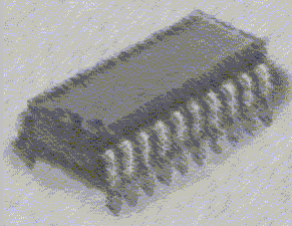


Tema 6. El sistema de entrada-salida

*Arquitectura de
Computadores I*



I. T. Informática de Sistemas

Curso 2009-2010

Tema 6:

Transparencia: 2 / 45

El Sistema de entrada-salida

Índice

- Características del sistema de entrada-salida
- Gestión de la comunicación de la entrada-salida
 - Selección del periférico
 - Sincronización
 - Gestión de las señales de control
- Buses
 - Tipos de buses
 - Obtención de acceso al bus
 - Arbitraje del bus
 - Estándares
- Entrada-salida y sistema operativo
- Ejemplo de hardware real. Apple Macintosh 7200
- Bibliografía

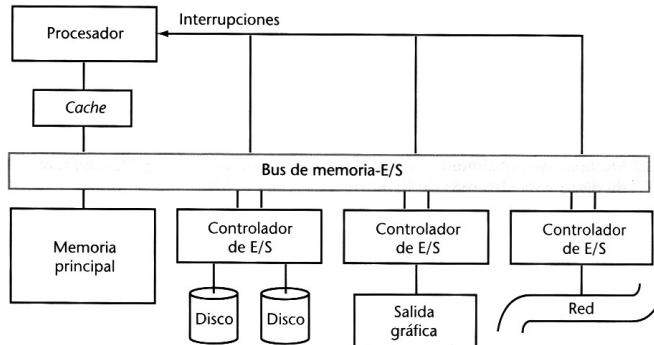


Departamento de Automática
Área de Arquitectura y Tecnología de Computadores

Arquitectura de Computadores I
I. T. Informática de Sistemas

Características del sistema de E/S (I)

- El objetivo del sistema de entrada-salida del computador es comunicarle con el mundo exterior. Para esta comunicación se emplean los periféricos



Características del sistema de E/S (II)

- Los periféricos pueden ser:
 - **Dispositivos de entrada de datos:** el teclado, el ratón, el escáner, los sensores de magnitudes mecánicas o eléctricas, etc.
 - **Dispositivos de salida de datos:** el monitor, la impresora, etc.
 - **Dispositivos de entrada y salida de datos:** como los discos

Componente del periférico	Descripción	Misión
Controlador	Se encarga de la comunicación con la CPU	<ul style="list-style-type: none"> • Transferencia de datos (objeto propio de la comunicación) • Protocolo de la transferencia ("ponerse de acuerdo" en el cómo)
Dispositivo	Puede ser mecánico, electromecánico o electromagnético con su electrónica de control o transductor en el caso de sensores	<ul style="list-style-type: none"> • Relación con el exterior



Gestión de la comunicación en E/S

- En la gestión de la comunicación de la entrada salida se deben tener en cuenta tres aspectos:
 - **Selección del periférico:** identificar el periférico entre todos los disponibles mediante el direccionamiento de los registros de datos y de control
 - **Sincronización con la CPU:** evitar los problemas surgidos de las diferentes velocidades de trabajo de la CPU y de los periféricos. Evitar inundar al periférico con datos provenientes de la CPU y evitar que ésta se quede esperando los datos del periférico
 - **Gestión de las señales de control:** determinar para cada tipos de sincronización y cada tipo de selección de periférico el conjunto de señales de control que deberán ser tenidos en cuenta por la Unidad de Control para la correcta gestión de la comunicación



Selección del periférico (I)

- Para elegir un periférico y comunicarse con él se emplean los puertos de entrada-salida que en su versión más simple se corresponden con un registro
- Para comunicarse con un periférico la CPU debe
 - **Especificar la dirección:** identificar el puerto de entrada-salida entre todos los que tenga (normalmente tienen varios puertos de entrada salida)
 - **Indicar el sentido de la comunicación:** especificar si se trata de una lectura o de una escritura.
 - **Enviar el dato:** si se trata de una escritura



Selección del periférico (y II)

- Especificación de las direcciones de los puertos
 - **Espacio de direcciones separado:** los espacios de direccionamiento de puertos y de direcciones de memoria son diferentes por lo que se deberán incluir instrucciones propias de entrada-salida. Por ejemplo, en el i80x86 instrucciones IN y OUT
 - **Espacio de direcciones común:** tanto direcciones de memoria como de puertos de entrada-salida forman un espacio de direcciones común por lo que las instrucciones del repertorio son las mismas para ambos tipos de transferencia
- Información enviada al periférico (escrituras) o recibida (lecturas):
 - **Datos:** el objeto de la transferencia
 - **Órdenes para el periférico:** de configuración del dispositivo, modo de operación, de reinicio del periférico, etc.
 - **Información de estado:** disponibilidad del periférico, indicación de error, etc.



Sincronización CPU-periférico (I)

- Ajustar la diferencia de velocidades de proceso. Los mecanismos más empleados son:
 - **Entrada-salida programada:** la CPU gestiona la comunicación con los periféricos mediante la ejecución de un programa
 - **Entrada-salida por interrupciones:** los periféricos avisan a la CPU cuando están listos para ser atendidos
 - **Entrada-salida mediante acceso directo a memoria (DMA):** los periféricos trabajan directamente con la memoria escribiendo o leyendo bloques de datos
 - **Entrada-salida mediante procesadores de entrada-salida:** emplean una CPU secundaria dedicada a la gestión de la operación de entrada-salida

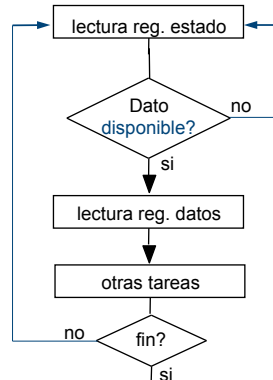


Sincronización CPU-periférico (II) Entrada-salida programada (I)

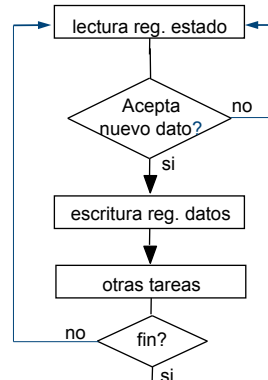
- La CPU gestiona la comunicación con los periféricos mediante la ejecución de un programa
 - La CPU está dedicada completamente a la operación de entrada salida, y la mayor parte a comprobar la disponibilidad del periférico. A este método se le conoce como sondeo
 - La programación es sencilla (para un único periférico) y el hardware asociado es simple
 - El programa de entrada salida se complica cuando hay que atender a diferentes periféricos ya que se debe modificar el programa de control de los mismos
 - La prioridad en el caso de varios periféricos es fija por programa ya que depende de la posición que ocupan en éste para ser atendidos



Sincronización CPU-periférico (III) Entrada-salida programada (y II)



LECTURA (ENTRADA)



ESCRITURA (SALIDA)



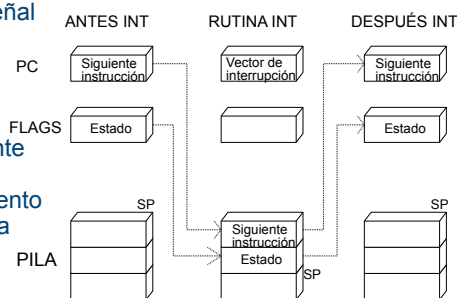
Sincronización CPU-periférico (IV) Entrada-salida por interrupciones (I)

- Los periféricos avisan a la CPU cuando están listos para ser atendidos. La CPU una vez ha indicado al periférico lo que desea, puede dedicarse a realizar otras tareas
- Al diseñar un sistema de entrada-salida mediante interrupciones se debe tener en cuenta:
 - Cómo se solicita la interrupción
 - Cómo se indica la aceptación de la interrupción
 - Cómo se identifica al periférico que ha interrumpido
 - Cómo se resuelve el esquema de prioridad si varios periféricos interrumpen simultáneamente



Sincronización CPU-periférico (V) Entrada-salida por interrupciones (II)

- En la E/S por interrupción:
 1. El periférico realiza la petición de servicio mediante una señal de control específica (interrupción hardware)
 2. Cuando la interrupción es aceptada por la CPU, ésta abandona momentáneamente el programa principal para ejecutar la rutina de tratamiento de la interrupción y realiza la transferencia
- Antes de comenzar la rutina de tratamiento de la interrupción la CPU debe salvar la dirección de retorno y el registro de estado



Sincronización CPU-periférico (VI)

Entrada-salida por interrupciones (III)

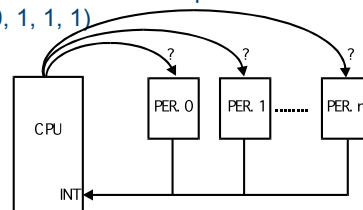
- Cuando existen varios periféricos que pueden interrumpir se deben gestionar las prioridades si realizan peticiones simultáneas
- Para ello se debe tener en cuenta:
 - Cómo conectar los periféricos a la CPU
 - Cómo establecer la política de prioridades
 - Cómo determinar la dirección de la rutina del tratamiento de la interrupción



Sincronización CPU-periférico (VII)

Entrada-salida por interrupciones (IV)

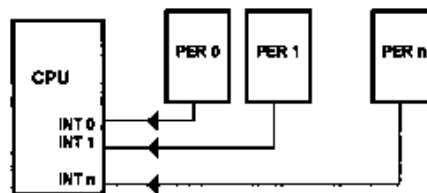
- **Conexión de todos los periféricos a una única línea de interrupción**
 - Todos los periféricos solicitan ser atendidos por la misma línea
 - La dirección de la rutina de tratamiento de la interrupción es fija y común para todos ellos
 - La rutina de tratamiento identifica mediante sondeo cuál de los periféricos ha interrumpido a la CPU
 - La prioridad se da por el orden en el que se sondea a los periféricos. (0, 1, 1, 1, 2, 3, n, 0, 1, 1, 1)



Sincronización CPU-periférico (VIII)

Entrada-salida por interrupciones (V)

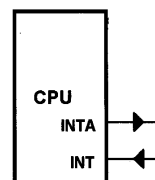
- **CPU con varias líneas de interrupción**
 - Cada periférico solicita ser atendido por una línea diferente
 - La dirección de la rutina de tratamiento de la interrupción es fija pero diferente para cada línea de interrupción
 - La prioridad la determina internamente la CPU



Sincronización CPU-periférico (IX)

Entrada-salida por interrupciones (VI)

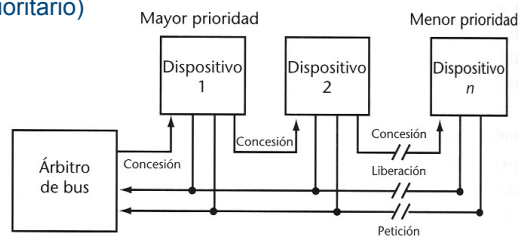
- **CPU con una única línea de petición y aceptación (I)**
 - La CPU cuenta con una línea de entrada INT por la que todos los periféricos solicitan ser atendidos
 - La CPU tiene una línea de salida INTA para dar el reconocimiento de la interrupción al periférico
 - Se debe determinar cómo se conectan los periféricos
 - Se deben gestionar las prioridades en caso de solicitud simultánea
 - Se debe determinar la dirección de la rutina de tratamiento de interrupción



Sincronización CPU-periférico (X) Entrada-salida por interrupciones (VII)

CPU con una única línea de petición y aceptación (II)

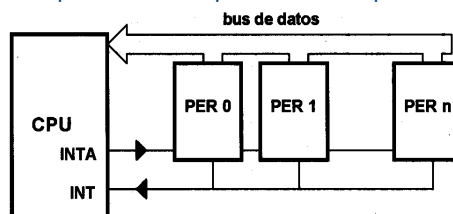
- Encadenamiento de periféricos o daisy-chain
 - Todos los periféricos solicitan la interrupción por la línea INT
 - Cuando la CPU acepta la interrupción activa la línea INTA
 - La dirección de la rutina de tratamiento de interrupción es fija y la resolución de prioridades se realiza por daisy-chain (más próximo es el más prioritario)



Sincronización CPU-periférico (XI) Entrada-salida por interrupciones (VIII)

CPU con una única línea de petición y aceptación (III)

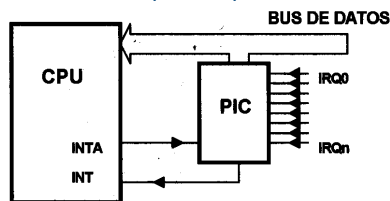
- Interrupciones vectorizadas
 - Todos los periféricos solicitan la interrupción por la línea INT
 - Cuando la CPU acepta la interrupción activa la línea INTA y el periférico más prioritario pone en el bus de datos la dirección de la rutina de tratamiento de interrupción
 - La resolución de prioridades se puede realizar por daisy-chain



Sincronización CPU-periférico (XII) Entrada-salida por interrupciones (IX)

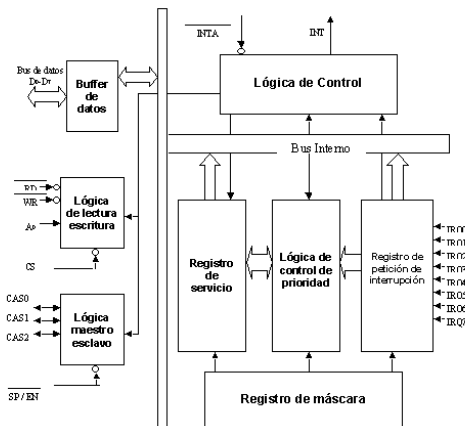
CPU con una única línea de petición y aceptación (IV)

- **Controlador programable de interrupciones (PIC)**
 - Un PIC permite la conexión de varios periféricos a la línea INT de la CPU expandiéndola en varios niveles
 - Realiza la gestión de prioridades de manera flexible
 - Se pueden enmascarar determinadas líneas
 - Admite controladores esclavos para expandir más niveles de interrupción



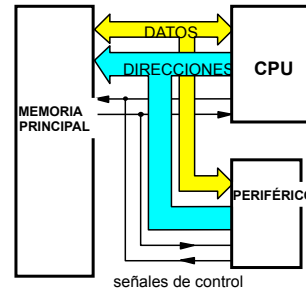
Sincronización CPU-periférico (XIII) Entrada-salida por interrupciones (y X)

PIC 8259



Sincronización CPU-periférico (XIV) Acceso directo a memoria (I)

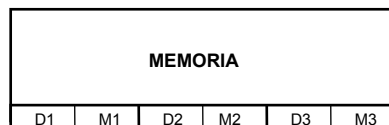
- Se trata de que el periférico pueda realizar la transferencia de un bloque de datos sin intervención de la CPU por cada dato. Para ello:
 - La CPU indica al periférico del tipo y características de la transferencia
 - El periférico se prepara y realiza la transferencia con la memoria
 - El periférico avisa a la CPU al terminar
- **Controlador de DMA.** Dado que la gestión de la transferencia en el bus es muy compleja, se crea este controlador para simplificar el diseño y la misión del periférico



Sincronización CPU-periférico (XV) Acceso directo a memoria (II)

Mediante memoria multipuerta

- Una memoria multipuerta es aquella que tiene posibilidad de realizar transferencias simultáneas con el exterior por medio de varias puertas
- A cada puerta se le asigna un registro de datos y otro de direcciones que indicarán el dato transferido y la dirección de transferencia



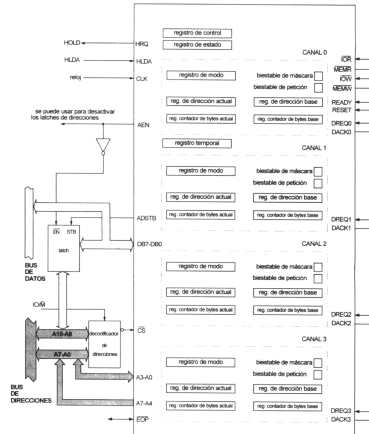
Mediante robo de ciclo

- La memoria tiene una sola puerta que es compartida entre el controlador de DMA y la CPU
- Cuando el periférico requiera una transferencia DMA debe hacer una petición de robo de ciclo a la CPU (HOLD)
- La CPU lo concede (HOLDA)
- El controlador de DMA gestiona los buses de direcciones y datos, y las señales de control de memoria y de E/S
- Terminada la transferencia, el controlador lo comunica a la CPU



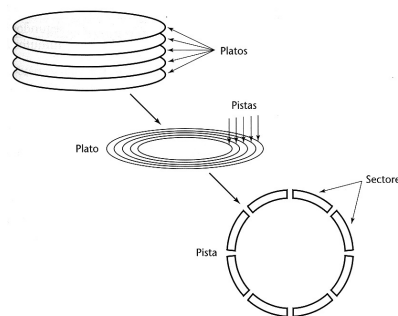
Sincronización CPU-periférico (XVI) Acceso directo a memoria (y III)

Controlador DMA 8237



Sincronización CPU-periférico (XVI) Procesadores de entrada-salida (I)

- Es emplear otra CPU para gestionar las operaciones de entrada-salida
- La CPU principal indica al procesador de entrada-salida la operación que hay que realizar y éste la lleva a cabo
- El procesador de entrada-salida puede gestionar el proceso mediante E/S programada, por interrupciones o por DMA
- El sistema de entrada-salida se vuelve más complejo pero más potente



Gestión de las señales de control

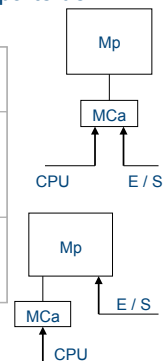
- Si se desea diseñar un sistema de E/S con las diferentes opciones explicadas con anterioridad se deberán introducir las señales siguientes:
 - **IO / M:** indicará si se trata de un acceso a un puerto de E/S o a memoria
 - **RD:** se desea leer del periférico
 - **WR:** se desea escribir en el periférico
 - **INTR:** solicitud de interrupción por el controlador del periférico
 - **INTA:** reconocimiento de la interrupción por la CPU
 - **HOLD:** petición de transferencia por DMA del controlador del periférico
 - **HOLDA:** concesión de transferencia por DMA por la CPU



Memoria cache y entrada-salida

- Además de los accesos al sistema de memoria realizados por el procesador, se deben tener en cuenta los accesos por parte del sistema de E / S

Método	Ventajas	Inconvenientes
CPU y E / S acceden a memoria a través de la MCa	Consistencia de datos ya que CPU y E / S ven la misma copia de la información	Incremento de tasa de fallos
CPU conectado a MCa y E / S a MP	La CPU aprovecha las ventajas de la MCa	Consistencia entre MP y MCa



Buses (I)

- **Bus.** Es un canal de comunicación compartido que emplea un conjunto de cables para conectar los diferentes subsistemas
- **Los buses están formados por:**
 - Un conjunto de señales de control
 - Un conjunto de líneas de datos
- **Ventajas:**
 - **Versatilidad**, ya que una vez definido el sistema de conexión se pueden añadir nuevos dispositivos
 - **Bajo coste**, puesto que los cables que lo forman se pueden compartir de diferentes maneras
- **Inconvenientes:**
 - Es un **cuello de botella** que limita la máxima productividad de entrada-salida del sistema
 - **Dificultad de diseño** debida a que depende del número de elementos que se conectarán y a la longitud del bus



Buses (II)

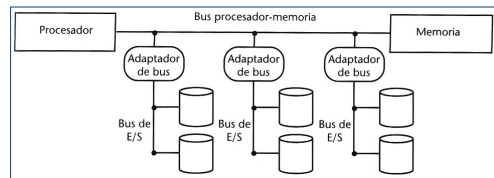
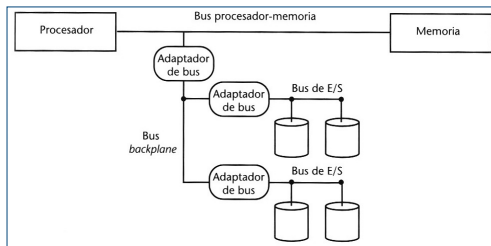
Tipos de buses (I)

- **Los buses se clasifican en tres tipos**
 - **Buses procesador-memoria.** Son de longitud pequeña, de alta velocidad y adaptados al sistema de memoria para maximizar el ancho de banda
 - **Buses backplane.** Suelen recibir este nombre por estar contruidos en el plano posterior. Es una estructura de conexión en el chasis y la memoria, la CPU y los periféricos lo emplean para comunicarse
 - **Buses de entrada-salida.** Pueden tener una longitud grande y sirven para conectar muchos dispositivos con anchos de banda muy diferentes entre si. Además para comunicarse con la memoria suelen emplear o buses procesador-memoria o buses *backplane*



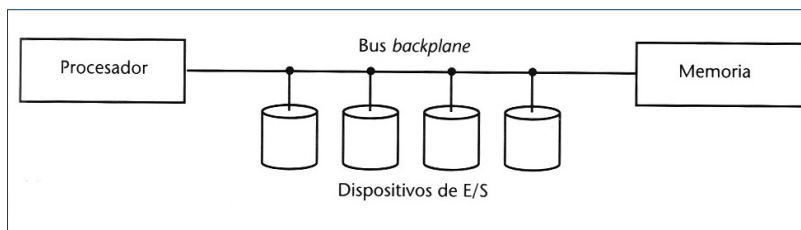
Buses (III)

Tipos de buses (II)



Buses (IV)

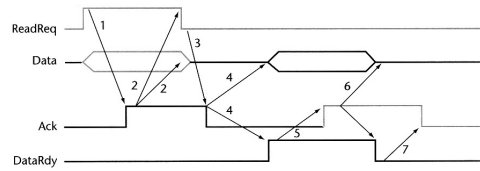
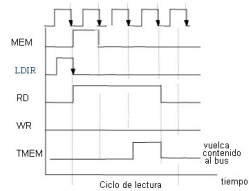
Tipos de buses (III)



Buses (V)

Modo de operación

- Los esquemas de comunicación, síncrono y asíncrono, se deben a las diferencias entre los buses procesador-memoria, backplane y de entrada-salida
- **Bus síncrono.** Tiene una señal de reloj entre las líneas de control y un protocolo para la comunicación gobernado por esta señal de reloj
- **Bus asíncrono.** No tiene señal de reloj y emplea un protocolo de *handshaking* para coordinar la comunicación
- Los buses síncronos suelen ser un poco más rápidos y los buses asíncronos más flexibles



Buses (VI)

Acceso al bus

- El control de acceso al bus tiene por objetivo que no se produzcan conflictos al acceder varios elementos al bus
- Es una manera de reservar el bus por un dispositivo
- El mecanismo de acceso es tener uno o más maestro del bus y varios esclavos
 - **Maestro de bus.** Controla el acceso al bus. Debe ser capaz de iniciar y controlar todas las peticiones de bus
 - **Esclavo de bus.** Responde a las peticiones del maestro, o de los maestros, pero nunca genera peticiones
- El procesador siempre es un maestro de bus y la memoria un esclavo de bus



Buses (VII)

Arbitraje del bus

- Ocurre cuando tenemos un esquema de varios maestros y se debe decidir cuál es el siguiente maestro que puede usar el bus
- Se debe garantizar que el maestro más prioritario sea atendido cuanto antes, pero que el menos prioritario no se quede sin su acceso al bus
- **Métodos de arbitraje:**
 - **Arbitraje en daisy-chain.** Una línea de concesión recorre todos los dispositivos
 - **Arbitraje centralizado.** Un árbitro centralizado selecciona al dispositivo y le nombra maestro del bus
 - **Arbitraje distribuido por auto selección.** Los dispositivos indican la prioridad de manera que el más prioritario se erige en maestro
 - **Arbitraje distribuido por detección de colisión.** Una vez detectada la colisión se emplea un esquema para seleccionar al maestro entre los dispositivos que causaron la colisión



Buses (VIII)

Temporización

- La temporización se refiere a la manera en que se realizan las fases de direccionamiento y de datos durante la transferencia
- **Pueden ser:**
 - **Bus de ciclo completo.** Las fases de direccionamiento y de datos van seguidas sin abandonar el bus. El diseño es sencillo, pero presenta una menor latencia y un ancho de banda más pequeño
 - **Bus de ciclo partido.** El bus se abandona entre las fases de direccionamiento y de datos. Permite la presencia de varios maestros de bus. Aumenta la latencia y el ancho de banda y que se transmita la identidad del maestro de bus. Por último, los esclavos deben ser capaces de solicitar el bus



Buses (IX)

Ancho de banda

- Aunque el ancho de banda del bus viene determinado casi por la elección de un protocolo síncrono o asíncrono y por las características de temporización del bus, existen otros factores que afectan al ancho de banda y que son:
 - **Anchura del bus de datos.** Al aumentar la anchura del bus de datos se pueden transmitir varias palabras en menos ciclos de bus
 - **Líneas de datos y direcciones separadas o multiplexadas.** Si las líneas están separadas se aumenta el ancho de banda en las escrituras
 - **Transferencias por bloques.** Si se transfieren múltiples palabras en ciclos consecutivos de bus sin enviar la dirección o sin liberar el bus se reduce el tiempo de transmitir un bloque grande de datos



Buses (X)

Tipos de operaciones en los buses

- Las operaciones que se puedan realizar por el bus determinarán los tipos de ciclos que tendremos en cuenta:
 - Ciclo de lectura, o escritura, simple
 - Ciclo de lectura, o escritura, múltiple
 - Ciclo de interrupción
 - Ciclo de acceso directo a memoria
 - Ciclo de lectura después de escritura



Buses (XI)

Buses de E/S. Rendimiento y coste

Opción	Alto rendimiento	Bajo coste
Anchura de bus	Líneas separadas de datos y direcciones	Multiplexado de las líneas
Anchura de datos	Más ancho implica más rápido	Más estrecho implica más barato
Tamaño de la transferencia	Transferir múltiples palabras disminuye la sobrecarga del bus	Transferir una única palabra es más sencillo
Número de maestros del bus	Múltiples maestros, requiere arbitraje	Un único maestro
Sincronización	Síncrono	Asíncrono
Temporización	Bus de ciclo partido	Bus de ciclo completo



Buses (XII)

Estándares de buses (I)

- Los estándares surgen para facilitar el diseño de los buses y de los periféricos que se pueden conectar a ellos
- Tipos de estándares:
 - **Estándar de facto.** Se populariza por el uso. Por ejemplo, el bus PC-AT o las normas MNP de los modem
 - **Estándar de la industria.** Varias empresas se ponen de acuerdo en las especificaciones. Por ejemplo, SCSI y Ethernet
 - **Estándar por un comité.** Son organizaciones autorizadas para la creación y aprobación de estándares. Por ejemplo, PCI



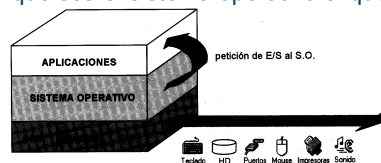
Buses (y XIII) Estándares de buses (y II)

Características	PCI	SCSI
Nombre estándar	PCI	ANSI X3.121
Tipo de bus	Backplane	Entrada-Salida
Anchura básica del bus de datos	32-64	8-32
Datos-direcciones	Multiplexadas	Multiplexadas
Número de maestros	Múltiples	Múltiples
Arbitraje	Centralizado	Auto selección
Sincronización	Síncrono	Asíncrono y síncrono
Número máximo de dispositivos	1.024 en segmentos de 32	7-31
Ancho de banda	133-512MB/sg.	2,5-40MB/sg.
Longitud máxima	0,5 metros	25 metros



Entrada-salida y sistema operativo (I)

- Al trabajar en lenguajes de alto nivel, las peticiones de E / S se compilan mediante llamadas al sistema
- El sistema operativo se encarga de gestionar todos los recursos del sistema operativo determinando las direcciones de memoria que se corresponden a los nombres asignados a los periféricos
 - **Ventajas:** más lenta pero es portable
 - **Inconvenientes:** es más lenta
- En aplicaciones en tiempo real se debe programar la E / S de forma directa sin dejar que sea el sistema operativo el que la controle



Entrada-salida y sistema operativo (II)

- El sistema operativo debe tener en cuenta:
 - Que el sistema de entrada-salida es compartido por múltiples programas que usan el procesador
 - Que un programa de usuario solamente accede a los dispositivos de entrada-salida sobre los que el usuario tiene acceso
 - Que los sistemas de entrada-salida emplean frecuentemente interrupciones para comunicar información sobre las operaciones de entrada-salida y que deberán ser atendidas por el sistema operativo
 - Que el control de bajo nivel de un dispositivo es complejo y que deben ser simplificados al programador mediante las llamadas al sistema operativo
 - El sistema operativo debe proporcionar un acceso equitativo a los recursos de entrada-salida compartido e intentar planificar los accesos para mantener la productividad del sistema



Entrada-salida y sistema operativo (III)

- El sistema operativo debe poder comunicarse con los dispositivos de entrada-salida
 - Enviando órdenes a los periféricos: lectura, escritura, posicionamiento de un disco, etc.
 - Recibiendo de los periféricos las notificaciones de que se han completado las operaciones solicitadas o de que se ha producido un error
 - Transfiriendo los datos entre memoria y el dispositivo de entrada-salida



Entrada-salida y sistema operativo (y IV)

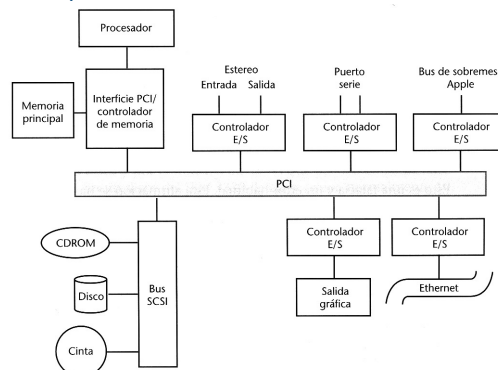
- Se han resaltado en negritas las llamadas al sistema a un sistema UNIX para la operación de entrada-salida en el programa C siguiente:

```
#define BUFSIZE 512
Void copy (from, to)
Char * from, *to;
{
    int fromfd, tofd, nread;
    char buf[BUFSIEZE];
    if ((fromfd = open( from, 0) == -1) syserr(from);
    if ((tofd = creat (to, 0666)) == -1) syserr(to);
    while ((nread = read(fromfd, buf, sizeof(buf))) > 0)
        if (write(tofd, bud, nread) !=nread) syserr ("write");
    if (nread == -1) syserr ("read");
    if (close(fromfd) == -1 || close (tofd) == -1) syserr ("close");
}
```



Apple Macintosh 7200

- El PCI se usa como bus *backplane* y los dispositivos de menor rendimiento emplean un SCSI



Bibliografía

- Estructura y diseño de computadores
David A. Patterson y John L. Hennessy. Reverté, 2000
Capítulo 8
- Arquitectura de computadores. Un enfoque cuantitativo
John L. Hennessy y David A. Patterson. Mc Graw Hill, 3ª ed, 2002
Capítulo 9
- Arquitectura de computadores
José A. de Frutos y Rafael Rico. Servicio de Publicaciones de la
Universidad de Alcalá, 1995
Capítulo 6
- Fundamentos de computadores
Pedro de Miguel Anasagasti. Paraninfo, 1999
Capítulo 8

