

Grupo ARCOS

**uc3m** | Universidad **Carlos III** de Madrid

# Tema 1

## Introducción a los computadores

Estructura de Computadores  
Grado en Ingeniería Informática



# Contenidos

- ▶ **Introducción:**
  - ▶ ¿Qué es un computador?
  - ▶ Elementos constructivos de un computador
  - ▶ Concepto de estructura y arquitectura
- ▶ **Computador Von Neumann:**
  - ▶ Modelo Von Newmann
  - ▶ Instrucciones máquina y programación
  - ▶ Fases de ejecución de una instrucción
- ▶ **Características de un computador y tipos:**
  - ▶ Parámetros característicos de un computador
  - ▶ Tipos de computadores
  - ▶ Evolución histórica

# Contenidos

## ▶ **Introducción:**

- ▶ **¿Qué es un computador?**
- ▶ Elementos constructivos de un computador
- ▶ Concepto de estructura y arquitectura

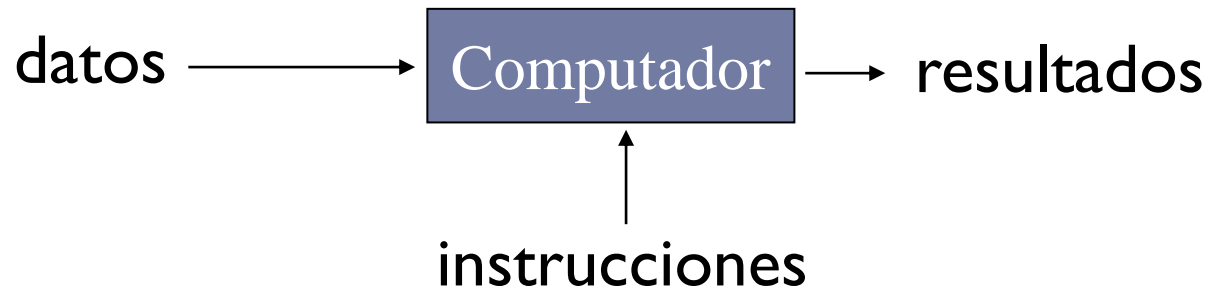
## ▶ **Computador Von Neumann:**

- ▶ Modelo Von Newmann
- ▶ Instrucciones máquina y programación
- ▶ Fases de ejecución de una instrucción

## ▶ **Características de un computador y tipos:**

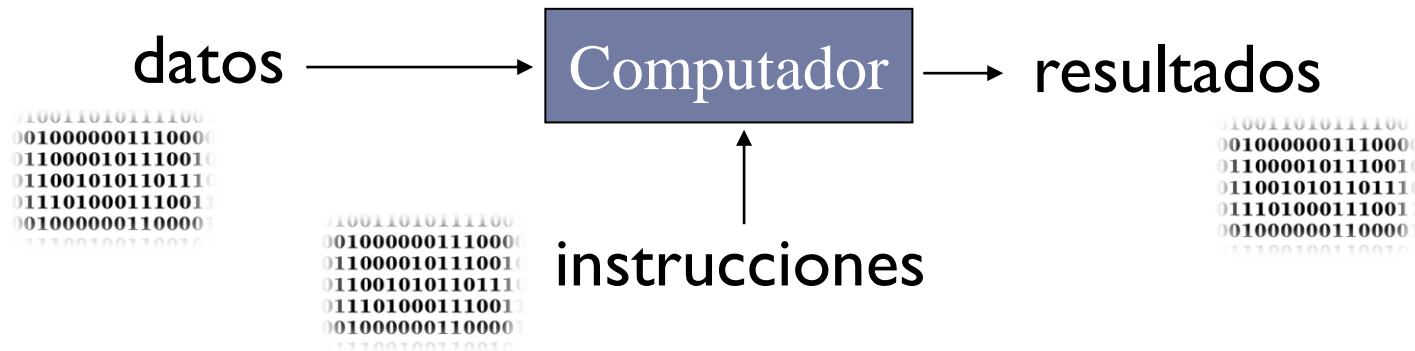
- ▶ Parámetros característicos de un computador
- ▶ Tipos de computadores
- ▶ Evolución histórica

# ¿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.

# Distintos tipo de computadores



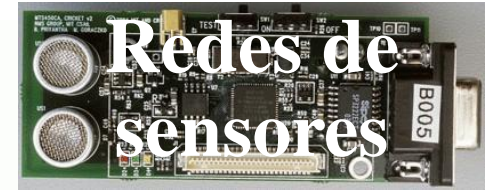
**Smart  
phones**



**Routers**



**Juegos**



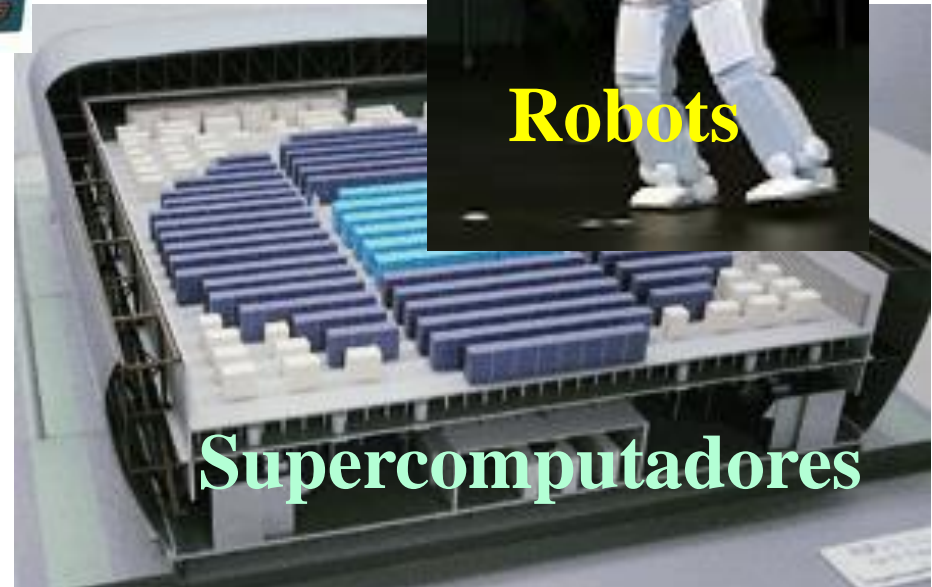
**Redes de  
sensores**



**Automóviles**



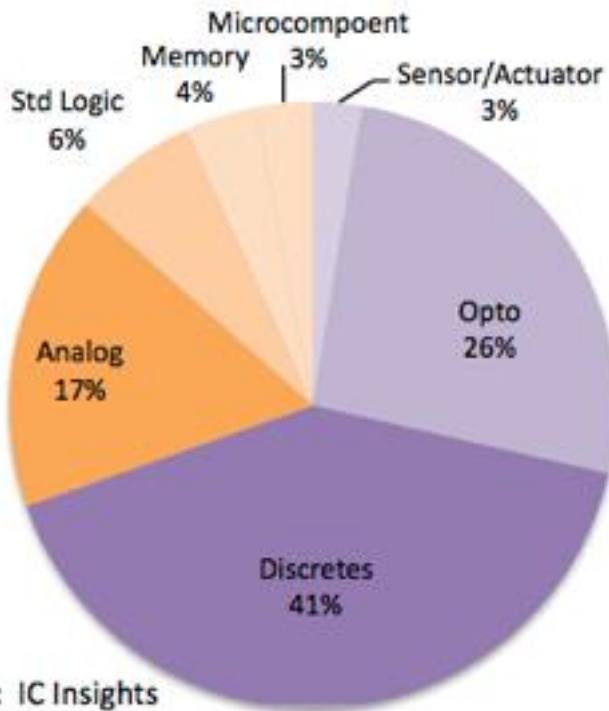
**Robots**



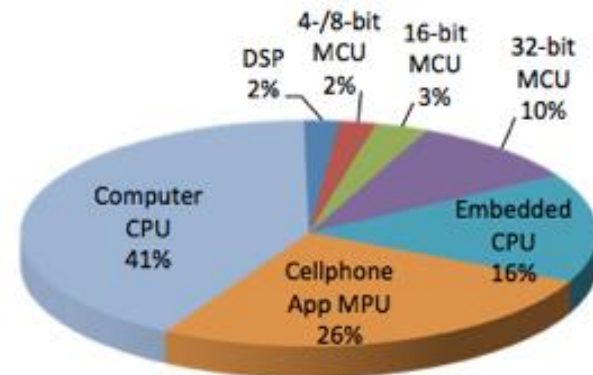
**Supercomputadores**

# Industria de los semiconductores

## 2019F Semiconductor Unit Shipments (1,142.6B)



- Procesadores:  
3% de la industria



# Contenidos

## ▶ **Introducción:**

- ▶ ¿Qué es un computador?
- ▶ **Elementos constructivos de un computador**
- ▶ Concepto de estructura y arquitectura

## ▶ **Computador Von Neumann:**

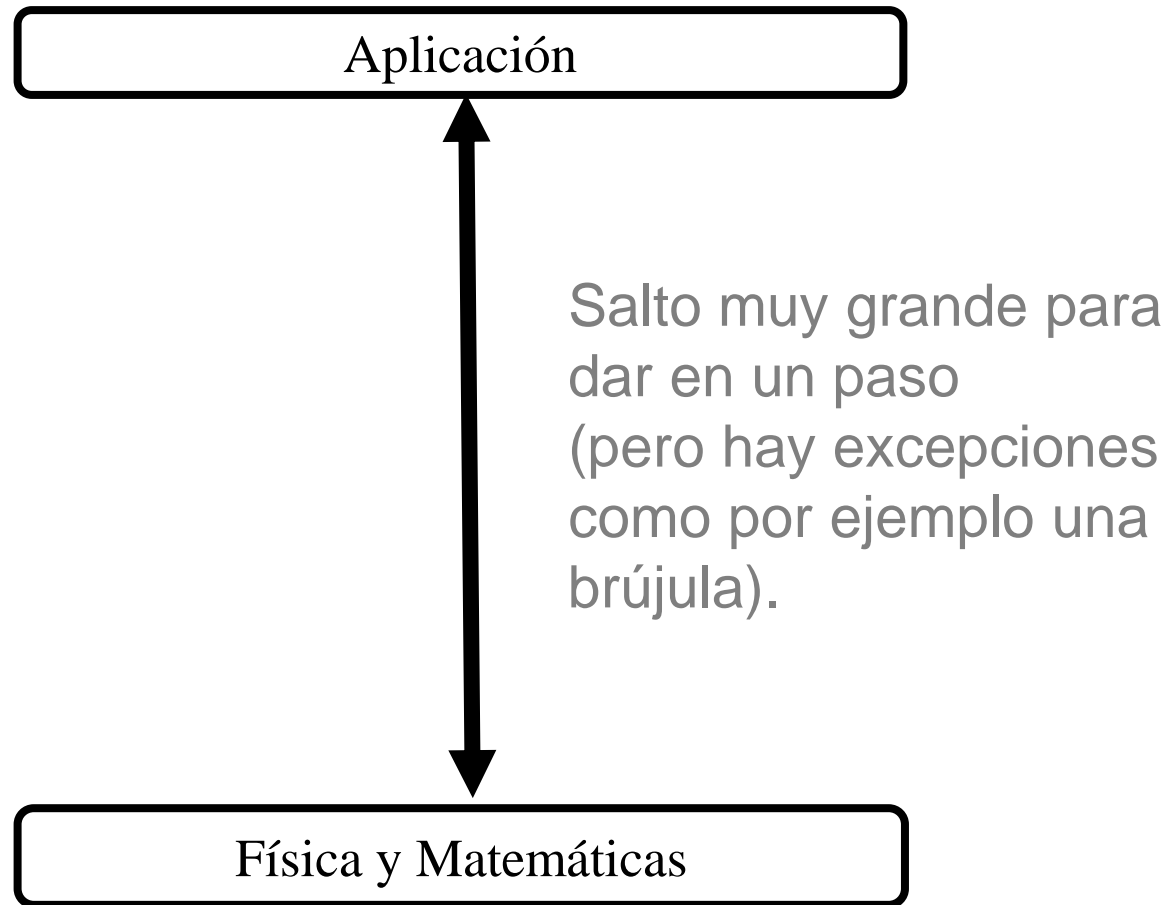
- ▶ Modelo Von Newmann
- ▶ Instrucciones máquina y programación
- ▶ Fases de ejecución de una instrucción

## ▶ **Características de un computador y tipos:**

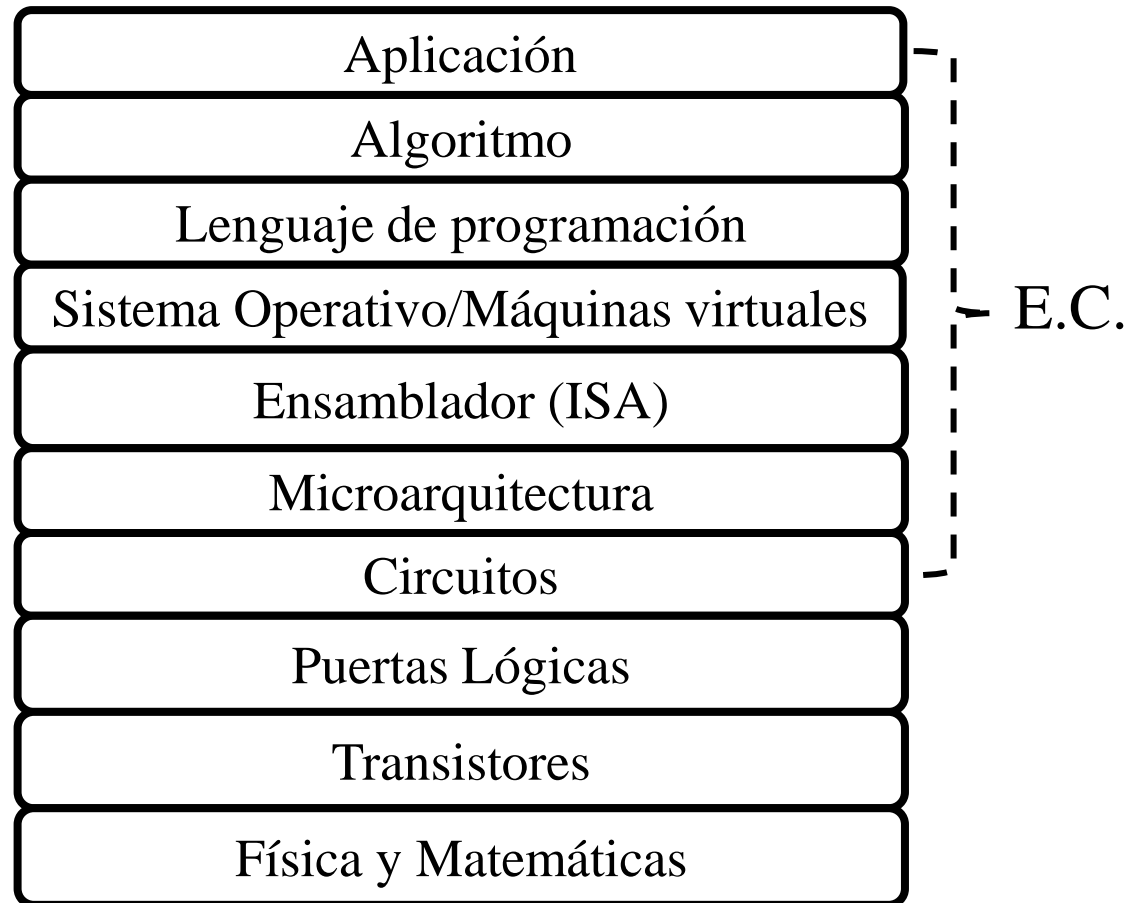
- ▶ Parámetros característicos de un computador
- ▶ Tipos de computadores
- ▶ Evolución histórica



# Elementos constructivos



# Elementos constructivos



# Elementos constructivos



Sistema digital basado en: 0 y 1

# Sistema binario

## ► Binario

$$\begin{array}{cccccccc} X = & 1 & 0 & 1 & 0 & 0 & 1 & 0 & \textcircled{1} \\ & \dots & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & \textcircled{2^0} \end{array}$$

dígito binario  $d_i$   
Peso  $p_i$

$$\text{► Valor} = d_{31} \times 2^{31} + d_{30} \times 2^{30} + \dots + d_i \times 2^i + d_0 \times 2^0$$

# Sistema binario

## ► Binario

$$\begin{array}{cccccccc} X = & 1 & 0 & 1 & 0 & 0 & 1 & 0 & \textcircled{1} \\ & \dots & 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & \textcircled{2^0} \end{array}$$

dígito binario  $d_i$   
Peso  $p_i$

- Valor =  $d_{31} \times 2^{31} + d_{30} \times 2^{30} + \dots + d_i \times 2^i + d_0 \times 2^0$
- ¿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

$$X = \begin{array}{cccccccc} 1 & 0 & 1 & 0 & 0 & 1 & 0 & \textcircled{1} \\ \dots 2^7 & 2^6 & 2^5 & 2^4 & 2^3 & 2^2 & 2^1 & \textcircled{2^0} \end{array}$$

dígito binario  $d_i$   
Peso  $p_i$

$$\text{► Valor} = d_{31} \times 2^{31} + d_{30} \times 2^{30} + \dots + d_i \times 2^i + d_0 \times 2^0$$

► ¿Cuántos valores se pueden representar con  $n$  bits?

$2^n$

► ¿Cuántos bits se necesitan para representar  $m$  'valores'?

$\text{Log}_2(m)$   
por exceso

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

$2^n - 1$

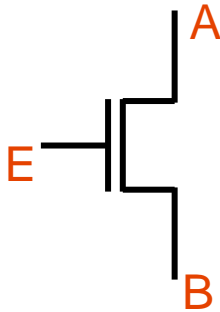
# Elementos constructivos



Elementos constructivos electrónicos...

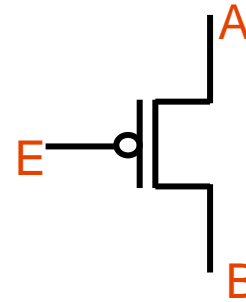
# Transistor

N-MOS



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

P-MOS

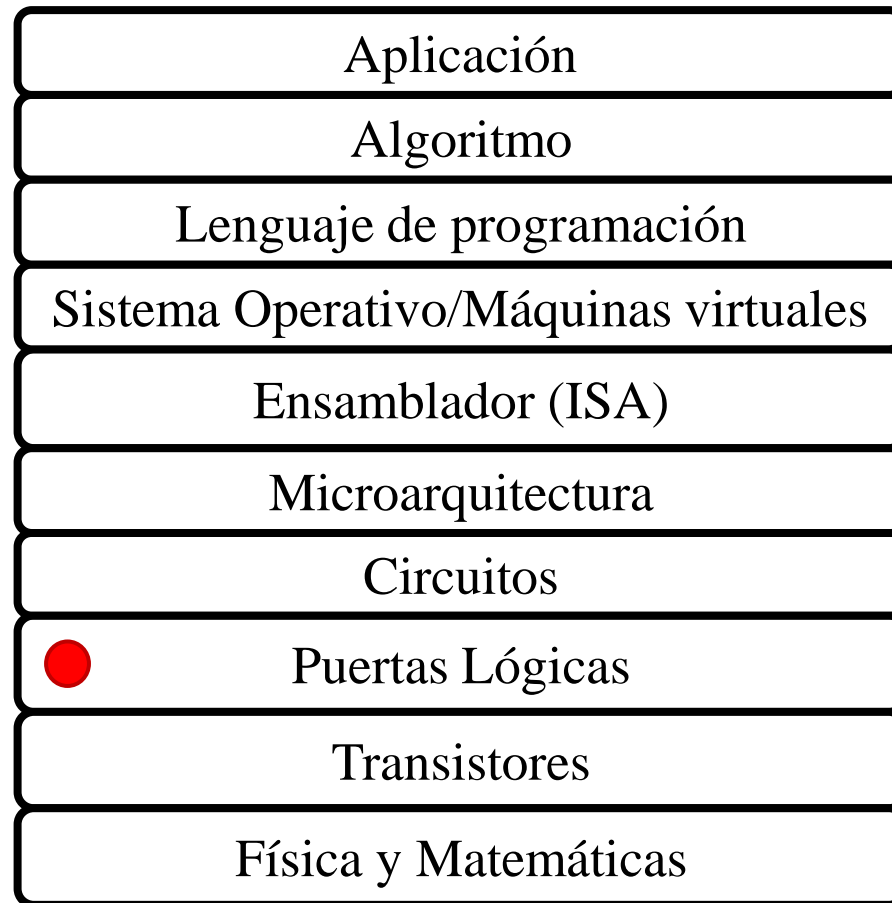


E	Funcionamiento
0	Conecta A con B (circuito abierto)
1	No conecta A con B (circuito cerrado)

- ▶ 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



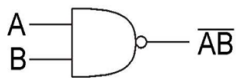
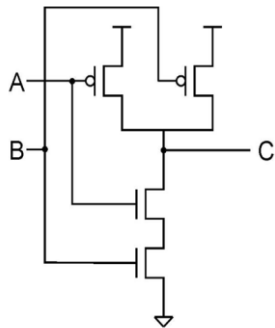
# Elementos constructivos



Elementos constructivos electrónicos...

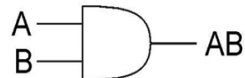
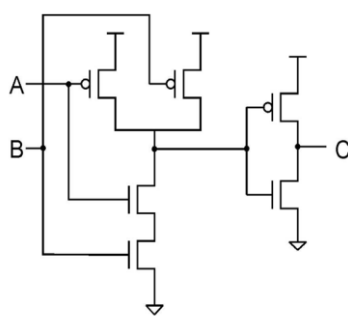
# Puertas lógicas

## NAND



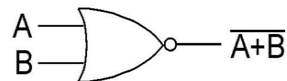
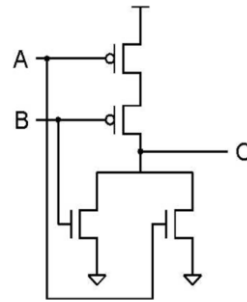
A	B	C
0	0	1
0	1	1
1	0	1
1	1	0

## AND



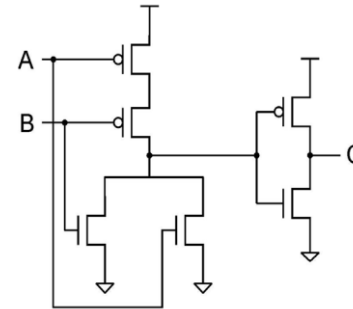
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

## NOR



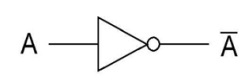
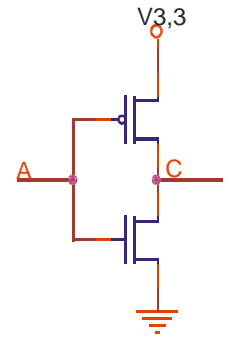
A	B	C
0	0	1
0	1	0
1	0	0
1	1	0

## OR



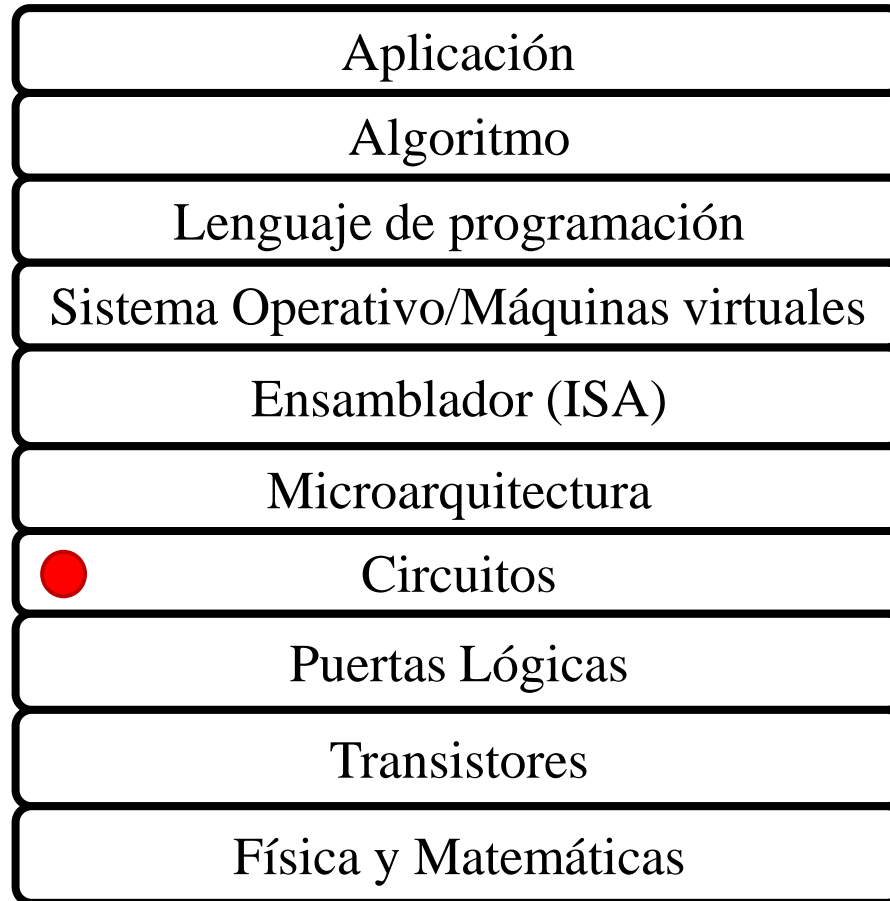
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

## NOT



A	C
1	0
0	1

# Elementos constructivos



Elementos constructivos electrónicos...

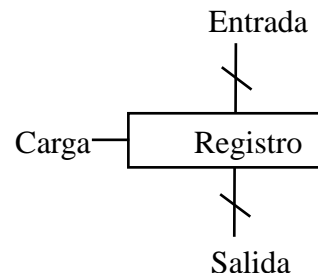
# Circuitos combinacionales y secuenciales

## ▶ Circuitos **combinacionales**:

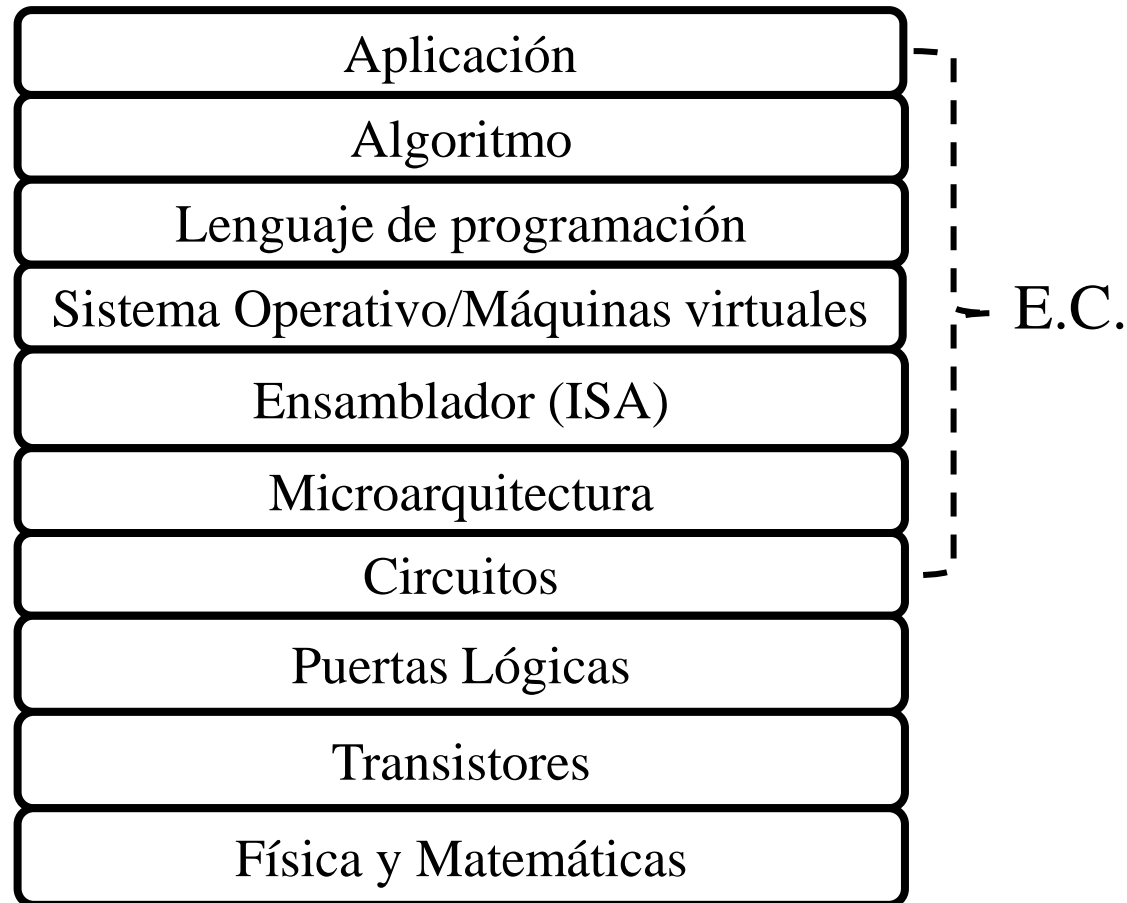
- ▶ La salida depende solo de los valores de entrada
- ▶ Ejemplos:
  - ▶ Decodificadores, Multiplexores, Operadores aritméticos y lógicos, ...

## ▶ Circuitos **secuenciales**:

- ▶ La salida depende de los valores de entrada Y del estado actual (almacenan estado)
- ▶ Ejemplos:
  - ▶ Biestables, Registros, ...



# Elementos constructivos



# Contenidos

## ▶ **Introducción:**

- ▶ ¿Qué es un computador?
- ▶ Elementos constructivos de un computador
- ▶ **Concepto de estructura y arquitectura**

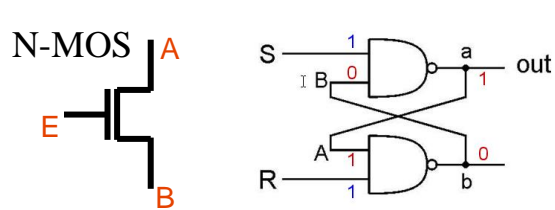
## ▶ **Computador Von Neumann:**

- ▶ Modelo Von Newmann
- ▶ Instrucciones máquina y programación
- ▶ Fases de ejecución de una instrucción

## ▶ **Características de un computador y tipos:**

- ▶ Parámetros característicos de un computador
- ▶ Tipos de computadores
- ▶ Evolución histórica

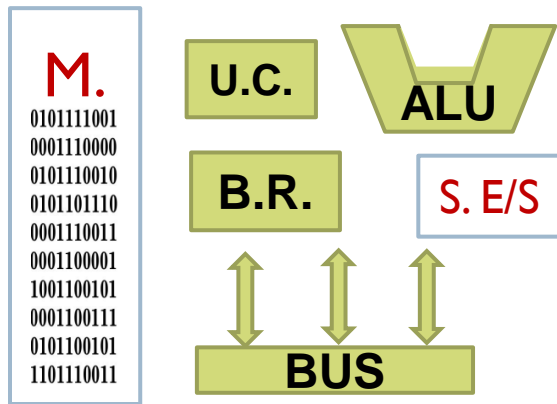
# ¿Qué aspectos hay que conocer en un computador?



**Tecnología:**

▶ Cómo se construyen los componentes

# ¿Qué aspectos hay que conocer en un computador?

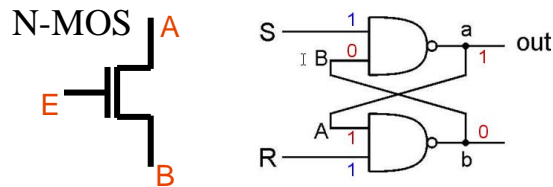


## ► Estructura:

- Componentes y su organización

## ► Tecnología:

- Cómo se construyen los componentes





# ¿Qué aspectos hay que conocer en un computador?



## ► **Arquitectura:**

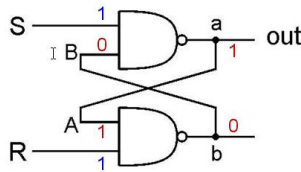
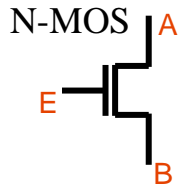
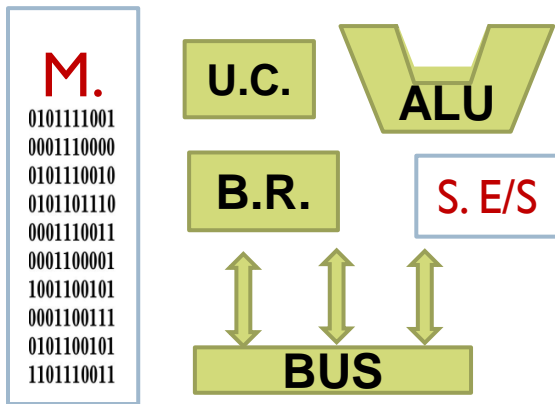
- Atributos visibles para un programador

## ► **Estructura:**

- Componentes y su organización

## ► **Tecnología:**

- Cómo se construyen los componentes



# Arquitectura de un computador:

## **Atributos visibles para un programador**

- ▶ Tipo y formato de datos que es capaz de utilizar el computador
- ▶ Juego de instrucciones que ofrece el computador (ISA, *Instruction Set Architecture*)
- ▶ Número y tamaño de los registros
- ▶ Técnicas de direccionamiento y acceso a la memoria
- ▶ Técnicas y mecanismos de Entrada/Salida (E/S)

# Contenidos

- ▶ **Introducción:**
  - ▶ ¿Qué es un computador?
  - ▶ Elementos constructivos de un computador
  - ▶ Concepto de estructura y arquitectura
- ▶ **Computador Von Neumann:**
  - ▶ **Modelo Von Newmann**
  - ▶ Instrucciones máquina y programación
  - ▶ Fases de ejecución de una instrucción
- ▶ **Características de un computador y tipos:**
  - ▶ Parámetros característicos de un computador
  - ▶ Tipos de computadores
  - ▶ Evolución histórica

# ¿Qué aspectos hay que conocer en un computador?



## ► **Arquitectura:**

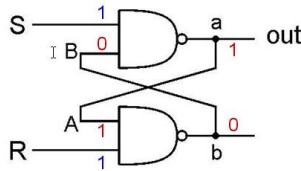
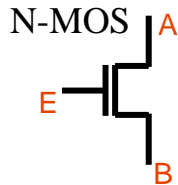
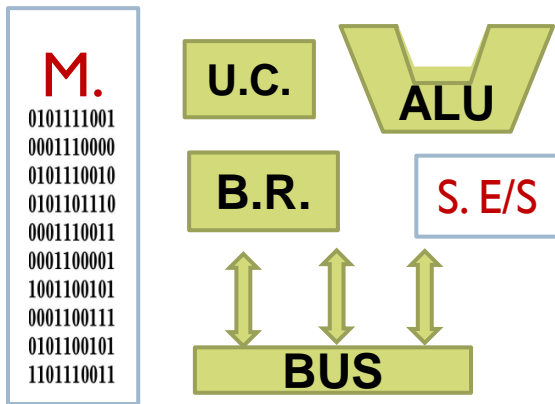
- Atributos visibles para un programador

## ► **Estructura:**

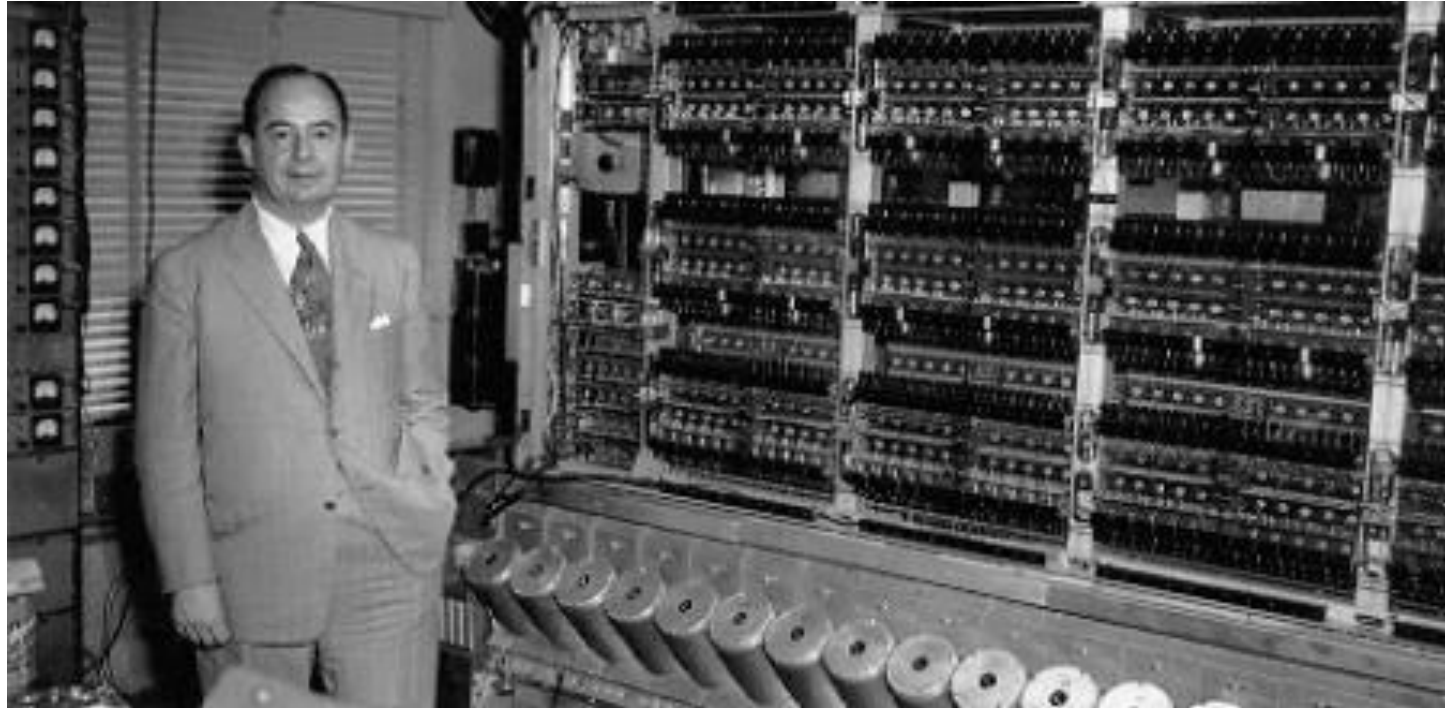
- Componentes y su organización

## ► **Tecnología:**

- Cómo se construyen los componentes

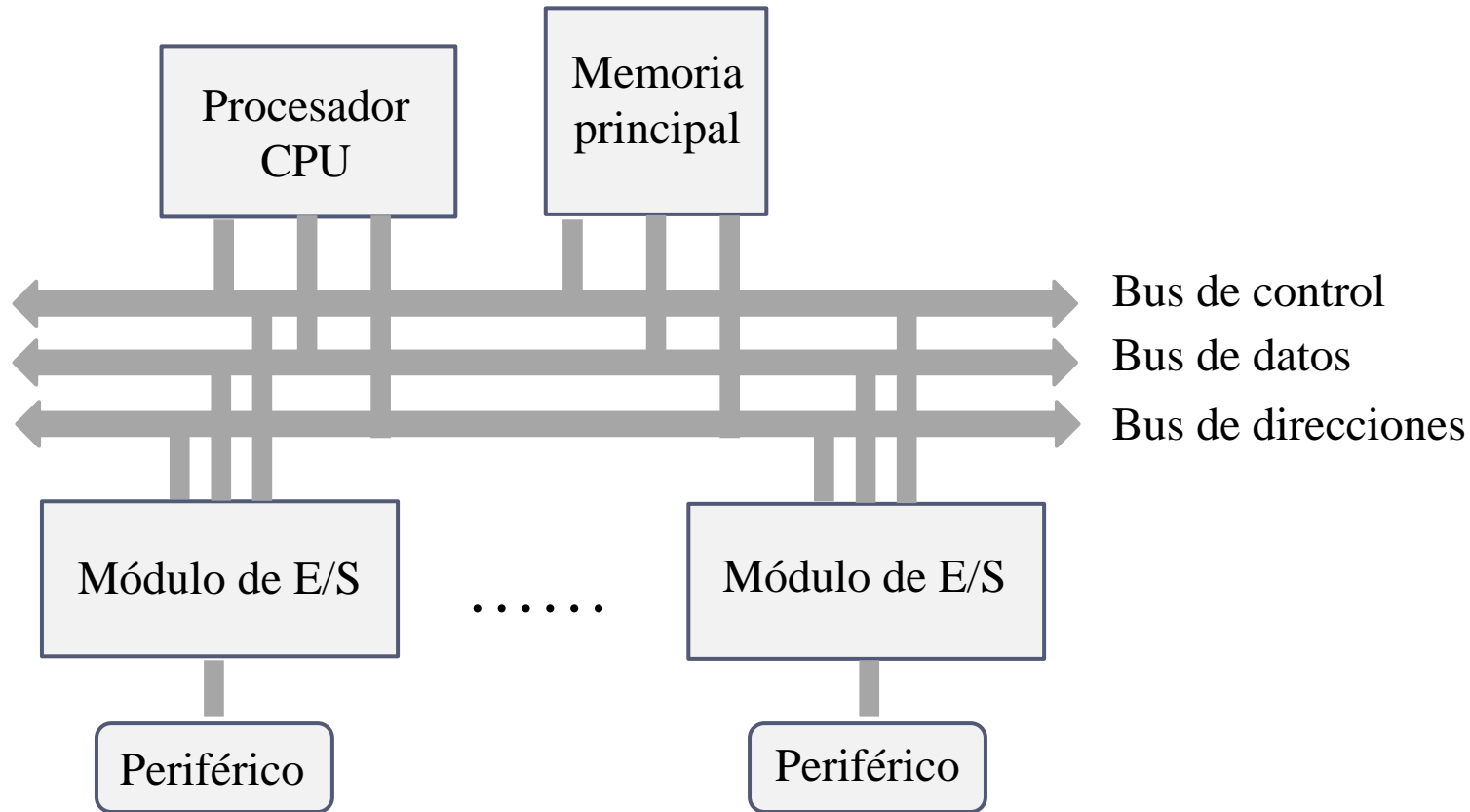


# Computador Von Neumann

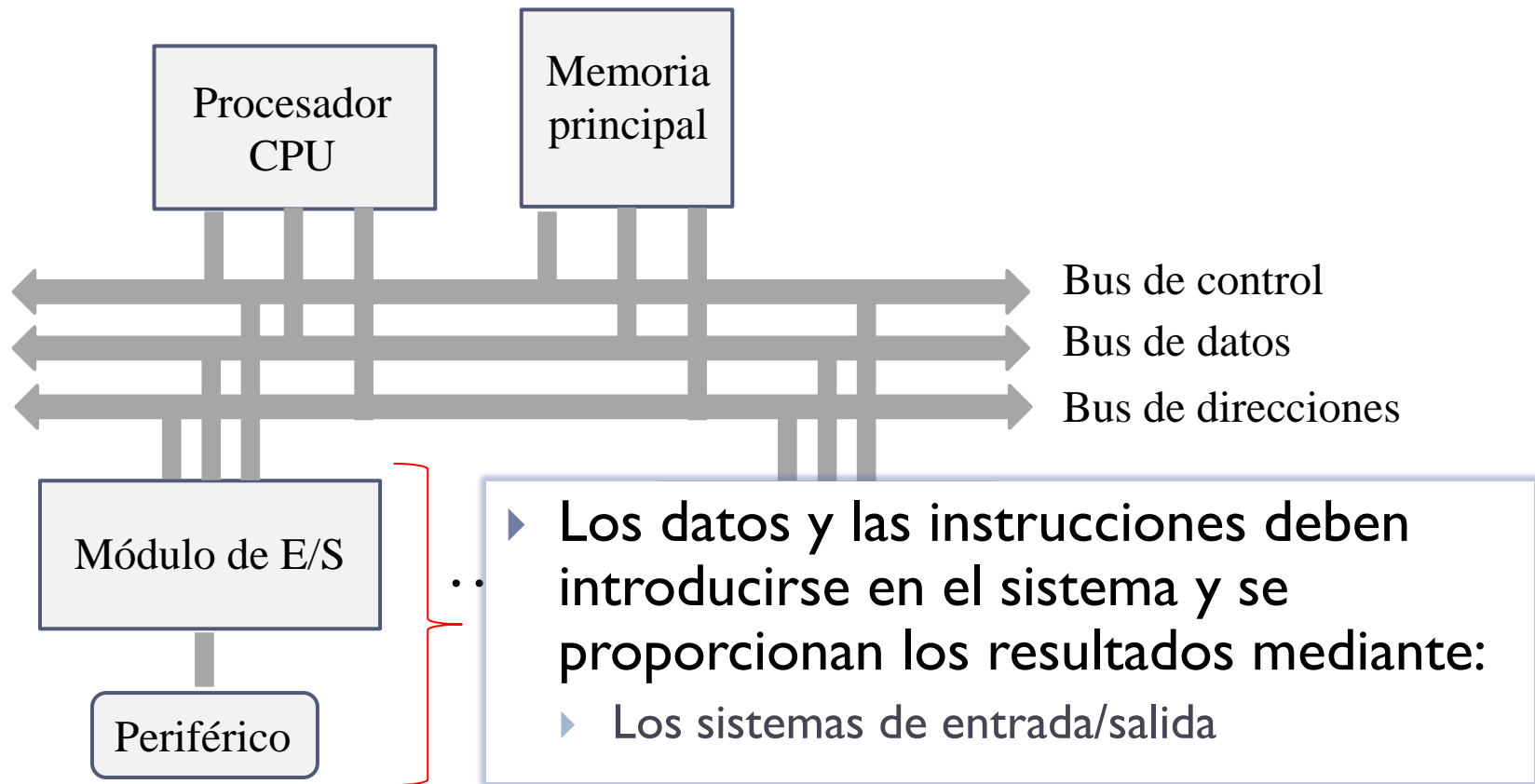


Máquina capaz de ejecutar una serie de instrucciones elementales (instrucciones máquina) que están almacenadas en memoria (son leídas y ejecutadas)

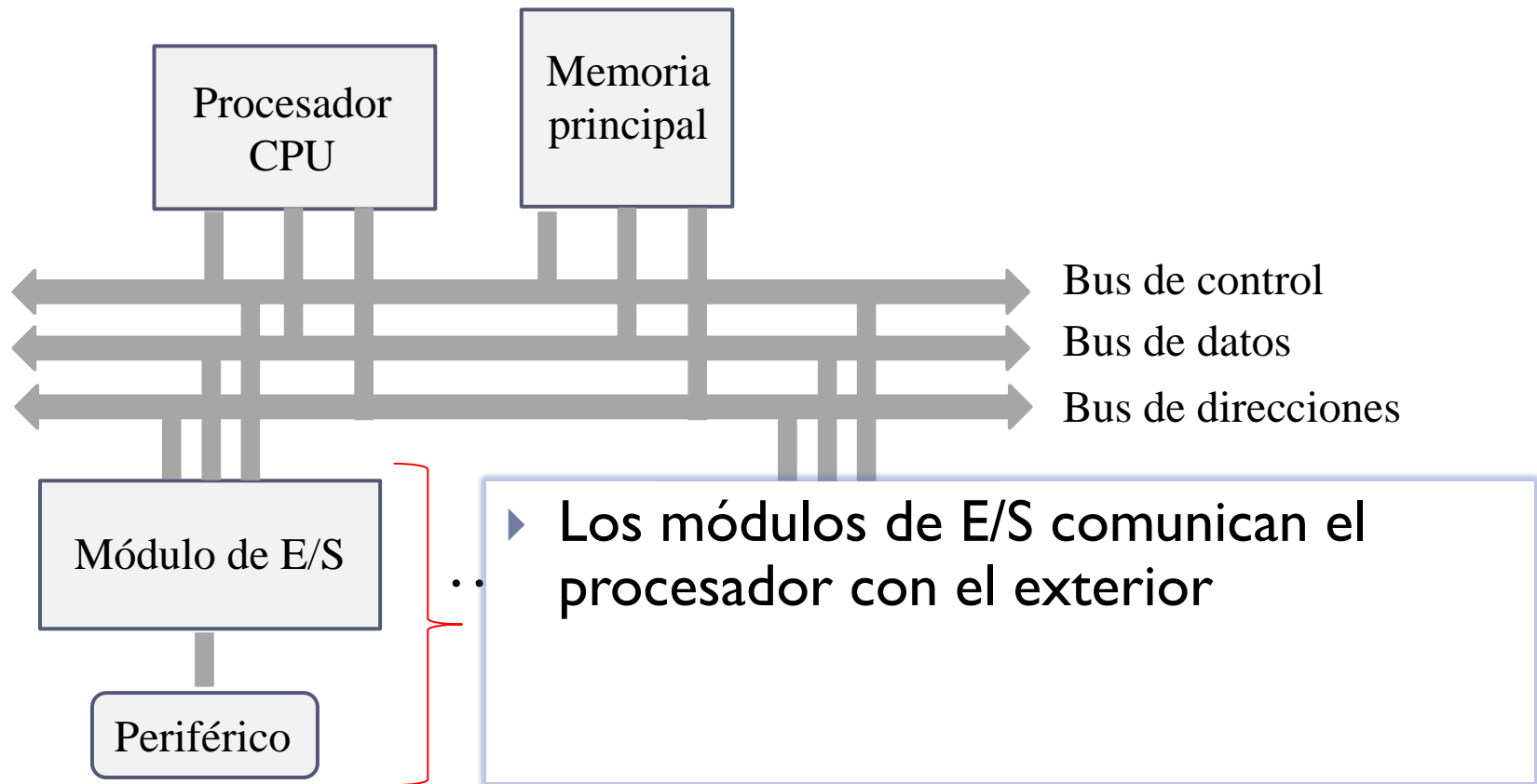
# Arquitectura Von Neumann



# Arquitectura Von Neumann (1 / 4)



# Arquitectura Von Neumann (1 / 4)





# Ejemplo de módulos + periféricos comunicación

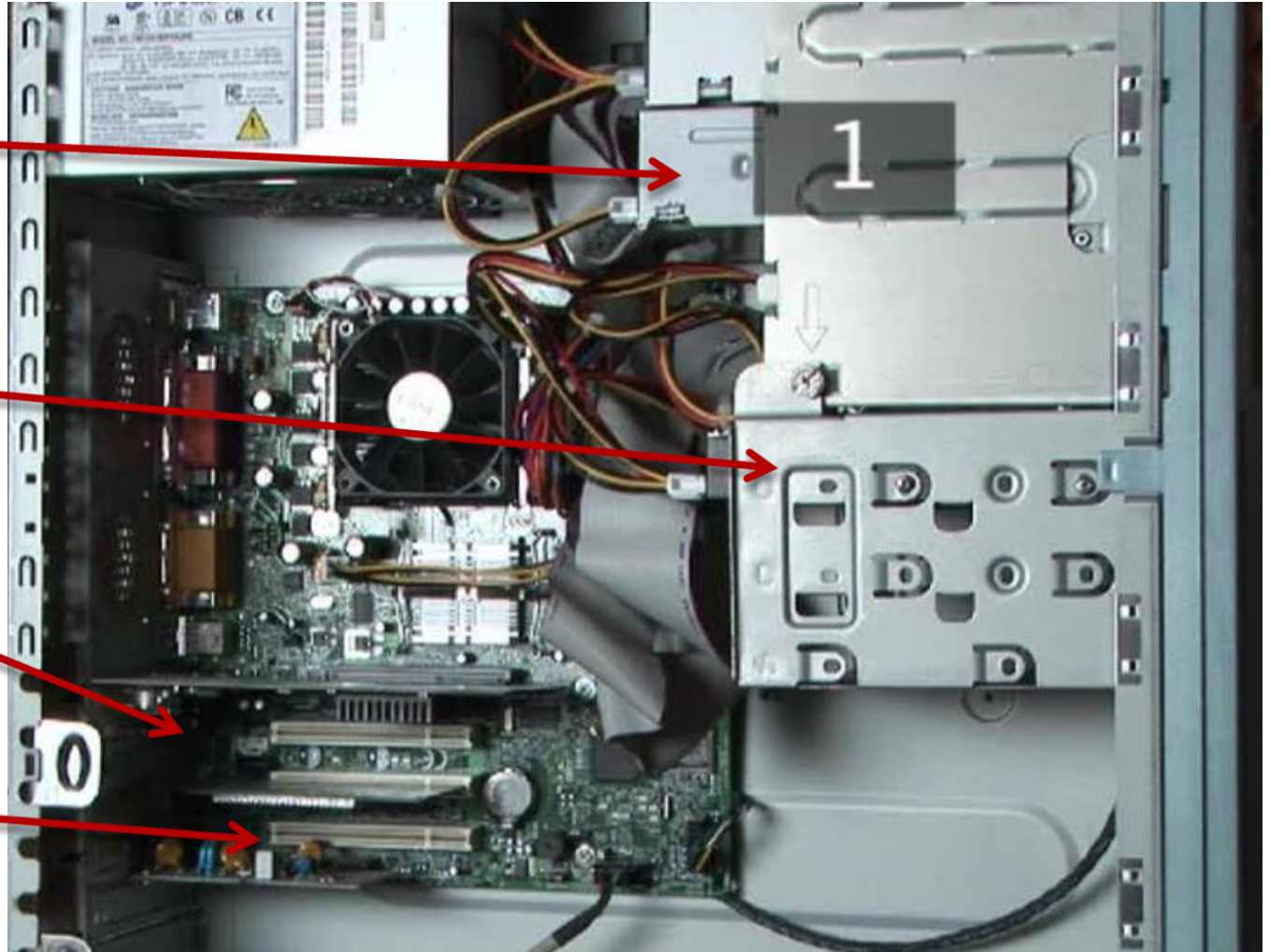
CD-ROM/  
DVD-ROM/  
BluRay/...

Disco duro

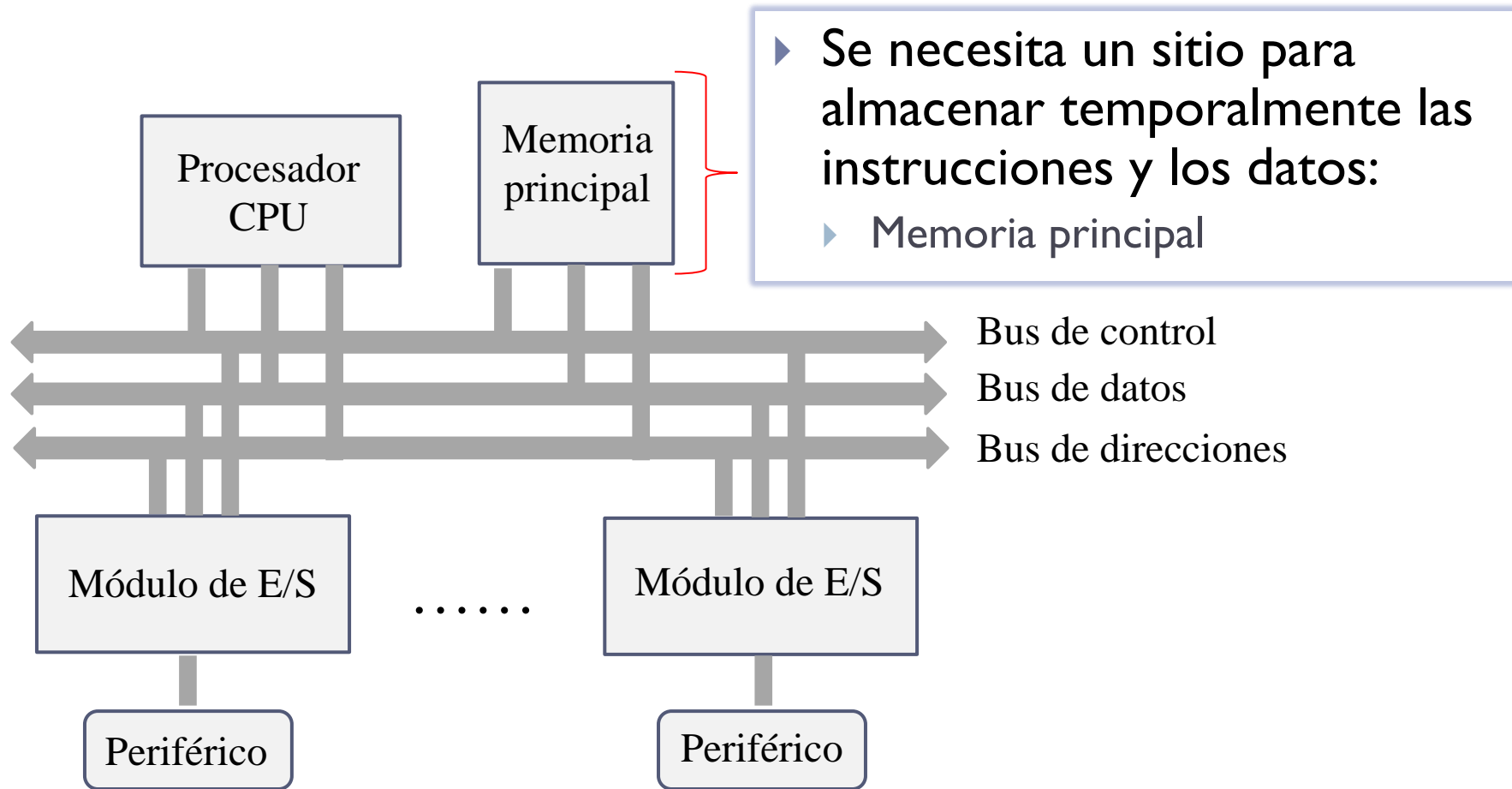


Tarjeta  
de red

Tarjeta  
de sonido

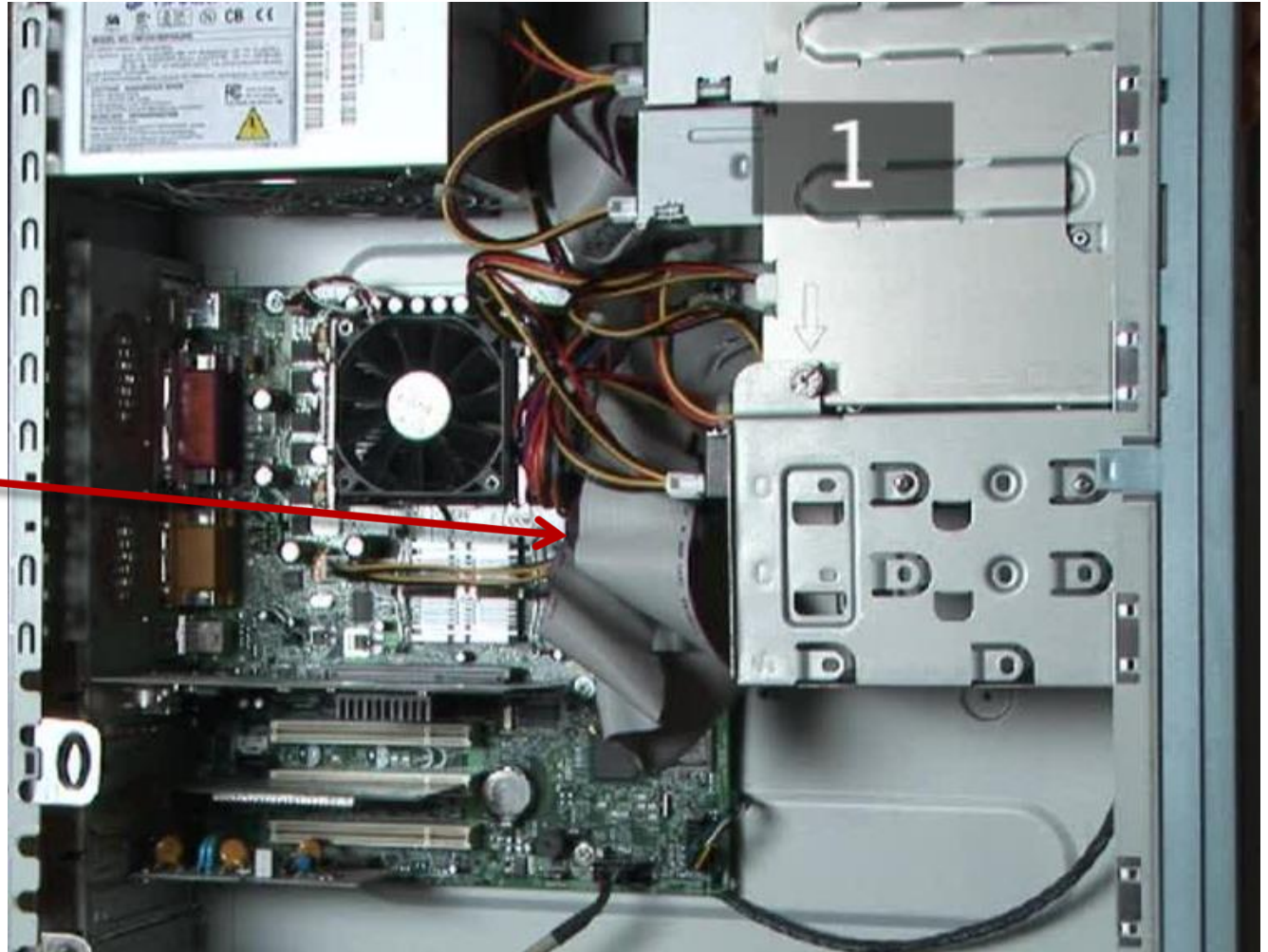
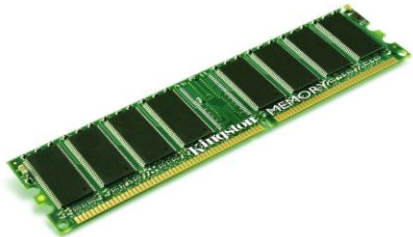


# Arquitectura Von Neumann (2/4)



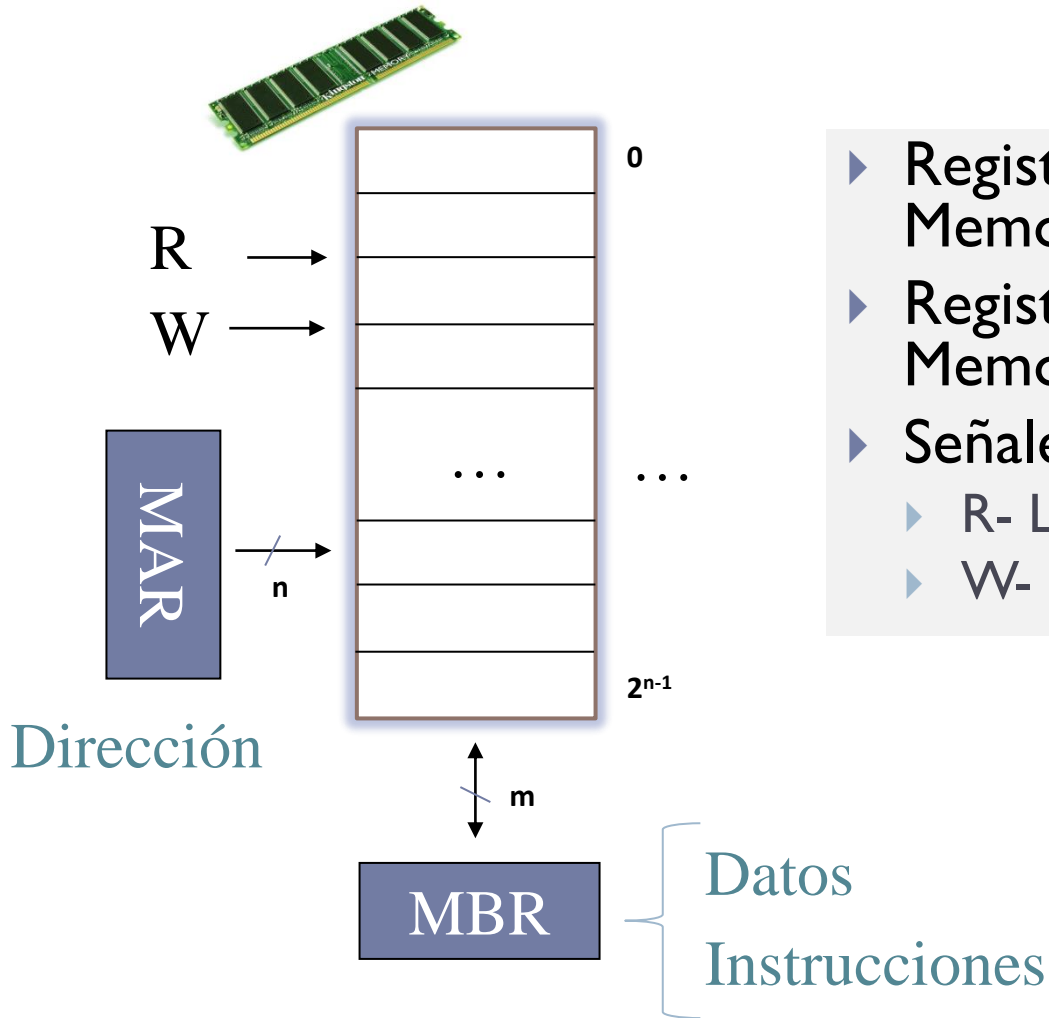
# Ejemplo de memoria principal

Memoria principal



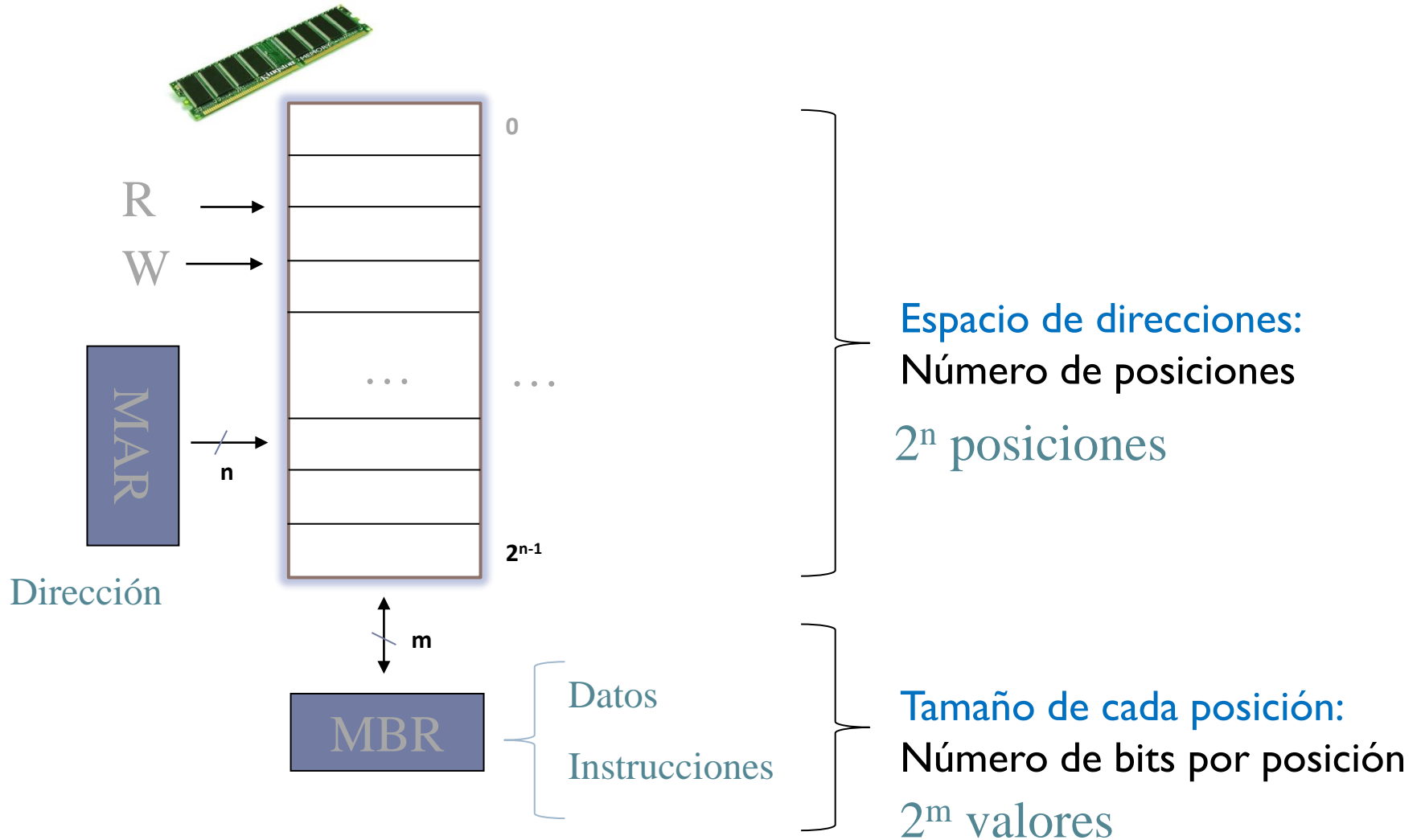
<http://www.videojug.com/film/what-components-are-inside-my-computer>

# Elementos de la memoria principal

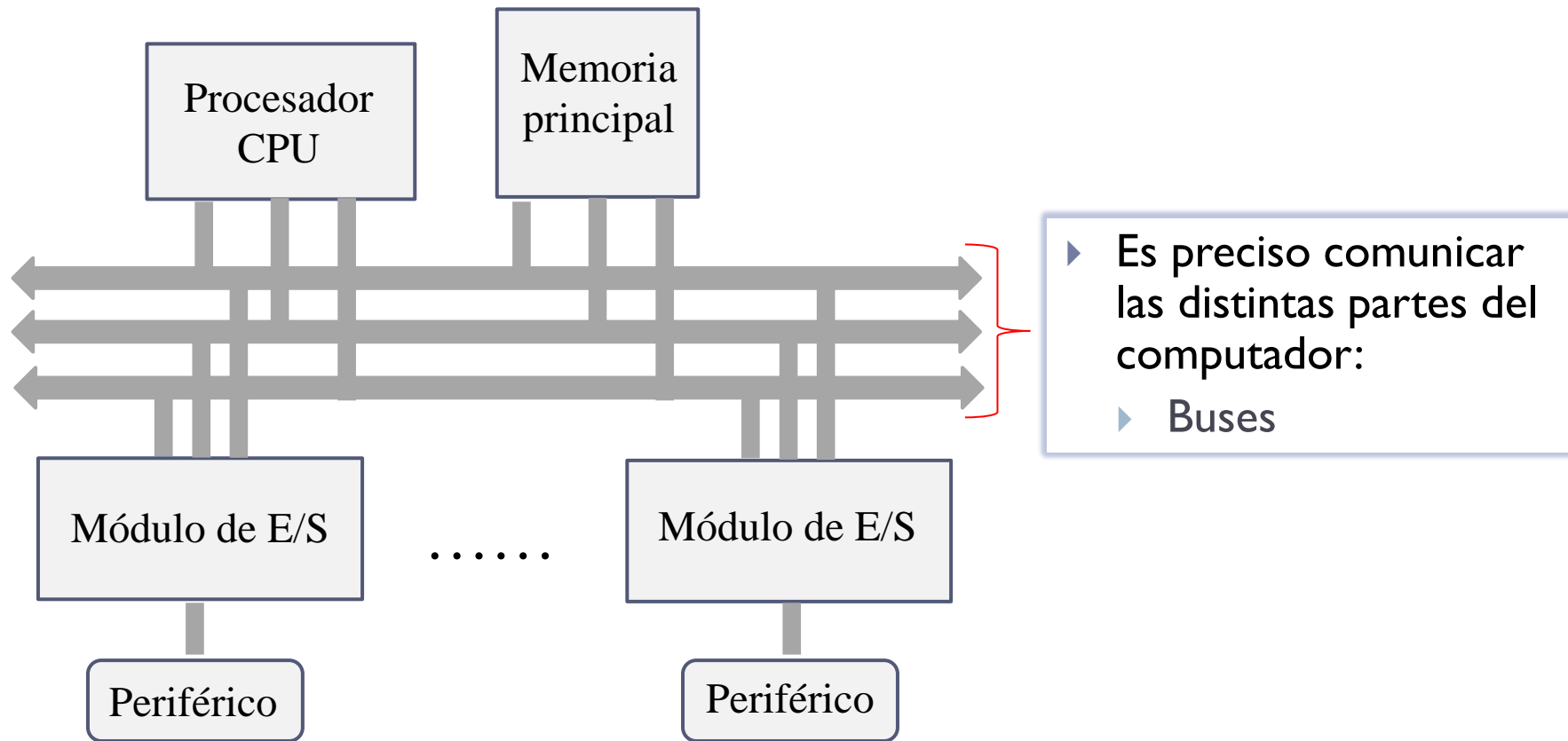


- ▶ Registro de direcciones (MAR, Memory Address Register)
- ▶ Registro de datos (MBR, Memory Buffer Register)
- ▶ Señales de control
  - ▶ R- Lectura (Read)
  - ▶ W- Escritura (Write)

# Elementos de la memoria principal



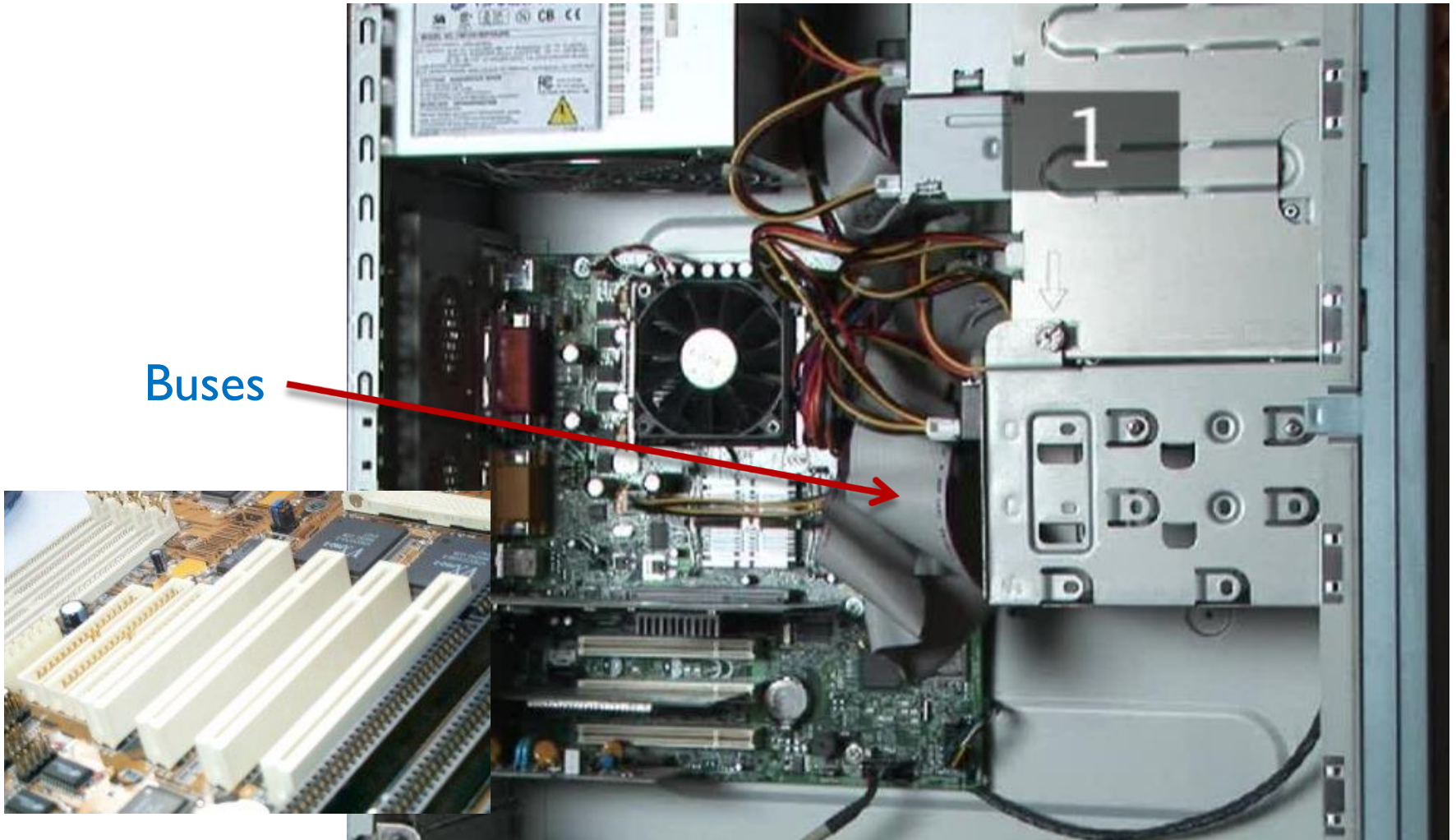
# Arquitectura Von Neumann (3/4)





# Ejemplo de buses

Buses



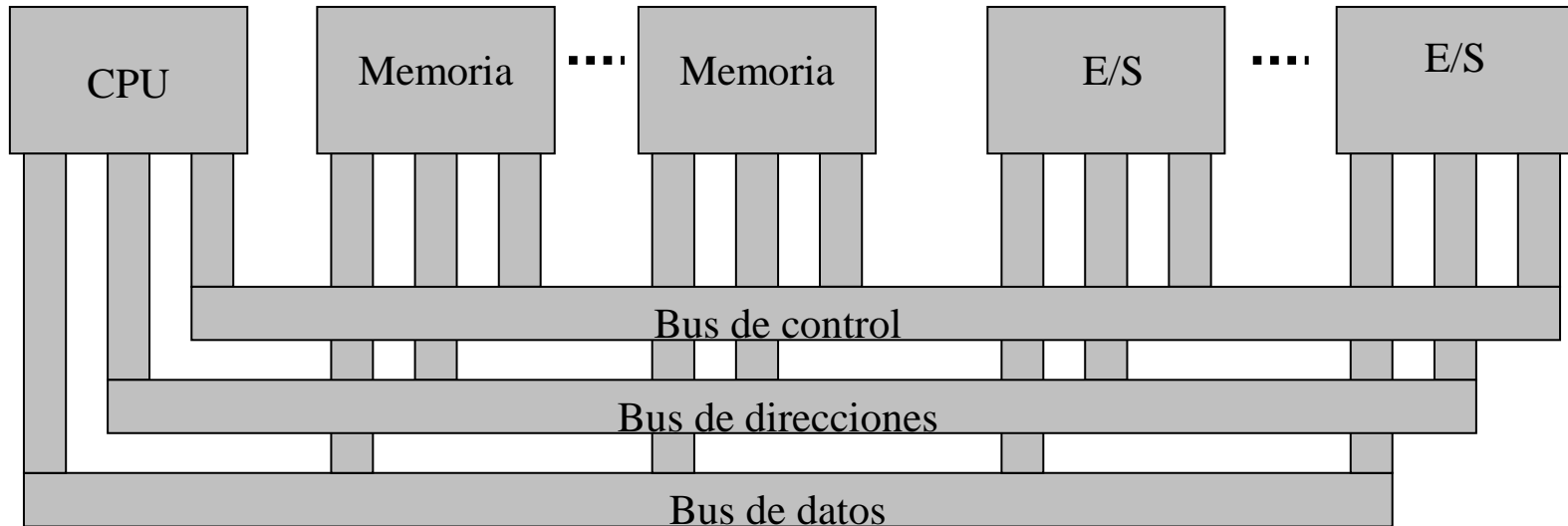
<http://www.videojug.com/film/what-components-are-inside-my-computer>

# 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**.



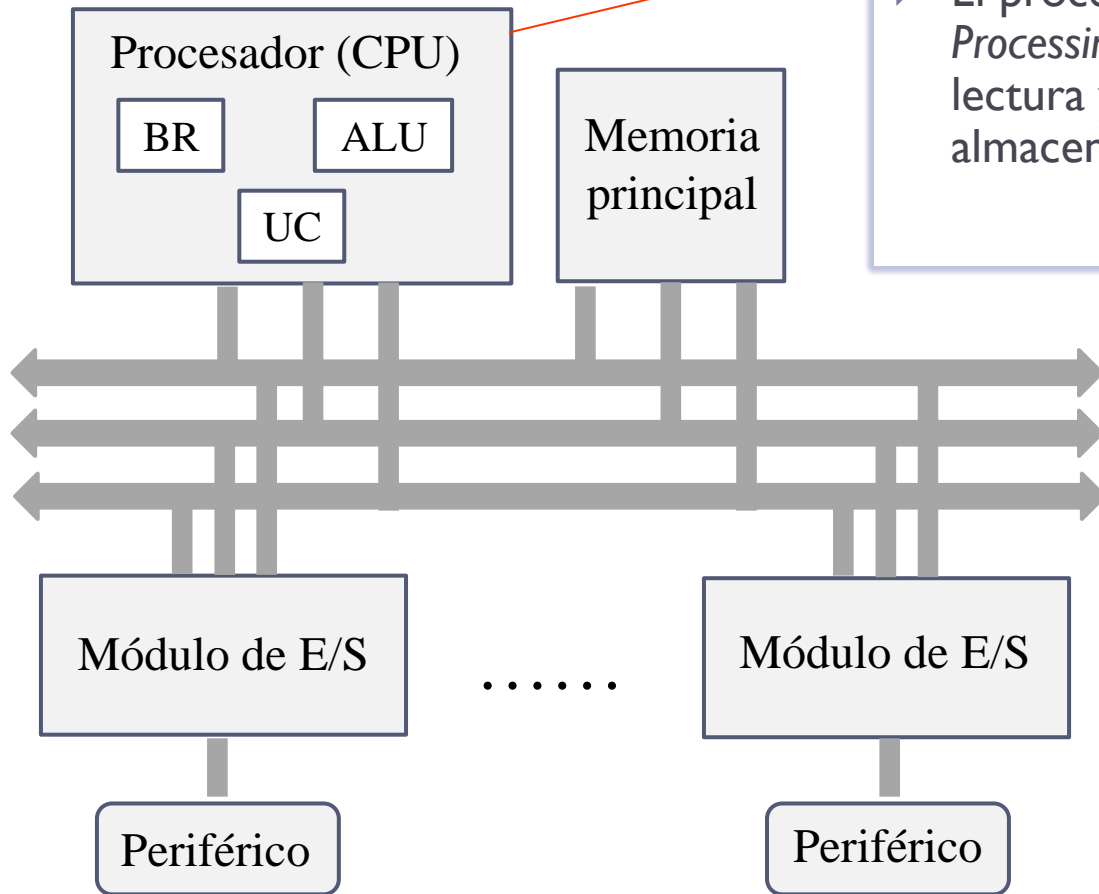
# Esquema de interconexión de bus



- ▶ **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)

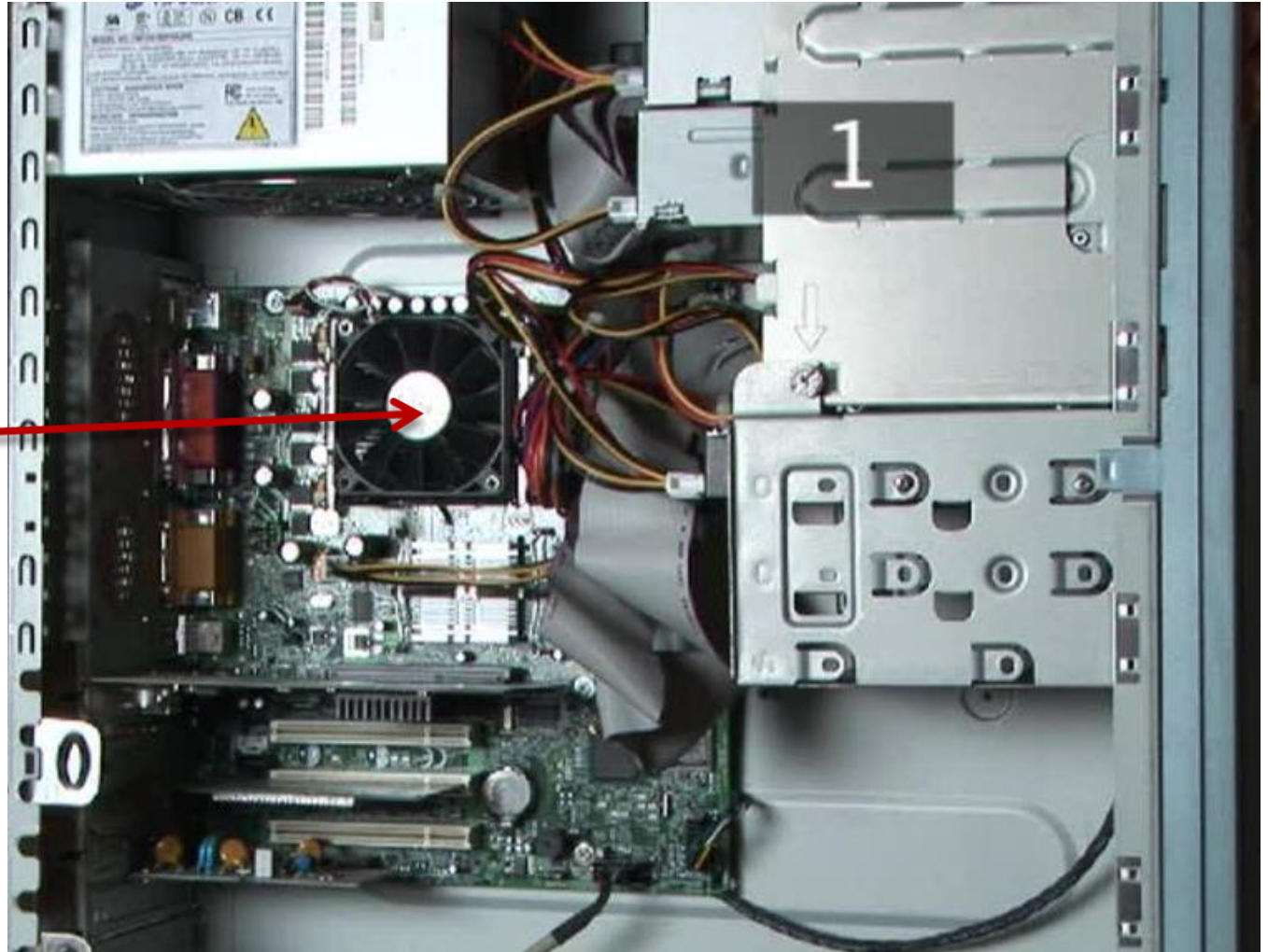
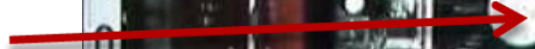
- El procesador o la CPU (*Central Processing Unit*) es responsable de la lectura y ejecución de las instrucciones almacenadas en memoria principal.



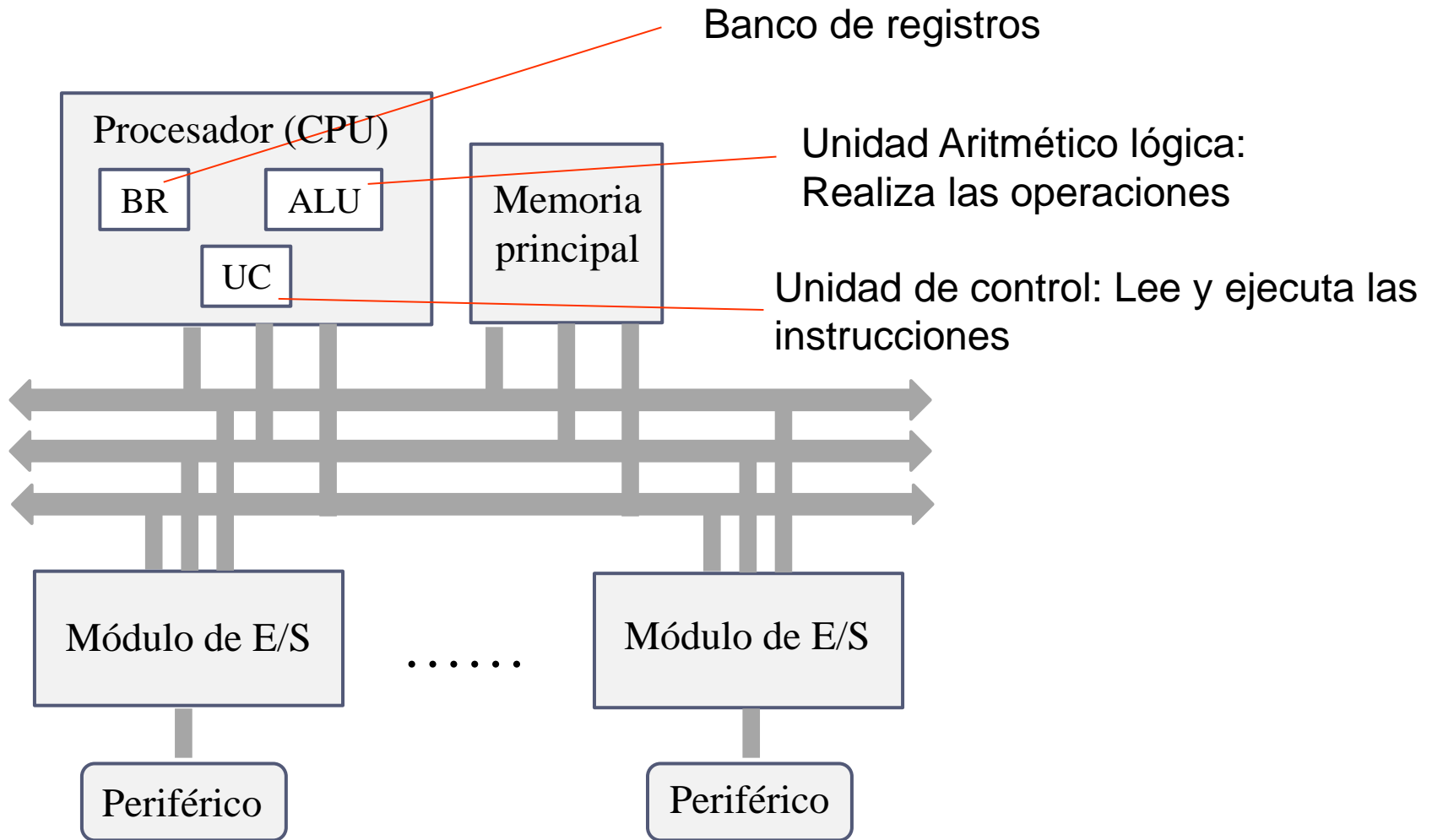
# Ejemplo de CPU



CPU

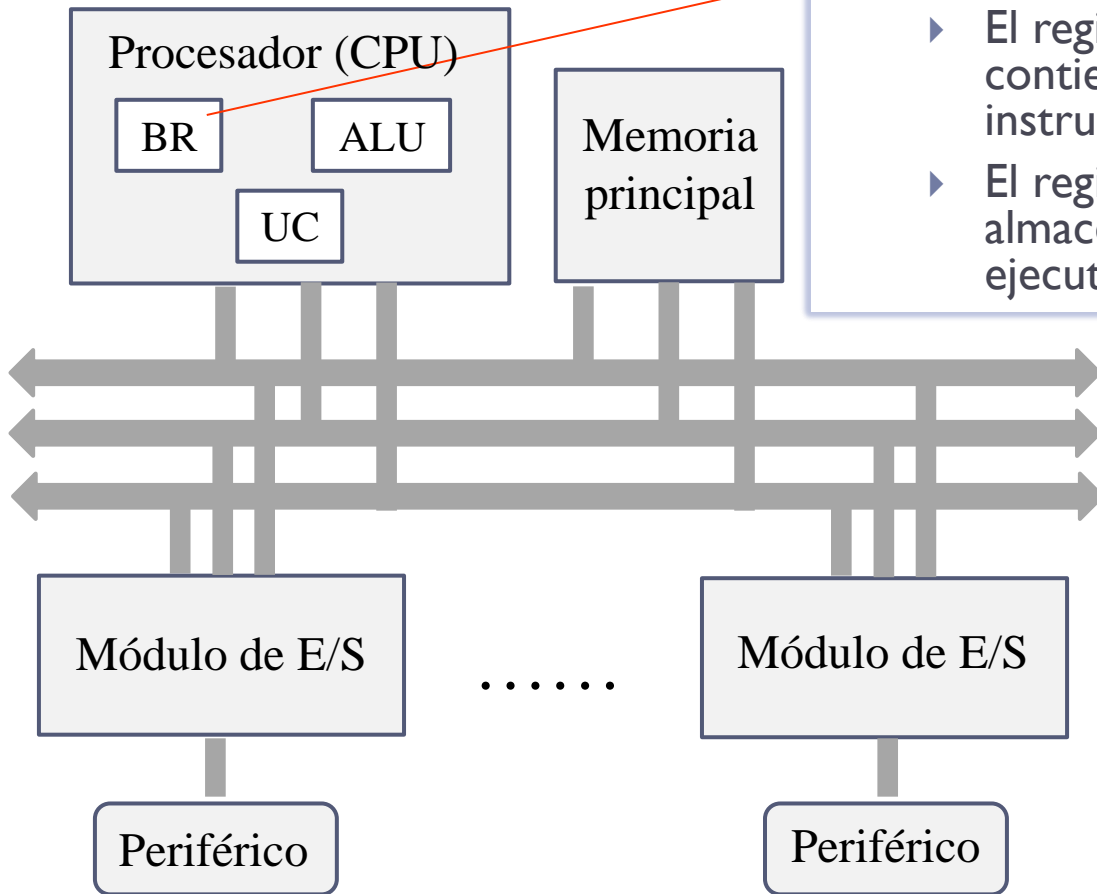


# Arquitectura Von Neumann (4/4)

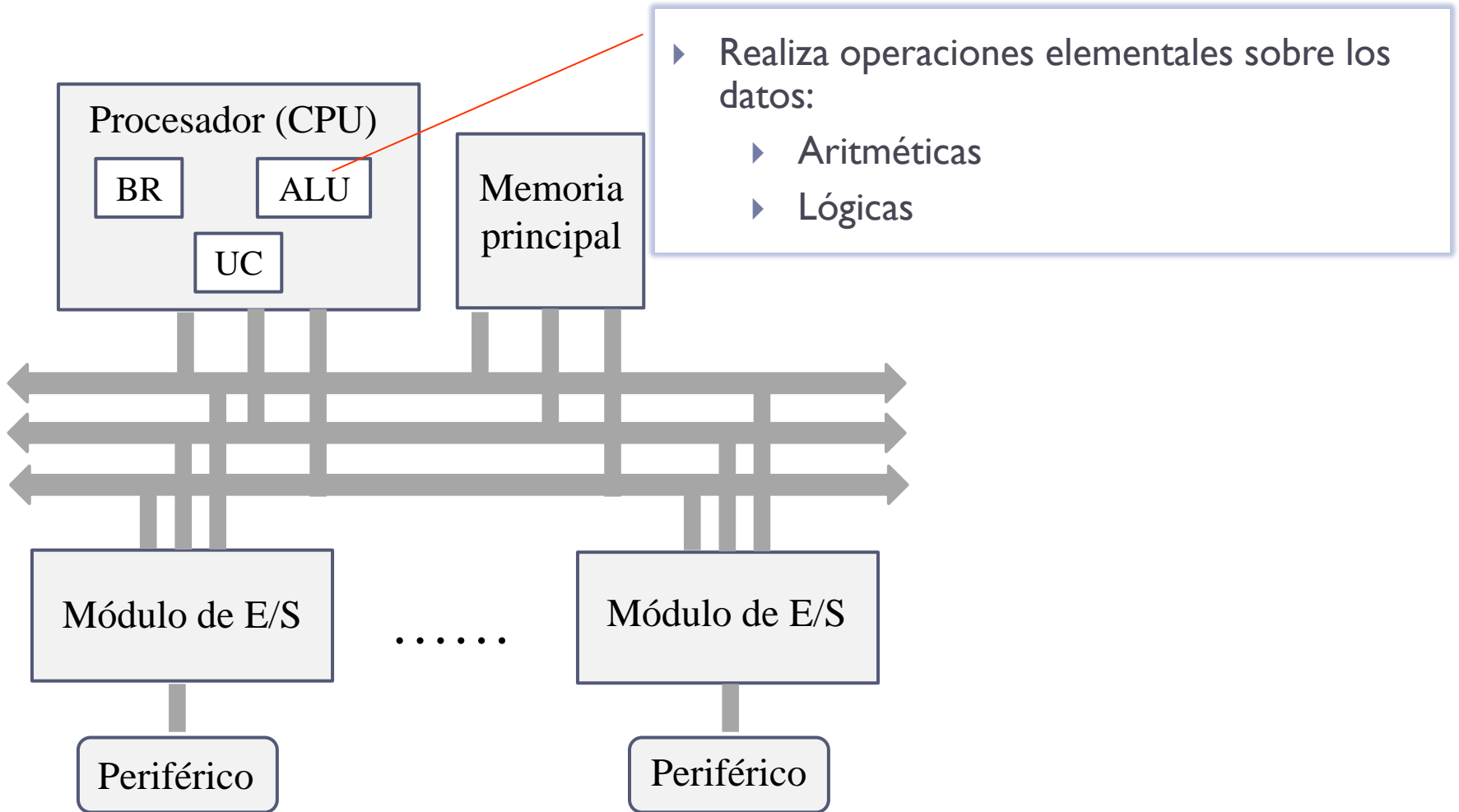


# Procesador: registros

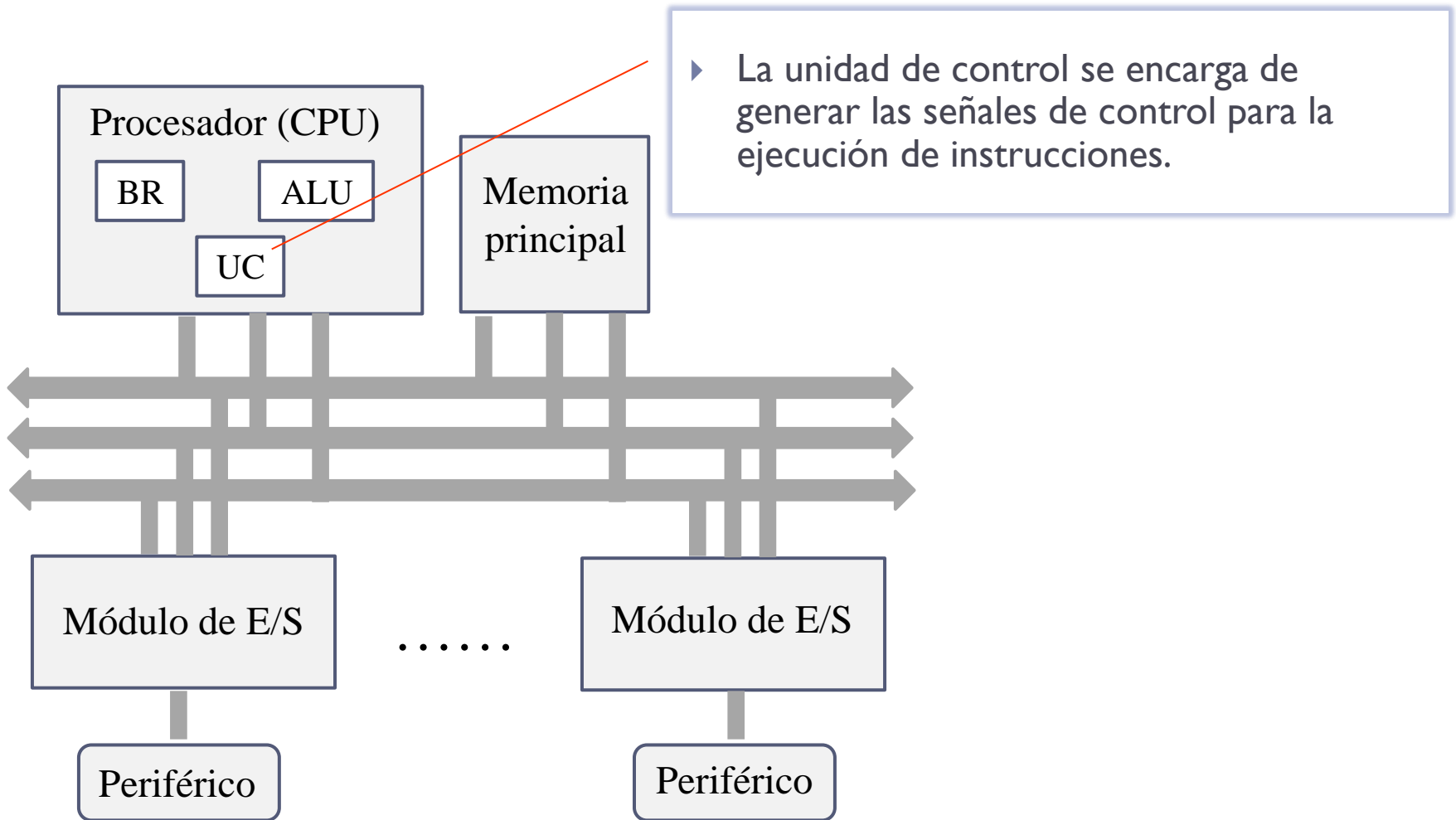
- ▶ Registros: almacenan una secuencia de bits.
- ▶ Dos registros especiales:
  - ▶ El registro PC (contador de programa) contiene la dirección de la siguiente instrucción a ejecutar.
  - ▶ El registro IR (registro de instrucción) almacena la instrucción que se está ejecutando



# Procesador: Unidad aritmético lógica ALU



# Procesador: Unidad de control, UC



# Contenidos

- ▶ **Introducción:**
  - ▶ ¿Qué es un computador?
  - ▶ Elementos constructivos de un computador
  - ▶ Concepto de estructura y arquitectura
- ▶ **Computador Von Neumann:**
  - ▶ Modelo Von Newmann
  - ▶ **Instrucciones máquina y programación**
  - ▶ Fases de ejecución de una instrucción
- ▶ **Características de un computador y tipos:**
  - ▶ Parámetros característicos de un computador
  - ▶ Tipos de computadores
  - ▶ Evolución histórica



# Programa

## ► Secuencia consecutiva de instrucciones máquina

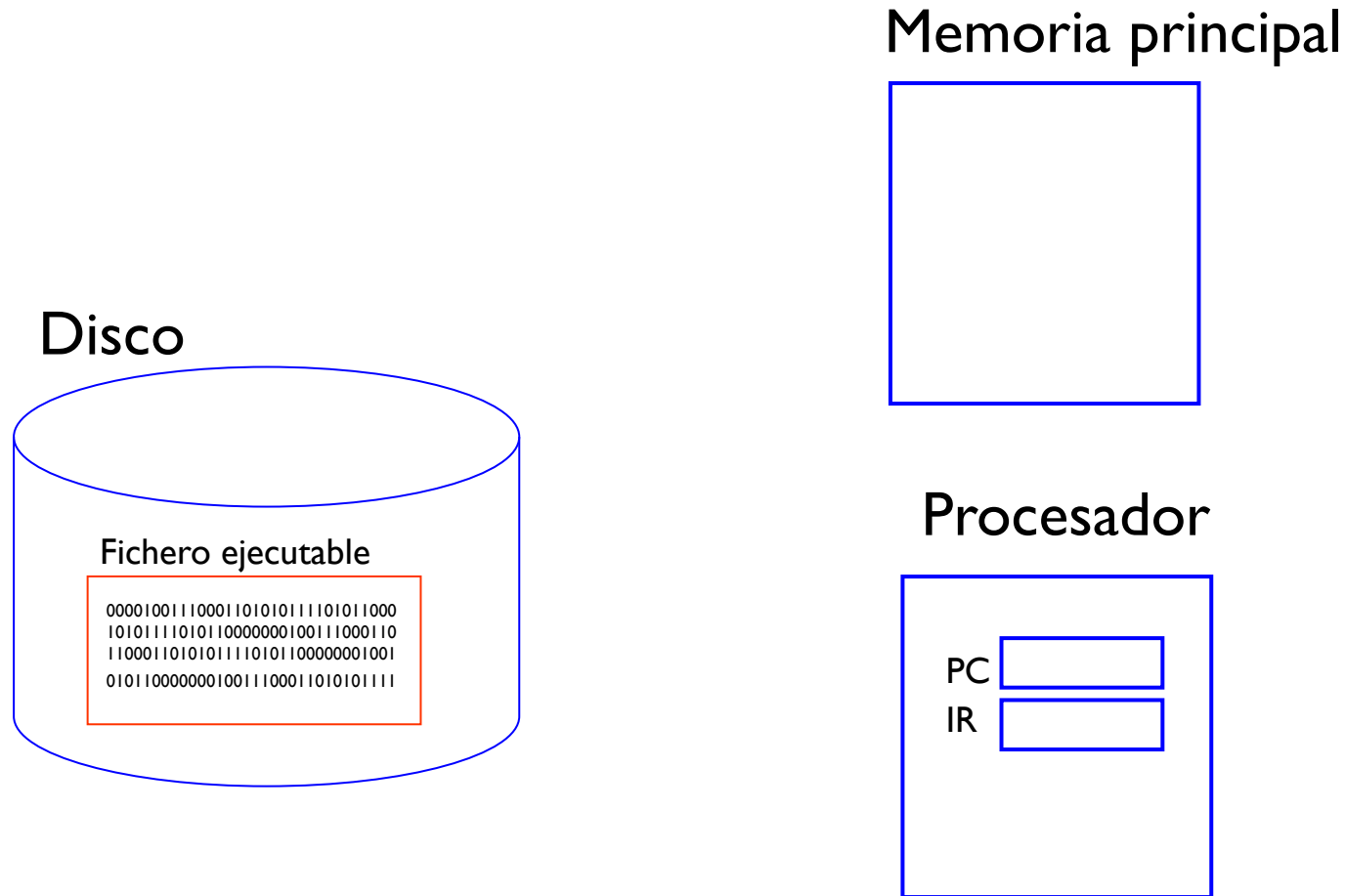
00001001110001101010111101011000	temp = v[k];
10101111010110000000100111000110	v[k] = v[k+1];
11000110101011110101100000001001	v[k+1] = temp;
01011000000010011100011010101111	
■	
■	
■	
■	
■	
■	
■	
■	

# Programa

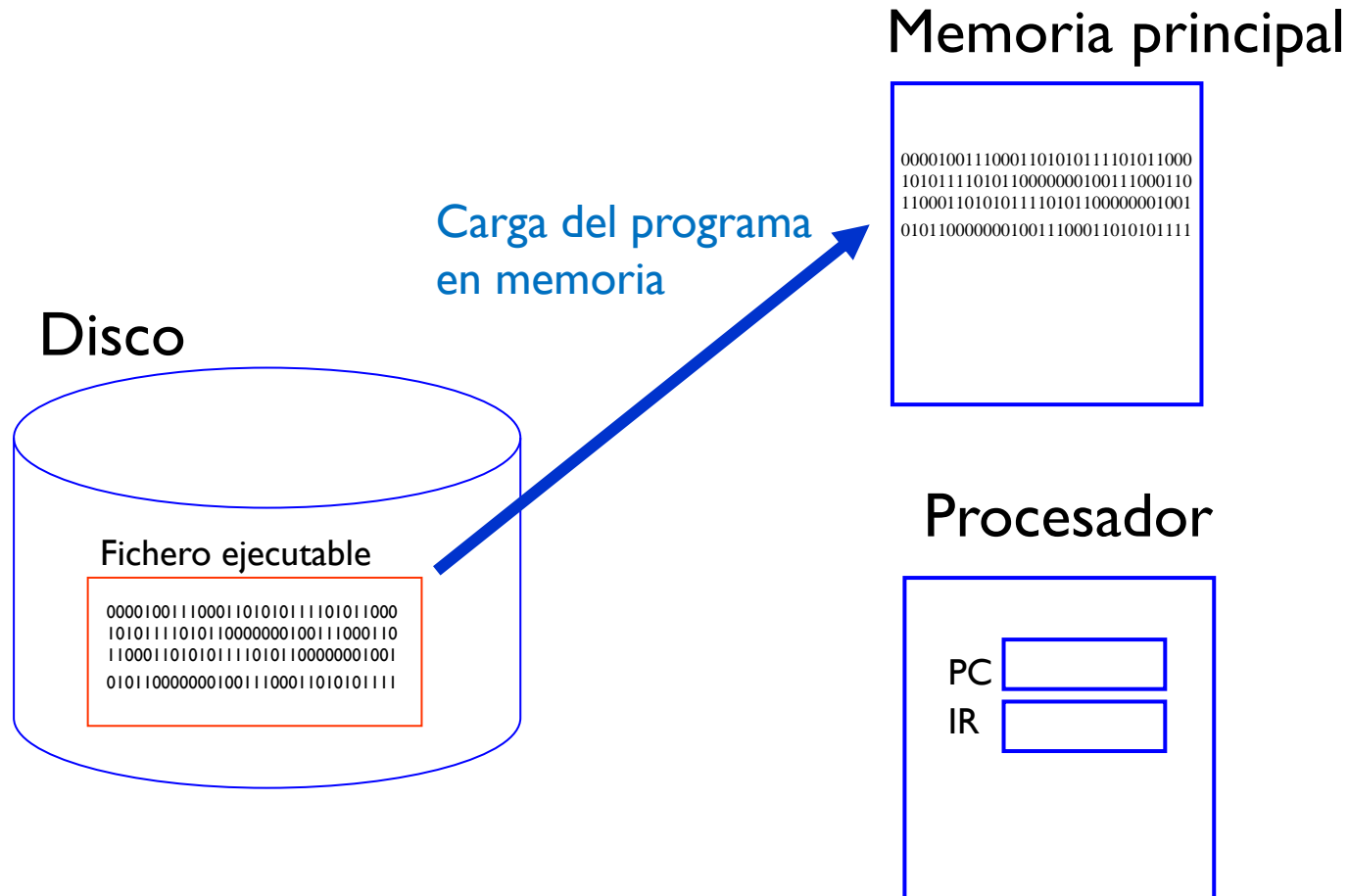
- ▶ Secuencia consecutiva de instrucciones máquina
- ▶ **Instrucción máquina**: operación elemental que puede ejecutar directamente un procesador
  - ▶ Codificación en binario

00001001110001101010111101011000	temp = v[k];
10101111010110000000100111000110	v[k] = v[k+1];
11000110101011110101100000001001	v[k+1] = temp;
01011000000010011100011010101111	
⋮	

# Cargar un programa



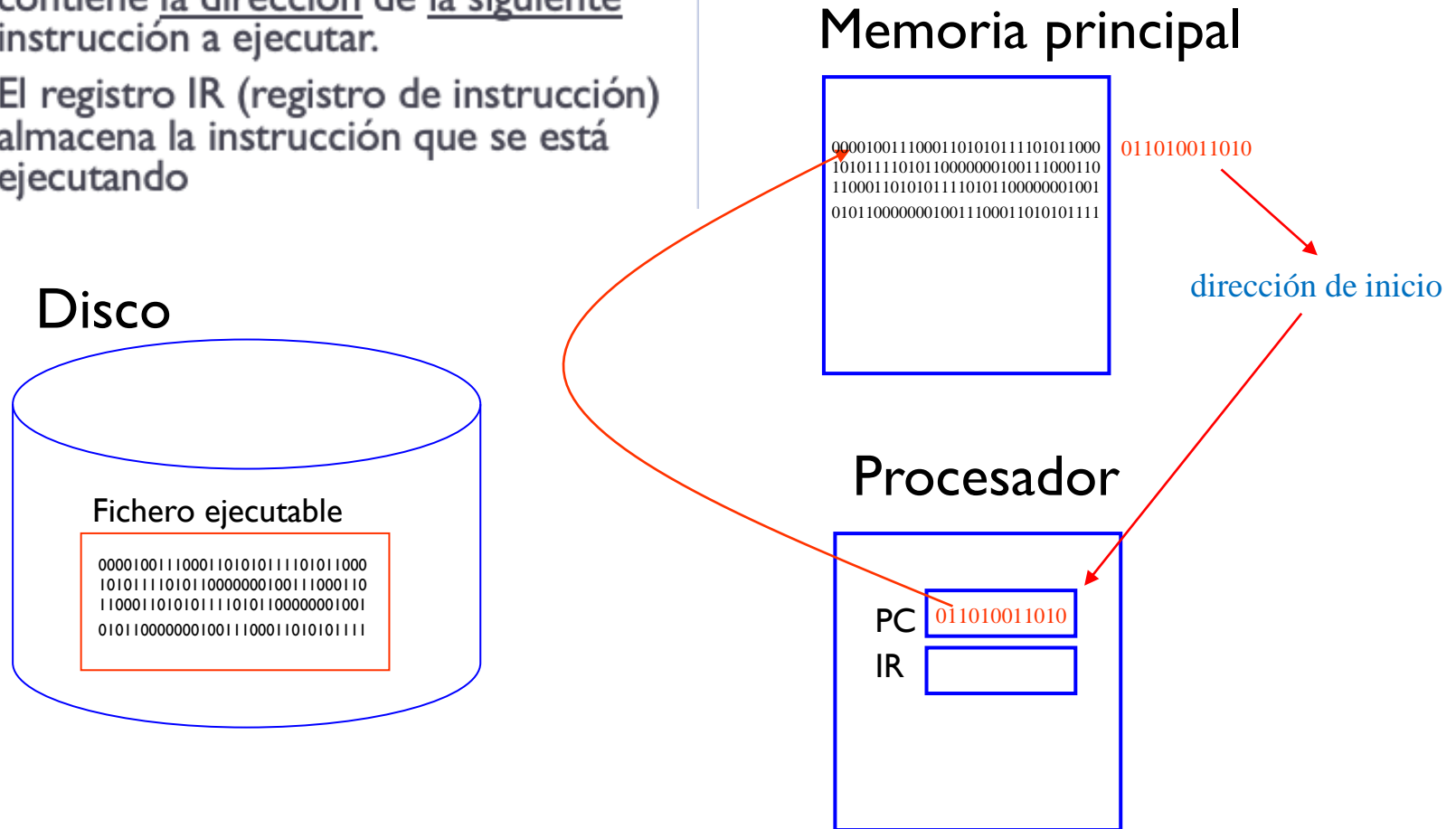
# Cargar un programa



# Cargar un programa

El registro PC (contador de programa) contiene la dirección de la siguiente instrucción a ejecutar.

El registro IR (registro de instrucción) almacena la instrucción que se está ejecutando

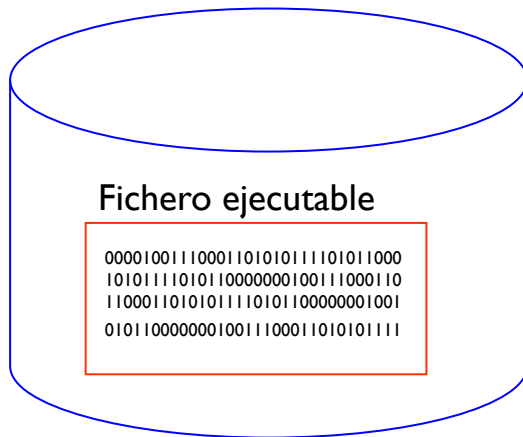


# Cargar un programa

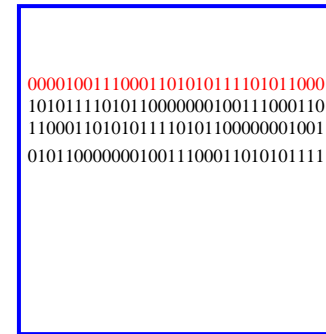
El registro PC (contador de programa) contiene la dirección de la siguiente instrucción a ejecutar.

El registro IR (registro de instrucción) almacena la instrucción que se está ejecutando

## Disco

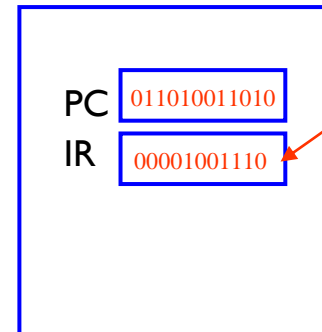


## Memoria principal



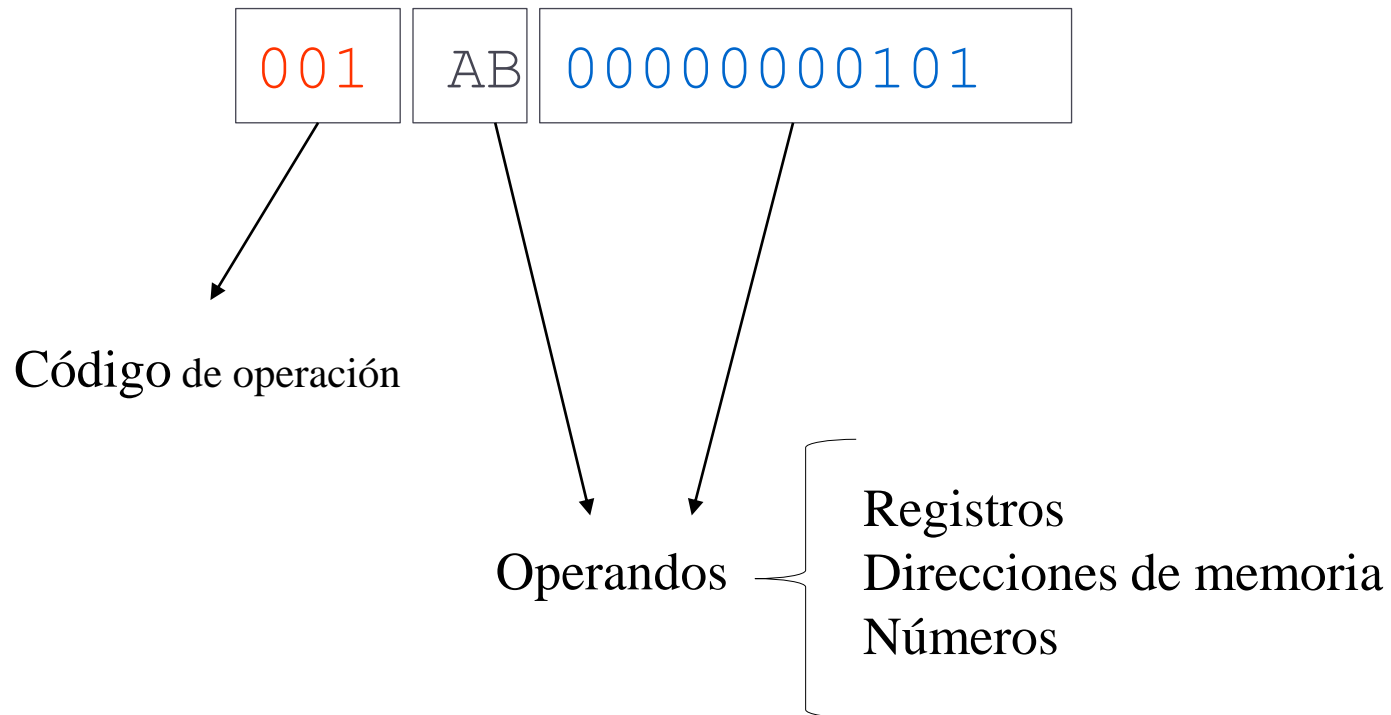
011010011010

## Procesador



contenido

# Formato de una instrucción máquina



# 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
```

## Memoria principal

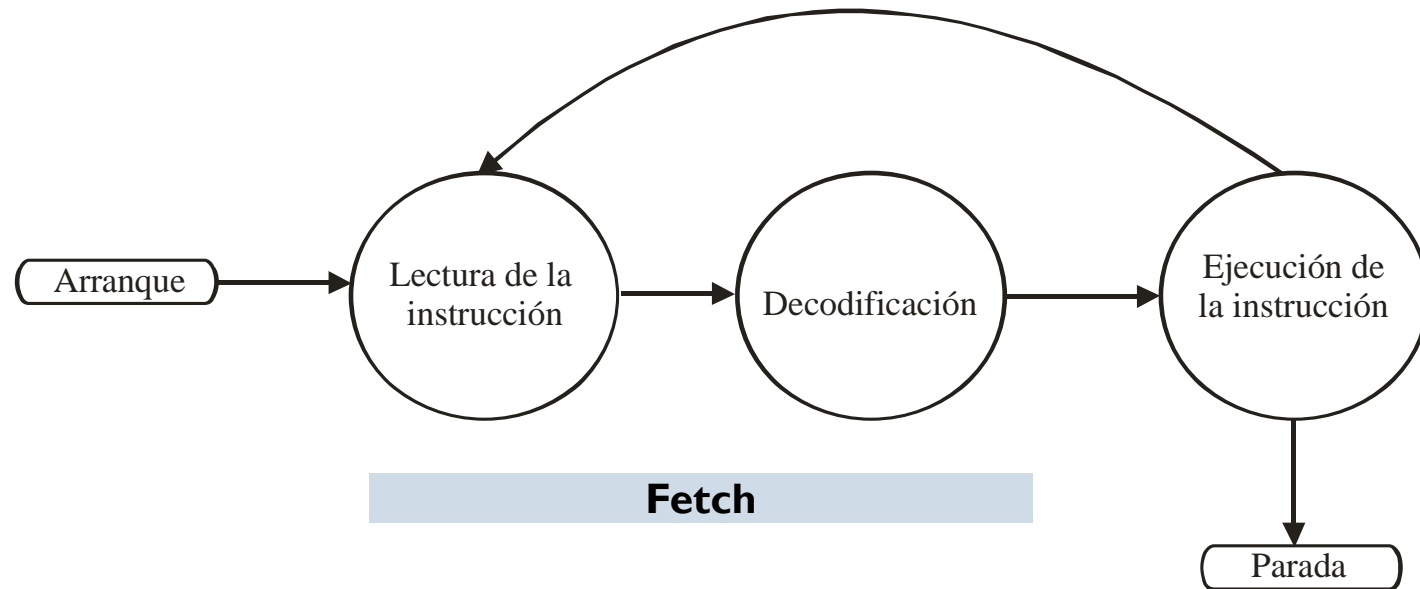
000100	0010000000000000
000101	0010100000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000



# Contenidos

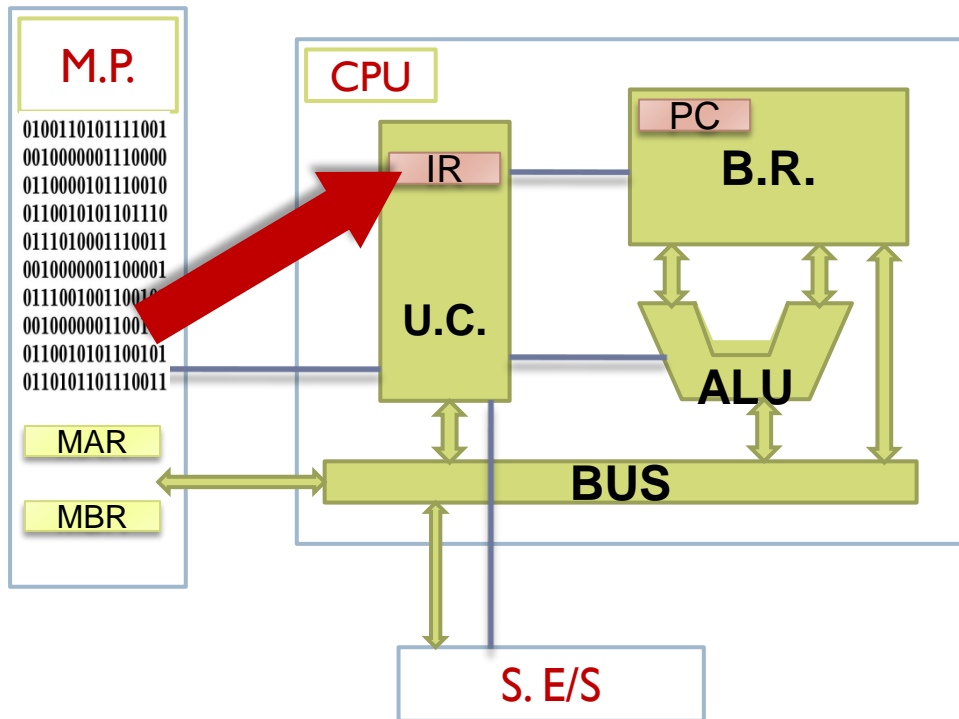
- ▶ **Introducción:**
  - ▶ ¿Qué es un computador?
  - ▶ Elementos constructivos de un computador
  - ▶ Concepto de estructura y arquitectura
- ▶ **Computador Von Neumann:**
  - ▶ Modelo Von Newmann
  - ▶ Instrucciones máquina y programación
  - ▶ **Fases de ejecución de una instrucción**
- ▶ **Características de un computador y tipos:**
  - ▶ Parámetros característicos de un computador
  - ▶ Tipos de computadores
  - ▶ Evolución histórica

# Fases de ejecución de una instrucción



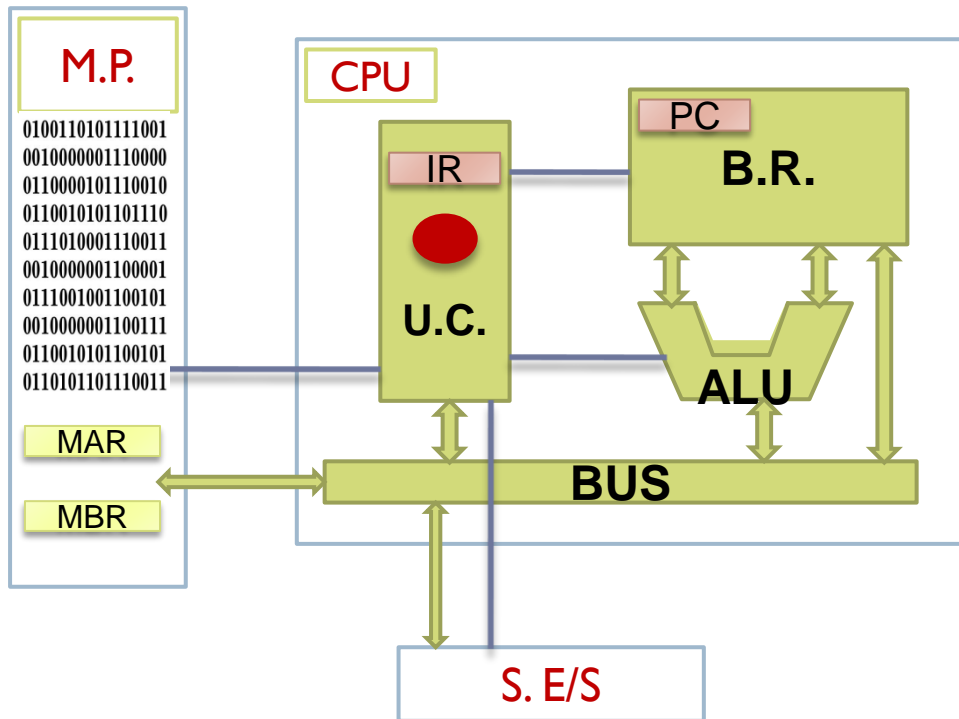
# Fases de ejecución:

## Lectura de la instrucción



- Leer de memoria principal la instrucción apuntada por el PC
  - El PC contiene la dirección de memoria donde se almacena la instrucción a ejecutar
  - La instrucción leída de M.P. se almacena en IR
- Incrementar PC
  - Incrementar la dirección almacenada en el PC para que apunte a la siguiente instrucción
- Decodificar instrucción
- Ejecutar la instrucción

# Fases de ejecución: Decodificación

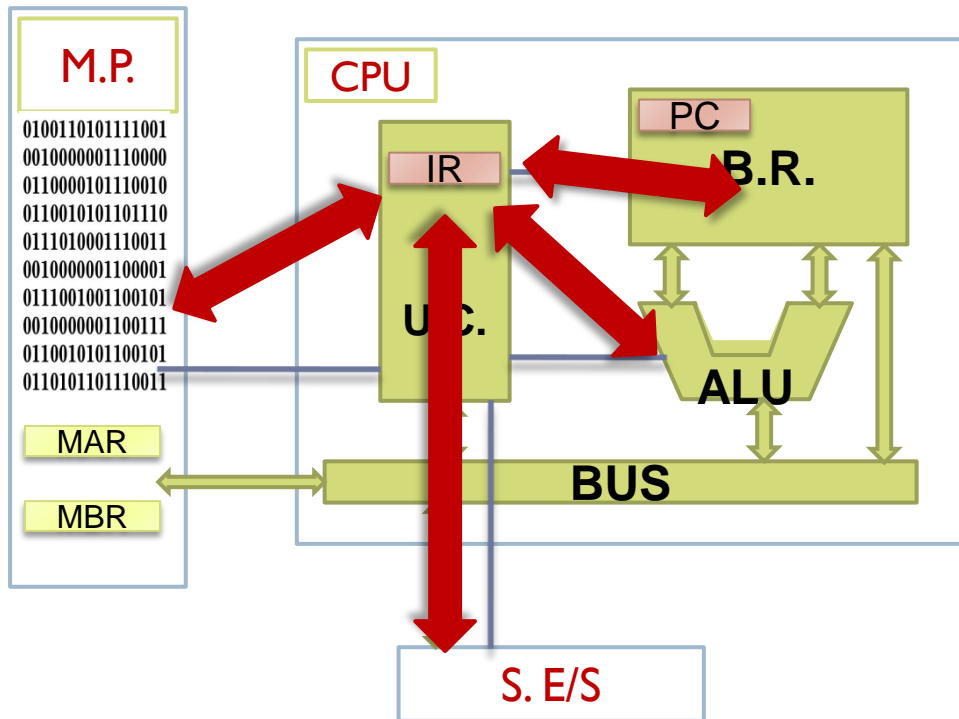


El registro PC (contador de programa) contiene la dirección de la siguiente instrucción a ejecutar.

El registro IR (registro de instrucción) almacena la instrucción que se está ejecutando

- Leer de memoria principal la instrucción apuntada por el PC
- El PC contiene la dirección de memoria donde se almacena la instrucción a ejecutar
- La instrucción leída de M.P. se almacena en IR
- Incrementar PC
- Incrementar la dirección almacenada en el PC para que apunte a la siguiente instrucción
- Decodificar instrucción
- Ejecutar la instrucción

# Fases de ejecución: Ejecución



- Leer de memoria principal la instrucción apuntada por el PC
- El PC contiene la dirección de memoria donde se almacena la instrucción a ejecutar
- La instrucción leída de M.P. se almacena en IR
- Incrementar PC
- Incrementar la dirección almacenada en el PC para que apunte a la siguiente instrucción
- Decodificar instrucción
- Ejecutar la instrucción

# Ejemplo de ejecución de un programa

## Procesador

PC	000100
RI	?
00	?
01	?
10	?
11	?

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

## Memoria principal

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

# Ejemplo de ejecución de un programa

## Procesador

PC	000100
RI	0010000000000000
00	?
01	?
10	?
11	?

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

## Memoria principal

Dirección	Contenido
000100	0010000000000000
000101	00101000000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

# Ejemplo de ejecución de un programa

## Procesador

PC	000101
RI	0010000000000000
00	?
01	?
10	?
11	?

- ▶ Lectura de la instrucción
- ▶ **Apuntar a la siguiente instrucción**
  - ▶  $PC \leftarrow PC + I$
- ▶ Decodificación de la instrucción
- ▶ Ejecución de la instrucción
- ▶ Volver a fetch

## Memoria principal

Dirección	Contenido
000100	0010000000000000
000101	00101000000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000



# Ejemplo de ejecución de un programa

## Procesador

PC	000101
RI	0010000000000000
00	?
01	?
10	?
11	?

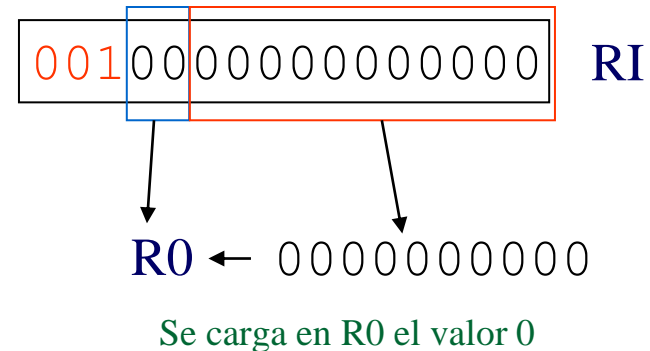
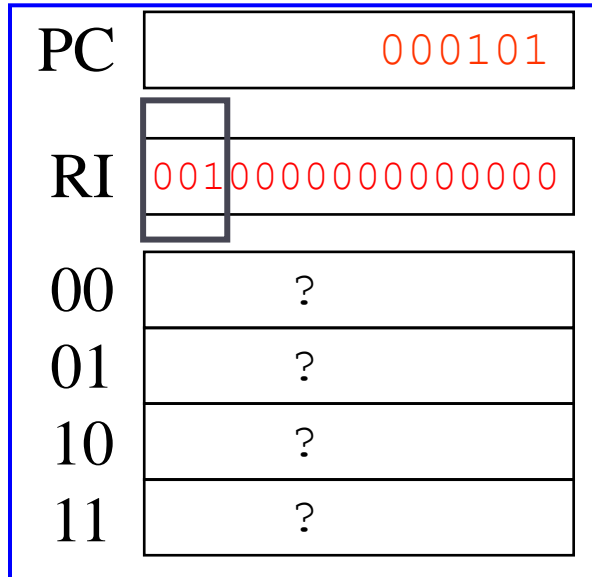
- ▶ Lectura de la instrucción
- ▶ Apuntar a la siguiente instrucción
- ▶ **Decodificación de la instrucción**
- ▶ Ejecución de la instrucción
- ▶ Volver a *fetch*

## Memoria principal

Dirección	Contenido
000100	0010000000000000
000101	00101000000000100
000110	0011000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	1000000000001000
001100	0111100000100000

# Ejemplo de ejecución de un programa

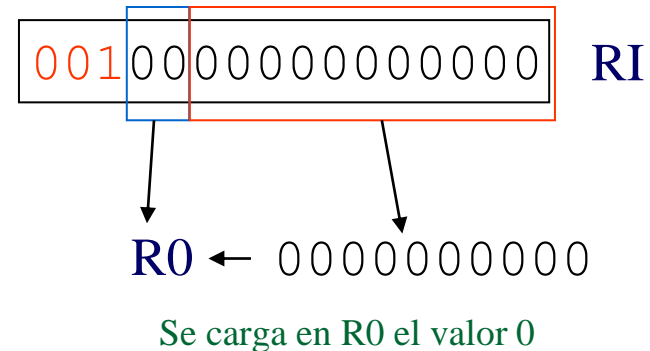
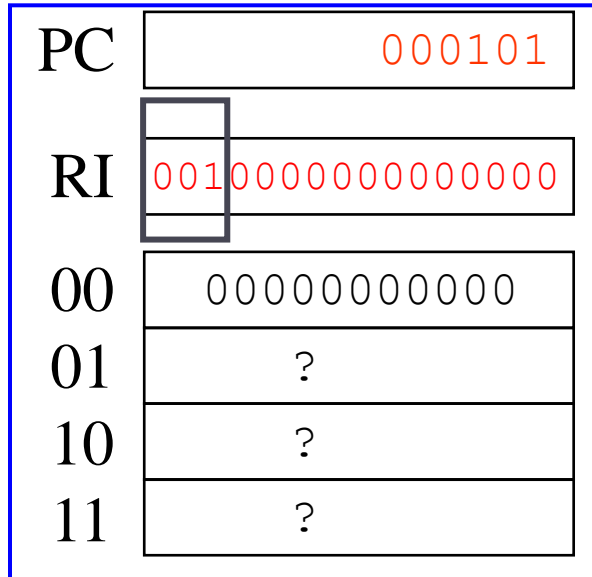
## Procesador



- ▶ 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

## Procesador



- ▶ 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

## Procesador

PC	000101
RI	0010000000000000
00	000000000000
01	?
10	?
11	?

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

## Memoria principal

Dirección	Contenido
000100	0010000000000000
000101	00101000000000100
000110	00110000000000001
000111	00111000000000000
001000	10100010000001100
001001	0001111100000000
001010	0000000100000000
001011	10000000000001000
001100	01111000000100000

# Ejemplo de ejecución de un programa

## Procesador

PC	000101
RI	0010100000000100
00	000000000000
01	?
10	?
11	?

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

## Memoria principal

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

# Ejemplo de ejecución de un programa

## Procesador

PC	000110
RI	0010100000000100
00	000000000000
01	?
10	?
11	?

- ▶ Lectura de la instrucción
- ▶ **Apuntar a la siguiente instrucción**
  - ▶  $PC \leftarrow PC + I$
- ▶ Decodificación de la instrucción
- ▶ Ejecución de la instrucción
- ▶ Volver a *fetch*

## Memoria principal

Dirección	Contenido
000100	0010000000000000
000101	00101000000000100
<b>000110</b>	00110000000000001
000111	0011100000000000
001000	1010001000001100
001001	0001111100000000
001010	0000000100000000
001011	10000000000001000
001100	0111100000100000

# Ejemplo de ejecución de un programa

## Procesador

PC	000110
RI	0010100000000100
00	000000000000
01	?
10	?
11	?

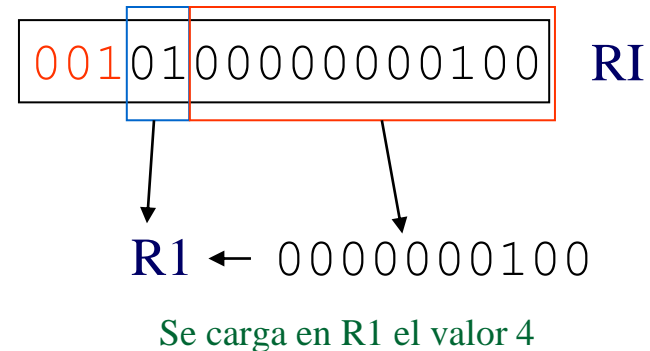
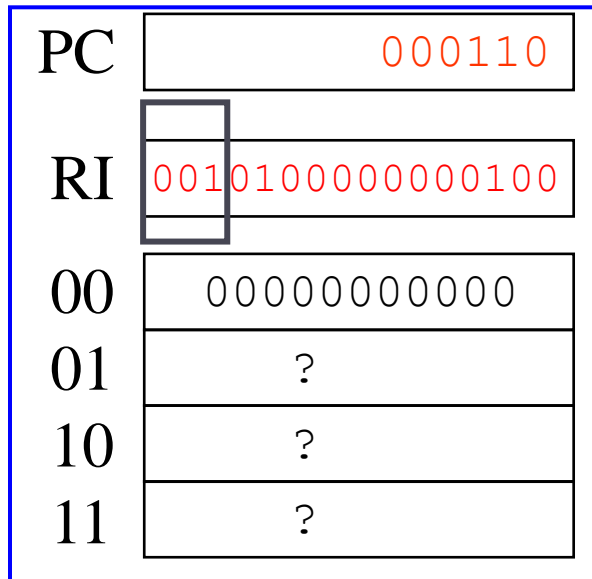
- ▶ Lectura de la instrucción
- ▶ Apuntar a la siguiente instrucción
- ▶ **Decodificación de la instrucción**
- ▶ Ejecución de la instrucción
- ▶ Volver a *fetch*

## Memoria principal

Dirección	Contenido
000100	0010000000000000
000101	00101000000000100
<b>000110</b>	00110000000000001
000111	00111000000000000
001000	10100010000001100
001001	00011111000000000
001010	00000001000000000
001011	10000000000001000
001100	01111000000100000

# Ejemplo de ejecución de un programa

## Procesador

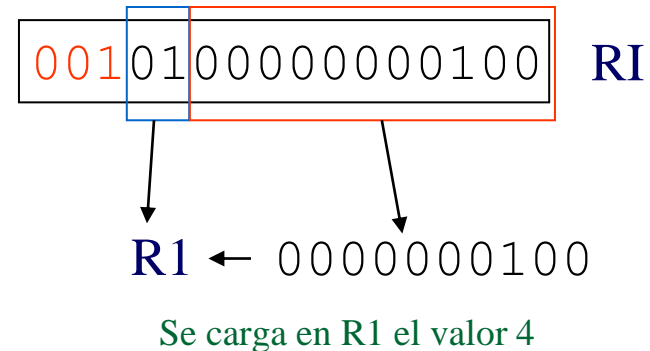
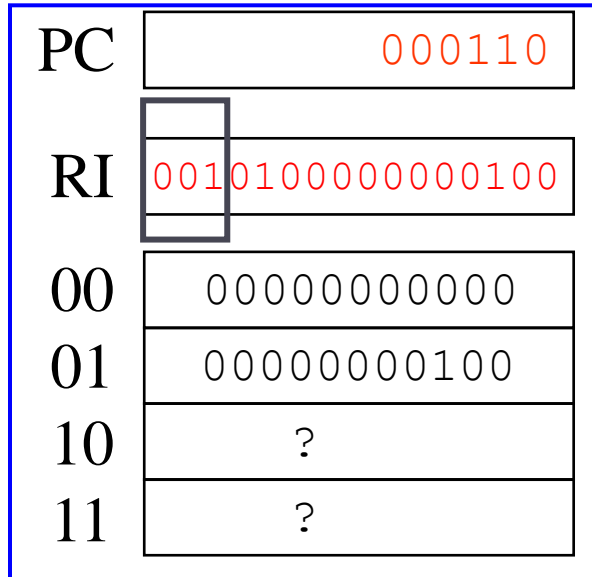


- ▶ Lectura de la instrucción
- ▶ Apuntar a la siguiente instrucción
- ▶ **Decodificación de la instrucción**
- ▶ Ejecución de la instrucción
- ▶ Volver a *fetch*



# Ejemplo de ejecución de un programa

## Procesador



- ▶ 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

## Procesador

PC	000110
RI	0010100000000100
00	000000000000
01	00000000100
10	?
11	?

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

## Memoria principal

Dirección	Contenido
000100	0010000000000000
000101	00101000000000100
000110	00110000000000001
000111	00111000000000000
001000	10100010000001100
001001	0001111100000000
001010	0000000100000000
001011	10000000000001000
001100	01111000000100000

# Ejemplo de ejecución de un programa

## Procesador

PC	000110
RI	0010100000000100
00	000000000000
01	00000000100
10	?
11	?

► Continúa la ejecución

## Memoria principal

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

# Contenidos

- ▶ **Introducción:**
  - ▶ ¿Qué es un computador?
  - ▶ Elementos constructivos de un computador
  - ▶ Concepto de estructura y arquitectura
- ▶ **Computador Von Neumann:**
  - ▶ Modelo Von Newmann
  - ▶ Instrucciones máquina y programación
  - ▶ Fases de ejecución de una instrucción
- ▶ **Características de un computador y tipos:**
  - ▶ **Parámetros característicos de un computador**
  - ▶ Tipos de computadores
  - ▶ 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
- ▶ Un computador con un ancho de palabra de  $n$  bits:
  - ▶ Direcciones de memoria de  $n$  bits
  - ▶ Los registros almacenan  $n$  bits
  - ▶ Números enteros de  $n$  bits
- ▶ Tamaños típicos → 32 bits, 64 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?

# 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



# Unidades para tamaño

- Normalmente se expresa en octetos o bytes:

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

Nombre	Prefijo binario	Prefijo SI
<b>Kilo</b>	kibibyte (KiB)	kilobyte (kB)
<b>Mega</b>	mebibyte (MiB)	megabyte (MB)
<b>Giga</b>	gibibyte (GiB)	gigabyte (GB)
<b>Tera</b>	tebibyte (TiB)	terabyte (TB)
<b>Peta</b>	pebibyte (PiB)	perabyte (PB)
<b>Exa</b>	exbibyte (EiB)	exabyte (EB)
<b>Zetta</b>	zebibyte (ZiB)	zettabyte (ZB)
<b>Yotta</b>	yobibyte (YiB)	yottabyte (YB)

# Unidades para tamaño

- ▶ 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
  - ▶ ...
- ▶ En **comunicación** se utilizan potencias de 10 y bits:
  - ▶ 1 Kb = 1000 bits = 125 bytes
  - ▶ 1 KB = 1000 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?
  - ▶  $200 \text{ GB} = 200 * 10^9 \text{ bytes} = 186.26 \text{ Gigabytes}$
- ▶ ¿Cuántos bytes por segundo transmite mi ADSL de 20 Mb?
  - ▶  $B \rightarrow \text{Byte}$
  - ▶  $b \rightarrow \text{bit.}$
  - ▶  $20 \text{ Mb} = 20 * 10^6 \text{ bits} = 20 * 10^6 / 8 \text{ bytes} = 2.38 \text{ Megabytes por segundo}$

# 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.
- ▶  $\text{MFLOPS} < \text{MIPS}$  (operación flotante más compleja que operación normal).
- ▶ Computadores vectoriales:  $\text{MFLOPS} > \text{MIPS}$
- ▶ Ejemplo: Itanium 2  $\rightarrow$  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

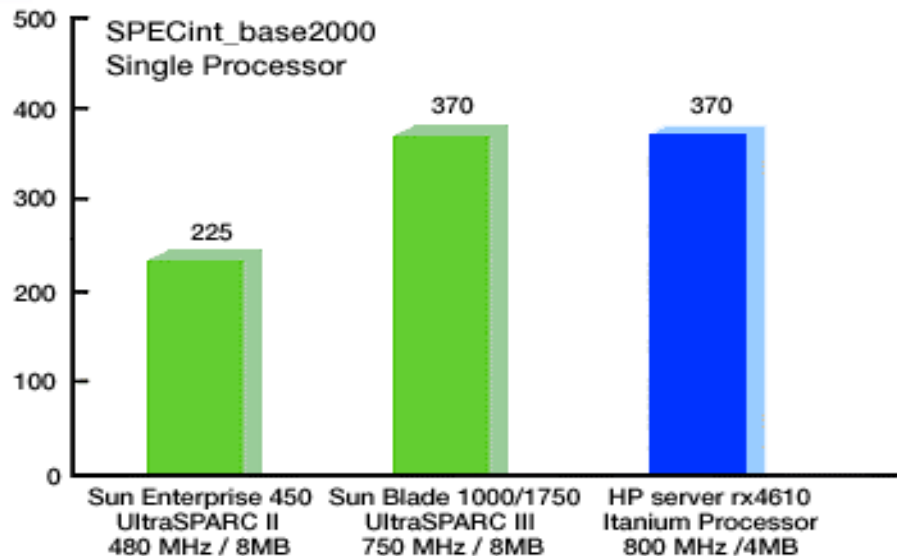
- ▶ MIPS y MFLOPS no válidos para comparar distintas máquinas.
- ▶ Tests basados en ejecutar un mismo programa en distintas máquinas para compararlas.
- ▶ Miden efectividad Compilador + CPU
- ▶ Los test sintéticos estandarizados (“oficiales”) buscan comparar la potencia de dos computadores.
- ▶ Es posible usar test sintéticos “no oficiales” para hacerse a la idea de la mejora con la carga de trabajo diaria

# Tests sintéticos “oficiales”

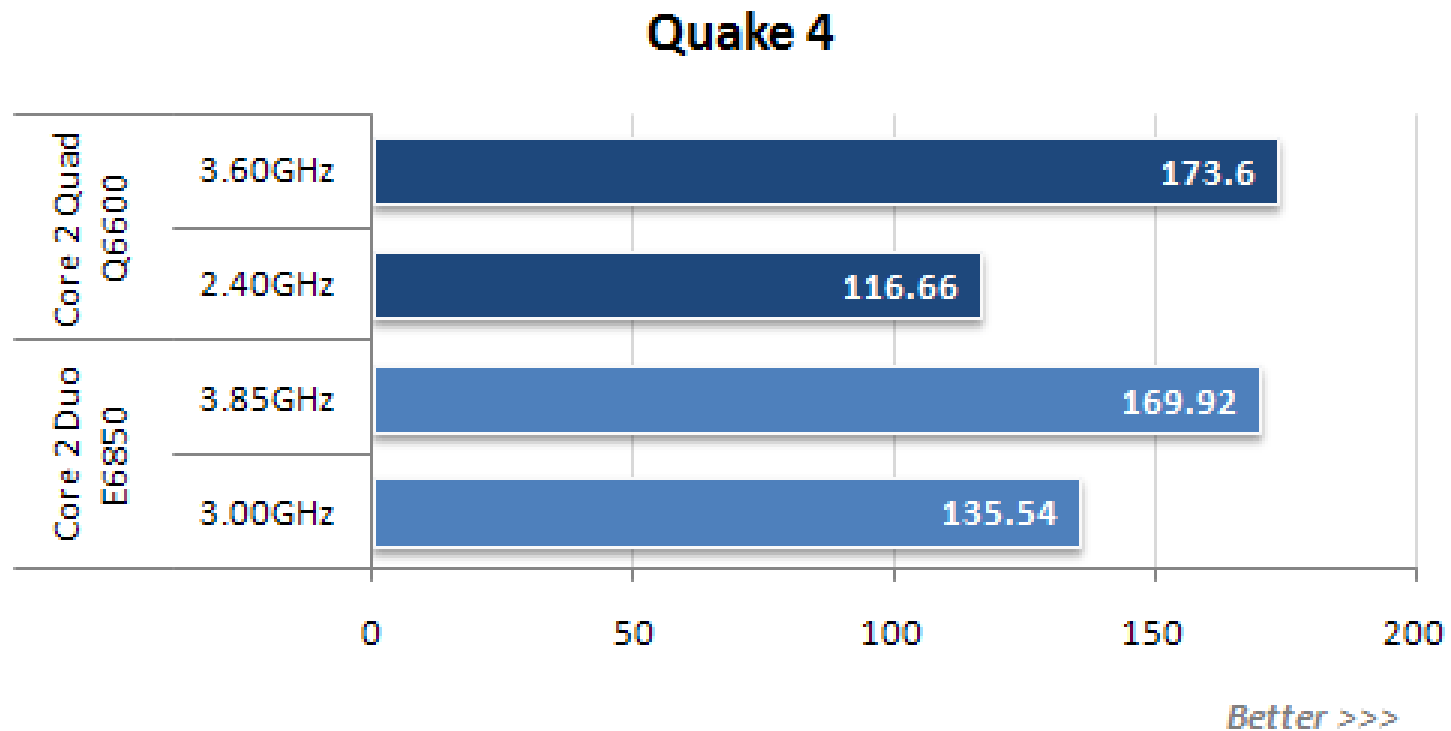
- ▶ Tests más usados:

- ▶ Linpack.
- ▶ SPEC.

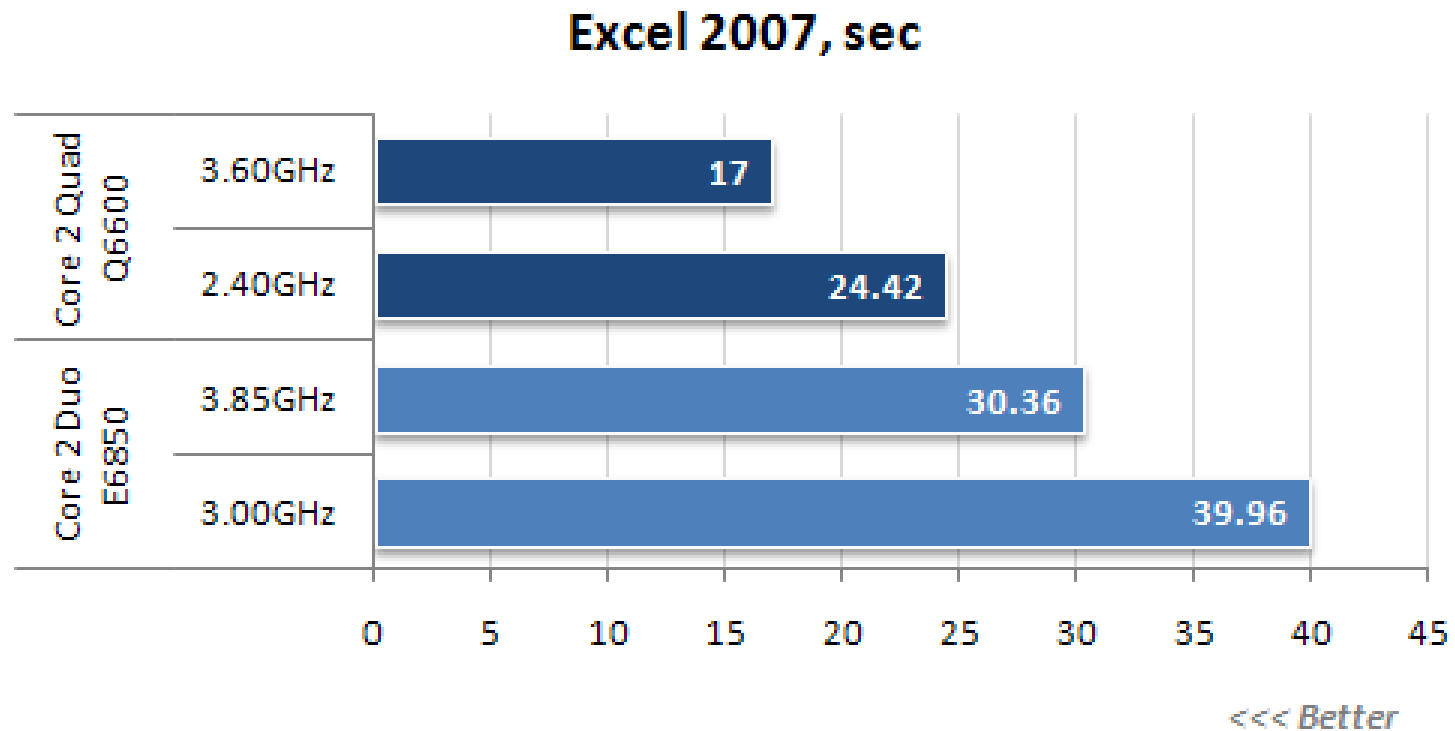
**SPEC CPU2000 Performance – SPECint2000**  
*Itanium™ Processor delivers best of class floating point performance and competitive integer performance*



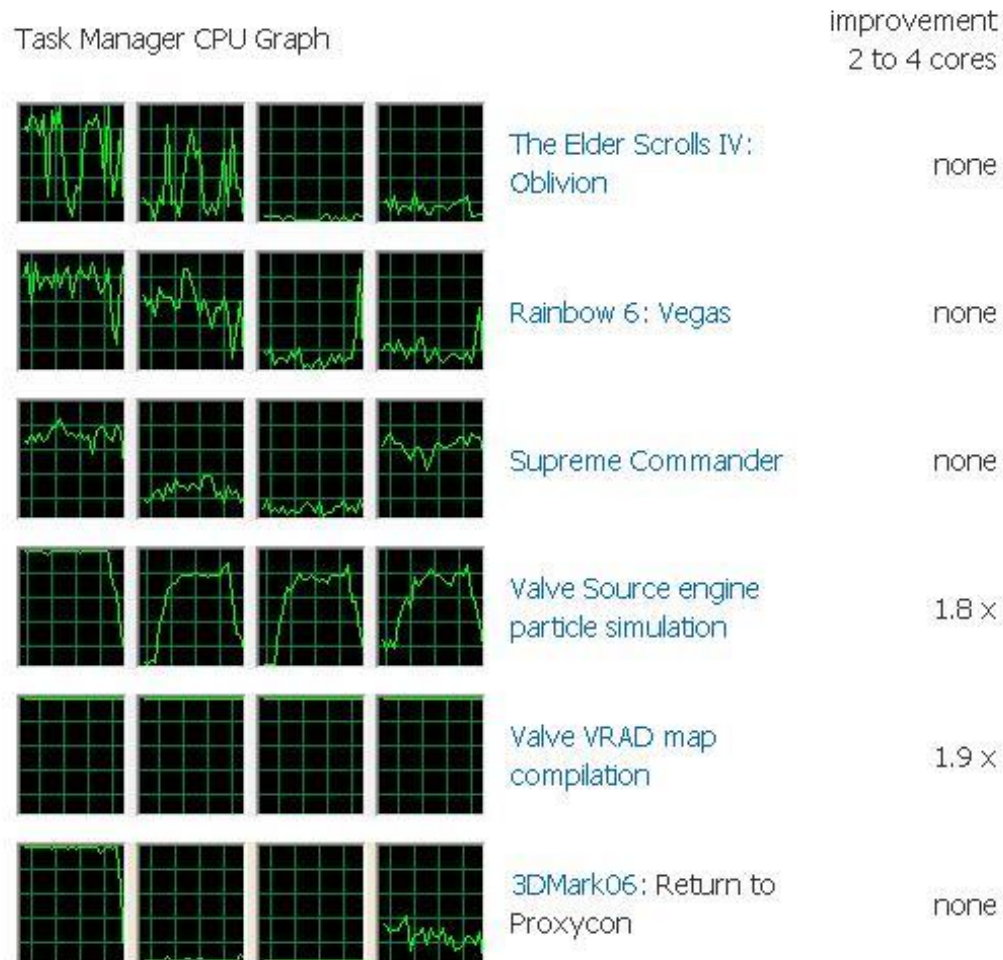
# Tests sintéticos “no oficiales”



# Tests sintéticos “no oficiales”



# Tests sintéticos “no oficiales”



# Contenidos

## ▶ Introducción:

- ▶ ¿Qué es un computador?
- ▶ Elementos constructivos de un computador
- ▶ Concepto de estructura y arquitectura

## ▶ Computador Von Neumann:

- ▶ Modelo Von Newmann
- ▶ Instrucciones máquina y programación
- ▶ Fases de ejecución de una instrucción

## ▶ **Características de un computador y tipos:**

- ▶ Parámetros característicos de un computador
- ▶ **Tipos de computadores**
- ▶ Evolución histórica



# Tipos de computadores

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

# Tipos de computadores

Nombre	Objetivos	Ejemplo	Aspectos de diseño
Desktop	Diseñados para ofrecer un buen rendimiento a los usuarios	Actualmente, la mayor parte son portátiles	<ul style="list-style-type: none"> <li>• Relación precio-rendimiento</li> <li>• Energía</li> <li>• Rendimiento de los gráficos</li> </ul>
Dispositivos móviles personales	Dispositivos sin cables con interfaz de usuario multimedia	Móviles, tablets,...	<ul style="list-style-type: none"> <li>• Precio</li> <li>• Energía</li> <li>• Rendimiento</li> <li>• Tiempo de respuesta</li> </ul>
Servidores	Usados para ejecutar aplicaciones de alto rendimiento o escala	Dan servicio a múltiples usuarios de forma simultánea	<ul style="list-style-type: none"> <li>• <i>Throughput</i> (Tasa de procesamiento)</li> <li>• Disponibilidad</li> <li>• Fiabilidad</li> <li>• Energía</li> <li>• Escalabilidad</li> </ul>
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	<ul style="list-style-type: none"> <li>• <i>Precio-rendimiento</i></li> <li>• <i>Throughput</i> (Tasa de procesamiento)</li> <li>• <i>Disponibilidad</i></li> <li>• <i>Fiabilidad</i></li> <li>• <i>Energía</i></li> <li>• <i>Escalabilidad</i></li> </ul>
Empotrados	Computador que se encuentra dentro de otro sistema para controlar su funcionamiento	Lavadoras, TV, MP3, consolas de videojuegos, etc.	<ul style="list-style-type: none"> <li>• Precio</li> <li>• Energía</li> <li>• Rendimiento de la aplicación específica</li> </ul>

# Contenidos

## ▶ Introducción:

- ▶ ¿Qué es un computador?
- ▶ Elementos constructivos de un computador
- ▶ Concepto de estructura y arquitectura

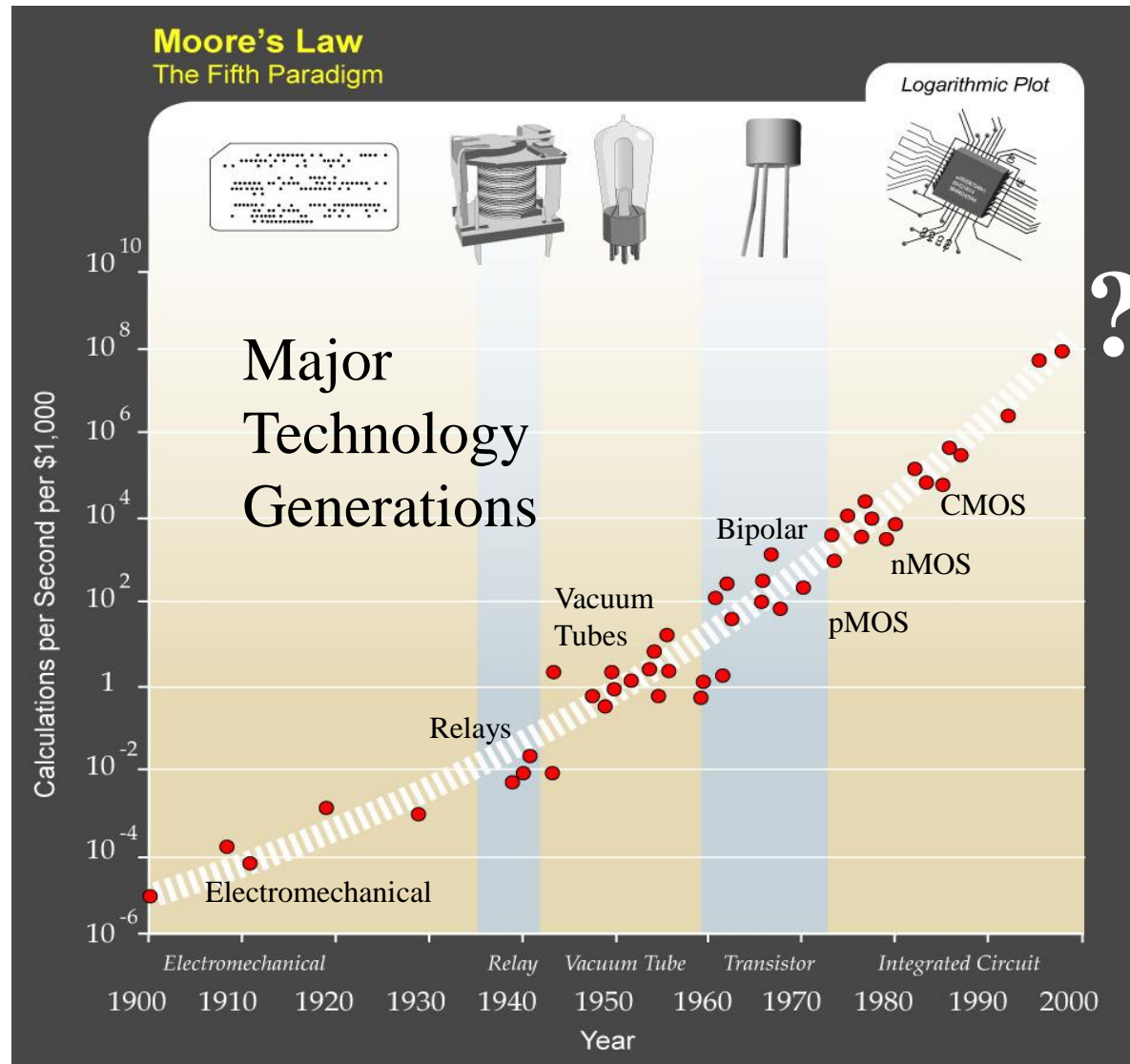
## ▶ Computador Von Neumann:

- ▶ Modelo Von Newmann
- ▶ Instrucciones máquina y programación
- ▶ Fases de ejecución de una instrucción

## ▶ **Características de un computador y tipos:**

- ▶ Parámetros característicos de un computador
- ▶ Tipos de computadores
- ▶ **Evolución histórica**

# Principales generaciones tecnológicas



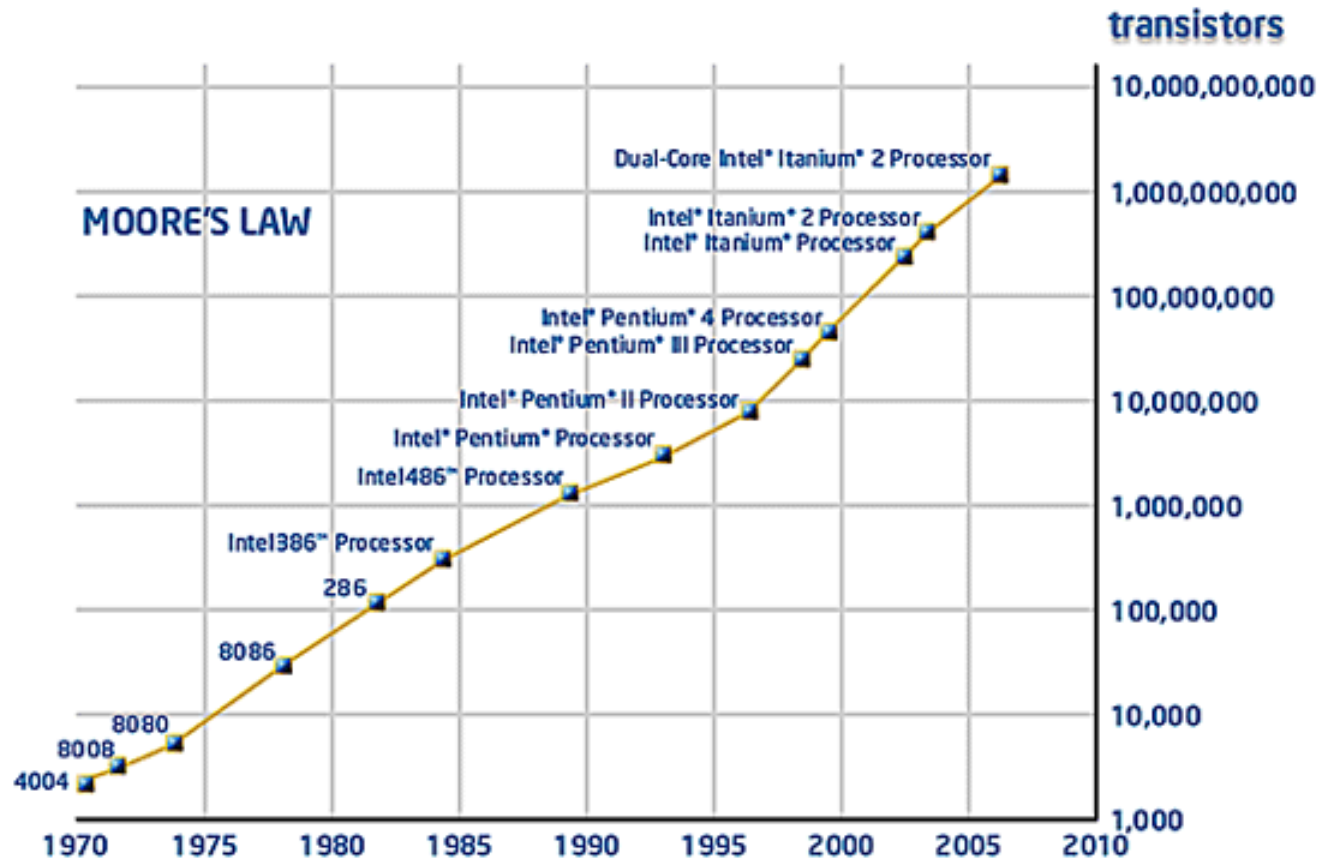
[from Kurzweil]

# Microprocesador

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

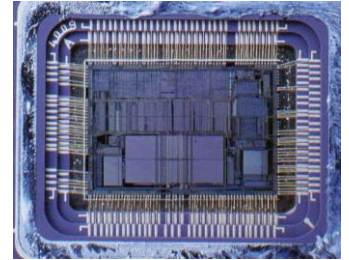


# Ley de Moore



# 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 micrómetros
- ▶ Hoy en día se consiguen chips con distancias de 5 nanómetros
- ▶ 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/computerI.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”

Grupo ARCOS

**uc3m** | Universidad **Carlos III** de Madrid

# Tema 1

## Introducción a los computadores

Estructura de Computadores  
Grado en Ingeniería Informática