4. ORGANIZACIÓN DE LA ENTRADA/SALIDA

- Importancia de la E/S en el rendimiento del sistema
- Medidas de rendimiento de la E/S
- Buses: interfaz periféricos-procesador-memoria

4.1. Importancia de la E/S en el rendimiento del sistema

- Error: medir el rendimiento de un sistema computador por el TUCP
- El rendimiento de un sistema depende de las diferentes partes que actúan en el camino entre la UCP y los dispositivos de entrada/salida; a saber:
  - UCP
  - Memorias caché y principal
  - Bus UCP-memoria y bus de E/S
  - Dispositivo de e/s y controlador, o canal, de E/S
  - Eficiencia del software de E/S
- La parte más lenta limitará el rendimiento del sistema => hay que equilibrarlo

Tiempo de ejecución de una carga de trabajo

- La actividad de E/S se solapa con la de la UCP =>

$$T_{CT} = T_{UCP} + T_{E/S} - T_{SOLAPAMIENTO}$$

  - Mejora de T_{CT} al mejorar, sólo T_{UCP}, G_{UCP} veces
    - ✓Caso *mejor*. Máximo tiempo de solapamiento:

$$T_{CT(MEJOR)} = T_{UCP}/G_{UCP} + T_{E/S} - Mín\ (T_{SOLAPAMIENTO}, T_{UCP}/G_{UCP})$$

    - ✓Caso *peor*. Mínimo t. de solapamiento: $T_{CT(PEOR)} = T_{UCP}/G_{UCP} + T_{E/S} - Máx\ (0, T_{SOLAPAMIENTO} - (T_{UCP} - T_{UCP}/G_{UCP}))$
    - ✓Caso *escalado*. El t. de solapamiento se reduce en la misma medida que el tiempo de UCP

$$T_{CT(ESCALADO)} = T_{UCP}/G_{UCP} + T_{E/S} - T_{SOLAPAMIENTO}/G_{UCP}$$

Tiempo de ejecución de una carga de trabajo (cont.)

- Mejora de T_{CT} al mejorar, T_{UCP}, G_{UCP} veces y $T_{E/S}$, $G_{E/S}$ veces
  - ✓Sean $\mathbf{NUEVOT_{UCP} \equiv NT_{UCP} \equiv T_{UCP}/G_{UCP}}$ y $\mathbf{NT_{E/S} \equiv T_{E/S}/G_{E/S}}$
  - ✓Caso *mejor*

$$\mathbf{T_{CT(MEJOR)} = NT_{UCP} + NT_{E/S} - Mín\ (T_{SOLAPAMIENTO}, NT_{UCP}, NT_{E/S})}$$

  - ✓Caso *peor*. $\mathbf{T_{CT(PEOR)} = NT_{UCP} + NT_{E/S} -}$

$$\mathbf{- Máx\ (0, T_{SOLAPAMIENTO} - Máx(T_{UCP} - NT_{UCP}, T_{E/S} - NT_{E/S}))}$$

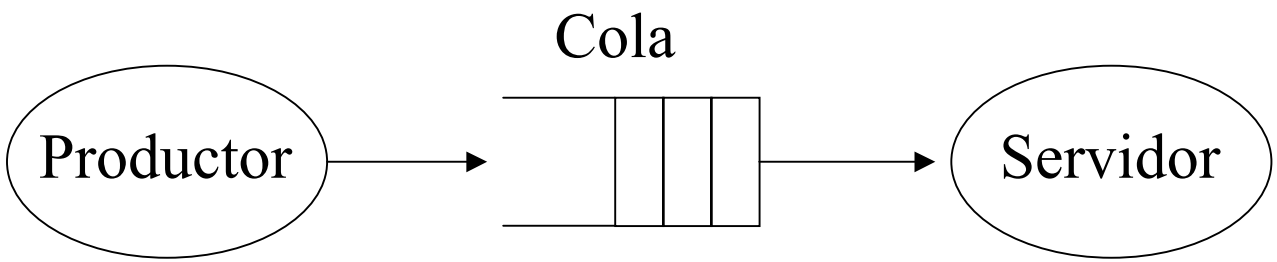
  - ✓Caso *escalado*. $\mathbf{T_{CT(ESCALADO)} = NT_{UCP} + NT_{E/S} -}$

$$\mathbf{- T_{SOLAPAMIENTO}/(Máx(G_{UCP}, G_{E/S}))}$$

Tiempo de ejecución de una carga de trabajo (cont.)

- No olvidar que la entrada/salida también repercute en TUCP
  - Si es por sondeo, la UCP ejecutará un bucle de espera durante un tiempo que dependerá del rendimiento del dispositivo
  - Si es por interrupciones, el manejo de las mismas también supone un tiempo de UCP

4.2. Medidas de rendimiento de la E/S

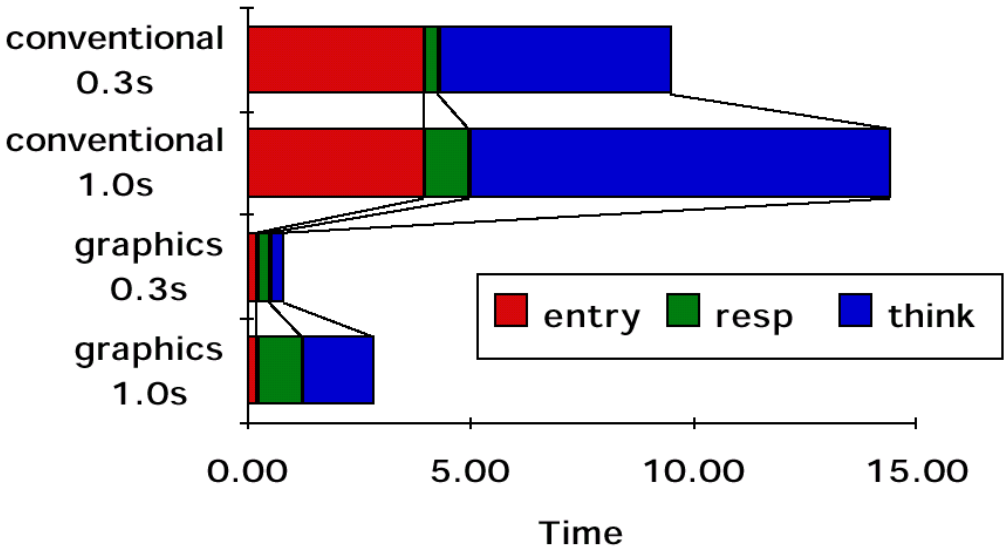


- Tiempo de respuesta: t. desde que el productor coloca una tarea en la cola hasta que el servidor finaliza esa tarea
  - Menor cuanto más vacía esté la cola
- Productividad: número de tareas completadas por el servidor en un período de tiempo
  - Menor si la cola queda vacía =>
  - Prácticamente imposible optimizar ambos objetivos a la vez

Tiempo de transacción persona-computador

- Tres componentes:
  - Tiempo de *entrada*. Tiempo que tarda el usuario en introducir una orden.
  - Tiempo de *respuesta del sistema*. Tiempo transcurrido desde que el usuario introduce la orden hasta que se visualiza la respuesta completa.
  - Tiempo de *asimilación* (*think time*). Tiempo que transcurre desde la recepción de la respuesta hasta que el usuario comienza a introducir la siguiente orden.

Tiempo de transacción persona-computador: algunos resultados



- Reducir t. de respuesta => reducir t. de asimilación
  - ✓ Un ahorro de 0,7s en el t. de respuesta conlleva un ahorro, en el t. de asimilación, de 4,9s en el sistema convencional y de 2s en el entorno gráfico
  - ✓ El tiempo de entrada no cambia

Tiempo de transacción persona-computador: el tiempo de respuesta y el factor humano

- Las personas necesitamos menos tiempo de asimilación cuando se nos da una respuesta más rápida
- La productividad humana aumenta mucho con tiempos de respuesta inferiores al segundo
  - La productividad de un principiante en un sistema con un tiempo de respuesta pequeño es similar a la de un experto en un sistema con un tiempo de respuesta grande

Benchmarks de rendimiento de E/S en disco

- Benchmarks TP (procesamiento de transacciones)
  - Miden el Nº. de transacciones por segundo (TPS)
  - Generalmente se presupone un sistema formado por un gran cuerpo de información compartido por muchos terminales. Ejemplos típicos:
    - ✓Sistema de reservas de billetes de una compañía aérea
    - ✓Red de cajeros automáticos de un banco

Benchmarks de rendimiento de E/S en disco: *Debit Credit*

- El benchmark ***DebitCredit*** ejecuta las operaciones de un cliente depositando o retirando dinero de un banco (***TPC-A, TPC-B***, etc. son versiones del *DebitCredit* con especificaciones más estrictas)
  - La E/S de disco para el *DebitCredit* son lecturas y escrituras aleatorias de registros de 100 bytes junto con escrituras secuenciales ocasionales

Benchmarks de rendimiento de E/S en disco: *Debit Credit* (cont.)

- A cada transacción corresponden
  - ✓Entre 2 y 10 entradas/salidas de disco
  - ✓Entre 5000 y 20000 instrucciones de UCP por cada e/s de disco
- Estos valores dependen de:
  - ✓La eficiencia del **software** de procesamiento de transacciones
  - ✓El ahorro de accesos a disco conseguido por mantener información en memoria principal

Benchmarks de rendimiento de E/S en disco: *Debit Credit* (cont.)

- Principal medida de rendimiento: TPS máximo bajo la restricción de que el 95% de las transacciones tienen un tiempo de respuesta menor que 1s
- Escalabilidad: Para que un sistema pueda tener un mayor TPS se le exige un mayor número de cajeros y un mayor tamaño de los archivos de las cuentas:

  Tamaño del archivo de cuentas = TPS x 0,01 GB;
  Nº cajeros = 100 x TPS

  ✓ La escalabilidad es necesaria para evitar falsear resultados usando una memoria principal grande y un número de cuentas pequeño

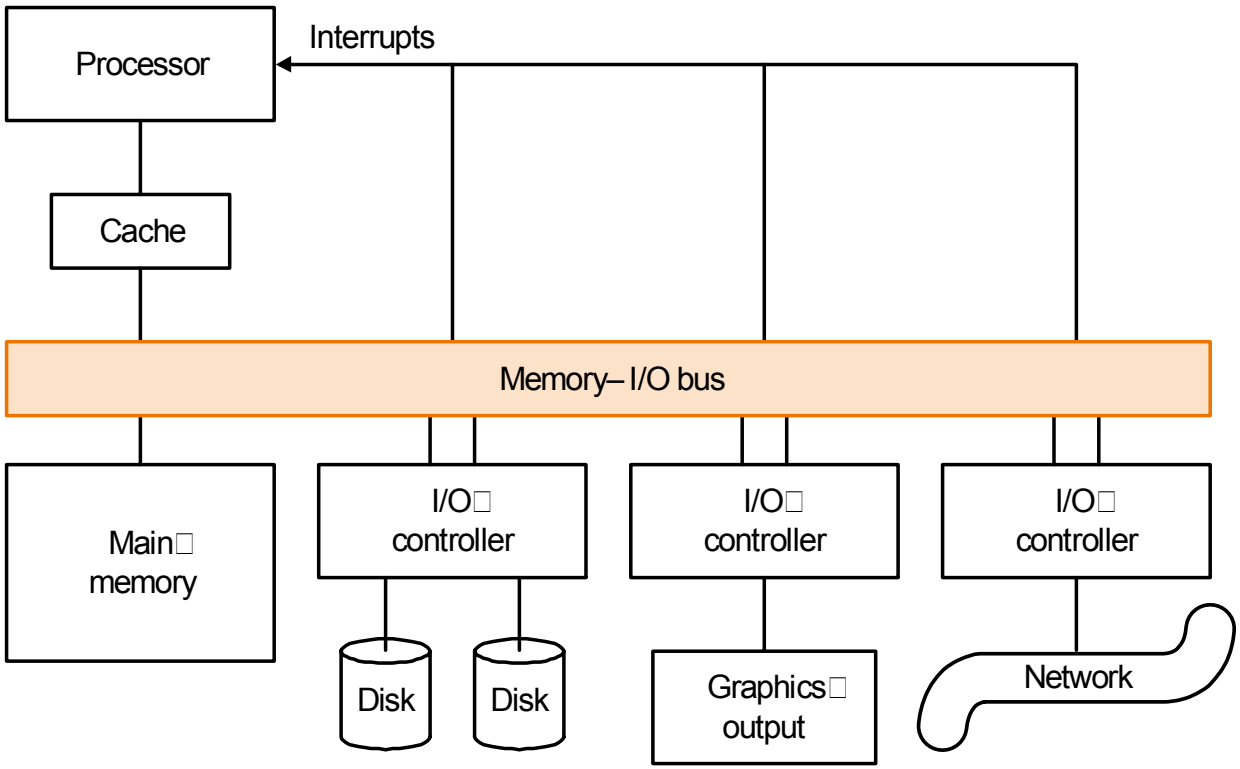
Benchmarks de rendimiento de E/S en disco: (cont.)

- Benchmark **SPECsfs** (*System-Level File Server*)
  - Programa sintético pensado para evaluar sistemas que ejecutan el servicio de archivos de red (*network file service* o NFS) de Sun Microsystems
  - Contiene una mezcla de lecturas, escrituras y otras operaciones en archivos, en un entorno de red
  - También se le exige escalabilidad y un tiempo de respuesta medio menor que 50 ms
  - Resultados: número de operaciones NFS por segundo frente a tiempo medio de respuesta

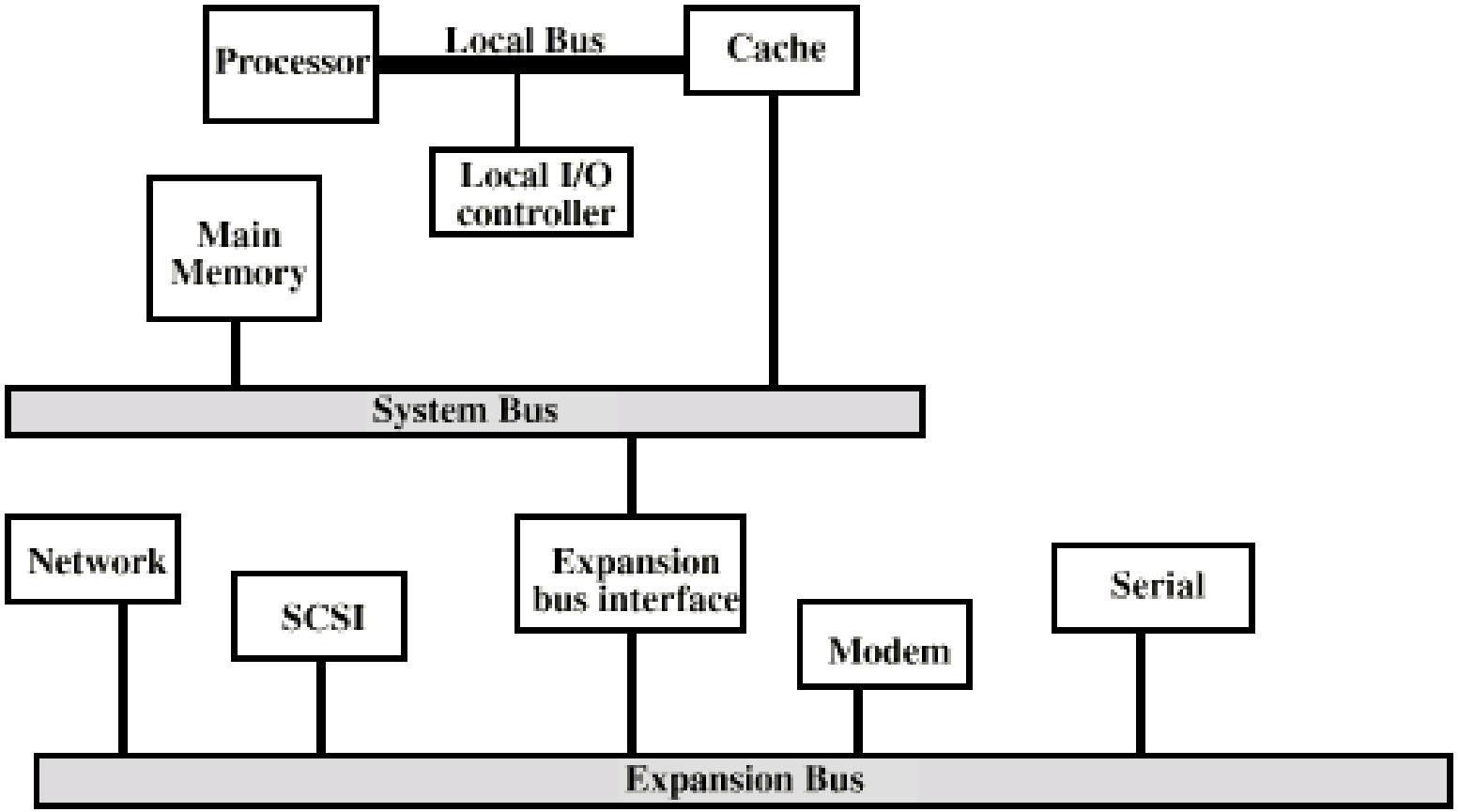
4.3. Buses: interfaz periféricos-procesador-memoria

- Clases básicas de buses en un sistema computador
  - Bus procesador-memoria o bus del sistema
    - ✓Corto, alta velocidad y adaptado al sistema de memoria para maximizar la anchura de banda memoria-procesador
  - Bus de E/S
    - ✓Puede ser largo y aceptar un amplio rango en el ancho de banda de los dispositivos conectados a él
    - ✓Normalmente siguen un estándar de bus

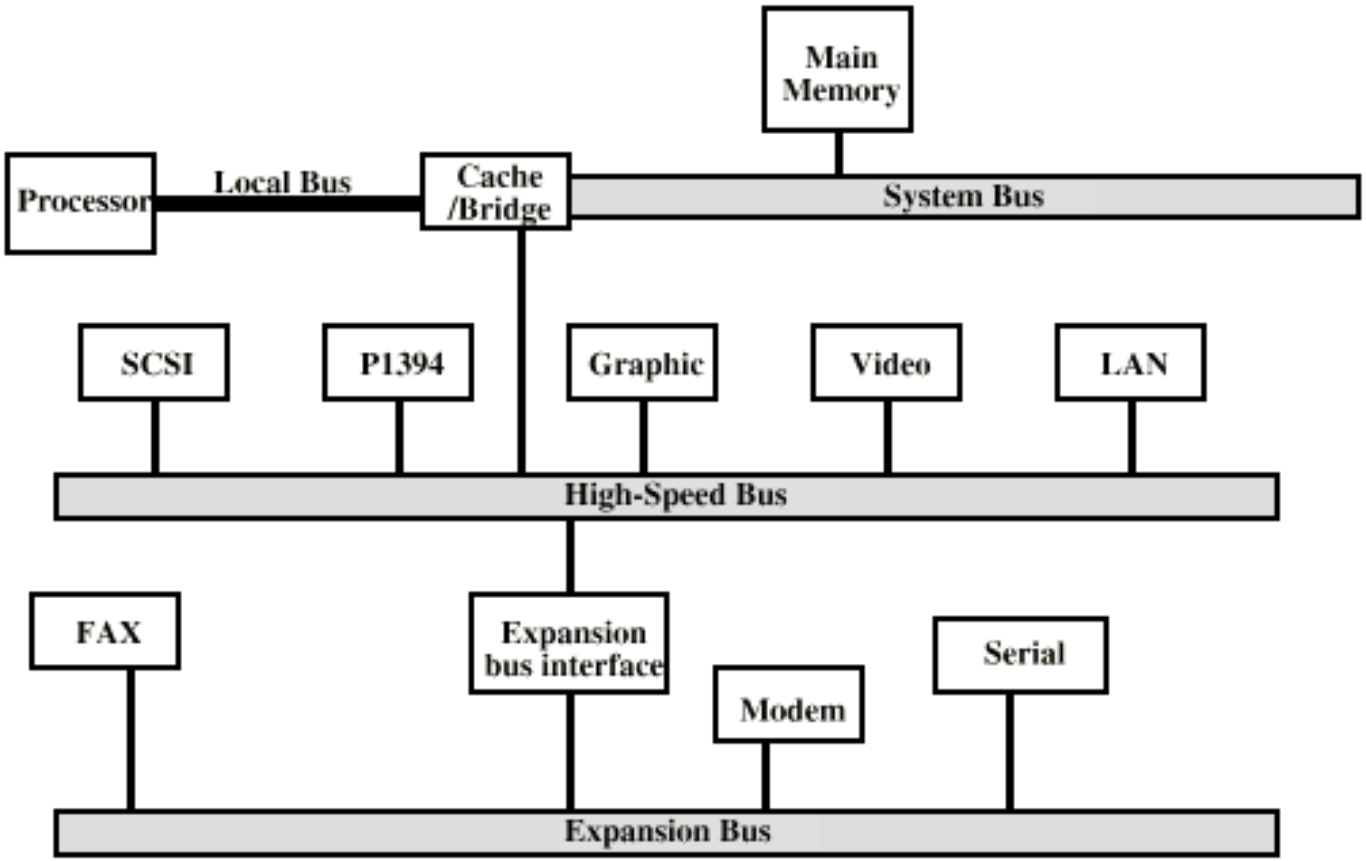
Buses en un sistema computador: bus único para memoria y E/S (sistemas de bajo coste)



Buses en un sistema computador: caso típico



Buses en un sistema computador: arquitectura de altas prestaciones



Tipos de líneas de un bus

- De datos
- De direcciones
  - La dirección determina la fuente o el destino del dato (también los puertos de E/S se pueden direccionar)
- De control
  - De temporización: determinan cuando empieza y termina cada transmisión
  - Comandos de tipo de transacción: lectura/escritura, datos/estado, etc.
  - Arbitraje: dan prioridad a unas transacciones sobre otras y deciden qué módulo tiene acceso al bus en cada momento
  - Petición/reconocimiento de interrupción y otras

Protocolos de bus: conceptos básicos

- Transacción típica a través de un bus
  - Enviar una dirección y recibir o enviar un dato
- Protocolo de comunicación
  - Especificación de la secuencia de eventos y temporización en la transferencia de información
- Controlador (*master*) del bus: tiene capacidad para iniciar una transacción
- Esclavo (*slave*) del bus: módulo activado por la transacción

Principales decisiones en el diseño de buses

- Anchura del bus de datos
- Diferentes/las mismas líneas (multiplexadas en el tiempo) para datos y direcciones (afecta a la anchura de bus)
- Tamaño del bloque transferido en una transacción
  - Transferir bloques de una palabra es más sencillo
  - Transferir bloques de varias palabras mejora el rendimiento (el tiempo gastado por palabra en transferencias de direcciones y señales de control, es menor)

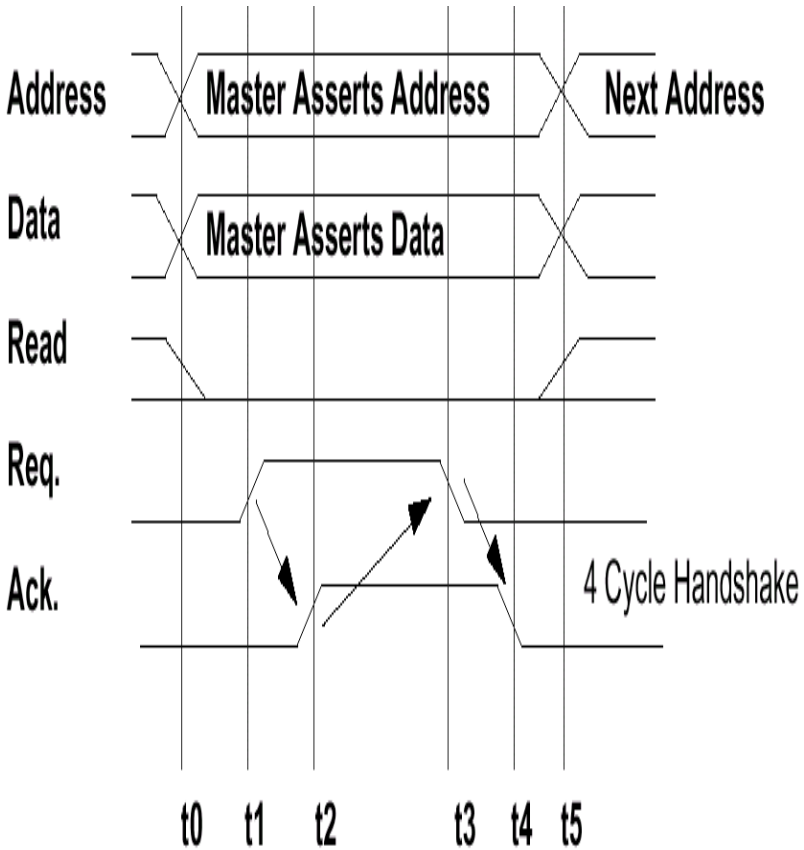
Principales decisiones en el diseño de buses: bus síncrono

- Protocolo sencillo basado en una señal de reloj
- Ventajas: rapidez y bajo coste
- Inconvenientes
  - Todos los elementos conectados al bus deben utilizar la misma señal de reloj
  - No pueden ser largos si son rápidos debido al sesgo (*skew*) del reloj
- Típicamente, los buses procesador-memoria son síncronos

Principales decisiones en el diseño de buses: bus asíncrono

- Carece de reloj => ventajas:
  - Admite dispositivos de velocidades diversas
  - Puede alargarse sin que por ello aparezcan problemas de sincronización ni de sesgo de reloj
- Protocolo entrelazado (*fully interlocked* o *handsaking*)
  - Requiere señales de control específicas (por ejemplo *Reqest* y *Acknowledge* [petición y reconocimiento])
    - ✓ La fiabilidad del protocolo se basa en que entre dos cambios de estado consecutivos de la señal *Req* tiene que haber un cambio de estado de *Ack* y viceversa
- Típicamente, los buses de E/S son asíncronos

Ejemplo de transacción asíncrona



- t0: El controlador (*Master*) coloca en el bus la dirección y los datos y especifica la operación escribir (baja *Read*)
- t1: Tras una espera, *Master* activa *Req.*
- t2: El *esclavo* activa *Ack.* (“he recibido la petición y los datos”)
- t3: *Master* desactiva *Req.* (“sé que has recibido mi petición y mis datos”)
- t4: El *esclavo* desactiva *Ack.* (“sé que has desactivado *Req.*”)

Principales decisiones en el diseño de buses: bus de ciclo partido (*split-cycle*)

- En una lectura el controlador transmite una dirección al esclavo y se desconecta del bus
  - Además transmite un identificador de sí mismo
- Otros controladores utilizan entonces el bus
- Cuando el esclavo en tiene listos los datos inicia la segunda parte del ciclo:
  - Accede al bus como controlador y transmite el dato al peticionario (al que tiene identificado y que ahora se comporta como esclavo)

Principales decisiones en el diseño de buses: bus de ciclo partido (cont.)

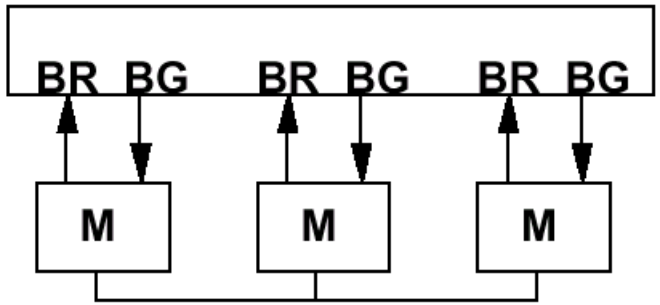
- Ventaja: mayor ancho de banda
  - El tiempo muerto de la transacción iniciada por un controlador es aprovechado por algún otro
- Inconvenientes:
  - El esclavo necesita una lógica para poder actuar como controlador
  - Debido a que hay varios controladores, se necesita un mecanismo de arbitraje del bus
  - Aumenta la latencia de las transacciones
- Es propio de sistemas con varios procesadores o dispositivos DMA conectados al bus

Tipos de arbitraje del bus

- Paralelo centralizado

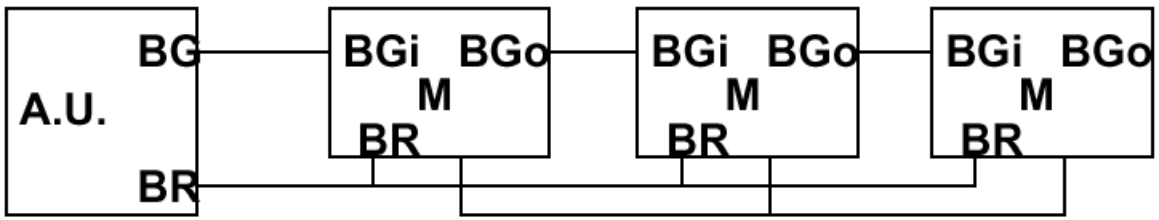
  BG: *bus grant*

  BR: *bus request*



- Serie (*daisy chain*)

A.U.:
arbitration unit



- Distribuido por autoselección
- Distribuido por detección de colisiones

Arbitraje paralelo centralizado

- Cada dispositivo tiene una línea de petición del bus (BR)
- Árbitro centralizado
  - Selecciona uno de los dispositivos que piden acceso al bus
  - Le concede el acceso activando su línea BG
- Inconveniente: el árbitro central puede ser un cuello de botella en la utilización del bus

Arbitraje serie

- Protocolo
  - Mientras un controlador controla el bus, activa la línea *bus busy* (sin nombre en el gráfico)
  - Si un controlador quiere acceder al bus activa BR
  - Si hay una petición pendiente y el bus está desocupado, el árbitro activa BG
  - Si un controlador no tiene petición pendiente deja pasar la señal BG al siguiente
  - Si el controlador tiene una petición pendiente y detecta un flanco de subida en BG, toma el control del bus (puede iniciar una transacción)

Arbitraje serie (cont.)

- ✓Tener en cuenta el flanco en lugar del nivel de BG evita que un controlador que haya dejado pasar dicha señal, acceda al bus al mismo tiempo que otro u otros de menor prioridad => fallo de arbitraje
- Ventaja: simplicidad
- Inconvenientes:
  - No garantiza la *imparcialidad:* los dispositivos de baja prioridad podrían no acceder nunca al bus
  - El que la señal BG tenga que atravesar varios controladores reduce la velocidad del bus

Arbitraje distribuido por autoselección

- Cada controlador que quiere acceso al bus coloca un código indicando su identidad
- Examinando el bus cada controlador puede determinar si es él el de mayor prioridad de los solicitantes de acceso
- Ventaja: no se necesita un árbitro central
- Inconveniente: se requieren más líneas para las señales de petición

Arbitraje distribuido por detección de colisiones

- Cada dispositivo pide el bus independientemente
- Pueden producirse colisiones como consecuencia de varias peticiones simultáneas
  - Los dispositivos leen el bus para detectar posibles colisiones (hay colisión si lo leído difiere de lo escrito en el bus)
- Los nodos que colisionan detienen la transmisión
  - Un esquema de selección determina quien debe tomar el control del bus
  - Por ejemplo, en Ethernet los dispositivos que colisionan retrasan la transmisión durante un intervalo aleatorio de tiempo

Ejemplos de buses

	Bus VME	FutureBus	IPI	SCSI
Tipo bus	CPU-mem.	CPU-mem.	E/S	E/S
Líneas de bus	128	96	16	8
Ancho dato	16-32 bits	32 bits	16 bits	8 bits
Nº controladores	Múltiple	Múltiple	Único	Múltiple
Arbitraje	Serie	Autosel.	-	Autosel.
Temporización	Asíncr.	Asíncr.	Asíncr.	Ambas
Ancho de banda (bloques 1 palabra)	12,9 MB/s	15,5 MB/s	25,0 MB/s	1,5 ó 5 MB/s
Ancho de banda (bloques ilimitados)	13,6 MB/s	20,8 MB/s	25,0 MB/s	1,5 ó 5 MB/s
Long. máx. bus	0,5 m	0,5 m	50 m	25 m