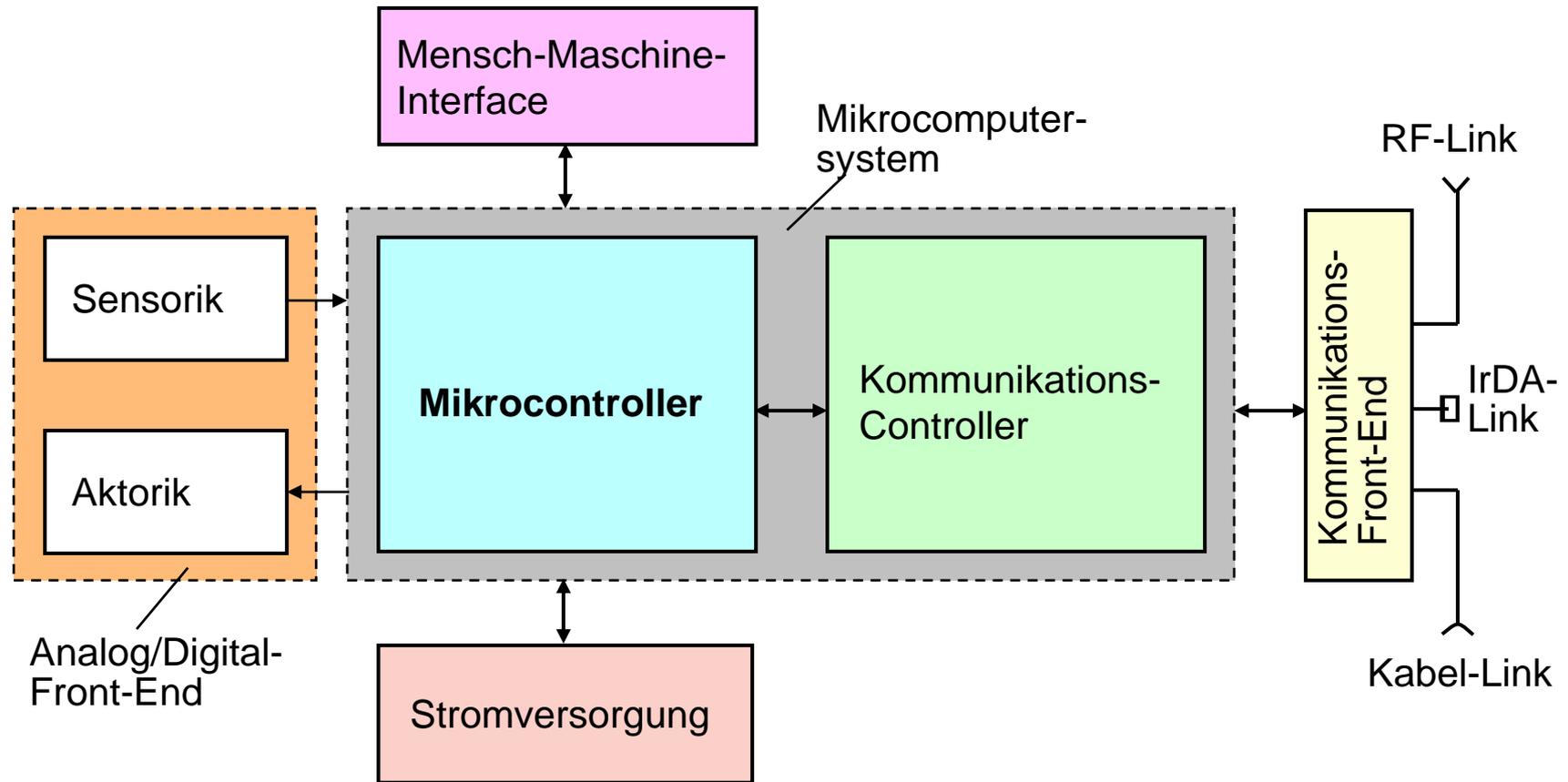


Kapitel 3

Rechnerarchitekturen für Echtzeitsysteme

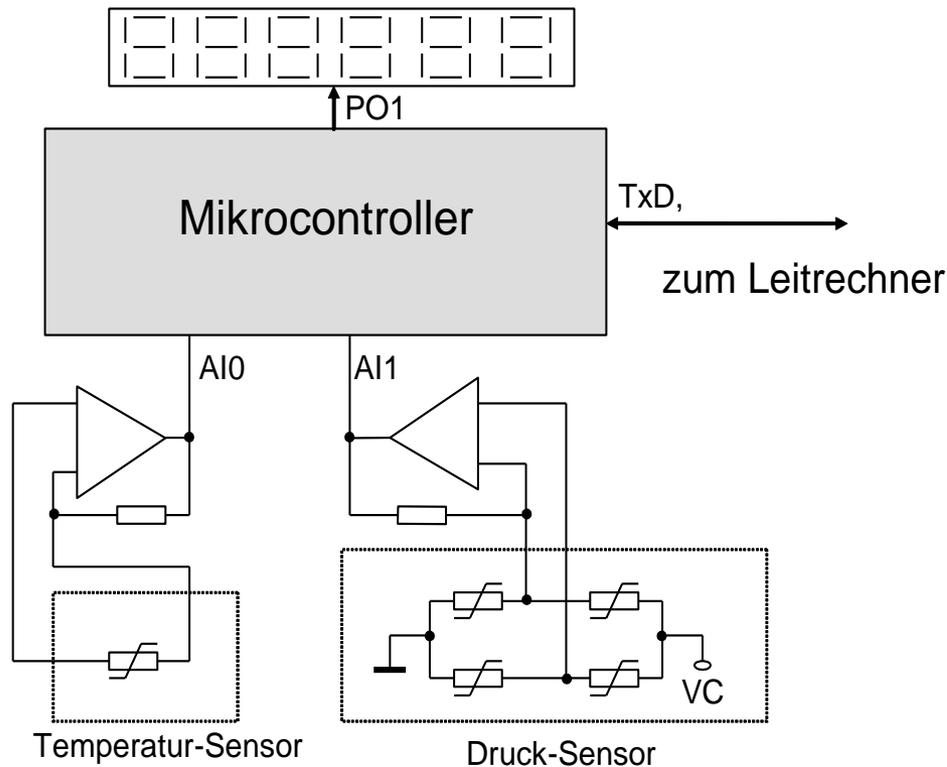
Eingebettete Echtzeit-Systeme (1)

Mobiles Gerät mit Schnittstellen



Intelligente Schnittstelle zwischen Mikrocontroller und WLAN

Temperaturkompensierte Druckmessung

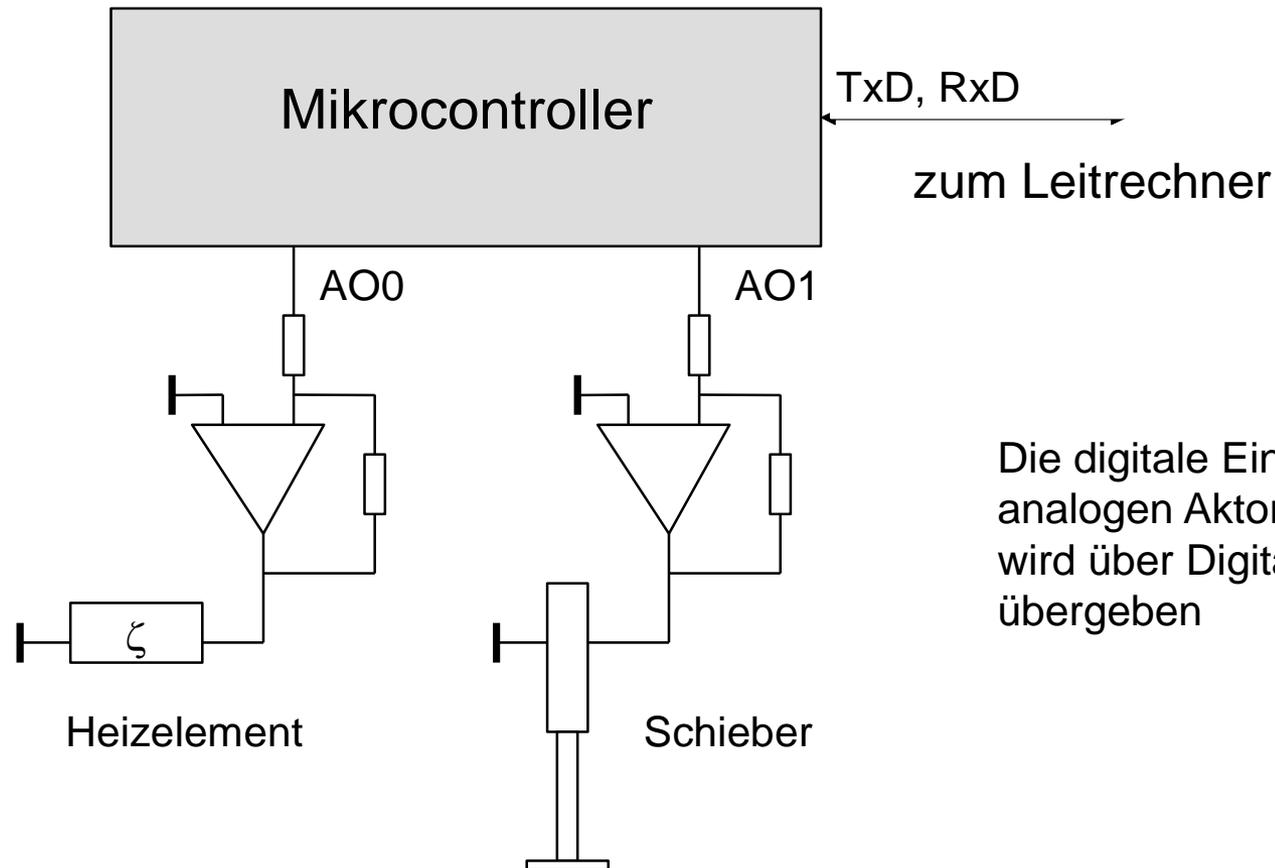


Prozesssteuerung:

- Messen
- Stellen
- Regeln

Kompressionsverfahren
dient der Reduktion des
Datenvolumens

Ansteuerung zweier analoger Aktoren



Die digitale Eingangsgröße an die analogen Aktoren (z.B. Heizelement) wird über Digital-/Analogwandlern übergeben

Ein Regelkreis mit einem Mikrocontroller

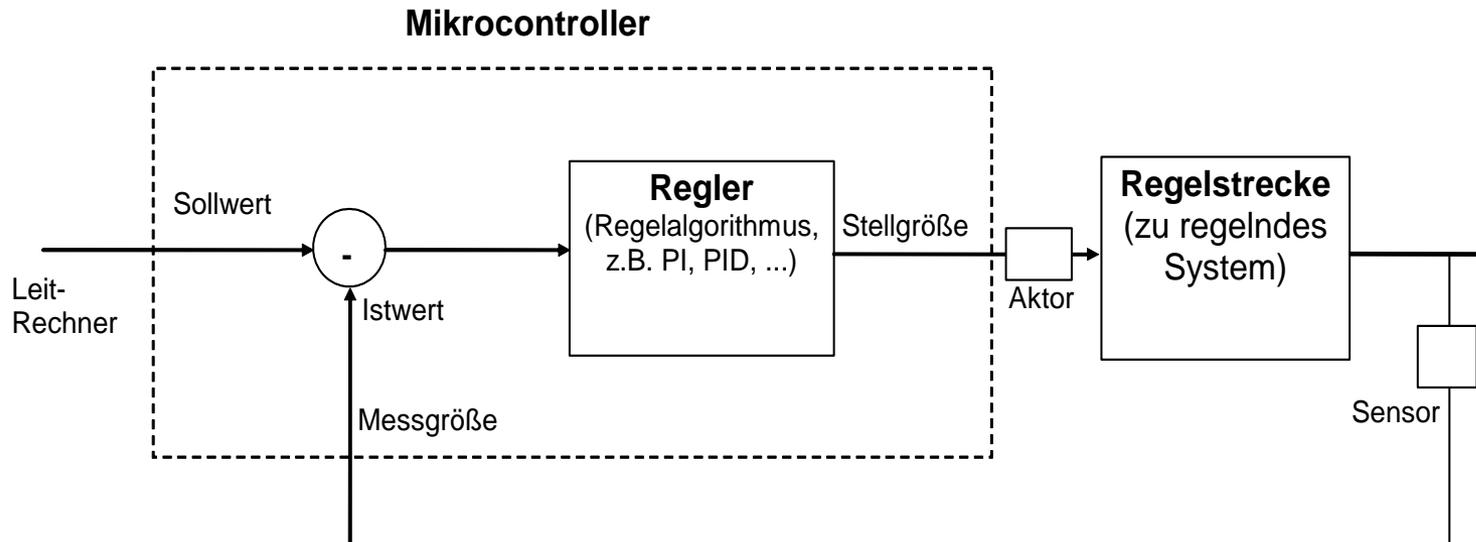
Die Teilschritten des **Regelns**:

- Messen des Istwertes
- Soll-/Istwert-Vergleich
- Ermittlung der Stellgröße

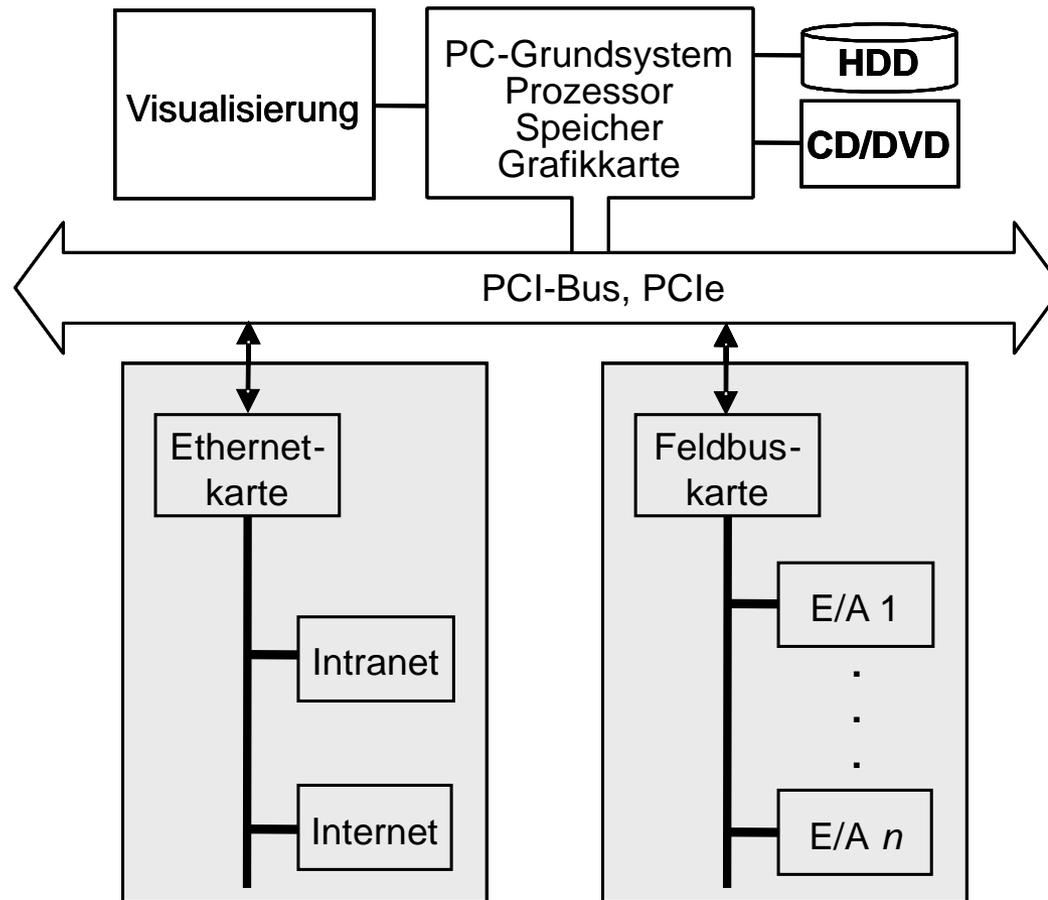
Aufgabe des **Regelalgorithmus**:

die Regelabweichung zu minimieren (den Istwert nah an den Sollwert heranzuführen)

Abhängig vom Regelalgorithmus werden Mikrocontroller mit verschiedener Leistung benötigt.



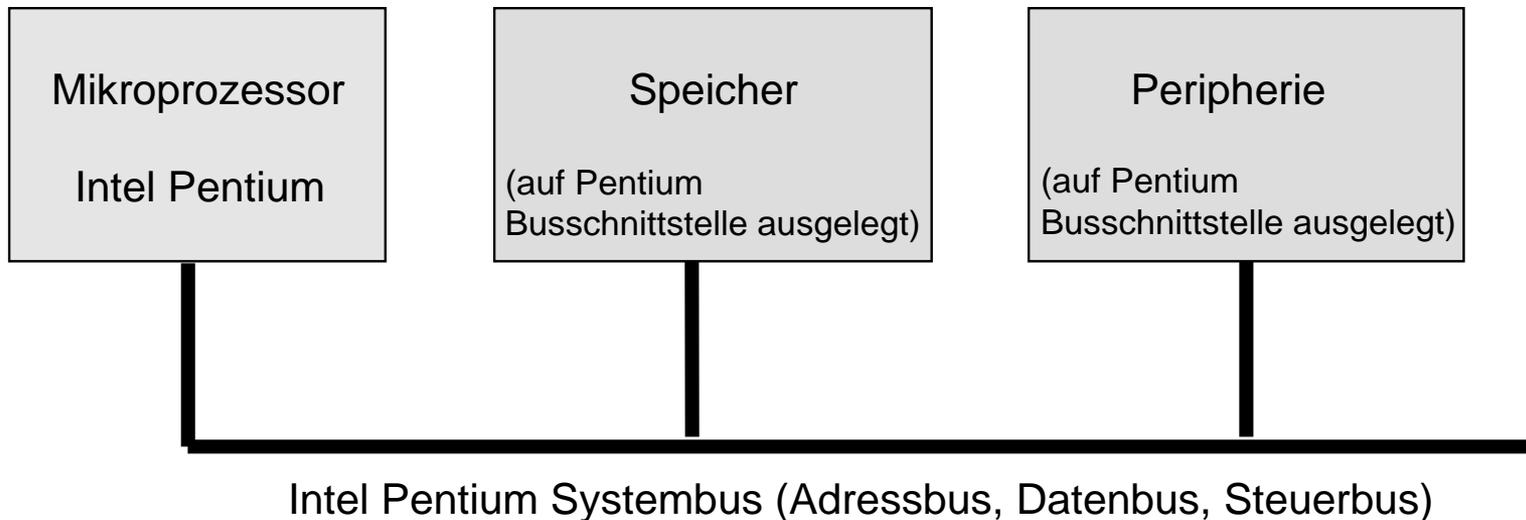
PC-basierte Echtzeitsysteme



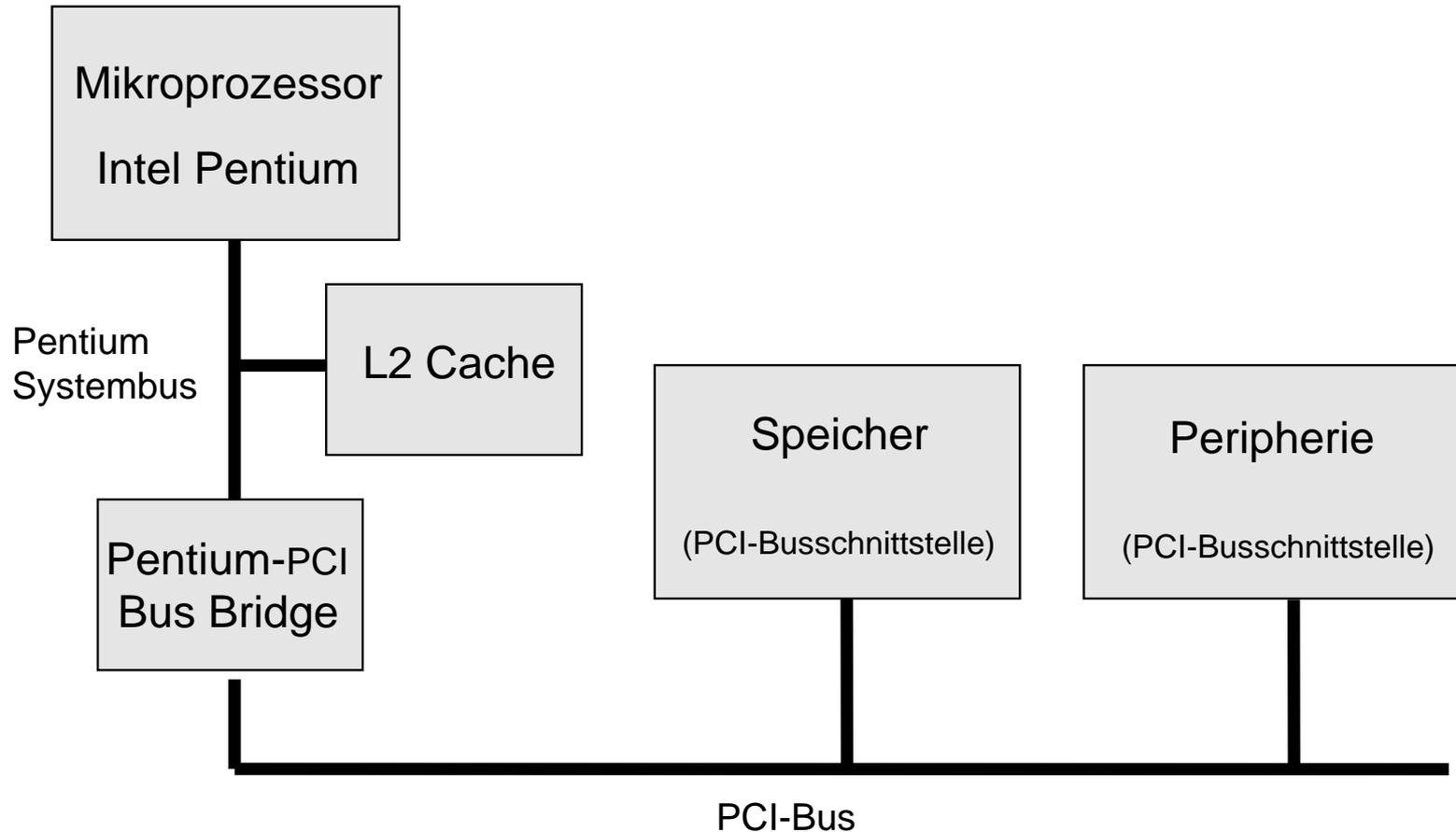
Systembusse für die Vernetzung innerhalb von PC-basierten Systemen

- Allgemeine Merkmale von Systembussen
- Paralleler PCI-Bus
- Serieller PCIe-Bus

Beispiel eines prozessorabhängigen Systembusses



Ein prozessorunabhängiger Systembus



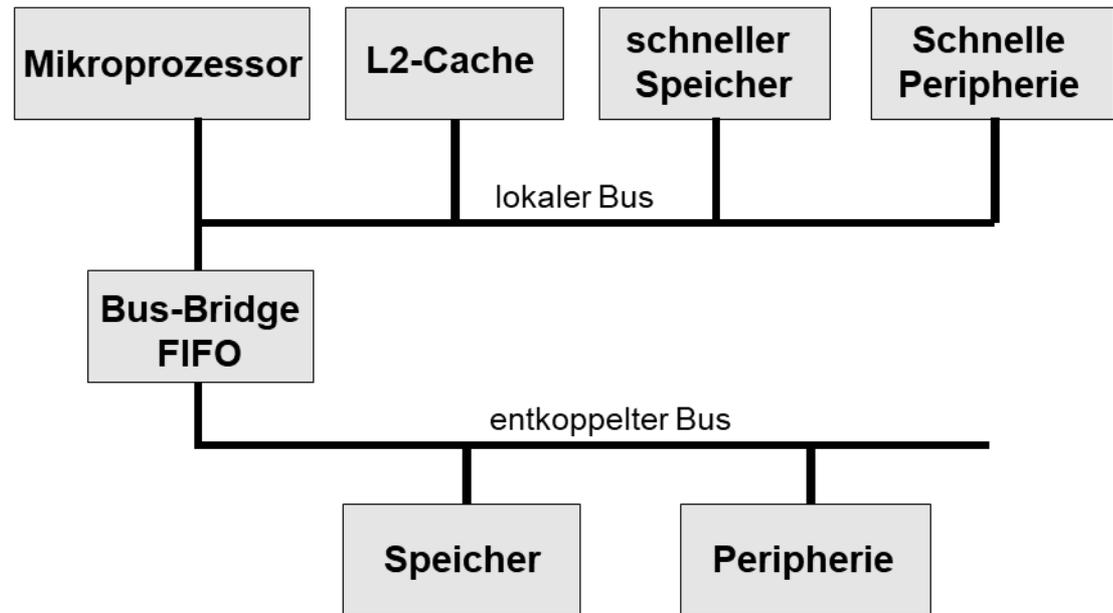
- Vorteile:* prozessorunabhängiger Bus \Rightarrow prozessorunabhängige Komponenten, insbesondere für modulare Steckkartensysteme
- Nachteile:* erhöhter Hardwareaufwand
- Anwendung:* modulare Automatisierungsrechner

Gepufferter, entkoppelter Bus

Erweiterung des Konzepts vom prozessorunabhängigen Bus

Bei einem zentralen Bus können langsame Geräte den Bus blockieren

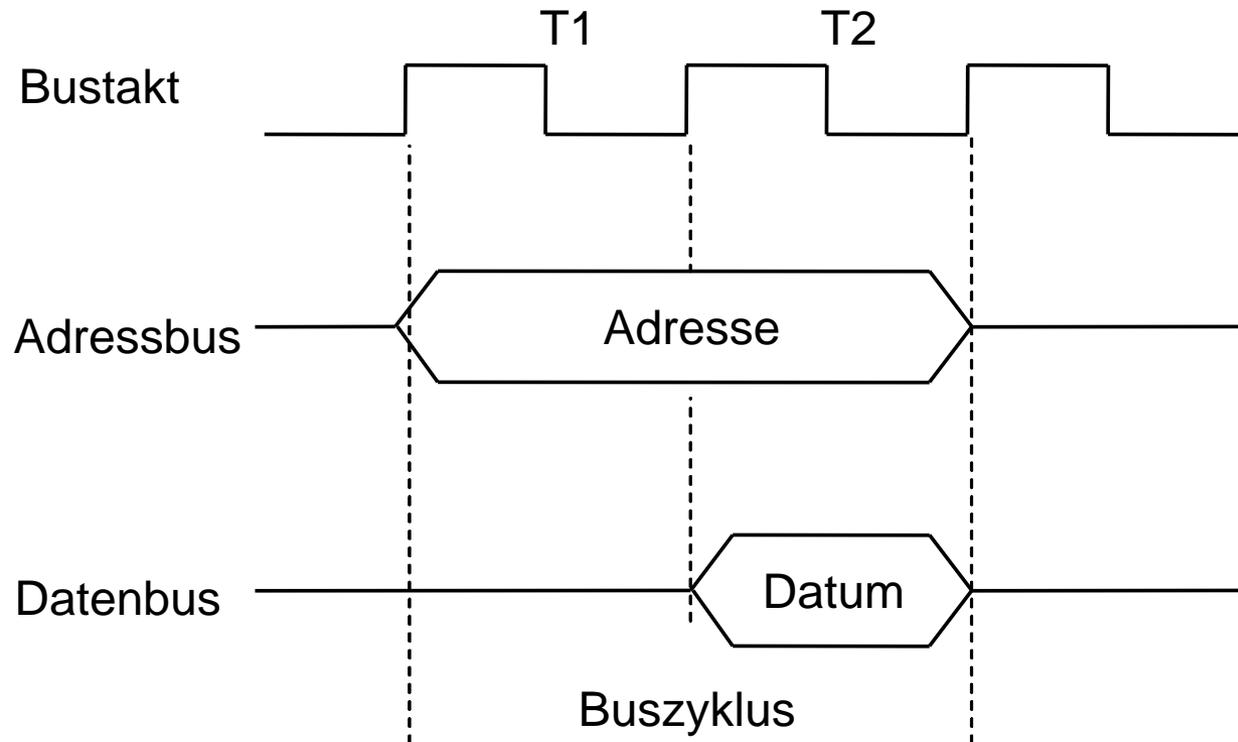
⇒ **Echtzeitbedingungen können verletzt werden!**



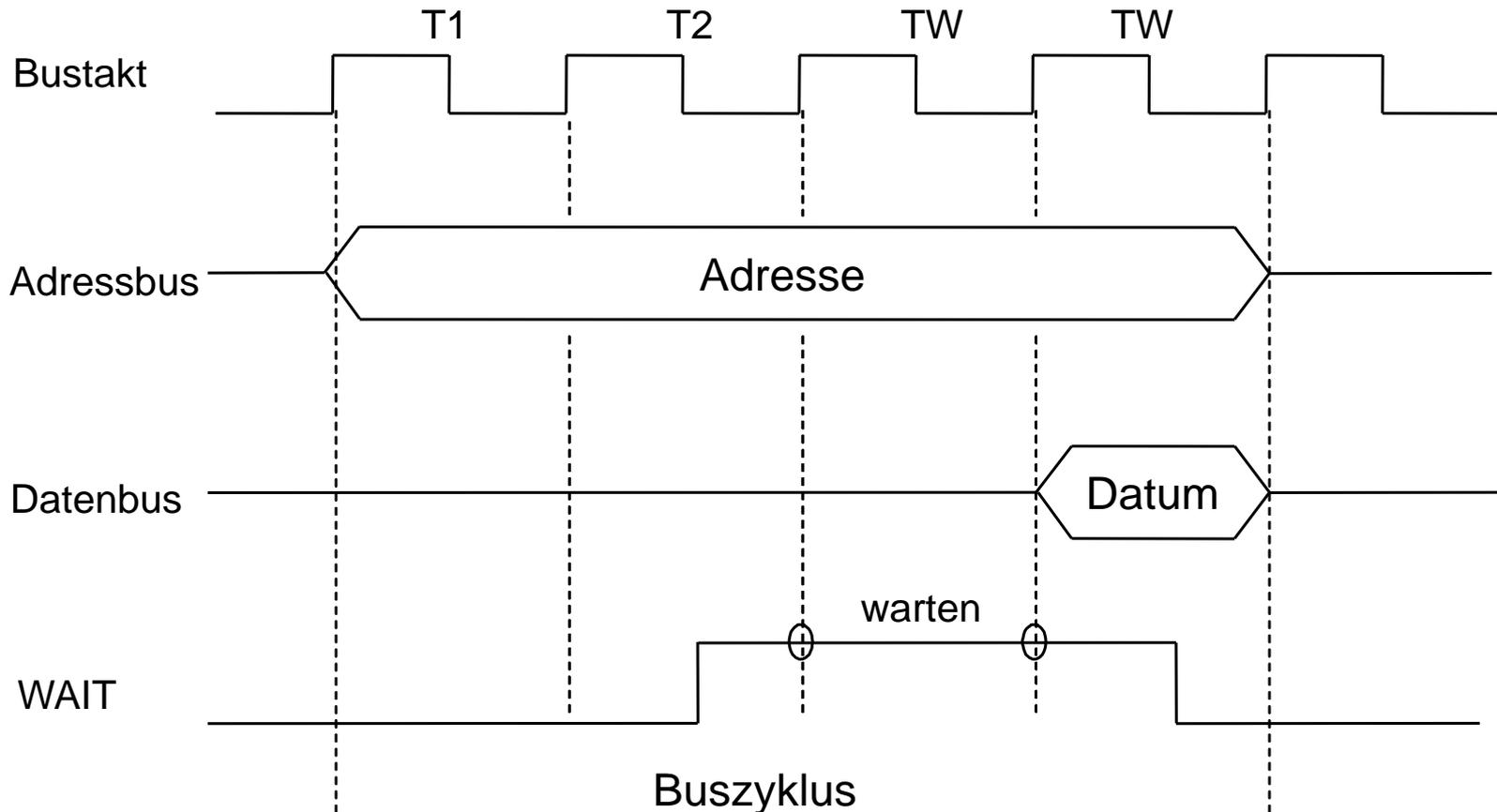
Entkoppelte Bussysteme können das Problem unterschiedlich schneller Module besser lösen

CPU und Bus werden hierbei durch einen FIFO-Speicher entkoppelt

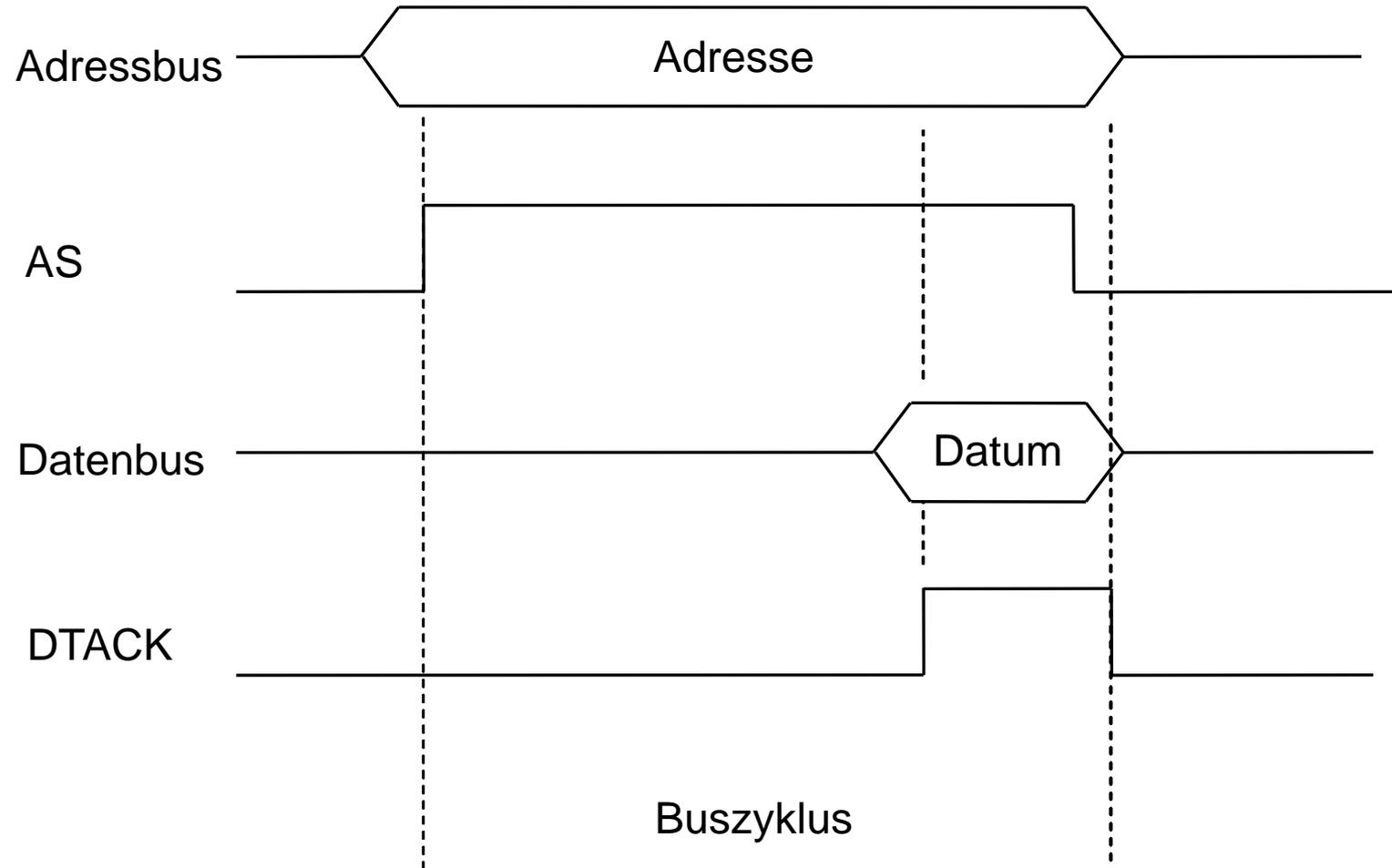
Synchroner Systembus



Synchroner Systembus mit Wartezyklen



Asynchroner Systembus



Merkmale Systembusse

Merkmal	PC-Busse				Workstation-Busse		Backplane-Busse		
	ISA-Bus	MCA	EISA-Bus	VL-Bus	SBus	MBus	VME-Bus	PCI-Bus	PCIe-Bus V3.0
Adressbus (Bits)	24	32	32	32	28	64	32	32/64	1*
Datenbus (Bits)	16	32	32	32	32		32		
max. Datentransportbreite (Bits)	16	32	32	32/64	64	64	32/64	32/64	1*
Bustaktfrequenz (MHz)	8	10	8.33	40	25	40	10	66	4000
max. Übertragungsrates (Mbyte/s)	8	20	33	80/160	200	320	40/80	266/533	985 / Lane*
Synchronisationsart	syn.	asyn.	syn.	syn.	syn.	syn.	asyn.	syn.	syn.

* - seriell, 1-16 zusammengehörige Lanes

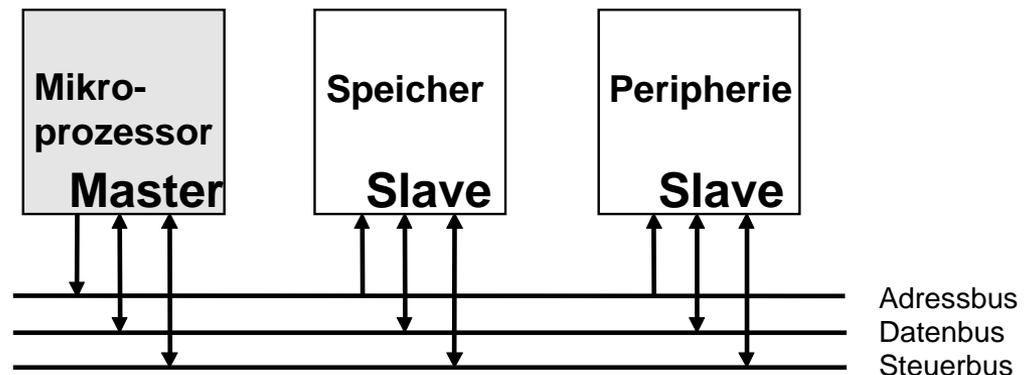
Beim Zugriff auf den Systembus sind zwei Arten zu unterscheiden: Zugriff als

- **Master:** aktiver Zugriff auf den Systembus, Auslösung von Buszyklen, Adressierung von Komponenten, ...
- **Slave:** passiver Zugriff, Reagieren auf einen Buszyklus, Lesen von Adressen, Datentransfer auf Anforderung, ...

Existiert nur ein Busmaster, sind die Verhältnisse einfach:

Der Master 'besitzt' ständig den Bus, das Zeitverhalten des Gesamtsystems ist leicht vorhersagbar

Beispiel:



Mehrere Busmaster

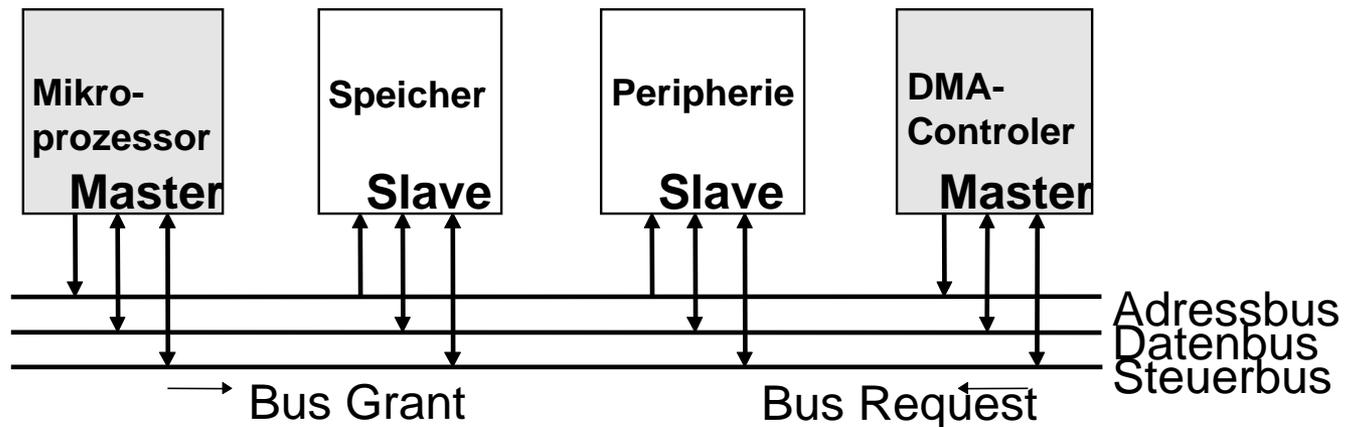
In einem komplexeren Mikrorechnersystem kann es jedoch mehrere Master geben

Beispiele: • mehrere Mikroprozessoren im Rechnersystem

- DMA-Transfer zwischen Speicher und/oder Peripherie

In diesem Fall darf immer nur ein Master gleichzeitig am Bus aktiv sein

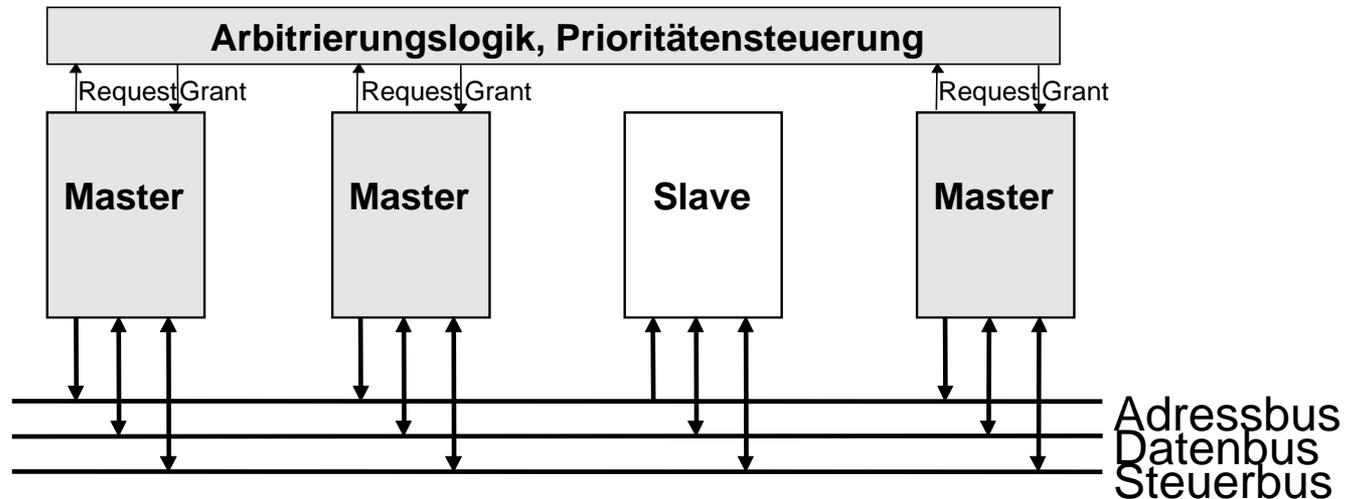
⇒ Es muss ein Verfahren zur Zuteilung des Systembusses an jeweils einen Master geben: **Buszuteilung, Bus-Arbitration**



In einem einfachen Fall mit nur einem Mikroprozessor und einem DMA-Controller als Busmaster kann diese Zuteilung durch den Prozessor geschehen

Externer Bus – Arbitrer

In komplexeren Systemen mit mehreren Bus-Mastern und ggf. gleichzeitigen Anforderungen ist ein **externer Bus-Arbitrer** erforderlich, der als Schiedsrichter über die Bus-Zuteilung wacht



Möchte eine Komponente aktiver Bus-Master werden, so fordert sie den Bus beim externen Bus-Arbitrer an (bus request).

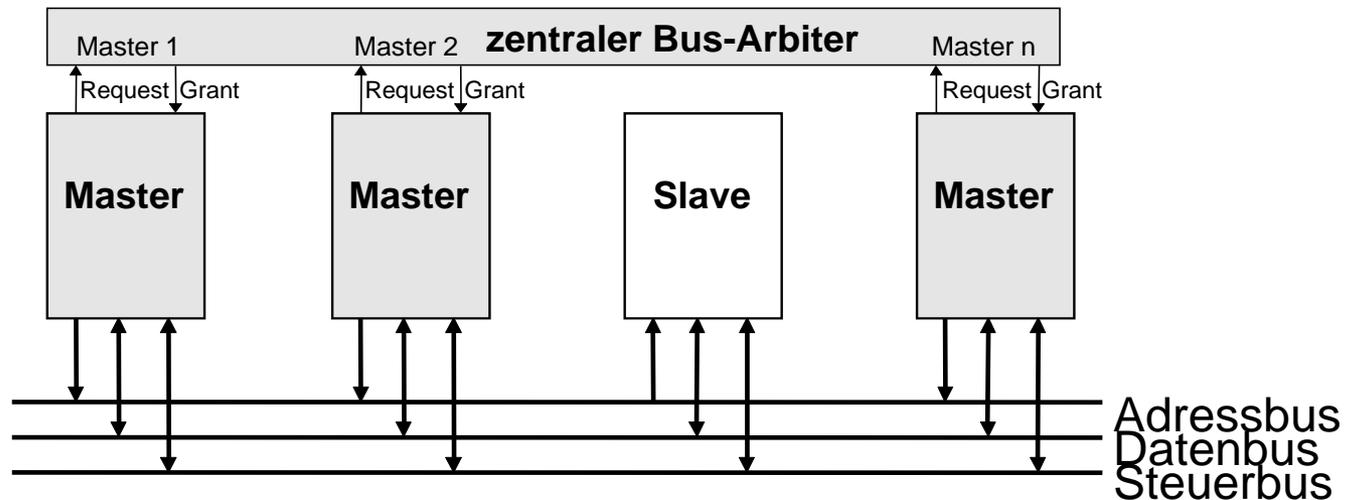
Sobald die Komponente an der Reihe ist, erhält sie die Zuteilung von diesem Arbitrer (bus grant)

Zentraler Bus – Arbitrer

Ein externer Bus-Arbiter kann nach verschiedenen Prinzipien (ähnlich der Interrupt-Verarbeitung, Daisy-Chain/Controller) aufgebaut werden:

Zentraler Bus-Arbiter

Die Bus-Zuteilung und Prioritäten-Steuerung erfolgt über einen zentralen Bus-Arbiter



Vorteile: einfacher Systemaufbau, beliebige und schnelle Priorisierung

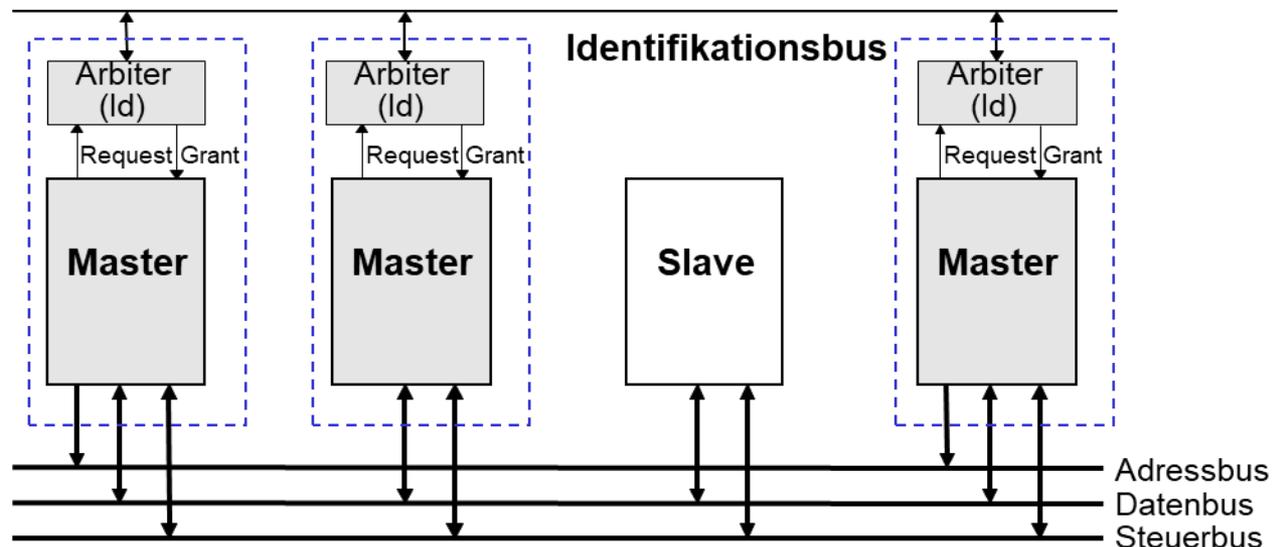
Nachteile: hoher Leitungsaufwand

Identifikationsbus

Daneben existieren auch dezentrale Verfahren, die nicht mit einer Daisy-Chain sondern einem Identifikationsbus arbeiten.

Hierbei enthält jeder lokale Arbiter eine ID-Nummer, die seine Priorität bestimmt.

Bei Bus-Anforderung geben alle anfordernden Arbiter ihre ID-Nummer auf den Identifikationsbus, in einem iterativen Verfahren wird hieraus die höchste Nummer (= höchste Priorität) ausgewählt



Dieses Verfahren wird z.B. bei Multibus II benutzt

Zeitliche Vorhersagbarkeit des Busses für Echtzeitsysteme von zentraler Bedeutung

Einfaches System, synchroner Bus, keine Wartezyklen:
Vollständig deterministisches Zeitverhalten

Einfaches System, synchroner Bus, Wartezyklen:
Ebenfalls deterministisch, wenn Wartezyklen bekannt.

Bei variablen Wartezyklen Obergrenze, Abbruch durch Busmaster wenn überschritten

Mehrere Busmaster:

das Echtzeitverhalten wird vom Verhalten des Busses im Konfliktfall bestimmt. Erforderliche Maßnahmen zur Wahrung von Echtzeitfähigkeit:

- Prioritäten
- Preemption
- Unterbrechbarkeit von Blocktransfers
(Vermeidung von Prioritäteninversion)
- Busmonitor

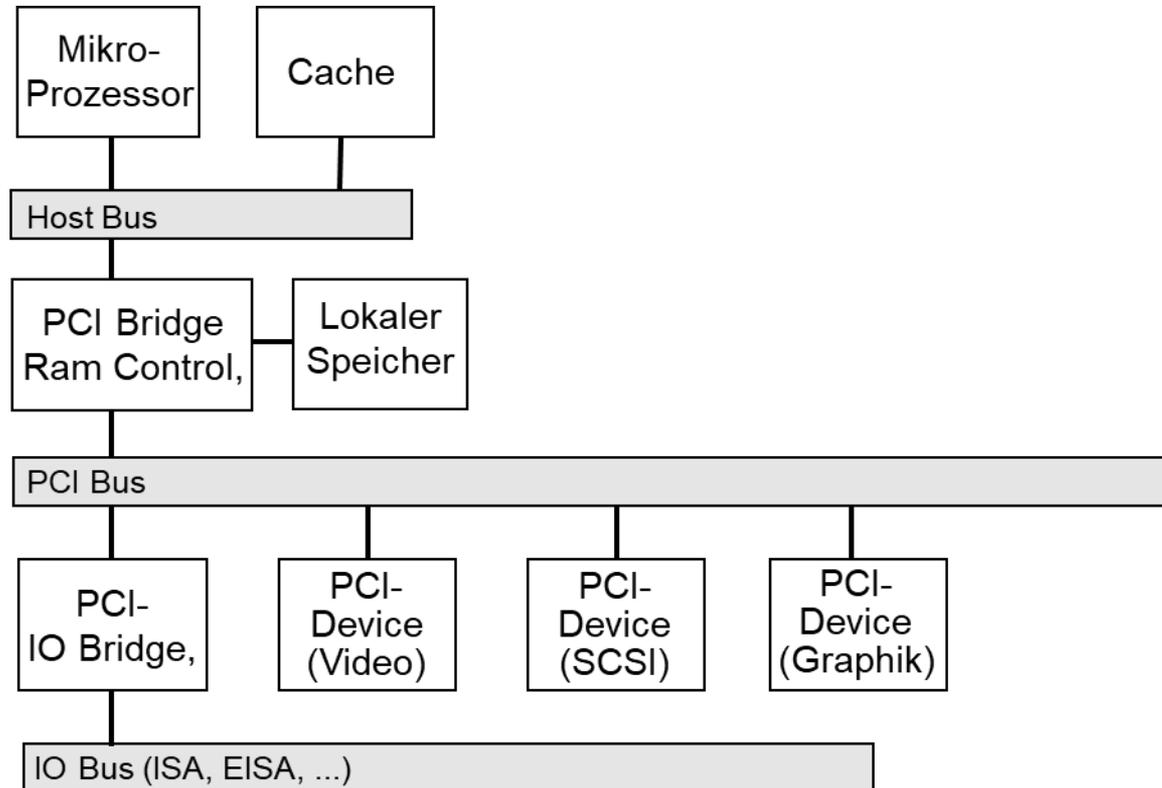
Beispiel 1: Der PCI – Bus

PCI-Bus: *Peripheral Component Interconnect* Bus (Entkoppelter, gepufferter, prozessorunabhängiger 32/64-Bit Bus)

Eigenschaften des PCI-Bus:

- Kommandoorientiert (Klassifizierung der Bus-Transfers über Kommandos, nicht Einzel-Signale)
- Synchron, Taktfrequenz 33 - 66 MHz
- Multiplex-Bus (Adressen/Daten)
- Je nach Ausbaustufe 32 Bit Adressen/Daten bzw. 64 Bit Adressen/Daten
- Adressbereiche per Software konfigurierbar
- Erlaubt Burst-Transfers
- Fehlererkennend (Parity-Fehlererkennung, Retry-Option, Timeouts, fatale Fehler können über ein Bussignal signalisiert werden)
- Multi-Master fähig (Bus-Zuteilungsprotokolle, Lock-Zyklen, Cache-Kohärenz, zentraler Arbiter)
- Echtzeitfähig (Prioritäten, zeitlich begrenzbare Busbelegung)
- Bridge-Konzept

Grundlegende PCI-Bus Architektur (Bridge-Konzept)



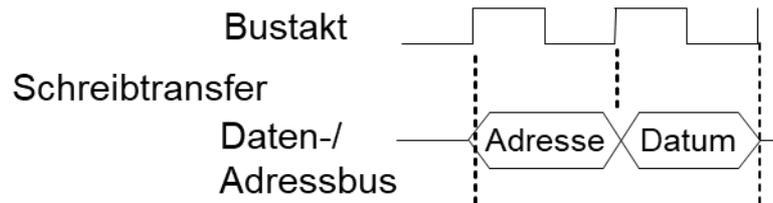
PCI-Bus Bridge wird wie ein normales PCI-Device behandelt
⇒ PCI-Bussysteme können hierarchisch kaskadiert werden

PCI-Bus Glossar

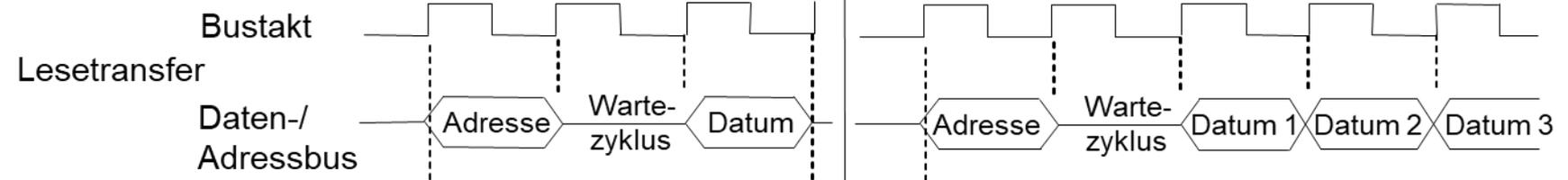
Transaktion	bezeichnet einen allgemeinen Datentransfer über den PCI-Bus, bestehend aus einer Adress- und einer oder mehrerer Datenphasen (1 Taktzyklus pro Phase)
Agent	bezeichnet jeden Busteilnehmer
Initiator	ist ein Agent, der eine Transaktion durchführt und steuert, ein Master
Target	ist ein Agent, der auf eine Transaktion antwortet, ein Slave
Arbiter	ist der zentrale Bus-Arbiter des PCI-Bus
Bridge	ist der Verbindungsbaustein zwischen PCI-Bus und einem weiteren PCI-Bus oder anderem Bussystem

PCI-Bus Transferarten

a) Standardtransfer



b) Bursttransfer



Anzahl benötigter Taktzyklen bei verschiedenen PCI-Transfers

Anzahl Taktzyklen	4 Worte	128 Worte	2^{30} Worte
Standard Lesen	12	384	$2^{31} + 2^{30}$
Standard Schreiben	8	256	2^{31}
Burst Lesen	6	130	$2^{30} + 2$
Burst Schreiben	5	129	$2^{30} + 1$

Übertragungsrate = Anzahl Bytes / (Taktzykluszeit * Anzahl Taktzyklen)

Resultierende Datenübertragungsraten bei einem 64-Bit 66MHz PCI-Bus

MBytes/sec	4 Worte	128 Worte	2^{30} Worte
Standard Lesen	177,78	177,78	177,78
Standard Schreiben	266,67	266,67	266,67
Burst Lesen	355,56	525,13	533,33
Burst Schreiben	426,67	529,20	533,33

Die wichtigsten Signale des PCI-Bus

Signal	Richtung	Beschreibung
AD[31:0]	bi	Adress/Datenbus, gemeinsame Leitungen zur Übertragung von Daten und Adressen
C/BE#[3:0]	out	Command/ByteEnable, identifiziert Transfertype (während der Adressphase) oder gibt Datenbytes frei (während der Datenphase)
PAR	bi	Parität, gerade Parität über AD[31:0] und C/BE#[3:0]
FRAME#	out	Frame, kennzeichnet Start und Länge eines Transfers
TRDY#	in	Target Ready, kennzeichnet die Bereitschaft des Target, den aktuellen Datentransfer abzuschließen
IRDY#	out	Initiator Ready, kennzeichnet die Bereitschaft des Initiators, den aktuellen Datentransfer abzuschließen
STOP#	in	Stop, durch Aktivieren dieses Signals kann ein Target einen vom Initiator gestarteten Datentransfer stoppen
DEVSEL#	in	Device Select, bestätigt die Adressdekodierung durch ein Target
IDSEL	in	Initialization Device Select, selektiert ein Device während der Konfigurationsphase
PERR#	bi	Parity Error, signalisiert das Auftreten eines Paritätsfehlers auf AD[31:0] oder C/BE#[3:0]
SERR#	bi	System-Error, signalisiert das Auftreten eines katastrophalen Fehlers (z.B. Paritätsfehler während der Adressierungsphase)
REQ#	out	Request, Bus-Anforderung an den zentralen Arbitrer durch einen Initiator
GNT#	in	Grant, Bus-Gewährung durch den Arbitrer
CLK	in	PCI System Takt, 0 .. 66MHz
RST#	in	System Reset, rücksetzen aller PCI-Devices

Arbitrierung und Echtzeitverhalten

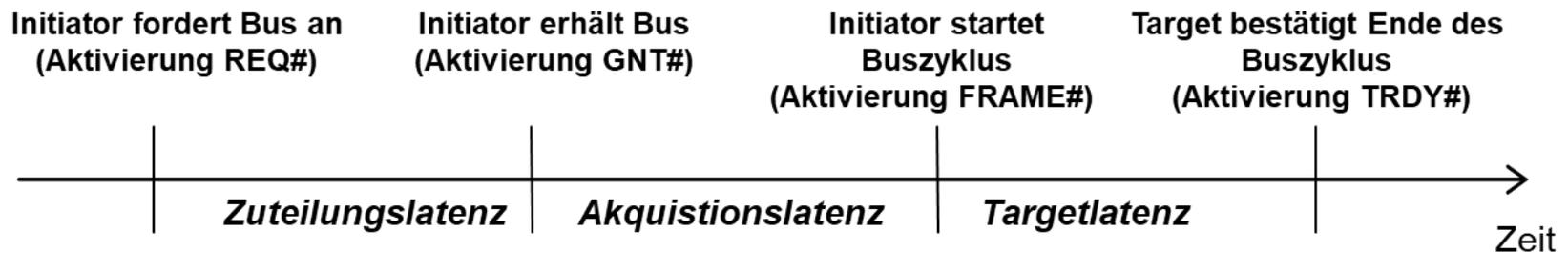
Der PCI-Bus verfügt über einen zentralen Arbiter auf dem Mainboard

Dieser erlaubt anwenderdefinierbare Prioritätsschema

Für jede PCI-Transaktion muss eine Buszuteilung beantragt werden

Dies geschieht über die Signale REQ# und GNT#

PCI-Transaktion mit Buszuteilung:

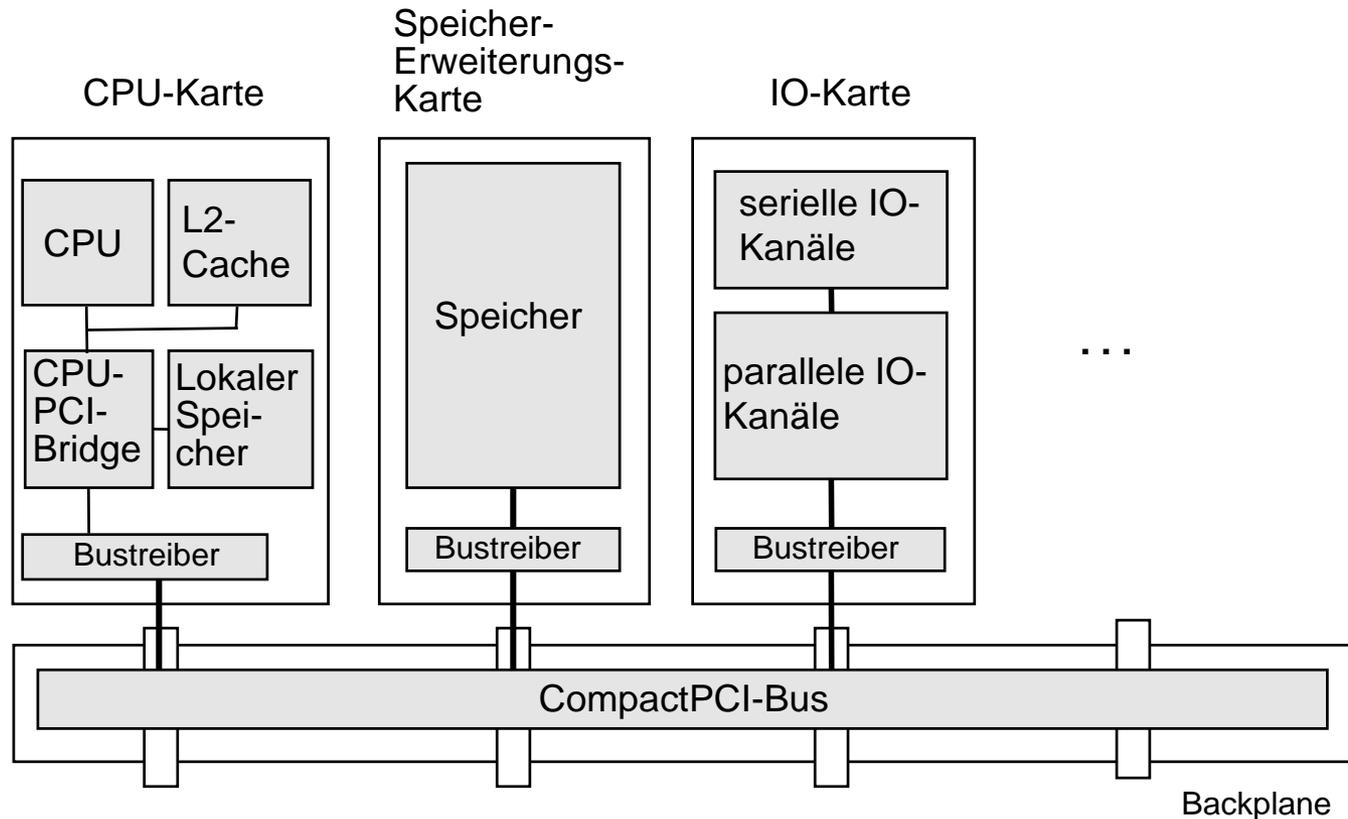


Will ein Master mehrere Transaktionen ohne neue Zuteilung ausführen, muss er REQ# aktiv lassen (*Back-to-Back-Transfers*)

Arbitrierungsregeln zur Gewährleistung des Echtzeitverhaltens:

- Datentransfer durch Initiator, solange das Target Daten aufnehmen kann und kein anderer Initiator den Bus beansprucht
- Latenzzähler: Sobald eine vorgebbare Anzahl von Taktzyklen abgelaufen ist, muss ein Initiator im Konfliktfall den Bus freigeben => garantierter Zeitschlitz
- Target muss Datentransfer abbrechen, wenn es nicht innerhalb von 8 Taktzyklen das nächste Datum bereitstellen kann => Schachtelung langsamer und schneller Transfers

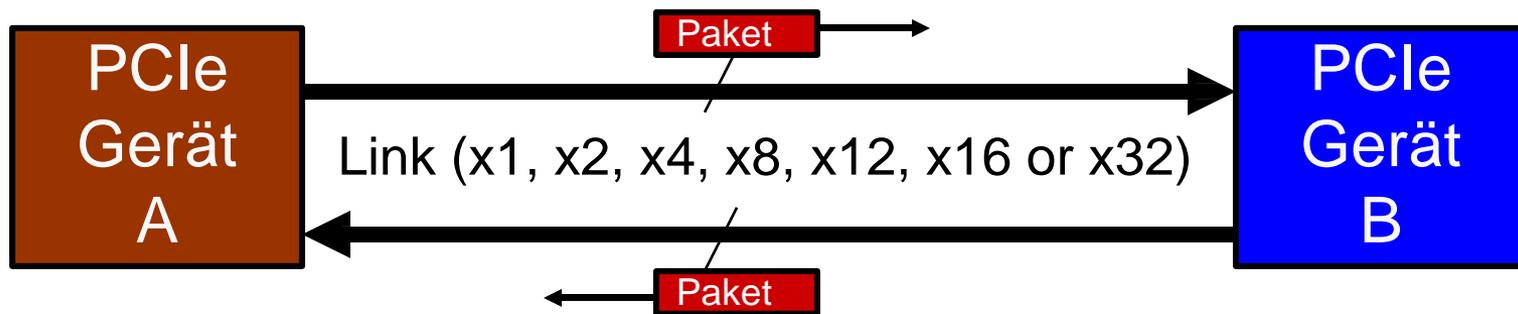
PC's in Backplane-Technik: Industrie-PC's



Vorteil: Software kann auf einem normalen PC entwickelt werden und direkt auf den Automatisierungsrechner übernommen werden

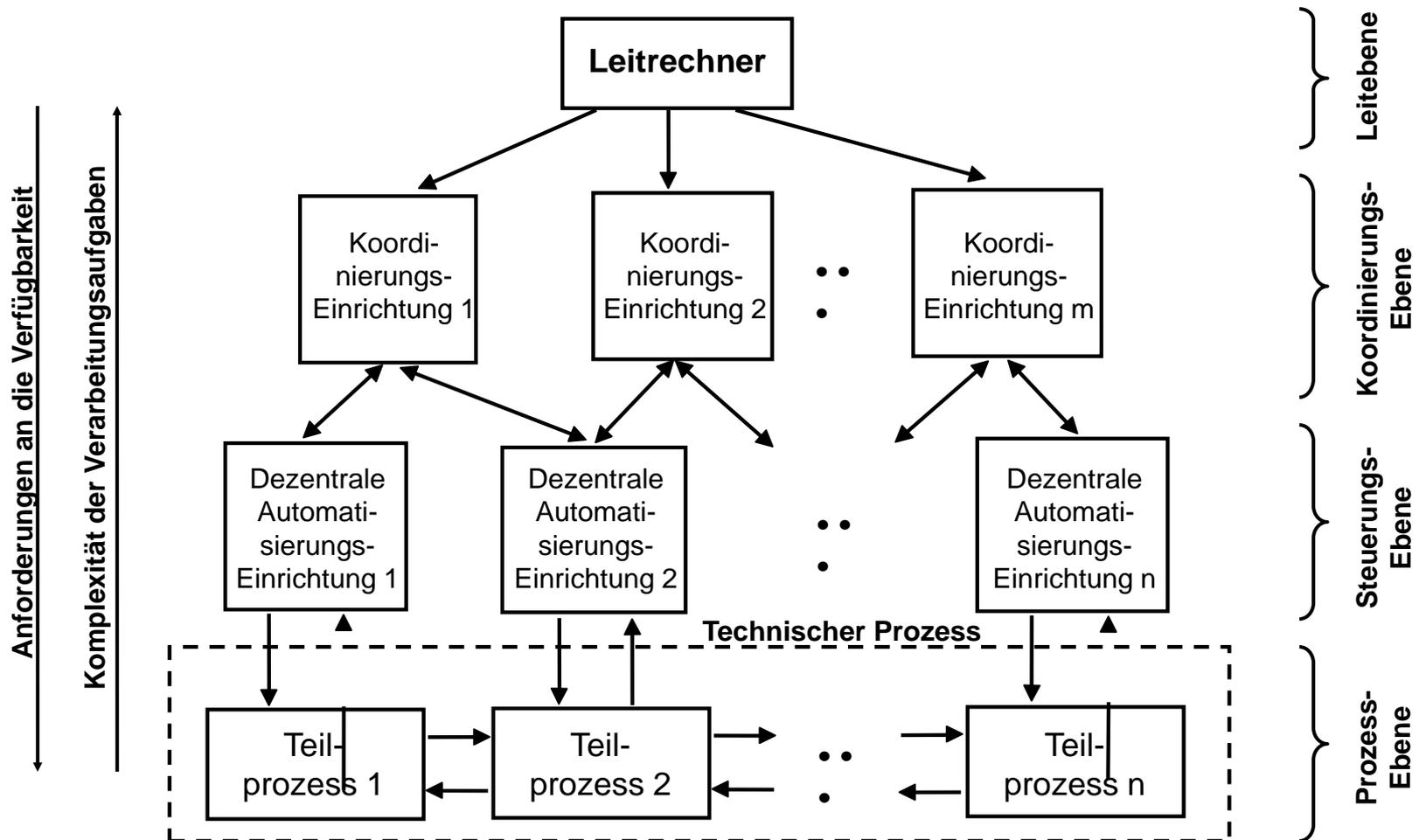
PCI Express Eigenschaften

- Punkt-zu-Punkt-Verbindungen
- Weniger Verbindungsleitungen durch seriellen Bus
- Skalierbar: x1, x2, x4, x8, x12, x16, x32 lanes/link
- Doppelte Simplex-Verbindung
- Differentielle Übertragung
- 2.5, 5.0 and 8.0 GT/s Transfer/Richtung/s
- Paketbasiertes Transaktionsprotokoll



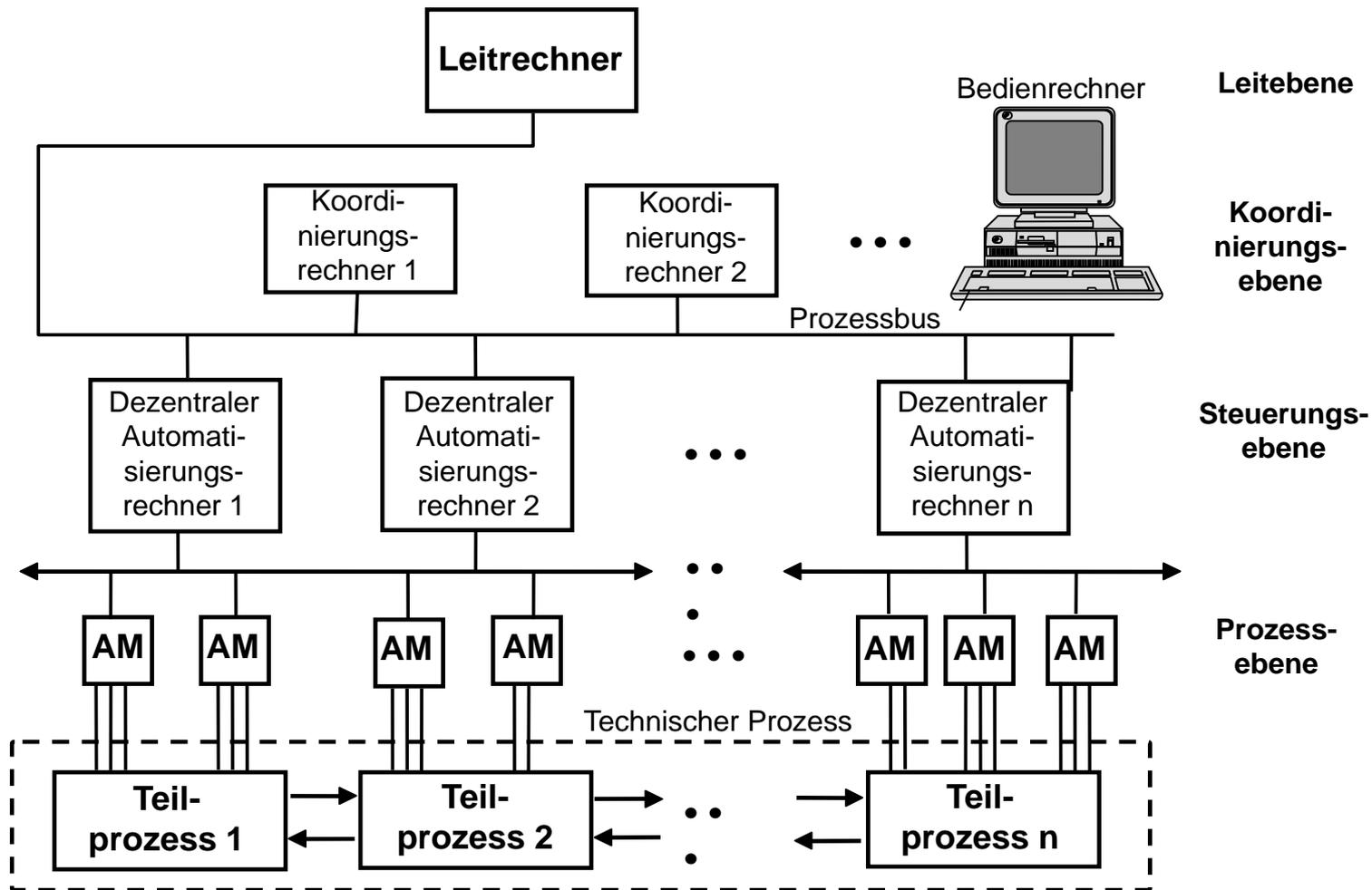
www.pcisig.com/developers/main/training_materials/

Automatisierungshierarchie



Kombination von zentraler und dezentraler Struktur durch Einführung einer Hierarchie von Automatisierungseinheiten

Automatisierungshierarchie mit Bussen



Realisierung einer Automatisierungshierarchie mit Hilfe von Bus-Systemen. AM = Anschluss-Modul für Sensoren/Aktoren