

Grupo ARCOS

uc3m | Universidad **Carlos III** de Madrid

Tema 6: E/S y dispositivos periféricos

Estructura de Computadores

Grado en Ingeniería Informática
Grado en Matemática aplicada y Computación
Doble Grado en Ingeniería Informática y Administración de Empresas



Contenidos

1. Introducción

2. Buses

- ▶ Estructura y funcionamiento
- ▶ Jerarquía de buses

3. Periféricos

- ▶ Concepto y tipos de periféricos
- ▶ Estructura general de un periférico
- ▶ Módulos de E/S

4. Interacción E/S: técnicas de E/S

5. Caso de estudio: disco duro y discos de estado sólido

Contenidos

1. Introducción

2. Buses

- ▶ Estructura y funcionamiento
- ▶ Jerarquía de buses

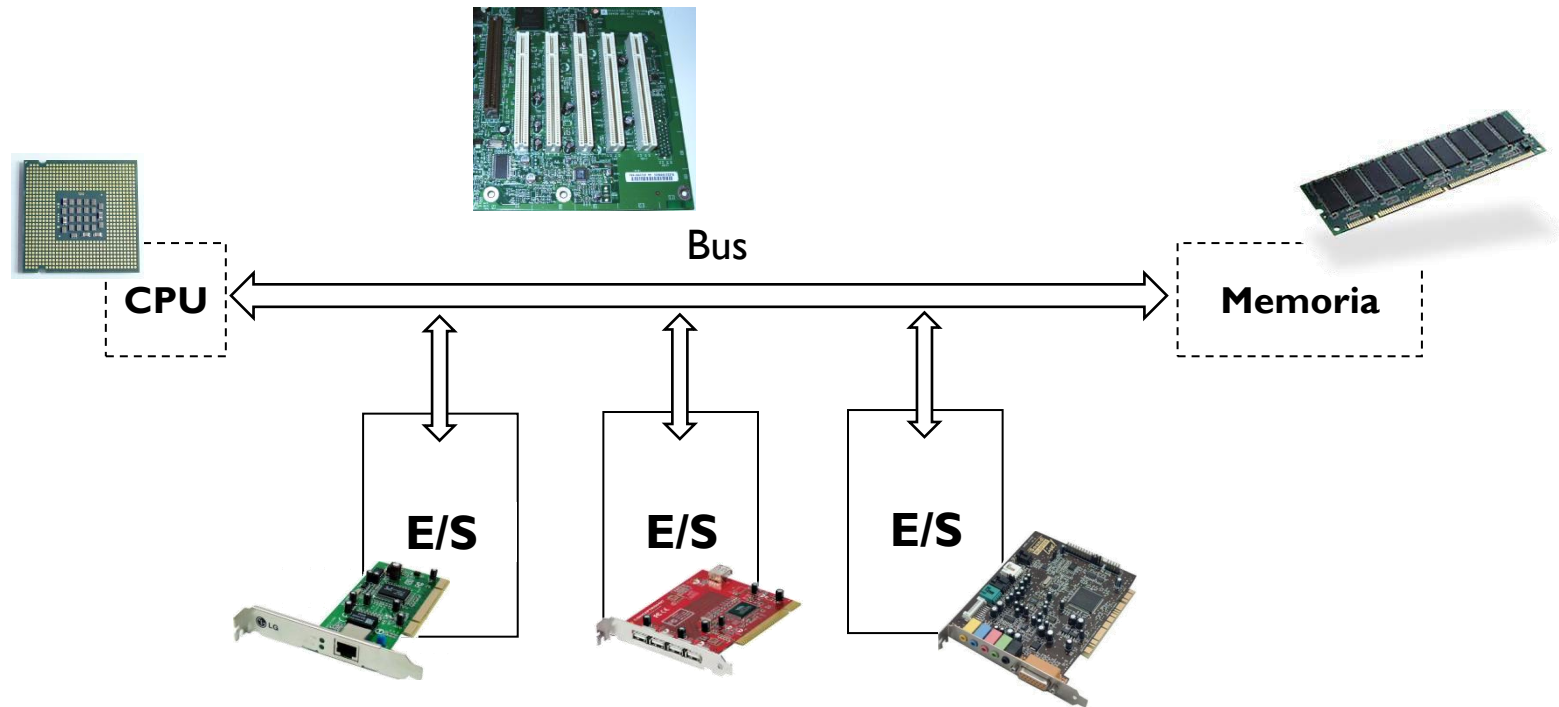
3. Periféricos

- ▶ Concepto y tipos de periféricos
- ▶ Estructura general de un periférico
- ▶ Módulos de E/S

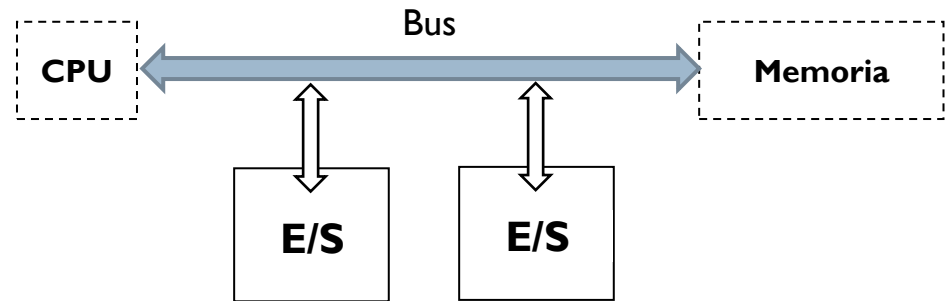
4. Interacción E/S: técnicas de E/S

5. Caso de estudio: disco duro y discos de estado sólido

Introducción

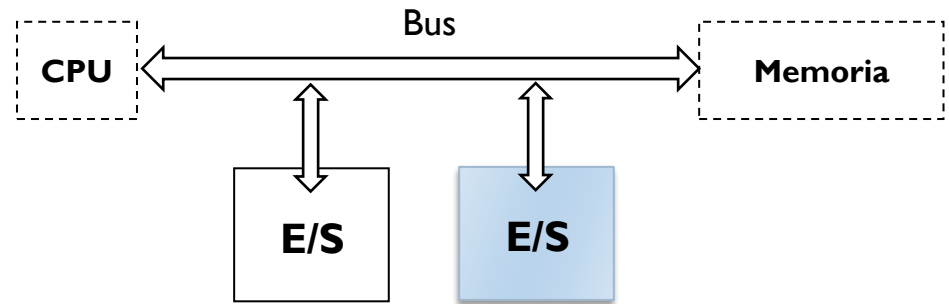


Introducción



- Qué es un bus de interconexión

Introducción



- ▶ Qué es un periférico
- ▶ Qué es un módulo de entrada/salida
- ▶ Cómo se acceden a los datos de los periféricos

Contenidos

1. Introducción

2. Buses

- ▶ Estructura y funcionamiento
- ▶ Jerarquía de buses

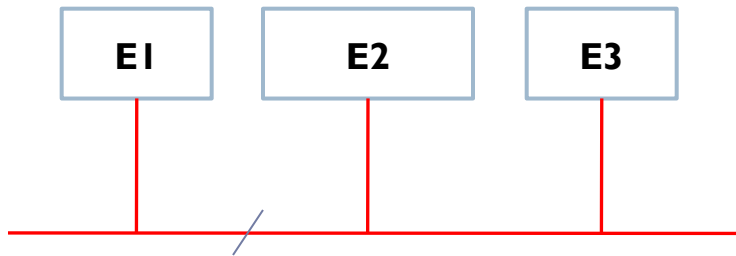
3. Periféricos

- ▶ Concepto y tipos de periféricos
- ▶ Estructura general de un periférico
- ▶ Módulos de E/S

4. Interacción E/S: técnicas de E/S

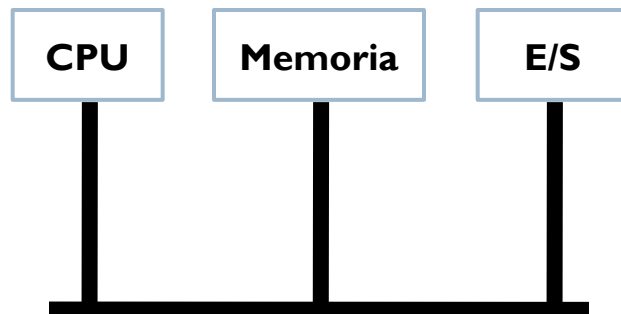
5. Caso de estudio: disco duro y discos de estado sólido

Bus



- ▶ Un bus es un **camino** de comunicación **entre dos o más dispositivos**.
- ▶ **Constituido** por **varias líneas** de transmisión de bit.
- ▶ **Medio compartido**, unívoco.
- ▶ Permite transmitir varios bits entre dos elementos conectados a él

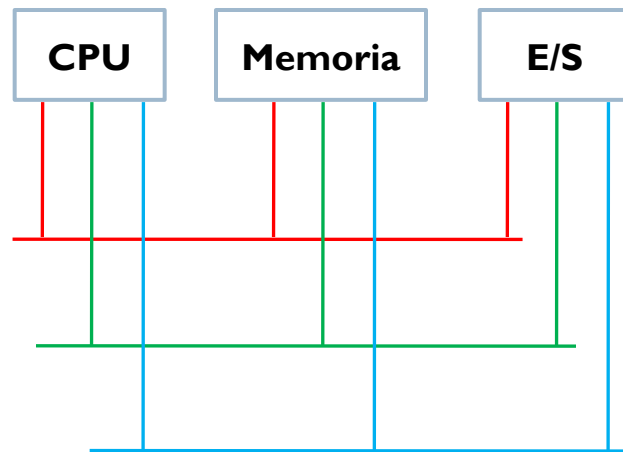
Bus del sistema



► Bus del sistema

- Conecta los principales componentes del computador
- Representa la unión de tres buses:
 - Control
 - Direcciones
 - Datos

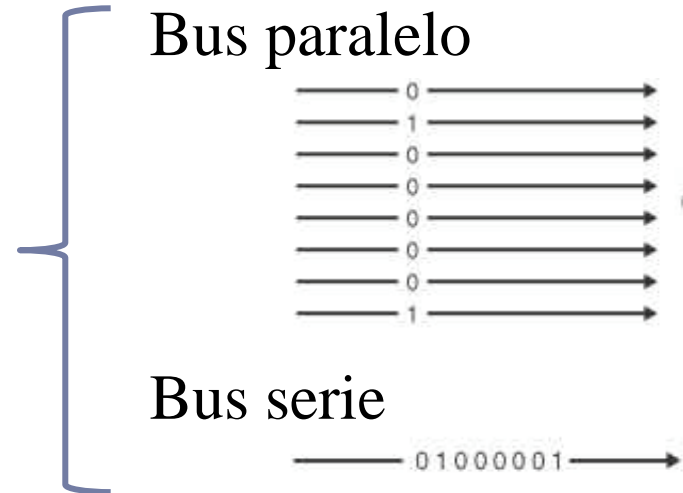
Buses



- ▶ Bus de **datos**
 - ▶ Transmite datos
 - ▶ Su anchura y velocidad influye mucho en las prestaciones
- ▶ Bus de **direcciones**
 - ▶ Direcciones de memoria y dispositivos de E/S
 - ▶ Su anchura determina la máxima capacidad de memoria
- ▶ Bus de **control**
 - ▶ Señales de control y temporización

Características de un bus

- ▶ **Ancho del bus:** determina el número de bits que pueden transmitirse simultáneamente



- ▶ **Frecuencia:** frecuencia de reloj con la que puede operar
- ▶ **Velocidad de transferencia:** número de bytes por ciclo de reloj
- ▶ **Ancho de banda** (tasa de transferencia): bytes transmitidos por segundo
 - ▶ Velocidad de transferencia X frecuencia

Ejercicio

- ▶ Calcular el ancho de banda en MBps de un bus de 32 bits y una frecuencia de 66 MHz

Ejercicio (solución)

- Calcular el ancho de banda en MBps de un bus de 32 bits y una frecuencia de 66 MHz

$$\text{Ancho de banda} = \frac{32 \text{ bits} \times 66 \text{ MHz}}{8 \text{ bits por byte}} = \frac{32 \times 66 \cdot 10^6}{8} = 264 \text{ MBps}$$

Método de arbitraje (protocolo del bus)

- ▶ Determina qué elemento de los que están conectados al bus puede acceder al bus
 - ▶ Esquema **centralizado**: un controlador del bus concede el uso del bus
 - ▶ Cuando un elemento quiere acceder al bus solicita permiso al controlador a través de las líneas de control (BUSRQ)
 - ▶ Cuando el bus está libre el controlador concede el uso (BUSACK)
 - ▶ Esquema **distribuido**: cada elemento conectado al bus incluye una lógica de control de acceso que permite usar de forma conjunta el bus (protocolo de acceso)

Buses síncronos y asíncronos

- ▶ Un bus **síncrono** está gobernado por una señal de reloj y un protocolo de comunicación ajustado al funcionamiento del reloj
 - ▶ Rápido
 - ▶ Todos los dispositivos conectados a él deben operar a la misma frecuencia de reloj
- ▶ Un bus **asíncrono** no utiliza un reloj, la comunicación se realiza mediante el envío de ordenes a través de las líneas de control del bus

Jerarquías de buses

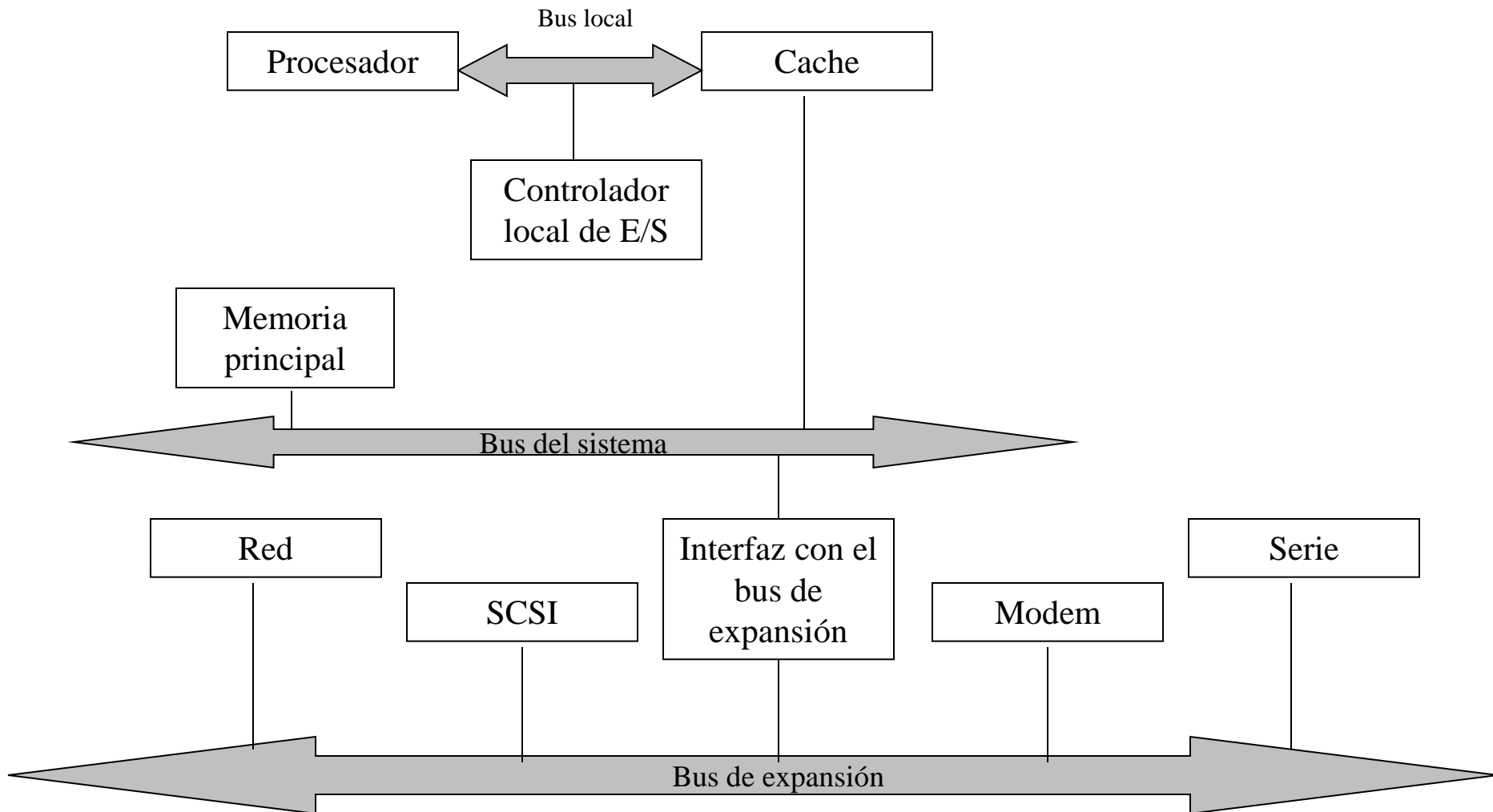
- ▶ Problema:

- ▶ A más dispositivos conectados al bus, mayor es el retardo de propagación.
- ▶ A medida que aumenta el número de peticiones de transferencia, se puede producir un cuello de botella.

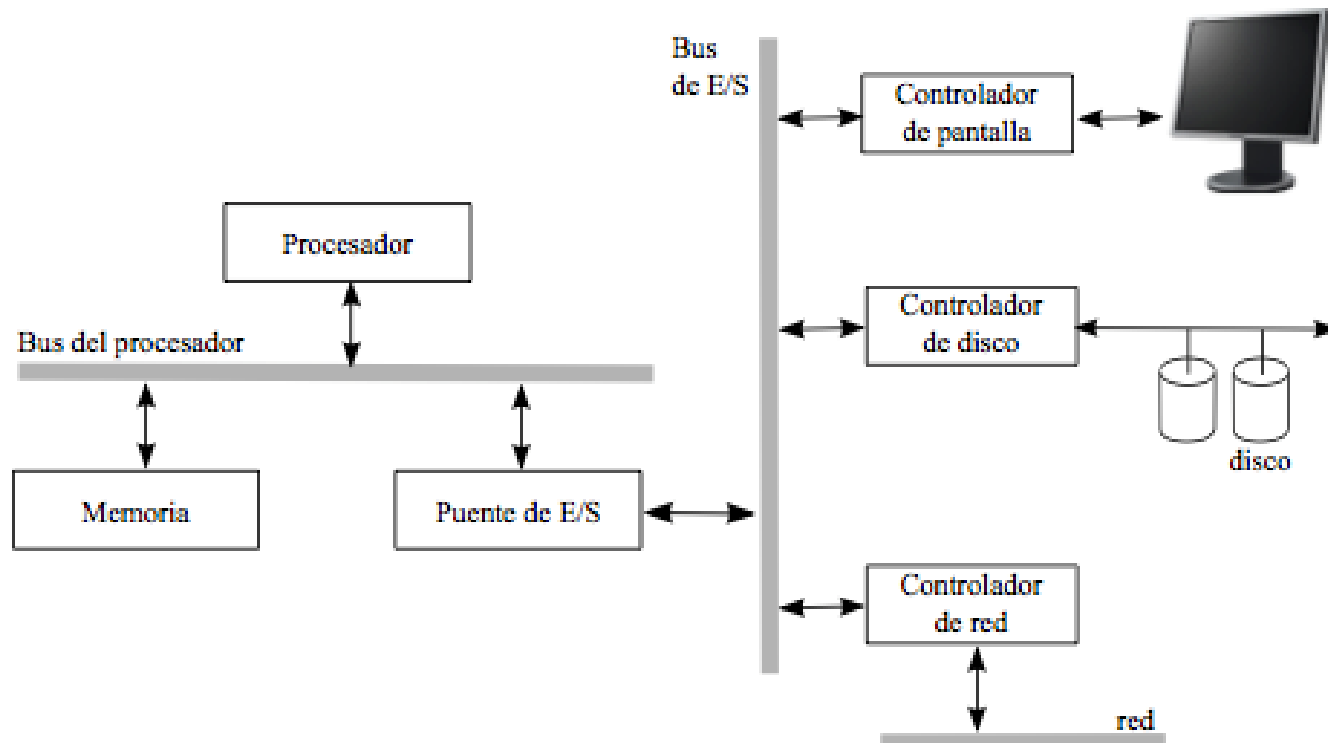
- ▶ Soluciones:

- ▶ Aumentar la velocidad de transmisión de datos con buses más anchos.
- ▶ Utilizar más buses de datos, organizados jerárquicamente.

Jerarquías de buses

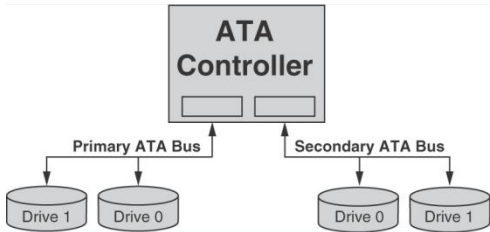
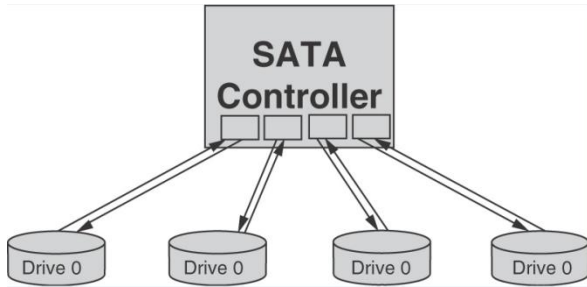
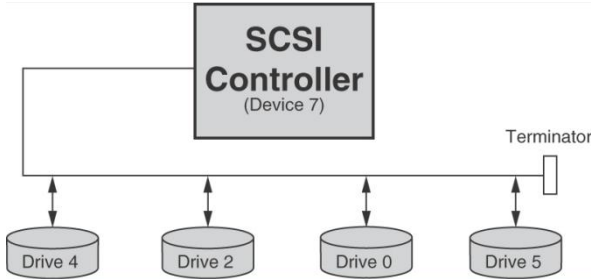
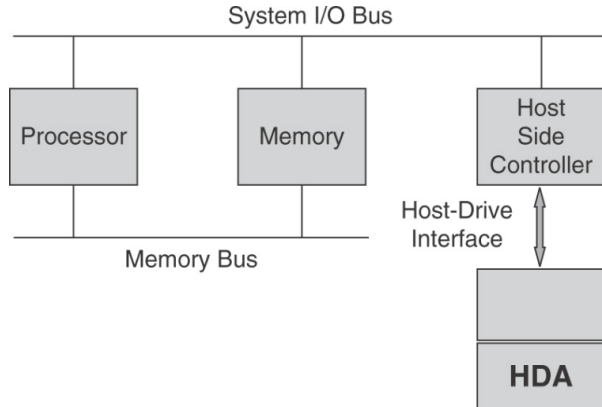


Esquema de buses en un sistema informático típico



Controladores de disco

Memory Systems
Cache, DRAM, Disk
Bruce Jacob, Spencer Ng, David Wang
Elsevier

	~ [1980 – 2010]	~ [2010 – 2022]
PC normal		
Servidores		

Curiosidades: Familia USB



	Transfer (por seg.)	Aparición
USB4	40 Gbps	2019
USB 3.2	20 Gbps	2017
USB 3.0	600 MB/s	2010
USB 2.0	60 MB/s	2000
USB 1.0	1.5 MB/s and 187 KB/s	1996

	Song / Pic	256 Flash	USB Flash	SD-Movie	USB Flash	HD-Movie
	4 MB	256 MB	1 GB	6 GB	16 GB	25 GB
USB 1.0	5.3 sec	5.7 min	22 min	2.2 hr	5.9 hr	9.3 hr
USB 2.0	0.1 sec	8.5 sec	33 sec	3.3 min	8.9 min	13.9 min
USB 3.0	0.01 sec	0.8 sec	3.3 sec	20 sec	53.3 sec	70 sec

<http://www.unp.co.in/f140/comparison-of-usb-3-0-port-with-usb-2-0-and-usb-1-0-a-70063/>

Contenidos

1. Introducción

2. Buses

- ▶ Estructura y funcionamiento
- ▶ Jerarquía de buses

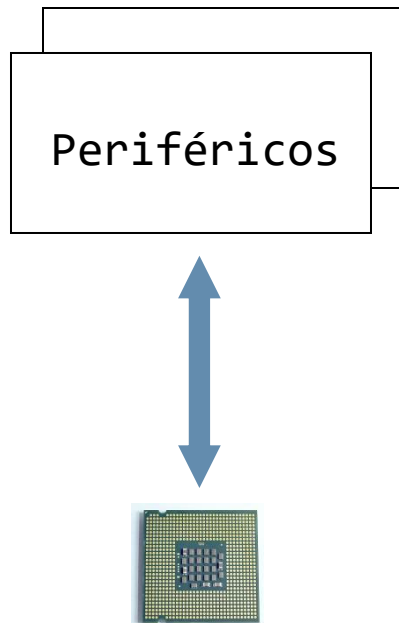
3. Periféricos

- ▶ Concepto y tipos de periféricos
- ▶ Estructura general de un periférico
- ▶ Módulos de E/S

4. Interacción E/S: técnicas de E/S

5. Caso de estudio: disco duro y discos de estado sólido

Concepto de periférico



► Periférico:

- Todo aquel dispositivo externo que se conecta a un procesador a través de las unidades o **módulos de entrada/salida** (E/S).
- Permiten almacenar información o comunicar el computador con el mundo exterior.

Clasificación de periféricos (por uso)



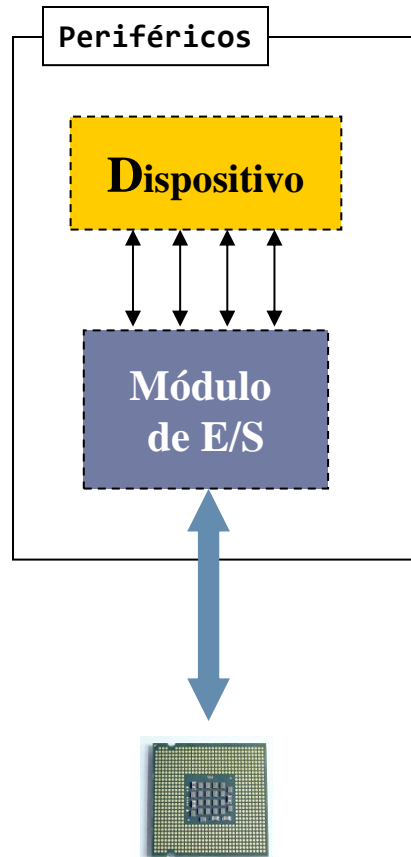
► Comunicación:

- Hombre-máquina
 - (Terminal) teclado, ratón, ...
 - (Impresa) plotter, escáner, ...
- Máquina-máquina (Módem, ...)
- Medio físico
 - (Lectura/accionamiento) x (analógico/digital)

► Almacenamiento:

- Acceso "directo" (Discos, DVD, ...)
- Acceso secuencial (Cintas)

Estructura general de un periférico



- ▶ Compuesto de:

- ▶ **Dispositivo**

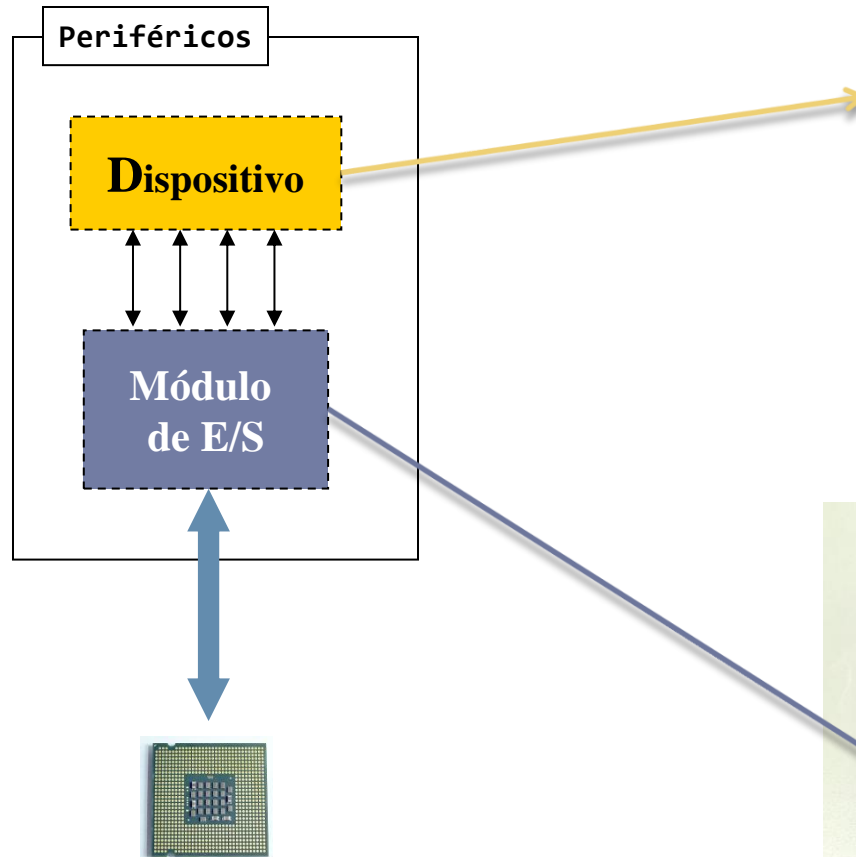
- ▶ Hardware que interactúa con el entorno

- ▶ **Módulo de Entrada/Salida**

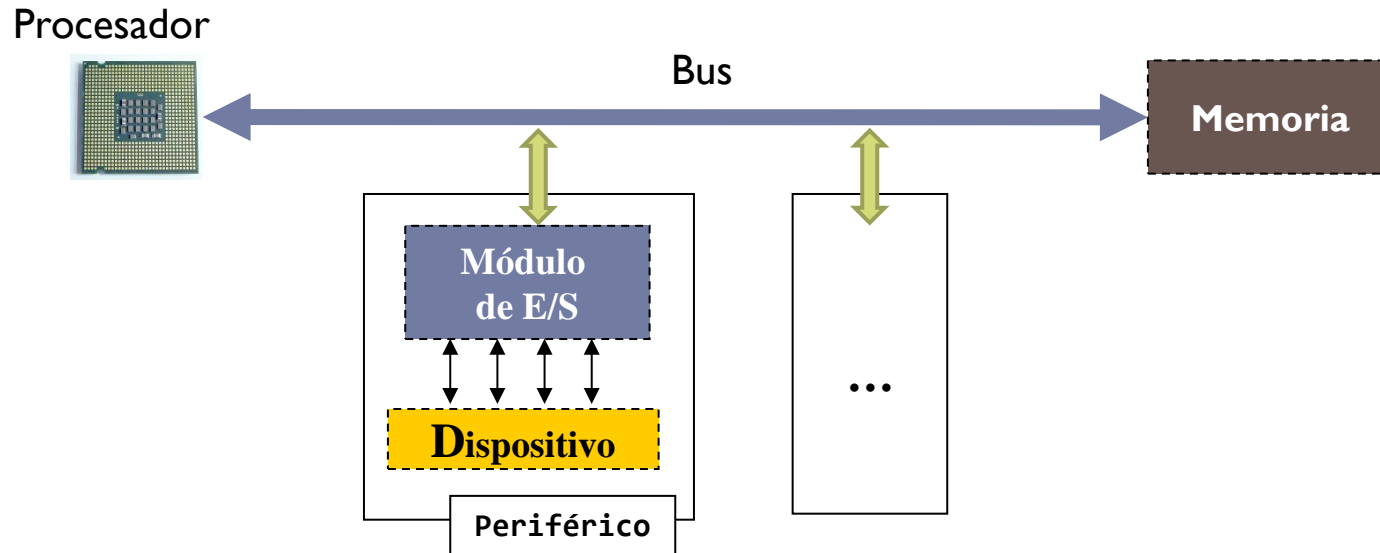
- ▶ También denominado **controlador**
 - ▶ Interfaz entre dispositivo y el procesador, que le oculta las particularidades de éste

Ejemplo

Disco magnético

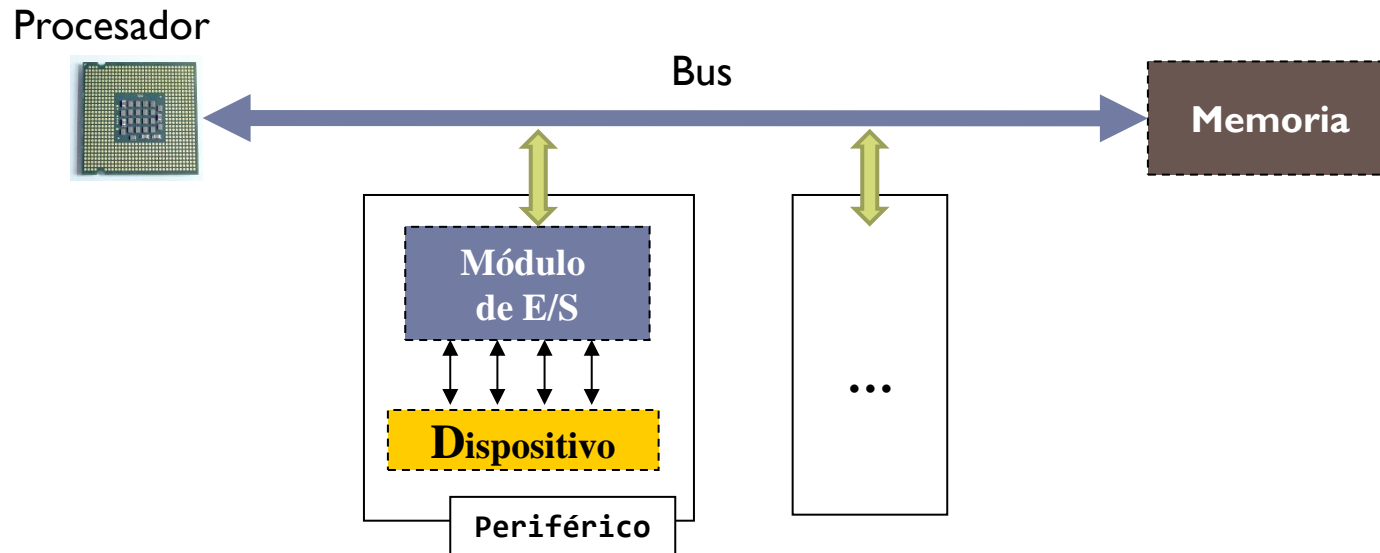


Módulo de E/S



- Las **unidades o módulos de E/S** realizan la conexión del procesador con los dispositivos periféricos.

Necesidad de los módulo de E/S



- ▶ Gran variedad de periféricos.
 - ▶ Los periféricos son 'raros'
- ▶ La velocidad de transferencia de datos de los periféricos es mucho menor que la de la memoria o el procesador.
 - ▶ Los periféricos son 'muy lentos'
- ▶ Formatos y tamaños de palabra de los periféricos distintos a los del computador al que se conectan.

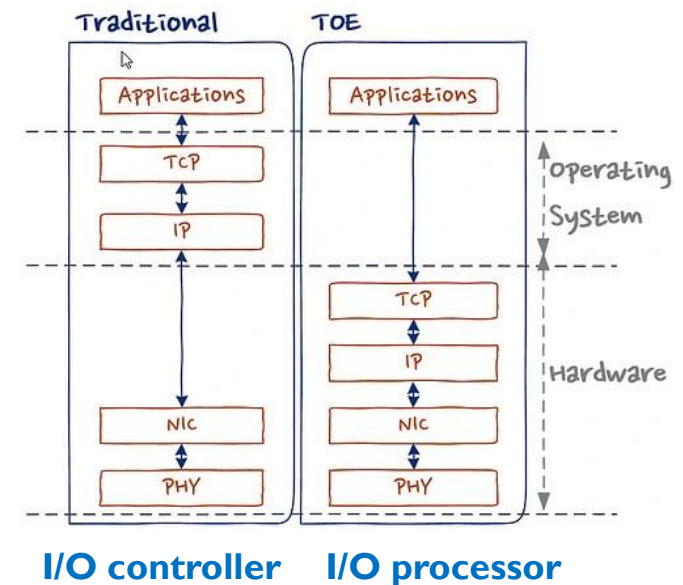
Módulos de E/S: posibles tareas

► Tareas comunes:

- Control y temporización
- Comunicación con CPU o memoria
- *Buffering* o almacenamiento intermedio
- Detección de errores
- Etc.

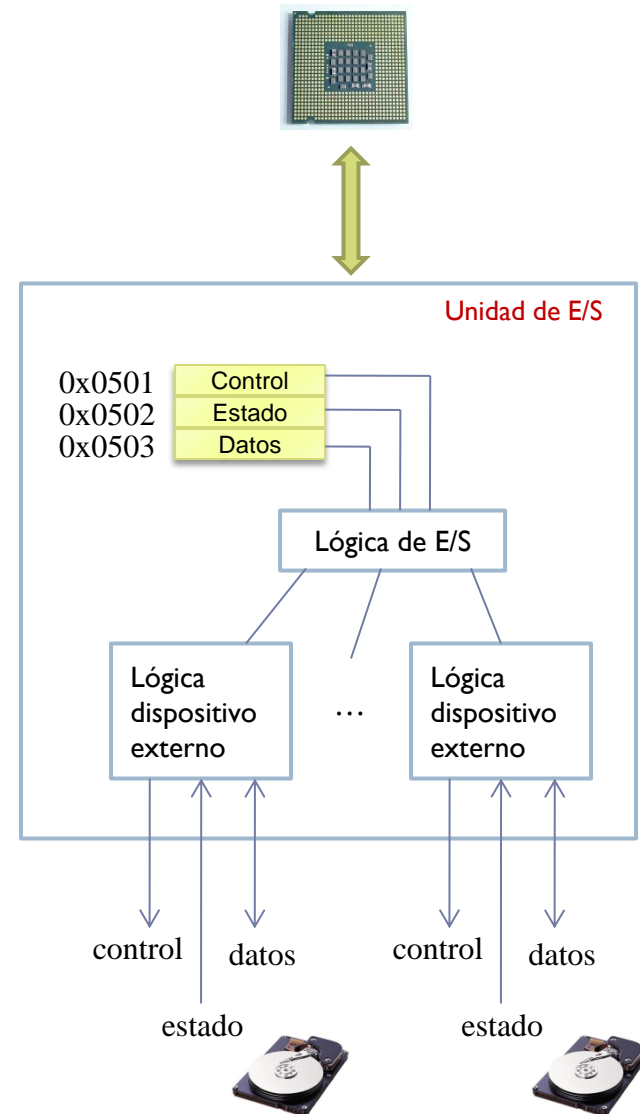
► Tipos de módulos por complejidad:

- Controladores E/S:
módulo simple que requiere de CPU para el control del dispositivo.
- Canales de E/S o procesador E/S:
realiza casi todo el control de E/S

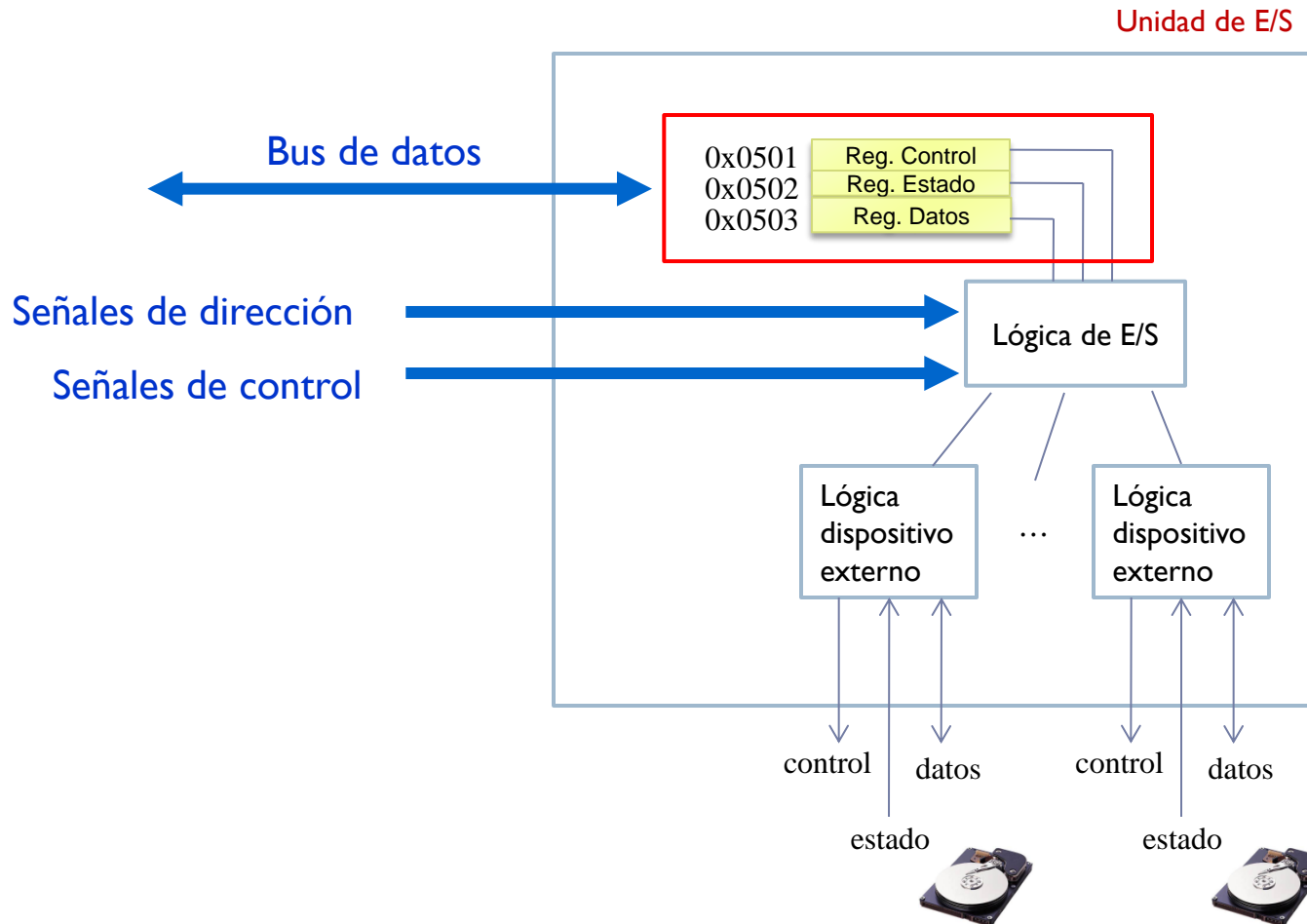


Módulo de E/S: funciones

- ▶ **Atender al procesador:**
 - ▶ Decodificación de órdenes
 - ▶ Información de estado
 - ▶ Control y temporización
 - ▶ Ej.: datos a memoria ppal.
- ▶ **Controlar periférico(s):**
 - ▶ Comunicación con dispositivos
 - ▶ Detección de errores
 - ▶ Almacenamiento temporal de datos
 - ▶ periférico->procesador

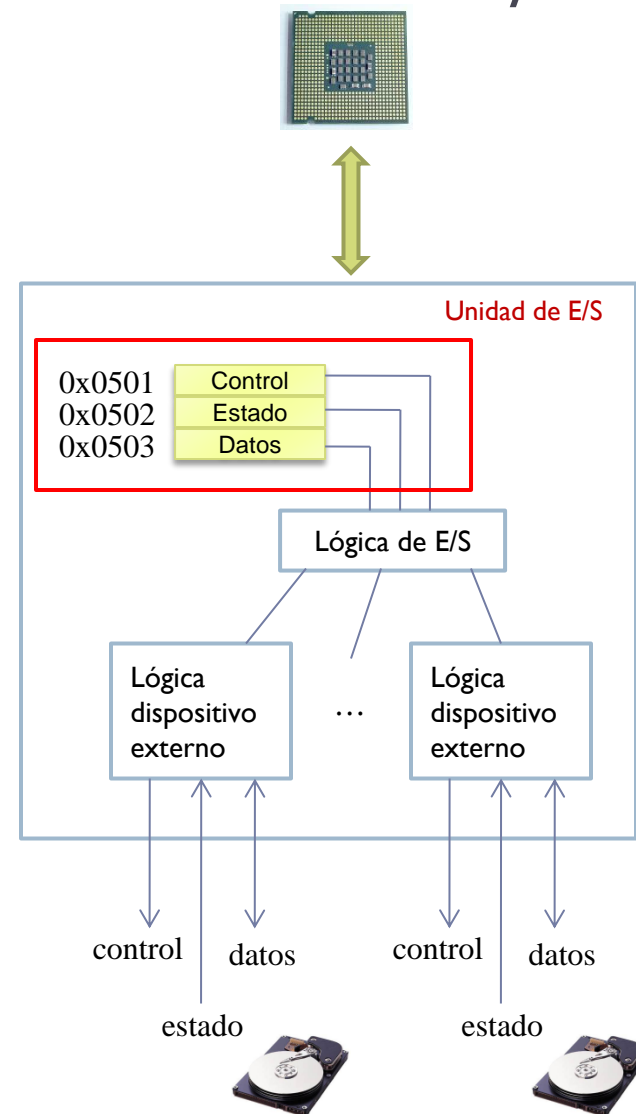


Modelo simplificado de módulo de E/S



Modelo simplificado de módulo de E/S

- ▶ Interacción entre procesador y Unidad de E/S a través de 3 registros:
 - ▶ Registro de **control**
 - ▶ Ordenes para el periférico
 - ▶ Registro de **estado**
 - ▶ Estado desde de la última orden
 - ▶ Registro de **datos**
 - ▶ Datos intercambiados Procesador/Perif.



Modelo simplificado de módulo de E/S

► Interacción periférico/unidad de E/S:

► **Líneas de datos:**
transferencia de información

► **Señales de estado:**
diagnóstico del periférico

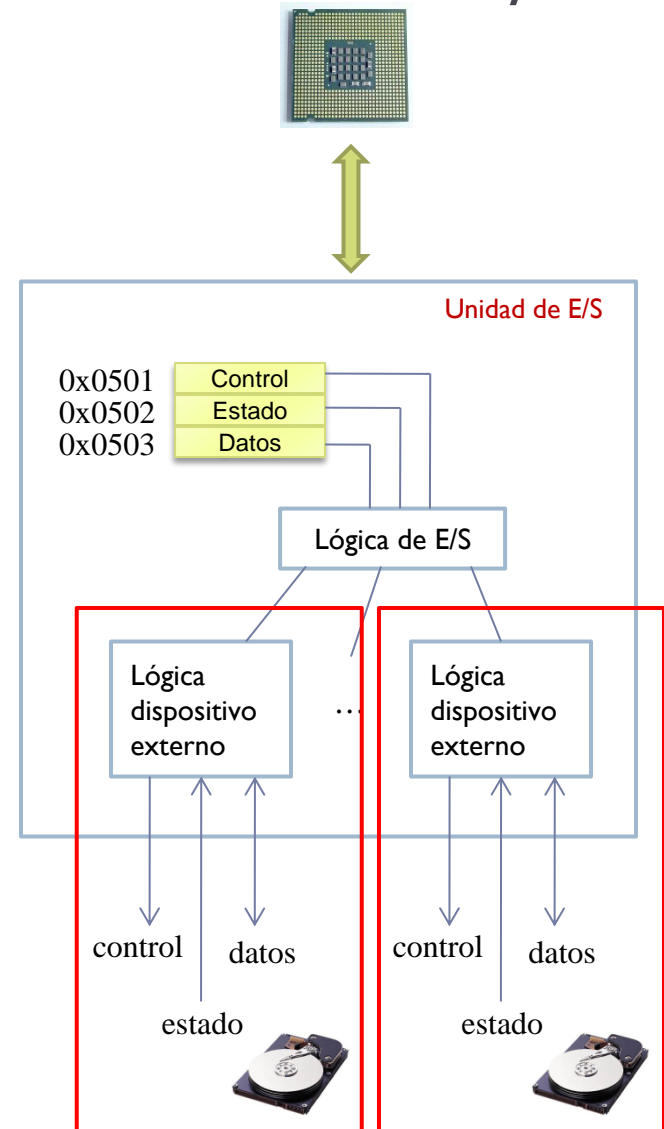
► Ejemplos:

- Nuevo dato disponible
- Periférico encendido/apagado
- Periférico ocupado
- Periférico operativo o no
- Error de operación
- ...

► **Señales de control:**
accionamiento del periférico

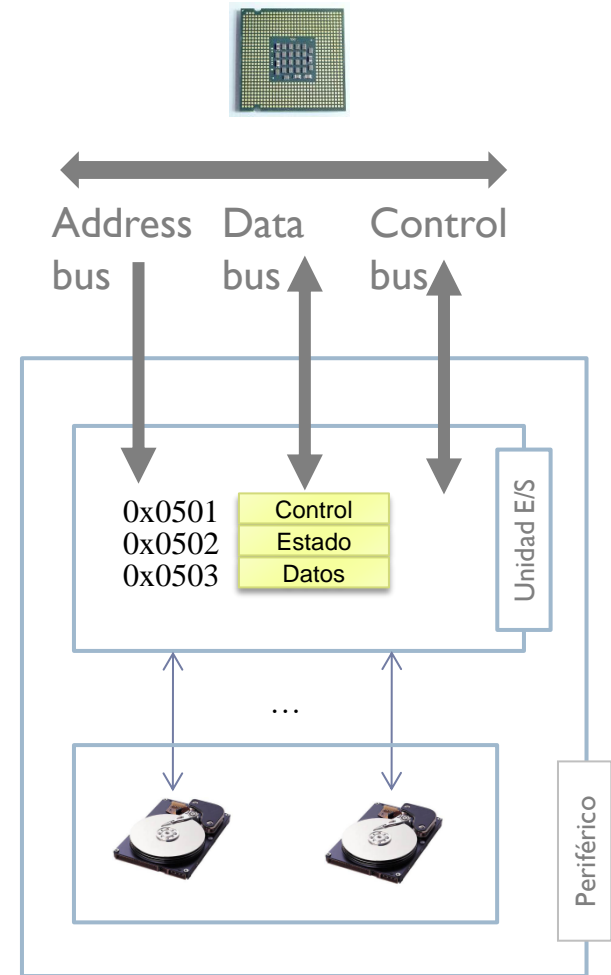
► Ejemplos:

- Encender o apagar
- Saltar página en impresoras
- Posicionar el brazo de un disco
- ...



Módulo de E/S: características

- ▶ **Unidad de transferencia**
 - ▶ Bloque
 - ▶ Caracter
- ▶ **Direccionamiento**
 - ▶ Mapeado a memoria
 - ▶ Mapeado a puerto
- ▶ **Técnicas de Entrada/Salida**
 - ▶ E/S programada
 - ▶ Por interrupciones
 - ▶ Por DMA



Características (1 / 3)

▶ Unidad de transferencia:

▶ **Dispositivos de bloque:**

- ▶ Unidad: **bloque** de bytes
- ▶ Acceso **secuencial** o **directo** a bloques
- ▶ Operaciones: leer, escribir, situarse, ...
- ▶ Ejemplos: discos y “cintas”

▶ **Dispositivos de carácter:**

- ▶ Unidad: **caracteres** (ASCII, Unicode, etc)
- ▶ Acceso **secuencial** a caracteres
- ▶ Operaciones: `get`, `put`,
- ▶ Ejemplo: terminales, impresoras, tarjetas de red

- ▶ **Unidad transferencia**
 - ▶ Bloque
 - ▶ Caracter
- ▶ **Direccionamiento**
 - ▶ Mapeado memoria
 - ▶ Mapeado a puerto
- ▶ **Técnicas de E/S**
 - ▶ Programada
 - ▶ Interrupciones
 - ▶ DMA

Características (2/3)

► Direccionamiento de E/S:

► **Espacio de memoria conjunto o común:**

- Los registros del 'controlador' se proyectan en memoria y usando un conjunto de direcciones de memoria se acceden a dichos registros.
- Ej: `sw a0 etiqueta_discoA`

► **Espacio de memoria separado (puertos):**

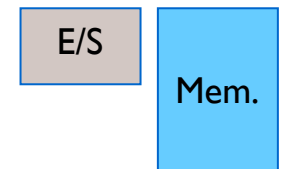
- Con instrucciones ensamblador especiales (in/out) se acceden a las direcciones de E/S (denominadas puertos) que representan los registros del 'controlador'.
- Ej: `out a0 0x105A`

- **Unidad transferencia**
 - Bloque
 - Caracter
- **Direccionamiento**
 - Mapeado memoria
 - Mapeado a puerto
- **Técnicas de E/S**
 - Programada
 - Interrupciones
 - DMA

lw Reg., add.
sw Reg., add.



in Reg., add. lw Reg., add.
out Reg., add. sw Reg., add.



Linux

```

phenix.arcos.inf.uc3m.es - default* - SSH Secure Shell
File Edit View Window Help
acaldero@phenix:~$ cat /proc/ioports
0000-0cf7 : PCI Bus 0000:00
0000-001f : dma1
0020-0021 : pic1
0040-0043 : timer0
0050-0053 : timer1
0060-0060 : keyboard
0064-0064 : keyboard
0070-0073 : rtc0
0080-008f : dma page reg
00a0-00a1 : pic2
00c0-00df : dma2
00f0-00ff : fpu
0290-029f : pnp 00:01
0290-0294 : pnp 00:01
02f8-02ff : serial
0378-037a : parport0
03c0-03df : vga+
03f2-03f2 : floppy
03f4-03f5 : floppy
03f7-03f7 : floppy
03f8-03ff : serial
0400-047f : 0000:00:1f.0
0400-0403 : ACPI PM1a_EVT_BLK
0404-0405 : ACPI PM1a_CNT_BLK
Connected to phoenix.arcos.inf.uc3m.es

```

Windows

Información del sistema

Archivo Editar Ver Ayuda

Resumen del sistema	Recurso	Dispositivo
Recursos de hardware	0x00000000-0x0000000F	Controladora de acceso directo a r
Conflictos/uso compartido	0x00000000-0x0000000F	Bus PCI
DMA	0x00000010-0x0000001F	Recursos de la placa base
Hardware forzado	0x00000020-0x000000...	Controladora programable de inte
E/S	0x00000022-0x0000003F	Recursos de la placa base
IRQs	0x00000040-0x000000...	Cronómetro del sistema
Memoria	0x00000044-0x0000005F	Recursos de la placa base
Componentes	0x00000060-0x000000...	Recursos de la placa base
Entorno de software	0x00000061-0x000000...	Altavoz del sistema
	0x00000062-0x000000...	Recursos de la placa base
	0x00000064-0x000000...	Recursos de la placa base
	0x00000065-0x0000006F	Recursos de la placa base
	0x00000070-0x000000...	Sistema CMOS/reloj en tiempo real
	0x00000072-0x0000007F	Recursos de la placa base
	0x00000080-0x000000...	Recursos de la placa base
	0x00000081-0x000000...	Controladora de acceso directo a r
	0x00000084-0x000000...	Recursos de la placa base
	0x00000087-0x000000...	Controladora de acceso directo a r
	0x00000088-0x000000...	Recursos de la placa base
	0x00000089-0x000000...	Controladora de acceso directo a r
	0x0000008C-0x000000...	Recursos de la placa base

Buscar esto:

☐ Buscar sólo la categoría seleccionada ☐ Buscar sólo nombres de categoría

Características (3/3)

- ▶ Unidad transferencia
 - ▶ Bloque
 - ▶ Caracter
- ▶ Direccionamiento
 - ▶ Mapeado memoria
 - ▶ Mapeado a puerto
- ▶ Técnicas de E/S
 - ▶ Programada
 - ▶ Interrupciones
 - ▶ DMA

- ▶ **Técnicas de E/S:** Interacción Procesador-Controlador
 - ▶ **E/S programada**
 - ▶ **E/S por interrupciones**
 - ▶ **E/S por DMA (acceso directo a memoria)**
- ▶ **A continuación...**

Contenidos

1. Introducción

2. Buses

- ▶ Estructura y funcionamiento
- ▶ Jerarquía de buses

3. Periféricos

- ▶ Concepto y tipos de periféricos
- ▶ Estructura general de un periférico
- ▶ Módulos de E/S

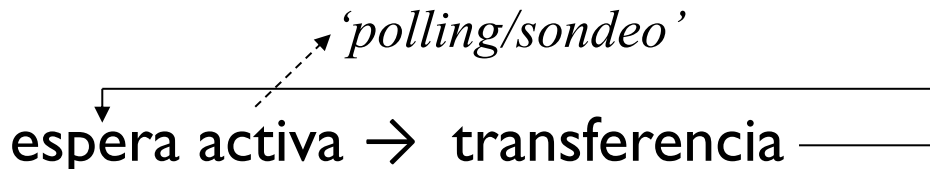
4. Interacción E/S: técnicas de E/S

5. Caso de estudio: disco duro y discos de estado sólido

Características (3/3)

Técnicas de E/S: Interacción CPU-Controlador

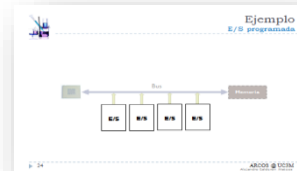
- ▶ Unidad transferencia
 - ▶ Bloque
 - ▶ Caracter
- ▶ Direccionamiento
 - ▶ Mapeado memoria
 - ▶ Mapeado a puerto
- ▶ Técnicas de E/S
 - ▶ Programada
 - ▶ Interrupciones
 - ▶ DMA

- ▶ **E/S programada**
 - ▶ CPU hace la E/S: espera activa → transferencia
- ▶ **E/S por interrupciones**
 - ▶ CPU no espera, solo transfiere datos
- ▶ **E/S por DMA (acceso directo a memoria)**
 - ▶ CPU ni espera ni transfiere, se notifica al final

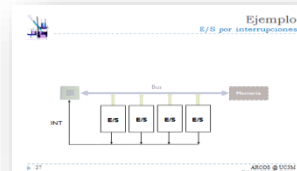
Características (3/3)

Técnicas de E/S: Interacción CPU-Controlador

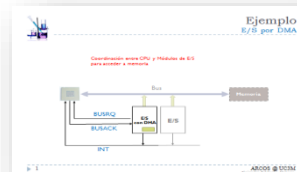
▶ E/S programada



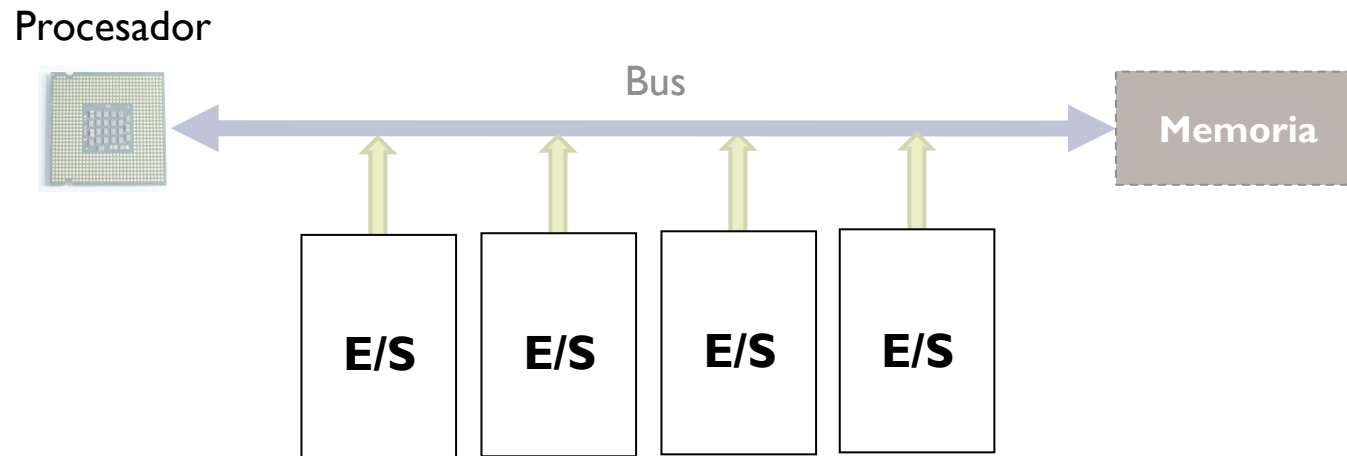
▶ E/S por interrupciones



▶ E/S por DMA



E/S programada



E/S programada: in/out vs lw/sw

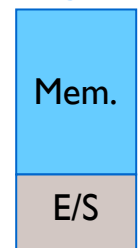
- ▶ La transferencia entre el procesador (o memoria) y la unidad de E/S se realiza bajo el control del procesador con instrucciones máquinas especiales privilegiadas: **instrucciones máquina de E/S**

- ▶ **Mapa memoria conjunto o común (ej. RISC-V):**

- ▶ Los registros del ‘controlador’ se proyectan en memoria y usando un conjunto de direcciones de memoria se acceden a dichos registros.

- ❑ Escribir en controlador: **sw** a0 etiqueta_discoA
 - ❑ Leer de controlador: **lw** a0 etiqueta_discoA

lw Reg., add.
sw Reg., add.

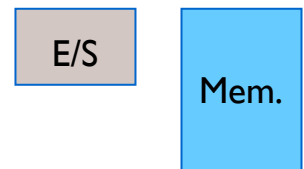


- ▶ **Mapa de memoria separado (ej. Intel x86):**

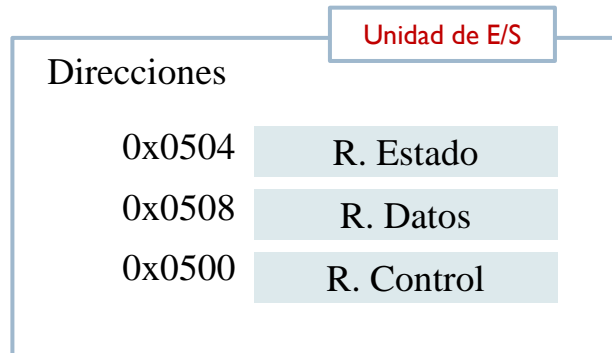
- ▶ Con instrucciones ensamblador especiales se acceden a las direcciones de E/S (denominadas puertos) que representan los registros del ‘controlador’.

- ❑ Escribir en controlador: **out** a0 0x105A
 - ❑ Leer de controlador: **in** a0 0x1050

in Reg., add. lw Reg., add.
out Reg., add. sw Reg., add.



Ejemplo

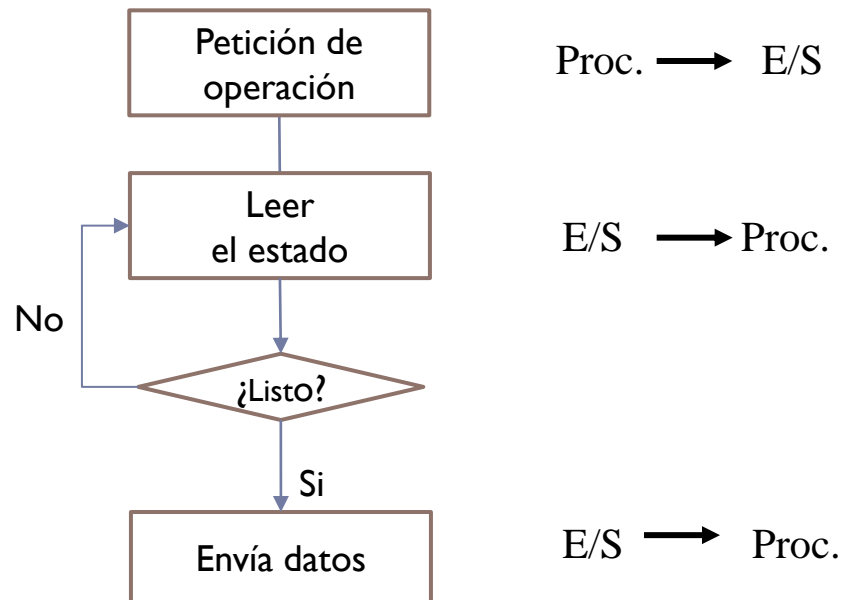


- ▶ Información de **control**
 - ▶ 0: leer
 - ▶ 1: escribir
- ▶ Información de **estado**
 - ▶ 0: dispositivo ocupado
 - ▶ 1: dispositivo (dato) listo
- ▶ Mapa de E/S común
 - ▶ lw y sw de RISC-V

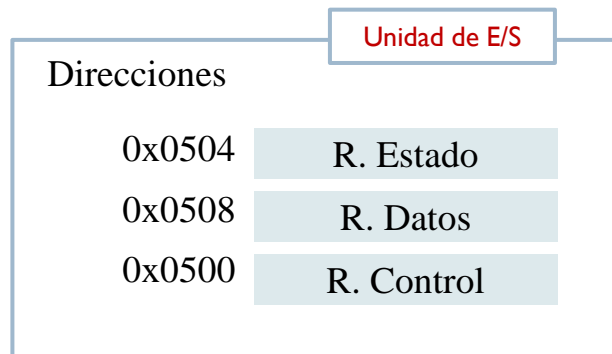
- ▶ ¿Instrucciones para escribir un 1 en el registro 0x0508 (de datos)?

```
li t0, 1
la t1, 0x0508
sw t0, 0(t1)
```

E/S programada: interacción



Ejemplo



- ▶ Información de **control**
 - ▶ 0: leer
 - ▶ 1: escribir
- ▶ Información de **estado**
 - ▶ 0: dispositivo ocupado
 - ▶ 1: dispositivo (dato) listo
- ▶ Mapa de E/S común
 - ▶ lw y sw del RISC-V

▶ ¿Operaciones para **leer** un dato?

1. Enviar la orden

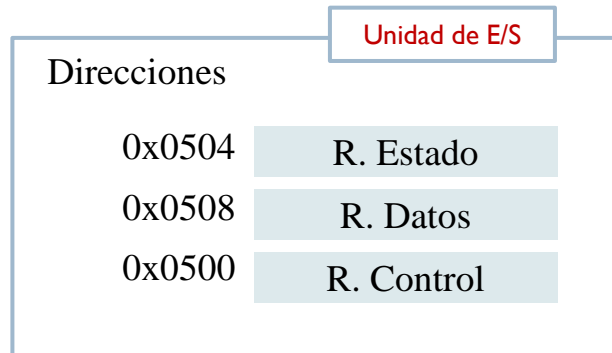
```
li t0, 0
la t1, 0x500
sw t0, 0(t1)
```
2. Leer el estado

```
b1: la t1, 0x0504
lw t0, 0(t1)
```
3. Comprobar el estado

```
beq t0, x0, b1
```
4. Leer el dato

```
la t1, 0x0508
lw t0, 0(t1)
```

Ejemplo



- ▶ Información de **control**
 - ▶ 0: leer
 - ▶ 1: escribir
- ▶ Información de **estado**
 - ▶ 0: dispositivo ocupado
 - ▶ 1: dispositivo (dato) listo
- ▶ Mapa de E/S común
 - ▶ lw y sw del RISC-V

▶ ¿Operaciones para **escribir** un dato?

1. Enviar el dato

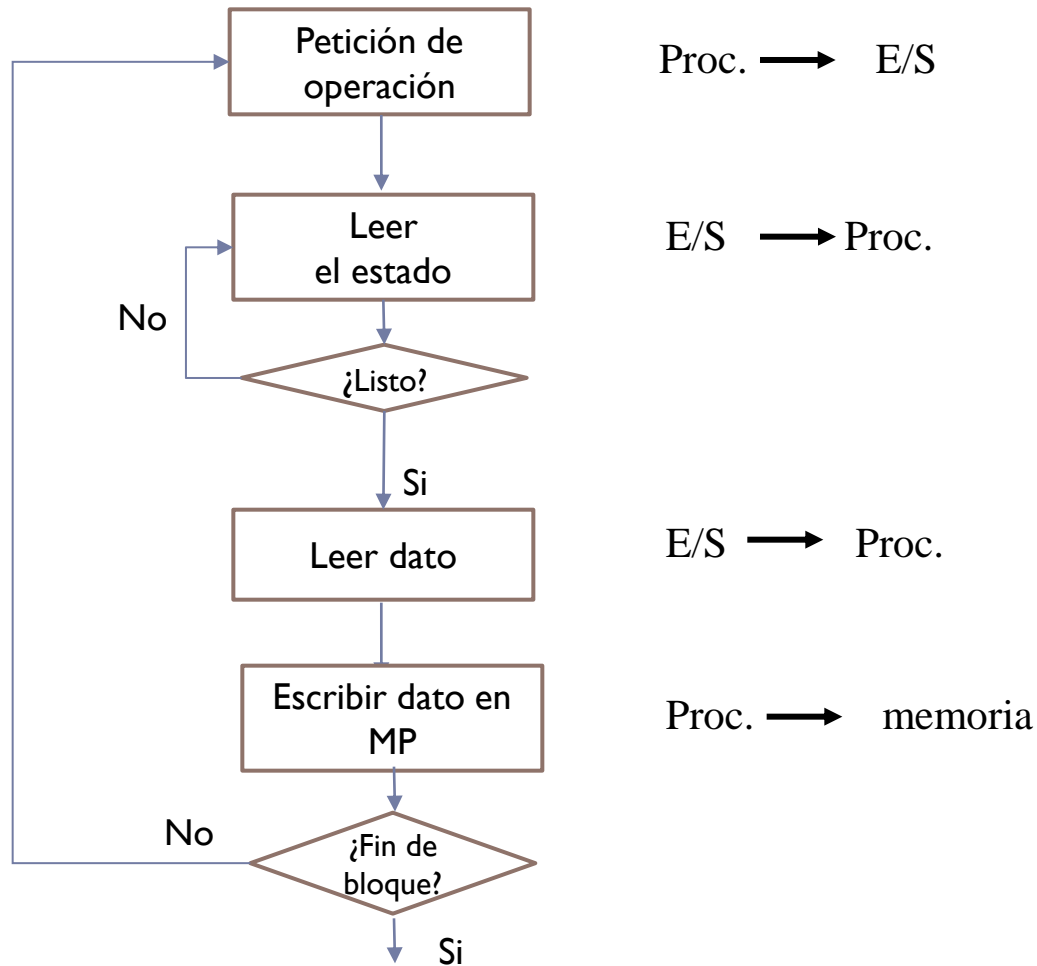
```
li t0, 123
la t1, 0x0508
sw t0, 0(t1)
```
2. Enviar la orden

```
li t0, 1
la t1, 0x500
sw t0, 0(t1)
```
3. Leer estado

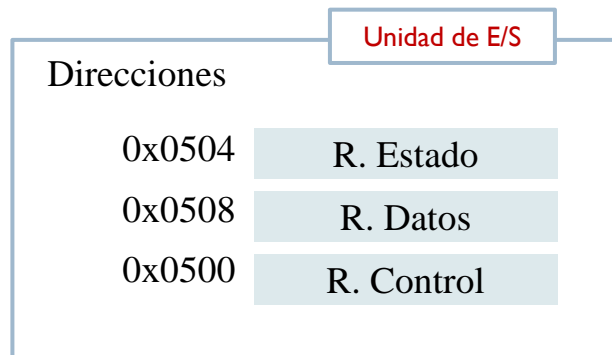
```
b1: la t0, 0x0504
lw t0, 0(t0)
```
4. Comprobar estado

```
beq t0, x0, b1
```

Lectura de un bloque de datos



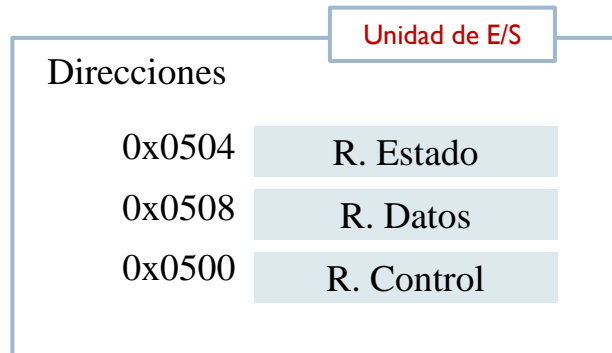
Ejercicio



- ▶ Información de control
 - ▶ 0: leer
 - ▶ 1: escribir
- ▶ Información de estado
 - ▶ 0: dispositivo ocupado
 - ▶ 1: dispositivo (dato) listo
- ▶ Mapa de E/S común
 - ▶ Instrucciones lw y sw

Codifique un programa en ensamblador que lee 100 datos usando la unidad de E/S descrita, y los almacena en la dirección de memoria principal dada por la etiqueta 'datos'.

Ejercicio (solución)

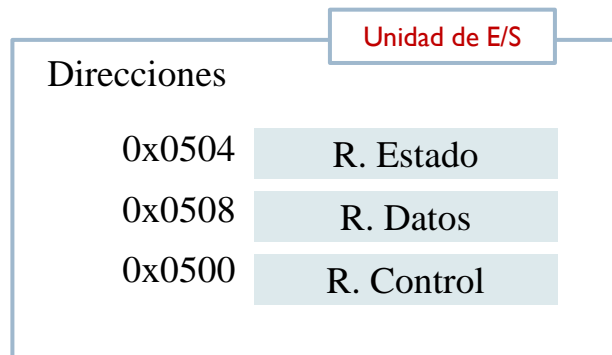


- ▶ Información de control
 - ▶ 0: leer
 - ▶ 1: escribir
- ▶ Información de estado
 - ▶ 0: dispositivo ocupado
 - ▶ 1: dispositivo (dato) listo
- ▶ Mapa de E/S común
 - ▶ Instrucciones lw y sw

```
.data
    datos: .zero 400

.text
main:    li    t3 0
         li    t4 400
bucle1:  la    t0 0x500
         sw    x0 0(t0)
         mv    t0 x0
bucle2:  la    t1 0x504
         lw    t1 0(t1)
         beq   t1 x0 bucle2
         la    t2 0x508
         lw    t2 0(t2)
         sw    t2 datos(t3)
         addi  t3 t3 4
         bne   t3 t4 bucle1
```

Ejercicio (solución)



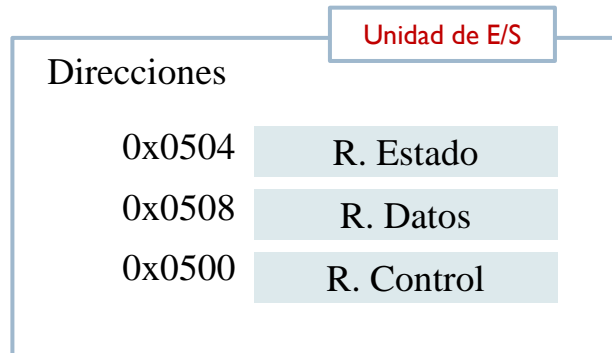
- ▶ Información de control
 - ▶ 0: leer
 - ▶ 1: escribir
- ▶ Información de estado
 - ▶ 0: dispositivo ocupado
 - ▶ 1: dispositivo (dato) listo
- ▶ Mapa de E/S común
 - ▶ Instrucciones lw y sw

```
.data
    datos: .zero 400

.text
main:    li    t3 0
         li    t4 400
bucle1:  la    t0 0x500
         sw    x0 0(t0)
         mv    t0 x0
bucle2:  la    t1 0x504
         lw    t1 0(t1)
         beq   t1 x0 bucle2
         la    t2 0x508
         lw    t2 0(t2)
         sw    t2 datos(t3)
         addi  t3 t3 4
         bne   t3 t4 bucle1
```

} Bucle de sincronización

Ejercicio (solución)



- ▶ Información de control
 - ▶ 0: leer
 - ▶ 1: escribir
- ▶ Información de estado
 - ▶ 0: dispositivo ocupado
 - ▶ 1: dispositivo (dato) listo
- ▶ Mapa de E/S común
 - ▶ Instrucciones lw y sw

```
.data
    datos: .zero 400

.text
main:    li    t3 0
         li    t4 400
bucle1:  la    t0 0x500
         sw    x0 0(t0)
         mv    t0 x0
bucle2:  la    t1 0x504
         lw    t1 0(t1)
         beq   t1 x0 bucle2
         la    t2 0x508
         lw    t2 0(t2)
         sw    t2 datos(t3)
         addi  t3 t3 4
         bne   t3 t4 bucle1
```

Bucle de sincronización

Bucle de transferencia

Ejercicio

- ▶ Sea un computador con la capacidad de ejecutar 200 millones de instrucciones por segundo (200 MIPS)
- ▶ Se conecta el módulo de E/S anteriormente descrito siendo el tiempo medio de espera de lectura de 5 ms
- ▶ Calcule cuantas instrucciones se ejecutan en el bucle de sincronización y en el bucle de transferencia para el programa mostrado

```
.data
    datos: .zero 400

.text
main:    li    t3 0
         li    t4 400
bucle1:  la    t0 0x500
         sw    x0 0(t0)
         mv    t0 x0
bucle2:  la    t1 0x504
         lw    t1 0(t1)
         beq   t1 x0 bucle2
         la    t2 0x508
         lw    t2 0(t2)
         sw    t2 datos(t3)
         addi   t3 t3 4
         bne   t3 t4 bucle1
```

Bucle de sincronización

Bucle de transferencia

Ejercicio (solución)

- ▶ Bucle de sincronización:
 - ▶ En media dura 5 ms
 - ▶ Se ejecuta 200 MIPS en media
 - ▶ $I_{bs} = 200 * 10^6 * 5 * 10^{-3} = 10^6$
- ▶ Bucle de transferencia:
 - ▶ $2 (li+li) + (11-3) * 100 + 10^6 (I_{bs})$
- ▶ Como puede comprobarse, en el bucle se ejecuta 1.000.802 instrucciones, de las cuales 1.000.000 corresponden al bucle de espera (el 99,9%)
 - ▶ Es un desperdicio de ciclos del procesador
 - ▶ El procesador no realiza trabajo útil

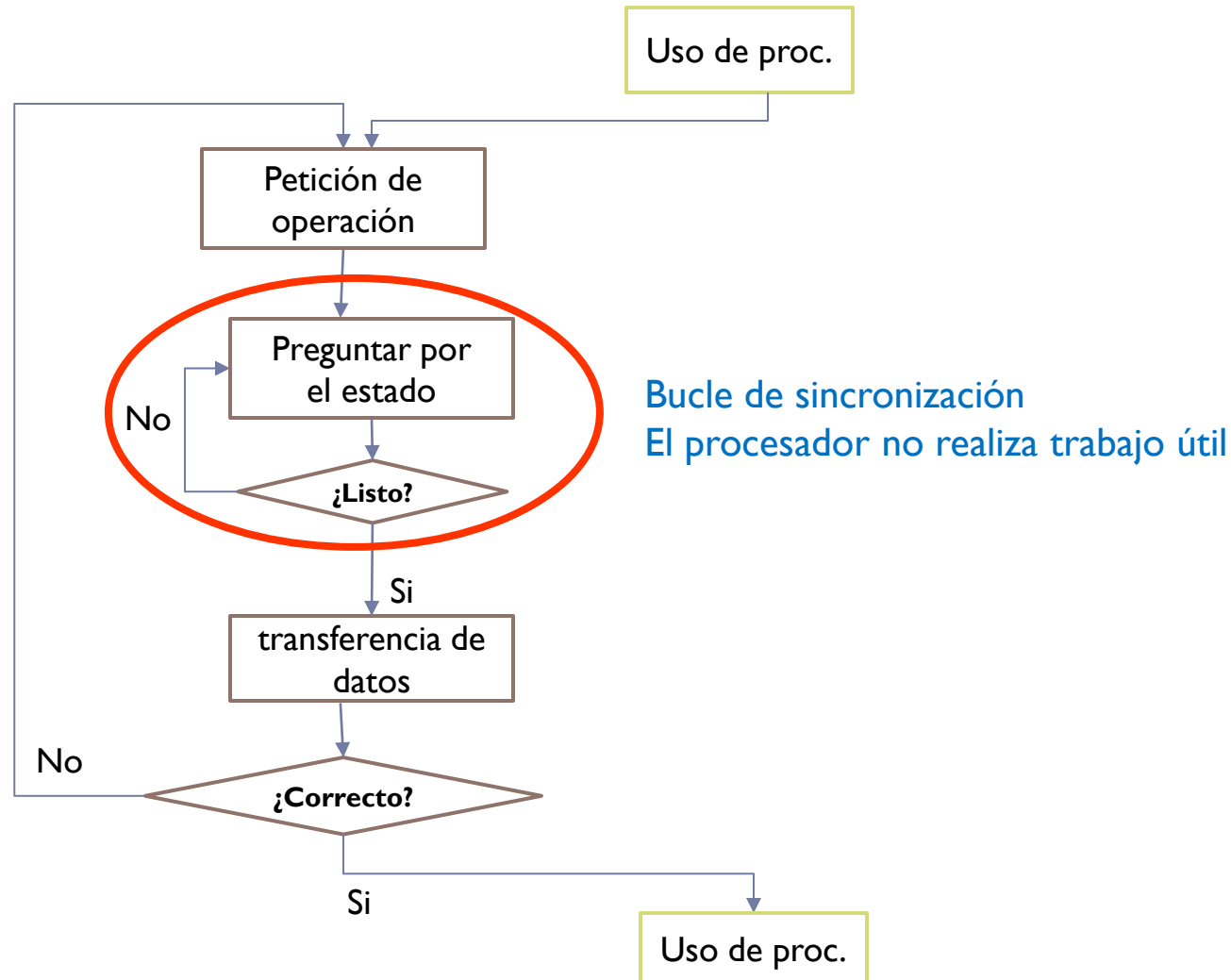
```
.data
    datos: .zero 400

.text
main:    li    t3 0
         li    t4 400
bucle1:  la    t0 0x500
         sw    x0 0(t0)
         mv    t0 x0
bucle2:  la    t1 0x504
         lw    t1 0(t1)
         beq   t1 x0 bucle2
         la    t2 0x508
         lw    t2 0(t2)
         sw    t2 datos(t3)
         addi   t3 t3 4
         bne   t3 t4 bucle1
```

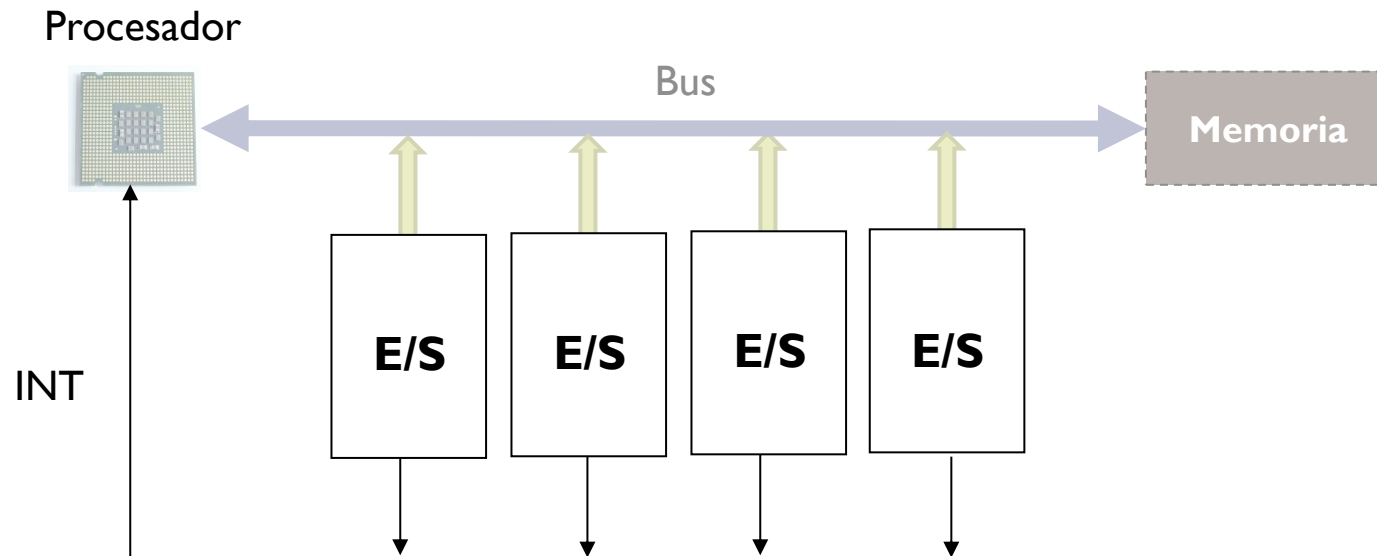
Bucle de sincronización

Bucle de transferencia

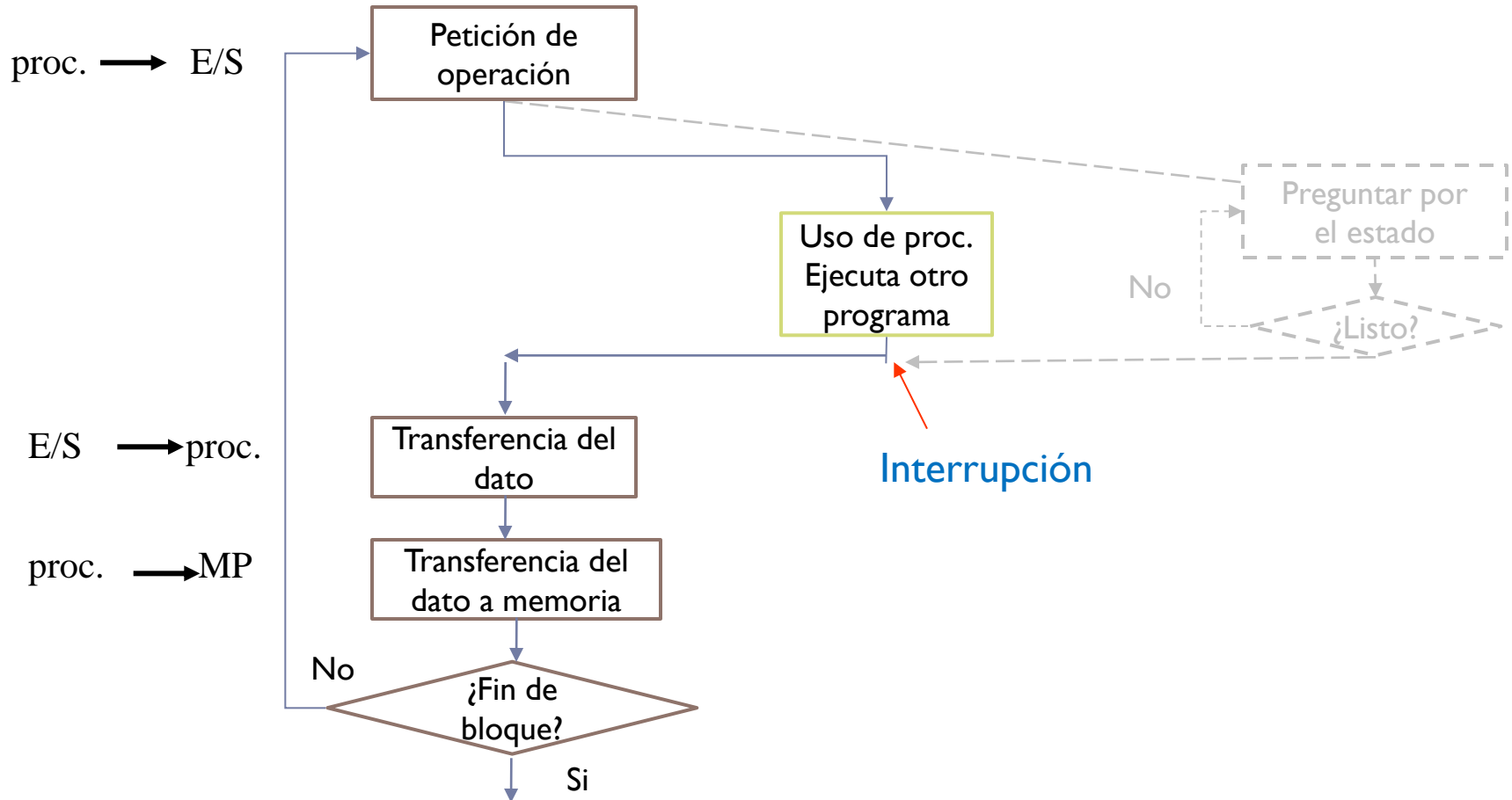
Problema de la E/S programada



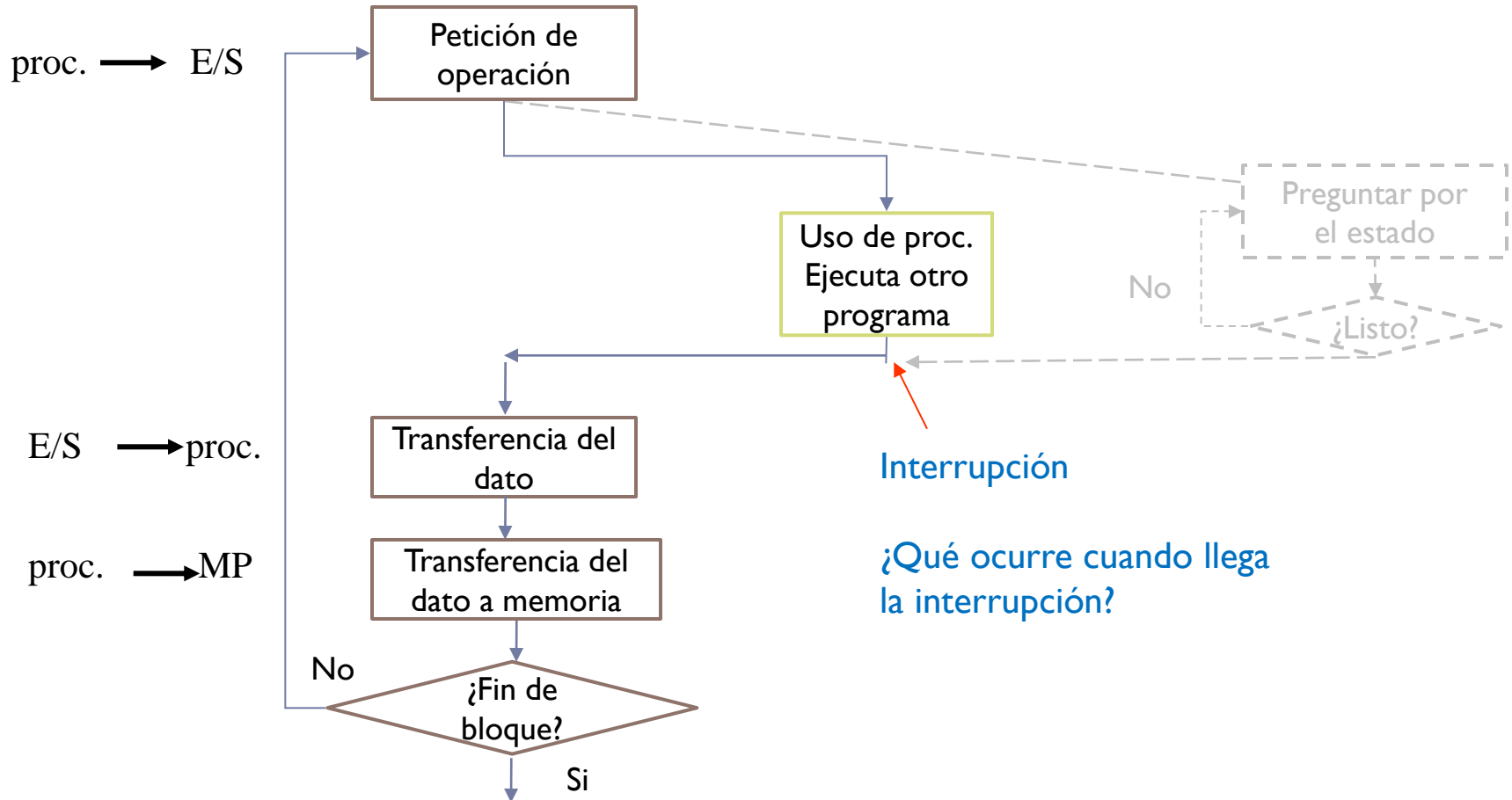
E/S mediante interrupciones



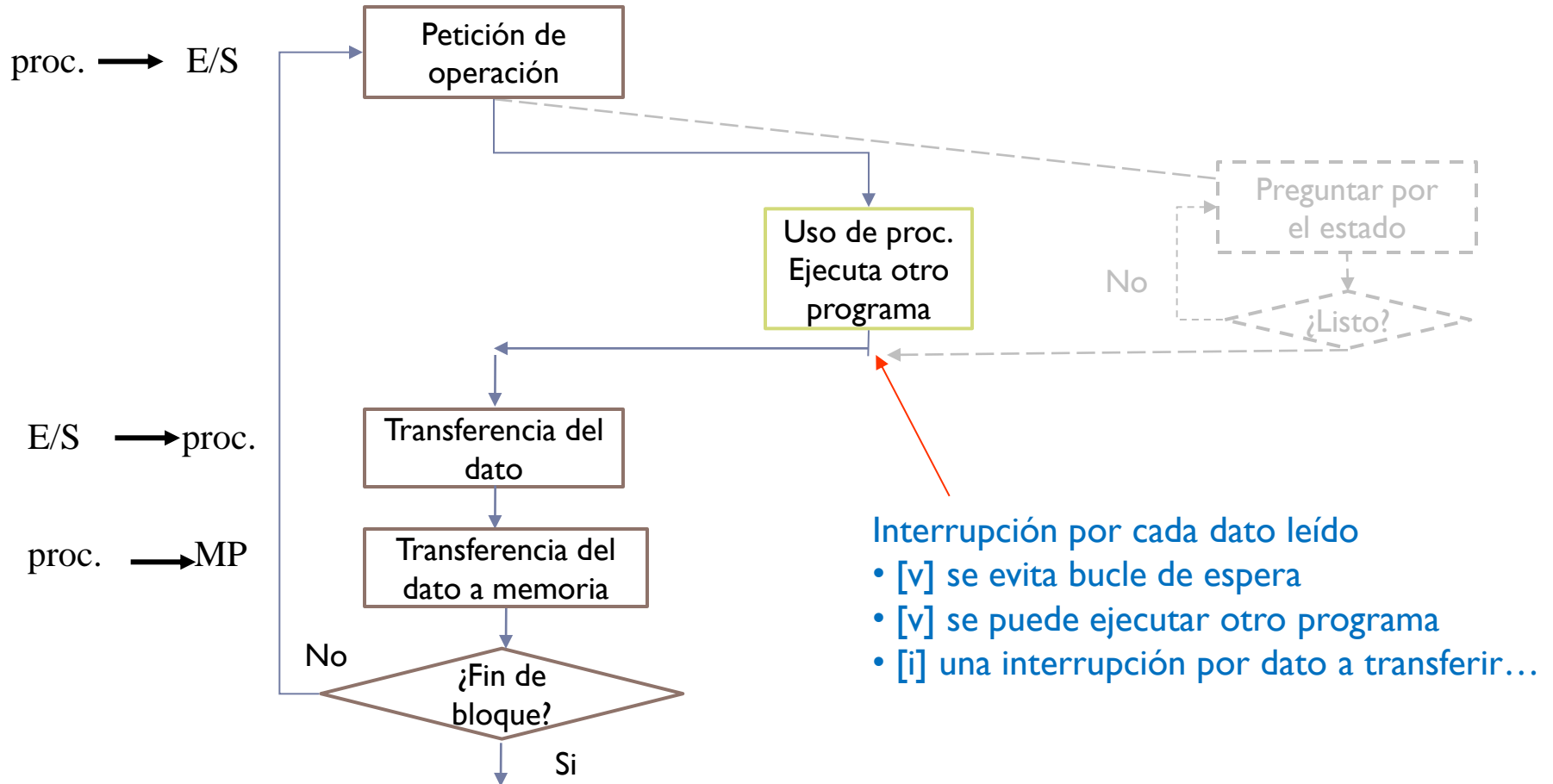
E/S mediante interrupciones



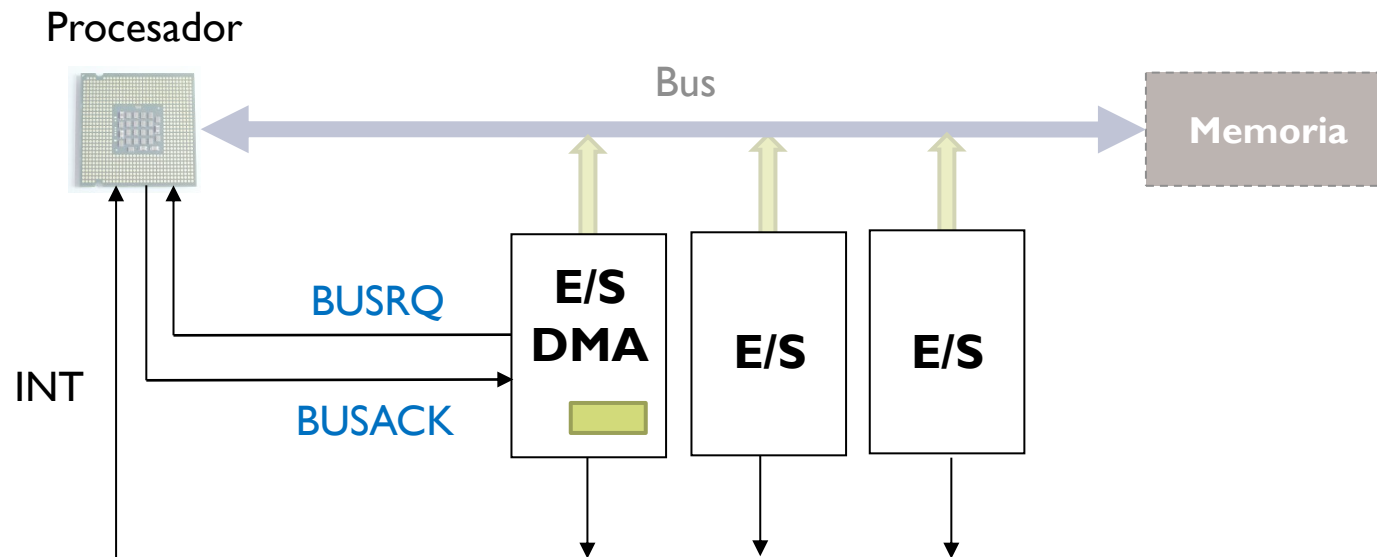
E/S mediante interrupciones



E/S mediante interrupciones

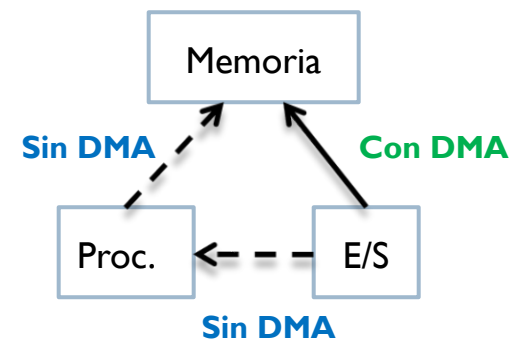


E/S por DMA



E/S mediante DMA

- ▶ **DMA (*Direct Memory Access*):**
Acceso directo a memoria
- ▶ La transferencia entre la unidad de E/S y la memoria:
 - ▶ Con interrupciones se evita el bucle de espera pero la transferencia la lleva a cabo el procesador
 - ▶ Con DMA toda la transferencia la realiza la unidad de E/S
 - ▶ El procesador no realiza la transferencia entre la unidad de E/S y la memoria
- ▶ **Interrupciones generadas:**
 - ▶ Con interrupciones para un bloque de N bytes se generan N interrupciones
 - ▶ Con DMA solo **una** interrupción al final



Transferencia de un bloque mediante DMA

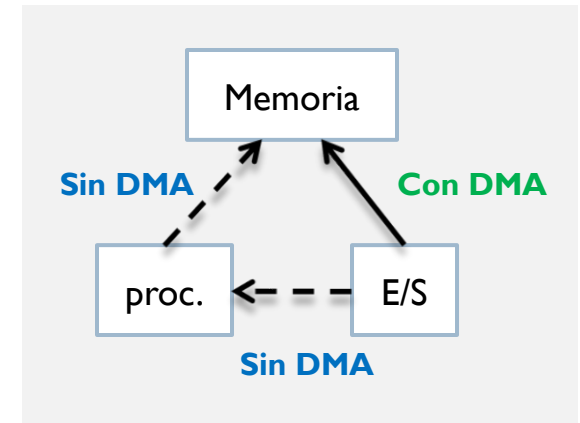
Proc. → E/S

Petición de
operación

Uso de proc.
Ejecuta otro
programa

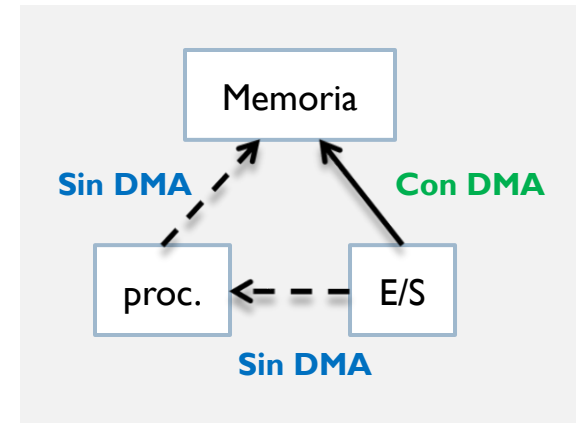
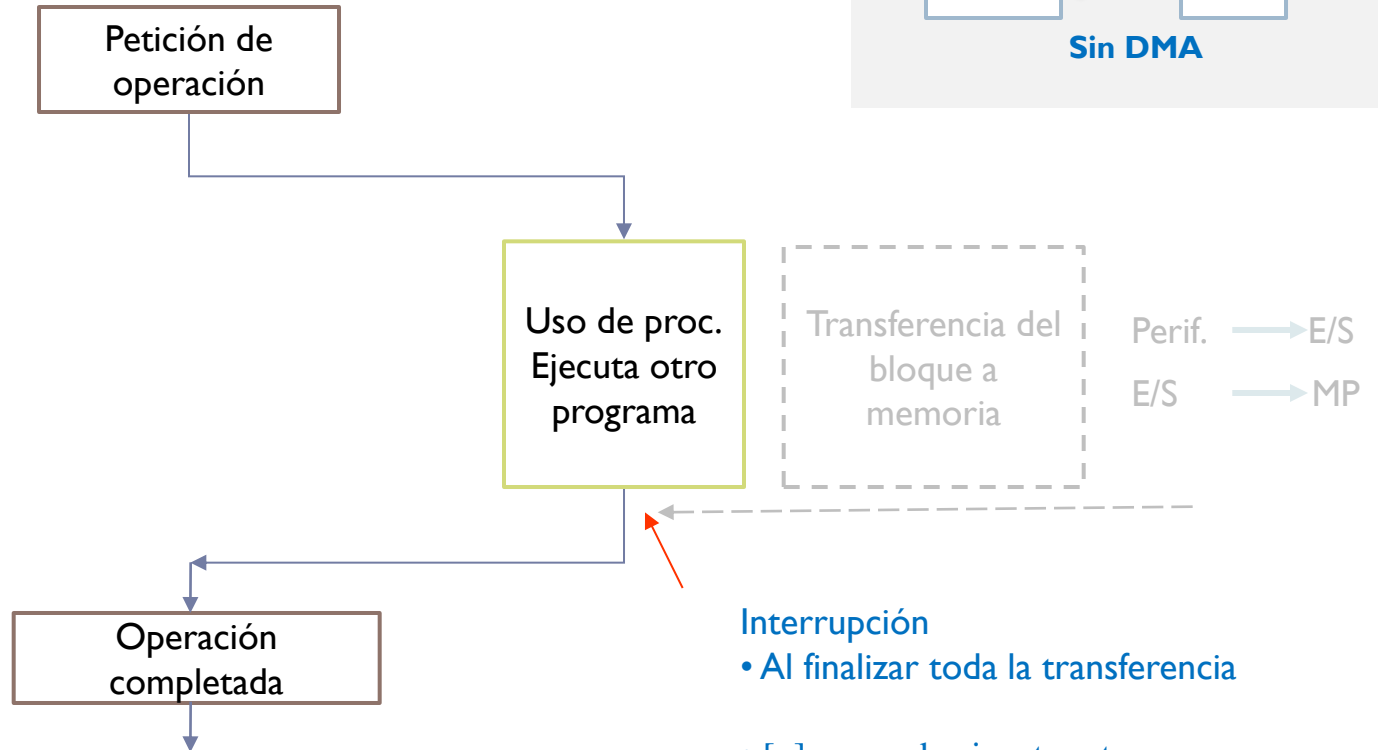
Transferencia del
bloque a
memoria

Perif. → E/S
E/S → MP



Transferencia de un bloque mediante DMA

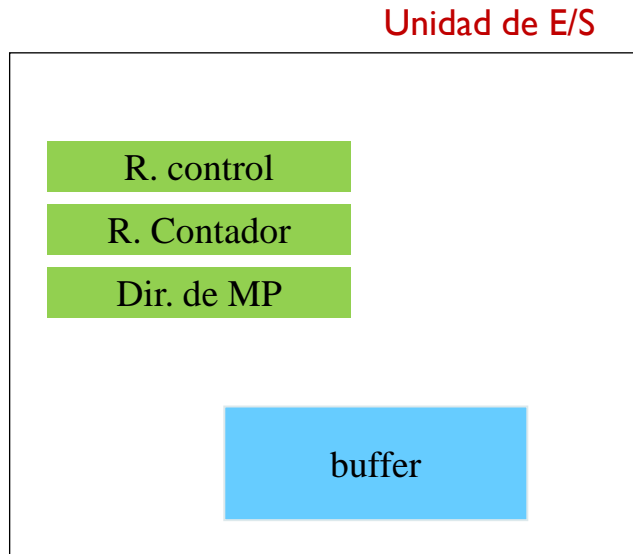
Proc. → E/S



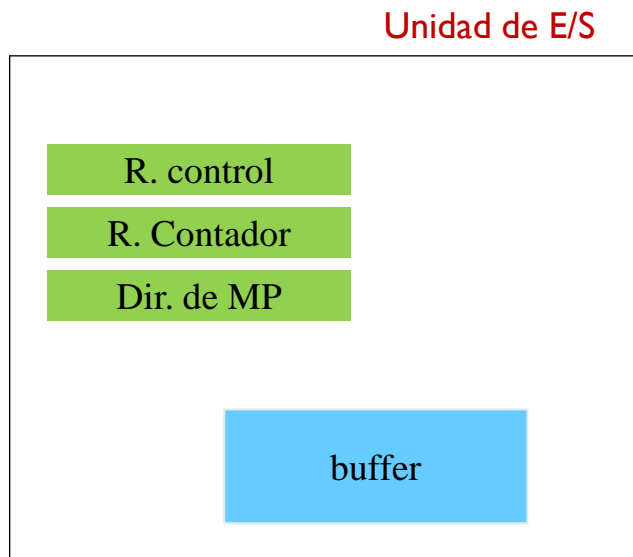
Interrupción

- Al finalizar toda la transferencia
- [v] se puede ejecutar otro programa
- [v] una única interrupción

Estructura simplificada de un módulo de E/S para DMA



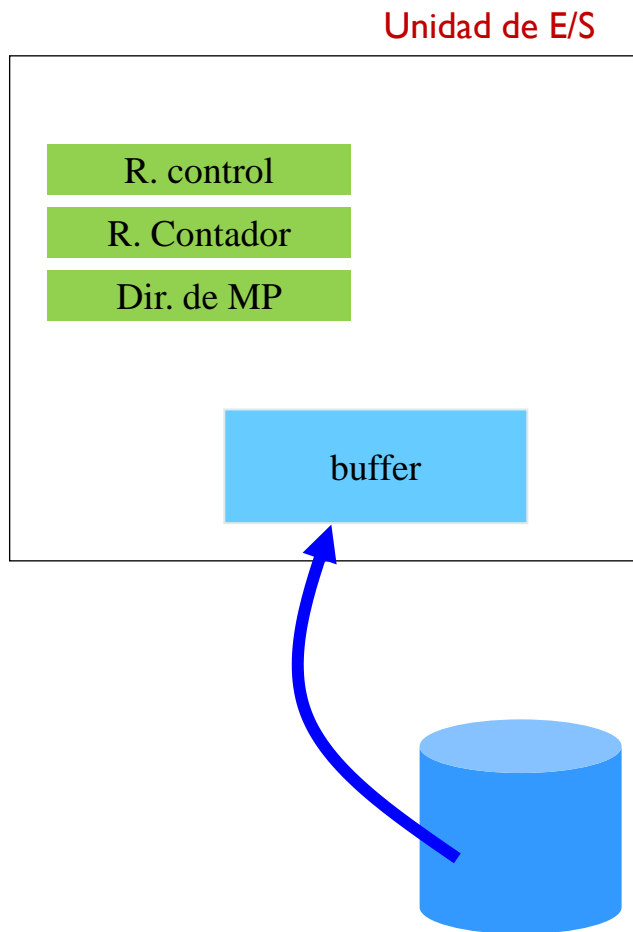
Transferencia con DMA



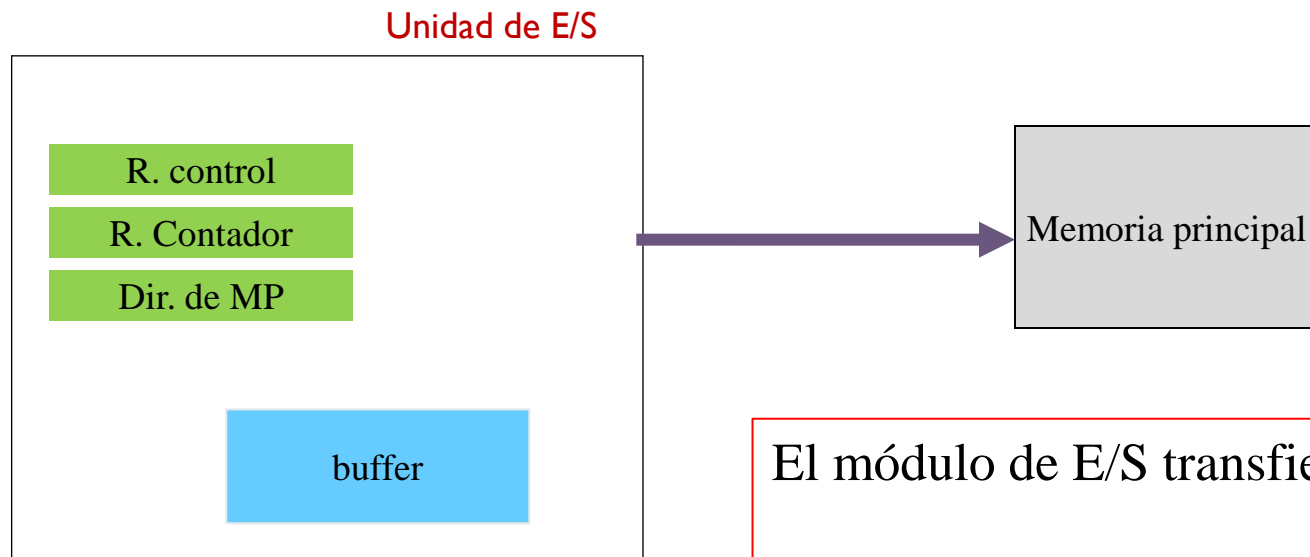
- ▶ El procesador escribe en los registros (con instrucciones de E/S)
 - ▶ La operación (**R. control**)
 - ▶ Lectura, escritura
 - ▶ El número de bytes a transferir (**R. contador**)
 - ▶ La **dirección de memoria principal** donde
 - ▶ Se almacenan los datos (escritura al periférico)
 - ▶ Almacenar los datos (lectura del periférico)

Transferencia con DMA

- ▶ La unidad de E/S transfiere todo el bloque de datos del periférico al buffer interno de la unidad de E/S (para lectura)



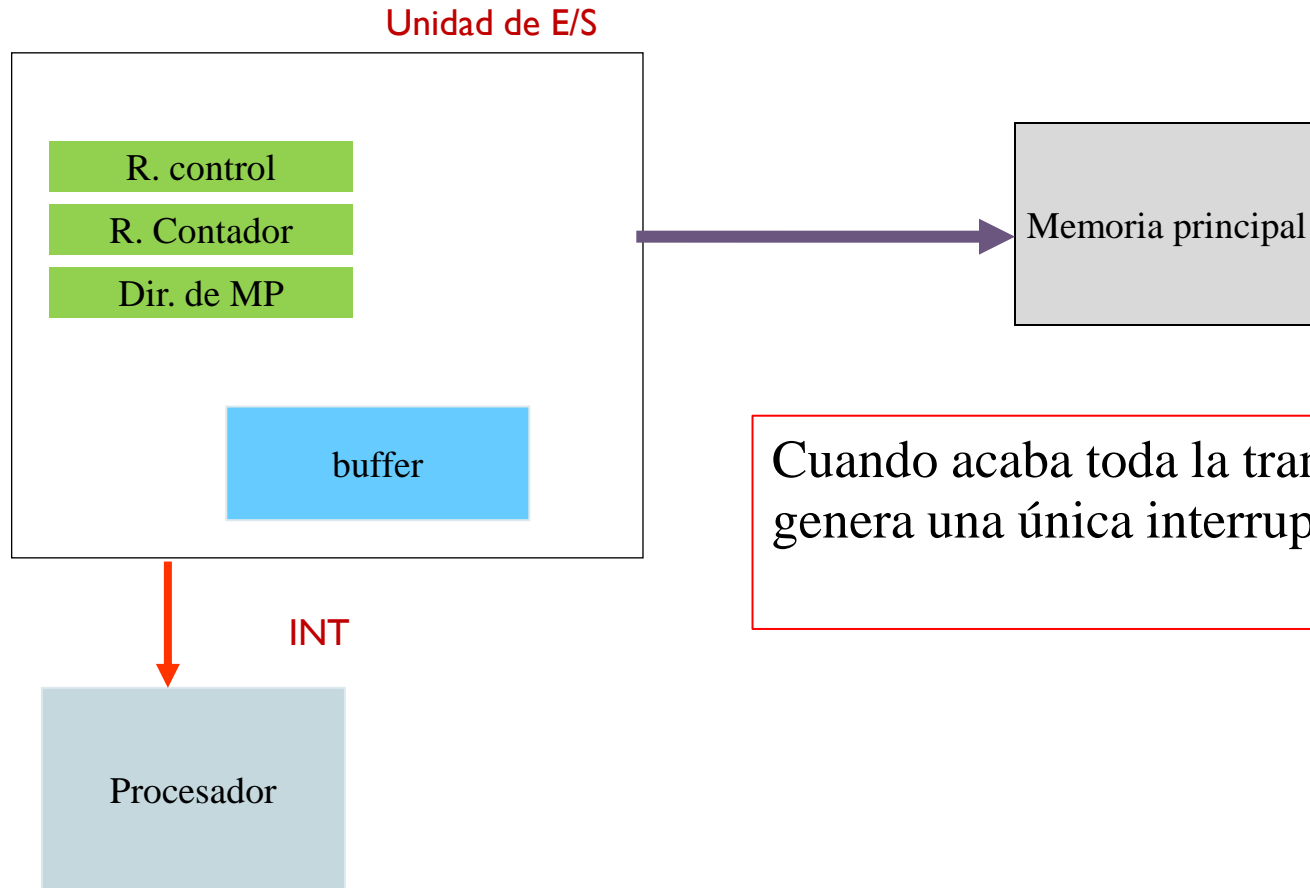
Transferencia con DMA



El módulo de E/S transfiere el bloque

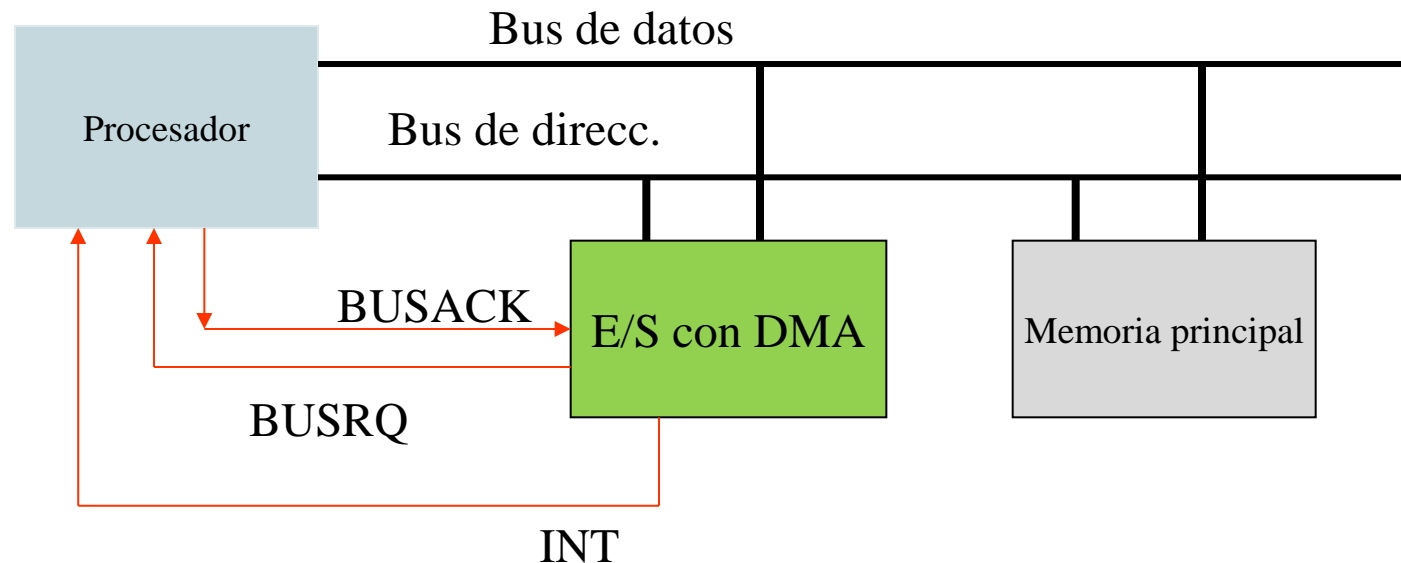
```
while (contador > 0)
{
    Byte (palabra) -> MP[Dir de MP]
    Dir de MP ++;
    contador --;
}
```

Transferencia con DMA



Cuando acaba toda la transferencia se genera una única interrupción

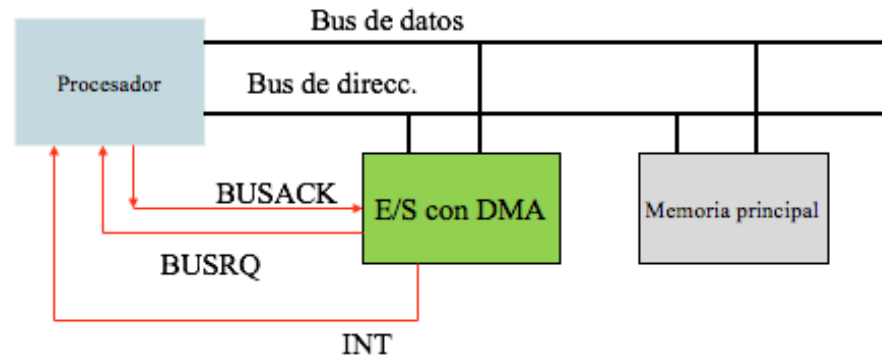
Acceso del módulo de E/S a MP



- Hay que coordinar el acceso a memoria entre el procesador y el módulo de E/S -> BUSRQ + BUSACK

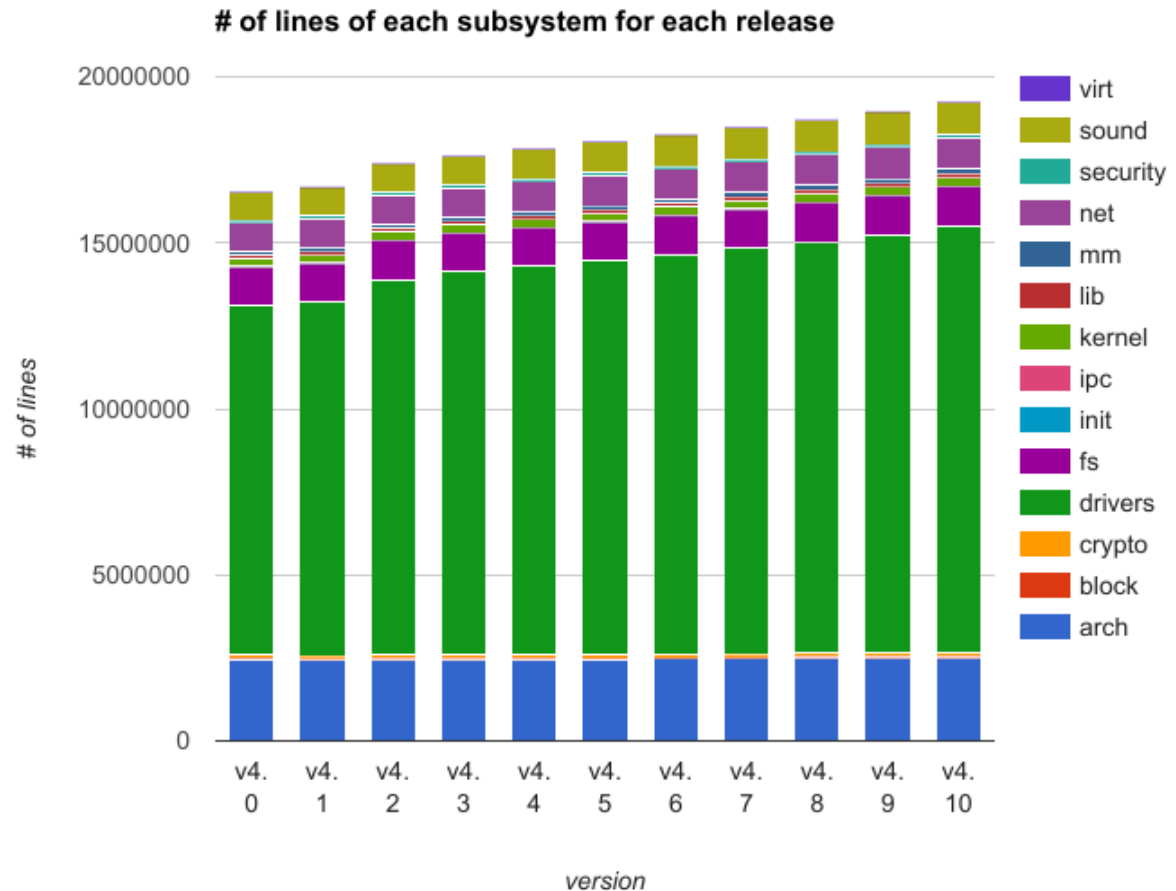
Acceso del módulo de E/S a MP:

Robo de ciclo



- ▶ Cuando la unidad de E/S está lista para transferir un dato:
 - ▶ Activa la señal **BUSRQ** para solicitar el acceso a los buses
 - ▶ Al final de cada fase de una instrucción el procesador comprueba la señal **BUSRQ**. Si está activa libera los buses y activa **BUSACK**
 - ▶ La unidad de E/S accede a MP y a continuación desactiva **BUSRQ**
 - ▶ El procesador continúa
 - ▶ La interrupción se genera al final

Curiosidad: Importancia de los controladores de Kernel de Linux



- El 70% del código de Linux son los controladores de dispositivo (o drivers)

Grupo ARCOS

uc3m | Universidad **Carlos III** de Madrid

Tema 6: E/S y dispositivos periféricos

Estructura de Computadores

Grado en Ingeniería Informática

Grado en Matemática aplicada y Computación

Doble Grado en Ingeniería Informática y Administración de Empresas

