technische universität
dortmund

fakultät für informatik
informatik 12

2.5 Kommunikation, Ein-/Ausgabe (E/A) - engl. *input/output (I/O)* -

Peter Marwedel

Informatik 12

Otto-Hahn-Str. 16

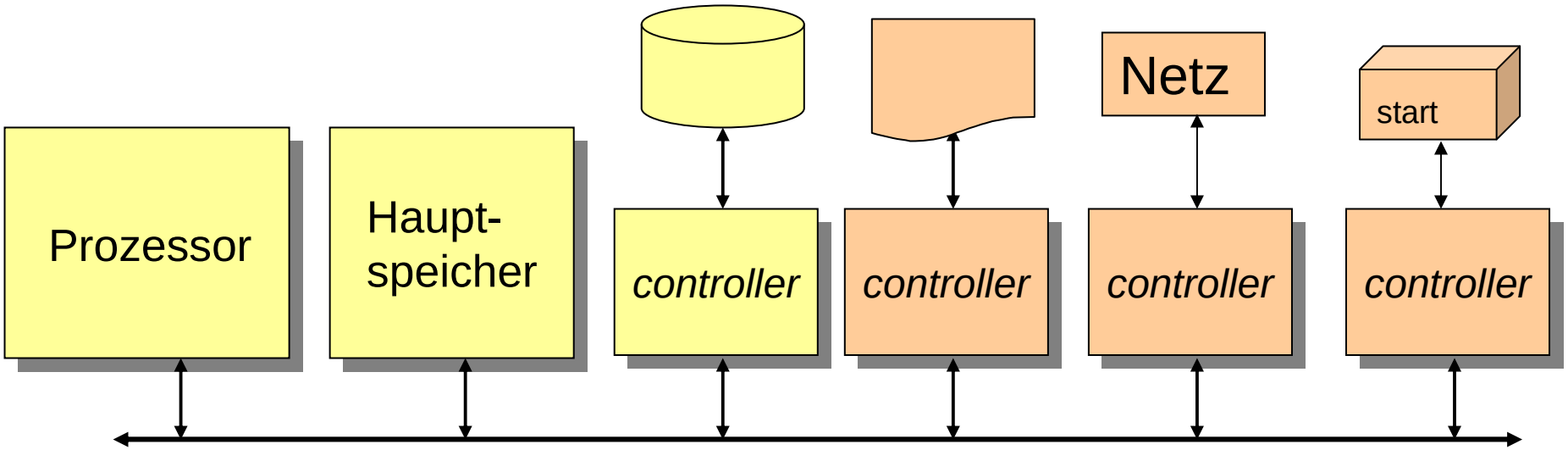
Tel. 755 6111

E-mail: peter.marwedel@tu-dortmund.de

Sprechstunde: Mo 13:00-14:00

WAV-Sound

Kontext


Prozessor
Haupt-
speicher
controller
controller
Netz
controller
start
controller

Wie sieht das interne Verbindungsnetzwerk aus?
☞ Bussysteme, Bustopologien

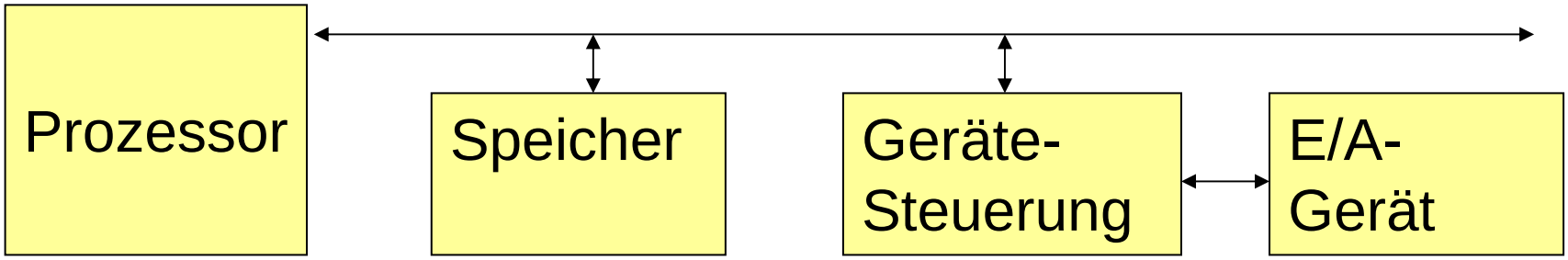
- Wie sieht das Verbindungsnetzwerk aus? 
- Wie adressiert man die E/A-Geräte und deren Steuerungen?
- Busse mit oder ohne Bestätigung?
- Wie stimmt man die Geschwindigkeit der E/A-Geräte mit der des Prozessors ab?

Viele Entwurfsbeschränkungen:

- Geschwindigkeit
- Kosten
- Kompatibilität
- ...

☞ viele Bus-Topologien.

1. Ein einziger Bus



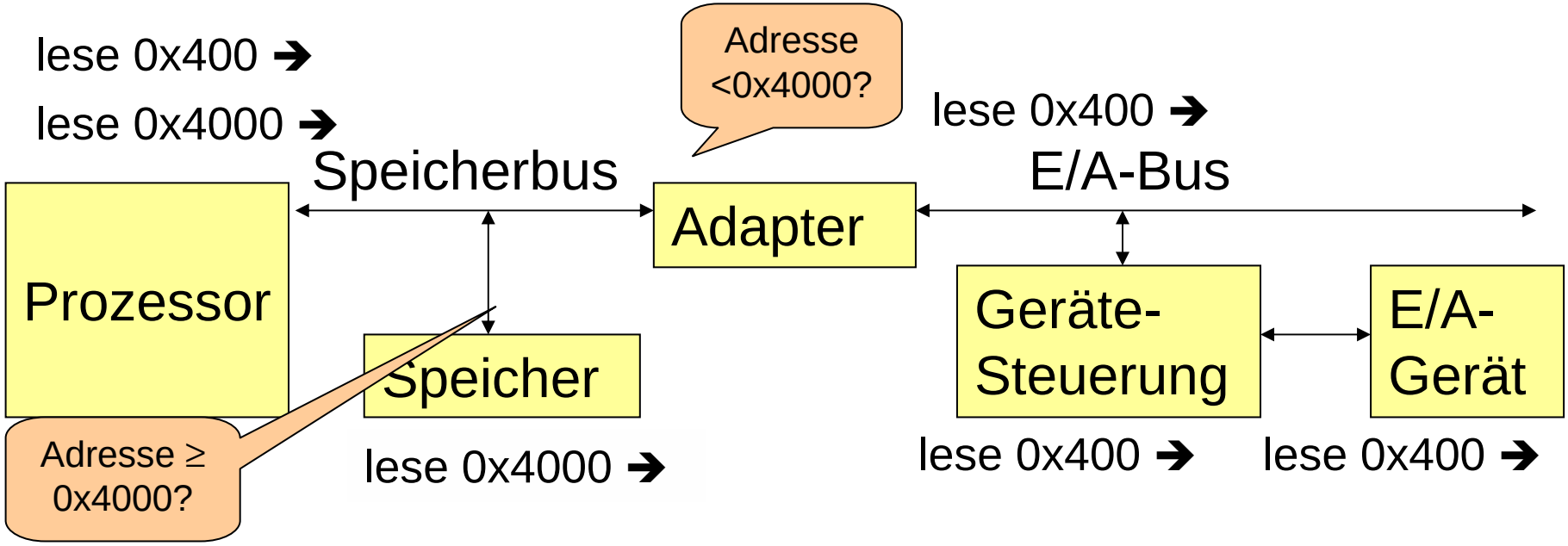
Sehr einfach;

Motiviert u.a. durch Pinbeschränkungen;

Enge Kopplung von Speicher- und E/A-Spezifikationen;

Eingeschränkte Parallelarbeit.

2. Einstufiger Adapter zum Übergang auf E/A-Bus

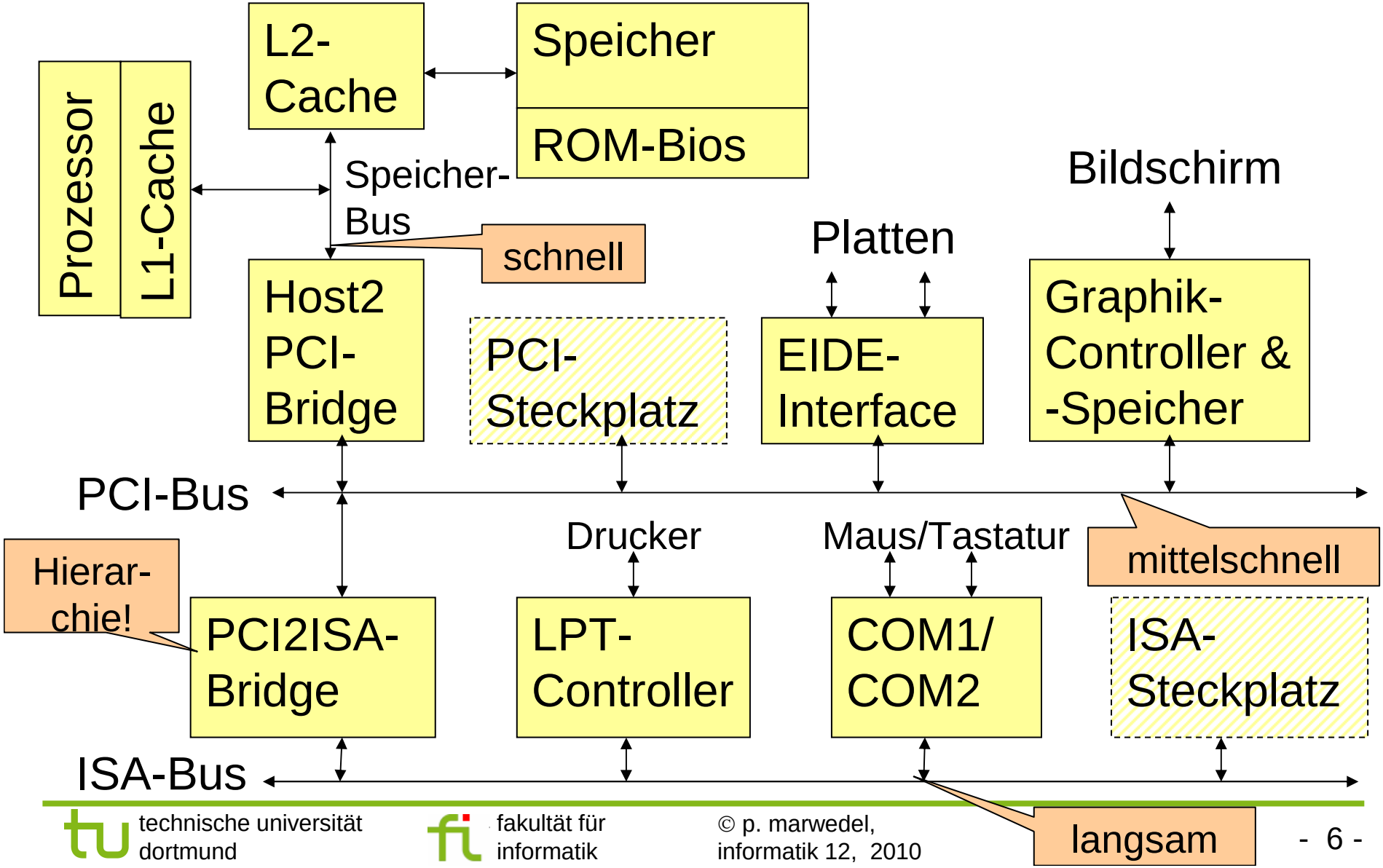


Spezielle Speicheradressen oder Kontrollleitung *IO/Memory.*

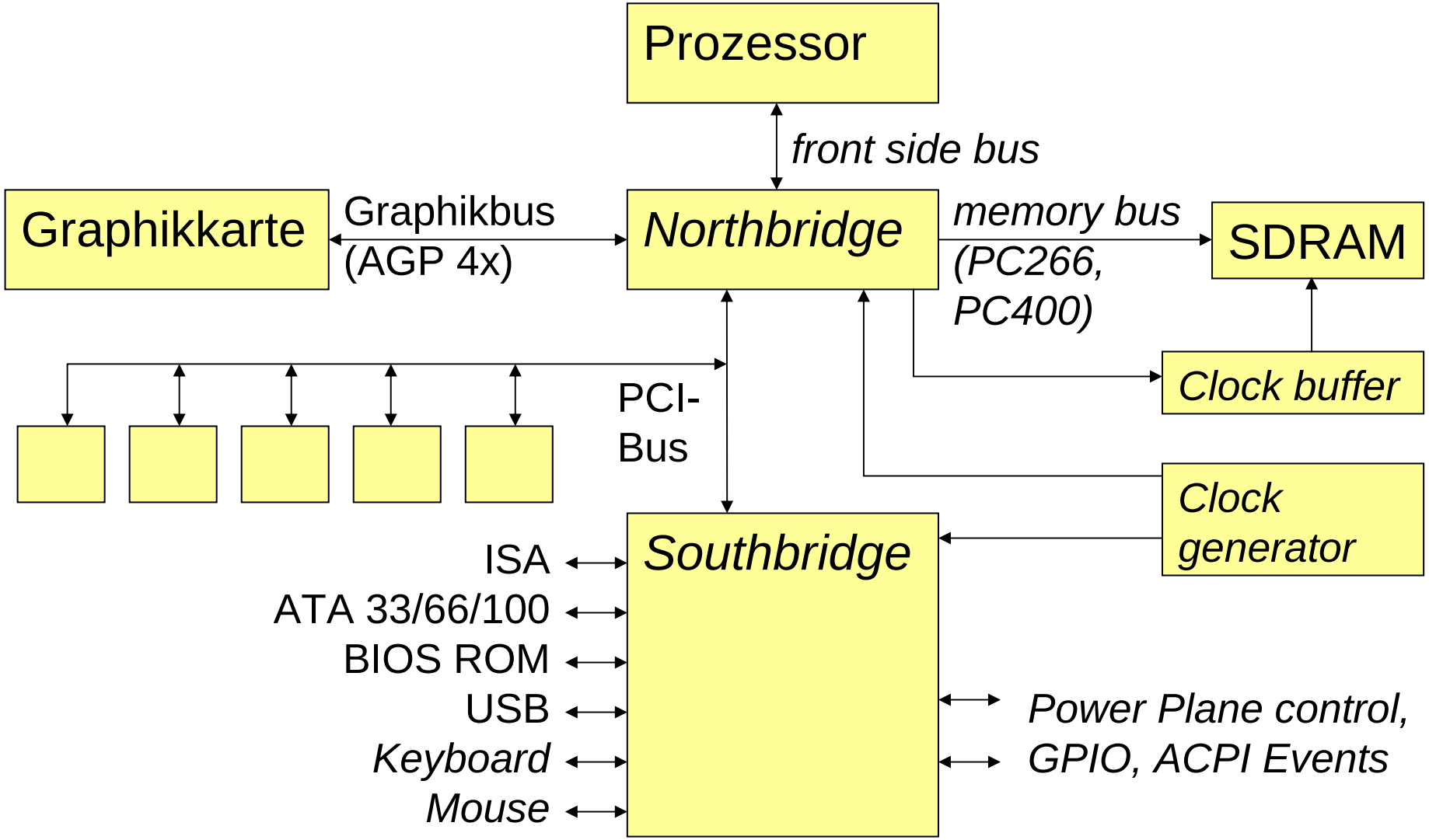
Speicherbus ggf. mit größerer Wortbreite;

höhere Geschwindigkeit.

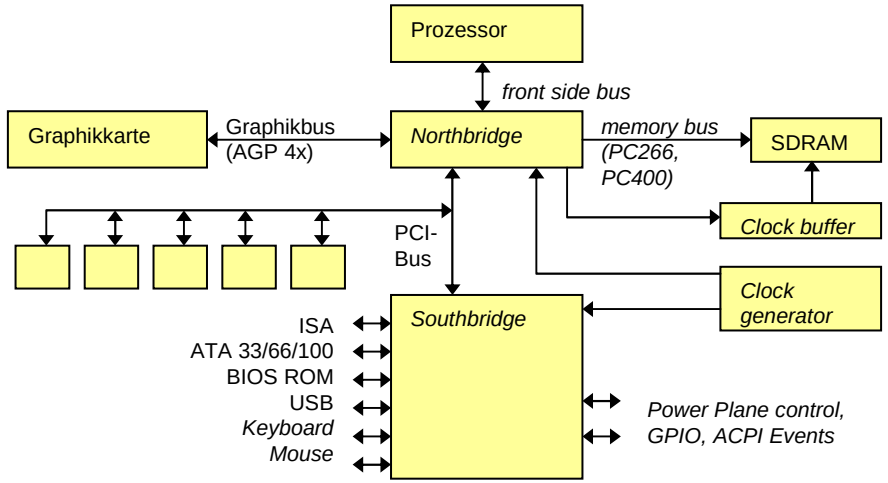
3. Grundstruktur der ersten PCI-basierten Systeme

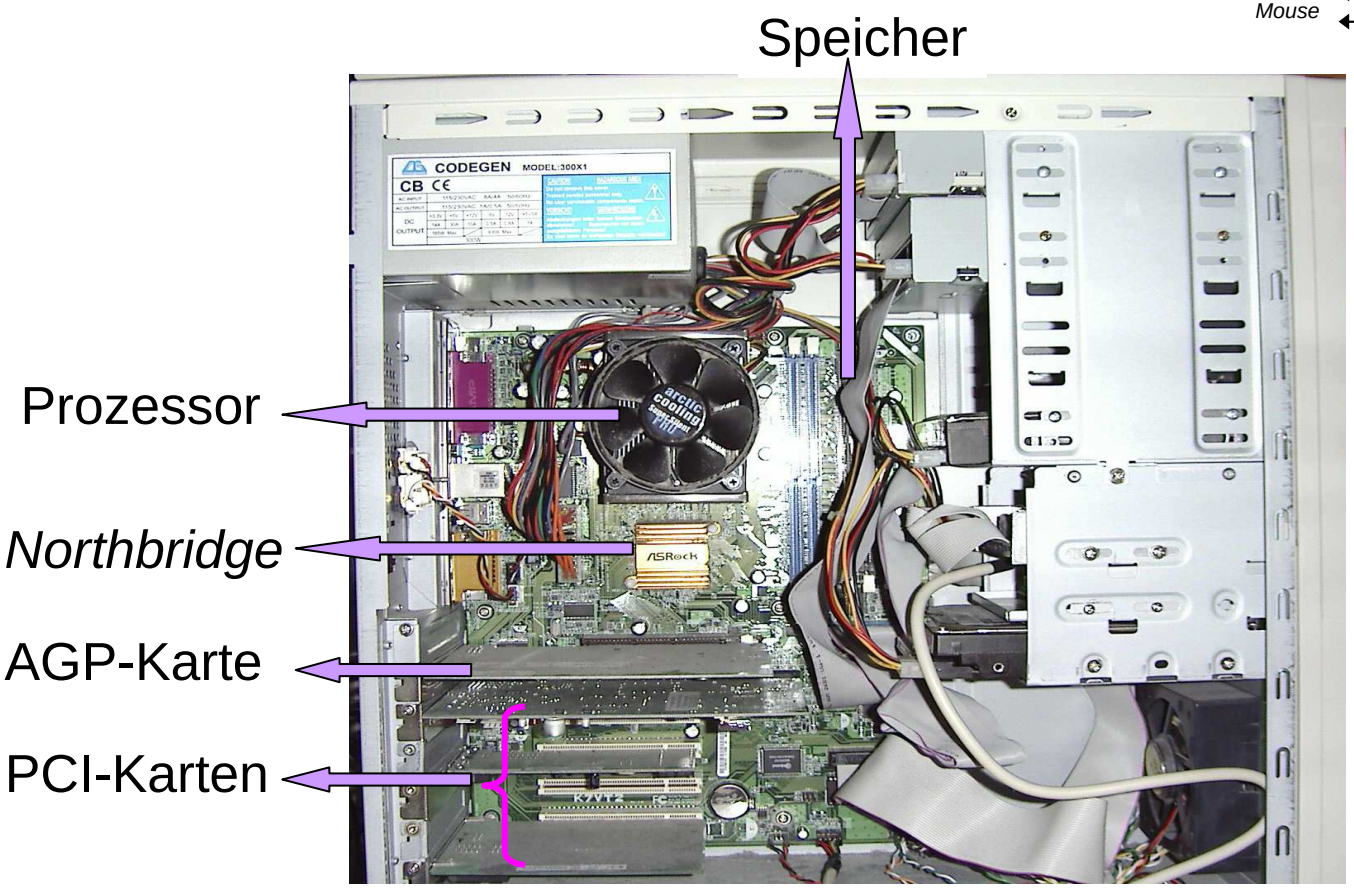


4. Grundstruktur späterer PCI-basierter Systeme

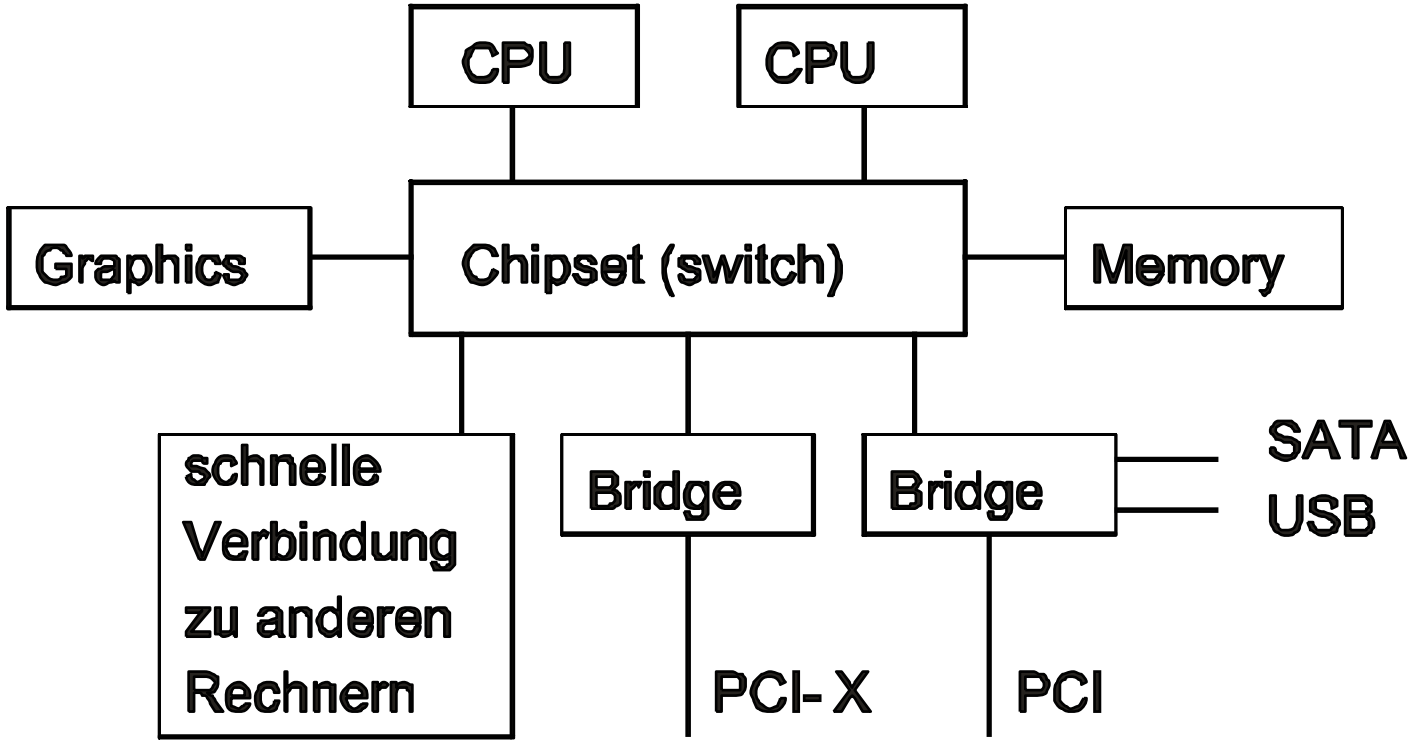


Wo sind diese Komponenten auf dem Board?





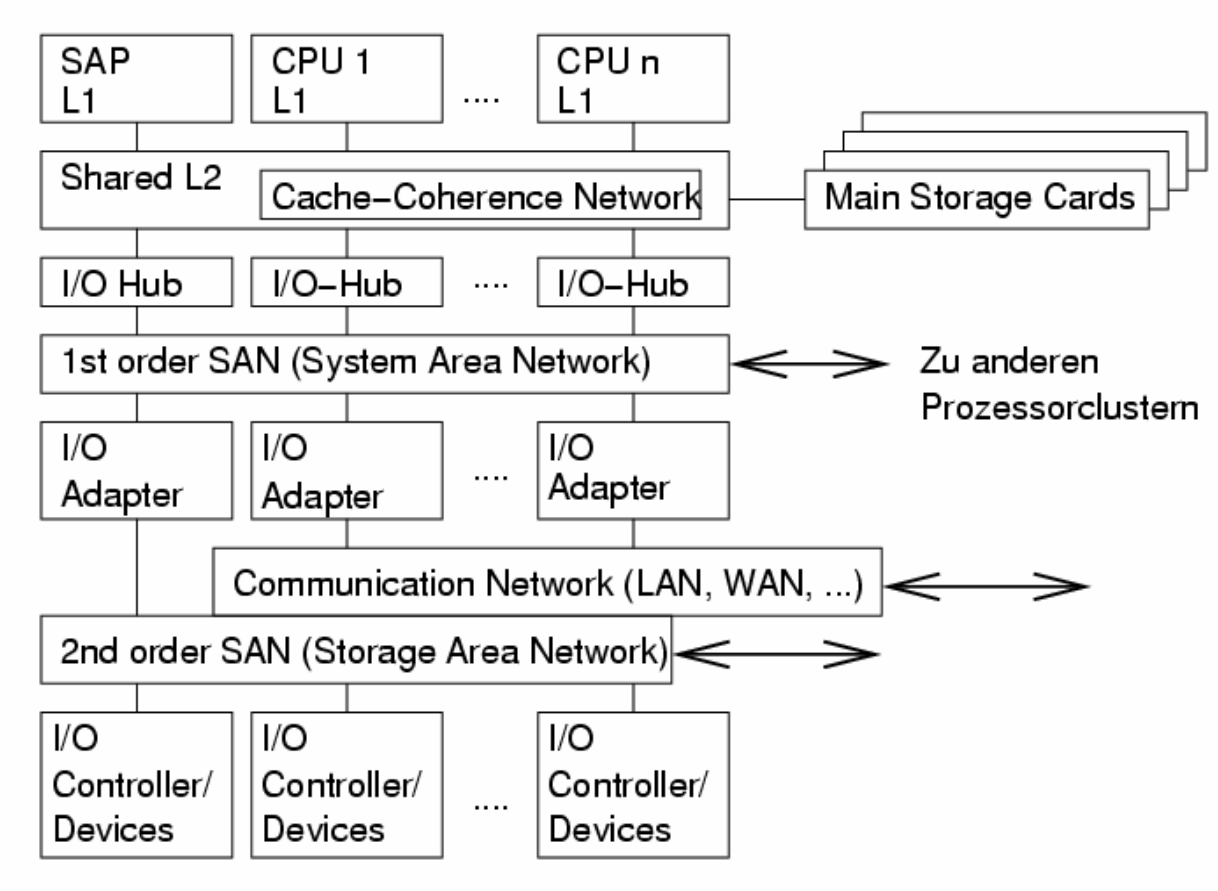
5. PCI-Express



Möglichkeit der Vermeidung eines Busses, an den mehrere Karten angeschlossen sind und die gegenseitig die Geschwindigkeit begrenzen.

6. Vollständig separate Speicher- und E/A-Busse (Großrechner)

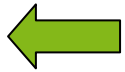
Der Speicher ist Multiport-Speicher, und es ex. Kanäle, welche ohne CPU-Belastung Daten zwischen Speicher und E/A-Steuerungen transportieren & eigene Programme ausführen.



zSerie (IBM)

Kanäle als *system assist processors* (SAPs) realisiert.
Gemeinsam genutzter L2-Cache verbindet ≤ 20 Prozessoren (16 CPUs, 3 SAPs, 1 Reserveprozessor) + 4 I/O-Hubs.
Prozessoren, L2-Cache, I/O-Hubs befinden sich auf MCM.
Jede CPU, und jeder SAP hat Bandbreite von 14,5 GByte/s mit einer Zugfriffszeit von 20 ns zum L2-Cache.
Bandbreite des Hauptspeichers beträgt 29 GByte/s.
Bandbreite für alle I/O-Hubs insges. theoretisch 36 GByte/s, davon 29 GByte/s praktisch realisierbar.
I/O-Prozessoren können Aufträge von verschiedenen Betriebssystemen entgegennehmen.
Komponenten auf Zuverlässigkeit von 99,999 % ausgelegt.

Adressierung

- Wie sieht das Verbindungsnetzwerk aus?
- Wie adressiert man die E/A-Geräte und deren Steuerungen? 
- Busse mit oder ohne Bestätigung?
- Wie stimmt man die Geschwindigkeit der E/A-Geräte mit der des Prozessors ab?

2 Techniken:

1. Speicherbezogene Adressierung
2. Eigene E/A-Adressen

1. Speicherbezogene Adressierung (*memory mapped I/O*)

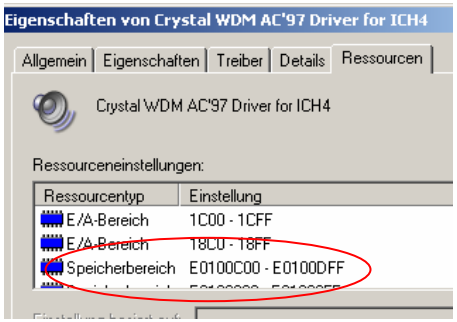
Geräte erhalten spezielle Speicheradressen,
Kommunikation mittels *load*- und *store*-Befehlen,
Prozessoren ohne separate Speicher- und E/A-Schnittstelle;
Trennung in E/A- und Speicherbus nur über Busadapter

Adressraum:

load = Eingabe;
store = Ausgabe

Muss von *Caching* ausgenommen werden.

Beispiel: MIPS-Maschine wie bei SPIM realisiert



Eigenschaften speicherbezogener E/A-Adressierung

Vorteile:

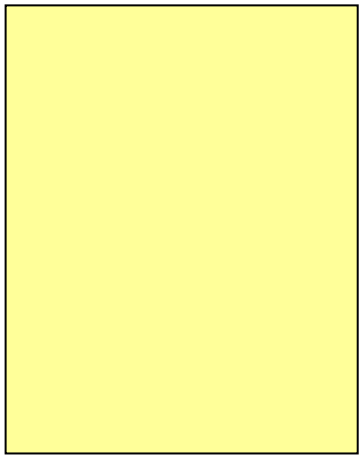
- Großer E/A-Adressraum aufwandsarm,
- Wenig Festlegung beim Prozessorentwurf,
- Keine separaten E/A-Befehle notwendig; kleiner Befehlssatz.

Nachteile:

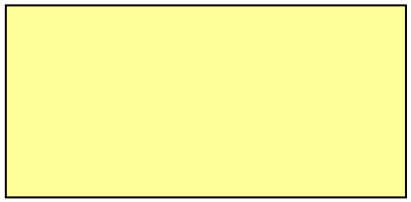
- Besonderes E/A-Timing nur über Busadapter möglich,
- Zugriffsschutz auf E/A-Geräte nur über Speicherverwaltung.

2. Separate E/A-Adressen

Speicheradressraum



E/A-Adressraum

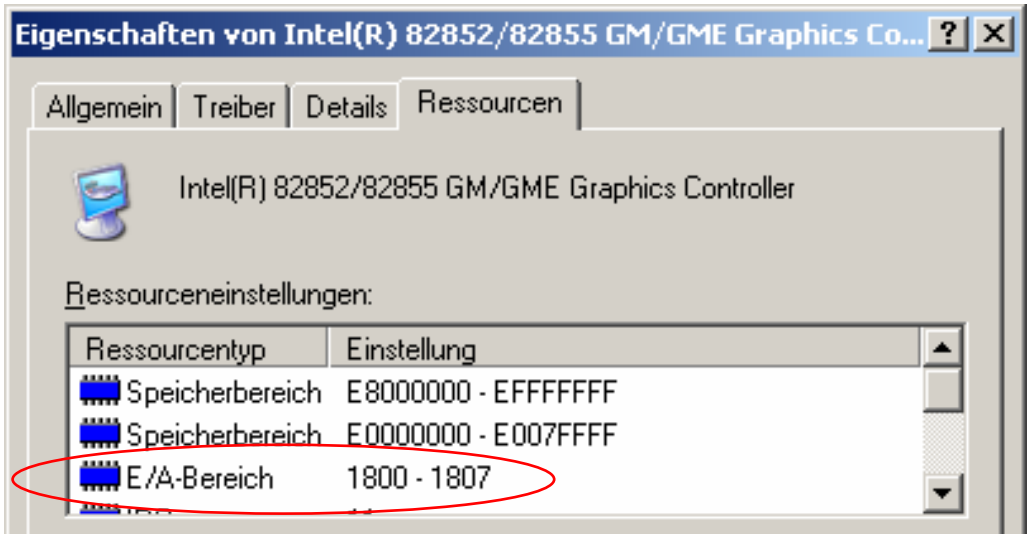


in = Eingabe;
out = Ausgabe

Umkehrung von Vor- und Nachteilen der speicherbezogenen Adressierung.

Prozessoren mit E/A-Adressen können auch speicherbezogene Adressierung nutzen.

Beispiel: Intel x86/Pentium-Familie.



2.5.1.3 Synchrone und asynchrone Busse

- Wie sieht das Verbindungsnetzwerk aus?
- Wie adressiert man die E/A-Geräte und deren Steuerungen?
- Busse mit oder ohne Bestätigung? 
- Wie stimmt man die Geschwindigkeit der E/A-Geräte mit der des Prozessors ab?

Unterscheidung zwischen:

- unidirektionalem Timing (bei synchronen Bussen) und
- bidirektionalem Timing (bei asynchronen Bussen).

Synchrone und asynchrone Busse

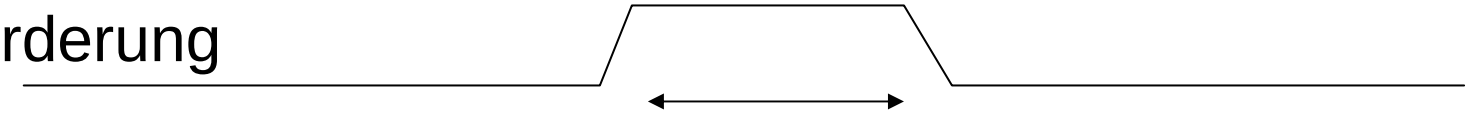
Unterscheidung basiert auf Unterscheidung zwischen unidirektionalem Timing (bei synchronen Bussen) und bidirektionalem Timing (bei asynchronen Bussen).

- **Unidirektionales Timing:**
  Kommunikationspartner verlassen sich darauf, dass Partner innerhalb festgelegter Zeit reagieren.
- **Bidirektionales Timing:**
  Kommunikationspartner bestätigen per Kontrollsignal (senden ein ***acknowledgement***), dass sie in der erwarteten Weise reagiert haben.

Schreiben beim synchronen Bus

Sender → Empfänger

Prinzip:



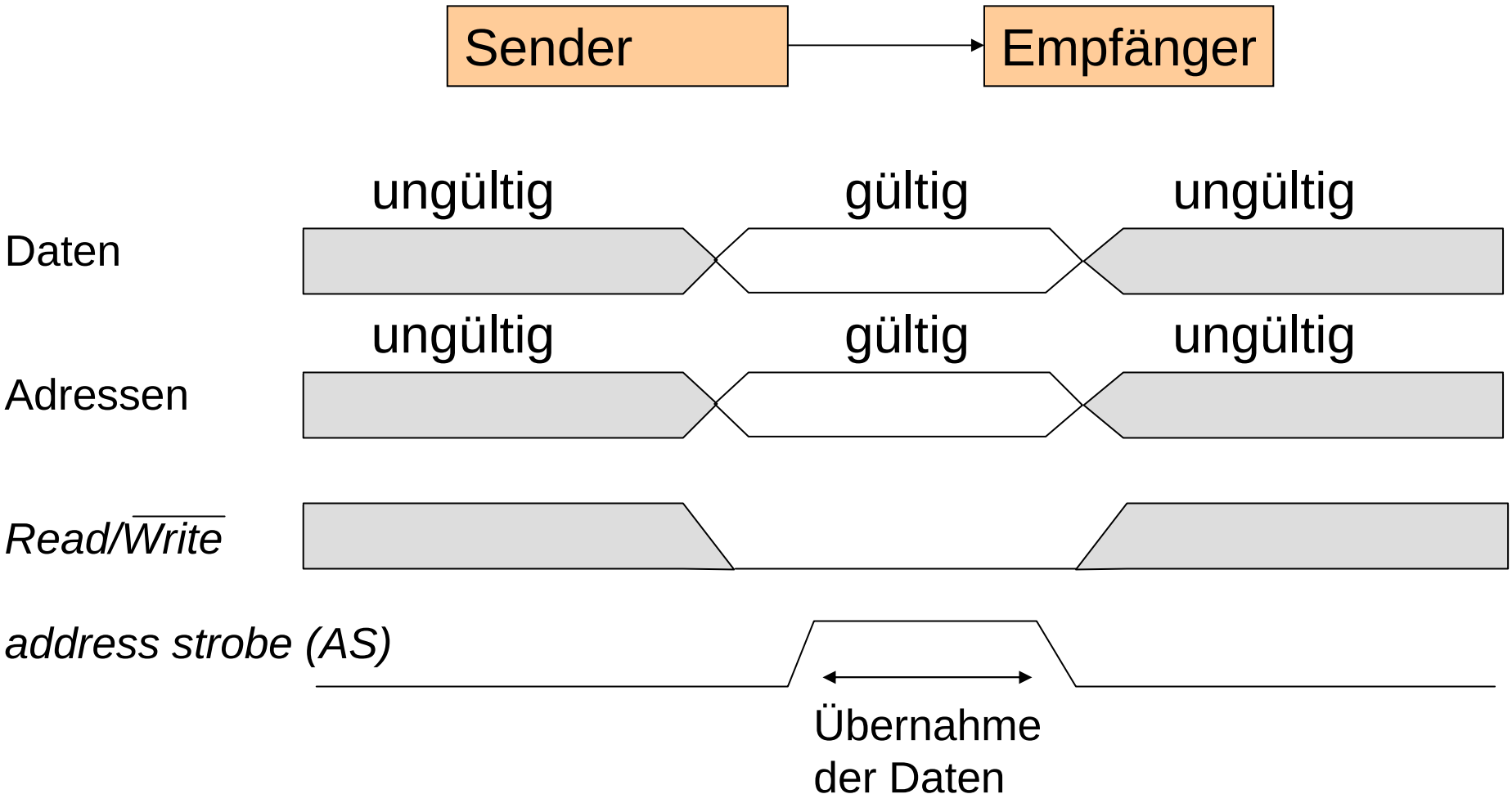
(*write request, write strobe*)

Übernahme der Daten innerhalb maximaler Zeit nach Flanke der Schreibanforderung

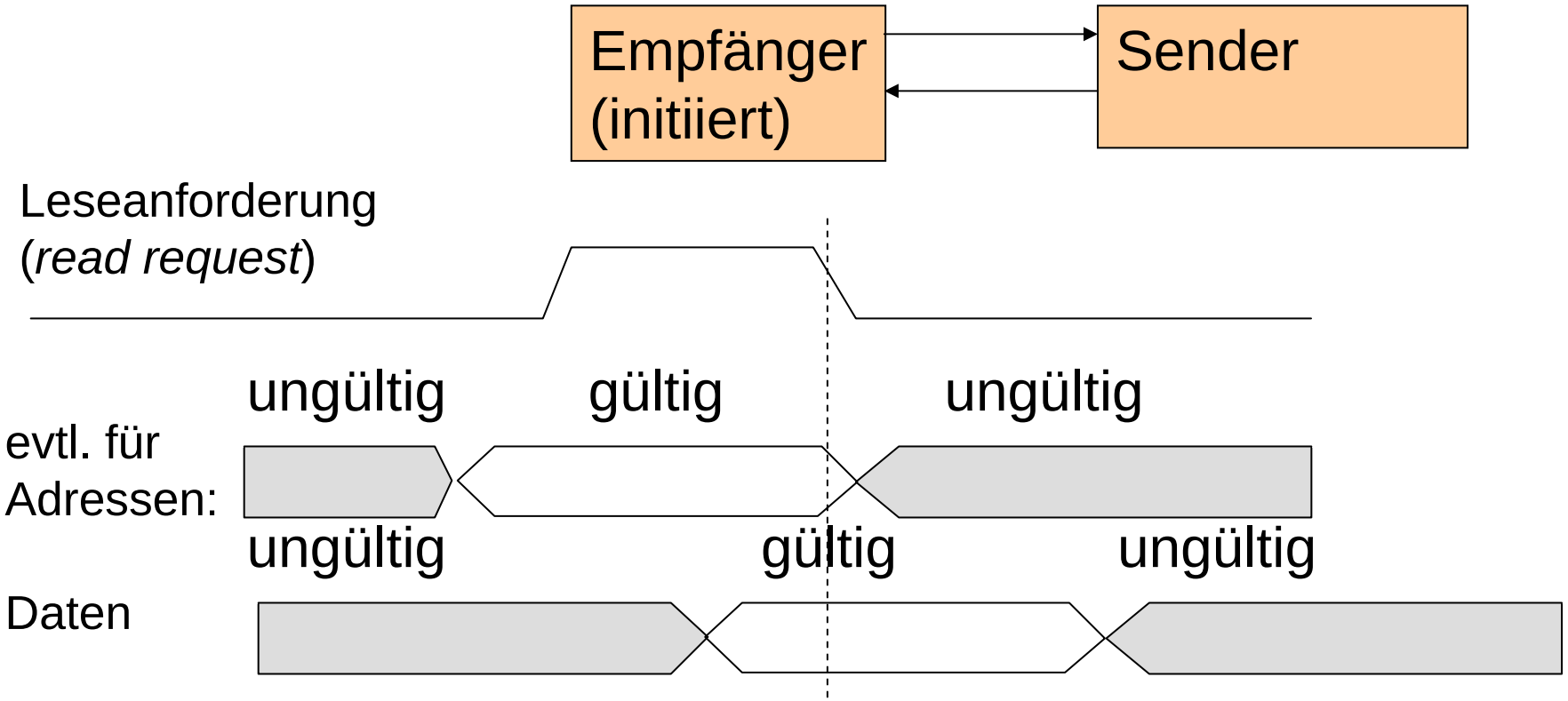
Sender verlässt sich darauf, dass der Empfänger die Daten annimmt.

Alle Signale laufen vom Sender zum Empfänger ☞ schnell.

Schreiben beim synchronen Bus
- mit Adressleitungen -

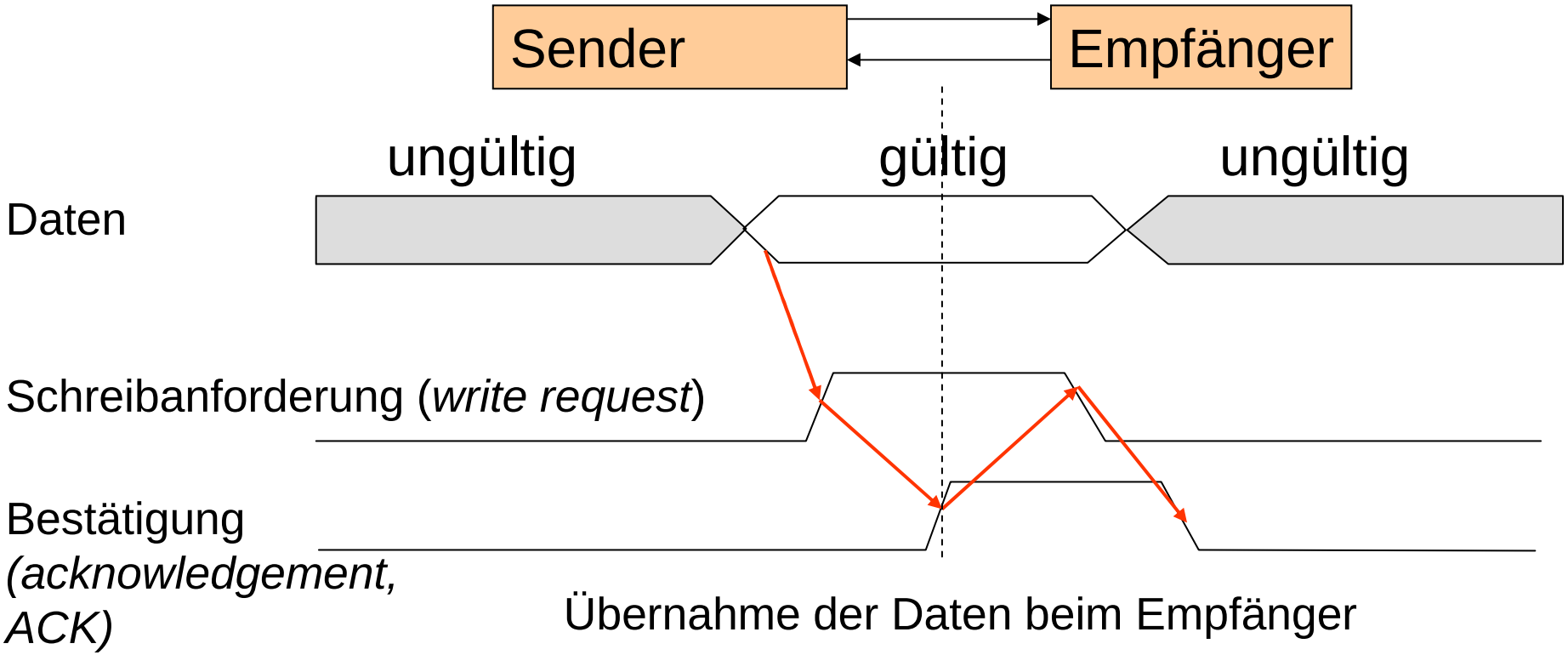


Lesen beim synchronen Bus



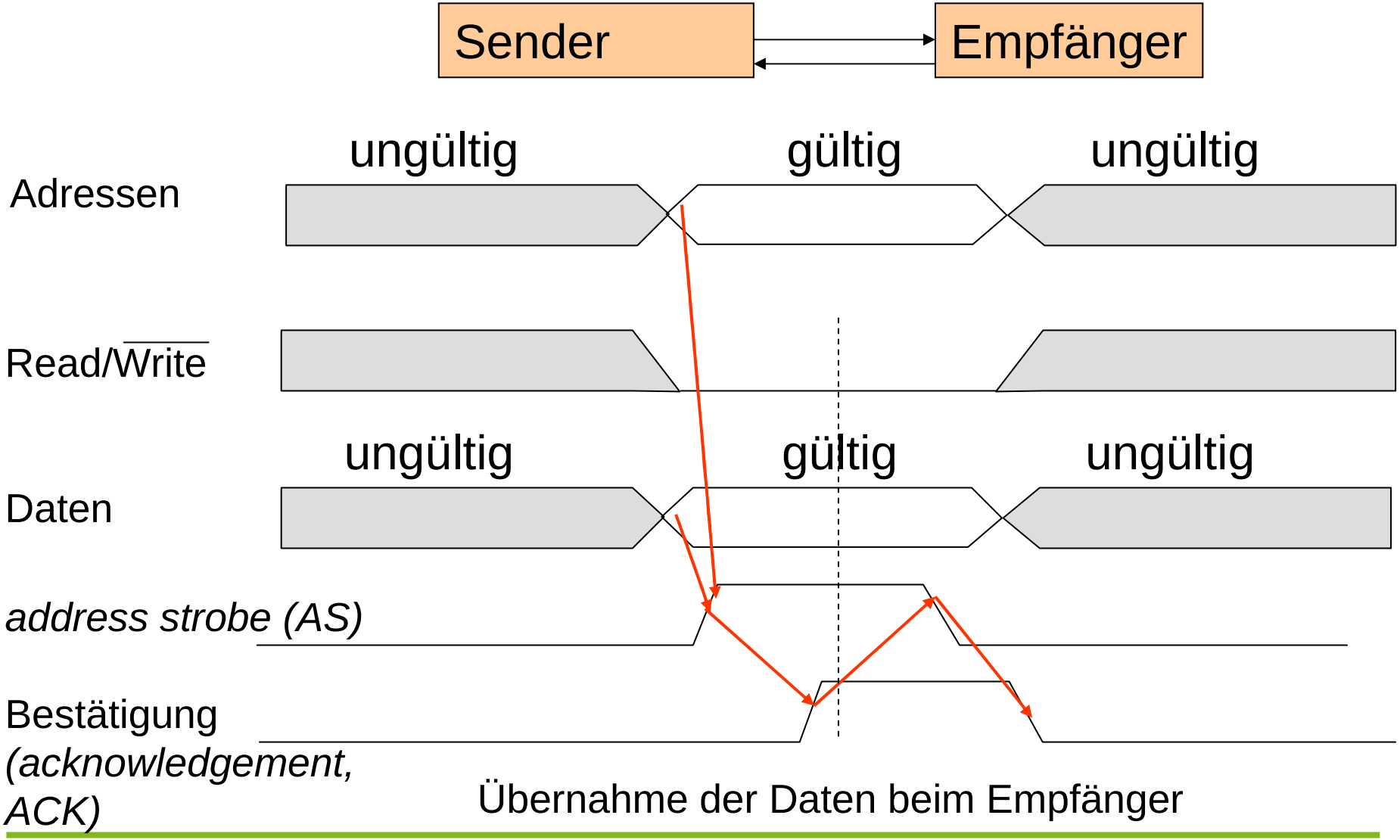
Empfänger und Sender verlassen sich auf verabredete Zeiten.
2 Signallaufzeiten.

Schreiben beim asynchronen Bus

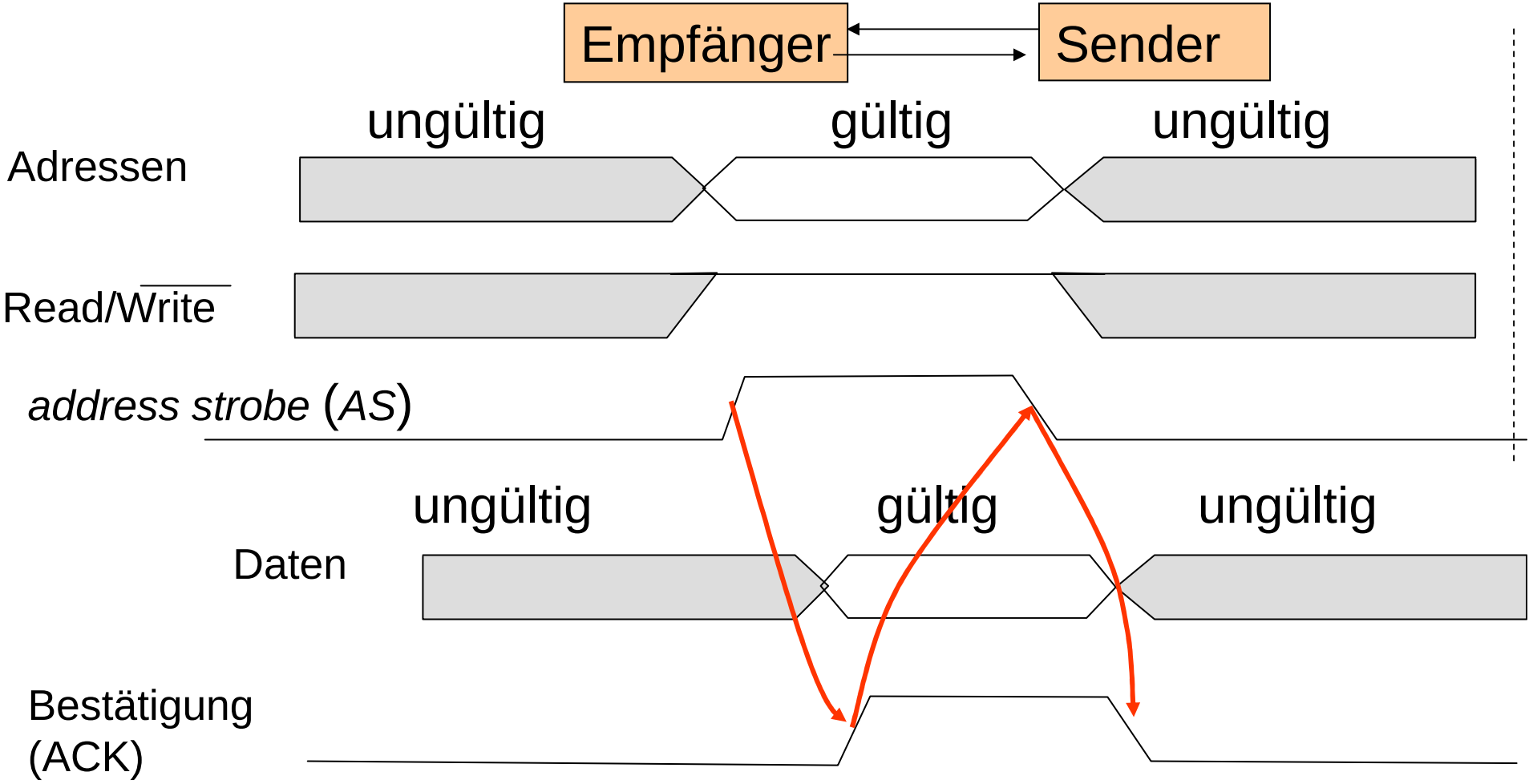


Der Sender muss Daten gültig halten, bis Bestätigung eintrifft.
Bis zu 4 Signallaufzeiten ☞ langsam.

Schreiben beim asynchronen Bus
- mit Adressleitungen -



Lesen beim asynchronen Bus
- mit Adressleitungen -

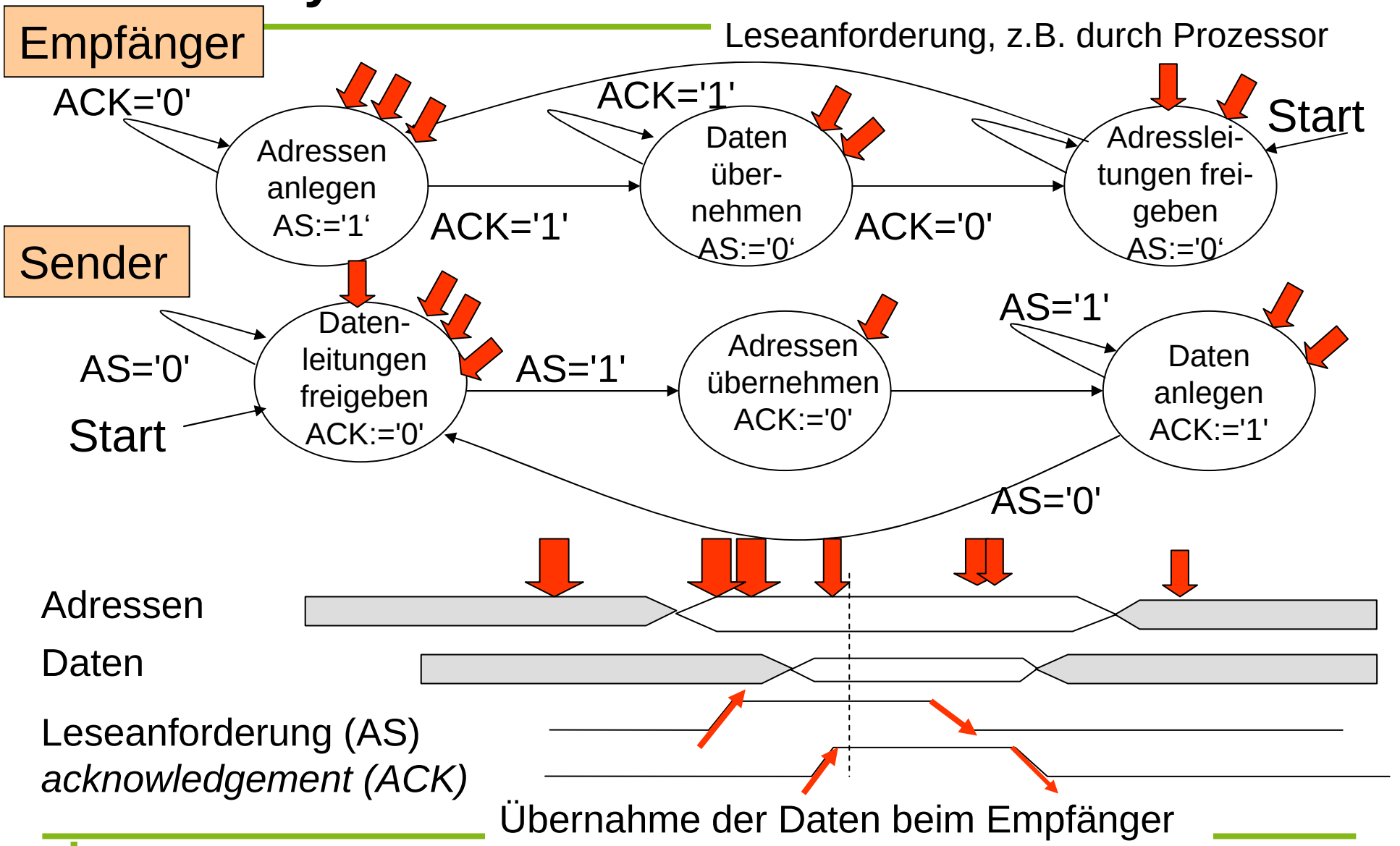


Sender muss Daten gültig halten, bis Bestätigung eintrifft. Mehrere Laufzeiten.

Vergleich der beiden Methoden

	unidirektional	**bidirektional**
Vorteile	einfach; bei konstanten Antwortzeiten schnell	passt sich unterschiedlichen Geschwindigkeiten an
Nachteile	Kommunikationspartner muss in bestimmter Zeit antworten	komplexer; Zeitüberwachung notwendig; evtl. langsam
	synchrone Busse, Speicherbusse	asynchrone Busse, E/A- und Peripheriebusse

Beispielhaftes Zustandsdiagramm eines asynchronen Busses für das Lesen



Sender und Empfänger als kommunizierende Automaten

- Sender und Empfänger stellen jeweils einen Automaten dar, der die Bussignale erzeugt.
- Diese Automaten kommunizieren bzw. synchronisieren sich über die Steuersignale wie *address strobe* (AS) und *acknowledge (ACK).*

Zusammenfassung

- Struktur der Verbindungsnetzwerke richtet sich nach einer Vielzahl von Randbedingungen; Spannbreite von
  - einfachen Leitungen
  - bis zu Kommunikationsprozessoren
- Adressierung:
  - Separate E/A-Adressen
  - *Memory-Mapped I/O*
- Busse mit oder ohne Bestätigung?
  - Unidirektionale Busse
  - Bidirektionale
- Wie stimmt man die Geschwindigkeit der E/A-Geräte mit der des Prozessors ab?

Ablauf der Kommunikation zwischen CPU und Gerätesteuerungen

- Wie sieht das Verbindungsnetzwerk aus?
- Wie adressiert man die E/A-Geräte und deren Steuerungen?
- Busse mit oder ohne Bestätigung?
- Wie stimmt man die Geschwindigkeit der E/A-Geräte mit der des Prozessors ab?

Techniken:

1. Direkte Ein-/Ausgabe (*Immediate devices*)
2. Status-Methode (*busy waiting*)
3. *Polling*
4. *Interrupts*
5. *Direct memory access* (DMA)

Programmed I/O

1. Direkte Ein-/Ausgabe (*immediate devices*)

In manchen Fällen: keine Synchronisation erforderlich, weil die Geräte immer zu Ein- und Ausgaben bereit sind und weil die Häufigkeit des Lesens bzw. Schreibens unwichtig ist.

Beispiele:

- Setzen von Schaltern/Lämpchen
- Ausgabe an Digital/Analog-Wandler
- Schreiben in Puffer
- Setzen von Kontrollregistern

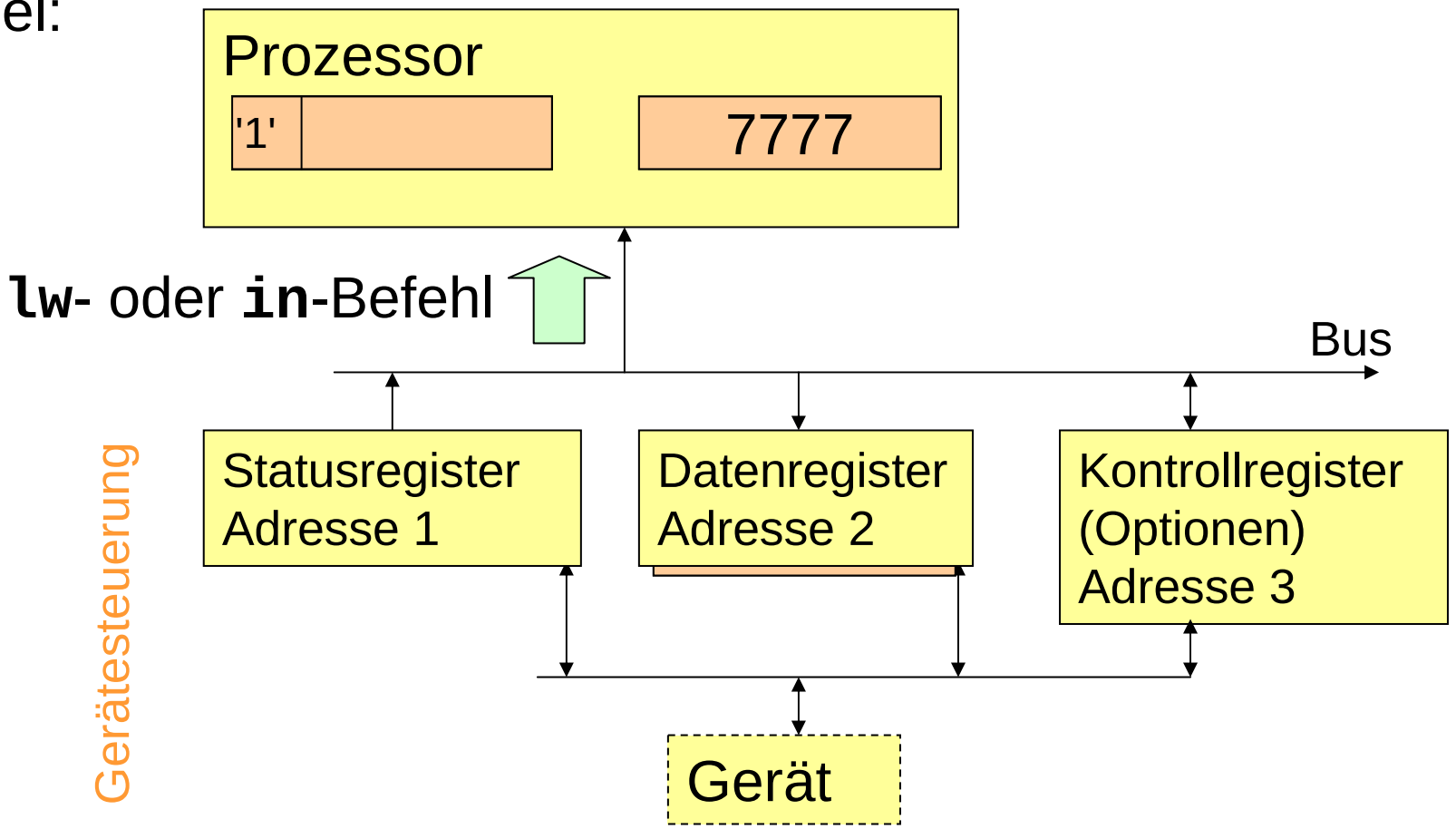
Programmieren:

- **`load`**- bzw. **`input`**-Befehl zur Eingabe
- **`store`** bzw. **`output`**-Befehl zur Ausgabe

2. Status-Methode (*Busy waiting*) - Hardwarestruktur -

Das Gerät- bzw. die Steuerung stellt Bereit-Bit zur Verfügung.

Beispiel:

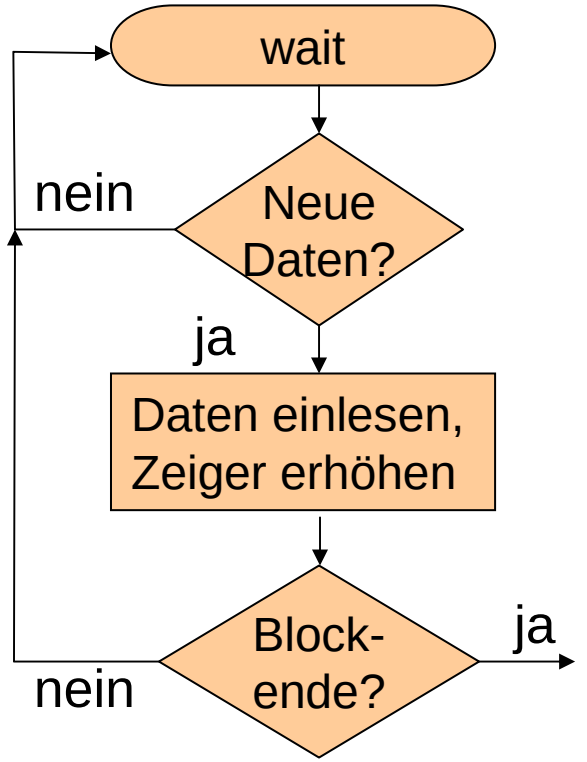


Status-Methode (*Busy waiting*) - Prinzip der Warteschleife -

Das „Bereit“-Bit wird in einer Warteschleife abgeprüft, bis nach Anzeige eines entsprechenden Werts die Ein- bzw. Ausgabe erfolgen kann.

Prinzip:

```
REPEAT
    REPEAT
        lese Statuswort des Gerätes
    UNTIL  Bereit-Bit im Statuswort ist gesetzt;
    lese Datenwort;
    erhöhe Blockzeiger;
UNTIL  Ende des Blocks ist erreicht;
```



Prozessor bleibt während des Wartens beschäftigt (*busy*).

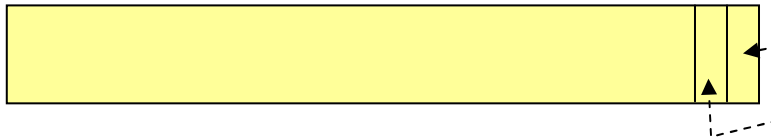
Status-Methode (*Busy waiting*)
- Assembler-Programm für die 80x86-Familie -

```
% Register B enthalte die Blocklänge
% H&L enthalte die Anfangsadresse
wait:
IN 1                % A:= Gerätestatus
CPI ready           % Z:= Bereit-Bit
JNZ wait            % if nicht bereit then goto wait
IN 2                % A := Datenregister der Steuerung
MOV M, A            % Speicher[(H&L)] := A
INX H               % (H&L) := (H&L)+1
DCR B               % B:= B-1; Z:= (B=0);
JNZ wait            % if Z<>'0' then goto wait
continue:           % Block eingelesen
```

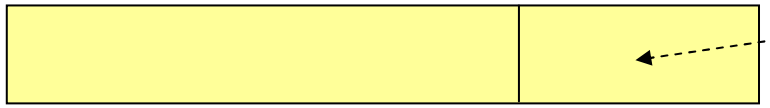
Terminal I/O im Simulator

Simulator simuliert Ein-/Ausgaben zum Terminal
(s. Skript, Anhang B):

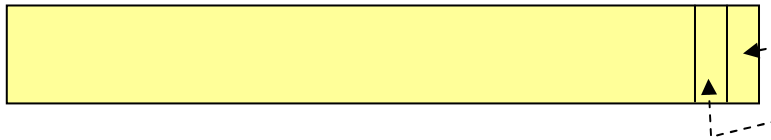
receiver control
(0xFFFF0000)



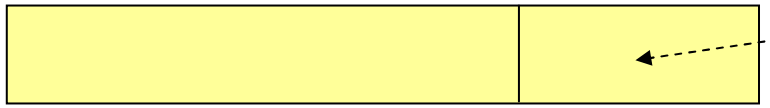
receiver data
(0xFFFF0004)



transmitter control
(0xFFFF0008)



transmitter data
(0xFFFF000c)



Eigenschaften der *busy waiting* Methode

Nachteile:

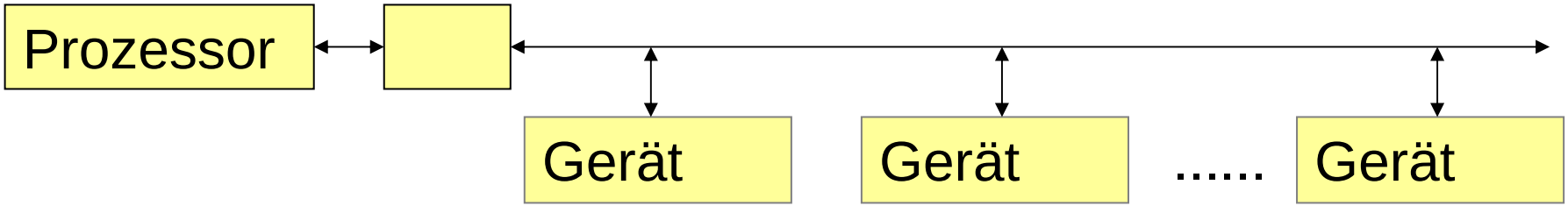
- Keine Möglichkeit der verzahnten Bearbeitung anderer Aufgaben.
- Geringe Übertragungsgeschwindigkeit.

Anwendung:

- Wenn keine anderen Aufgaben vorliegen.
- Wenn das Gerät so schnell ist, dass die Schleife selten durchlaufen wird und für die Umschaltung auf andere Aufgaben keine Zeit bleibt.
- Wenn das Gerät zu keiner anderen Methode fähig ist.

Polling

Polling = gelegentliches Prüfen des Bereit-Bits mit verzahnter Bearbeitung anderer Aufgaben.
Anwendungsbeispiel:
Erfassung von Übertragungswünschen einer großen Anzahl von Geräten:



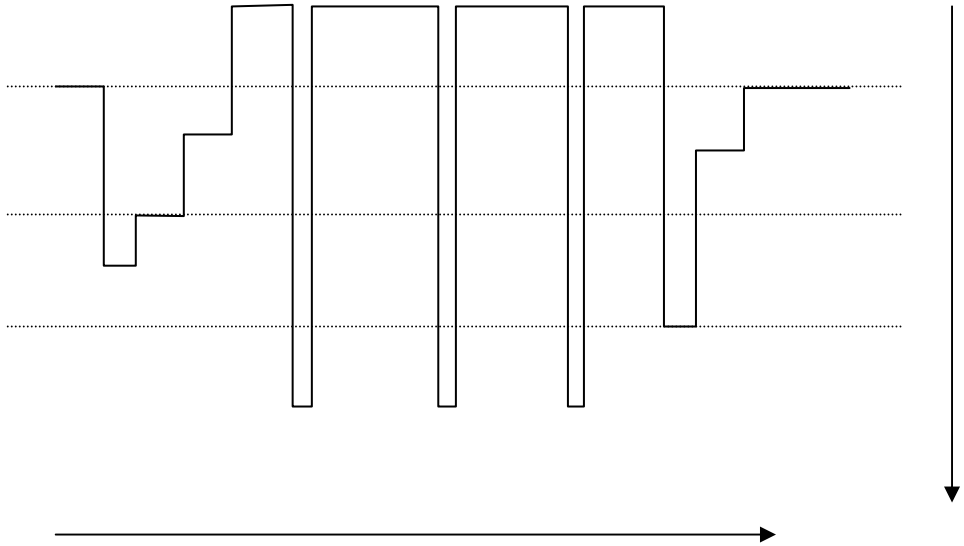
Erlaubt eine gerechte Bedienung vieler Geräte (z.B. Terminals oder USB-Geräte). Vermeidet Überlastungen durch viele Übertragungswünsche; Vorteilhaft bei *denial-of-service*-Angriffen; unter Umständen große Reaktionszeit.

Unterbrechungen (1)


*

Gerätesteuerung unterbricht den Prozessor, falls Datentransport(e) erforderlich werden oder falls Fehler auftreten.

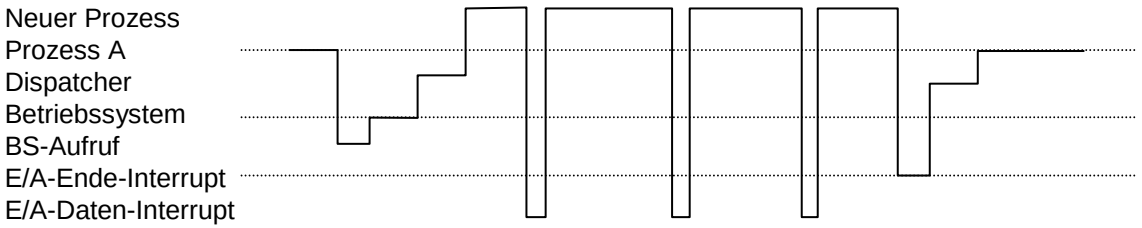
Neuer Prozess
Prozess A
Dispatcher
Betriebssystem
BS-Aufruf
E/A-Ende-Interrupt
E/A-Daten-Interrupt
....
Stromausfall



* http://www.privatedoor.com/AAAImages/PDPicBoss1-200x170.jpg

Ablauf

- Prozess A überträgt E/A-Auftrag an das Betriebssystem.
- *Dispatcher* schaltet auf neuen Prozess um.
- Gerät sendet *Interrupt*.
- Falls *Interrupt* nicht gesperrt ist und die Priorität ausreichend hoch ist und der laufende Maschinenbefehl abgeschlossen ist:
- Sicherung des Zustandes (Software oder Mikroprogramm).
- Feststellen der *Interrupt*-Ursache (ggf. *vectored interrupt*).
- Datentransport: Ein-/Ausgabe, Lesen/Schreiben Speicher, Pufferzeiger erhöhen, Test auf Blockende.
- Restaurieren des Kontexts, *return from interrupt*.
- Prozess fährt in Bearbeitung fort.
- Nach einer Reihe von Datenübertragungsinterrupts erfolgt ein Blockende-*Interrup*t welcher den *Dispatcher* startet.



Eigenschaften

- Wegen des *Overheads* nicht für die schnelle Datenübertragung geeignet

Zu programmieren:

- Starten der Geräte vor der Übertragung.
- Übertragung per *Interrupt*-Routine.

Anwendungen:

- Häufig zur Übertragung zum Drucker eingesetzt.

Für Rechner ohne *direct memory access* (s.u.) ist diese Methode erforderlich.

Direct memory access (DMA) & Buszuteilung

Beim ***direct memory access*** (DMA) werden die Datentransporte **direkt** zwischen Gerätesteuerung und Speicher durchgeführt, **ohne** Beteiligung des Prozessors.

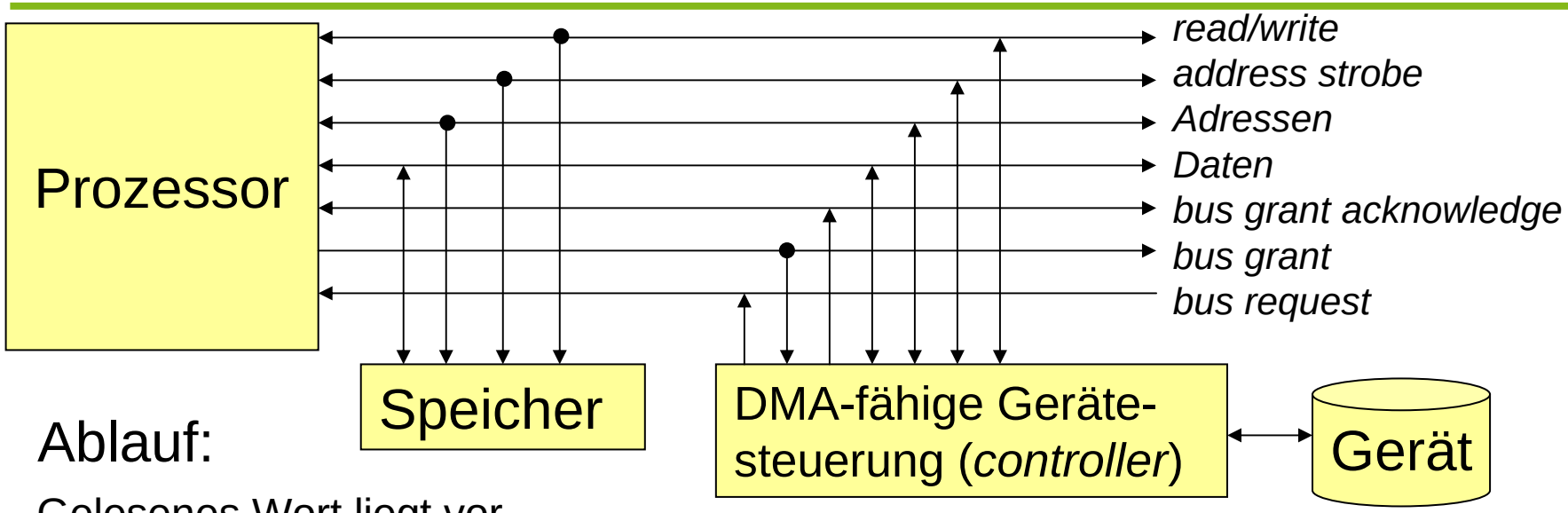
Bislang: nur der Prozessor hat die volle Kontrolle über den Bus (legt Adressen an und erzeugt Kontrollsignale, ..),
andere Einheiten dürfen nur auf seine Anforderung tätig werden.

Diese Funktion des Prozessors heißt ***Bus-Master*-Funktion**.

DMA macht es erforderlich, dass Gerätesteuerungen *Bus-Master* werden können.

Die Vergabe der *Bus-Master*-Funktion an einen Bewerber heisst **Buszuteilung (*bus arbitration*).**

Direct memory access (DMA) - Beispiel: Motorola 68000er/FreeScale ColdFire Prozessoren -



Ablauf:

Gelesenes Wort liegt vor

- ☞ *controller* setzt *bus request*
- ☞ Prozessor setzt *bus grant*, wenn bereit
- ☞ *controller* setzt *bus grant acknowledge*
- ☞ *bus grant* wird zurückgenommen

} Buszuteilung (einfacher Fall, ohne Wettbewerb)

- ☞ *controller* belegt Adress- und Datenbus
- ☞ *controller* setzt address strobe
- ☞ Speicher übernimmt Information vom Datenbus

} *controller* ist *bus master*

- ☞ *controller* löscht *bus grant acknowledge*, erhöht Pufferzeiger, prüft auf Ende
- ☞ Prozessor kann Bus wieder benutzen.

Eigenschaften der DMA-Technik

Bessere Datenrate: mehrere Datenworte *en bloc* über den Bus übertragen (***block mode***), ohne neue Buszuteilung.

Besonders sinnvoll, falls die Geräte selbst Puffer haben (z.B. *track buffer* bei Plattenlaufwerken).

Zu programmieren:

- Initialisierung der DMA-Hardware
- Ende-Behandlung
- Fehlerbehandlung

☞ Aus Programmierersicht ist DMA die einfachste Technik.

Zusammenfassung

Ablauf der Kommunikation

Techniken:

1. Direkte Ein-/Ausgabe (*Immediate devices*)
2. Status-Methode (*busy waiting*)
3. *Polling*
4. *Interrupts*
5. *Direct memory access* (DMA)

Buszuteilung

Wie entscheidet man bei der Busvergabe im Fall von mehreren Wettbewerbern?

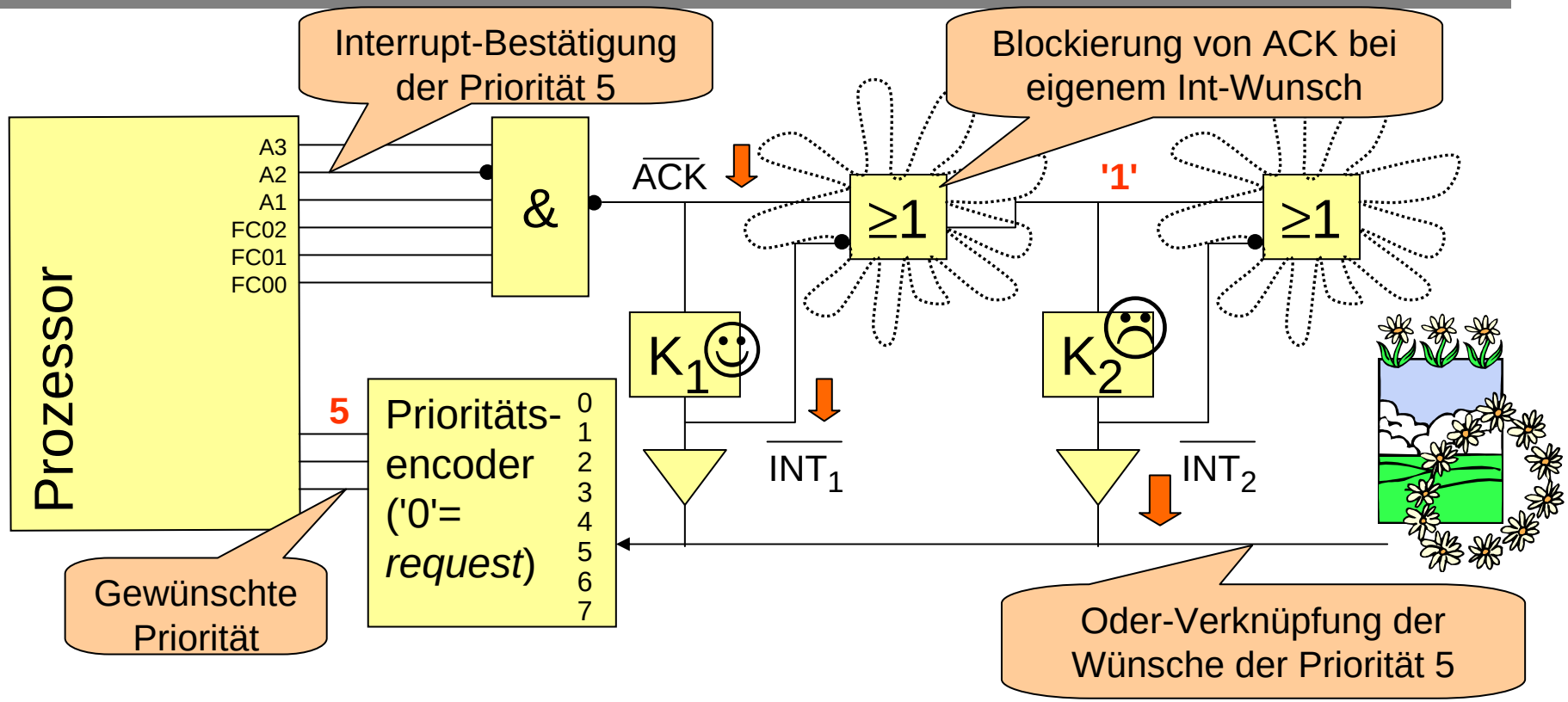
Methoden der Buszuteilung (*bus arbitration*):

1. *Daisy chaining* (Gänseblümchen-Verkettung)
2. *Centralized, parallel arbitration*
3. *Distributed arbitration by self selection*
4. *Distributed arbitration by collision detection*
5. *Token ring, token bus,* weitere LAN-Verfahren (siehe Abschnitt 2.5.4.4)

Siehe auch: Hennessy/ Patterson

Daisy chaining (Gänseblümchen-Verkettung)

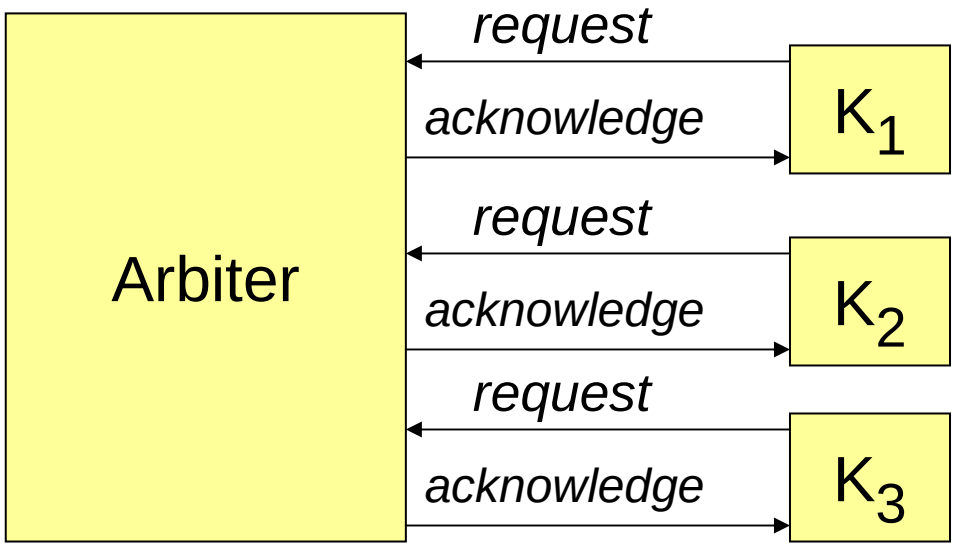
Relative Position der *Controller* am Bus bestimmt Zuteilung.



Daisy chaining ist einfach, aber nicht fair (*controller* können ggf. nie die Zuteilung bekommen („verhungern“)).

Centralized, parallel arbitration

... multiple request lines,
... devices independently request the bus [HP95]

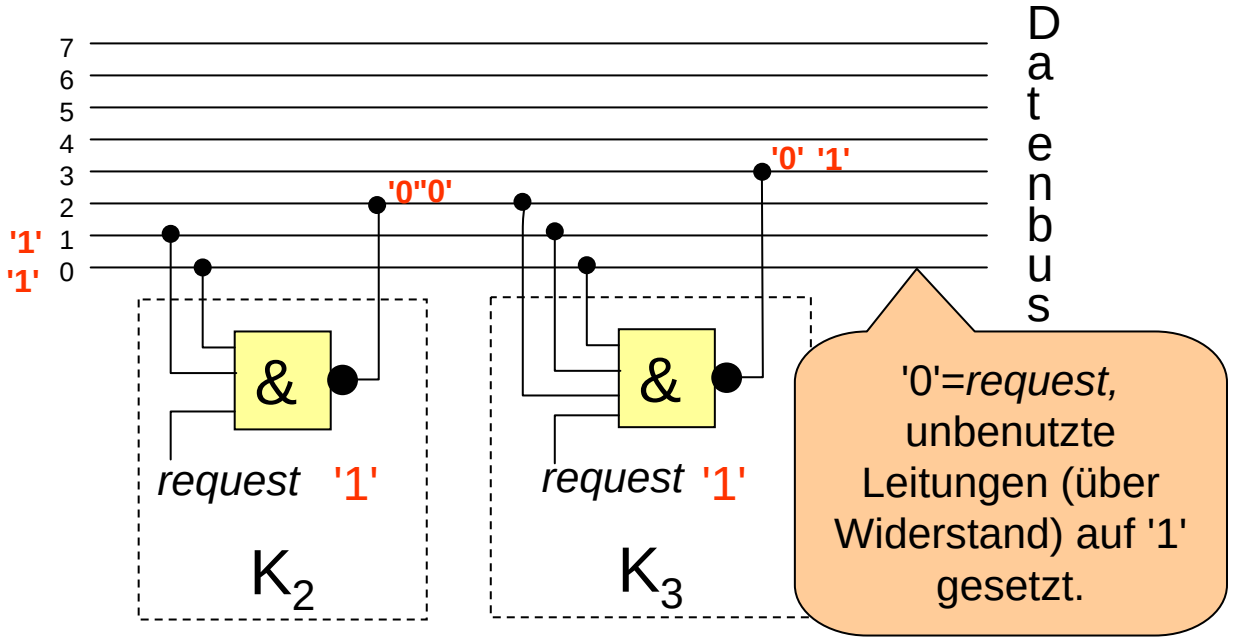


Central arbiter ... may become the bottleneck for bus usage

[HP95]

Distributed arbitration by self selection

These schemes also use multiple request lines, but the devices requesting bus access determine who will be granted access. [HP96]

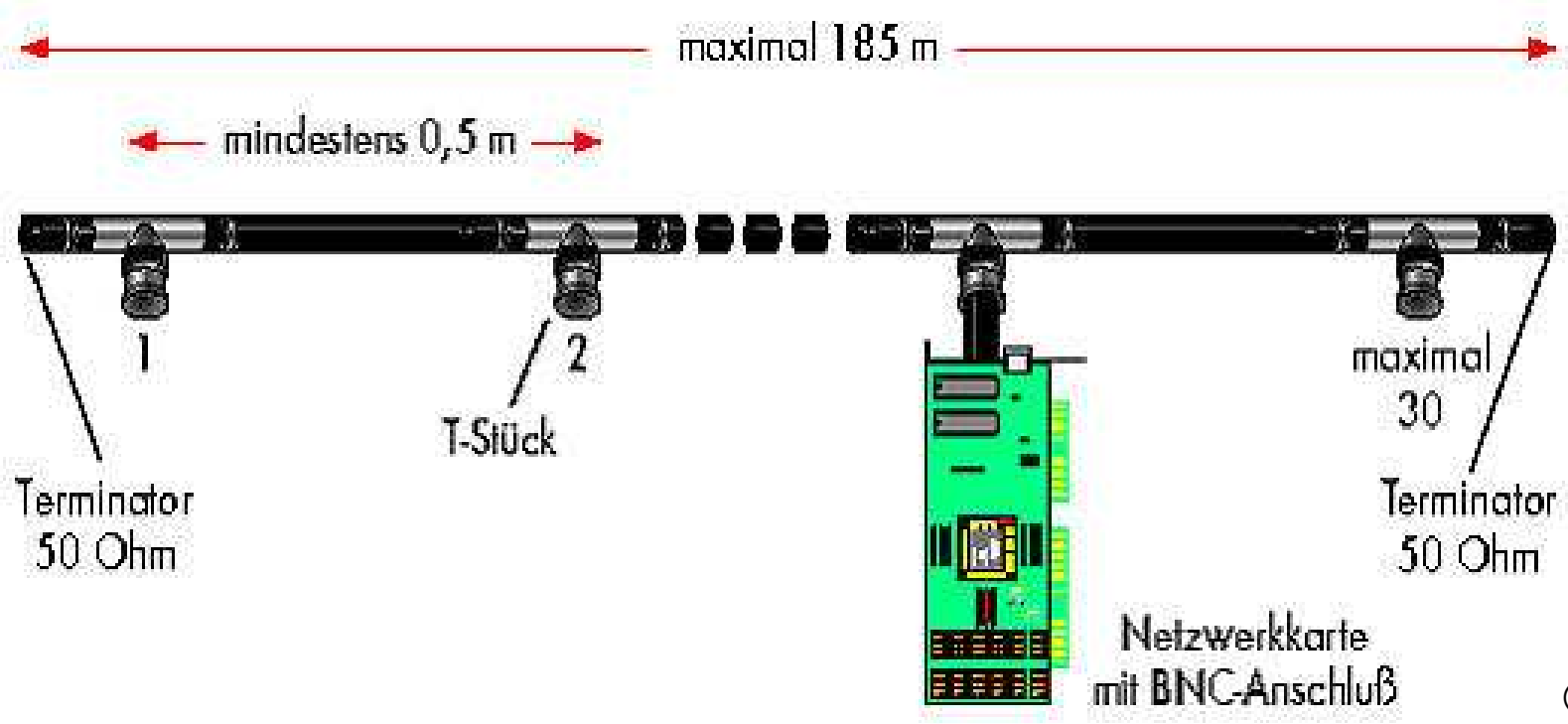


Phase, in der Datenbus für *abitration* benutzt wird. Ein eigener *request* erscheint nur dann als '0' auf dem Bus, wenn kein Wunsch einer höheren Priorität auf dem Bus angemeldet ist.

Max. Anzahl der *controller* = Anzahl der Datenleitungen.
Anwendungsbeispiel: SCSI.

Distributed arbitration by collision detection

In this scheme, each device independently requests the bus. Multiple simultaneous requests result in a collision. The collision is detected and a scheme for selecting among the colliding parties is used. [HP95]



Weitere Eigenschaften von Bussen

- ***Multiplexing* von Daten und Adressen:**
  Daten und Adressen werden nacheinander übertragen.
  Ziel: Einsparung von Leitungen, insbesondere bei großen (> 32 Bit) Adressräumen.
- ***Split transaction busses:***
  Zwischen dem Senden einer Adresse und dem Lesen der Daten können weitere Aufträge abgewickelt werden.
  Zu diesem Zweck wird jedem Auftrag ein Auftrags-Identifikator (aus wenigen Bits) mitgegeben.
  Hilft beim Verstecken der Zugriffszeit (Latenzzeit) auf den Speicher.

Standardbusse

Speicherbusse: schnellste Busse

Lokale Busse:

- **AT-Bus, ISA-Bus**
  AT: Speicher- und Peripheriebus für IBM-PCs ab 1981
- **PCI-Bus**
  Für Einsatz in PCs konzipiert, 32-64 Daten und Adressen
- **AGP** *(„advanced graphics port“)*
  Spezieller Bus für Graphikkarten

Charakteristische Daten von lokalen Bussen

	ISA-Bus	PCI-Bus	PCI-100	AGP (4x)	PCI-Express
Einführungsjahr	1981	~1994			2004
Datenleitungen	8/16	32 (64)	32 (64)		
Adressenleitungen	20	32 (64)	32 (64)		
synchron/asynchron	sync.	sync.	sync.		Paket-vermittlung
Bustakt [MHz]	4,77	33	100	133*	2500
Transferrate, Einzelworte, bei T_{acc}=0 [MB/s]					
Transferrate, Blockmode, bei T_{acc}=0 [MB/s]	9,54	111	333	1000	4GB/s je Verbindung mit 16 „*Lanes*“
Bemerkung				*Nutzung beider Flanken	Serielle Punkt zu Punkt-Verbindungen

Zusammenfassung

Methoden der Buszuteilung (*bus arbitration*):

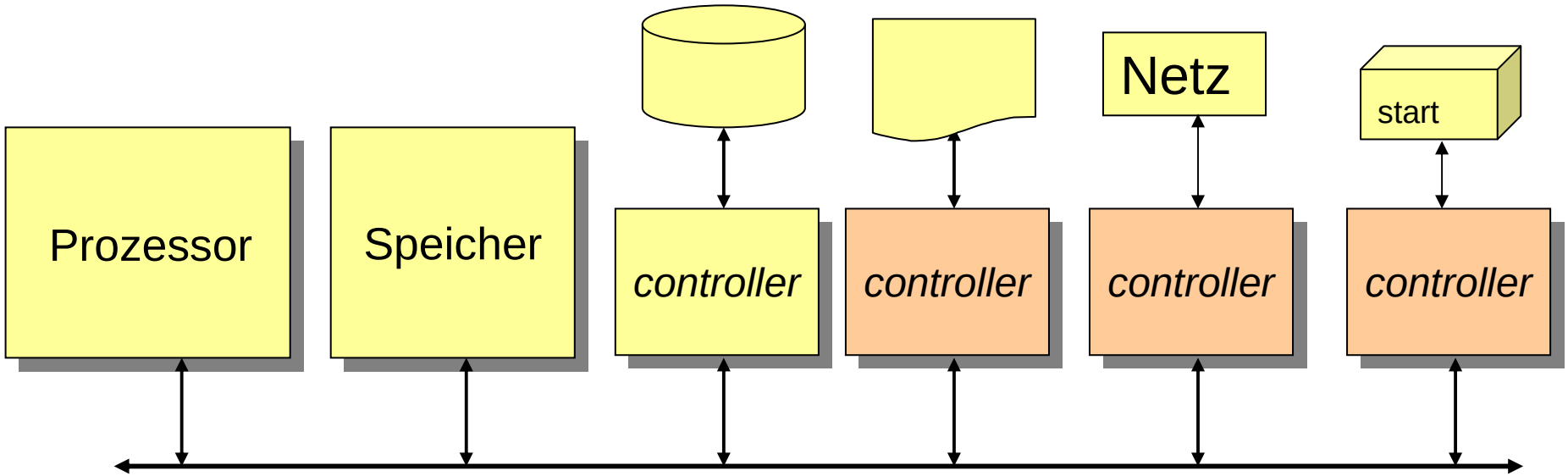
1. *Daisy chaining* (Gänseblümchen-Verkettung)
2. *Centralized, parallel arbitration*
3. *Distributed arbitration by self selection*
4. *Distributed arbitration by collision detection*
5. *Token ring, token bus,* weitere LAN-Verfahren

(siehe Abschnitt 2.5.4.4)

Charakterisierung von Bussen durch

- Übertragungsrate, *split transactions*, Multiplexing, u.a.

Datenübertragung, lokale Peripherie



Datenübertragungsraten gemessen in Bit/s oder Bytes/s.

Datenübertragung, lokale Peripherie

Aus Aufwandsgründen auf parallele Übertragung von Bytes begrenzt oder generell bitserielle Übertragung

Beispiele:

- Asynchrone & synchrone serielle Schnittstellen
- USB (*universal serial bus*)
- *FireWire* (IEEE 1394)
- SCSI (*small computer's system interconnect*)
- SATA
- *Ethernet*
- *Token ring*
- Drahtlose Übertragung

Asynchrone serielle Schnittstellen

Asynchrone serielle Schnittstellen sind

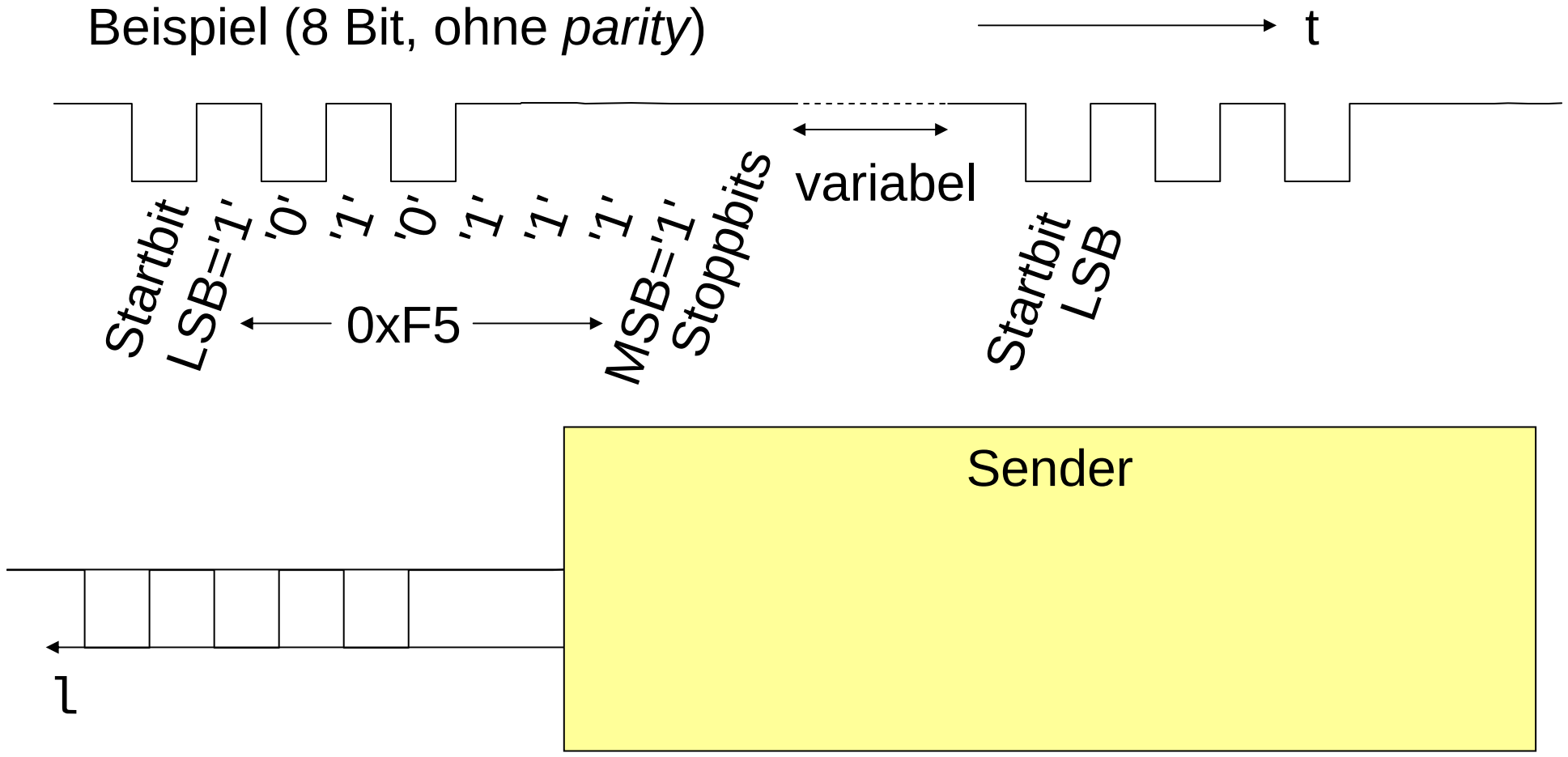
- **seriell** (d.h. die Bits werden nacheinander über eine Leitung übertragen) und
- **asynchron** (das bedeutet hier: der zeitliche Abstand zwischen aufeinanderfolgenden **Zeichen** ist variabel)

Nacheinander werden folgende Werte übertragen:

- das **Startbit**: dies ist immer eine '0'. Sie dient zur Synchronisierung zwischen Sender und Empfänger
- [5,] 7 oder 8 **Datenbits**, *LSB travels first*
- **Paritätsbit**: gerade, ungerade oder keine Parität
- 1-2 **Stoppbits**: Mindestlänge der Pause zwischen 2 Zeichen

Format

Beispiel (8 Bit, ohne *parity*)



Eigenschaften

- Die Zeit für jeweils ein Bit richtet sich nach der Datenübertragungsrate, gemessen in Bit/s.
- Auf kurzen Entfernungen (im Raum) sind Übertragungsraten bis knapp oberhalb von 100 kbit/s erreichbar.
- Die Übertragungsraten sind bei Sender und Empfänger gleich einzustellen.

Unterscheidung bezüglich Übertragungsrichtungen:

- **Voll-Duplex-Betrieb**: Übertragung in beide Richtungen gleichzeitig.
- **Halb-Duplex-Betrieb**: Übertragung in beide Richtungen, aber nur in eine zur Zeit.
- **Simplex-Betrieb**: Übertragung in eine Richtung.

V.24-Norm

V.24-Standard = RS 232-Standard.

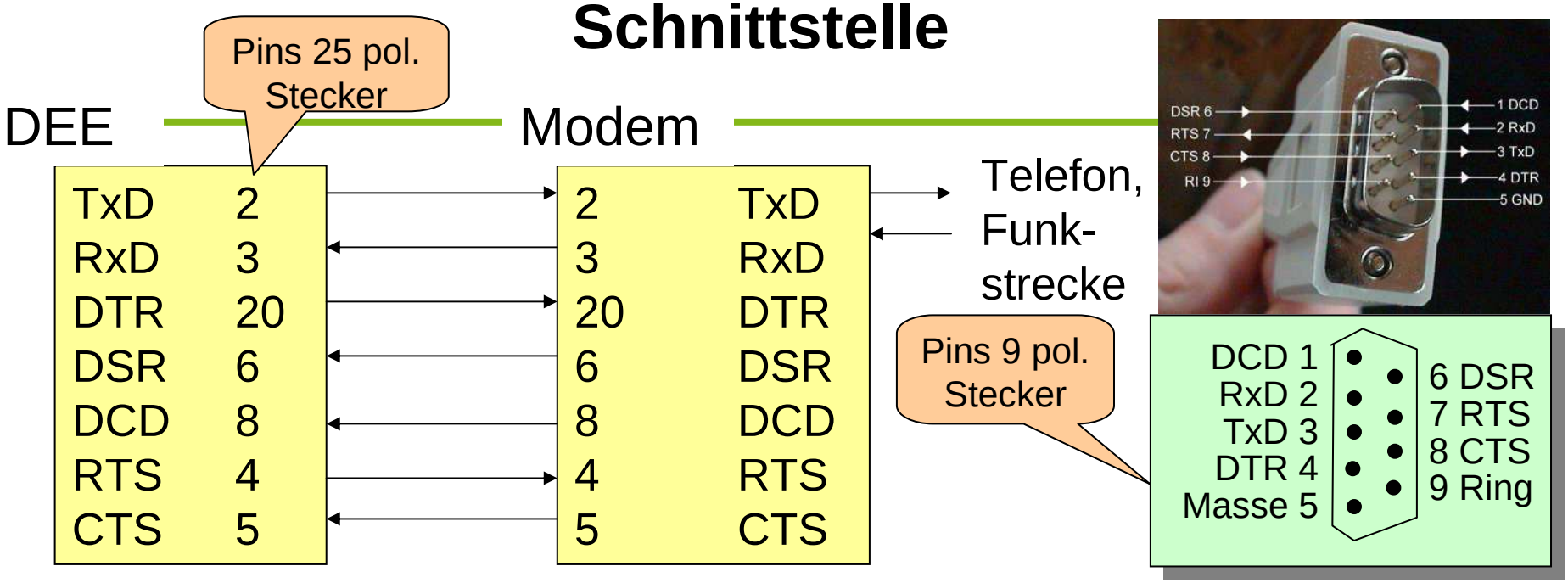
Normierung der asynchronen Übertragung zwischen:

- einem Daten-Endgerät DEE (*data terminal equipment DTE*), d.h. Rechner oder Terminal,
- und einer Datenübertragungseinrichtung DÜE (*data communication equipment DCE*), z.B. einem Modem (Modulator/Demodulator).

☞ Besonderheiten bei der Übertragung zwischen zwei Rechnern.

www.hsg-kl.de/.../msr/seriell/prowo04/TCPD.php

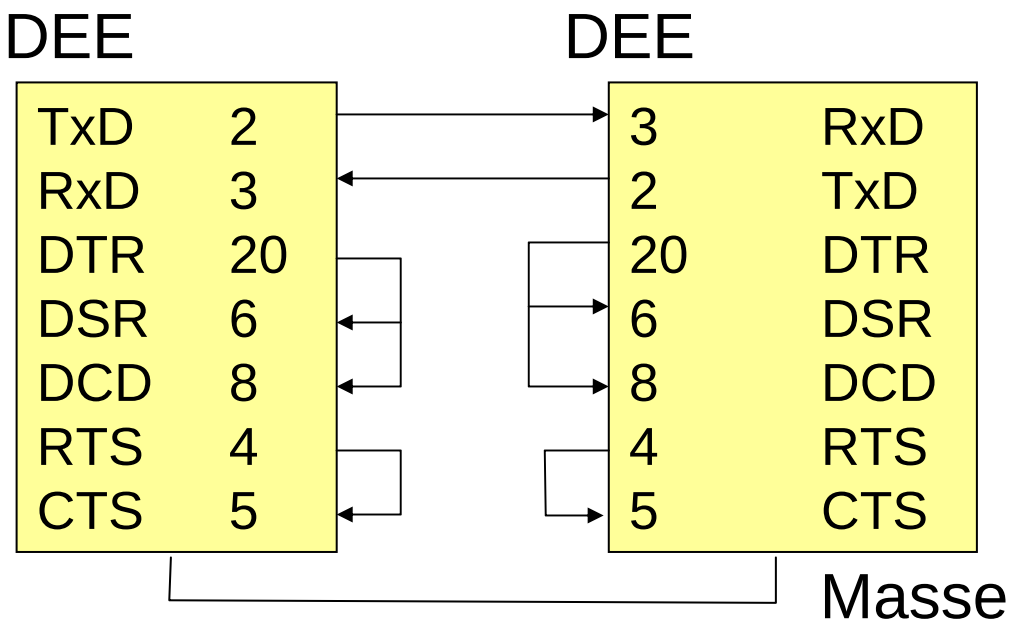
Schnittstelle



- **TxD** (*transmit data*): Sendedaten
- **RxD** (*receive data*): Empfangsdaten
- **DTR** (*data terminal ready*): DEE ist betriebsbereit
- **DSR** (*data set ready*): Partner ist betriebsbereit
- **RTS** (*request to send*): DEE möchte Daten senden
- **CTS** (*clear to send*): Partner kann Daten empfangen
- **DCD** (*carrier detect*): Partner empfängt analoges Trägersignal

Kommunikation zwischen Endgeräten ohne Hardware-Flusskontrolle

Häufig steht nur eine 4-Draht –Verbindung (2x Daten, 2x Masse) zur Verfügung. Verwendung:



Spezielle Kabel (2/3 vertauscht)

Interpretation:

Wenn die DEE eingeschaltet ist, nimmt sie an, der Partner ist es auch.

Bei Sendewunsch über RTS erfolgt über CTS sofort die Bestätigung.

☞ keine Prüfung auf Überlastung.

Zur Absicherung gegen Überlastung Benutzung einer Software-Flusskontrolle (*handshaking*, z.B. mit ↑ S und ↑ Q).

Eigenschaften der Software-Flusskontrolle

Zur Absicherung gegen Überlastung Benutzung einer Software-Flusskontrolle (z.B. mit ↑S und ↑Q).

+: geringe Anzahl von Leitungen

-: ↑S , ↑Q dürfen in Daten nicht vorkommen,

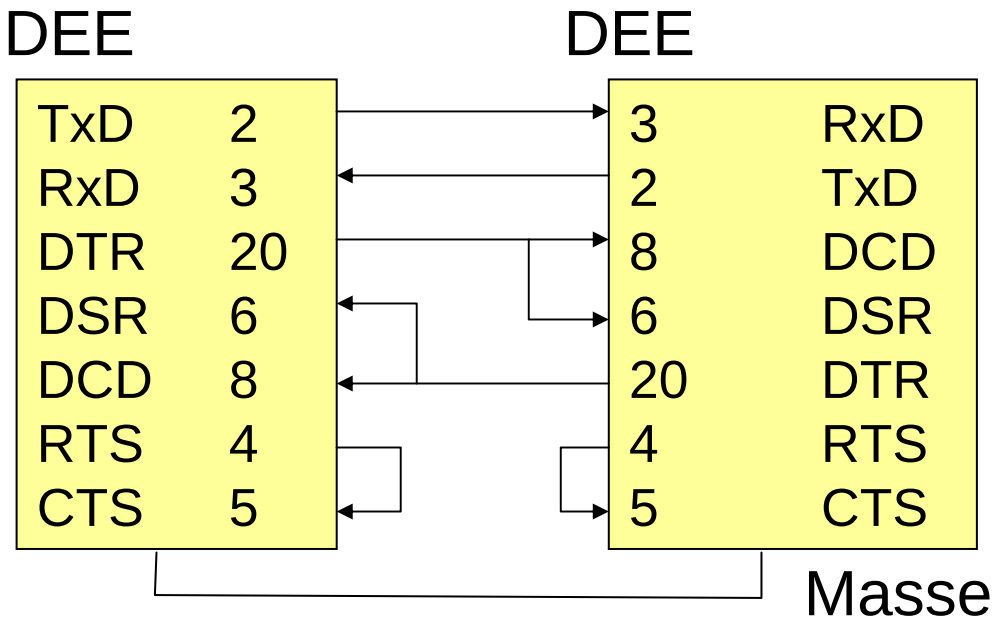
-: Bei Überlast können weitere Zeichen zu spät kommen oder ganz verloren gehen.

Kommunikation zwischen zwei Endgeräten mit Hardware-Flusskontrolle (*handshaking*)

Bei Anschluss von DEE an Modem: alle Anschlüsse mit gleicher Nummer miteinander verbunden.

Passt nicht bei Verbindung 2er DEE (z.B. Rechner).

Mögliche Lösung mit **Hardware-Flusskontrolle**:



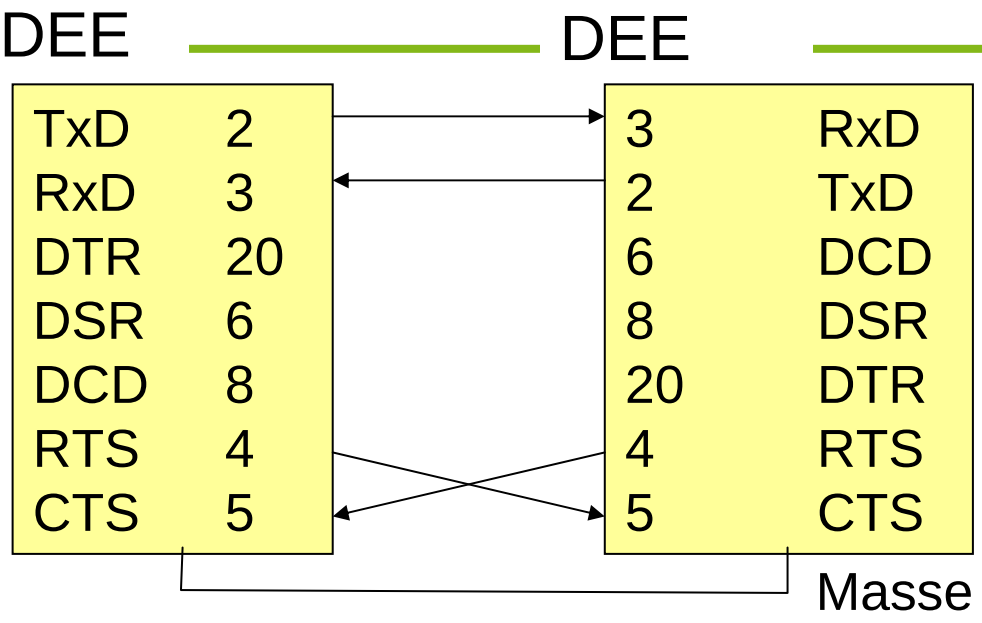
Interpretation:

Senden setzt voraus, dass Partner über DTR seine Bereitschaft zum Empfang von Daten signalisiert.

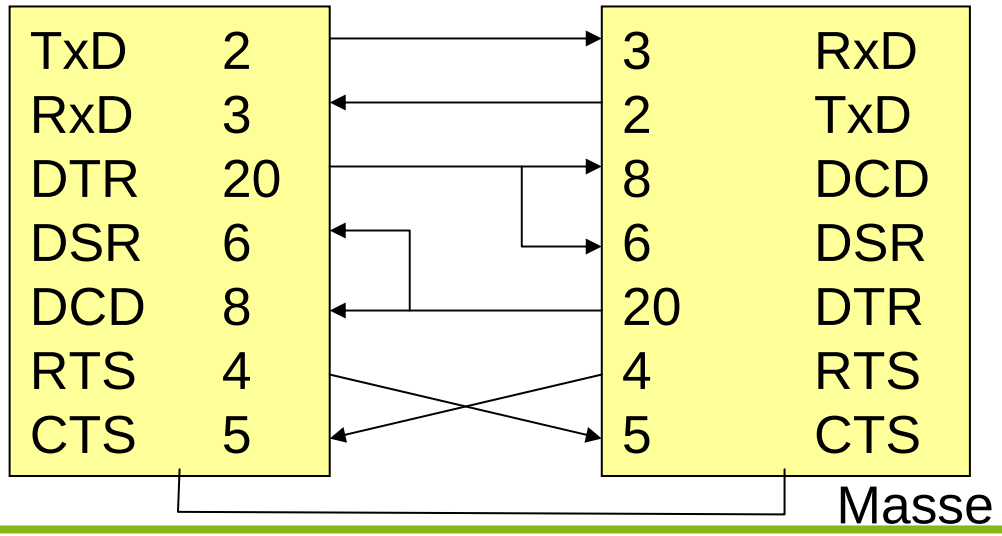
CTS, RTS: Mitteilung eines Übertragungswunsches und der Empfangsbereitschaft; Pins z.Tl. offen oder kreuzweise verbunden.

Neben den durchverbundenen Kabeln („Verlängerungskabeln“) weitere Kabelsorte („Nullmodemkabel“).

Weitere Versionen des *Hardware-Handshaking*



RTS signalisiert hier **Empfangs**bereitschaft. Passt nicht zur ursprünglichen Definition.



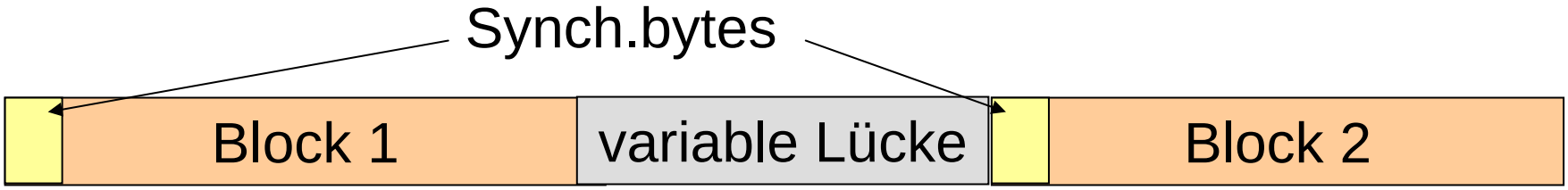
Vom Veranstalter erworbenes Kabel; gemäß Durchtesten; passt für die beiden o.a. Versionen.

Synchrone serielle Schnittstellen

Synchrone serielle Schnittstellen sind **seriell** und **synchron** (das bedeutet hier: der zeitliche Abstand zwischen aufeinanderfolgenden **Zeichen** ist **fest**);

Der Abstand zwischen benachbarten Blöcken bleibt variabel.

Vor einem neuen Block ist jeweils eine Synchronisation (Abstimmung der Lage und exakten Länge der Zeitfenster der einzelnen Bits) durchzuführen*. Dazu werden bestimmte Synchronisationsbytes benutzt.



* Entsprechend der Synchronisation vor jedem Zeichen bei der asynchronen Übertragung

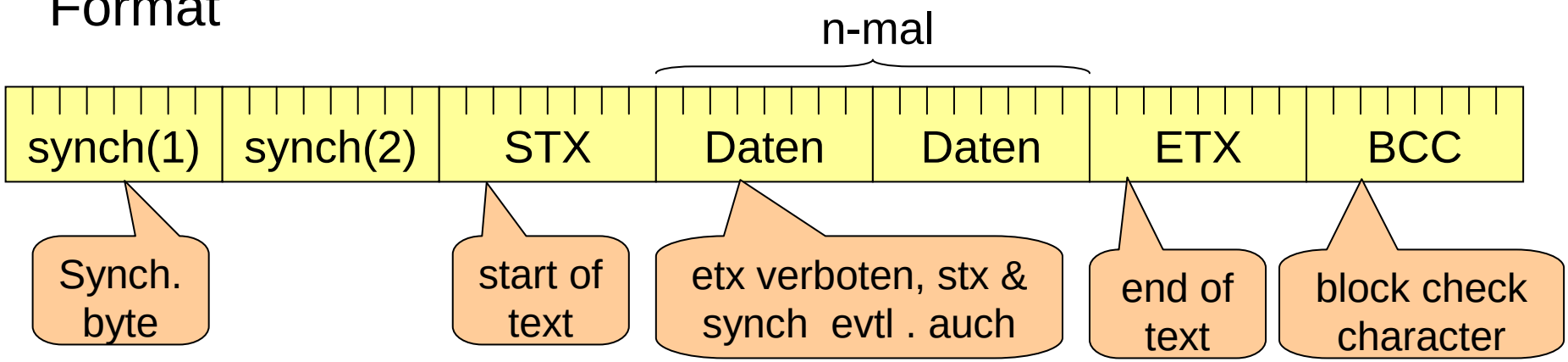
Unterscheidung zwischen zeichenorientierter und bitorientierter Übertragung

Dürfen beliebige Zeichen in den Daten selbst vorkommen?

1. „zeichenorientierte Übertragung“: nein
2. „bitorientierte Übertragung“: ja

Die zeichenorientierte Übertragung

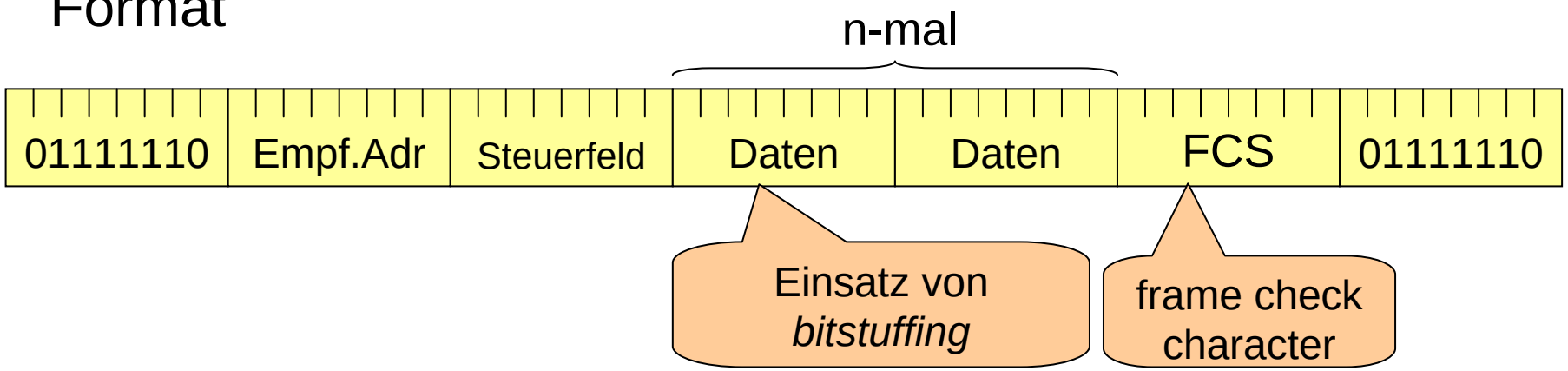
Format



- Es wird immer eine durch 8 teilbare Anzahl von Bits übertragen ☞ byte- bzw. zeichenorientierte Übertragung.
- Funktioniert mit byteorientierter Hardware.
- Kein Problem bei der Übertragung von Textdateien (STX, ETX, SYNCH sind keine druckbaren Zeichen), Problem bei Übertragung beliebiger Dateien.

Bitorientierte Übertragung

Format



- Beim Sender: innerhalb der Daten wird nach jeweils 5 Einsen eine '0' eingefügt (*bit stuffing*).
- Beim Empfänger: nach 5 x '1': folgt '0', so wird sie ausgefügt, folgt '1', so ist der Block zu Ende.
- Es wird **nicht** immer eine durch 8 teilbare Anzahl von Bits übertragen ☞ bitorientierte Übertragung (HDLC ☞ ISDN).

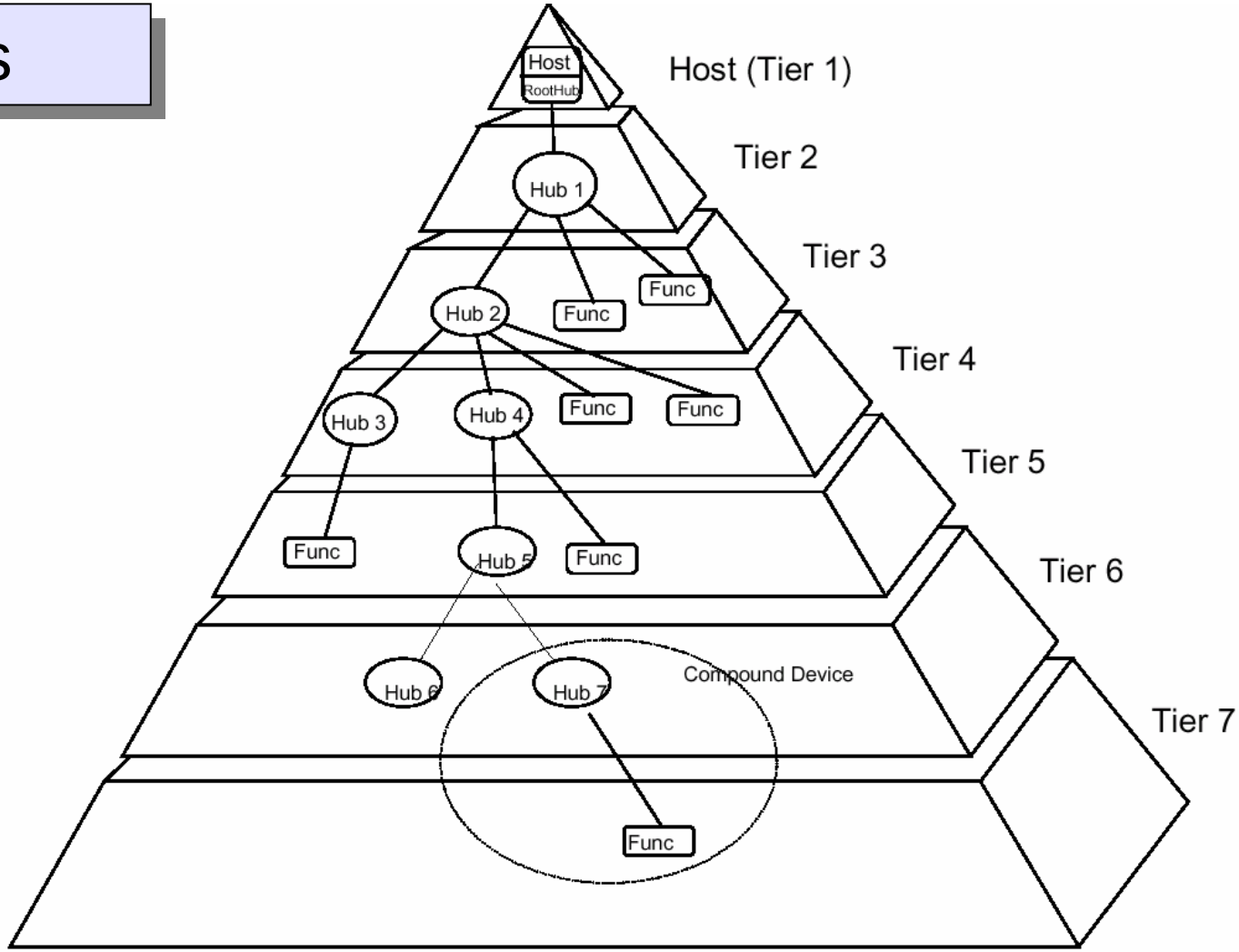
USB-Bus *(Universal serial bus)*

Eigenschaften:

- USB setzt stets einen Master (Rechner) voraus
- Datentransfers werden durch Polling von diesem Rechner aus eingeleitet
- Datenraten: 1,5 Mbit/s oder 12 Mbit/s (USB 1.1), 500 Mbit/s (USB 2.0), 5 Gbit/s (USB 3.0)
- 3 Übertragungsmodi:
  - max. 54 Byte, garantiert pro Zeitintervall
  - *bulk transfer*, nur wenn Bandbreite vorhanden ist
  - isochroner Transfer, zeitgenau, keine Fehlerbehandlung
- USB 1.1 nicht für Platten und Kameras geeignet.

Hierarchisch aufgebautes Hub-System

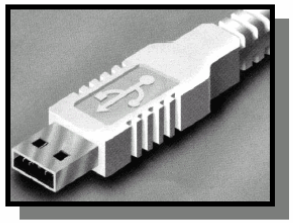
max. 5 Hubs



Verbindungen zwischen Rechnern und Hubs (USB 1.0, USB 2.0)

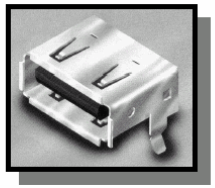
Series "A" Connectors

- Series "A" plugs are always oriented **upstream** towards the *Host System*



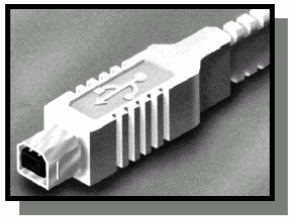
"A" Plugs (*From the USB Device*)

"A" Receptacles (*Downstream Output from the USB Host or Hub*)



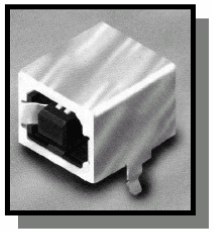
Series "B" Connectors

- Series "B" plugs are always oriented **downstream** towards the *USB Device*



"B" Plugs (*From the Host System*)

"B" Receptacles (*Upstream Input to the USB Device or Hub*)



Series "mini-B" Connectors

- Series "mini-B" plugs are always oriented **downstream** towards the *USB Device*

"mini-B" Plugs (*From the Host System*)

"mini-B" Receptacles (*Upstream Input to the USB Device or Hub*)

Maximale Länge von 5m pro Hub
Unterscheidung in Stecker vom Typ A (Rechner) und B (angeschl. Gerät)
Verlängerungskabel Stecker A auf Buchse A verletzt Spezifikation.
Stecker A auf Stecker A gefährlich!

Aufbau von Netzwerken mit USB 1.0, USB 2.0?

USB-Rechner-zu-Rechner-Verbindungen nicht vorgesehen, Netzwerke nur über spezielle Brücken.

Auszug aus FAQ auf //www.usb.org:

Q7: You mean I can't make a direct cable connection like a null modem?

A7: Correct. In fact, if you try this with an illegal A to A USB cable, you'll short the two PCs' power supplies together, possibly destroying one or both machines or causing a fire hazard. Even (if) there were no danger to the machines from the problem with two power supplies, there still wouldn't be any way to get the two PCs talking to each other, since USB doesn't support that particular kind of communication. A reasonably priced solution to handle this need is the USB bridge.

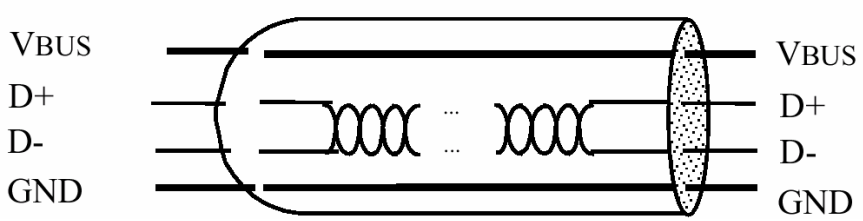
Übertragung der Informationen

Serieller Bus mit differenzieller Übertragung

Stromversorgung für angeschlossene Geräte

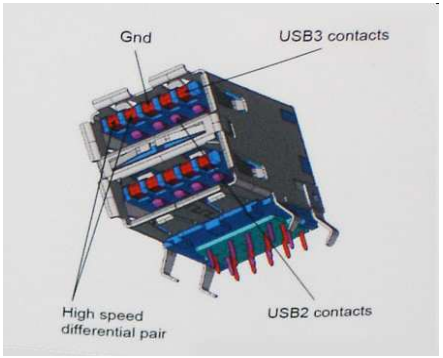
NRZI-Kodierung der Daten

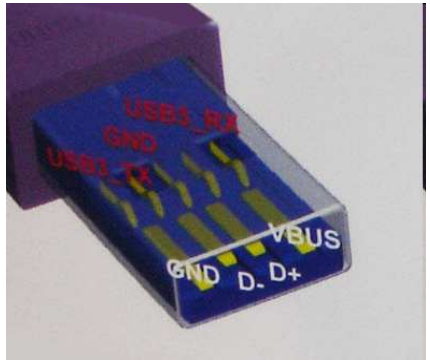
- '1' zu übertragen: Pegel bleibt erhalten
- '0' zu übertragen: Wechsel des Pegels
- *Bit stuffing*
  (Einfügen von Nullen, um Takt erzeugen zu können)



USB 3.0

- Höhere Geschwindigkeit
- Voll-Duplex
- Abkehr vom reinen Polling
- Mögliche Stromentnahme
  - Unangemeldet 150 mA statt bisher 100 mA
  - Angemeldet 900 mA statt bisher 500 mA
- Zusätzliche Kontakte für Duplexfähigkeit und als Masseanschluss
- Abwärtskompatibel: alte Stecker passen in neue Buchsen (diese sind tiefer)
- Einführung ~2010





http://www.reghardware.co.uk

FireWire (IEEE 1394)

Serieller Bus geht zurück auf Ideen () von Apple 1987

- Gleichberechtigte Netzwerkpartner
- Übertragungsgeschwindigkeit:
- IEEE 1394: 100/200/400 Mbit/s
- IEEE 1394b: 800/1600 Mbits/s
- Einzelne FireWire S3200-Geräte
- im Entwurf: bis 4,8 Gbit/s
- Geeignet auch für hochauflösende Kameras, Anschluss von externen Plattenlaufwerken
- Erlaubt Isochron-Betrieb mit garantierten Übertragungs-zeiten, Echtzeit-fähig (wichtig für Audio und Video)
- Selbstkonfigurierend
- max. 63 Geräte, max. 4,5 m zwischen Geräten
- Keine feste Prioritätszuordnung, fair

Eigenschaften

- Keine externen Hubs (Geräte selbst sind Hubs)
- *Peer-to-Peer*-Verbindungen möglich
  (zur Realisierung von Netzwerken geeignet)
- Wurzelknoten wird nach Einschalten oder *Reset* festgelegt
- Jedes Gerät kann zu jedem Zeitpunkt den Bus anfordern
  (Anforderungen werden an Vorgängerknoten weitergereicht)
- Busanforderung nutzt 3. Zustand ('Z') mit aus.

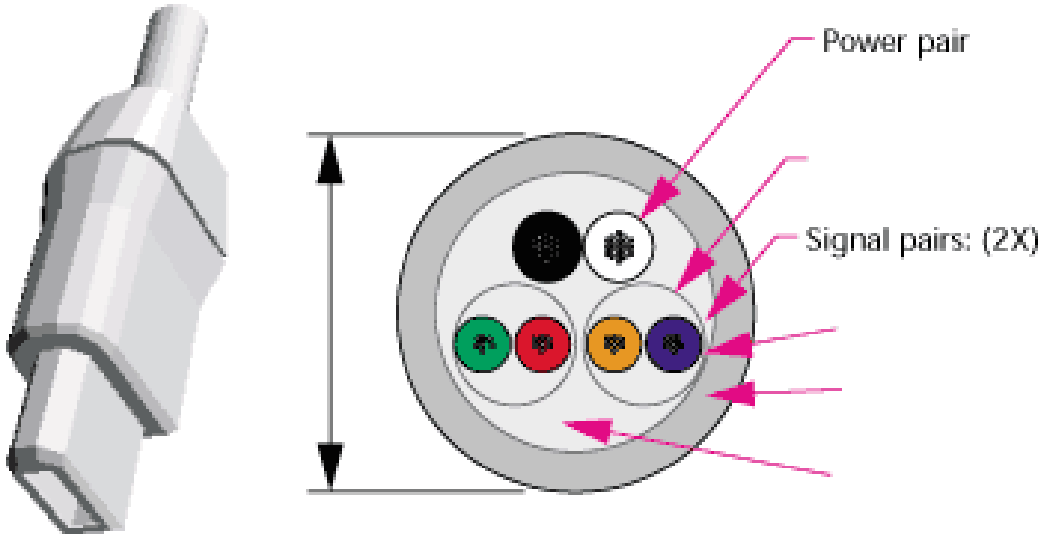
siehe auch:
- Bähring, Mikrorechnertechnik, Bd. II (3. Auflage);
- K. Dembrowski, Feuerdraht, c't 2/97
- //www.1394ta.org

Kabel

2 verdrillte Kabelpaare

- 1394a: getrennte Übertragung von Daten und Takt
- 1394b: *full-duplex*

Optionales 3 Adernpaar bei 6-pol. Stecker für Spannungsversorgung (8-40 V, bis 1,5 A -bis 60 W-)



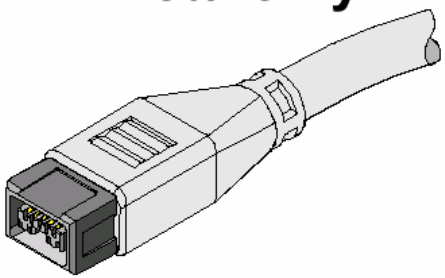
Nutzung des 1394-Protokolls über *Gigabit-Ethernet*-Kabel?

Neue Verbindungstechnik für 1394b

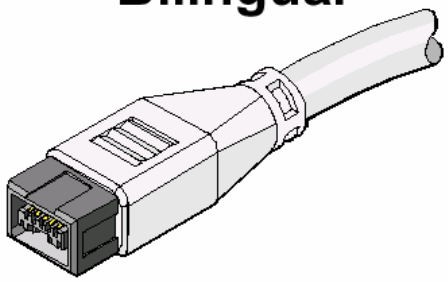
Bilingual and Beta connector plug and socket

- **9 pin**
  - includes power
  - 2 extra pins for signal integrity
  - one pin for reserved for future use
- **Small size - mating interface 8mm x 5 mm**

Beta-only



Bilingual

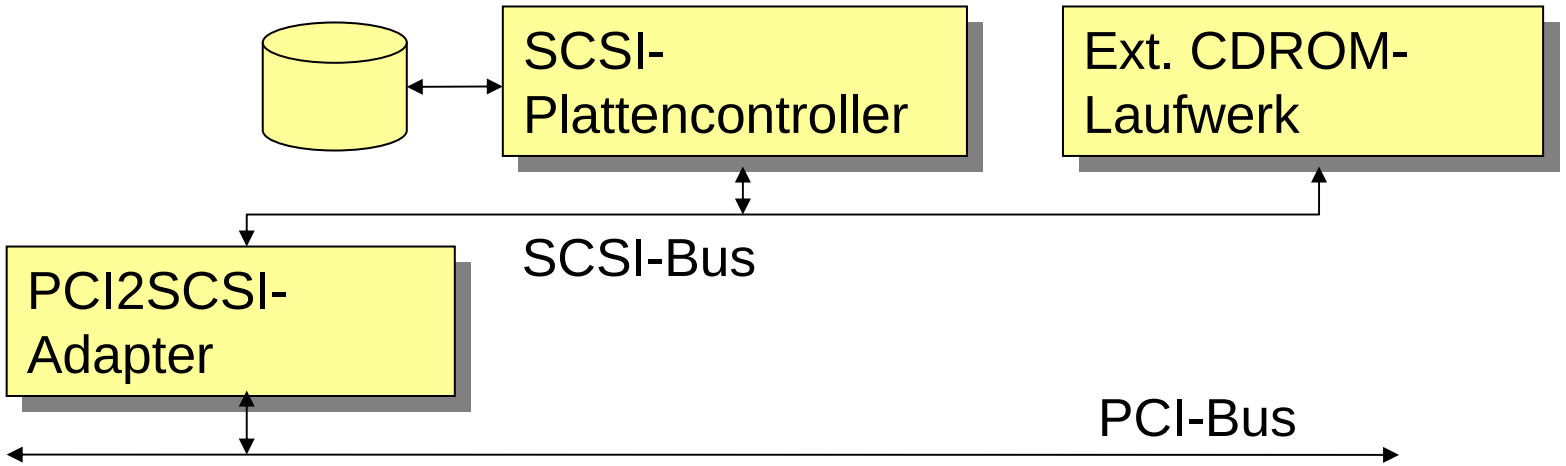


Nov ember 13, 2000 Slide 17

ZAYANTE

Grundsätzliche Eigenschaften des SCSI-Busses

- Ursprünglich für den Anschluss von Platten konzipiert.
- Erlaubt durch standardisierte Befehlsschnittstellen weitgehende Abstraktion vom physikalischen Medium und Entlastung des Prozessors.
- Jeder Teilnehmer kann Übertragung initiieren.
- Max.8 Anschlüsse pro Bus (16 bei *wide* SCSI).



Varianten des SCSI-Busses

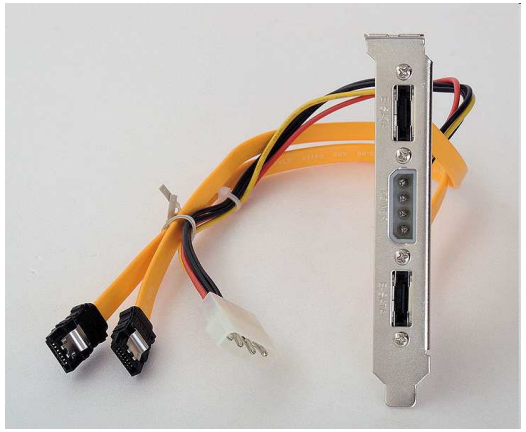
- **SCSI-1:** Ursprüngliche Definition, für Platten; v.a. asynchron (3,3 MB/s).
- **SCSI-2:** andere Geräte, nur synchron, 10MB/s als fast SCSI.
- ***Narrow/Wide* SCSI:** *wide* SCSI ist mögliche Variante von SCSI-2 mit 16 oder 32 Datenleitungen.
- **Fast/Ultra/Ultra2/Ultra160:** Bezeichnungen für Taktrate.
- **differenzielle (symmetrische) Signale:**
  (PC-) übliches SCSI kodiert Werte über Spannungen zwischen einer Signalleitung und Masse (*single ended*, asymmetrisch). Bei differenziellen Signalen benötigt jedes Signal zwei Leitungen. Ist eine positiv gegenüber der andere, bedeutet dies eine '0', ist sie negativ, eine '1'. Bessere Störunempfindlichkeit.

Varianten des SCSI-Busses

	Transfer [MB/s]	Bits	Länge [m]	Geräte, max.	Übertragung
SCSI-1	5	8	6	7	*single-ended*
Fast SCSI	10	8	3	7	*single-ended*
Fast/Wide SCSI	20	16	3	15	*single-ended*
Ultra/Narrow	20	8	3	4	*single-ended*
	20	8	1,5	7	*single-ended*
Ultra/Wide	40	16	3	4	*single-ended*
	40	16	1,5	7	*single-ended*
Ultra2/Wide	80	16	12		differenziell

(e)Serial ATA

- Serielle Version des älteren internen parallelen ATA-Festplattenanschlusses
- eSATA ist eine Erweiterung von SATA auf den Anschluss an externe Geräte
- eSATA schließt keine Stromversorgung ein
- SATA-Geräte lassen sich an *serial attached* SCSI nutzen



Zusammenfassung

Datenübertragung lokale Peripherie & Netzwerke

- Serielle Schnittstellen
- USB
- Firewire
- SCSI
- (e)SATA
- Ethernet
- Drahtlose Techniken (Bluetooth, WLAN, …)

Ethernet

Arbitrierungsverfahren (fast) wie beim „Äther“:
jeder sendet, wenn er meint, der „Äther“ sei frei,
und zieht sich zurück, wenn es eine Kollision gibt

„*carrier sense multiple access/collision detect*“ (***CSMA/CD***)
Reduktion der effektiven Datenrate durch Kollisionen.

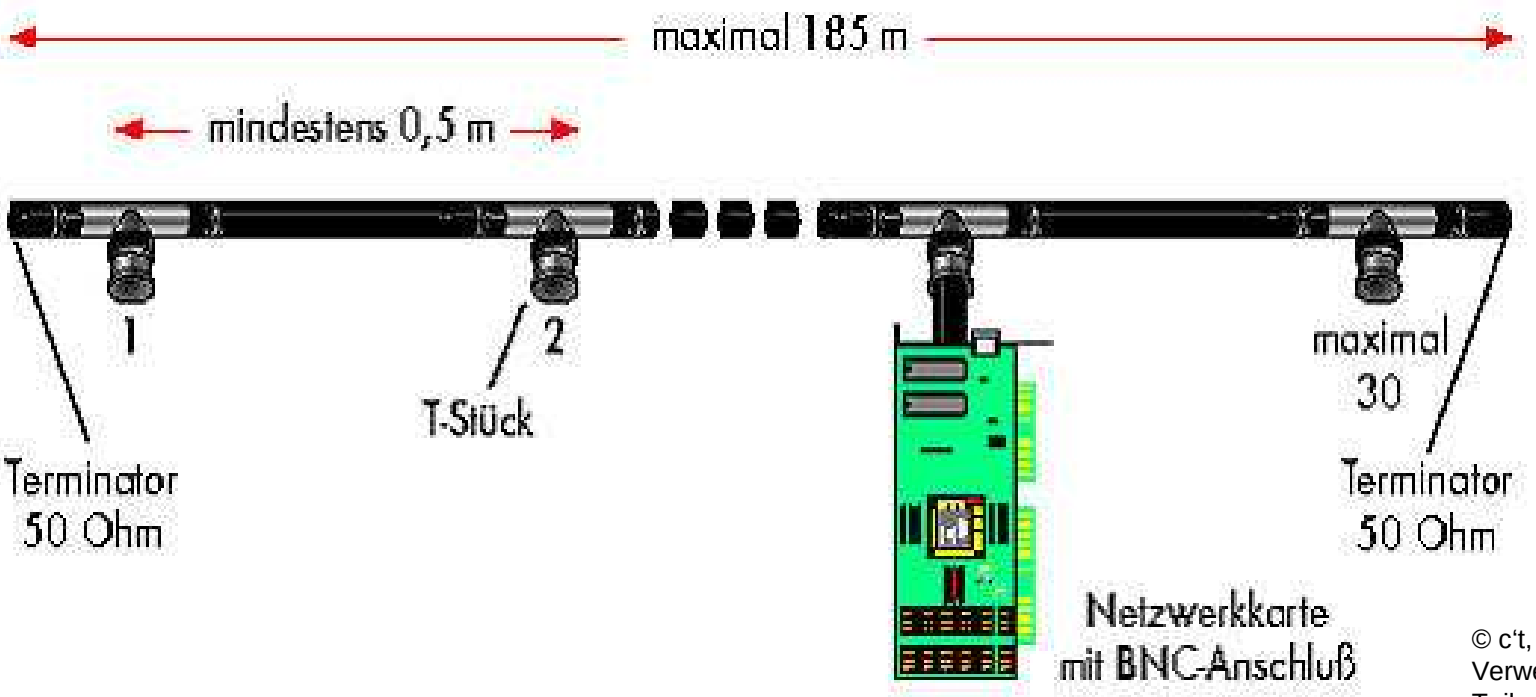
10 Mbit-*Ethernet*

- **10BASE5, *Thick Wire***
  Max. 500m lang, „dickes“ Koaxial-Kabel,
  an Anschlussstellen durchbohrt, an den
  Anschlussstellen *Transceiver*
  (Transmitter/Receiver),
  Kabel vom *Transceiver* zum Rechner

www.answers.com/topic/10base5

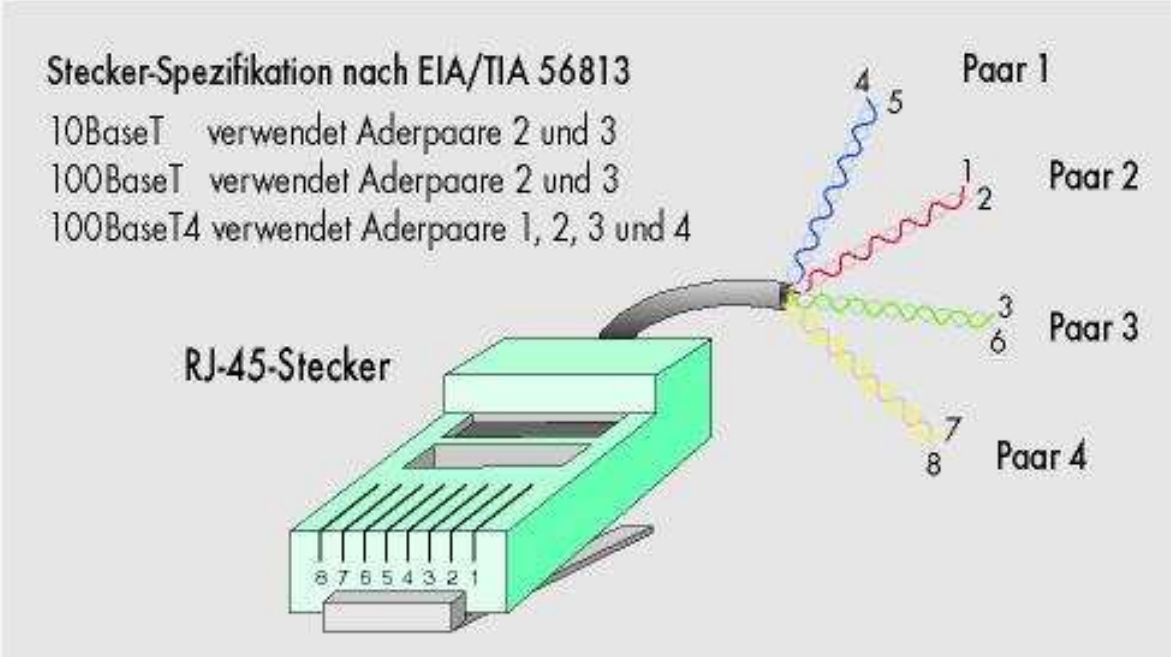
10MBit-Ethernet

- **10BASE2, *Thin Wire, Cheapernet***
  Max.185 m lang, dünnes Koaxial-Kabel, an den Anschlussstellen wird die Verbindung aufgetrennt, Halbduplex-Verbindung.



10MBit-Ethernet

- **10BASE-T, *TwistedPair***
  Max 100m lang, verdrillte Leitungen, differenzielle Übertragung mit Vollduplex-Fähigkeit, Punkt-zu-Punkt-Verkabelung, Einsatz in „strukturierter Verkabelung“

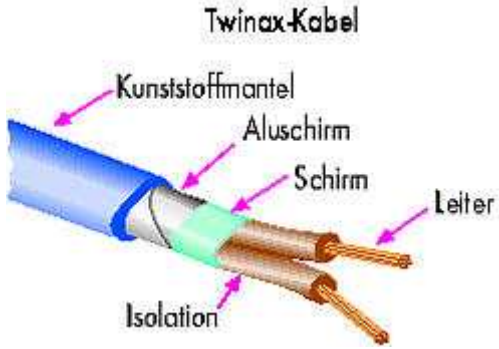


100MBit-Ethernet (Fast-Ethernet)

- **100BASE-T4**
  Kabel der Kategorie 3 (8-polige Telefonkabel), Halbduplex.
- **100BASE-Tx**
  Punkt-zu-Punkt-Verkabelung der Kategorie 5 (Cat 5) (UTP/STP-Kabel); Aufteilung der Daten auf Adernpaare und 3-wertige Logik zur Reduktion der Störstrahlung.
- **100BASE-Fx**
  Sternförmige Glasfaserverkabelung.

1000MBit-Ethernet, Gigabit Ethernet

- **1000BaseLx:** Lichtleiter,
  Segment 2-550m lang, teuer
- **1000BaseCx** (IEEE 802.3z): Max 25 m,
  2 verdrillte *Twinax* verdrillte Kabel oder
  1 Quad-Kabel pro Verbindung



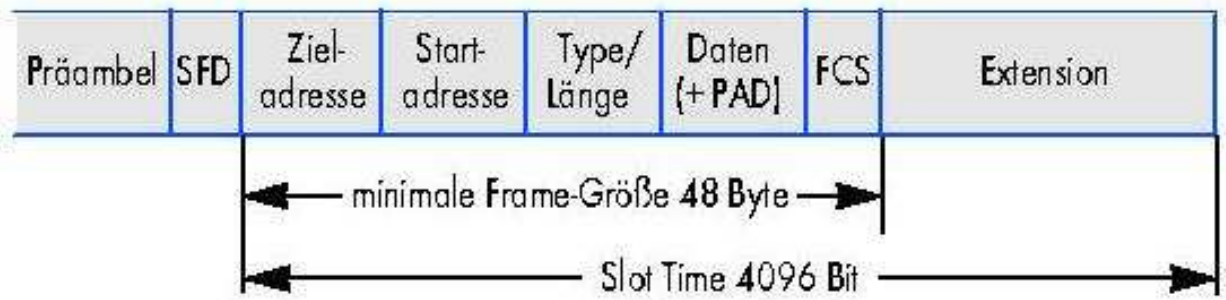
- GigaBit-Ethernet über „Kategorie 5“-Kupferkabel
  unter bestimmten Bedingungen möglich:
  - tatsächlich 4 Adernpaare verlegt,
  - nicht verdrillte Anschlussenden <13mm
  - ≤ 100m lang

  Investitionsschutz für viele Kat. 5 –Netzwerke;
  Besser: *upgrade* auf Kat. 5e- oder Kat. 6-Kabel

1000MBit-Ethernet, Gigabit Ethernet

Alternativ

- Vollduplex-Betrieb ohne Kollisionen
  (d.h.: Abkehr von CSMA/CD)
- oder Halbduplex-Betrieb mit Kollisionen,
  zur Konflikterkennung: Erhöhung der Paketlänge nötig

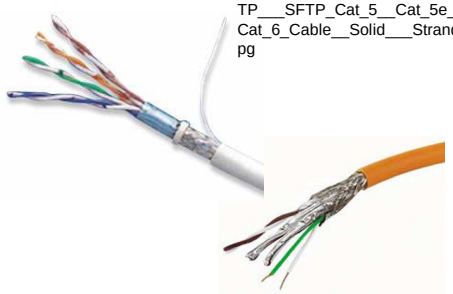


Beim Standard-PCI-Bus (32Bit, 33 MHz) können Gigabit-Ethernet-Karten evtl. die maximale Übertragungsrate nicht mehr erzielen.

10 Gbit-Ethernet

- Das kollisionsbehaftete CSMA/CD-Verfahren wird abgeschafft
- 10 Gigabit Ethernet über **Glasfaser** im Gebrauch.
  Löst andere Techniken für Weitverkehrsnetze ab.
  Lokal als *Backbone* und zur Anbindung von *storage-area networks* (SANs) eingesetzt.
- Verabschiedung des IEEE-Standards für 10 Gigabit Ethernet über **Kupferkabel** im Sommer 2006. Bis ca. 100 m. Alternativen derzeit:
  - Kat. 6-Kabel (mit bisherigen Steckern
  - Kat. 7-Kabel (mit neuen Steckern)

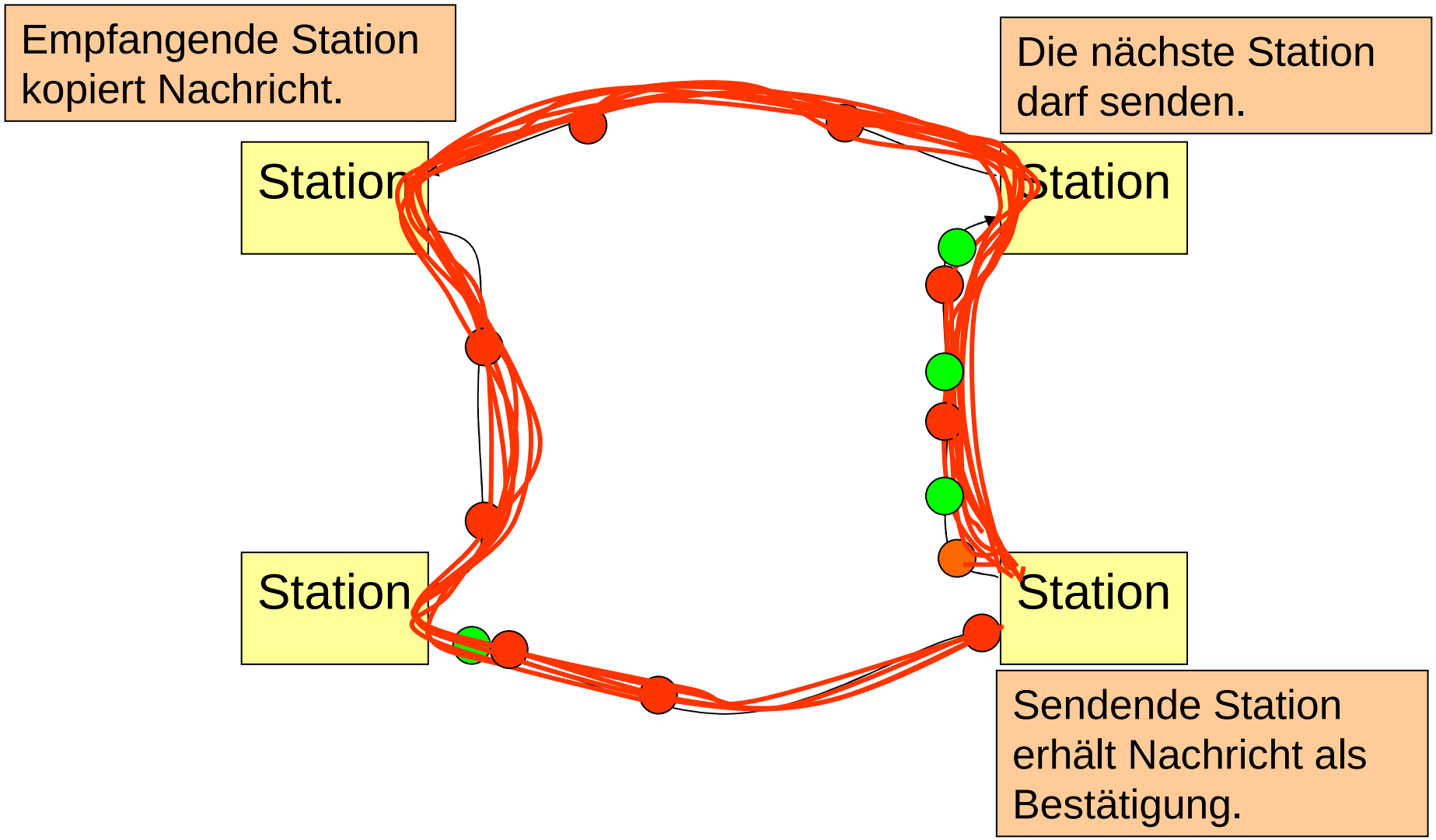
http://www.germes-online.com/direct/dbimage/50244126/UTP___FTP___SFTP_Cat_5__Cat_5e__and_Cat_6_Cable__Solid___Stranded_.jpg



100 Gbit-Ethernet

- 100 Gbit-Ethernet ist gegenwärtig in der Entwicklung
- Basis: Glasfaserverbindung

Token-Ring-Netze



Eigenschaften von *Token-Ring*-Netzen

Vorteile

- Auch bei starker Belastung bleibt bei *n* Stationen für jede Station 1/*n* der Übertragungsleistung erhalten.
  Kein ***thrashing***, d.h. keine Reduktion der effektiven Übertragungsrate bei Überlast
  (wie beim CSMA/CD-Verfahren)
- Auffrischung der Signale an jeder Station

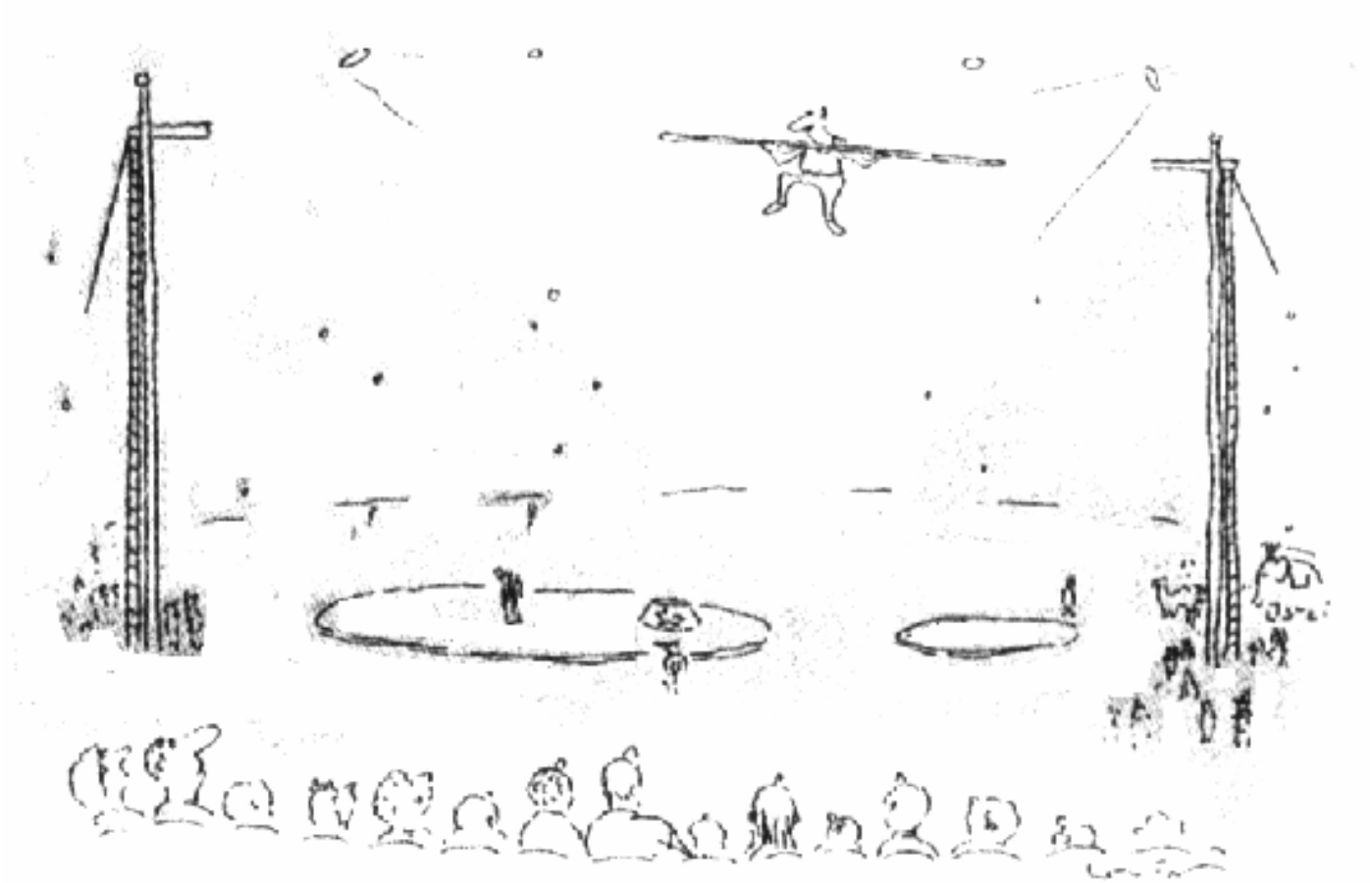
Nachteile

- Installation aufwendiger als beim *Ethernet*

Vorkommen

- IBM *Token Ring*, IEEE 802.5
- seit 1980 an Workstations

Wireless communication



"It appears to be some new kind of wireless technology."

WLAN

IEEE 802.11 a/b/g/n: Varianten der Netzwerk-Norm IEEE 802:

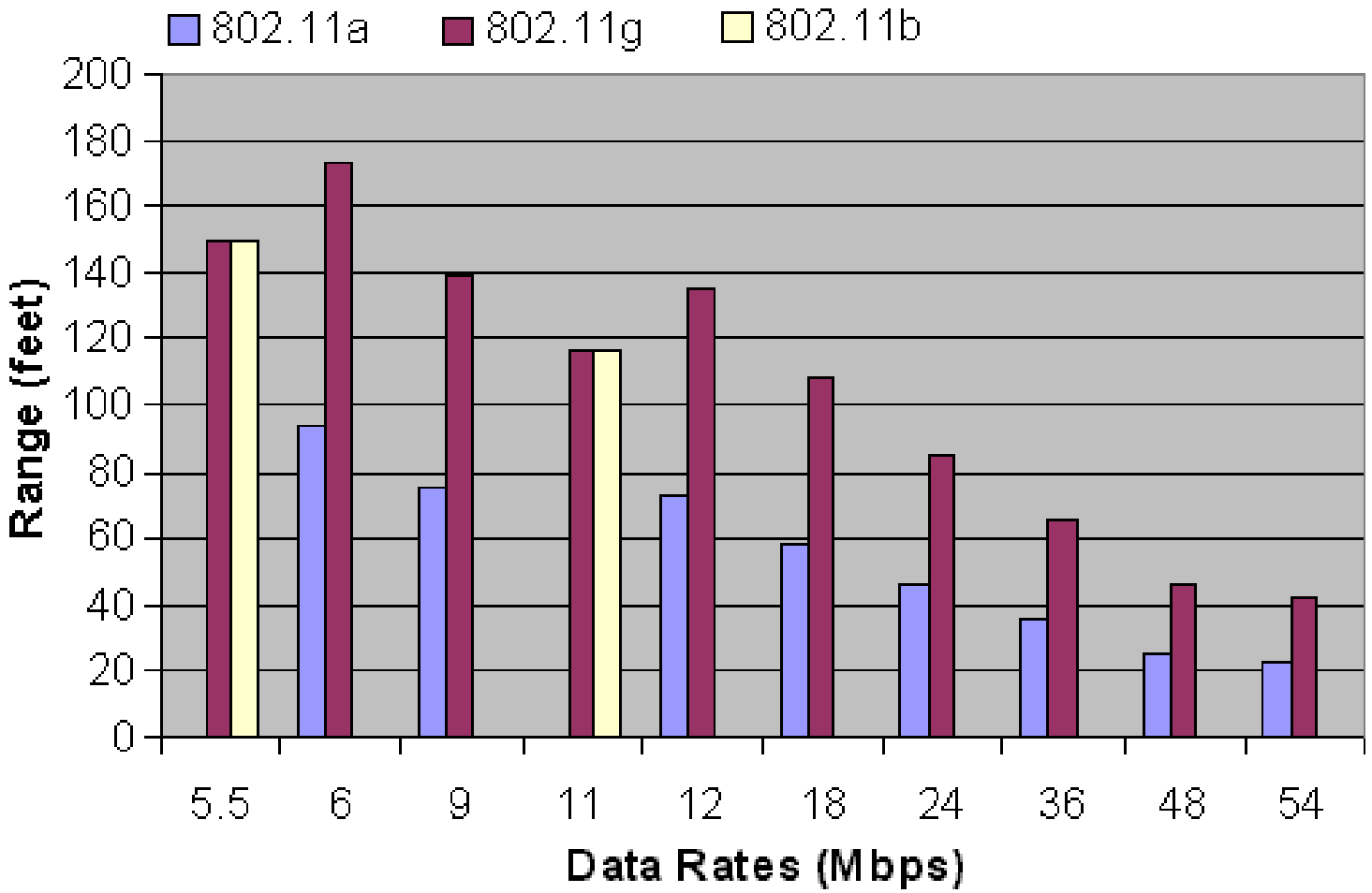
Datenraten:

- IEEE 802.11b: 1, 2, 5.5 oder 11 Mbit/s im 2,4 GHz-Band (meistens eingesetzte Karten)
- IEEE 802.11a: bis 54 Mbit/s im 5 GHz-Band
- IEEE 802.11g; bis 54 Mbit/s, auch im 2,4 GHz-Band, abwärtskompatibel mit 802.11a und b.
  „Super G“ (kein Standard): 108 Mbit/s.
- IEEE 802.11n (draft): s.u.

Funkkanäle störanfälliger als drahtgebundene Kommunikation
☞ aufgrund zusätzlicher Korrekturmaßnahmen in der Regel:
effektive Datenrate WLAN < Datenrate von 10BASE-T oder
effektive Datenrate WLAN << Datenrate von 100BASE-Tx.

Beziehung Entfernung/Datenrate

From Computer Desktop Encyclopedia
Reproduced with permission.
© 2002 Intersil Corporation.



Vergleich der IEEE-Standards

Protokoll	Veröffent-licht	Frequenz	Datenrate ("brutto")	Reich-weite (im Haus)	Reich-weite(im Freien)
802.11	1997	2,4 GHz	2 Mbit/s	~20 m	~100 m
802.11a	1999	5 GHz	54 Mbit/s	~35 m	~120 m
802.11b	1999	2,4 GHz	4,3 Mbit/s	~38 m	~140 m
802.11g	2003	2,4 GHz	54 Mbit/s	~38 m	~140 m
802.11n	Sep. 2009	2,4 GHz 5 GHz	248 Mbit/s	~70 m	~250 m

http://de.wikipedia.org/wiki/IEEE_802.11n

802.11n (1)

- Datenübertragungsraten von 100 – 600 Mbit/s
- Nutzung von 2-4 Antennen
  („*Multiple Input, Multiple Output*“ (MIMO))
- Ausnutzung der Richtwirkung mehrerer Antennen
- *Beamforming* (Sender kann bestimmten Empfänger anstrahlen, dadurch Gewinn an Übertragungskapazität)
- Nutzung von bis 2x20=40 Mhz breiten Frequenzbändern (anstelle von 20 Mhz)
- Anderes Modulationsverfahren als bisher
- Komplexere Steuerung der Geschwindigkeitsanpassung
- Stromsparfunktion für mobile Geräte

802.11n (2)

Gründe für die verzögerte Verabschiedung

- Starke Konkurrenzsituation im 2,4 GHz-Band (insgesamt nur 70 Mhz)
  ☞ komplexe Wahl des zweiten 20 MHz-Bandes
- Vergrößerung der Paketgröße zur Reduktion des *Overheads*
- *Backbones* können nicht mehr mit 100 Mbit-Netzwerken betrieben werden.
- Reduktion des Datenverkehrs durch dezentrale Organisation erforderlich
- Ältere *Switches* unterstützen kein "*power over* Ethernet" in Gigabitnetzwerken

http://wiki.computerwoche.de/doku.php/comm/ieee_802.11n

Bluetooth (1)

Benannt nach Harald „Blauzahn“,
dänischer König von 940-985, der das Land einigte.

Datenrate:

„synchron“	64kbit/s	für Sprachanwendung
„asynchron symmetrisch“	432,6 kbit/s	(Netzwerkanwendung)
„asynchron asymmetrisch“	721/57,6 kbit/s	*downloads*

Reichweiten:

„Pico bluetooth“	10cm..10m	1mW Sendeleistung
„Mega bluetooth“	>100m	100mW Sendeleistung

Frequenzband: 2402-2480 MHz, lizenzfrei für industrielle, medizinische und wissenschaftliche Zwecke bereitgestellt; gemeinsam genutzt mit IEEE 802.11 und Mikrowellenöfen.

Bluetooth (2)

Kanäle:

79, je 1 MHz; schneller Wechsel (nach jeder Nachricht) zwischen Kanälen, um die Wahrscheinlichkeit von Störungen zu reduzieren.

Vollduplex-Übertragung

Spezielle Stromsparmodi

optionale Verschlüsselung

optionale Authentifizierung

Verschiedene Profile ermöglichen Interoperabilität

Hardware relativ preisgünstig (Einbau z.B. in Köpfhörer)

Weitere Informationen:

http://www.bluetooth.com, c‘t 2/2003

Mobilfunk-basierte Techniken

- GSM: Modem begrenzt auf 9,6 kbit/s
- GPRS (General Packet Radio Service):
  Paket-orientierte Nachrichtenübertragung in 2G und 3G-Systemen ☞ „2,5G GSM“
  Datenraten von 56 bis 114 kbit/s
- Edge (*Enhanced data rates for* GSM Evolution),
  „2,75 G“; Enhanced GPRS: max. of 473,6 kbit/s
- *High Speed Downlink Packet Access* (HSDPA) “3.5 G”;
  UMTS-Erweiterung; max. 14 Mbit/s
  (in der Praxis 7,2 Mbit/s)
- LTE (Long Term Evolution)

Alternativen

- **WLAN**
- Datenkommunikation über **DECT**
- **HIPERLAN:**
  vom ETSI entwickelter Standard f. drahtlose Netze:
  - Typ 1: 24 Mbit/s im 5 GHz-Band
  - Typ 2: Wireless ATM mit 20 Mbit/s, 5 GHz-Band
- **WAN**
- **ATM (Festnetz)**
- **GSM, UMTS, HSDPA etc.**

Zusammenfassung

Datenübertragung lokale Peripherie & Netzwerke

- Serielle Schnittstellen
- USB
- Firewire
- SCSI
- (e)SATA
- Ethernet
- Drahtlose Techniken (Bluetooth, WLAN, …)