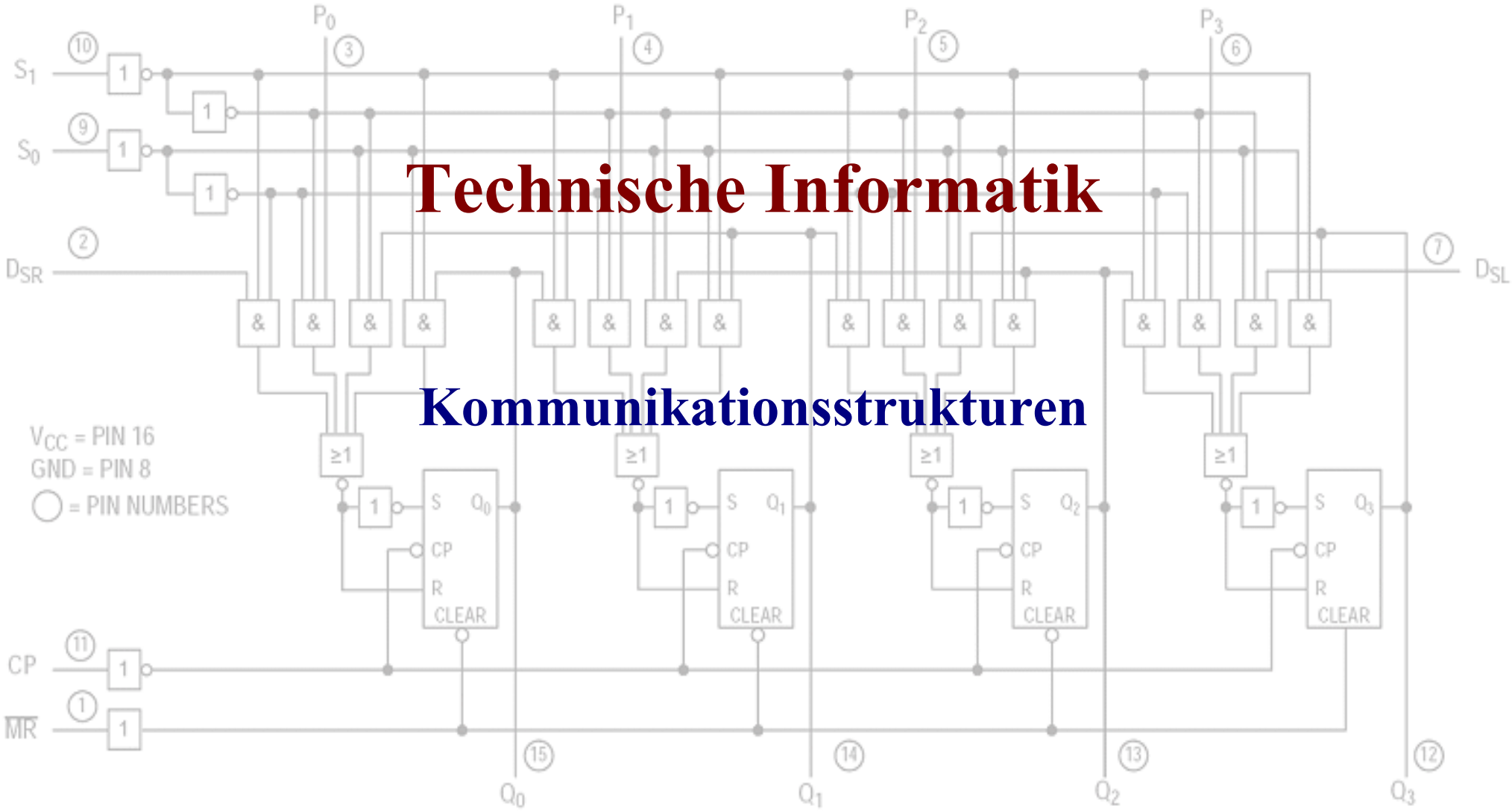
Technische Informatik

Kommunikationsstrukturen

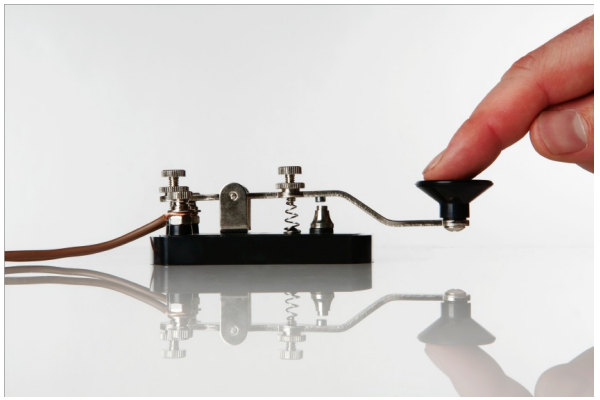


Eigenschaften von Kommunikationswegen

- Eine Form der Kommunikation: Der „Morse-Code“

▪▪▪▪ ▪ ▪▬▪▪ ▪▬▪▪ ▬▬▬ ▪▬▬ ▬▬▬ ▪▬▪ ▪▬▪▪ ▬▪▪

- Welche Anforderungen an Kommunikationsstrukturen gibt es?
- Welche charakteristischen Merkmale haben Kommunikationswege?

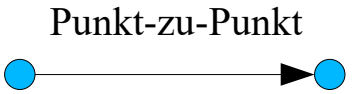


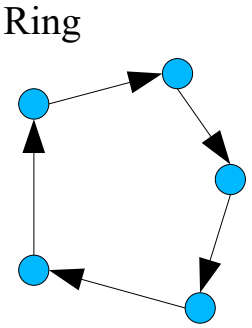
Eigenschaften von Kommunikationswegen

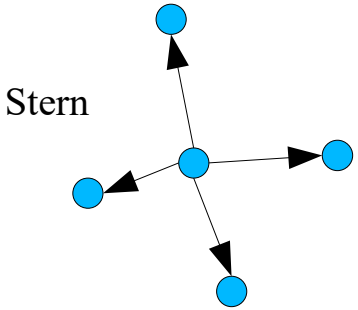
- Kommunikation = Nachrichtenübertragung
  - Dazu: Nachrichten bzw. Informationen müssen physikalisch dargestellt und interpretiert werden
- Übertragung
  - Erfordert ein Übertragungsmedium bzw. einen Träger:
    - Elektrisch, optisch, akustisch, drahtlos, …
  - Die zu übertragenen Informationen müssen zunächst geeignet dargestellt werden
    - Daten-Codierung
  - Anschließend erfolgt physikalische / nachrichtentechnische Umsetzung
    - Leitungscodierung/Modulation
    - Signalpegel, Funkmodulation etc.

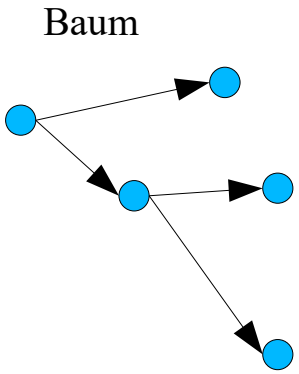
Eigenschaften von Kommunikationswegen

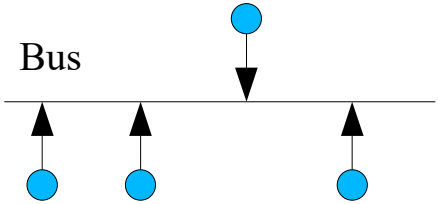
- Topologie:
  - 1-zu-1, 1-zu-N (Multicast), 1-zu-alle (Broadcast)
  - Punkt-zu-Punkt, Ring, Baum, Stern, Bus...
  - Kommunikationsrichtung:
    - simplex: Immer nur in eine Richtung
    - duplex: In beide Richtungen, auch gleichzeitig
    - semiduplex: In beide Richtungen, aber nur abwechselnd









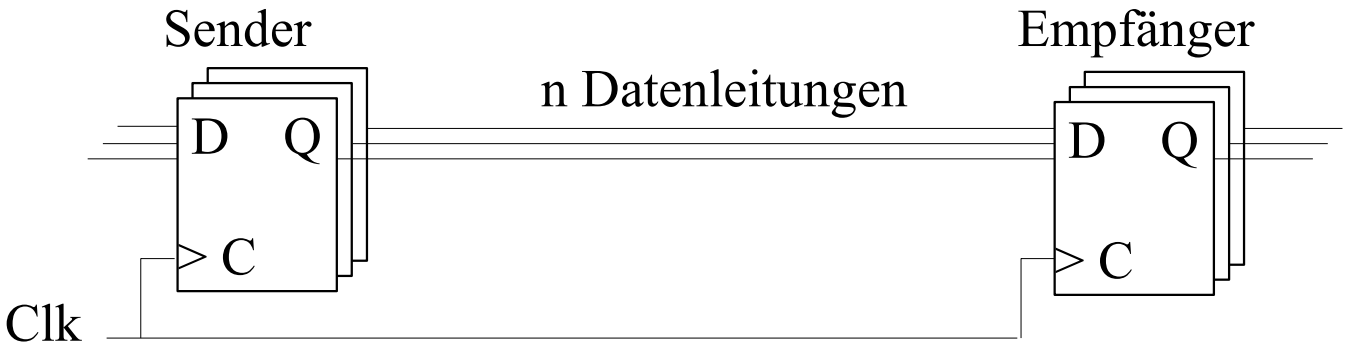


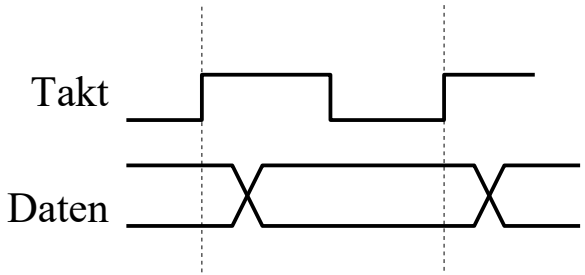
Eigenschaften von Kommunikationswegen

- Zeitverhalten:
  - Daten können synchron, asynchron oder isochron übertragen werden
  - Kenngrößen:
    - Datenrate (Datenmenge pro Zeit)
    - Latenz (Zeitverzug zwischen Sender und Empfänger)
- Datenflusskontrolle:
  - Handshaking
    - Master-Servant: Master kontrolliert alleine Buszugriff
    - Multi-Master: Erfordert Bus-Arbitrierung
  - Auf- und Abbau der Verbindung, ggf. im laufenden Betrieb: „hot plugging“
  - Ziel-Adressierung
  - Fehlererkennung und Korrektur

Synchrone Datenübertragung

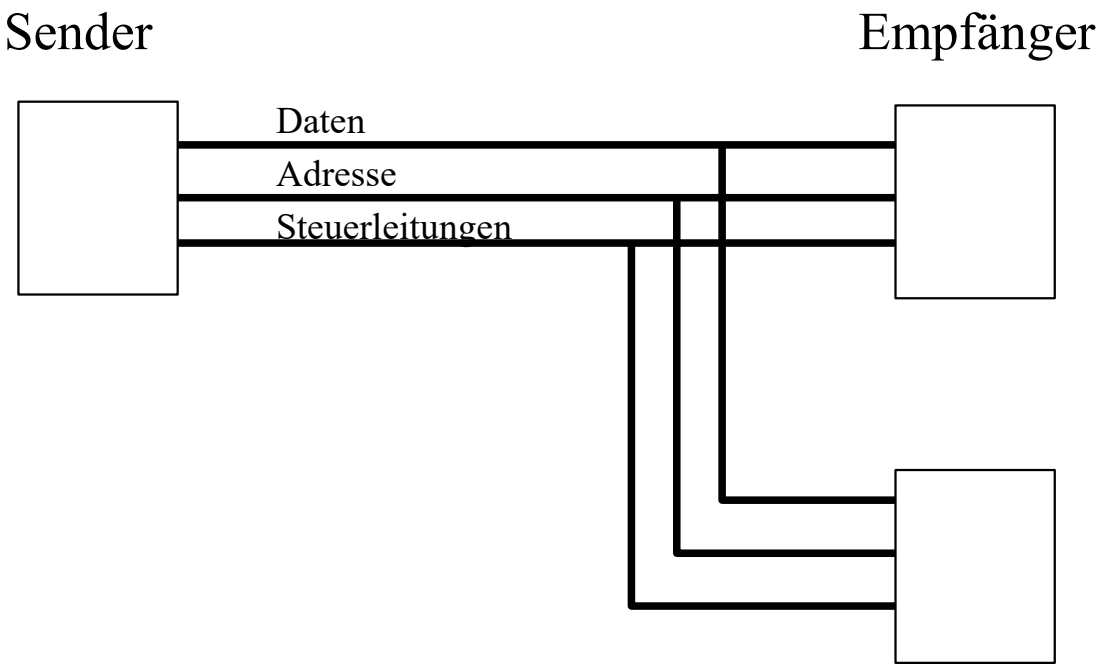
- Ausgangspunkt: Schaltwerk mit Datenübertragung zw. Speicherelementen
- Hier:
  - n Datenbits werden taktsynchron von einem (Speicher-) Element zu einem anderen parallel übertragen. Datenübernahme mit steigender Taktflanke





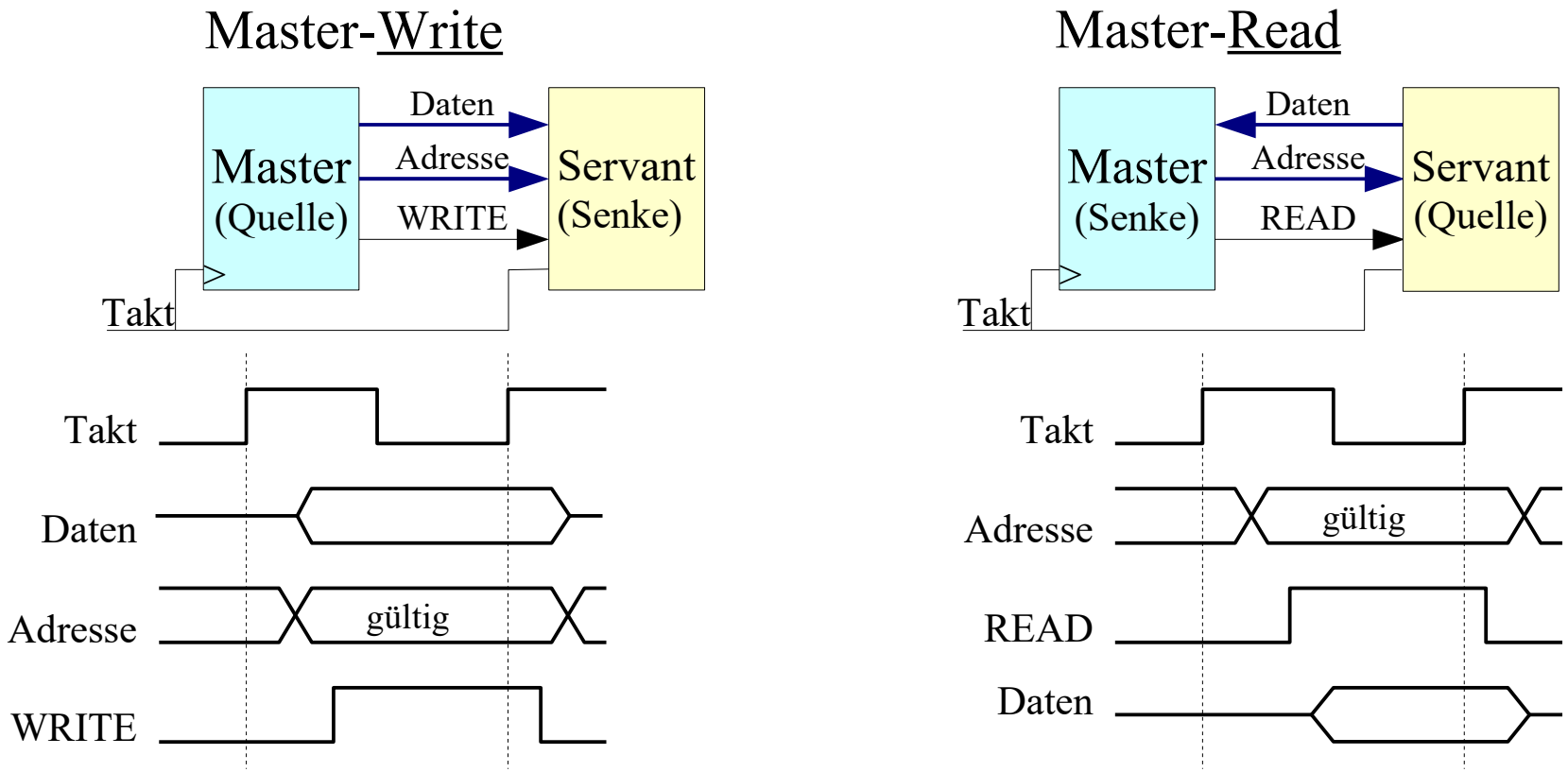
Synchrone Datenübertragung

- Adressierung:
  - Mit zusätzlichen Adressleitungen können Daten unterschiedlichen Empfängern (oder innerhalb eines Empfängers unterschiedlichen Registern) zugeordnet werden.



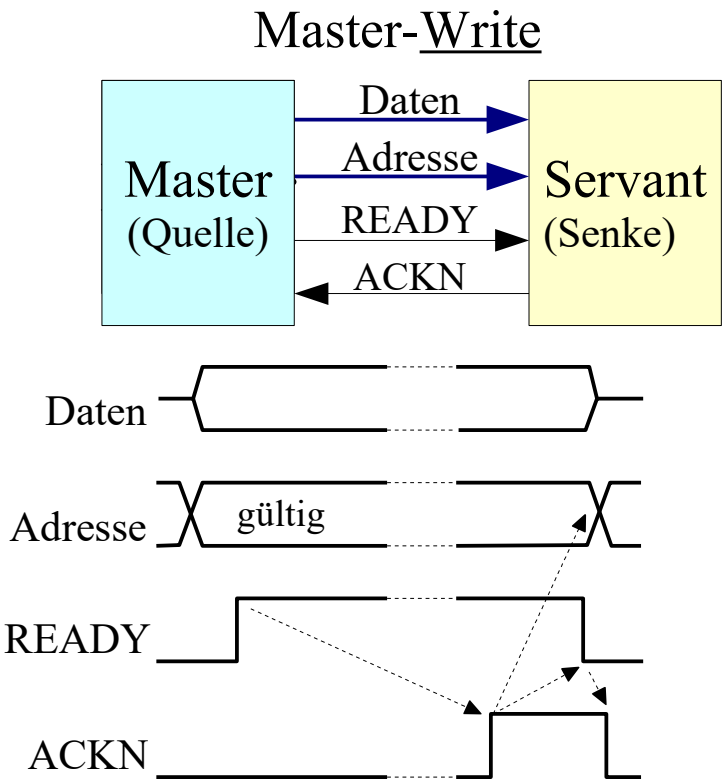
Synchrone Datenübertragung

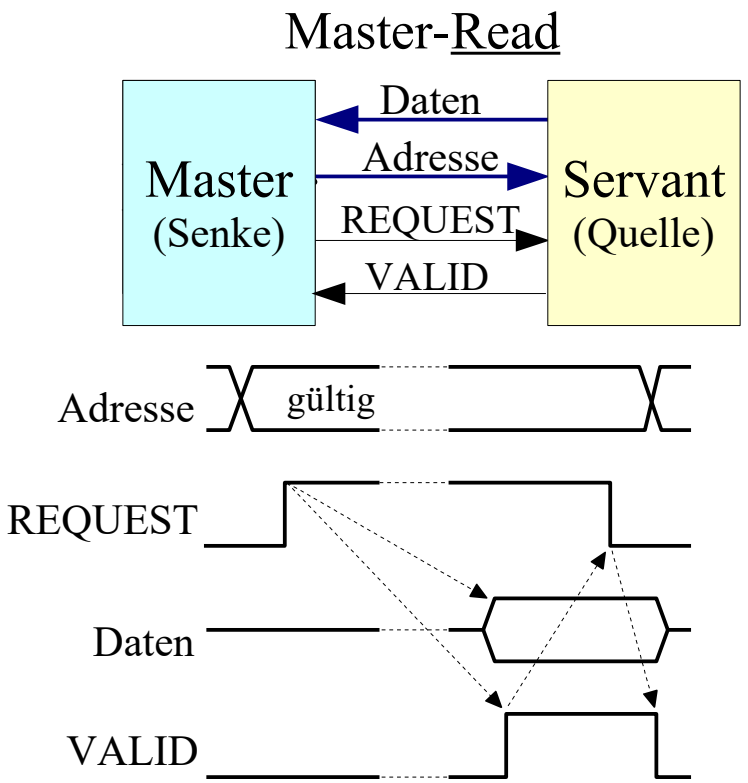
- Bidirektionale synchrone Datenübertragung mit Handshake-Signalen (READ/WRITE)
  - Master-Write: Master stetzt Adress-/Datenleitungen und signalisiert deren Gültigkeit mit dem WRITE-Signal. Datenübernahme durch Servant mit steigender Taktflanke
  - Master-Read: Master setzt nur Adress-Leitungen und fordert Servant mit READ-Signal auf, Daten bereitzustellen. Datenübernahme durch Master mit steigender Taktflanke



Asynchrone Datenübertragung

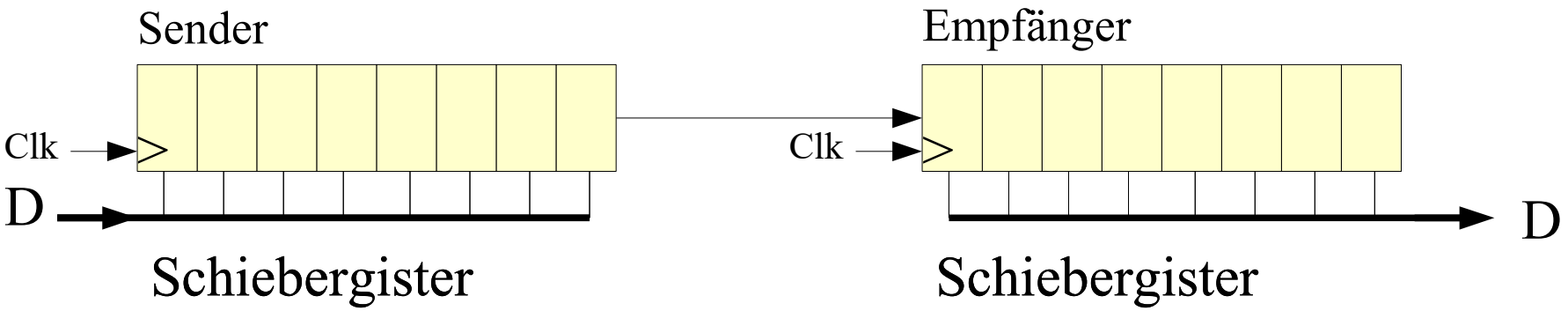
- Bidirektionale asynchrone Datenübertragung mit Handshake-Signalen
  - Master-Write: Master setzt Adress-/Datenleitungen und signalisiert deren Gültigkeit mit dem READY-Signal. Datenübernahme durch Servant, der diese anschließend durch ACKN bestätigt. Master nimmt daraufhin READY-Signal zurück.
  - Master-Read: Master setzt nur Adress-Leitungen und fordert Servant mit REQUEST-Signal auf, Daten bereitzustellen. Datenübernahme durch Master, sobald Servant VALID signalisiert. Master nimmt REQUEST zurück.





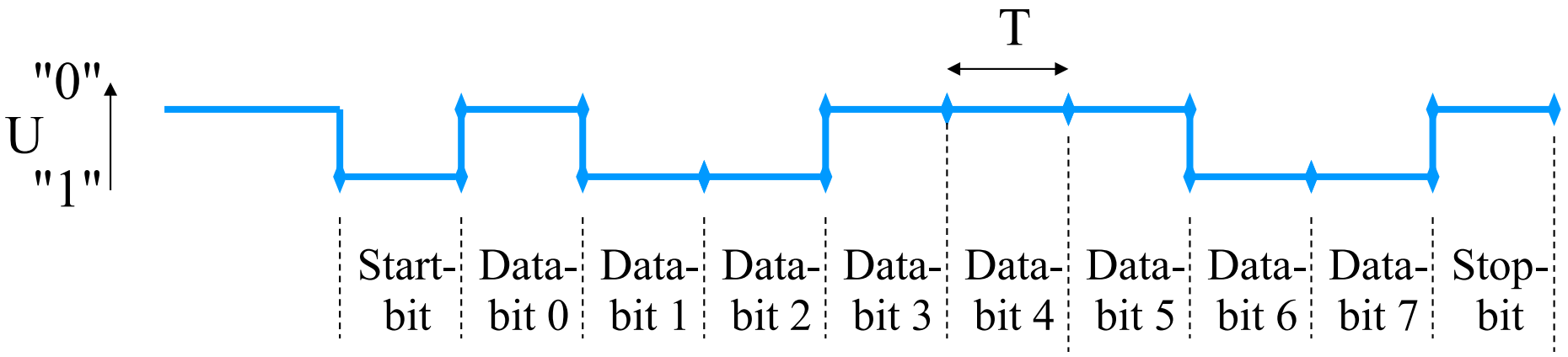
Serielle Datenübertragung

- Serielle Datenübertragung
  - Datenwort (typisch: 8 Bit = 1 Byte) wird in ein Schieberegister geladen. Anschließend wird der Inhalt des Registers taktsynchron an den Empfänger übertragen
  - Im Empfänger wird das Signal ebenfalls taktsynchron in ein Schieberegister geschoben. Nach N Takten liegt im Empfänger das Datenwort vor
  - Auch bei serieller Datenübertragung: Adress- und Steuerinformationen, Handshake etc. ggf im Protokollframe abgebildet



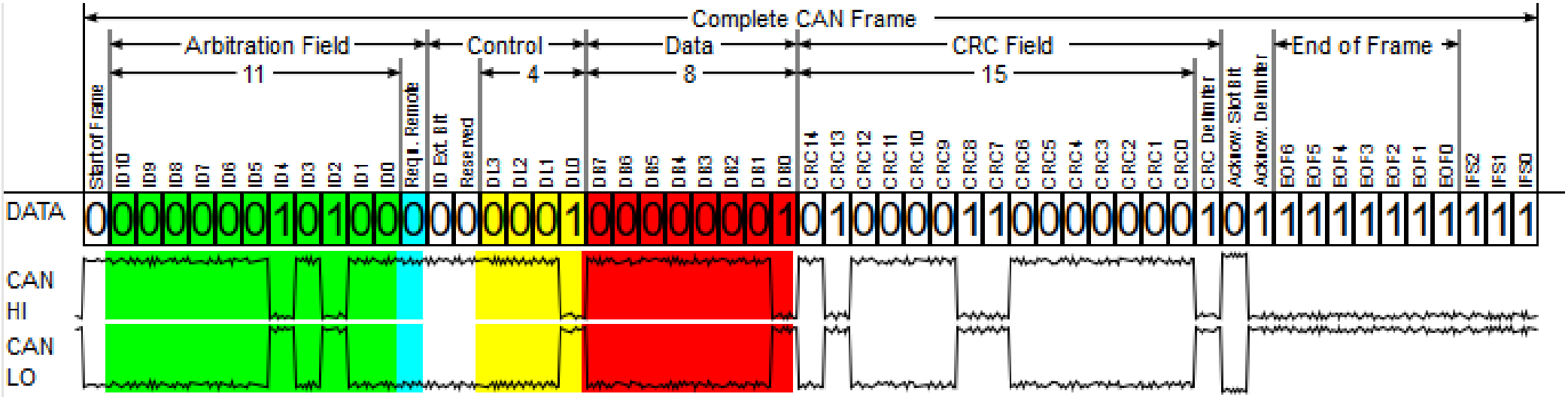
Serielle Datenübertragung

- Vereinbarung einer festen Datenrate
  - T = "Dauer eines Bits" und zulässige Toleranz ΔT
- Vereinbarung eines Startsignals
  - "Jetzt geht's los"
- Vereinbarung der Paketlänge
  - "Synchronisationsrahmen"; Lesbarkeit trotz Toleranz ΔT
- Verfahren kompensiert Taktabweichungen zwischen Sender und Empfänger



Serielle Datenübertragung: CAN

- Beispiel für serielle Datenübertragung:
- Controller Area Network (CAN)
  - Multi-Masterfähiger Bus, Anwendung im Automobilbereich, aber auch Medizin-Automatisierungstechnik, Avionik, …
  - Differentielle Datenleitungen (gespiegelte Signalpegel)
  - Nutzdaten werden in Frames „gepackt“, die Zusatzinformationen (Acknowledge, Fehler, etc.) enthalten

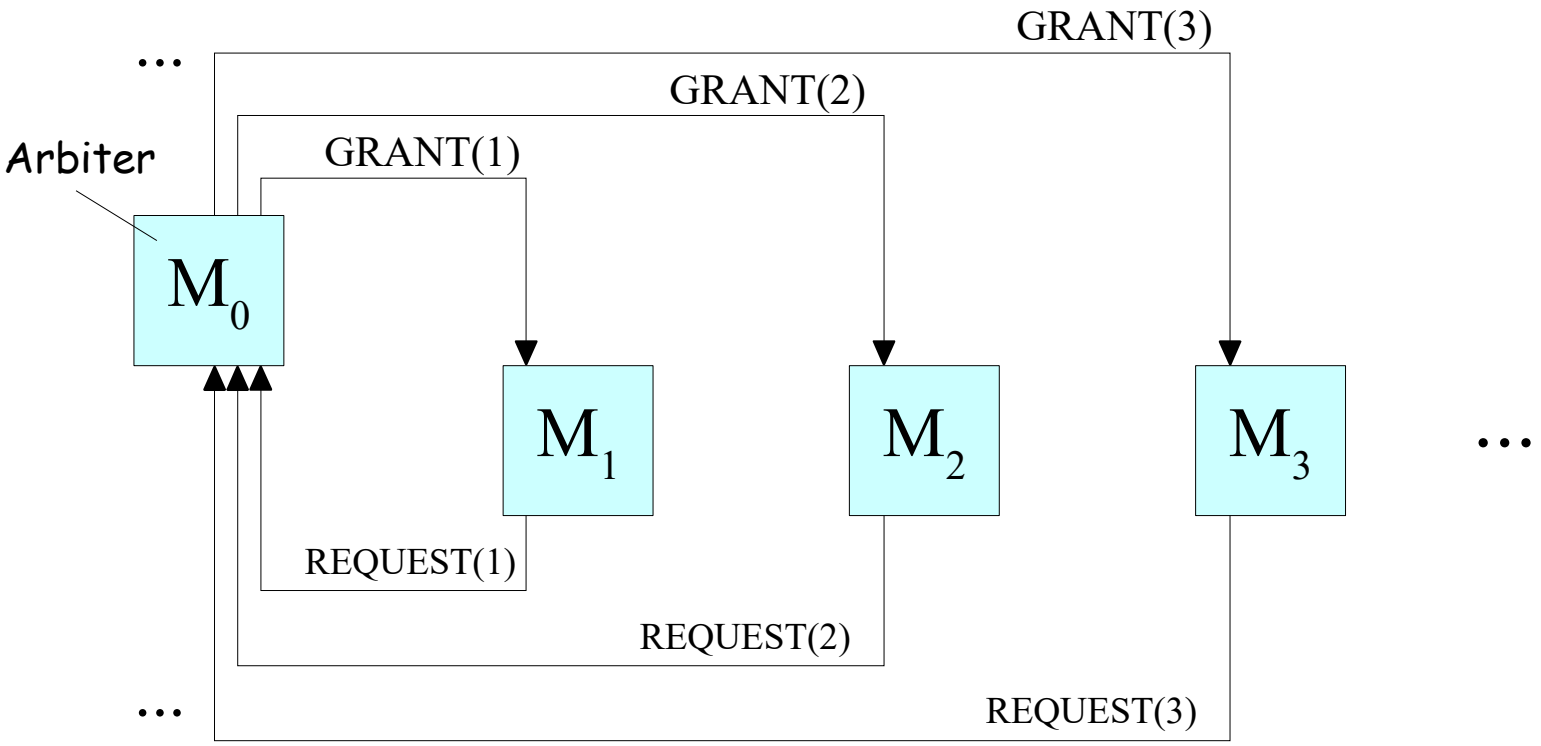


Arbitrierung

- Arbitrierung (arbiter = Schiedsricher, Gebieter) regelt Buszuteilung, wenn mehrere Busmaster vorhanden sind:
  - Ein Master (Arbiter) nimmt Transfer-Anfrage von Teilnehmer entgegen
    oder: Mehrere Master "verhandeln" über den Buszugriff
  - Arbiter weist den Bus gemäß den vorgegeben Kriterien zu
    oder: Der "Sieger" der Verhandlungen reserviert den Bus für sich
  - Der aktive Bus-Teilnehmer signalisiert, dass er aktiv ist
  - Der aktive Bus-Teilnehmer erzeugt Zieladressen und schickt Daten mit den vereinbarten Protokollen
  - Die Kommunikation wird beendet und der Bus freigegeben
    oder: Ein anderer Teilnehmer "beantragt" die Freigabe des Busses. Gemäß der vorgegeben Kriterien (Liste der Prioritäten, Wartezeiten, ..) wird der Antrag bearbeitet

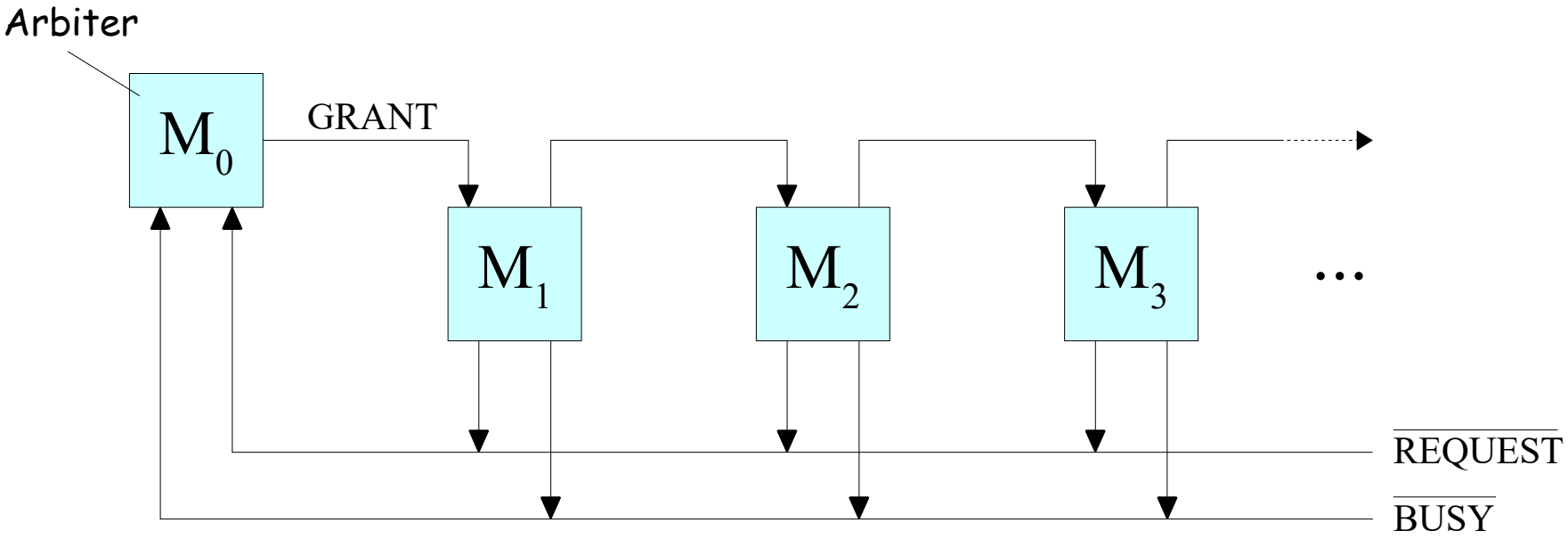
Zentrale Arbitrierung mit Stichleitungen

- Ablauf:
  - Master $\mathbf{M}_i$ setzt REQUEST(i)
  - Arbiter erteilt GRANT(i)



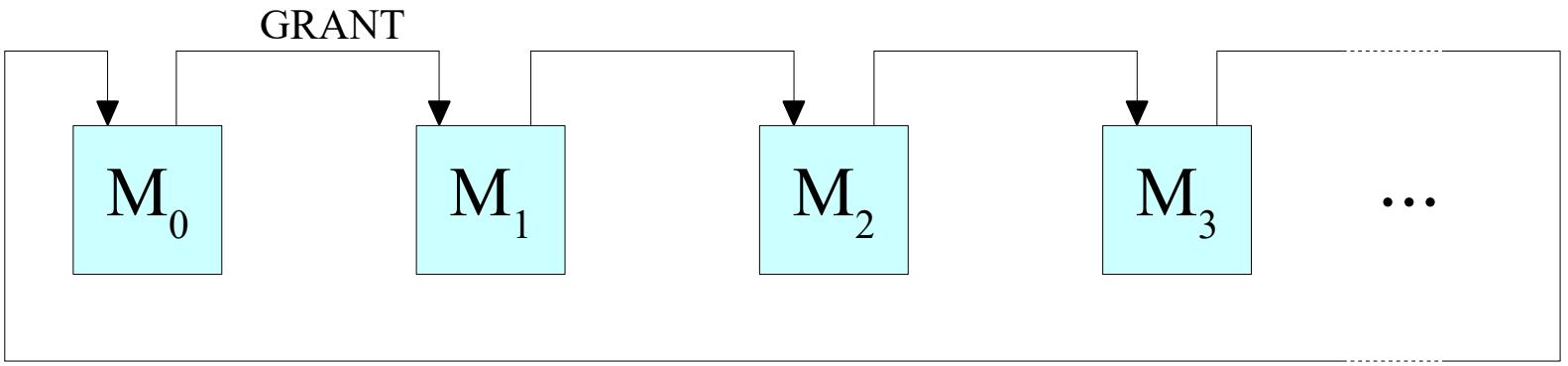
Zentrale Arbitrierung mit Daisy-Chain

- Ablauf:
  - Master $\mathbf{M}_i$ setzt REQUEST
  - Arbiter generiert GRANT (Zuteilungssignal), wenn Bus frei ist
  - GRANT wird von Master zu Master weitergereicht, bis $\mathbf{M}_i$ GRANT erhält
  - $\mathbf{M}_i$ setzt BUSY solange Bus benötigt wird



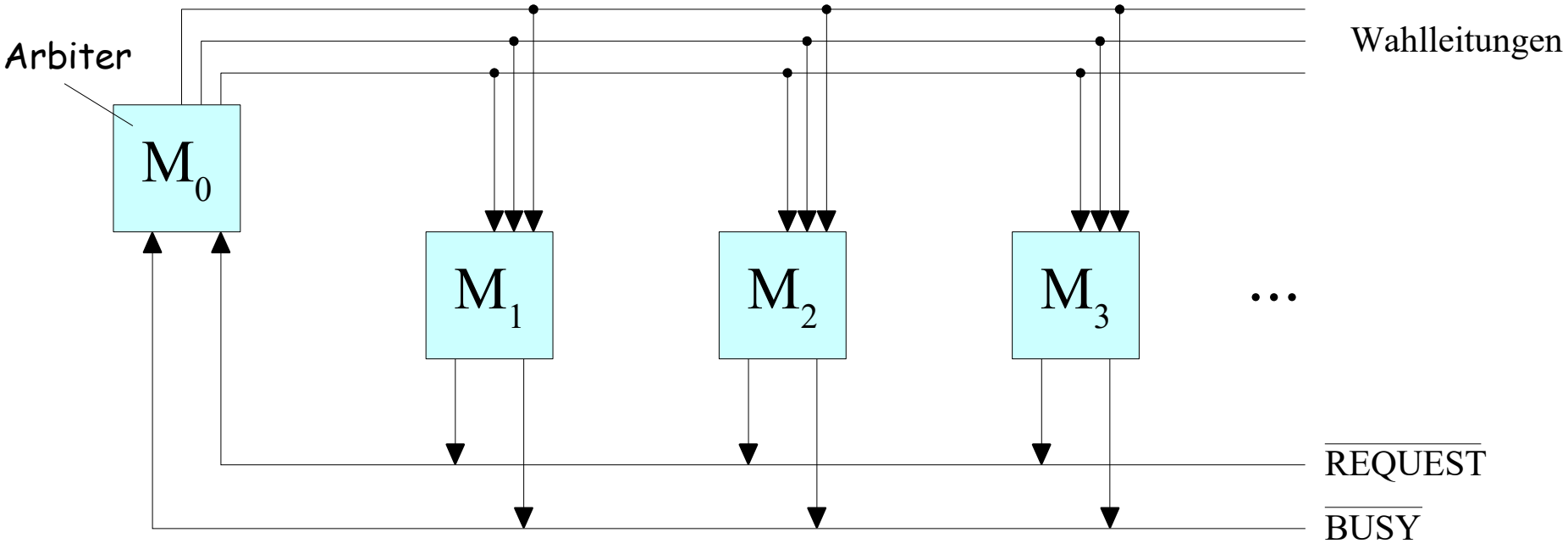
Dezentrale Arbitrierung mit Daisy-Chain

- Ablauf:
  - GRANT wird von Master zu Master weitergereicht
  - Master $\mathbf{M}_i$ behält GRANT solange Bus benötigt wird (Token-Ring)



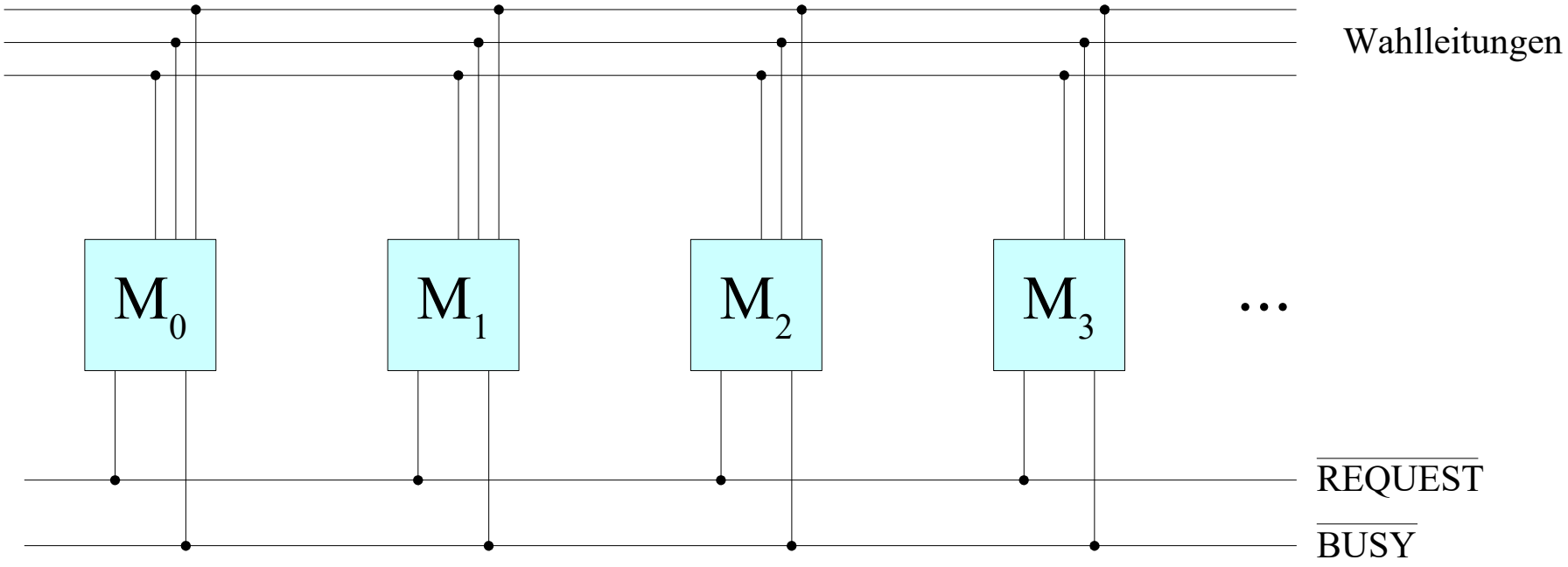
Zentrale Arbitrierung mit Polling

- Ablauf:
  - Master $\mathbf{M}_i$ setzt REQUEST
  - Arbiter sendet nacheinander die Gerätenummern aller Master
  - Wenn Gerätenummer von $\mathbf{M}_i$ anliegt, setzt $\mathbf{M}_i$ BUSY solange Bus benötigt wird



Dezentrale Arbitrierung mit Polling

- Ablauf:
  - Master $\mathbf{M}_i$ setzt REQUEST
  - Master, der zuletzt aktiv war, sendet nacheinander die Gerätenummern aller anderen Master
  - Wenn Gerätenummer von $\mathbf{M}_i$ anliegt, setzt $\mathbf{M}_i$ BUSY solange Bus benötigt wird



Asynchroner Bus-Zugriff

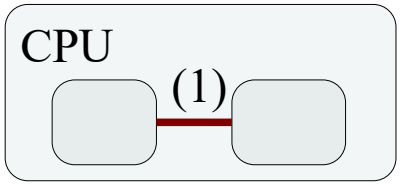
- Carrier Sense Multiple Access / Collision Detection (CSMA/CD)
- Prinzip:
  - Vor dem Senden wird geprüft, ob der Kanal für eine definierte Zeit frei ist. (Carrier Sense)
  - Falls dies der Fall ist, wird gesendet. Falls dabei eine Kollision auftritt (Collision Detection), wird Sendeversuch nach Ablauf einer zufälligen Zeitspanne wiederholt (Multiple Access)
- Varianten:
  - Collision Resolution (CSMA/CR)
    - Sendeversuch wird wiederholt, wenn Kollision mit einer höher-prioren Nachricht aufgetreten ist
  - Collision Avoidance (CSMA/CA)
    - In Funknetzen kann Kollision nicht sicher erkannt werden
    - Sendeversuch wird wiederholt, wenn Bestätigungssignal ausbleibt
- Beispiele:
  - CAN
  - WLAN

Kommunikation Rechnersystemen

- In (vernetzten) Rechnersystemen findet Kommunikation auf unterschiedlichsten Ebenen statt
  - Innerhalb der CPU (µm)
    - Implementiert in Halbleiterstrukturen
    - höchster Datendurchsatz
    - keine Anforderungen an Standardisierung
  - Auf Board-Ebene (Motherboard) (mm)
    - Implementiert in gedruckten Schaltungen (PCB)
  - Gehäuse (m)
    - Feste Verkabelung, Stecksysteme
    - Standardisierung erforderlich
  - Netzwerkebene (km)
    - Elektrische, optische (Lichtwellenleiter) oder Funk (WLAN,Richtfunk,Satellit, ...)

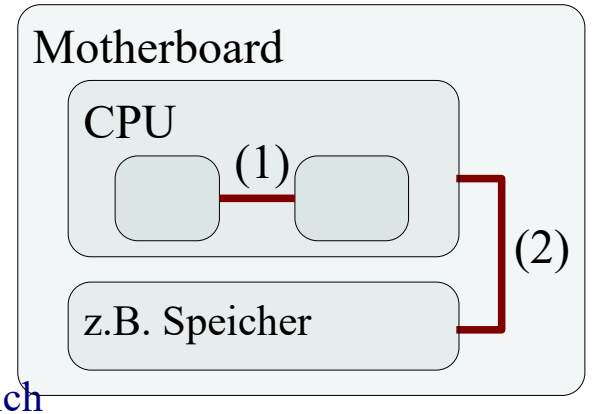
Eigenschaften von Kommunikationswegen

- Prozessor-Ebene
  - Interne, prozessorspezifische Schnittstellen zwischen CPU-Komponenten (Steuerwerk, Verarbeitungseinheit etc.)
  - Überwiegend als Parallelbus realisiert



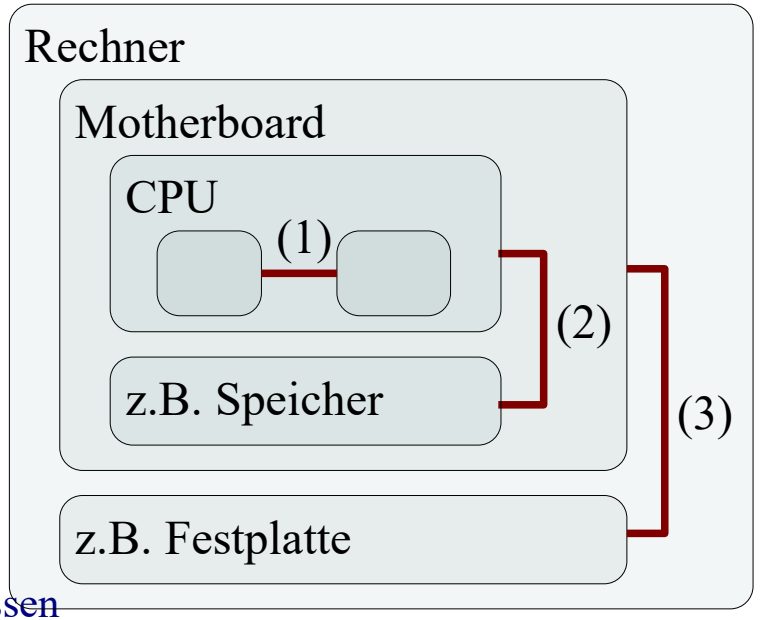
Eigenschaften von Kommunikationswegen

- Motherboard (CPU-Platine)
  - Ggf. Prozessorspezifischer Chipset, auf unmittelbare Prozessorumgebung begrenzt
  - Einheitliches Hardware-Übertragungsprotokoll
    - Parallel oder seriell
  - Speicher-Bus
    - DDRx SDRAM (JEDEC-Standard)
  - Weitere Peripherie-Komponenten
    - Proprietäre oder standardisierte Protokolle möglich
    - Media Interface, AGP Bus, FSB, SMBus, I2C, SPI, ...



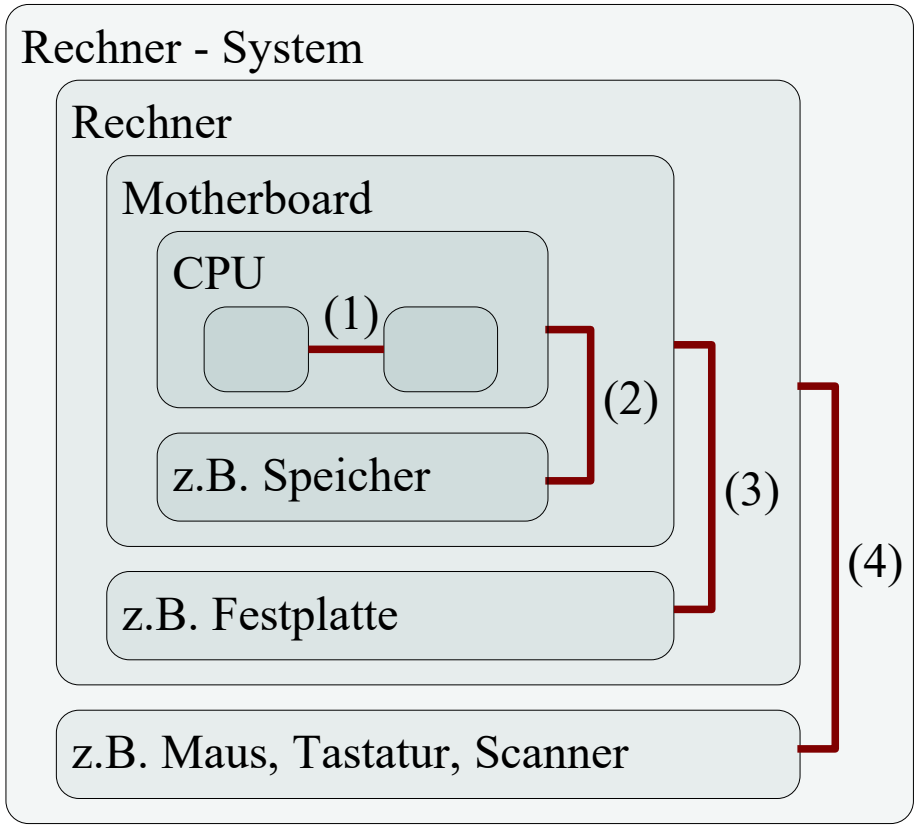
Eigenschaften von Kommunikationswegen

- Rechner-Ebene (Gehäuse)
  - Systembus, verschiedene Standards gebräuchlich, z.B.
    - VME (Versa Module Europe)
    - PCI (Peripheral Component Interconnect)
    - PCI-Express
    - SATA
  - Komplexere Bus-Arbitrierung und Hardware-Protokolle, z.B. auch
    - Gemeinsame Signalwege für Daten und Adressen
    - Blocktransfer: Adressen werden nur für das erste Wort erzeugt
  - Zunehmend serielle Kommunikationsprotokolle



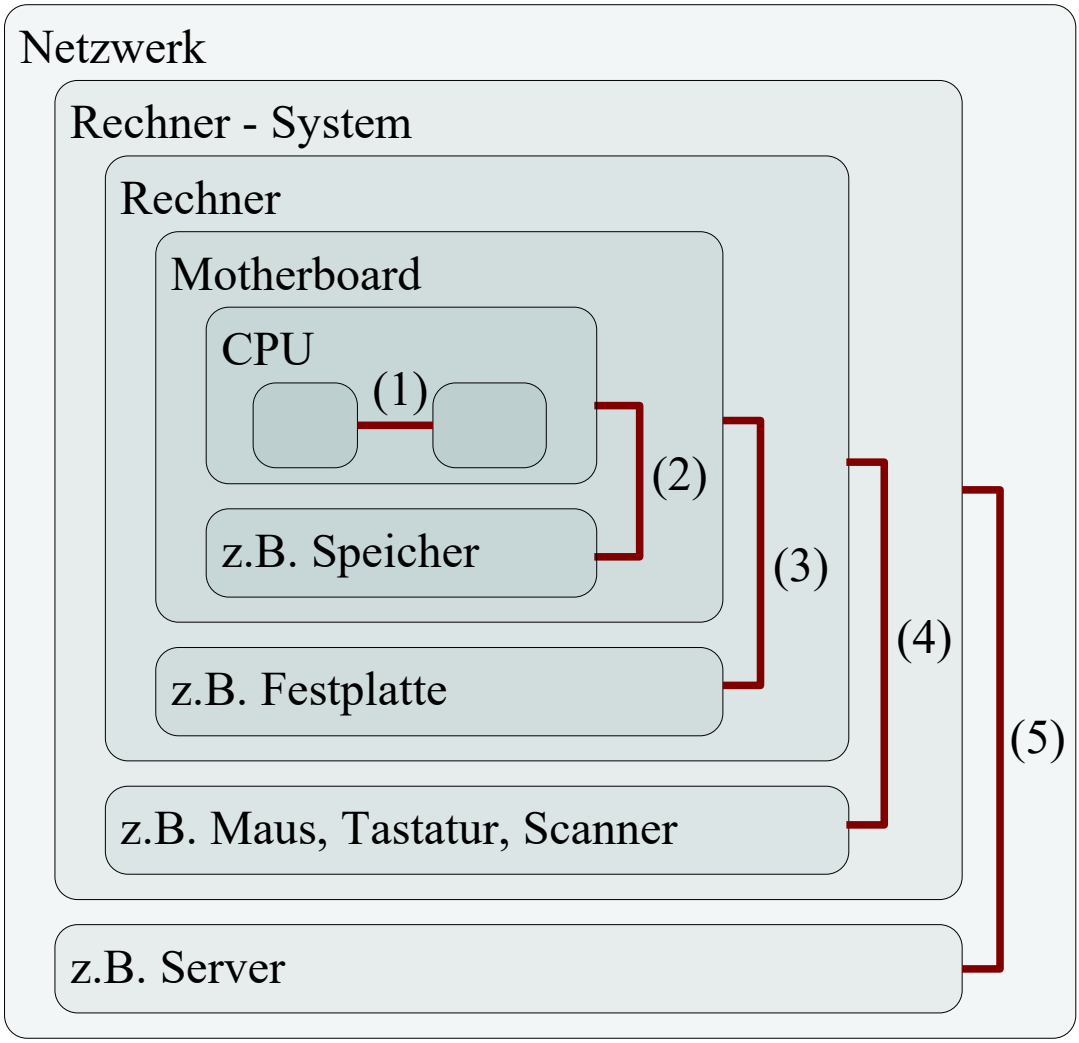
Eigenschaften von Kommunikationswegen

- Rechner-System
  - Peripherie-Bus, typische Standards:
    - SCSI (Small Computer System Interface)
    - Fibre Channel
    - USB (Universal Serial Bus)
    - Bluetooth
    - IEEE-488
  - Elektrische, optische und kabellose Übertragung gebräuchlich
  - Physikalische Ausdehnung begrenzt, aber deutlich über Gerätegehäuse hinaus
  - Serielle Datenübertragung



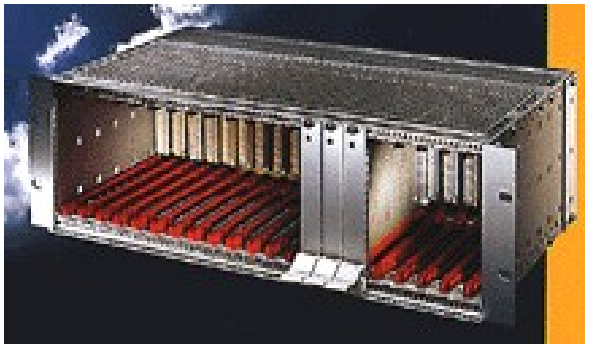
Übersicht

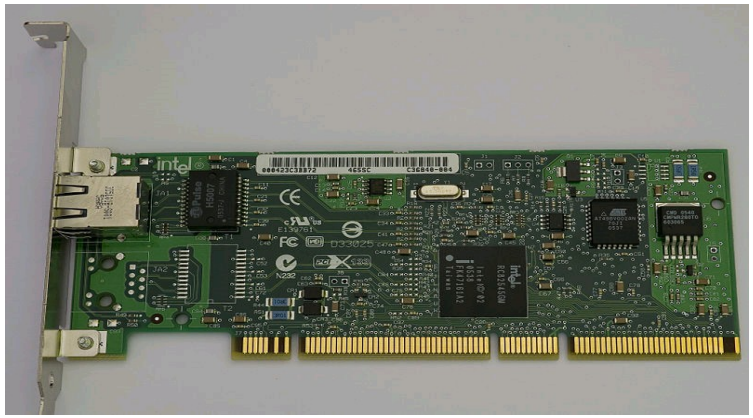
- Netzwerk-Ebene
  - Intersystem-Kommunikation
    - Ethernet, WLAN...
    - Telekommunikation
    - Industrie-Bus / Feld-Bus
    - Fahrzeugbusse (CAN, ...)
  - Zahlreiche Standards mit unterschiedlichen Anforderungsprofilen:
    - Hohe Datenrate
    - Echtzeitfähigkeit
    - Zuverlässigkeit
    - Sicherheit ...



PCI-Bus

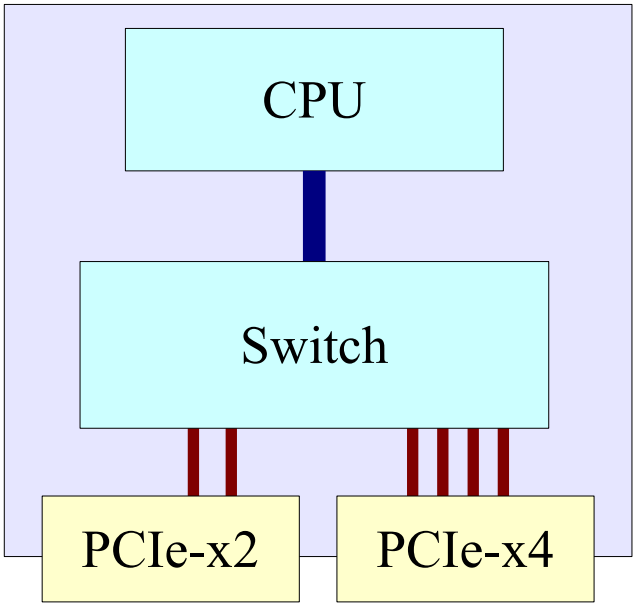
- Peripheral Component Interconnect Bus
- Heute durch PCI-Express weitgehend verdrängt
- Anwendungen:
  - PC, Workstation, Server, Industriesysteme
- Standardisierung:
  - PCI Special Interest Group,
  - PICMG (PCI Industrial Computer Manufactorers Group)
- Viele Anbieter für Hard- und Softwarekomponenten

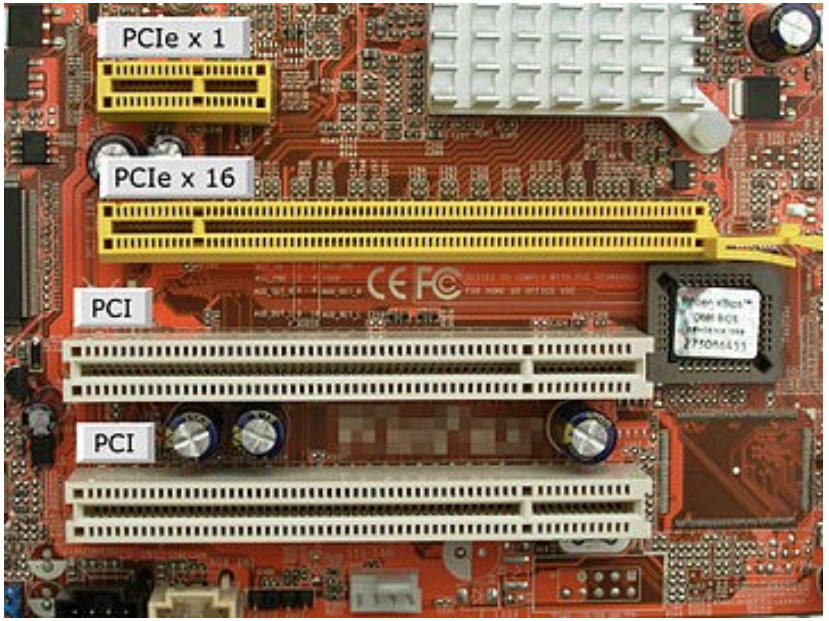




PCI Express

- PCI Express (PCIe)
  - Serielle Punkt-zu-Punkt-Verbindung zwischen einem Knoten (Switch, befindet sich auf dem Motherboard) und den Komponenten (Einsteckkarten)
  - Jede Einsteckkarte kann eine oder mehrere Lanes (serielle, bidirektionale Datenleitung) verwenden (PCIe x1,..., PCIe x16)
  - Datenübertragungsrate bis zu 1969 MB/s je Lane
  - Hot-plugging ist möglich

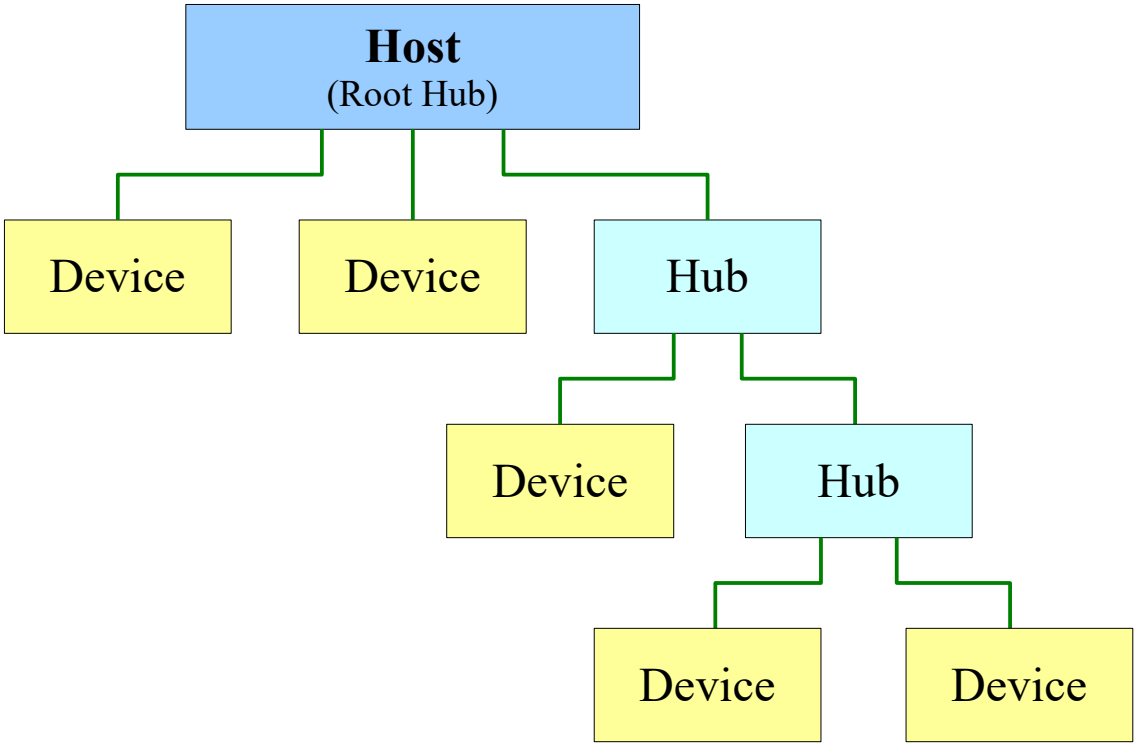




USB

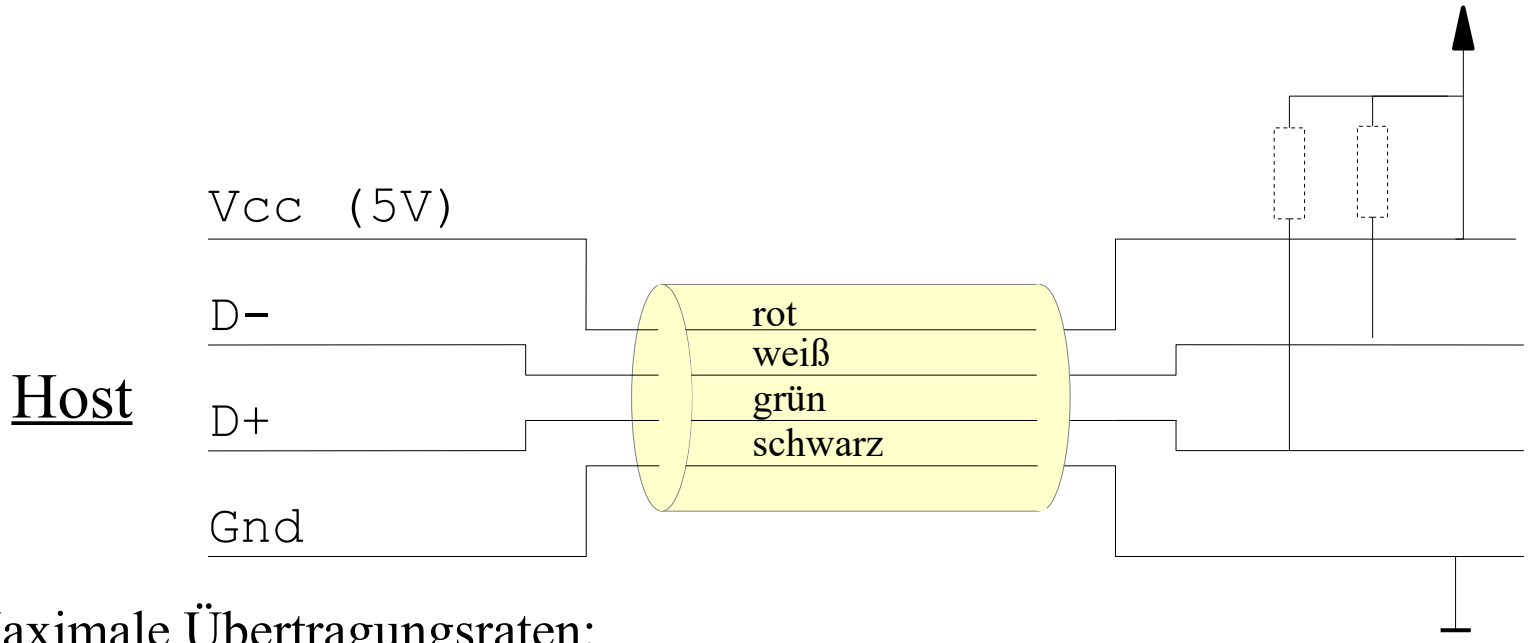
- USB = **U**niversal **S**erial **B**us
  - Hardware Interface für low-speed Peripherals
  - Keyboard, Maus, Joystick, Scanner, Printer, Modem, ...
  - Unterstützt MPEG-1 and MPEG-2 Digital Video
  - "Core-Team": Compaq, Hewlett Packard, Intel, Lucent, Microsoft, NEC, Philips
  - USB definiert Ende 1995, praktisch in PCs seit 1997, ab 2000: USB 2.0, ab 2008: USB 3.0
  - Unterstützung seit Windows 98
  - Konkurrenz zu FireWire (IEEE 1394)?

- Topologie:
  - Bis zu 127 Geräte in einem System
  - Leitungslänge max. 5 m zwischen Geräten
  - Kommunikation ausschließlich zwischen **Host** und **Device**



USB

- Differentielle Signalübertragung



- Maximale Übertragungsraten:
  - Low Speed: 1,5 Mbit/s (USB 1.0)
  - Full Speed: 12 Mbit/s (USB 2.0)
  - High Speed: 480 Mbit/s (USB 2.0)
  - Super Speed: 5 Gbit/s (USB 3.0)

- Integrierte Stromversorgung für periphere Geräte

USB

- Eigenschaften der Signalübertragung (USB 2.0)
  - Kabel mit 4 Leitungen (VCC, D-,D+,Gnd)
  - Schnelle Geräte können volle Bandbreite nutzen
  - Langsame Geräte nutzen Subkanal mit 1,5 Mbit/s
  - System unterstützt "Hot Swap": Anschließen neuer Einheiten ohne Abschalten des Systems
- Transferarten
  - Control: Identifikation + Konfiguration
  - Interrupt: Reservierte Bandbreite, aber keine Übertragungsrate, Fehlerkorrektur
  - Bulk: Keine reservierte Bandbreite, Fehlerkorrektur
  - Isochron: Reservierte Bandbreite + garantierte Übertragungsrate / Latenzzeit, keine Fehlerkorrektur

- Geräteerkennung und Identifikation
  - Hot-Plugging möglich
  - Enumeration nach Anschließen des USB-Gerätes,
    der Host erfragt dabei u.a. die „Device Configuration“
    - Vendor-ID (VID)
    - Product-ID (PID)
    - Device-Class
    - usw.
  - Betriebssystem stellt passenden Treiber zur Verfügung (erfordert ggf. Treiber des Herstellers)
  - Geräteklassen
    - Human Interface Device (HID)
    - Mass Storage Device (MSD)
    - Audio
    - Communication (CDC)
    - usw.

- Anwendungskommunikation
  - Master-Servant-Prinzip: Host initiiert Kommunikation
  - Austausch von Datenpaketen, aufgeteilt in Frames und Microframes
  - Endpunkte als logische Datensenke / Datenquelle
    - Eindeutige Adressierung
    - Endpunkte und Host bilden eine „Pipe“
  - Datentransfer
    - Token Packet (kennzeichnet Inhalt)
    - Data Packet
    - Status Packet (Bestätigung, Fehlerkorrektur)