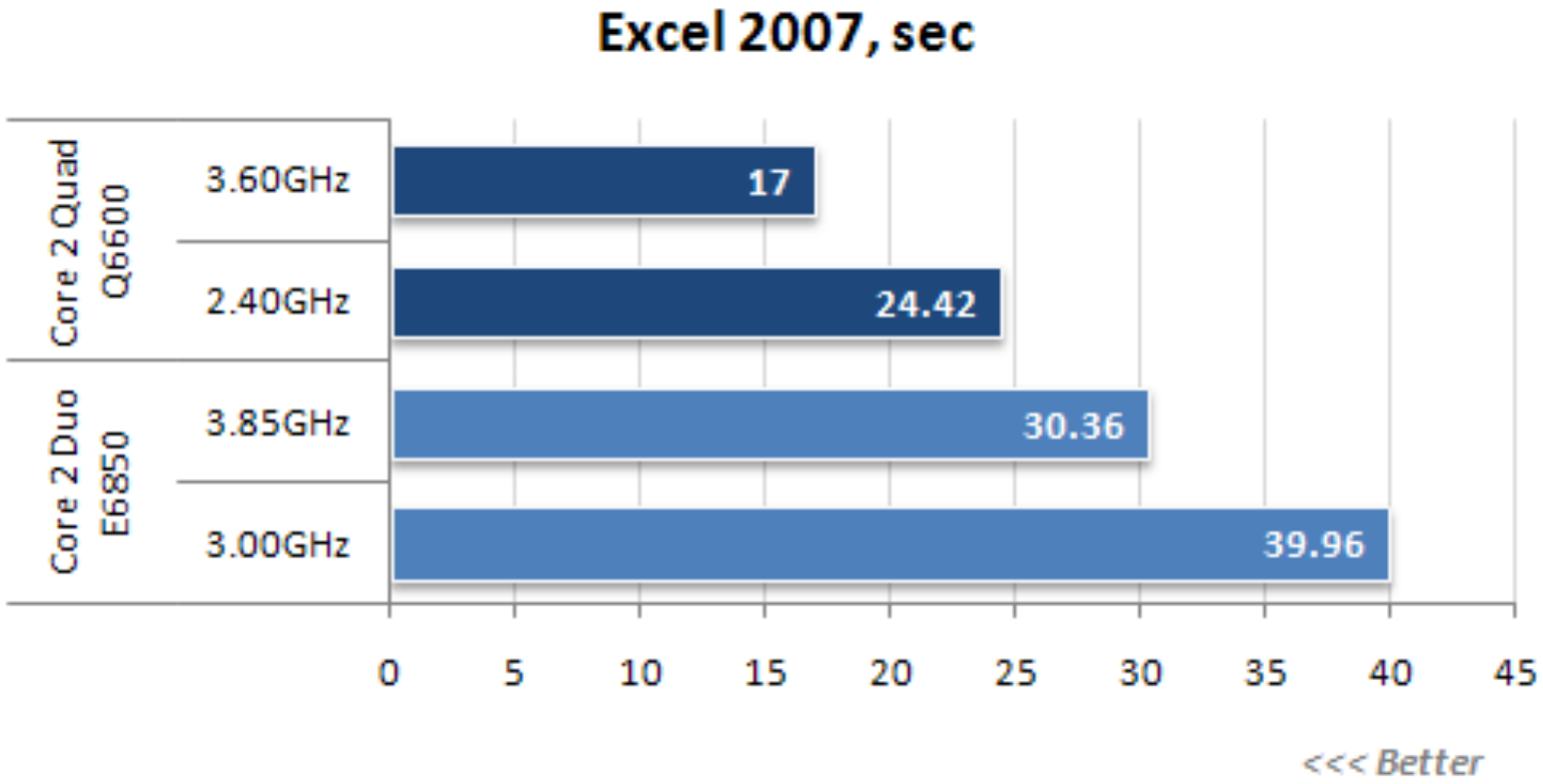
http://www.xbitlabs.com/articles/cpu/display/core2quad-q6600_11.html

Tests sintéticos “no oficiales”



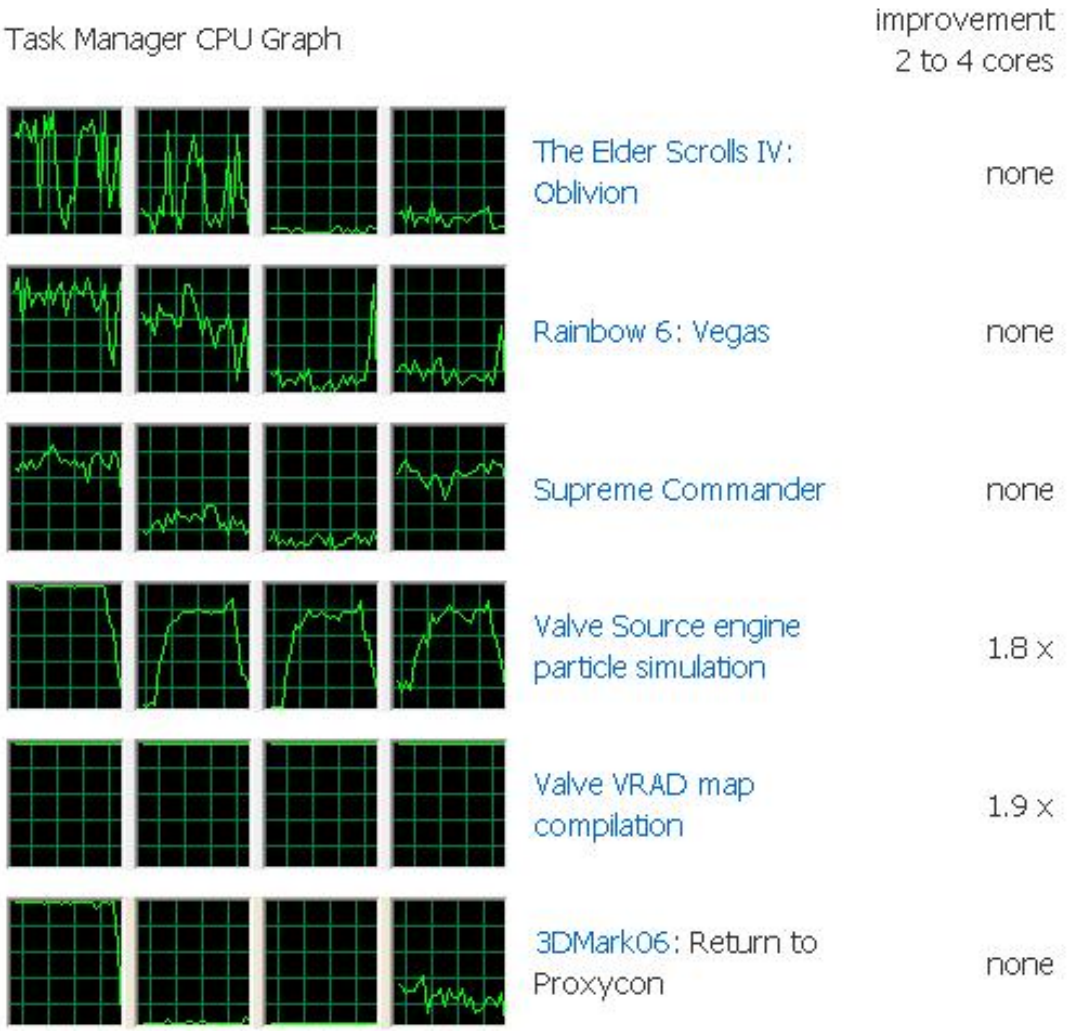
http://www.xbitlabs.com/articles/cpu/display/core2quad-q6600_11.html

Tests sintéticos “no oficiales”



http://www.xbitlabs.com/articles/cpu/display/core2quad-q6600_11.html

Tests sintéticos “no oficiales”



http://www.codinghorror.com/blog/archives/000942.html

Contenidos

1. ¿Qué es un computador?
2. Concepto de estructura y arquitectura
3. Elementos constructivos de un computador
4. Computador Von Neumann
5. Instrucciones máquina y programación
6. Fases de ejecución de una instrucción
7. Parámetros característicos de un computador
8. Tipos de computadores
9. Evolución histórica

Tipos de computadores

- Dispositivos móviles personales
- Desktop
- Servidores
- Clusters
- Empotrados

Tipos de computadores

- Desktop
  - Diseñados para ofrecer un buen rendimiento a los usuarios
  - Actualmente, la mayor parte son portátiles
  - Aspectos de diseño:
    - Relación precio-rendimiento
    - Energía
    - Rendimiento de los gráficos

Tipos de computadores

- Dispositivos móviles personales
  - Dispositivos sin cables con interfaz de usuario multimedia
  - Móviles, tablets,…
  - Aspectos de diseño:
    - Precio
    - Energía
    - Rendimiento
    - Tiempo de respuesta

Tipos de computadores

- Servidores
  - Usados para ejecutar aplicaciones de alto rendimiento o escala
  - Dan servicio a múltiples usuarios de forma simultánea
  - Aspectos de diseño:
    - *Throughput* (Tasa de procesamiento)
    - Disponibilidad
    - Fiabilidad
    - Energía
    - Escalabilidad

Tipos de computadores

- Clusters
  - Conjunto de computadores conectados mediante una red que actúa como un único computador de más prestaciones
  - Utilizando en supercomputadores y grandes centros de datos
  - Aspctos de diseño:
    - Precio-rendimiento
    - *Throughput* (Tasa de procesamiento)
    - Disponibilidad
    - Fiabilidad
    - Energía
    - Escalabilidad

Tipos de computadores

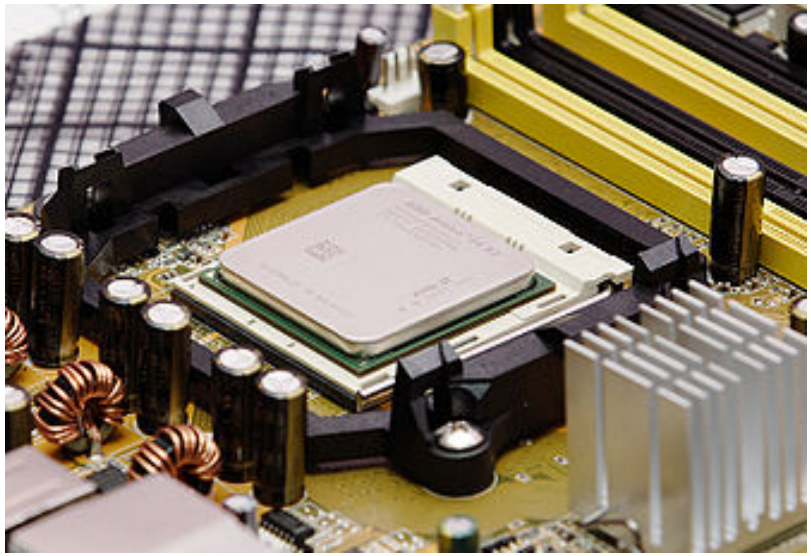
- Empotrados
  - Computador que se encuentra dentro de otro sistema para controlar su funcionamiento
    - Lavadoras, TV, MP3, consolas de videojuegos, etc.
  - Aspectos de diseño:
    - Precio
    - Energía
    - Rendimiento de la aplicación específica

Contenidos

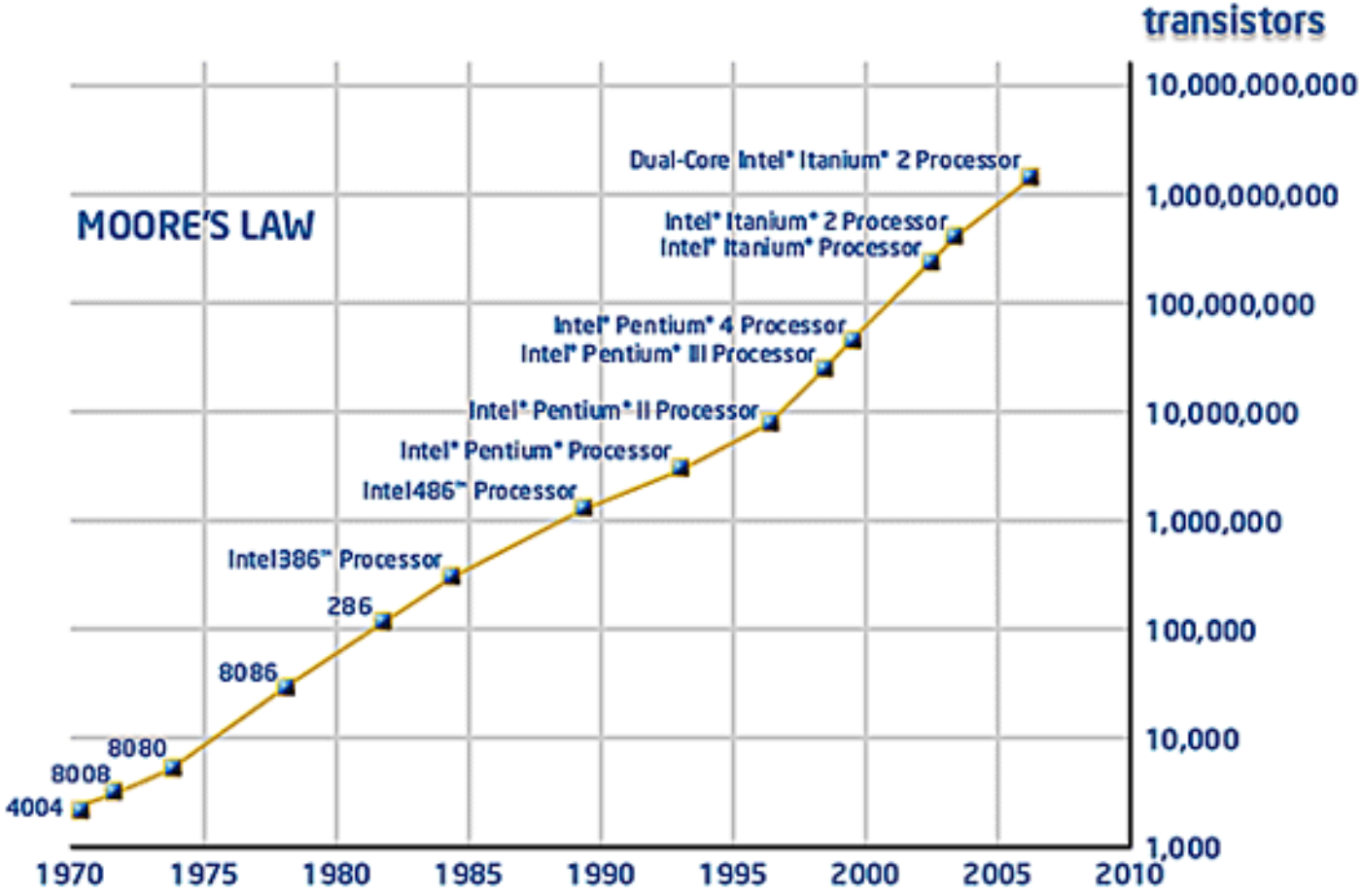
1. ¿Qué es un computador?
2. Concepto de estructura y arquitectura
3. Elementos constructivos de un computador
4. Computador Von Neumann
5. Instrucciones máquina y programación
6. Fases de ejecución de una instrucción
7. Parámetros característicos de un computador
8. Tipos de computadores
9. Evolución histórica

Microprocesador

- Incorpora las funciones de la CPU de un computador en un único circuito integrado

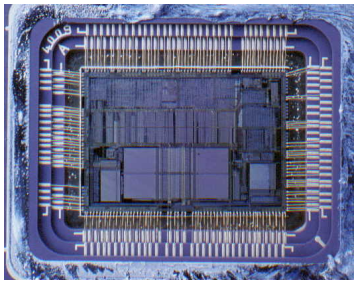


Ley de Moore


transistors
MOORE'S LAW
Dual-Core Intel® Itanium® 2 Processor
Intel® Itanium® 2 Processor
Intel® Itanium® Processor
Intel® Pentium® 4 Processor
Intel® Pentium® III Processor
Intel® Pentium® II Processor
Intel® Pentium® Processor
Intel486™ Processor
Intel386™ Processor
286
8086
8080
8008
4004
10,000,000,000
1,000,000,000
100,000,000
10,000,000
1,000,000
100,000
10,000
1,000
1970
1975
1980
1985
1990
1995
2000
2005
2010

Ley de Moore

- Doblar la densidad implica reducir las dimensiones de sus elementos en un 30%
- En 1971 el Intel 4004 tenía 2.300 transistores con tamaños de 10 micrometros
- Hoy en día se consiguen chips con distancias de 14 nanometros
- Para cumplir la ley de Moore se necesita tecnología cuyo precio se dobla cada 4,4 años



Mejoras en la tecnología

- Memoria
  - Capacidad de DRAM: 2x / 2 años (desde 1996);
    64x en la última década.
- Procesador
  - Velocidad: 2x / 1.5 años (desde 1985);
    100X en la última década.
- Discos
  - Capacidad: 2x / 1 año (desde 1997)
    250X en la última década.

Evolución histórica: bibliografía

- http://history.sandiego.edu/GEN/recording/computer1.html
- http://www.computerhope.com/history/
- http://www.computerhistory.org/
- http://www.computersciencelab.com/ComputerHistory/History.htm
- Museos de informática
- Buscar en google: "Computer history"