



**UNCUYO**  
UNIVERSIDAD  
NACIONAL DE CUYO



**FACULTAD  
DE INGENIERÍA**

**Licenciatura en Ciencias de la  
Computación**

# **Arquitectura de Computadoras**

## **Unidad 4:**

### **Entrada y Salida**

### **Buses de Entrada Salida**



# Temario

## 1. Entrada Salida - Modos de transferencia

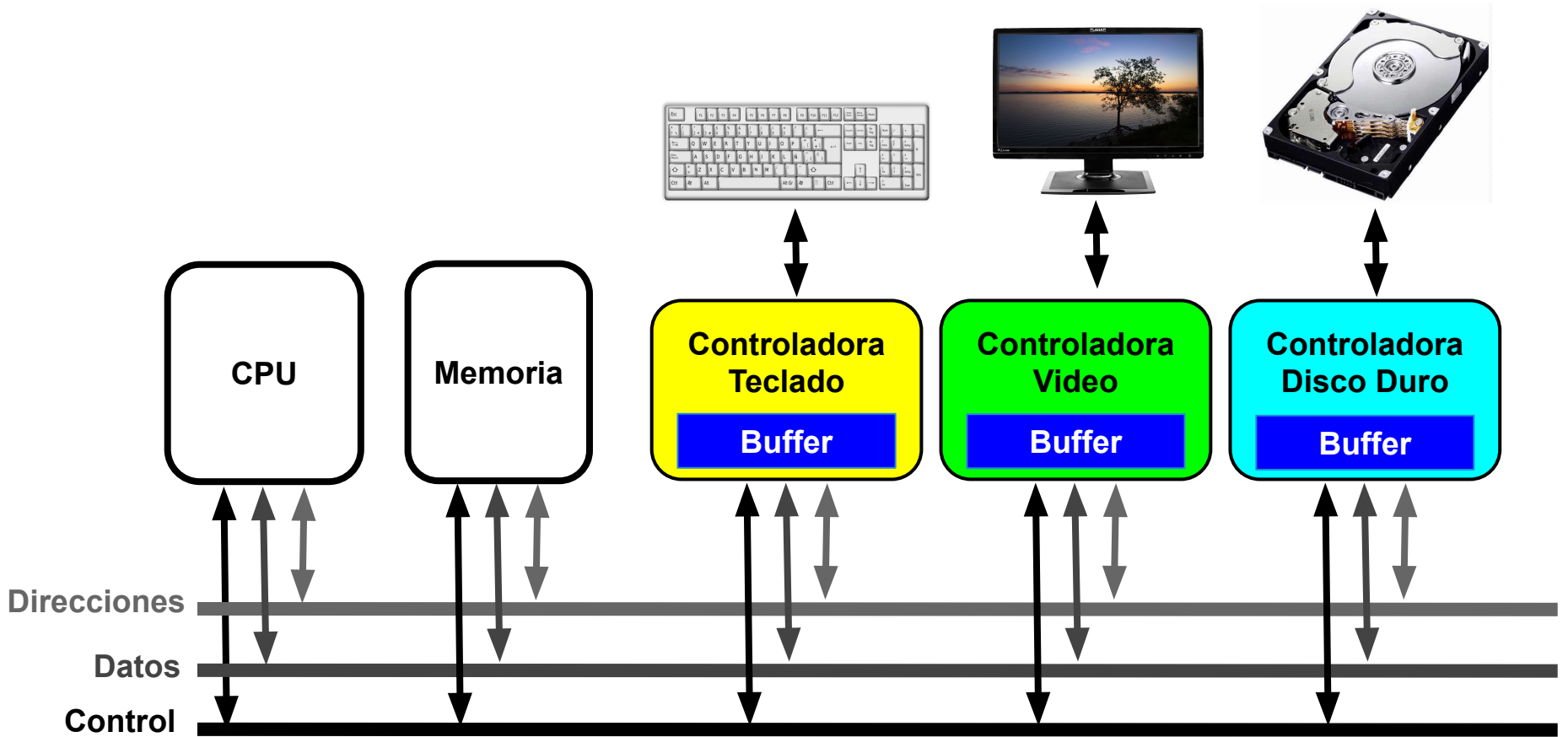
- 1.1. Microprocesador -> Dispositivos E/S
- 1.2. Dispositivos E/S -> Microprocesador

## 2. Componentes del sistema de entrada salida

- 2.1. Controladores
- 2.2. Manejador de E/S (Driver)
- 2.3. Puertos
- 2.4. Interfaces dispositivos almacenamiento masivo
- 2.5. Procesador de E/S

## 3. Buses de Entrada Salida

- 3.1. Tipos de buses
- 3.2. Necesidad y características
- 3.3. Bus ISA
- 3.4. BUS PCI
- 3.5. BUS PCI Express
- 3.6. BUS USB





## Entrada salida

- Se requiere de una **memoria intermedia** entre los dispositivos de entrada salida y el procesador para compensar las diferencias de velocidades.
  - Esta memoria está dividida en:
    - Instrucciones o comandos.
    - Estado (para que el dispositivo indique su estado, como ocupado, listo para trabajar, trabajo finalizado, códigos de error, etc.).
    - Datos.
- El procesador debe **direccionar** esta memoria para leerla o escribirla.



# Comunicación Microprocesador -> dispositivo de e/s

¿Cómo lo hacemos?: Ejecutando Instrucciones que escriban la memoria del dispositivo de e/s.

**Problema: Es necesario direccionar el periférico**

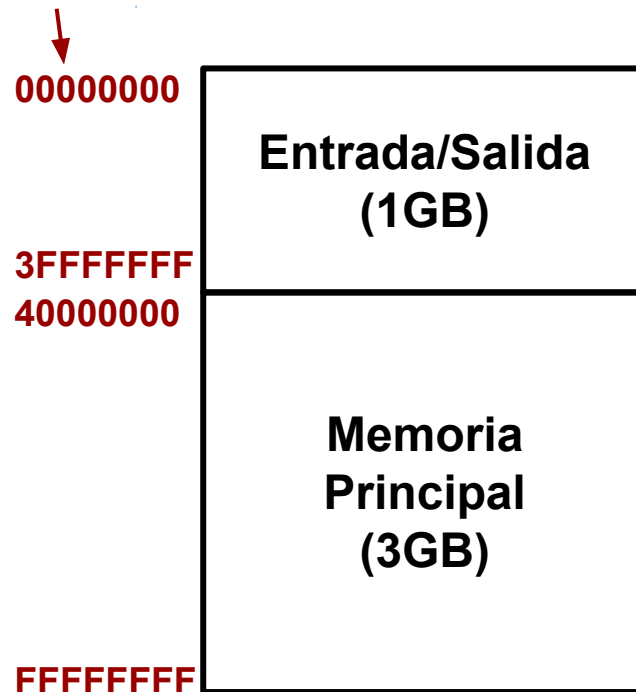
- **Dos métodos:**
  - **Entrada-salida mapeada en memoria (ARM, RISC).**
  - **Instrucciones especiales de entrada salida (Intel x86, IBM 370, Algunos micro-controladores sencillos)**

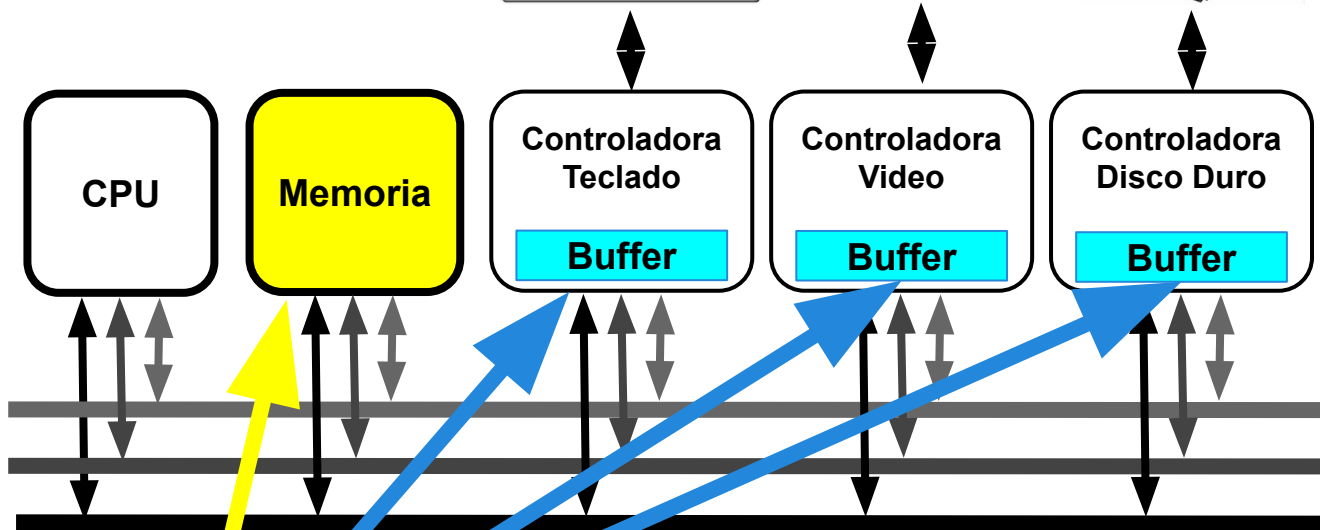
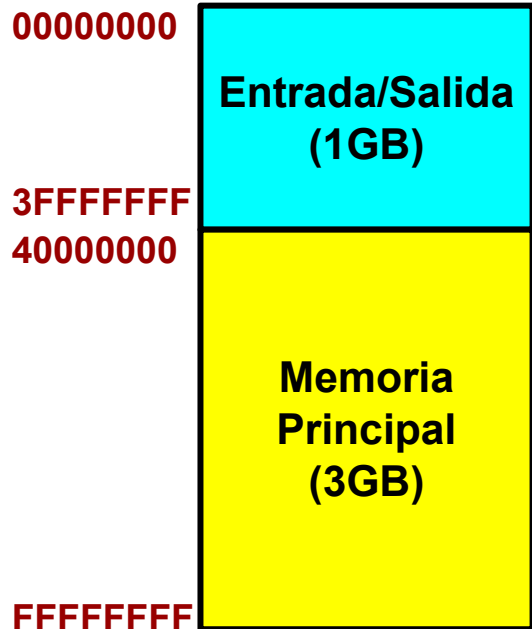


## Entrada salida mapeada en memoria

- Una parte del espacio de direcciones se asigna a dispositivos de E/S.
- El microprocesador escribe o lee datos desde el dispositivo con las **mismas instrucciones de acceso a memoria**.
  - Mismas señales en el bus de control.

Posiciones que el procesador puede direccionar





**MOVB A,[0x40000001]**  
**MOVB B,[0x20000001]**

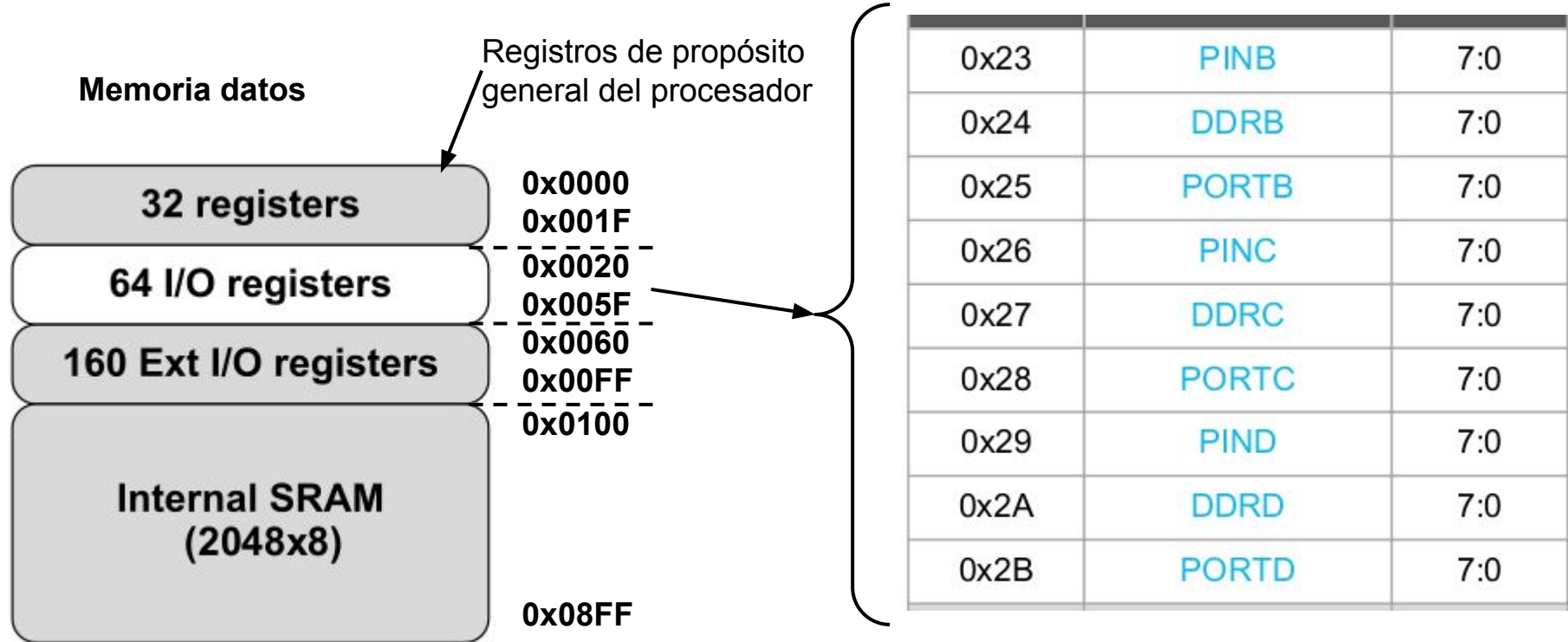


## **Entrada salida mapeada en memoria**

- **Ejemplo escritura: El procesador coloca una dirección de un dispositivo de E/S en el bus de direcciones y un dato en el bus de datos.**
- **La memoria decodifica esa dirección, sabe que son para E/S e ignora la transacción.**
- **Uno de los controladores de E/S sabe que esta dirección en el bus le corresponde.**
  - **Decodifica la dirección**
  - **Captura el dato**
  - **Lo transmite al dispositivo**
- **Si tengo x líneas de direcciones, la suma de las direcciones de memoria más las direcciones de I/O no podrá ser mayor a  $2^x$ .**



## Ejemplo 2: Atmel ATmega328/P (Arduino Uno) E/S mapeada memoria



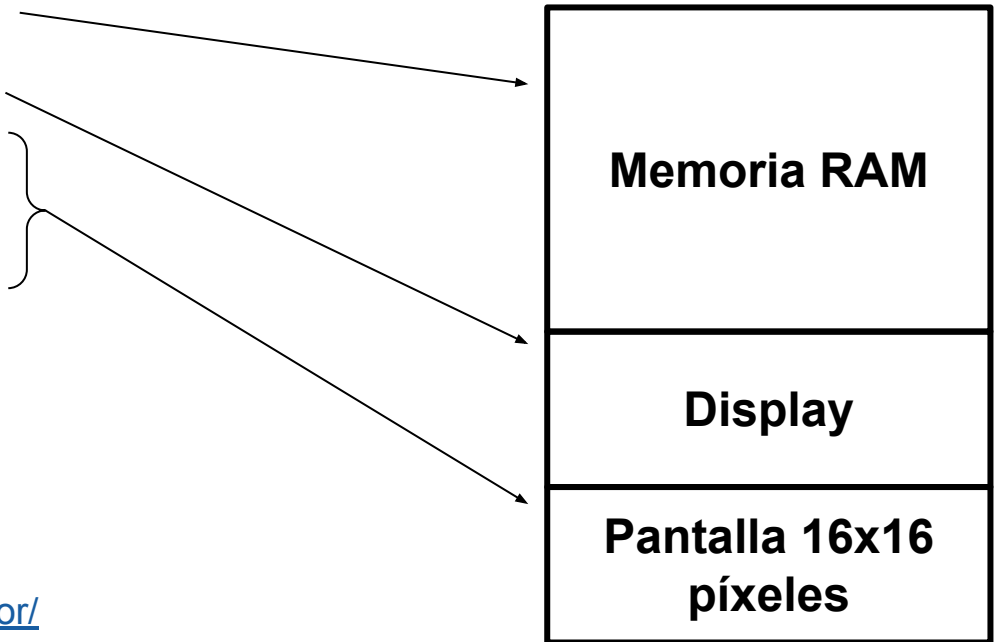
**MOVB [0x0001],0x20; escribe en el registro 1.**  
**MOVB [0x0025],0x20; operación de entrada/salida**  
**MOVB [0x0125],0x20; escribe en memoria SRAM**



# Entrada salida mapeada en memoria

## Ejemplo con Assembler Simulator

```
MOVB [0x0030],0x43  
MOVB [0x02E0],0x43  
MOVB [0x0300],0x43  
MOVB [0x0301],0xF4  
MOVB [0x0302],0x43  
HLT
```



0x0000

0x02DF  
0x02E0

0x02FF  
0x0300

0x03FF



# Entrada salida mediante Instrucciones especiales

- **Diferentes instrucciones** para acceder a los periféricos o memoria.
  - **Líneas de control separadas** para diferenciar si la operación es sobre la memoria o un periférico.
- **Diferentes mapas de direcciones** para acceder a memoria o a los periféricos.
- **x86 Intel microprocessors (Instrucciones in, out).**
- **Si tengo X líneas de direcciones, puedo tener  $2^X$  direcciones de memoria y  $2^X$  direcciones de I/O.**

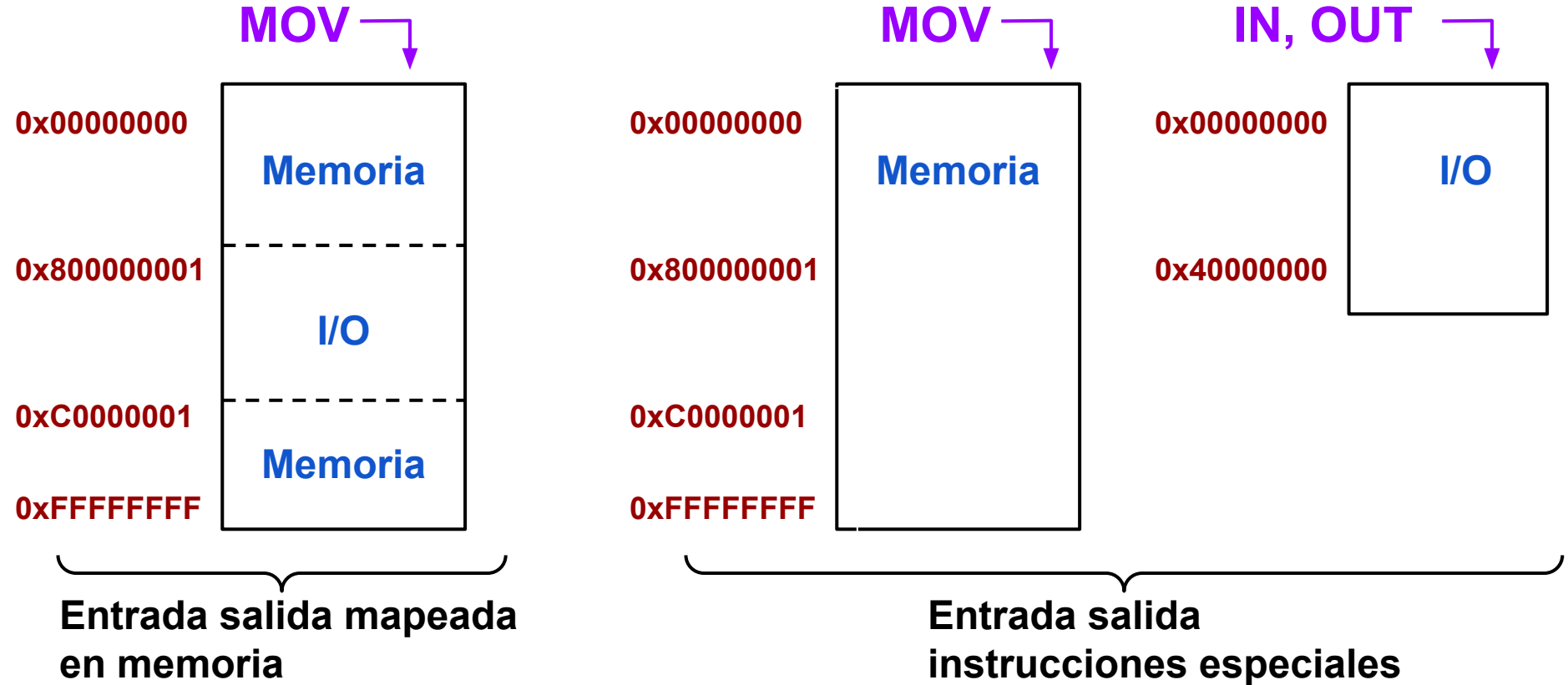
**Cuidado!:** Mismo bus de direcciones, pero diferentes líneas de control para diferenciar el destinatario.

Diferentes nombres según diferentes autores:

Morris Nano y William Stalling: isolated I/O (Entrada salida aislada)

Tanenbaum: Entrada salida mediante instrucciones especiales.

# Entrada salida mediante Instrucciones especiales



# Entrada salida mediante Instrucciones especiales

(Ejemplo en Assembler Simulator)

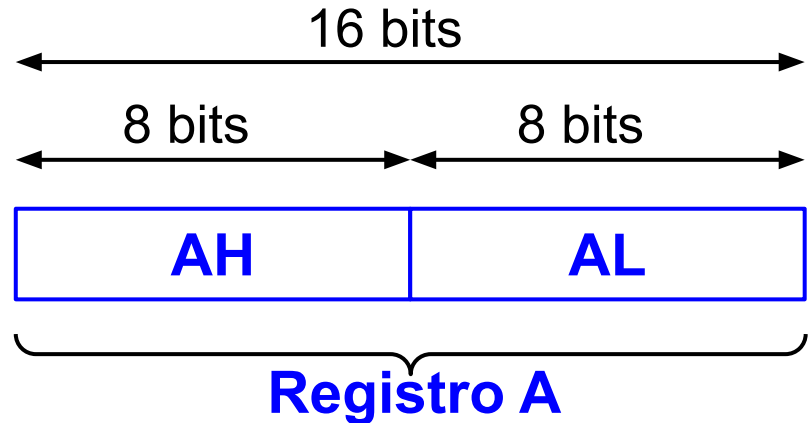
```
IN 0x0006  
MOVB [0x02E2],AL  
HLT
```

Mueve al registro A el contenido del registro KPDDATA, que contiene el código ASCII de la última tecla presionada



Nota: El simulador simula un procesador de 16 bits. Los registros tienen 16 bits. Pero cada posición de memoria tiene 8 bits.

Los registros se dividen en dos partes, parte alta o AH y parte baja o AL.





# Comunicación Dispositivo de e/s -> Microprocesador

- **Problema: El dispositivo necesita enviar o pedir datos al procesador (que pueden estar en memoria o registros), pero el microprocesador está siempre ocupado**

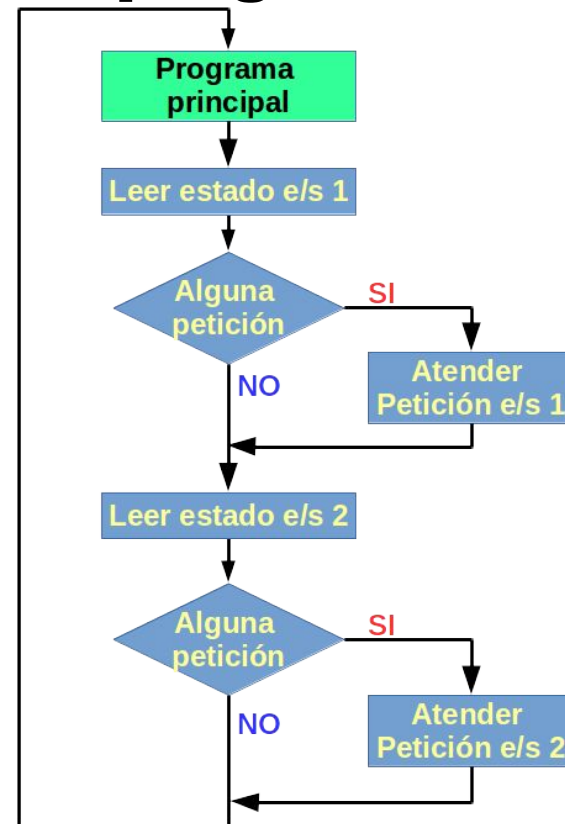
## Cinco métodos:

- **Transferencia controlada por programa (también llamada Polling o E/S programada).**
- **Interrupciones**
- **DMA (Direct Memory Access)**
- **Procesador de E/S**
- **Bus Mastering**



# 1) Transferencia controlada por programa

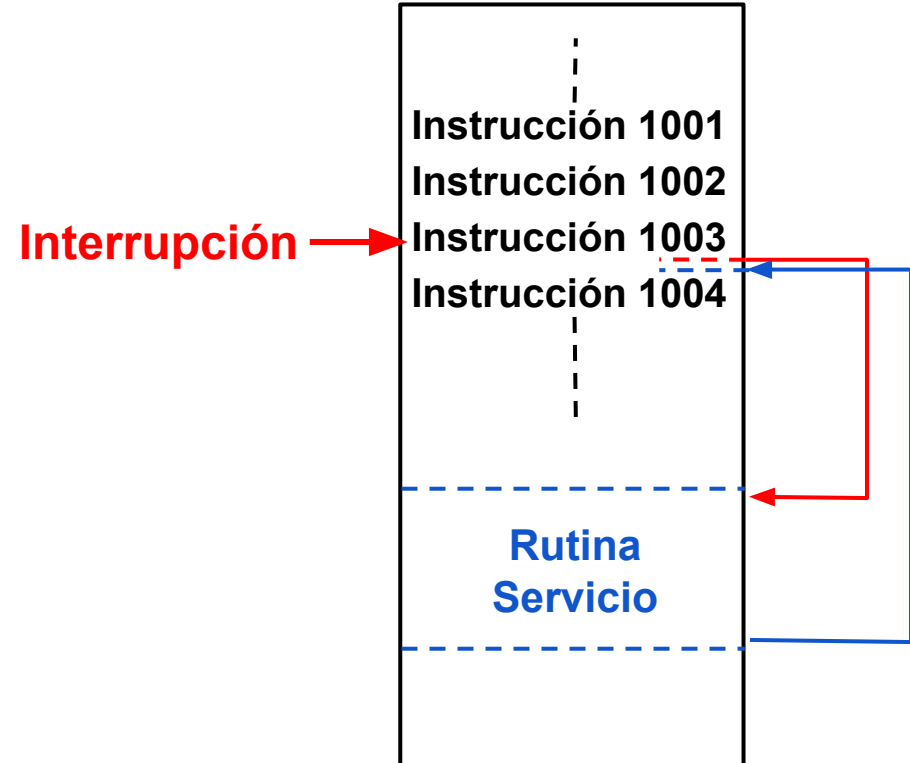
- **Comprobación periódica del estado de los dispositivos.**
- **Técnica sencilla**
  - **No requiere hardware adicional.**
  - **Fácil implementación en bajo nivel.**
- **Desperdicia tiempo del procesador**
  - **Gran número de periféricos**
  - **Los periféricos son lentos**
- **Pueden perderse datos**
- **Útil para procesadores poco potentes o dedicados a monitorear un solo periférico.**





## 2) Interrupciones de Hardware

- **Interrupción** del programa en curso, causada por un evento **asíncrono** no relacionado con las instrucciones en ejecución, provocan que el procesador ejecute una **rutina de servicio**, y luego volver al programa en curso.
- No impide que las instrucciones en curso completen su ejecución.







# Interrupciones

## Problemas a resolver:

- **Detectar la fuente de interrupción. N° de IRQ (Interrupt Request)**
  - Línea de control para cada fuente de interrupción
  - Bus de datos
- **Decidir en caso de dos dispositivos interrumpiendo al mismo tiempo. Prioridades.**
  - Por hardware
  - Por software (según el orden en que el procesador pregunte).
- **Decidir si “interrumpir” una rutina de servicio de una interrupción. Interrupciones anidadas o no.**
- **Guardar la palabra de estado o contexto.**
- **Ejecutar un “programa” que atienda al periférico que interrumpió. Rutina de servicio**

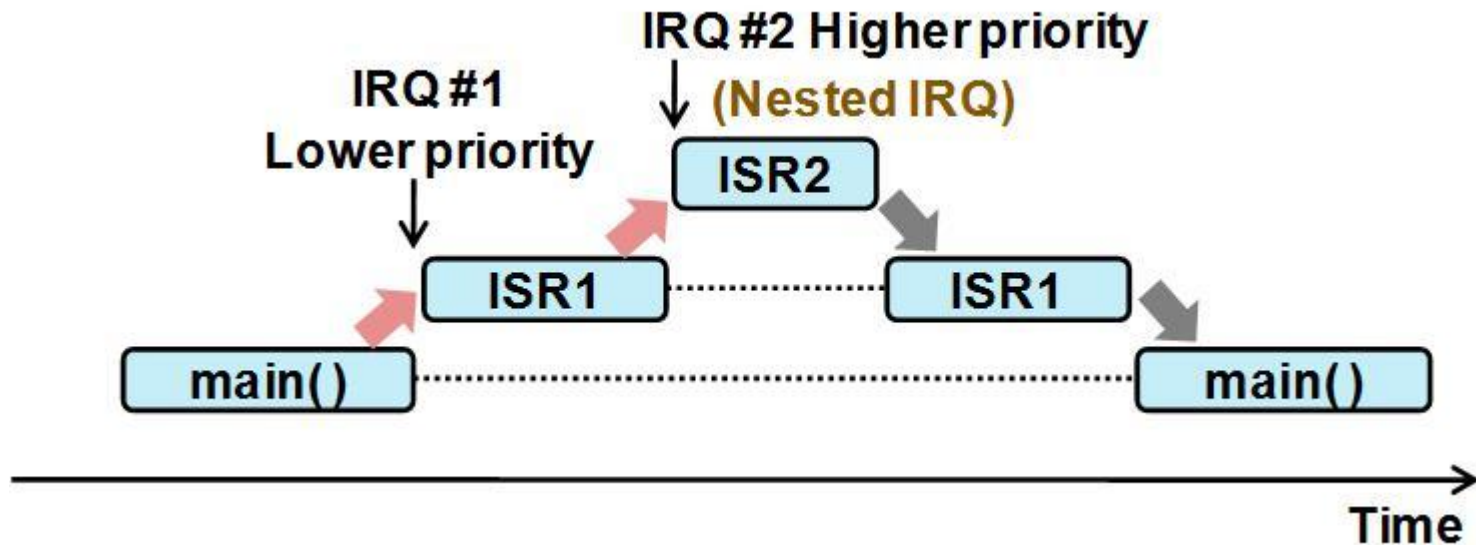


## Interrupciones x86

- **Diferenciación de las interrupciones: N° de IRQ.**
  - IRQ0: Reloj del sistema**
  - IRQ1: Teclado**
  - IRQ3: Puerto serie COM 2**
  - IRQ4: Puerto serie COM 1**
  - IRQ5: Floppy disk controller**
  - IRQ12: PS/2 mouse**
  - IRQ124: Placa Ethernet**
- **Prioridad configurable por:**
  - **Fija (no configurable).**
  - **Configurable:**
    - **Hardware.**
    - **Software (según el orden en que el procesador pregunte ¿Quién interrumpió).**

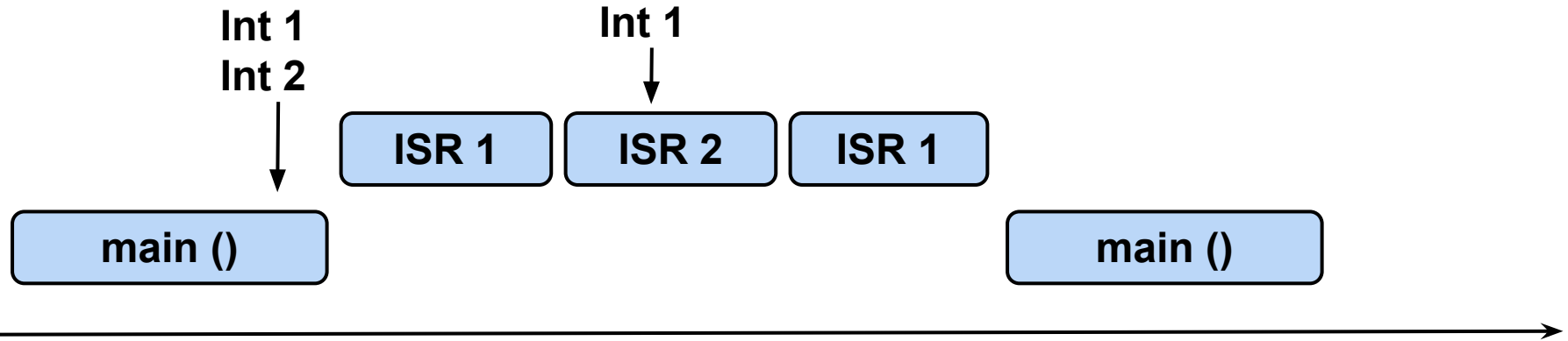


# Interrupciones Anidadas



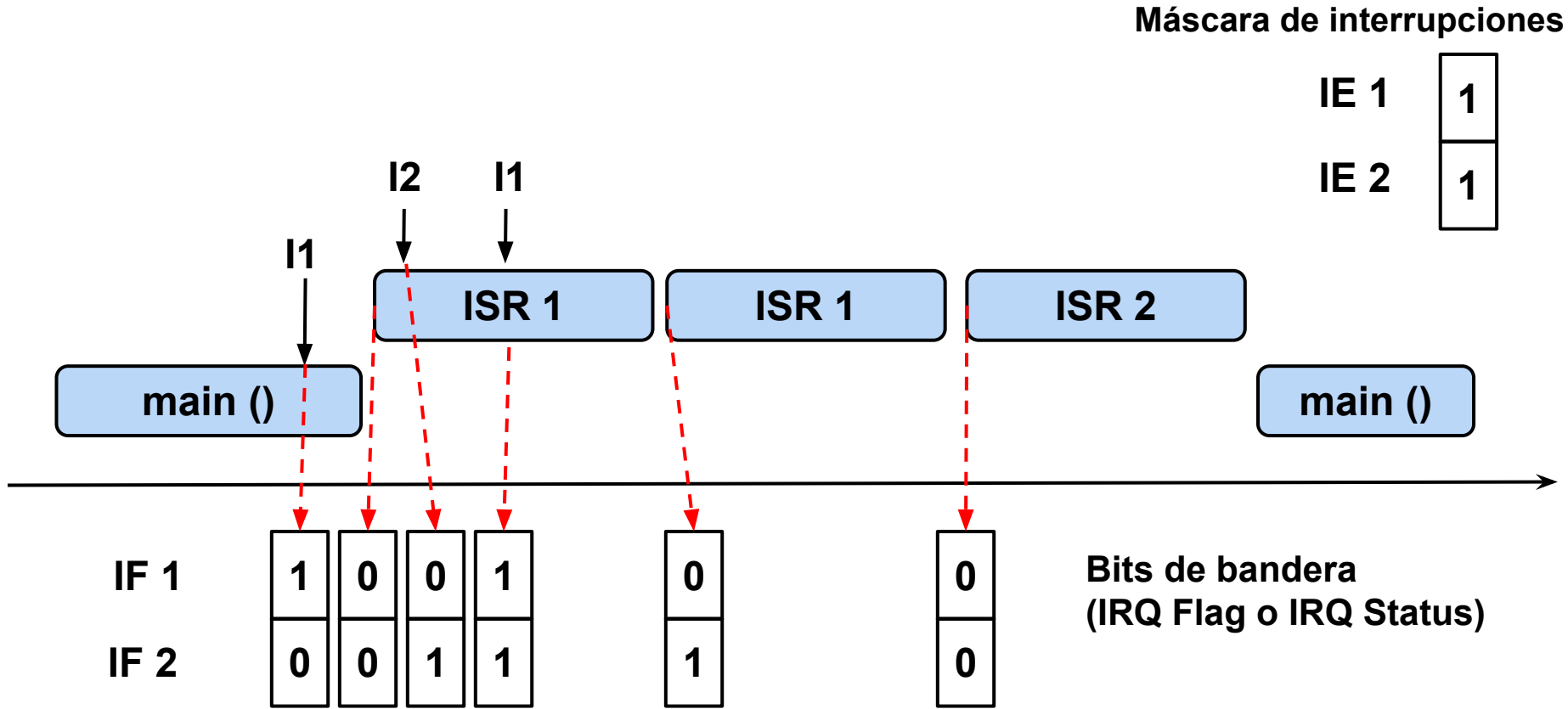


# Interrupciones no anidadas



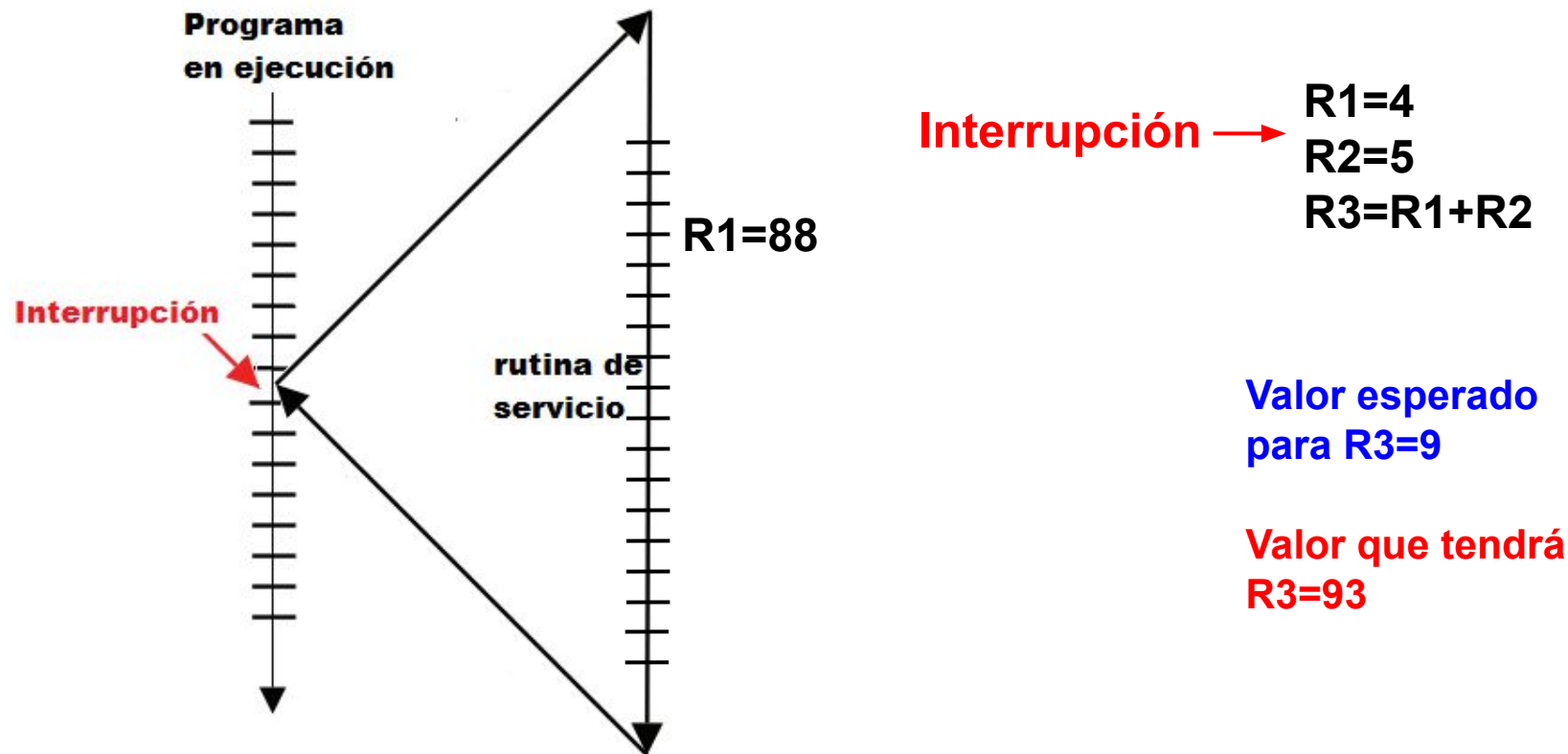
**Int 1: Mayor prioridad**  
**Int 2: Menor prioridad**

# Interrupciones - Registro de interrupciones



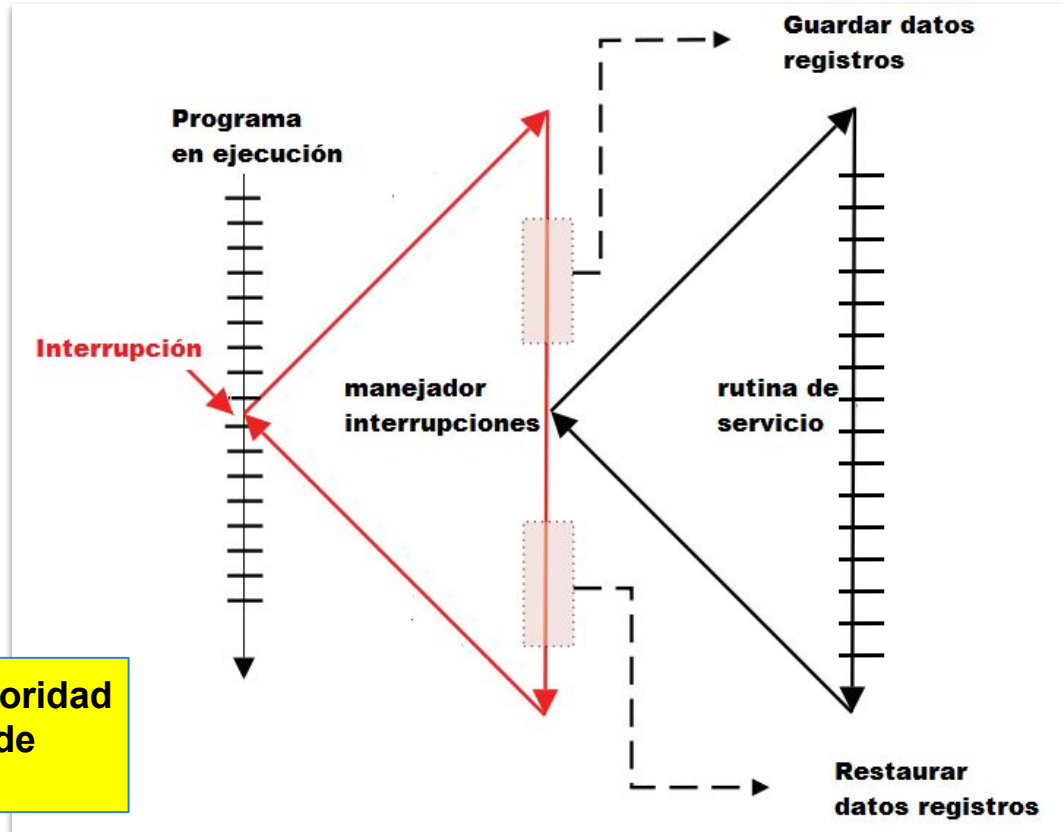


# Interrupciones - Necesidad de salvar el contexto



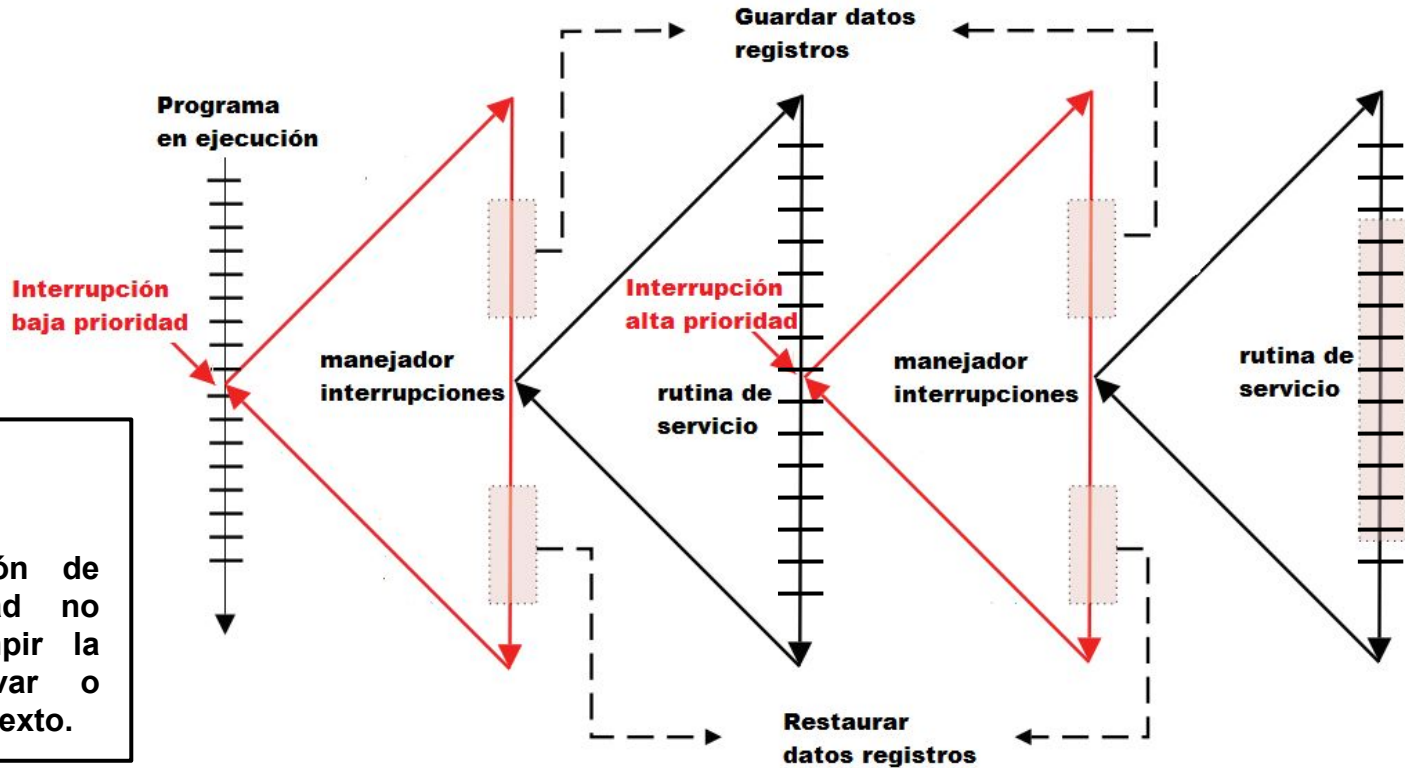
# Interrupciones - Necesidad de salvar el contexto

- Necesidad de **guardar el contexto**, contenido de los registros, incluido el **CP** (salvar las palabras de estado del programa, PSW: program status words).



Una interrupción de mayor prioridad no puede interrumpir la tarea de salvar o restaurar el contexto.

# Interrupciones anidadas - Salvar el contexto

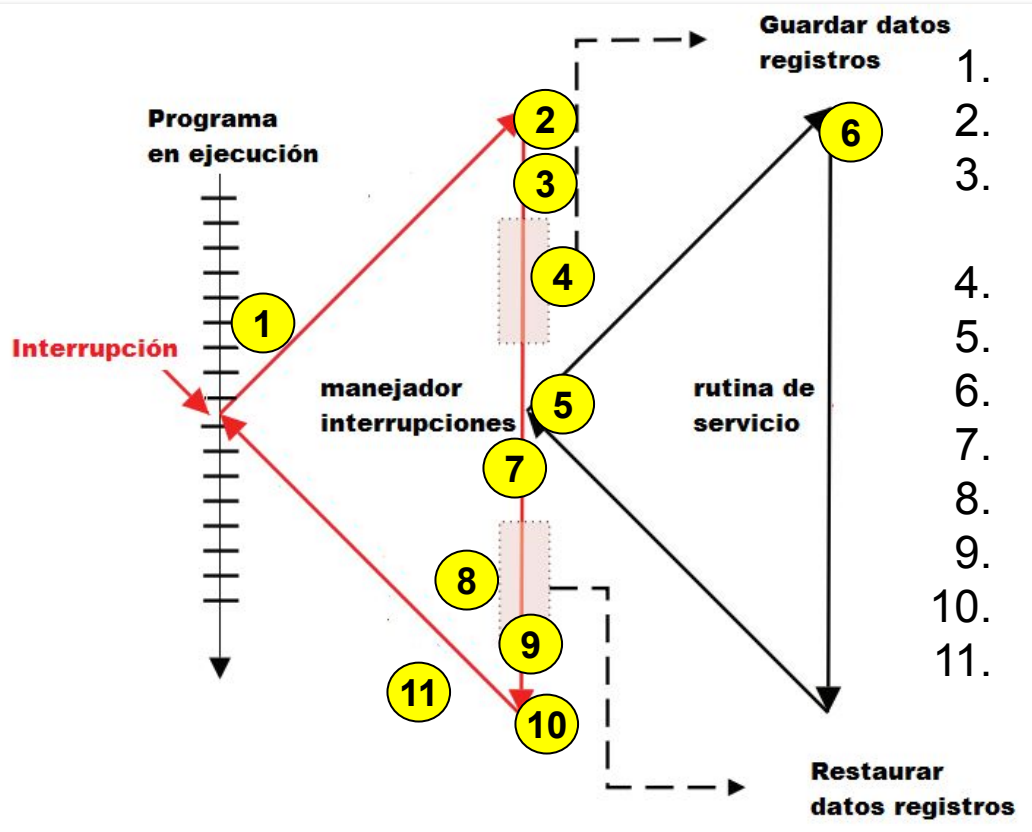


Una interrupción de mayor prioridad no puede interrumpir la tarea de salvar o restaurar el contexto.





# Interrupciones - Necesidad de salvar el contexto



1. Guardar el CP (lo realiza el procesador)
2. Deshabilitar interrupciones globalmente
3. Deshabilitar interrupciones de menor prioridad
4. Guardar contenido registros
5. Habilitar interrupciones globalmente
6. Ir a la rutina de servicio
7. Deshabilitar interrupciones globalmente
8. Restaurar registros (menos el CP)
9. Habilitar interrupciones de menor prioridad
10. Habilitar interrupciones globalmente
11. Restaurar el CP.



## Ejemplo ciclo de una Interrupción

Interrupción →

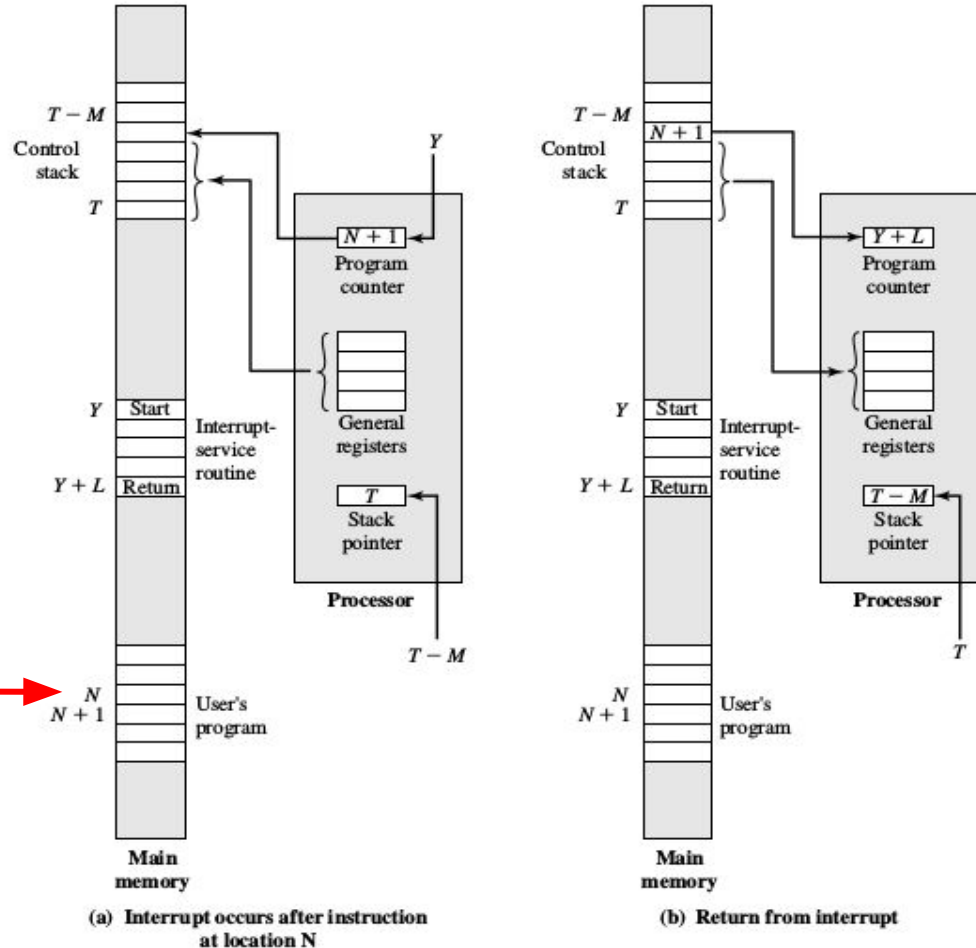


Figura obtenida de William Stallings  
"Computer Organization and Architecture"

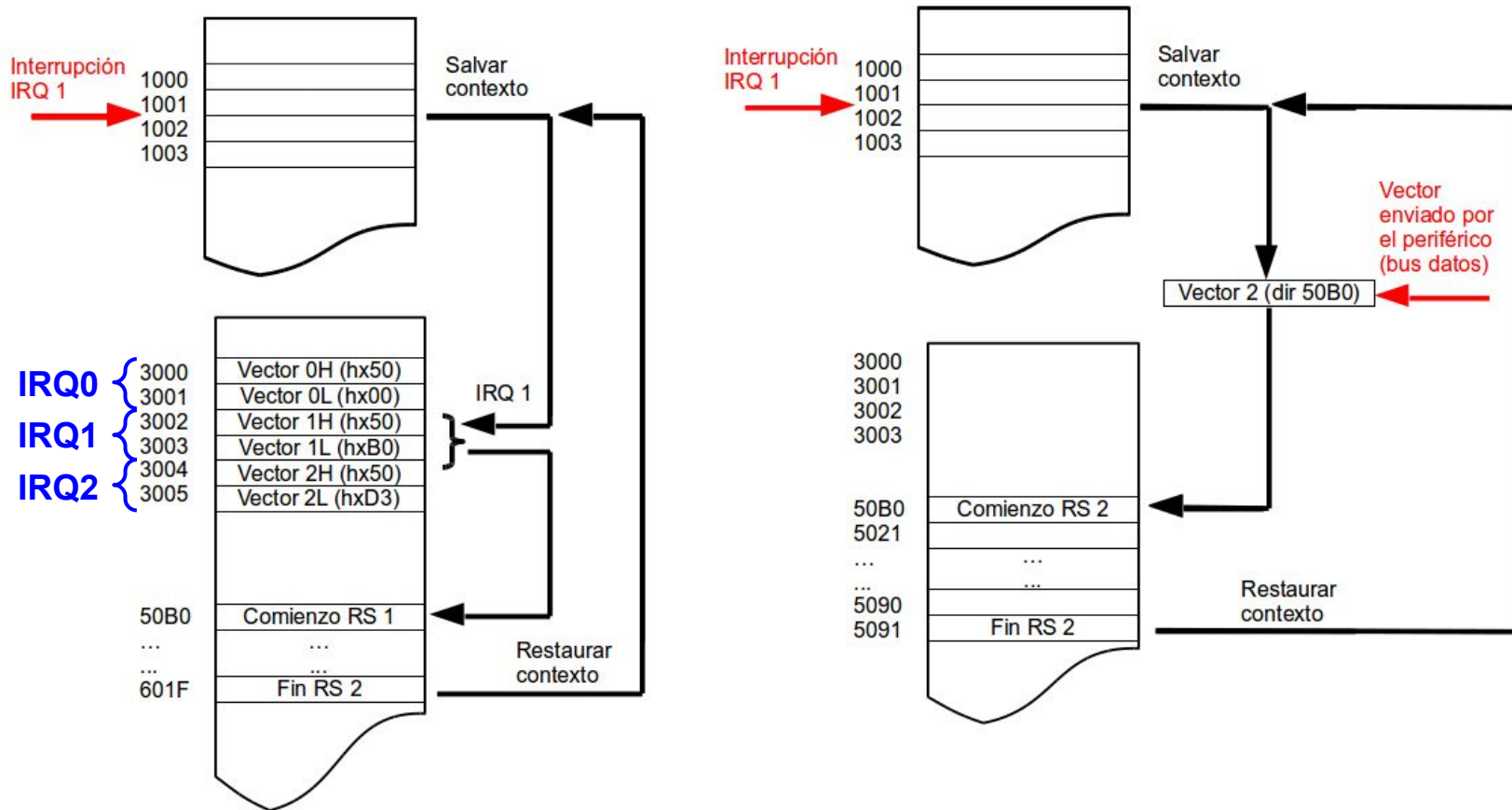


## Rutina de servicio

- **Subprograma que atiende al periférico que interrumpió.**
  - **Provista por los fabricantes o sistema operativo.**
- **Dirección de la rutina de servicio**
  - **Fija (cableada)**
  - **Vectorizada (dirección de comienzo de la rutina de servicio configurable).**
    - **Almacenado en la memoria RAM (tabla de vectores)**
    - **Provisto por el periférico**
    - **Sistema híbrido**

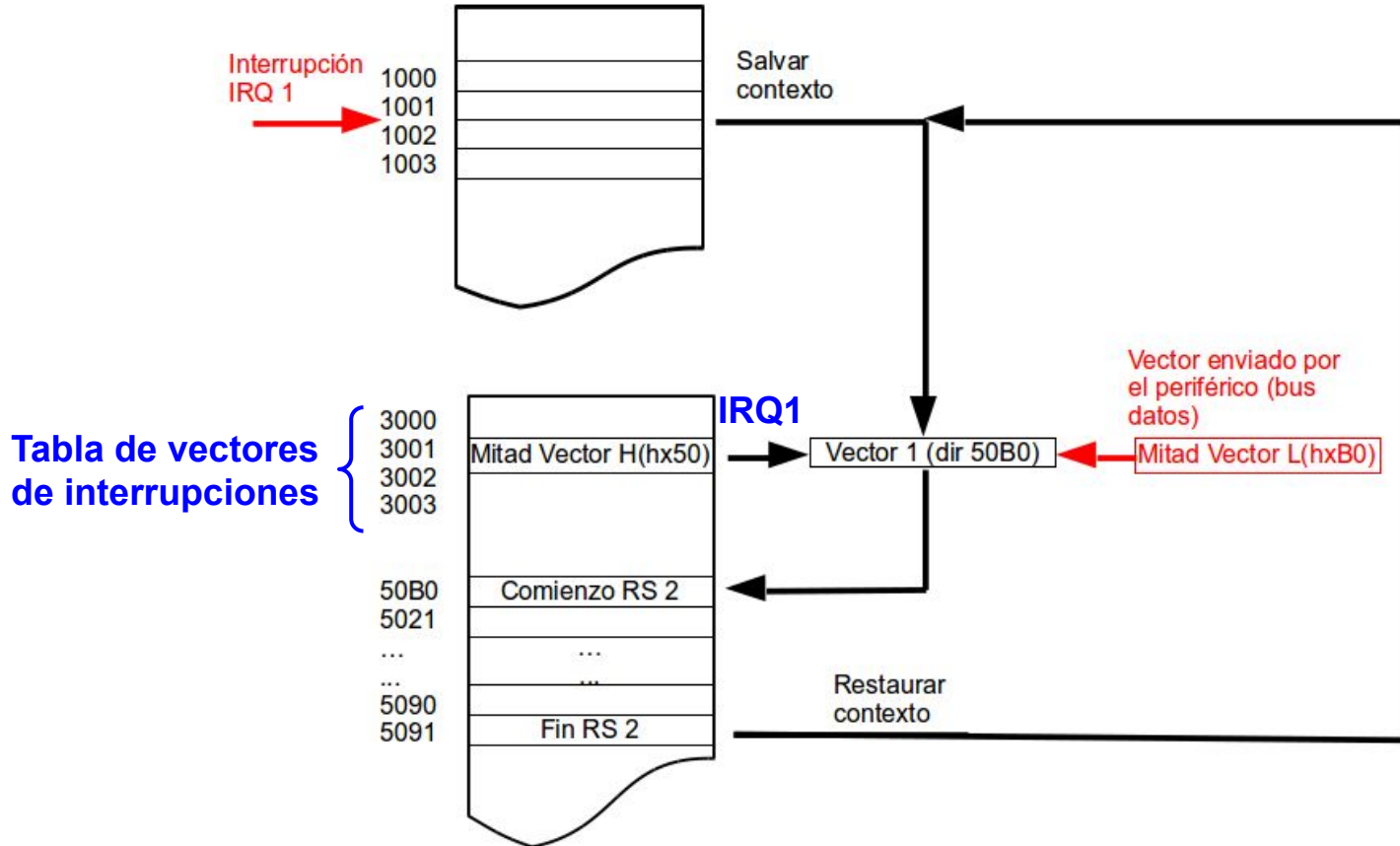


# Vector de Interrupción

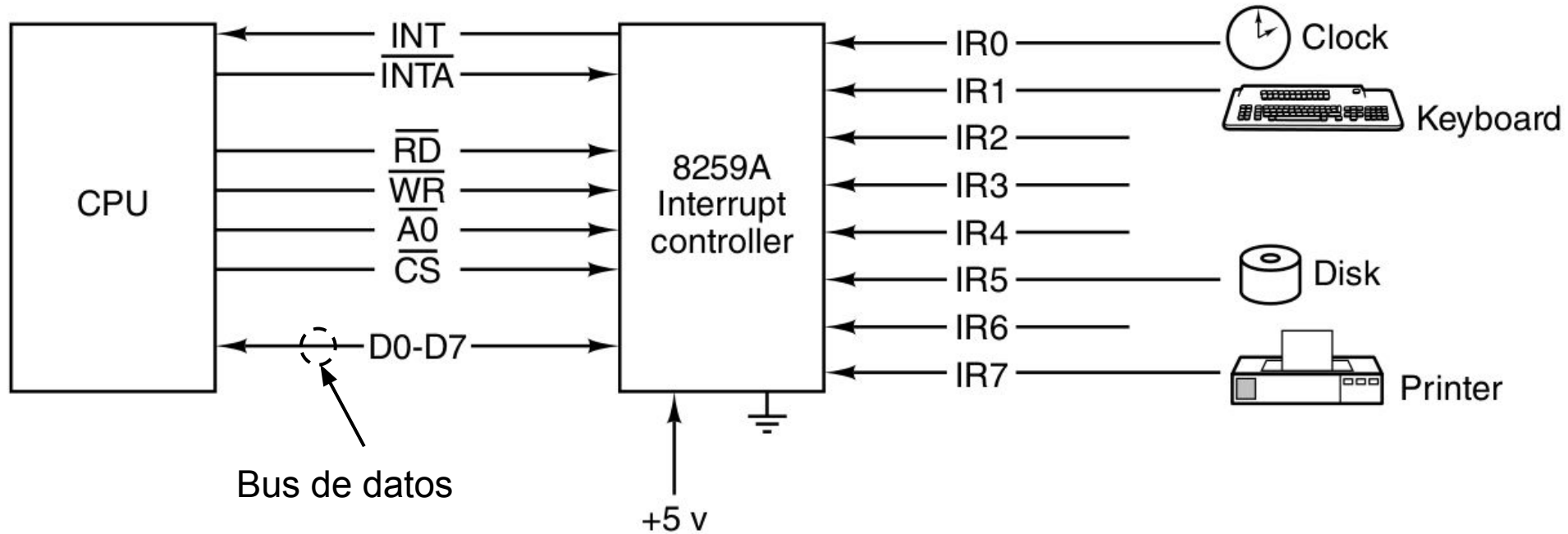




# Vector de Interrupción



# Ejemplo 1: Controlador de Interrupciones Programable (PIC): 8259A PIC





**Name:** EIMSK

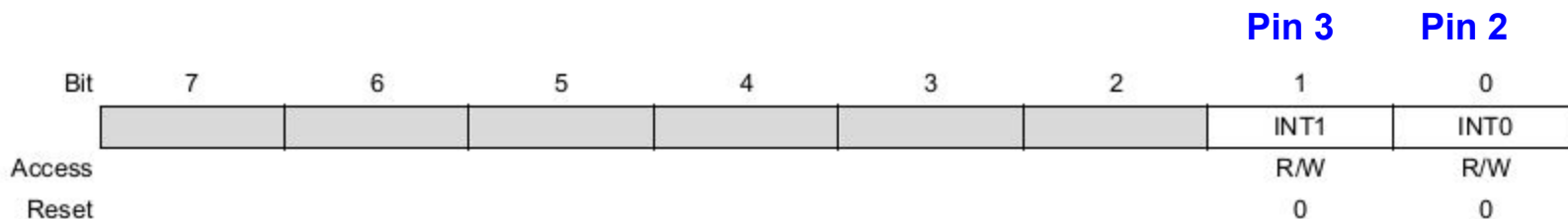
**Offset:** 0x3D

**Reset:** 0x00

**Property:** When addressing as I/O Register: address offset is 0x1D

## External Interrupt Mask Register

## Ejemplo 4: ATmega328



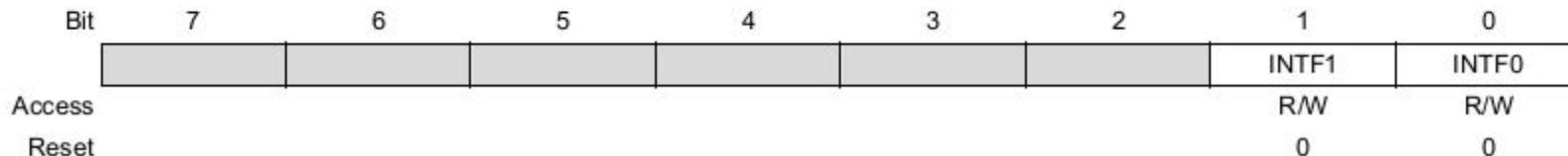
**Name:** EIFR

**Offset:** 0x3C

**Reset:** 0x00

**Property:** When addressing as I/O Register: address offset is 0x1C

## External Interrupt Flag Register



## Interrupciones en alto nivel

- En lenguajes de alto nivel las rutinas de servicio se escriben como **funciones (o subrutinas) callback**
  - Serán llamadas cuando se produzca la interrupción.
  - Deben ser escritas por el programador

```
attachInterrupt (digitalPinToInterrupt (2) , ISR_pin2 , FALLING) ;
```

```
.....  
void ISR_pin2(){  
  println("Rutina de servicio");  
  for(int i=1;i<8;i++){  
    .....  
    .....  
  }  
}
```

**Rutina de  
servicio**

```
call ISR_pin2()
```



<sup>1</sup>Función que se pasa como un argumento a otra función. Se invoca dentro de la función externa para completar alguna tarea.



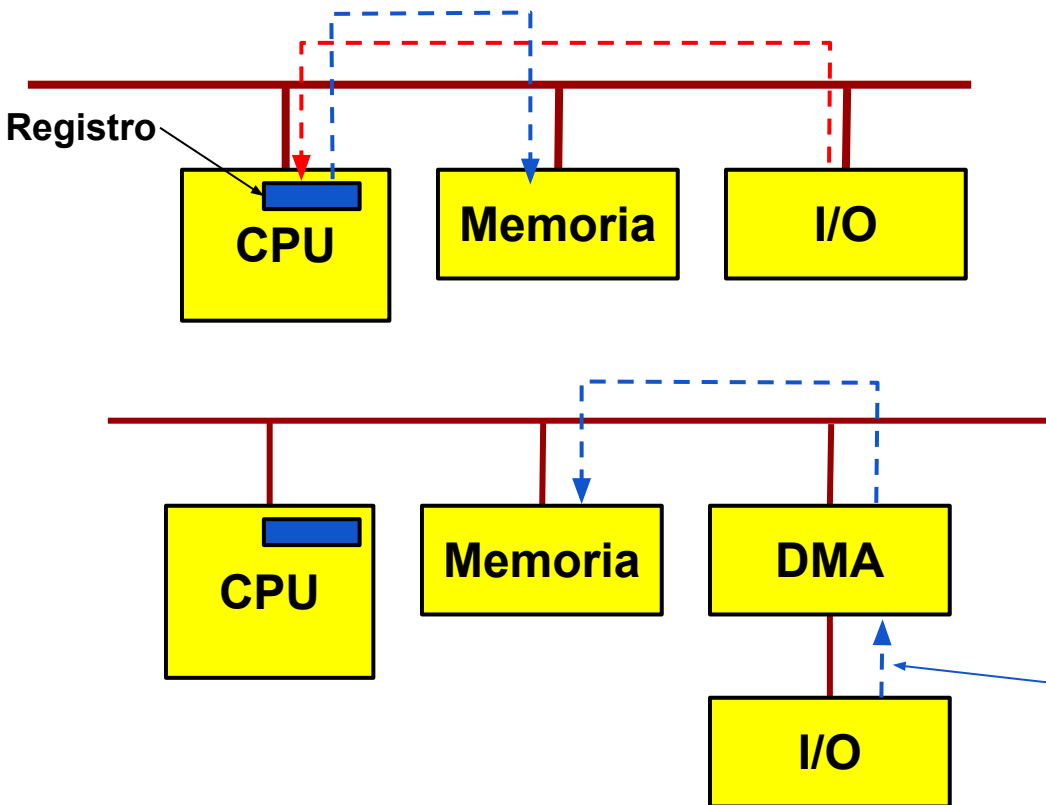


### 3) DMA (Direct Memory Access)

- **Útil cuando deben transferirse grandes volúmenes de datos (que se almacenan en memoria).**
- **El procesador cede al módulo DMA la tarea de transferir los datos.**
- **Nuevo bloque funcional: controlador DMA**
- **Tareas realizadas por el módulo DMA cuando realiza una transferencia:**
  - **El controlador DMA se convierte en master del bus principal.**
  - **Gestiona la transferencia directamente con la memoria.**
  - **Cuando se completa la transferencia el controlador DMA interrumpe al procesador.**
- **Incorporado con el Bus ISA.**



# DMA (Direct Memory Access)



Para transferir cada palabra desde un dispositivo de E/S a la memoria (o viceversa) se ocupa el bus de direcciones:

- **2 veces sin DMA (2 direccionamientos)**
- **1 vez con DMA (1 direccionamiento).**

Esta transferencia no ocupa el bus principal



### 3) DMA (Direct Memory Access)

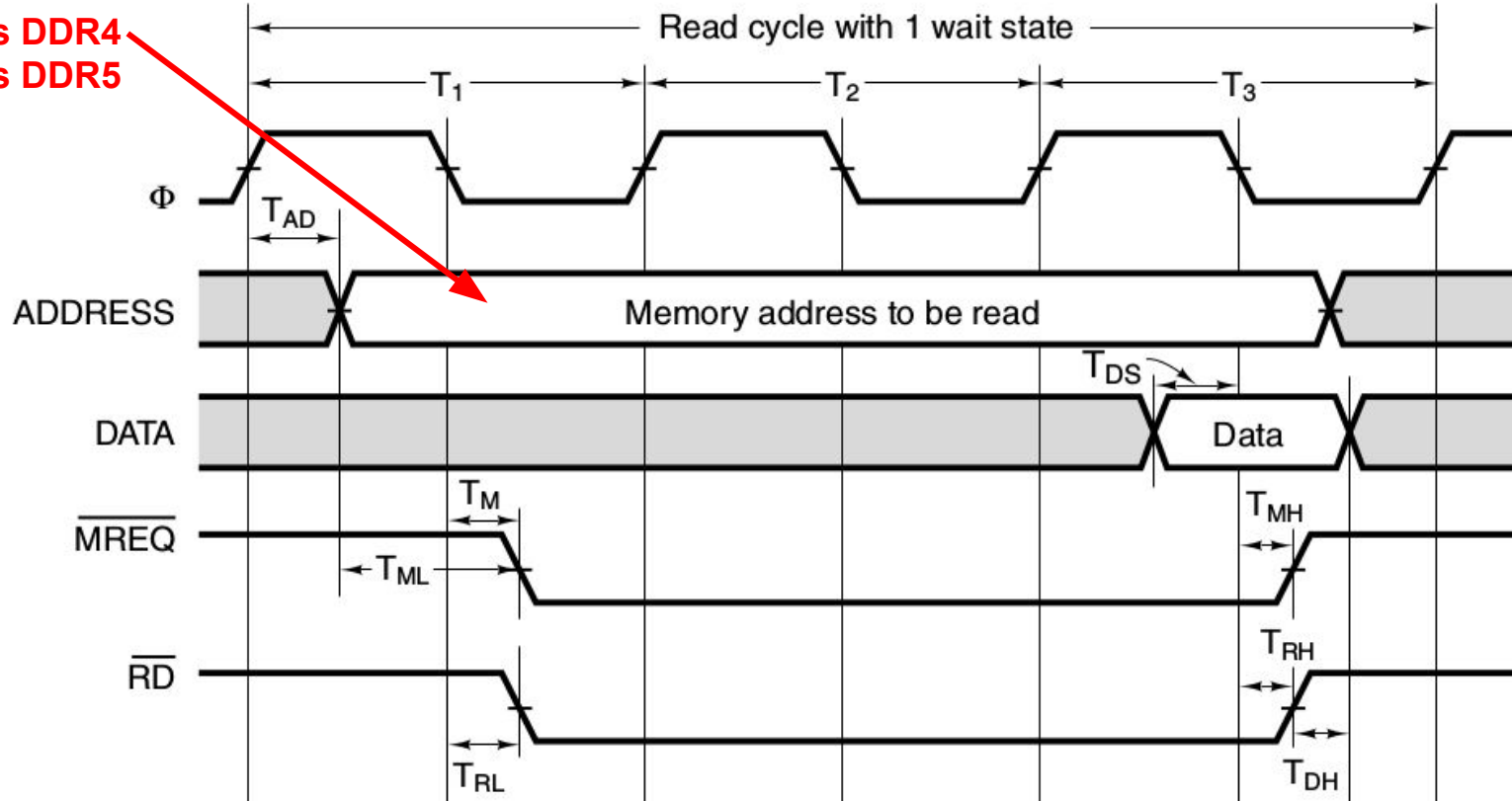
#### Fundamentos:

1. Para transferir un dato, **sin DMA se requieren 2 accesos** al bus principal, **con DMA, se requiere un solo acceso.**
2. Aprovecha la capacidad de **transferencia de bloques de datos** de las memorias actuales.
  - a. Se direcciona una sola vez, luego se accede a posiciones consecutivas de memoria.
3. Muy eficiente en combinación con memorias caché.

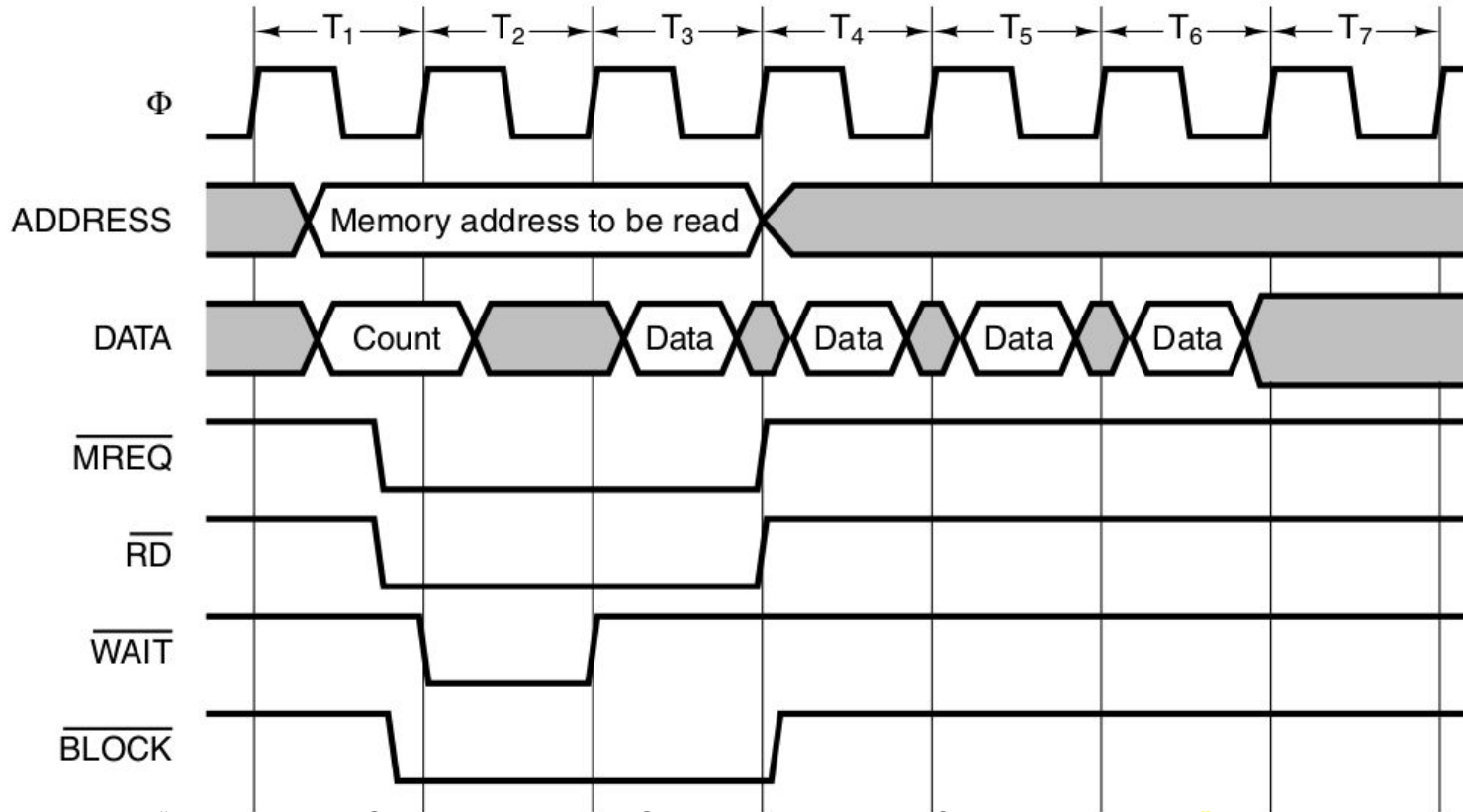


# Bus síncrono

18 ciclos DDR4  
46 ciclos DDR5

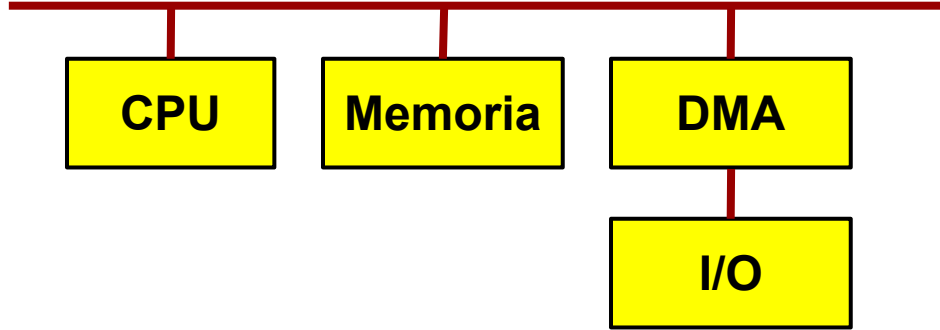


# Transferencia de bloques de datos (Usado por el DMA)

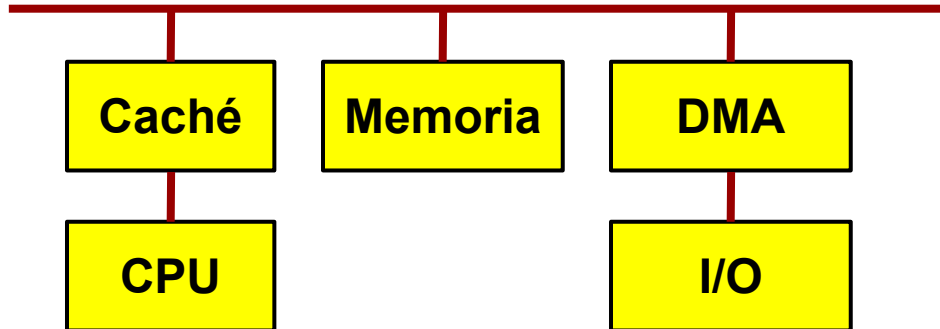




# DMA (Direct Memory Access)



Mientras el DMA accede a memoria, el procesador no puede acceder (no puede leer la siguiente instrucción)



Mientras el DMA accede a memoria, el procesador puede seguir ejecutando instrucciones desde la caché.

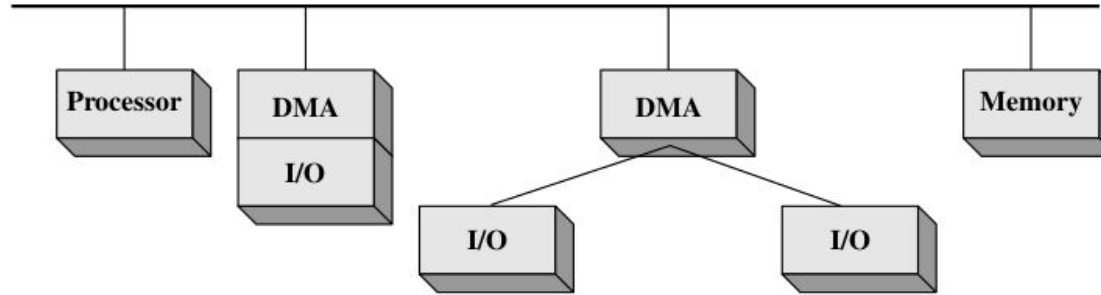


## **DMA: Consideraciones avanzadas**

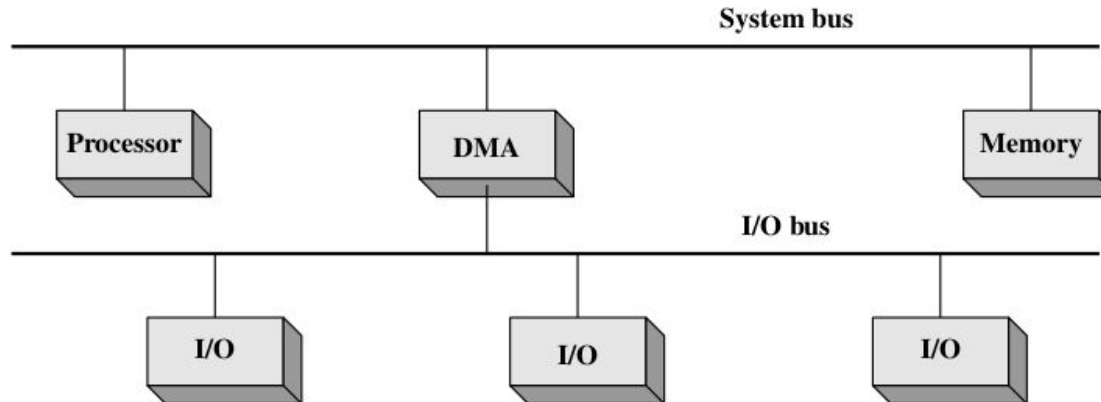
- **Puede haber varios controladores DMA en un sistema.**
- **Muchos controladores DMA tienen memoria para:**
  - **Absorber demoras en las transferencias**
  - **Cuando tiene que esperar para ser el master del bus.**
- **Modos de transferencia:**
  - **Robo de ciclos: El controlador DMA y el CPU comparten el bus en ciclos intercalados.**
  - **Burst (Ráfaga): El controlador DMA toma el control del bus hasta terminar la transferencia. **Permite transferencia por bloques.****
    - **Sin Cache (El CPU pierde ciclos)**
    - **Con Caché (En CPU no pierde ciclos). También es llamado: Modo transparente.**



# DMA (Direct Memory Access)



(b) Single-bus, integrated DMA-I/O





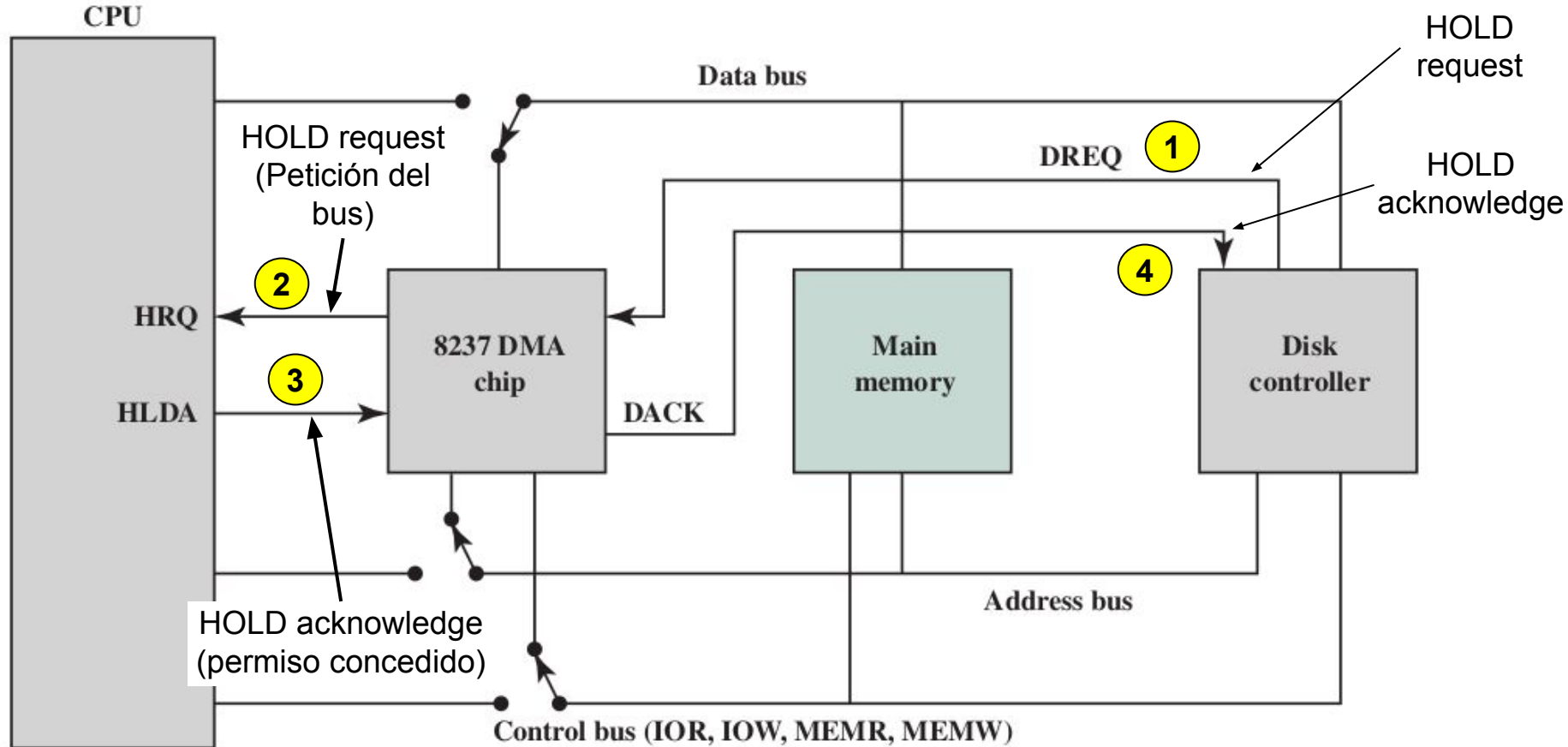


Figura obtenida de William Stallings. "Computer Organization and Architecture". 7° edición. Página 233.

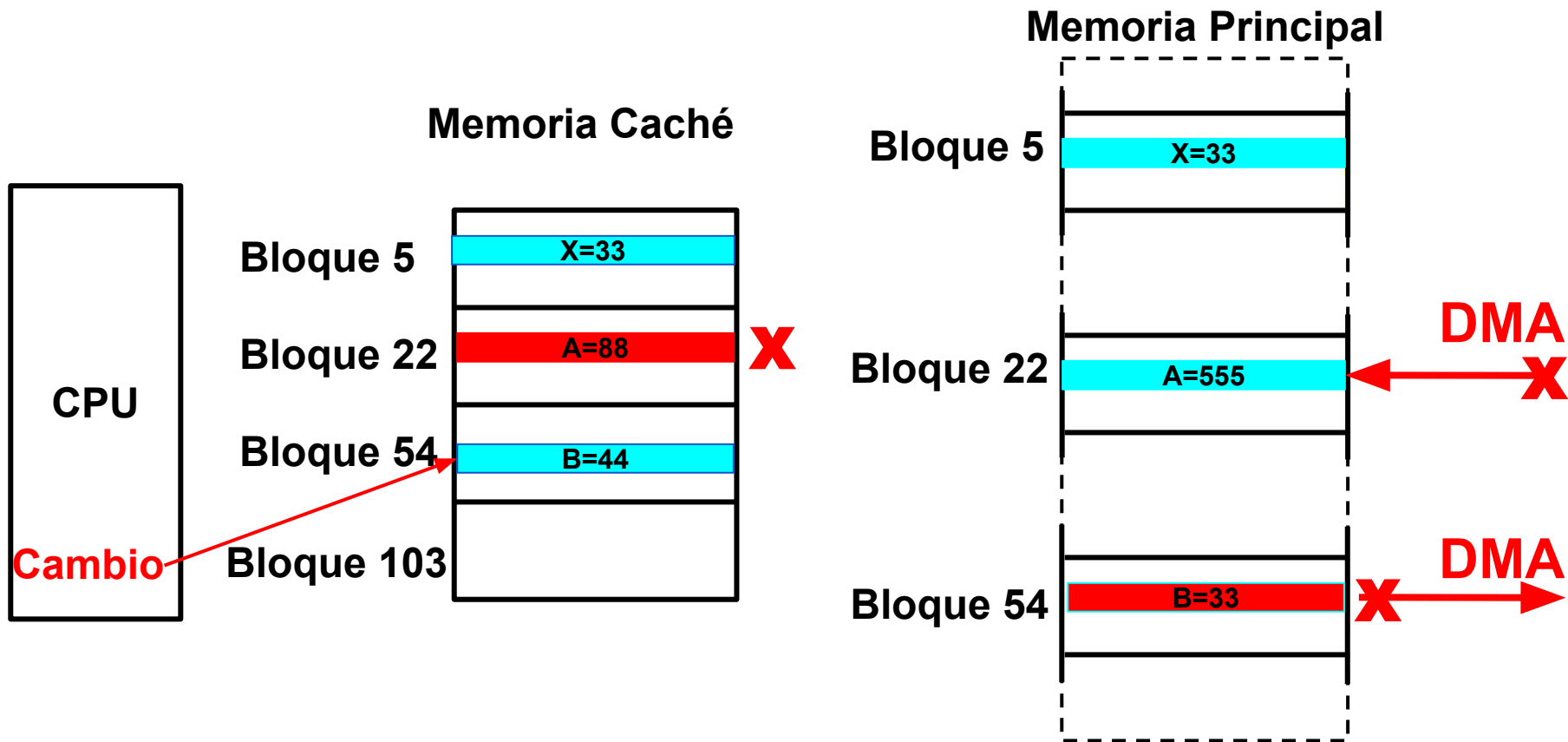


# Problema de coherencia DMA - Memoria Caché

- **El contenido de la Caché debe ser una copia exacta de una porción de la memoria RAM.**
- **El controlador de DMA puede modificar la memoria RAM, y hacer que haya diferencias con el contenido en la memoria Caché**
- **El DMA podría requerir leer un dato desde la memoria RAM, pero el procesador podría haber modificado el dato en la Caché, y el DMA leería un dato inválido**

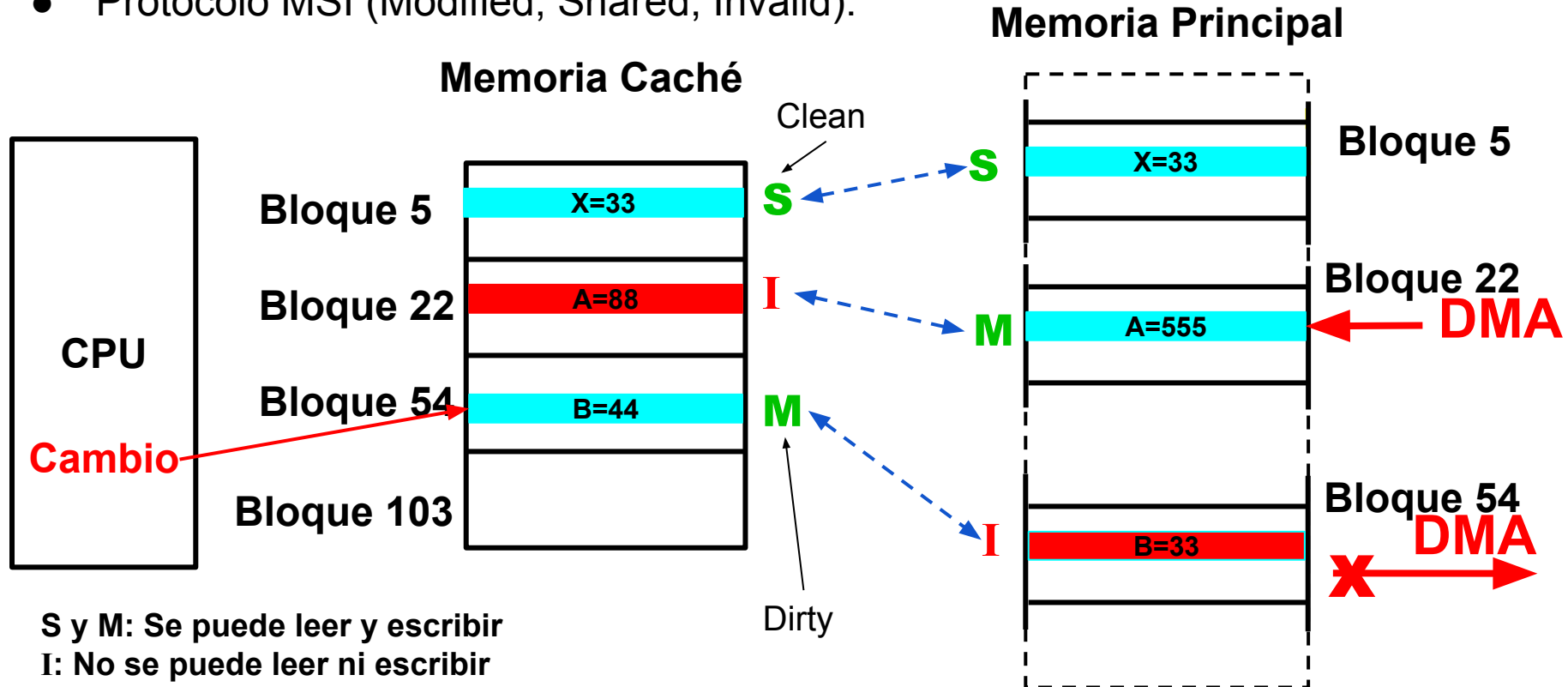


# Problema de coherencia DMA - Memoria Caché



# Algoritmos de coherencia de memoria caché

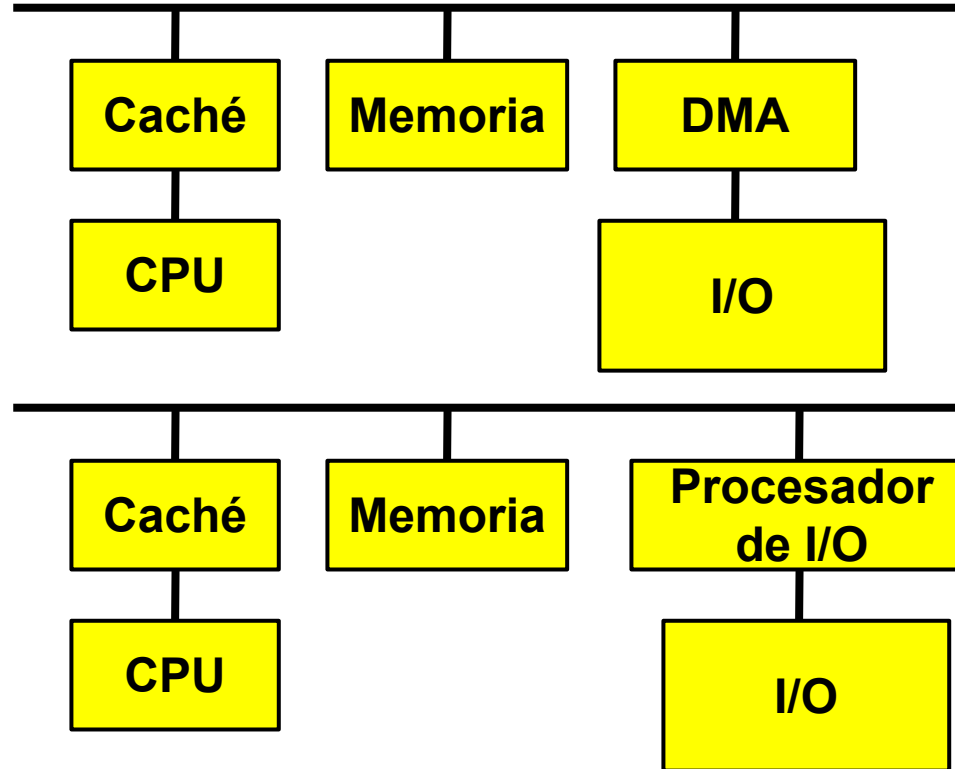
- Protocolo MSI (Modified, Shared, Invalid).





## 4) Procesador de Entrada Salida

**Procesador E/S:  
DMA + capacidad  
de ejecutar  
instrucciones**





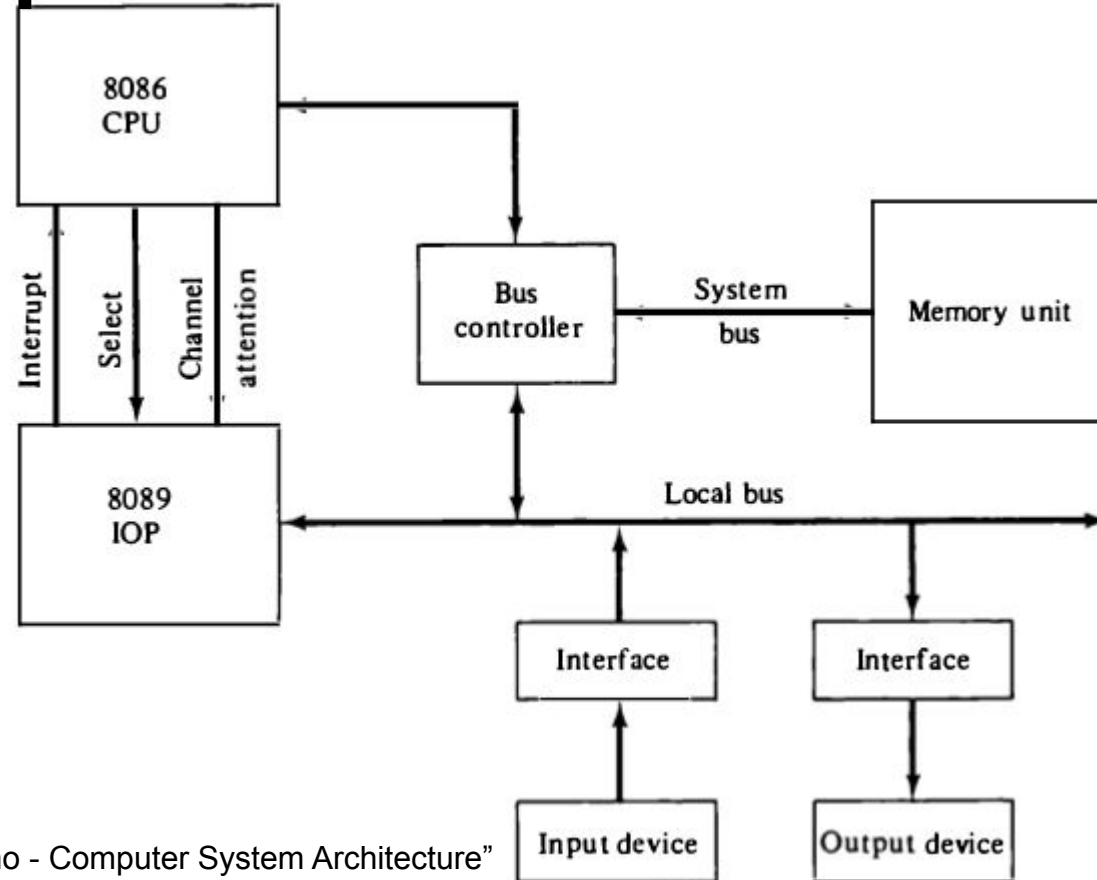
# Procesador de Entrada salida (canal de entrada salida)

## Características

- **Ejecuta instrucciones de entrada salida (libera al microprocesador de esta tarea).**
  - **Concepto de **co-procesador****
- **Puede acceder a la memoria RAM.**
- **Tiene memoria RAM interna.**
- **Ejecuta operaciones de DMA.**
- **Ejecuta rutinas de servicio de interrupciones, rutinas de iniciación, ejecución de operaciones, etc.**
- **Se fabrican especialmente para trabajar con un procesador o familia de procesadores específicos.**



# Ejemplo1: Co-Procesador E/S 8089





## Ejemplo 1: Procesador E/S 8089

- **1979**
- **Denominado Co-procesador de e/s 8089**
- **50 instrucciones**
  - **Transferencia de datos**
  - **Operaciones aritméticas y lógicas básicas**
  - **Saltos condicionales**
  - **Llamadas a subrutinas**
  - **Inicializa operaciones de DMA e interrupciones**
- **2 Núcleos virtuales**
- **Direccionamiento:**
  - **Memoria RAM: 20 bits (1 MB)**
  - **I/O: 16 bits (64 KB)**





## 5) Bus Mastering

- **Los dispositivos pueden tomar el control del bus e iniciar una transferencia con la memoria SIN intervención del procesador ni de un módulo de DMA o procesador de E/S.**
  - **Se lleva el procesador de entrada/salida a cada periférico.**
- **Dispositivo bus master: Dispositivo capaz de iniciar y controlar los buses del sistema.**
- **Un dispositivo bus mastering requiere que su interfaz posea su propio procesador, para que pueda controlar la operación sin intervención del procesador principal.**
- **Necesidad de mecanismo de arbitración.**
- **Implementado a partir del bus PCI (1992).**



# Temario

## 1. Entrada Salida - Modos de transferencia

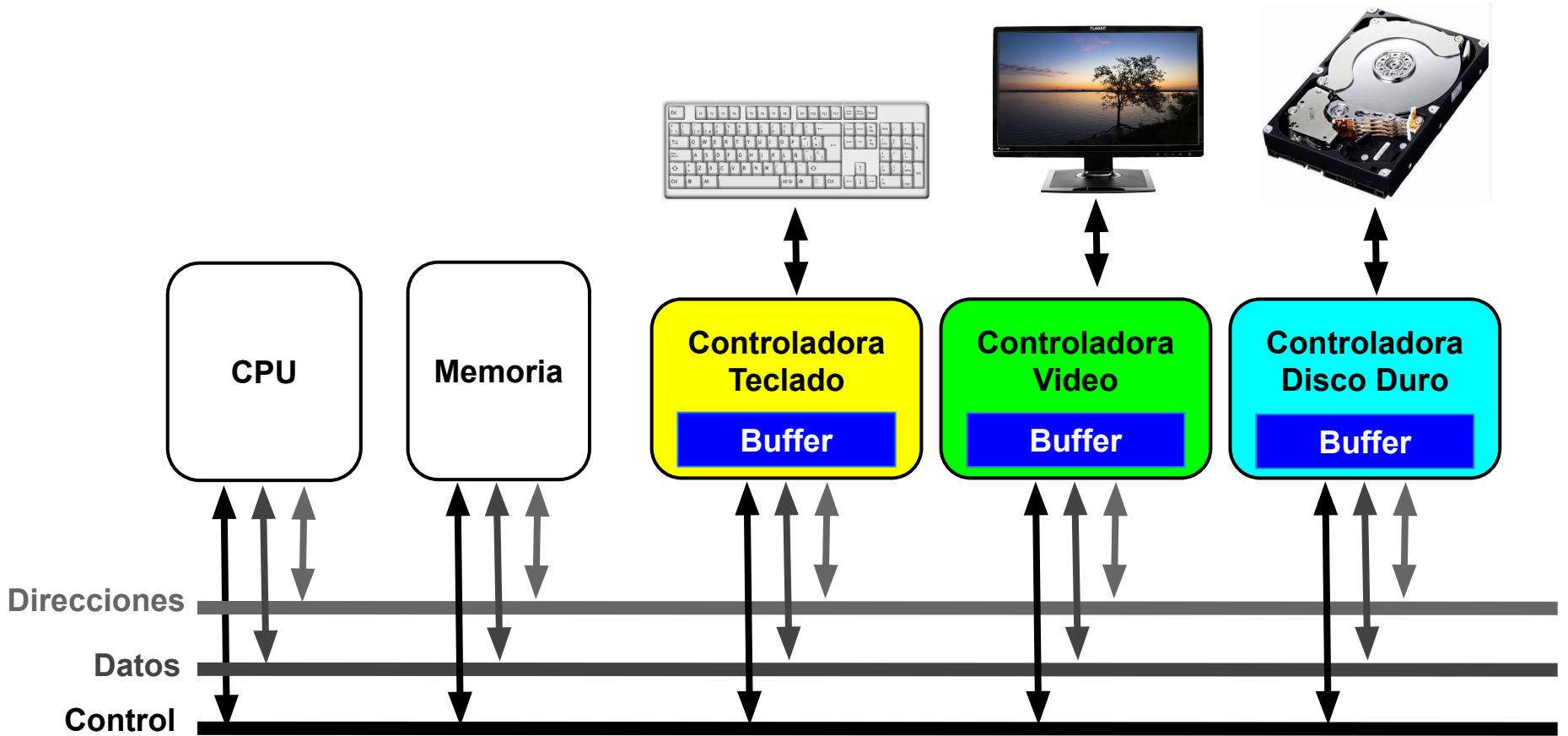
- 1.1. Microprocesador -> Dispositivos E/S
- 1.2. Dispositivos E/S -> Microprocesador

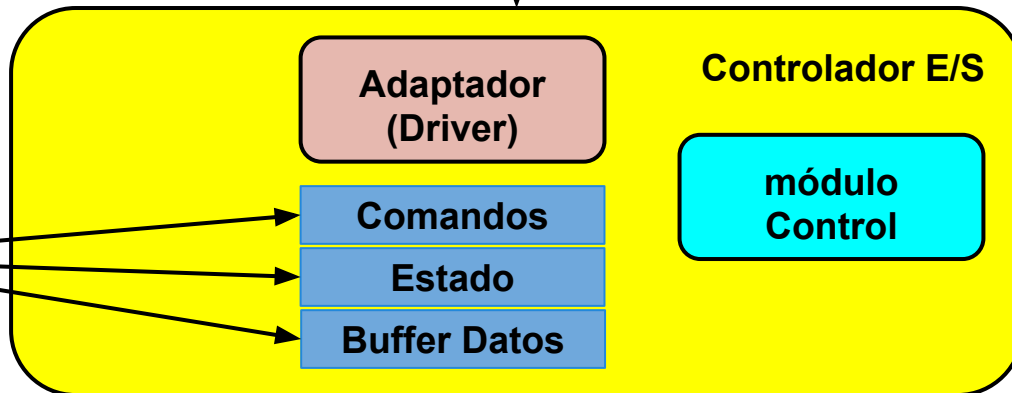
## 2. Componentes del sistema de entrada salida

- 2.1. Controladores
- 2.2. Manejador de E/S (Driver)
- 2.3. Puertos
- 2.4. Interfaces dispositivos almacenamiento masivo
- 2.5. Procesador de E/S

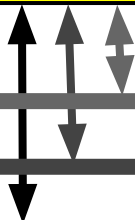
## 3. Buses de Entrada Salida

- 3.1. Tipos de buses
- 3.2. Necesidad y características
- 3.3. Bus ISA
- 3.4. BUS PCI
- 3.5. BUS PCI Express
- 3.6. BUS USB





Registros  
o Buffers



Direcciones



Datos



Control



No confundir Driver como componente de hardware de adaptación eléctrica con Driver como software de manejo del periférico



## Controladores o Interfaz de E/S

- Funciones:
  - Controlar la electrónica del periférico.
  - Controlar el acceso al bus (Interfaz entre el bus y el periférico).
    - Permite transferencia de datos
    - Genera Interrupciones
    - Genera peticiones de DMA
    - Toma control del bus (bus mastering).
- Contenido en una **placa de expansión**, en la **placa-madre** (dentro de un chip llamado **chipset** que reúne varios controladores y puentes), o dentro del procesador.

Diferentes nombres según autor:

Tanenbaum y Patricia Quiroga: Controlador de entrada/salida

Morris Nano: Interfaz de entrada/salida

William Stallings: Módulo de entrada/salida

- Módulo de E/S con poca capacidad de procesamiento: Controlador de E/S
- Módulo de E/S con elevada capacidad de procesamiento: Procesador de E/S





# Software manejador de dispositivo (Driver)

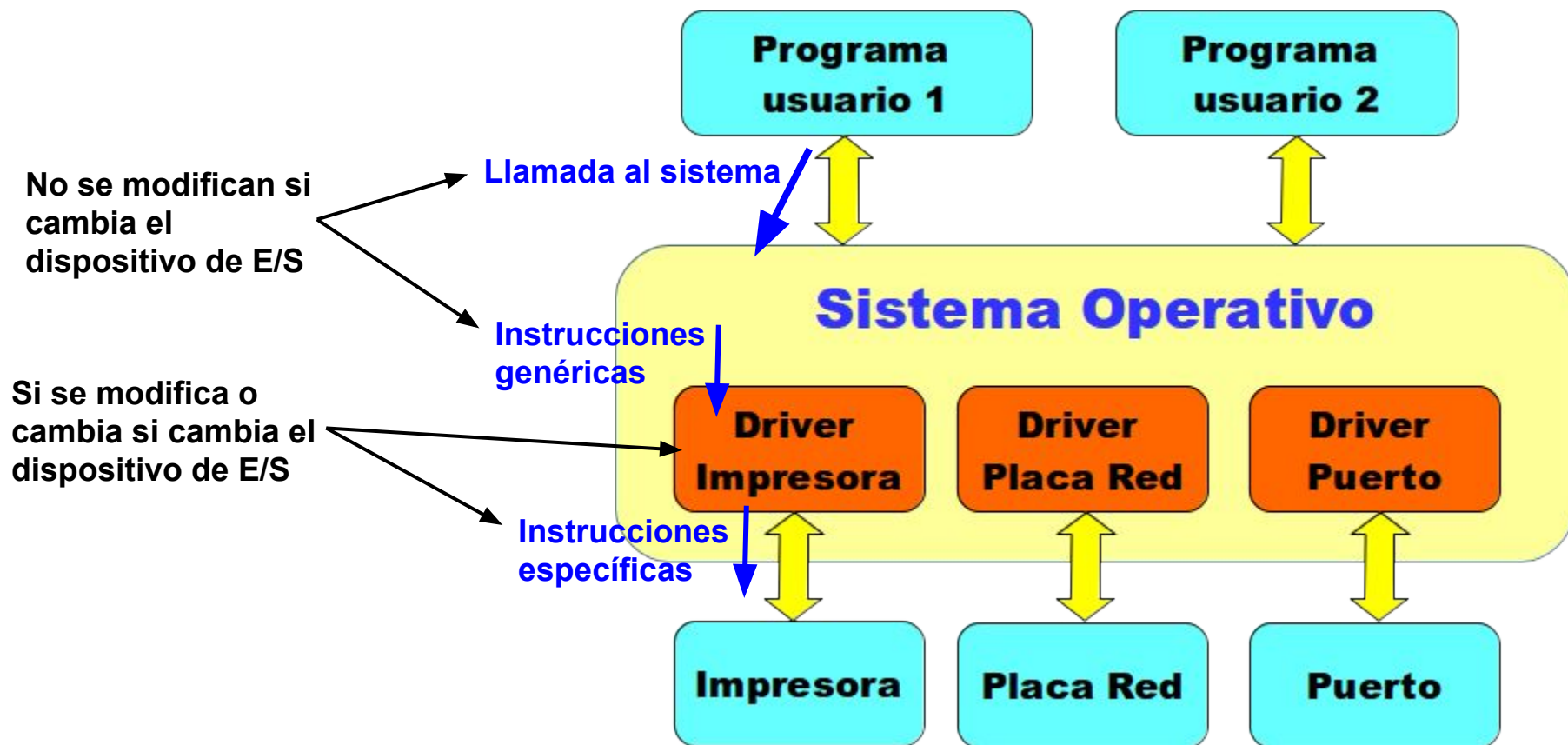
**Software que controla un dispositivo de hardware**

**Problema: Distintos fabricantes diseñan sus dispositivos de hardware de manera diferente: Distintos comandos, estados, secuencias temporales, rutinas de servicio, etc.**

- **Cada dispositivo de E/S particular tiene su driver, que traduce instrucciones genéricas (propias de cada sistema operativo) a las instrucciones particulares de cada dispositivo de E/S**
- **Desarrollado por:**
  - **Fabricante del dispositivo**
  - **Diseñadores de sistemas operativos**
  - **Colaboración entre ambos**



## Comunicación con un Driver





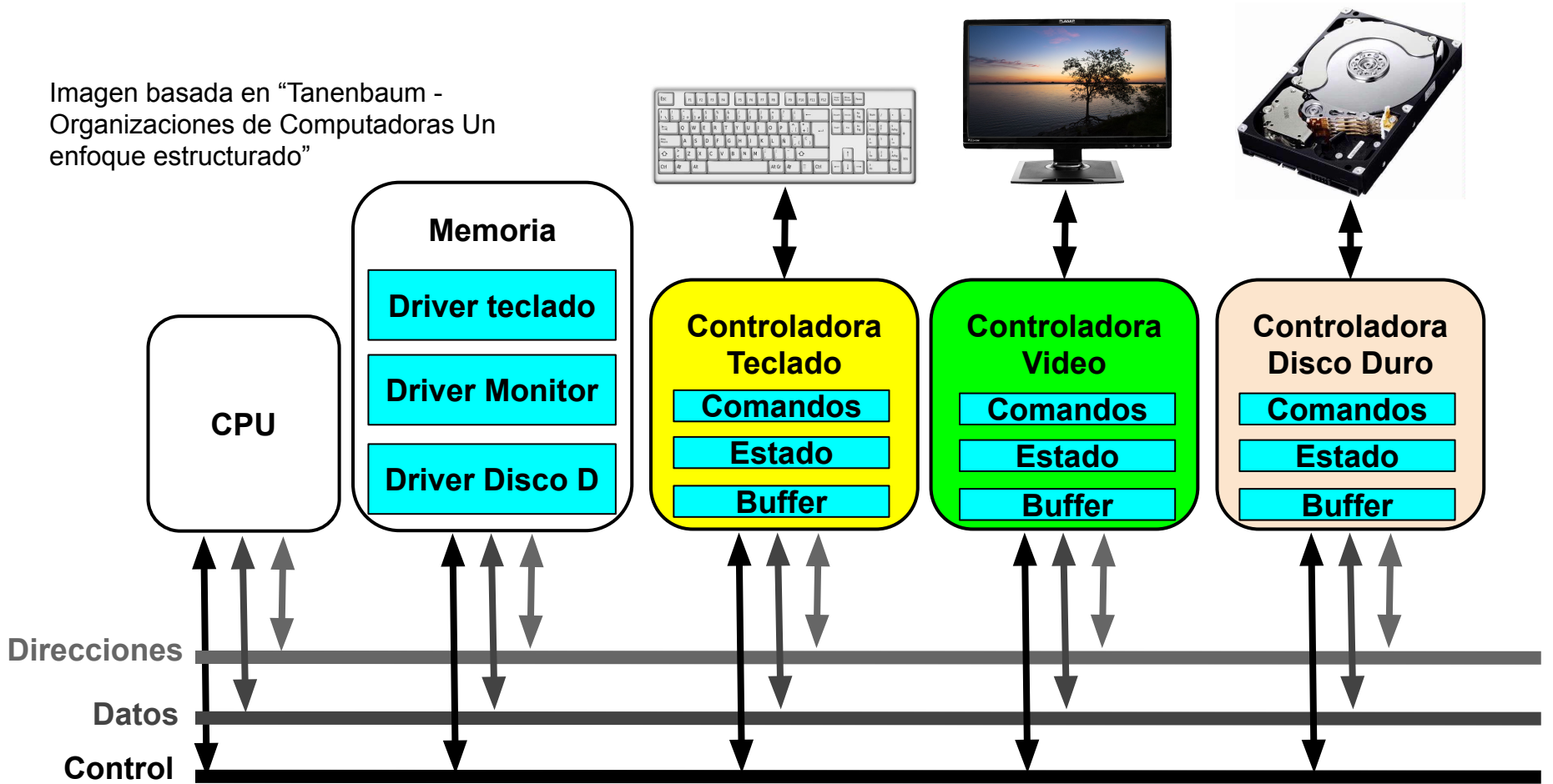
## **Manejador de dispositivo (Driver)**

### **Tareas que realiza:**

- **Enviar comandos al periférico.**
- **Verificar los registros de estado.**
- **Respetar la secuencia de tiempos de cada tarea.**
- **Inicializar el periférico.**
- **Verificar errores y ejecutar rutinas de solución de problemas.**
- **Contiene rutinas de servicio de interrupciones.**
- **Controlar DMA.**



Imagen basada en “Tanenbaum -  
Organizaciones de Computadoras Un  
enfoque estructurado”





# Manejador de dispositivo (Driver)

## Consideraciones adicionales:

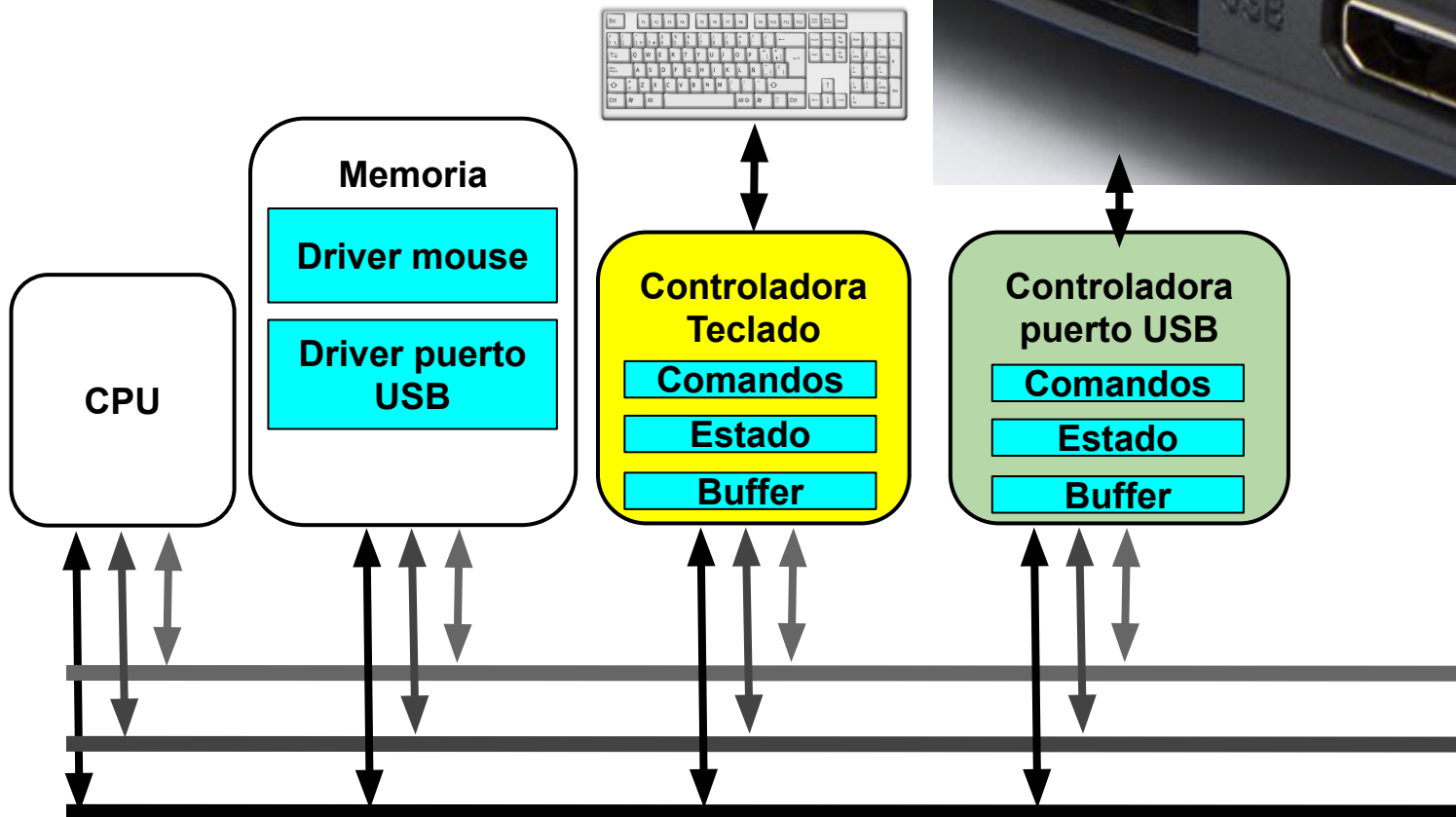
- Realiza una **abstracción** del dispositivo
- Es invocado por el sistema operativo
- Específico para **cada hardware y Sistema Operativo**
- Parte fundamental de un Sistema Operativo



## Puertos/Interfaces de entrada salida

- **Dispositivo de E/S** destinado a **comunicaciones** con dispositivos de E/S.
- **Ejemplos:**
  - **USB, HDMI, etc.**
  - **Interfaces con dispositivos de almacenamiento masivo.**
- **Características**
  - **Estandarizan:**
    - **Conector (o familia de conectores).**
    - **Protocolo de comunicaciones.**
  - **Puede ser de propósito específico (PS/2 teclado) o general (puerto serie, puerto USB).**
  - **Arquitectura abierta:** Permite **ampliar** la computadora a nuevos tipos de periféricos

Tanenbaum los llama I/O Chips  
William Stallings: External interface



# Interfaces para dispositivos de almacenamiento masivo (discos duros, CDs, Floppy disks)

- **Interfaz de E/S (puerto) interno para conectar dispositivos de almacenamiento masivo. (Controlador + conector + buses)**
- **2 estándares:**
  - **IDE (Integrated Drive Electronics) o ATA (Advanced Technology Attachment):**
    - Creado por Western Digital (WD)
    - Toshiba, Seagate, HP, Kingston, Samsung, etc.
    - Es una interfaz específica para dispositivos de almacenamiento masivo.
  - **SCSI (Small Computer System Interface) o “scuzzy”:**
    - American National Standards Institute (ANSI)
    - Seagate, HP, Dell, Compaq.
    - Interfaz de propósito general, usado hoy para dispositivos de almacenamiento masivo.

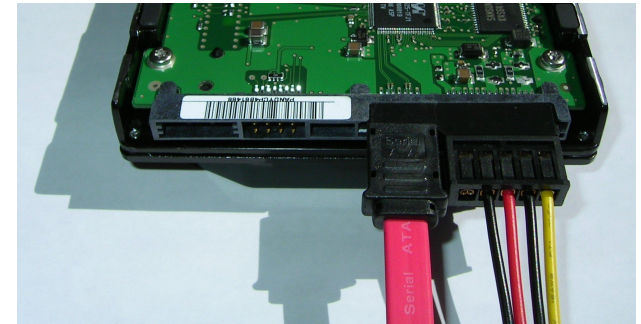
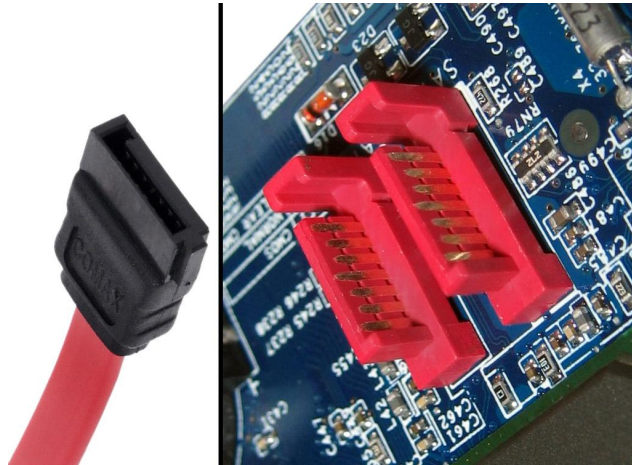
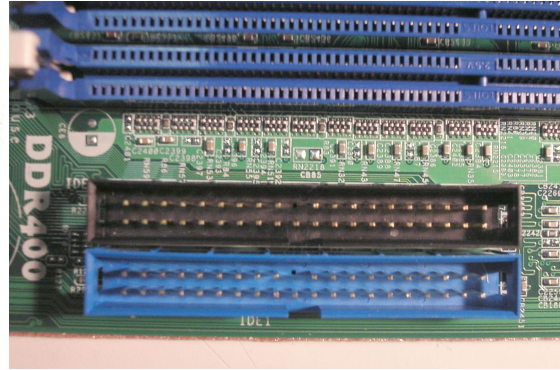
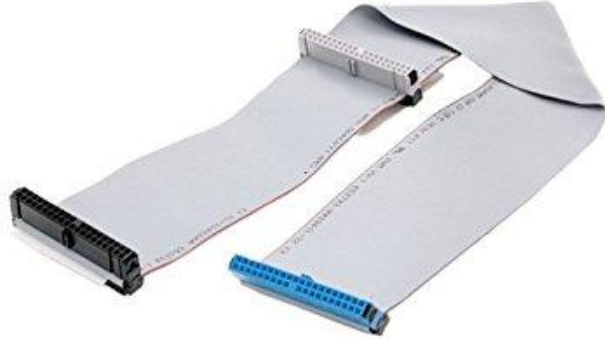


# Evolución de las Interfaz ATA y SCSI

	IDE/ATA	SCSI
<b>Paralelo</b>	<ul style="list-style-type: none"><li>• <b>ATA (133 MB/s)</b></li><li>• <b>ATAPI (ATA Packet Interface)</b></li><li>• <b>EIDE (Extended IDE)</b></li><li>• <b>UATA (Ultra ATA)</b></li></ul>	<ul style="list-style-type: none"><li>• <b>SCSI (SPI: SCSI Parallel Interface)</b></li><li>• <b>SCSI 1.0, 2.0 - 5.0</b></li></ul>
<b>Serie</b>	<ul style="list-style-type: none"><li>• <b>SATA (Serial ATA) (SATA 4.0, 2017, 16 Gb/s)</b></li><li>• <b>mSATA (mini SATA)</b></li><li>• <b>eSATA (external SATA)</b></li></ul>	<ul style="list-style-type: none"><li>• <b>SAS (Serial Attached SCSI) (SAS 4.0, 2017, 22.5 Gb/s)</b></li></ul>
<b>Sobre otros protocolos</b>	<ul style="list-style-type: none"><li>• <b>SATA Express, M.2, U.2 (PCI-Express con SATA) (32 Gb/s), SATA IP</b></li></ul>	<ul style="list-style-type: none"><li>• <b>UAS ( USB Attached SCSI), SCSI Express (SCSI sobre PCIe)</b></li></ul>



# Interfaces IDE/ATA





# Interfaces SCSI



© 2006 HowStuffWorks

Conector SFF-8482







# Temario

## 1. Entrada Salida - Modos de transferencia

- 1.1. Microprocesador -> Dispositivos E/S
- 1.2. Dispositivos E/S -> Microprocesador

## 2. Componentes del sistema de entrada salida

- 2.1. Controladores
- 2.2. Manejador de E/S (Driver)
- 2.3. Puertos
- 2.4. Interfaces dispositivos almacenamiento masivo

## 3. Buses de Entrada Salida

- 3.1. Tipos de buses
- 3.2. Necesidad y características
- 3.3. Bus ISA
- 3.4. BUS PCI
- 3.5. BUS PCI Express
- 3.6. BUS USB



# Tipos de Buses

- **Internos del procesador**

- **Comunicación con la ALU, entre núcleos de un micro, cachés internas, registros, puentes internos, etc.**
- **Muy rápidos y muy cortos.**
- **Diferentes** para distintos procesador (a medida).
- **Ejemplos:**
  - **AMBA (ARM)**
  - **Backside bus (Intel)**
  - **AMD HyperTransport (HT)**



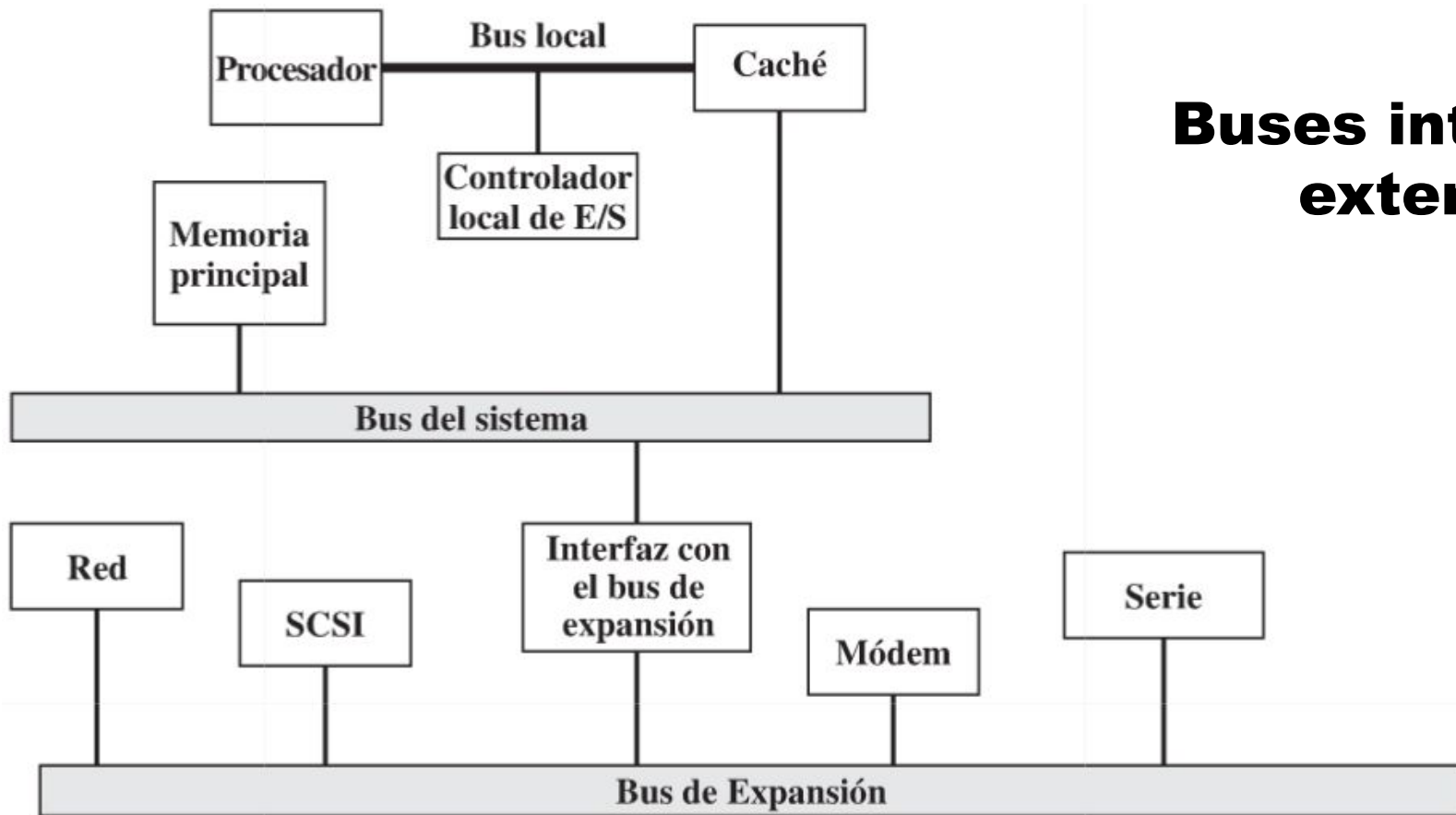
## Tipos de Buses

- **Externos al procesador Propósito específico**
  - **Ejemplos:**
    - **Bus de memoria.**
    - **Conexión Procesador-Northbridge (puente norte)**
      - **Intel QuickPath Interconnect (QPI), 25.6 GB/s**
      - **Intel Front-side bus (FSB), 8GB/s**
      - **AMD HyperTransport (HT), 25.6 GB/s**
    - **Conexión Northbridge-Southbridge**
      - **Intel DMI, 8GT/s. (similar a PCIe).**
      - **AMD A-Link Express (PCIe).**
  - **Muy rápidos y cortos.**
  - **Diferentes** para distintos micros (a medida).



## Tipos de Buses

- **Externos al procesador de entrada salida (también llamados de expansión)**
  - **Comunicación con dispositivos I/O**
  - **Más largo y lentos.**
  - **Distintos tipos de periféricos.**
  - **Estandarizados** para permitir la conexión de dispositivos externos de otros fabricantes.
  - **Propiedad de consorcios**
  - **Ejemplos:**
    - **Internos a la computadora:**
      - **PCI Express**
      - **PCI**
      - **ISA**
    - **Externos a la computadora:**
      - **USB**

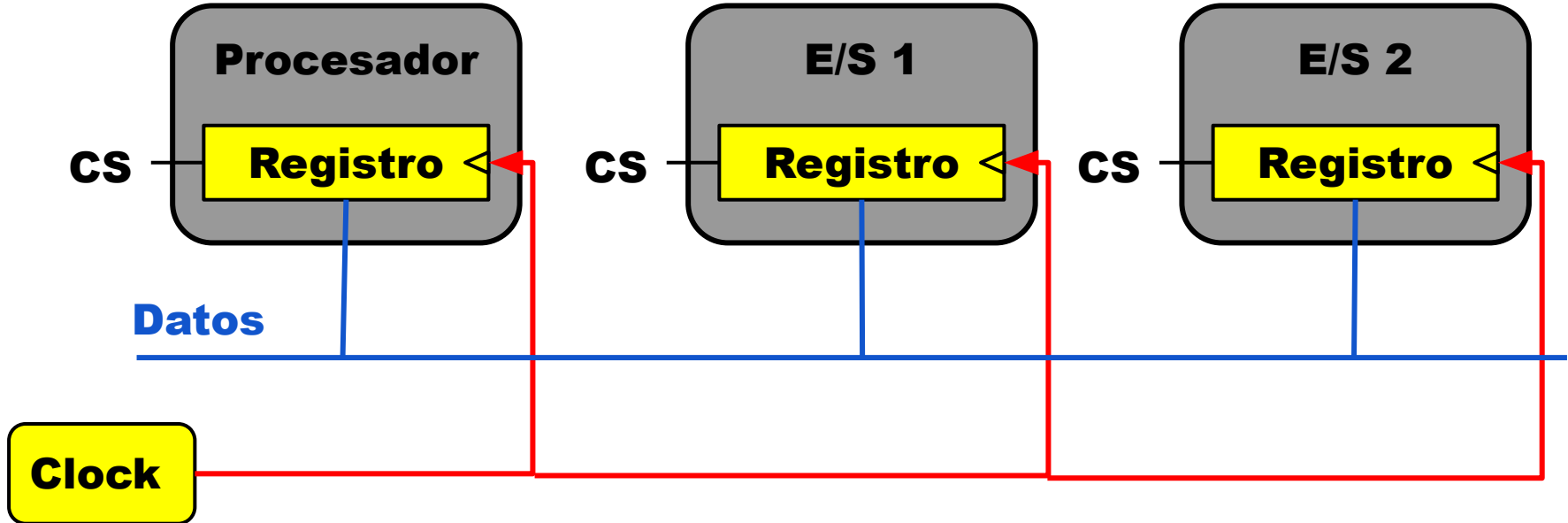


## Buses internos y externos



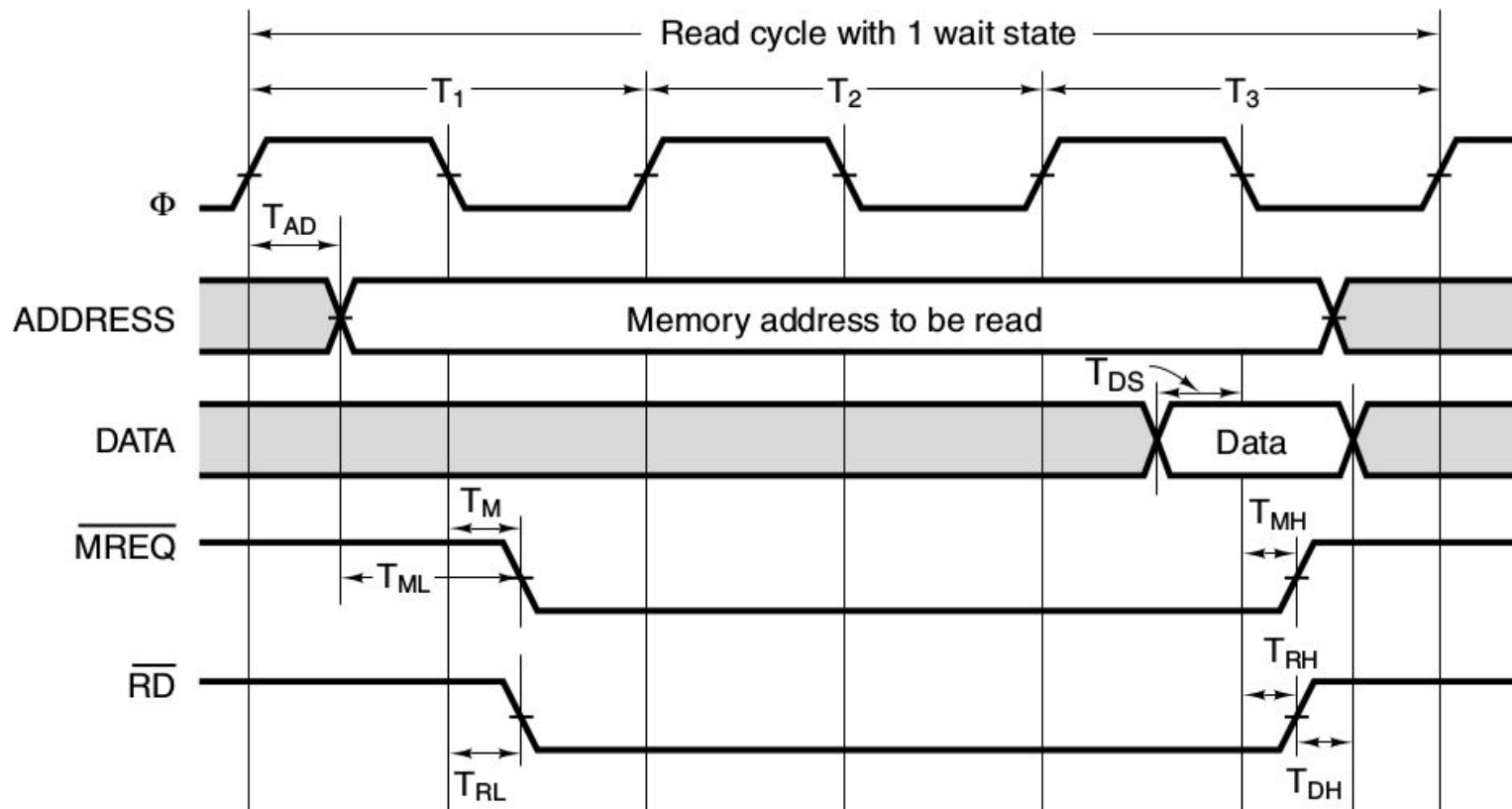
# Transferencia entre registros síncrona

Transferencia síncrona: **Mismo Clock**  
No hay necesidad de **Control de flujo**.





# Bus síncrono (hay clock)

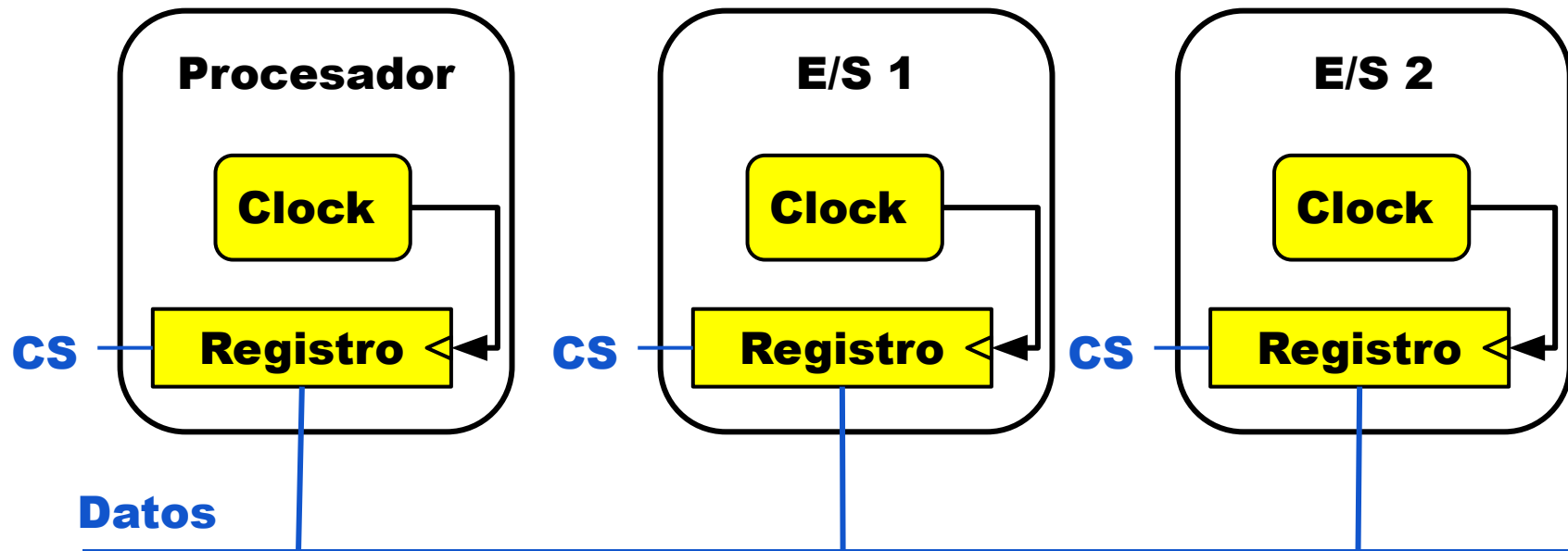




# Transferencia entre registros Asíncrona

Transferencia asíncrona: **SIN Clock común**

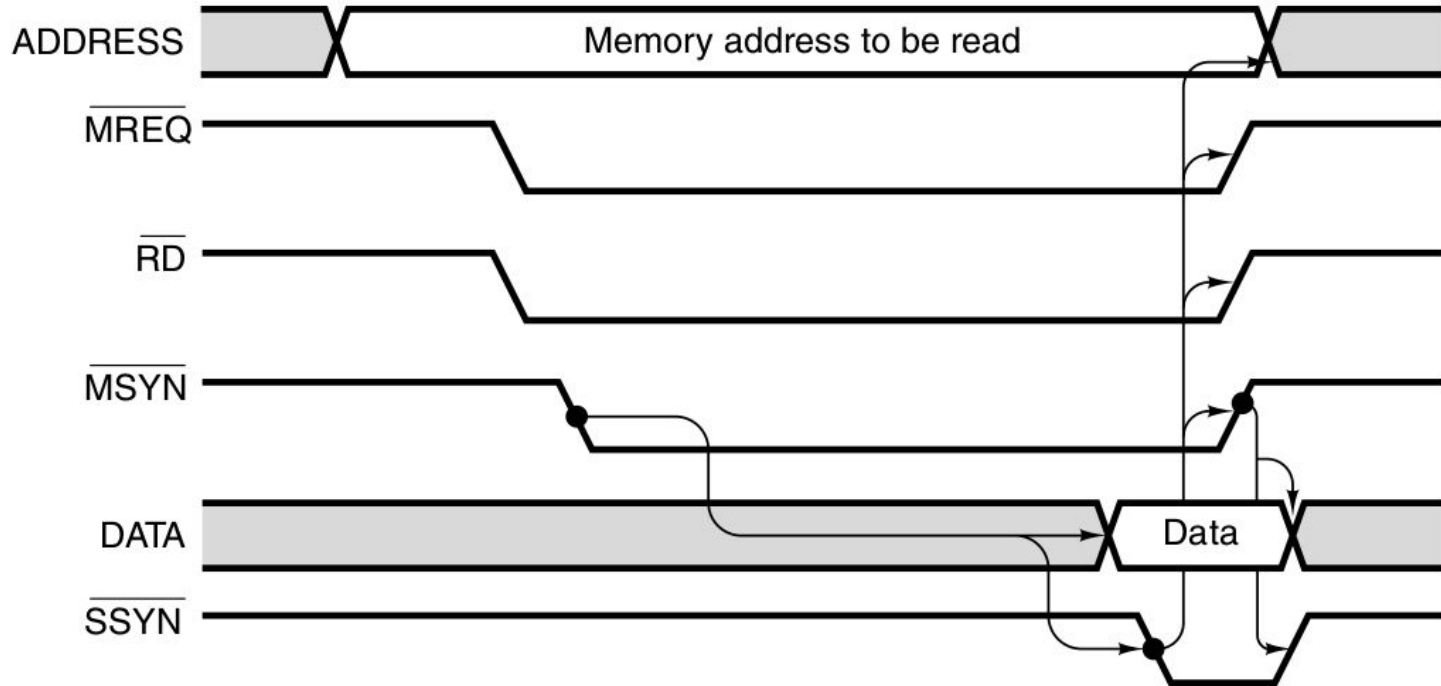
Necesidad de mecanismo de **Control de flujo (handshake)**







# Bus asíncrono (**NO** hay clock)



**MSYN: Sincronismo del maestro (Procesador).**

**SSYN: Sincronismo del esclavo (Memoria o dispositivo de E/S).**



# Transferencia entre registros síncrona

- **Ventajas**
  - **Menor costo**
  - **No requiere handshake**
- **Desventajas:**
  - **Pérdidas de tiempo (al tener que esperar el clock)**
  - **Dificultad para adaptarse a avances tecnológicos de componentes individuales**
  - **La velocidad queda limitada por el componente más lento**
    - **Solución Intermedia: Distintos buses.**

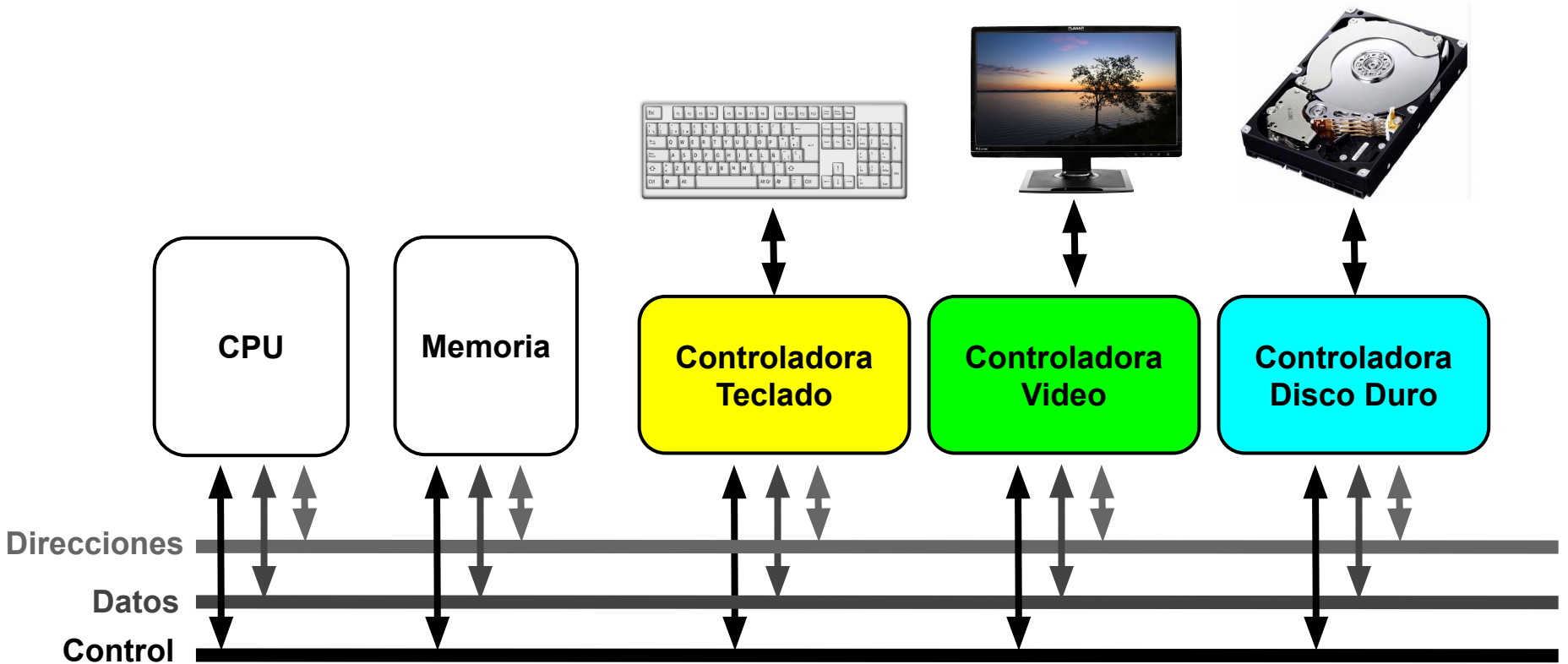
# Transferencia entre registros asíncrona

- **Ventajas**

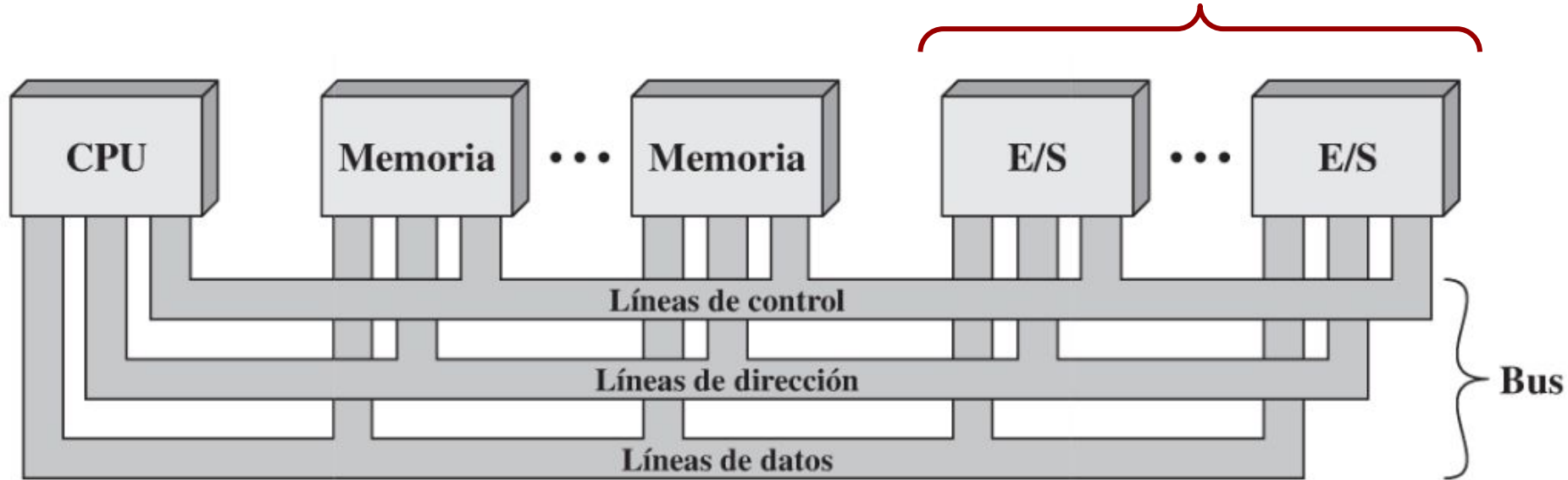
- **No hay pérdidas de tiempo por tener que “esperar el clock”**
- **Facilidad para adaptarse a avances tecnológicos de componentes individuales**
- **Cada periférico puede funcionar a su velocidad máxima**
- **Pueden convivir dispositivos rápidos y lentos**

- **Desventajas:**

- **Mayor complejidad**
  - **Necesidad de mecanismo de handshaking complejo**
- **Mayor costo.**



**Cada vez más dispositivos**

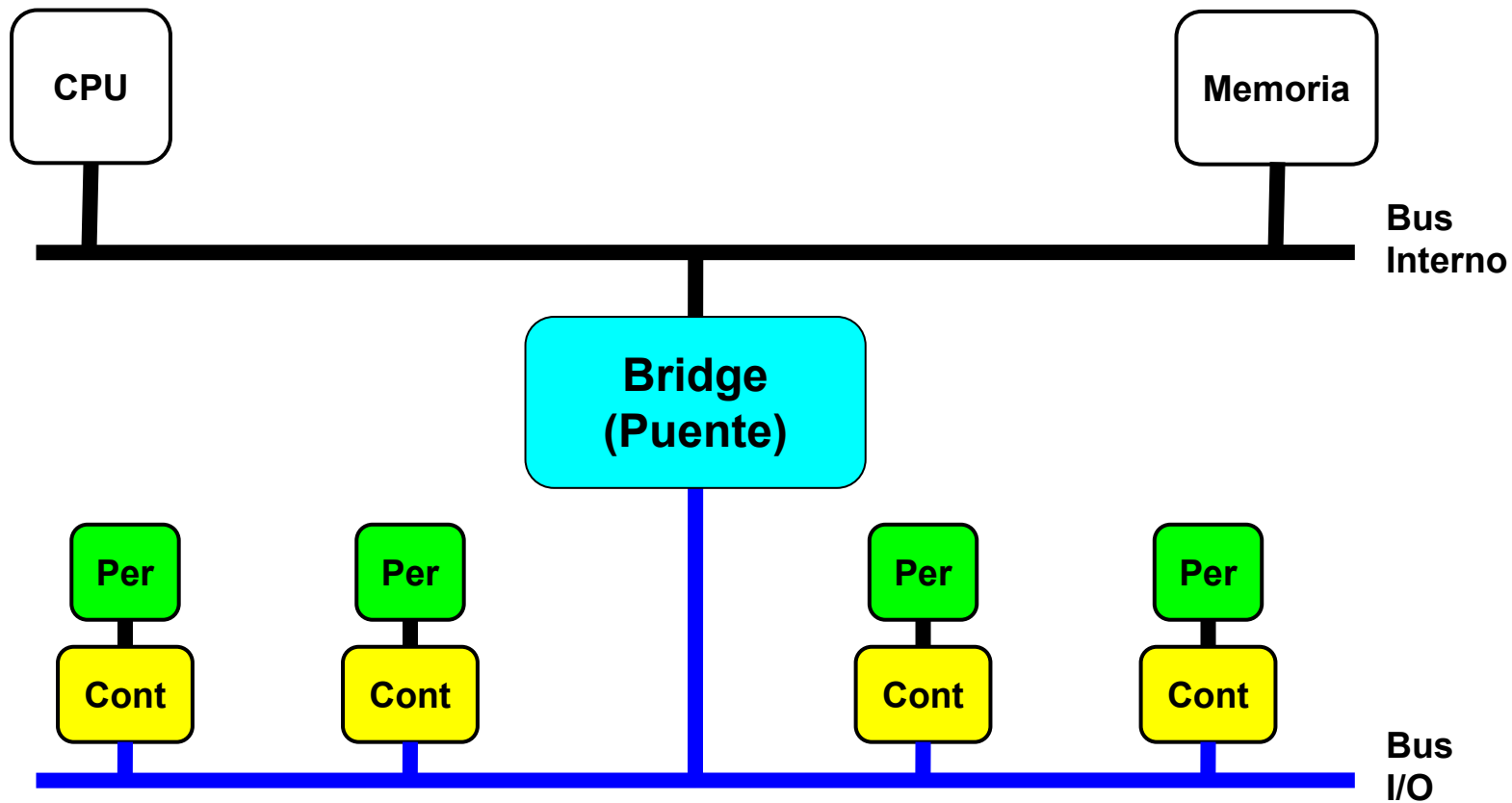


## Problema 1:

### Cada vez más y más periféricos...

- **Los periféricos ocupan el bus -> “Roban ciclos al CPU” (cycle stealing\*) (mientras un periférico ocupa el bus, el CPU no puede acceder al bus para buscar instrucciones en la memoria principal)**
- **Deterioro de características eléctricas**

\* Tanenbaum, computer-organization-and-architecture



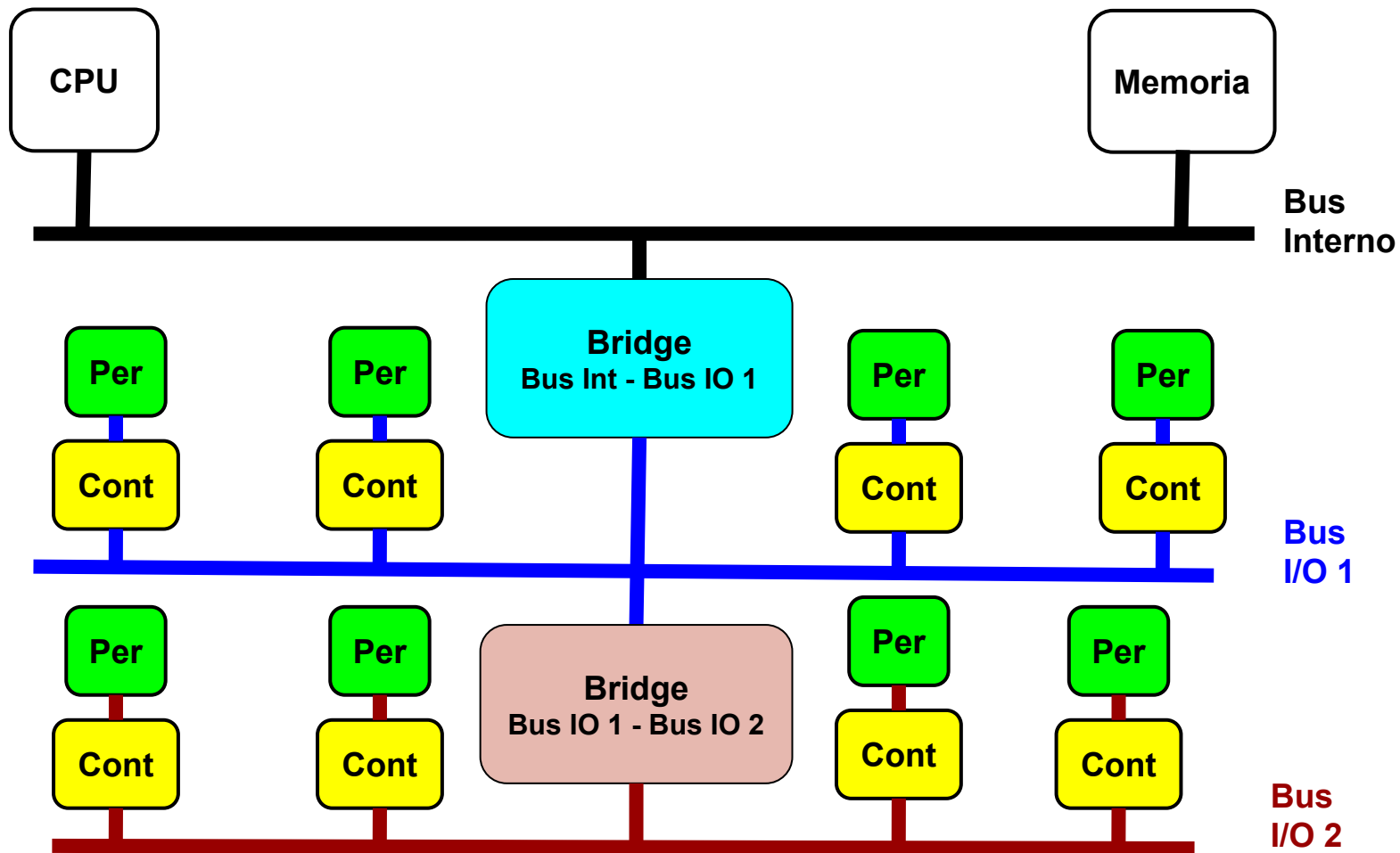


## Problema 2:

**Cada vez más y más periféricos... diferentes...**

- **Diferentes velocidades**
- **Necesidad de compatibilidad con periféricos anteriores**








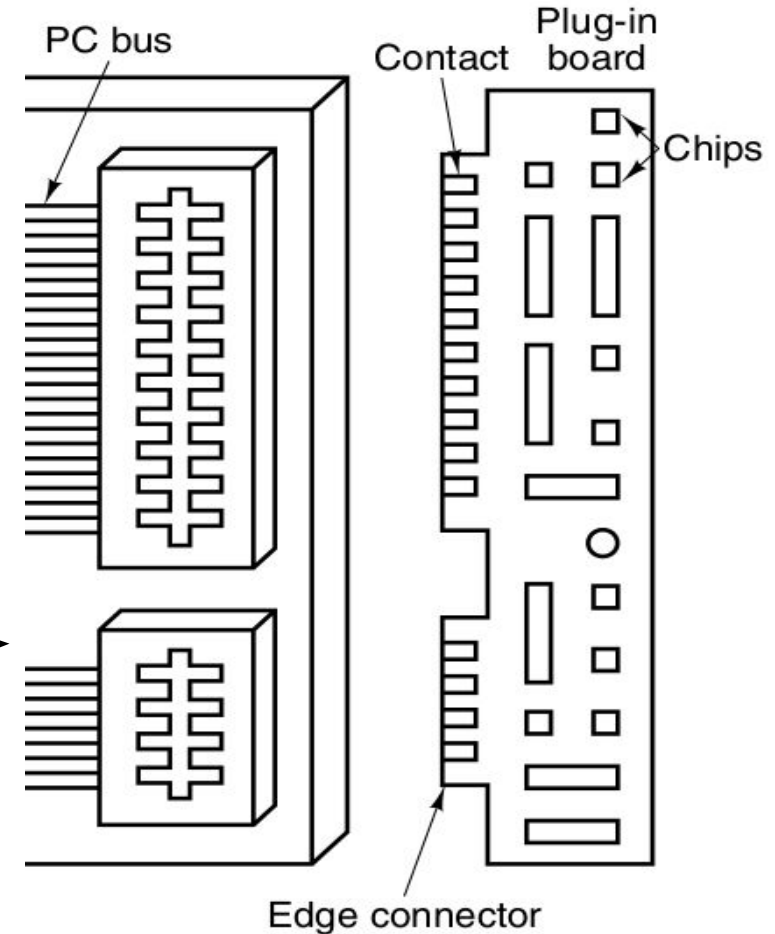
## **Bus ISA (Industry Standard Architecture)**

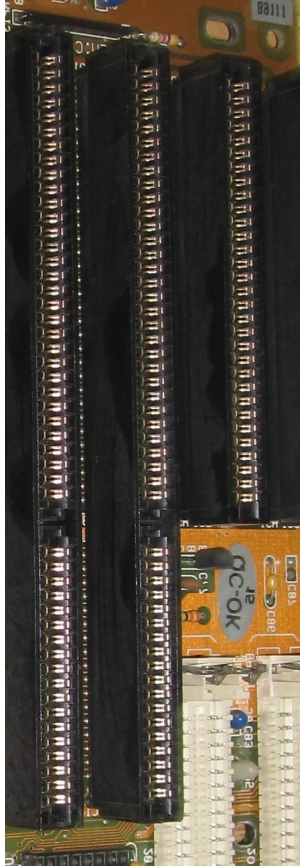
- **Creado por IBM (1981), estandarizado e impuesto por el mercado**
- **Inicialmente llamado IBM PC bus**
- **Características técnicas:**
  - **Bus paralelo**
  - **8 - 16 bits datos**
  - **20 - 24 bits de direcciones (1 MB, 16 MB)**
  - **4.77 - 8 MHz. 2 bytes por ciclo (16.7 MB/sec).**
  - **Soporta interrupciones y DMA**



## Compatibilidad 8 bits - 16 bits

- **Existe un bus IBM de 8 bits compatible con muchos periféricos.**
- **IBM saca al mercado la 80286 con bus PC/AT de 16 bits.**
- **Existían muchas placas compatibles con el bus ISA de 8 bits.**
- **Solución:** 







Habilitación  
direccionamiento

Habilitación  
bus 16 bits

A 1	<u>I/O CHCK</u>
A 2	Data 7
A 3	Data 6
A 4	Data 5
A 5	Data 4
A 6	Data 3
A 7	Data 2
A 8	Data 1
A 9	Data 0
A 10	<u>I/O CH RDY</u>
A 11	AEN
A 12	Addr 19
A 13	Addr 18
A 14	Addr 17
A 15	Addr 16
A 16	Addr 15
A 17	Addr 14
A 18	Addr 13
A 19	Addr 12
A 20	Addr 11
A 21	Addr 10
A 22	Addr 9
A 23	Addr 8
A 24	Addr 7
A 25	Addr 6
A 26	Addr 5
A 27	Addr 4
A 28	Addr 3
A 29	Addr 2
A 30	Addr 1
A 31	Addr 0

C 1	SBHE
C 2	Addr 23
C 3	Addr 22
C 4	Addr 21
C 5	Addr 20
C 6	Addr 19
C 7	Addr 18
C 8	Addr 17
C 9	<u>MEMR</u>
C 10	<u>MEMW</u>
C 11	Data 8
C 12	Data 9
C 13	Data 10
C 14	Data 11
C 15	Data 12
C 16	Data 13
C 17	Data 14
C 18	Data 15

GND	B 1
RESET DRV	B 2
+5V	B 3
IRQ 9	B 4
-5V	B 5
DRQ 2	B 6
-12V	B 7
Reserved, NC	B 8
+12V	B 9
GND	B 10
<u>MEMW</u>	B 11
<u>MEMR</u>	B 12
<u>IOW</u>	B 13
<u>IOR</u>	B 14
<u>DACK 3</u>	B 15
DRQ 3	B 16
<u>DACK 1</u>	B 17
DRQ 1	B 18
<u>REFRESH</u>	B 19
CLK	B 20
IRQ 7	B 21
IRQ 6	B 22
IRQ 5	B 23
IRQ 4	B 24
IRQ 3	B 25
<u>DACK 2</u>	B 26
T/C	B 27
ALE	B 28
+5V	B 29
OSC	B 30
GND	B 31

<u>MEM CS 16</u>	D 1
<u>I/O CS 16</u>	D 2
IRQ 10	D 3
IRQ 11	D 4
IRQ 12	D 5
IRQ 15	D 6
IRQ 14	D 7
<u>DACK 0</u>	D 8
DRQ 0	D 9
<u>DACK 5</u>	D 10
DRQ 5	D 11
<u>DACK 6</u>	D 12
DRQ 6	D 13
<u>DACK 7</u>	D 14
DRQ 7	D 15
+5V	D 16
<u>MASTER</u>	D 17
GND	D 18

Escritura

lectura

DRQ: DMA Request

DACK: ACK de los DMA



## Nacimiento del estándar ISA

- **Problema para IBM:**
  - **Muchos periféricos IBM compatibles no fabricados por IBM.**
  - **Muchos clones de la IBM PC (IBM PC AT).**
- **IBM Creó la IBM PS/2 de 16 bits no IBM compatible, (fue un fracaso).**
- **Los competidores reaccionan y crean el estándar ISA.**



## **Evolución del estándar ISA**

**Problema: Necesidad de mayor velocidad**

### **Sucesores del BUS ISA:**

- **EISA (Extended ISA), 32 bits, 33.3 MB/sec (80386)**
- **Bus VESA (Video Electronics Standards Association)**



# Bus **PCI** (Peripheral Component Interconnect)

- Creado por **Intel** (1992)
- Intel hace públicas las patentes.
- Hace “públicas” las especificaciones de todas las versiones del bus **PCI**.
- Forma el consorcio **PCI Special Interest Group** (Agilent, AMD, Dell, HP, Intel, Synopsys, NVIDIA, and Qualcomm) <https://pcisig.com/><sup>1</sup>.
- **3600 Compañías** fabrican productos **PCI** (2018)

<sup>1</sup> Especificaciones no gratuitas. Sección noticias con mucha información

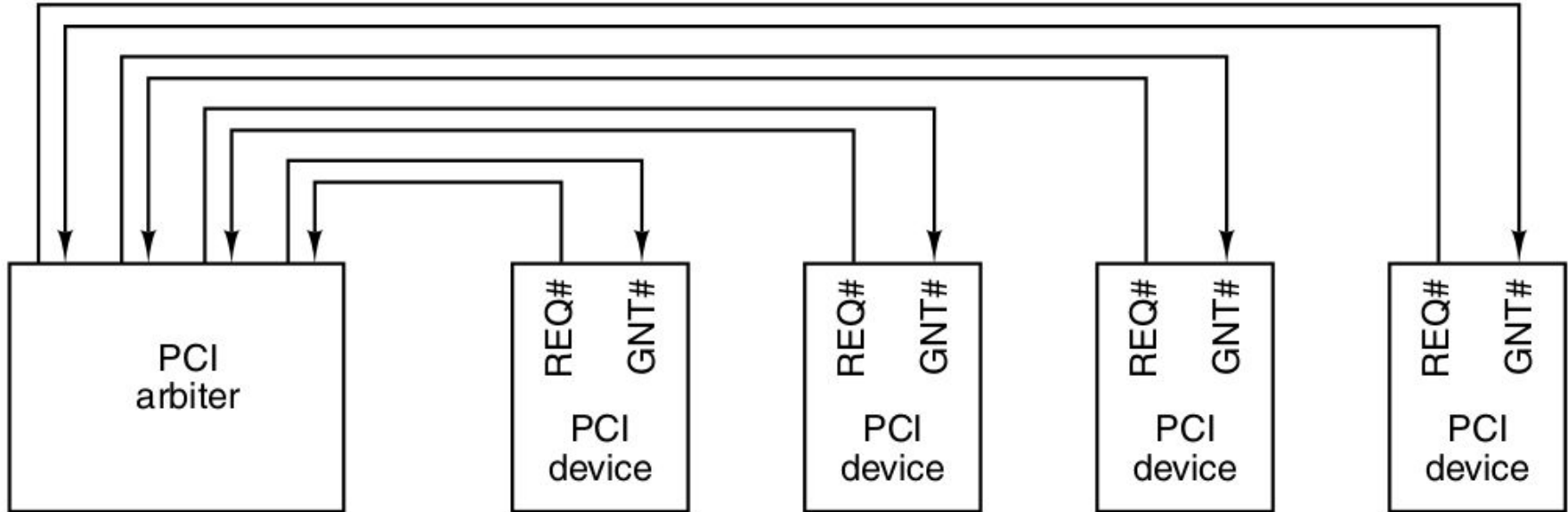




## Bus PCI características

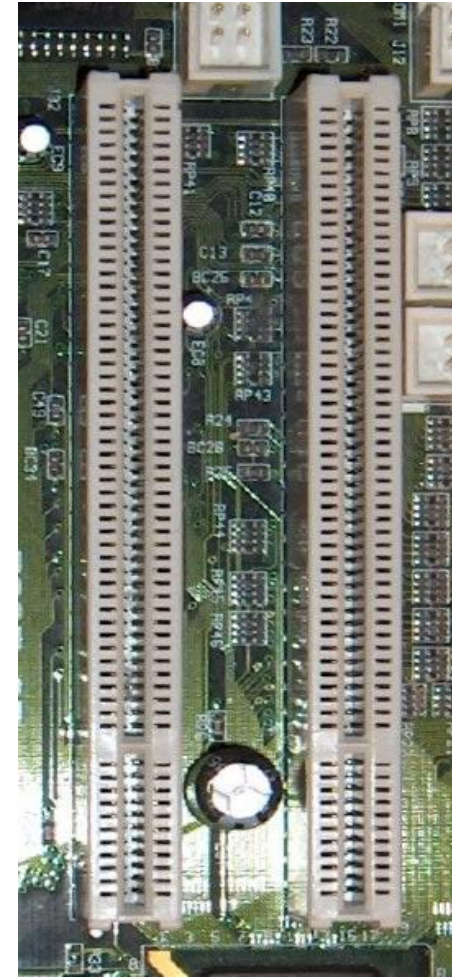
- **32 bits direcciones**
- **32 - 64 bits datos**
- **12 - 28 bit control**
- **33 - 66 MHz.**
- **Bus paralelo**
- **Compatibilidad con bus ISA a través de puentes.**
- **Bus mastering -> Necesidad de arbitración.**

## Arbitración centralizada del bus



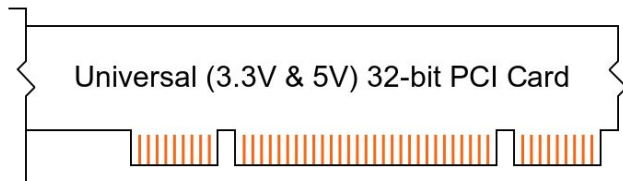
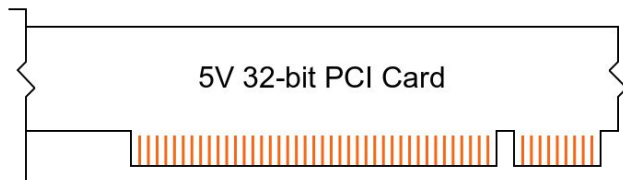
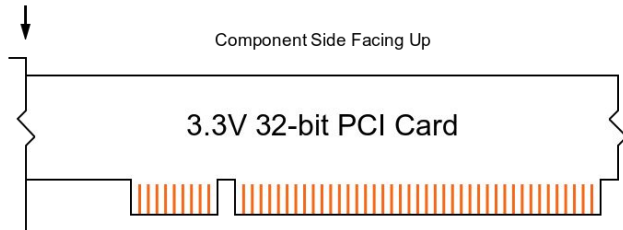


# Placa y ranuras PCI

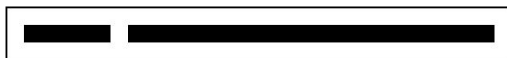




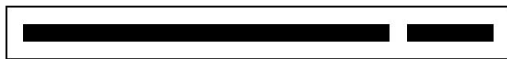
PCI Bracket



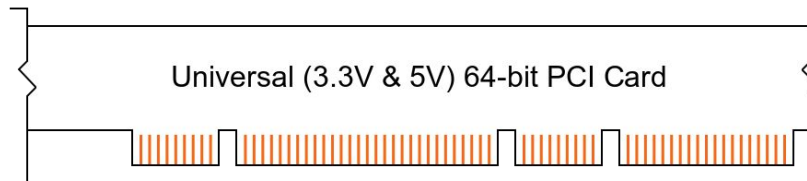
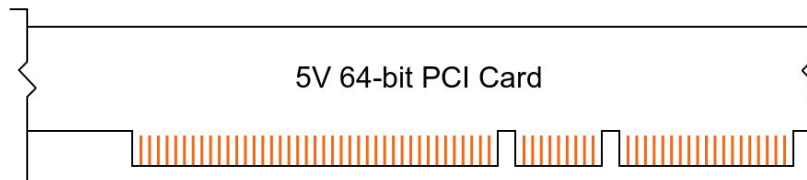
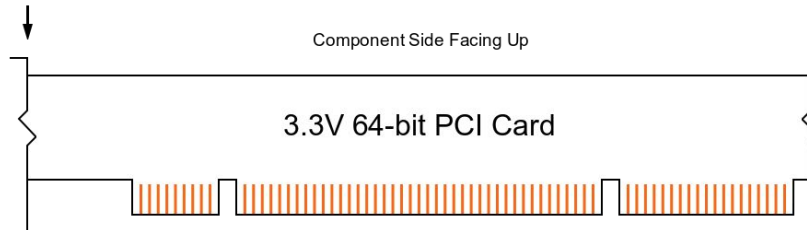
3.3V 32-bit PCI Slot



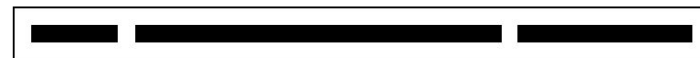
5V 32-bit PCI Slot



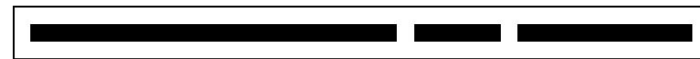
PCI Bracket

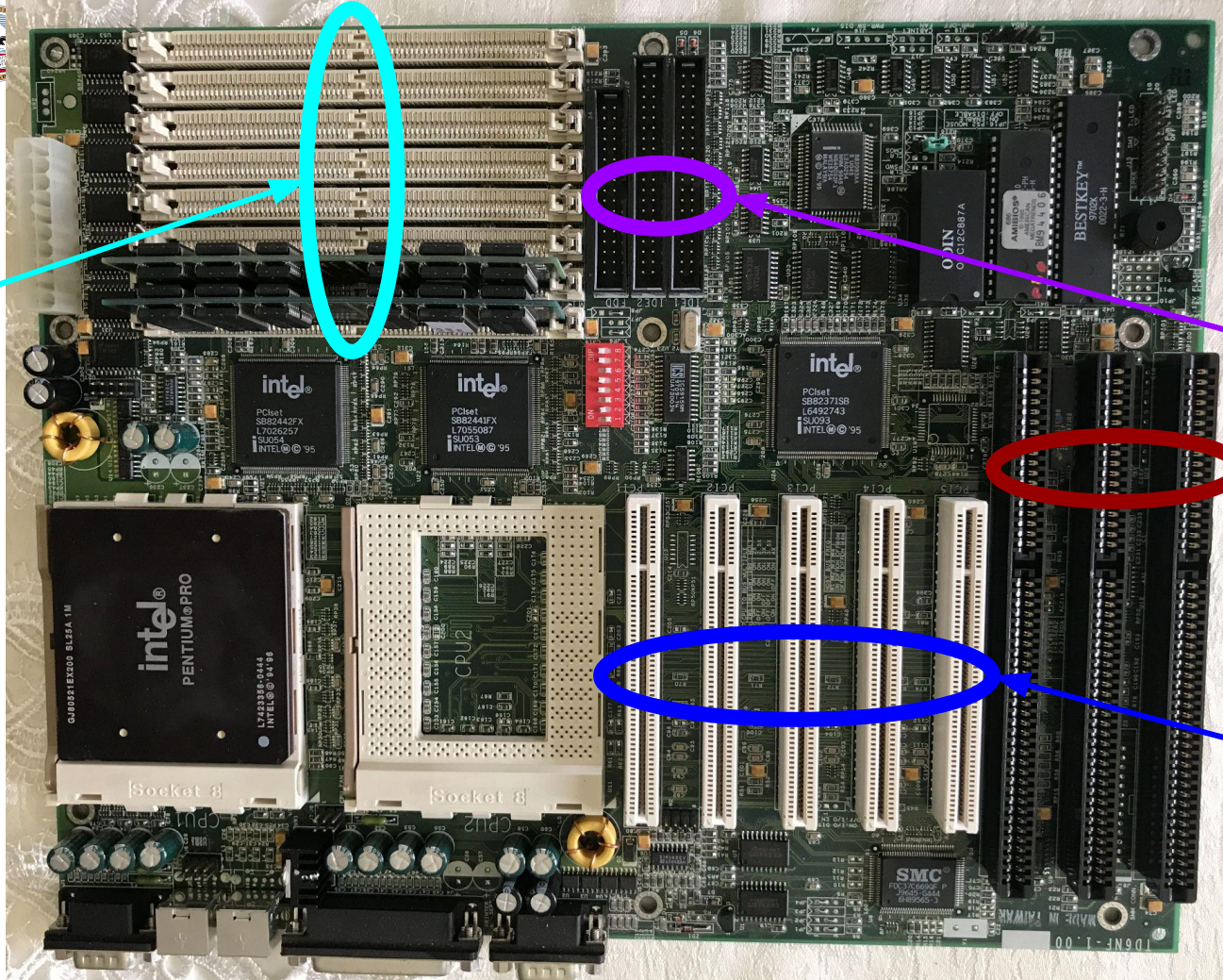


3.3V 64-bit PCI Slot



5V 64-bit PCI Slot





RAM

Motherboard  
Pentium 2  
(1997)

Interfaz  
IDE

Ranuras  
expansión  
ISA

Ranuras  
expansión  
PCI



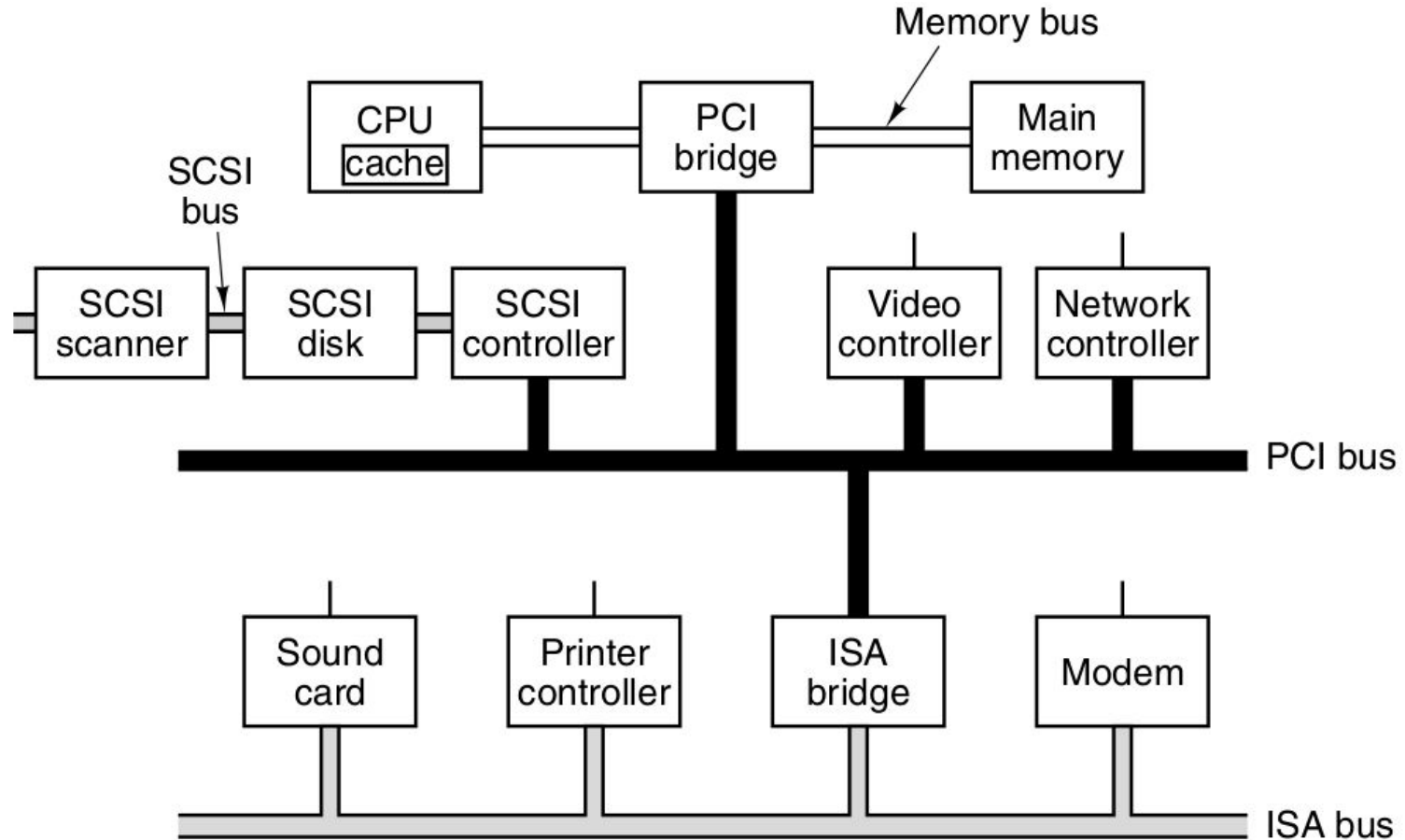
## **Evolución del bus PCI**

- **Bus PCI: 1.0, 2.0, 2.1, 2.2, 2.3, 3.0**
- **PCI-X (PCI eXtended):**
  - **Bus paralelo**
  - **Pensado para servidores**
  - **133 MHz**
  - **Compatible con PCI**

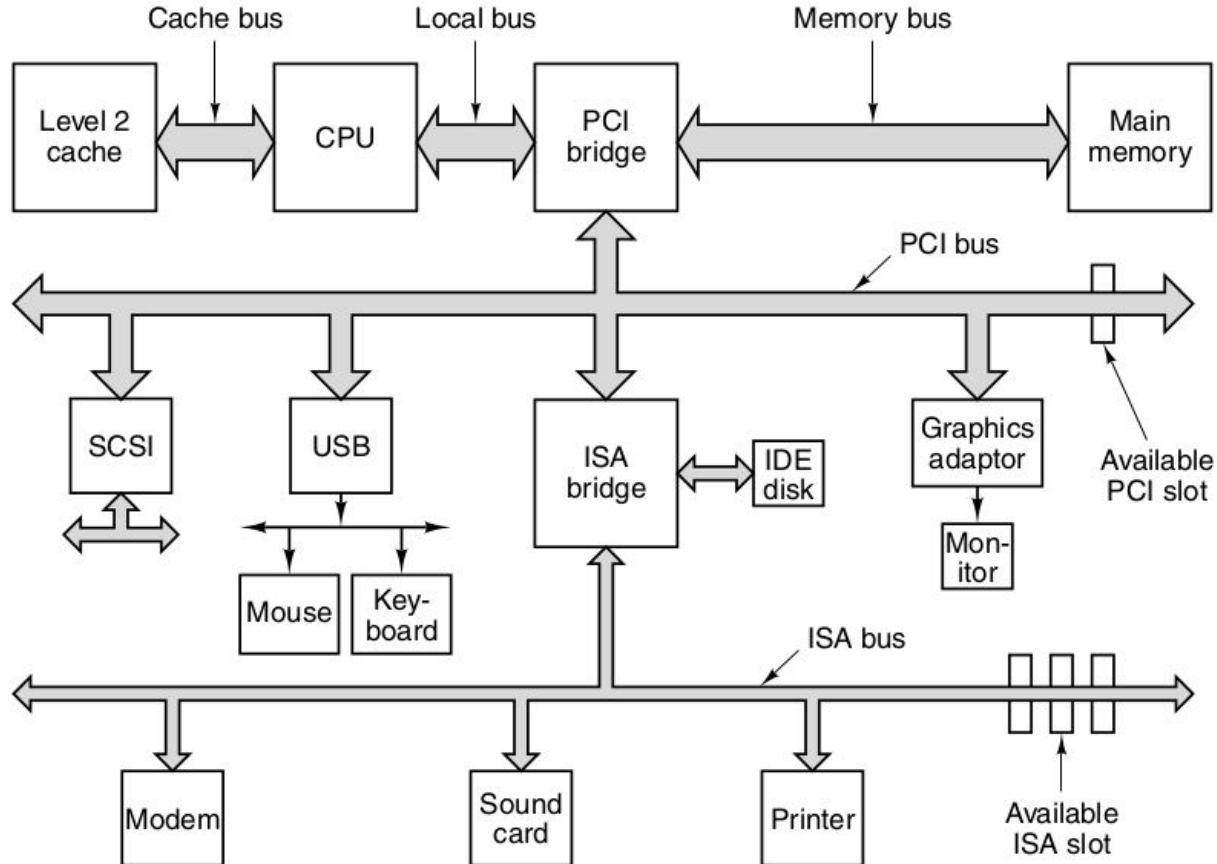


## **Bus AGP (Accelerated Graphics Port bus)**

- **El bus PCI (antes del PCI Express) no ofrecía suficiente velocidad para placas de video por la cantidad de dispositivos PCI existentes.**
- **2004 - 2008**
- **Creado por Intel**
- **Bus paralelo**
- **Versiones:**
  - **AGP 1.0 264 MB/s**
  - **AGP 2.0 1 GB/s**
  - **AGP 3.0 2.1 GB/s.**

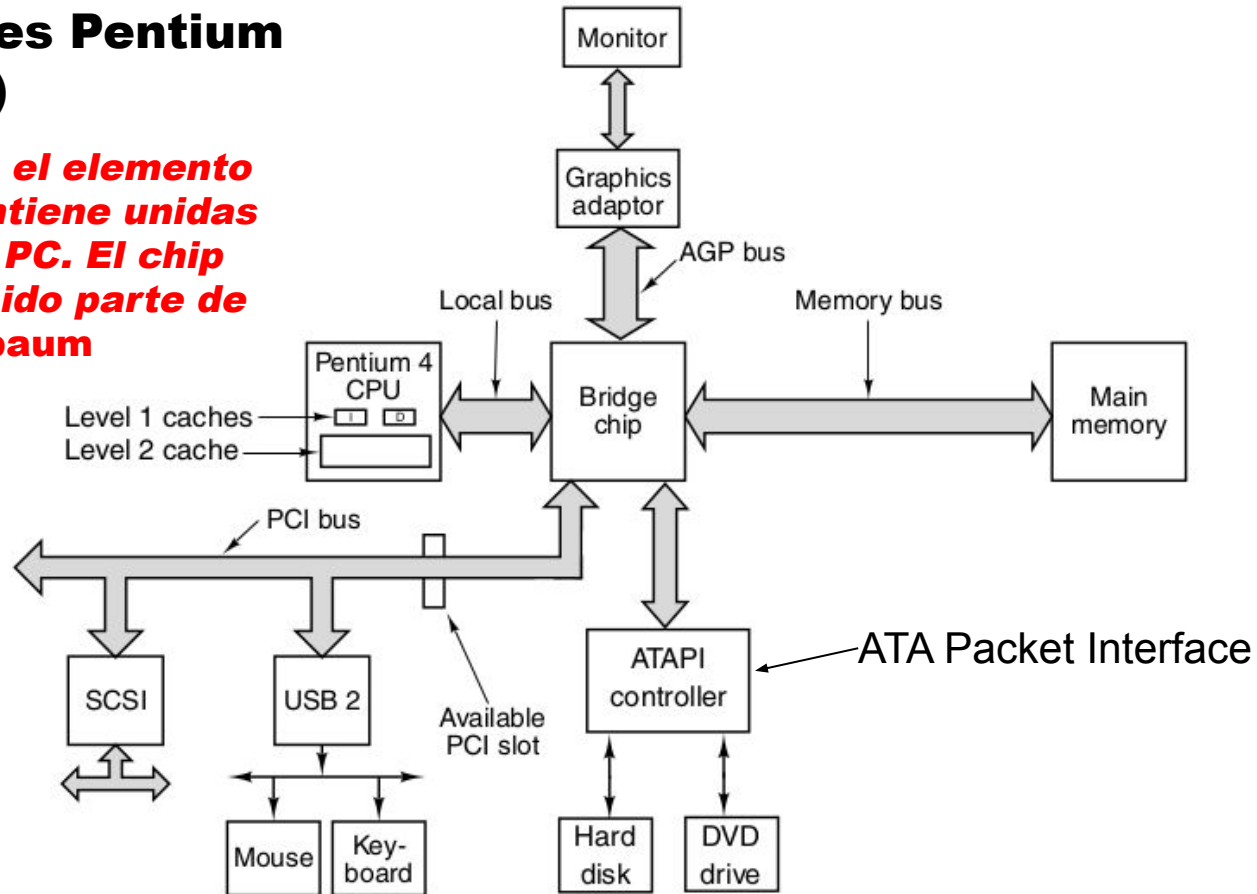






## Ejemplo: Buses Pentium 4 (2000-2008)

**“El bus ya no es el elemento central que mantiene unidas las partes de la PC. El chip puente ha asumido parte de ese rol”. Tanenbaum**





## Bus PCI Problemas

- **Periféricos que necesitan más velocidad, por ejemplo:**
  - **Placas de video**
  - **Discos duros**
  - **Placas de red**
- **Gran tamaño físico (Muchas pistas)**
- **Dificultad para incrementar la velocidad de los buses paralelo:**
  - **Interferencia entre pistas.**
  - **Interferencias externas.**



# Transmisión Paralelos vs serie (Interfaces, buses, etc.)

## Transmisión Paralelo

### Ventajas:

- Transmite  $n$  bits al mismo tiempo (en 1 ciclo).

### Desventajas:

- Necesidad de muchas líneas (costo)
- Muy afectado por interferencia entre pistas (comunicación cruzada) e interferencia externa (depende de la velocidad).
- Límite práctico 66MHz.

## Transmisión Serie

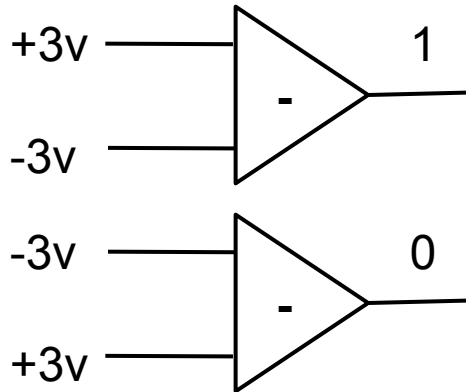
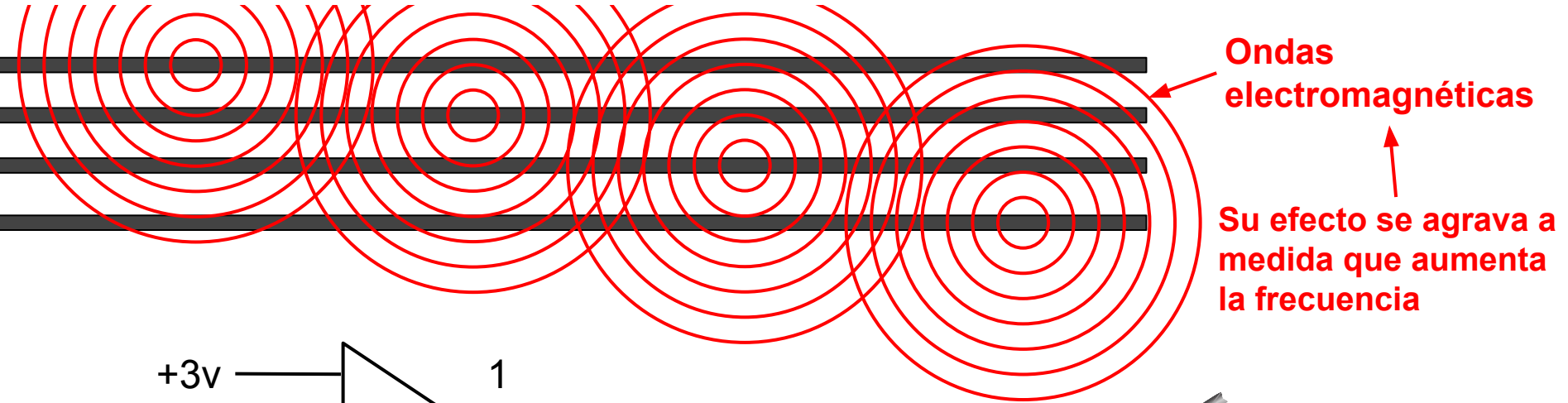
### Desventajas:

- Transmitir  $n$  bits requiere al menos  $n$  ciclos.

### Ventajas:

- Gran inmunidad a interferencia entre pistas o externa si se usa par trenzado.
- Menos líneas (menor espacio y costo)

# Transmisión Paralelos vs serie (Interfaces, buses, etc.)



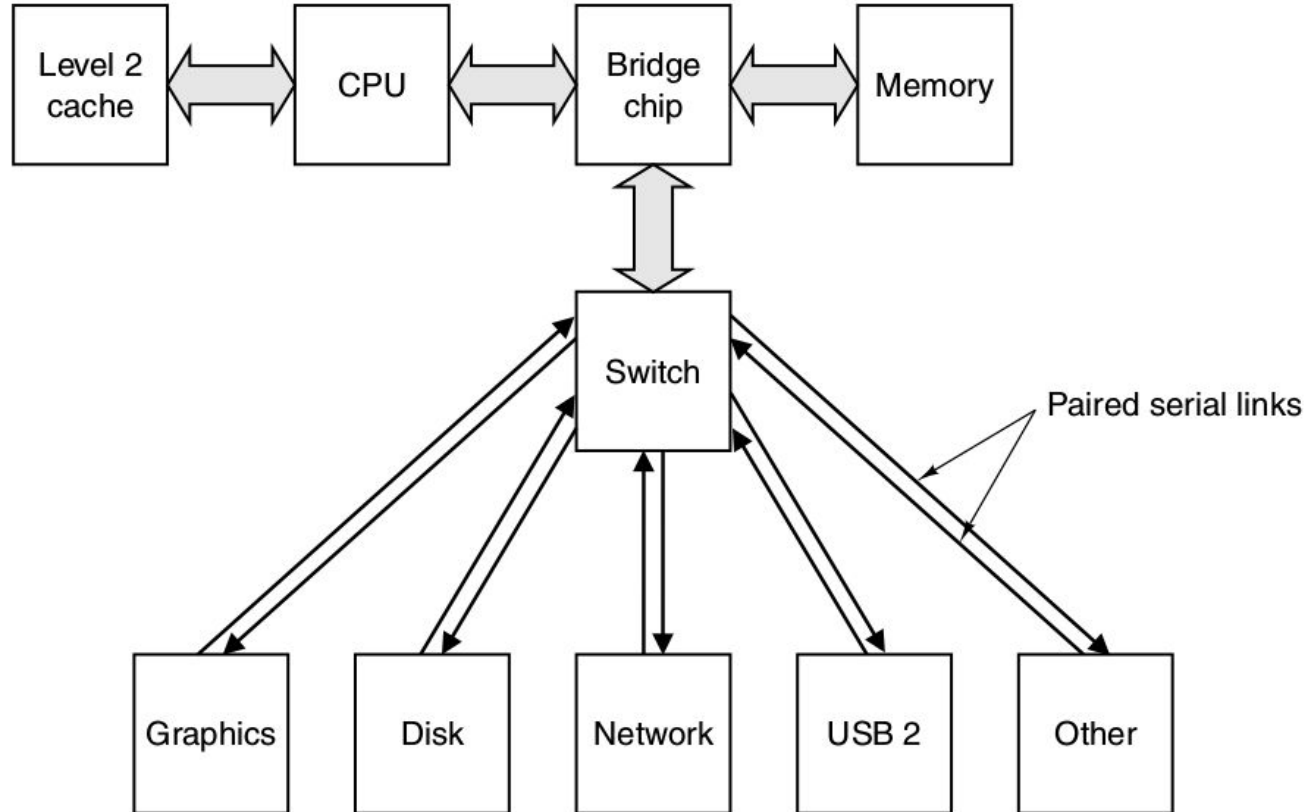


## Bus PCI Express (PCIe)

- **2002**
- **Serie (enlaces de par trenzado)**
- **Asíncrono**
- **Conexiones punto a punto a través de un dispositivo llamado **Switch** (conmutador).**
- **Arquitectura es más cercanas a una red LAN (Ethernet conmutada) que a un bus.**
- **Según Tanenbaum:**
  - **No es un bus**
  - **Nada que ver con el bus PCI**
- **No es compatible con el bus PCI**
- **Toma su nombre por motivos de **marketing****



# Bus PCI Express





## **Bus PCI Express (PCIe)**

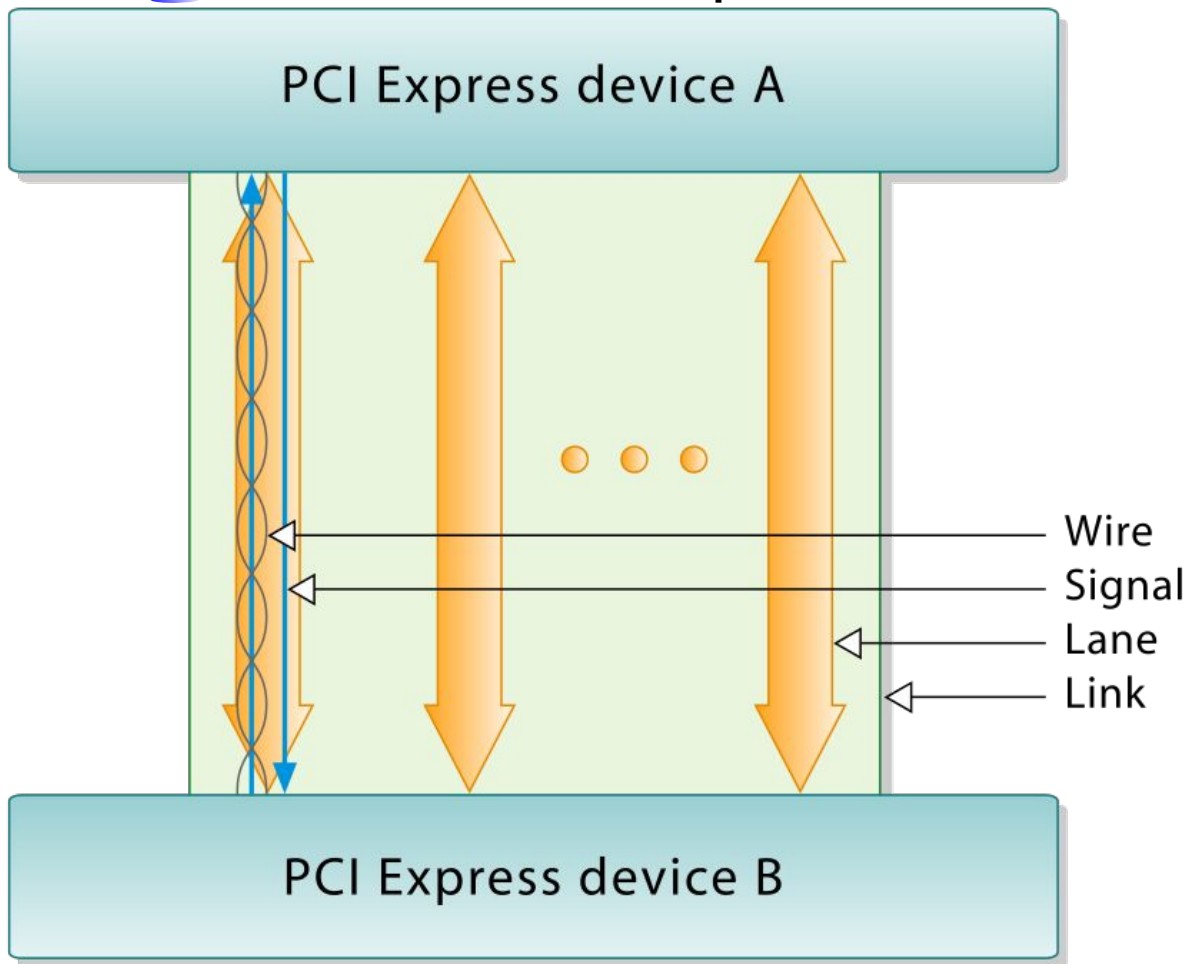
- **Cada conexión (link) consiste de 1,2,4,8,....,32 canales bidireccionales (lane). Cada canal consiste en cuatro “cables” (dos para enviar datos, dos para recibir).**
- **Utiliza señales diferenciales**





Wire: cable  
Signal: señal  
Link: enlace

1 lane: 15.8  
Gbps (4.0)

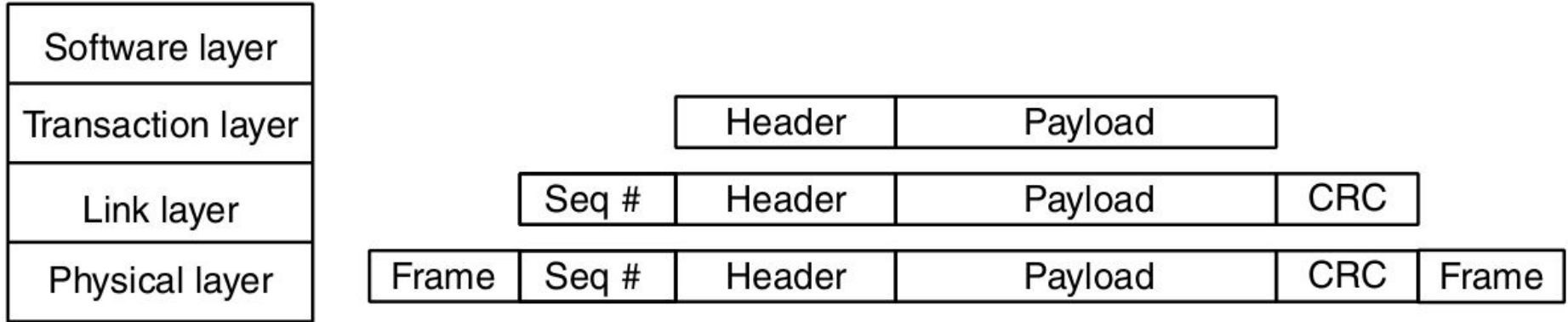




## Bus PCI Express (PCIe)

- **Comunicación basada en paquetes de datos**
  - **Encabezado: información (señales) de control -> No son necesarias líneas de control**
  - **Payload: contiene los datos a transmitir**
  - **Utiliza un código de detección de errores**
  - **Paquetes ack.**
  - **Control de flujo (evita que un emisor pueda enviar datos a mayor velocidad de la que el receptor puede recibir).**
- **Un dispositivo PCIe puede comportarse como un puente PCIe -> arquitectura tipo árboles.**
- **Utiliza una pila de protocolos**

# Bus PCI Express (PCIe) Pila protocolos



Transaction layer: distinguir tipos de mensajes (datos, configuración, etc.)

Link layer: Control de errores. Agrega CRC (detectar errores) y Seq (número de secuencia, para detectar paquetes perdidos o duplicados). Paquetes ACK y NACK.

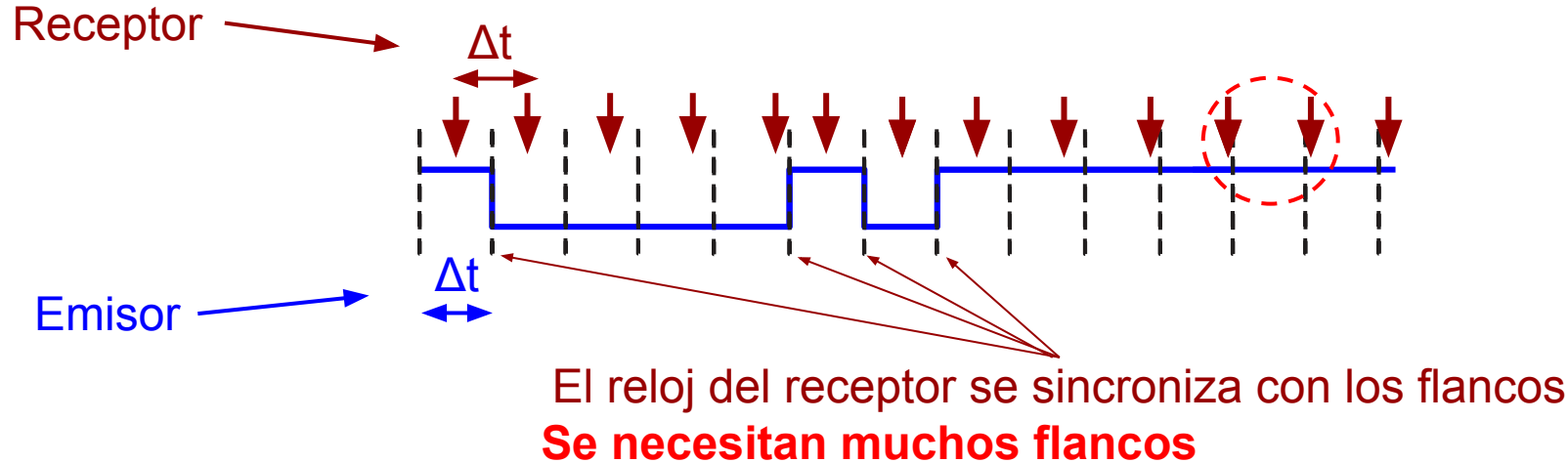
Physical layer: Separar paquetes de datos y sincronizar relojes.



## Bus PCI Express (PCIe)

- **Bus Asíncrono: NO utiliza un clock -> 2 Problemas:**
  - **Necesidad de sincronizar transmisor y receptor:**
    - **Solución: código 8b/10b** (uso diez bits para transmitir 8 bits de información). Se usan los símbolos con mayor cantidad de transiciones.
      - **Diferencia entre GT/s (total de bits) y Gb/s (bits de datos).**
  - **¿Cómo sabe el receptor cuando un dispositivo esta por comenzar una transmisión o cuando ha finalizado una?**
    - **Solución: bits de “frame” al comienzo y final del paquete.**

## Bus PCI Express - Códigos de sincronismo



- Se necesitan “bits” adicionales para sincronismo.
- Los fabricantes pueden indicar la velocidad como:
  - GT/s: Indican todos los bits por segundo, pero no todos los bits son útiles.
  - GB/s: Bits útiles por segundo.



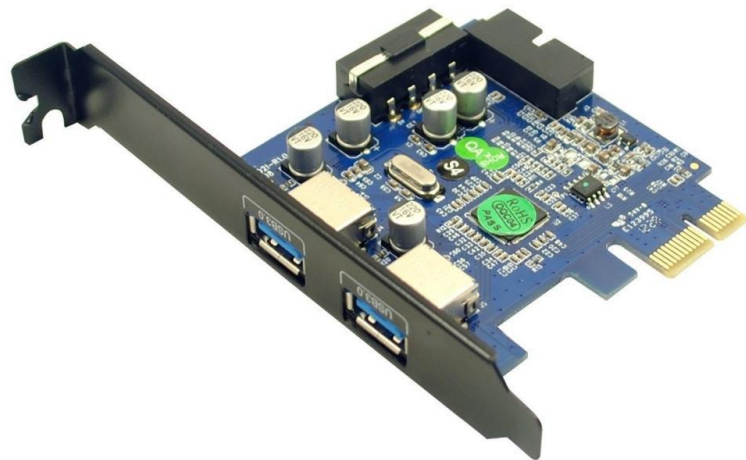
**UNCUYO**  
UNIVERSIDAD  
NACIONAL DE CUYO



**FACULTAD  
DE INGENIERÍA**

**Placas y ranuras PCI Express**

**Licenciatura en Ciencias de la  
Computación**



PCI x1



PCI x16

Ultra Durable



## Placa De Video Evga Geforce Rtx 30 Series Rtx 3070.

- Conectividad: HDMI, DisplayPort.
- PCI-Express 4.0
- Cantidad de núcleos 5888.
- Resolución 7680x4320.
- 650 W



## Diferencia con el bus PCI y PCI Express

	<b>PCI</b>	<b>PCI Express</b>
<b>Serie o paralelo</b>	<b>Bus paralelo</b>	<b>Bus serie</b>
<b>Topología</b>	<b>Bus (Conexión compartida)</b>	<b>Estrella (Switch compartido o conmutada)</b>
<b>Unidades de transmisión</b>	<b>Se transmiten bloques de bits en paralelo.</b>	<b>Se transmiten paquetes</b>
<b>Sincronización</b>	<b>Bus síncrono</b>	<b>Bus asíncrono</b>





# PCIe Bandwidth & Frequency



Year	Bandwidth	Frequency/Speed
1992	133MB/s (32 bit simplex)	33 Mhz (PCI)
1993	533MB/s (64 bit simplex)	66 Mhz (PCI 2.0)
1999	1.06GB/s (64 bit simplex)	133 Mhz (PCI-X)
2002	2.13GB/s (64 bit simplex)	266 Mhz (PCI-X 2.0)
2002	8GB/s (x16 duplex)	2.5 GHz (PCIe 1.x)
2006	16GB/s (x16 duplex)	5.0 GHz (PCIe 2.x)
2010	32GB/s (x16 duplex)	8.0 GHz (PCIe 3.x)
2017	64GB/s (x16 duplex)	16.0 GHz (PCIe 4.0)
2019	128GB/s (x16 duplex)	32.0 GHz (PCIe 5.0)
2022	256 GB/s (x16 duplex)	64.0 GHz (PCIe 6.0)

Figura Obtenida de página web del “PCI Special Interest Group”.

## Aplicaciones del bus PCI express

- Es el **bus primario** de las computadoras actuales
- Ha reemplazado a los buses dedicados a vídeo.
- **AMD se usa para interconectar sus chipsets (northbridge y southbridge).**
- **Se está usando para conectar dispositivos de almacenamiento masivo (SATA Express, SCSI Express, Dispositivos almacenamiento masivo PCIe)**



**PCIe**



## Bus USB

### ¿Porqué?

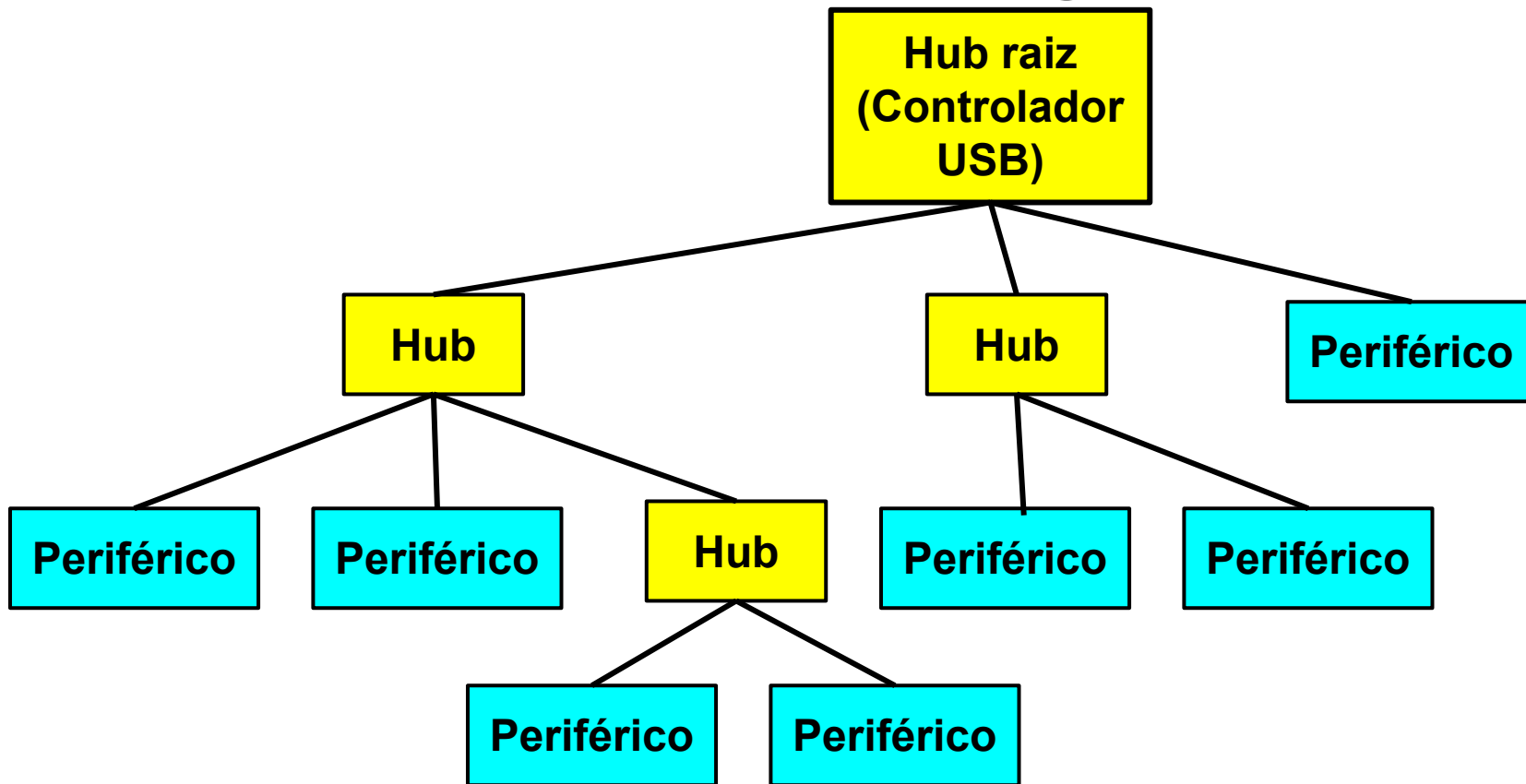
- **ISA, PCI y PCIe demasiado costosos para dispositivos I/O de baja velocidad externos.**
- **Necesidad de abrir la PC para insertar la placa ISA o PCI.**

### Solución: **Bus USB** (Universal Serial Bus)

- **Objetivo: conectar dispositivos externos.**
- **USB Implementers Forum** (Intel, Microsoft, IBM, Apple, Compaq, DEC, NEC y Northern Telecom) (<http://www.usb.org/>).
- **1996.**
- **Bus serie.**
- **4 cables (2 alimentación, 2 datos).**
- **Topología Árbol -> ¿Es un bus?**
- **Comunicación por paquetes + pila de protocolos + maestro/esclavo.**



## Bus USB: Topología





**Hub USB externo**



Hub USB interno  
(panel USB)

Cable Alimentación  
eléctrica



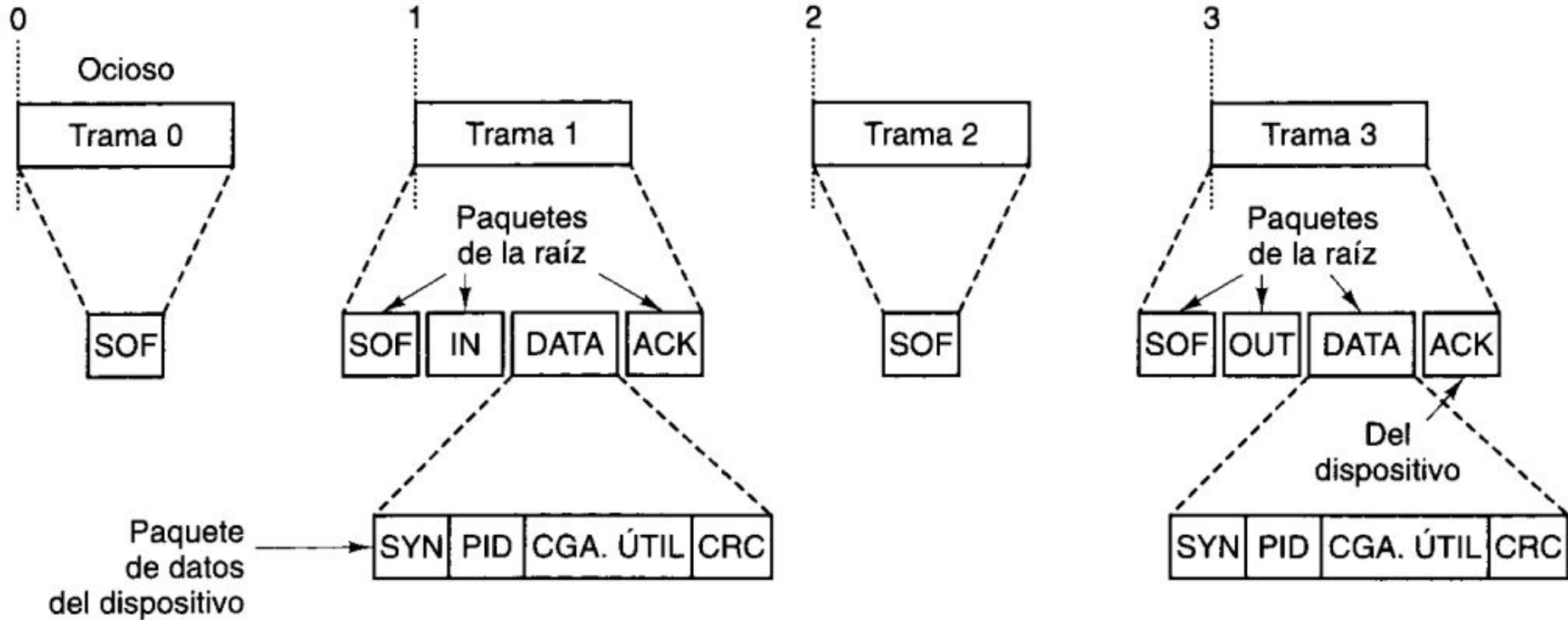
Conector USB  
placa madre



## **Objetivos originales**

**(resumen)**

- 1. Que los usuarios no tengan que abrir la PC para instalar nuevos dispositivos I/O.**
- 2. Un único tipo de cable.**
- 3. Que los dispositivos tomen su energía del propio cable.**
- 4. Hasta 127 dispositivos.**
- 5. Que los dispositivos puedan agregarse “en caliente” (sin apagar la computadora).**
- 6. Bajo costo.**







## **Al conectar un dispositivo USB**

- **El Hub raíz lo detecta (detecta cambios de voltaje en los pines de datos) y genera una interrupción.**
- **El Hub raíz le asigna un identificador (uno entre los 127).**
- **El Hub raíz solicita al dispositivo información (fabricante, dispositivo, versión, etc.).**
- **El sistema operativo busca el driver y lo carga en memoria.**

### **Comunicación con los dispositivos**

- **El hub raíz envía cada 1 ms un mensaje SOF (Start of Frame o inicio de trama).**
- **El hub raíz envía un mensaje IN (indicando que espera datos del dispositivo) u OUT (indicando que enviará datos del dispositivo).**
- **Se intercambian los datos.**
- **Se envía un mensaje de reconocimiento (ACK).**
- **El El hub raíz comienza el proceso con el siguiente dispositivo.**
- **Los dispositivos USB no interrumpen. Solo el hub raíz puede interrumpir.**



## **Versiones Bus USB**

<b>Versión</b>	<b>Año</b>	<b>Velocidad</b>	<b>Potencia</b>
<b>USB 1.0</b>	<b>1996</b>	<b>1.5 Mb/s</b>	<b>&lt; 1 W</b>
<b>USB 1.1</b>	<b>1998</b>	<b>12 Mb/s</b>	<b>&lt; 1 W</b>
<b>USB 2.0</b>	<b>2000</b>	<b>480 Mb/s</b>	<b>2.5 W</b>
<b>USB 3.0</b>	<b>2008</b>	<b>5 Gb/s</b>	<b>4.5 W</b>
<b>USB 3.2</b>	<b>22/09/2017</b>	<b>10 Gb/s</b>	<b>100 W</b>
<b>USB 4.0</b>	<b>2020</b>	<b>40 Gb/s</b>	<b>100 W</b>



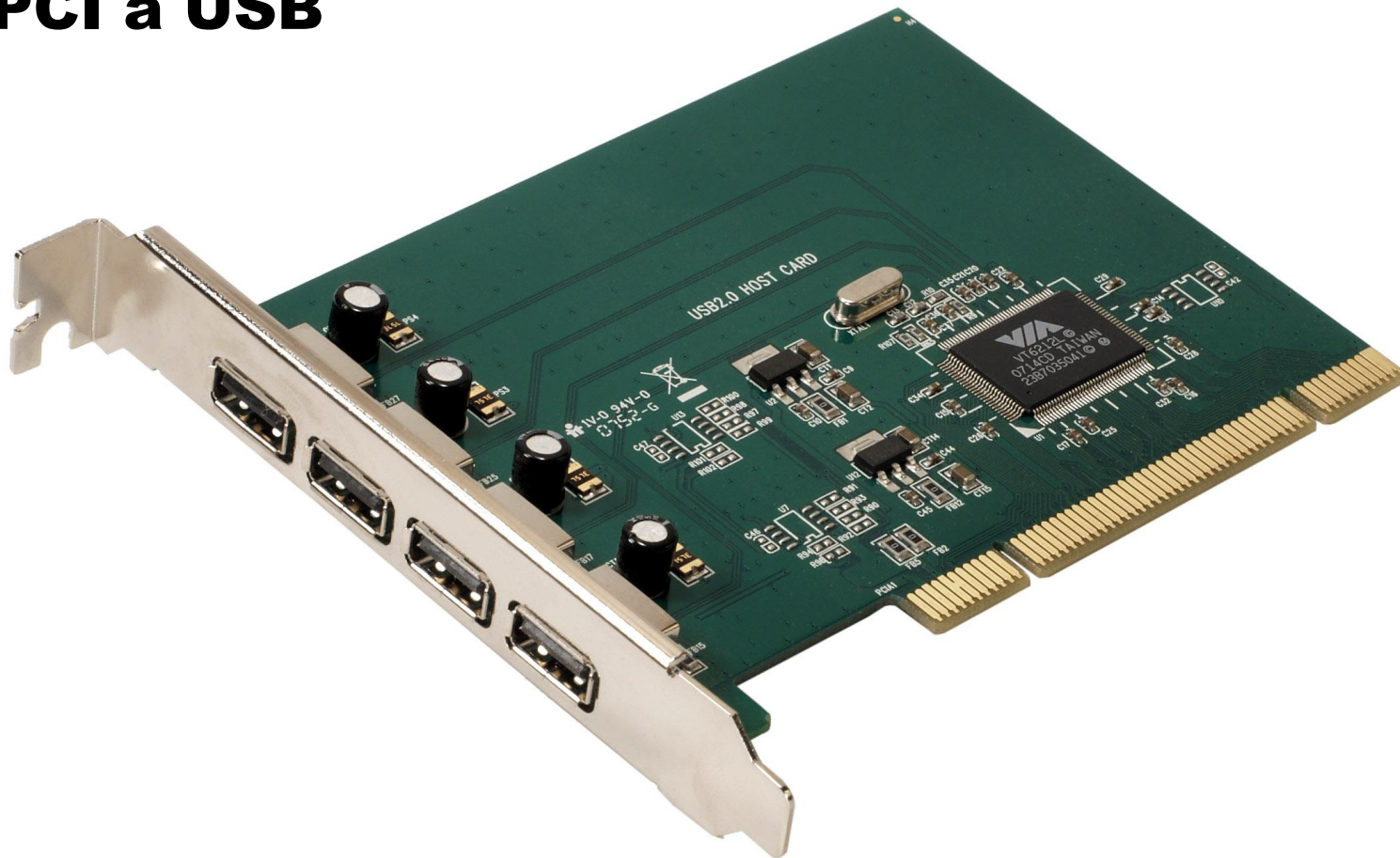
UNCUYO  
UNIVERSIDAD  
NACIONAL DE CUYO



FACULTAD  
DE INGENIERÍA

Licenciatura en Ciencias de la  
Computación

# PCI a USB





# PCIe a USB





USB Type A

USB Type B

USB 3.0

USB Mini

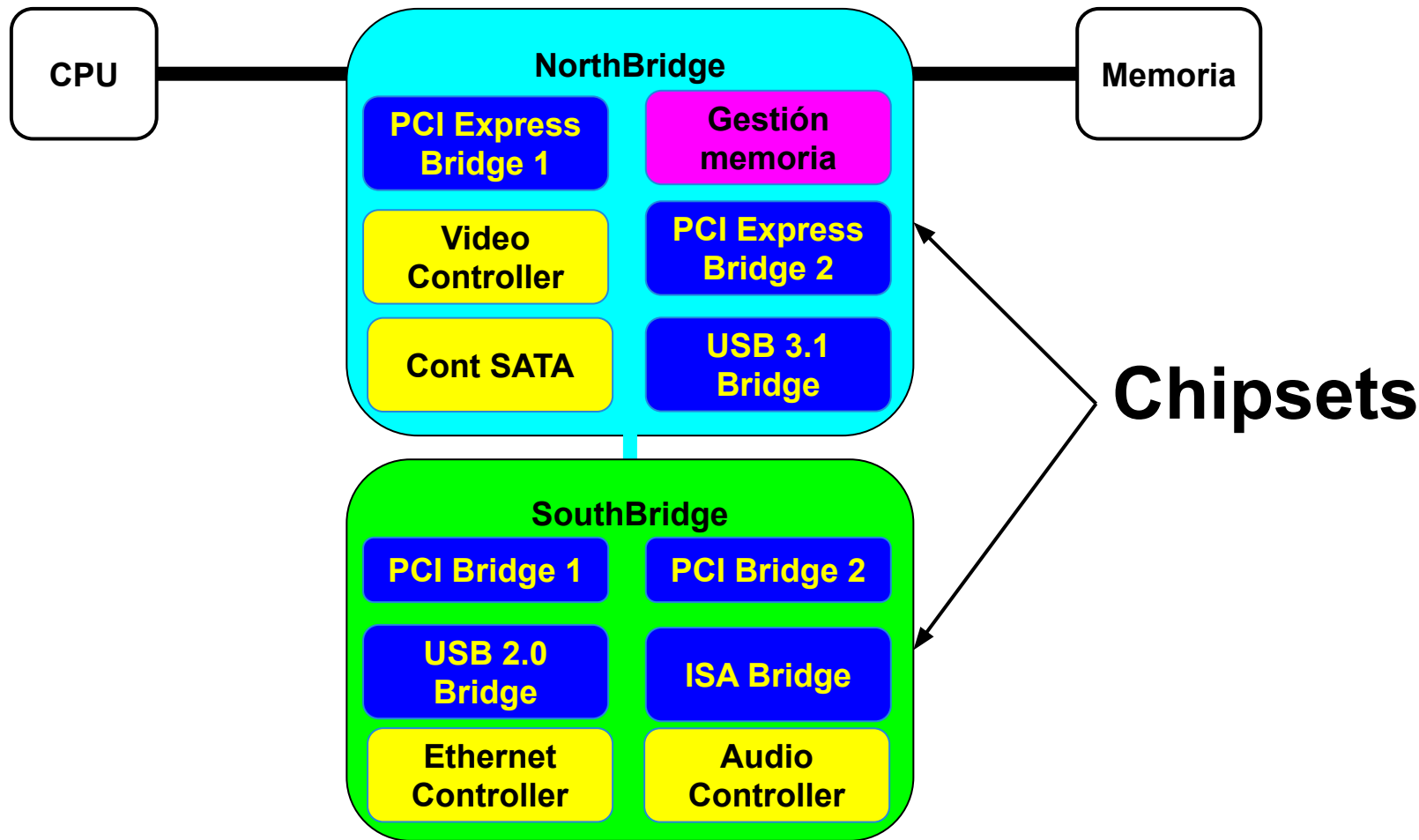
USB Micro

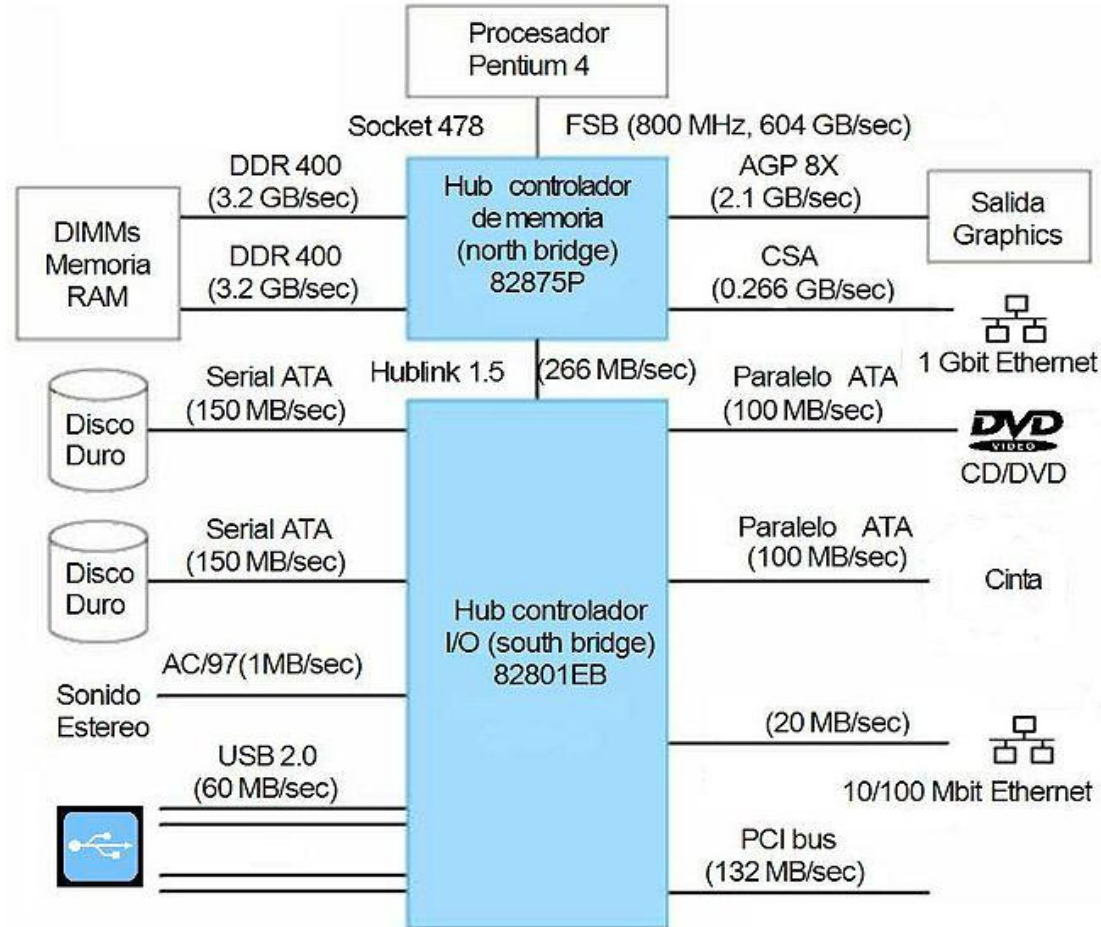
USB Type C



## Otros buses

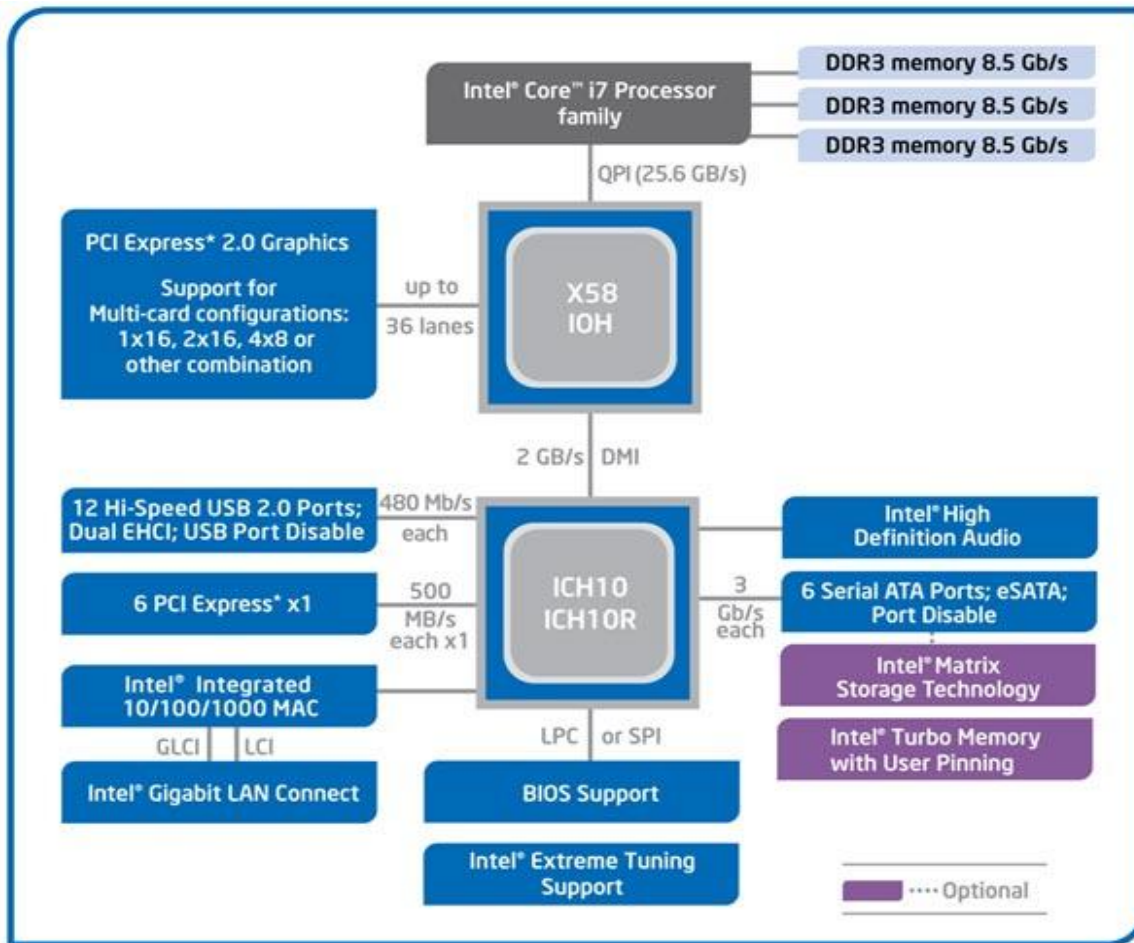
- **SMBus (System Management Bus), 10 KHz to 100 KHz**
  - Tareas de configuración básicas, errores críticos, etc.
  - Intel y Duracell
- **PMBus (Power Management Bus), 400 KHz**
- **FireWire (IEEE 1394, competencia de USB)**
  - Apple, 1990
  - También conocido como i.LINK (Sony) y Lynx (Texas Instruments)
  - Diferencia con USB: permite comunicación per-to-per.
  - Bus serie
- **Thunderbolt**
  - Sucesor de FireWire
  - Desarrollado por Intel y Apple





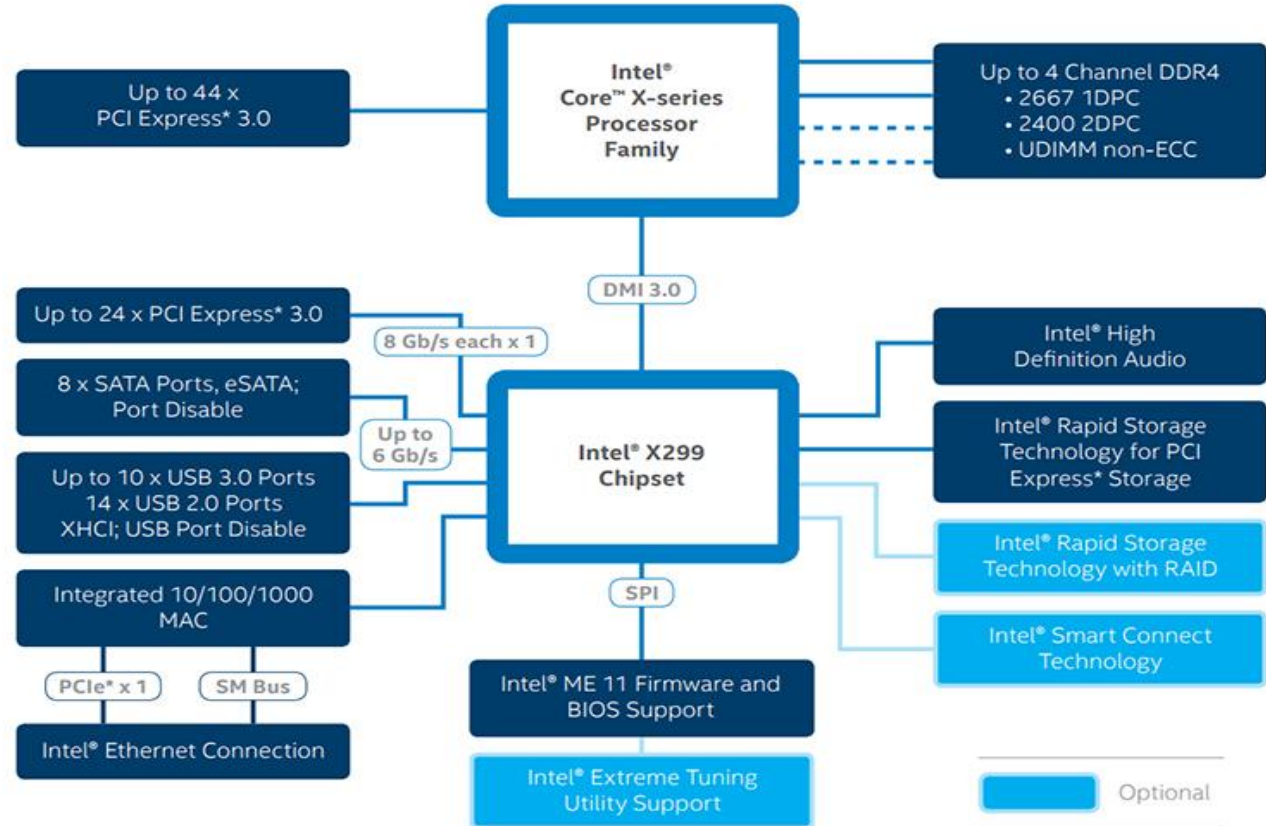


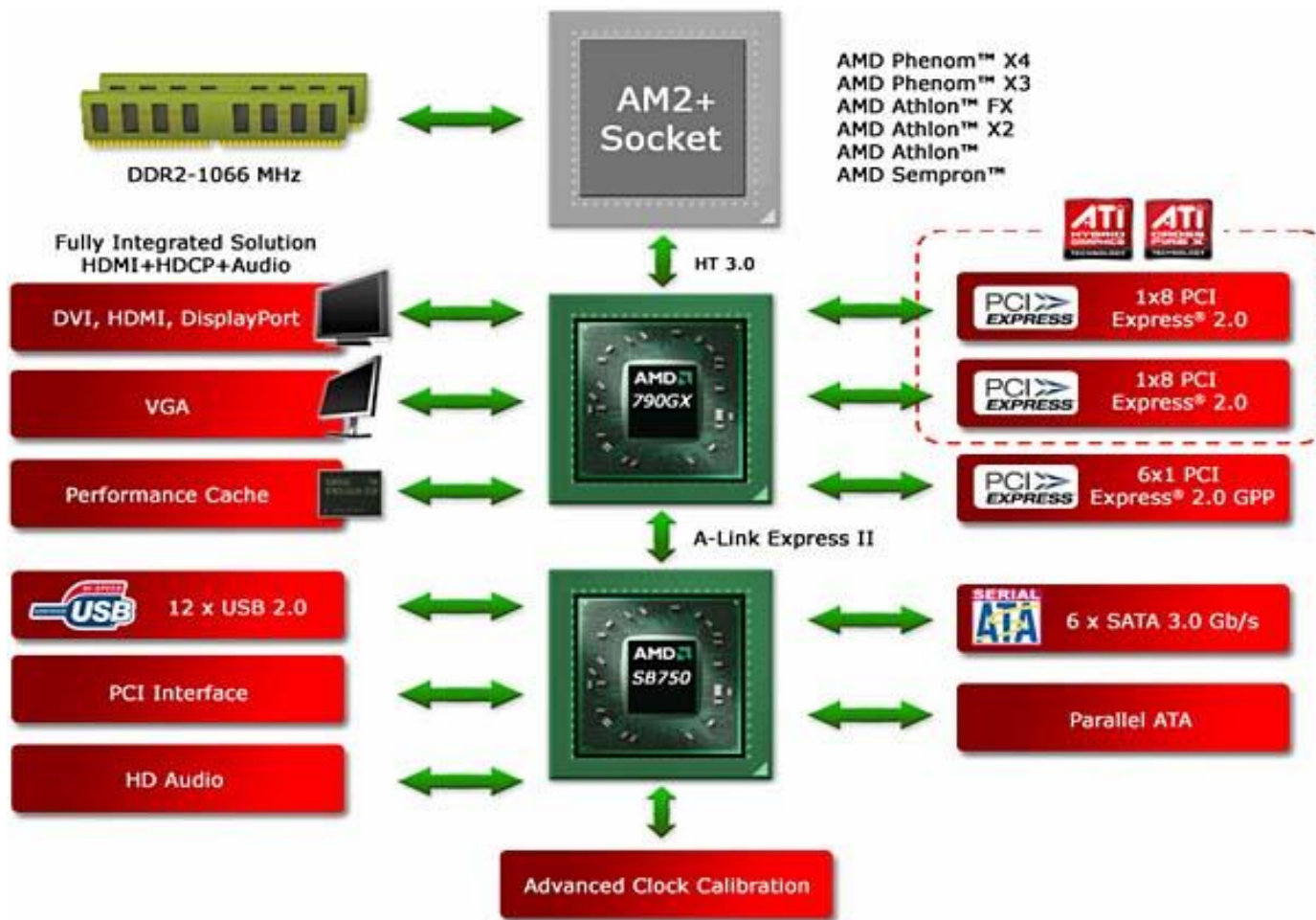
Primeras  
 arquitectur  
 as para  
 procesador  
 es i3, i5 e  
 i7 (dos  
 chipsets  
 externos)



Arquitecturas modernas para procesadores i3, i5 e i7 (un chipset interno y uno externos)

INTEL® X299 CHIPSET BLOCK DIAGRAM

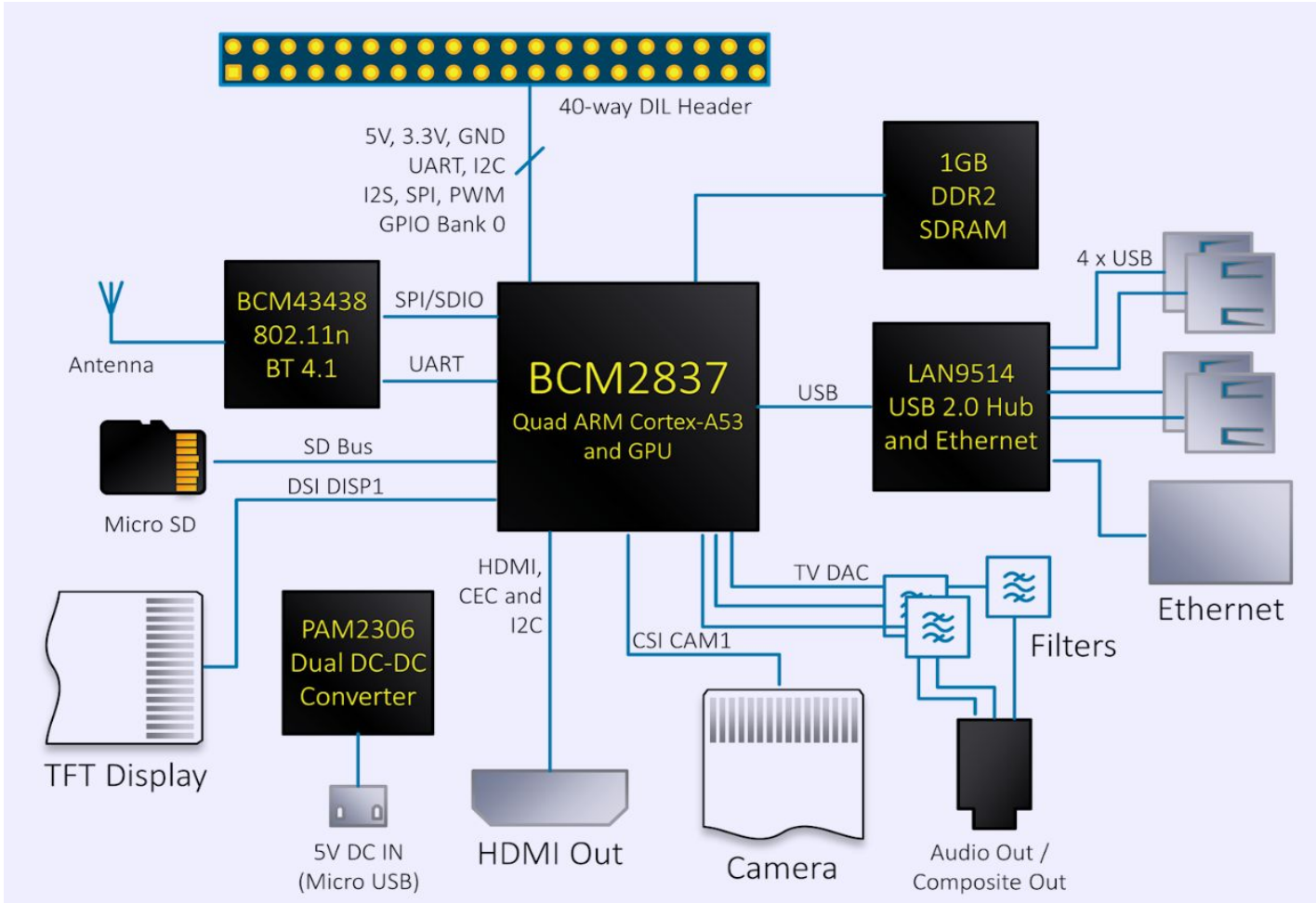


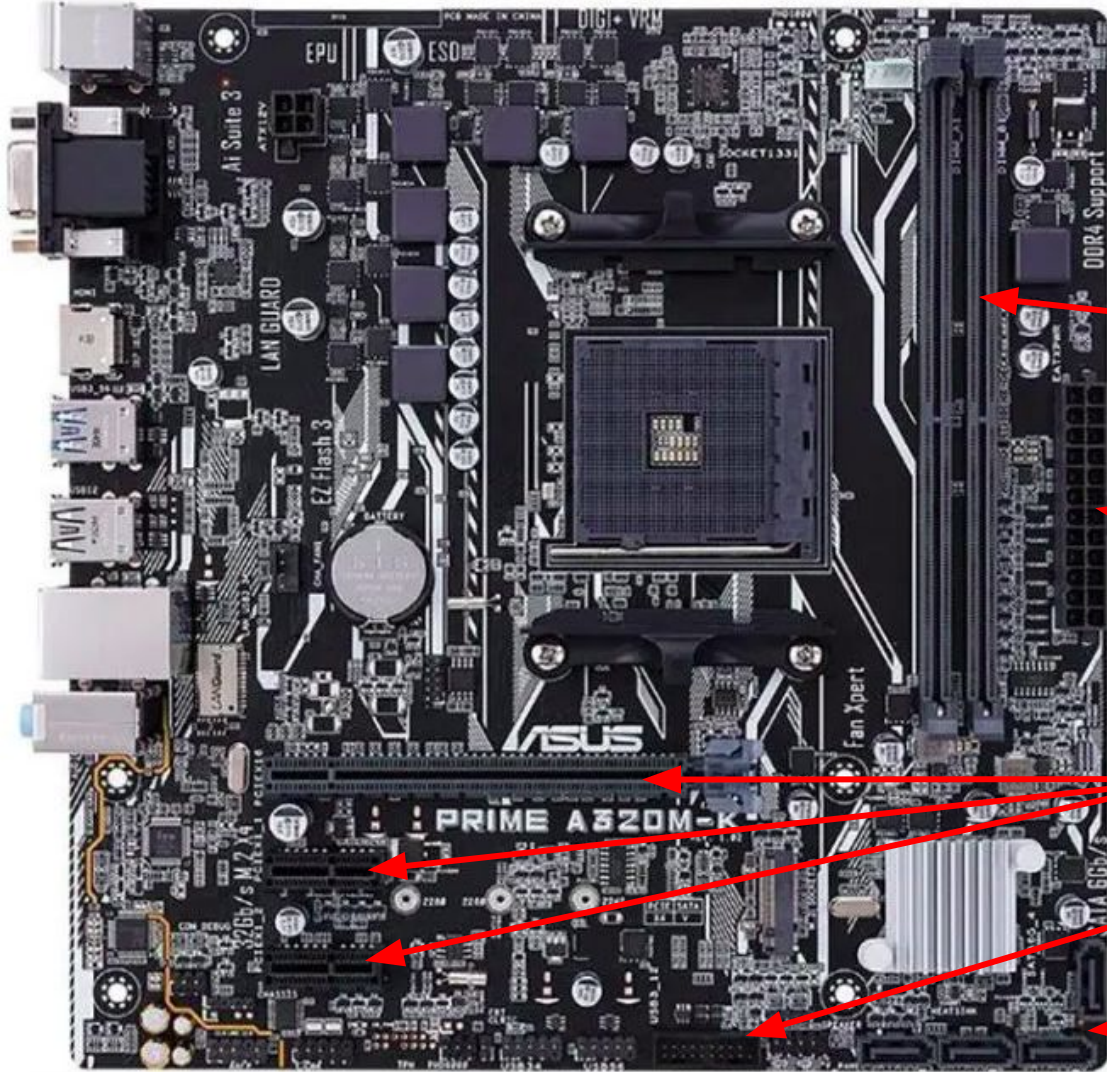






# Raspberry Pi3





Mother Asus Prime  
A320m-k Am4 Ddr4 A320  
Hdmi M2

DDR4

Conectores  
alimentación eléctrica

PCI Express

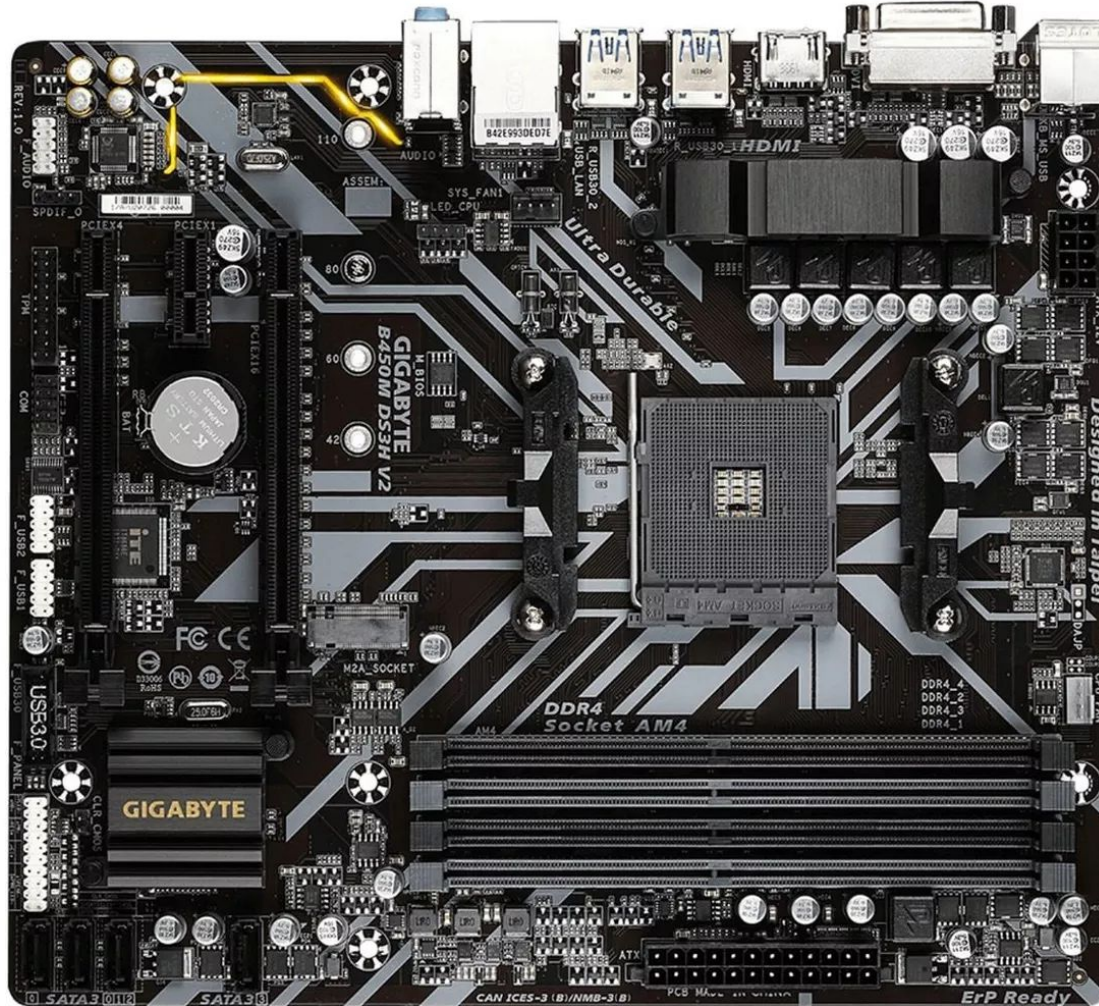
USB

SATA



Motherboard Asus Tuf Gaming B460m-plus

(<https://www.asus.com/latin/motherboards-components/motherboards/tuf-gaming/tuf-gaming-b460m-plus-wi-fi/>)



itura en Ciencias de la  
ación

Motherboard  
Gigabyte Amd  
B450m  
(<https://www.gigabyte.com/ar/Motherboard/B450M-DS3H-rev-1x#kf>)





# Computadora Raspberry Pi

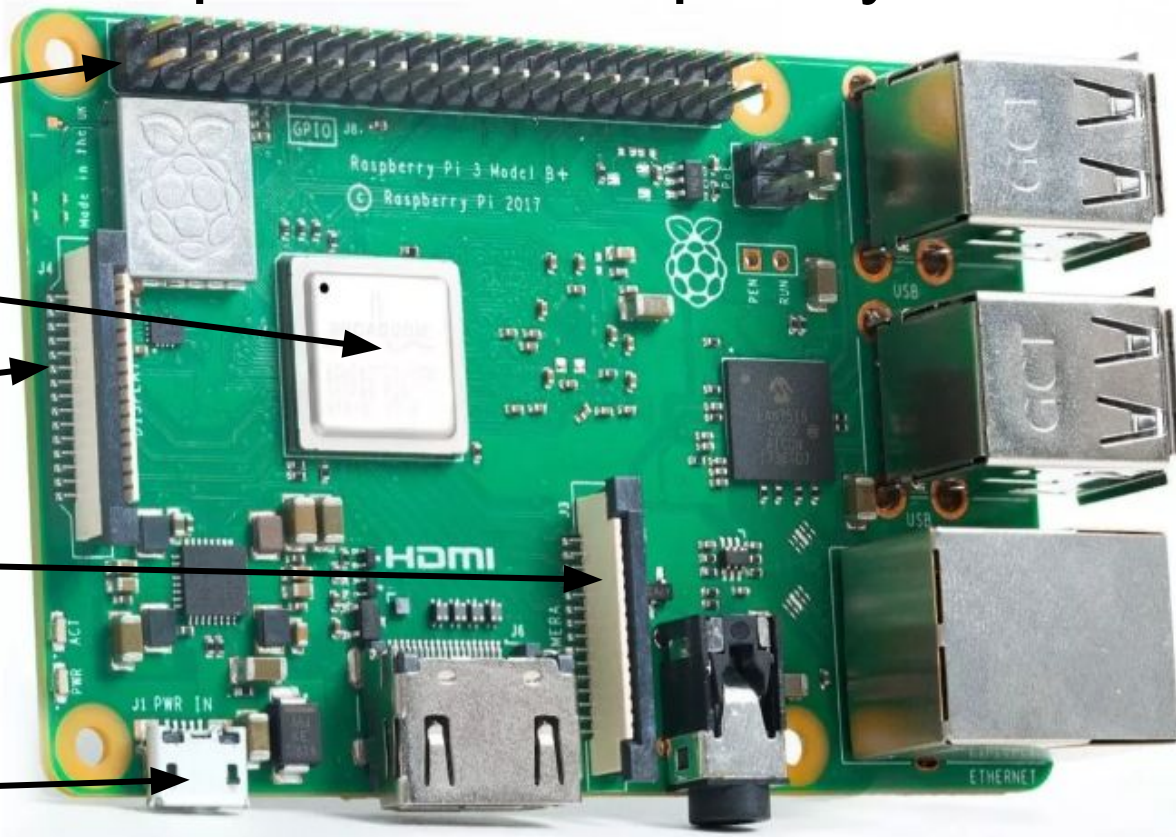
Pines de propósito general

Procesador

Salida Video DSI

Entrada Video DSI

Alimentación (micro USB o USB C)



USB (4 puertos)

Conector Red



## Bibliografía

### Libros:

- Organización de computadoras; Andrew Tanenbaum; 2008
- Computer Organization and Architecture; William Stallings; 2010.

### Páginas web:

- USB Implementers Forum; <http://www.usb.org/>
- PCI Special Interest Group; <https://pcisig.com/>

### Especificaciones:

- Hoja de datos de: Intel ® 200 (including X299) and Intel ® Z370 Series Chipset Families Platform Controller Hub (PCH), 2018.



## Apéndice: Comandos Linux útiles sobre I/O

- **watch -n1 cat /proc/interrupts**: Para ver el número de interrupciones que cada IRQ ha generado (arquitectura x86)
- **cat /proc/dma**: Muestra el número de canales de DMA (dispositivos capaces de generar peticiones de DMA) del bus ISA (nota: 4 cascade es un canal de DMA esclavo).
- **cat /proc/iomem**: mapa de memoria principal (incluye periféricos mapeados en memoria).
- **cat /proc/ioports**: Direcciones de periféricos que solo pueden ser accedidos mediante instrucciones especiales (no mapeados en memoria)
- **cat /proc/meminfo**: información de memoria (mapped: memoria usada para mapear dispositivos de E/S)



# Apéndice: Comandos usados en clase

## Linux

- **lspci**: Muestra dispositivos PIC. Algunas opciones:
  - **-k**: muestra módulos de kernel (entre ellos los drivers)
  - **-v**: verbose; **--v**: very (muy) verbose
- **lsusb**: Muestra dispositivos USB. Algunas opciones:
  - **t**: árbol