

Dept. de Teoría de la Señal y Comunicaciones y Sistemas
Telemáticos y Computación
Área de Telemática (GSyC)

Almacenamiento y resto de Entrada/Salida

Katia Leal Algara

katia.leal@urjc.es

<http://gsyc.escet.urjc.es/~katia/>

Introducción

- ❑ El sistema de E/S está formado por:
 - ❑ Una serie de **dispositivos periféricos** que permiten la transferencia de información entre la computadora y el exterior
 - ❑ **Otros elementos** imprescindibles para su conexión y control

Introducción

- ❑ Los sistemas de E/S ponen más énfasis en la **confiabilidad** y en el coste. Además, se debe planificar su capacidad de **expansión**
- ❑ Sin embargo, los procesadores y la memoria se centran en el **rendimiento** y en el coste
- ❑ El rendimiento de la E/S depende de:
 - ❑ las características del dispositivo
 - ❑ de la conexión entre el dispositivo y el resto del sistema
 - ❑ la jerarquía de memoria
 - ❑ el sistema operativo

Introducción

- ❑ Gran olvidado en el campo de la Arquitectura de Computadores
- ❑ Sin embargo, por mucho que se optimice el funcionamiento del procesador y de la jerarquía de memoria, el **sistema de E/S** puede suponer un gran cuello de botella afectando negativamente al rendimiento global del sistema

Introducción

- ❑ Para obtener el rendimiento deseado del sistema de E/S no sólo hay que tener en cuenta el funcionamiento de los dispositivos de E/S, sino también la tecnología que se emplea para conectar los dispositivos con la computadora y la forma en que se gestiona la transferencia de información

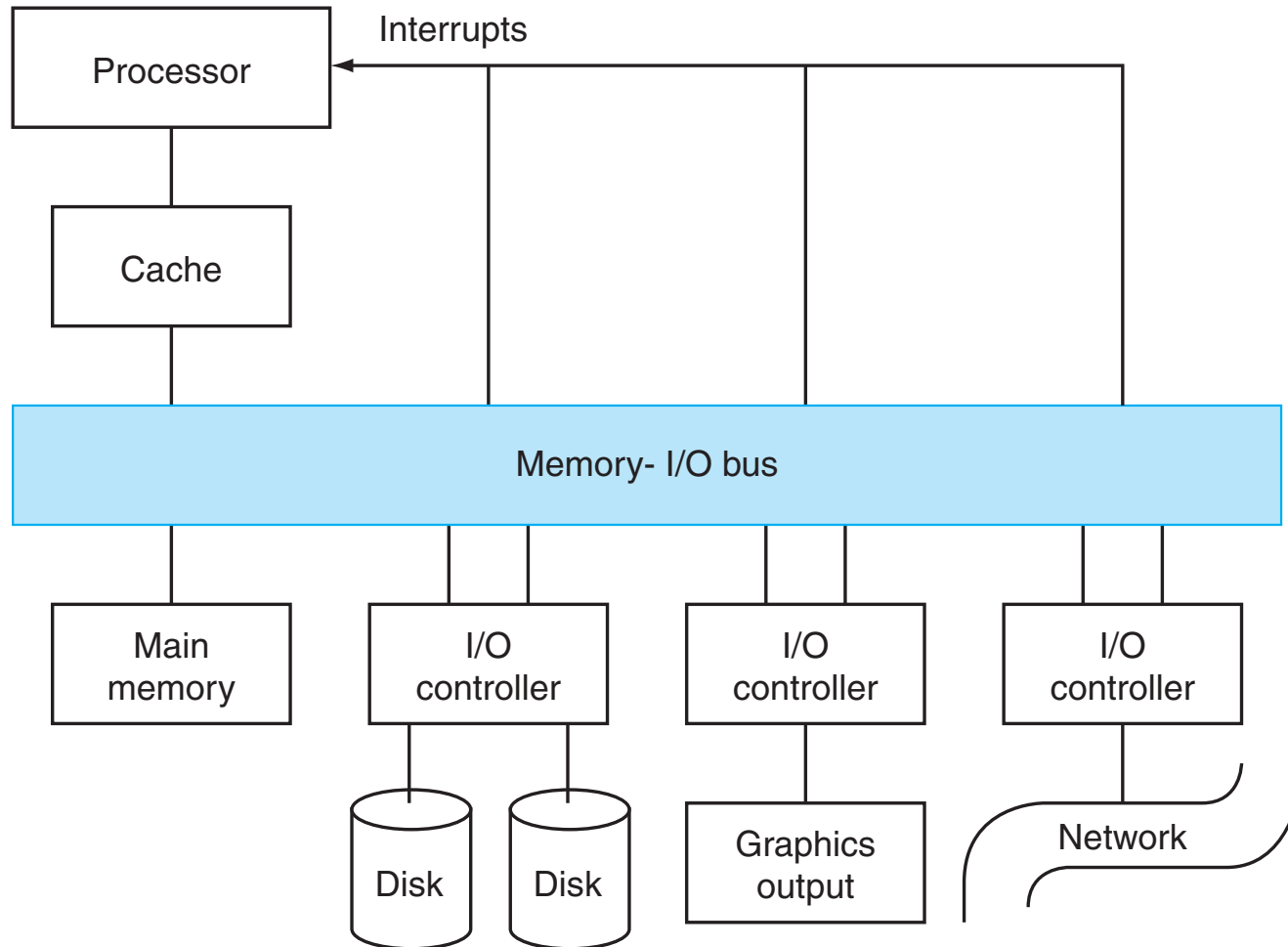
Contenidos

- ❑ Se estudian tres grandes aspectos del sistema de E/S
 - ❑ Buses
 - ❑ Dispositivos de E/S
 - ❑ **Gestión del sistema de E/S**

Mecanismo completo de una operación de E/S

- ❑ Aunque nos centraremos en la parte hardware, tanto el SO como los drivers también tienen un papel destacado en la transferencia de información
- ❑ Las conexiones entre los dispositivos de E/S, el procesador y la memoria se denominan históricamente **buses**:
 - ❑ Líneas/cables paralelos compartidos
- ❑ Sin embargo, actualmente las conexiones de E/S están más cerca de ser líneas serie dedicadas

Mecanismo completo de una operación de E/S



Mecanismo completo de una operación de E/S

- ☐ Existe una gran variedad de dispositivos que se pueden conectar a una computadora
- ☐ **E/S básica:** teclado, ratón, monitor
- ☐ **Almacenamiento:** disco duro, CD, DVD
- ☐ **Impresión:** impresora, escáner
- ☐ **Comunicaciones:** modem, acceso a red LAN
- ☐ **Multimedia:** micrófono, altavoces, capturadora, cámara de vídeo
- ☐ **Automatización y control:** sensores, alarmas, adquisición de datos

Clasificación de los periféricos

- ☐ Comportamiento:
 - ☐ **Entrada** (lectura)
 - ☐ **Salida** (escritura)
 - ☐ **Almacenamiento** (lectura-escritura)
- ☐ Interlocutor: en el otro extremo del dispositivo de E/S puede haber un **humano** o una **máquina**
- ☐ Velocidad de transferencia de datos: velocidad pico a la que se pueden transferir datos entre un dispositivo de E/S y la memoria principal o el procesador

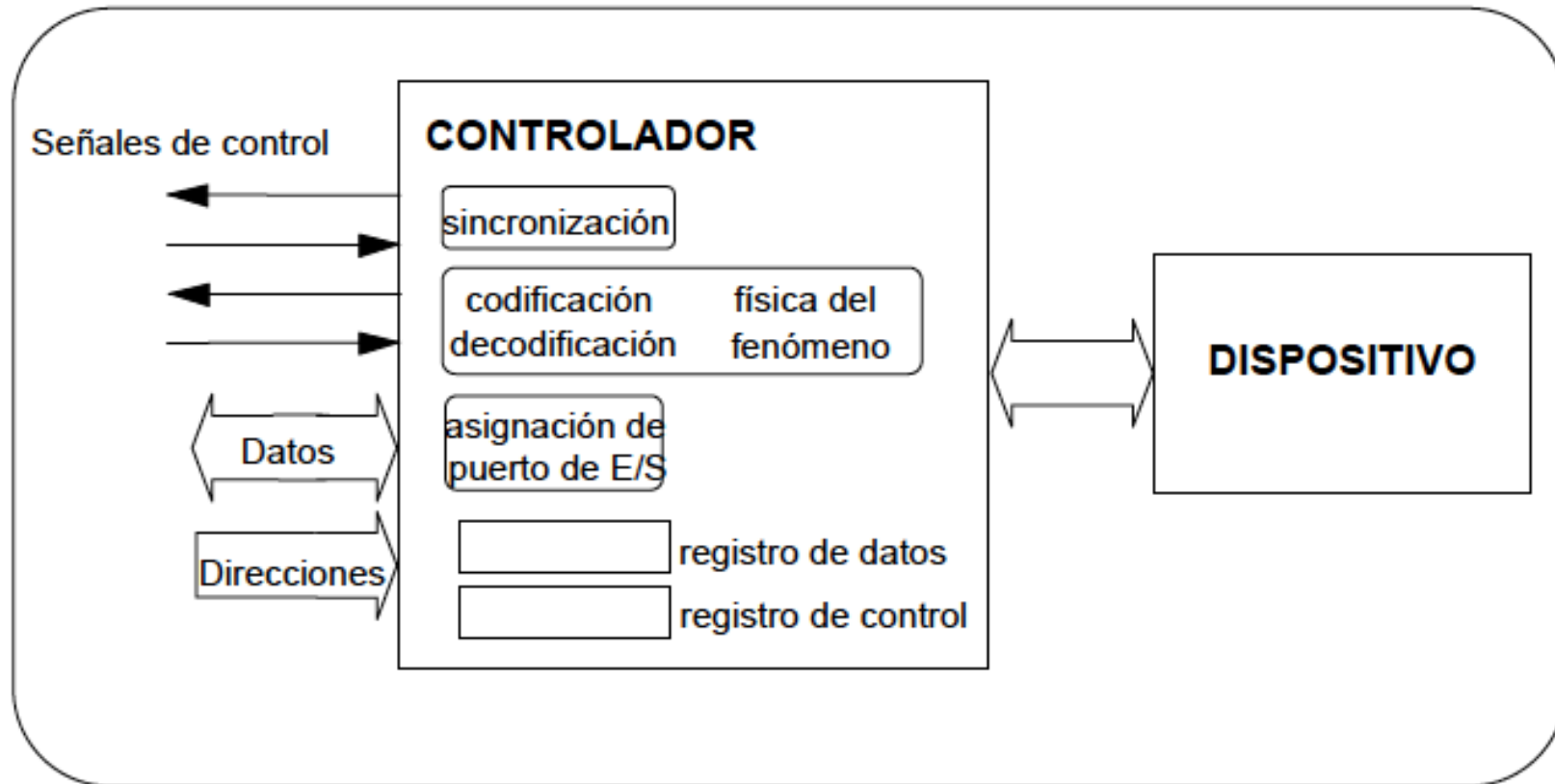
Clasificación de los periféricos

Device	Behavior	Partner	Data rate (Mbit/sec)
Keyboard	input	human	0.0001
Mouse	input	human	0.0038
Voice input	input	human	0.2640
Sound input	input	machine	3.0000
Scanner	input	human	3.2000
Voice output	output	human	0.2640
Sound output	output	human	8.0000
Laser printer	output	human	3.2000
Graphics display	output	human	800.0000–8000.0000
Modem	input or output	machine	0.0160–0.0640
Network/LAN	input or output	machine	100.0000–1000.0000
Network/wireless LAN	input or output	machine	11.0000–54.0000
Optical disk	storage	machine	80.0000
Magnetic tape	storage	machine	32.0000
Magnetic disk	storage	machine	240.0000–2560.0000

Componentes de un periférico

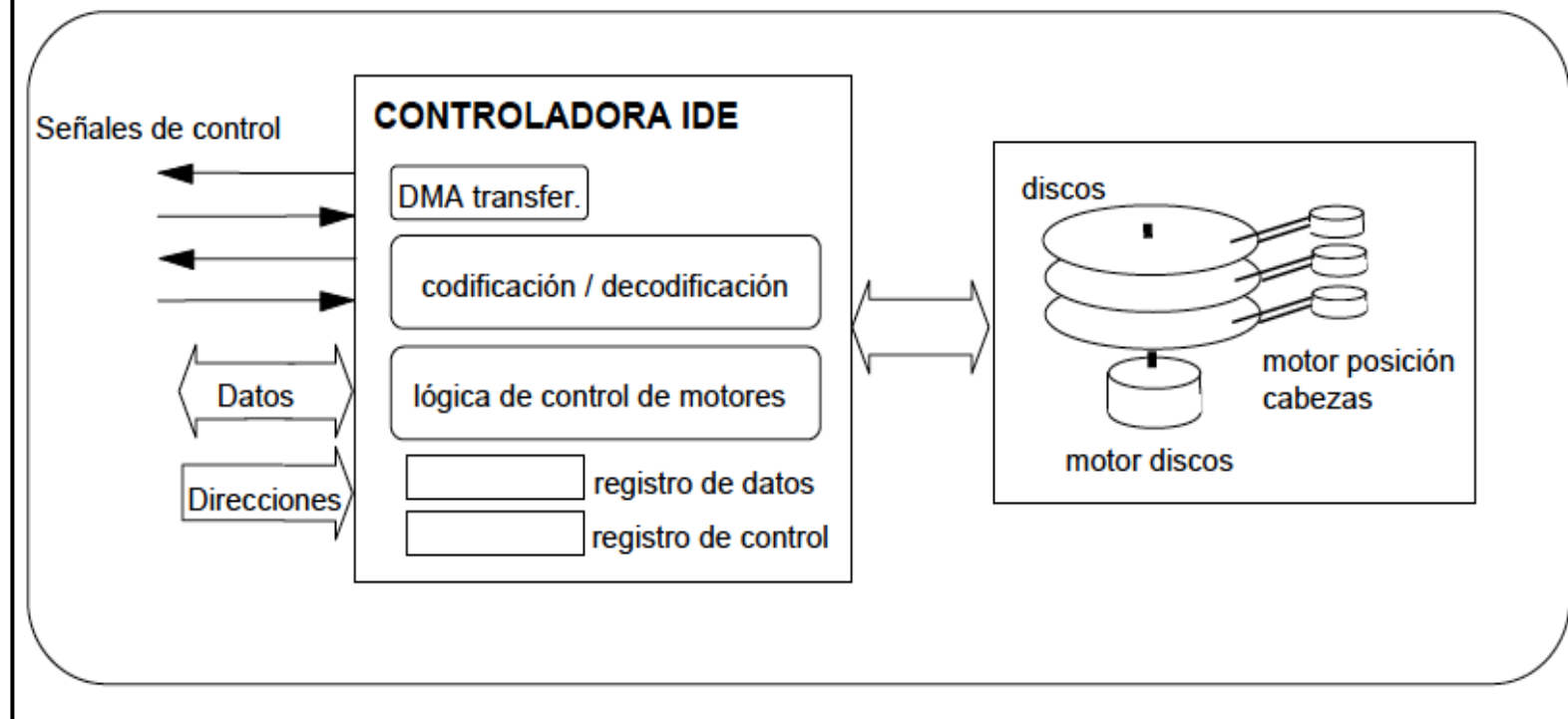
- ❑ Un periférico tiene dos partes
 - ❑ Controlador: comunicación con la CPU
 - ❑ Dispositivo: mecánico, electromecánico o electromagnético
- ❑ Misión del **controlador**
 - ❑ Transferencia de datos (objeto propio de la comunicación)
 - ❑ Protocolo de la transferencia (“ponerse de acuerdo” en el cómo y el cuándo)
- ❑ Misión del **dispositivo**
 - ❑ Relacionarse con el exterior

Ejemplos de periféricos: GENÉRICO



Ejemplos de periféricos: UNIDAD DE DISCO

- ❑ El interfaz **ATA** (Advanced Technology Attachment) o PATA, originalmente conocido como **IDE** (Integrated device Electronics), es un estándar de interfaz para la conexión de los dispositivos de almacenamiento masivo de datos y las unidades ópticas que utiliza el estándar derivado de ATA y el estándar ATAPI



Mecanismo completo de una operación de E/S

- ❑ Todos los dispositivos se conectan al sistema a través de un **interfaz** cuyas funciones principales son:
 - ❑ Interpretar las órdenes que recibe del procesador y transmitir las al periférico
 - ❑ Controlar las transferencias de datos entre el procesador y el periférico (convertir formatos, adaptar velocidades)
 - ❑ Informar al procesador del estado del periférico
- ❑ **Unidad hardware/software** ubicada en una tarjeta o adaptador
 - ❑ La parte software está ligada a los drivers y al SO

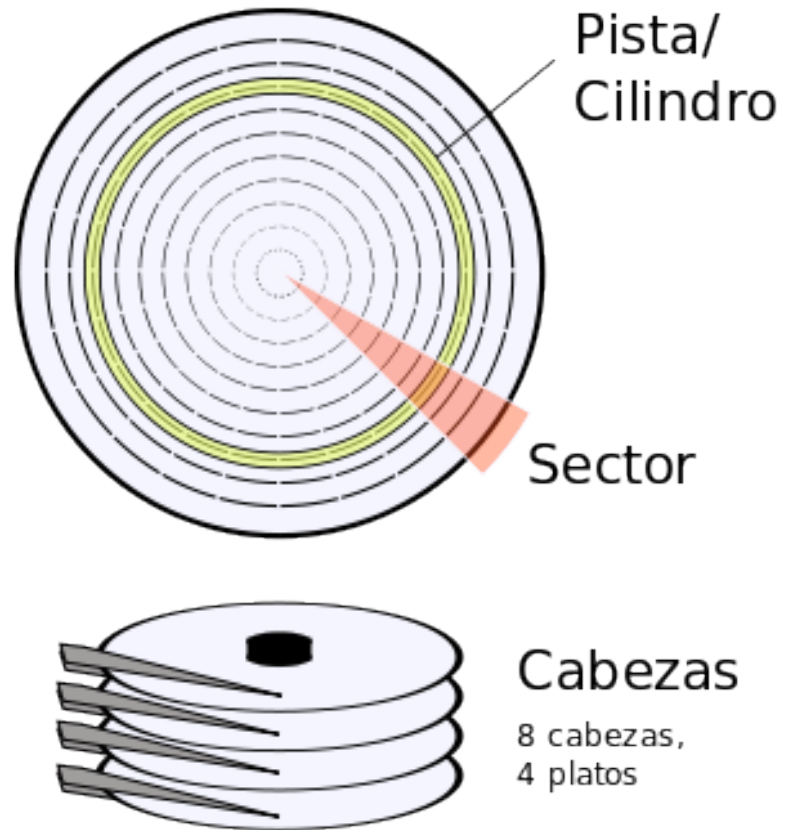
Mecanismo completo de una operación de E/S

- ❑ Los interfaces se clasifican como:
 - ❑ **Serie o paralelo**: según cómo manejen la información
 - ❑ **Generales o específicos**: según estén diseñados para dar soporte a un gran conjunto de dispositivos o a un tipo de dispositivo concreto
- ❑ La unidad física que permite conectar el dispositivo con el interfaz se denomina **puerto**
- ❑ El interfaz se conecta mediante un **bus de E/S** con un controlador, situado en un chipset o hub, que se encarga de la transferencia de información con el procesador o con la memoria según los mecanismos de gestión de E/S disponibles

Disco duro

- ❑ Dentro de un disco duro hay uno o varios discos concéntricos llamados **platos** que giran todos a la vez sobre el mismo eje, al que están unidos
- ❑ El **cabezal** está formado por un conjunto de brazos paralelos a los platos, en cuya punta están las cabezas de lectura/escritura. Hay una cabeza de lectura/escritura para cada **cara** de cada plato
 - ❑ Los cabezales pueden moverse hacia el interior o el exterior de los platos, combinado con la rotación, permite que los cabezales puedan alcanzar cualquier posición de la superficie de los platos

Disco duro



Disco duro

- ❑ Para acceder a un dato, el SO debe dirigir al disco en un proceso de tres pasos
- ❑ **Paso 1**: posicionar la cabeza sobre la pista adecuada, *seek*. El tiempo empleado en mover la cabeza se denomina ***seek time***
- ❑ **Paso 2**: una vez hemos alcanzado la pista, hay que esperar a que el sector deseado pase debajo de la cabeza de lectura/escritura. A este tiempo se le denomina ***latencia rotacional***
- ❑ **Paso 3**: el acceso al disco lo completa el ***tiempo de transferencia***, que es el tiempo necesario para transferir un bloque de bits.
- ❑ Finalmente, el controlador de disco gestiona detalles de control del disco y de la transferencia entre el disco y la memoria. El controlador suma un tiempo final al tiempo total de acceso al disco, el ***tiempo de controlador***

Disco duro: *Interfaces de controlador de disco*

- ❑ **IDE**: Integrated Drive Electronics o ATA (Advanced Technology Attachment). Principal estándar por su versatilidad hasta 2004.
- ❑ **SATA** (Serial ATA): el más novedoso de los estándares de conexión, utiliza un bus serie para la transmisión de datos. Más rápido, pequeño, eficiente y cómodo que IDE. Además de permitir **conexión en caliente**.
- ❑ **SCSI**: interfaces para discos duros de gran capacidad de almacenamiento y velocidad de rotación (servidores). Pueden trabajar asincrónicamente respecto al microprocesador, lo que posibilita una mayor velocidad de transferencia.
- ❑ **SAS** (Serial Attached SCSI): Interfaz de transferencia de datos en serie. Aumenta la velocidad y permite la conexión y desconexión en caliente. La tecnología SAS irá reemplazando a su predecesora SCSI.

Memoria flash

- ☐ Derivada de la memoria EEPROM, permite la lecto-escritura de múltiples posiciones de memoria en la misma operación
- ☐ Comparada con los discos duros, estos tienen un coste muy inferior, pero una capacidad muy superior
- ☐ Gran resistencia a los golpes, bajo consumo y muy silenciosa, ya que no contiene ni actuadores mecánicos ni partes móviles
- ☐ Pequeño tamaño y ligereza: factor determinante a la hora de utilizarla en dispositivos portátiles
- ☐ Los distintos tipos de memoria flash sólo permiten un número limitado de escrituras y borrados
- ☐ Los sistemas de archivos para estas memorias están en pleno desarrollo, algunos ya en funcionamiento como por ejemplo **JFFS**. Sin embargo, en la práctica se emplea un sistema de archivos **FAT** por compatibilidad, sobre todo en las tarjetas de memoria extraíble

Memoria flash Vs Discos duros

- ☐ Una unidad de estado sólido (no volátil) o **SSD** (*Solid-State Drive*) es un dispositivo de almacenamiento de datos
- ☐ Están construidas con chips de memoria flash
- ☐ No son discos, pero juegan el mismo papel a efectos prácticos
- ☐ Puede que en el futuro terminen sustituyendo al disco duro
- ☐ Son muy rápidas ya que **no** tienen partes móviles y consumen menos energía. Son fiables y físicamente duraderas. Sin embargo, su coste por GB es aún muy elevado
- ☐ **Los discos que no son discos:** en ocasiones, a las unidades de estado sólido se las clasificadas como “discos”. Esto es un error puesto que sus datos no se almacenan sobre superficies cilíndricas ni platos. Esta confusión conlleva habitualmente a creer que “SSD” significa *Solid State Disk*, en vez de *Solid State Drive*

Buses

- ❑ La principal desventaja del bus es que crea un cuello de botella, limitando el *throughput* de la E/S
 - ❑ Como la E/S debe pasar a través del bus, el ancho de banda de éste limita el máximo *throughput*
- ❑ Los buses se clasifican como:
 - ❑ **Buses procesador-memoria:** corto, de alta velocidad y específicos para dicho procesador y sistema de memoria, maximizando así el ancho de banda procesador-memoria
 - ❑ **Buses de E/S:** largos, soportan una gran variedad de dispositivos de E/S

Buses

- ☐ Diseñar un bus es complicado puesto que la velocidad del bus viene limitada por factores físicos:
 - ☐ Longitud del bus
 - ☐ Número de dispositivos conectados
- ☐ Además, el bus se debe diseñar para soportar un amplio abanico de dispositivos, cada uno con su propia latencia y tasa de transferencia de datos

Conexiones básicas

- ☐ Transacción de E/S
 - ☐ Enviar la dirección
 - ☐ Recibir o enviar los datos
- ☐ Transacción de lectura: leer datos de la memoria ==> ***Input***
- ☐ Transacción de escritura: escribir en memoria ==> ***Output***
- ☐ La industria ha desarrollado varios estándares
- ☐ Un **estándar** sirve como especificación tanto al fabricante de ordenadores como al fabricante de periféricos

Conexiones básicas

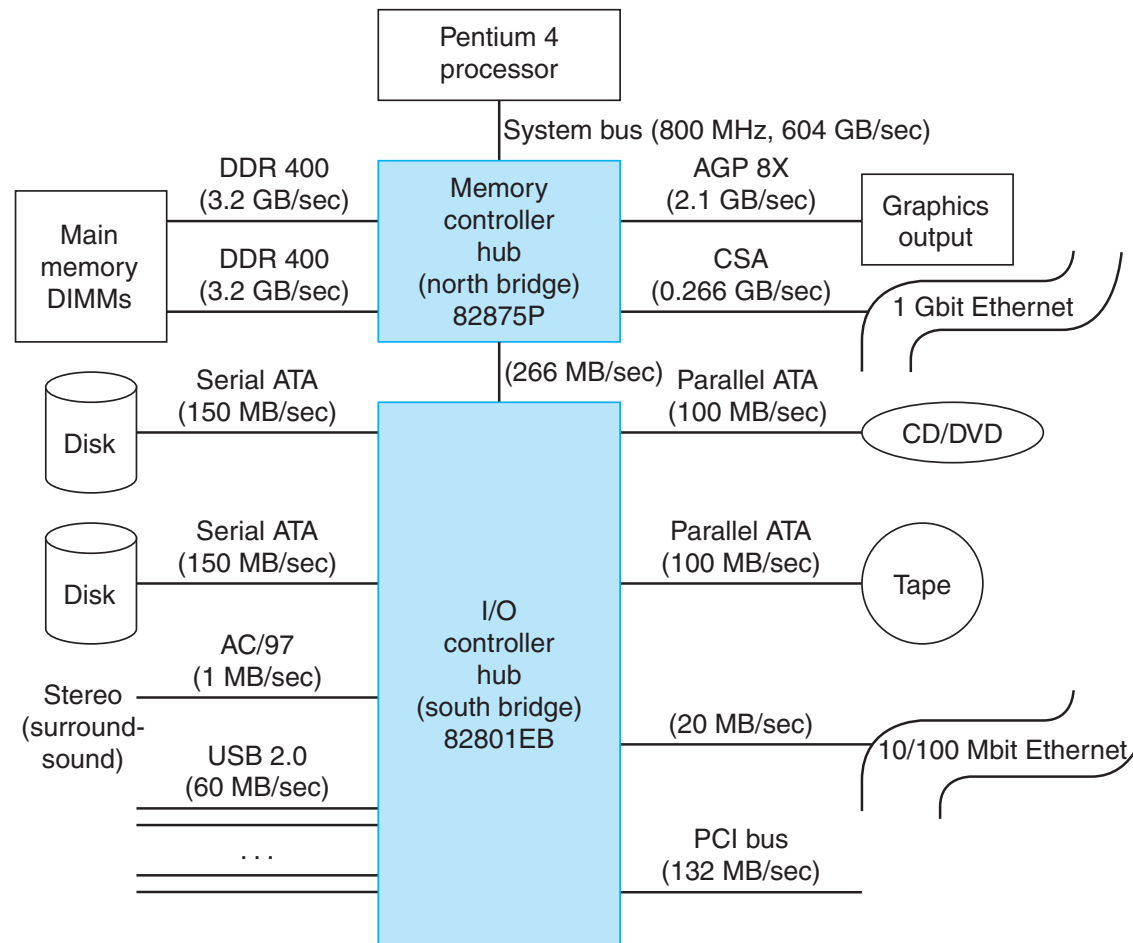
- ☐ Los buses tradicionales son **síncronos**
- ☐ Estos buses incluyen:
 - ☐ Un **reloj** en la líneas de control
 - ☐ Un protocolo fijo para la comunicación relativo al reloj
- ☐ Ejemplo protocolo:
 - ☐ Transmitir dirección y comando en el primer ciclo de reloj
 - ☐ La memoria debe responder con el dato en el quinto ciclo de reloj
- ☐ Ventajas:
 - ☐ El protocolo se puede implementar con una máquina de estados finita
 - ☐ Interfaz lógica pequeña y velocidad del bus alta
- ☐ Desventajas:
 - ☐ Todos los dispositivos deben trabajar a la misma velocidad, la que marca el reloj del bus
 - ☐ Debido a problemas en la señal de reloj, los buses no pueden ser muy largos si quieres ser rápidos

Conexiones básicas

- ☐ **Interconexiones asíncronas:** no están marcadas por una señal de reloj
- ☐ Se puede acomodar a una gran variedad de dispositivos
- ☐ Se puede alargar el bus sin preocuparse por problemas de sincronización con la señal de reloj
- ☐ Los cinco estándares más populares, USB, PCI Express (PCIe), Serial ATA (SATA) y Serial SCSI (SAS), son todos asíncronos
- ☐ Para coordinar la transmisión de datos entre el emisor y el receptor, los buses asíncronos utilizan un protocolo de apretón de manos, ***handshaking protocol***
- ☐ Este protocolo consiste en un serie de pasos en los que emisor y receptor proceden con el siguiente paso sólo cuando ambas partes están de acuerdo

Conectando el procesador, la memoria y los dispositivos de E/S

Interconexiones del procesador x86



Interfaz del sistemas de E/S

- ☐ Tareas necesarias para transferir datos desde un dispositivo hasta el espacio de direcciones de memoria de un proceso de usuario
- ☐ ¿Cómo se transforma una petición de usuario de E/S en un comando para el dispositivo y cómo se comunica dicho comando?
- ☐ ¿Cómo se transfieren los datos hacia y desde memoria?
- ☐ ¿Cuál es el papel del SO?

Interfaz del sistema de E/S

- ❑ El SO hace de interfaz entre el hardware y el programa que realiza la petición de E/S
- ❑ Las responsabilidades del SO se derivan de tres características del sistema de E/S
 - ❑ Varios programas que utilizan el procesador **comparten** el sistema E/S
 - ❑ Los sistemas de E/S por lo general utilizan **interrupciones** para comunicar información sobre el estado de las operaciones de E/S
 - ❑ El control de bajo nivel de un dispositivo de E/S es **complejo**

Interfaz del sistema de E/S

- ☐ El SO debe garantizar que un programa sólo accede a la porción de un dispositivo de E/S para la que tiene permiso
- ☐ El SO proporciona abstracciones para acceder a los dispositivos por medio de rutinas que manejan las operaciones de bajo nivel del dispositivo
- ☐ El SO gestiona las interrupciones de la misma manera que maneja las excepciones generadas por un programa
- ☐ El SO trata de proporcionar un acceso equitativo a los dispositivos de E/S compartidos, al mismo tiempo que planifica el acceso para incrementar la productividad

Interfaz del sistema de E/S

- ☐ Para realizar estas funciones en beneficio de los programas de usuario, el SO debe:
 - ☐ ser capaz de comunicarse con los dispositivos de E/S
 - ☐ evitar que los programas de usuario se comuniquen directamente con los dispositivos
- ☐ Se necesitan tres tipos de comunicación:
 1. El SO debe ser capaz de **ordenar comandos** a los dispositivos: read, write, disk seek ...
 2. El dispositivo debe **notificar al SO la finalización** de una operación de E/S o un error
 3. Los datos se deben transferir **entre la memoria y el dispositivo** de E/S

1. Ordenar comandos a los dispositivos de E/S

- ☐ Para poder dar órdenes a los dispositivos, el procesador debe ser capaz de **direccionar** a los dispositivos
- ☐ Existen dos maneras de direccionar dispositivos:
 - ☐ E/S mapeada en memoria
 - ☐ Instrucciones especiales para E/S

1. Ordenar comandos a los dispositivos de E/S

- ☐ La E/S mapeada en memoria consiste en **reservar parte del espacio de direcciones para los dispositivos de E/S**
 - ☐ Las lecturas y escrituras en dichas direcciones se interpretan como comandos para los dispositivos
- ☐ Ejemplo: una operación de escritura sobre un dispositivo se puede utilizar para enviar datos a un dispositivo donde los datos serán interpretados como un comando
 - ☐ El procesador pone el dato y la dirección en el bus de memoria, pero estos son ignorados por el controlador de memoria al tratarse de una dirección reservada a la E/S
 - ☐ El controlador del dispositivo sí que ve la operación, graba el dato y lo transmite al dispositivo como un comando
- ☐ Los programas de usuario no pueden realizar operaciones de E/S directamente
 - ☐ **Las direcciones de E/S están protegidas por el mecanismo de traducción**
- ☐ En cada evento, la dirección utilizada codifica la identidad del dispositivo y el tipo de transmisión entre éste y el procesador

1. Ordenar comandos a los dispositivos de E/S

- ☐ En la actualidad, realizar la lectura o escritura de datos para completar una petición de un programa de usuario normalmente requiere de varias operaciones de E/S separadas
- ☐ El procesador debe comprobar el estado del dispositivo entre comandos individuales para determinar si el comando se completó correctamente
- ☐ Ejemplo, una impresora tiene:
 - ☐ *Status register*: estado de la impresora
 - ☐ *Done bit*, indica si el carácter se ha imprimido
 - ☐ *Error bit*, indica falta de papel o atasco
 - ☐ *Data register*: almacena los bytes a imprimir

1. Ordenar comandos a los dispositivos de E/S

- ☐ La alternativa a la E/S mapeada en memoria es el uso de **instrucciones específicas de E/S**
- ☐ Estas instrucciones especifican el número de dispositivo y el comando (o la localización del comando en memoria)
- ☐ La comunicación se realiza por medio de una serie de cables del bus de E/S
- ☐ Para evitar que los usuarios puedan ejecutar instrucciones específicas para dispositivos de E/S, **sólo se permite la ejecución de dichas instrucciones en modo supervisor**

2. Comunicación con el procesador

- ❑ **Polling:** proceso por el que periódicamente se comprueba el estado de un dispositivo de E/S
 - ❑ Es la forma más simple que tiene un dispositivo de comunicarse con el procesador
 - ❑ El dispositivo pone la información en un registro de estado
 - ❑ El procesador obtiene dicha información
 - ❑ El procesador tiene un control completo y realiza todo el trabajo
 - ❑ Ejemplo de utilización: en aplicaciones en sistemas de tiempo real embebidos
 - ❑ La desventaja es que **se desperdician muchos ciclos del procesador** debido a la gran diferencia de velocidad entre los procesadores y los dispositivos de E/S

2. Comunicación con el procesador

- ❑ **E/S por interrupciones:** la sobrecarga que introduce el *polling* llevó al desarrollo de la E/S por interrupciones
 - ❑ Se trata de un esquema de E/S en el que se utilizan interrupciones para indicar al procesador que un dispositivo de E/S requiere su atención
 - ❑ La E/S por interrupciones se utilizan en casi todos los sistemas, al menos para algunos dispositivos
 - ❑ Cuando un dispositivo quiere notificar al procesador que ha finalizado una operación de E/S, hace que el procesador se interrumpa
- ❑ Las interrupciones son como las excepciones, con dos diferencias significativas

2. Comunicación con el procesador

- ❑ ... **1) Una interrupción es asíncrona** con respecto a la ejecución de una instrucción, no está asociada con ninguna instrucción y no puede evitar que la instrucción se complete. Antes de comenzar la ejecución de una nueva instrucción, la unidad de control comprueba si existen interrupciones de E/S pendientes
- ❑ ... **2)** Además de saber que se ha producido una interrupción, necesitamos conocer más información, como la **identidad del dispositivo que ha generado la interrupción**, así como la prioridad de las peticiones de interrupción

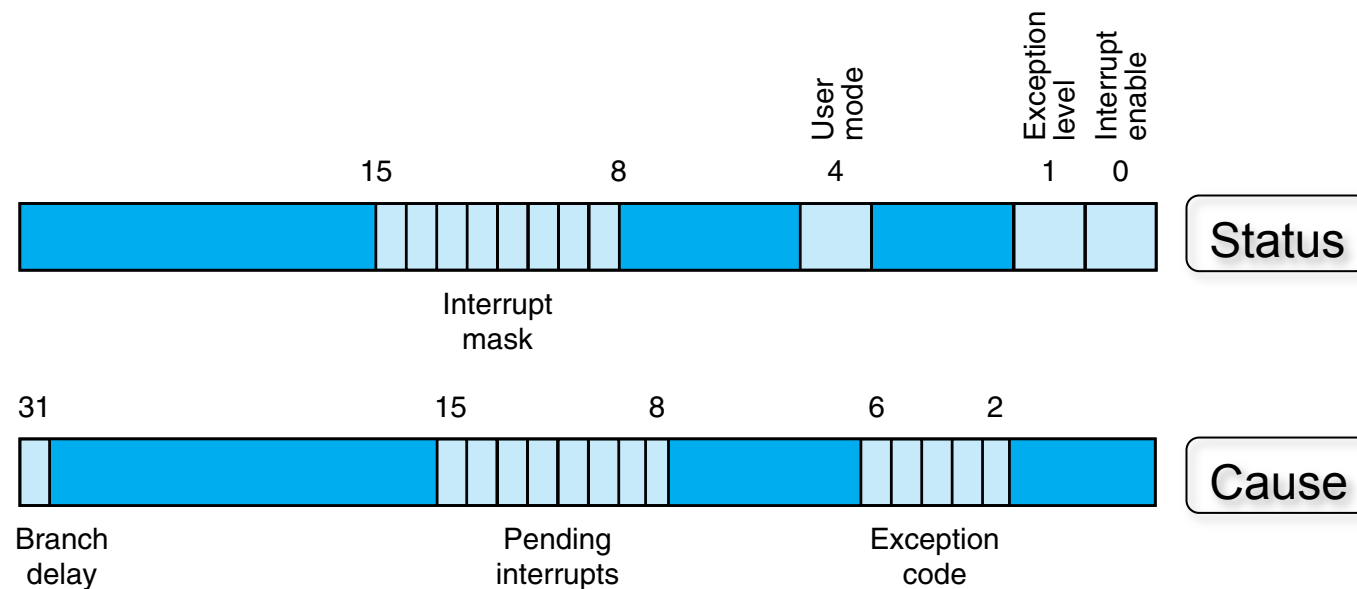
Niveles de prioridad de interrupción

- ☐ La mayoría de los mecanismos de interrupción tiene varios niveles de prioridad
 - ☐ Los SSOO Unix utilizan entre cuatro y seis niveles
- ☐ Este nivel indica el orden en que el procesador debe procesar las interrupciones
- ☐ Tanto las excepciones internas como las interrupciones externas tienen prioridades
 - ☐ Las interrupciones tienen menor prioridad que las excepciones internas
 - ☐ A los dispositivos más rápidos se les suele asociar una mayor prioridad
- ☐ Para soportar niveles de prioridad, MIPS proporciona las primitivas que permiten al SO la implementación de una política

Niveles de prioridad de interrupción

☐ Registros clave:

- ☐ **Status**: determina quién puede interrumpir. Si el bit de interrupción está a 0, nadie puede interrumpir
- ☐ **Cause**: el campo con el código de la excepción le indica al SO el motivo de la interrupción



Pasos en el manejo de una interrupción

1. AND lógico de los campos *máscara de interrupción* e *interrupciones pendientes*: *interrupt mask AND pending interrupts*
2. **Seleccionar la interrupción de mayor prioridad** de las posibles: las que están más a la izquierda tienen una prioridad mayor
3. **Salvar** la máscara de interrupciones del registro de estado
4. Cambiar la máscara de interrupciones para deshabilitar todas aquellas interrupciones con una prioridad menor o igual
5. Salvar el estado del procesador necesario para manejar la interrupción
6. Para permitir interrupciones de mayor prioridad, poner a 1 el bit de habilitación de interrupción
7. **Llamar a la correspondiente rutina de tratamiento de interrupción**
8. Antes de **restaurar** el estado, poner el bit de habilitación de interrupción a 0 para restaurar la máscara de interrupción

IPL: *interrupt priority levels*

- ☐ El IPL es un mecanismo inventado por el SO
- ☐ Forma parte de la información de un proceso
- ☐ A cada proceso del sistema se le asigna un IPL
- ☐ Con el nivel más bajo de IPL, todas las interrupciones están permitidas
- ☐ Con el nivel más alto de IPL, todas las interrupciones están bloqueadas
- ☐ Incrementar o decrementar el IPL implica modificar la máscara de interrupciones

3. Transferir datos entre los dispositivos y la memoria

- ☐ Los dos métodos que permiten a un dispositivo comunicarse con el procesador, polling y E/S por interrupciones, son la base para implementar dos mecanismo de transferencia de datos entre un dispositivo y la memoria
- ☐ Estos métodos son **apropiados para dispositivos con un ancho de banda pequeño**
- ☐ El peso de la **transferencia por *polling* y dirigida por interrupciones** recae en el procesador

3. Transferir datos entre los dispositivos y la memoria

- ☐ **Transferencia por *polling*:** el procesador carga los datos de los registros del dispositivo y los almacena en memoria
- ☐ **Transferencia por interrupciones:** el SO transfiere datos por medio de un número pequeño de bytes desde o hacia los dispositivos
 - ☐ Mientras se completan las transferencias, el SO realiza otras tareas
 - ☐ Cuando el último byte de una petición de E/S ha sido transferido y la operación de E/S ha finalizado, el SO informa al programa
- ☐ El SO y el procesador realizan todo el trabajo

3. Transferir datos entre los dispositivos y la memoria

- ☐ La E/S por interrupciones introduce mucha sobrecarga en el caso transferir datos desde o hacia un disco duro
- ☐ Para **dispositivos con un ancho de banda grande** las transferencias consisten en mover bloques grandes de datos (cientos o miles de bytes)
- ☐ Se inventó un mecanismo para que el controlador del dispositivo transfiriese datos directamente con la memoria sin involucrar al procesador: *direct memory access* (**DMA**)
- ☐ El dispositivo todavía necesita del mecanismo de interrupción, pero sólo para indicarle al procesador la finalización de la operación o en caso de error

DMA

- ☐ El mecanismo de DMA se implementa por medio de un **controlador especializado** que transfiere datos entre memoria y un dispositivo de E/S de forma **totalmente independiente al procesador**
- ☐ El controlador DMA se convierte en un **master** encargado de dirigir las lecturas o escrituras entre él mismo y la memoria

DMA: transferencia en tres pasos

1. El procesador inicializa el DMA indicando:
 - El identificador del dispositivo
 - La operación a realizar en el dispositivo
 - La dirección de memoria fuente o destino de los datos
 - El número de bytes a transferir
2. El DMA comienza la operación en el dispositivo y arbitra la interconexión:
 - Cuando el dato está disponible, lo transfiere
 - El DMA proporciona la dirección de memoria para la lectura o escritura
 - Si la petición requiere más de una transferencia, la unidad de DMA genera la siguiente dirección de memoria e inicia la siguiente transferencia
 - La unidad de DMA puede completar una transferencia entera sin necesidad de molestar al procesador
3. Una vez que la transferencia por DMA ha finalizado, el controlador interrumpe al procesador

DMA

- ☐ En un computador puede haber múltiples dispositivos DMA. Por ejemplo, en un sistema con un bus procesador-memoria y varios buses de E/S, cada controlador de bus de E/S puede contener un procesador DMA para que controle las transferencias entre un dispositivo del bus de E/S y la memoria
- ☐ A diferencia de la transferencia por *polling* o por interrupciones, DMA se puede utilizar para hacer de interfaz con un disco duro sin consumir todos los ciclos del procesador en una única operación de E/S
- ☐ En aquellas ocasiones en las que el procesador tenga que acceder a memoria, si esta está ocupada realizando una transferencia por DMA, **el procesador tendrá que esperar**
- ☐ Sin embargo, en la mayoría de los casos, **la utilización de cachés evita tener que acceder a la memoria principal**, dejando el ancho de banda de la memoria libre para ser utilizado por los dispositivos de E/S

DMA y el sistema de memoria

- ☐ Cuando se introduce DMA en un sistemas de E/S, la relación entre el procesador y el sistema de memoria cambia
- ☐ Sin DMA, todos los accesos a memoria provienen del procesador
- ☐ Con DMA, aparece un nuevo camino de acceso al sistema de memoria
 - ☐ Uno que **no pasa por el mecanismo de traducción de direcciones ni por la jerarquía de memoria caché**
 - ☐ Esto genera problemas tanto al sistema de memoria virtual como al sistema con caché
 - ☐ Estos problemas se resuelve por medio de una combinación de técnicas hardware y soporte software

DMA, ¿memoria virtual o memoria física?

- ☐ Si la unidad de **DMA utiliza direcciones físicas**, debemos restringir las transferencias por DMA dentro de una misma página
 - ☐ Si sobrepasamos el límite de una página, dichas posiciones de memoria no tienen por qué ser contiguas en la memoria virtual
- ☐ Si la unidad de **DMA utiliza direcciones virtuales**, podemos sobrepasar los límites de una página
 - ☐ En este caso, la unidad de DMA tiene un pequeño número de entradas con las traducciones de dirección virtual a física para una transferencia
 - ☐ Este conjunto de entradas se lo proporciona el SO cuando se inicia la operación de E/S
- ☐ En cualquier caso, el SO no debe cambiar la ubicación de una página mientras se esté realizando una transferencia por DMA que involucre a dicha página

DMA y jerarquía de caché

- ☐ Puede haber dos copias de un dato: una en la caché y otra en la memoria
- ☐ Como las peticiones de la unidad de DMA van directamente a la memoria, **el contenido de una posición de memoria desde el punto de vista del procesador y de la unidad de DMA pueden ser diferentes**
- ☐ Una transferencia del disco a la memoria puede provocar que algunas posiciones que están en caché queden desactualizadas. Una lectura posterior por parte del procesador de dichas ubicaciones en caché devolverá un contenido antiguo
- ☐ Del mismo modo, si la caché utiliza post-escritura, la unidad de DMA al leer directamente de la memoria obtendría un valor desactualizado
- ☐ A estos problemas se los denomina **problemas de coherencia**

DMA y jerarquía de caché: soluciones

1. Hacer que todo el tráfico de E/S pase por la caché
 - Las lecturas verán el valor más actualizado
 - Las escrituras actualizarán el dato en la caché
 - Caro y puede tener un impacto negativo en el rendimiento del procesador
2. Invalidaciones selectivas de caché por parte del SO para una lectura de E/S o forzar que la post-escritura se realice en una escritura de E/S, cache *flushing*
 - Requiere menos soporte hardware y puede ser más eficiente si el software hace su trabajo de forma sencilla y eficiente
3. Proporcionar un mecanismo hardware qué, de forma selectiva, invalide o vuelque entradas de caché
 - La invalidación por hardware para asegurar la coherencia de la caché es típica en sistemas multiprocesador
 - Esta misma técnica se puede utilizar para el sistema de E/S