Tema 13: Jerarquía de buses y buses estándares

Departamento de
Informática e Ingeniería
de Sistemas

Universidad Zaragoza

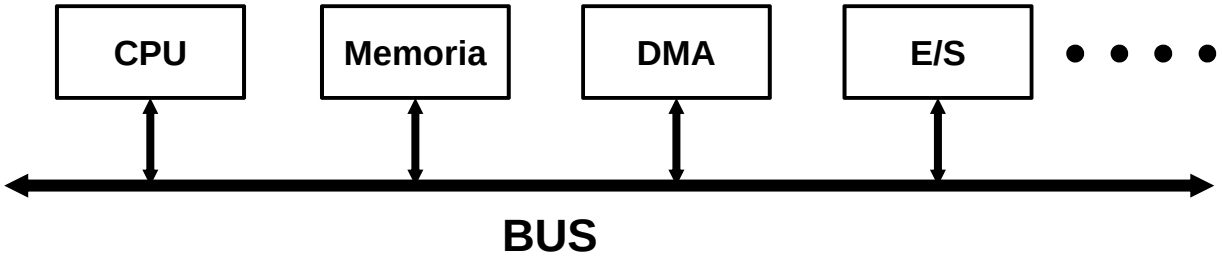
Jerarquía de buses y buses estándares

Jerarquía de buses y buses estándares

1. Limitaciones de los sistemas basados en un único bus
2. Jerarquía de buses
3. Jerarquía de buses en los PC
4. Ejemplos de buses estándar

Problemas de los sistemas basados en un único bus

- Conectar un gran número de dispositivos heterogéneos a un mismo bus provoca una *disminución del rendimiento global del sistema*
- Además, si los dispositivos han sido diseñados por compañías distintas a la que diseñó el bus pueden existir *incompatibilidades del bus con los dispositivos*

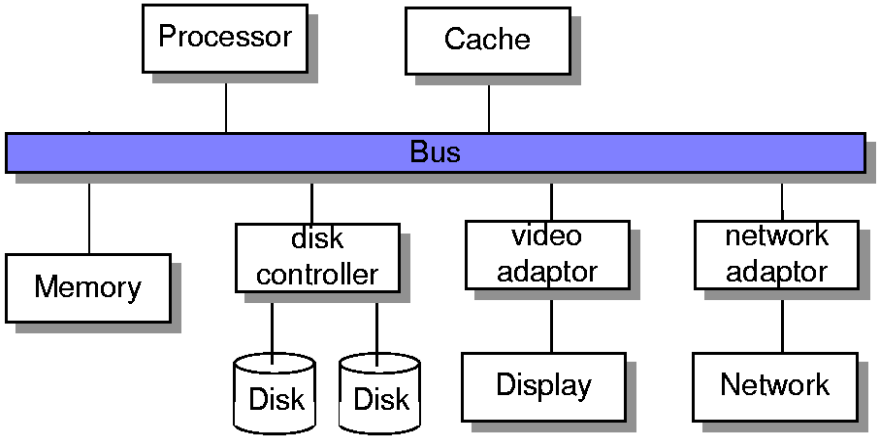


Problema 1: disminución del rendimiento

- Al conectar muchos dispositivos heterogéneos
  - Aumenta el ***retardo de propagación*** de las señales
    - El bus debe tener ***mayor longitud*** para soportar mayor número de dispositivos
    - Las ***señales de arbitraje*** (GRANT), si son encadenadas, debe propagarse a través de un mayor número de posibles masters
  - El bus puede actuar como un ***"cuello de botella"***
    - Si la demanda de la transferencia es mayor que la capacidad del bus los dispositivos deberán esperar mucho tiempo para poder transmitir
  - La ***diferencia de velocidad de los dispositivos*** afecta negativamente al rendimiento global
    - En el mismo tiempo que un dispositivo lento realiza una transferencia, uno rápido podría haber realizado miles de transferencias

Problema 1: disminución del rendimiento

- Procesador a 200 MHz (tiempo ciclo = 5 ns.)
- Ciclo medio por instrucción: CPI = 2 ciclos
- El procesador se conecta a la cache y al resto de dispositivos a través de un único bus del sistema
- El disco tiene un tiempo de acceso de 10 ms y una velocidad de transferencia de 10 MB/seg:
  - ¿Cuánto se tarda en realizar una transferencia de 512 KB de disco a memoria?
  - ¿Cuántas instrucciones podrían haberse ejecutado?



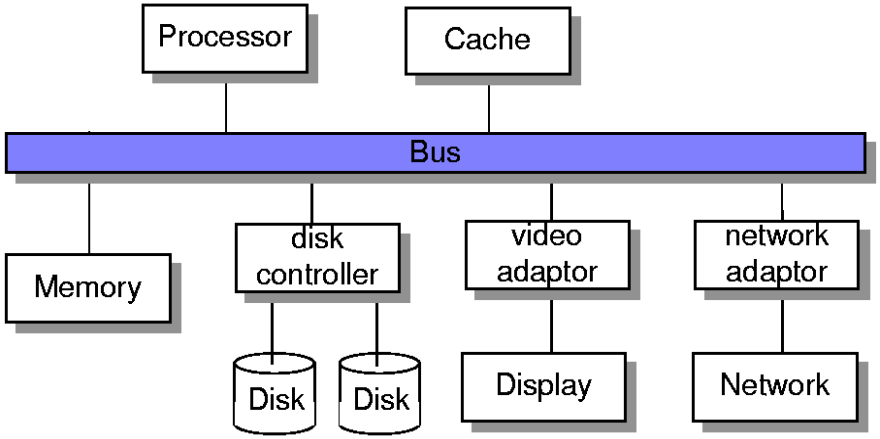
Problema 1: disminución del rendimiento

Queremos realizar una transferencia de 512 KB de disco a memoria

$$Tiempo = 10\ ms + \frac{512\ KB}{10.000\ KB/s} = 61{,}2\ ms$$

En ese tiempo, la CPU podría haber ejecutado:

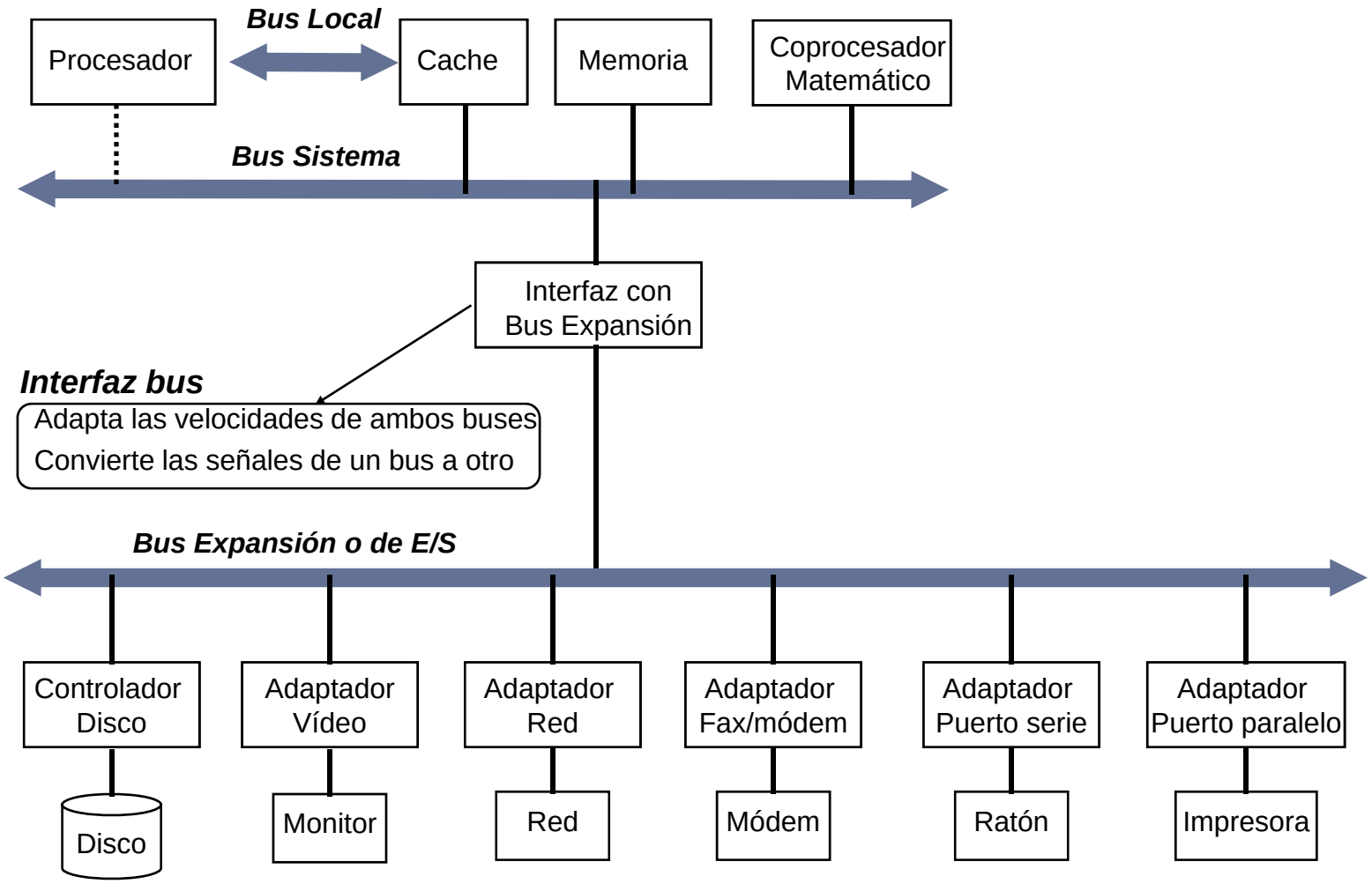
(0,0612 s) x (100 x 106 instruc /s) = 6,12 millones de instrucciones



Problema 2: incompatibilidades

- Un computador está compuesto por muchos elementos que, frecuentemente, han sido desarrollados por empresas distintas
- El bus del sistema es clave para el rendimiento y en general está diseñado para un procesador concreto
- Los fabricantes de dispositivos de E/S no quieren diseñar un dispositivo distinto para cada procesador
- ***Solución ideal:*** que todos los computadores utilizasen un estándar de bus uniforme
- ***Problema:*** un cada fabricante diseña sus propios buses optimizados para sus arquitecturas, por lo que es muy difícil que todos se pongan de acuerdo

Jerarquía de buses: buses local, del sistema y de expansión



Jerarquía de buses

- Bus Local y Bus del Sistema
  - Buses rápidos, cortos
  - Buses Propietarios (no estándares)
  - Optimizados para la arquitectura
  - Nº fijo de dispositivos de prestaciones conocidas
- Bus de expansión
  - Buses más largos y lentos
  - Bus abierto (estándar)
  - Accesible por el usuario
  - Nº indeterminado de dispositivos de distintas prestaciones

Ventajas de la jerarquía de buses

- El bus local entre el procesador y la cache aísla el tráfico de E/S del procesador
  - Se puede transferir información entre la memoria y la E/S sin interrumpir la actividad del procesador
- El bus de expansión reduce el tráfico en el bus del sistema
  - La transferencia entre cache y memoria principal se pueden realizar de forma más eficiente
- Se elimina el problema de la incompatibilidad
  - El bus local y del sistema suelen ser propietarios (no estándar) y están optimizados para cada arquitectura particular
  - Los buses de expansión son buses estándares o abiertos (ISA, PCI, PCI-Express, etc.)

Interfaces, o puentes entre buses

- Adaptar las velocidades de ambos buses
  - El bus del sistema es, en general, más rápido que el bus de expansión
    - El adaptador debe actuar como buffer de almacenamiento intermedio para evitar la pérdida de datos
- Conversión de líneas del bus
  - Los buses pueden tener utilizar señales distintas para realizar funciones similares
  - **Ejemplos:**

    1) **Líneas de operación distintas**

    Bus sistema: Una única línea RD/WR*

    Bus expansión: Dos líneas READ - WRITE separadas

    2) **Líneas multiplexadas y dedicadas**

    Bus sistema: líneas de dirección/datos multiplexadas (AD0, AD15, A16-A19*)*

    Bus expansión: líneas de dirección y datos dedicadas (A0-A19, D0-D15)

    3) **Distinto número de líneas de datos**

    Bus sistema: D0-D31

    Bus expansión: D0-D15

    ⇒ El adaptador debe dividir cada transferencia de 32 bits en dos transferencias de 16 bits

    4) **Distinto mecanismo de sincronización**

    Bus sistema: síncrono

    Bus expansión: asíncrono

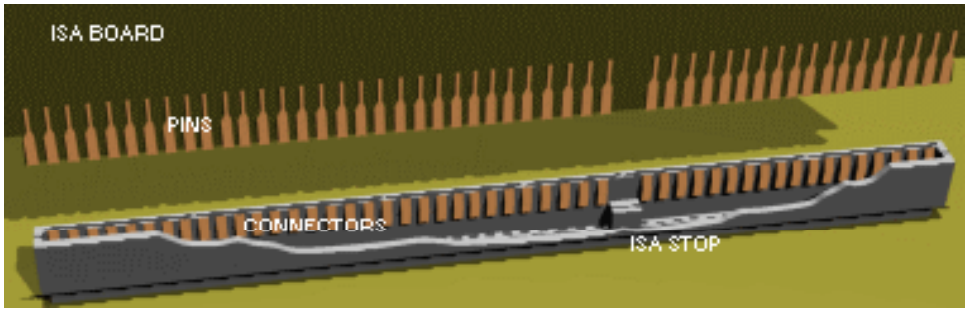
    ⇒ El adaptador deberá comunicarse de forma síncrona con el bus del sistema y de forma asíncrona con el bus de expansión

    ⇒ El adaptador deberá ser capaz las señales de sincronización adecuadas dependiendo del bus con el que se comunique

    Etc.

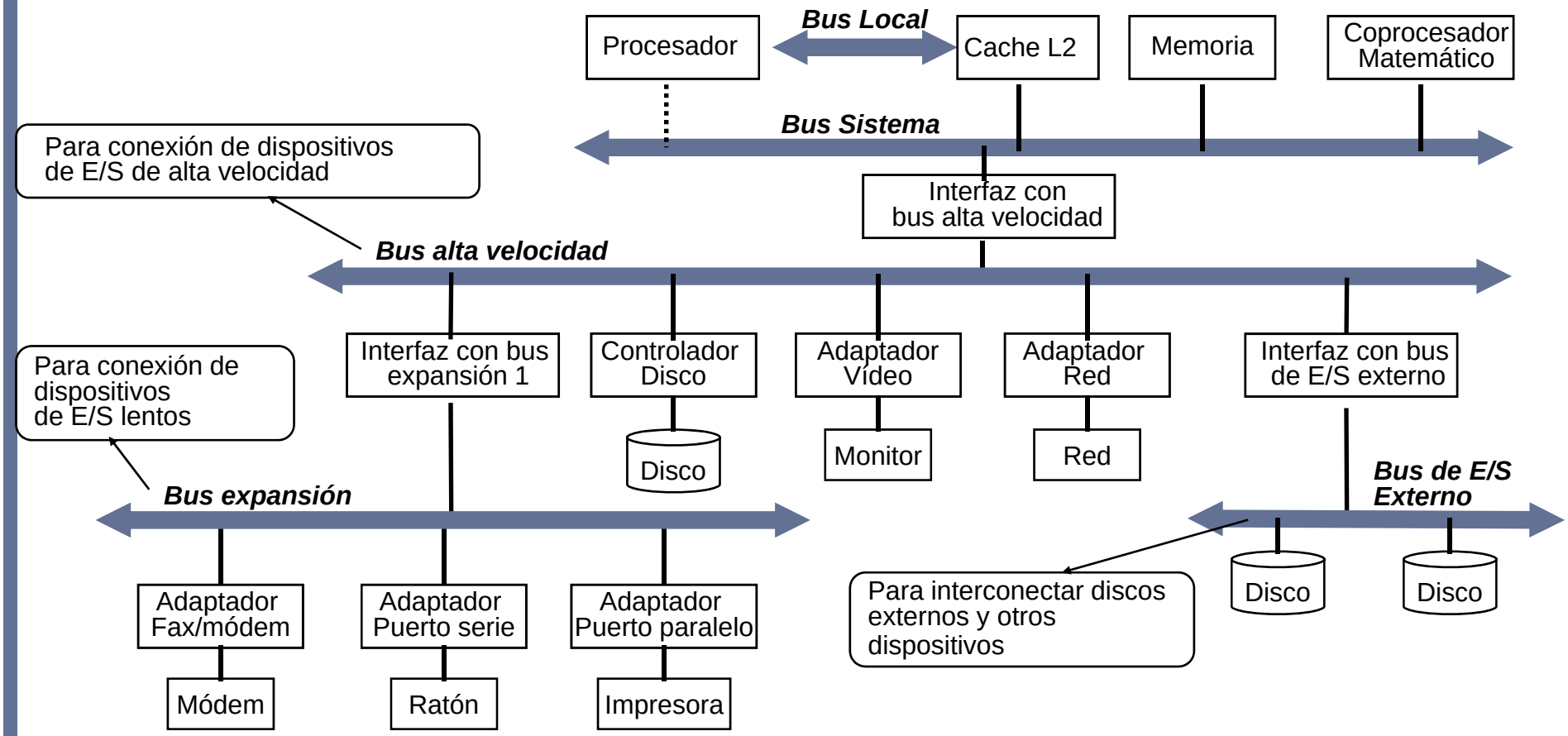
Especificaciones de un bus estándar

- Las especificaciones de un bus estándar deben estar perfectamente definidas y recogidas en un documento de estandarización
- En las especificaciones se distinguen varios niveles:
  - **Nivel eléctrico**
    - Valores de las tensiones de alimentación
    - Límites de valores eléctricos de las señales lógicas
      - P. ej. 1 lógico → de 0,2 V a 0,5 V;
        0 lógico → de -0,2 V a -0,5 V
  - **Nivel mecánico**
    - Forma y tamaño de los conectores
    - Número de contactos del conector
    - Número de dispositivos que soporta



  - **Nivel lógico**
    - Funciones a cada señal (bus de datos, bus de direcciones, bus de control)
    - Asignación de señales a los contactos del conector
  - **Nivel de temporización básico**
    - Protocolos de sincronización empleados
  - **Nivel de arbitraje**
    - Protocolos de arbitraje empleados

Jerarquía de buses de un PC

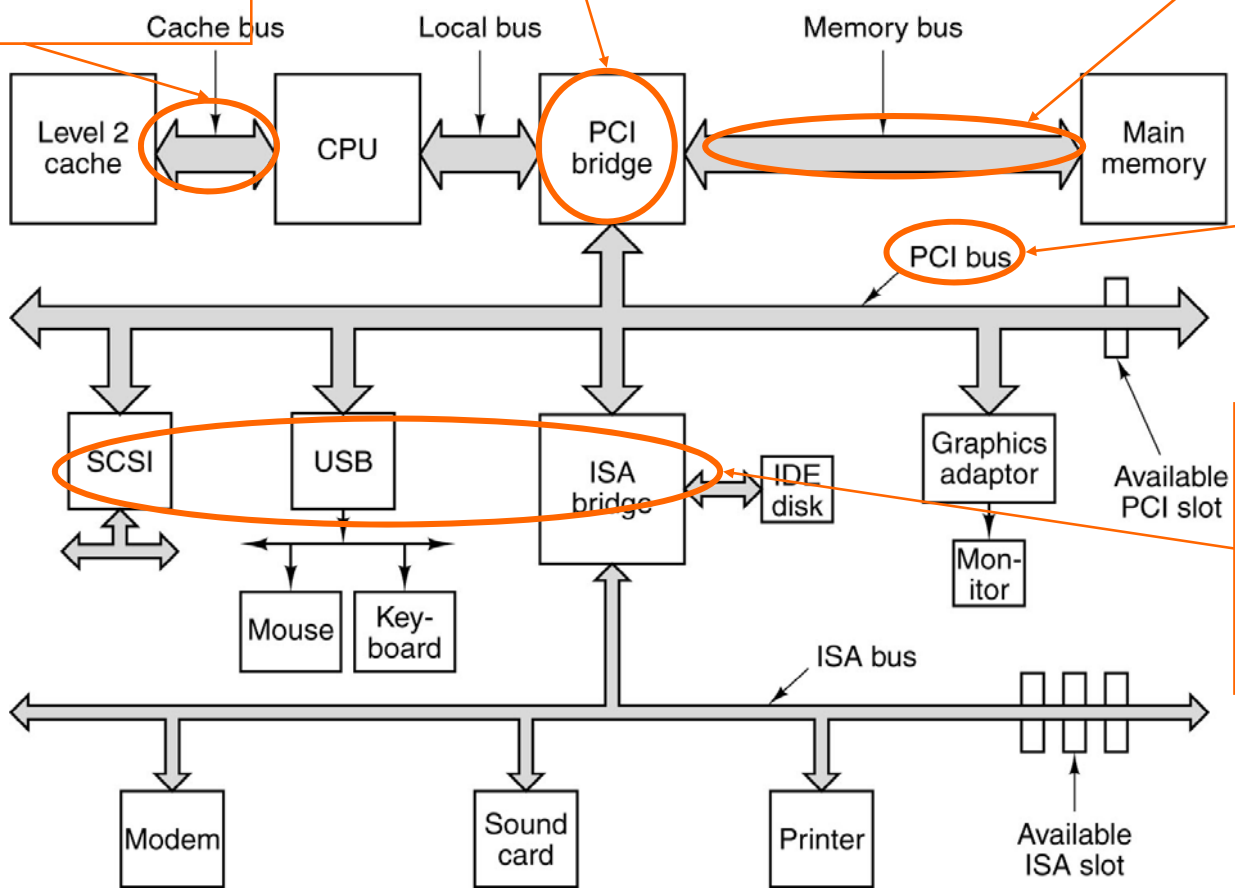


Ejemplo 1: jerarquía de buses en un PC pentium

Bus de caché: permite al procesador acceder a la caché mientras otros dispositivos acceden a memoria

Adaptador del bus PCI (puente): conecta el bus del sistema con el bus PCI

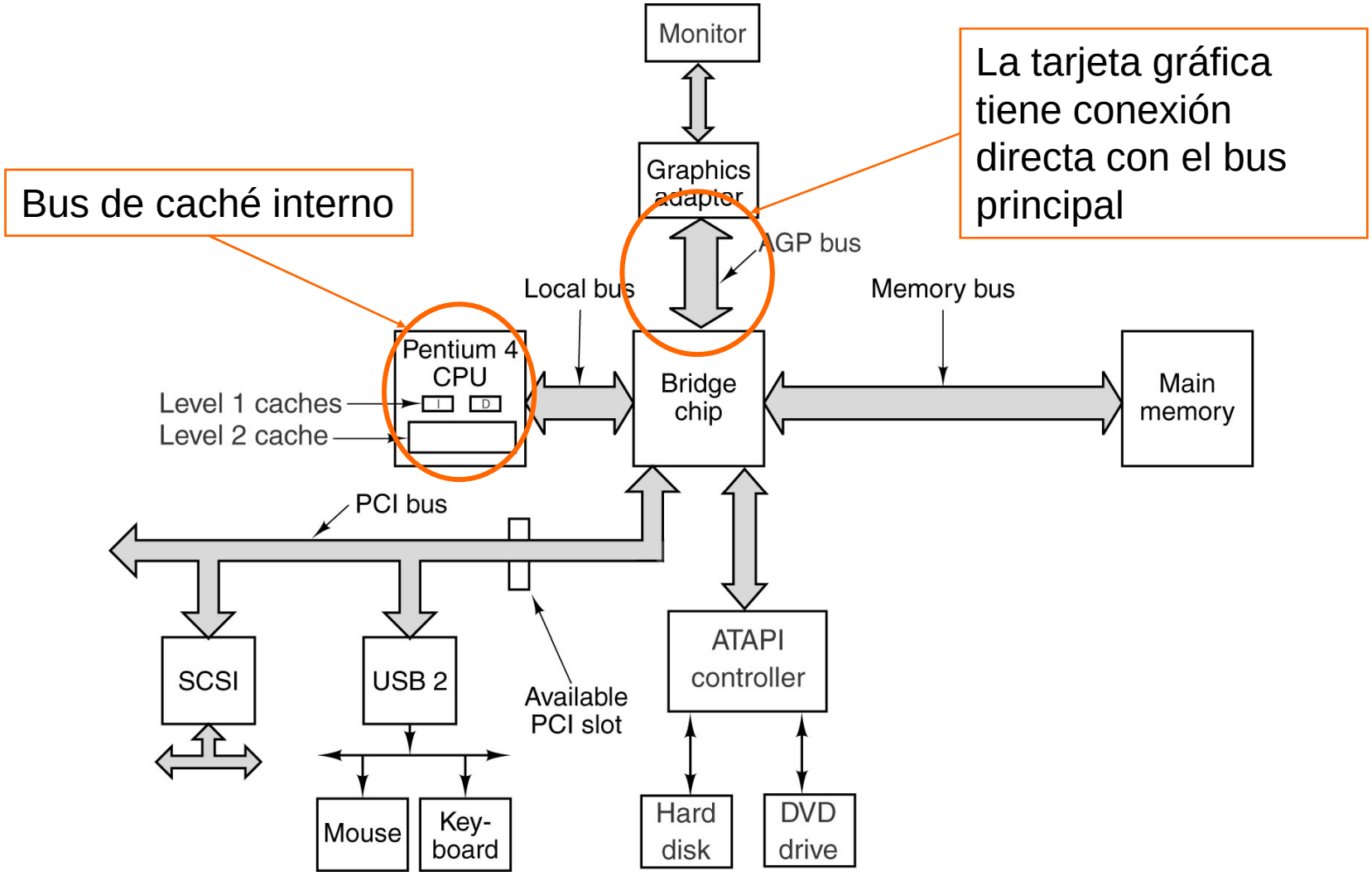
Bus de memoria: permite acceder a la memoria principal



Bus PCI para conectar dispositivos de alta velocidad

Adaptadores a otros buses estándar:
SCSI: entrada salida
USB: conexión serie
ISA: dispositivos lentos

Ejemplo 2: jerarquía de buses en un Pentium IV



Jerarquía de buses en un PC x86

Comparación de las prestaciones de los buses de un PC Pentium

Bus	***Ancho datos***	***Frec. reloj***	***Velocidad transmisión***	
Sistema	64	133 MHz	1.064 Gbytes/s	
PCI (V 2.0)	32	33 MHz	132 Mbytes/s	Buses de expansión de alta velocidad
PCI (V 2.1)	64	66 MHz	528 Mbytes/s	
EISA	32	8 MHz	32 Mbytes/s	Buses de expansión de baja velocidad
ISA	16	8 MHz	5-8 Mbytes/s	
SCSI-1	8	5 MHz	4 Mbytes/s	Buses de E/S Externos
SCSI-2	16/32	10 MHz	20/40 Mbytes/s	
IDE	16	1.6 MHz	3.18 Mbytes/s	“Buses” para conexión de discos
EIDE	16	5.5 MHz	10.6 Mbytes/s	
USB 2	Serie	--	60 Mbits/s	Bus serie

Se consigue mejorar el ancho de banda aumentando el número de líneas de datos y la frecuencia de reloj

Jerarquía de buses en un PC Pentium

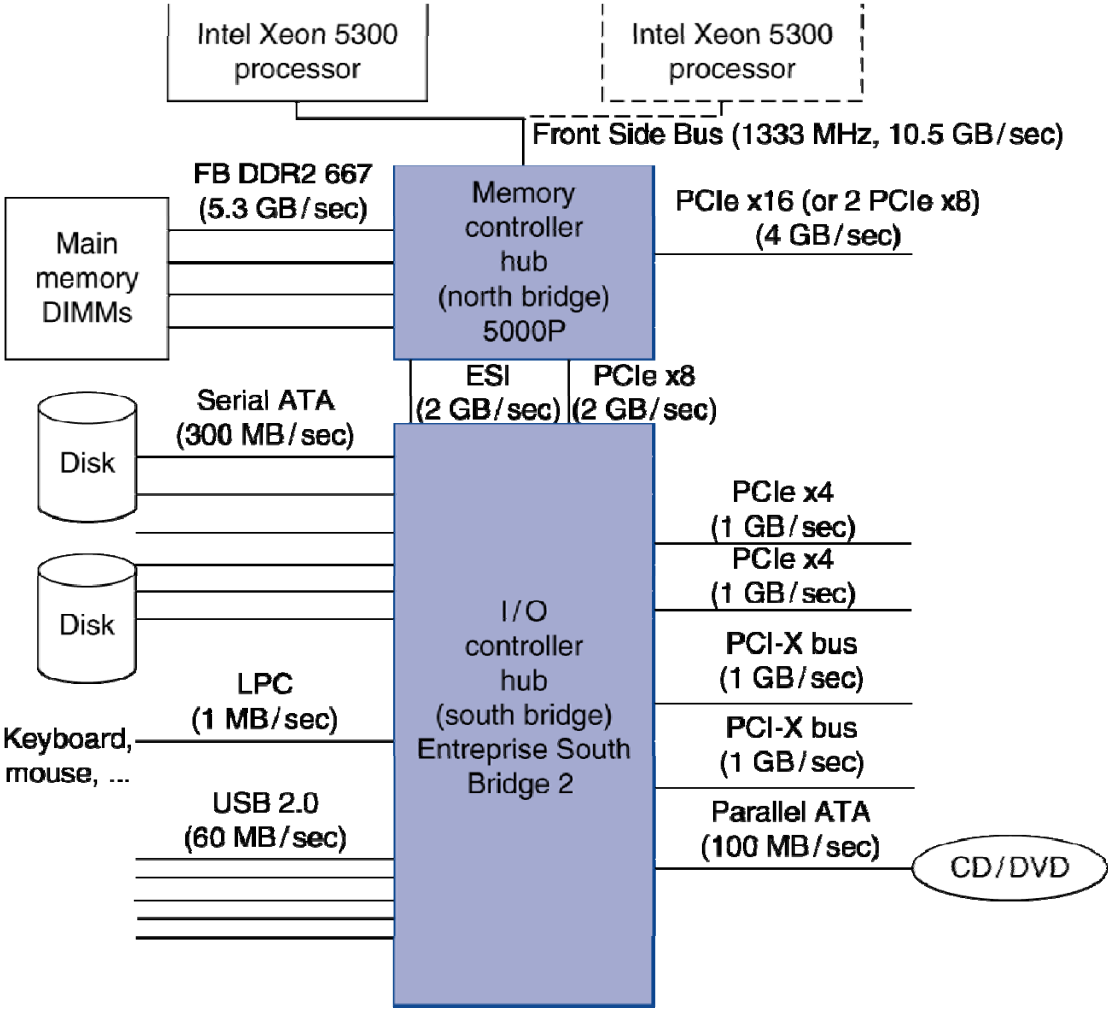
Buses más recientes

Bus	Año	Líneas de datos	Freq. reloj	Ancho de banda	Uso
USB 2.0	2000	Serie	---	480Mbit/s	periféricos
AGP	1996	32	66 MHz	264 MB/s	Tarjetas gráficas
AGP 8X	2002	32	533 MHz	2GB/s	Tarjetas gráficas
PCI-X	1998	64	133 MHz	1GB/s	Dispositivos rápidos
PCI-Express	2004	De 1 a 16 conexiones serie dobles	---	250MB/ conexión	Dispositivos rápidos, tarjetas gráficas

Tendencia:

- Buses on-chip síncronos
- Conexiones serie de alta velocidad para conectar dispositivos rápidos
  - Razón: *tiempo de desplazamiento relativo* (*skew time*)

Typical x86 PC I/O System



Ejemplo de bus estándar: bus PCI

Bus PCI (Peripheral Component Interconnect Bus, 1993)

- Bus de expansión diseñado para el i80486 y Pentium
  - **Bus de datos:**
    - Versión 2.0: 32 bits de datos
    - Versión 2.1: 64 bits de datos
  - **Bus de direcciones:** 32 bits (4 GB direccionables)
  - **Ciclo de reloj:**
    - Versión 2.0: 33 MHz
    - Versión 2.1: 66 MHz
  - **Velocidad de transferencia máxima:**
    - Versión 2.0: 132 Mbytes/s
    - Versión 2.1: 528 Mbytes/s
  - **Protocolo de bus:** semisíncrono
  - **Protocolo de arbitraje:** centralizado en estrella
  - **Otras características**
    - Hasta 16 slots de expansión
    - Soporte para gran variedad de controladores de dispositivos de E/S de alta velocidad
      - Vídeo, Sonido, Redes alta velocidad, Adaptadores SCSI, etc.
    - Soporte Plug-and-Play (conecta y listo)
      - Tarjetas controladoras autoconfigurables (línea de interrupción, dirección de E/S, etc.)

Ejemplo de bus estándar: bus PCI

Protocolo de sincronización del bus PCI

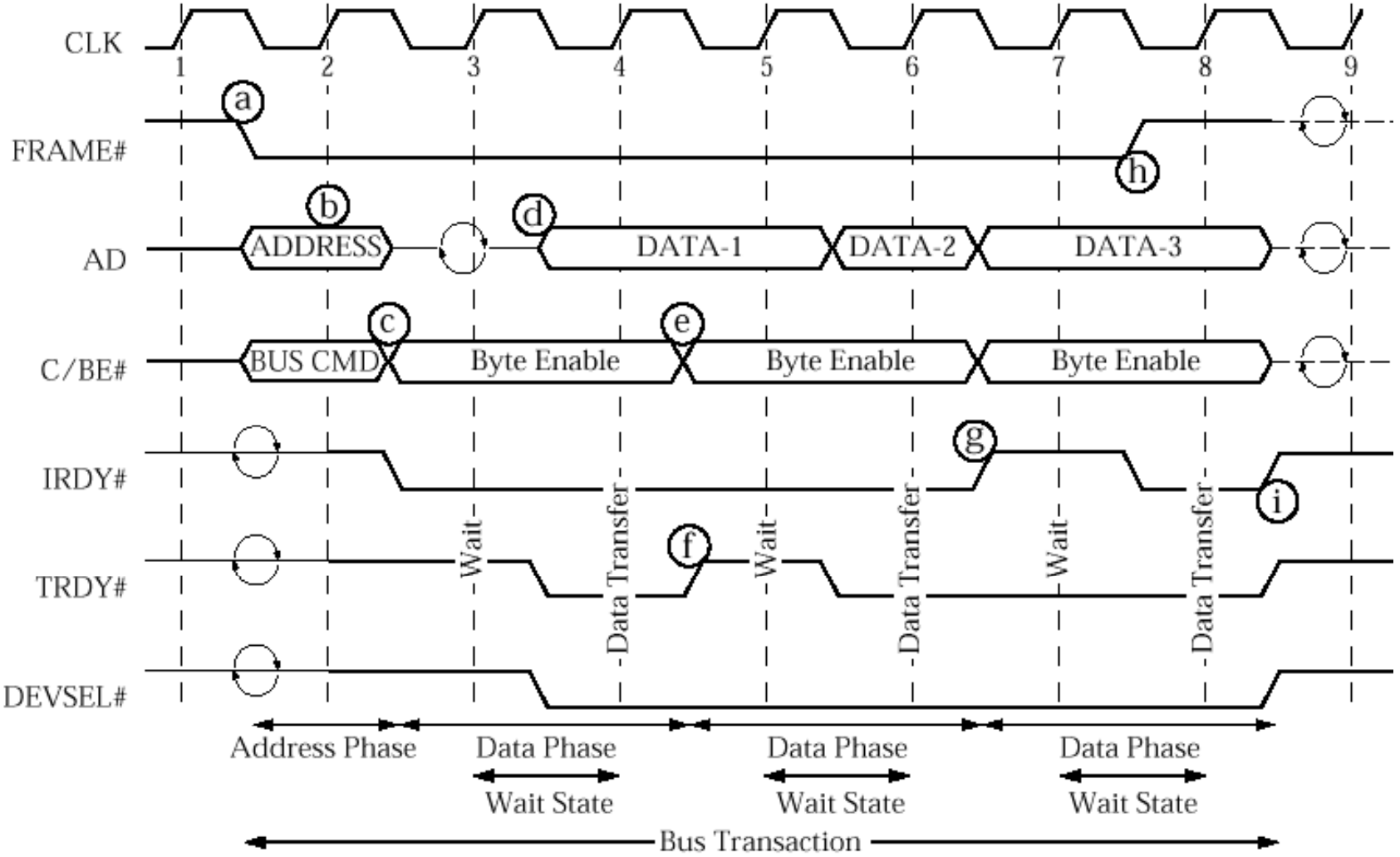
- Protocolo semisíncrono
- Modos de transferencias
  - **Modo palabra**
    - Se transmite una única palabra a una dirección de memoria o E/S específica
    - Las palabras pueden ser de 1, 2, 3 ó 4 bytes
  - **Modo bloque**
    - Se transfiere un bloque de datos desde/hacia posiciones de memoria consecutivas, a partir de una posición inicial
- Lineas del bus
  - **CLK:** señal de reloj
  - **AD0-AD31:** Líneas multiplexadas de datos y direcciones
  - **C0*-C3*/BE0*-BE3*:** Líneas multiplexadas de *orden y byte activo (byte enabled)*
    - **Orden (C0*-C3*):** la activa el master durante el primer ciclo de la transferencia para especificar el tipo de transferencia a realizar:
      - Lectura de memoria, escritura de memoria, lectura de E/S, escritura de E/S, etc.
    - **Byte activo (BE0*-BE3*):** la activa el master durante la transferencia de datos para indicar qué líneas del bus transportan los datos
      - BE0* activada $\Rightarrow$ AD0-AD7 transporta datos
      - BE1* activada $\Rightarrow$ AD8-AD15 transporta datos
      - BE2* activada $\Rightarrow$ AD16-AD23 transporta datos
      - BE3* activada $\Rightarrow$ AD24-AD31 transporta datos

Ejemplo de bus estándar: bus PCI

- Líneas del bus (cont.)
  - **FRAME*:** Señal para indicar el comienzo y la duración de una transferencia
    - La activa el master al poner la dirección en el bus para indicar el comienzo de la transferencia
    - Si la transferencia es *modo bloque* la señal se mantiene activa durante toda la transferencia del bloque y se desactiva al transferir la última palabra
  - **DEVSEL*:** Señal de *dispositivo seleccionado* (device selected)
    - La activa el slave para indicar que ha reconocido su dirección
  - **TRDY*:** Señal de *slave preparado (target ready)*
    - La activa el slave al inicio de la transferencia junto con DEVSEL*
    - El slave desactiva esta señal en caso de que no pueda completar la transferencia en un solo ciclo de reloj
  - **IRDY*:** Señal de *master preparado (initiator ready)*
    - La activa el master al inicio de la transferencia
    - El master desactiva esta señal en caso de que no pueda completar la transferencia en un solo ciclo de reloj, por ejemplo en caso de que el master se quede temporalmente sin capacidad de almacenamiento

Ejemplo de bus estándar: bus PCI

Ejemplo: Lectura de un bloque de 3 palabras



Ejemplo de bus estándar: bus PCI

a El master realiza las siguietes acciones
- Pone la dirección en el bus (AD0-AD31)
- Indica el tipo de operación a realizar (C0*-C3*)
- Activa FRAME* para indicar el inicio de la transferencia

b El slave descodifica y reconoce su dirección en el bus

c El master deja libre el bus de datos e indica en BE0*-BE3* qué líneas transportarán los datos y activa IRDY* para indicar que está preparado para recibir el 1er dato

d Cuando el slave tiene el 1er dato válido realiza las siguientes acciones
- Activa DEVSEL* para indicar ha reconocido su dirección
- Pone el dato en el bus y activa TRDY* para indicar que el dato está en el bus

e El master lee el dato
- A partir de aquí, mientras esté la señal FRAME* activada, se leerá un dato en cada ciclo de reloj (siempre que el slave no desactive TRDY*)

f El slave necesita más de 1 ciclo para poner el 2º dato en el bus
- Desactiva TRDY* hasta que tiene el nuevo dato preparado

g El master no está preparado para recibir el 3er dato
- Desactiva IRDY* hasta que está preparado para poder recibir correctamente el siguiente dato

h Transferencia del último dato
- El master desactiva FRAME* para indicar el final de la transferencia del bloque

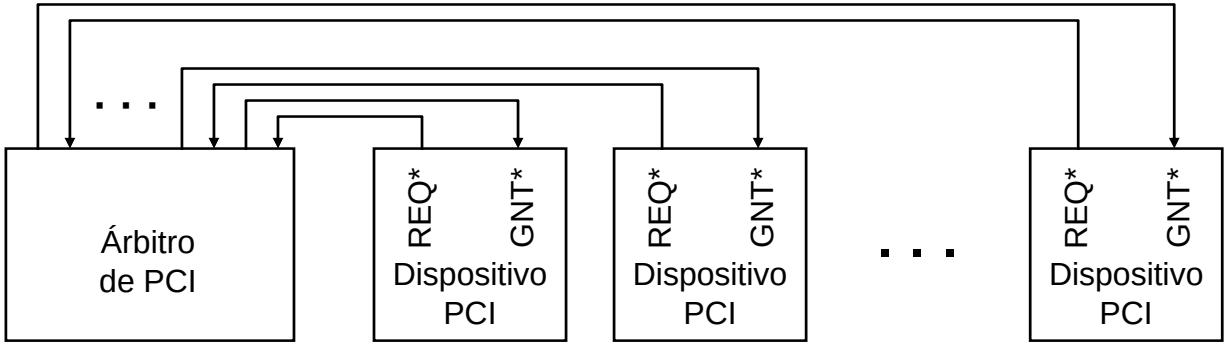
i El master dsactiva IRDY* y el slave desactiva TRDY* y DEVSEL*
- El bus queda libre para la siguiente transferencia

Ejemplo de bus estándar: bus PCI

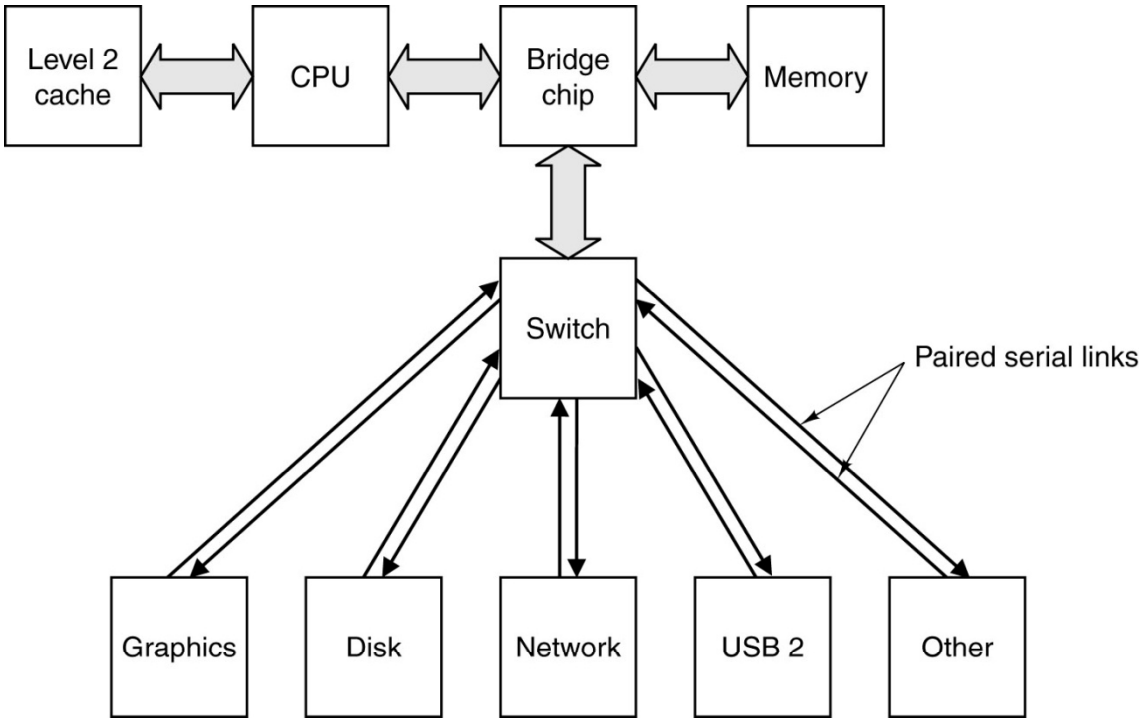
Protocolo de arbitraje del bus PCI

- Protocolo centralizado en estrella
  - Cada máster se conecta al árbitro mediante 2 líneas dedicadas
    - REQ: petición del bus
    - GNT: concesión del bus
  - La especificación de PCI no indica un algoritmo de arbitraje particular
    - Pueden utilizarse distintos tipos de algoritmos
      - FIFO
      - Prioridad fija
      - Prioridad variable
      - Rotativo
      - etc.

Líneas de arbitraje del bus PCI



PCI Express



- Conexiones serie dedicadas
- Cada conexión proporciona 250 Mbs para entrada de datos y otros tantos para salida
- Se pueden utilizar hasta 16 conexiones para un mismo dispositivo

Conclusiones

- Los sistemas basados en un único bus no funcionan bien cuando se deben conectar muchos dispositivos heterogéneos
- Incluyendo una jerarquía de buses se puede trabajar con muchos dispositivos de una forma eficiente
- Los dispositivos lentos se conectan a buses de expansión que a su vez pueden comunicarse con el procesador atravesando uno o más interfaces
- Con este esquema todos los dispositivos lentos tienen asignado el mismo ancho de banda que un dispositivo rápido
- Además los buses de expansión pueden ser estándar, con lo que cualquier compañía puede desarrollar componentes para estos buses
- Por otro lado los buses de caché y de sistema pueden ser propietario y estar optimizados para un determinado procesador