

Tema 7: Gestión de la entrada/salida

Mayo de 2011

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 Implementación de la E/S
 - Concepto de bus
 - Elementos de diseño de un bus
 - Parámetros de los buses
 - Protocolos de acceso al bus
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

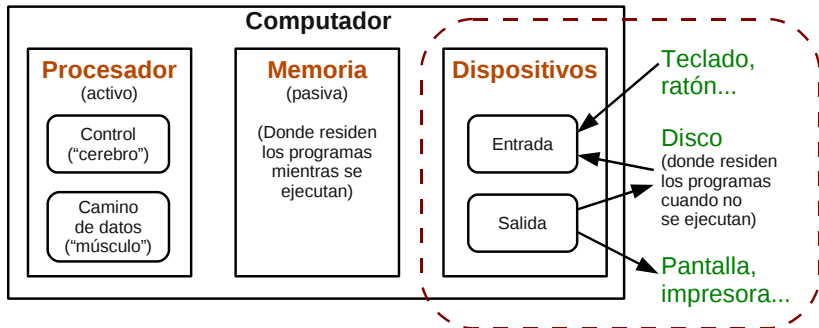
Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 Implementación de la E/S
 - Concepto de bus
 - Elementos de diseño de un bus
 - Parámetros de los buses
 - Protocolos de acceso al bus
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

Introducción

- Hasta ahora nos habíamos centrado en sólo dos componentes de un ordenador:
 - Procesador.
 - Sistema de memoria.
- El **sistema de E/S** proporciona información a la CPU y le permite comunicarse con el exterior y otros dispositivos.
- Objetivos del tema:
 - Clasificación de los diferentes periféricos de E/S.
 - Programación de los dispositivos de E/S.
 - El papel del S.O.
 - Implementación de la E/S: buses y protocolos de acceso.
 - Visión general de algunos buses estándares.

Esquema de Von Neumann de un ordenador



Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 Implementación de la E/S
 - Concepto de bus
 - Elementos de diseño de un bus
 - Parámetros de los buses
 - Protocolos de acceso al bus
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

Clasificación de los dispositivos de E/S

Según la funcionalidad a la que se destinen:

- **Almacenamiento:** discos duros, diversos tipos de disquetes (unidades ZIP, etc.), memorias flash, CDROMs, DVDs, grabadoras de CD/DVD, unidades de cinta, etc.
- **Interfaz con el usuario:** teclados, ratones, tabletas gráficas, joysticks, etc.
- **Visualización y multimedia:** tarjetas gráficas, monitores, impresoras, tarjetas de sonido, altavoces, etc.
- **Comunicaciones:** tarjetas de red (Ethernet, inalámbricas), módems, etc.
- **Adquisición de datos:** cámaras de vídeo, micrófono+tarjeta de sonido, etc.

Clasificación de los dispositivos de E/S

Según el ancho de banda (bytes/segundo)

- Gran diversidad (desde unos pocos bytes a MB o incluso GB).

Dispositivo	Tipo	Interacción	Ancho de banda (KB/s)
Teclado	Entrada	Humana	0,01
Ratón	Entrada	Humana	0,02
Impresora de líneas	Salida	Humana	1,00
Impresora láser	Salida	Humana	100,00
Memoria flash	Almacenamiento	Máquina	20.000,00
Disco duro	Almacenamiento	Máquina	60.000,00
Tarjeta de red	Entrada/Salida	Máquina	125.000,00
Adaptador gráfico	Salida	Humana	3.000.000,00

Clasificación de los dispositivos de E/S

- Los dispositivos de E/S mencionados son muy diferentes entre sí:
 - Tienen usos y modos de funcionamiento diversos.
 - Muchas de sus características vienen determinadas por la tecnología.
- Sin embargo, pueden estudiarse de una forma más homogénea en función de:
 - La **programación** del dispositivo de E/S:
 - Sondeo, interrupciones, DMA, ...
 - La **implementación** de su comunicación con la CPU+memoria:
 - Buses, protocolos de acceso, arbitraje, ...

Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 Implementación de la E/S
 - Concepto de bus
 - Elementos de diseño de un bus
 - Parámetros de los buses
 - Protocolos de acceso al bus
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

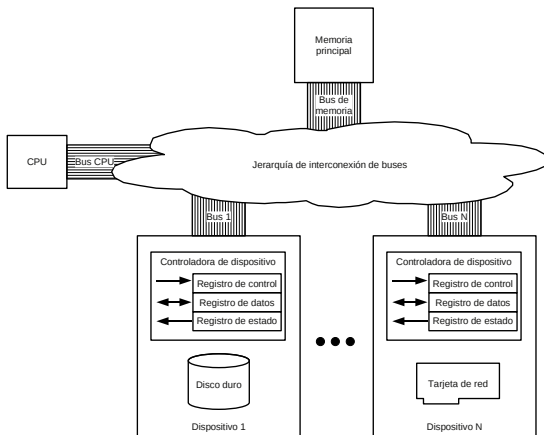
Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 Implementación de la E/S
 - Concepto de bus
 - Elementos de diseño de un bus
 - Parámetros de los buses
 - Protocolos de acceso al bus
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

Puertos, controladoras y canales

- Diversidad de dispositivos de E/S, pero ... comunicación y programación siguen ciertas reglas comunes.
- **Puertos de E/S:** registros (externos a la CPU) a través de los que se comunican la CPU y los dispositivos. En particular:
 - **Registro de datos:** lectura/escritura del dato a comunicar entre CPU y dispositivo.
 - **Registro de control:** en él la CPU escribe las órdenes a realizar.
 - **Registro de estado:** información de listo/no listo, estado, etc.
 - Integrados en un chip (llamado **controladora** del dispositivo).
- **Canales (= buses)**
 - Líneas de comunicación entre distintos componentes (CPU – memoria – dispositivos lentos – dispositivos rápidos, etc.)
 - Como veremos después, se organizan jerárquicamente.

Puertos, controladoras y canales



Esquema de comunicación de la CPU con la E/S a través de los puertos.

Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 Implementación de la E/S
 - Concepto de bus
 - Elementos de diseño de un bus
 - Parámetros de los buses
 - Protocolos de acceso al bus
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

Programación de la entrada/salida

- Estudiaremos primero la E/S desde el punto de vista del programador:
 - Modos de **direccionamiento y acceso** a los puertos de los dispositivos:
 - Espacio de direcciones mapeado en memoria normal, o bien,
 - Espacio de direcciones de E/S aislado.
 - **Instrucciones** utilizadas.
 - Mecanismos de **interacción CPU–Dispositivos**:
 - **Sondeo** (= encuesta = escrutinio = *polling*).
 - **Interrupciones**.
 - **Interrupciones + DMA** (acceso directo a memoria).

E/S mapeada en memoria

- También conocida como E/S de correspondencia directa.
- Parte del espacio de direcciones se asocia a los dispositivos de E/S (son direcciones normales, tipo RAM), pero ...
- ... una operación de L/E en una posición de memoria reservada a los dispositivos es ignorada por la memoria, siendo **capturada por la controladora de dispositivo** correspondiente.
- La L/E de un dato requiere **varias operaciones separadas** de E/S en los distintos puertos del dispositivo. Por ejemplo:
 - 1 Escribir código de orden en **registro de control**.
 - 2 Leer estado del dispositivo en **registro de estado**.
 - 3 Leer/escribir dato en **registro de datos**.
- **Ventaja:** gran cantidad de instrucciones disponibles (todas las de acceso a memoria normales).
- **Inconveniente:** se utiliza parte del ¿valioso? espacio de direcciones.

E/S aislada

- Espacio de direcciones dedicado a la E/S.
- Cada dirección especial especifica tanto el dispositivo como el puerto.
- Son necesarias **instrucciones especiales** (no valen las de carga y almacenamiento normales).
 - **Ejemplo:** procesadores de la familia Intel 8086:
 - IN reg, dir_puerto
 - OUT dir_puerto, reg
- **Ventajas:** no se utiliza una parte de la memoria normal.
- **Inconvenientes:** menor repertorio de instrucciones disponibles.

Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 Implementación de la E/S
 - Concepto de bus
 - Elementos de diseño de un bus
 - Parámetros de los buses
 - Protocolos de acceso al bus
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

Sondeo (*polling*)

- También llamado encuesta o escrutinio.
- El procesador es el que realiza todo el trabajo, “**sondeando**” **continua** o **periódicamente** al puerto correspondiente.
- El dispositivo informa de su situación modificando su **registro de estado**.
- El procesador lee dicha palabra de estado y obra en consecuencia.
- **Problema:** la CPU sondea muchas veces el dispositivo comprobando que no ha cambiado.
 - E/S mucho más lenta que CPU → **Gran pérdida de tiempo**

Sondeo (*polling*)

- **Ejemplo:** ratón; el procesador verifica periódicamente el estado del ratón, para ver si el usuario lo ha movido, en cuyo caso el SO informa al programa asociado del cambio de situación.
- Normalmente, bucles de espera activa (**encuesta continua**) sólo permisibles en dispositivos dedicados (sist. empotrados).
- Para computadores normales (SO de tiempo compartido), más habitual la **encuesta periódica** (p.e., para el ratón).

Rendimiento del sondeo

- Supongamos que en una CPU a 500 MHz una operación de sondeo consume 400 ciclos de reloj (llamar a la rutina de sondeo, acceder al dispositivo y volver). Determinar el **% de tiempo que la CPU gasta realizando sondeos** para los siguientes dispositivos:
 - **Ratón:** debe escrutarse 30 veces/seg para no perder el movimiento del usuario.
 - **Disquete:** transfiere datos en unidades de 2 bytes y tiene un ancho de banda de 50 KB/seg. No pueden perderse datos.
 - **Disco duro:** transfiere datos en unidades de 16 bytes y su ancho de banda es de 8 MB/seg. Tampoco pueden perderse datos.

Rendimiento del sondeo

- Ciclos por segundo consumidos en el sondeo del **ratón**:
 - $30 * 400 = 12.000$ ciclos/seg consumidos \Rightarrow
 - % CPU: $12.000 / (500 * 10^6) * 100 = 0,0024$ % \Rightarrow **impacto despreciable.**
- Ciclos por segundo consumidos en el sondeo del **disquete**:
 - $50 \text{ KB/s} / 2 \text{ bytes por transferencia} = 25.600$ encuestas/seg \Rightarrow
 - $25.600 * 400 = 10.240.000$ ciclos/seg consumidos \Rightarrow
 - % CPU: $10.240.000 / (500 * 10^6) * 100 = 2,048$ % \rightarrow **impacto apreciable, pero permisible** si no hay muchos dispositivos.
- Ciclos por segundo consumidos en el sondeo del **disco duro**:
 - $8 \text{ MB/s} / 16 \text{ bytes por transferencia} = 524.288$ encuestas/seg \Rightarrow
 - $524.288 * 400 = 209.715.200$ ciclos/seg consumidos \Rightarrow
 - % CPU: $209.715.200 / (500 * 10^6) * 100 = 41,95$ % \Rightarrow **impacto inaceptable**

Interrupciones

- Interrupciones vs. sondeo (analogía secretaria):
 1. Encargamos a nuestra secretaria un trabajo (= **orden de transacción de E/S**).
 2. Para saber cuándo ha terminado su trabajo, tenemos dos opciones:
 - 2a. Nos quedamos mirándola continuamente, esperando a que termine (= **sondeo continuo**), o bien hacemos otra cosa, pero cada X minutos nos acercamos a preguntarle si ha terminado (= **sondeo periódico**).
 - 2b. O bien, continuamos haciendo otras cosas, sin preocuparnos ya de vigilar, y pedimos a la secretaria que nos avise (p.e., llamando por teléfono) cuando haya terminado el trabajo (= **interrupción externa a la CPU**).
 3. En ambos casos, una vez detectado que la subtarea ha sido terminada, utilizarla adecuadamente (= **tratamiento de los datos**: lect./escrib. de los mismos desde/a buffers adecuados, etc.)

Interrupciones

- En el sondeo, el tratamiento de los datos lo hacen las instrucciones que siguen al bucle de espera activa, pero...
- ...una interrupción puede llegar en cualquier momento y la CPU estará haciendo otra cosa (son **asíncronas** respecto al programa en ejecución).
- Por tanto, hay que **habilitar mecanismos** para que:
 - 1 La CPU deje automáticamente lo que esté haciendo y pase a ejecutar la rutina de tratamiento (**RSI o rutina de servicio de interrupción**; también llamada manejador).
 - 2 Se **salve la información** mínima para que luego pueda recuperarse el estado de ejecución en el que estaba el programa justo en el momento en que llegó la interrupción.
 - 3 Al finalizar la RSI, se vuelva justo a dicho estado y se **reanude el proceso interrumpido** con normalidad.

Interrupciones

- Ventaja principal de interrupciones frente al sondeo:
 - La CPU no gasta ciclos útiles en comprobar si el dispositivo está preparado.
 - Simplemente, cuando éste lo esté, y lo indique mediante la interrupción correspondiente, se ejecutará la RSI asociada.
- Cada dispositivo tendrá su propio **número de interrupción asociada**:
 - P.ej., un pin de entrada de interrupción en la CPU por cada dispositivo (MIPS), o bien
 - Un solo pin de interrupción (igual para todos los dispositivos), y después de activarla el dispositivo pone su código (número de interrupción) en el bus de datos de la CPU (Intel IA-32).

Interrupciones

- Para los mecanismos anteriores (salto a la RSI + salvar estado de la CPU + reanudar proceso interrumpido) hay que resolver algunas cuestiones:
- **A. Salto a la RSI:**
 - ¿Dónde saltar? Dos opciones:
 - Siempre a un mismo lugar del código (**dirección física fija**) al producirse la interrupción, en la cual se mira qué dispositivo la provocó y entonces se bifurca al tratamiento adecuado.
MIPS salta a la dirección 0x80000180.
 - A una dirección variable, dependiendo del número de interrupción producida (**interrupciones vectorizadas**).
Intel IA-32 busca la dirección correspondiente en un array de direcciones indexadas por dicho número¹.

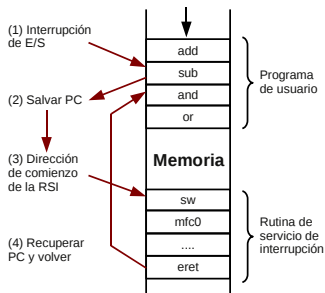
Interrupciones

● B. Salvar estado de la CPU:

- Mínimo, hay que salvar siempre el valor del PC: en MIPS, hay un registro a tal efecto, el **EPC** (exception program counter).
- En máquinas que dependen de un **registro de estado para las condiciones** (p.e., flags en Intel IA-32), salvarlo también.
- Por supuesto, todos aquellos **registros modificados por la RSI** deben ser salvados, pero esto lo hace el propio software de la misma (análogo a como se hace en una rutina normal con los registros preservados entre llamadas, pero para todos los registros).
- En realidad, hay dos registros que en MIPS no se salvan: \$k0 y \$k1 (**registros reservados para el kernel**; es lo mínimo necesario para poner en marcha la RSI, como veremos) → ¡El programador de aplicaciones no debe usarlos!

Interrupciones

- **C. Recuperar estado y reanudar proceso interrumpido:**
 - Se recuperan los registros (que la RSI guardó antes de modificar).
 - Se vuelve a escribir el PC con el **EPC** guardado, lo cual provoca la reanudación del proceso.
- Resumen del acceso completo:



Interrupciones

- **Rendimiento (consumo de CPU con interrupciones):**
 - Suponemos un gasto de 500 ciclos por cada transferencia (tiempo de ejecución de la RSI). También que el disco duro se usa durante un 5 % del tiempo. Misma CPU y HD que antes.
 - Interrupciones de disco duro/seg:
 $8 \text{ MB/s} / 16 \text{ bytes por transf.} = 524.288 \text{ interrupciones/seg.}$
 - Ciclos/seg: $524.288 * 500 = 262.144.000 \text{ ciclos/seg.}$
 - % CPU (durante las transferencias): $262.144.000 / (500 * 10^6) * 100 = 52,4 \%$.
 - % CPU (total; sólo 5 % tiempo activo): $52,4 \% * 5 \% = \mathbf{2,62 \%}$.
 - **Nota:** en el ejemplo del sondeo suponíamos que estábamos sondeando siempre (no sólo el 5 % del tiempo):
 - Quizá no muy realista para el disco duro (ya que sabemos cuándo hay peticiones pendientes), pero ...
 - ... p.e. tarjeta de red → pueden llegar datos en cualquier momento → habría que estar sondeando siempre.

Interrupciones

- Una **interrupción de E/S** es como una **excepción** (necesita rutina de tratamiento, capacidad de recuperación, etc.), pero:
 - La interrupción de E/S es “**asíncrona**” respecto al programa (a priori, no se sabe en qué momento puede producirse; no se haya asociada a ningún tipo de instrucción concreta).
 - Necesitamos saber **qué dispositivo la provocó**.
- **Terminología** (muy variable en la literatura):
 - **Excepción**^a: una instrucción provoca una situación anormal o no permitida durante su ejecución (ej., *violación de acceso*, *overflow*, etc.).
 - **Interrupción**: un dispositivo externo a la CPU señala a la CPU que necesita su atención.
 - **Trap**: es una instrucción especial que el programador puede usar para provocar una llamada a una rutina de servicio (llamada al sistema operativo).

^aA veces, por abuso del lenguaje, se usa “excepción” como un término genérico para los 3 casos.

Interrupciones y DMA

- El sondeo y las interrupciones son adecuados para dispositivos con **escaso ancho de banda**: se reduce el coste de la controladora y de la interfaz, ya que CPU y SO realizan casi todo el trabajo:
 - **Cada transferencia de una palabra** a/desde dispositivo → una secuencia de instrucciones de CPU para transferirla desde/a memoria (mediante rutina de sondeo o mediante RSI).
- Con dispositivos de gran ancho de banda las transferencias constan principalmente de grandes bloques de datos:
 - Sondeo o interrupciones solamente no son métodos adecuados, pues se mantendría ocupado el procesador demasiado tiempo.
- **Acceso directo a memoria (DMA)**: mecanismo para conseguir que la transferencia de datos desde el dispositivo a la memoria o viceversa se realice sin la intervención del procesador.

Interrupciones y DMA

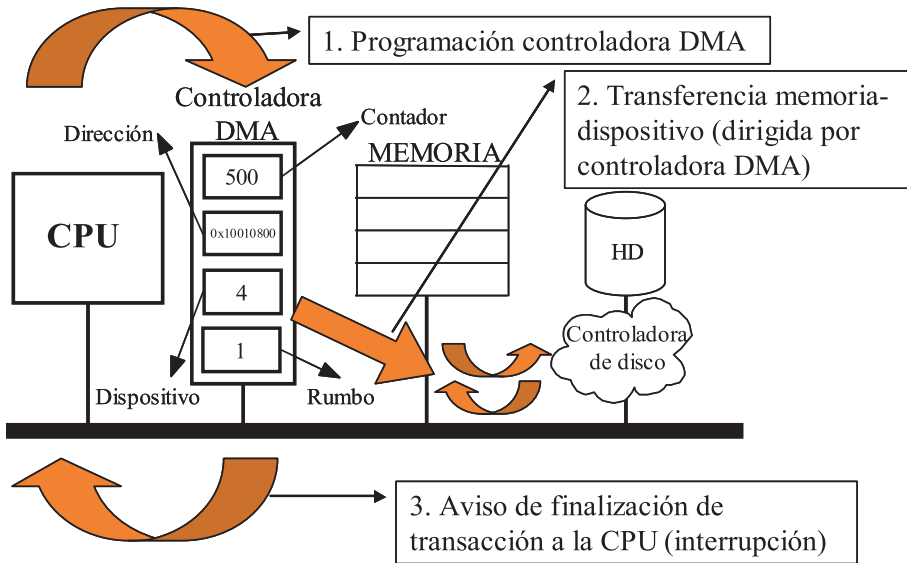
- El DMA se implementa con un chip especializado (**controladora de DMA**), que transfiere datos entre un dispositivo de E/S y memoria independientemente del procesador.
- Tendrá **implicaciones en el hardware**, como veremos:
 - Bus: múltiples amos (CPU y DMA) → sistema de arbitraje.
- **Pasos en una transferencia DMA:**
 - 1 El procesador inicializa la controladora de DMA: identidad del dispositivo, operación a realizar, dirección de memoria donde leer o escribir, número de bytes a transferir y sentido del desplazamiento.
 - 2 La controladora de DMA va pidiendo el bus y, cuando lo consigue, va realizando tantas operaciones de transferencia entre el dispositivo y memoria como sean necesarias.
 - 3 Finalmente, la controladora de DMA genera una interrupción informando al procesador de la finalización de la transferencia o de una condición de error.

Interrupciones y DMA

- **Analogía jefe–secretaria:**

- 1 Encargamos un trabajo bastante grande a nuestra secretaria (p.e., varios informes) (= **programación de la controladora DMA**).
- 2 Dejamos que haga todos los informes en su despacho, sin interrumpirnos cada vez que termine uno de ellos para enseñárnoslo. En vez de ello, cada vez que termina uno, lo va acumulando en una esquina de nuestra mesa, sin interrumpirnos (= **acceso directo a memoria por parte de la controladora**).
- 3 Una vez que termina de realizar y colocar el último informe, nos avisa de que ya lo tenemos todo en nuestra mesa, listo para ser utilizado (= **interrupción de notificación de finalización**).

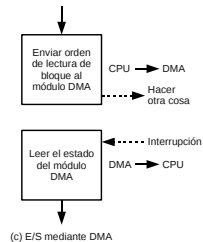
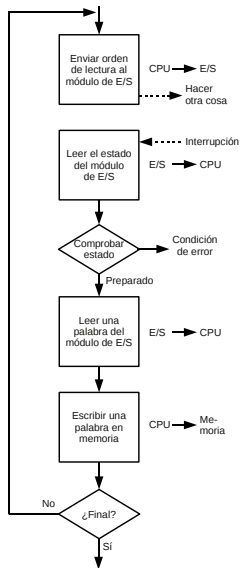
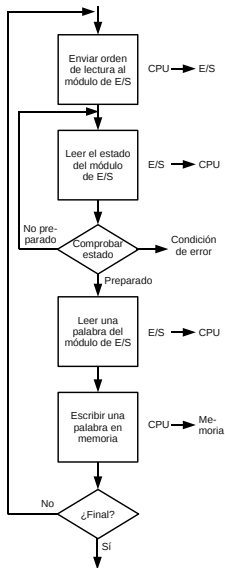
Secuencia de acciones (DMA)



Rendimiento del DMA

- Supongamos misma CPU y disco duro de los ejemplos anteriores.
- Supongamos que el proceso de inicio de la controladora de DMA (programación de los puertos correspondientes) tarda 1.000 ciclos y que la RSI de tratamiento una vez finalizada la transferencia tarda 500.
- Además, supondremos que cada transferencia DMA es de 4KB.
- Casos:
 - Si el disco estuviese continuamente transmitiendo: 8 MB por seg. / 4KB por transf. = 2048 transf. DMA por segundo.
Ciclos CPU/seg: $(1.000+500) * 2048 = 3.072.000$ ciclos/seg.
% CPU (durante las transferencias): $3.072.000 / 500 * 10^6 = \mathbf{0,62\%}$.
 - Como sólo transmite el 5% del tiempo: % CPU (total; sólo 5% tiempo activo): $0,62 * 5\% = \mathbf{0,031\%} \rightarrow$ **prácticamente despreciable**.
- **Nota:** el ejemplo supone que el DMA no interfiere con la CPU, lo cual puede no ser muy realista, pues los accesos a memoria de la CPU pueden colisionar con los del DMA.

Resumen (sondeo, interrupciones y DMA)



Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 **El papel del sistema operativo**
- 5 Implementación de la E/S
 - Concepto de bus
 - Elementos de diseño de un bus
 - Parámetros de los buses
 - Protocolos de acceso al bus
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

El papel del sistema operativo

- Control de dispositivos de E/S heterogéneo y complicado:
 - ¿Quién mantiene el control del estado de los dispositivos, conoce las direcciones de los puertos para leer/escribir los datos y las órdenes asociadas a cada dispositivo, maneja los errores, etc.?
- Respuesta: el **sistema operativo (SO)**.
 - El código del SO es el **primero que se instala** al arrancar el ordenador.
 - En particular, instala las rutinas de petición de E/S de los distintos dispositivos y las posibles RSI asociadas (**drivers**) → Sólo el SO conoce los puertos, órdenes, etc.
 - El usuario solicita sus servicios a través de mecanismos sencillos con parámetros bien definidos (**llamadas al sistema**: `syscall`).
 - Además, por seguridad, estas llamadas preestablecidas serán la única forma de acceder a la E/S (los usuarios no podrán utilizar directamente los puertos) → Necesidad de **soporte en CPU para seguridad**.

Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 Implementación de la E/S
 - Concepto de bus
 - Elementos de diseño de un bus
 - Parámetros de los buses
 - Protocolos de acceso al bus
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 **Implementación de la E/S**
 - **Concepto de bus**
 - Elementos de diseño de un bus
 - Parámetros de los buses
 - Protocolos de acceso al bus
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

Concepto de bus

- Los diferentes subsistemas que componen un computador (procesador, memoria y dispositivos de E/S) necesitan comunicarse entre sí \Rightarrow **Bus** o **canal**.
- Bus: canal de comunicación compartido que usa un conjunto de líneas (cables o hilos) para conectar múltiples subsistemas.
 - Cualquier señal transmitida está disponible para los dispositivos conectados al bus.
- Aspectos claves del diseño de un bus:
 - Líneas del bus (de datos, control y direcciones).
 - Anchura, longitud y frecuencia del bus \rightarrow ancho de banda.
 - Temporización: síncrona y asíncrona.
 - Control de acceso al bus: arbitraje.

Jerarquía de buses múltiples

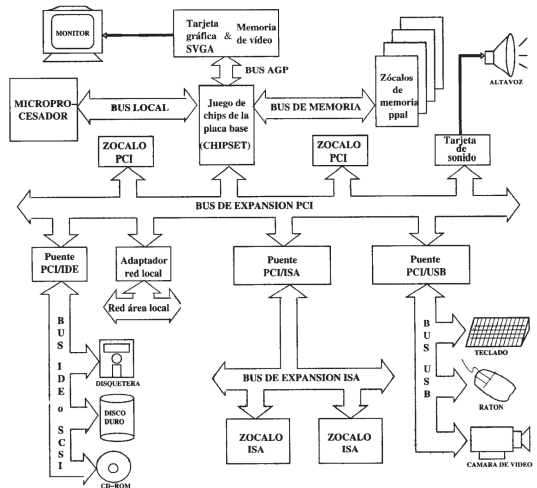
- Gran número de dispositivos conectados al bus:
 - Aumenta el retardo de propagación debido al tiempo de coordinación de los diferentes dispositivos para el uso del bus.
 - Cuello de botella: los dispositivos están continuamente esperando que sea su turno → **congestión**.
- **Posible solución:** aumentar el ancho de banda del bus.
- **Solución más efectiva:**
 - Utilizar varios buses: para evitar que interfieran diferentes subsistemas.
 - Organizarlos de forma **jerárquica**: buses con diferentes velocidades.

Los dispositivos de E/S más exigentes se acercan al procesador para evitar cuellos de botella.

Jerarquía de buses múltiples

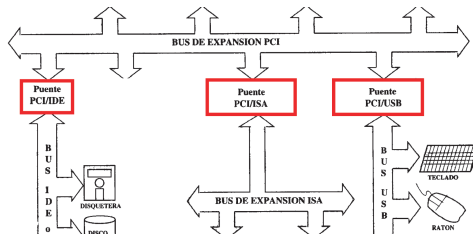
Clasificación:

- 1 Bus local y de memoria.
 - Conectan CPU y memoria.
 - Son los de mayor rendimiento.
- 2 Buses de expansión.
 - PCI, ISA.
- 3 Buses dedicados.
 - De propósito específico.
 - IDE/SCSI, AGP, firewire, USB, ...



Puentes de conexión al bus

- Los **puentes** de conexión al bus conectan dos buses diferentes y son capaces de interpretar el contenido de los paquetes que reciben.



- Funciones:**

- Traducción:** conecta buses diferentes y traduce paquetes. Ej: PCI \Leftrightarrow IDE.
- Gestión del tráfico:** decide si trasmite un paquete por los otros extremos en función de la dirección destino.

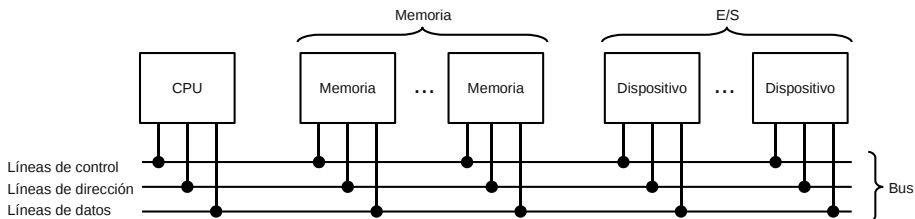
Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 **Implementación de la E/S**
 - Concepto de bus
 - **Elementos de diseño de un bus**
 - Parámetros de los buses
 - Protocolos de acceso al bus
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

Elementos de diseño de un bus

- Ya vimos que un bus es un **canal de comunicación compartido** que conecta a los componentes.
- Compartido → las comunicaciones son escuchadas por todos los dispositivos conectados en el bus.
- Organiza la información comunicada en 3 grupos:
 - Información a ser comunicada:
 - **Direcciones.**
 - **Datos.**
 - Todo lo demás:
 - **Control.**

Elementos de diseño de un bus



Esquema de interconexión de un bus.

Elementos de diseño de un bus

- Puesto que las líneas de información son compartidas por todos los componentes ...
- ... debemos diferenciar un conjunto de **líneas de control** y un conjunto de **líneas de información**.
- **Líneas de control:**
 - Gestionan el acceso y uso de las líneas de información.
 - Determinan el tipo de información en las líneas de información.
 - Transmiten ordenes e información de temporización, que especifican las operaciones a realizar.

Ejemplos de líneas de control

Línea	Efecto que produce su activación
Escritura en memoria	Hace que el dato del bus se escriba en la posición direccionada
Lectura de memoria	Hace que el dato de la posición direccionada se sitúe en el bus
Escritura de E/S	Hace que el dato del bus se transfiera al puerto de E/S direccionado
Lectura de E/S	Hace que el dato del puerto de E/S direccionado se sitúe en el bus
Petición de bus	Indica que un módulo necesita disponer del control del bus
Concesión de bus	Indica que se cede el control del bus a un módulo que lo solicitó
Petición de interrupción (INTR)	Indica que hay una interrupción pendiente
Interrupción reconocida (INTA)	Indica que la interrupción pendiente se ha reconocido
Reloj	Se utiliza para sincronizar las operaciones

Líneas de información

- **Líneas de datos:**

- Transmiten datos entre los módulos.
- Constituyen el bus de datos.
- Anchura del bus de datos: número de líneas → Prestaciones.
- Ejemplo: anchura de 8 bits e instrucciones de 16 bits (2 accesos a memoria por cada ciclo de instrucción).

- **Líneas de dirección:**

- Determinan la fuente o el destino del dato (dirección).
- Constituyen el bus de direcciones.
- Anchura del bus direcciones → Posiciones de memoria direccionables.

Líneas de información: alternativas de diseño

- **Multiplexadas:** información para direcciones y datos se transmite a través de las mismas líneas → Desempeñan varias funciones por lo que se necesita una Línea de Control de Dirección Válida (LCDV).
 - 1 Se lleva la dirección al bus. Activa LCDV.
 - 2 Período de tiempo: cada módulo copia la dirección y determina si es el direccionado.
 - 3 Se quita del bus la dirección y se realizan las mismas operaciones para transmitir los datos. Desactiva LCDV.



- **Ventajas:** ahorro de espacio y bajo coste del bus.
- **Desventajas:** circuitería más compleja en cada módulo y reducción de prestaciones.

Líneas de información: alternativas de diseño

- **Dedicadas:** cada grupo de líneas tiene una sola función.



- **Ventajas:** se mejoran las prestaciones del bus.
- **Desventajas:** se necesita un bus con un número mayor de líneas por lo que se incrementa el coste.

Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 **Implementación de la E/S**
 - Concepto de bus
 - Elementos de diseño de un bus
 - **Parámetros de los buses**
 - Protocolos de acceso al bus
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

Parámetros de los buses

Anchura del bus: se define como el número de líneas de información de las que consta dicho bus. Podemos diferenciar buses serie (USB, Firewire) y paralelo (PCI, bus del sistema).

Longitud del bus: puede oscilar desde menos de un metro (bus del sistema, PCI), varios metros (bus SCSI) e incluso cientos de metros (Ethernet). A mayor longitud, menor frecuencia de funcionamiento y menor anchura del bus.

Frecuencia de funcionamiento (para buses síncronos): frecuencia de la señal de reloj que rige las transferencias.

Parámetros de los buses

Ancho de banda teórico: cantidad de información que somos capaces de transmitir. Se suele expresar en MB/s.

$AB = \frac{n}{8} \times f$, donde n es el nº de líneas de datos.

Ancho de banda efectivo: cantidad de información que somos realmente capaces de transmitir. En este caso se tiene en cuenta que para transmitir datos necesitamos utilizar ciclos de bus en el protocolo de acceso y en el arbitraje. Más adelante veremos la diferencia con un ejemplo.

Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 **Implementación de la E/S**
 - Concepto de bus
 - Elementos de diseño de un bus
 - Parámetros de los buses
 - **Protocolos de acceso al bus**
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

Protocolos de acceso al bus

- Señalamiento y secuenciación que permite la interacciones entre los componentes. Ejemplo: lectura de memoria:
 - El procesador pone una dirección válida en el bus y activa la señal de control de “leer de memoria”.
 - La memoria recibe la señal de lectura, lee la dirección, obtiene el dato apropiado, pone el dato en el bus.
 - El procesador espera; después, lee el dato del bus.
- Pueden ser:
 - **Síncronos**: sincronizados por un reloj; organizan el protocolo a través de “pulsos” de reloj.
 - **Asíncronos**: sin un reloj común que marque el ritmo.

Temporización síncrona

- **Ventajas:**

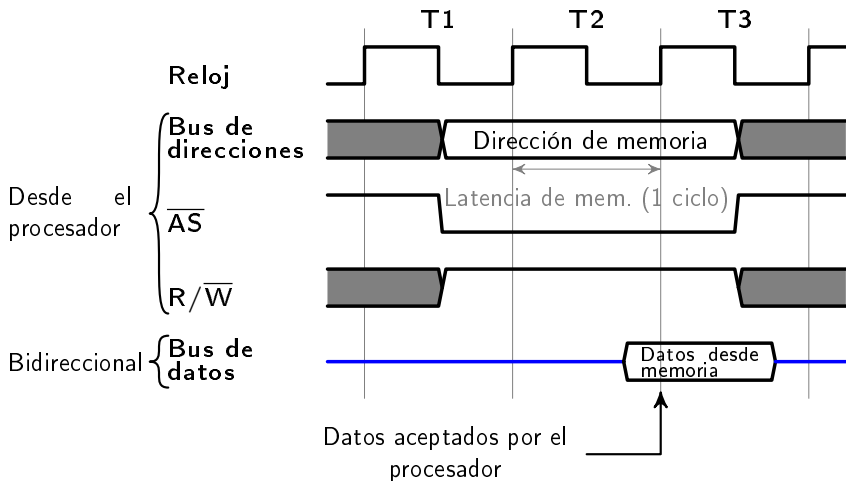
- Protocolo sencillo.
- Interfaz lógica simple.
- Bus muy rápido.

- **Inconvenientes:**

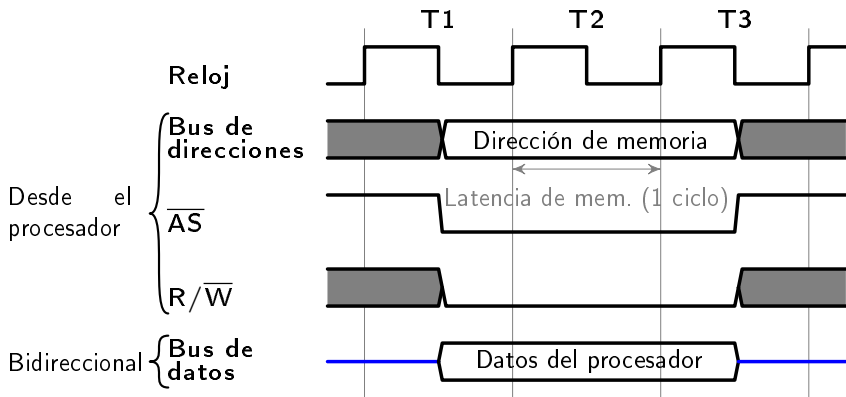
- Los dispositivos conectados deben funcionar a la misma frecuencia de reloj.
- Problema de sesgo de reloj: diferencia de tiempo absoluto desde que dos elementos ven un flanco de reloj → **desincronización interna.**
 - Buses síncronos no pueden ser largos y rápidos simultáneamente.

- Ejemplo: bus **procesador–memoria.**

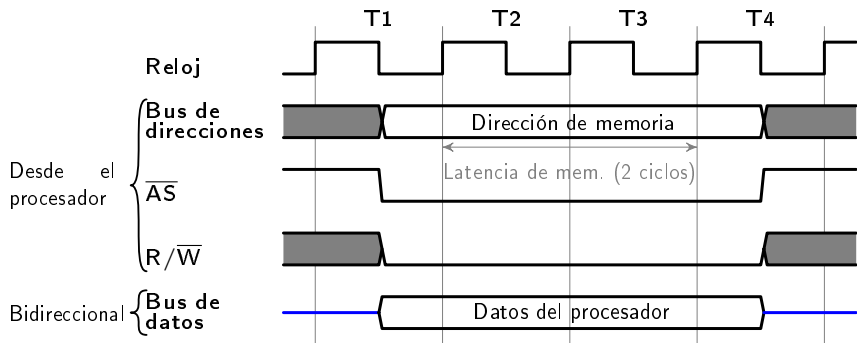
Ejemplo de transferencia síncrona: lectura



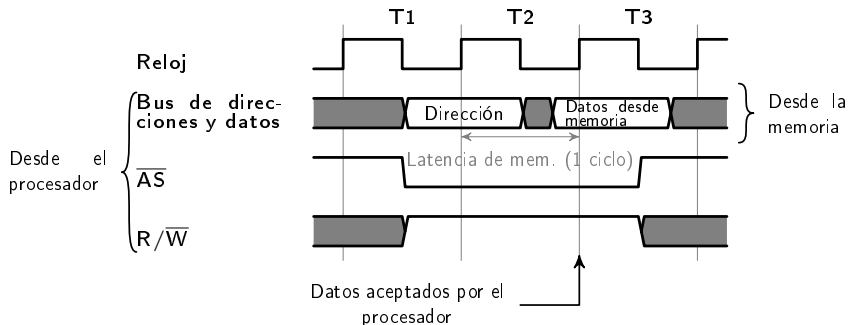
Ejemplo de transferencia síncrona: escritura



Transferencia síncrona: escritura (2 ciclos)

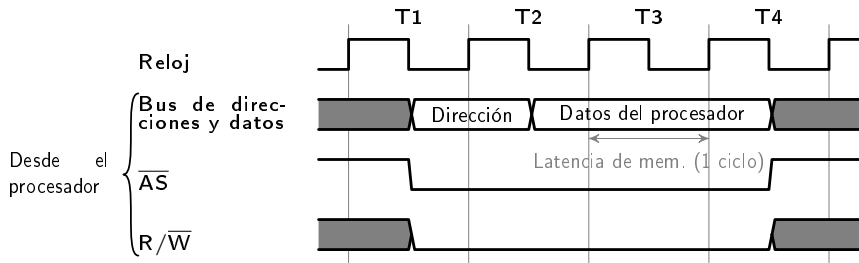


Transferencia sínc. en bus multiplexado: lectura



En una lectura no hay diferencia en el número de ciclos entre un bus multiplexado y otro que no lo es.

Transferencia sínc. en bus multiplexado: escritura



En una escritura se pierde un ciclo en el bus multiplexado (puesto que no puede ponerse en el mismo ciclo la dirección y el dato).

Ejemplo de cálculo del AB teórico/efectivo (1/3)

- Sea un bus con separación de datos de 32 bits y direcciones de 32 bits, con una temporización como la vista:
 - 1^{er} ciclo se pone la dirección (y el dato si es una escritura).
 - 2^o y 3^{er} ciclos: ciclos de espera (latencia de la memoria de 2 ciclos).
 - 4^o ciclo: se realiza la lectura (escritura efectiva).

Además, la memoria soporta el modo ráfaga (se puede seguir leyendo/escribiendo hasta un máximo de 4 palabras de 32 bits) sin tener que iniciar otra operación.

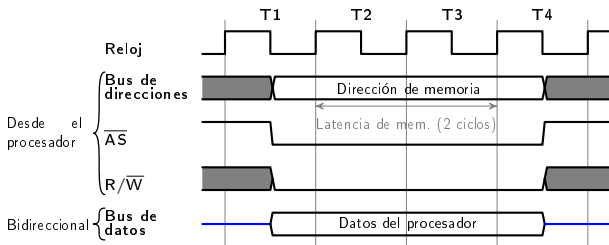
- ¿Cual sería el ancho de banda teórico del bus si funciona a 200 MHz?

En un ciclo se puede transmitir 32 bits de datos = **4 bytes**. Si $f=200$ Mhz, entonces en 1 segundo tengo 200×10^6 ciclos $\Rightarrow AB \approx 4 \times 200$ MB/s = **800 MB/s**.

- ¿Y el ancho de banda efectivo si no consideramos el modo ráfaga?

Ejemplo de cálculo del AB teórico/efectivo (2/3)

- Si no consideramos el modo ráfaga, una operación en el bus (lectura o escritura) necesita cuatro ciclos:

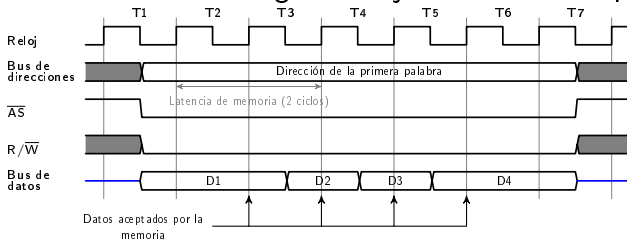


Por lo tanto, transmitimos 4 bytes/4 ciclos. Multiplicando por f podemos expresar esta cantidad en MB/s. Es decir, el ancho de banda efectivo es:

$$AB = 4/3 \times 200 \times 10^6 \text{ B/s} = 200 \text{ MB/s.}$$

Ejemplo de cálculo del AB teórico/efectivo (3/3)

- Si consideramos el modo ráfaga, la mejor situación es apurarla al máximo:



Por lo tanto, transmitimos 4×4 bytes / $(4 + 3)$ ciclos.

Multiplicando por f podemos expresar esta cantidad en MB/s.

Es decir, el ancho de banda efectivo del bus es:

$$AB = 16/7 \times 200 \times 10^6 \text{ B/s} \approx 457,14 \text{ MB/s.}$$

Que no llega al ancho de banda teórico pero que es más del doble del valor obtenido en el apartado anterior.

Temporización asíncrona

- No hay ninguna línea de reloj.
- Coordinación de la transmisión → **Protocolo de presentación:**
 - Consta de una serie de pasos, de forma que tanto emisor como receptor sólo pasan al paso siguiente cuando ambos están de acuerdo.
 - En el caso de las lecturas, el protocolo de presentación requiere, al menos, de las siguientes tres líneas de control:
 - 1 **ReadReq:** petición de lectura a memoria (dirección en líneas de información).
 - 2 **DataRdy:** datos preparados (datos en líneas de información).
 - 3 **Ack:** reconocimiento de las otras dos señales.
 - Las señales de control *ReadReq* y *DataRdy* se mantienen activas hasta que sean vistas por la otra parte, que reconoce mediante *Ack*.

Temporización asíncrona

- **Ventajas:**

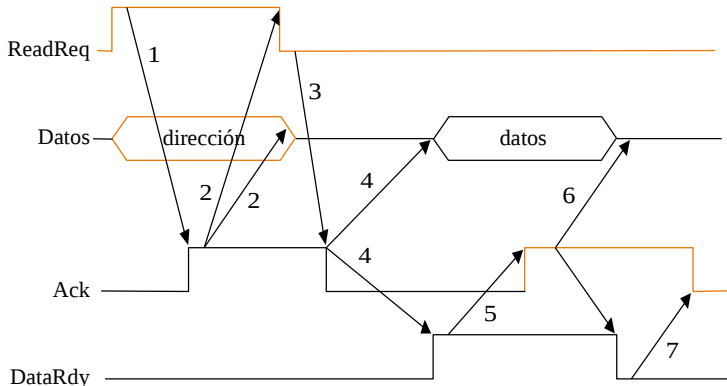
- Dispositivos conectados pueden trabajar a frecuencias de reloj variables.
- El bus puede conectar gran variedad de dispositivos (largo).

- **Inconvenientes:**

- Más lento que el síncrono.
- Necesidad de líneas de control adicionales.
- Fallo de sincronización → Sincronizadores (dispositivo síncrono actúa correctamente ante la llegada de una señal asíncrona).

- **Ejemplo:** algunos buses de E/S (Firewire, USB, ...)

Cronograma de la transferencia asíncrona



- Se activa la petición de lectura y se pone la dirección en el bus.
- La memoria activa la señal de Ack una vez guardada la dirección. El dispositivo desactiva *ReadReq* y libera el bus.
- Se desactiva Ack.
- Una vez encontrado el dato ...
- Se pone en el bus y se activa *DataRdy*.
- Cuando el dispositivo lee el dato del bus, activa la señal Ack.
- En respuesta, se desactiva *DataRdy* y se libera el bus. Se desactiva Ack.

Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 **Implementación de la E/S**
 - Concepto de bus
 - Elementos de diseño de un bus
 - Parámetros de los buses
 - Protocolos de acceso al bus
 - **Mecanismos de control de acceso**
 - Algunos ejemplos de buses comerciales

Mecanismos de control de acceso

- ¿Por qué se necesita un mecanismo para controlar los accesos al bus?
 - Los dispositivos tratarían de utilizar las líneas de control del bus simultáneamente lo que nos llevaría a una situación caótica.
 - Ejemplo: DMA.
- ¿De qué forma puede un dispositivo reservar el bus para utilizarlo en una operación de E/S?
- **Amo o dueño del bus (Bus Master)**: inicia y controla todas las peticiones de uso del bus. Esquema sencillo:
 - El procesador es el único **dueño del bus**.
 - La memoria es un **esclavo**, que responde a las peticiones de lectura y escritura.
- ① Dispositivo A quiere comunicar con B y hace la petición al procesador (amo).
- ② Cuando el procesador puede, atiende la petición de A. Durante toda la comunicación el procesador se encuentra ocupado.

Mecanismos de control de acceso

- **Problema (un dueño):** el procesador debe intervenir en cada operación de E/S.
- **Solución:** múltiples dueños del bus.
 - Cada módulo conectado al bus dispone de lógica para acceder al bus.
 - Necesidad de un **método de arbitraje** para decidir el amo del bus.
- **Características deseables de un método de arbitraje:**
 - Respetar las prioridades.
 - Imparcialidad.
 - Reducir el tiempo para arbitrar el bus (solapar con las transferencias).

Agenda

- 1 Introducción
- 2 Clasificación de los dispositivos de E/S
- 3 Programación de la entrada/salida
 - Puertos, controladoras y canales
 - E/S mapeada en memoria vs. E/S aislada
 - Técnicas de comunicación CPU-E/S
- 4 El papel del sistema operativo
- 5 Implementación de la E/S
 - Concepto de bus
 - Elementos de diseño de un bus
 - Parámetros de los buses
 - Protocolos de acceso al bus
 - Mecanismos de control de acceso
 - Algunos ejemplos de buses comerciales

Algunos ejemplos de buses comerciales

Bus	Anchura (bits)	Velocidad (Mhz)	Ancho de banda (MBytes/seg)
8-bit ISA	8	8,3	7,9
16-bit ISA	16	8,3	15,9
EISA	32	8,3	31,8
VLB	32	33	127,2
PCI	32	33	127,2
64-bit PCI 2.1	64	66	508,6
PCI express (x1,x2, x4, x16)	1 por sentido	–	200x2x(1, 4, 8, 16)
IDE (Ultra DMA)	16	16	33,3
SCSI (Ultra3 Wide)	16	80	160
AGP	32	66	254,3
AGP (x4 mode)	32	66x4	1017,3
AGP (x8 mode)	32	66x8	2034,6
USB 1.1	1	12	1,5
USB 2.0	1	–	60