



Tema 6

E/S y dispositivos periféricos

Grupo ARCOS

Estructura de Computadores
Grado en Ingeniería Informática
Universidad Carlos III de Madrid

Contenidos

- ❑ **Introducción**
- ❑ **Buses**
 - ❑ Estructura y funcionamiento
 - ❑ Jerarquía de buses
- ❑ **Periféricos**
 - ❑ Concepto de periférico
 - ❑ Clasificación y tipos de periféricos
 - ❑ Estructura general de un periférico
 - ❑ Caso de estudio: disco duro
- ❑ **Módulos de E/S**
 - ❑ Estructura
 - ❑ Funcionamiento general
 - ❑ Características

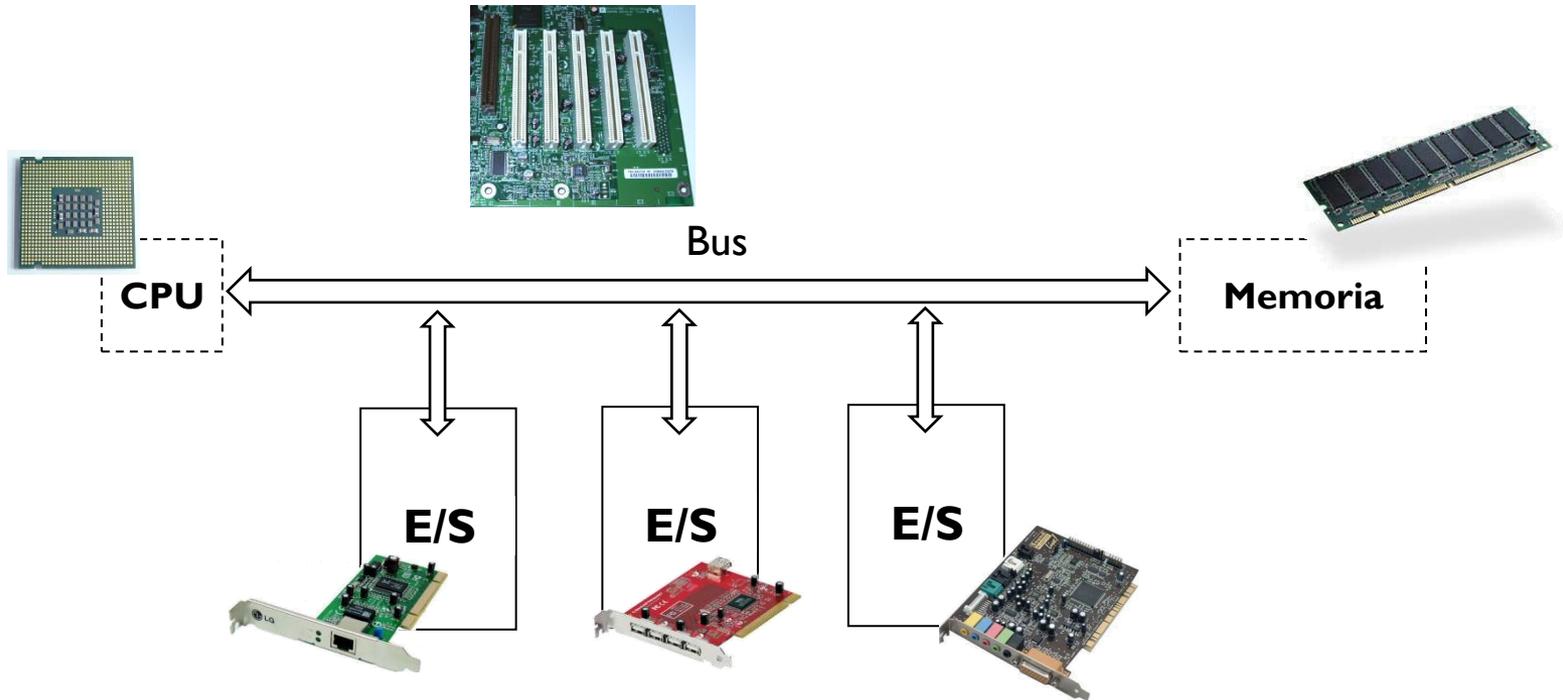
¡ATENCIÓN!

- ❑ Estas transparencias son un guión para la clase
- ❑ Los libros dados en la bibliografía junto con lo explicado en clase representa el material de estudio para el temario de la asignatura
- ❑ Para la preparación de los exámenes se ha de utilizar todo el material de estudios

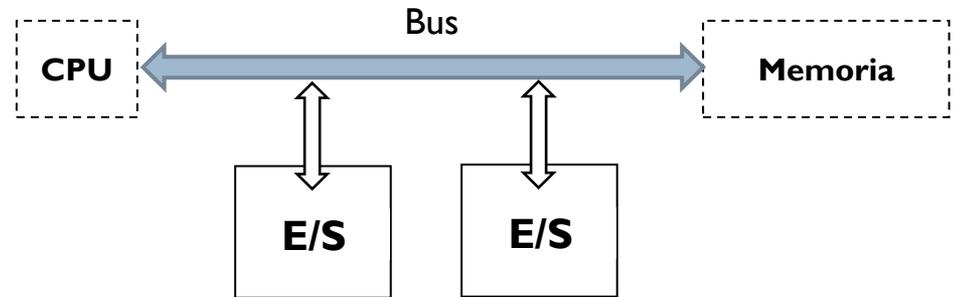
Contenidos

- ❑ **Introducción**
- ❑ **Buses**
 - ❑ Estructura y funcionamiento
 - ❑ Jerarquía de buses
- ❑ **Periféricos**
 - ❑ Concepto de periférico
 - ❑ Clasificación y tipos de periféricos
 - ❑ Estructura general de un periférico
 - ❑ Caso de estudio: disco duro
- ❑ **Módulos de E/S**
 - ❑ Estructura
 - ❑ Funcionamiento general
 - ❑ Características

Introducción

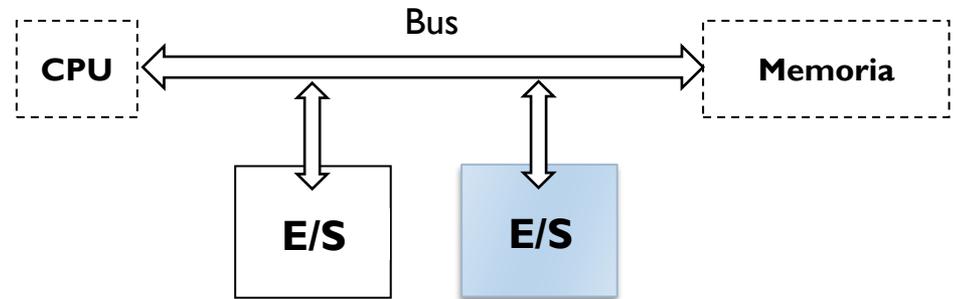


Introducción



- ▶ Qué es un bus de interconexión

Introducción

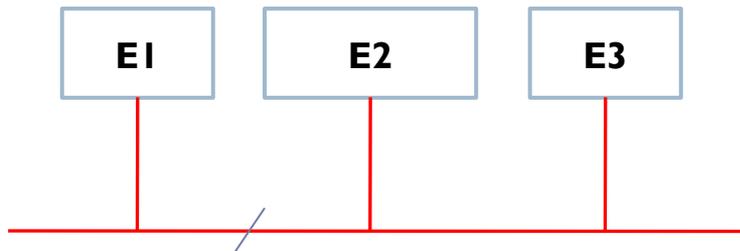


- ▶ Qué es un periférico
- ▶ Qué es un módulo de entrada/salida

Contenidos

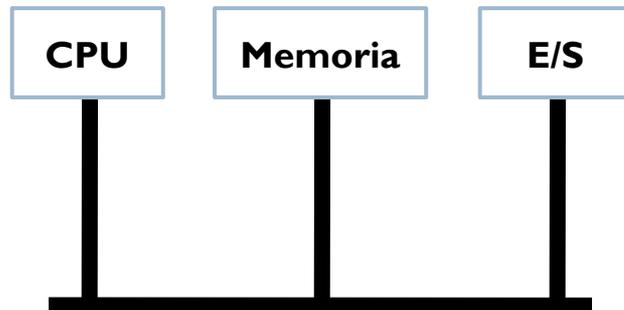
- ❑ Introducción
- ❑ **Buses**
 - ❑ Estructura y funcionamiento
 - ❑ Jerarquía de buses
- ❑ Periféricos
 - ❑ Concepto de periférico
 - ❑ Clasificación y tipos de periféricos
 - ❑ Estructura general de un periférico
 - ❑ Caso de estudio: disco duro
- ❑ Módulos de E/S
 - ❑ Estructura
 - ❑ Funcionamiento general
 - ❑ Características

Bus



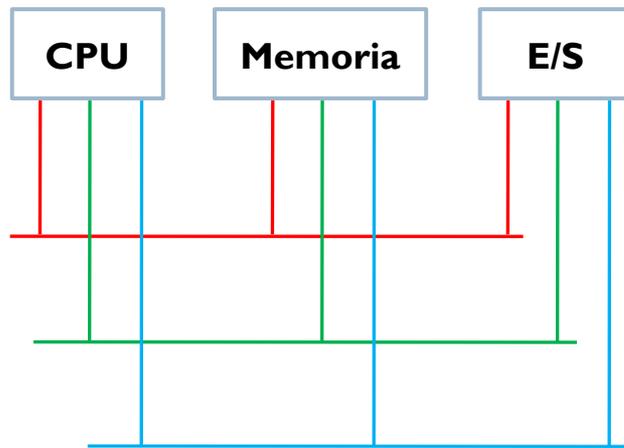
- ▶ Un bus es un **camino** de comunicación **entre dos o más dispositivos**.
- ▶ **Constituido** por **varias líneas** de transmisión de bit.
- ▶ **Medio compartido**, unívoco.
- ▶ Permite transmitir varios bits entre dos elementos conectados a él

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

Comunicación entre módulos

- ▶ **Envío de datos** de un módulo a otro:

- 1º) Obtener el uso del bus.

- 2º) Transferir el dato a través de las líneas de datos del bus.

- ▶ **Petición de datos** a otro módulo:

- 1º) Obtener el uso del bus.

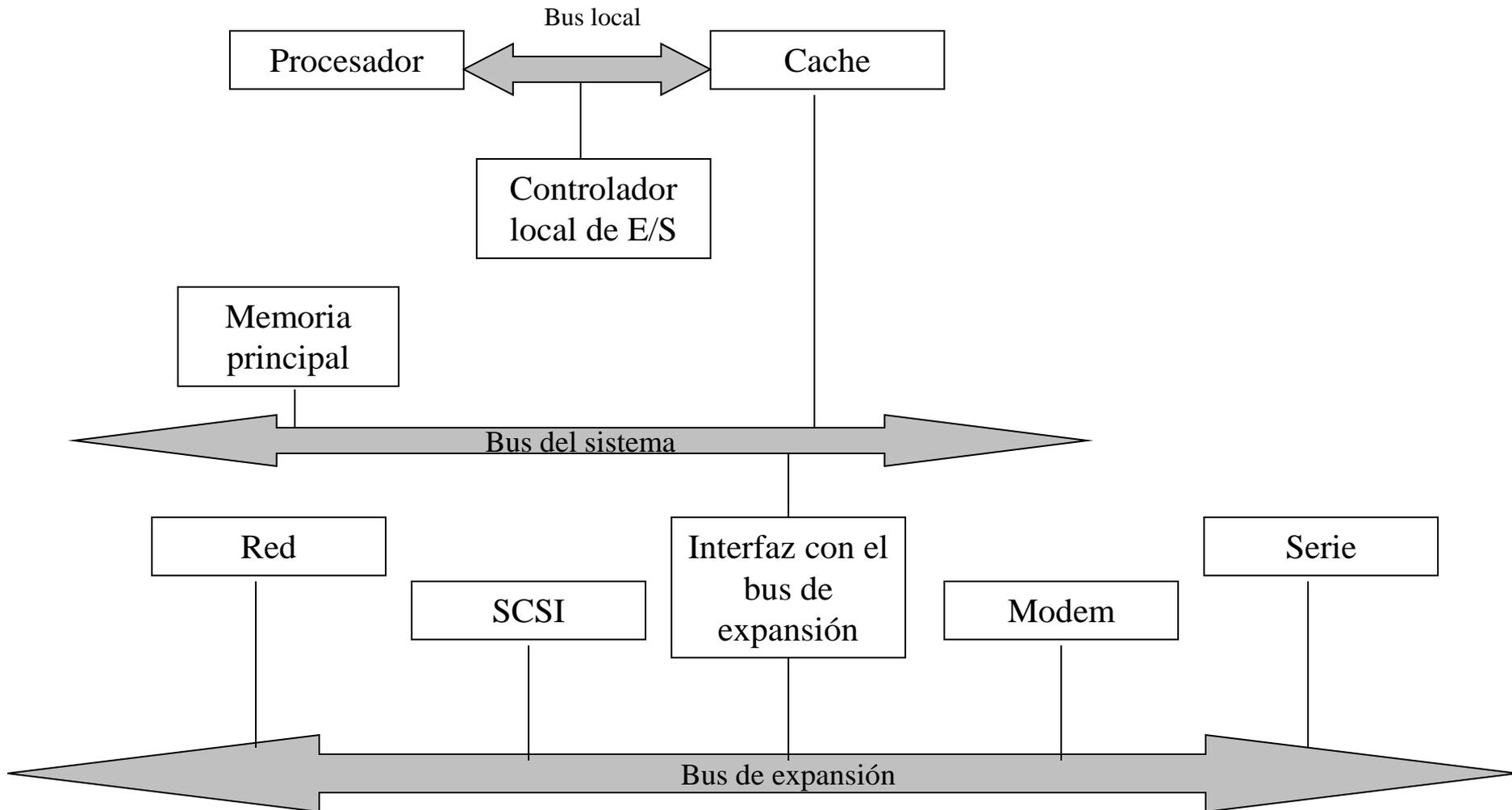
- 2º) Transferir la petición de información a través de las líneas de control y dirección.

- 3º) Esperar que el segundo módulo envíe el dato.

Jerarquías de buses

- ▶ 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



Curiosidades: Familia USB



	Transfer. (por seg.)	Aparición
USB 3.0	600 MB/s	2010
USB 2.0	60 MB/s	2000
USB 1.0	1.5 MB/s y 187 KB/s	1996

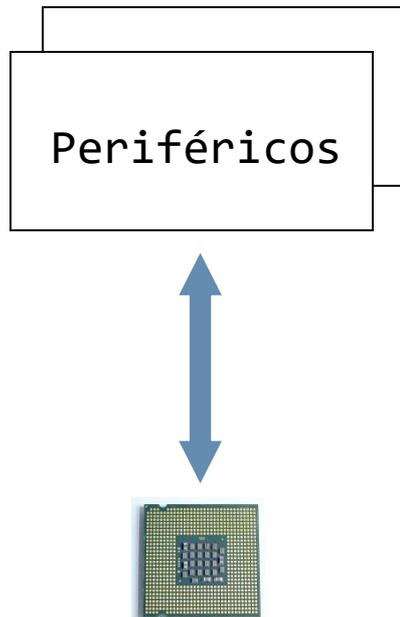
	Song / Pic	256 Flash	USB Flash	SD-Movie	USB Flash	HD-Movie
	4 MB	256 MB	1 GB	6 GB	16 GB	25 GB
USB 1.0	5.3 sec	5.7 min	22 min	2.2 hr	5.9 hr	9.3 hr
USB 2.0	0.1 sec	8.5 sec	33 sec	3.3 min	8.9 min	13.9 min
USB 3.0	0.01 sec	0.8 sec	3.3 sec	20 sec	53.3 sec	70 sec



Contenidos

- ❑ **Introducción**
- ❑ **Buses**
 - ❑ Estructura y funcionamiento
 - ❑ Jerarquía de buses
- ❑ **Periféricos**
 - ❑ **Concepto de periférico**
 - ❑ **Clasificación y tipos de periféricos**
 - ❑ **Estructura general de un periférico**
 - ❑ Caso de estudio: disco duro
- ❑ **Módulos de E/S**
 - ❑ Estructura
 - ❑ Funcionamiento general
 - ❑ Características

Concepto de periférico



▶ Periférico:

- ▶ Todo aquel dispositivo externo que se conecta a una CPU 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, ...)

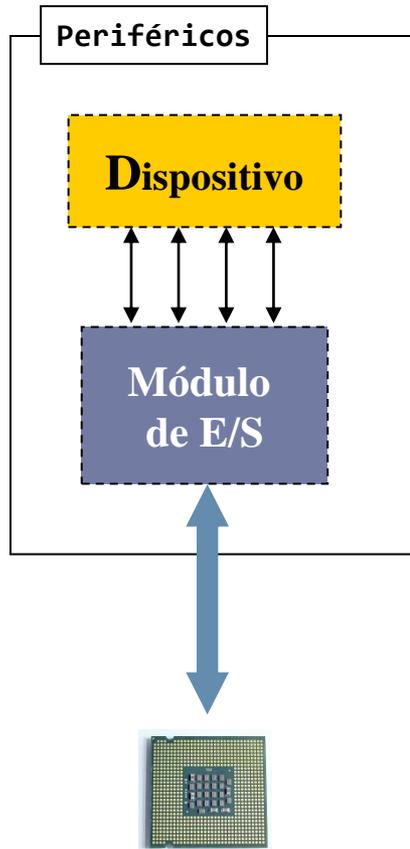
► Medio físico

- (Lectura/accionamiento) x (analógico/digital)

► Almacenamiento:

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

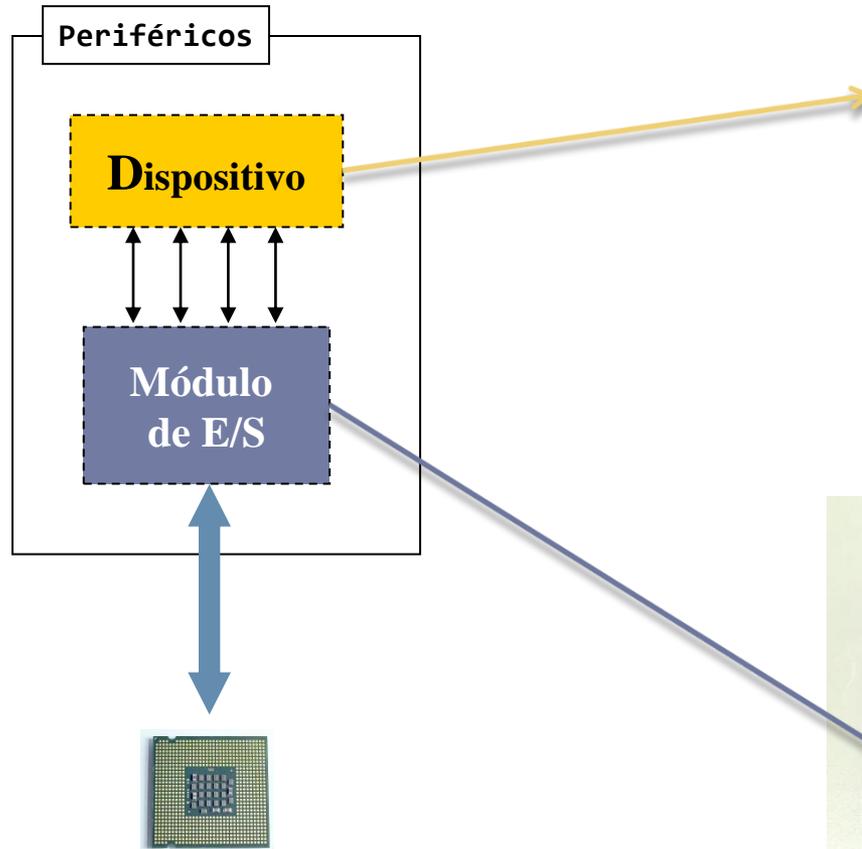
Estructura general de un periférico



- ▶ Compuesto de:
 - ▶ **Dispositivo**
 - ▶ Hardware que interactúa con el entorno
 - ▶ **Módulo de Entrada/Salida**
 - ▶ También denominado **controlador**
 - ▶ Interfaz entre dispositivo y la CPU, que le oculta las particularidades de este

Ejemplo

Disco duro



Contenidos

- ❑ Introducción
- ❑ Buses
 - ❑ Estructura y funcionamiento
 - ❑ Jerarquía de buses
- ❑ **Periféricos**
 - ❑ Concepto de periférico
 - ❑ Clasificación y tipos de periféricos
 - ❑ Estructura general de un periférico
 - ❑ **Caso de estudio: disco duro**
- ❑ Módulos de E/S
 - ❑ Estructura
 - ❑ Funcionamiento general
 - ❑ Características

Un poco de historia...



- ▶ El primer disco duro apareció en 1956
 - ▶ Encargo de las fuerzas Aéreas de EEUU
 - ▶ Se le llamó IBM RAMAC 305
 - ▶ 50 discos de aluminio de 61 cm de diámetro
 - ▶ 5 MB de datos
 - ▶ Giraba a 3.600 revoluciones por minuto
 - ▶ Contaba con una velocidad de transferencia de 8,8 Kbps
 - ▶ Pesaba cerca de una tonelada

Un poco de historia...



1956



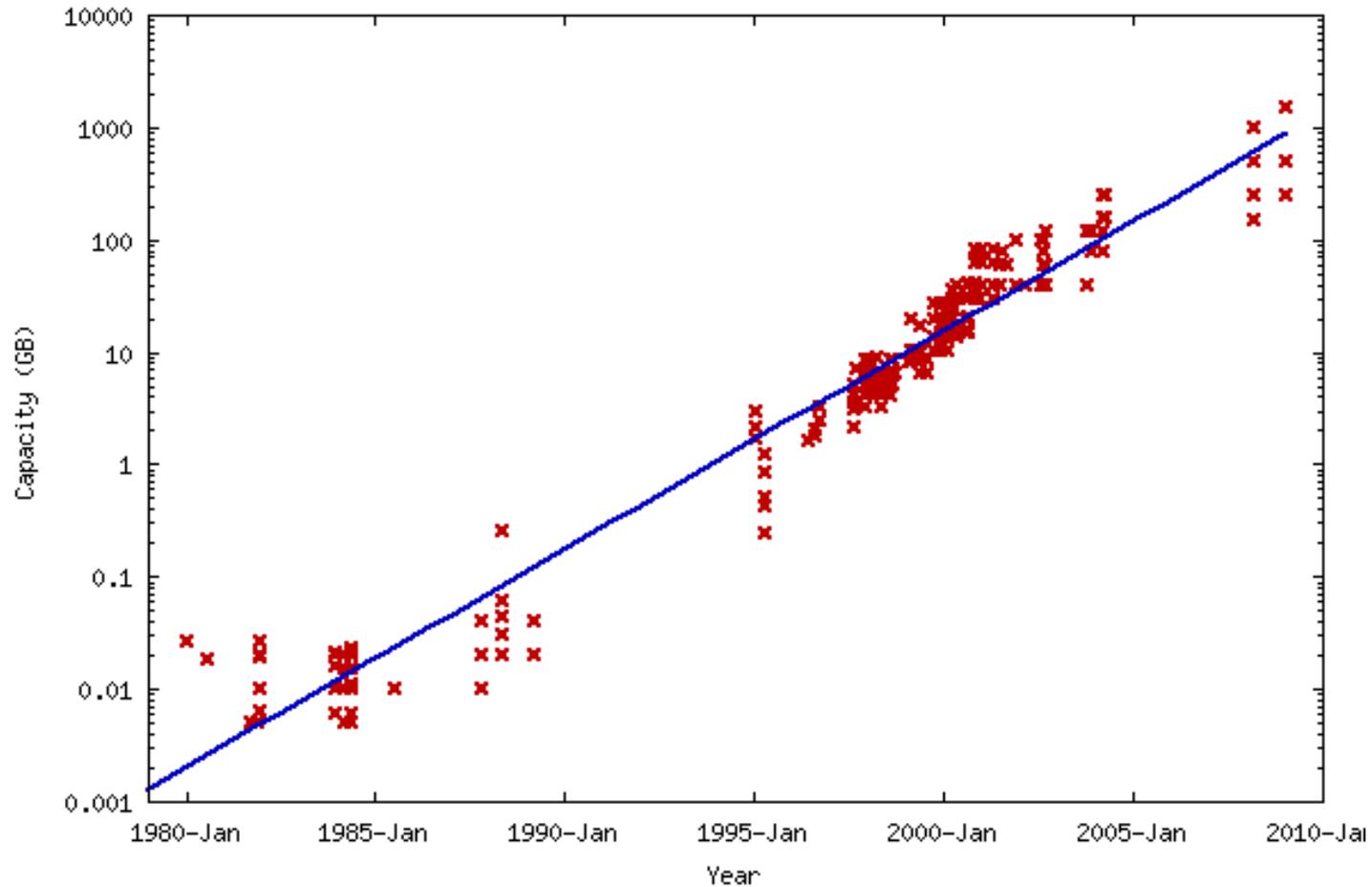
80's



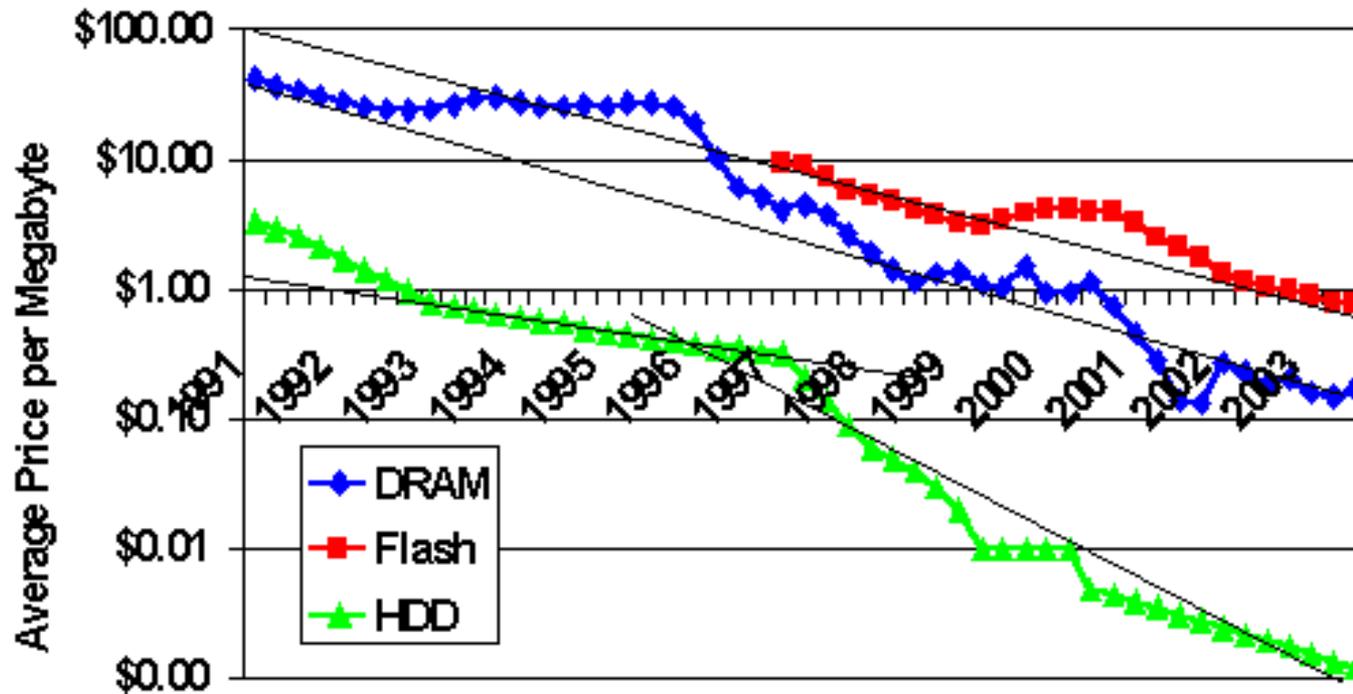
2005



Un poco de historia...



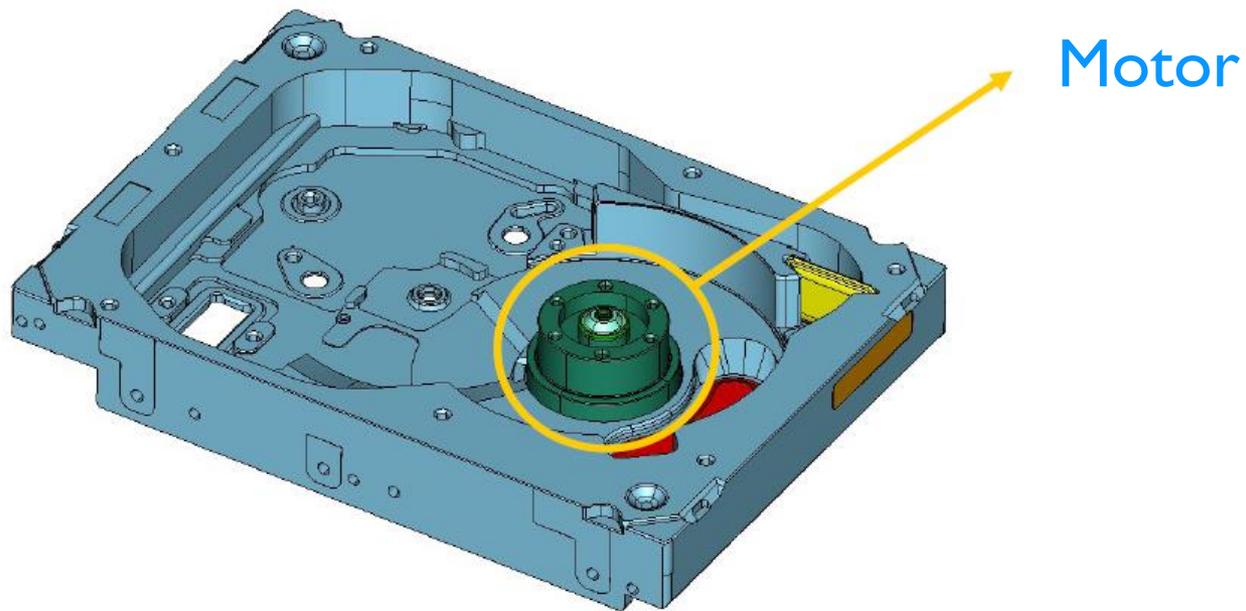
Un poco de historia...



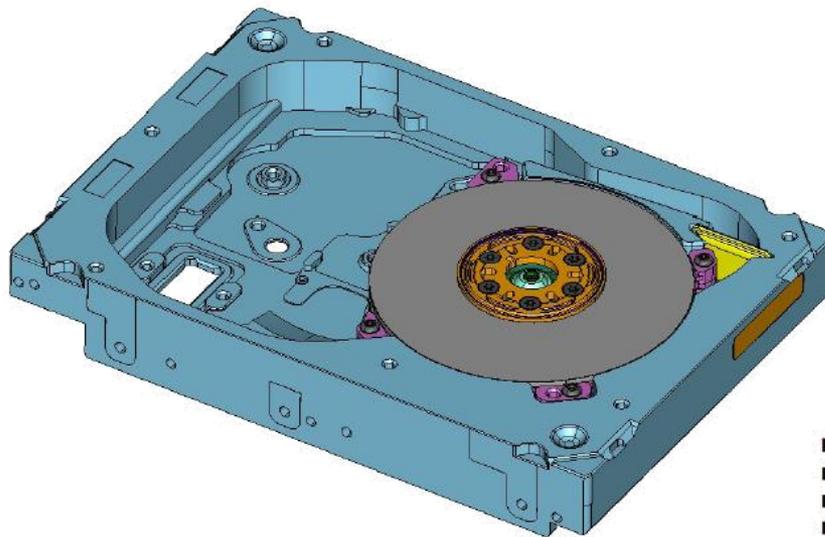
Un poco de historia...

	Tasa de crecimiento por año
Capacidad	1.93/año
Coste	0.6/año
Prestaciones	0.05/año

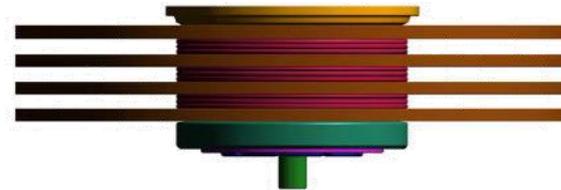
Anatomía de un disco duro...



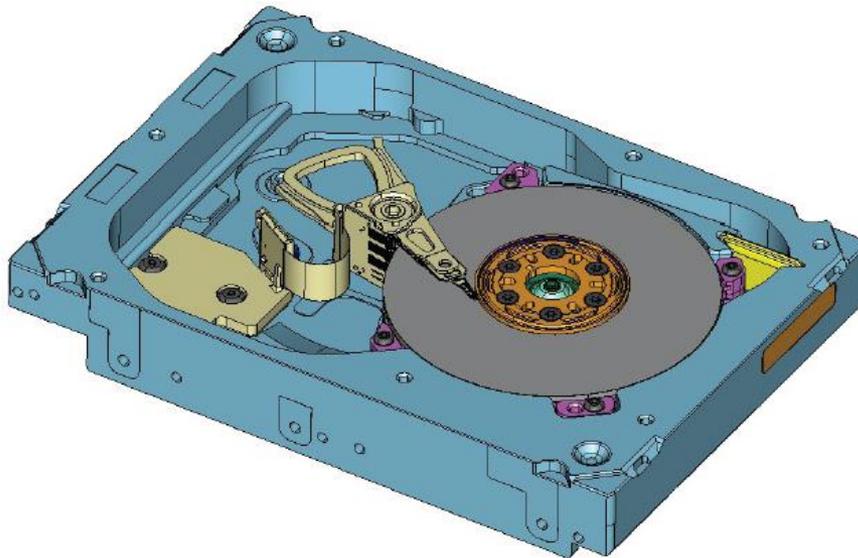
Anatomía de un disco duro...



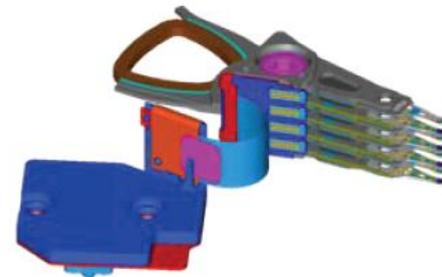
Discos



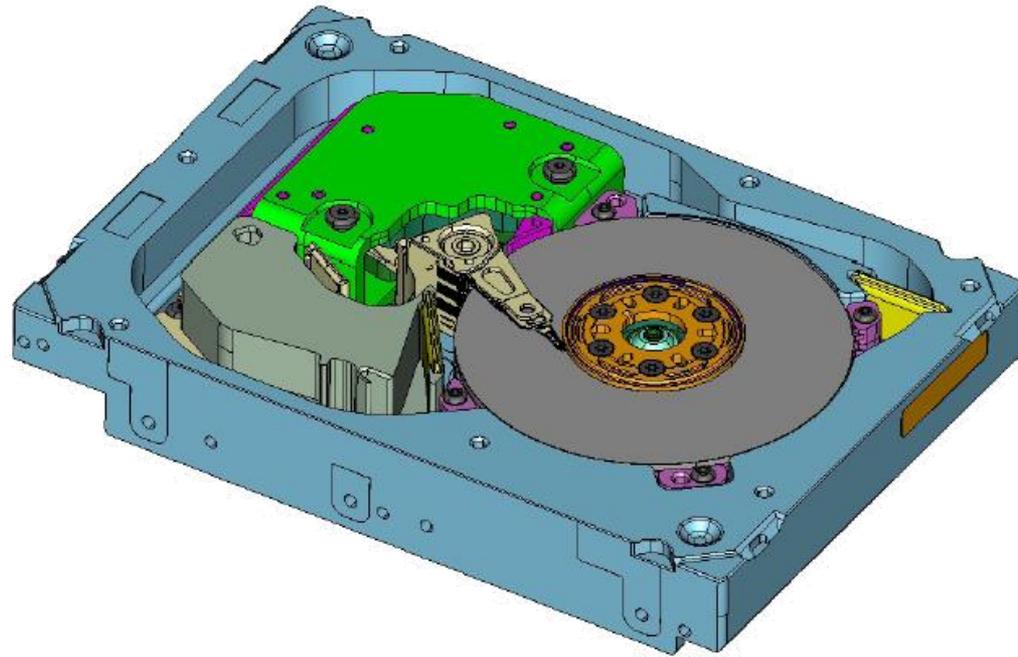
Anatomía de un disco duro...



Cabezas
lectoras/escriptoras



Anatomía de un disco duro...



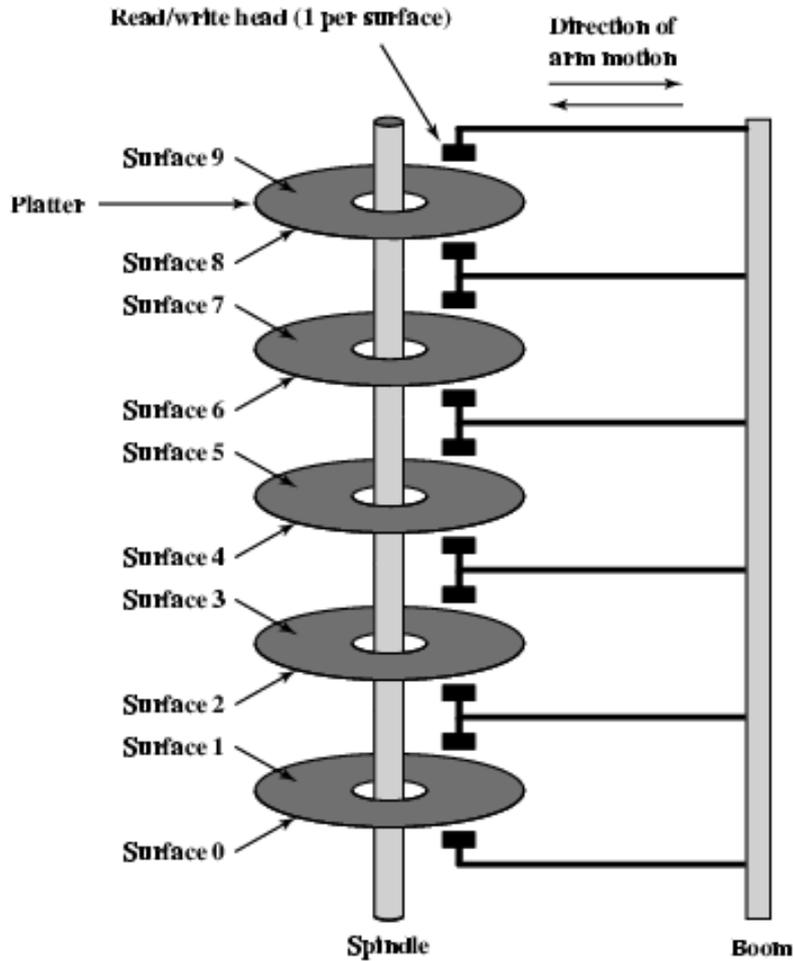
Módulo de control y
mecánica

Anatomía de un disco duro...

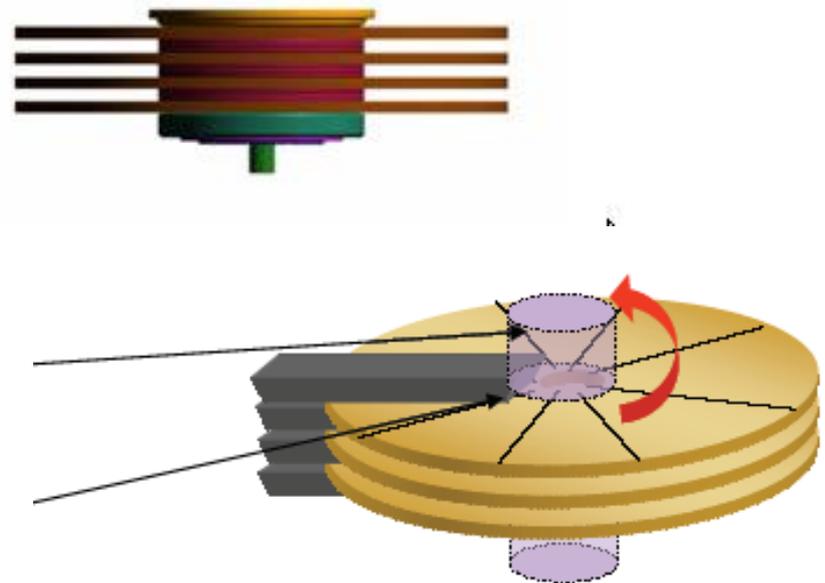


- ▶ **Electrónica**
 - ▶ Planificación de comandos
 - ▶ Corrección de errores
 - ▶ Optimización
 - ▶ Comprobación de integridad
 - ▶ Control de las revoluciones por minuto (RPM)

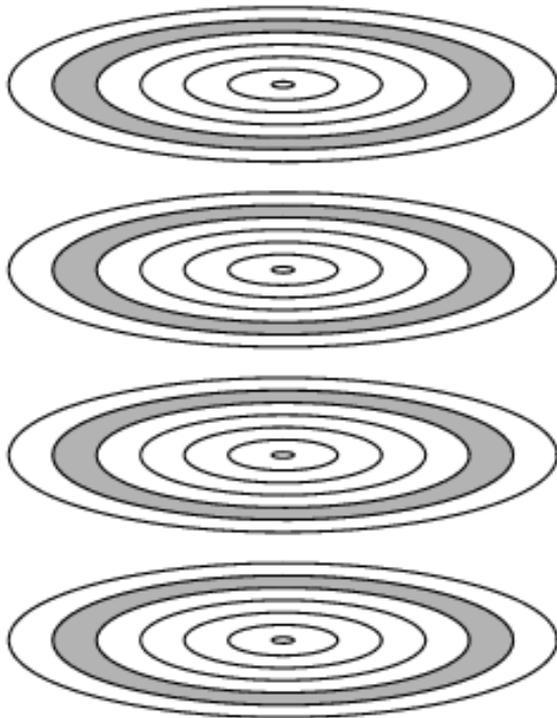
Almacenamiento



Rotación

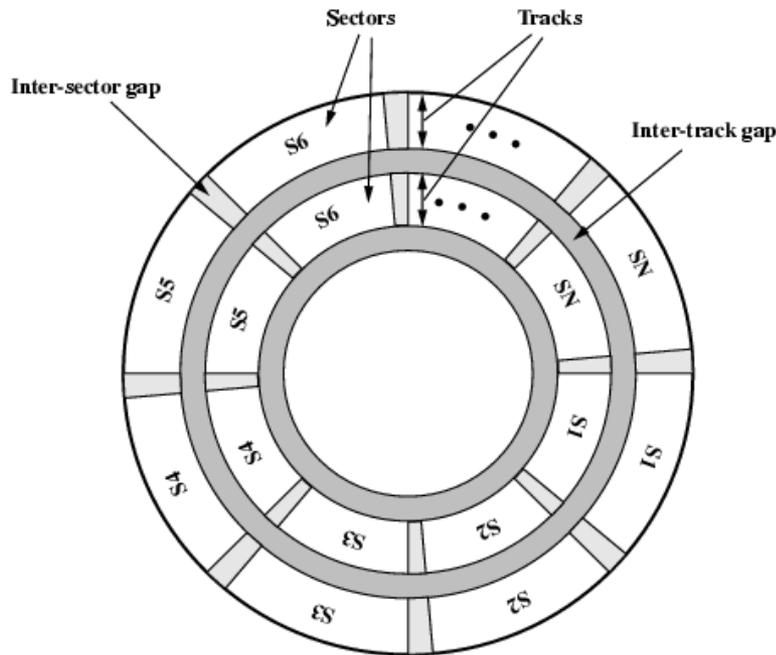


Almacenamiento: cilindro



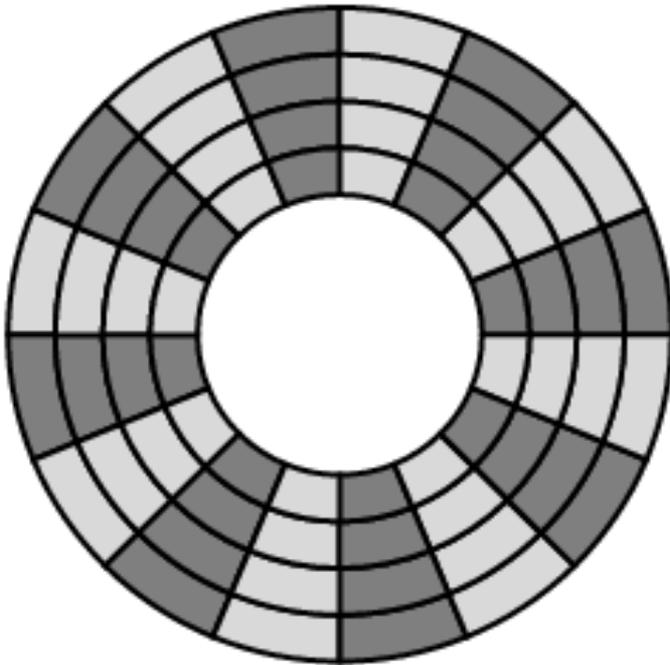
- ▶ **Cilindro:**
información accedida por todas las cabezas en una rotación

Almacenamiento: pistas y sectores

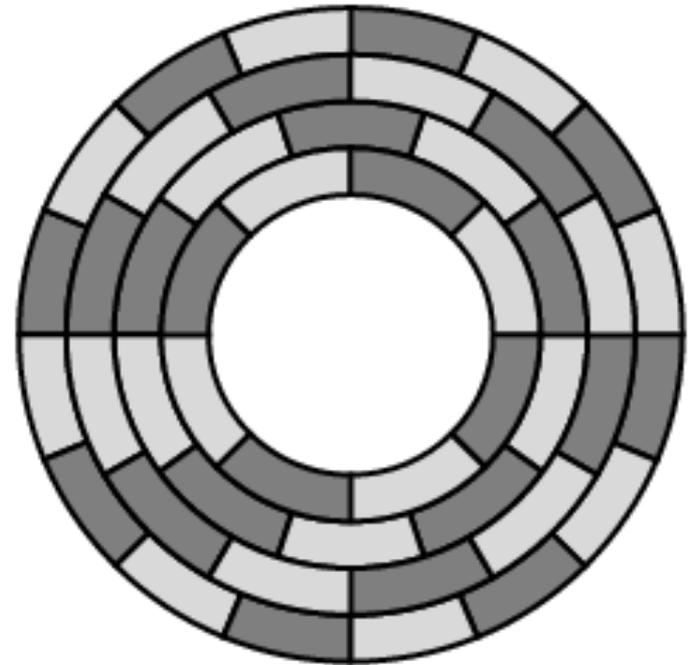


- ▶ **Pista:**
 - ▶ Un anillo del plato
- ▶ **Sector:**
 - ▶ División de la superficie del disco realizada en el formateo (típicamente 512 bytes)
- ▶ **Bloques:**
 - ▶ El sistema de ficheros escribe en bloques
 - ▶ Grupo de sectores

Distribución de sectores



(a) Constant angular velocity



(b) Multiple zoned recording

Capacidad de almacenamiento

- ▶ Para discos con velocidad angular constante

- ▶ n_s : número de superficies
- ▶ p : número de pistas por superficie
- ▶ s : número de sectores por pista
- ▶ t_s : bytes por sector

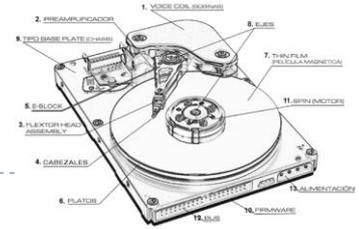
$$Capacidad = n_s \times p \times s \times t_s$$

- ▶ Para discos con múltiples zonas

- ▶ z : número de zonas
- ▶ p_i : pistas de la zona i
- ▶ s_i : sectores por pista de la zona i

$$Capacidad = n_s \times t_s \times \sum_{i=1}^z (p_i \times s_i)$$

Tiempo de acceso



- ▶ $T_{\text{acceso}} = n * T_{\text{búsqueda}} + T_{\text{latencia}} + T_{\text{transferencia}}$
- ▶ **n**: número de cilindros a desplazar desde el cilindro actual al cilindro sobre el que se quiere operar
- ▶ Tiempo de búsqueda ($T_{\text{búsqueda}}$): tiempo necesario para mover la cabeza un cilindro de distancia
- ▶ Latencia de rotación (T_{latencia}): tiempo que pasa hasta que el sector deseado pasa por debajo de la cabeza de lectura/escritura
 - ▶ $T_{\text{latencia}} =$ Tiempo medio para recorrer media pista
- ▶ Tiempo de transferencia (T_t): tiempo necesario para recorrer un sector y transferir los datos de él

Ejercicio



- ▶ Sea un disco con un solo plato con las siguientes características:
 - ▶ Velocidad de rotación: 7200 rpm
 - ▶ Platos: 5, con 2 superficies por plato
 - ▶ Número de pistas de una cara del plato: 30000
 - ▶ Sectores por pista: 600 (de 512 bytes)
 - ▶ Tiempo de búsqueda: 1 ms por cada 100 pistas atravesadas
- ▶ Suponiendo que la cabeza está en la pista 0 y se solicita un sector de la pista 600, calcular:
 - ▶ Capacidad del disco duro
 - ▶ La latencia de rotación
 - ▶ Tiempo de transferencia de un sector
 - ▶ Tiempo de búsqueda del sector pedido

Ejercicio (sol.)



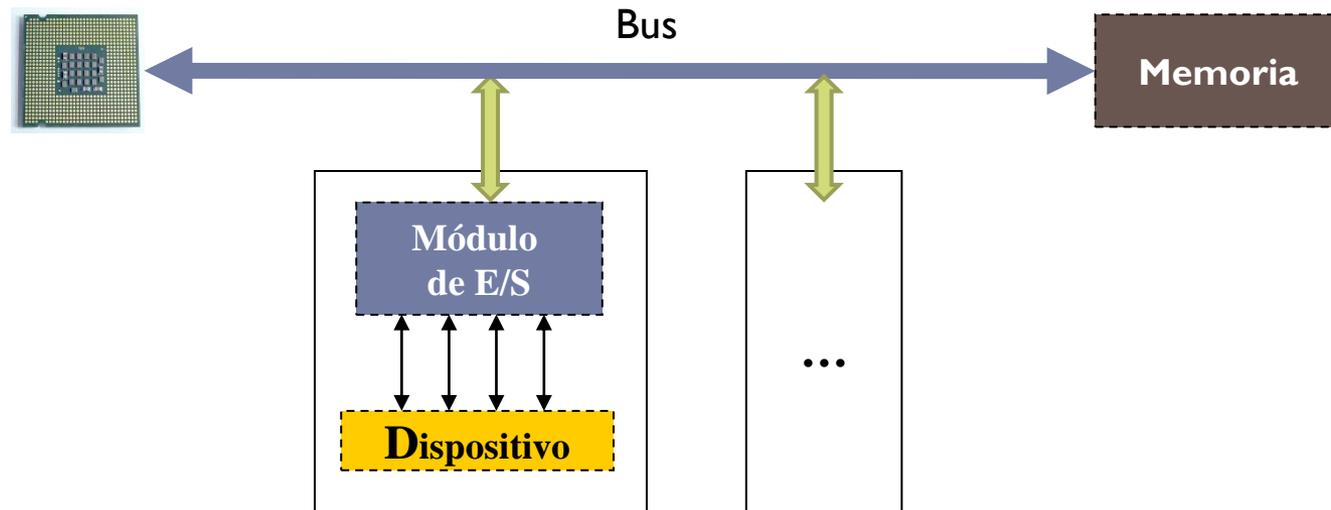
- ▶ **Capacidad:**
 - ▶ $5 \text{ platos} * 2 \text{ caras/plato} * 30.000 \text{ pistas/cara} * 600 \text{ sectores/pista} * 512 \text{ bytes/sector} = 85,8 \text{ GB}$
- ▶ **Latencia de rotación:**
 - ▶ $L_r = \text{Tiempo de media vuelta a una pista}$
 - ▶ $7.200 \text{ vueltas/minuto} \rightarrow 120 \text{ vuelta/segundo}$
 $\rightarrow 0,0083 \text{ segundos/vuelta} \rightarrow 4,2 \text{ milisegundos (media vuelta)}$
- ▶ **Tiempo de transferencia de un sector:**
 - ▶ Hay 600 sectores por pista y la pista se lee en 8,3 milisegundos
 - ▶ $8,3 / 600 \rightarrow 0,014 \text{ milisegundos}$
- ▶ **Tiempo de búsqueda:**
 - ▶ Cada 100 pistas 1 ms, y hay que ir a la pista 600
 - ▶ $600 / 100 = 6 \text{ milisegundos}$

Contenidos

- ❑ **Introducción**
- ❑ **Buses**
 - ❑ Estructura y funcionamiento
 - ❑ Jerarquía de buses
- ❑ **Periféricos**
 - ❑ Concepto de periférico
 - ❑ Clasificación y tipos de periféricos
 - ❑ Estructura general de un periférico
 - ❑ Caso de estudio: disco duro
- ❑ **Módulos de E/S**
 - ❑ **Estructura**
 - ❑ **Funcionamiento general**
 - ❑ **Características**

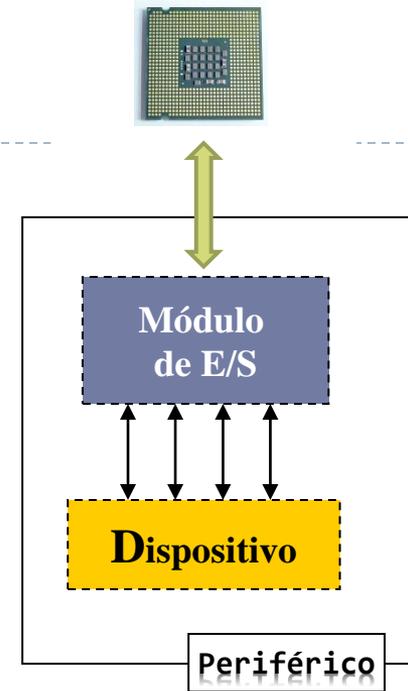
Módulo de E/S: qué son

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



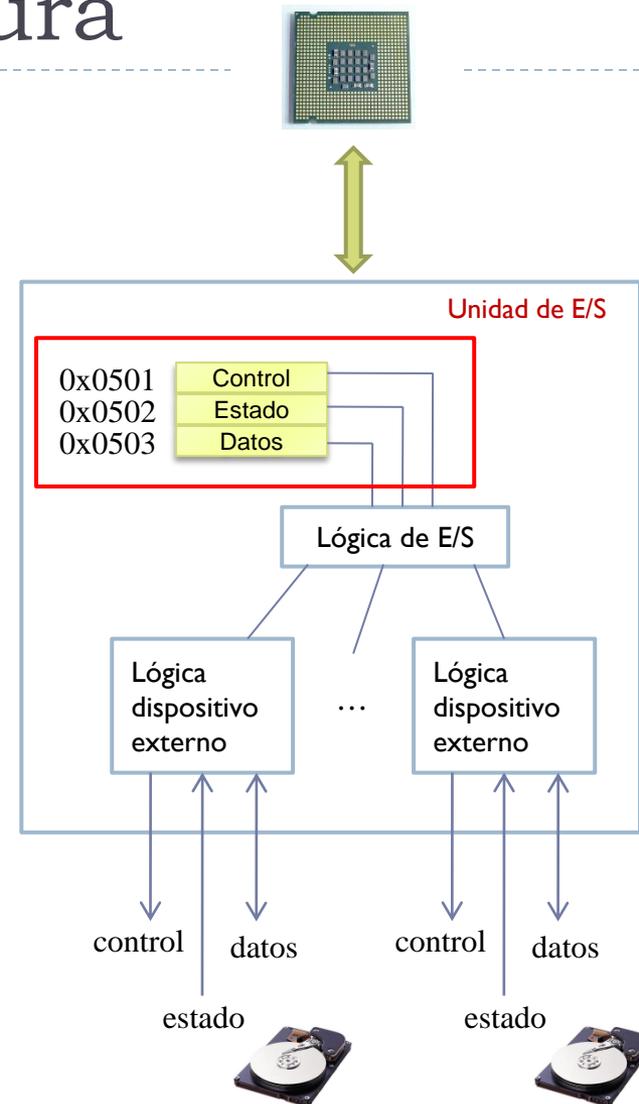
Módulo de E/S: necesidad

- ▶ Son necesarios debido a:
 - ▶ 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.



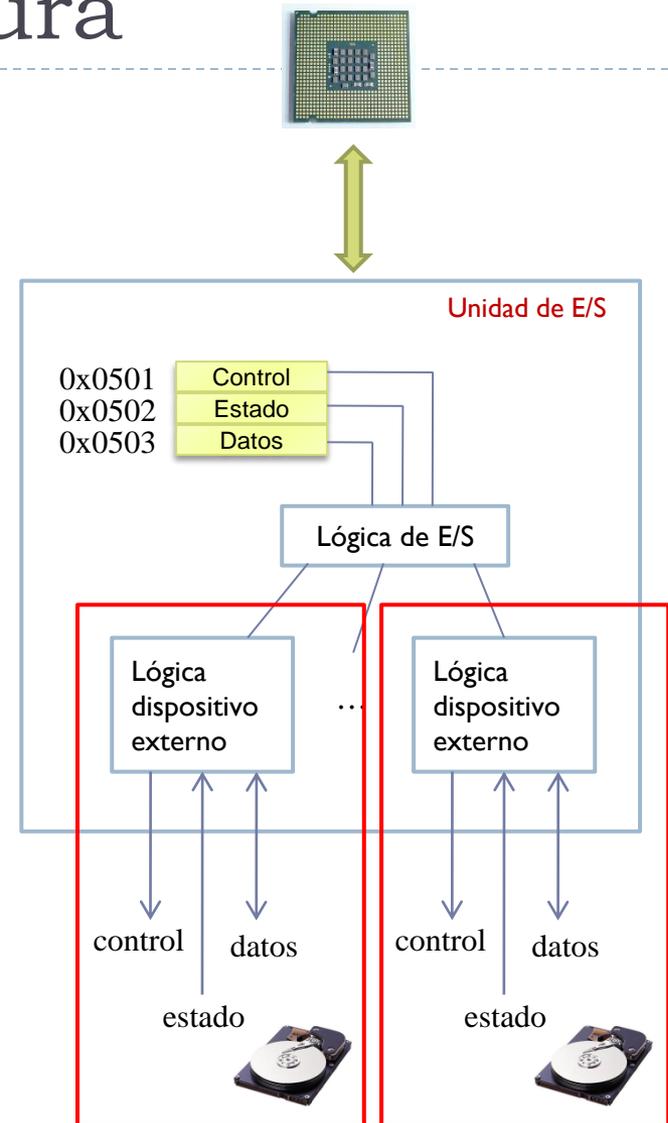
Módulo de E/S: estructura

- ▶ Interacción entre CPU 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 CPU/Perif.



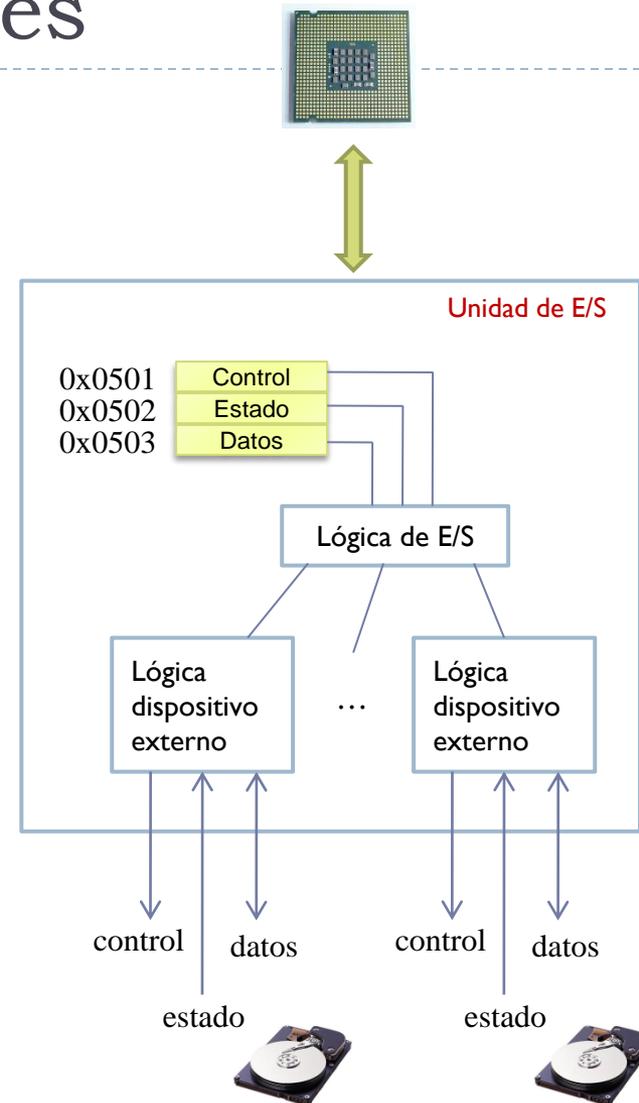
Módulo de E/S: estructura

- ▶ 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: funciones

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



Módulo de E/S: tipos

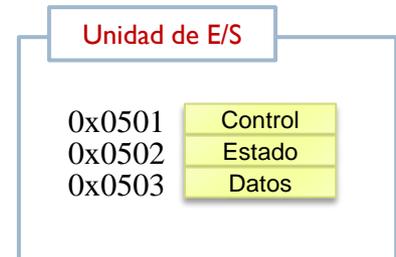
- ▶ Tipos de módulos por complejidad:

- ▶ **Canal de E/S o procesador de E/S:**

- se encarga de la mayoría de los detalles del procesamiento.

- ▶ **Controlador de E/S o controlador de dispositivo:**

- módulo más simple, que requiere que el procesador tenga un control detallado del dispositivo.



Contenidos

- ❑ **Introducción**
- ❑ **Buses**
 - ❑ Estructura y funcionamiento
 - ❑ Jerarquía de buses
- ❑ **Periféricos**
 - ❑ Concepto de periférico
 - ❑ Clasificación y tipos de periféricos
 - ❑ Estructura general de un periférico
 - ❑ Caso de estudio: disco duro
- ❑ **Módulos de E/S**
 - ❑ Estructura
 - ❑ Funcionamiento general
 - ❑ **Características**

Módulo de E/S: características

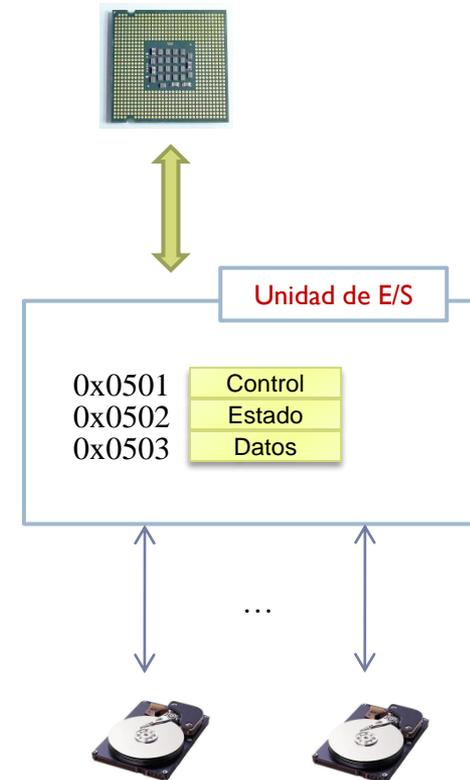
- ▶ Características fundamentales:

- ▶ **Unidad de transferencia**

- ▶ **Direccionamiento (y ancho de palabra)**

- ▶ **Interacción computador-controlador**

¡ Importante !



Características (1 / 3)

- ▶ Unidad de transferencia:
 - ▶ **Dispositivos de bloque:**
 - ▶ Acceso secuencial o directo a bloques
 - ▶ Operaciones: leer, escribir, situarse, ...
 - ▶ Acceso posible a través de ficheros proyectados en memoria
 - ▶ Ejemplos: discos y “cintas”
 - ▶ **Dispositivos de carácter:**
 - ▶ Acceso secuencial a caracteres
 - ▶ Operaciones: `get`, `put`,
 - ▶ Ejemplo: terminales, impresoras, tarjetas de red

Características (2/3)

▶ Direccionamiento de E/S:

▶ **Proyectados en memoria**

- ▶ 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`



▶ **Mediante puertos:**

- ▶ Con instrucciones ensamblador especiales (In/Out) se acceden a unos puertos que representan los registros del ‘controlador’.

- ▶ Ej: `out $a0 0x105A`

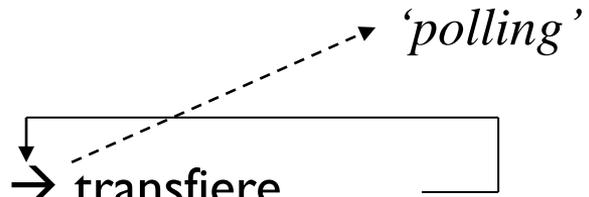


Características (3/3)

▶ Interacción Computador-Controlador:

▶ **E/S programada o directa**

- ▶ CPU no hace otra cosa que E/S: espera → transfiere



▶ **E/S por interrupciones**

- ▶ CPU: sólo transfiere

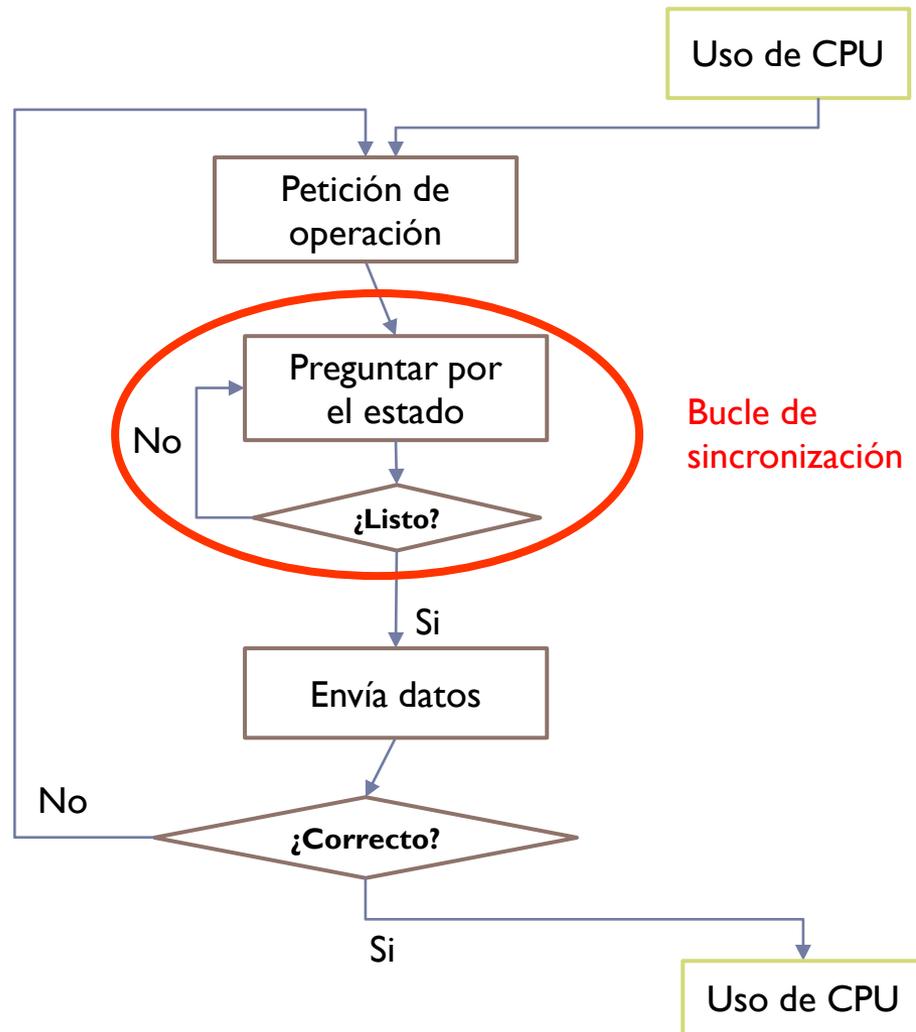
▶ **E/S por DMA (acceso directo a memoria)**

- ▶ CPU no transfiere solo se le avisa del fin del bloque transferido
 - controlador de periférico más sofisticado
 - lógica para DMA: contadores, señales de control, etc.

Interacción mediante E/S programada



Interacción mediante E/S programada



Ejemplo



- ▶ 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

Ejemplo (lectura de dato)



- ▶ 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

1. **Enviar la orden:**
 - `li $t0 0`
 - `sw $t0 0x500`
2. **Leer estado hasta que esté listo:**
 - bucle: `lw $t1 0x504`
 - `beqz $t1 bucle`
3. **Leer el dato:**
 - `lw $t2 0x508`

Ejemplo (escritura de dato)



- ▶ 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

1. Enviar el dato:
 - `li $t0 123`
 - `sw $t0 0x508`
2. Enviar la orden:
 - `li $t0 1`
 - `sw $t0 0x500`
3. Leer estado hasta que esté listo:
 - bucle: `lw $t1 0x504`
 - `beqz $t1 bucle`

Ejercicio



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

- ▶ 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

Ejercicio (sol.)



- ▶ 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: .space 400

.text

.globl main

main:    li $t3 0
        bucle1: li $t0 0
            sw $t0 0x500

        bucle2: lw $t1 0x504
            beqz $t1 bucle2
            lw $t2 0x508
            sw $t2 datos($t3)
            add $t3 $t3 4
            bne $t3 400 bucle1
```

Ejercicio (sol.)



- ▶ 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: .space 400

.text
.globl main
main:    li $t3 0
        bucle1: li $t0 0
            sw $t0 0x500
        bucle2: lw $t1 0x504
            beqz $t1 bucle2
            lw $t2 0x508
            sw $t2 datos($t3)
            add $t3 $t3 4
            bne $t3 400 bucle1
```

} Bucle de sincronización

Ejercicio (sol.)



- ▶ 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: .space 400

.text

.globl main

main:    li $t3 0
         bucle1: li $t0 0
                    sw $t0 0x500

         bucle2: lw $t1 0x504
                    beqz $t1 bucle2
                    lw $t2 0x508
                    sw $t2 datos($t3)
                    add $t3 $t3 4
                    bne $t3 400 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: .space 400

.text
.globl main
main:    li $t3 0
        bucle1: li $t0 0
            sw $t0 0x500
        bucle2: lw $t1 0x504
            beqz $t1 bucle2
            lw $t2 0x508
            sw $t2 datos($t3)
            add $t3 $t3 4
            bne $t3 400 bucle1
```



Ejercicio (sol.)



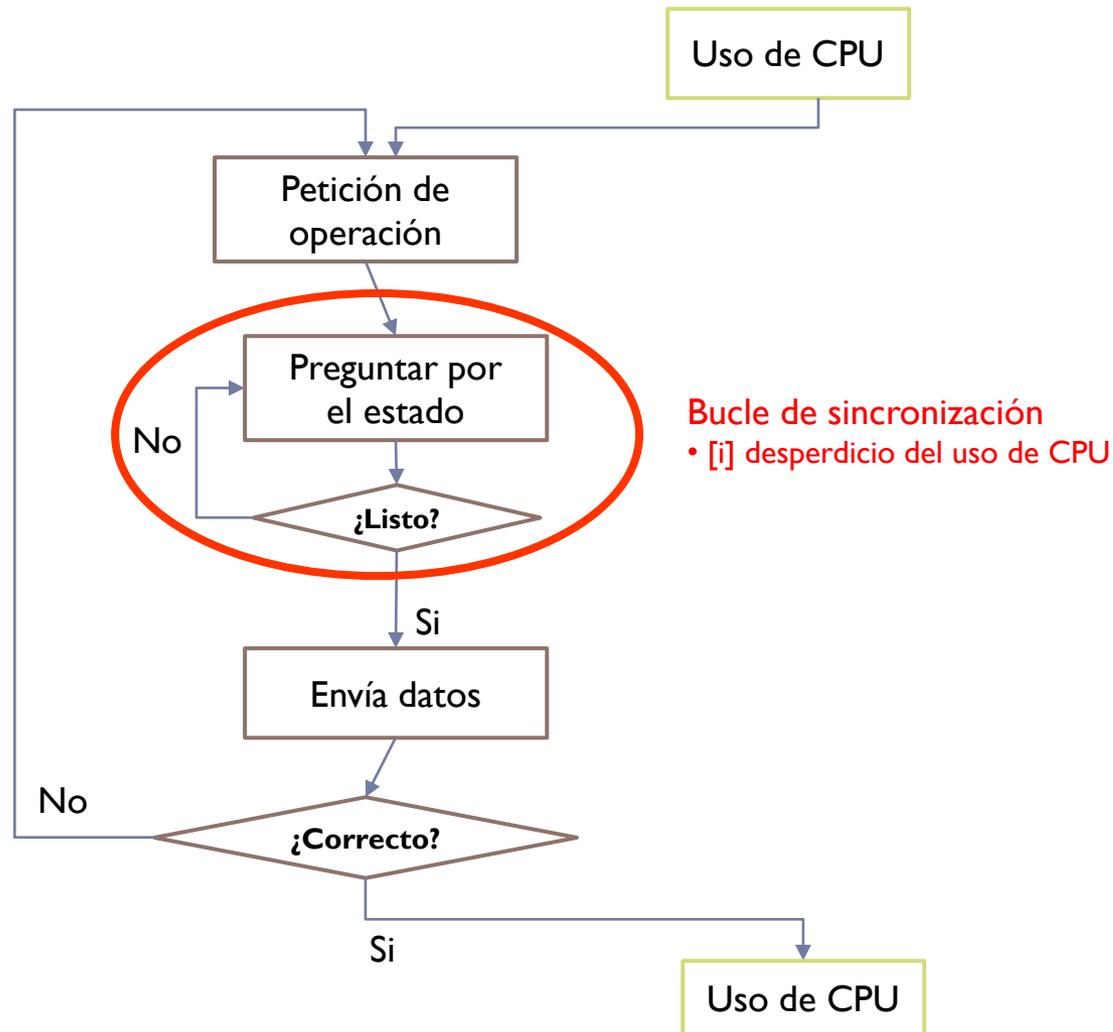
- ▶ Bucle de sincronización:
 - ▶ En media dura 5 ms
 - ▶ Se ejecuta 200 MIPS en media
 - ▶ $I_{bs} = 200 * 10^6 * 5 * 10^{-3} = 10^6$
- ▶ Bucle de transferencia:
 - ▶ $I (li \$t3 0) + 6 * 100 + 10^6 (I_{bs})$
- ▶ Como puede comprobarse, en el bucle se ejecuta 1.000.601 instrucciones, de las cuales 1.000.000 corresponden al bucle de espera (el 99,9%)
 - ▶ Es un desperdicio de ciclos de la CPU

```
.data
    datos: .space 400

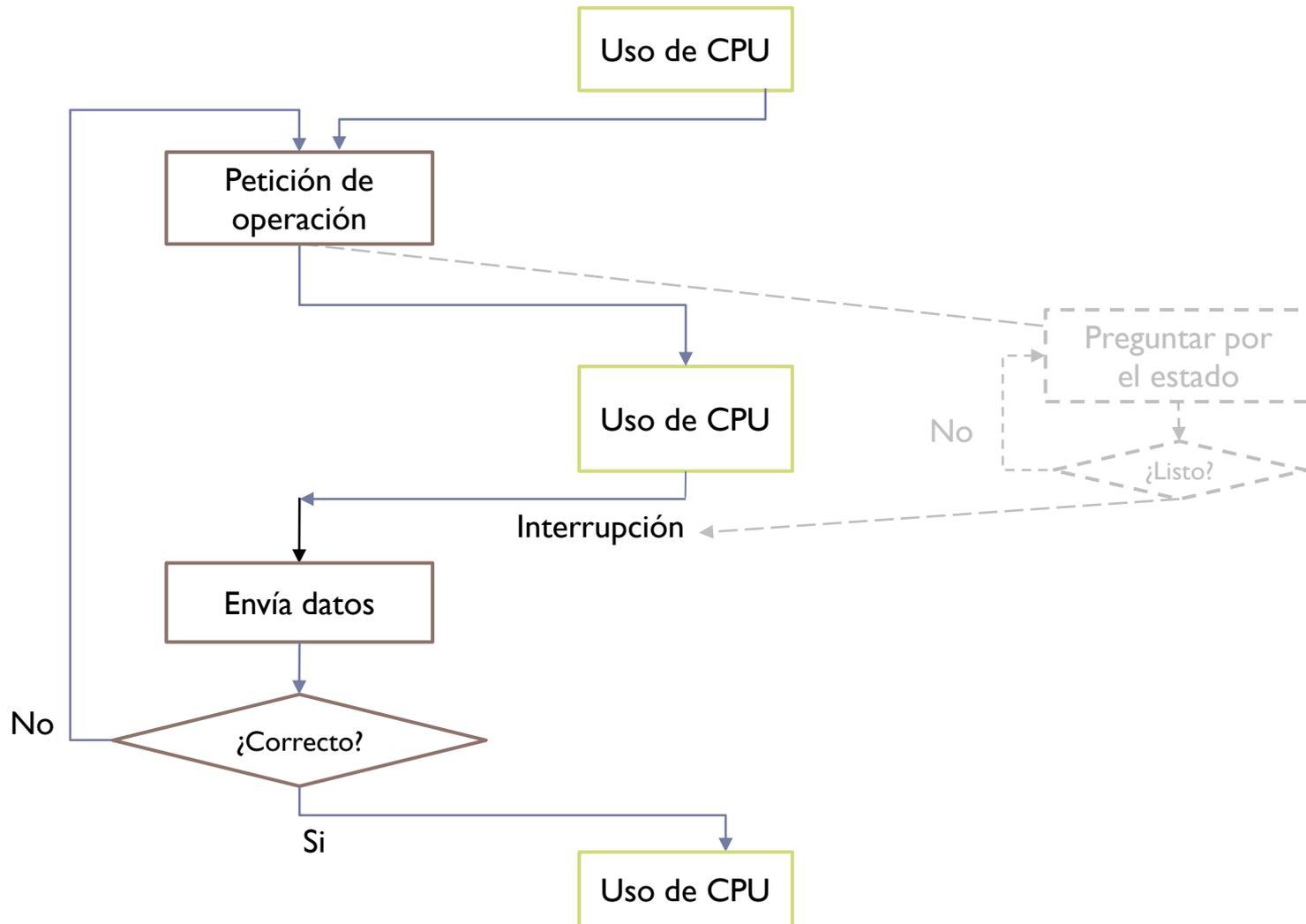
.text
.globl main
main:    li $t3 0
        bucle1: li $t0 0
                sw $t0 0x500
        bucle2: lw $t1 0x504
                beqz $t1 bucle2
                lw $t2 0x508
                sw $t2 datos($t3)
                add $t3 $t3 4
                bne $t3 400 bucle1
```



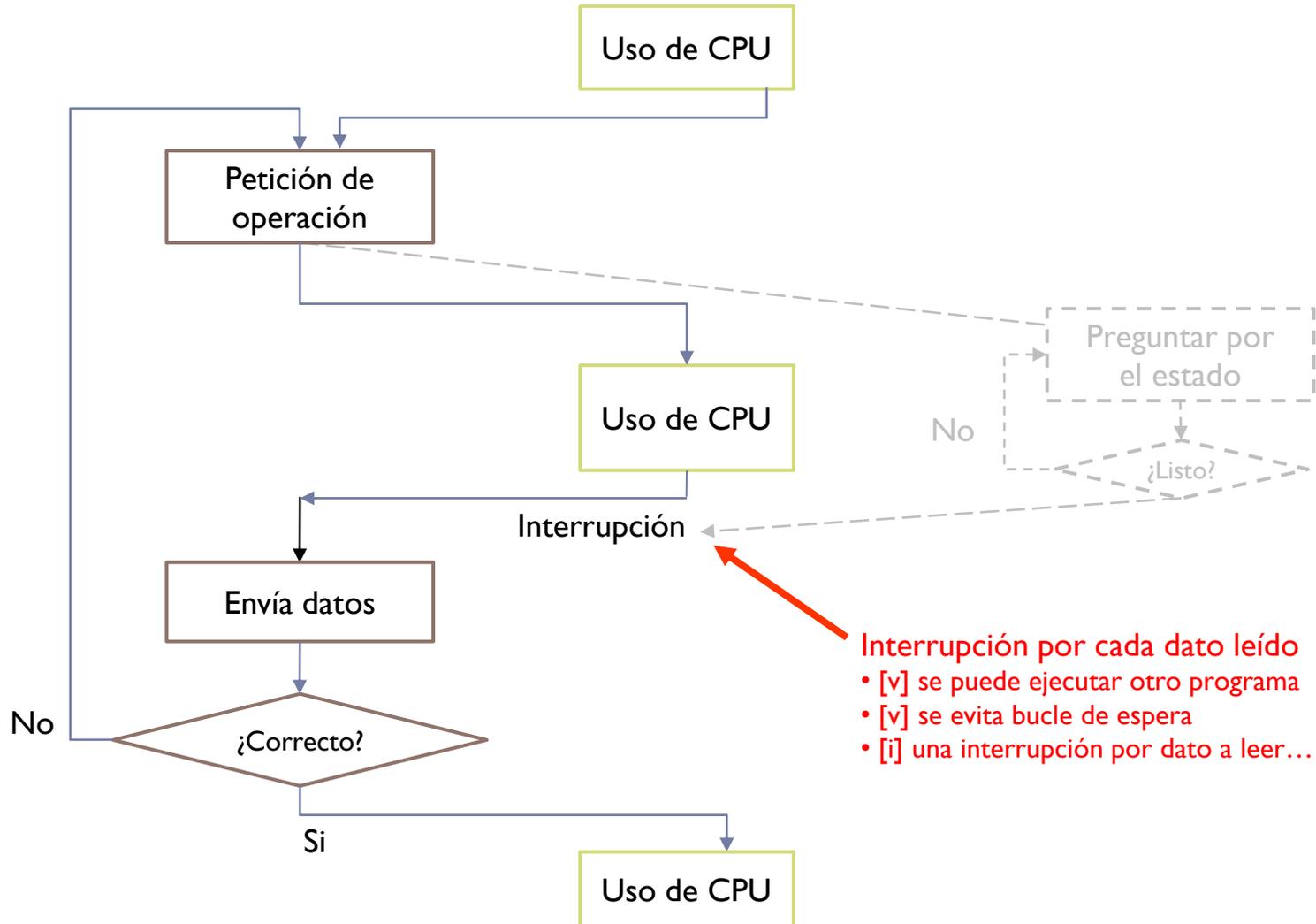
Interacción mediante E/S programada



Interacción mediante interrupciones



Interacción mediante interrupciones



Ejemplo (lectura de 100 datos)

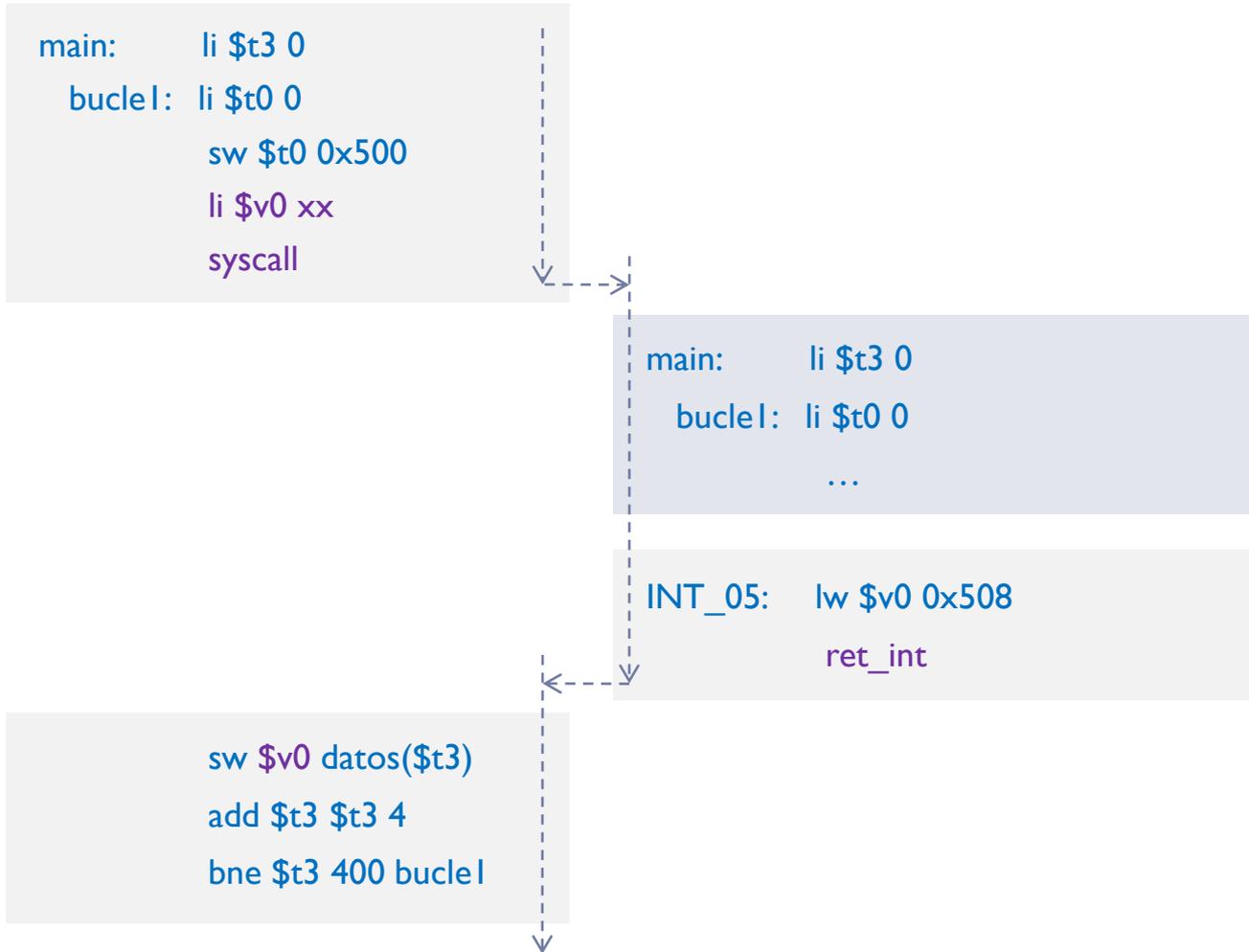


- ▶ Información de control
 - ▶ 0: leer
 - ▶ 1: escribir
- ▶ Información de estado
 - ▶ 0: dispositivo ocupado
 - ▶ 1: dispositivo (dato) listo
- ▶ Mapa de E/S común
 - ▶ Instrucciones lw y sw

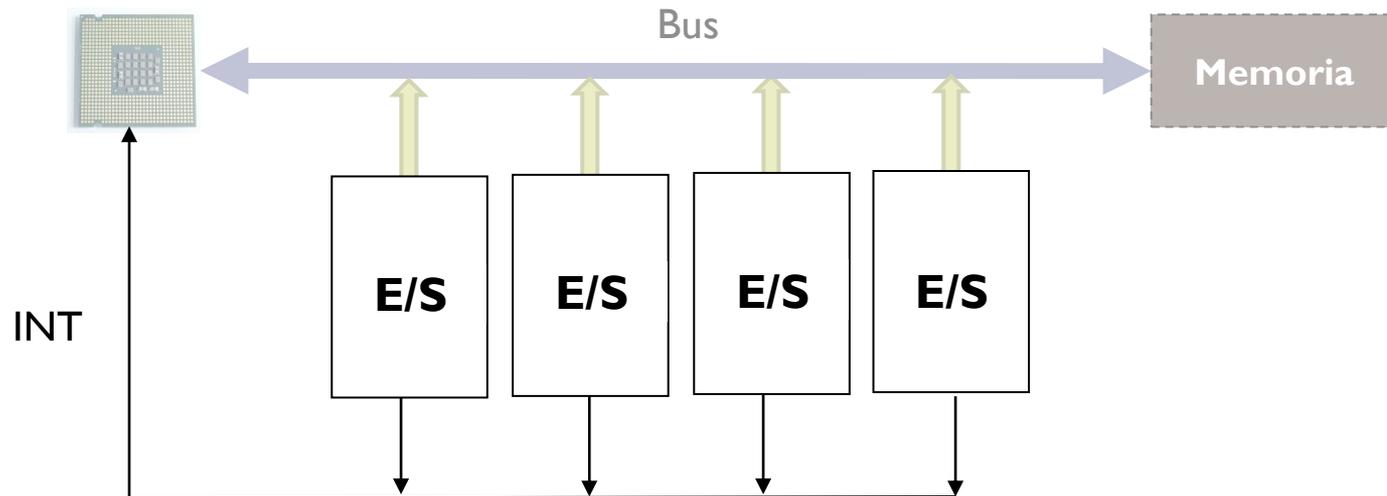
```
main:    li $t3 0
        bucleI: li $t0 0
           sw $t0 0x500
           li $v0 xx
           syscall # sleep till interrupt ☺
           sw $v0 datos($t3)
           add $t3 $t3 4
           bne $t3 400 bucleI
```

```
INT_05:  lw $v0 0x508
           ret_int # restore registers & return
```

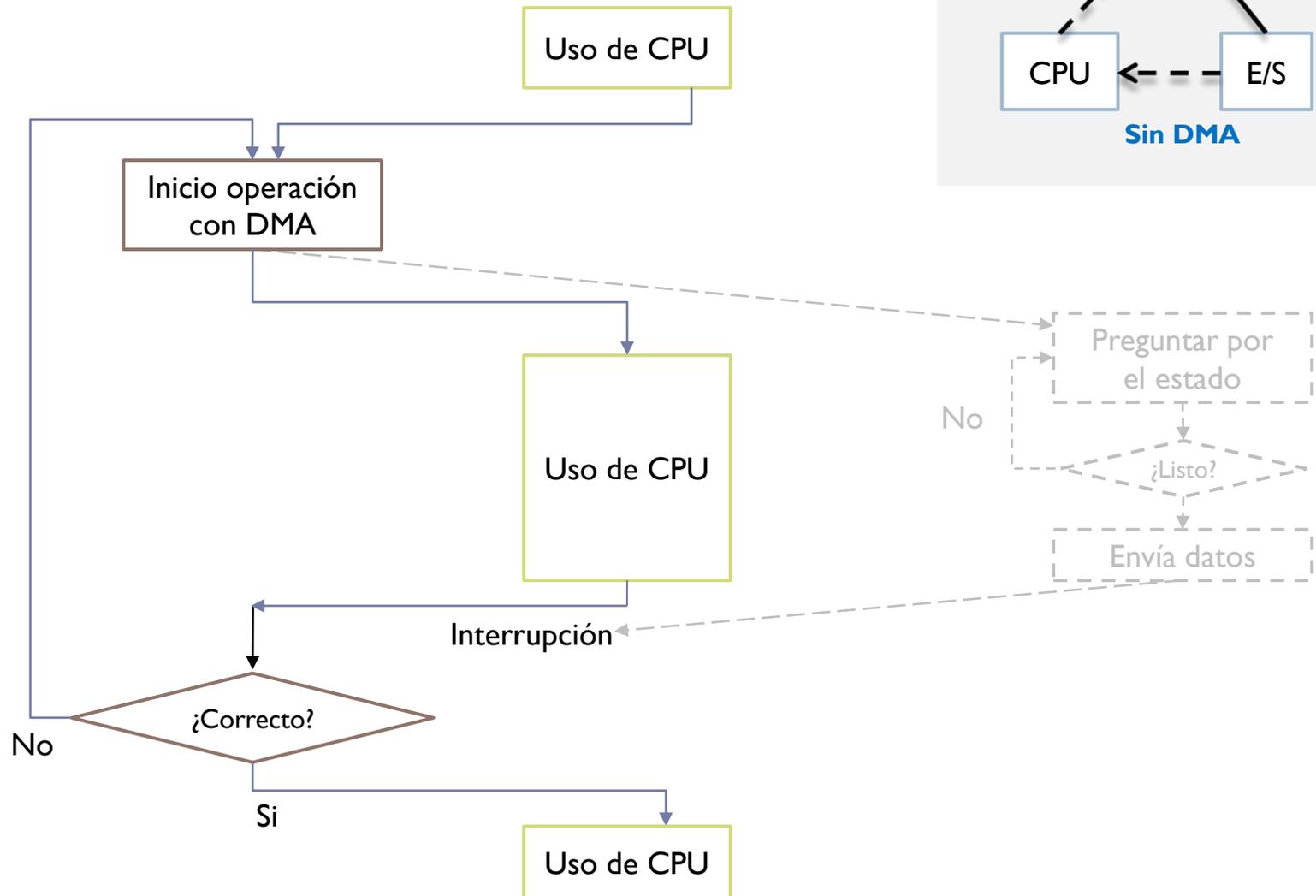
Ejemplo (lectura de 100 datos)



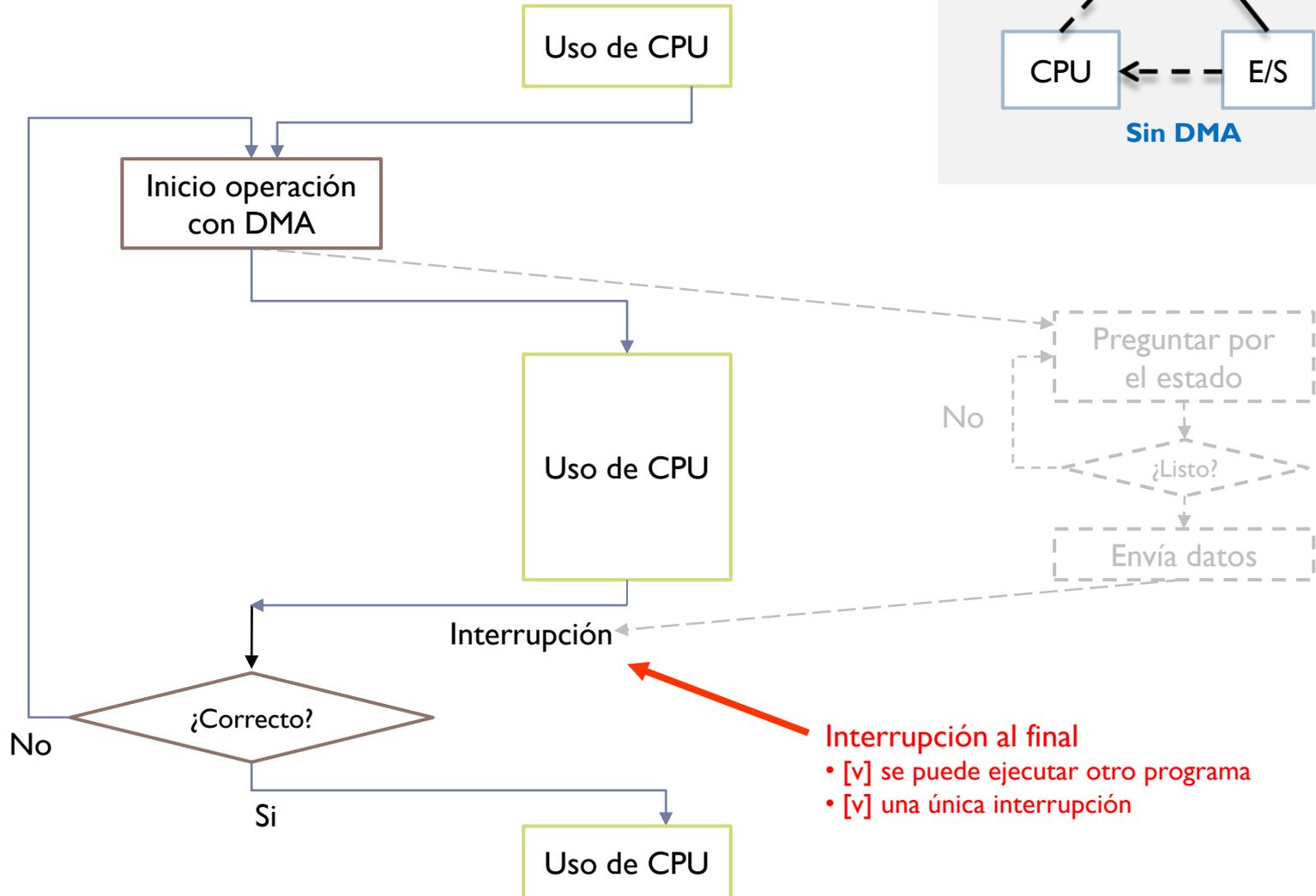
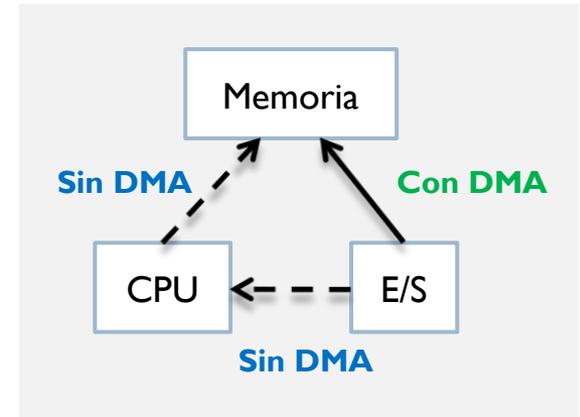
Interacción mediante interrupciones



Interacción mediante DMA



Interacción mediante DMA



Ejemplo (lectura de 100 datos)



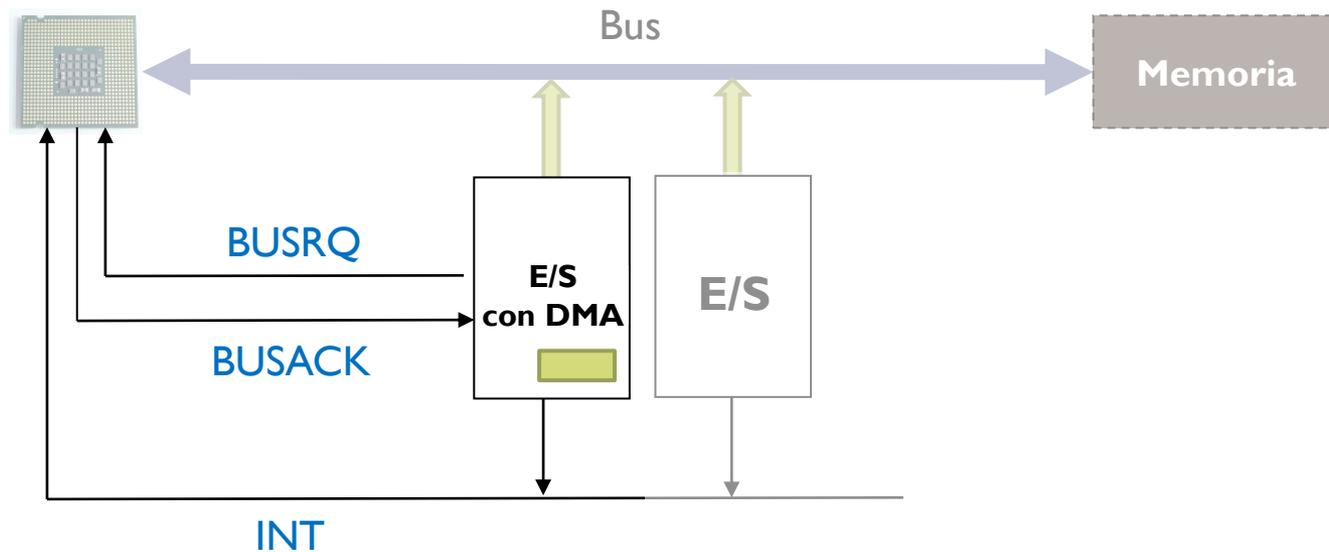
- ▶ Información de control
 - ▶ 0: leer
 - ▶ I: escribir
 - ▶ I0: dirección
 - ▶ 20: número de palabras
- ▶ Mapa de E/S común
 - ▶ Instrucciones lw y sw

```
main:    la $t0 datos
         sw $t0 0x500
         li $t0 100
         sw $t0 0x500
         li $t0 0
         sw $t0 0x500
         li $v0 xx
         syscall # sleep till interrupt ☺
         beq $v0 400 error
```

```
INT_05: lw $v0 0x508
         ret_int # restore registers & return
```

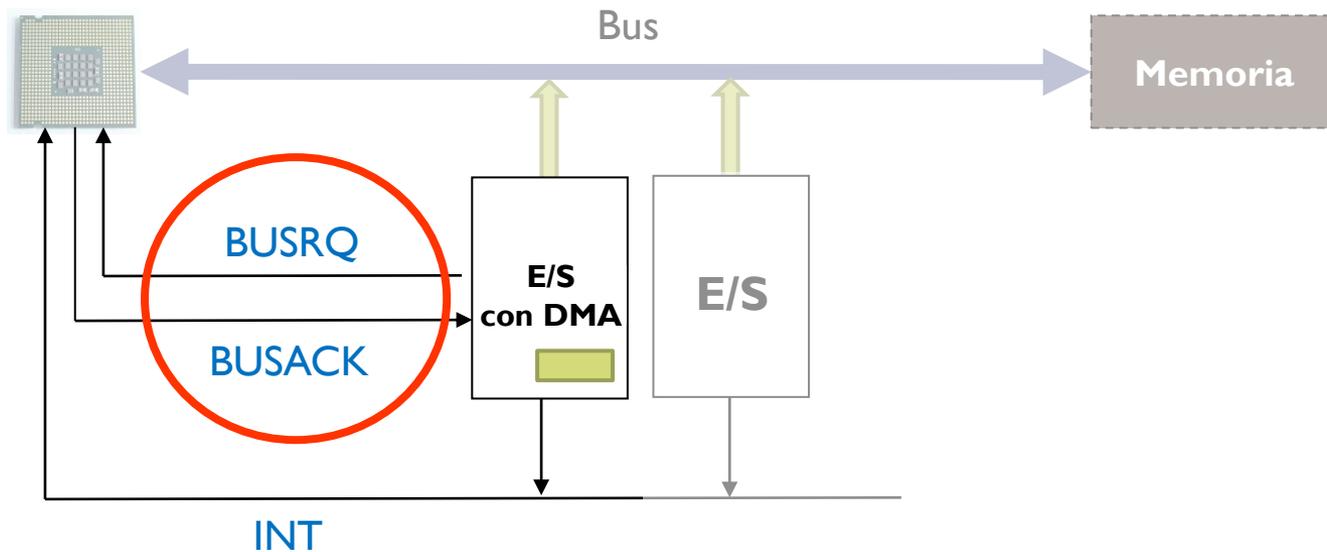
Interacción mediante DMA

Coordinación entre CPU y Módulos de E/S para acceder a memoria



Interacción mediante DMA

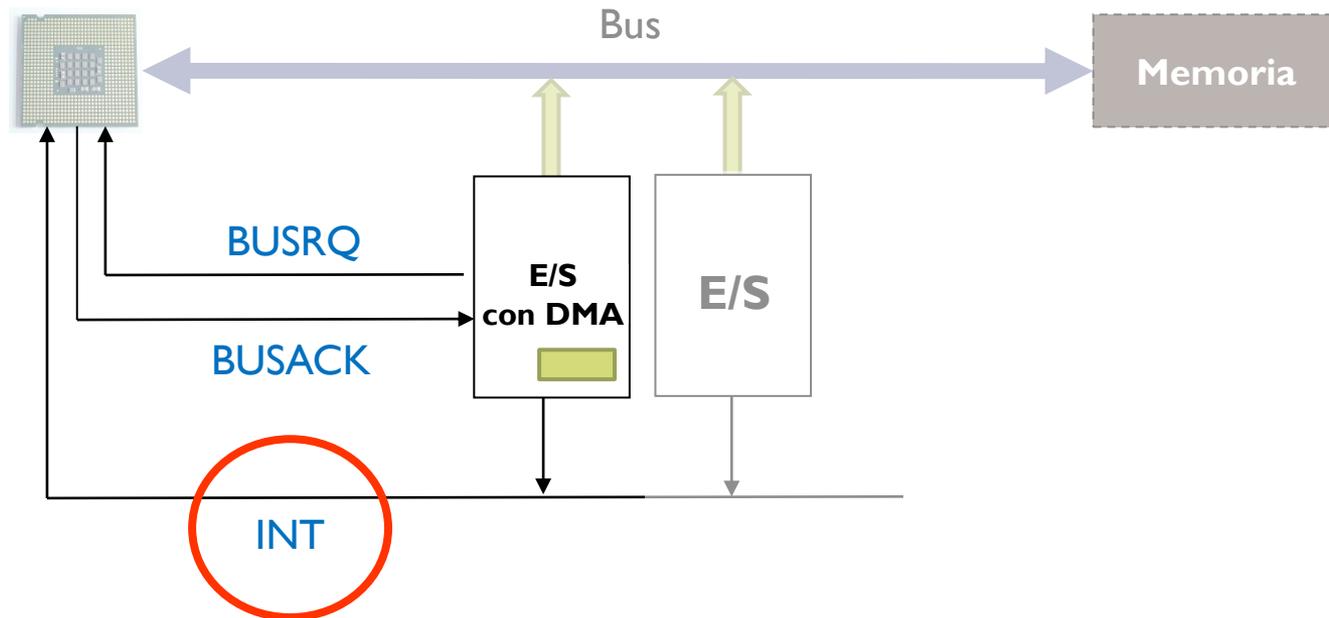
- Cada dato transferido a memoria supone:
- Pedir permiso para acceder a memoria (BUSRQ)
 - Esperar el permiso (BUSACK)
 - Transfiere a memoria
 - Desactiva petición de permiso (BUSRQ)



Interacción mediante DMA

Una vez transferido todos los datos:

- Generar una interrupción (INT) para avisar a la CPU



Curiosidades: Importancia de los controladores



- ▶ Estadísticas del kernel de Linux (2007-2008):
 - ▶ 9,2 millones de líneas de código.
 - ▶ Se incrementa un 10% cada año:
 - ▶ En media cada día se:
 - Añaden 4.500 líneas,
 - Borran 1.800 líneas
 - Modifican 1.500 líneas
 - ▶ La mayor parte del código es de los drivers:
 - ▶ El 55% del código son los controladores de dispositivo (o drivers)
 - Software parte del sistema operativo que la CPU ejecuta para trabajar con el dispositivo asociado
 - ▶ El núcleo del kernel ocupa un 5% y el resto (40%) se reparte entre soporte para las distintas arquitectura, el código de red, etc.



Tema 6

E/S y dispositivos periféricos

Grupo ARCOS

Estructura de Computadores
Grado en Ingeniería Informática
Universidad Carlos III de Madrid