

# Entrada / Salida Buses e Interconexión

Agustín Fernández, Josep Llosa, Fermín Sánchez

Estructura de Computadors II  
Departament d'Arquitectura de Computadors  
Facultat d'Informàtica de Barcelona



## Índice

- Introducción
- Tipos de líneas de un bus
- Buses Serie/Paralelos
- Buses Síncronos/Asíncronos
- Buses de ciclo Completo/Partido
- Rendimiento
- Arbitraje del bus
- Detección y corrección de errores
- Jerarquía de buses
- Buses del PC

## Introducción

### Definiciones básicas

- Bus
  - Canal de comunicación compartido
  - Conjunto de "cables" que permite conectar múltiples elementos
- La velocidad de transmisión depende de la longitud, del soporte (cable, fibra, pista...) y de la lógica de control
- Dueño del bus (*bus master*): elemento que inicia la comunicación
- Esclavo (*slave*) elemento que responde a la comunicación
- Dispositivos fuente y destino
- Ciclo de bus: operación básica de un bus que permite realizar una transferencia de un dato entre dos dispositivos
- Transacción: Agrupación de varias transferencias en una sola operación



## Introducción

### Clasificación de los buses

- En función del grado de paralelismo
  - Bus Paralelo
  - Bus Multiplexado
  - Bus Serie
- En función de la sincronización
  - Bus síncrono
  - Bus asíncrono
- En función de la temporización
  - Bus de ciclo completo
  - Bus de ciclo partido



## Introducción

- El bus proporciona:
  - Una **especificación** que define un zócalo de conexión
  - Un **protocolo** de comunicación: conjunto de reglas que gobiernan el comportamiento de los dispositivos conectados al bus
- Para que un dispositivo se conecte a un bus debe tener:
  - **Compatibilidad física** (zócalo)
  - **Compatibilidad lógica** (entender el protocolo de comunicación)
- El dispositivo debe tener
  - Un **interfaz de bus**: circuitería responsable de cumplir con la especificación física del bus
  - Un **controlador hardware**: ejecuta las operaciones internas propias del dispositivo
  - Un **controlador software** (*driver*): rutinas de tratamiento para las interrupciones asociadas al dispositivo



## Introducción

- Ejemplo: Lector/Grabador de CD
  - **Interfaz de bus**: Permite a la circuitería del controlador hardware conectarse a un bus determinado: SCSI o EIDE
  - **Controlador hardware**: motor de rotación + brazo de posicionamiento + láser de lectura + registros internos
  - **Controlador software**: Permite usar el CD bajo LINUX o Windows, por ejemplo



## Introducción

- **Ventajas** de comunicarse mediante un bus
  - Versatilidad y bajo coste
  - Una vez definida la forma de interconexión, es muy simple añadir nuevos dispositivos
- **Inconvenientes**
  - Cuello de botella en el rendimiento del computador
  - Factores físicos
    - limitan la longitud de la línea
    - limitan el número de dispositivos que podemos conectar
    - impiden usar el bus a cualquier velocidad



## Tipos de líneas de un bus

- **Líneas de Datos (y direcciones)**
  - Llevan información desde el elemento fuente al destino
    - datos
    - direcciones
    - órdenes complejas
  - Líneas de dirección y datos separadas en algunos buses
    - Ejemplo: Bus ISA
  - Líneas de dirección y datos multiplexadas (en tiempo) en algunos buses: reducción del número de líneas
    - Ejemplo: BUS PCI
  - Líneas de dirección multiplexadas (en tiempo) en algunos buses
    - Ejemplo: memorias DRAM actuales, fila y columna
- **Líneas de Control.**
  - Se usan para:
    - hacer peticiones
    - indicar reconocimientos
    - indicar qué tipo de información pasa por las señales de datos
- **Líneas de alimentación**



## Tipos de líneas de un bus

- **Líneas de Datos** (determinan el rendimiento)
  - Determinan anchura de la información a transmitir
  - Métrica de rendimiento: **ancho de banda**
    - $\text{Ancho de Banda} = \text{Frecuencia} * \# \text{líneas} / 8$
    - Suele medirse en Kbytes/s, Mbytes/s o Gbytes/s
    - Potencias de 10
  - Ejemplo: Bus local Pentium 4, 64 líneas a 400 Mhz  
 $(400 * 10^6 \text{ ciclos/s} * 64 \text{ bits/ciclo} / 8 \text{ bits/byte}) / 10^9 \text{ bytes/Gbyte} = 3,2 \text{ Gbytes/s}$
- **Líneas de Direcciones** (acceso)
  - El procesador ha de acceder a los registros del controlador del bus y el controlador de bus a los registros de los dispositivos
  - $\# \text{líneas @ del controlador de bus}$  menor que  $\# \text{líneas @ del procesador}$
  - La placa base incluye la circuitería de decodificación



## Tipos de líneas de un bus

- **Líneas de Control** (protocolo)
  - líneas de sincronización y temporización (reloj)
  - Líneas de interfaz: arbitraje, control de acceso, líneas de petición/respuesta en buses asíncronos. Implementan el protocolo de bus en general
  - Líneas de fiabilidad: p.e. los bits de paridad para detectar errores
  - Señal de L/E de la memoria
  - Señales para indicar el tamaño de los operandos
- **Líneas de Alimentación**
  - Alimentan dispositivos de poco consumo (ej: teclado y ratón) y dispositivos de consumo medio (ej.: modem, placa gráfica, etc. conectados mediante zócalo en la placa base)
  - Dispositivos de mayor consumo se conectan directamente a la fuente de alimentación (ej.: DVD, CD-ROM, disco, las nuevas placas gráficas 3D de última generación, ...)



## Buses Paralelos

- **Bus paralelo:**
  - Tiene múltiples líneas de datos.
  - Los datos se transmiten de forma simultánea por todas las líneas usando un cable para cada bit del dato transmitido
  - Ejemplo: PCI, ATA
- **Bus Multiplexado:**
  - Es un caso particular de bus paralelo, en el que algunas líneas de comunicación se comparten (en tiempo) entre varias señales en distintos momentos
  - Ejemplo: transmisión de fila/columna en el bus de direcciones de una memoria DRAM
- **Ventajas de la transmisión paralela**
  - Un bus paralelo es mucho mas rápido que un bus serie a la misma frecuencia, ya que utiliza varios cables de transmisión de datos en vez de uno (pero se produce diafonía entre los cables)
  - Se puede aumentar la velocidad de transmisión aumentando el número de líneas
  - Ejemplo: ISA (16 bits) -> EISA (extensión a 32 bits)



## Buses Serie

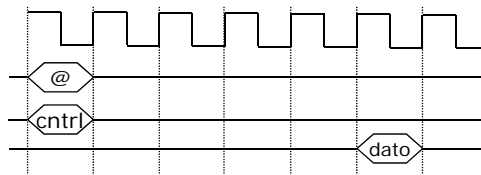
- **Bus serie:**
  - Tiene una única línea de datos
  - Los datos se transmiten bit a bit
  - Ejemplo: PCIe, SATA, USB
- **Ventajas de la transmisión serie**
  - Incrementar la frecuencia de reloj en un bus serie es mucho más fácil que en uno paralelo
  - El coste del cable serie es inferior
    - Menor número de líneas
    - Conectores con menos pins
    - Aislamiento de menor coste
  - Permite cables de mayor longitud
    - No tiene problemas de skew (las señales no han de llegar al mismo tiempo)
  - Más inmune a las interferencias



## Buses Síncronos

- **Bus Síncrono**

- Incorpora una señal de reloj en las líneas de control, que no es necesariamente el reloj del procesador (menor frecuencia)
- La comunicación se realiza utilizando un protocolo basado en ese reloj. Una transferencia puede durar varios ciclos
- Ejemplo de una lectura sencilla:



- Este protocolo puede realizarse con un pequeño autómata
- El protocolo está perfectamente determinado
  - puede funcionar a gran velocidad
  - requiere poca lógica de interficie

## Buses Síncronos

- Los dispositivos conectados al bus leen y escriben datos en el bus mediante *buffers*
- El momento de lectura y escritura de los buffers coincide con un flanco de reloj según el protocolo del bus
- Existen señales de control que permiten comunicarse a los dos dispositivos involucrados en la transferencia indicando cuándo han realizado una determinada acción
- **Problemas de los buses síncronos:**
  - Todos los dispositivos conectados al bus han de funcionar a la misma velocidad: la del dispositivo más lento
  - Debido a los problemas del sesgo (*skew*) de reloj, los buses síncronos no pueden ser a la vez rápidos y largos
- Ejemplo: Los buses procesador-memoria suelen ser síncronos porque los dispositivos involucrados están muy cercanos, son pocos y están preparados para trabajar a velocidades de reloj altas
- Ejemplo: bus PCI

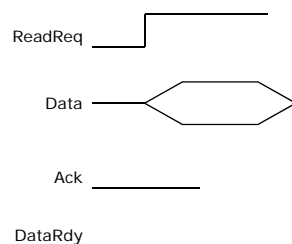
## Buses Asíncronos

- **Bus Asíncrono**
  - Carece de señal de reloj
  - El reloj es reemplazado por dos líneas de control temporales, que indican cuándo están listos los dos dispositivos involucrados en la transferencia
  - Admite la conexión de una variedad muy grande de dispositivos
  - La longitud del bus no necesita tener en cuenta los problemas de sincronización y el sesgo del reloj
  - Usa un protocolo de *handshaking* para coordinar la transmisión de datos entre el emisor y el receptor
  - El protocolo *handshaking* es una secuencia de pasos a seguir, de forma que el emisor y el receptor sólo avanzan al paso siguiente cuando ambos están de acuerdo
  - Ejemplo: Bus ISA



## Buses Asíncronos

- **Protocolo Handshaking:**
  - Ejemplo de comunicación dispositivo E/S-Memoria asíncrona



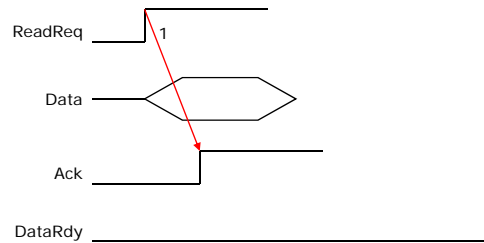
0: El dispositivo (**dueño del bus**) realiza la petición **activando ReadReq** y poniendo en el bus de datos la **dirección** a la que quiere acceder





## Buses Asíncronos

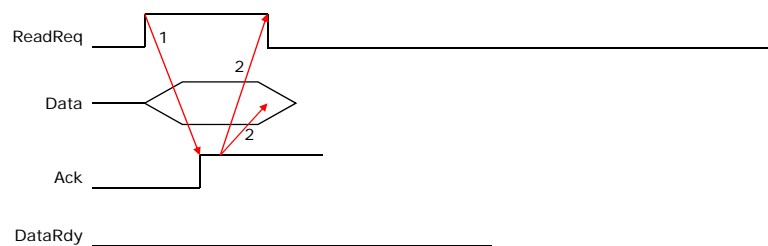
- Protocolo Handshaking:
  - Ejemplo de comunicación dispositivo E/S-Memoria asíncrona



1: Cuando la memoria (**dispositivo esclavo**) detecta el ReadReq, lee la dirección del bus de datos y **activa** el Ack para avisar que ha recibido la dirección

## Buses Asíncronos

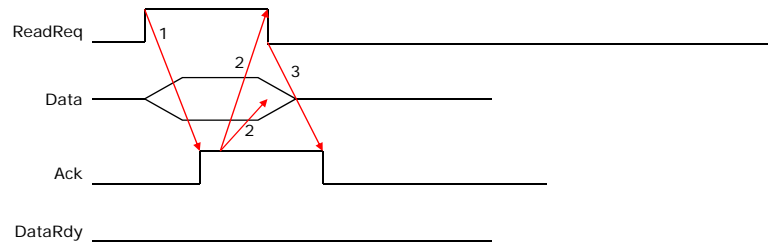
- Protocolo Handshaking:
  - Ejemplo de comunicación dispositivo E/S-Memoria asíncrona



2: El dispositivo de E/S detecta el Ack, **desactiva** el ReadReq y libera el bus de datos

## Buses Asíncronos

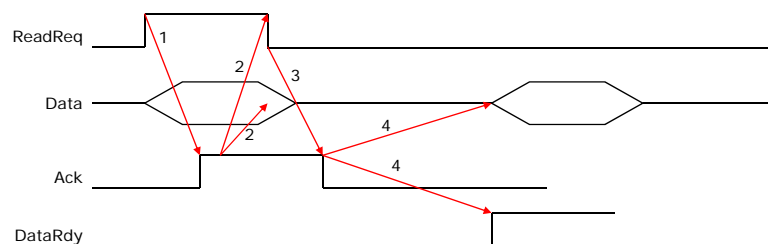
- Protocolo Handshaking:
  - Ejemplo de comunicación dispositivo E/S-Memoria asíncrona



3: La memoria detecta que se ha desactivado el ReadReq y **desactiva el Ack**

## Buses Asíncronos

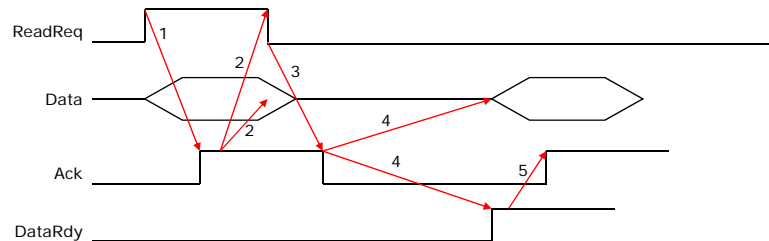
- Protocolo Handshaking:
  - Ejemplo de comunicación dispositivo E/S-Memoria asíncrona



4: Cuando la Memoria ha leído el dato solicitado, lo pone en el bus de datos y **activa la señal DataRdy**

## Buses Asíncronos

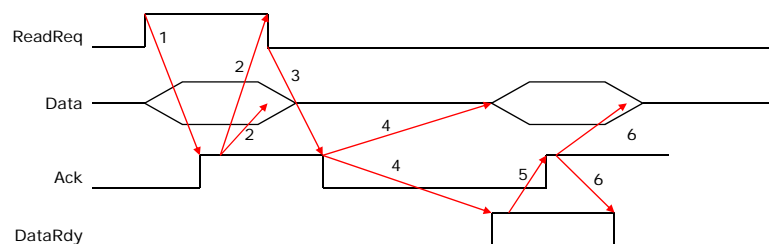
- Protocolo Handshaking:
  - Ejemplo de comunicación dispositivo E/S-Memoria asíncrona



5: El dispositivo de E/S detecta el DataRdy, lee los datos y activa la señal Ack para indicar que ya tiene los datos

## Buses Asíncronos

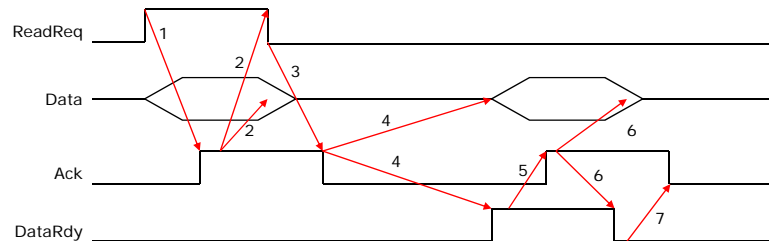
- Protocolo Handshaking:
  - Ejemplo de comunicación dispositivo E/S-Memoria asíncrona



6: La Memoria detecta el Ack, desactiva el DataRdy y libera el bus de datos

## Buses Asíncronos

- Protocolo Handshaking:
  - Ejemplo de comunicación dispositivo E/S-Memoria asíncrona



7: El Dispositivo **detecta** la **desactivación** del DataRdy y **desactiva** el Ack. Ya puede iniciarse otra transacción

## Buses Ciclo Completo / Partido

- **Bus de ciclo completo**
  - Se ocupa el bus durante todo el tiempo que dura la transacción
  - Se pierden ciclos debido a la velocidad de respuesta de los dispositivos
  - Ejemplo: Bus PCI
- **Bus de ciclo partido**
  - Se divide el tiempo en ranuras (*time slots*)
  - En cada ranura se envía un mensaje
  - La transferencia elemental se compone de dos ranuras (ida y vuelta)
  - El bus puede ser utilizado por otros dispositivos entre dos ranuras (por ejemplo, mientras dura la latencia de memoria)
  - Ejemplo: Bus PCIe

## Preguntas

- ¿Qué tipo de bus es mejor?
  - ¿serie o paralelo?
  - ¿síncrono o asíncrono?
  - ¿ciclo partido o ciclo completo?
- ¿Qué tipo de bus es capaz de transmitir más datos por unidad de tiempo?
- ¿Qué tipo de bus se usa más?



## Rendimiento

- Los buses son uno de los **cuellos de botella** más importantes del computador:
  - Los procesadores son cada vez más rápidos
  - Las tarjetas gráficas y de sonido proporcionan cada vez mejores prestaciones (necesitan más ancho de banda)
- **Métricas de rendimiento del bus**
  - **Tiempo de respuesta:** Se puede reducir el tiempo de acceso al bus simplificando el camino de comunicación
  - **Velocidad de E/S:** Es necesario aumentar el **ancho de banda**. Se puede hacer añadiendo buffers y transmitiendo bloques de información → aumenta el tiempo de acceso al bus

**OBJETIVOS INCOMPATIBLES ENTRE SÍ**

- La necesidad de soportar varios tipos de dispositivos con latencias y velocidades de transferencia muy diferentes, contribuye a complicar aún más el diseño del bus



## Rendimiento

- Factores fundamentales que afectan al Ancho de Banda
  - Bus síncrono o asíncrono
  - Temporización utilizada
- Otros factores
  - a) Anchura del bus de datos
  - b) Líneas de datos y direcciones separadas o multiplexadas
  - c) Transferencias por bloques
- Cada una de estas alternativas incrementa el rendimiento, pero a costa de tener (en cada caso):
  - más líneas (a)
  - mayor complejidad (b)
  - mayor tiempo de respuesta (c)



## Rendimiento

### Estrategias para aumentar el rendimiento del bus

- Especialización:
  - Diseñar buses especiales para dispositivos que requieren gran ancho de banda (AGP y Fire Wire)
- Evitación:
  - Un dispositivo lento repercute menos en la velocidad del sistema cuanto menos se use
  - Evitar transacciones en los buses aumentando el tamaño de las caches integradas (L1, L2 y recientemente L3)
  - Incorporar coprocesamiento en los dispositivos para reducir el tráfico
- Eliminación
  - Concentrar funciones en la placa base, liberando al bus



## Arbitraje del bus

- El Bus es un recurso COMPARTIDO.
- PREGUNTAS
  - ¿A quién se concede el uso del bus?
    - ⇒ Política de selección
  - ¿En qué condiciones el dispositivo que tiene el control del bus debe cederlo a otro dispositivo que lo solicita?
    - ⇒ Política de cesión
  - ¿Quién se encarga de hacer que estas políticas se cumplan?
    - ⇒ Árbitro del bus
- Arbitraje del bus
  - Proceso por el cual se selecciona el siguiente dispositivo que tendrá el control del bus
  - La selección debe tener en cuenta las necesidades de los diferentes dispositivos



## Arbitraje del bus

- Políticas de selección
  - Estática: Ciclos "repartidos" previamente entre los *bus masters*. Típico de sistemas de tiempo real (necesidades conocidas de antemano)
  - Dinámica
    - Por prioridad: Cada *bus master* tiene preasignada una prioridad distinta
    - Por equitatividad: Todos los *bus masters* tienen la misma prioridad. Asignación aleatoria o rotativa. Típico de multiprocesadores
- Políticas de cesión
  - Bajo petición: Un *bus master* posee el control del bus, aunque no lo necesite, mientras otro dispositivo no lo solicite (ejemplo: CPU y DMA)
  - Cesión al terminar la transacción: El *bus master* devuelve el control al árbitro una vez acabada la transacción
  - Expulsión: Un *bus master* con mayor prioridad expulsa del bus al *bus master* que tiene el control del bus. Sólo tiene sentido cuando se transfieren bloques de información



## Arbitraje del bus

- **Tipos de arbitraje**
  - **Centralizado:** Un único árbitro elige al *bus master* que va a controlar el bus
  - **Distribuido:** Todos los dispositivos participan en la selección del siguiente *bus master* que va a controlar el bus



## Arbitraje del bus

- **Arbitraje centralizado**
  - El dueño del bus puede ser el procesador u otros dispositivos conectados al bus (por ejemplo, controladores de DMA)
  - Los sistemas sencillos (microcontroladores) usan este esquema
  - Cuando el árbitro es el procesador, también puede actuar como dueño del bus.
    - **Problema:** el procesador está implicado en todas las transacciones del bus, ya que es el responsable de iniciarlas
  - Ejemplo: bus PCI
- **Arbitraje distribuido**
  - Los dispositivos conectados al bus dialogan entre sí para escoger al siguiente dueño del bus
  - Cada dispositivo tiene capacidad para iniciar una transferencia y ser dueño del bus
  - Ofrece alta fiabilidad, ya que el arbitraje no depende de un único dispositivo.
  - Ejemplo: bus SCSI. Cada dispositivo tiene un identificador (número). El dispositivo con mayor identificador obtienen el bus.





## Arbitraje del bus

- **Protocolo de arbitraje centralizado**
  - El dispositivo que desee usar el bus
    - Debe realizar una petición de bus (*bus request*)
    - y esperar la concesión del bus (*bus grant*)
    - Después de la concesión, el dispositivo puede usar el bus
    - Cuando acabe, debe indicar al árbitro del bus que ya no lo necesita (*liberación*)
  - Si hay más de un dispositivo solicitando el bus, el árbitro debe decidir a quién se le concede
  - Factores a tener en cuenta:
    - **Prioridad de bus.** Cada dispositivo tiene asignada una prioridad. El dispositivo más prioritario debe ser atendido en primer lugar
    - **Imparcialidad.** Todos los dispositivos tienen derecho a acceder al bus, incluso los menos prioritarios
      - **Problema: Inanición.** Un dispositivo poco prioritario podría no acceder nunca al bus si los dispositivos prioritarios lo usan con mucha frecuencia



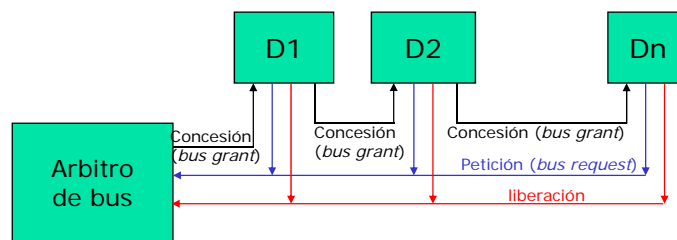
## Arbitraje del bus

- **Tipos de arbitraje centralizado**
  - Centralizado en serie (*Daisy chain*)
  - Centralizado en paralelo
- **Tipos de arbitraje distribuido**
  - Distribuido por autoselección
  - Distribuido por detección de colisión



## Arbitraje del bus

- Arbitraje Centralizado en serie (*daisy chain arbitration*)
  - Arbitraje por lista encadenada
  - La prioridad se establece por la posición en la cadena (el primero es el más prioritario)
  - Es muy simple, pero **no respeta el principio de imparcialidad** (los últimos dispositivos de la cadena podrían no acceder nunca al bus si los primeros lo requieren continuamente)
  - La velocidad del bus está limitada por el *daisy chain*
  - Ejemplo: Bus VME



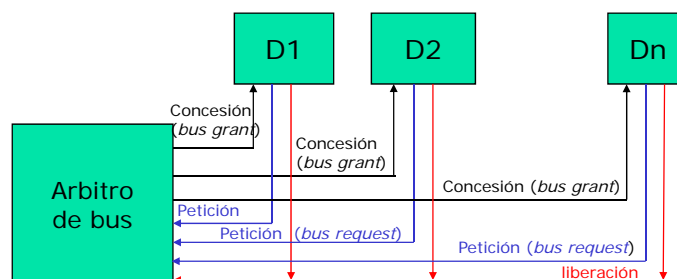
Entrada/Salida: Buses e interconexión

35



## Arbitraje del bus

- Arbitraje Centralizado en paralelo
  - Hay múltiples líneas de petición de bus que van al árbitro de forma independiente
  - El árbitro selecciona uno de los dispositivos que han pedido el bus y notifica al elegido que ahora es el dueño del bus
  - Ejemplo: bus PCI



Entrada/Salida: Buses e interconexión

36



## Arbitraje del bus

- **Arbitraje Distribuido por autoselección**
  - Hay múltiples líneas de petición de bus
  - Cuando un dispositivo quiere el bus, lo solicita indicando su código
  - Los propios dispositivos, examinando el bus, determinan cuál es el más prioritario y deciden si pueden utilizar el bus o no
  - No se necesita un árbitro centralizado
  - Ejemplo: bus NuBus de Apple Macintosh II



## Arbitraje del bus

- **Arbitraje Distribuido por detección de colisión**
  - Los dispositivos solicitan el bus de forma independiente
  - Se produce una colisión si existen múltiples peticiones simultáneas
  - Una vez detectada la colisión, el árbitro selecciona quién usa el bus de entre los dispositivos que han provocado la colisión
  - Ejemplo: Ethernet



## Arbitraje del bus

- ¿Qué esquema de arbitraje es mejor?
- La idoneidad de los diferentes esquemas de arbitraje está determinada por varios factores
  - La expansibilidad del bus
  - El número de dispositivos
  - La longitud del bus
  - La rapidez del arbitraje
  - El grado de imparcialidad deseado



## Detección y corrección de errores

- Buses paralelo
  - Error de transmisión: por ruido, interferencias, etc. Se detecta y corrige con códigos redundantes. Ej: bit de paridad
  - Error de direccionamiento
    - Intento de acceso a posición de memoria o registro inexistente. En buses asíncronos se evita limitando la duración máxima del ciclo de bus (*time out*)
    - Intento de escritura en una posición protegida (ej: ROM). Requiere hardware especial para la detección
  - Error de control
    - El árbitro cede el control del bus a más de un dueño. Se puede detectar como error de transmisión
    - Se sobrepasa el tiempo preestablecido antes de que el árbitro ceda el control del bus a un dispositivo que lo ha solicitado
- Buses serie
  - Error de transmisión: El formato de los bloques a enviar y el protocolo de comunicación deben estar diseñados para detectar y corregir (si es posible) los errores. Ej: CRC



## Jerarquía de buses

- **Nivel 0:** Buses internos de los circuitos integrados
- **Nivel 1:** Interconectan los distintos componentes de una placa de circuito impreso
- **Nivel 2:** Interconectan las distintas placas de un módulo
- **Nivel 3:** Interconectan los distintos módulos de un computador
- **Nivel 4:** Buses paralelos que conectan dispositivos de E/S con el computador
- **Nivel 5:** Buses serie que conectan dispositivos de E/S con el computador



## Jerarquía de buses

- **Buses de nivel 0**
  - Buses internos de los circuitos integrados
  - El usuario no tiene acceso
  - Son de uso específico
  - Pueden ser muy anchos (256 ó 512 bits)
  - Ejemplo: buses que conectan las caches integradas del procesador entre sí o con el procesador



## Jerarquía de buses

- **Buses de nivel 1**
  - Interconectan los componentes de un circuito impreso
  - Son cortos (decenas de cm.)
  - Los dispositivos suelen conectarse directamente al bus (sin necesidad de amplificadores de señal)
  - Suelen ser síncronos y tener un único dueño, que establece la temporización de las transferencias (hace de árbitro)
  - Son poco flexibles, porque se diseñan para minimizar el coste de las placas maximizando su rendimiento
  - Ejemplo: Bus procesador-memoria



## Jerarquía de buses

- **Buses de nivel 2**
  - Interconectan las distintas placas (tarjetas) de un módulo, formando el "panel posterior" (*backplane*)
  - La mayoría de los computadores actuales tienen un sólo módulo (placa base), por lo que no existen buses de nivel 3 y el bus de nivel 2 se convierte en el **bus del sistema**
  - Longitud inferior al metro
  - Un centenar de señales
  - Son buses estándar, ya que deben conectar dispositivos de muy distintos tipos
  - Pueden ser de ciclo completo o de ciclo partido, en función del caudal de datos a transmitir
  - Ejemplos de ciclo completo: ISA, EISA y PCI
  - Ejemplos de ciclo partido: PCIe, FutureBus, Megabus de Honeywel y SBI de DEC



## Jerarquía de buses

- **Buses de nivel 3**
  - Interconectan los diversos módulos de un computador
  - Hoy en día hay pocos sistemas que contengan varios módulos (grandes computadores), por lo que coinciden los buses de nivel 2 y 3
  - Longitud hasta 10 metros
  - Los buses "largos" deben ser tratados como líneas de transmisión:
    - Se necesitan dispositivos terminales
    - Presentan problemas de ruido (*crosstalk*). Este problema se agudiza cuando cambian "simultáneamente" todos los bits de direcciones o de datos
    - Para evitar el ruido se apantallan los cables y se incrementa la corriente de las señales
    - El tiempo de propagación de las señales es crítico
  - Es frecuente que se requiera el uso de repetidores o *buffers* para interconectar los módulos
  - Ejemplos: BXP-bus de Intel y Unibus de DEC.



## Jerarquía de buses

- **Buses de nivel 4**
  - Bus paralelo para la conexión de dispositivos de E/S
  - Se pueden considerar cuatro situaciones:
    - Controladores de los periféricos se conectan directamente al bus de sistema (de nivel 2 ó 3)
    - Se dispone de un bus especial de E/S de propósito general (de nivel 4) al que se conectan TODOS los controladores de los periféricos
      - Ej: Unibus en la arquitectura VAX o PCI y EISA en PCs
    - Se diseña un bus específico para conectar algunos periféricos determinados. Este bus cuelga del bus de sistema. Estos buses no suelen estar normalizados (aplicaciones muy específicas)
    - Se diseña un bus general para conectar (algunos) periféricos según una norma preestablecida. Este bus cuelga del bus de sistema.
      - Ej: Puerto paralelo Centronics para conexión de impresoras o bus SCSI para conexión de discos



## Jerarquía de buses

- **Buses de nivel 5**
  - Bus serie para la conexión de:
    - Dispositivos de E/S
    - Otros computadores
  - Pueden cubrir "grandes" distancias
  - La información se empaqueta en bloques, con cabecera y cola que permiten identificar al destinatario y posibles errores en la transmisión
  - Tradicionalmente se han empleado para conectar dispositivos de baja velocidad (teclado, ratón, etc.)
  - Pueden ser simples (una línea, transmisión *semiduplex* –en ambos sentidos alternativamente-) o dobles (dos líneas, una de ida y otra de vuelta (transmisión *full-duplex* –en ambos sentidos simultáneamente-)
  - El desarrollo de la redes locales y dispositivos que requieren transmitir mucha información (cámaras digitales, escáners) ha aumentado mucho la velocidad de transmisión
  - Ej: USB (480 Mb/s), Fire Wire (800 Mb/s), Fast Ethernet 100 Mb/s



## Jerarquía de Buses: Un PC actual

- **Buses integrados (nivel 0)**
- **Bus local y bus de memoria (nivel 1)**
  - Conecta el procesador con memoria principal y con el resto de buses
  - 33 MHz en el 80486, 50/66MHz en los Pentium, 66/100MHz en los PentiumII, 100/133 MHz en los Pentium III, ..., hasta los 400MHz en el K7, 800 MHz en el Pentium 4 y 1 GHz en el K8.
- **Bus de Expansión (de propósito general) (nivel 2)**
  - Conecta procesador con periféricos de E/S
  - PCI/ISA en los 90, hoy sólo PCI y PCI express (PCIe)
- **Buses de propósito específico (niveles 4 y 5)**
  - SCSI (80's) o IDE (90's) para HD,
  - AGP (90's) para tarjetas gráficas
  - USB y Fire Wire (2004) para multimedia



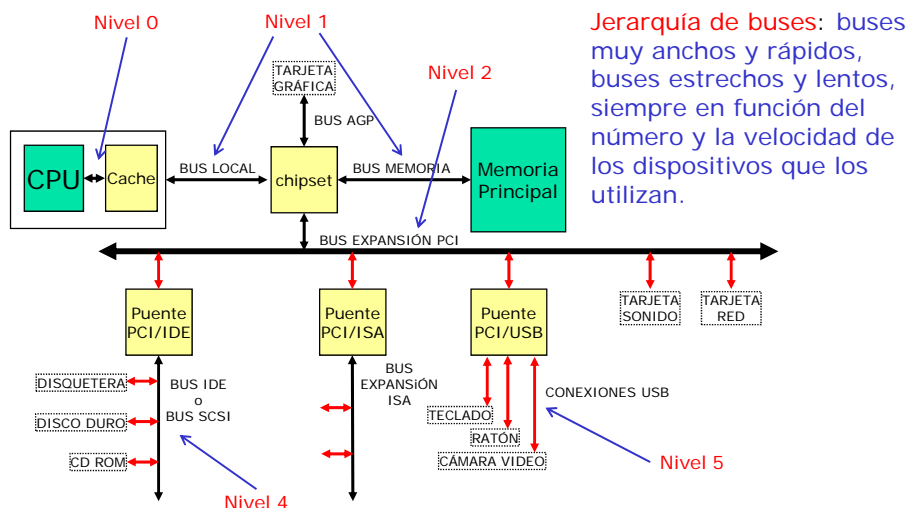


## Jerarquía de Buses: Un PC actual

	Bus Local	Bus de Expansión	Bus Dedicado
Principal cualidad que debe tener	Velocidad	Estandarización	Especialización
Responsabilidad que asume	Rendimiento	Versatilidad	Funcionalidad
Rango de Frecuencias más usuales	200-600 MHz	8-100 MHz	Muy heterogéneo
Ejemplos Comerciales	100x4 y 133x4 (Pentium 4) 133x2 (K7)	PCI ISA	AGP EIDE, SCSI FireWire, USB



## Jerarquía de Buses: Un PC actual (2004)



## Jerarquía de Buses: Un PC actual (2005)

Jerarquía de buses: buses muy anchos y rápidos, buses estrechos y lentos, siempre en función del número y la velocidad de los dispositivos que los utilizan.

