TEMA 6: Buses comunes y Dispositivos de E/S

Contenidos

Objetivo: conocer las características de los buses más extendidos así como la diversidad de dispositivos de E/S que pueden utilizar estos buses.

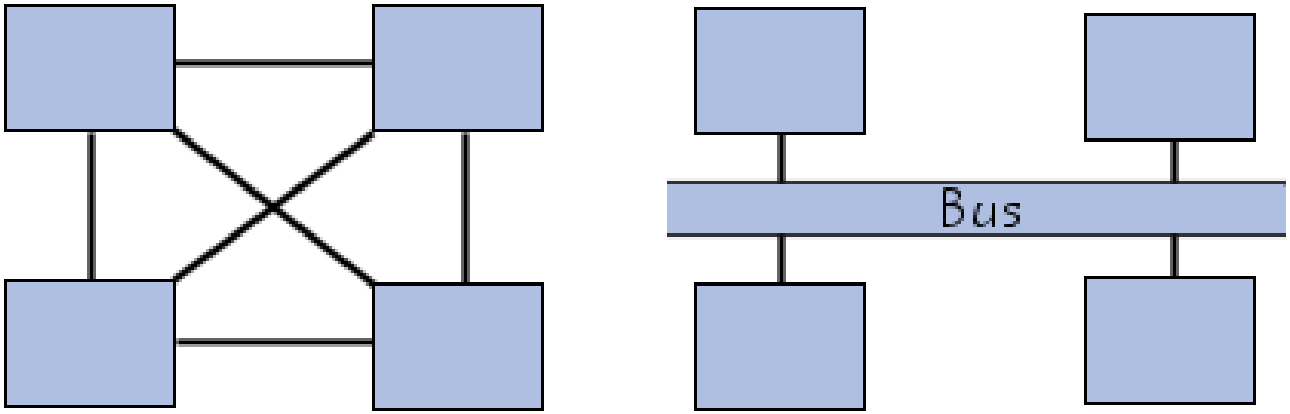
- Introducción
- Bus I2C
- Bus PCI
- Bus USB
- Otros buses

1. Introducción

¿Qué es un bus?:

Un bus es un sistema especializado en enviar y recibir datos entre varios dispositivos.

Un bus es un "medio" de comunicación (cables, circuitos impresos, etc.) compartido por dos o más componentes de hardware.



¿Para qué sirven?

El propósito de buses es reducir el número de "vías individuales" necesarias para la comunicación entre varios dispositivos, a un único canal de datos.

Clasificaciones:

- Síncronos - Asíncronos
- Serie - Paralelo
- Compartido - Dedicado
- Pasivo - inteligente
- Horizontal - Jerárquico
- Cableados – inalámbricos
- Internos – externos

Consideraciones a la hora de diseñar/analizar un bus:

- Estructura física / limitaciones físicas
- Accesibilidad
- Velocidad
- Fiabilidad (Manejo de errores)
- Extensibilidad
- Cuellos de botella
- Ruido (eléctrico)
- Flexibilidad
- Facilidad de Interconexión, PnP o no PnP
- Potencia ¿Lleva alimentación o no?
- Capacidad de acceso compartido (Fan out/in)
- Protocolo de comunicación

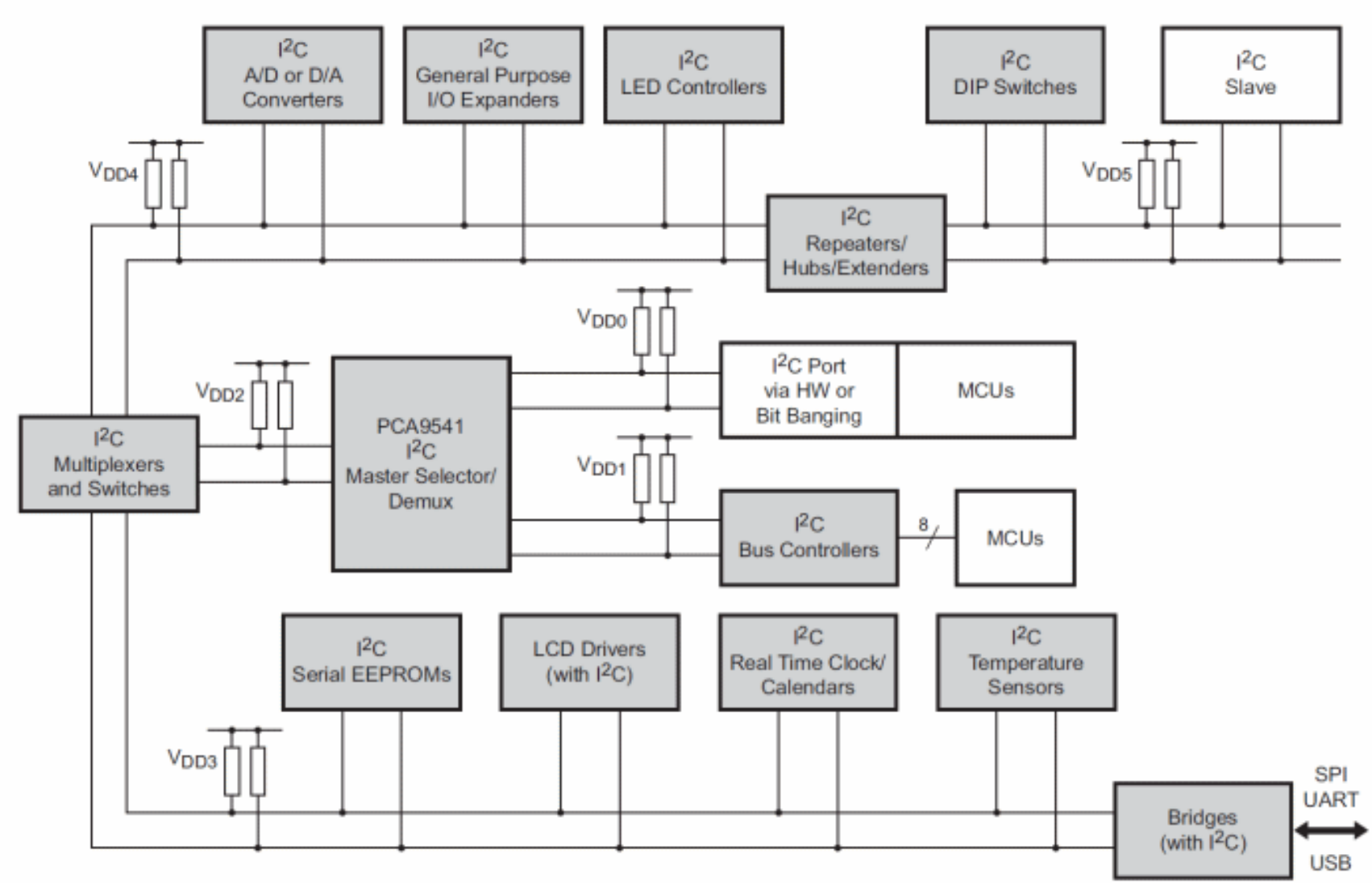
2. Bus I2C

Es un bus pensado para conectar sistemas basados en microcontroladores que no precisen de una alta velocidad

Un sistema embebido típico consiste en una o más microcontroladores y dispositivos periféricos, como memorias, convertidores, I/O, expansiones, controladores de LCD, sensores, conmutadores matriciales, etc. La complejidad y el costo de la conexión de todos los dispositivos entre sí deben mantenerse al mínimo.

El bus I2C permite que los dispositivos más lentos puedan comunicarse con el sistema sin ralentizar los más rápidos.

Topología de un Bus I2C



Características

Los dispositivos I2C pueden ser **maestros** y **esclavos**. Los dispositivos maestros también pueden funcionar como esclavos, cuando sea necesario.

Es un bus serie síncrono: existe sincronización entre la línea de datos (SDA) y la línea de reloj (SCL), lo que permite validar los datos, y una serie de opciones de control en el protocolo I2C

Todos los dispositivos están conectados únicamente con dos cables: SCL y SDA. Ambas señales son bidireccionales. Están conectadas a través de resistencias de polarización (Pull up), a una tensión de alimentación positiva. Esto significa que cuando el bus está libre, ambas líneas están a nivel alto.

La activación de las líneas (SCL y SDA) se realiza “obligando” a nivel bajo. Activas a nivel bajo.

El número de los dispositivos en un único bus es casi ilimitada - el único requisito es que la capacidad del bus no exceda de 400 pF.

Como el nivel lógico alto depende de la tensión de alimentación, no hay tensión de bus estándar. Pero todos los dispositivos de un bus deben trabajar a la misma tensión.

Los dispositivos maestros generan reloj de bus e inician la comunicación en el bus.

Con el fin de comunicarse con un dispositivo específico, cada dispositivo esclavo debe tener una dirección asociada.

Los dispositivos maestros I2C (generalmente microcontroladores) no necesitan una dirección ya que ningún otro (esclavo) dispositivo envía comandos al maestro.

El sentido del enlace: será “half dúplex” (semi-bidireccional), ya que existe solo una línea de datos que podrá utilizarse para el flujo de datos en ambos sentidos, pero nunca simultáneamente.

El bus permite la conexión de varios Masters

Los datos y direcciones se transmiten con palabras de 8 bits.

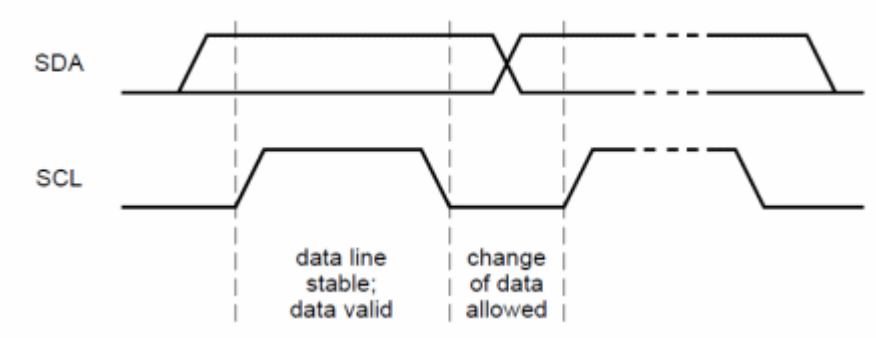
Velocidad de transferencia: debemos adaptar la velocidad entra maestro-esclavo, la tasa máxima de transferencia máxima será entre 100 y 400 kilobits por segundo.

Terminología I2C

Transmisor es el dispositivo que transmite los datos al bus
Receptor: es el dispositivo que recibe los datos del maestro desde el bus
Maestro: es el dispositivo que genera un reloj, se inicia la comunicación, envía comandos I2C y se detiene la comunicación
Esclavo: es el dispositivo que escucha el bus y es direccionado por el maestro
Multi-maestro: I2C puede tener más de un maestro y cada uno puede enviar comandos
Arbitraje: es el proceso que determinar cuál de los maestros puede utilizar el bus, cuando hay más de un maestro
Sincronización de bus: es el proceso para sincronizar los relojes de dos o más dispositivos

Inicio y parada del bus

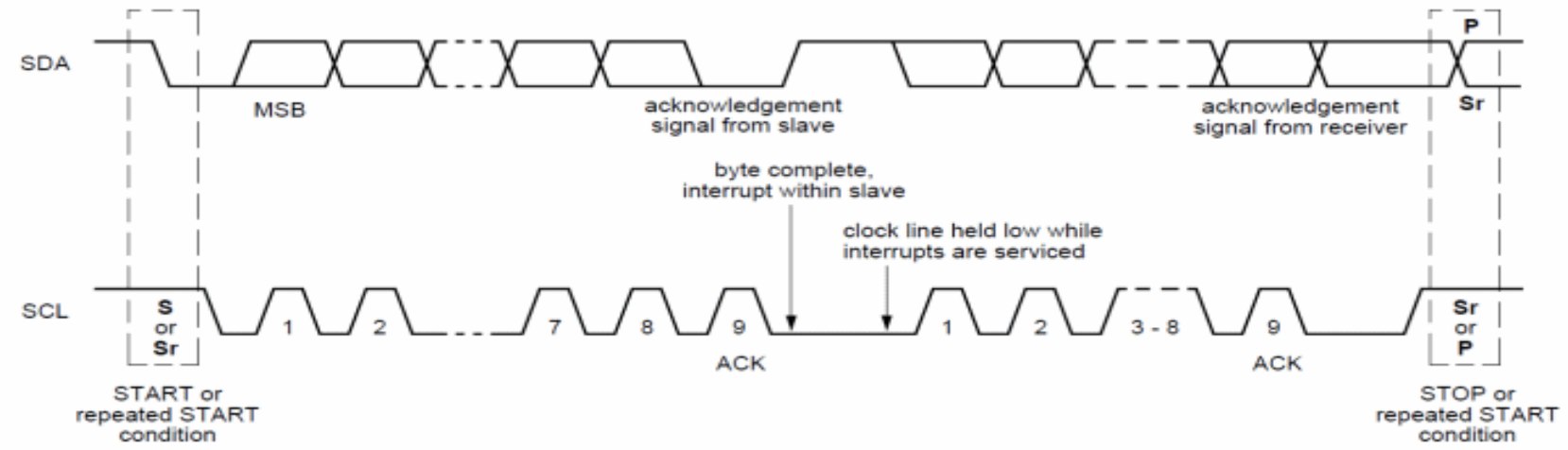
Cada comando I2C enviado por un dispositivo maestro comienza con una condición START y termina con una condición STOP. Para ambas condiciones SCL tiene que estar a nivel alto. Un flanco de bajada en SDA es considerado como START y un flanco de subida, como STOP.



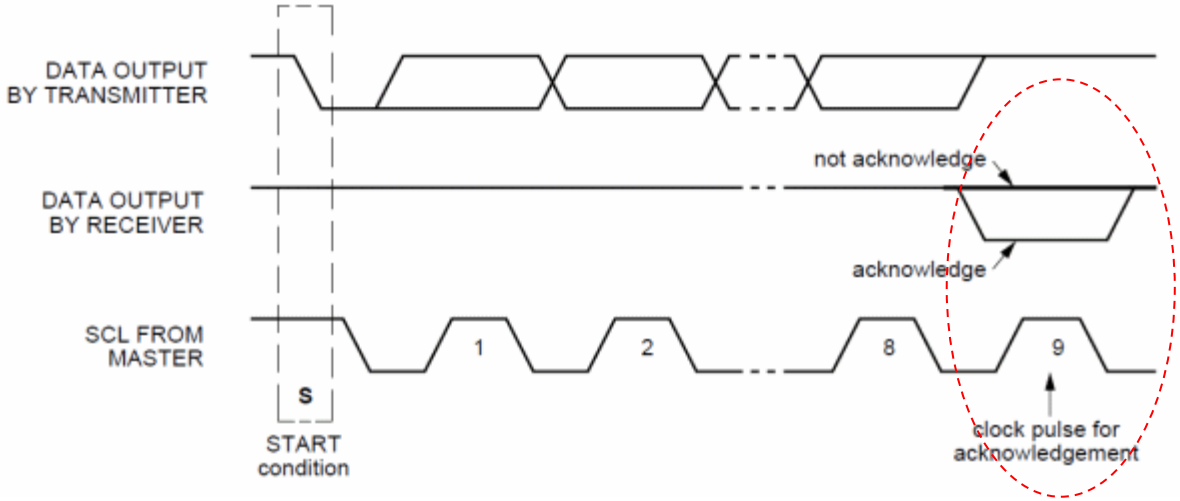
Después de la condición de arranque del bus es considerado como ocupado y sólo puede ser utilizado por otro maestro sólo después de que se detecte una condición de parada.

Después de la condición de inicio el dispositivo maestro puede generar un arranque de nuevo. Esto es equivalente a un arranque normal y por lo general es seguida por la dirección I2C esclavo.

Transferencia de datos en I2C



Los datos sobre el bus I2C se transfieren en paquetes de 8 bits (bytes). No hay limitación en el número de bytes, sin embargo, cada byte debe ser seguido por un bit de reconocimiento. Este bit indica si el dispositivo está listo para proceder con el siguiente byte. Para todos los bits de datos, incluyendo el bit de reconocimiento, el maestro debe generar pulsos de reloj.



Comunicación con direcciones I2C de 7 bits

Cada dispositivo esclavo tiene una dirección única de 7 bits.

La comunicación se inicia con la condición de inicio, seguido de la dirección del esclavo 7 bits y el bit de dirección de datos.

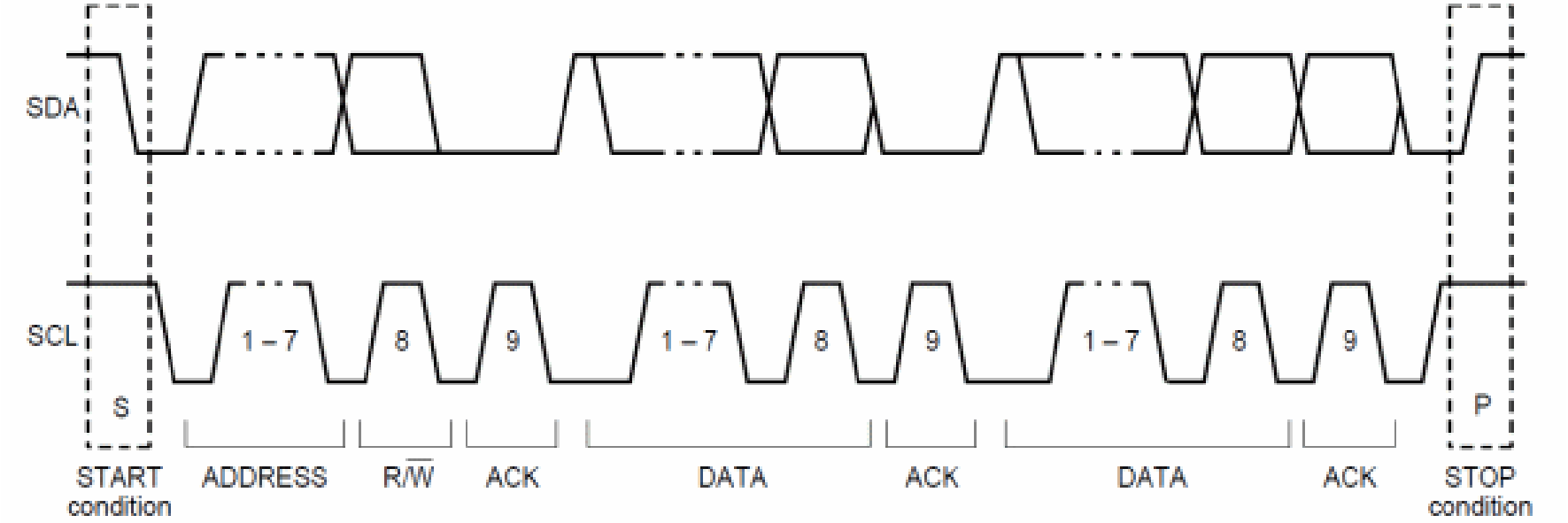
Si este bit es 0, entonces el maestro va a escribir en el dispositivo esclavo. De lo contrario, si el bit de dirección de datos es 1, el maestro lee del dispositivo esclavo.

Después se envía la dirección del esclavo y la dirección de los datos, el maestro puede continuar con la lectura o la escritura.

La comunicación termina con la condición de parada, que también indica que el bus I2C está libre.

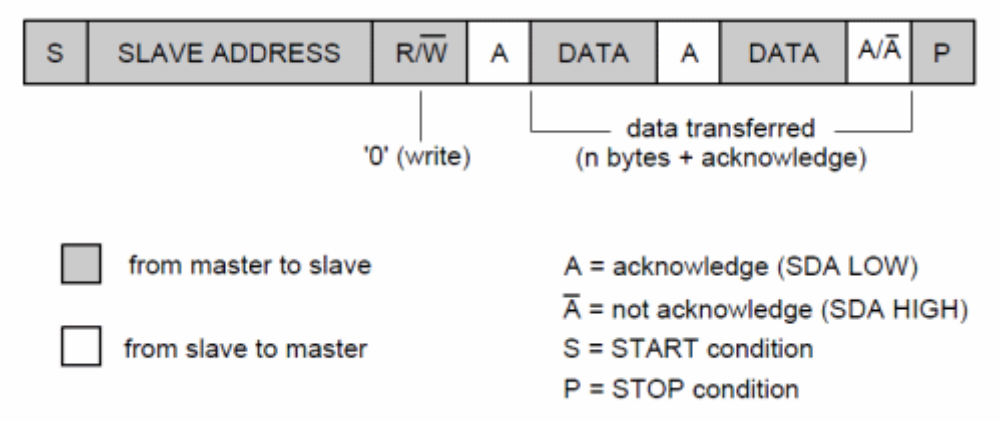
Si el maestro tiene que comunicarse con otros esclavos, puede generar un arranque repetido con otra dirección (esclavo) sin la generación de condiciones de parada.

Todos los bytes son transferidos con el bit MSB cambió primero. Si el maestro sólo se escribe en el dispositivo esclavo a continuación, la dirección de transferencia de datos no se cambia.

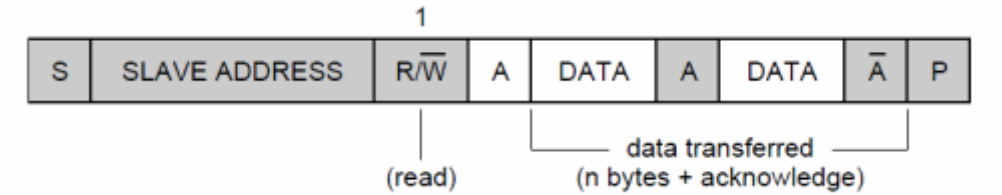


Formato de lectura o escritura de datos

Escritura en un dispositivo esclavo



Lectura de un dispositivo esclavo



Understanding the I2C Bus Enlace
Manejo y aplicaciones del Bus I2C de Arduino Enlace
Application note I2C Enlace

Tareas a realizar: AT

a. Indicar las diferencias y coincidencias entre I2C y SPI
b. ¿Qué podríamos tomar de I2C para completar nuestro proyecto de bus SPI+?

15 minutos

3. Bus PCI

PCI = Peripheral Component Interconnect (Bus)

PCI es un bus habitual debido a las sus características plug-and-play y capacidad para funcionar con el bus de datos de 64 bits

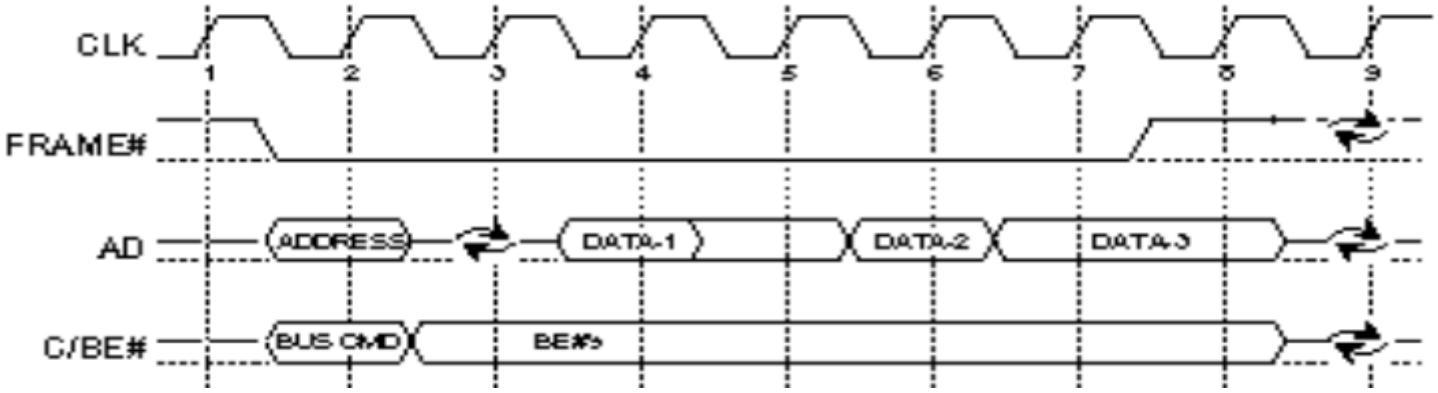
El interfaz PCI contiene una pequeña memoria pequeña con la información de la tarjeta que conectamos.

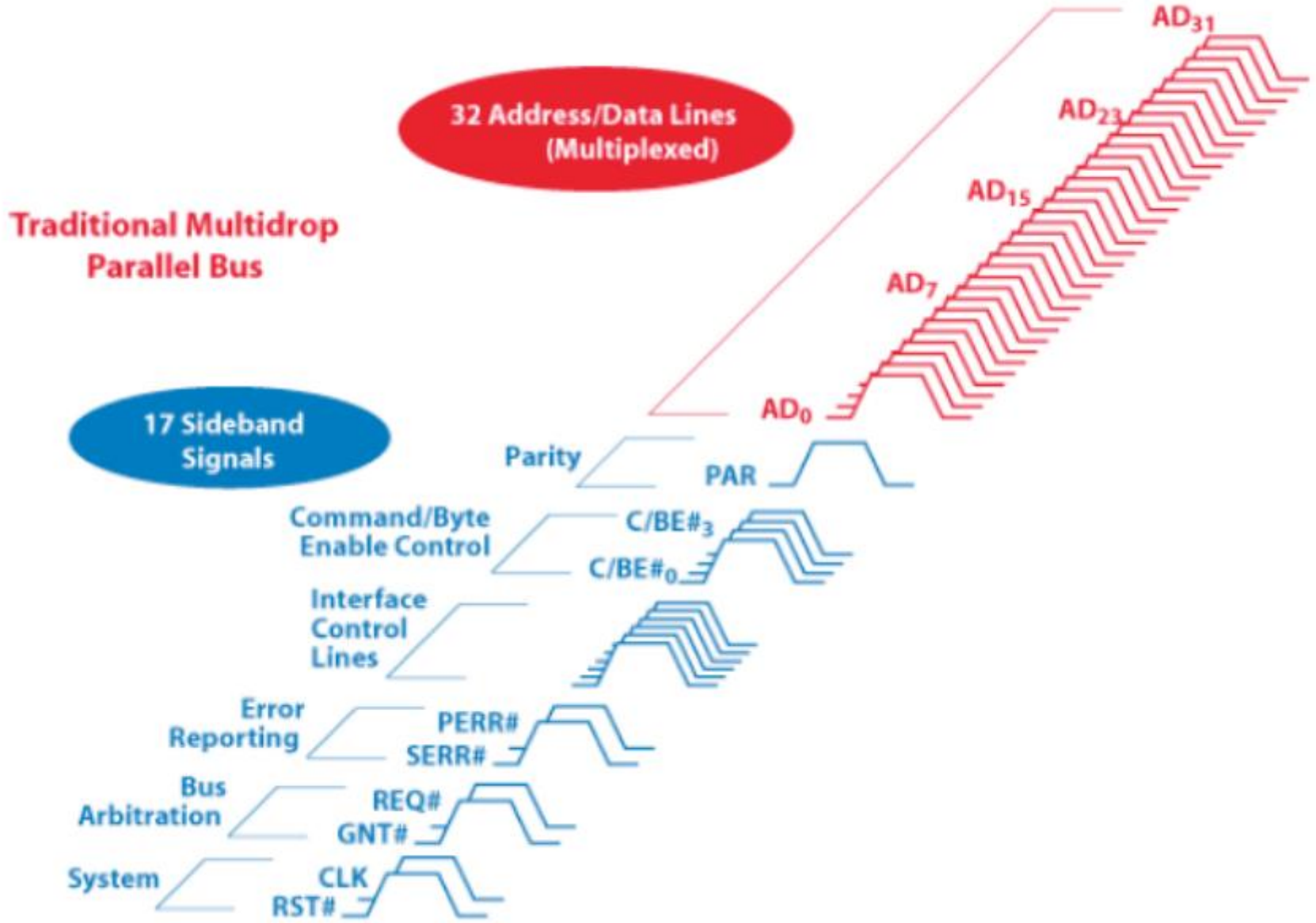
La información de esta memoria permite al ordenador configurar automáticamente la tarjeta (PnP)

El bus PCI se accede a través de un circuito integrado llamado PCI bridge, independiente del tipo de procesador y la arquitectura interna del mismo.

Funciones PCI, permiten utilizar dirección y datos de 32 bits o de 64 bits, indistintamente.

La dirección y los buses de datos se multiplexan para reducir el tamaño del conector.




Traditional Multidrop Parallel Bus
32 Address/Data Lines (Multiplexed)
17 Sideband Signals
AD31
AD23
AD15
AD7
AD0
Parity
PAR
Command/Byte Enable Control
C/BE#3
C/BE#0
Interface Control Lines
Error Reporting
PERR#
SERR#
Bus Arbitration
REQ#
GNT#
System
CLK
RST#

PCI Bus Signals(1)

Signal	Lines	Master	Slave	Description
CLK	1			Clock (33 MHz or 66 MHz)
AD	32	×	×	Multiplexed address and data lines
PAR	1	×		Address or data parity bit
C/BE	4	×		Bus command/bit map for bytes enabled
FRAME#	1	×		Indicates that AD and C/BE are asserted
IRDY#	1	×		Read: master will accept; write: data present
IDSEL	1	×		Select configuration space instead of memory
DEVSEL#	1		×	Slave has decoded its address and is listening
TRDY#	1		×	Read: data present; write: slave will accept
STOP#	1		×	Slave wants to stop transaction immediately
PERR#	1			Data parity error detected by receiver
SERR#	1			Address parity error or system error detected
REQ#	1			Bus arbitration: request for bus ownership
GNT#	1			Bus arbitration: grant of bus ownership
RST#	1			Reset the system and all devices

Mandatory PCI bus signals.

PCI Bus Signals(2)

Signal	Lines	Master	Slave	Description
REQ64#	1	×		Request to run a 64-bit transaction
ACK64#	1		×	Permission is granted for a 64-bit transaction
AD	32	×		Additional 32 bits of address or data
PAR64	1	×		Parity for the extra 32 address/data bits
C/BE#	4	×		Additional 4 bits for byte enables
LOCK	1	×		Lock the bus to allow multiple transactions
SBO#	1			Hit on a remote cache (for a multiprocessor)
SDONE	1			Snooping done (for a multiprocessor)
INTx	4			Request an interrupt
JTAG	5			IEEE 1149.1 JTAG test signals
M66EN	1			Wired to power or ground (66 MHz or 33 MHz)

Optional PCI bus signals.

Especificaciones hardware básicas

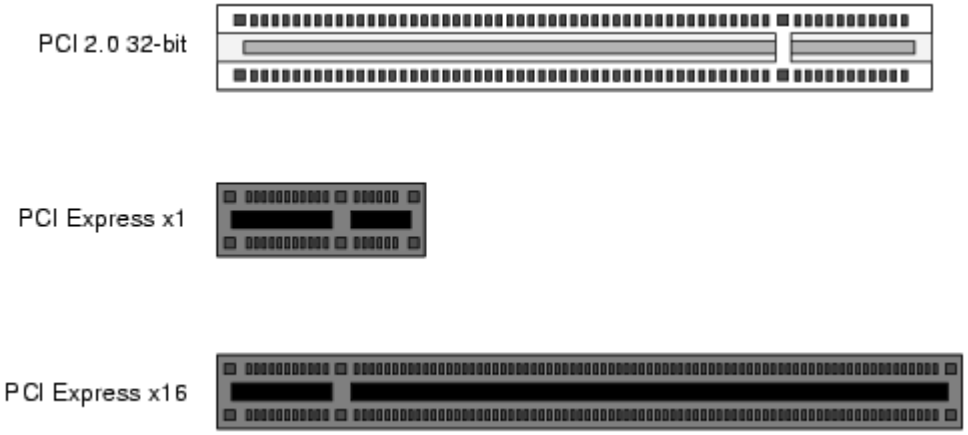
- Reloj de 33,33 MHz con transferencias síncronas
- Ancho de bus de 32 bits o 64 bits
- Tasa de transferencia máxima de 133 MB por segundo en el bus de 32 bits (33,33 MHz × 32 bits ÷ 8 bits/byte = 133 MB/s)
- Tasa de transferencia máxima de 266 MB/s en el bus de 64 bits.
- Espacio de dirección de 32 bits (4 GB)
- 256 bytes de espacio de configuración.
- Alimentación de 3,3 V o 5 V, dependiendo del dispositivo

Mejoras del Bus PCI

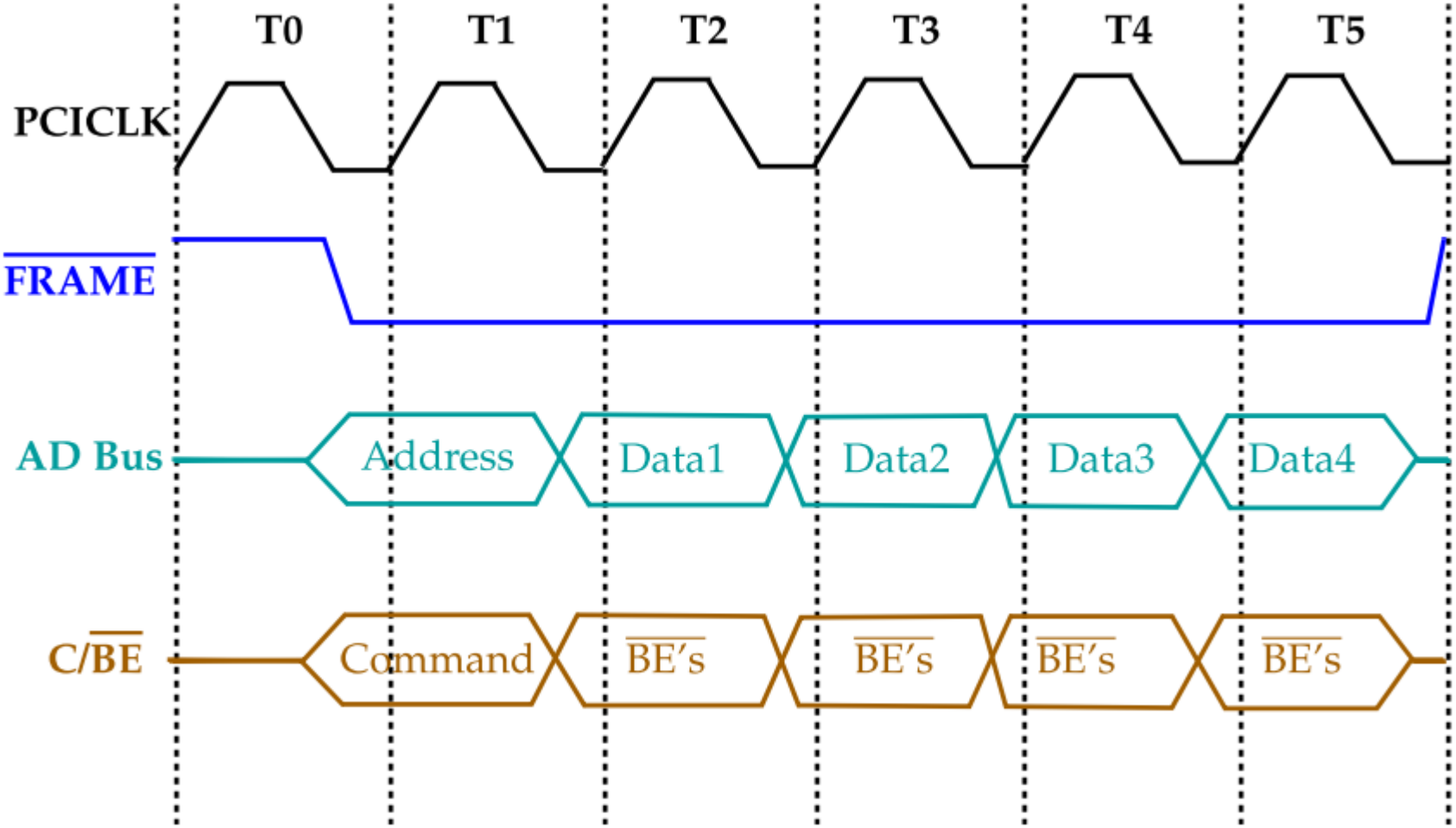
- PCI 2.2 funciona a 66 MHz (requiere 3,3 voltios en las señales) (índice de transferencia máximo de 533MB/s.
- PCI 2.3 permite el uso de 3,3 voltios y señalizador universal, pero no soporta los 5 voltios en las tarjetas.
- PCI 3.0 es el estándar final oficial del bus, con el soporte de 5 voltios completamente quitado.
- PCI-X cambia el protocolo levemente y aumenta la transferencia de datos a 133 MHz (índice de transferencia máximo de 1014 MiB/s).
- PCI-X 2.0 especifica un ratio de 266 MHz (índice de transferencia máximo de 2035 MiB/s) y también de 533 MHz, expande el espacio de configuración a 4096 bytes, añade una variante de bus de 16 bits y utiliza señales de 1,5 voltios.
- Mini PCI es un nuevo formato de PCI 2.2 (portátiles)
- PC/104-Plus es un bus industrial que utiliza las señales PCI con diferentes conectores.
- PXI es la extensión del bus PCI para instrumentación y control.

- PC104/Plus es un bus industrial que utiliza las señales PCI con diferentes conectores.

- PCI Express (anteriormente conocido por las siglas PCIe) es un desarrollo del bus PCI que usa comunicación serie. Consigue aumentar el ancho de banda mediante el incremento de la frecuencia, llegando a ser 32 veces más rápido que el PCI 2.1 ya que, aunque su velocidad es mayor que PCI Express, presenta el inconveniente de que al instalar más de un dispositivo la frecuencia base se reduce y pierde velocidad de transmisión.



Señales y temporización

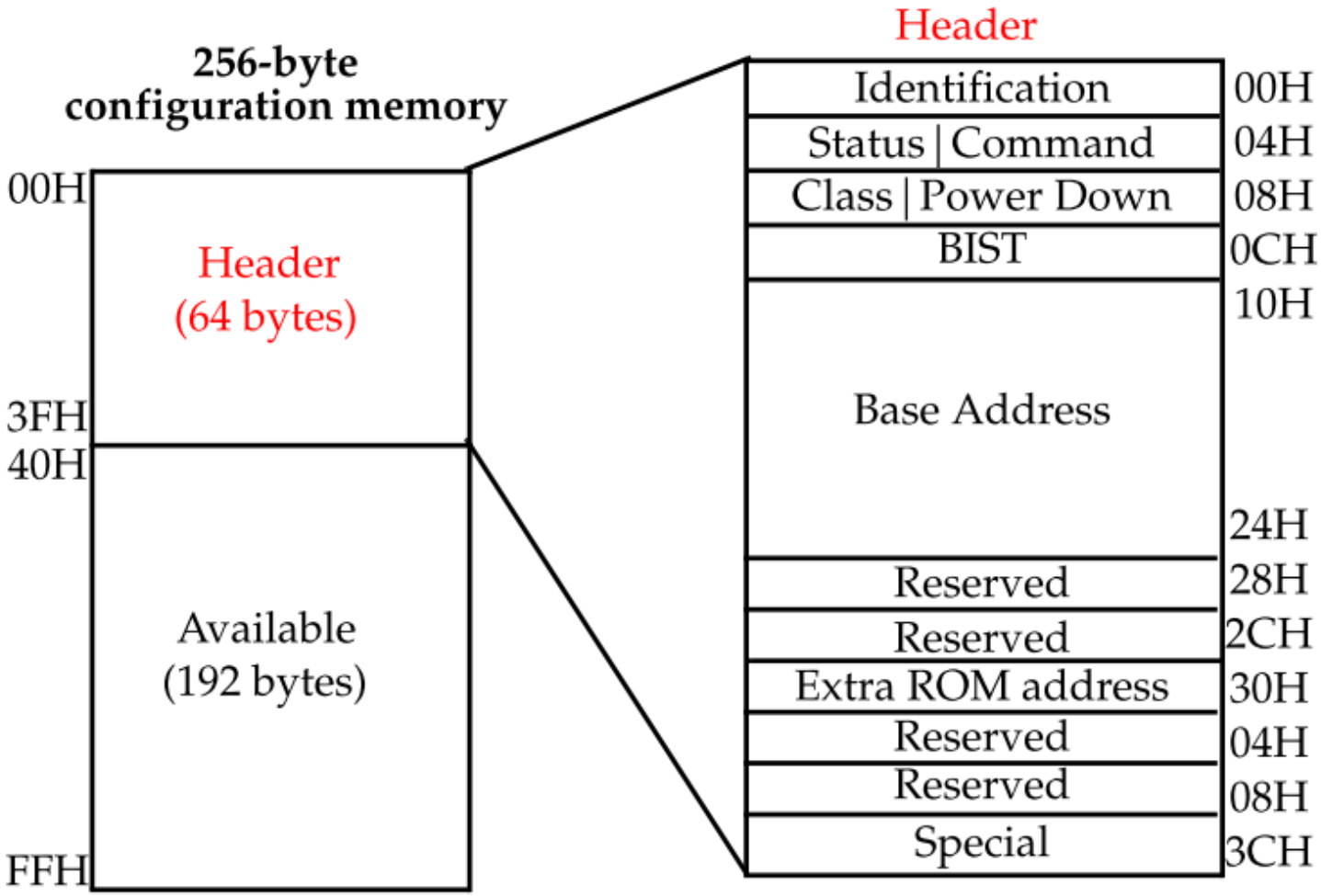


Comandos PCI: Los comandos se envían por medio de los terminales C/BE durante el ciclo T1.

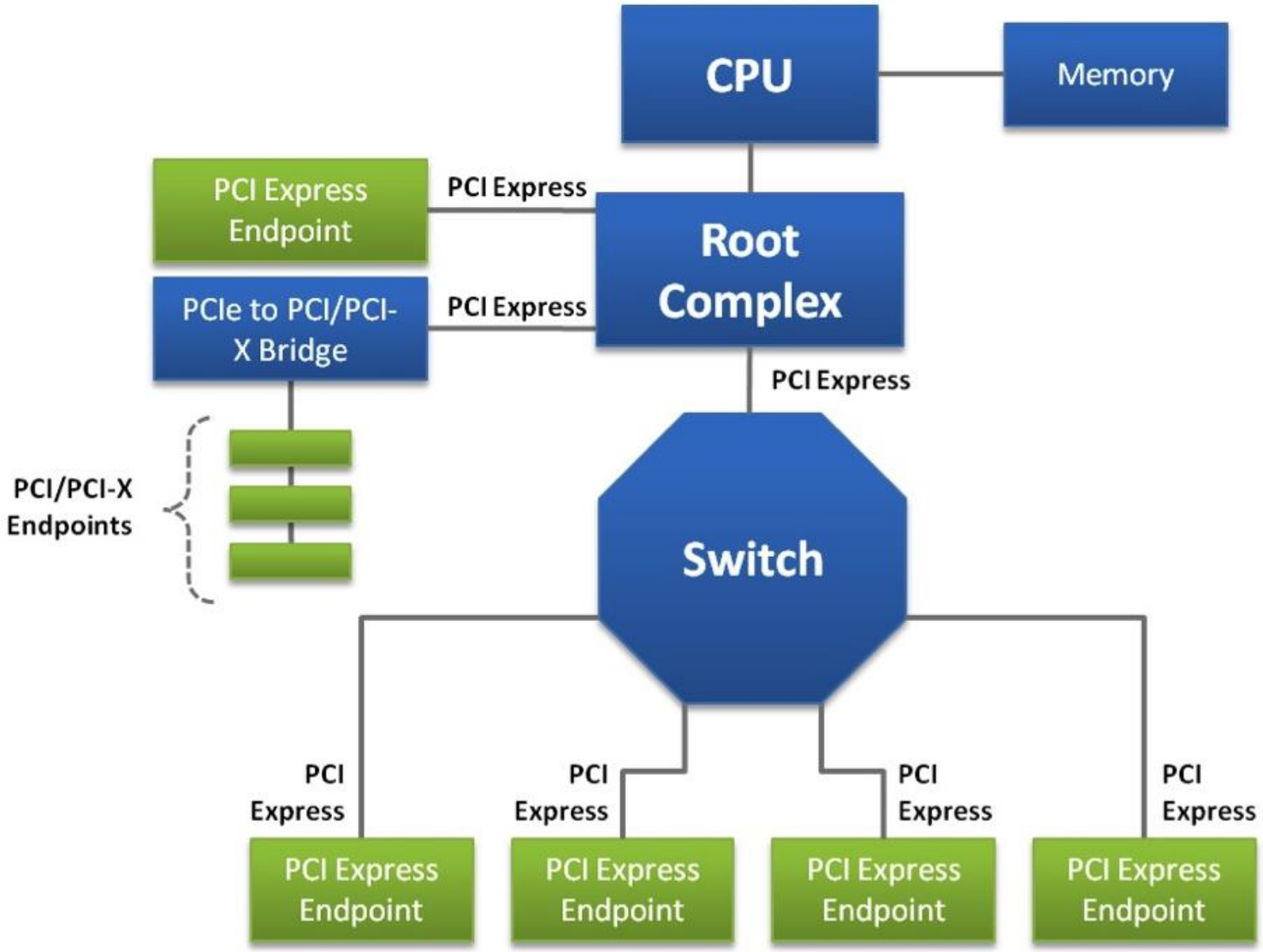
- **INTA Sequence**: Obtener el vector de interrupción del controlador de interrupciones.
- **Special Cycle**: Se utiliza para transferir datos a todos los componentes de PCI, por ejemplo, apagado del procesador.
- **I/O Read Cycle**: Los datos se leen desde un dispositivo de E / S en la dirección AD0-AD15.
- **I/O Write Cycle**: Los datos se escriben en un dispositivo de E / S.
- **Memory Read Cycle**: los datos se leen desde un dispositivo de memoria.
- **Memory Write Cycle**: Los datos se escriben en el dispositivo de memoria

- **Configuration Read**: La información de configuración se lee desde el dispositivo PCI
- **Configuration Write**: La información de configuración se escribe en el dispositivo PCI.
- **Memory Multiple Access**: Se leen varios datos desde la memoria del dispositivo.
- **Dual Addressing Cycle**: Se utiliza para tranferir datos de 64 bits a un dispositivo PCI que sólo tiene disponible un bus de 32 bits.
- **Line Memory Access**: Permite leer más de dos datos de 32 bits.
- **Memory Write with Invalidation**: Actúa igual que Line Memory Access, pero se utiliza para escritura (Write back) en caché.

Espacio de configuración del Bus PCI (actualmente aumentado a 512 bytes)



Topología PCI Express (similar a USB)



3. Bus USB

Generalidades:

- Bus serie orientado a conexión de dispositivos con PC
- Varias versiones 1.0, 1.1, 2.0, 3.0, 3.1… por ahora
- Fácilmente reconfigurable
- Fácilmente ampliable

Facilidad de uso:

- Unificación de conectores y cableado de los periféricos
- Transparencia para los usuarios de detalles técnicos

USB 1.0. El primero en aparecer. Pensado para funcionar con teclados, ratones y dispositivos que en principio necesitan de un ancho de banda pequeño. Permite trabajar a una velocidad aproximada de 1.5 Megabits/s. 1996.

USB 1.1. En este caso su velocidad se multiplica por 8 hasta los 12 Megabits/s. 1998.

USB 2.0. Mejorar la velocidad en 40 veces y llegar a los 480 Megabits/s. Compatible con 1.0 y 1.1. 2000

USB 3.0. Multiplica la velocidad por 10 hasta 4.8 Gigabits. 2008

USB 3.1. Tiene una tasa de transferencia de: 10 GB

USB On-The-Go (OTG): Dispositivos móviles

Wireless USB: (W-USB o WUSB) es un protocolo de comunicación inalámbrica que aúna la sencillez de uso de USB con la versatilidad de las redes inalámbricas. Se basa en la plataforma Ultra-WideBand (WiMedia Alliance), 480 Mbit/s en un entorno de tres metros y 110 Mbit/s en un entorno de diez metros. Actualmente está en todavía desarrollándose. No hay muchos dispositivos

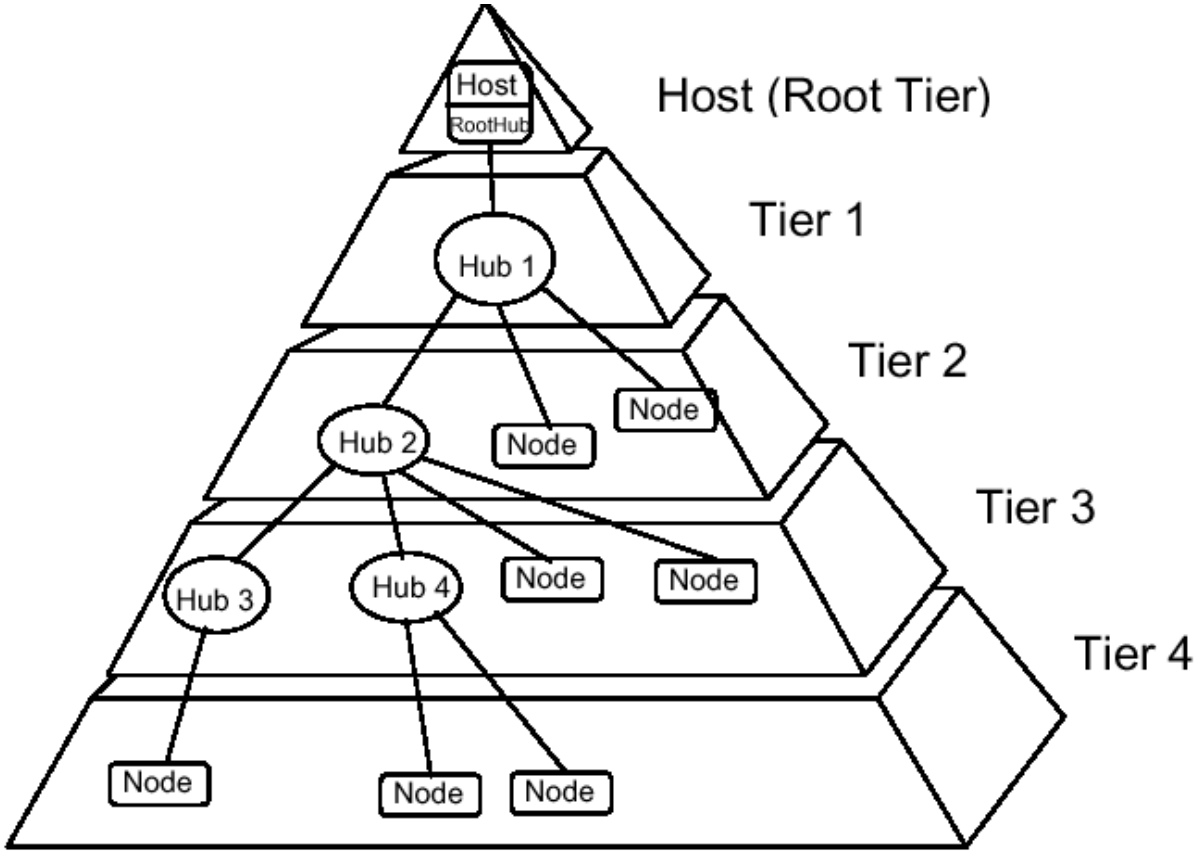
Características generales:

- Reducción del puertos en PC sin limitar la expansión (hasta 127 dispositivos)
- Dispositivos síncronos o asíncronos con el mismo cableado
- Dispositivos de diferente velocidad de transferencia
- Auto identificación y auto configuración de los dispositivos
- Auto detección de nuevos dispositivos en el bus
- Auto detección de baja de dispositivos del bus
- Soporte de "PnP" en caliente
- "Sin necesidad" de alimentación en el dispositivo

Problemas del bus USB:

- Longitud del cable: 5m de máximo y utilizando hubs 30m
- Intercomunicación: no existe posibilidad de que dos hosts o dos dispositivos se comuniquen entre sí, directamente
- Difusión de datos múltiple: no existe la posibilidad de mandar un mismo dato a varios dispositivos a la vez
- Complejidad del protocolo:
  - No se puede acceder a nivel de puerto
  - Son necesarios dispositivos inteligentes
  - Son necesarias varias capas de hard soft para que funcione

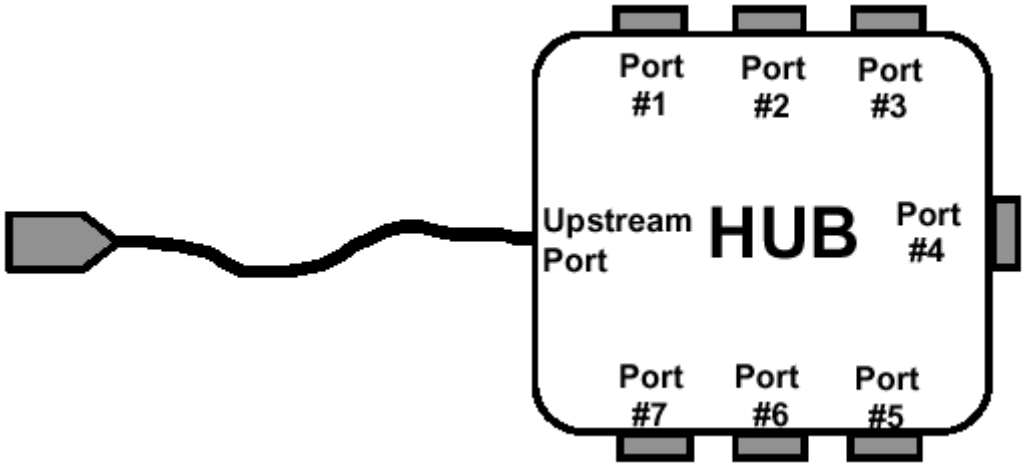
Arquitectura



Elementos: Hubs y Dispositivos

- Hubs: añade conectividad al sistema
- Dispositivo: Proporciona la funcionalidad del sistema

El Hub Raíz reside en el PC



El Host:

Es un PC u otro computador que contiene un controlador host USB y un hub root

Controlador host:

- Da formato a los datos para trasmitir en el bus.
- Traduce los datos recibidos a un formato que el SO pueda entender.

Hub root: Tiene uno o varios conectores para conectar dispositivos.

Las tareas del Host:

- Detección de dispositivos
- Enumeración: el host asigna una dirección y solicita información adicional de cada dispositivo.
- Arbitra BUS.
- Gestiona el flujo de datos: Varios dispositivos pueden querer transferir datos al mismo tiempo, el host debe planificar el tiempo para cada dispositivo
- Detecta Errores
- Suministra y gestiona la energía
- Gestiona el intercambio de datos con periféricos

Funciones del Dispositivo:

- Función USB relacionada con la acción concreta de entrada/salida
- Dispositivo lógico USB: coordina una serie de funciones
- Interfaz de comunicación USB: ofrece a los niveles superiores el servicio de comunicación con el bus
- Detectar comunicaciones dirigidas hacia el
  - Un periférico no puede iniciar una comunicación por sí solo. En cambio este debe esperar y responder a una comunicación del host
  - Cada dispositivo monitorea la dirección de dispositivo contenida en cada comunicación en el bus
- Detección de errores

Elementos lógicos internos a USB:

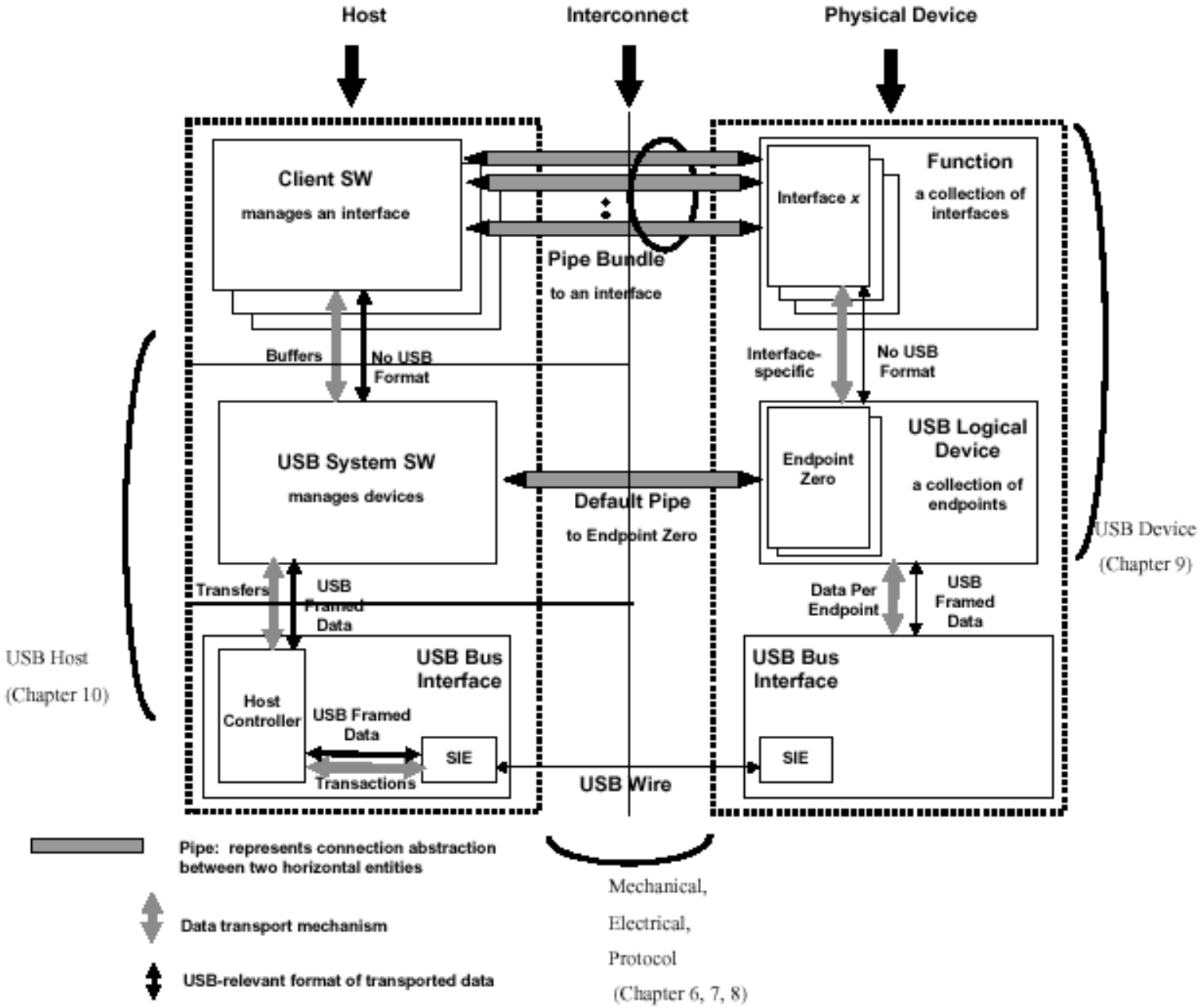
- **Función**: En la especificación USB se define como un dispositivo que proporciona una capacidad al host.
- **Puerto**: En un sentido amplio, un puerto de ordenador es una localización direccionable disponible para conectar circuitos adicionales.
- **Endpoint**: Puerto al que puede accederse indirectamente a través de los drivers del dispositivo USB.
- **Low/full/high-speed**: Modo de funcionamiento en el que la tasa de transferencia del bus es de 1,5/12/480… Mbps.
- **Frame**: Intervalo de 1 ms (125 µs en modo high-speed) de envío y recepción de datos

- **Transacción**: Agrupación de paquetes. Una transacción es de entrada cuando los datos son leídos por el host desde el sistema USB destino, o de salida cuando se transfieren datos desde el sistema al dispositivo USB destino.

- **Transferencia**: Una o más transacciones.

- Otras consideraciones: **downstream** se refiere a una transferencia desde host, y **upstream** a una transferencia hacia el host; **frame** es trama, y **payload** equivale a carga útil.

Flujo de la Información:



Tipos de transferencias

Son formatos de comunicación predefinidos por USB entre los que el usuario puede escoger según sus necesidades

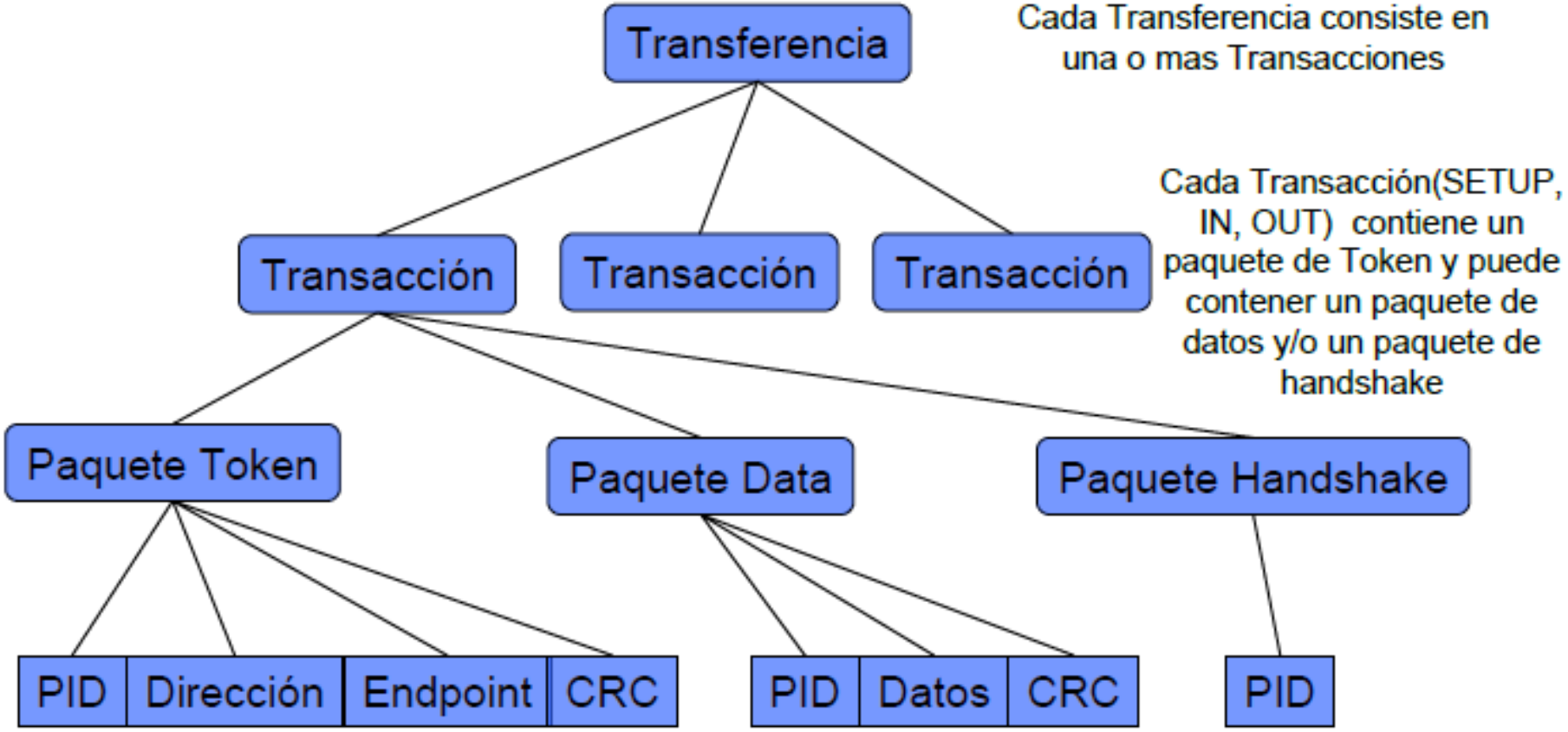
Cada tipo de transferencia tiene un conjunto de características propias de la comunicación:

- Formato de los datos,
- Dirección de la comunicación
- Restricciones de tamaño de paquete
- Control de errores
- Latencia

Transferencias: Planificación

- Una transferencia USB consiste en un grupo de transacciones.
- El host planifica las transacciones dentro de frames.
- La especificación pone límites acerca de esta planificación: No más del 90% de c/frame puede ser usado para transferencias periódicas (iso, int). Mínimo 10% de c/frame para transferencias de control

- La especificación USB define una transacción como la entrega de servicios a un endpoint.
- Las transacciones consisten en uno, dos, o tres paquetes
- Existen tres tipos de transacciones que se definen según el sentido del flujo de datos y el propósito:
  - SETUP
  - OUT
  - IN


Transferencia
Cada Transferencia consiste en una o mas Transacciones
Transacción
Transacción
Transacción
Cada Transacción(SETUP, IN, OUT) contiene un paquete de Token y puede contener un paquete de datos y/o un paquete de handshake
Paquete Token
Paquete Data
Paquete Handshake
PID
Dirección
Endpoint
CRC
PID
Datos
CRC
PID

4. Otros buses

SATA: Serial ATA, S-ATA o SATA (Serial Advanced Technology Attachment)

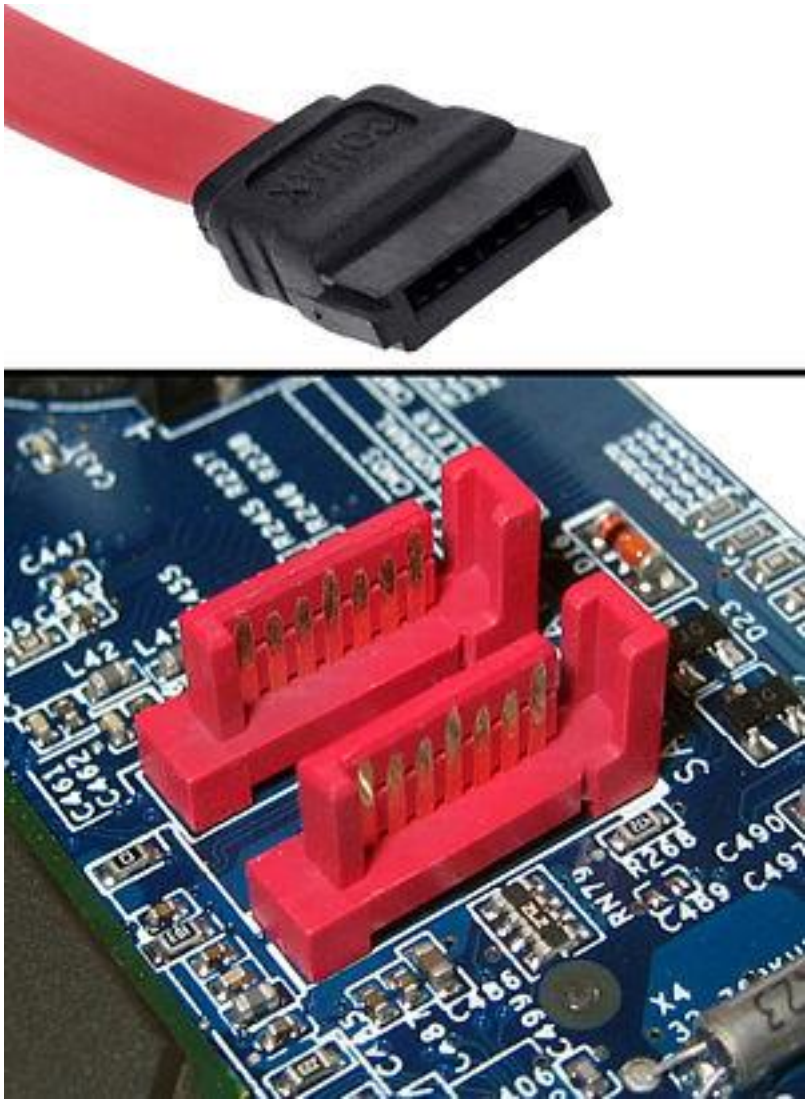
Es un bus de transferencia de datos entre la placa base y algunos dispositivos de almacenamiento, como la unidad de disco duro, lectora y grabadora de discos ópticos (unidad de disco óptico), unidad de estado sólido, etc… Serial ATA sustituye a la tradicional ATA o P-ATA.

SATA aumenta la velocidad de trasferencia, mejora el uso del ancho de banda cuando hay varias unidades, mayor longitud del cable de transmisión de datos y PnP en caliente. Se puede insertar el dispositivo sin tener que apagar la computadora

Características:

	SATA I	SATA II	SATA III
Frecuencia	1500 MHz	3000 MHz	6000 MHz
Bits/clock	1	1	1
Codificación 8b10b	80%	80%	80%
bits/Byte	8	8	8
Velocidad real	150 MB/s	300 MB/s	600 MB/s

Conectores

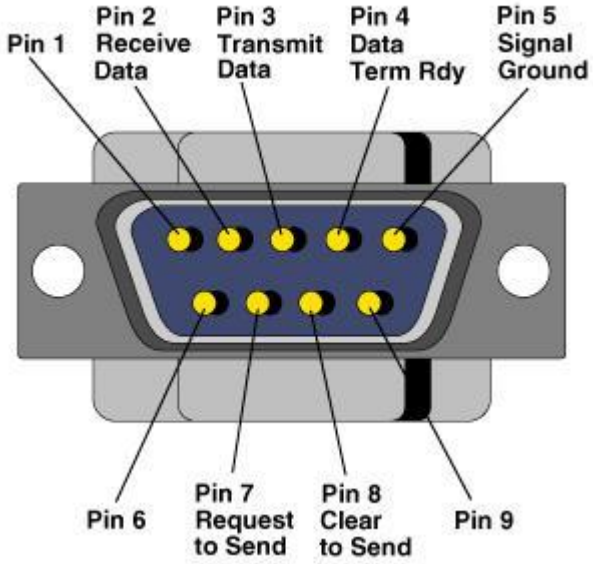


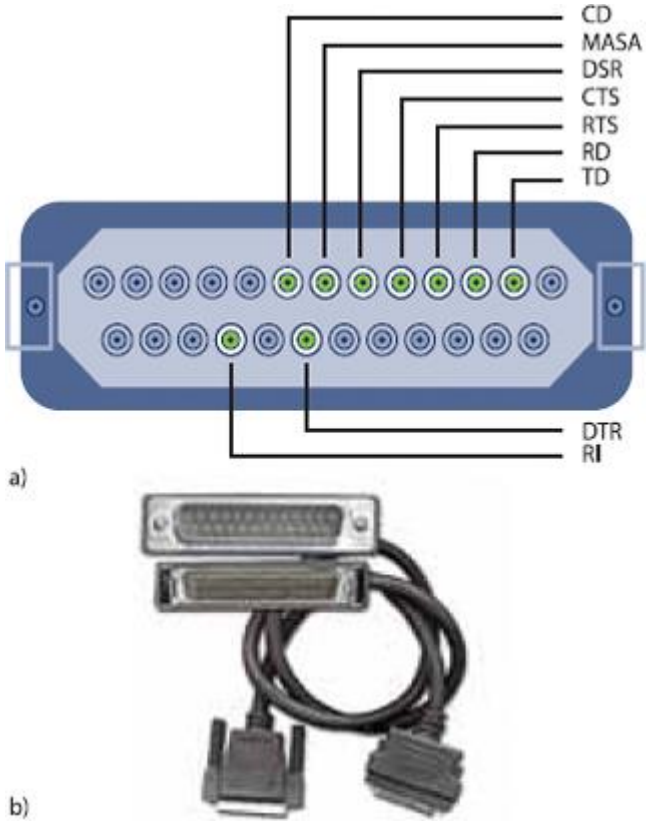
Críticas y problemas

- La organización privada y mercantil que desarrolla el estándar, es poco clara y transparente: no comparte ni hace públicas sus reuniones y votaciones ni las especificaciones del interfaz

- Frente a los conectores grandes y de fuerte acoplamiento de PATA, en SATA se optó por unos conectores más pequeños y de enganche más suave, lo que a veces provoca desconexiones accidentales.

RS 232C: Los puertos serie también conocidos como puertos de comunicaciones (COM) son ya un clásico que no termina de desaparecer. Este puerto/bus de E/S ha acompañado al PC desde hace más de veinte años.





Puede funcionar como:

- DTE (**Equipo terminal de datos**)
- DCE (**Equipo de Comunicación de datos**)

Terminales

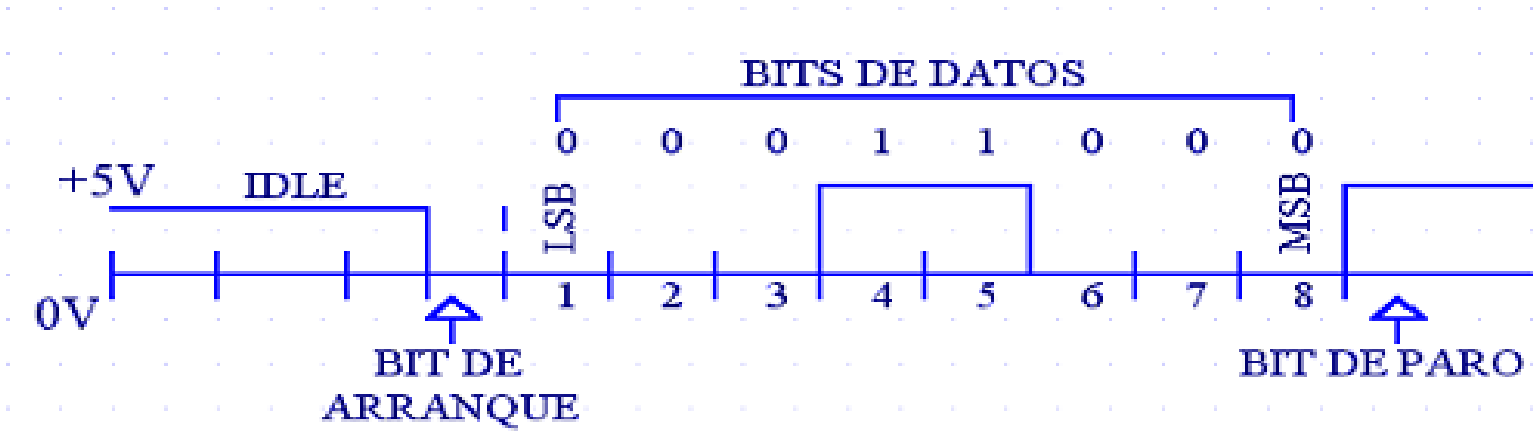
Pin	Función
TXD	(Transmitir Datos) señal de salida
RXD	(Recibir Datos) señal de entrada
DTR	(Terminal de Datos Listo) señal de salida
DSR	(Equipo de Datos Listo) señal de entrada
RTS	(Solicitud de Envío) señal de salida
CTS	(Libre para Envío) señal de entrada
DCD	(Detección de Portadora) señal de entrada

Para la transmisión de datos hay dos opciones:

Por Software: XON/XOFF. Por hardware: RTS y CTS

La transferencia puede ser:

Síncrona: El emisor y el receptor son sincronizados usando una señal de reloj que indica el tiempo entre cada bit.
Asíncrona: porque el receptor se re sincroniza el mismo con el transmisor usando el bit de inicio



FIN