Tema 8. Buses: conexión E/S con CPU/Memoria

ESTRUCTURA DE COMPUTADORES

Grupo de Arquitectura de Computadores (GAC)

Objetivos del tema

En este tema se estudiará:

- La interconexión de los distintos componentes de un computador
- Algunas de las principales soluciones de interconexión basadas en buses
- Aspectos básicos de diseño del sistema de E/S de un computador

Bibliografía

Básica

- *Computer Organization and Design: The hardware/software interface (3rd ed.)*. David A. Patterson and John L. Hennessy. Morgan Kaufmann Publishers, Inc. 2007
- *Computer Architecture: A Quantitative Approach (3rd or 4th ed.)*. John L. Hennessy y David A. Patterson. Morgan Kaufmann Publishers, Inc. 2004/2006
- *Organización y arquitectura de computadores (7th ed.)*. William Stallings. Prentice Hall. 2006

Complementaria

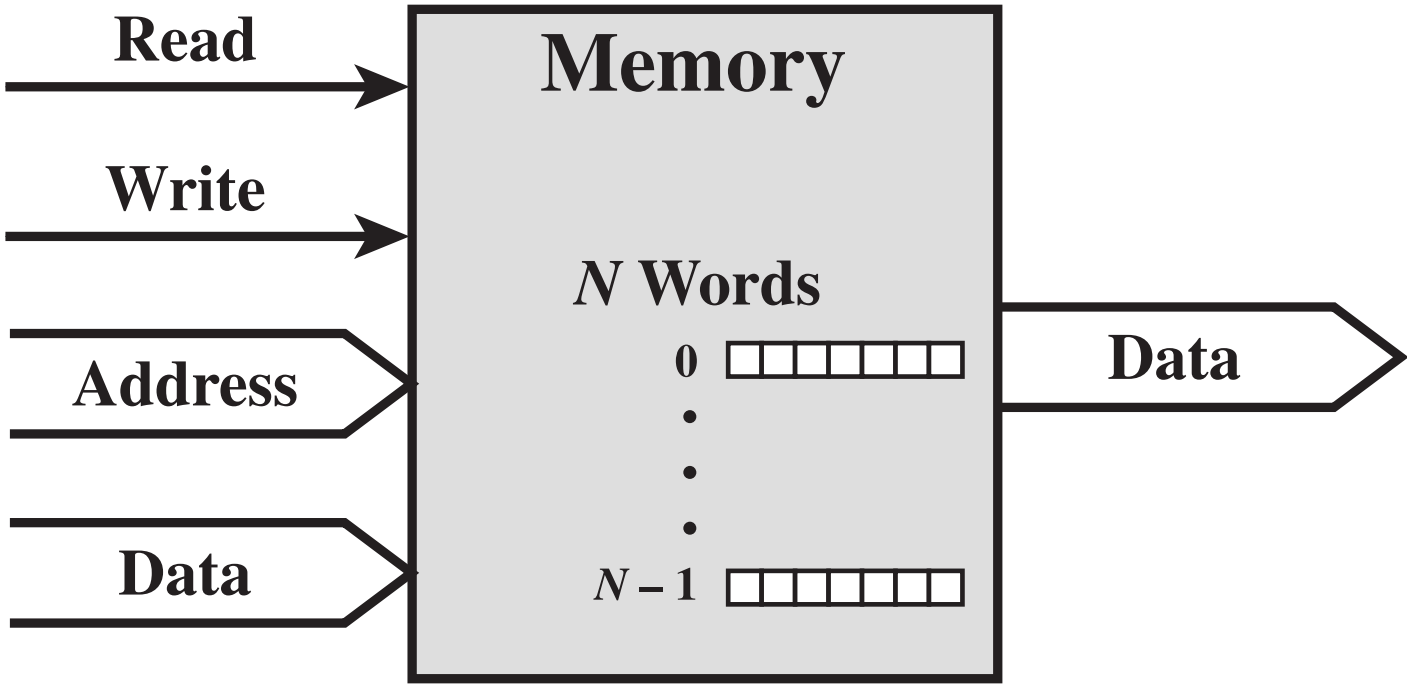
- *Organización de Computadores*. C. Hamacher, Z. Vranesic y S. Zaky. Mc Graw Hill. 2003
- *Problemas Resueltos de Estructura de Computadores*. F. García, J. Carretero, J.D. García y D. Expósito. Paraninfo, 2009.

Índice

Índice

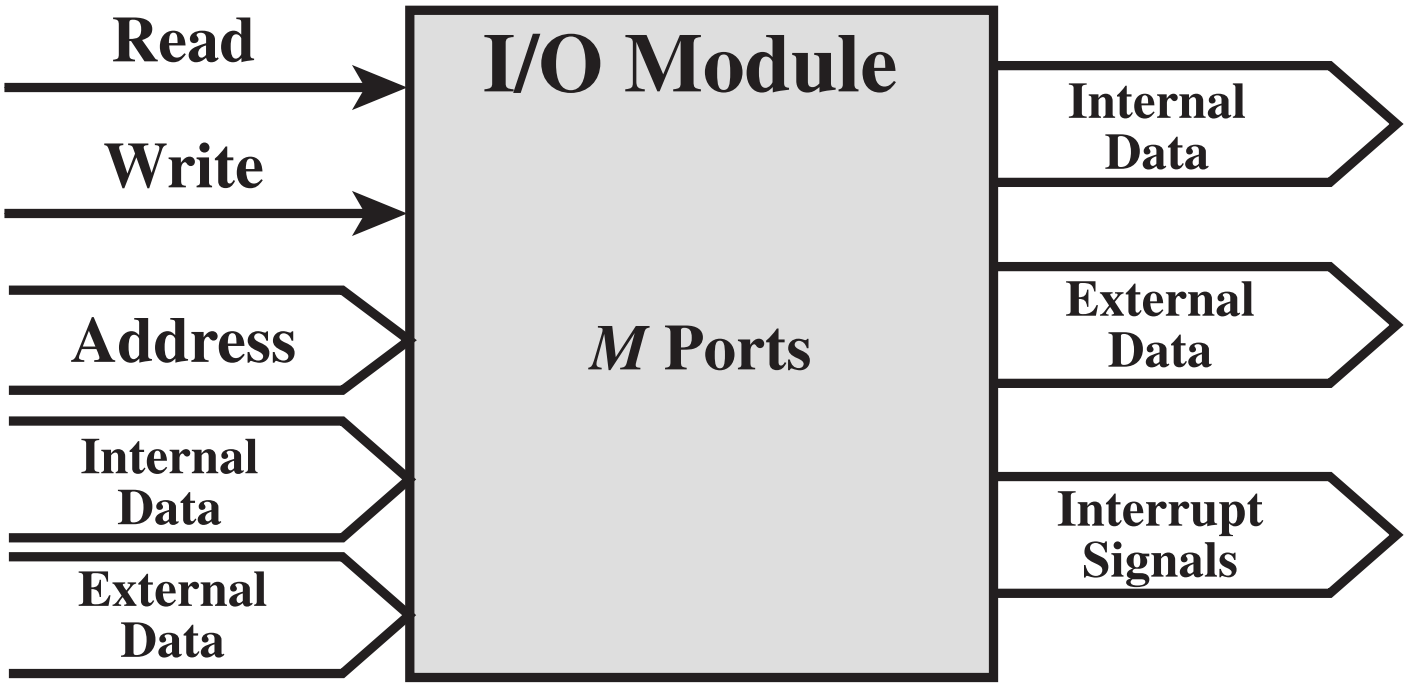
Introducción

- Comunicación entre unidades funcionales



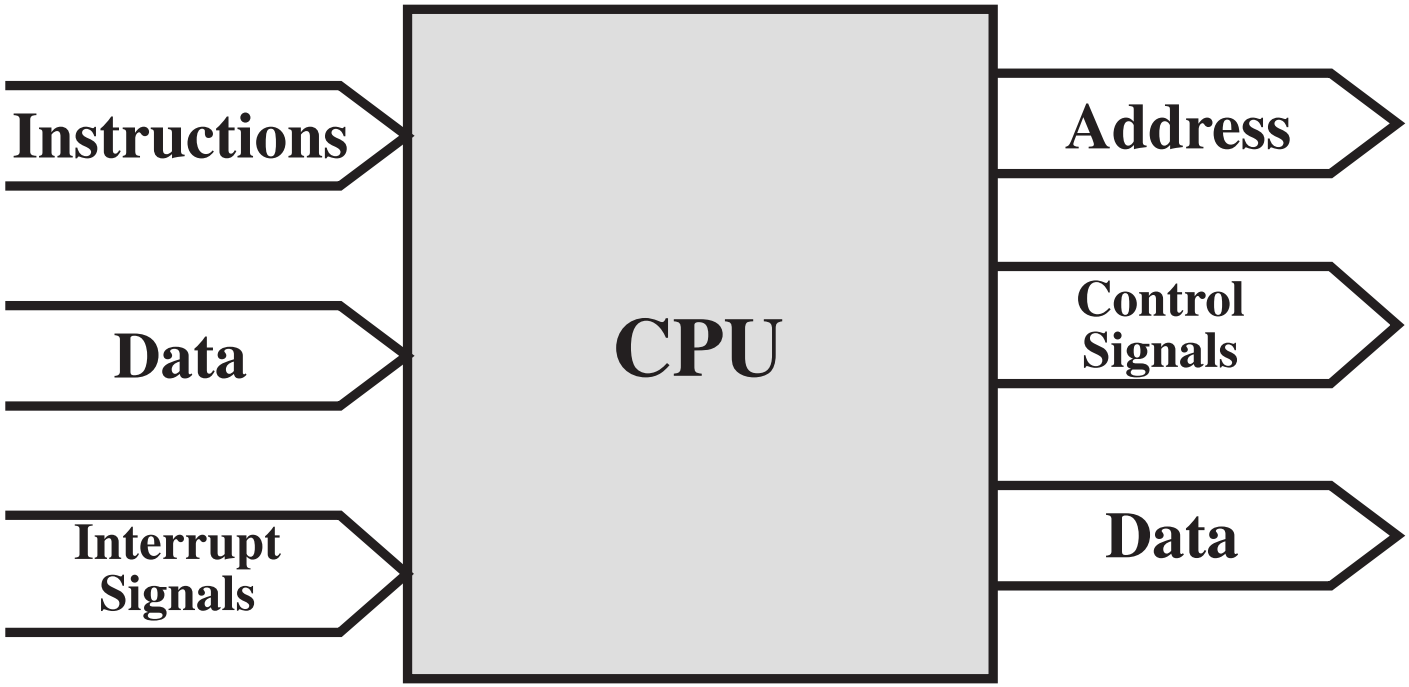
Introducción

- Comunicación entre unidades funcionales



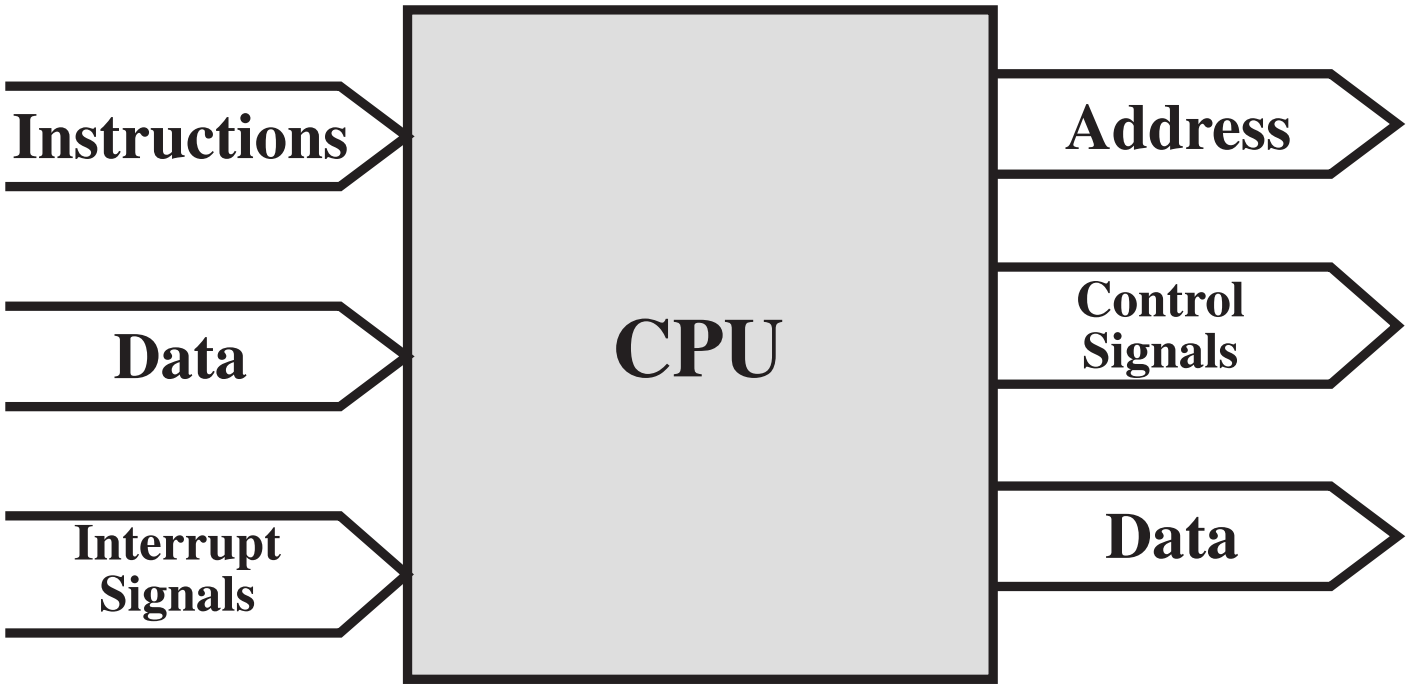
Introducción

- Comunicación entre unidades funcionales



Introducción

- Comunicación entre unidades funcionales



- Tipos de transferencias a interconectar:
  - Procesador $\Leftrightarrow$ Memoria
  - Procesador $\Leftrightarrow$ E/S
  - Memoria $\Leftrightarrow$ E/S (DMA)

Interconexión con **buses**

BUS

Canal de comunicación compartido que utiliza un conjunto de líneas para conectar varios subsistemas

- En cada instante cada línea transmite un único bit de información
- Ventajas:
  - versatilidad
    - facilidad para añadir nuevos dipositivos al sistema
    - posibilidad de estandarizar conexiones entre dispositivos
  - bajo coste
- Desventaja:
  - cuello de botella $\Rightarrow$ limita la productividad máxima del sistema
- Diseño de un sistema de buses
  - cubrir la demanda de comunicación del sistema y permitir la conexión de un gran número de disp. E/S

Interconexión con **buses**

BUS

Canal de comunicación compartido que utiliza un conjunto de líneas para conectar varios subsistemas

- En cada instante cada línea transmite un único bit de información
- Ventajas:
  - versatilidad
    - ★ facilidad para añadir nuevos dipositivos al sistema
    - ★ posibilidad de estandarizar conexiones entre dispositivos
  - bajo coste
- Desventaja:
  - cuello de botella $\Rightarrow$ limita la productividad máxima del sistema
- Diseño de un sistema de buses
  - cubrir la demanda de comunicación del sistema y permitir la conexión de un gran número de disp. E/S

Interconexión con **buses**

BUS

Canal de comunicación compartido que utiliza un conjunto de líneas para conectar varios subsistemas

- En cada instante cada línea transmite un único bit de información
- Ventajas:
  - versatilidad
    - ⋆ facilidad para añadir nuevos dipositivos al sistema
    - ⋆ posibilidad de estandarizar conexiones entre dispositivos
  - bajo coste
- Desventaja:
  - cuello de botella $\Rightarrow$ limita la productividad máxima del sistema
- Diseño de un sistema de buses
  - cubrir la demanda de comunicación del sistema y permitir la conexión de un gran número de disp. E/S

Interconexión con **buses** (y II)

- Características deseables en un BUS
  - heterogeneidad
  - escalabilidad
  - baja latencia
  - alto ancho de banda
- Principal problema en el diseño de un BUS
  - velocidad máxima (rendimiento) fuertemente limitada por cuestiones físicas:
    - longitud del bus
    - número de dispositivos conectados
  - y también limitada por la necesidad de dar soporte a una gran variedad de dispositivos con muy distintas latencias y anchos de banda
- Industria en transición:

buses paralelos compartidos (aproximación clásica, pe. PCI)

$\Downarrow$

interconexiones punto-a-punto de alta velocidad con switches (pe. PCI Express)

Interconexión con **buses** (y II)

- Características deseables en un BUS
  - heterogeneidad
  - escalabilidad
  - baja latencia
  - alto ancho de banda
- Principal problema en el diseño de un BUS
  - velocidad máxima (rendimiento) fuertemente limitada por cuestiones físicas:
    - ★ longitud del bus
    - ★ número de dispositivos conectados
  - y también limitada por la necesidad de dar soporte a una gran variedad de dispositivos con muy distintas latencias y anchos de banda
- Industria en transición:

  buses paralelos compartidos (aproximación clásica, pe. PCI)

  ⇓

  interconexiones punto-a-punto de alta velocidad con switches (pe. PCI Express)

Interconexión con **buses** (y II)

- Características deseables en un BUS
  - heterogeneidad
  - escalabilidad
  - baja latencia
  - alto ancho de banda
- Principal problema en el diseño de un BUS
  - velocidad máxima (rendimiento) fuertemente limitada por cuestiones físicas:
    - ★ longitud del bus
    - ★ número de dispositivos conectados
  - y también limitada por la necesidad de dar soporte a una gran variedad de dispositivos con muy distintas latencias y anchos de banda
- Industria en transición:

buses paralelos compartidos (aproximación clásica, pe. PCI)

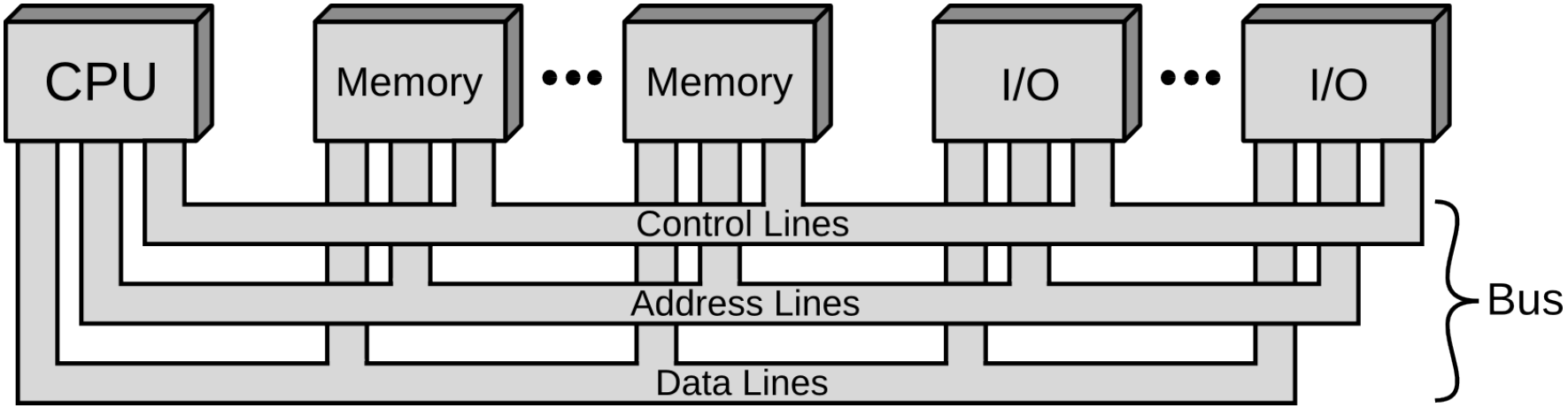
⇓

interconexiones punto-a-punto de alta velocidad con switches (pe. PCI Express)

Índice

Estructura de un BUS

- 3 grupos funcionales de líneas:



líneas de datos Camino para transmitir datos entre módulos.

- Nº líneas de datos determina máximo número de bits que es posible transmitir al mismo tiempo.

líneas de dirección Designan (direccionan) fuente o destino de datos.

- Anchura determina máxima cantidad de memoria (y de dispositivos de E/S) direccionable en sistema.

líneas de control Gestionan acceso y uso de líneas de datos y dirección.

- Señalizan peticiones y reconocimientos, indicando qué tipo de información pasa por las líneas de datos.

Funcionamiento de un BUS

- Transacción de bus: 2 pasos
  1. Petición (request): Se emite un comando, normalmente acompañado de una dirección
  2. Acción (action): Se transfieren datos
- Funcionamiento basado en paradigma Master/Slave:

  Maestro de bus (*BUS Master*) Inicia una transacción en el bus, emitiendo una *request*, y controla toda la operación
  - En un sistema puede haber múltiples dispositivos potenciales maestros de bus: CPUs, DMAs...

    $\Rightarrow$ Proceso de arbitraje previo para gestionar uso del bus

  Disp. esclavo El dispositivo que ha sido direccionado por el maestro. Responde a las peticiones del maestro:
  - Envía datos a través del bus si maestro pidió datos (lectura)
  - Recibe datos a través del bus si maestro quiere enviar datos (escritura)

Funcionamiento de un BUS

- Transacción de bus: 2 pasos
  1. Petición (request): Se emite un comando, normalmente acompañado de una dirección
  2. Acción (action): Se transfieren datos
- Funcionamiento basado en paradigma Master/Slave:

  Maestro de bus (*BUS Master*) Inicia una transacción en el bus, emitiendo una *request*, y controla toda la operación
  - En un sistema puede haber múltiples dispositivos potenciales maestros de bus: CPUs, DMAs...

    $\Rightarrow$ Proceso de arbitraje previo para gestionar uso del bus

  Disp. esclavo El dispositivo que ha sido direccionado por el maestro. Responde a las peticiones del maestro:
  - Envía datos a través del bus si maestro pidió datos (lectura)
  - Recibe datos a través del bus si maestro quiere enviar datos (escritura)

Arbitraje de BUS

- En un sistema con múltiples maestros de bus: esquema de arbitraje
  1. Cuando un maestro de bus quiere usar el bus, activa línea de petición de bus (bus request)
  2. Un maestro de bus no puede iniciar una transacción hasta que bus le es concedido (línea de control bus grant)
  3. Un maestro de bus avisa cuando ha terminado de utilizar bus
- Compromiso entre dos factores a considerar en el arbitraje:
  - Prioridad
  - Equidad (*fairness*)
- Clasificación genérica de esquemas de arbitraje
  - Arbitraje en serie (*daisy chain arbitration*)
  - Arbitraje paralelo centralizado
  - Arbitraje distribuido por autoselección
  - Arbitraje distribuido por detección de colisión

Arbitraje de BUS

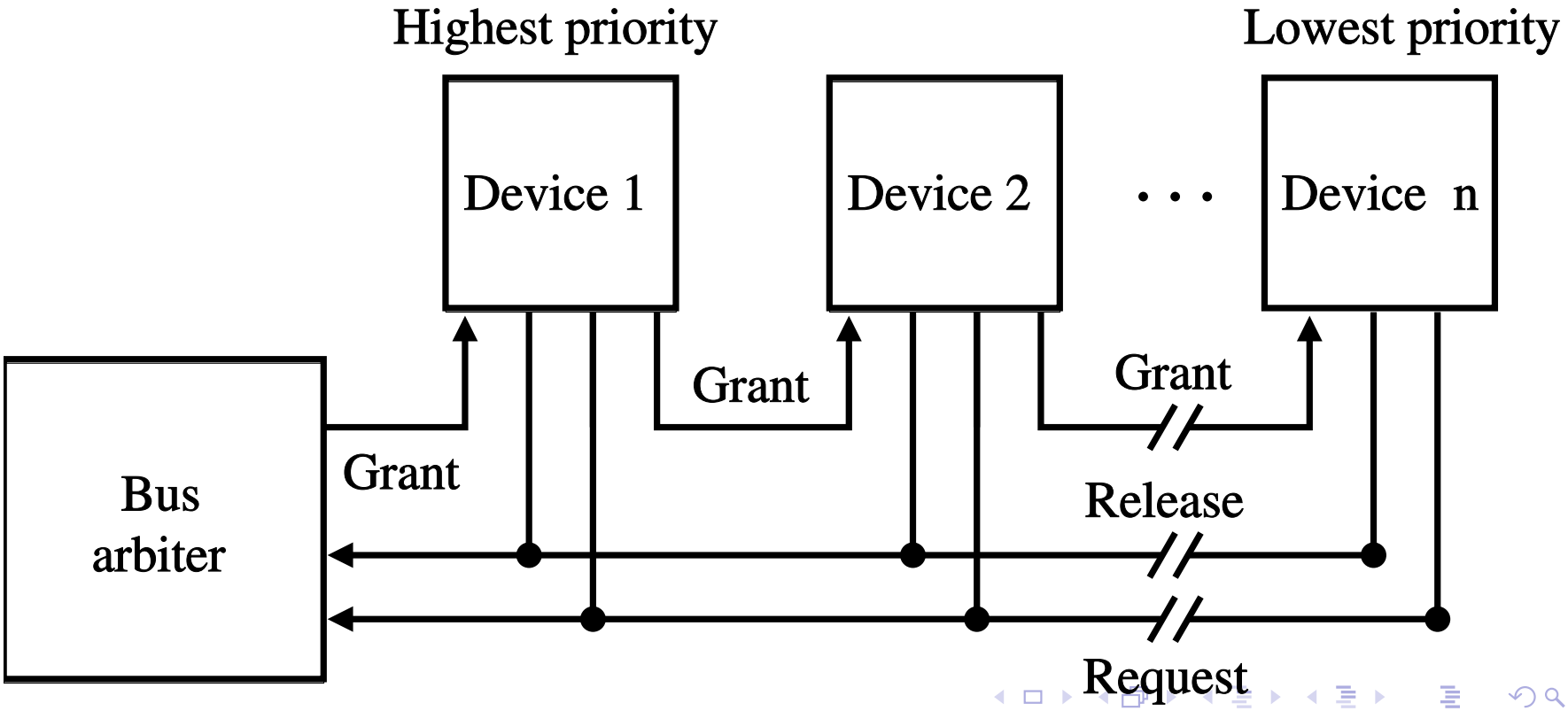
Clasificación genérica de esquemas de arbitraje

- Arbitraje en serie (*daisy chain arbitration*)
- Arbitraje paralelo centralizado
- Arbitraje distribuido por autoselección
- Arbitraje distribuido por detección de colisión

Arbitraje de BUS

Clasificación genérica de esquemas de arbitraje

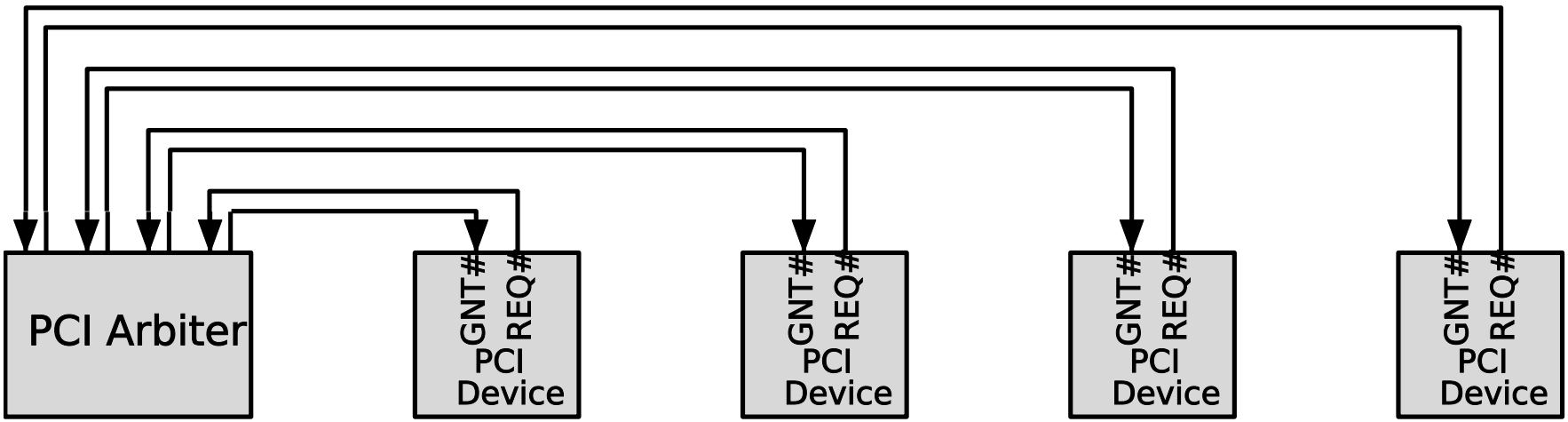
- **Arbitraje en serie** (*daisy chain arbitration*)
  - La línea de concesión de bus (*grant*) recorre los dispositivos de más a menos prioritario.
  - Prioridad determinada por posición de dispositivo en el bus.
  - ✗ Rendimiento limitado por propagación de señal de concesión
  - ✗ Difícil implementación de mecanismo que asegure equidad



Arbitraje de BUS

Clasificación genérica de esquemas de arbitraje

- Arbitraje en serie (*daisy chain arbitration*)
- **Arbitraje paralelo centralizado**
  - Múltiples líneas de petición, por las que los diferentes dispositivos piden acceso al bus de forma independiente.
  - Un árbitro centralizado selecciona uno de entre los dispositivos que han solicitado el bus y le notifica que ahora es el maestro del bus.



- Arbitraje distribuido por autoselección
- Arbitraje distribuido por detección de colisión

Arbitraje de BUS

Clasificación genérica de esquemas de arbitraje

- Arbitraje en serie (*daisy chain arbitration*)
- Arbitraje paralelo centralizado
- **Arbitraje distribuido por autoselección**
  - También múltiples líneas de petición de bus, por las que cada dispositivo solicita de forma independiente el bus.
  - Cada cada dispositivo determina de forma independiente si él es el solicitante de mayor prioridad sin necesidad de un árbitro.
- Arbitraje distribuido por detección de colisión

Arbitraje de BUS

Clasificación genérica de esquemas de arbitraje

- Arbitraje en serie (*daisy chain arbitration*)
- Arbitraje paralelo centralizado
- Arbitraje distribuido por autoselección
- **Arbitraje distribuido por detección de colisión**
  - Cada dispositivo solicita de forma independiente el bus
  - En caso de múltiples peticiones simultáneas de bus se produce una colisión.
  - Una vez detectada la colisión se aplica un esquema que determine el dispositivo que será maestro de bus entre las partes en colisión.

Características de un BUS: Cuestiones de diseño

Dedicación

Líneas dedicadas vs. multiplexadas

Temporización

Bus síncrono vs. asíncrono

Tipos de transacciones

- Operaciones básicas: 1 fase direc. + 1 fase datos
  - Lectura
  - Escritura
- Operaciones más sofisticadas: 1 fase direc. + múltiples fases datos
  - Lectura-modificación-escritura
  - Lectura-después de-escritura
  - Lectura o escritura de un bloque
- Posibilidad de transacción partida (*packet-switched bus*) cuando hay posibilidad de múltiples maestros de bus
  - Se dividen eventos del bus en *peticiones* y *respuestas*
  - Mejor aprovechamiento del bus: varias transacciones en curso, aprovechando esperas

Características de un BUS: Cuestiones de diseño

Dedicación

Líneas dedicadas vs. multiplexadas

Temporización

Bus síncrono vs. asíncrono

- Inclusión o no de señal de reloj entre líneas de control

Características de un BUS: Cuestiones de diseño

Dedicación

Líneas dedicadas vs. multiplexadas

Temporización

Bus síncrono vs. asíncrono

- Inclusión o no de señal de reloj entre líneas de control

Ventajas de tener reloj

- Protocolo para transacción fijo y gobernado por el reloj: poca lógica de control (máquina de estados sencilla)
- Alta velocidad de funcionamiento

Inconvenientes

- Imposibilidad para mezclar dispositivos con grandes diferencias de velocidad
- Limitados en longitud: problema del sesgo del reloj (*clock skew*)

Características de un BUS: Cuestiones de diseño

Dedicación

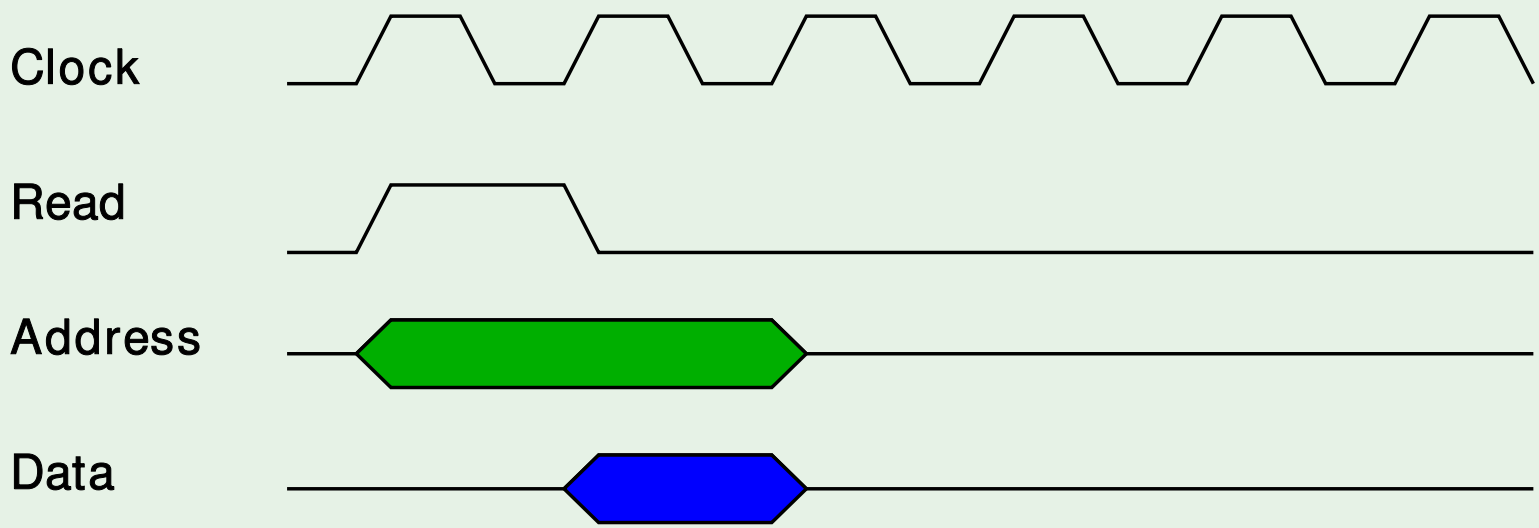
Líneas dedicadas vs. multiplexadas

Temporización

Bus síncrono vs. asíncrono

- Inclusión o no de señal de reloj entre líneas de control

Ejemplo de transacción: **lectura síncrona**



Características de un BUS: Cuestiones de diseño

Dedicación

Líneas dedicadas vs. multiplexadas

Temporización

Bus síncrono vs. asíncrono

- Inclusión o no de señal de reloj entre líneas de control

Ventajas de **no** tener reloj

- Conexión disp. amplio rango diferentes velocidades
- Escalan mejor en nº disp. y cambios tecnológicos en ellos
- Más largos (no problema sesgo del reloj)

Inconvenientes

- Transacción se coordina con protocolo más complejo (*handshaking*): más lentos (mayor sobrecarga sincronización)
- Líneas de control adicionales
- Más difícil acotar duración transacción

Características de un BUS: Cuestiones de diseño

Dedicación

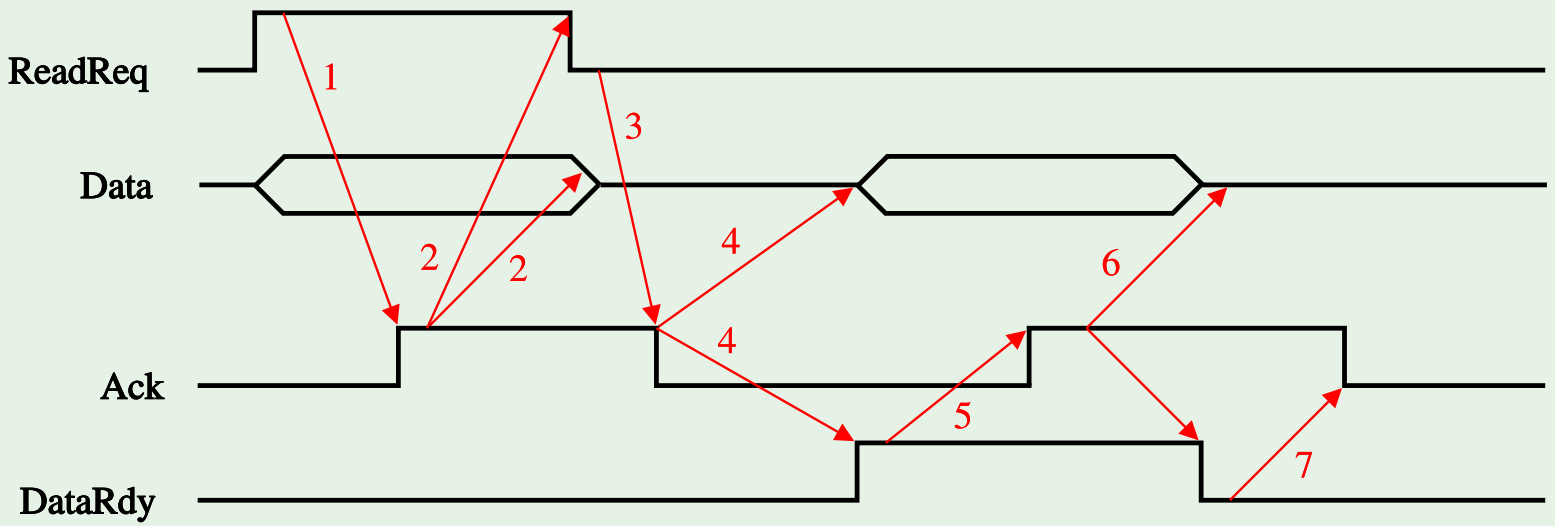
Líneas dedicadas vs. multiplexadas

Temporización

Bus síncrono vs. asíncrono

- Inclusión o no de señal de reloj entre líneas de control

Ejemplo de transacción: **lectura asíncrona**



Características de un BUS: Cuestiones de diseño

Dedicación

Líneas dedicadas vs. multiplexadas

Temporización

Bus síncrono vs. asíncrono

Tipos de transacciones

- Operaciones básicas: 1 fase direc. + 1 fase datos
  - Lectura
  - Escritura
- Operaciones más sofisticadas: 1 fase direc. + múltiples fases datos
  - Lectura-modificación-escritura
  - Lectura-después de-escritura
  - Lectura o escritura de un bloque
- Posibilidad de transacción partida (*packet-switched bus*) cuando hay posibilidad de múltiples maestros de bus
  - Se dividen eventos del bus en *peticiones* y *respuestas*
  - Mejor aprovechamiento del bus: varias transacciones en curso, aprovechando esperas

Índice

Tipos de BUSES

- Bus **Procesador-memoria**
  - Bus síncrono de alta velocidad
  - Poco versátil, diseño no estandarizado
- Bus **E/S**
  - Asíncrono o con frecuencia de reloj baja
  - Estándar y versátil: gran variedad de dispositivos con distintas velocidades

✗ Prestaciones en bus disminuyen con conexión de muchos dispositivos (bus = cuello de botella)

✗ Problemas ante distintas velocidades y anchos de banda de diferentes dispositivos

Solución

Configuraciones con múltiples buses: jerárquía de buses

- Bus ***Backplane*** o ***Mezzanine*** (Literalmente: *«entresuelo»*. Ejemplo, Bus PCI)

Tipos de BUSES

- Bus **Procesador-memoria**
  - Bus síncrono de alta velocidad
  - Poco versátil, diseño no estandarizado
- Bus **E/S**
  - Asíncrono o con frecuencia de reloj baja
  - Estándar y versátil: gran variedad de dispositivos con distintas velocidades

✗ Prestaciones en bus disminuyen con conexión de muchos dispositivos (bus = cuello de botella)

✗ Problemas ante distintas velocidades y anchos de banda de diferentes dispositivos

Solución

Configuraciones con múltiples buses: jerárquía de buses

- Bus ***Backplane*** o ***Mezzanine*** (Literalmente: *«entresuelo»*. Ejemplo, Bus PCI)

Tipos de BUSES

- Bus **Procesador-memoria**
  - Bus síncrono de alta velocidad
  - Poco versátil, diseño no estandarizado
- Bus **E/S**
  - Asíncrono o con frecuencia de reloj baja
  - Estándar y versátil: gran variedad de dispositivos con distintas velocidades

✗ Prestaciones en bus disminuyen con conexión de muchos dispositivos (bus = cuello de botella)

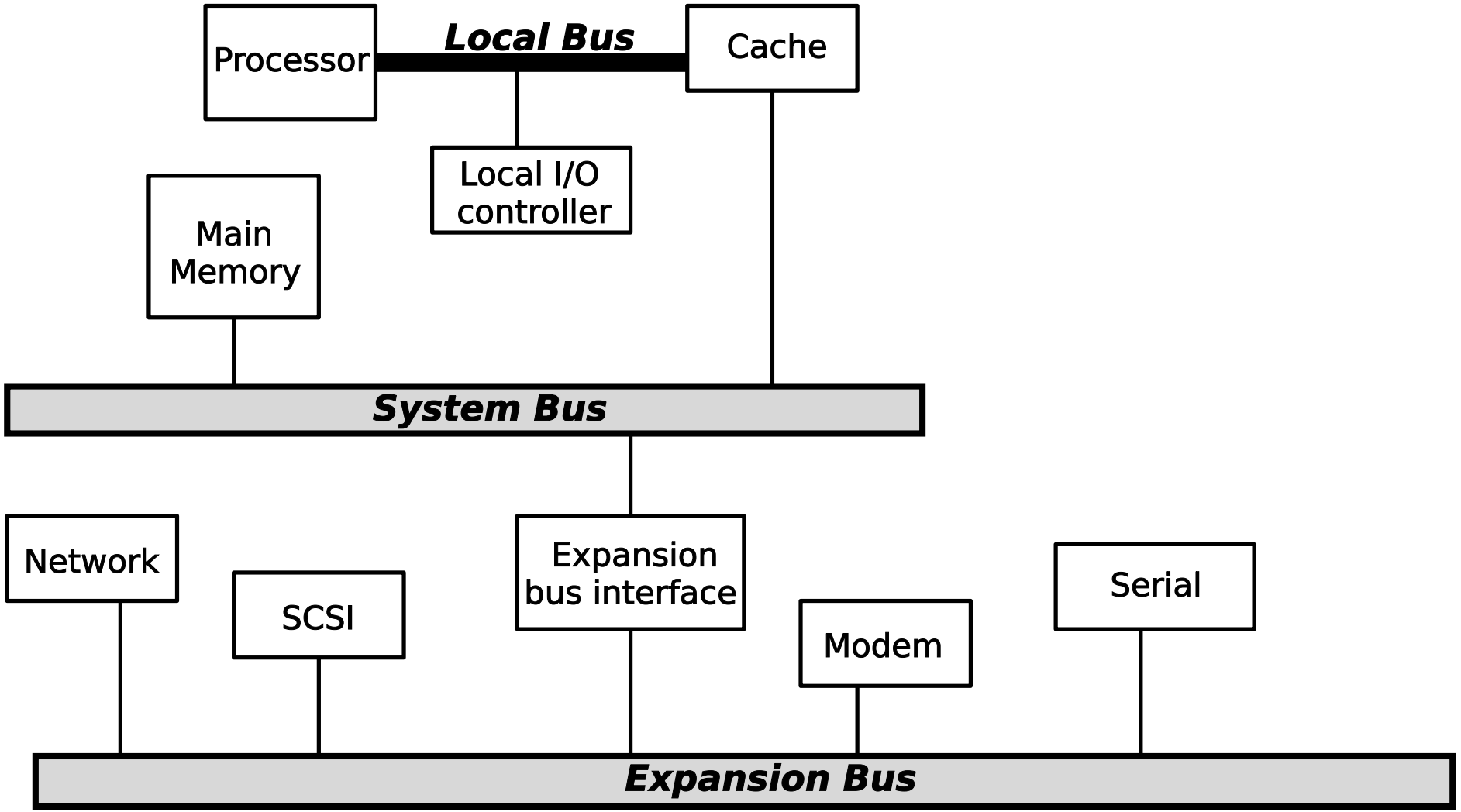
✗ Problemas ante distintas velocidades y anchos de banda de diferentes dispositivos

Solución

Configuraciones con múltiples buses: jerárquía de buses

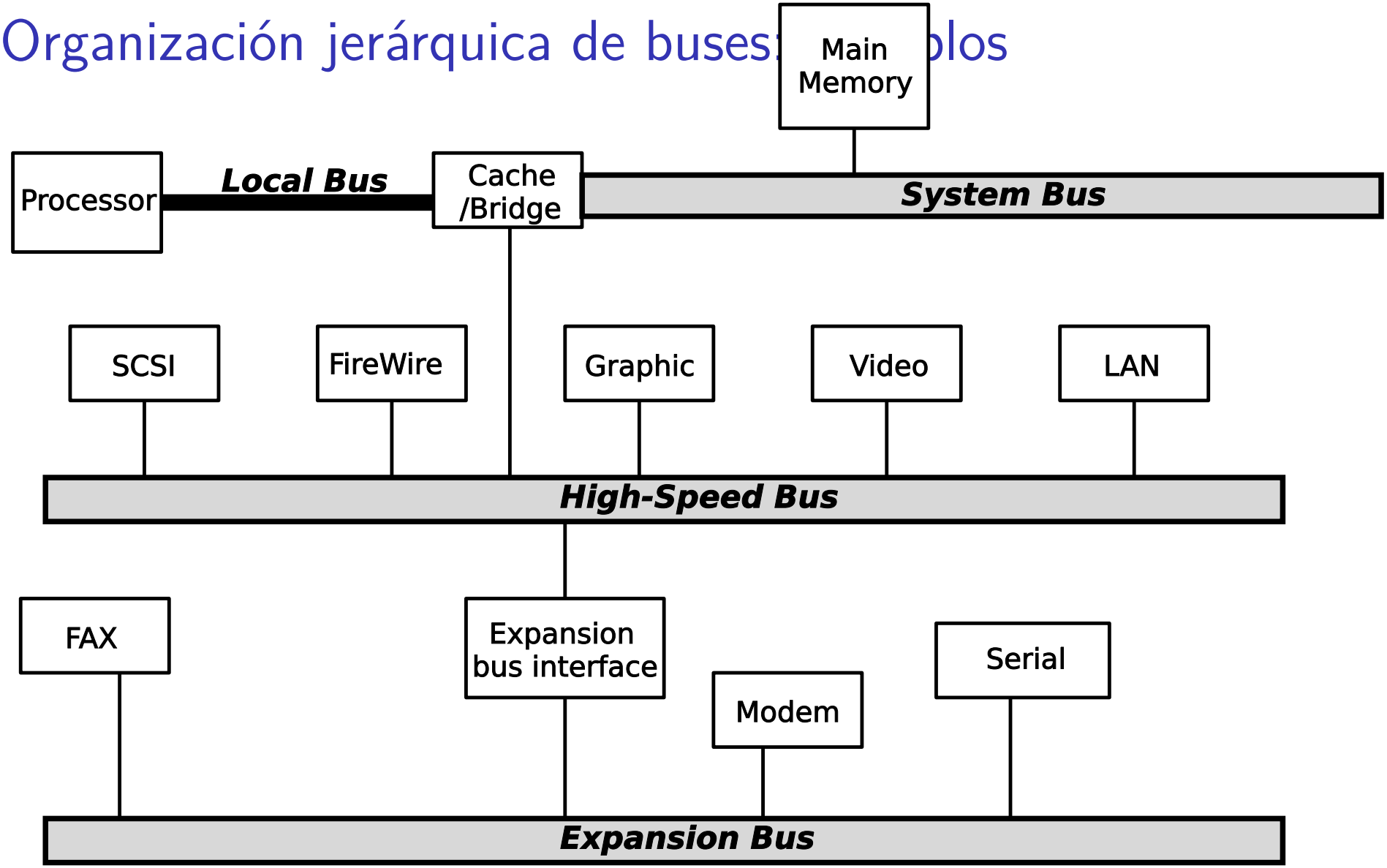
- Bus ***Backplane*** o ***Mezzanine*** (Literalmente: *«entresuelo»*. Ejemplo, Bus PCI)

Organización jerárquica de buses: ejemplos



(a) Traditional Bus Architecture

Organización jerárquica de buses: ejemplos



(b) High-Performance Architecture