



**Universidad  
Europea de Madrid**

**LAUREATE** INTERNATIONAL UNIVERSITIES

## **FUNDAMENTOS DE COMPUTADORES**

### **BUSES II**

© Todos los derechos de propiedad intelectual de esta obra pertenecen en exclusiva a la Universidad Europea de Madrid, S.L.U. Queda terminantemente prohibida la reproducción, puesta a disposición del público y en general cualquier otra forma de explotación de toda o parte de la misma.

La utilización no autorizada de esta obra, así como los perjuicios ocasionados en los derechos de propiedad intelectual e industrial de la Universidad Europea de Madrid, S.L.U., darán lugar al ejercicio de las acciones que legalmente le correspondan y, en su caso, a las responsabilidades que de dicho ejercicio se deriven.

## Índice

Presentación .....	4
Elementos de diseño de un bus .....	6
Tipos de buses .....	7
Dedicadas .....	7
Multiplexadas .....	7
Dedicado físicamente .....	7
Método de arbitraje .....	9
Temporización síncrona .....	11
Temporización síncrona .....	11
Temporización asíncrona .....	13
Tipos de transferencia de datos .....	16
Ejemplo: el bus PCI .....	19
Arbitraje del bus PCI: .....	19
Ejemplo: configuración del bus PCI I .....	21
Ejemplo: configuración del bus PCI II .....	22
Ejemplo: estructura del bus PCI .....	24
Ejemplo: órdenes del PCI .....	29
Resumen .....	31

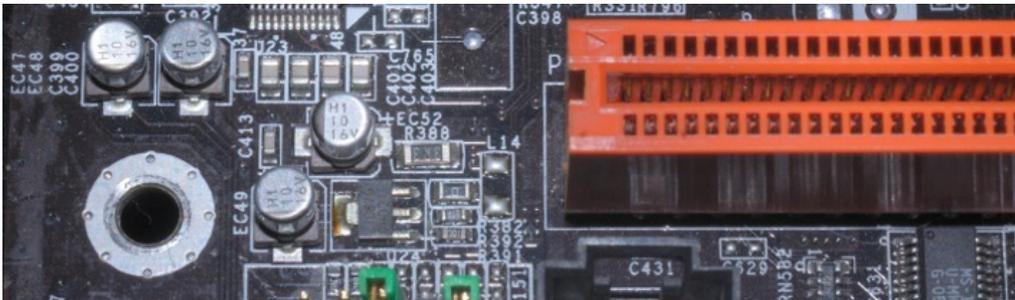
## Presentación

En este tema terminaremos de ampliar nuestros conocimientos sobre los buses, pasando de una visión más genérica a una vista en detalle de cómo puede ser un bus, de su comportamiento y de sus diferentes maneras de implementar cada una de sus características.

Los puntos que desarrollaremos en este tema son:

- Elementos de diseño de un bus.
- Tipos de buses.
- Métodos de arbitraje.
- Temporización síncrona y asíncrona.

Para terminar con un ejemplo, veremos uno de los buses más utilizados hoy en los ordenadores actuales, tanto ordenadores de uso domestico como ordenadores servidores de prestaciones medio-altas: el bus PCI. Veremos algunas de sus características y podremos sacar una relación directa con las características básicas tratadas en temas anteriores.





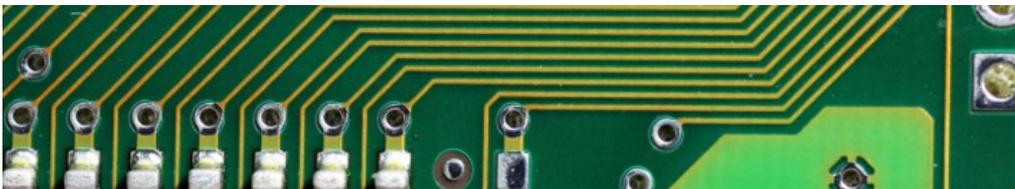
## Elementos de diseño de un bus

A pesar de que hay una gran diversidad de buses, lo que provoca multitud de especificaciones y tipos, en ellos hay una serie de parámetros o elementos de diseño que se pueden distinguir y clasificar para una mejor comprensión de los mismos.

Los puntos que vamos a diferenciar son los siguientes.

Tipo	Describe cómo y cuántos dispositivos se conectan al bus, así como la organización física de las señales que lo componen.
Método de Arbitraje	Cómo se coordina el derecho a uso del bus.
Temporización	Cómo es el método de transmisión de información por el bus, si está o no controlado por un reloj, etc.
Anchura del Bus	Tamaño del bus. Cuántas señales contiene, cómo se dividen, etc.
Tipo de Transferencia de Datos	No todos los accesos a los datos siguen la secuencia dirección-dato. Aunque esta es quizás la más utilizada, no es la única. Aquí veremos de qué modo se puede transferir datos por un bus.

En las siguientes pantallas, describiremos en detalle cada una de estas características, haciendo especial hincapié en las ventajas que cada característica ofrece.



## Tipos de buses

Las líneas que forman un bus, se pueden dividir en dos tipos genéricos: dedicadas y multiplexadas.

### Dedicadas

Hay dos maneras de entender el concepto de dedicado:

- Se define como línea dedicada a aquella que está permanentemente asignada a una función.
- Se define como línea dedicada a aquella que está permanentemente asignada a un conjunto de componentes del computador.

Por lo tanto, una línea puede ser dedicada funcional o físicamente. Un ejemplo de dedicación funcional, como vimos en el bus de sistema, es el uso de líneas dedicadas a datos y otras a direcciones.

### Multiplexadas

Sin embargo esto no es obligatorio ya que, en muchos buses, las mismas líneas que transmiten datos transmiten direcciones y unas líneas de control especial permiten diferenciar que tipo de información se está transmitiendo y facilitan la información de dato válido o dirección válida. De esta manera, hay un tiempo límite para la lectura de los datos y de las direcciones ya que, con el tiempo, el dato y el tipo de dato que se transmite es diferente. A este formato se le conoce como *multiplexado en el tiempo*.

La ventaja del multiplexado en el tiempo, es el ahorro de número de líneas para la comunicación de datos, que a su vez conlleva el consiguiente ahorro económico.

Por contraposición, el bus dedicado funcionalmente es más rápido ya que en un intervalo de tiempo puede transmitir la dirección y el dato simultáneamente, lo que sería el doble de rápido que uno multiplexado, ya que este necesitaría dos intervalos de tiempo: uno para la dirección y otro para el dato.

### Dedicado físicamente

La dedicación física es mucho menos frecuente y se restringe a buses de muy alto volumen de uso sin que los costes sean importantes (por tanto, buses pequeños en longitud). Los buses dedicados conectan un conjunto cerrado de dispositivos y a este bus no se le pueden añadir más componentes. Un ejemplo es el bus local, que conecta exclusivamente la microprocesador y la caché.

**Ahorro en el número de líneas**

Por ejemplo, si pensamos en el cable de teléfono que une dos hogares intercontinentales, no es lo mismo que este cable sea de dos hilos que de 100.

### Método de arbitraje

Son muy pocos los buses en los que solo hay un emisor de información y un receptor. Lo normal, es que un bus se comparta entre muchos emisores y muchos receptores.

Es más, lo más normal es que todos los componentes conectados al bus puedan funcionar **como emisores y receptores a intervalos de tiempo**. Por tanto, surge el problema de coordinar y establecer protocolos de uso del bus, que se solventa a través de los denominados **métodos de arbitraje**.

El método más sencillo es el que **no dispone de árbitro**, transmitiendo cada uno cuando puede y quiere. Este tipo de arbitraje es barato pero el rendimiento se reduce mucho, ya que al transmitir cuando se quiera, lo más normal es que sobre el bus se pretendan hacer muchas comunicaciones simultáneamente. En los buses omnidireccionales, esto provoca que las transmisiones de unos interrumpen las de otros, y al final nadie pueda transmitir. A pesar de que este método parece ridículo, y su funcionamiento imposible, hay dispositivos que lo utilizan, como es el caso de Ethernet por cable físico (la red de ordenadores clásica de cable de 4 pares o el de cable coaxial).

Otra manera de implementar un orden a las comunicaciones, es poner un **moderador del bus**, al que llamaremos árbitro. Dependiendo de dónde se sitúa el árbitro y de quien actúa como tal, podemos diferenciar los siguientes tipos.

<b>Árbitro centralizado</b>	Es un componente hardware fijo del bus, que se encarga de distribuir los tiempos y prioridades en el uso del mismo. Este componente siempre está presente y forma parte de la implementación de bus.
<b>Árbitro distribuido</b>	En este caso no hay un controlador físico, o un dispositivo que solo está presente para tener este rol, en el bus. Por el contrario, cada módulo que se conecta al bus, posee las funcionalidades de árbitro o, al menos, parte de ellas.

Por lo tanto, cada dispositivo puede funcionar en un momento dado como árbitro del bus (al que llamaremos maestro), siendo el encargado de distribuir los tiempos a los demás componentes (llamados esclavos) y manteniendo este rol durante el tiempo que permanezca conectado al bus, independientemente de si tiene o no que transmitir.

Por el contrario, también puede funcionar como árbitro solo durante el tiempo que necesita utilizar el bus. Terminadas estas necesidades, pasa el rol de árbitro a otro dispositivo para que este pueda hacer uso del bus, repitiéndose esta situación sucesivamente.

**La situación se repite sucesivamente**

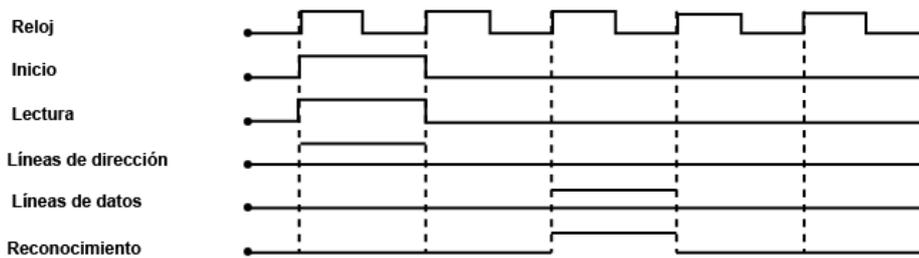
Es muy parecido al poseedor del *token* (*testigo*): el dispositivo que posea el token tiene el uso exclusivo del bus y, cuando termina su actividad, pasa el token a otro dispositivo para que pueda transmitir.

### Temporización síncrona

El término temporización hace referencia a cómo se coordinan los eventos en el bus. Puede ser síncrona o asíncrona.

#### Temporización síncrona

Los eventos en el bus están determinados por el ciclo de reloj (un pulso de reloj, un evento). El bus incluye una línea de reloj a través de la que se transmite una señal de sincronización.



Veamos un ejemplo. Supongamos que la CPU activa una señal de lectura y sitúa una dirección de memoria en las líneas de dirección durante el primer ciclo. Una vez que las líneas se han estabilizado, el procesador activa una señal de inicio para indicar la presencia de la dirección en el bus.

En el caso de una lectura	En el caso de una escritura
<ul style="list-style-type: none"> <li>● El procesador pone una orden de lectura al comienzo del segundo ciclo.</li> <li>● El módulo de memoria reconoce la dirección y, después de un retardo de un ciclo, sitúa el dato en las líneas de datos.</li> <li>● El procesador lee el dato de dichas líneas y quita la señal de lectura.</li> </ul>	<ul style="list-style-type: none"> <li>● El procesador pone el dato en las líneas de datos al comienzo del segundo ciclo y pone la orden de escritura una vez estabilizadas las líneas de datos.</li> <li>● El módulo de memoria copia la información de las líneas de datos durante el tercer ciclo de reloj.</li> </ul>

#### Síncrona o asíncrona

La temporización puede clasificarse como:

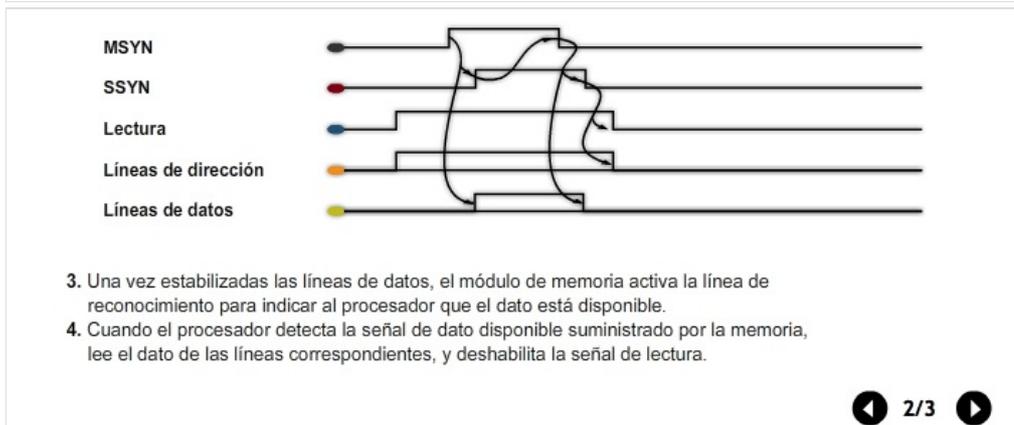
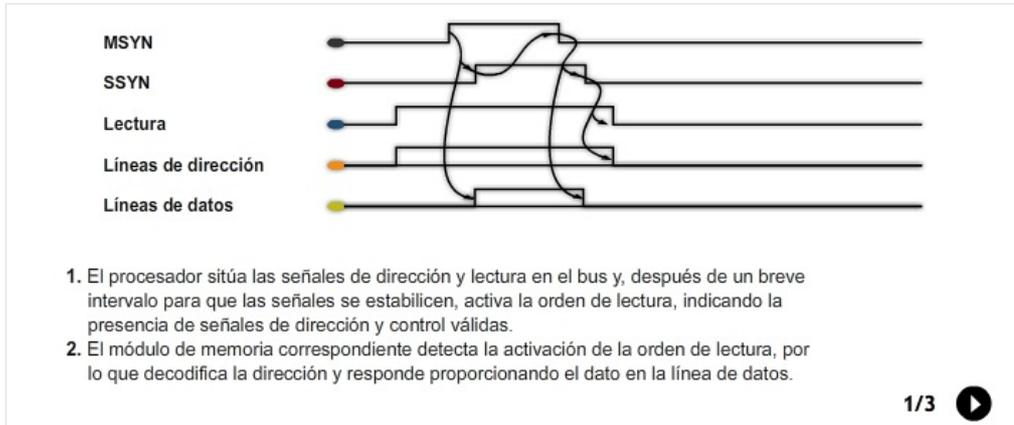
- **Síncrona**, si los dispositivos utilizan unos tiempos prefijados por un reloj.
- **Asíncrona**, si no hay un reloj que determine el fin y principio del siguiente evento.

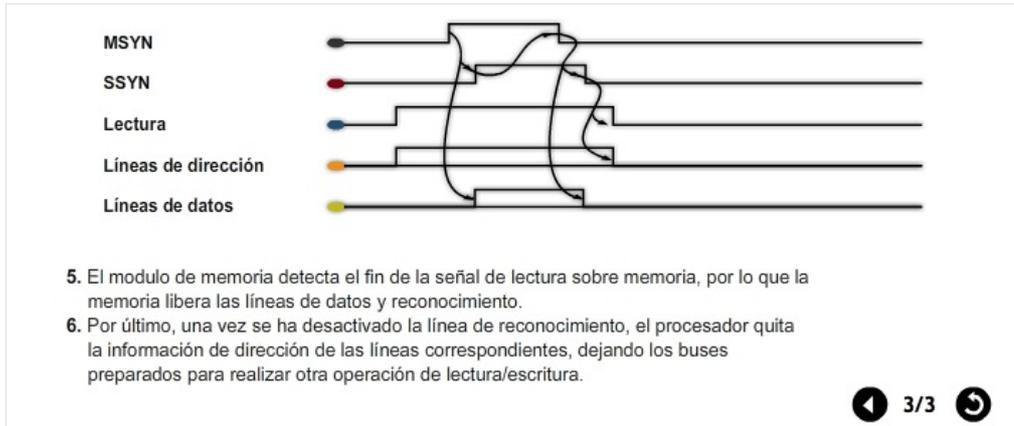
**Sincronización**

Todos los dispositivos del bus pueden leer la línea de reloj, y todos los eventos empiezan y terminan a cada ciclo de reloj. Ojo, el evento termina en un ciclo de reloj, pero la tarea no tiene por qué. Es decir, una operación de lectura sobre memoria es una tarea, no un evento.

### Temporización asíncrona

Con la temporización asíncrona, al no haber una señal de reloj que determine el principio y fin del tiempo de actuación, la presencia de un evento en el bus es consecuencia, y depende, de que se produzca un evento previo que indique que una fase se ha terminado y se puede proceder a la siguiente fase de la acción (siguiente evento de bus). Veamos un ejemplo:





En un bus asíncrono no hay señal de reloj, pero tienen que existir señales que permitan la sincronización de las fases. En este caso, las señales de *lectura/reconocimiento* son las que funcionan como señales de sincronización.

Si la ventaja de un bus síncrono es la facilidad de implementación de dispositivos que puedan transmitir información a través del mismo, el inconveniente es que la [mejora en hardware](#) no tiene repercusión en la velocidad de transmisión de datos, ya que estos están limitados por la velocidad del reloj. Los defectos y virtudes de un bus síncrono se corresponden, respectivamente, con las ventajas y desventajas de un bus asíncrono.

**Mejora en hardware**

Para explicar mejor esta circunstancia de la mejora en el hardware, podríamos hacer el siguiente símil: imaginemos que estamos aprendiendo a hablar un idioma extranjero, por ejemplo el inglés.

Al principio nos costará mucho hablar, por lo que pronunciaremos una palabra por segundo, pero conforme vayamos cogiendo más dominio del nuevo idioma, podremos ir aumentando la velocidad de transmisión de palabras a 2, 3, etc., por segundo, siempre que nuestro interlocutor haya aprendido al mismo ritmo. En caso contrario la comunicación tendrá que ajustarse a la velocidad del más lento.

Así, en una comunicación asíncrona pueden cohabitar dos dispositivos de generaciones diferentes y de velocidades diferentes sin problema, ya que la transmisión se ajusta a la velocidad el más lento. Por el contrario, esto no puede ocurrir en un bus síncrono. Aunque remplacemos dispositivos lentos por rápidos la comunicación entre ellos no puede evolucionar, ya que es el reloj la que indica la tasa de transferencia de datos. Es decir, si la transmisión se diseña para pronunciar una palabra por segundo, siempre hablaremos a una palabra por segundo, aunque los interlocutores sean bilingües.

### Tipos de transferencia de datos

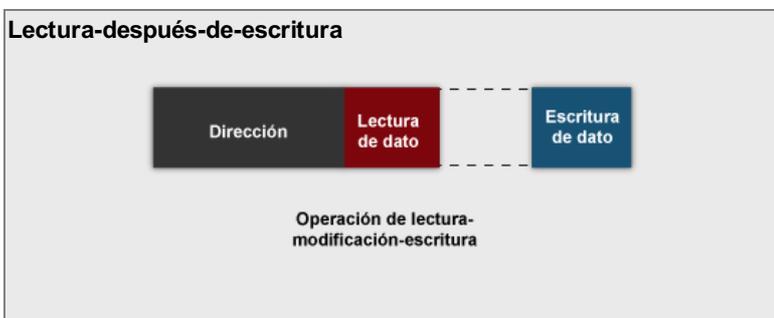
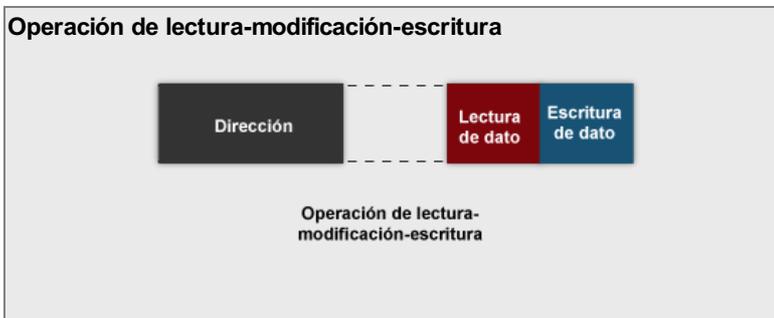
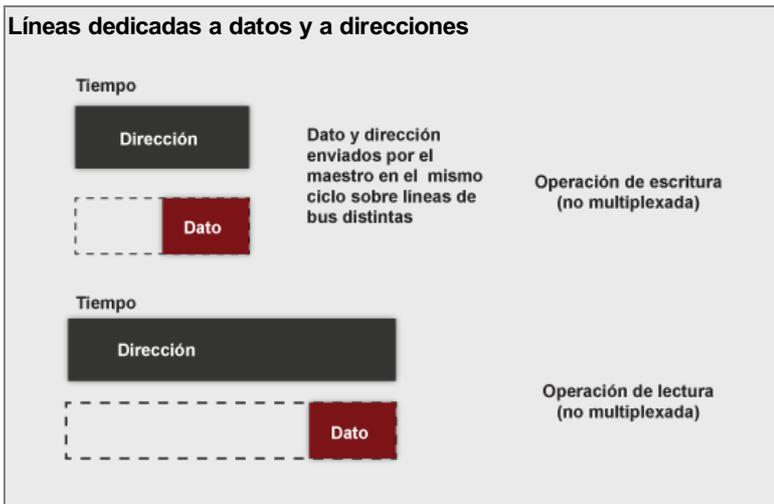
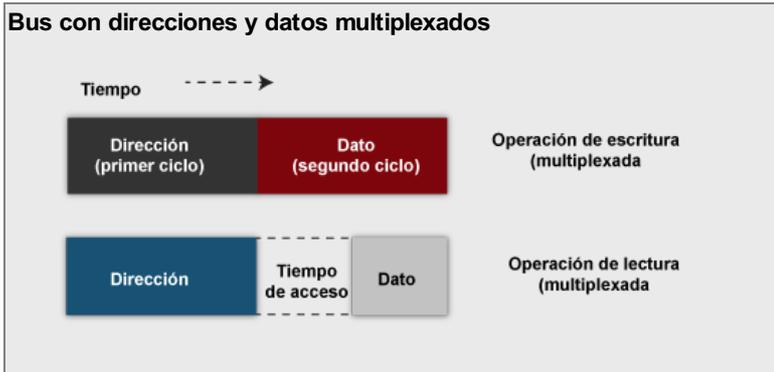
El formato de dirección-dato es un esquema muy utilizado para las operaciones de lectura o escritura, pero no el único. De hecho, este sistema de dirección-dato es utilizado casi únicamente sobre la memoria. Por consiguiente hay otros modos de transmisión de datos, de los que podemos describir:

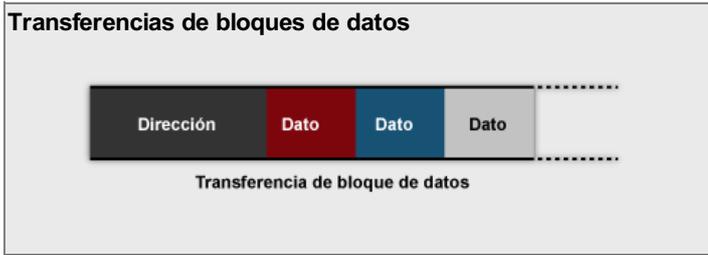
- En el caso de un bus con direcciones y datos multiplexados, el bus se utiliza primero para especificar la dirección y luego para transferir el dato.
- En el caso de que haya líneas dedicadas a datos y a direcciones, la dirección se sitúa en el bus de direcciones y se mantiene ahí mientras que el dato se ubica en el bus de datos.

En ciertos buses también son posibles algunas operaciones combinadas.

- Una operación de lectura-modificación-escritura es simplemente una lectura seguida inmediatamente de una escritura en la misma dirección. La dirección se proporciona una sola vez al comienzo de la operación. La operación completa es generalmente indivisible de cara a evitar cualquier acceso al dato por otros posibles maestros del bus. El objetivo primordial de esta posibilidad es proteger los recursos de memoria compartida en un sistema con multiprogramación.
- La lectura-después-de-escritura es una operación indivisible que consiste en una escritura seguida inmediatamente de una lectura en la misma dirección. La operación de lectura se puede realizar con el propósito de comprobar el resultado.

Algunos buses también permiten transferencias de bloques de datos. En este caso, un ciclo de dirección viene seguido por  $n$  ciclos de datos. El primer dato se transfiere a/desde la dirección especificada mientras que el resto de datos se transfieren a/desde las direcciones siguientes.





### Ejemplo: el bus PCI

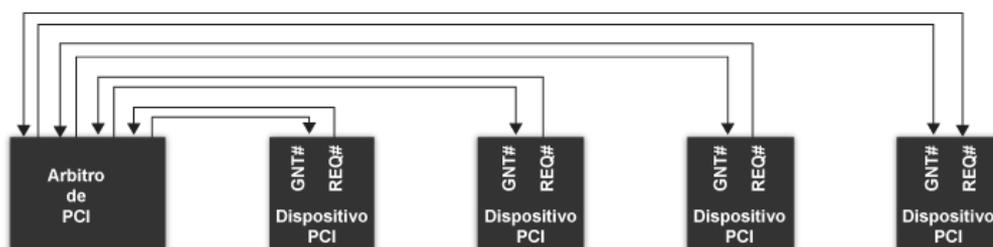
Como ejemplo de bus real, vamos a ver una de los buses más extendidos en los ordenadores actuales: **el bus PCI**. El bus PCI es un bus de ancho de banda elevado, independiente del procesador, que se puede utilizar como bus de altas prestaciones o bus para una arquitectura multinivel. El bus PCI proporciona muy buenas prestaciones para los subsistemas de E/S de alta velocidad.

Haciendo una relación directa con las características vistas anteriormente sobre los buses, el estándar actual permite el uso de hasta 64 líneas de datos a 66 MHz, lo que permite una tasa de transferencia de entre los 528 MB/s y los 4,2 Gbps. La principal característica que ha hecho que el bus PCI se haya convertido en un estándar es su diseño físico, ya que ha sido diseñado específicamente para ajustarse económicamente a los requisitos de E/S de los sistemas actuales (se implementa con muy pocos circuitos integrados y permite que otros buses se conecten al bus PCI) de manera muy eficiente y económica.

#### Arbitraje del bus PCI:

El bus PCI utiliza un esquema de arbitraje centralizado síncrono en el que cada maestro tiene una única señal de petición (REQ) y cesión (GNT) del bus. Estas líneas se conectan a un árbitro central y se utiliza un simple intercambio de las señales de petición y cesión para permitir el acceso al bus.

La especificación PCI no indica un algoritmo particular de arbitraje. El árbitro puede utilizar un procedimiento de *primero-en-llegar-primero-en-servirse*, un procedimiento de cesión cíclica (*round-robin*) o cualquier clase de esquema de prioridad.



**Convertido en estándar**

El bus PCI ha sido desarrollado por Intel pero las especificaciones son de dominio público, lo que ha permitido que esté soportado por prácticamente la totalidad de los procesadores y periféricos.

**PCI**

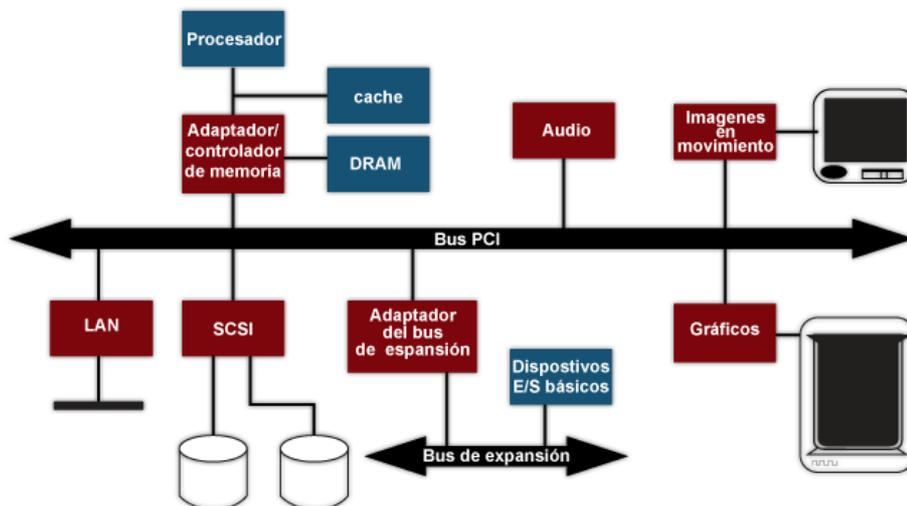
Las siglas PCI corresponden a **P**eripheral **C**omponent **I**nterconnect (Interconexión de Componente Periférico).

**Esquema de prioridad**

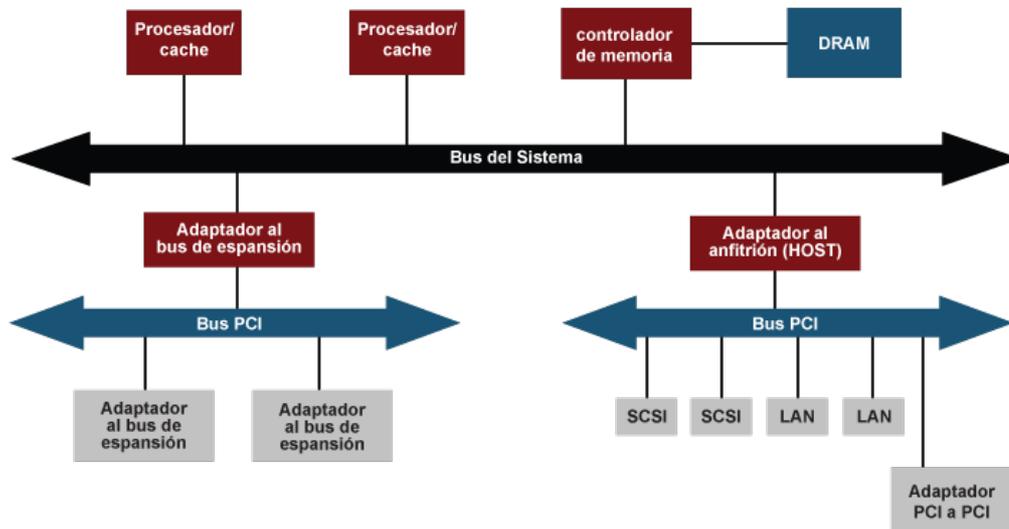
El maestro del PCI establece, para cada transferencia que desee hacer, si tras la fase de dirección sigue una o más fases de datos consecutivas.

### Ejemplo: configuración del bus PCI I

La siguiente figura muestra la forma usual de utilizar el bus PCI en un sistema uniprocador. Un dispositivo que integra el controlador (caché) de DRAM y el adaptador al bus PCI que proporciona el acoplamiento al procesador y la posibilidad de generar datos a velocidades elevadas. El adaptador actúa como un registro de acoplo (buffer) de datos puesto que la velocidad del bus PCI puede diferir de la capacidad de E/S del procesador.



## Ejemplo: configuración del bus PCI II



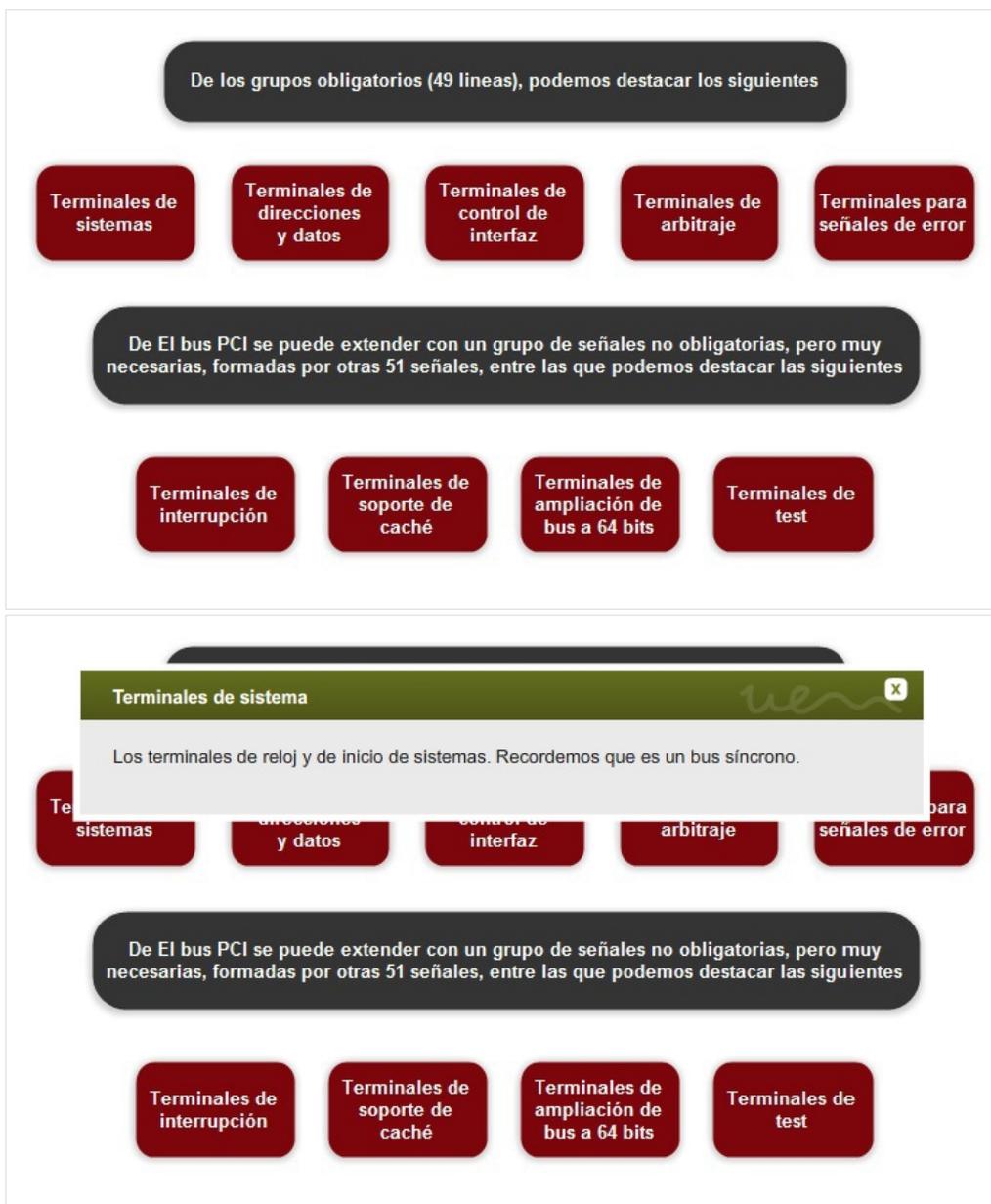
El esquema que se muestra sobre estas líneas podría corresponder con un sistema multiprocesador, donde se pueden conectar mediante adaptadores una o varias configuraciones PCI al bus de sistema del procesador. Al bus de sistema se conectan únicamente las unidades procesador/caché, la memoria principal y los adaptadores de PCI que, de nuevo mediante adaptadores, mantienen al PCI independiente de la velocidad del procesador y proporcionan la posibilidad de recibir y enviar datos rápidamente.



### Ejemplo: estructura del bus PCI

El bus PCI se divide en una serie de grupos funcionales, de los cuales algunos son obligatorios y otros opcionales.

Veamos algunos de esos grupos funcionales:



**Terminales de direcciones y datos**

Están formadas por varias líneas, de la cuales obligatoriamente son 32 líneas, solo 32, para datos y direcciones, por lo que están multiplexadas en el tiempo. Dentro de este grupo hay otras líneas que sirven para validar si las señales de datos y direcciones son correctas (señales de corrección de errores).

De El bus PCI se puede extender con un grupo de señales no obligatorias, pero muy necesarias, formadas por otras 51 señales, entre las que podemos destacar las siguientes

- Terminales de interrupción
- Terminales de soporte de caché
- Terminales de ampliación de bus a 64 bits
- Terminales de test

**Terminales de control de interfaz**

Controlan la temporización y la coordinación de las transmisiones de datos y direcciones por el bus.

De El bus PCI se puede extender con un grupo de señales no obligatorias, pero muy necesarias, formadas por otras 51 señales, entre las que podemos destacar las siguientes

- Terminales de interrupción
- Terminales de soporte de caché
- Terminales de ampliación de bus a 64 bits
- Terminales de test

**Terminales de arbitraje**

A diferencia de las otras líneas que son comunes para todos los dispositivos, hay dos señales (líneas) a cada uno de los dispositivos que se conectan a un bus PCI y no son compartidas por otros periféricos (son exclusivas de cada periférico). Sirven para asignar el uso de bus por parte del árbitro central y para solicitar la necesidad de uso del bus por parte del periférico.

De El bus PCI se puede extender con un grupo de señales no obligatorias, pero muy necesarias, formadas por otras 51 señales, entre las que podemos destacar las siguientes

- Terminales de interrupción
- Terminales de soporte de caché
- Terminales de ampliación de bus a 64 bits
- Terminales de test

**Terminales para señales de error**

Indican errores de paridad, etc.

Terminales de sistemas, direcciones y datos, control de interfaz, terminales de arbitraje, terminales para señales de error

De El bus PCI se puede extender con un grupo de señales no obligatorias, pero muy necesarias, formadas por otras 51 señales, entre las que podemos destacar las siguientes

- Terminales de interrupción
- Terminales de soporte de caché
- Terminales de ampliación de bus a 64 bits
- Terminales de test

**Terminales de interrupción**

Utilizadas por los terminales conectados que generan interrupciones. Al igual que las líneas de arbitraje, tampoco son compartidas sino que cada terminal tiene su propia señal de petición de interrupción.

De El bus PCI se puede extender con un grupo de señales no obligatorias, pero muy necesarias, formadas por otras 51 señales, entre las que podemos destacar las siguientes

- Terminales de interrupción
- Terminales de soporte de caché
- Terminales de ampliación de bus a 64 bits
- Terminales de test

**Terminales de soporte de caché**

Especialmente indicadas para dar soporte a la memoria caché. Es decir, dar soporte a los protocolos de coherencia de caché entre múltiples procesadores (lo veremos más adelante en la unidad de memoria).

De El bus PCI se puede extender con un grupo de señales no obligatorias, pero muy necesarias, formadas por otras 51 señales, entre las que podemos destacar las siguientes

- Terminales de interrupción
- Terminales de soporte de caché
- Terminales de ampliación de bus a 64 bits
- Terminales de test

**Terminales de ampliación de bus a 64 bits**

Como ya hemos visto, el bus PCI solo soporta 32 bits de manera obligatoria, por lo que se ha extendido a 64 bits. Al igual que las otras señales de datos y direcciones, son multiplexadas en el tiempo. En este grupo también se incluyen las señales de interpretación y validación, así como las de activación del modo de 64 bits.

De El bus PCI se puede extender con un grupo de señales no obligatorias, pero muy necesarias, formadas por otras 51 señales, entre las que podemos destacar las siguientes

- Terminales de interrupción
- Terminales de soporte de caché
- Terminales de ampliación de bus a 64 bits
- Terminales de test

**Terminales de test**

Utilizadas para procedimientos de test, según estándares de la IEEE.

- Terminales de sistemas
- direcciones y datos
- control de interfaz
- Terminales de arbitraje
- Terminales para señales de error

De El bus PCI se puede extender con un grupo de señales no obligatorias, pero muy necesarias, formadas por otras 51 señales, entre las que podemos destacar las siguientes

- Terminales de interrupción
- Terminales de soporte de caché
- Terminales de ampliación de bus a 64 bits
- Terminales de test

### Ejemplo: órdenes del PCI

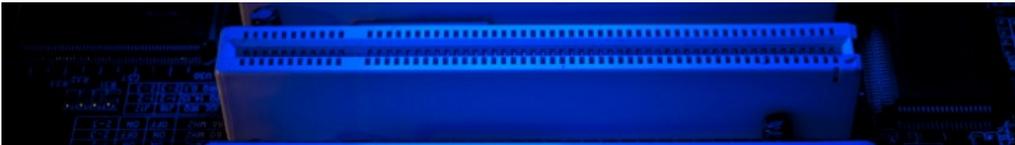
La función básica del bus PCI (igual que la de todos los buses) es la transferencia de datos. Al dispositivo que inicia la transferencia se le llama maestro y al otro interlocutor en la transferencia, esclavo.

La existencia de un maestro y un esclavo no implica que la información sea del maestro al esclavo, sino que puede ser bidireccional y, en la misma fase de asignación, haber movimiento de datos en ambos sentidos.

Cuando un maestro adquiere el control del bus, **especifica el tipo de transferencia**: reconocimiento de interrupción, ciclo especial, lectura de E/S, escritura en E/S, lectura de memoria, escritura en memoria, lectura múltiple en memoria, escritura e invalidación de memoria, actualización de configuración de dispositivos, lectura/escritura de configuración, ciclo de dirección dual, etc.

Gracias a todos estos tipos de transferencia:

- Se permiten transmisiones de mensajes de difusión. Es decir, de uno a muchos.
- Para las comunicaciones con dispositivos de entrada/salida, cada dispositivo conectado al bus tiene asociado un rango de direcciones privado que se utiliza para transmitir datos para o desde el dispositivo de E/S, permitiéndose mensajes de movimiento de datos por byte por palabra, e incluso de bloque.
- Para la comunicación con memoria se pueden utilizar uno o más ciclos de reloj de bus, permitiendo transmitir un dato por ciclo con solo una transferencia de dirección (datos consecutivos). El bus, al tener además soporte de caché, permite movimientos de datos de línea de caché si el protocolo está activo.
- Se permite invalidar direcciones de memoria, modificación de datos en caché, etc. Lo que permite la implementación de protocolos de coherencia de caché.
- Se permiten las órdenes de actualización de dispositivos, admitiéndose la transmisión de órdenes de configuración tanto para lectura como actualización, como por ejemplo durante la inicialización del sistema.



## Resumen

En este tema hemos visto, más en detalle, las configuraciones típicas que puede tener un bus y las diferentes maneras de implementarlas. Hemos destacado tanto las diferencias existentes entre ellas como las ventajas/inconvenientes de cada una de ellas.

Las características sobre las que hemos hecho desglose son:

- Elementos de diseño de un bus.
- Tipos de buses: dedicado / multiplexado.
- Métodos de arbitraje: centralizado / distribuido
- Temporización: síncrona / asíncrona.
- Modos de transferencias: Byte, palabra, línea o bloque, etc.
- Etc.

Hemos terminado el tema viendo unos de los buses más estandarizados, en cuanto a uso, en los ordenadores actuales: el bus PCI, sobre el que hemos analizado cada uno de los puntos que ya habíamos visto, buscando las similitudes y diferencias.