

Grupo ARCOS

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

# Tema 6

## E/S y dispositivos periféricos

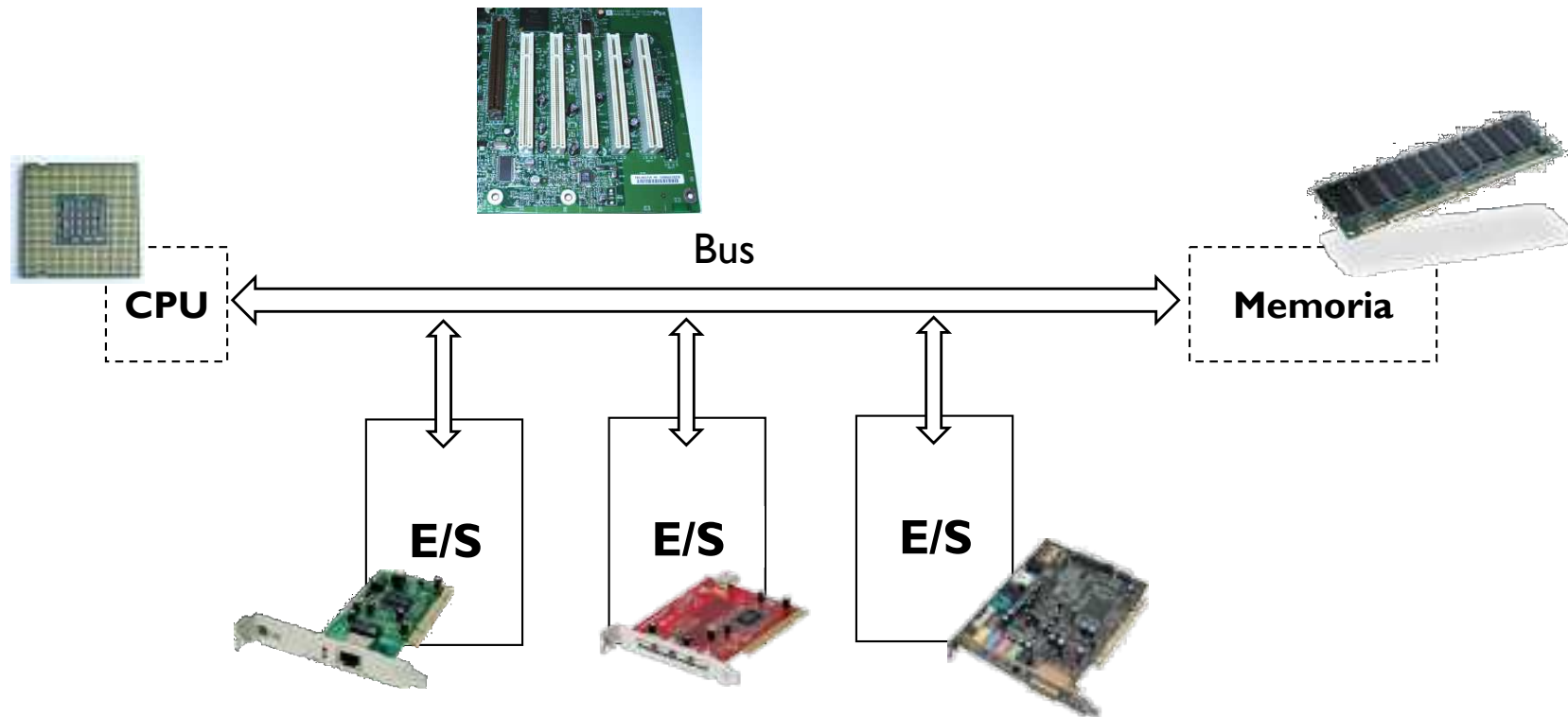
Estructura de Computadores  
Grado en Ingeniería Informática



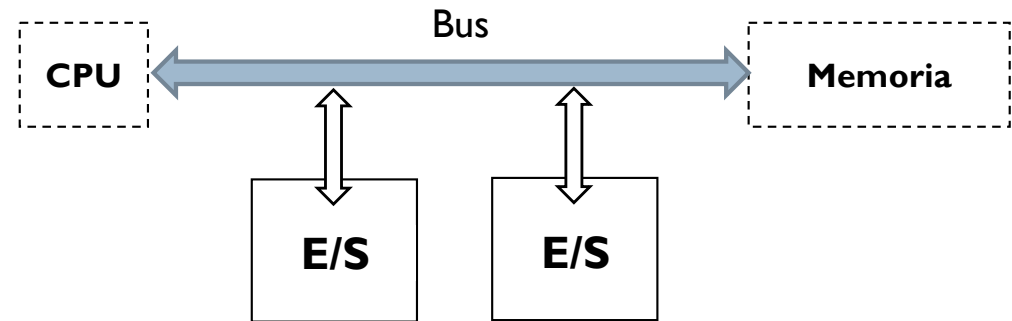
# Contenidos

1. Introducción
2. Periféricos
  - Concepto de periférico
  - Clasificación y tipos de periféricos
  - Estructura general de un periférico
    - Caso de estudio: disco duro, discos de estado sólido
3. Buses
  - Estructura y funcionamiento
  - Jerarquía de buses
4. Módulos de E/S
  - Estructura
  - Características
  - Técnicas de E/S

# Introducción

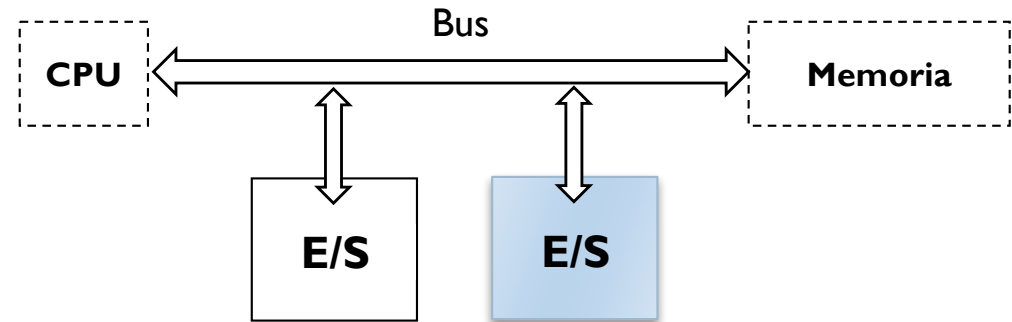


# Introducción



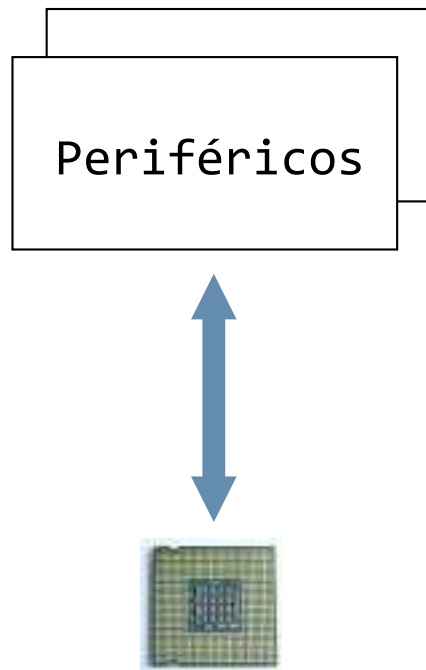
- Qué es un bus de interconexión

# Introducción



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

# Concepto de periférico



## ► Periférico:

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

# Clasificación de periféricos (por uso)



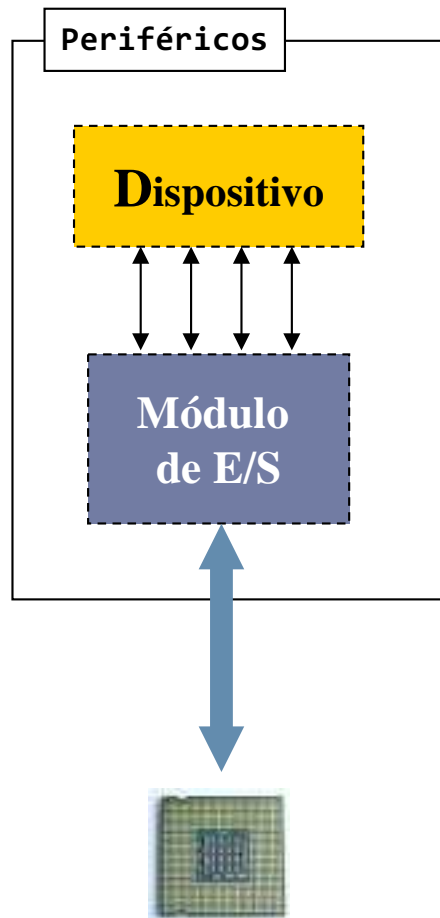
## ► **Comunicación:**

- **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

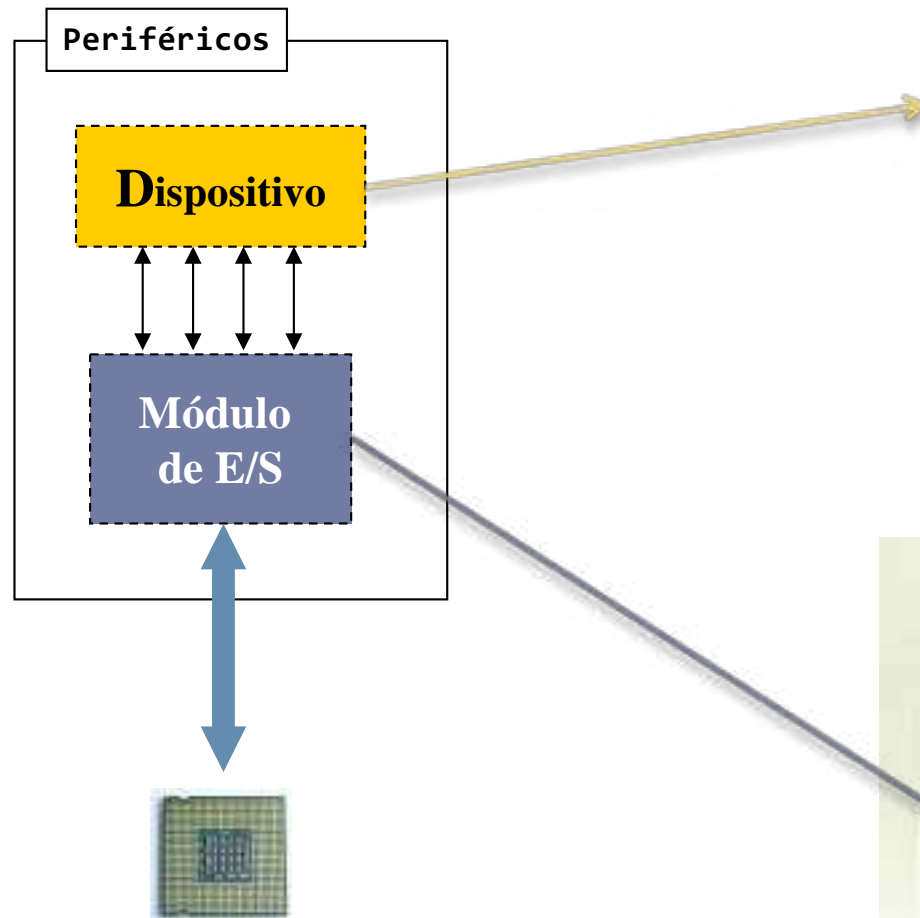


- ▶ Compuesto de:
  - ▶ **Dispositivo**
    - ▶ Hardware que interactúa con el entorno
  - ▶ **Módulo de Entrada/Salida**
    - ▶ También denominado **controlador**
    - ▶ Interfaz entre dispositivo y el procesador, que le oculta las particularidades de éste



# Ejemplo

## Disco magnético



# 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 (24") de diámetro
    - ▶ 5 MB de datos
    - ▶ Giraba a 3.600 revoluciones por minuto
    - ▶ Contaba con una velocidad de transferencia de 8,8 Kbps
    - ▶ Alquiler por año 35000 \$
    - ▶ Pesaba cerca de una tonelada

# Un poco de historia...

- ▶ En 1980 aparece el primer disco de 5 1/4”
  - ▶ 5 MB
  - ▶ 10.000 \$



- ▶ En 1997 aparece el primer disco a 15000 RPM

# Un poco de historia...



1956

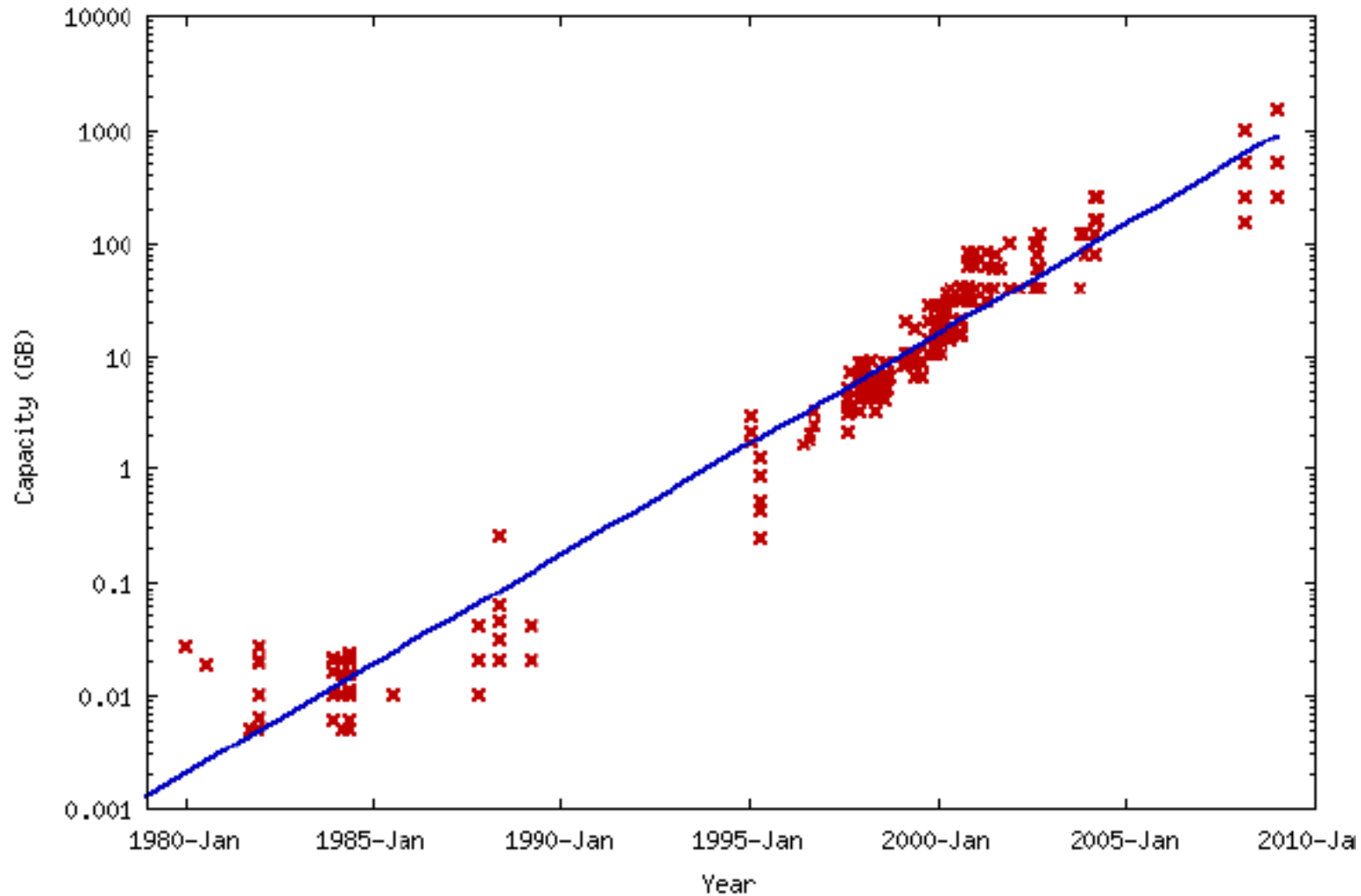


80's



2005

# Un poco de historia...

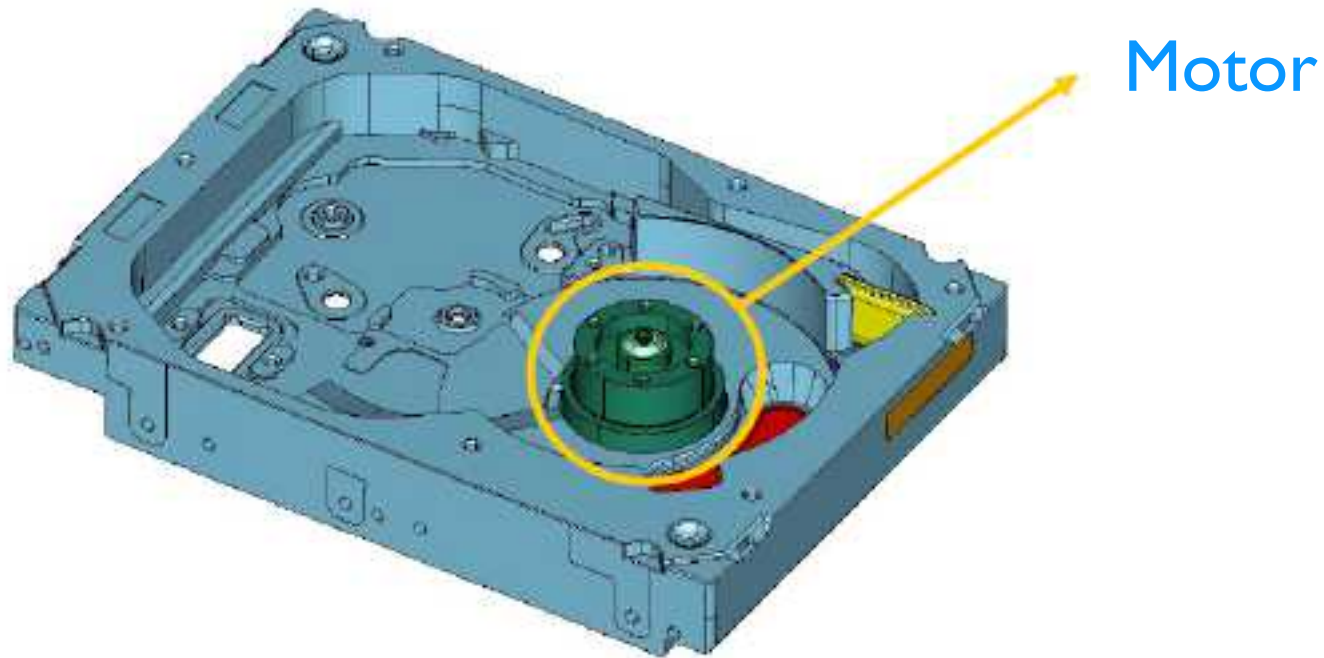


[http://en.wikipedia.org/wiki/File:Hard\\_drive\\_capacity\\_over\\_time.svg](http://en.wikipedia.org/wiki/File:Hard_drive_capacity_over_time.svg)

# Un poco de historia...

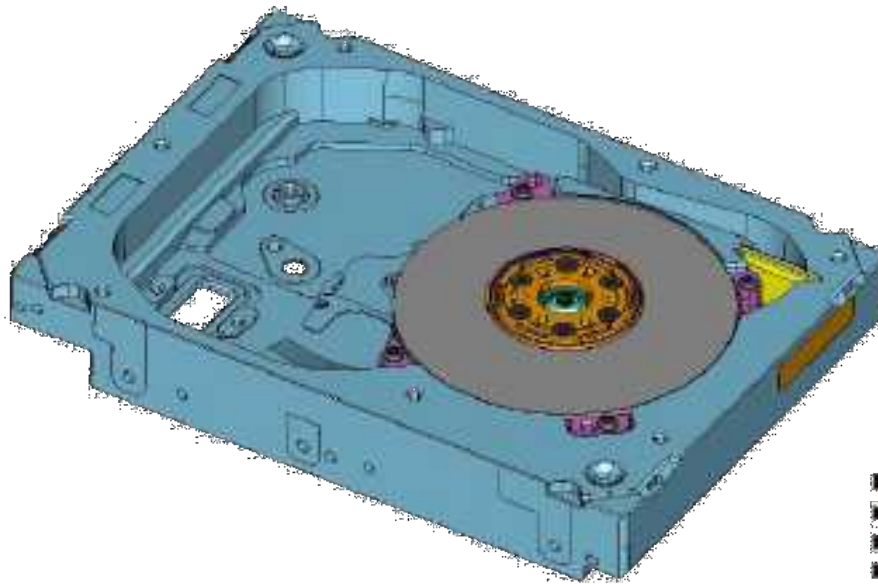
	<b>Tasa de crecimiento por año</b>
Capacidad	1.93/año
Coste	0.6/año
Prestaciones	0.05/año

# Anatomía de un disco duro



# Anatomía de un disco duro

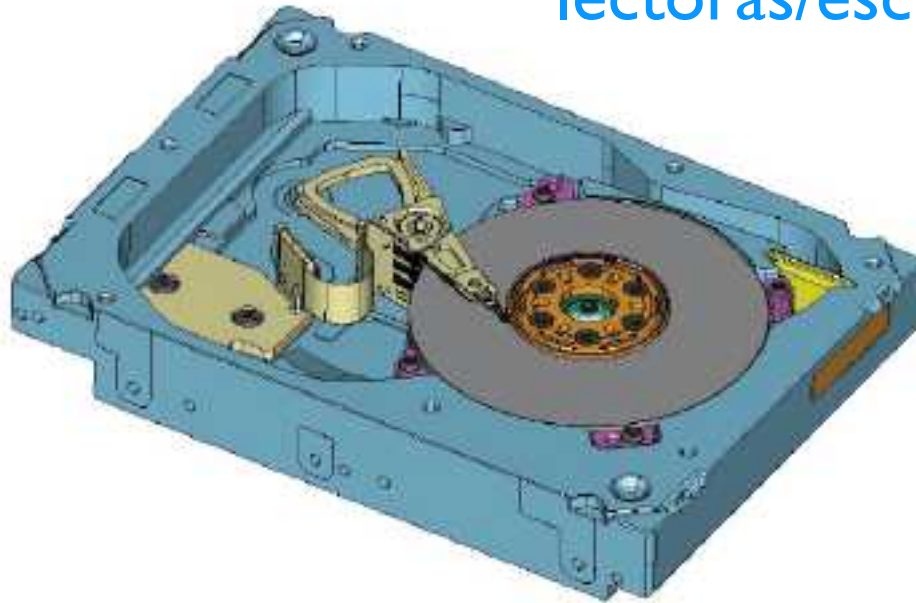
## Discos



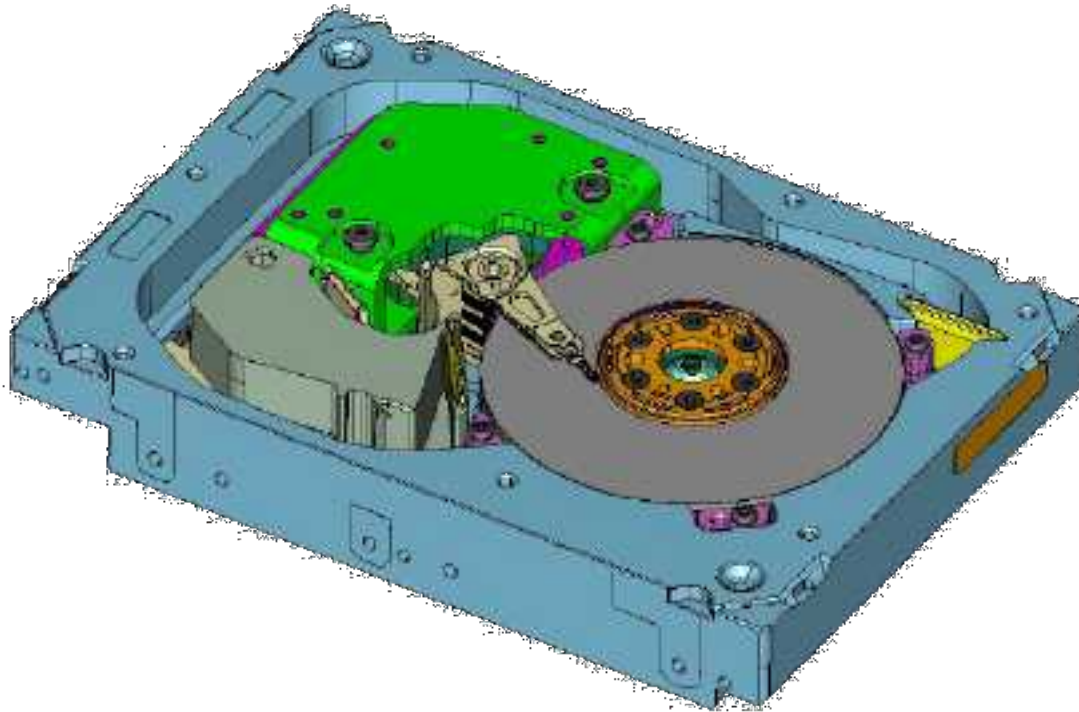


# Anatomía de un disco duro

Cabezas  
lectoras/escriptoras



# Anatomía de un disco duro



Módulo de control y  
mecánica

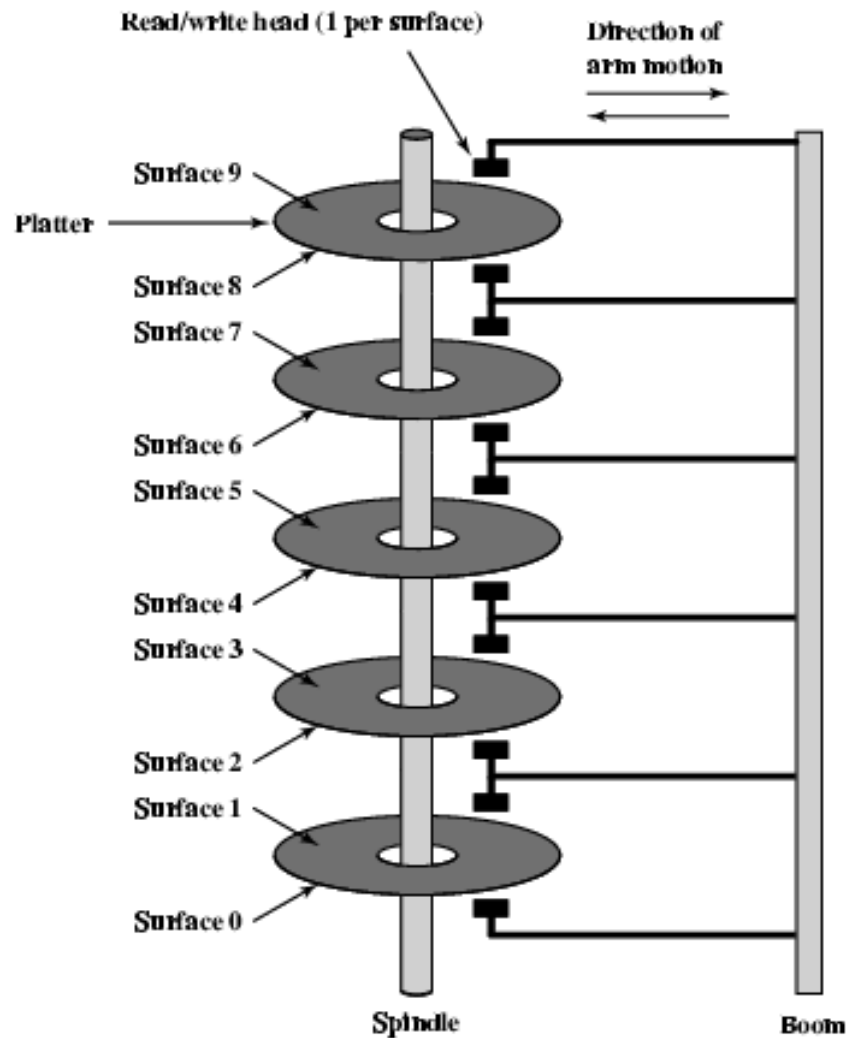
# Anatomía de un disco duro



<http://www.snia.org>

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

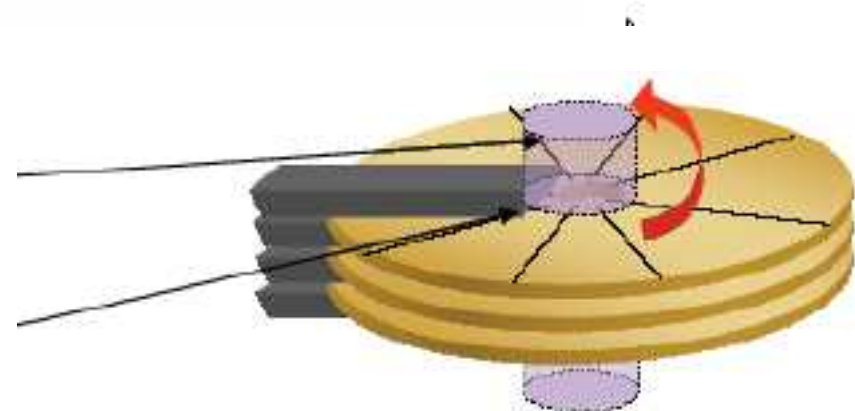
# Múltiples platos



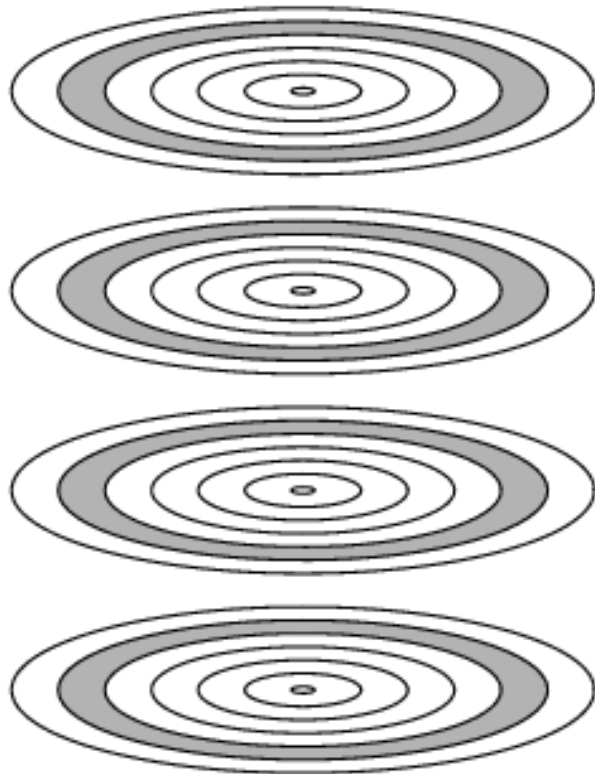
<http://www.snia.org>



Rotación

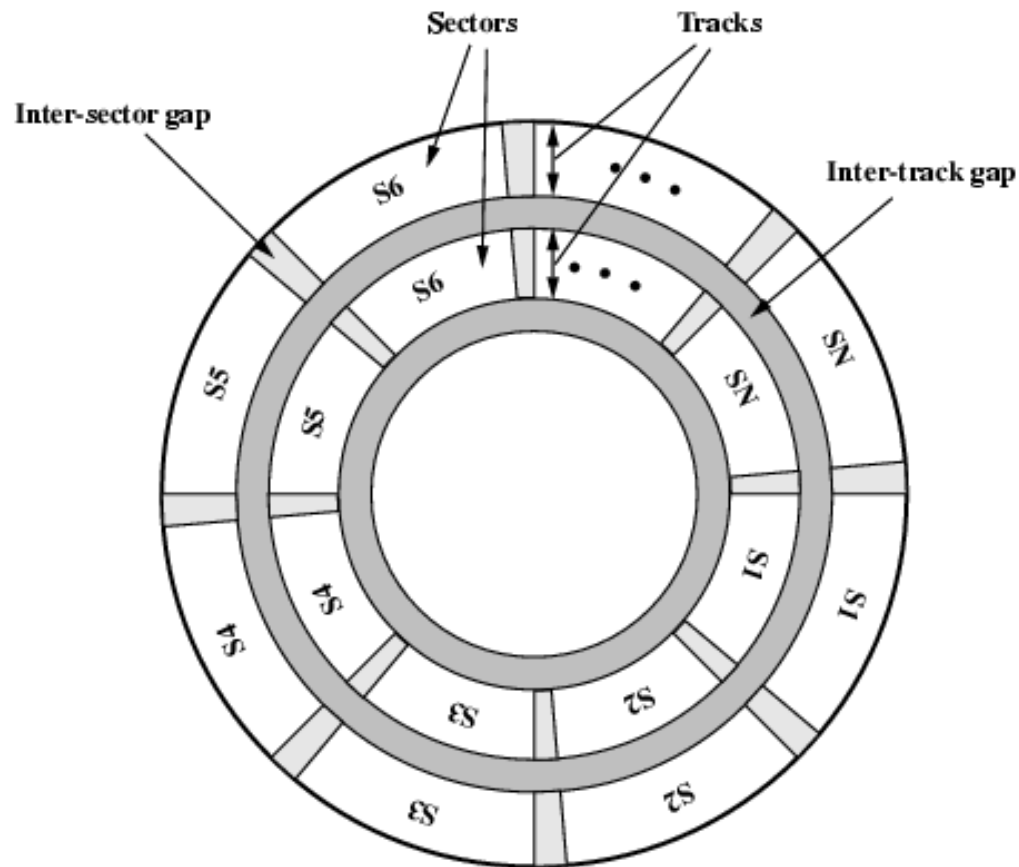


# Cilindros



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

# Almacenamiento: pistas y sectores



## Pista:

- Un anillo del plato

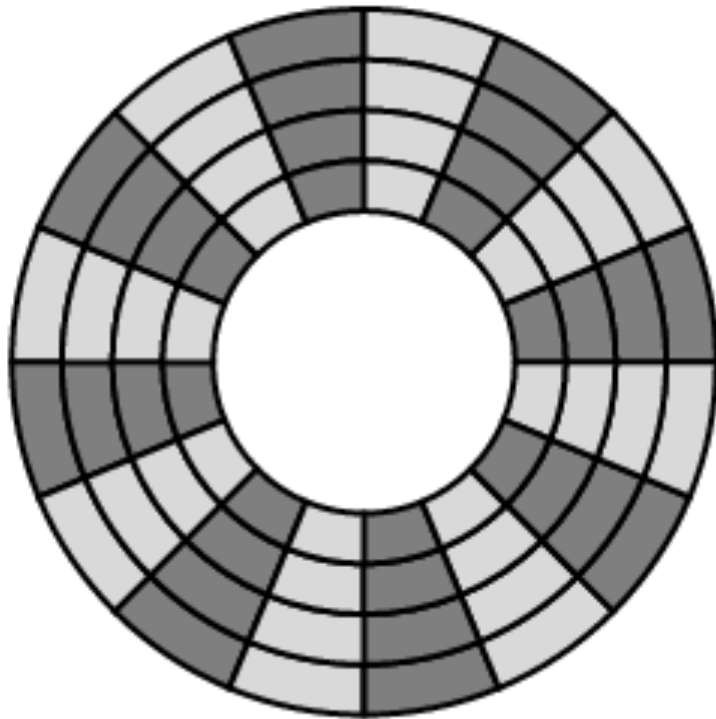
## Sector:

- División de la superficie del disco realizada en el formateo (típicamente 512 bytes)

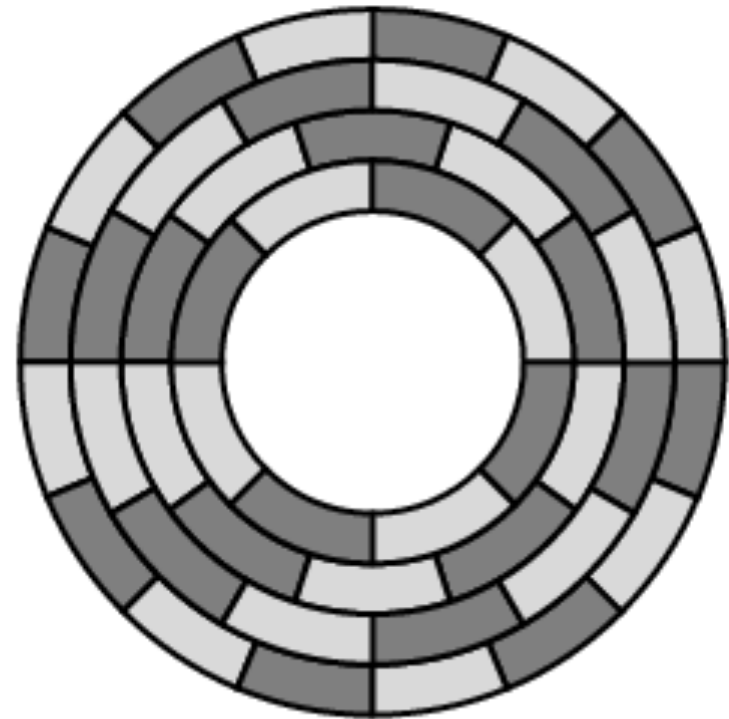
## Bloques:

- El sistema de ficheros escribe en bloques
- Grupo de sectores

# Distribución de sectores



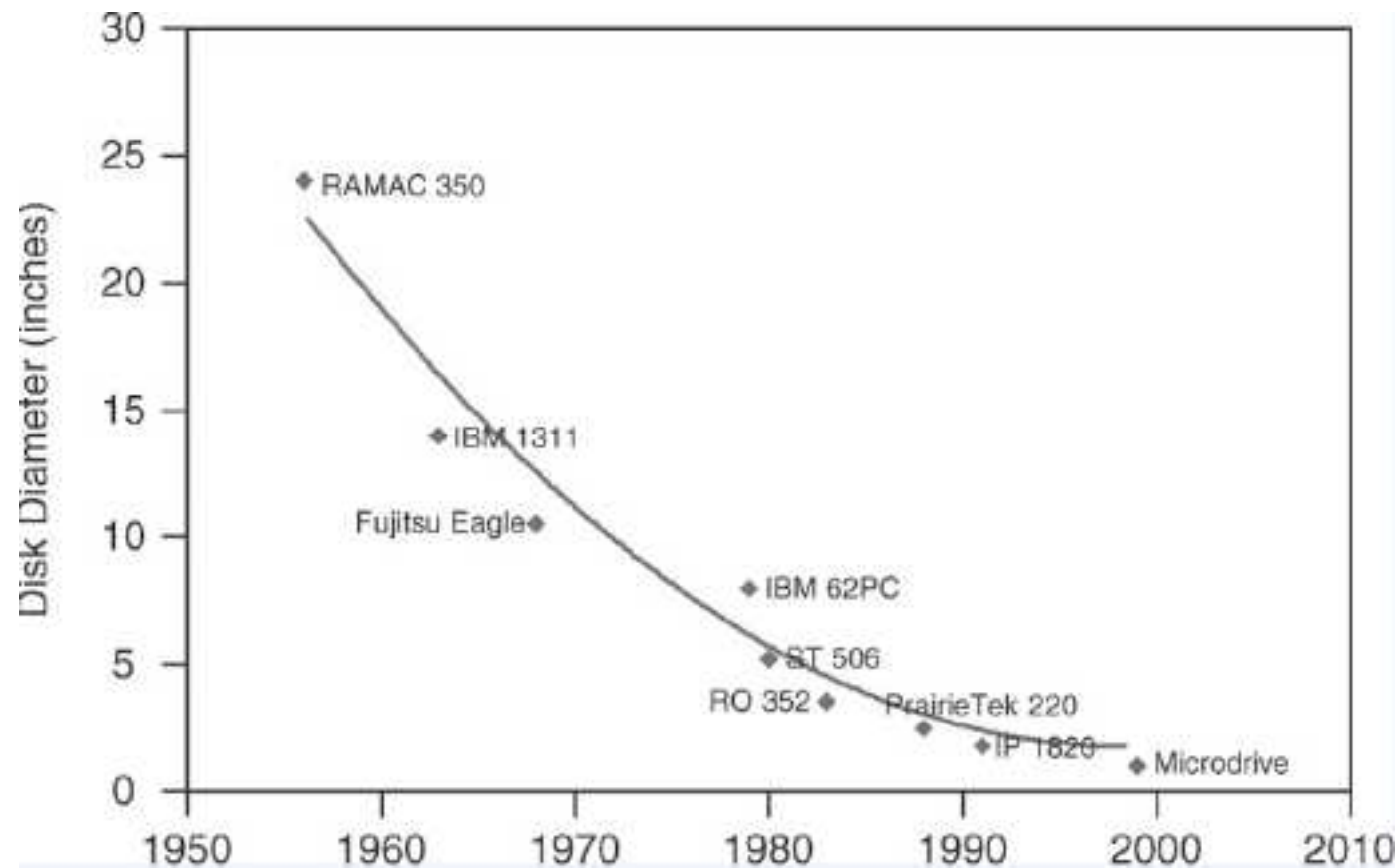
**(a) Constant angular velocity**



**(b) Multiple zoned recording**



# Evolución del tamaño de los discos



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



# Medida de la capacidad

- ▶ Bits por pulgada cuadrada
  - ▶ Dependen de la cabeza de lectura/escritura, del medio de grabación, de la rotación del disco y de la velocidad a la que el bus puede aceptar datos.
- ▶ Pistas por pulgada
  - ▶ Dependen de la cabeza de lectura/escritura, el medio de grabación, la precisión con la que la cabeza puede posicionarse y la capacidad del disco para girar en un círculo perfecto

# 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

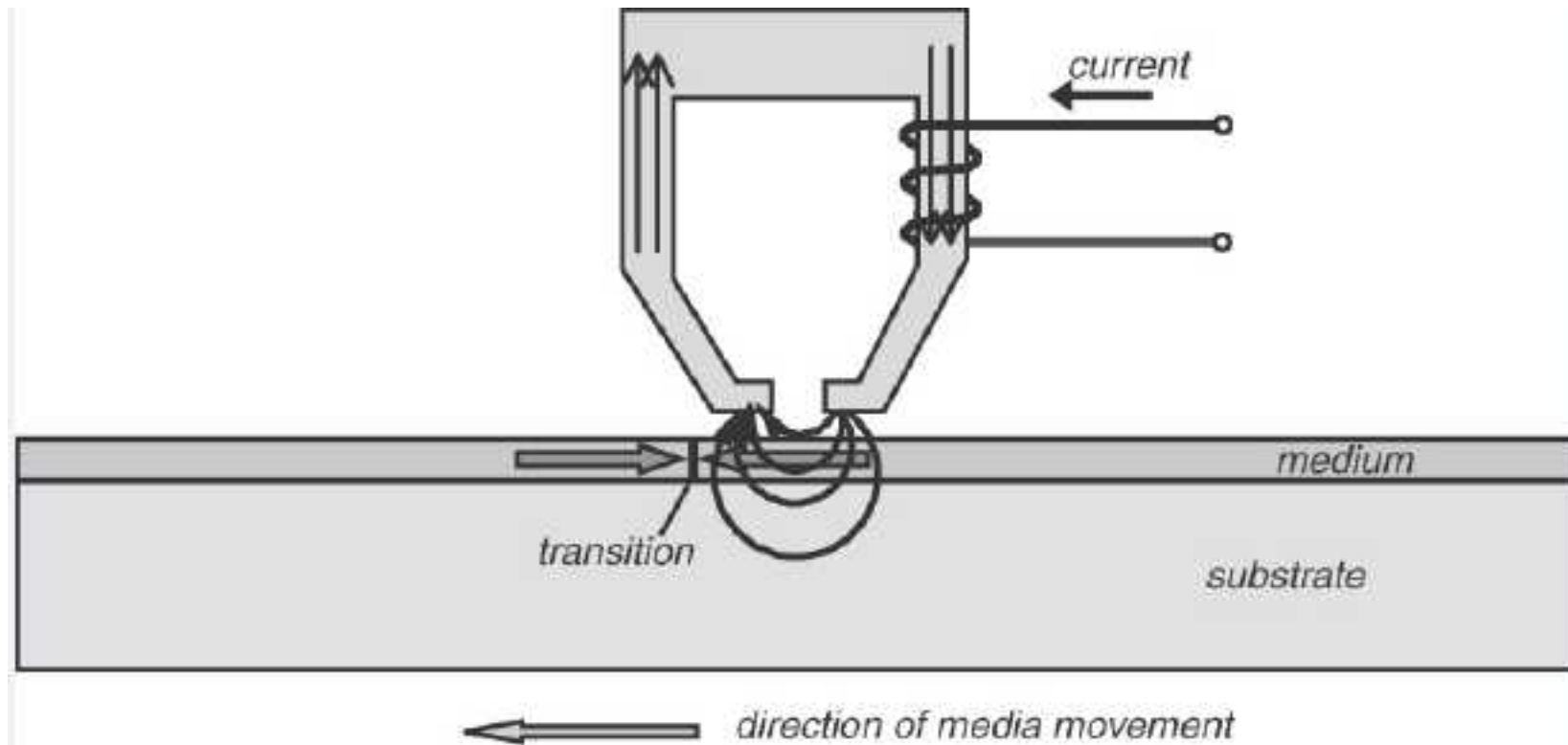
- ▶  $z$ : número de zonas
- ▶  $p_i$ : pistas de la zona  $i$
- ▶  $s_i$ : sectores por pista de la zona  $i$

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

# Métodos de grabación

- ▶ En la última década se ha incrementando en un 100% la densidad de grabación
- ▶ Cada bit en una pista consta de múltiples granos magnéticos
- ▶ El tamaño de estos granos no puede reducirse por debajo de los 10 nanómetros debido a:
  - ▶ Efecto superparamagnético
  - ▶ La temperatura ambiente podría afectar a la estabilidad de los granos
- ▶ Técnicas de grabación:
  - ▶ Longitudinal: guardan datos de forma longitudinal sobre un plano horizontal
  - ▶ Perpendicular: los datos se alinean de forma vertical, lo que permite disponer de mayor espacio y guardar más información

# Cabeza lectora/escritora



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

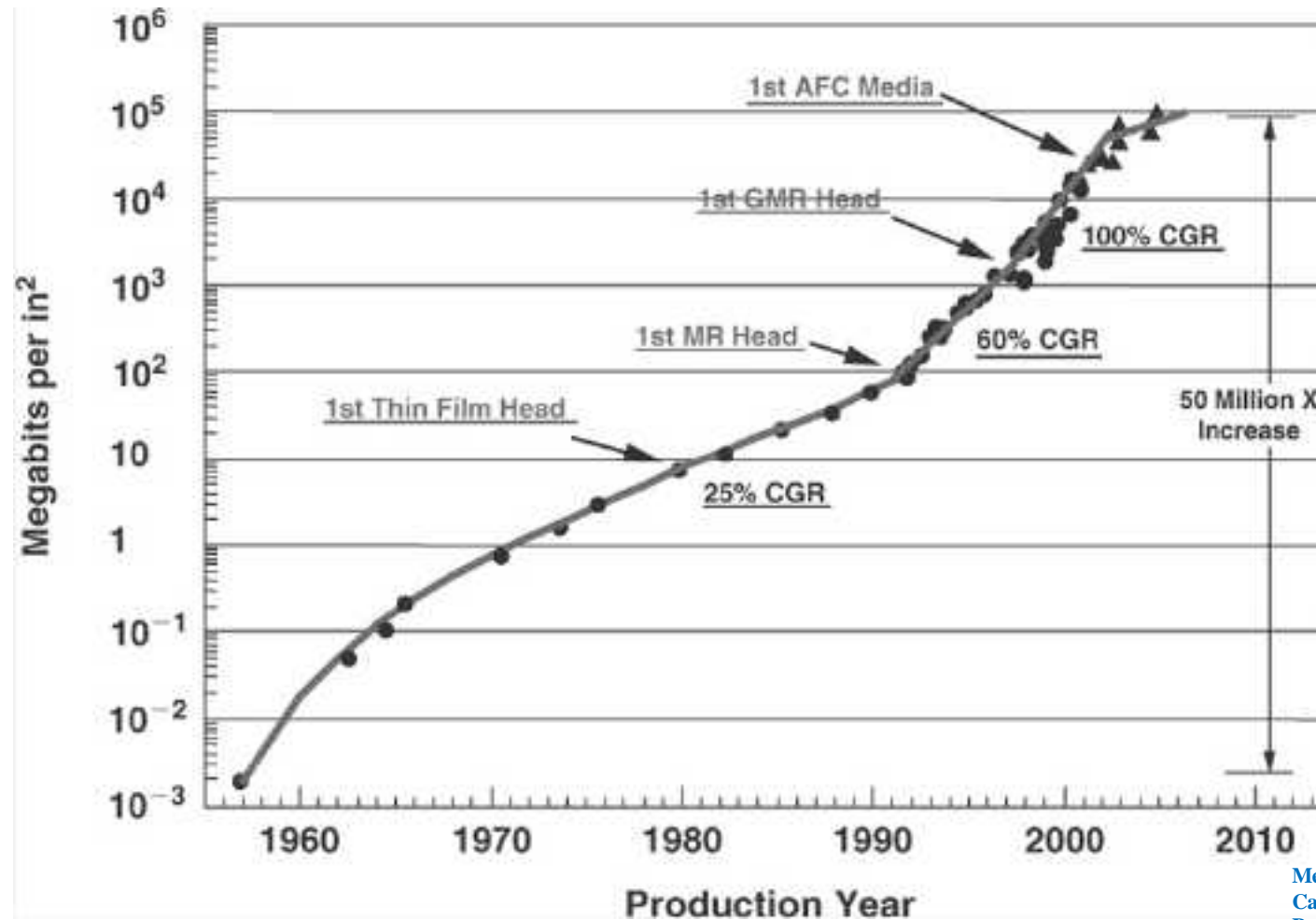
# Densidad superficial

- ▶ Las mejoras en la capacidad de los disco se expresa como mejorar en la **densidad superficial**:

$$\text{Areal density (AD)} = \frac{\text{Tracks on a disk surface}}{\text{Inch}} \times \frac{\text{Bits on a track}}{\text{Inch}}$$

- ▶ Hasta 1988 la tasa de mejora anual en AD fue del 29%
- ▶ Entre 1988 y 1996 la tasa de mejora fue del 60% al año
- ▶ Entre 1997 y 2003 la tasa de incremento fue del 100% anual
- ▶ Entre 2003-2011 la tasa de incremento ha sido del 30% anual
- ▶ En 2011 la mayor densidad en productos comerciales fue de 400 billones de bits por pulgada cuadrada
- ▶ El coste por bit se ha mejorado en un factor de 1.000.000 entre 1983 y 2011

# Evolución de la densidad

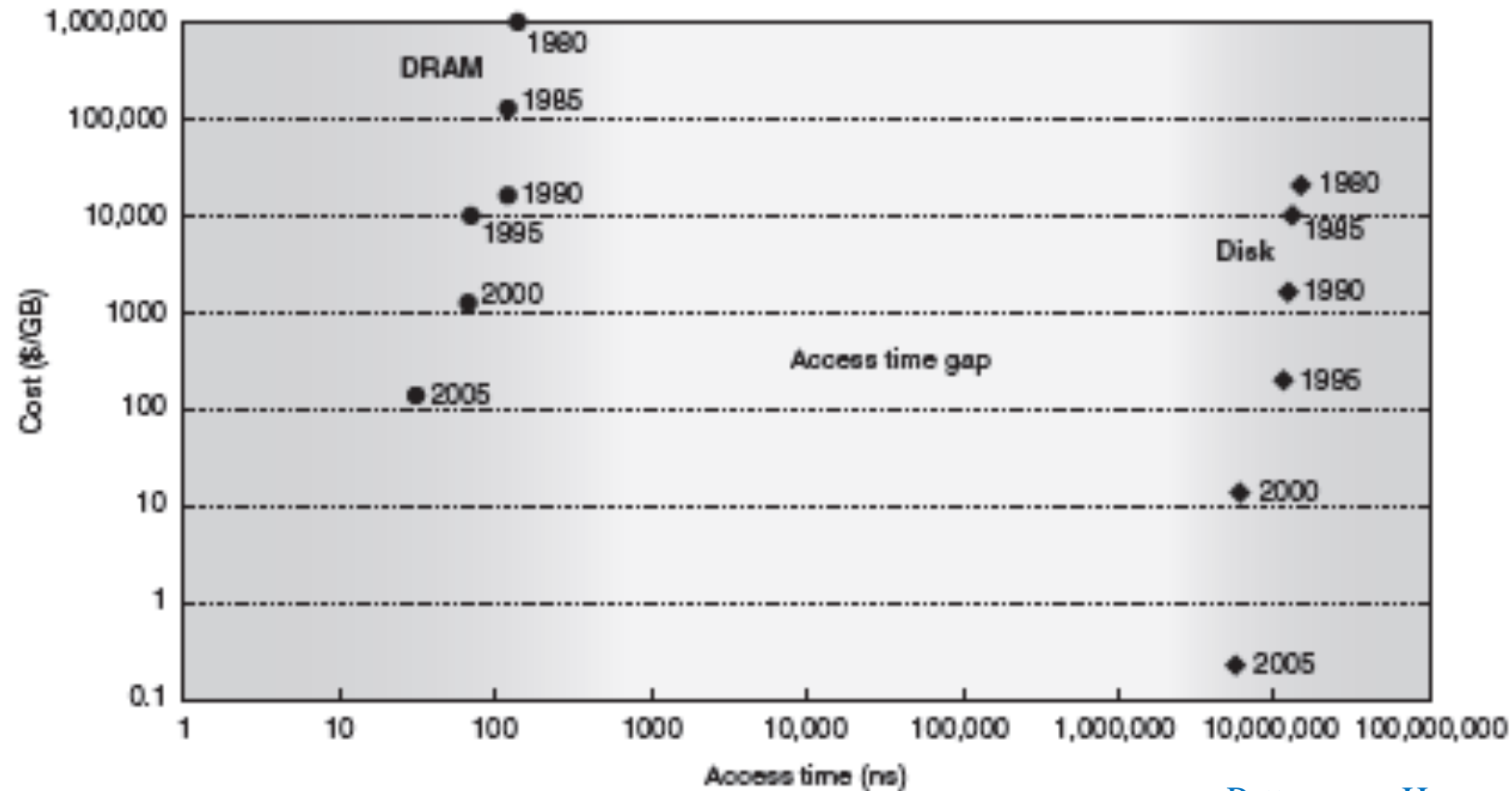


Memory Systems  
Cache, DRAM, Disk  
Bruce Jacob, Spencer Ng, David Wang  
Elsevier

# Discos y memoria principal

- ▶ La latencia de una memoria DRAM es 100.000 veces menor que la de un disco
- ▶ El coste por GB para una memoria DRAM está entre 30 y 150 veces el coste por GB de un disco duro
- ▶ En 2015:
  - ▶ Un disco de 8 TB transfiere 190 MB/s con un coste 250 \$
  - ▶ Un módulo de 8 GB de DDR4 transfiere 25 GB/s y cuesta aprox. 70\$

# Discos y memoria principal



Patterson y Hennesy

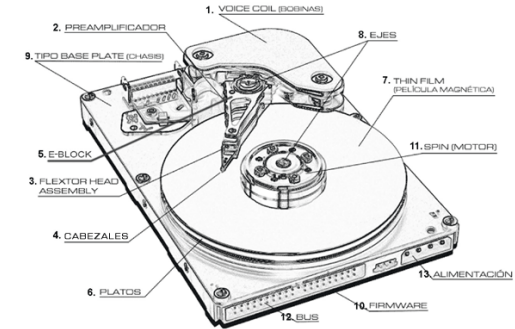


# Direccionamiento

- ▶ Tipos de direccionamiento
  - ▶ **Direccionamiento físico**: cilindro-pista-sector. Un sector queda determinado por estos tres valores
  - ▶ Direccionamiento de **bloques lógicos** (LBN)
    - ▶ Cada sector se identifica con un bloque lógico y la correspondencia la hace el propio disco
- ▶ Los controladores de disco actuales realizan la traducción de LBN a dirección física

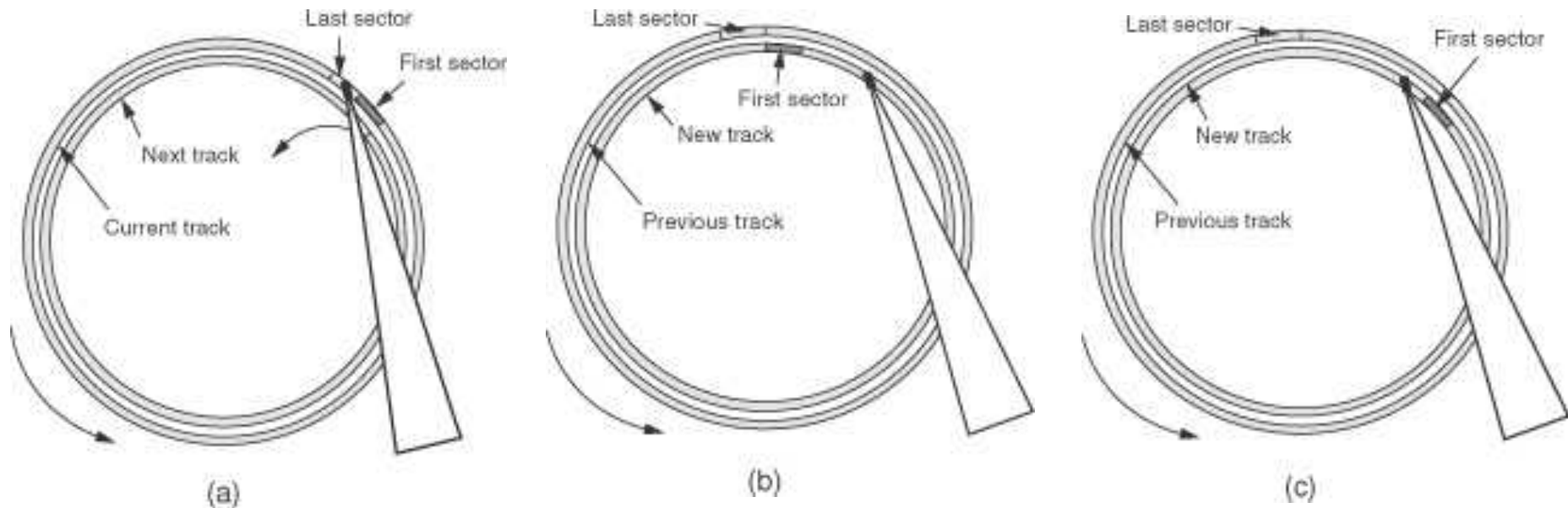
# Tiempo de acceso

►  $T_{\text{acceso}} = T_{\text{búsqueda}} + T_{\text{latencia}} + T_{\text{transferencia}}$



- **Tiempo de búsqueda** ( $T_{\text{búsqueda}}$ ): tiempo necesario para mover la cabeza desde el cilindro actual al cilindro sobre el que se quiere operar
- **Latencia de rotación** ( $T_{\text{latencia}}$ ): 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** ( $T_{\text{transferencia}}$ ): tiempo necesario para recorrer un sector y transferir los datos de él

# Posicionamiento y rotación



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

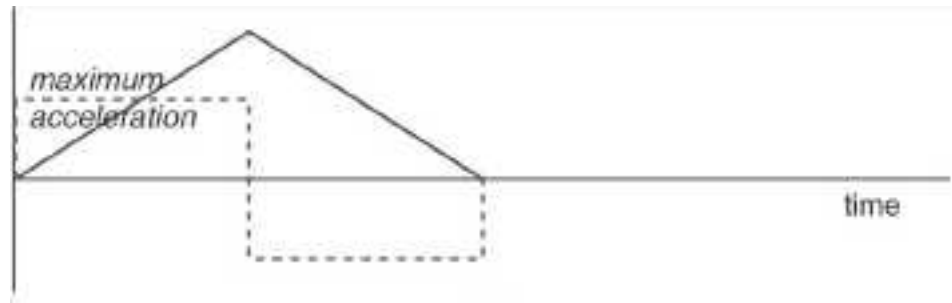
# Tiempo de posicionamiento

- ▶ Elementos que incluye:
  - ▶ Tiempo de aceleración de la cabeza ( $T_{acc}$ )
  - ▶ Tiempo con velocidad constante ( $T_c$ ) para distancias largas
  - ▶ Tiempo de deceleración ( $T_{dec}$ )
  - ▶ Tiempo de colocación ( $T_s$ ). Este tipo domina al resto de tiempos
  - ▶ El tiempo de posicionamiento medio se toma como el tiempo de posicionamiento entre dos pitas aleatorias

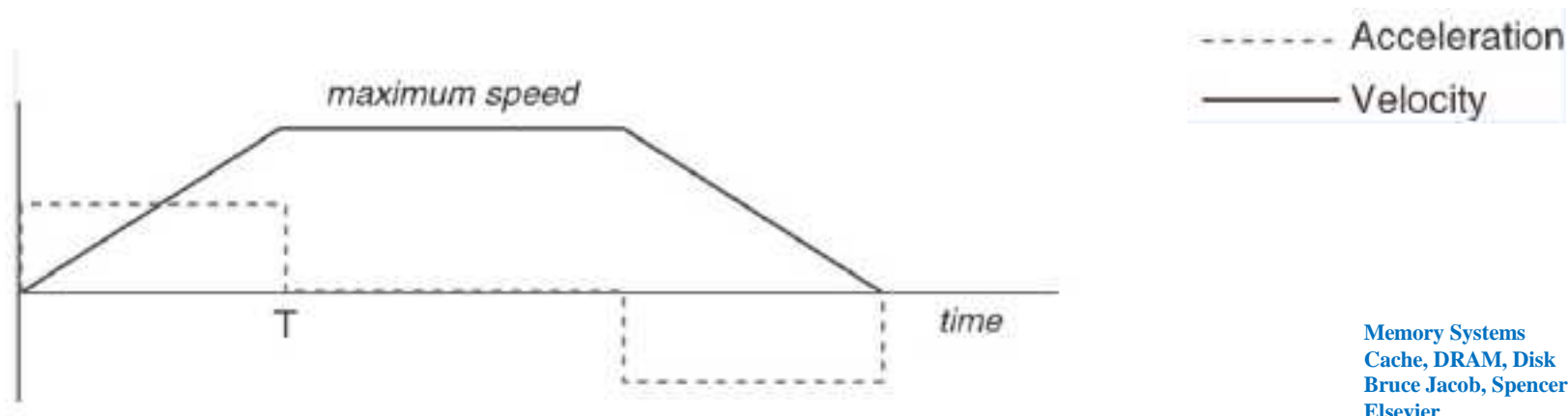
$$D_{average} = \frac{1}{2} \times a_{acc} \times t_{acc}^2 + a_{dec} \times t_{dec}^2$$

# Tiempo de posicionamiento

## ► Posicionamiento entre pistas cercanas



## ► Posicionamiento entre pistas lejanas



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

# Tiempo de posicionamiento

- Cuando la aceleración y deceleración son iguales:

$$t_{acc} = t_{dec} = \sqrt{\frac{D_{average}}{a}}$$

$$t_{seek} = 2 \times \sqrt{\frac{D_{average}}{a}} + t_s$$

# Tiempo de posicionamiento

- ▶ El aumento en la densidad de grabación da lugar a cilindros con mayor capacidad
  - ▶ Incrementa la probabilidad de reducir el número de posicionamientos de la cabeza
  - ▶ Aumenta la probabilidad de que los siguientes datos a solicitar estén en el mismo cilindro lo que reduce los posicionamientos

# Latencia rotacional

- ▶ La latencia media es el tiempo en dar media vuelta

$$T_{rotate} = \frac{1}{2} \times \frac{60 \times 10^3}{RPM}$$

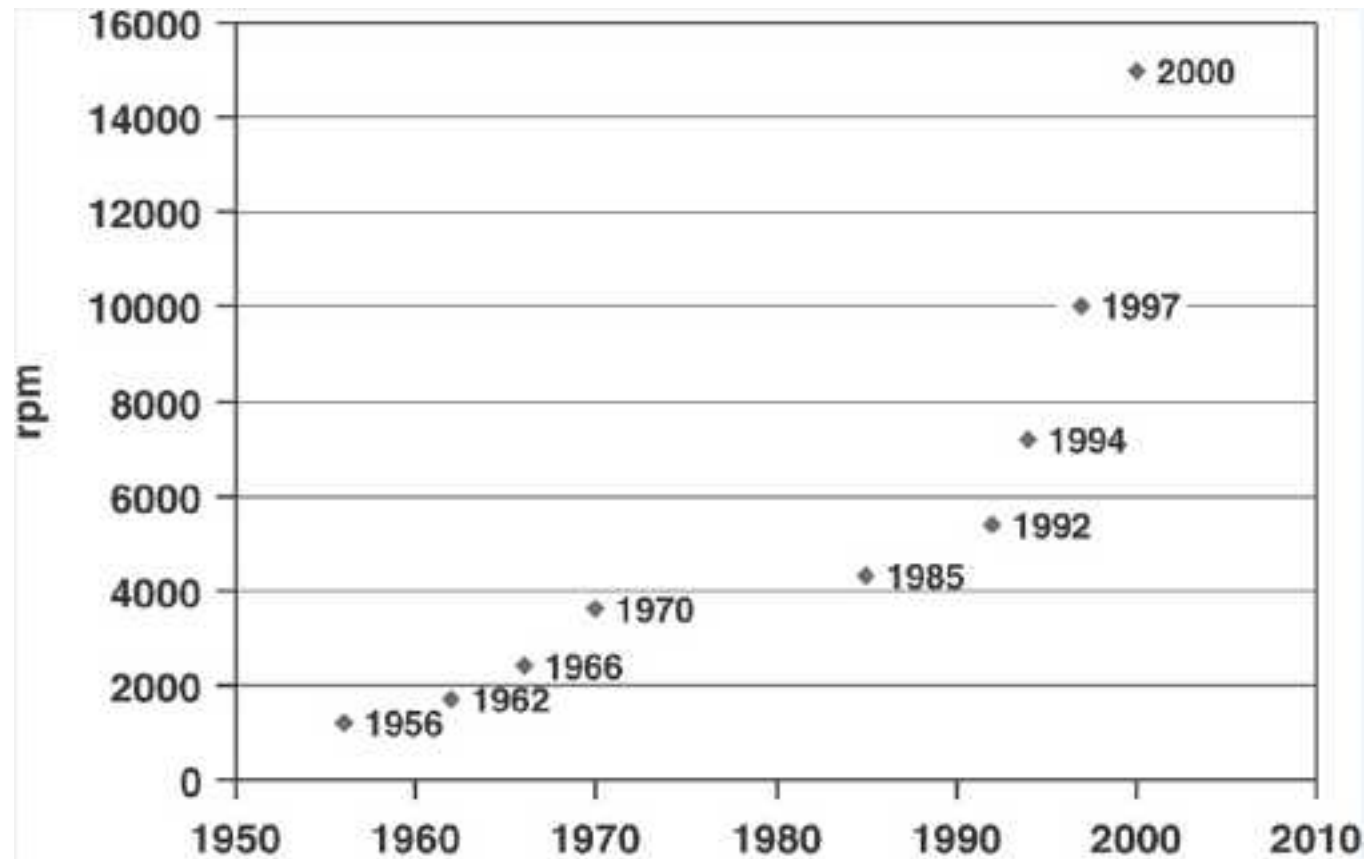
- Disco con latencia de acceso cero (*Zero-latency access*)
  - ▶ Transfieren los datos tan pronto como la cabeza se encuentra en la pista deseada a un bufer donde los bloques son después reordenados



# Velocidad y latencia

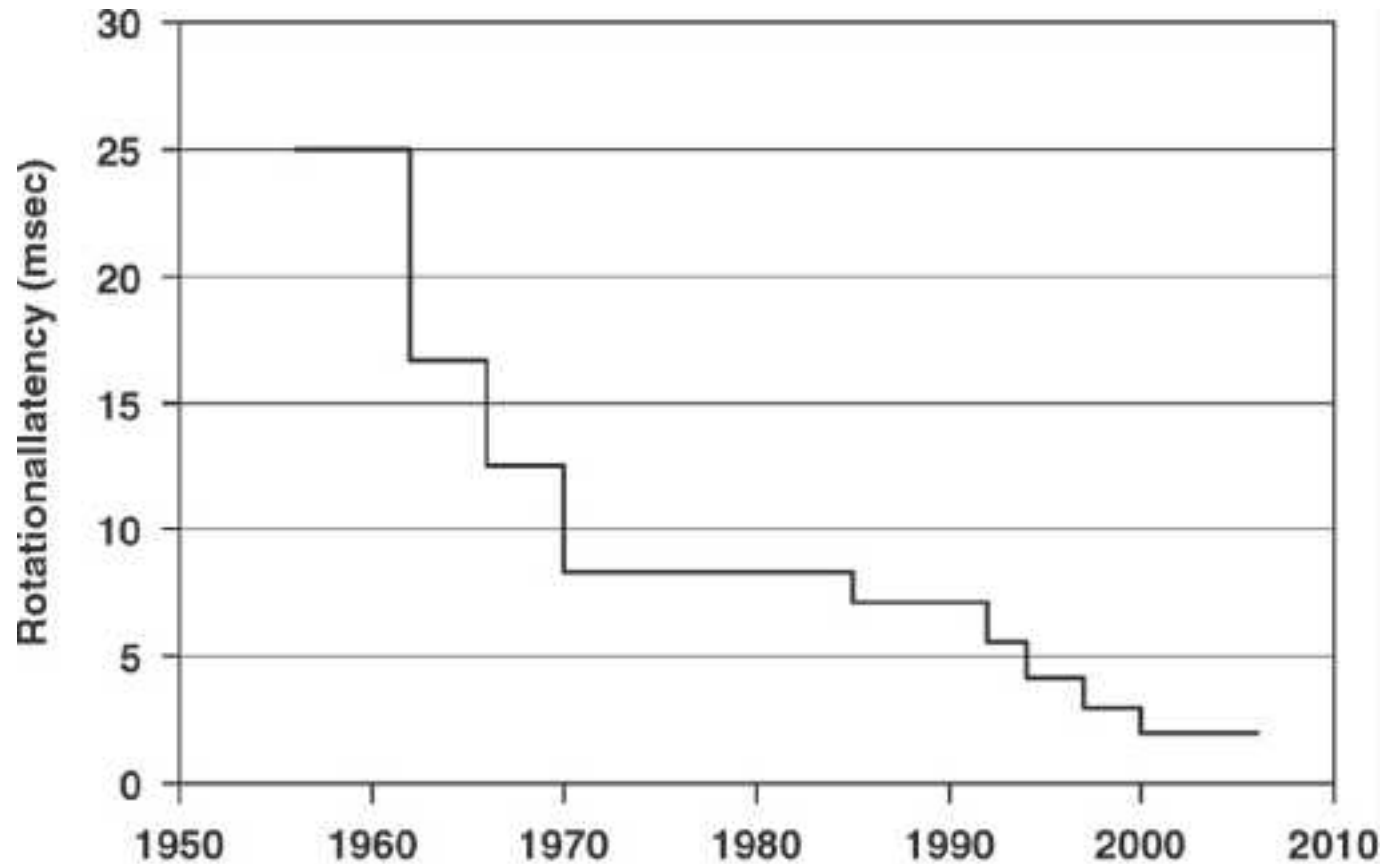
Rotational Speed	Rotational Latency (in ms)
5400	5.6
7200	4.2
10000	3.0
12000	2.5
15000	2.0

# Evolución de las RPM



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

# Evolución de la latencia rotacional



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

# Tiempo de transferencia

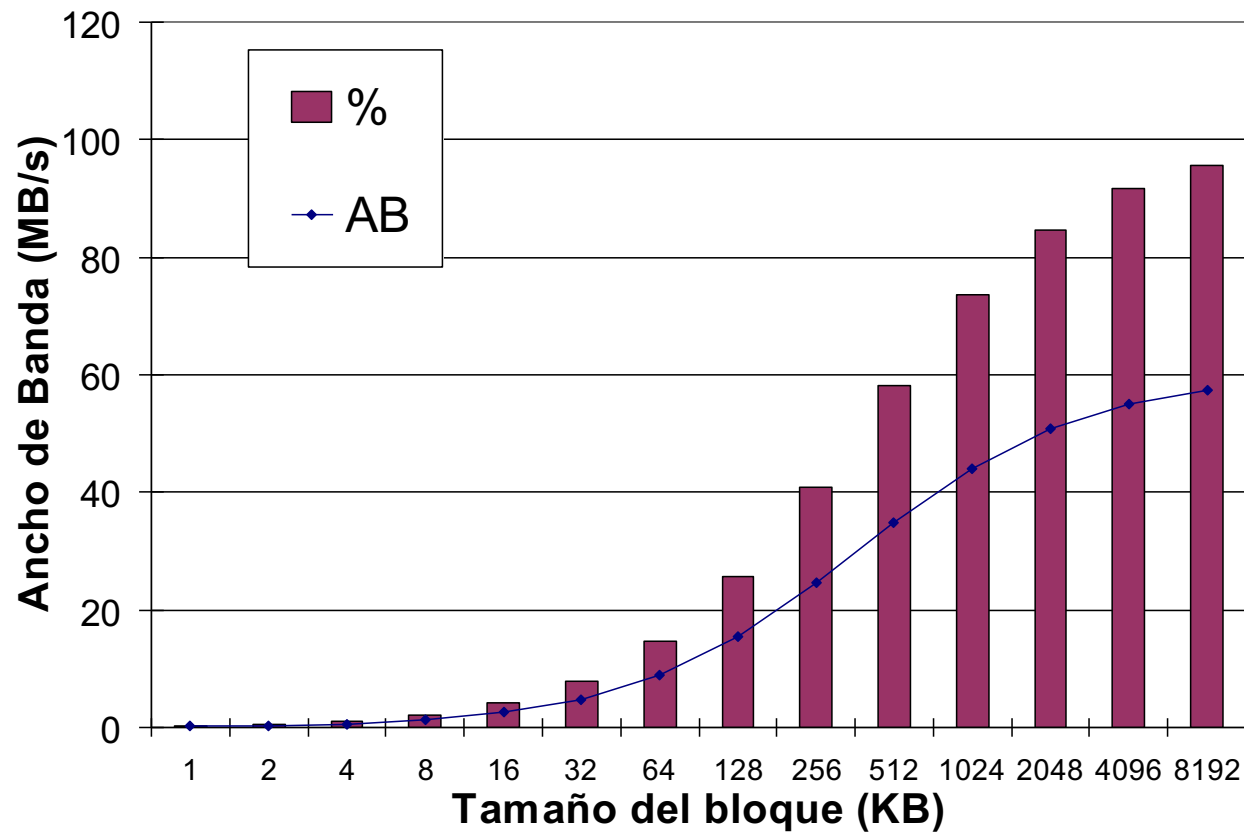
- ▶ Consta de dos partes:
  - ▶ Tasa de transferencia externa entre la memoria y la caché del disco
  - ▶ Tasa de transferencia interna entre la caché de disco y el medio de almacenamiento
- ▶ La tasa de transferencia interna se mide como:

$$T_{transfer} = \frac{N_{request}}{N_{track}} \times \frac{60}{RPM}$$

- ▶ Donde  $N_{track}$  es el número de sectores en una pista y  $N_{request}$  la cantidad de datos solicitados en número de sectores.
- ▶ En los discos con pistas externas más densas la relación de sectores de las pistas más externas a las más internas suele estar entre 1.43 y 1.58.

# Efecto del tamaño de la petición

- Efecto del tamaño de bloque ( $t_a=6$  ms y  $AB = 60$  MB/s)



# Controlador de disco

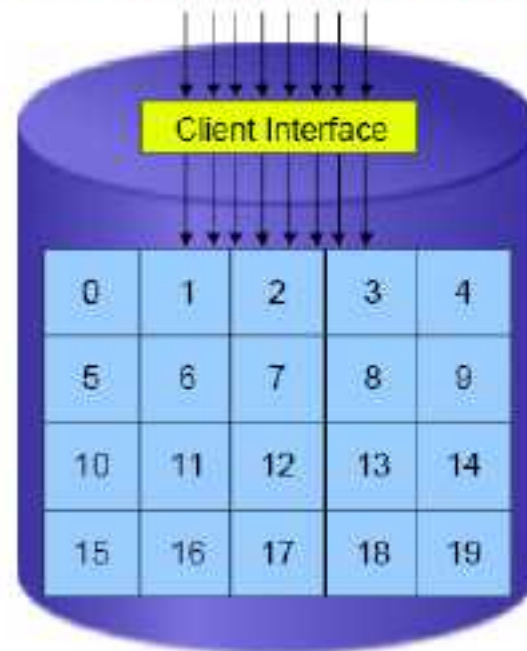
- ▶ Circuitos y componentes responsables de controlar el disco:
  - ▶ Interfaz de almacenamiento
  - ▶ Secuenciador de disco
  - ▶ Códigos correctores de errores (ECC)
  - ▶ Control del motor
  - ▶ Microprocesador
  - ▶ Controlador del bufer
  - ▶ Caché de disco

# Interfaz de almacenamiento

- ▶ Ofrece un protocolo estándar para comunicar el disco con el sistema.
  - ▶ IDE, SCSI, FC, SATA
- ▶ Suelen ofrecer la capacidad del disco como un array lineal de bloques de tamaño fijo
- ▶ El acceso se especifica con un LBN y el número de bloques a transferir
  - ▶ El controlador del disco traduce el LBN a dirección física (Cilindro, pista, sector)
- ▶ Existen otros tipos de dispositivos: *Object-based Storage Devices* (OSD) que crean objetos en el dispositivo mediante interfaces más potentes y amplias

# Acceso a un disco

SCSI, SAS, FCP, SRP, iSCSI, iSER, ATA, SATA



Physical Blocks:  
e.g. 512 bytes



# Caché de disco

- ▶ Explota la localidad
- ▶ Reduce los accesos físicos al disco
  - ▶ Reduce la disipación de calor
  - ▶ Mejora el rendimiento
- ▶ No suele exhibir localidad temporal debido a las cachés de datos de los SSOO
- ▶ Normalmente implementa lectura adelantada (*prefetch*) para mejorar la localidad espacial
- ▶ Las cachés de disco normalmente se dividen en segmentos independientes correspondientes a flujos secuenciales de datos

# Caché de disco

- ▶ Algoritmos de reemplazo
  - ▶ Random Replacement
  - ▶ Least Frequently Used (LFU)
  - ▶ Least Recently Used (LRU)
- ▶ Su tamaño debería estar entre e. 0.1 y el 0.3 % del tamaño del disco

# Planificador de disco

- ▶ Los dispositivos actuales incluyen un planificador de las peticiones a disco cuyo objetivo es mejorar el tiempo de respuesta y minimizar el movimiento de las cabezas
- ▶ Políticas:
  - ▶ First Come First Served (FCFS)
  - ▶ Shortest Seek Time First (SSTF)
  - ▶ SCAN
  - ▶ C-SCAN
  - ▶ LOOK

# Otros elementos del controlador

- ▶ **Secuenciador de disco**: gestiona las transferencias de datos entre la interfaz de almacenamiento y el buffer de datos
- ▶ **ECC**: comprueba y corrige errores mediante códigos de corrección
- ▶ **Control del motor**: detecta la posición actual de la cabeza y controla el avance a la pista futura
- ▶ **Microprocesador**: controla el funcionamiento global del disco
- ▶ **Controlador del buffer**: controla las señales para el buffer de memoria

# Ejercicio

- ▶ ¿Cuántos bytes almacena un disco duro de 250 GB ?

# Recordatorio

- ▶ 1 KB = 1024 bytes, pero en el SI es 1000 bytes
- ▶ Los fabricantes de disco duros y en telecomunicaciones emplea el SI.
  - ▶ Un disco duro de 30 GB almacena  $30 \times 10^9$  bytes
  - ▶ Una red de 1 Mbit/s transfiere  $10^6$  bps.

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

# Ejercicio

- ▶ Un disco duro tiene una velocidad de rotación de 7200 rpm y una densidad constante de 604 sectores por pista. El tiempo medio de búsqueda es de 4 ms.
- ▶ Calcular el tiempo de acceso a un sector

# 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 (solución)

- ▶ 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
  - ▶  $600 / 100 = 6 \text{ milisegundos}$

# Ejercicio

- ▶ Sea un disco duro con tiempo medio de búsqueda de 4 ms, una velocidad de rotación de 15000 rpm y sectores de 512 bytes con 500 sectores por pista. Se quiere leer un fichero que consta de 2500 sectores con un total de 1,22 MB. Estimar el tiempo necesario para leer este fichero en dos escenarios:
  - ▶ El fichero está almacenado de forma secuencial, es decir, el fichero ocupa los sectores de 5 pistas adyacentes
  - ▶ Los sectores del fichero están distribuidos de forma aleatoria por el disco

# Fiabilidad

- ▶ Sea TMF el tiempo medio hasta el fallo
- ▶ Sea TMR el tiempo medio de reparación
- ▶ Se define la disponibilidad de un sistema como:

$$\text{disponibilidad} = \frac{TMF}{TMF + TMR}$$

- ▶ ¿Qué significa una disponibilidad del 99%?
  - ▶ En 365 días funciona correctamente  $99 \cdot 365 / 100 = 361.3$  días
  - ▶ Está sin servicio 3.65 días
- ▶ Los fallos en los discos duros contribuyen al 20-55% de los fallos en el sistema de almacenamiento

# Consumo de energía

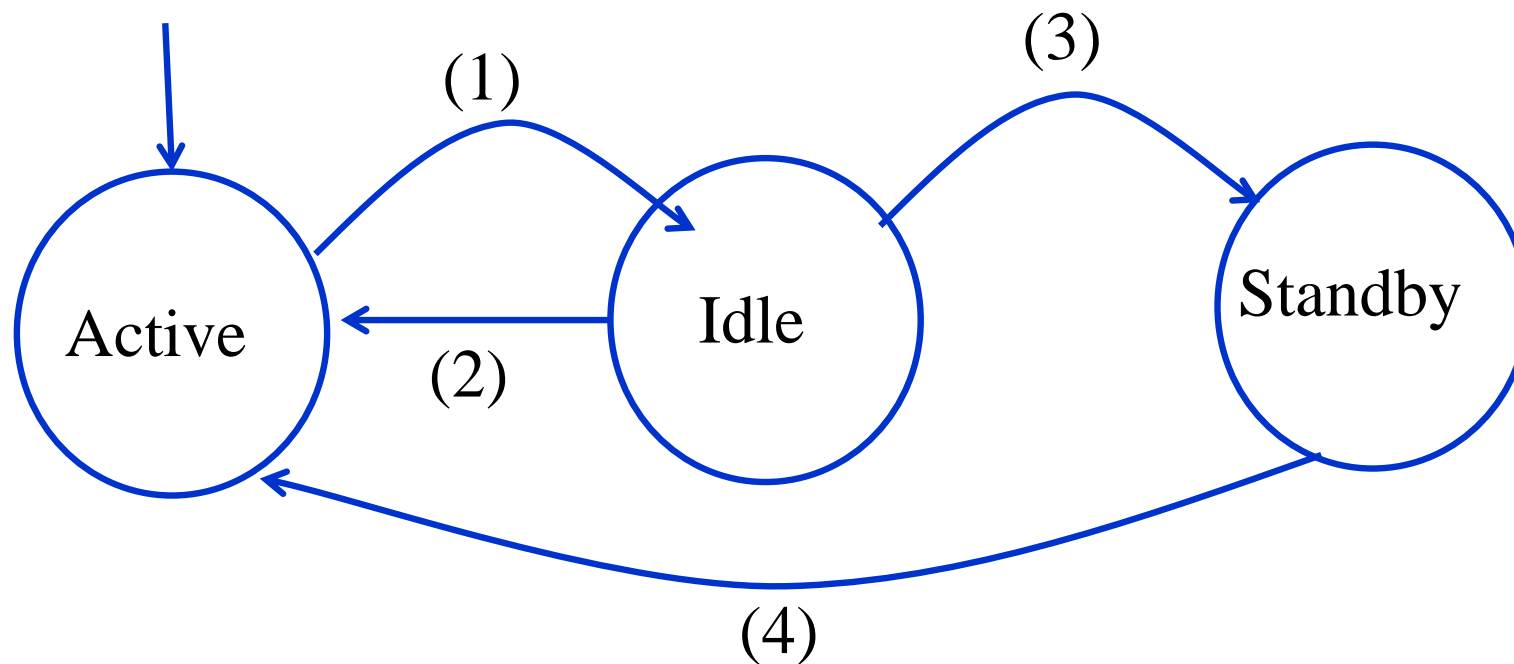
- ▶ Un típico disco ATA del 2011 consume:
  - ▶ 9 w cuando está inactivo
  - ▶ 11 w cuando está leyendo o escribiendo
  - ▶ 13 w en una operación de posicionamiento
- ▶ Estimación del consumo en un disco

$$Power = N_{platter} \times D_{platter}^{4.6} \times RPM^{2.8}$$

- ▶ Donde  $N_{platter}$  es el número de discos y  $D_{platter}$  el diámetro de los platos
- ▶ La temperatura es el factor que más afecta a la fiabilidad de los discos
  - ▶ Cada 10° de incremento por encima de los 21° reduce la fiabilidad de la electrónica en un 50%

# Diagrama de transición de estados de consumo de energía

R/W requests



# Transición de estados de consumo de energía

- ▶ (1): No hay peticiones pendientes pero los platos siguen girando, las cabezas se aparcan
- ▶ (2): El disco recibe una petición en el estado *idle*
- ▶ (3): Para reducir la energía el disco se para y las cabezas se sitúan fuera del disco
- ▶ (4): Nuevas peticiones, el disco tiene que activarse

# Estrategias de ahorro de energía

- ▶ Basada en *timeouts*. Cuando el disco está en estado *idle* durante un cierto tiempo pasa a *standby*
- ▶ Predicción dinámica. Basada en el funcionamiento de las aplicaciones
- ▶ Mecanismos estocásticos.
- ▶ Gestión de la energía guiado por la aplicación
  - ▶ La aplicación informa sobre el patrón de acceso ( en el código o generados por el compilador)

# Problemas del apagado de los discos

- ▶ Se incrementa el consumo cuando los platos tienen que volver a girar
- ▶ Reduce la fiabilidad de los discos. Los fabricantes suelen indicar el número de ciclos de start/stop que puede soportar un disco. Por encima de este valor la probabilidad de fallos se incrementa en un 50%
- ▶ Los métodos de ahorro de energía se suelen aplicar a dispositivos portátiles y no se aplican a servidores por las cargas de datos intensivas.



# Discos de estado sólido SSD (Solid State Drive)

- ▶ Dispositivo de almacenamiento de bloques basado en semiconductores que actúa como una unidad de disco
  - ▶ Basados en memorias Flash
    - ▶ Almacenamiento no volátil
  - ▶ Basados en memorias DDR
    - ▶ Requiere baterías y backup en disco para conseguir almacenamiento no volátil

HDD

con partes móviles



SDD

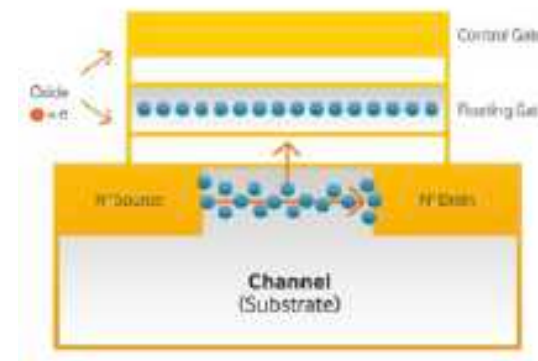
sin partes móviles

# Memorias Flash

- ▶ Memorias no volátiles que se pueden borrar y grabar eléctricamente
- ▶ Tipos:
  - ▶ Flash NOR
    - ▶ Basadas en puertas NOR
    - ▶ Permite acceder a datos de 1 byte
    - ▶ Buenas para accesos aleatorios de alta velocidad
    - ▶ Usada en memorias BIOS (función de arranque)
    - ▶ Operaciones de lectura más rápidas
  - ▶ Flash NAND
    - ▶ Basadas en puertas NAND
    - ▶ Lee y escribe a alta velocidad en modo secuencial en tamaños de bloques
    - ▶ No puede acceder a bytes individuales
    - ▶ Mayor densidad y más baratas. Usada en SSD
    - ▶ Más duraderas, menos caras, más densas, operaciones de escritura/borrado más rápidas

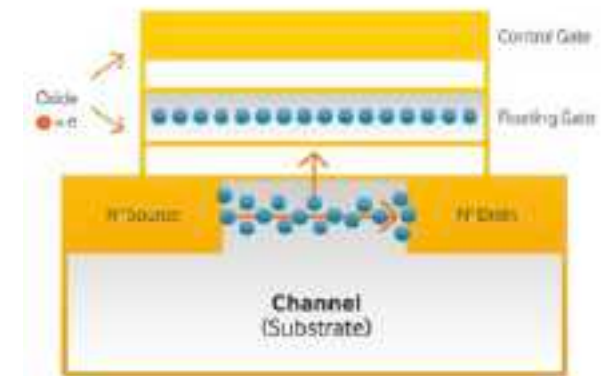
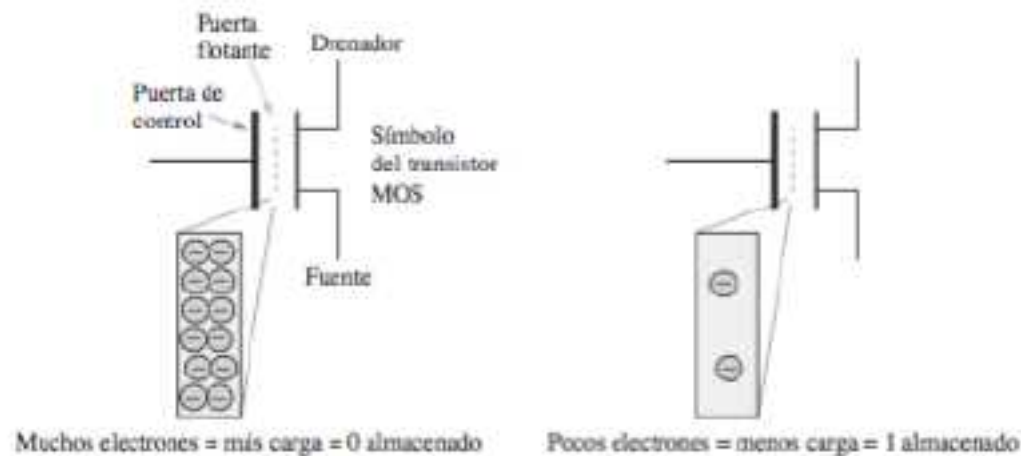
# Celdas de memoria

- ▶ Cada celda de almacenamiento está compuesta por un transistor MOS de puerta flotante
- ▶ Hay dos puertas aisladas por una capa de óxido
  - ▶ Puerta de control
  - ▶ Puerta flotante
- ▶ Los electrones fluyen libremente entre ambas puertas
- ▶ La puerta flotante está eléctricamente aislada atrapando los electrones



# Celdas de memoria

- El bit de datos se almacena como una carga o ausencia de carga en la puerta flotante, dependiendo de si se desea almacenar un 0 o un 1

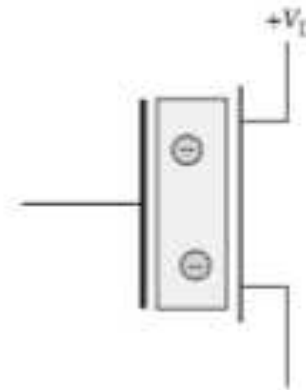
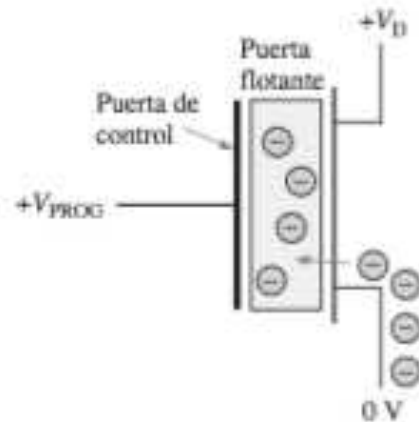


Fundamentos de Sistemas Digitales  
Thomas L. Floyd

# Operaciones básicas de una celda de memoria Flash

- ▶ Programación
  - ▶ Inicialmente todas las celdas están en el estado 1, porque la carga está eliminada
- ▶ Operación de lectura
- ▶ Operación de borrado

# Programación de una celda de memoria Flash

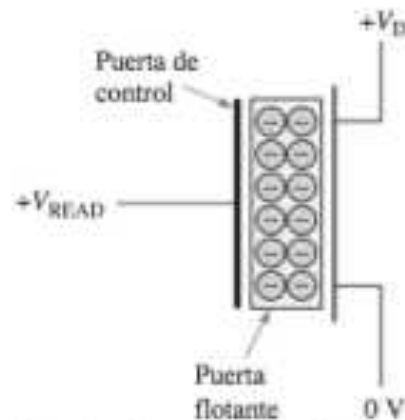


Fundamentos de Sistemas Digitales  
Thomas L. Floyd

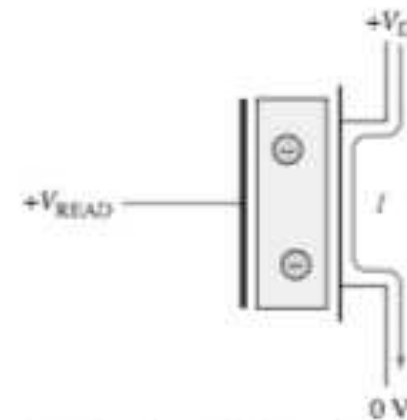
- ▶ Para almacenar un 0, se aplica a la puerta de control una tensión suficientemente positiva con respecto a la fuente, para añadir carga a la puerta flotante durante la programación (atrae los electrones)
- ▶ En el proceso de escritura se añaden electrones a aquellas puertas que deban almacenar un 0 y no se añade a las que deben almacenar un 1.
- ▶ Solo se escriben "0"
- ▶ Para almacenar un 1, no se añade ninguna carga, dejándose la celda en el estado de borrado

# Lectura de una celda de memoria Flash

- Se aplica una tensión positiva a la puerta de control



Cuando se lee un 0, el transistor permanece desactivado, porque la carga de la puerta flotante impide a la tensión de lectura exceder el umbral de activación.

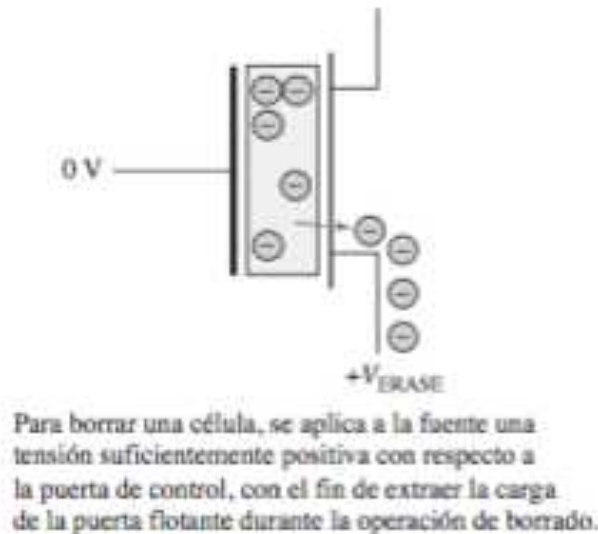


Cuando se lee un 1, el transistor se activa, porque la ausencia de carga en la puerta flotante permite que la tensión de lectura exceda el umbral de activación.

Fundamentos de Sistemas Digitales  
Thomas L. Floyd

# Borrado

- ▶ Durante el borrado se elimina la carga de todas las celdas de memoria. Se aplica un voltaje en sentido contrario para eliminar los electrones



Fundamentos de Sistemas Digitales  
Thomas L. Floyd



# Tipos de memorias Flash NAND

- ▶ Flash de celdas de un nivel (SLC)
  - ▶ Almacenan 1 bit por celda
  - ▶ 50000 - 100000 escrituras por celda
  - ▶ Usado fundamentalmente en aplicaciones militares e industriales
- ▶ Flash de celdas de múltiples niveles (MLC)
  - ▶ Almacenan varios bits por celda, en función del número de electrones almacenados en la celda
  - ▶ Ofrecen más capacidad pero menos duración (se desgastan más)
  - ▶ < 10000 escrituras por celda
  - ▶ Usado en electrónica de consumo
  - ▶ Menor coste
  - ▶ Mitad de prestaciones que las SLC

# Nivelación del desgaste

- ▶ Una memoria flash NAND solo puede escribir un determinado número de veces en cada bloque (o celda)
- ▶ Cuando se supera el límite, la celda se desgasta (su capa de óxido) y ya no almacena correctamente los electrones
- ▶ Nivelación del desgaste: proceso que usa el controlador de una unidad SSD para maximizar la duración de la memoria Flash
- ▶ Esta técnica nivela el desgaste en todos los bloques distribuyendo la escritura de datos por todos los bloques
  - ▶ Cuando se va a modificar un bloque se escribe en uno nuevo

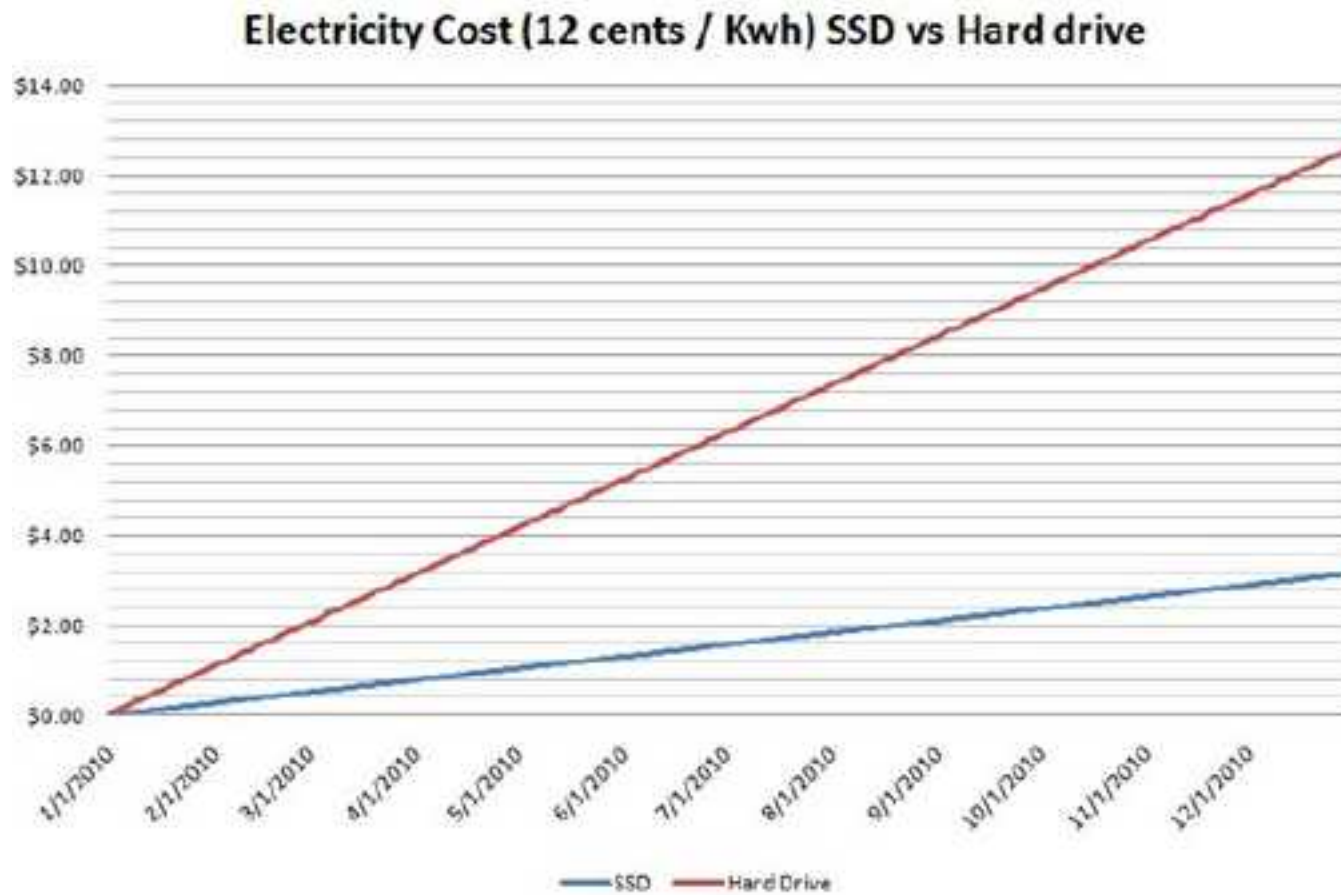
# Estructura de una Flash Nand

- ▶ Las flash NAND se dividen en bloques de 128 KB que se subdividen en páginas de 2KB
- ▶ El borrado se hace de un bloque completo
- ▶ La programación y lectura de una página

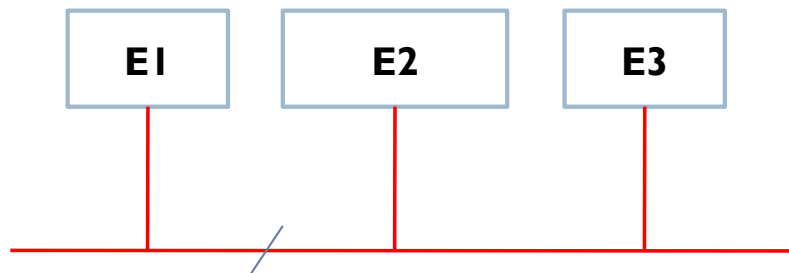
# Comparativa SDD vs HDD

	<b>SDD</b>	<b>HDD</b>
Tiempo de acceso	0.1 ms	5-8 ms
Operaciones de E/S por segundo	6000 io/s	400 io/s
consumo	2-5 watios	6-15 watios

# Consumo energético

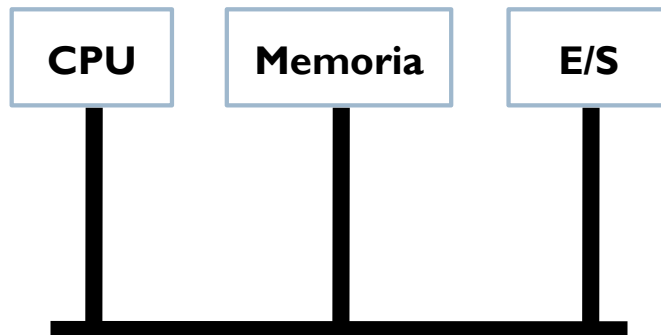


# Bus



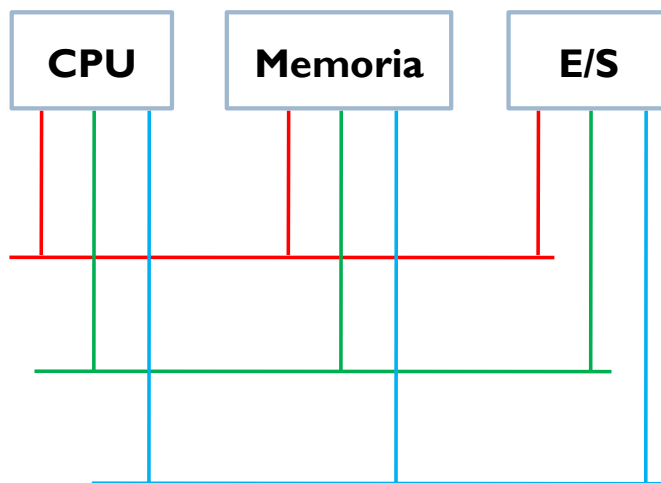
- ▶ Un bus es un **camino** de comunicación **entre dos o más dispositivos**.
- ▶ **Constituido** por **varias líneas** de transmisión de bit.
- ▶ **Medio compartido**, unívoco.
- ▶ Permite transmitir varios bits entre dos elementos conectados a él

# 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

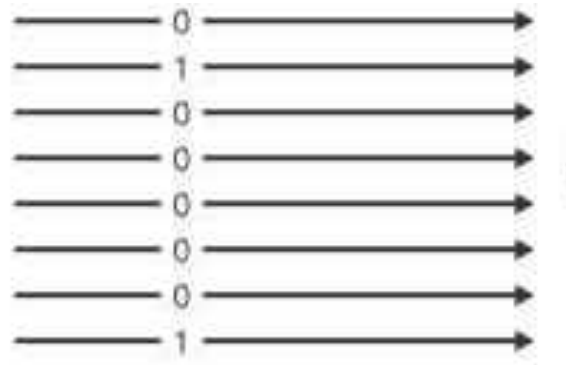


# Características de un bus

- ▶ Ancho el bus
- ▶ Frecuencia
- ▶ Velocidad de transferencia
- ▶ Ancho de banda

# Características de un bus

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



bus paralelo



bus serie

# Características de un bus

- ▶ **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  $\times$  frecuencia

# Ejercicio

- ▶ Calcular el ancho de banda 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

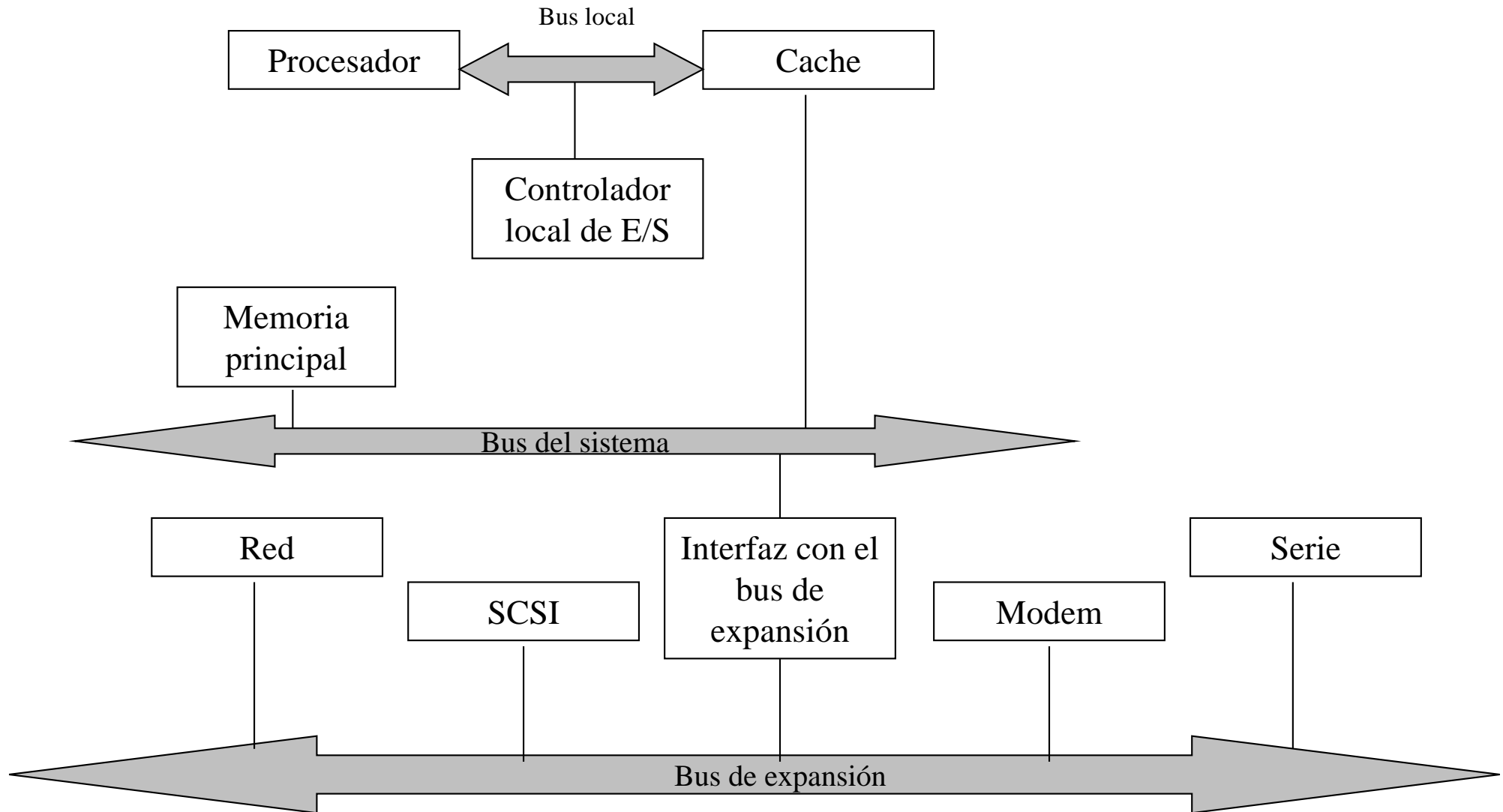
- ▶ Un bus **síncrono** está gobernado por una señal de reloj y un protocolo de comunicación ajustado al funcionamiento del reloj
  - ▶ Rápido
  - ▶ Todos los dispositivos conectados a él deben operar a la misma frecuencia de reloj
- ▶ Un bus **asíncrono** no utiliza un reloj, la comunicación se realiza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

# Jerarquías de buses

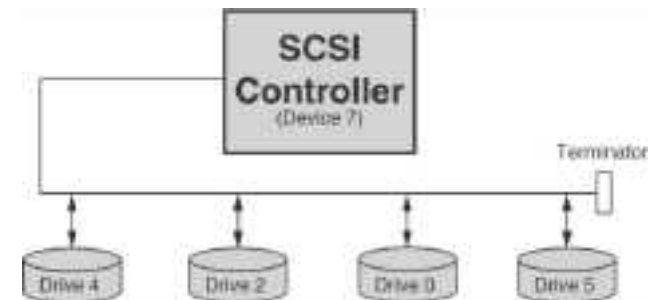
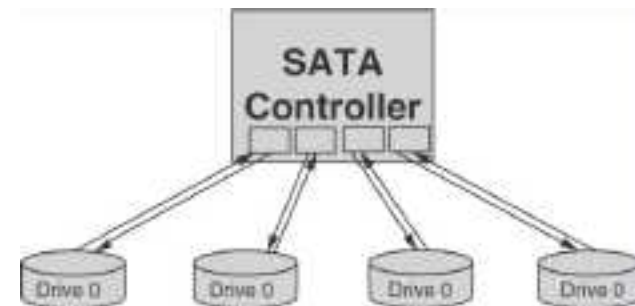
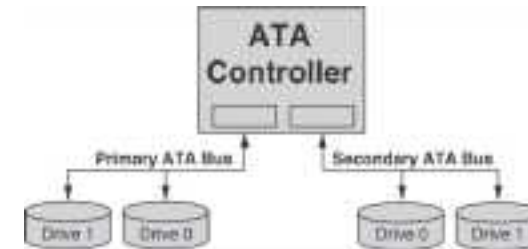
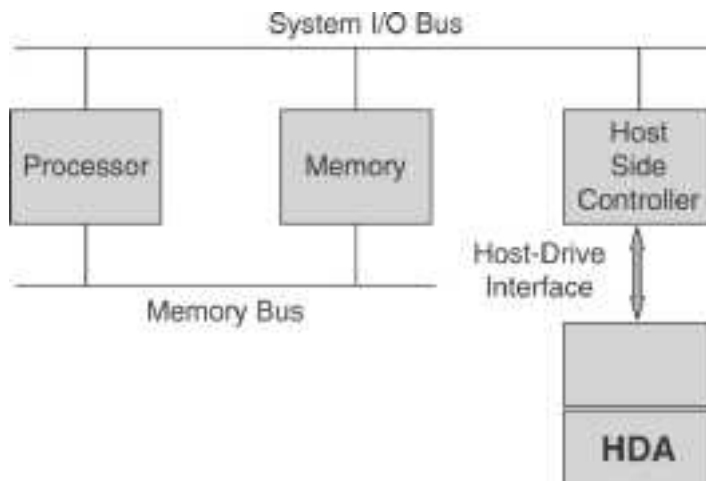
- ▶ A más dispositivos conectados al bus, mayor es el retardo de propagación.
- ▶ A medida que aumenta el número de peticiones de transferencia, se puede producir un cuello de botella.
- ▶ Soluciones:
  - ▶ Aumentar la velocidad de transmisión de datos con buses más anchos.
  - ▶ Utilizar más buses de datos, organizados jerárquicamente.



# Jerarquías de buses

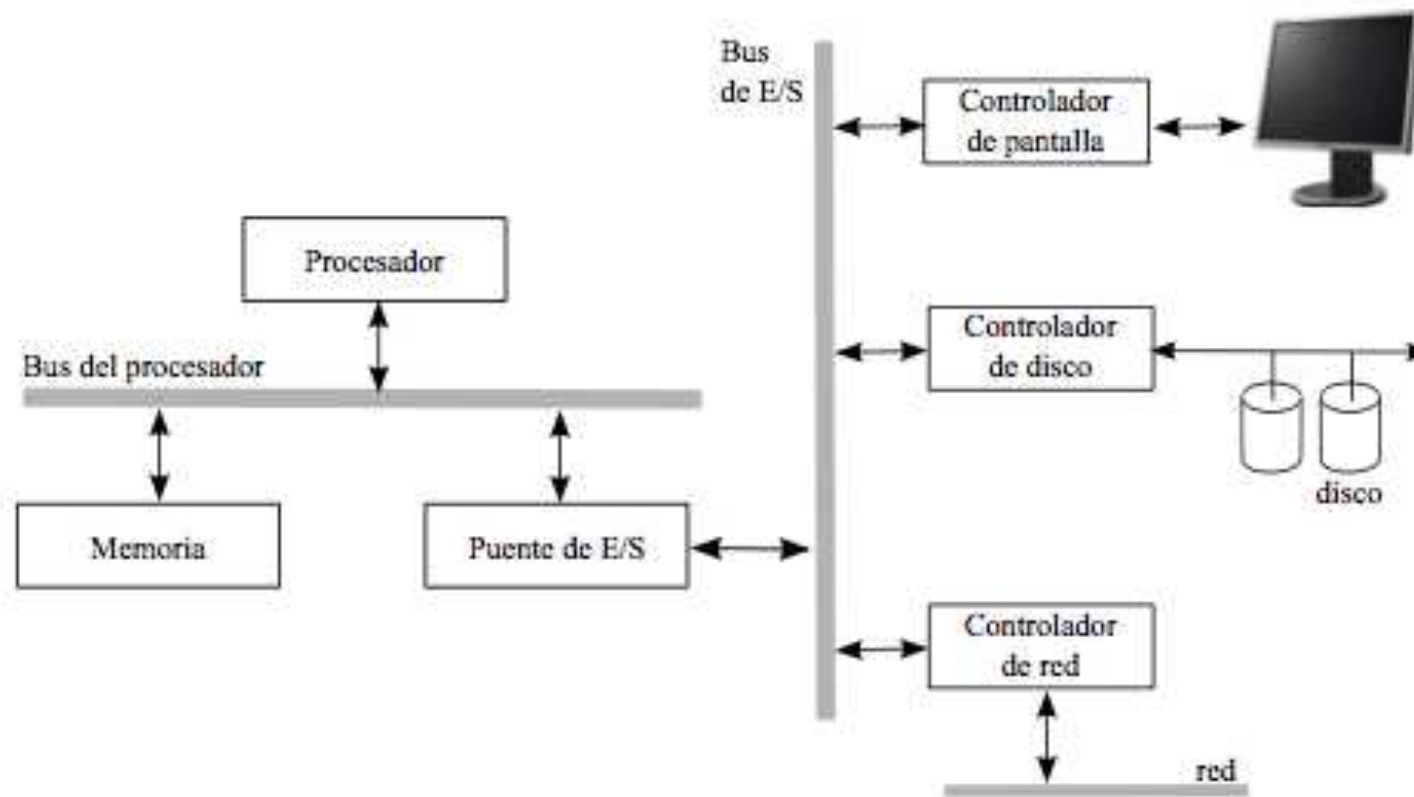


# Controladores de disco



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

# Esquema de buses en un sistema informático típico



# Curiosidades: Familia USB



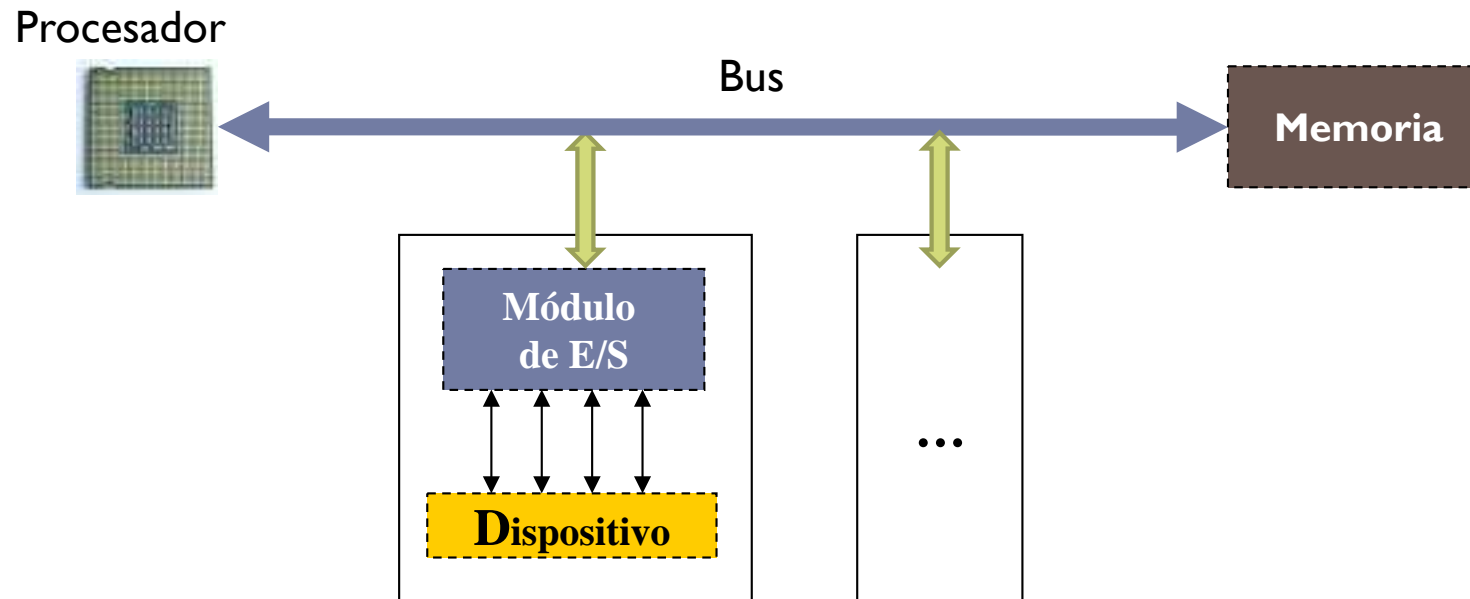
	Transfer. (por seg.)	Aparición
<b>USB 3.0</b>	<b>600 MB/s</b>	<b>2010</b>
<b>USB 2.0</b>	<b>60 MB/s</b>	<b>2000</b>
<b>USB 1.0</b>	<b>1.5 MB/s y 187 KB/s</b>	<b>1996</b>

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

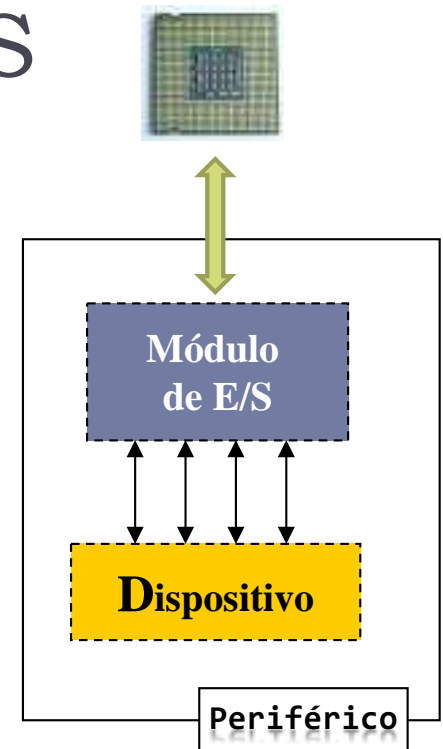
<http://www.unp.co.in/f140/comparison-of-usb-3-0-port-with-usb-2-0-and-usb-1-0-a-70063/>

# 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

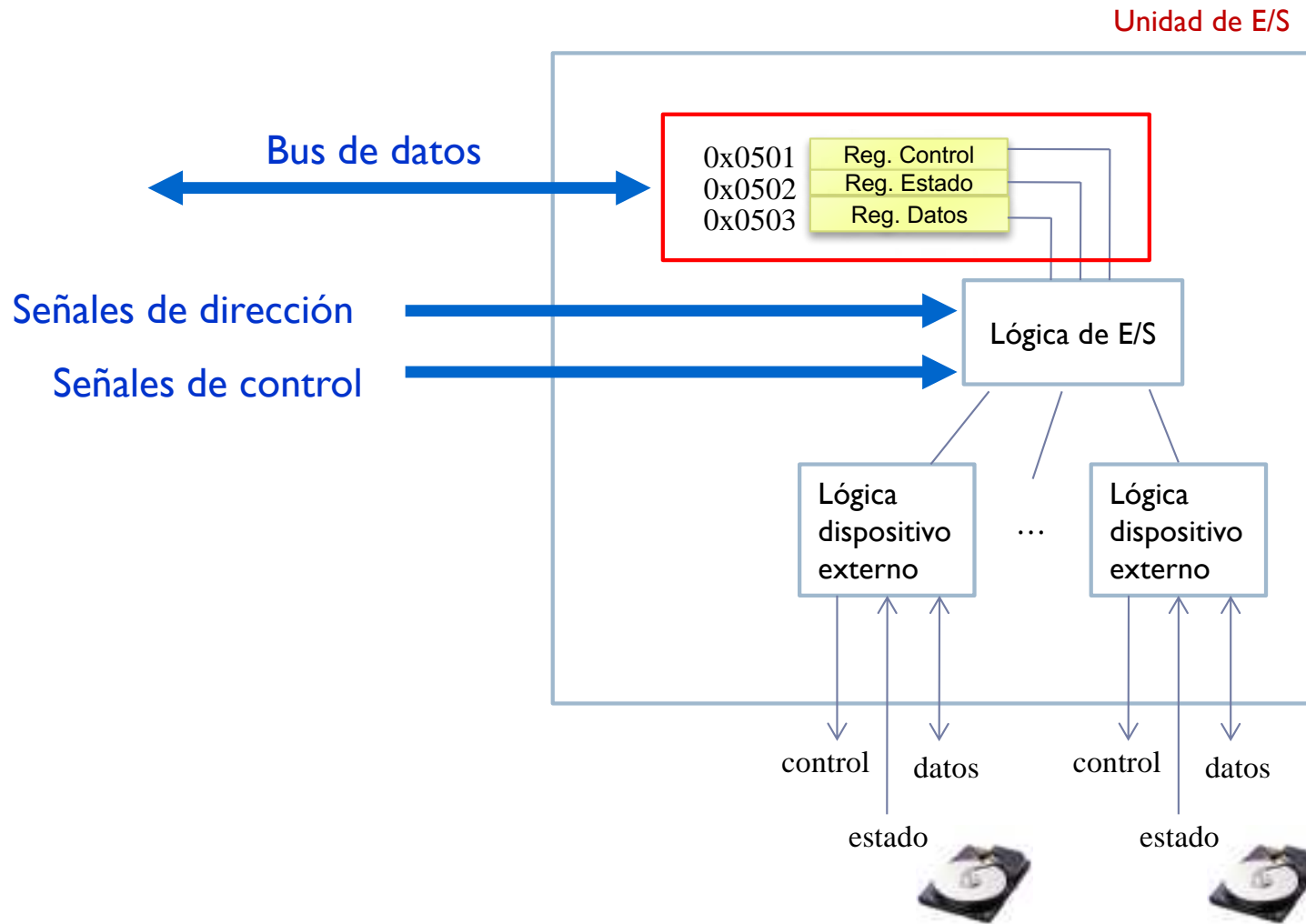


- ▶ Son necesarios debido a:
  - ▶ Gran variedad de periféricos.
    - ▶ Los periféricos son 'raros'
  - ▶ La velocidad de transferencia de datos de los periféricos es mucho menor que la de la memoria o el procesador.
    - ▶ Los periféricos son 'muy lentos'
  - ▶ Formatos y tamaños de palabra de los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

# Necesidades de los módulos de E/S

- ▶ Control y temporización
- ▶ Comunicación con el procesador o la memoria
- ▶ Comunicación con el periférico
- ▶ *Buffering* o almacenamiento intermedio
- ▶ Detección de errores

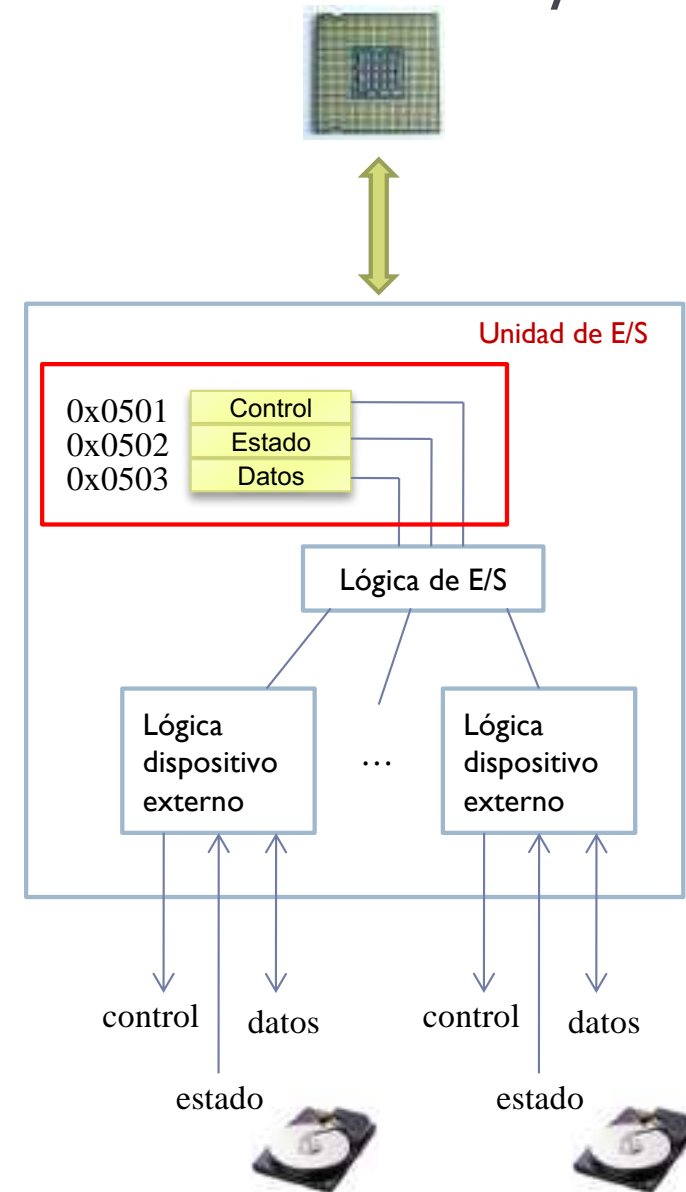
# Modelo simplificado de módulo de E/S





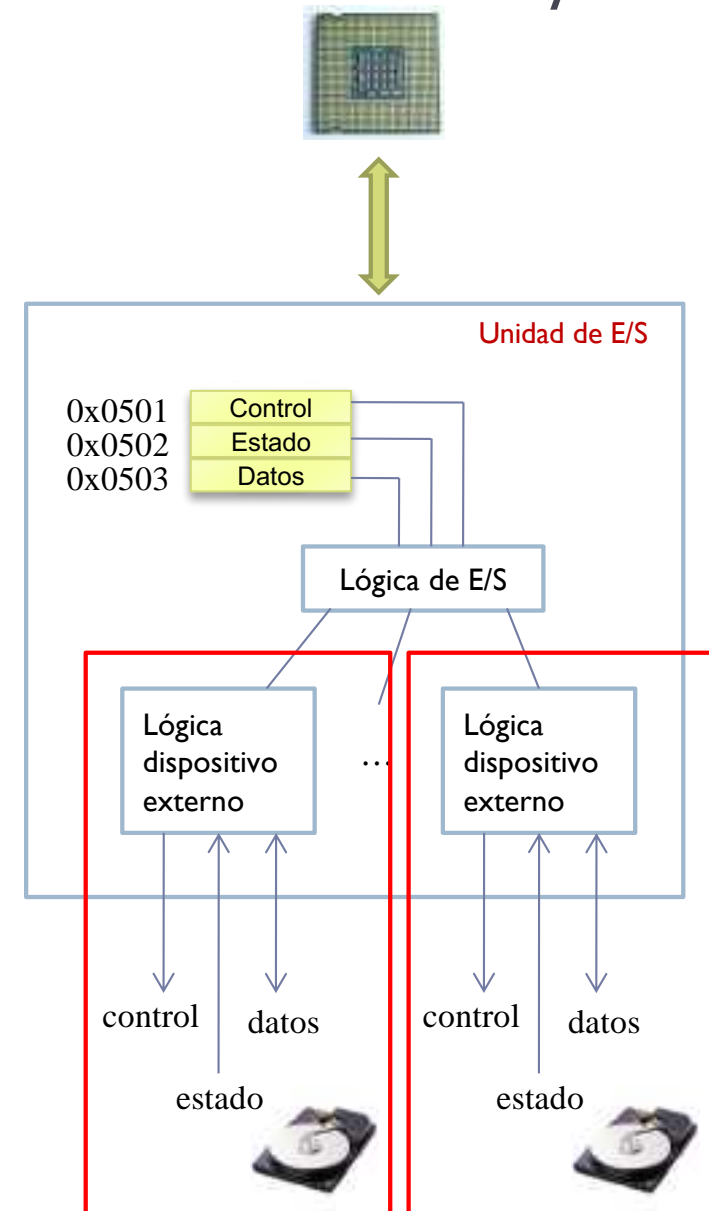
# Modelo simplificado de módulo de E/S

- ▶ 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.



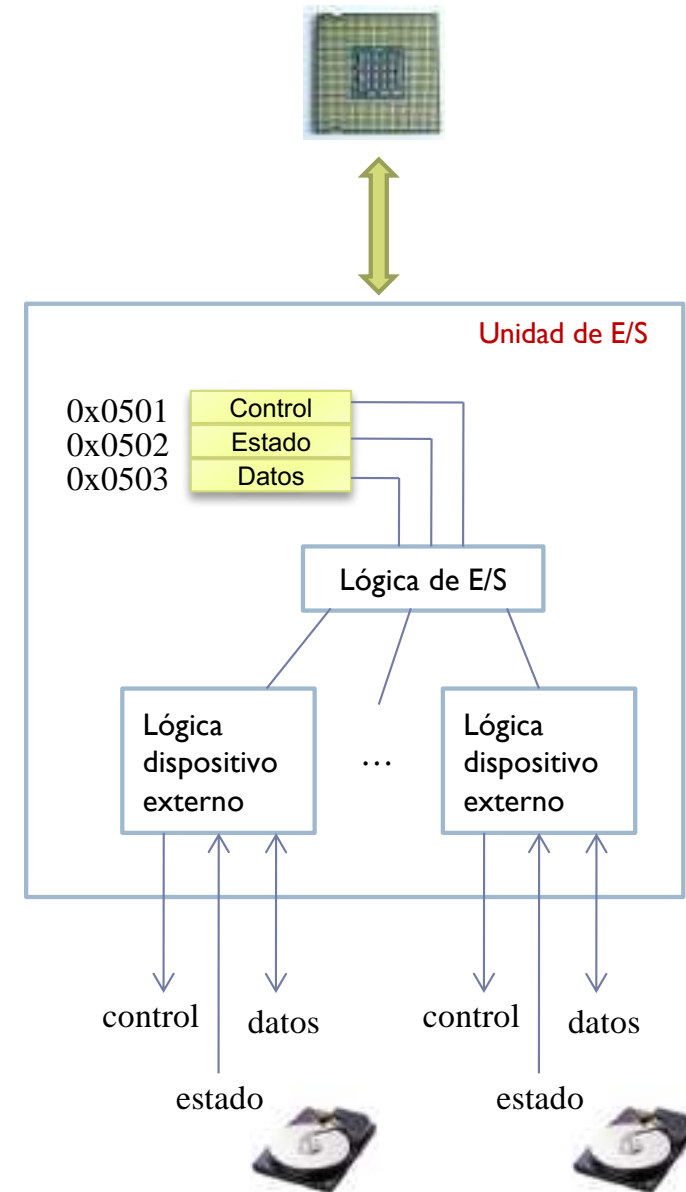
# Modelo simplificado de módulo de E/S

- ▶ 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: 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->procesador



# Módulo de E/S: tipos

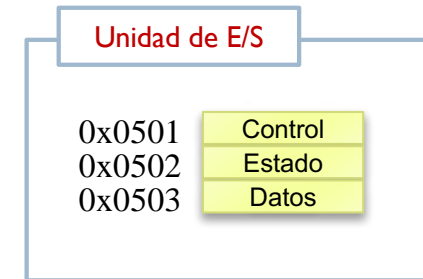
- ▶ Tipos de módulos por complejidad:

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

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

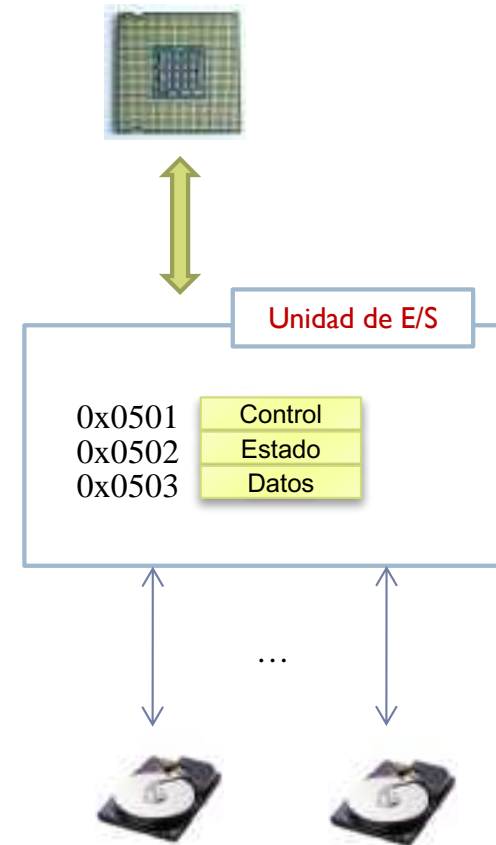
- ▶ **Controlador de E/S o controlador de dispositivo:**

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



# Módulo de E/S: características

- ▶ Características fundamentales:
  - ▶ **Unidad de transferencia**
  - ▶ **Direccionamiento**
  - ▶ **Técnicas de Entrada/Salida**

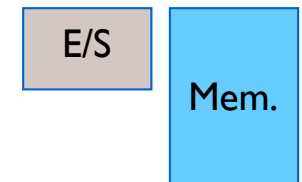
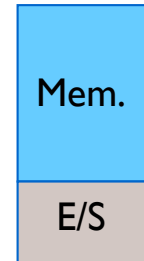


# Características (1 / 3)

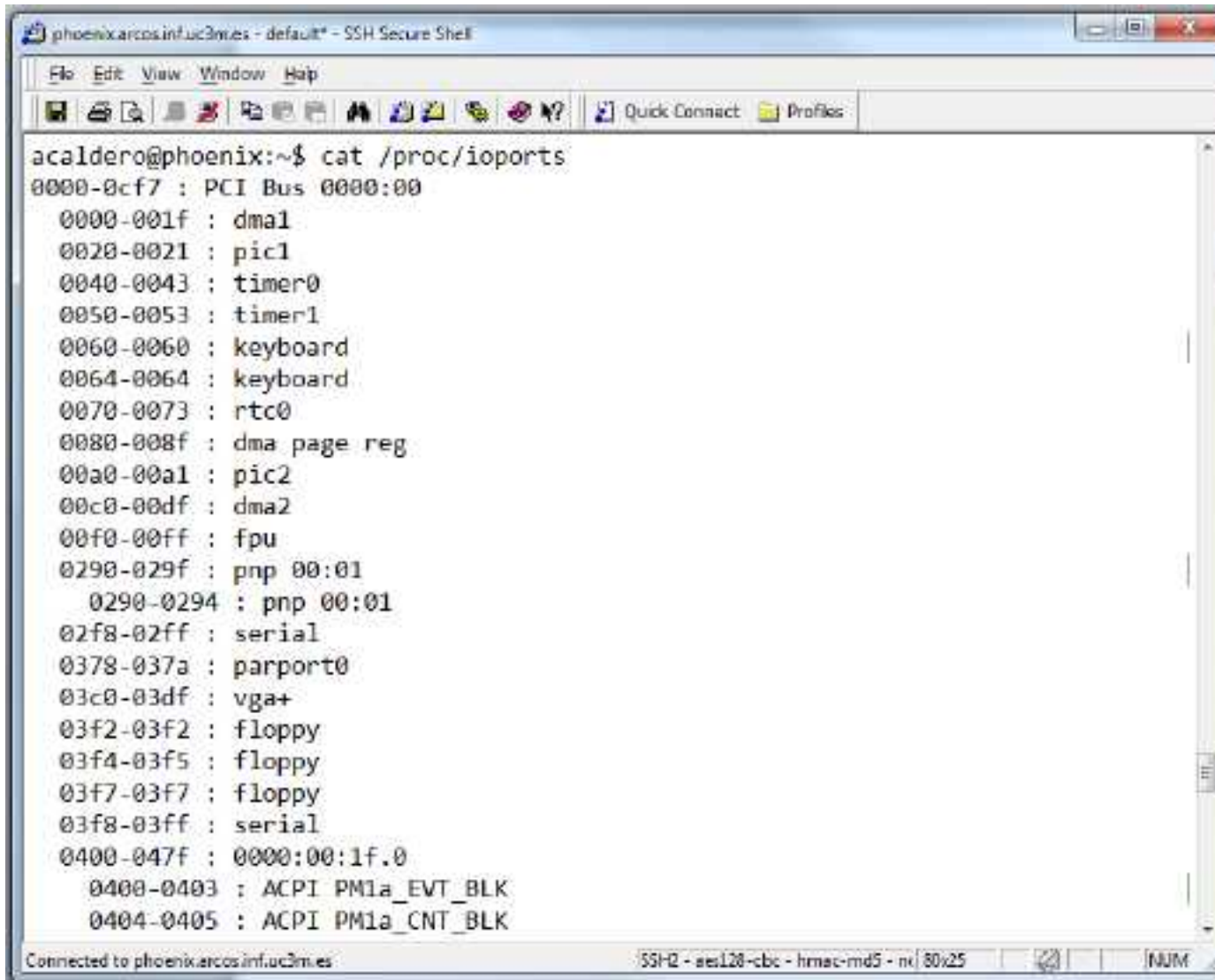
- ▶ Unidad de transferencia:
  - ▶ **Dispositivos de bloque:**
    - ▶ Unidad: **bloque** de bytes
    - ▶ Acceso **secuencial** o **directo** a bloques
    - ▶ Operaciones: leer, escribir, situarse, ...
    - ▶ Ejemplos: discos y “cintas”
  - ▶ **Dispositivos de carácter:**
    - ▶ Unidad: **caracteres** (ASCII, Unicode, etc)
    - ▶ Acceso **secuencial** a caracteres
    - ▶ Operaciones: get, put, ....
    - ▶ Ejemplo: terminales, impresoras, tarjetas de red

# Características (2/3)

- ▶ Direccionamiento de E/S:
  - ▶ **Espacio de memoria conjunto**
    - ▶ 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`



# Direcccionamiento Linux

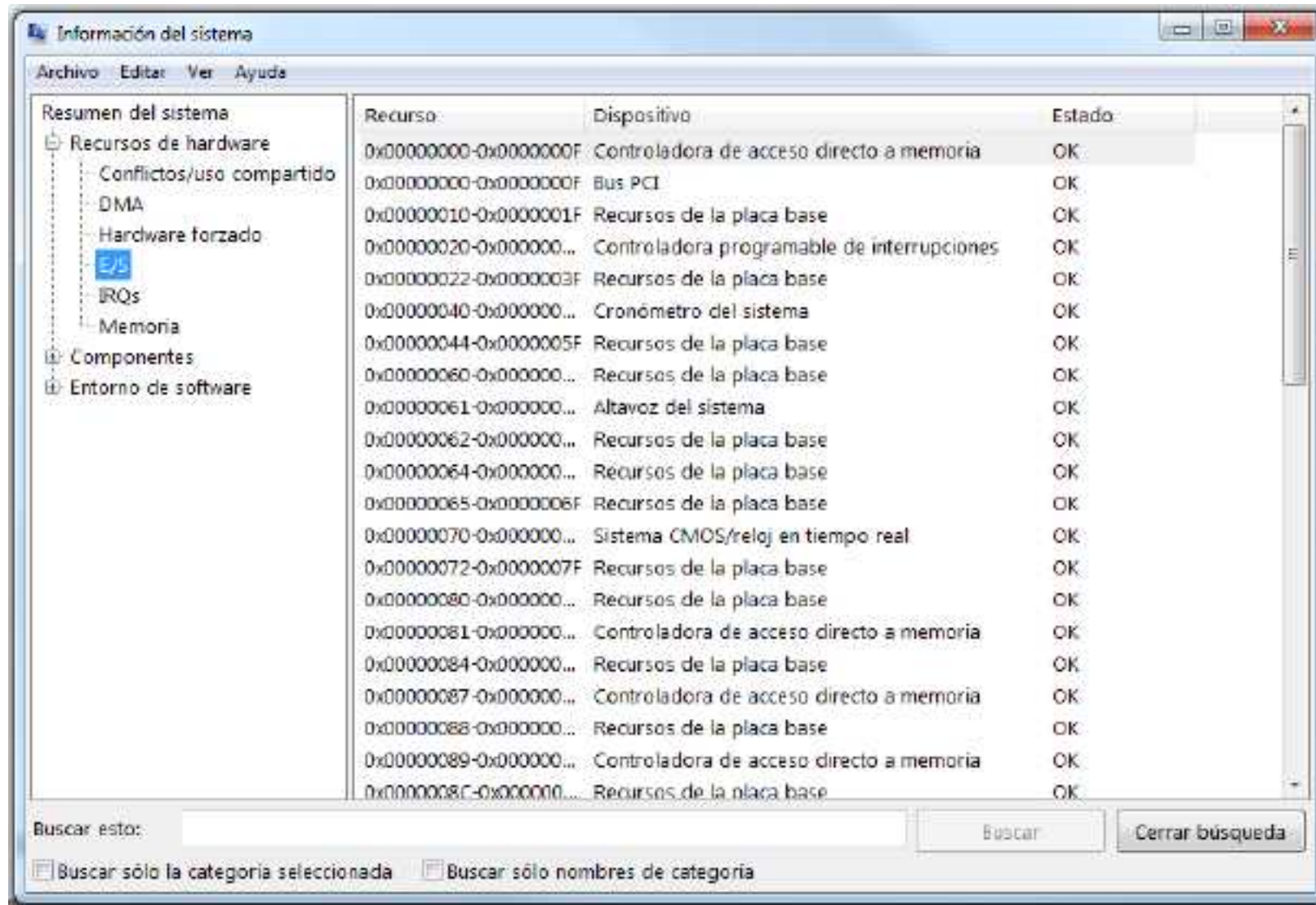


The image shows a terminal window titled "phoenixarcos.inf.uc3m.es - default\* - SSH Secure Shell". The window has a menu bar (File, Edit, View, Window, Help) and a toolbar with various icons. The terminal content shows the command "cat /proc/ioprocs" being executed by the user "acaldero" at the "phoenix" host. The output lists various I/O ports and their corresponding hardware devices, such as PCI Bus, dma1, pic1, timer0, timer1, keyboard, rtc0, dma page reg, pic2, dma2, fpu, pnp 00:01, serial, parport0, vga+, floppy, and ACPI PM1a\_EVT\_BLK. The status bar at the bottom indicates the connection details: "Connected to phoenixarcos.inf.uc3m.es" and "SSH2 - aes128-cbc - hmac-md5 - nx | 80x25".

```
acaldero@phoenix:~$ cat /proc/ioprocs
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
```



# Direcccionamiento Windows



# Características (3/3)

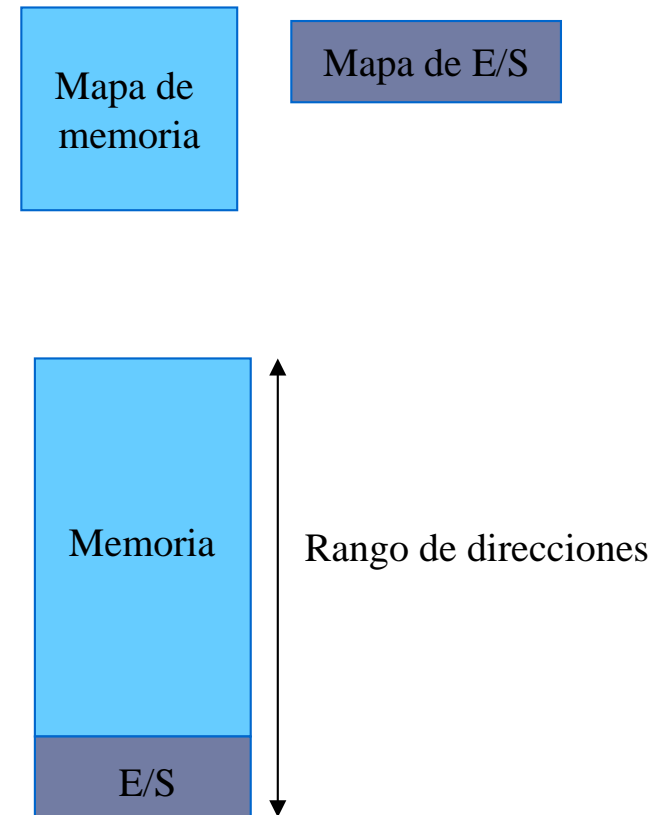
- ▶ **Técnicas de E/S:** Interacción Procesador-Controlador
  - ▶ **E/S programada**
  - ▶ **E/S por interrupciones**
  - ▶ **E/S por DMA (acceso directo a memoria)**

# E/S programada

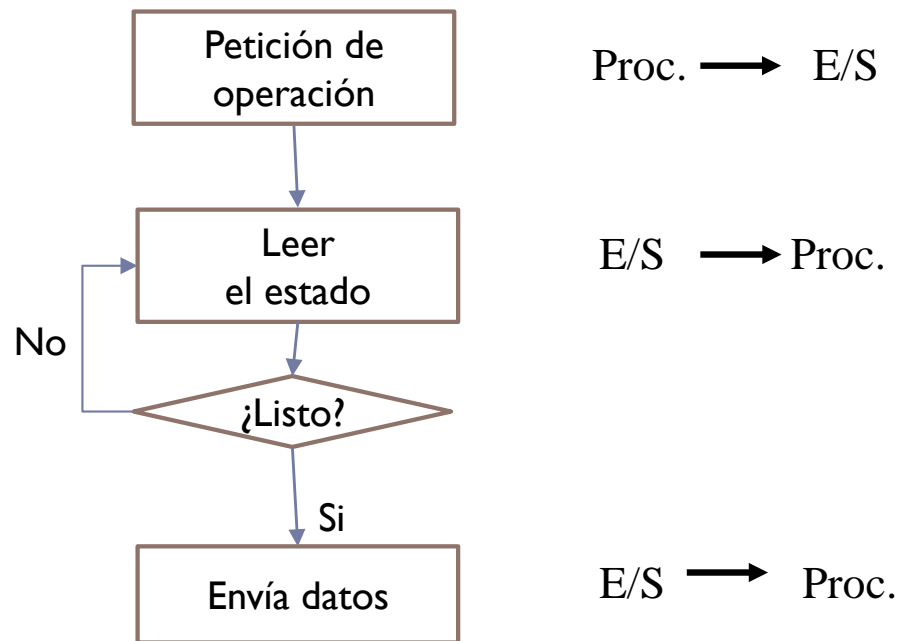
- ▶ Toda la transferencia entre el procesador (o memoria) y la unidad de E/S se realiza bajo el control del procesador mediante la ejecución de **instrucciones máquina de E/S**
- ▶ Instrucciones de E/S:
  - ▶ Instrucciones máquina especiales (similares a LW y SW)
  - ▶ Son instrucciones privilegiadas
- ▶ Ejemplo de hipotética instrucción de E/S
  - ▶ **IN Reg, dirección**
    - ▶ Carga en el registro del procesador Reg el dato que se encuentra en el registro del módulo de E/S cuya dirección es dirección
  - ▶ **OUT Reg, dirección**
    - ▶ Realiza la escritura en el módulo de E/S

# Mapa de E/S

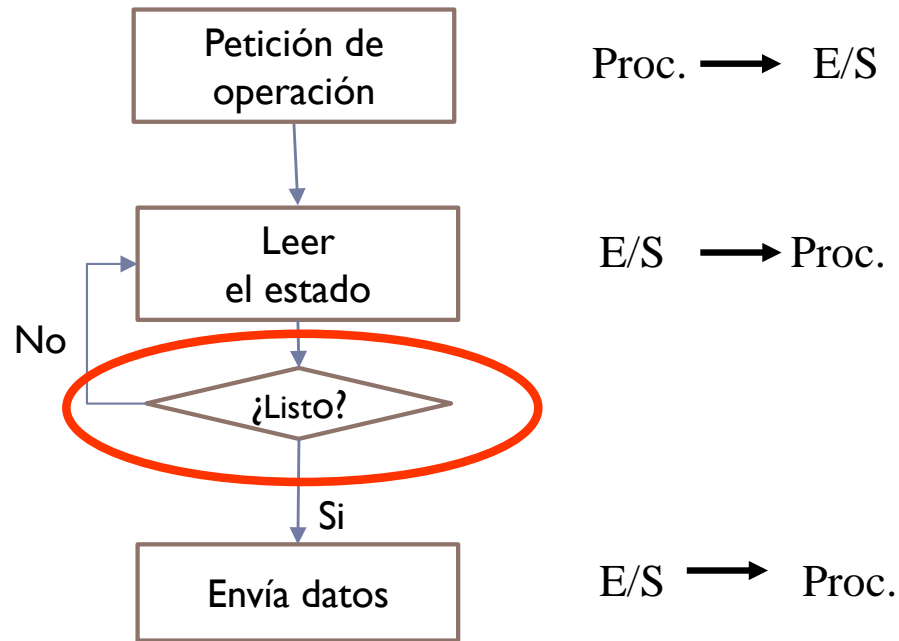
- ▶ Conjunto de todas las direcciones de E/S
  - ▶ Con  $p$  bits,  $2^p$  direcciones posibles
- ▶ Tipos:
  - ▶ Mapa de E/S separado
    - ▶ Incluye instrucciones especiales de E/S (IN, OUT)
  - ▶ Mapa común
    - ▶ El acceso a los módulos de E/S se realiza con las mismas instrucciones que se utilizan para acceder a memoria (LOAD, STORE)



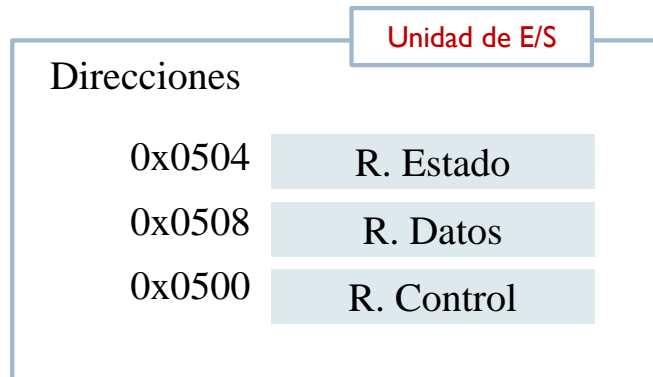
# Interacción mediante E/S programada



# Interacción mediante E/S programada

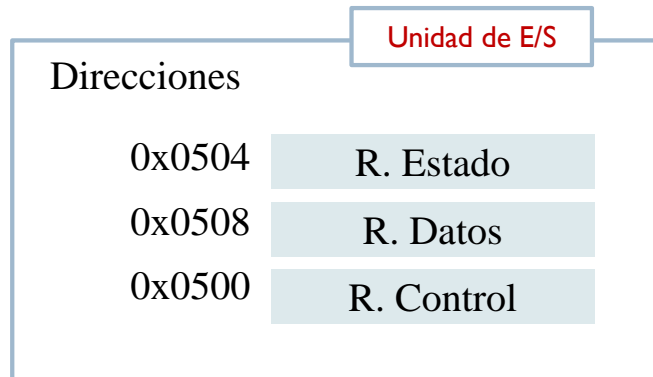


# Ejemplo



- ▶ 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 del MIPS

# Ejemplo

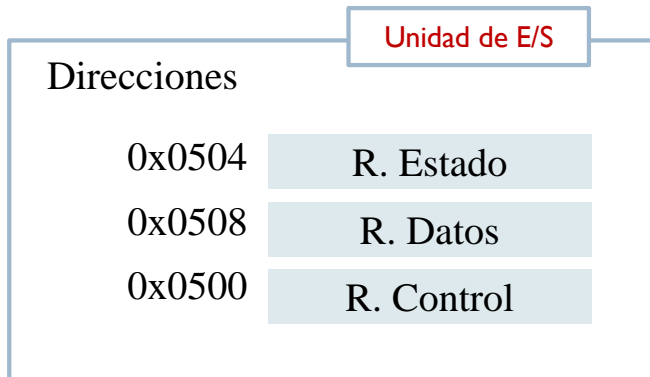


- ▶ ¿Instrucciones para escribir un I en el registro 0x0508 (de datos)?

- ▶ 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 del MIPS



# Ejemplo

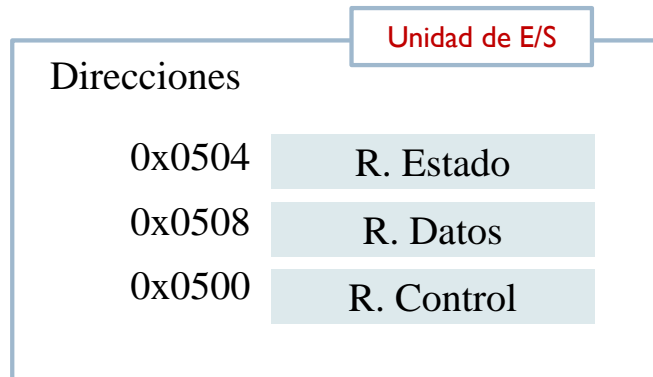


- ▶ Información de control
  - ▶ 0: leer
  - ▶ 1: escribir
- ▶ Información de estado
  - ▶ 0: dispositivo ocupado
  - ▶ 1: dispositivo (dato) listo
- ▶ Mapa de E/S común
  - ▶ Instrucciones lw y sw del MIPS

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

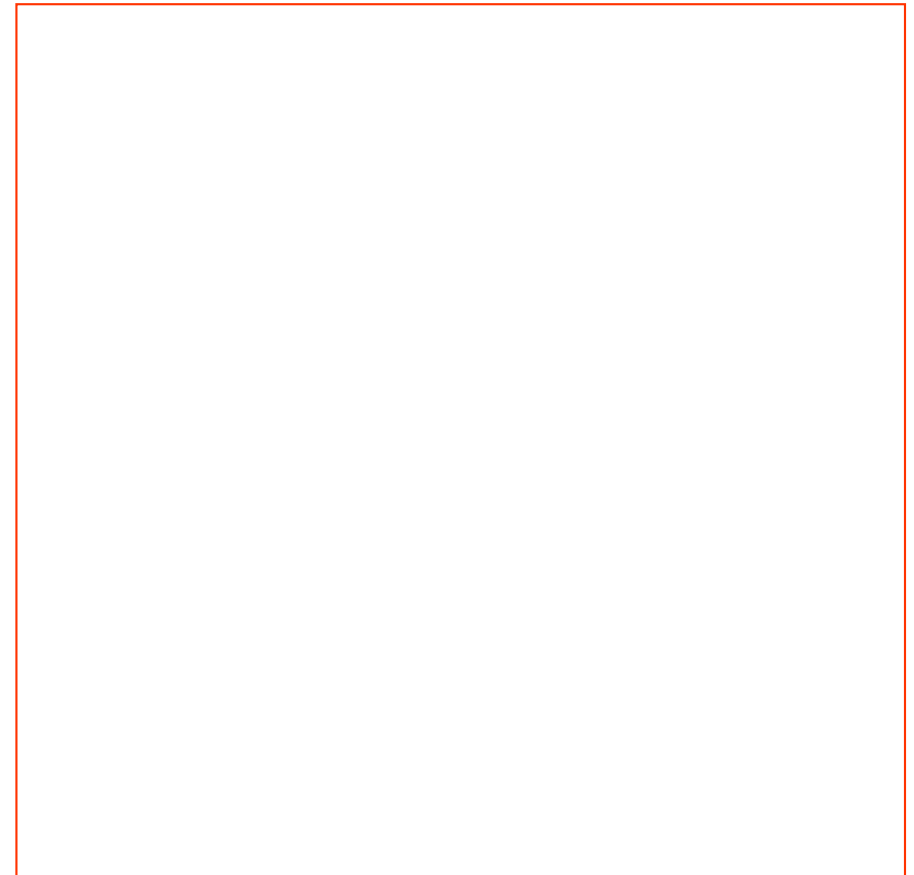
```
li $t0, 1
sw $t0, 0x0508
```

# Ejemplo

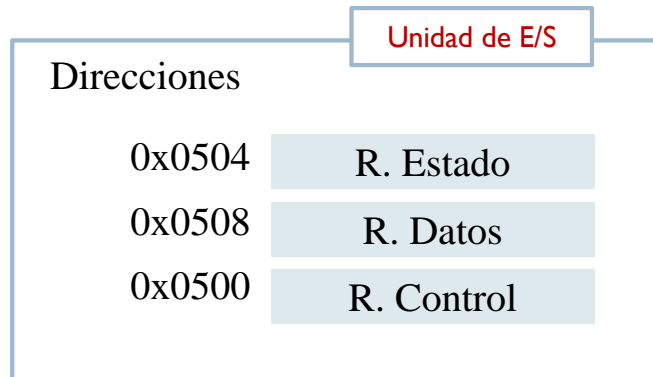


- ▶ Información de control
  - ▶ 0: leer
  - ▶ 1: escribir
- ▶ Información de estado
  - ▶ 0: dispositivo ocupado
  - ▶ 1: dispositivo (dato) listo
- ▶ Mapa de E/S común
  - ▶ Instrucciones lw y sw del MIPS

- ▶ ¿Operaciones para leer un dato?



# 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 del MIPS

## ▶ ¿Operaciones para leer un dato?

1. Enviar la orden

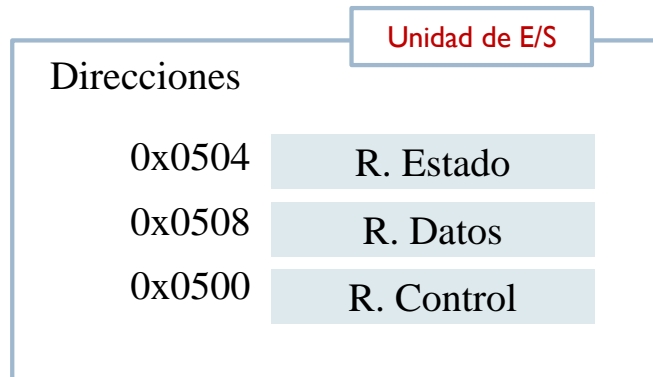
```
li $t0, 0
sw $t0, 0x0500
```
2. Leer el estado

```
bucle: lw $t0, 0x0504
```
3. Comprobar el estado

```
beqz $t0, bucle
```
4. Leer el dato

```
lw $t0, 0x0508
```

# 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 del MIPS

## ▶ ¿Operaciones para escribir un dato?

### 1. Enviar el dato

```
li $t0, 123
sw $t0, 0x0508
```

### 2. Enviar la orden

```
li $t0, 1
sw $t0, 0x0500
```

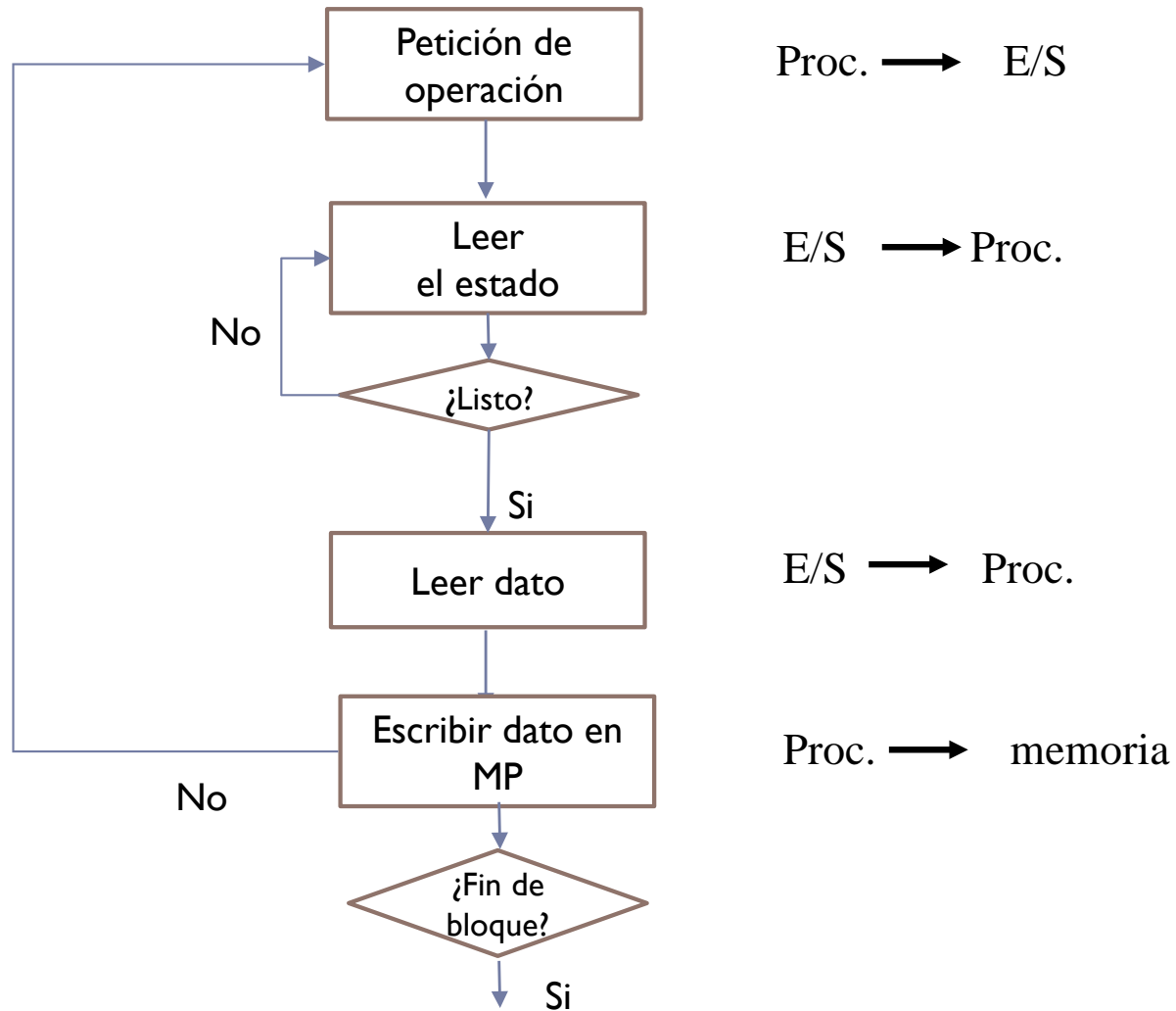
### 3. leer estado

```
bucle: lw $t0, 0x0504
```

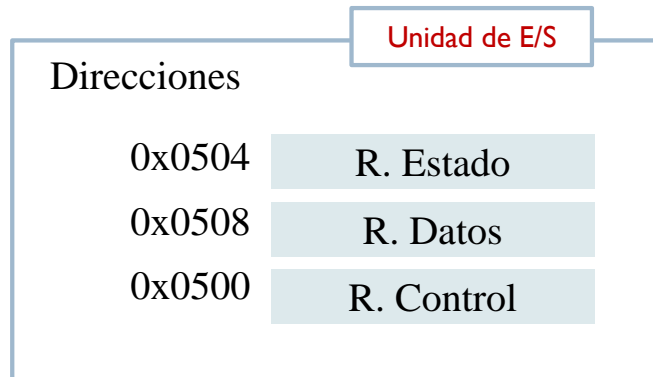
### 4. comprobar estado

```
beqz $t0, bucle
```

# Lectura de un bloque de datos



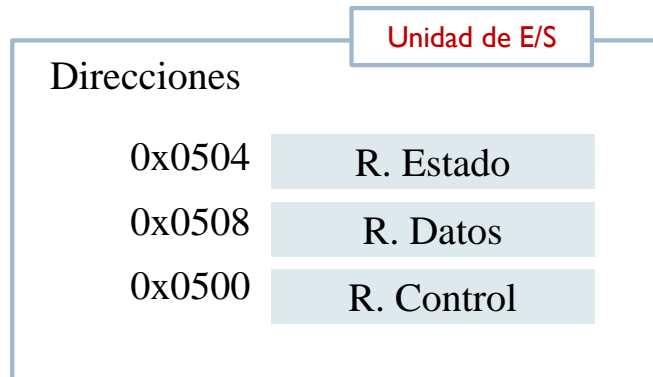
# Ejercicio



- ▶ 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)



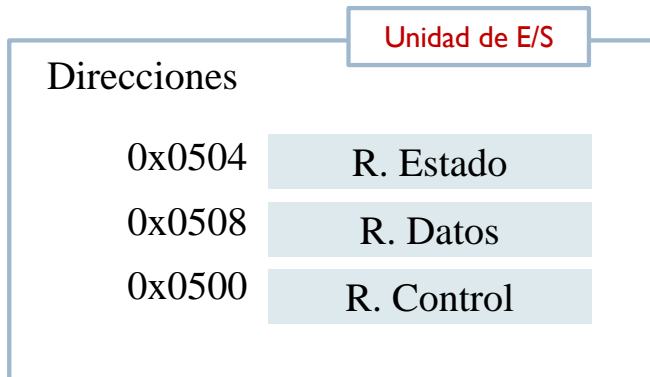
- ▶ Información de control
  - ▶ 0: leer
  - ▶ 1: escribir
- ▶ Información de estado
  - ▶ 0: dispositivo ocupado
  - ▶ 1: dispositivo (dato) listo
- ▶ Mapa de E/S común
  - ▶ Instrucciones lw y sw

```
.data
    datos: .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 (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: .space 400

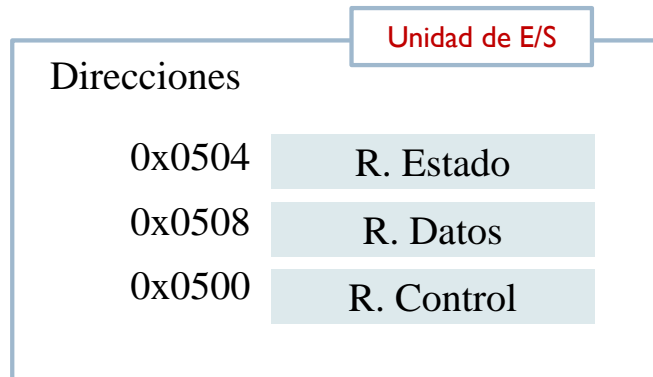
.text
.globl main
main:    li $t3 0
        bucle1: li $t0 0
                sw $t0 0x500

        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 (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: .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
```



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:
  - ▶  $I (li \$t3 0) + 6 * 100 + 10^6 (I_{bs})$
- ▶ 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 del procesador
  - ▶ El procesador no realiza trabajo útil

```
.data
    datos: .space 400

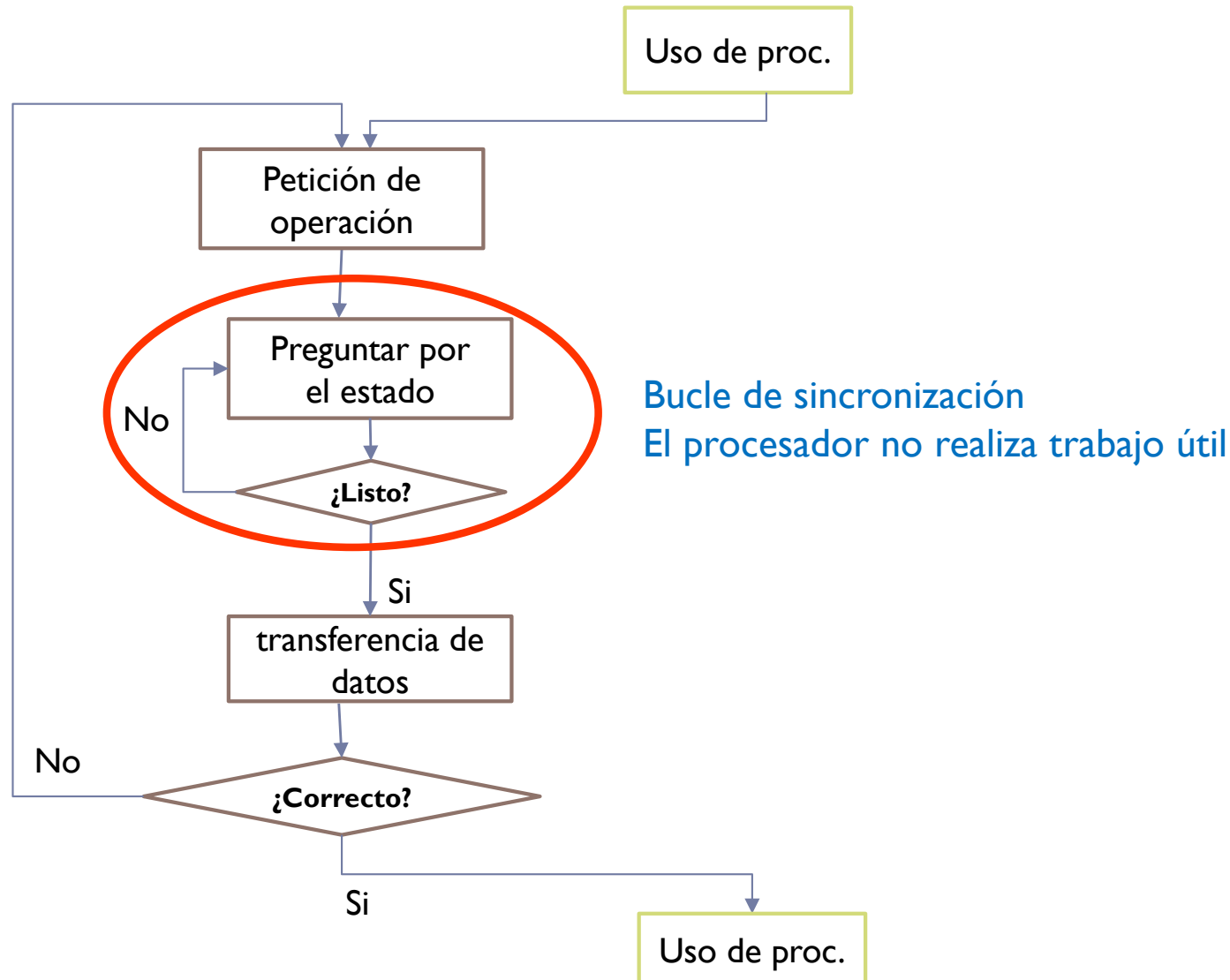
.text
.globl main
main:    li $t3 0
        bucle1: li $t0 0
                sw $t0 0x500
        bucle2: lw $t1 0x504
                beqz $t1 bucle2
                lw $t2 0x508
                sw $t2 datos($t3)
                add $t3 $t3 4
                bne $t3 400 bucle1
```



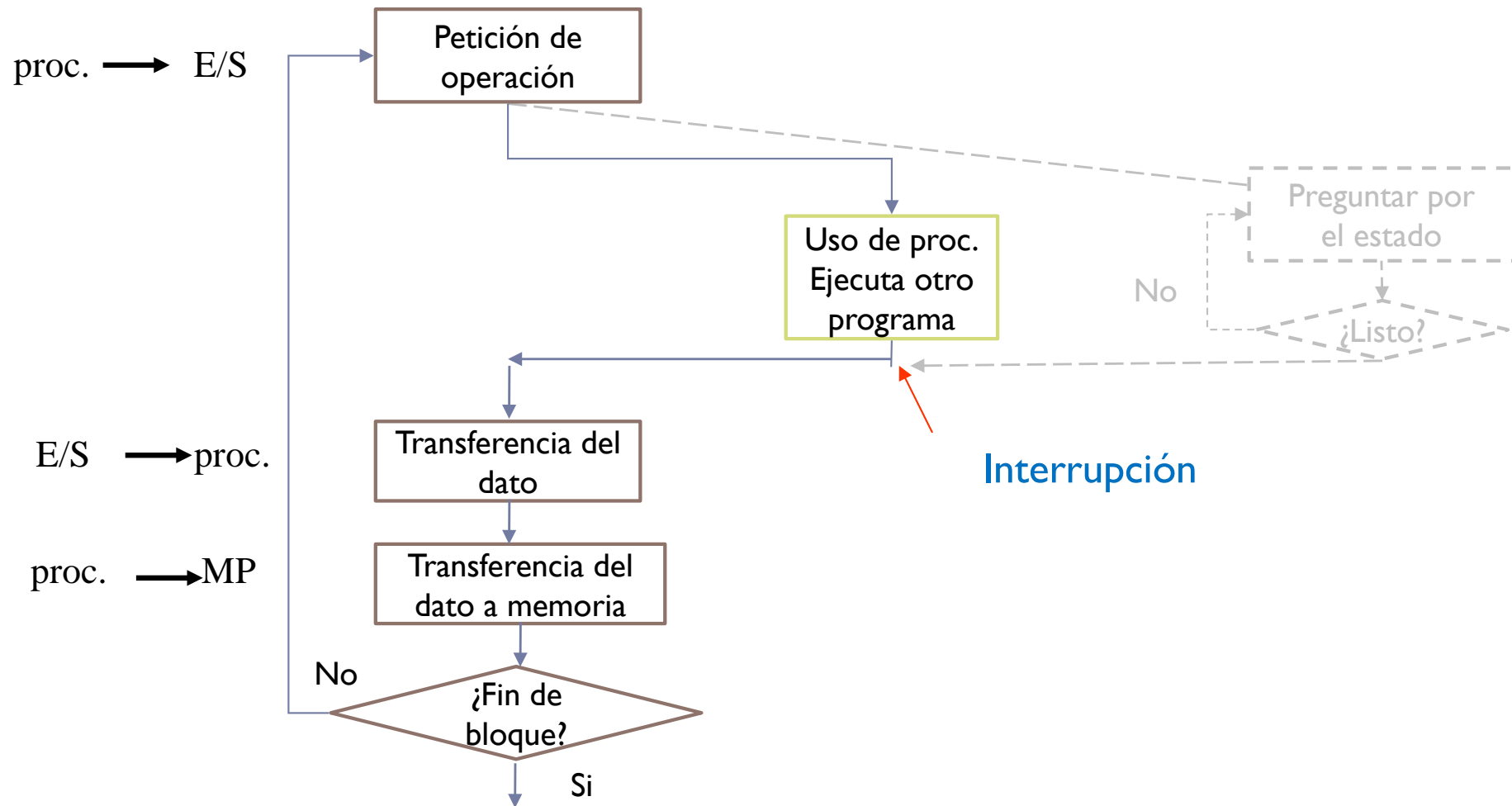
Bucle de sincronización

Bucle de transferencia

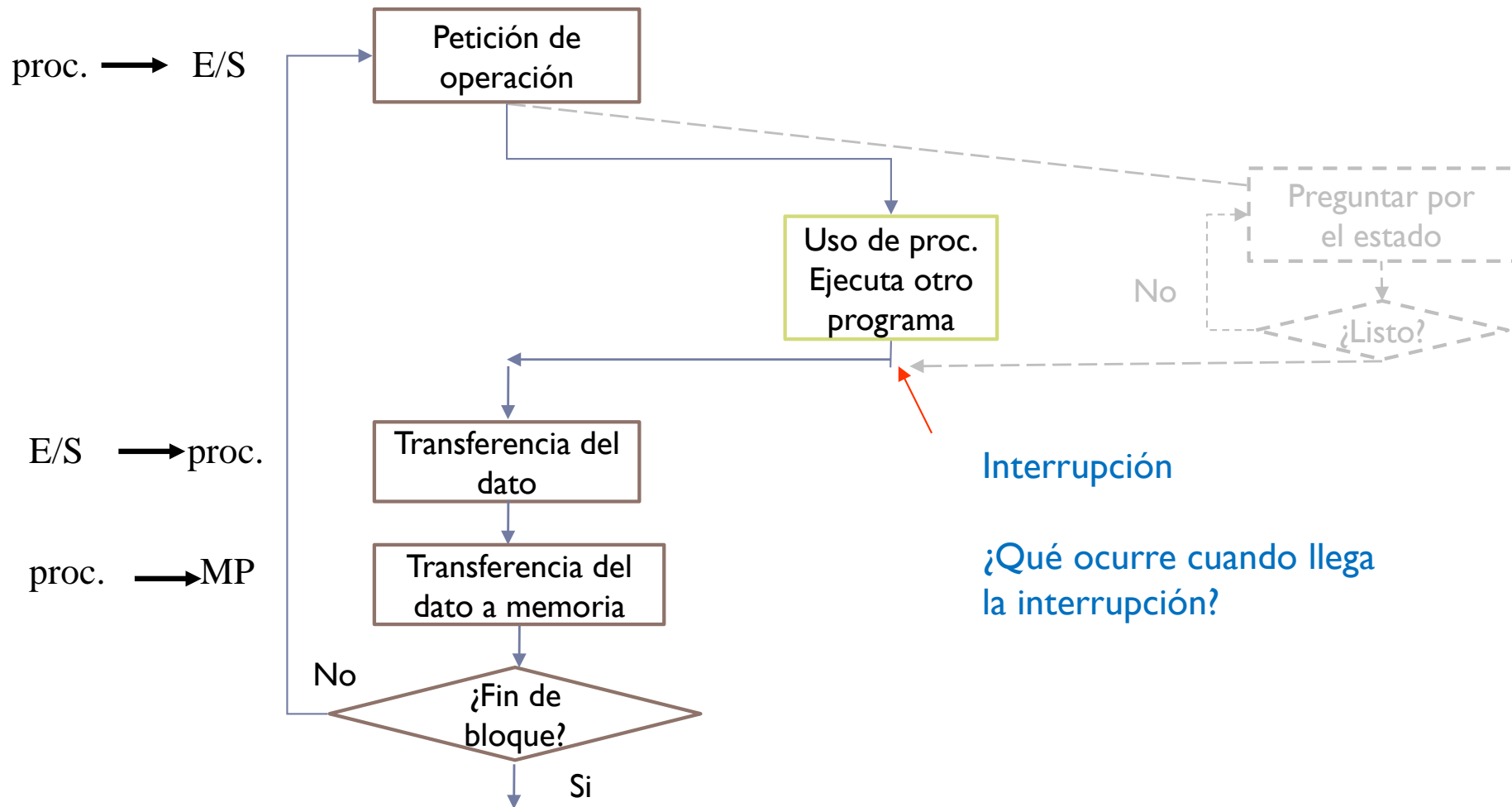
# Problema de la E/S programada



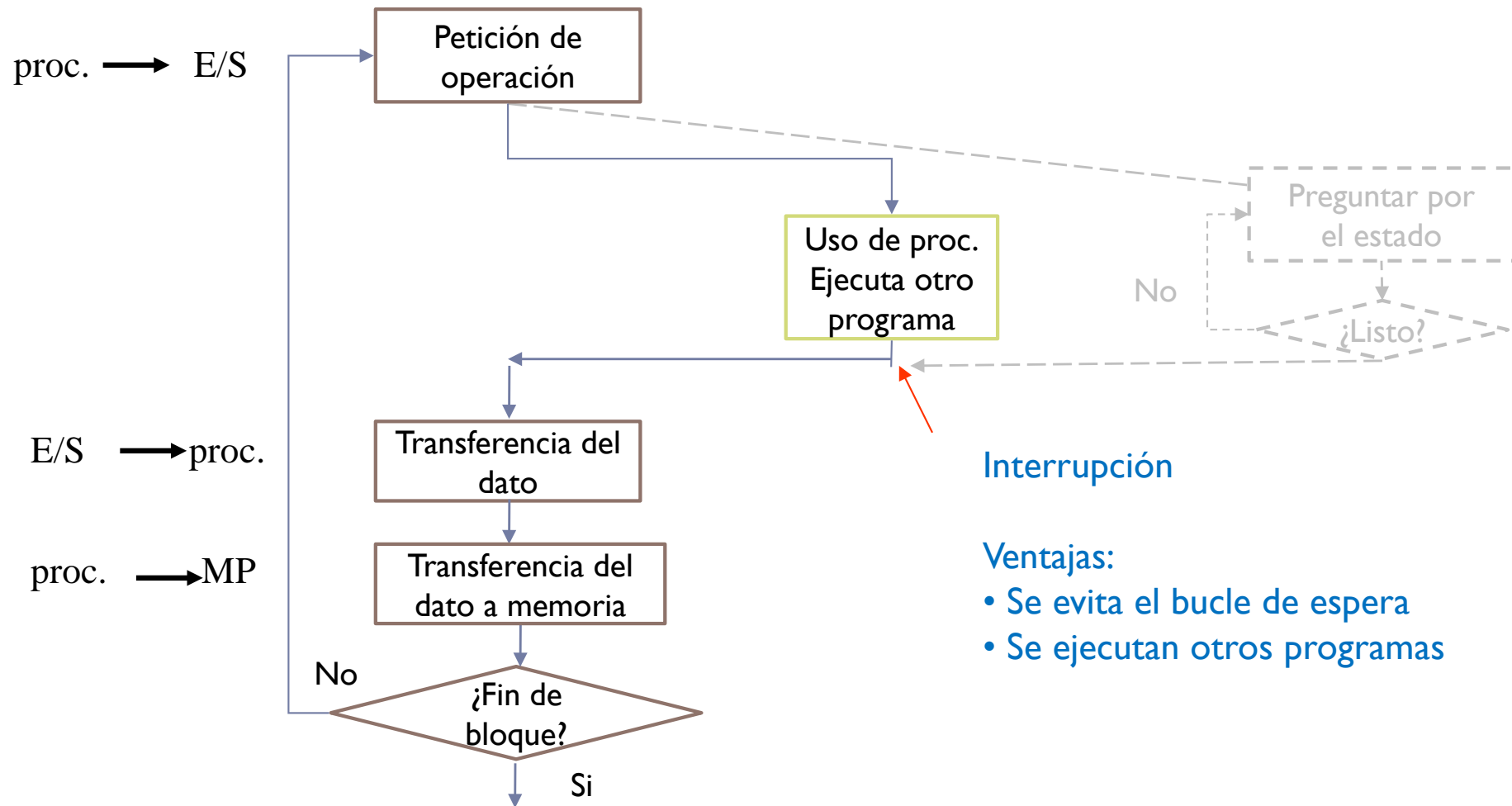
# E/S mediante interrupciones



# E/S mediante interrupciones

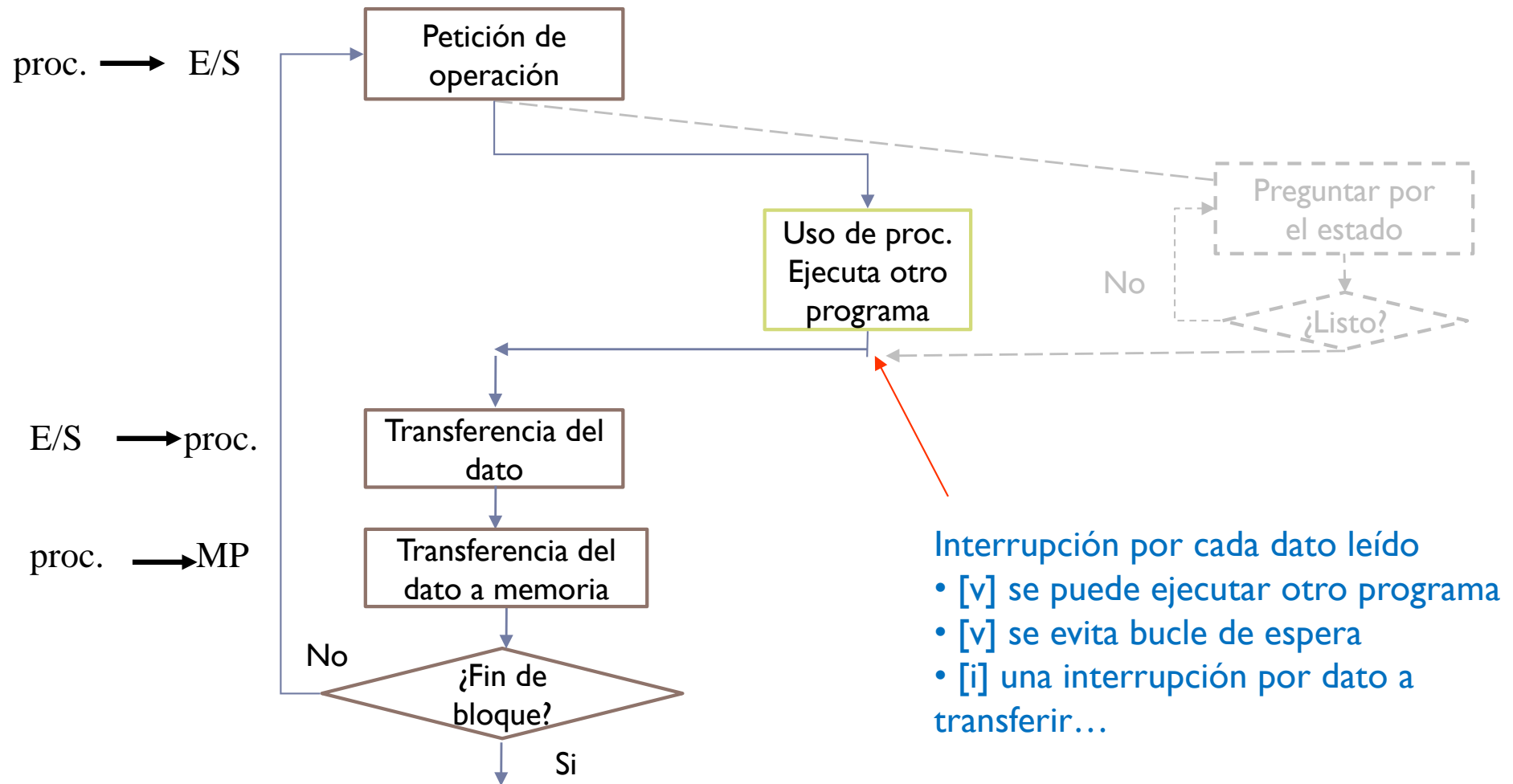


# E/S mediante interrupciones



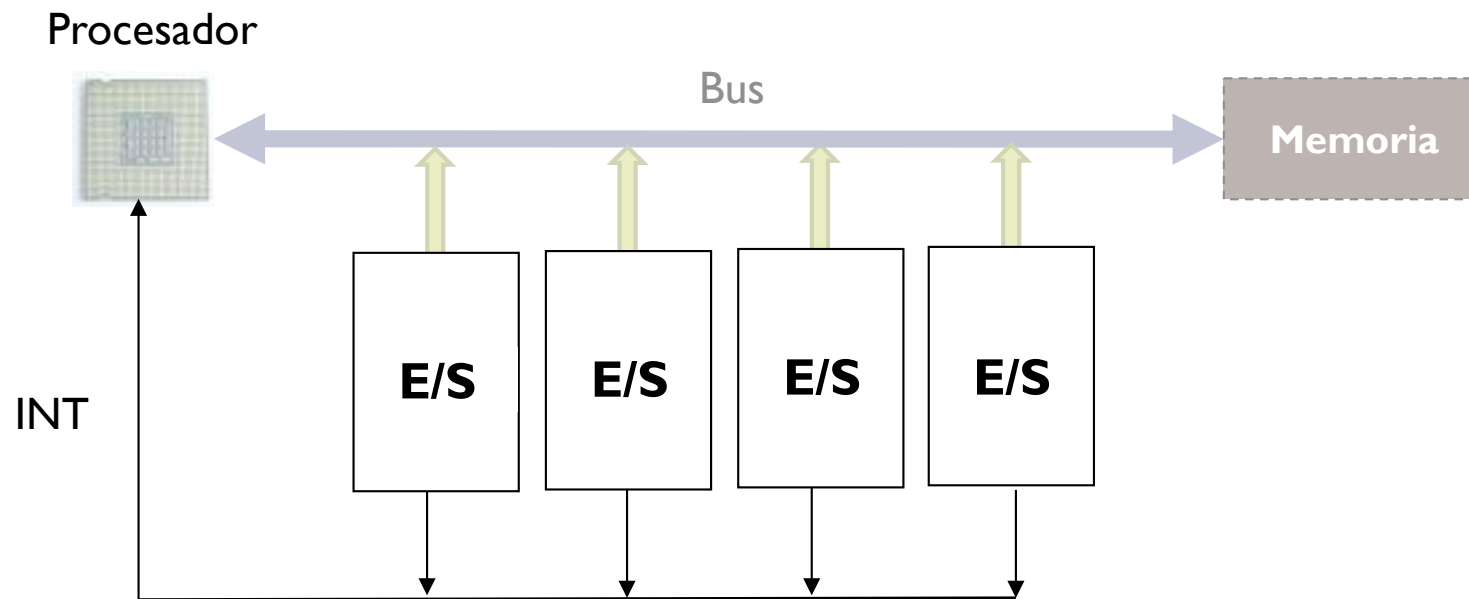
- Ventajas:
- Se evita el bucle de espera
  - Se ejecutan otros programas

# Interacción mediante interrupciones



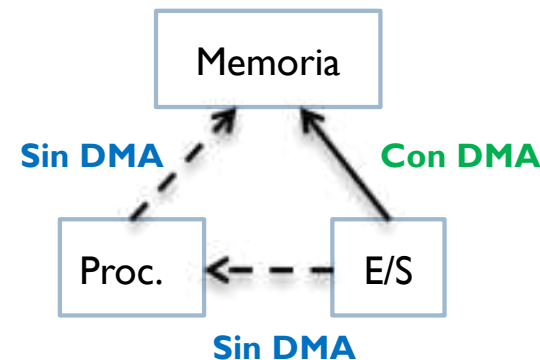


# E/S mediante interrupciones

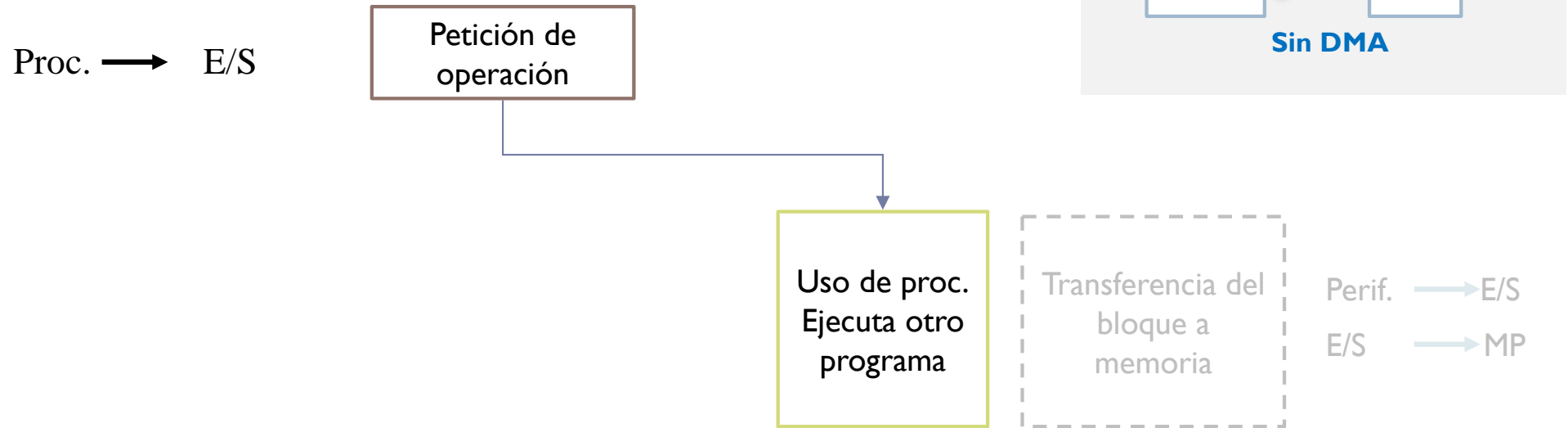


# E/S mediante DMA

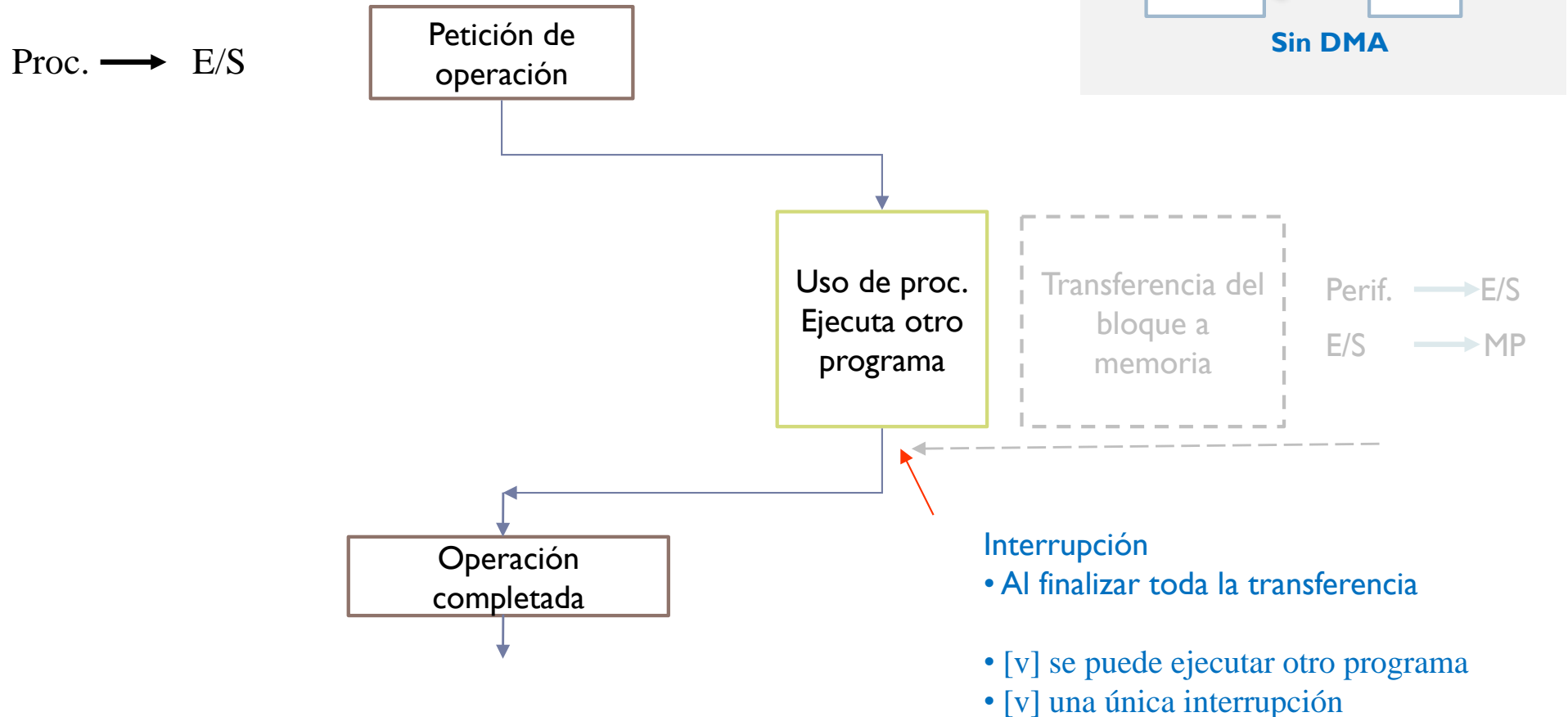
- ▶ DMA (Direct Memory Access): Acceso directo a memoria
- ▶ El procesador no realiza la transferencia entre la unidad de E/S y la memoria
  - ▶ Con interrupciones se evita el bucle de espera pero la transferencia la lleva a cabo el procesador
  - ▶ Para un bloque de N bytes, se generan N interrupciones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



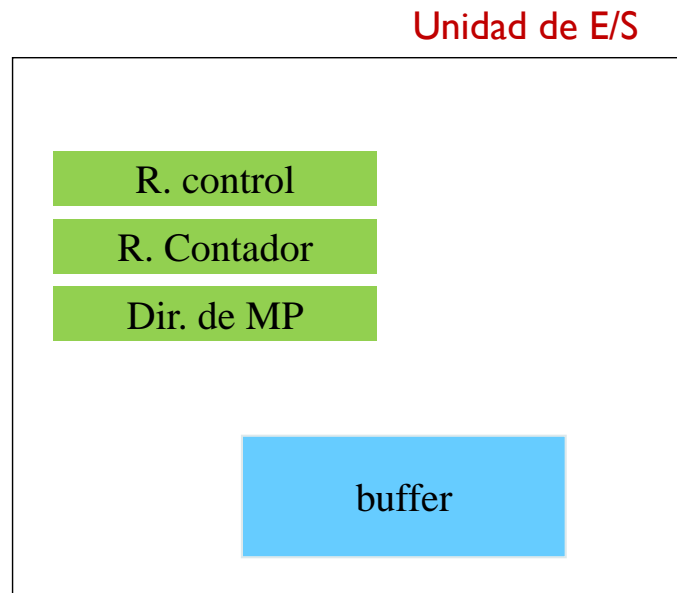
# Transferencia de un bloque mediante DMA



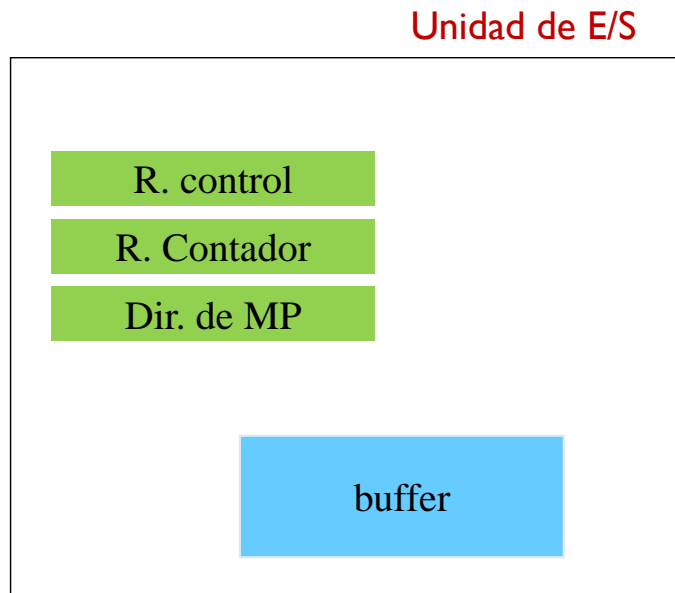
# Transferencia de un bloque mediante DMA



# Estructura simplificada de un módulo de E/S para DMA



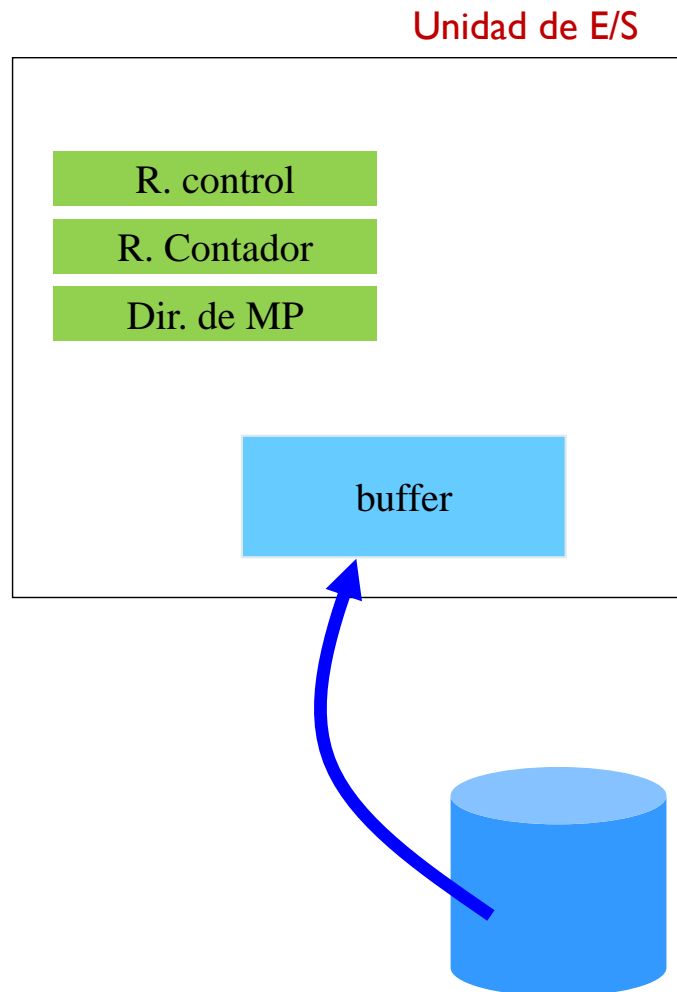
# Transferencia con DMA



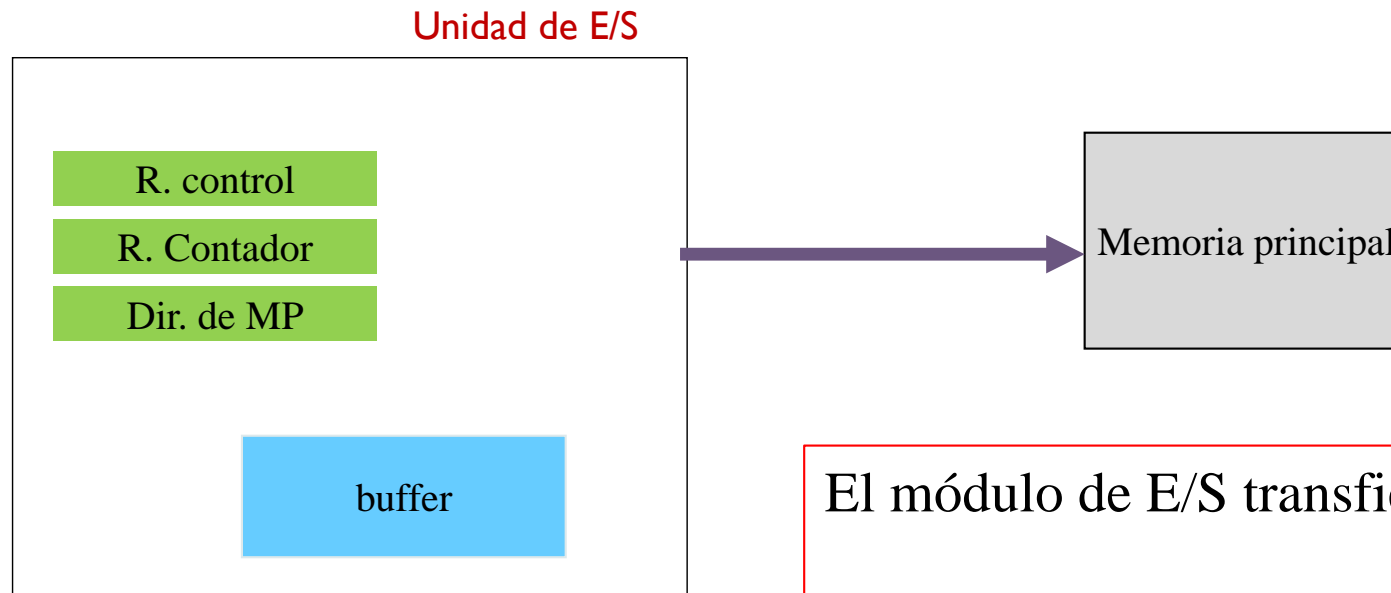
- ▶ El procesador escribe en los registros (con instrucciones de E/S)
  - ▶ La operación (**R. control**)
    - ▶ Lectura, escritura
  - ▶ El número de bytes a transferir (**R. contador**)
  - ▶ La **dirección de memoria principal** donde
    - ▶ Se almacenan los datos (escritura al periférico)
    -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

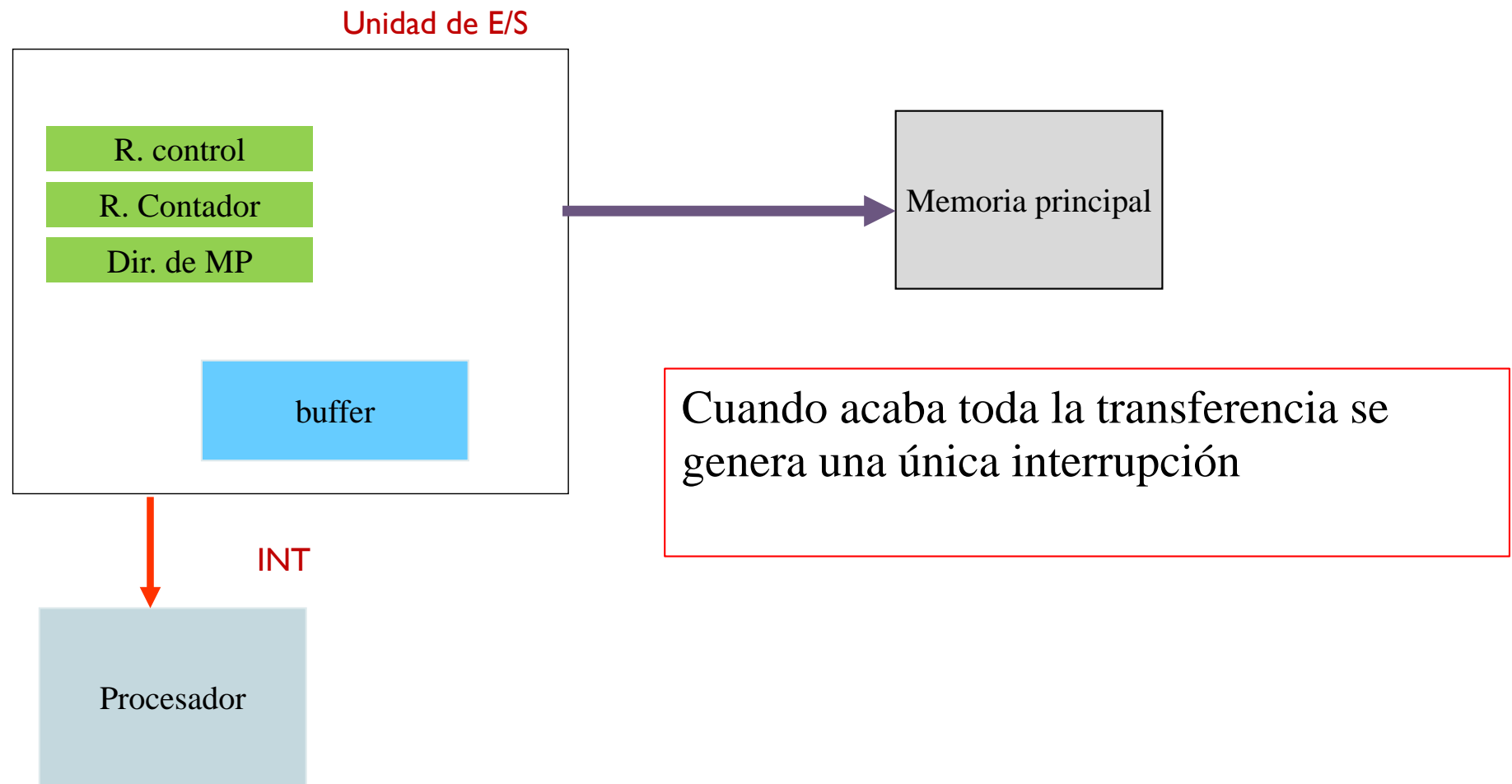


El módulo de E/S transfiere el bloque

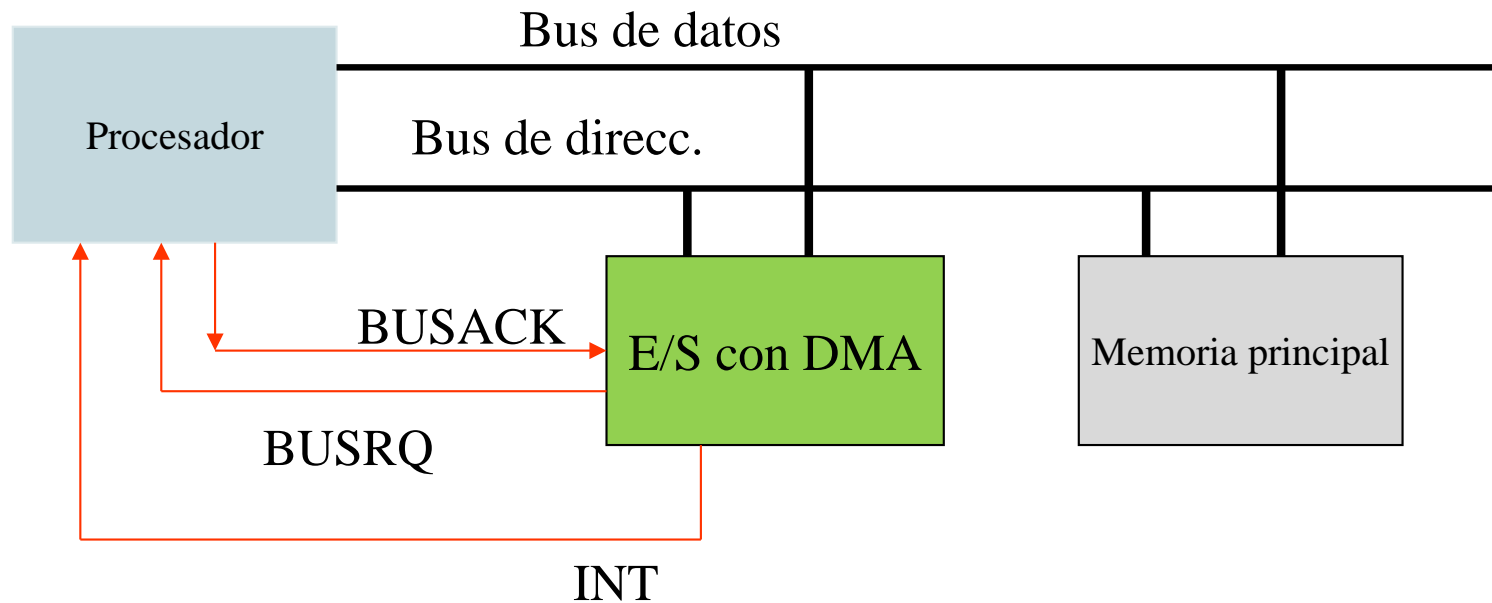
```
while (contador > 0)
{
    Byte (palabra)      Dir de MP
    Dir de MP ++;
    contador --;
}
```



# Transferencia con DMA



# Acceso del módulo de E/S a MP



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

# Acceso del módulo de E/S a MP

## Robo de ciclo



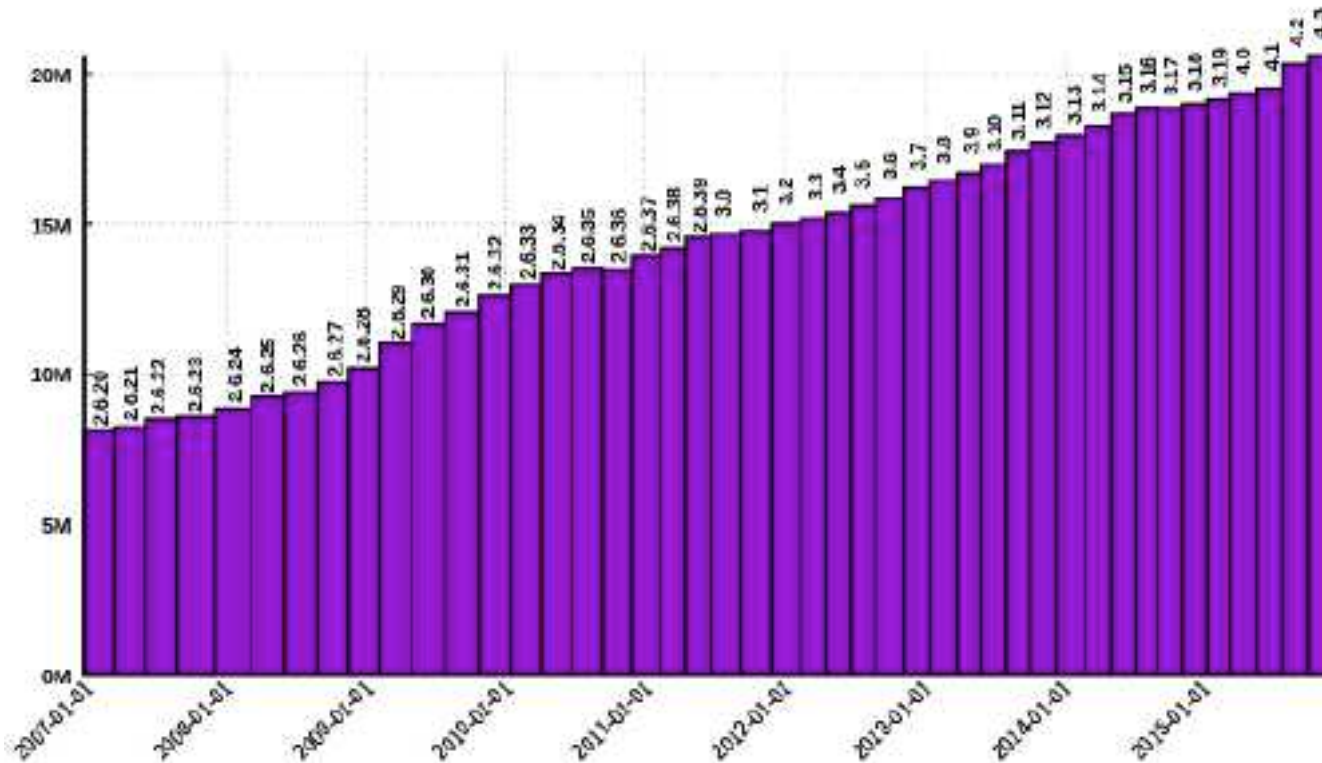
- ▶ Cuando la unidad de E/S está lista para transferir un dato:
  - ▶ Activa la señal **BUSRQ** para solicitar el acceso a los buses
  - ▶ Al final de cada fase de una instrucción el procesador comprueba la señal **BUSRQ**. Si está activa libera los buses y activa **BUSACK**
  - ▶ La unidad de E/S accede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

# 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.

# Importancia de los controladores Kernel de Linux



Líneas de  
código del  
kernel de  
Linux

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

# Ejercicio 1

- ▶ Sea un disco con 600 sectores por pista, con un con una velocidad de rotación de 7200 rpm y un tiempo medio de búsqueda de 2ms. Calcule el tiempo medio de acceso a un sector.

# Ejercicio 2

- ▶ Se desea desarrollar un controlador para un semáforo. El controlador dispone de una CPU de 32 bits, mapa de E/S común y juego de instrucciones del MIPS 32. A esta CPU se le conectan dos módulos de E/S. El primero es un cronómetro y el segundo es el módulo de E/S que controla el funcionamiento del semáforo. El módulo cronómetro dispone de los tres registros siguientes:
  - ▶ Registro con dirección 1000. En este registro se carga el valor correspondiente a la cuenta atrás en segundos.
  - ▶ Registro con dirección 1004. En este registro se carga un 1 cuando se quiere comenzar la cuenta atrás.
  - ▶ Registro con dirección 1008. Cuando la cuenta atrás llega a 0, en este registro se carga un 1. Mientras se está realizando la cuenta atrás el valor de este registro es 0.
- ▶ El módulo de E/S que controla el semáforo dispone de tres registros:
  - ▶ Registro con dirección 1012. En este registro se codifica el valor correspondiente al color del semáforo: 100 para el rojo, 010 para el amarillo y 001 para el verde.
- ▶ Se pide:
  - ▶ Escriba el programa ensamblador que controla el funcionamiento de este semáforo. El semáforo siempre comienza su funcionamiento en rojo. La duración del semáforo en rojo y verde es de 90 segundos y en amarillo de 20 segundos.

## Ejercicio 3

- Un computador tiene conectado un ratón que debe consultarse al menos 30 veces por segundo para poder actualizar su posición en la pantalla. La rutina que consulta su posición y vuelve a dibujar el puntero en la pantalla requiere 2000 ciclos para su ejecución. Si el computador tiene una frecuencia de 2.7 GHz, ¿qué sobrecarga supone la mencionada rutina de tratamiento de interrupciones, es decir, qué porcentaje de tiempo dedica el computador a ejecutar esta rutina.



# Ejercicio 4

- ▶ Se dispone de un computador que tiene conectados un sensor que mide la temperatura de un horno y una alarma. El sensor de temperatura se conecta al computador a través de un módulo de E/S que dispone de los tres siguientes registros:
  - ▶ **Registro de control** (dirección: ST\_REG\_CONTROL). Este registro se utiliza para indicar la operación a realizar sobre el sensor. Se pueden indicar dos operaciones:
    - ▶ Inicialización del dispositivo. El dispositivo se inicializa mediante el valor 0
    - ▶ Lectura de la temperatura. Esta operación se indica con el valor 1.
  - ▶ **Registro de datos** (dirección: ST\_REG\_DATOS). En este registro se almacena el valor de la temperatura tomado por el sensor.
  - ▶ **Registro de estado** (dirección: ST\_REG\_ESTADO). Este registro puede almacenar dos valores:
    - ▶ Listo (valor 1): cuando el dispositivo se ha inicializado o se dispone de una medida de temperatura válida.
    - ▶ Ocupado (valor 0): cuando el dispositivo está inicializándose o realizando una toma de medida.

## Ejercicio 4 (cont)

- ▶ Por su parte, la alarma se conecta a un módulo de E/S que dispone de dos registros:
  - ▶ **Registro de control** (dirección: A\_REG\_CONTROL). Este registro se utiliza para activar o desactivar la alarma. Para activar la alarma se escribe en este registro el valor 1. Para desactivarla se escribe el valor 0.
  - ▶ **Registro de estado** (dirección: A\_REG\_ESTADO). Este registro permite conocer el estado de la alarma. Si el valor de este registro es 0, la alarma se encuentra desactivada. Si el valor almacenado es 1, la alarma está activada.
- ▶ El computador dispone de un mapa de E/S separado y dos instrucciones de E/S:
  - ▶ `in RegProcesador, RegE/S` que carga en el registro del procesador RegProcesador el dato almacenado en el registro de un módulo de E/S con dirección RegE/S.
  - ▶ `out RegProcesador, RegE/S` que carga en el registro del módulo de E/S con dirección regE/S el dato almacenado en el registro del procesador RegProcesador.
- ▶ Todos los registros de los módulos de E/S son registros de 32 bits. Escriba utilizando el ensamblador del MIPS un programa que lea continuamente la temperatura del horno. Si la temperatura supera los 100° C, se debe activar la alarma y dejarla activada mientras la temperatura se encuentre por encima de los 100° C. Cuando la temperatura caiga por debajo de los 100° se debe desactivar la alarma.

# Ejercicio 5

- ▶ Sea un computador de 32 bits con direccionamiento de bytes y mapas de direcciones separados para memoria y entrada/salida. Su arquitectura ofrece el juego de instrucciones del MIPS R2000 más las instrucciones in y out, que permiten leer y, respectivamente, escribir en los registros de los módulos de E/S:
  - ▶ in rdest, dirección
  - ▶ out rsrc, dirección
- ▶ A este computador se encuentra conectado un sensor de nivel de aguas Px con, entre otras, las siguientes características:
  - ▶ Su módulo o unidad de E/S dispone de los siguientes registros de 32 bits:
    - ▶ Reg. de Control (R\_CONTROL\_Px)
    - ▶ Reg. de Estado (R\_ESTADO\_Px)
    - ▶ Reg. de Datos (R\_DATOS\_Px)
  - ▶ Sólo dispone de dos mandatos o ‘comandos’, ON (activa el sensor) y OFF (desactiva el sensor).
  - ▶ El Reg. de Estado tiene tres valores posibles: MIDIENDO, NUEVO y ERROR.
  - ▶ El modo de operación del sensor es el siguiente: cuando se activa pone su estado inicialmente a MIDIENDO, a partir de ese momento devolverá (en su Reg. de Datos) una nueva medida (32 bits) del nivel del agua cada vez que detecte una variación en el nivel mayor de un determinado umbral. Cuando suministra una nueva medida en el Reg. de Estado aparece el valor NUEVO, y cuando detecta que se ha leído éste (del Reg. de Datos) cambia su valor a MIDIENDO, hasta la llegada de una nueva medida. Además es posible que se produzca alguna anomalía en el sensor, dejando de estar operativo, circunstancia que indica con el valor ERROR en el Reg. de Estado.

# Ejercicio 5 (cont.)

- ▶ Se pide:
  - ▶ Escribir un programa (*driver*) para manejar este periférico mediante *E/S Programada* con el siguiente funcionamiento:
    1. Se activa el sensor.
    2. Hasta que se complete un total de 100 medidas, cada vez que llega una nueva se deberá almacenar en la siguiente posición de la zona de Memoria asignada, que comienza en la dirección `M_ALMACEN`.
    3. Si en cualquier momento se detecta que se ha producido un error de funcionamiento del periférico, se deberá interrumpir la lectura de las 100 medidas y escribir en la posición de Memoria `M_CODIGO` el valor `PROBLEMAS`.
    4. Si se completan las 100 medidas se escribirá en la posición de Memoria `M_CODIGO` el valor `TODO_OK`. Indicar qué inconvenientes tiene el uso aquí de esta técnica de *E/S Programada* o *Directa* y cómo se podría utilizar en su lugar la de *E/S por Interrupciones*.
  - ▶ Indique brevemente el beneficio que supondría su uso en este caso.

NOTA: todos los identificadores (nombres simbólicos) que aparecen en el enunciado tendrán un determinado valor (no especificado aquí) y corresponden siempre a datos de 32 bits.