

# IT-Komponenten



**JProf. Dr. Gunnar Stevens**

Human Computer Interaction

University of Siegen

[gunnar.stevens@uni-siegen.de](mailto:gunnar.stevens@uni-siegen.de)

# Agenda

2

- Geschichtliche Entwicklung
- Rechnerklassen
- Aufbau der CPU
- Speichermedien
- Busse
- Peripheriegeräte

# Geschichtliche Entwicklung

# Geschichte 1/2

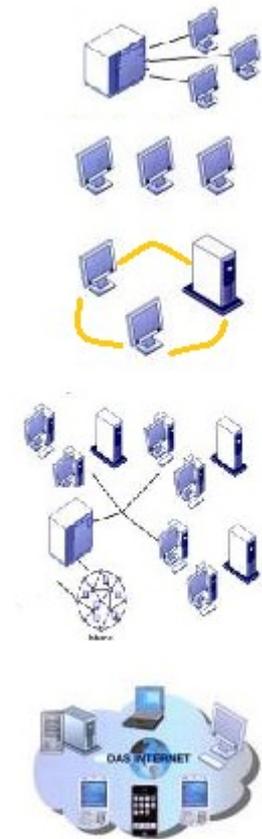
4

Jahr	Name	Hersteller	Beschreibung
1834	Analytical Engine	Charles Babbage	Entwurf einer mechanischen Rechenmaschine
1941	Z3	Konrad Zuse	Erster programmierbarer, turingmächtiger Computer
1952	IAS	John von Neumann	Erster Rechner mit Von-Neumann-Architektur
1960	PDP-1	DEC	Erster Minicomputer

# Geschichte 2/2

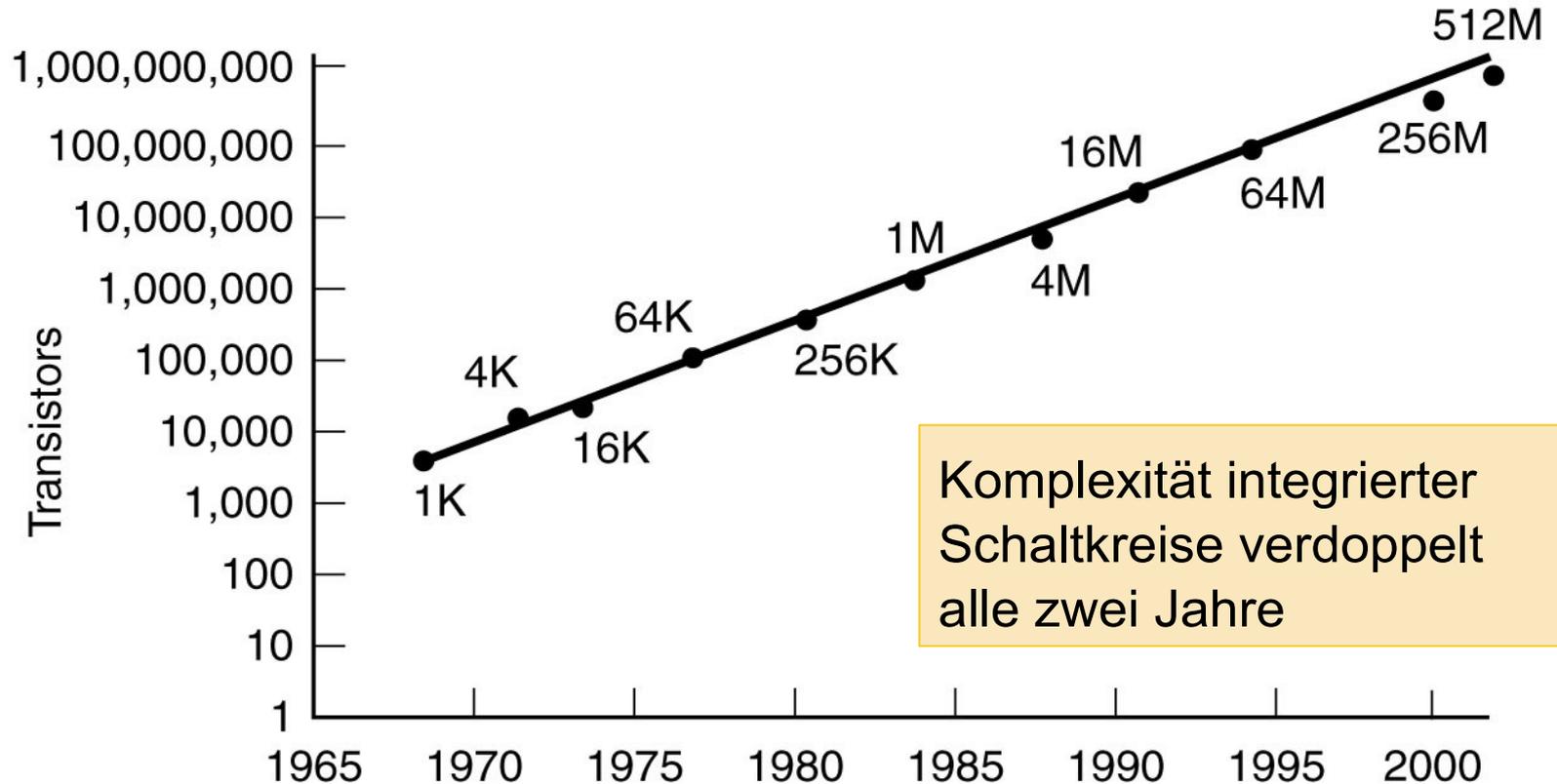
5

- 1959 bis heute
  - ▣ Epoche der Groß- und Minirechner
- 1981 bis heute
  - ▣ Arbeitsplatzrechner und Personal Computing
- 1983 bis heute
  - ▣ Client-Server Computing
- 1992 bis heute
  - ▣ Betriebliche und Internetbasierte Informationssysteme
- 2000 bis heute
  - ▣ Cloud Computing

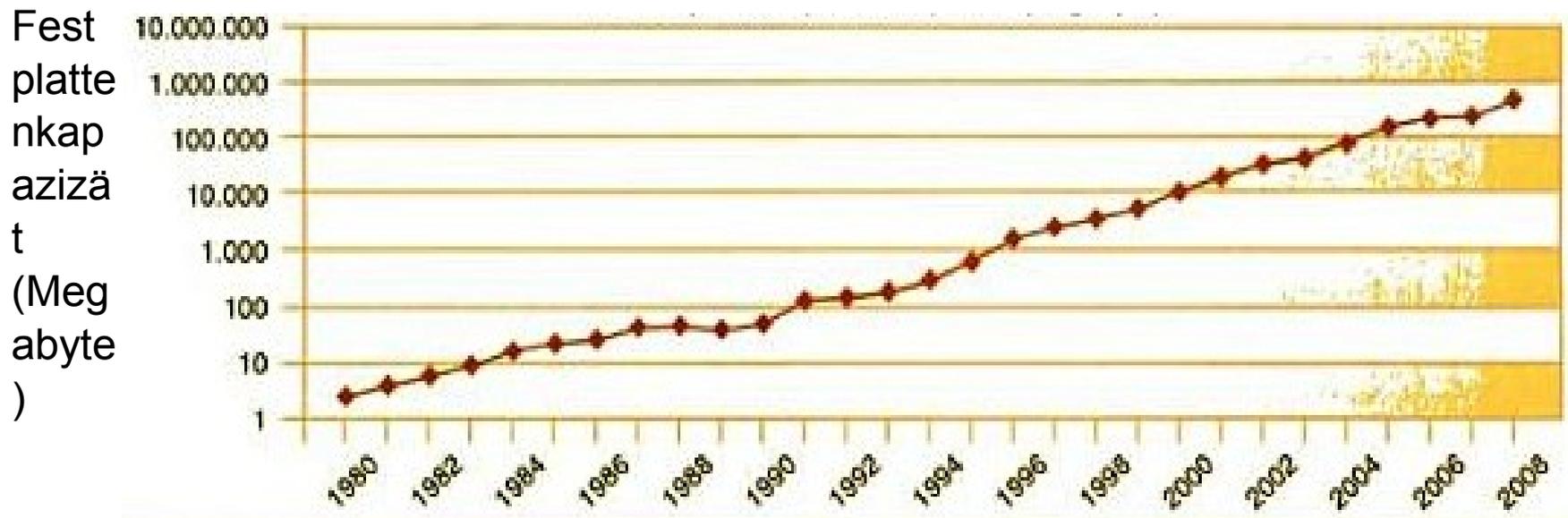


# Moore's Law

6



# Wachstum an 7 Speicherkapazität

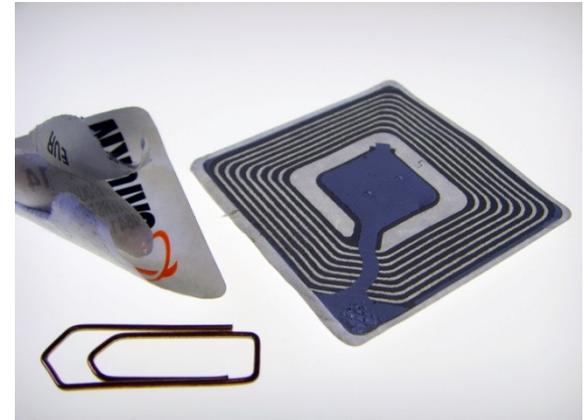


# Rechnerklassen

# Wegwerfcomputer

9

- Beschreibung
  - klein (wenige mm)
  - billig (<1€)
  - Minimalsysteme  
(kein „vollständiger“ Computer)
- Beispielanwendungen
  - Produktkennzeichnung
  - ePass
- R



# Mikrocontroller

10

- Beschreibung
  - ▣ Preis ca. 0,50-20 €
  - ▣ Vollständige Computer
  - ▣ Werden oft in andere Systeme eingebettet
- Beispielanwendungen
  - ▣ Mobiltelefone
  - ▣ Fernseher
  - ▣ Uhren
- Rechenleistung



# Personal Computer (PC)



11

- Beschreibung
  - ▣ Preis ca. 200-2.000€
  - ▣ Verfügen über ein Betriebssystem
- Beispielanwendungen
  - ▣ Desktop- oder Notebook-Computer
- Rechenleistung



# Server

12

- Beschreibung
  - ▣ Besonders leistungsfähig
  - ▣ Preis ca. 2.000-8.000 €
- Beispielanwendungen
  - ▣ Datenbankserver
  - ▣ Webserver
- Rechenleistung



# Mainframe

13

- Beschreibung
  - ▣ Sehr hohe E/A Kapazität (E/A = Eingabe/Ausgabe)
  - ▣ Preis > 100.000€ (oft 1-3 Mio.€)
- Beispielanwendungen
  - ▣ Webserver
  - ▣ Buchhaltung
- Rechenleistung



# Cluster

14

- Beispielanwendungen
  - ▣ Wissenschaftliche Berechnung
  - ▣ Produktentwicklung
  - ▣ Webserver (Load-Balancing Cluster)
- Beschreibung
  - ▣ Aus vielen einzelnen Servern zusammengesetzt
  - ▣ Preis >25.000€
- Rechenleistung

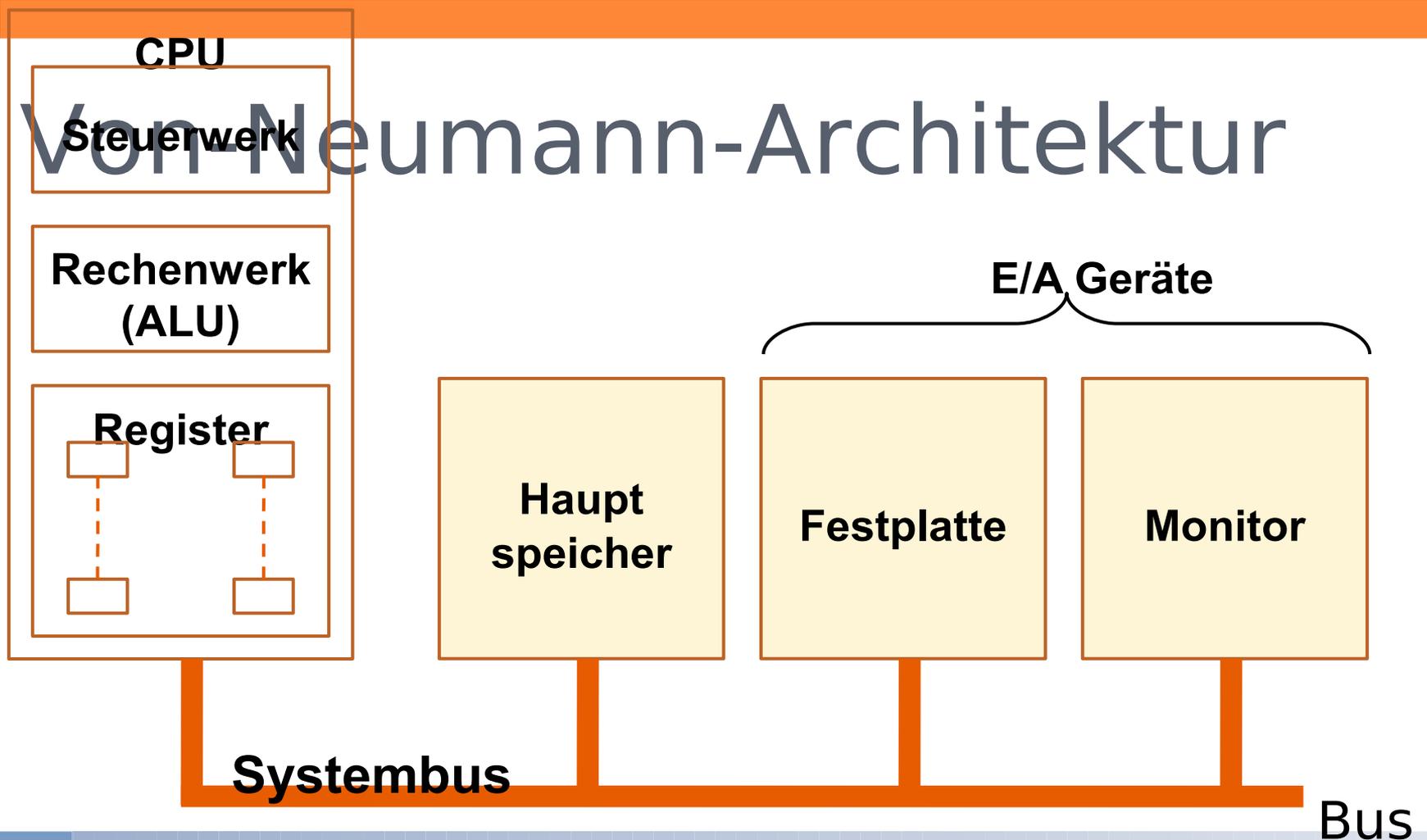


# Aufbau der CPU

# Einfacher Rechner nach der

16

## Von-Neumann-Architektur



# Register

17

- Register sind Speicherzellen auf der CPU
- Spezialregister
  - ▣ dienen einem bestimmten Zweck
  - ▣ Beispiele
    - Akkumulator (EAX)
      - Speichert Ergebnisse und Operanden
    - Befehlszeiger (EIP)
      - Speichert die Speicheradresse des nächsten Befehls
- Mehrzweckregister

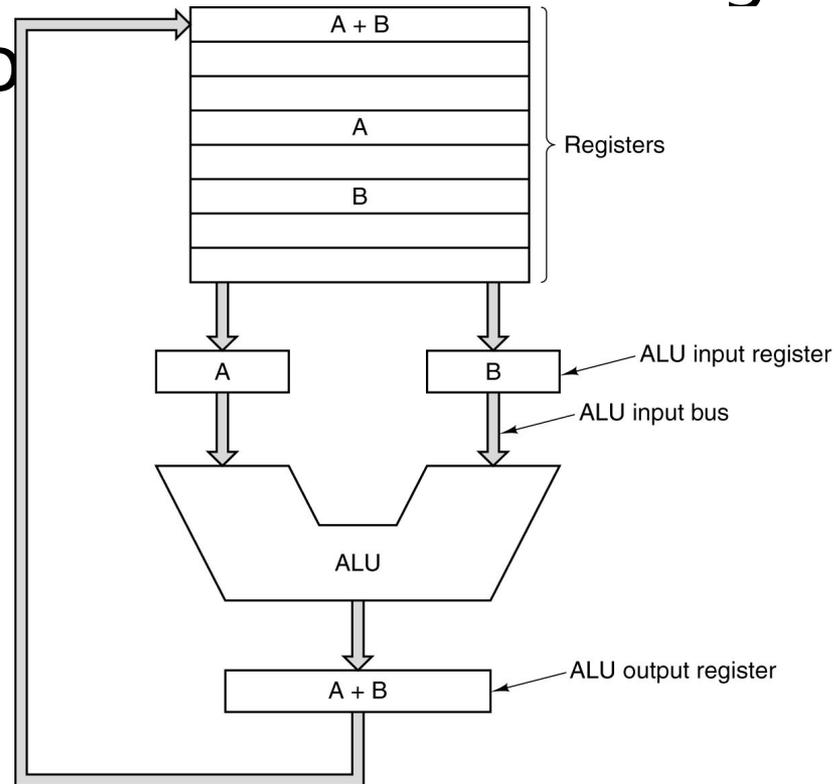
□ Diese Register können vom Programmierer beliebig genutzt werden

# Rechenwerk (Arithmetic

## Logic Unit)

18

- Führt arithmetische und logische Operationen



# Das Steuerwerk

19

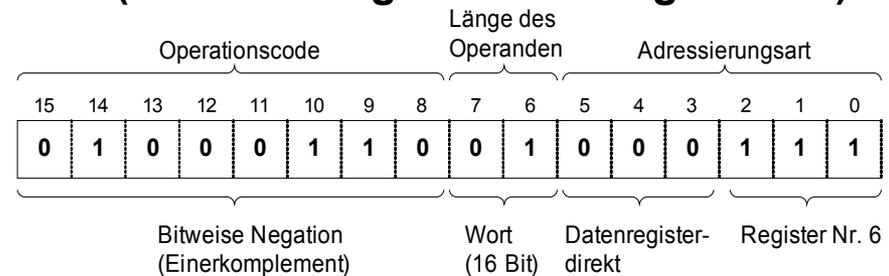
- Lädt Befehle aus dem Arbeitsspeicher
- Übersetzt die Maschinenbefehle (Opcodes) in Steuersignale für Register und ALU (Microcode)
- Führt die Steuerbefehle aus

# Maschinenbefehle

- Maschinenbefehle bestehen aus einem Befehlstypen und ggf. ein oder zwei Operanden

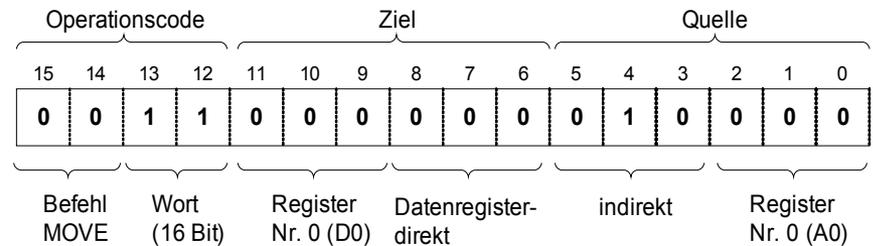
Beispiel (Motorola 68000):

## NOT D6 (Bitweise Negation von Register D6):



- Befehlstypen sind z.B.
  - ▣ Logische oder arithmetische Operationen
  - ▣ Laden und speichern von Registern in den Arbeitsspeicher

## MOVE (A0), D0 (Übertrage das Wort an der mit A0 adressierten Speicheradresse in das Register D0):



- Operanden können sein
  - ▣ Adressen im Speicher
  - ▣ Register

## □ Die Werte selbst



# Assembler

Der Assembler ist eine Kurzschreibweise für Maschinenbefehle, um diese für den Menschen besser lesbar zu machen.

Einige Beispiele für Befehle des 68000-Assemblers:  
Wertzuzuweisung:

MOVE D0,D1                   *(schreibe den 16-Bit Wert von Register D0 nach D1)*

Arithmetische Befehle:

ADD #35,D0                   *( D0 <= D0 + 35),*

MULU D2,D3                   *( D3 <= D2 \* D3),*

SUB (A0),D4                   *( D4 <= D4 - (A0)),*

Sprungbefehle:

BRA                           *(unbedingter Sprung)*

CMP D1,D2                   *(bildet Differenz  $d = D2 - D1$ ), anschließend Vergleich: BEQ*  
*(falls  $d = 0$ ), BGT (falls  $d > 0$ ), BLT (falls  $d < 0$ ), ...*  
Prof. Dr. Gunnar Stevens  
Human-Computer-Interaction, University of Siegen  
gunnar.stevens@uni-siegen.de

# Assembler

Ein einfaches Assemblerprogramm für die Summe der ersten  $n$  Zahlen.

Eingabe:  $n$ .

Hier wird solange  $k$  erhöht und zur Summe  $s$  addiert, bis  $k = n$ .

	MOVE	#0,D0	Summe $s$ (Register D0) auf 0 setzen
	MOVE	#0,D1	Zählwert $k$ (Register D1) auf 0 setzen
	MOVE	$n$ ,D2	Wert $n$ in Register D2 schreiben
marke1	CMP	D2,D1	vergleiche $n$ und $k$ (in Register D2 und D1)
	BLE	marke2	falls nicht größer, springe zu marke2
	ADD	#1,D1	Zählwert $k$ in D1 um 1 erhöhen
	ADD	D1,D0	Zählwert $k$ zu $s$ addieren
	BRA	marke1	unbedingter Sprung zu marke1
marke2			Ende

# Abstraktionsebenen

```
int add(int a, int b)
{
    int ret = a + b;
    return ret;
}
```

```
_add
    MOVEA.L A7, A6
    MOVEM D0-D7, -(A7)
    MOVE.L +(A6), D3
    MOVE.L +(A6), D2
    ADD.L D3, D2
    MOVE.L D2, -(A6)
    MOVEM (A7)+, D0-D7
    RTS
```

C-Programm (.c)

Compiler (gcc -S)

Assembler-Programm (.s)

Assembler (as)

Object-Code (.o)

Linker (ld)

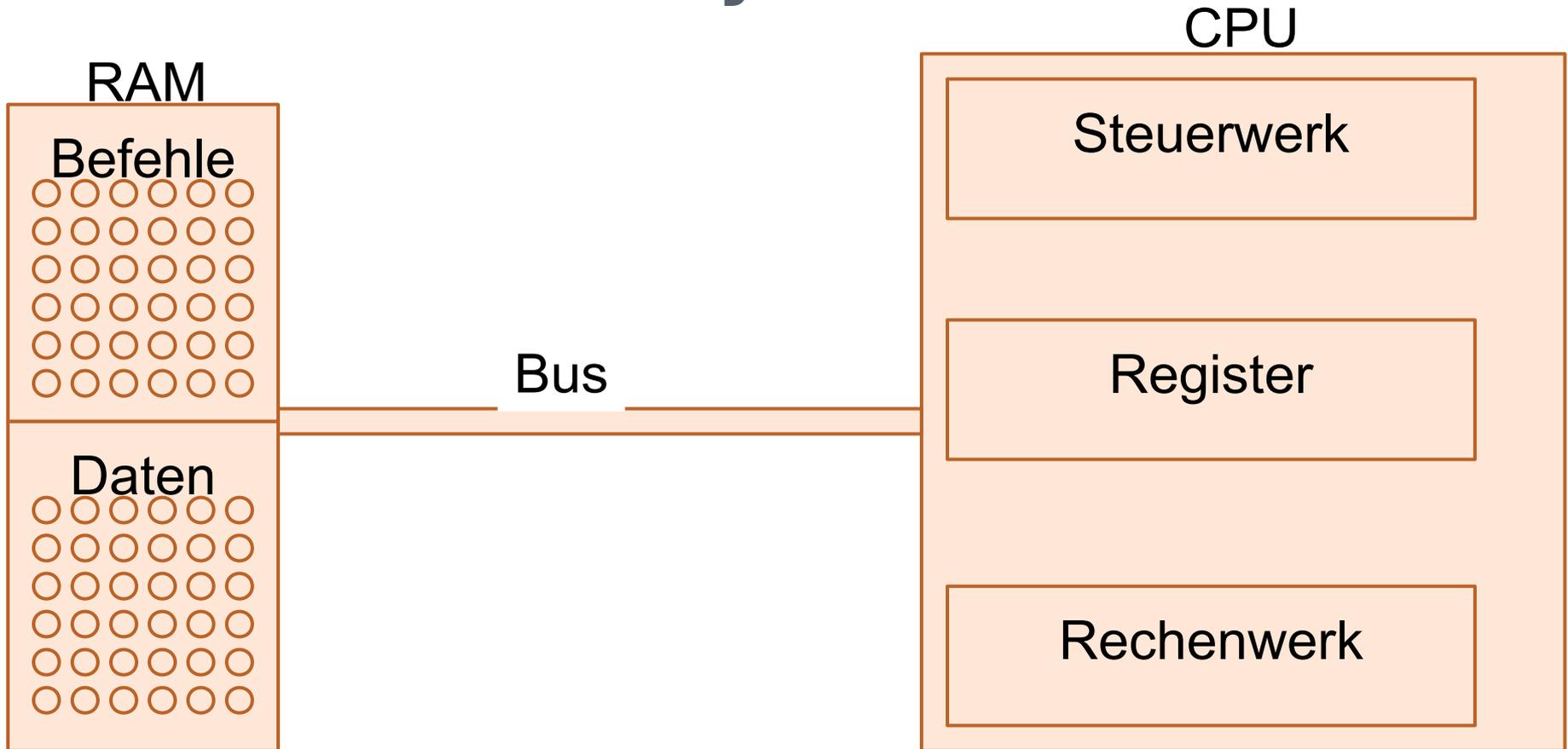
Ausführbarer Code

# Abrufen-Dekodieren-

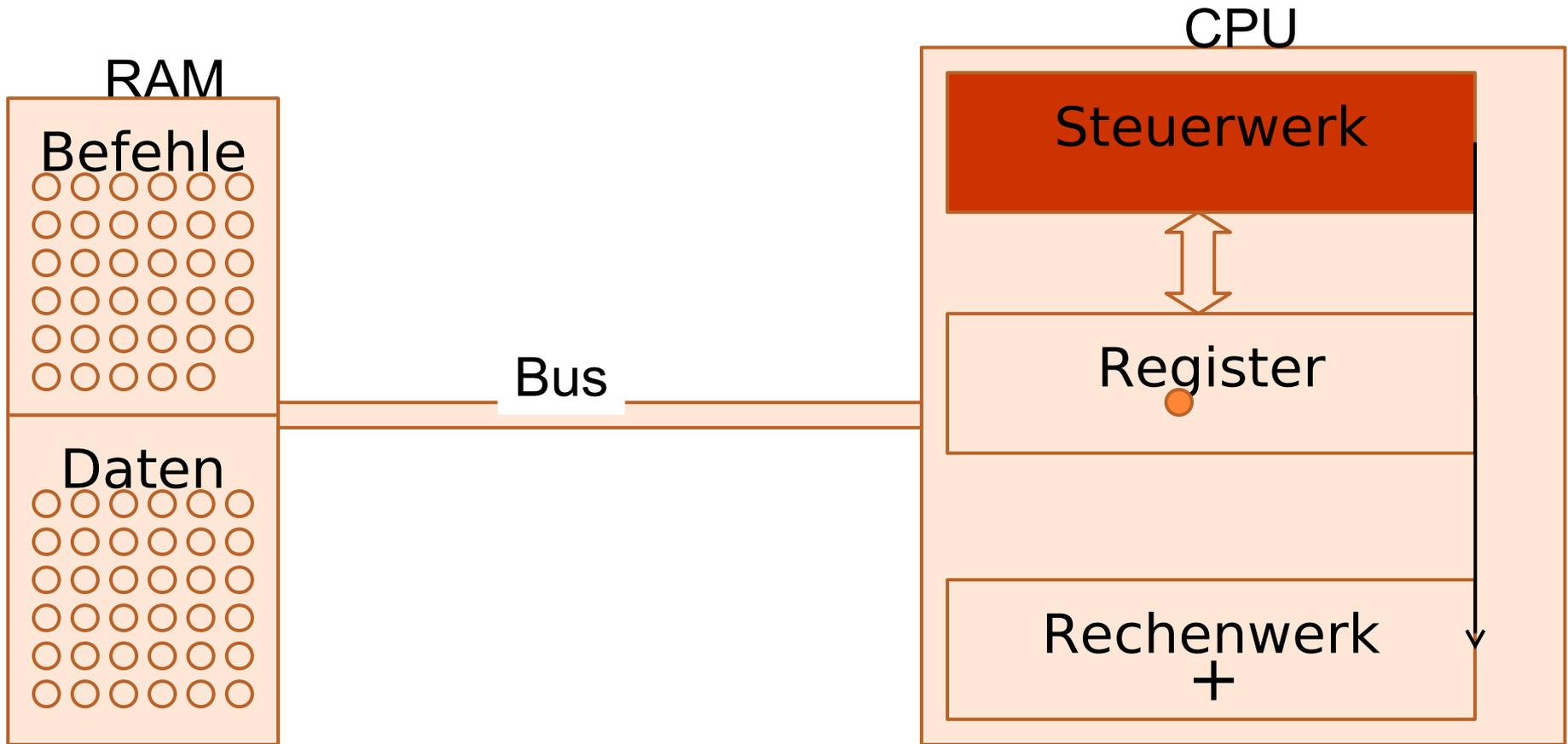
Abrufen

# Ausführen-Zyklus

24



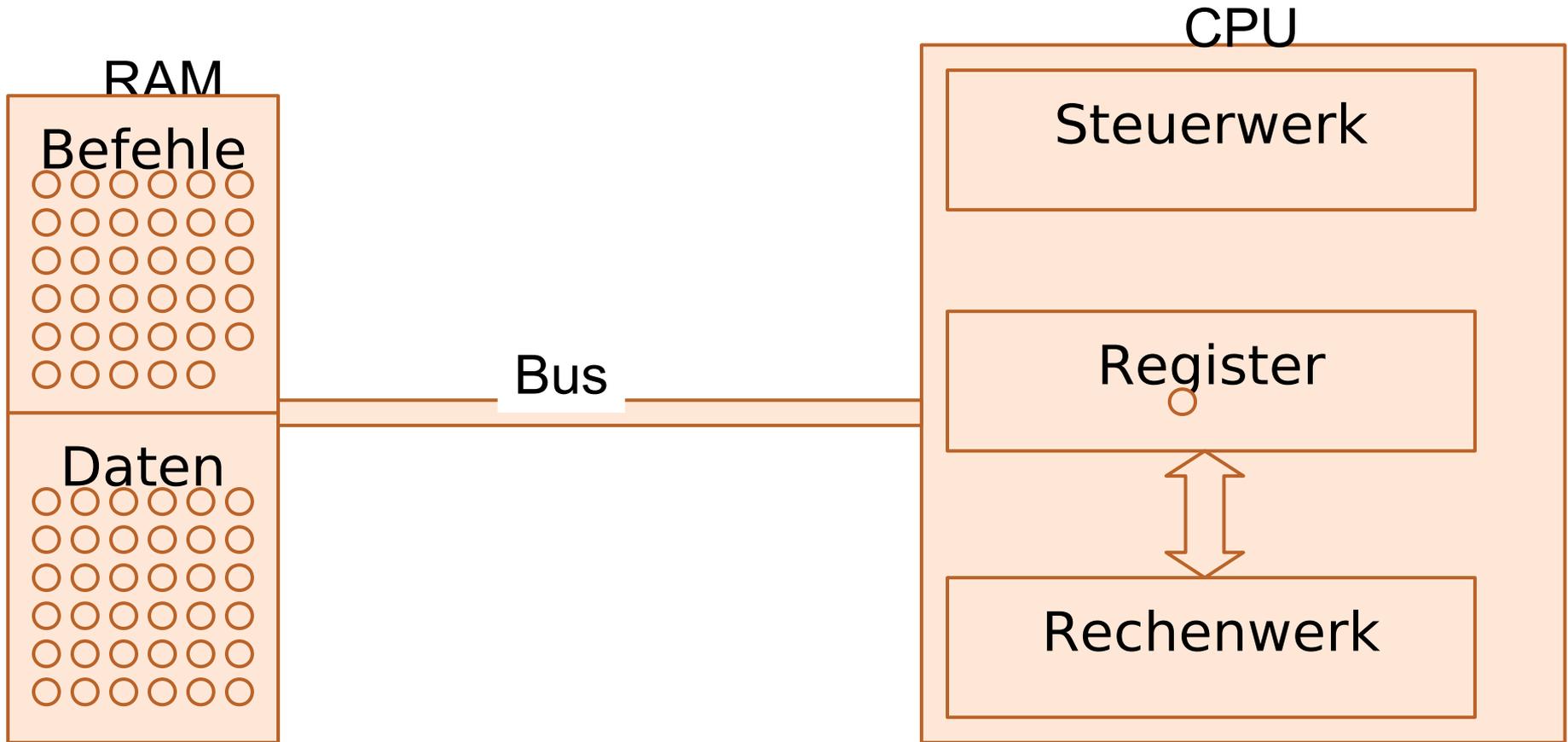
# Abrufen-Dekodieren- Dekodieren Ausführen-Zyklus



# Abrufen-Dekodieren- Ausführen

26

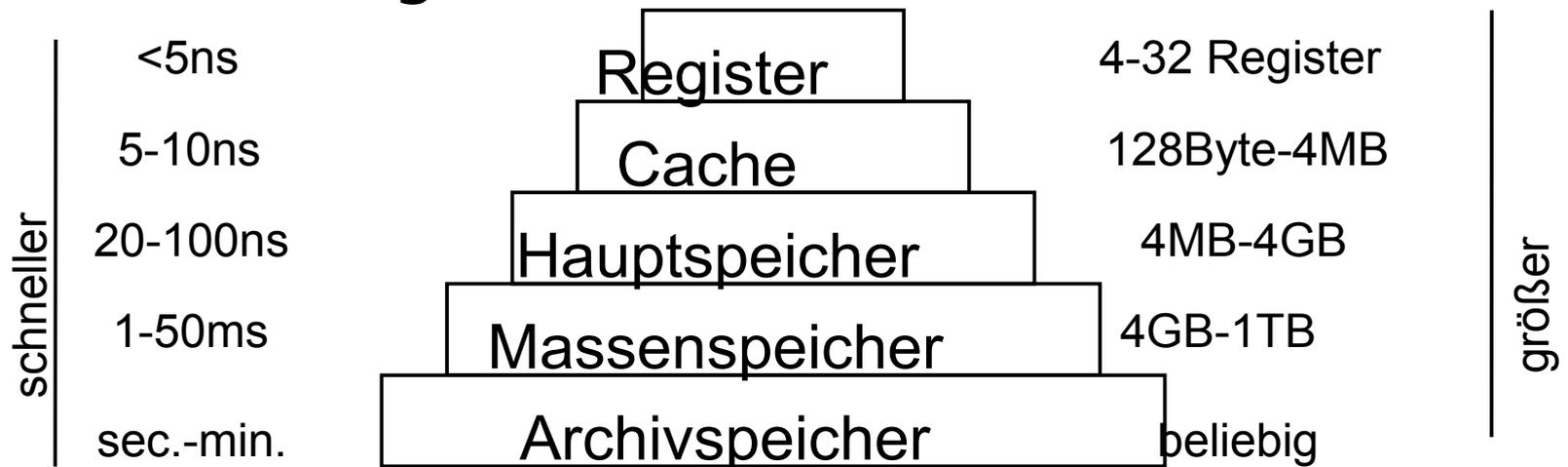
# Ausführen-Zyklus



# Speicher

# Speicherhierarchie

Die Speicherhierarchie entsteht im Zusammenwirken von Speichern unterschiedlicher Größe und Zugriffszeiten  
Geschwindigkeit:



# Hauptspeicher

29

- ist flüchtig  
(verliert ohne Strom sein Zustand)
- besteht aus Speicher-stellen, die je eine Informationseinheit aufnehmen

kann

JProf. Dr. Gunnar Stevens

Human Computer Interaction, University of Siegen

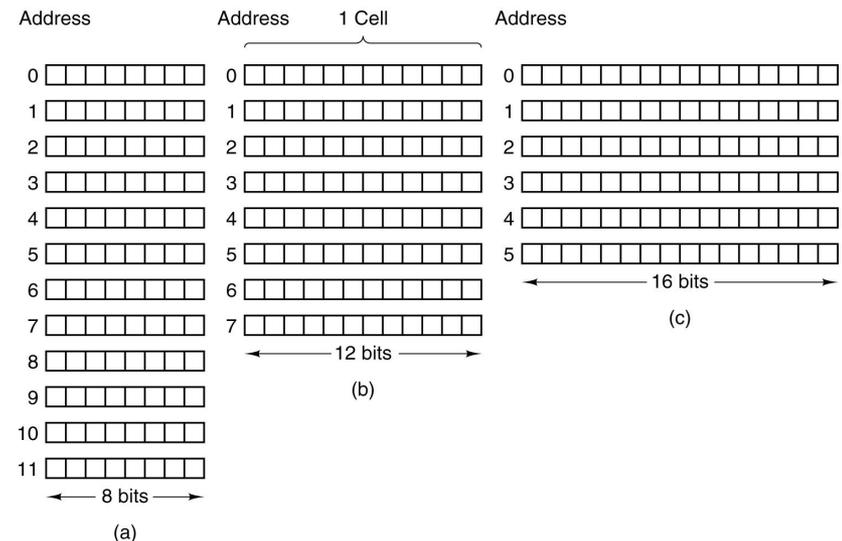
gunnar.stevens@uni-siegen.de



# Speicherstelle und Adressraum

30

- Speicherstelle
  - ▣ Größe meist an Register-größe der CPU angepasst
- Adressraum
  - ▣ Menge der adressierbaren Speicherstellen
  - ▣ Größe meist an Register-größe der CPU angepasst



*Drei Möglichkeiten einen 96 bit großen Speicher in Speicherstellen einzuteilen*

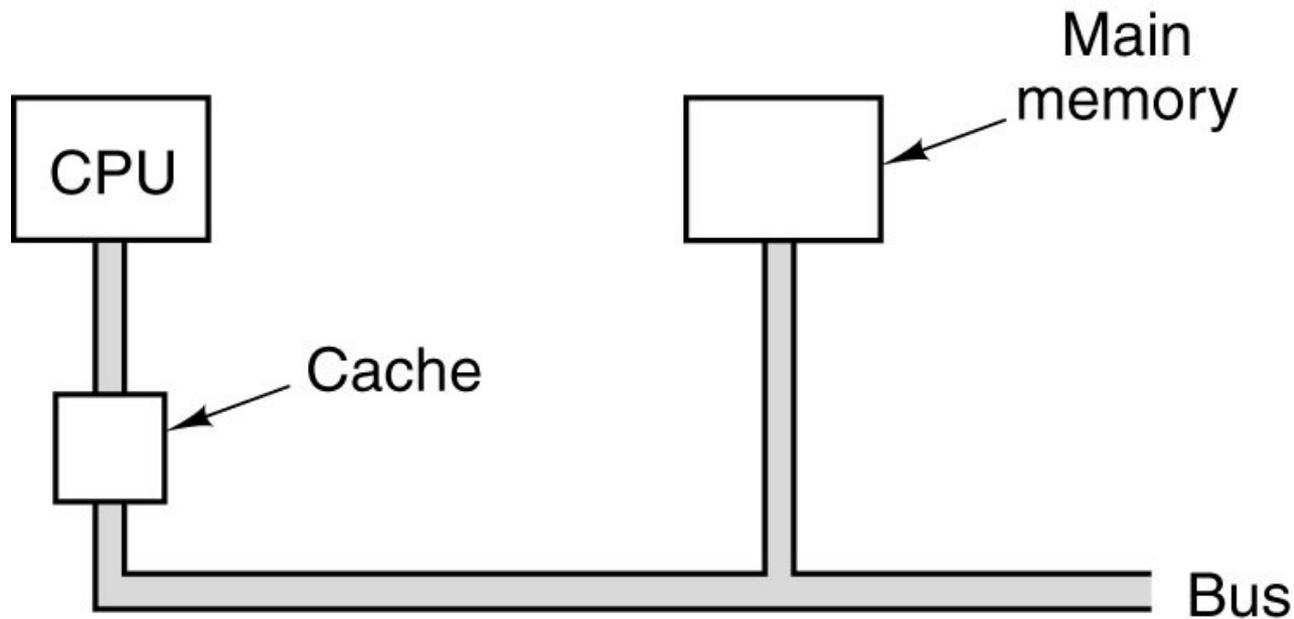
# Zugriff

31

- Im Hauptspeicher ist ein wahlfreier Zugriff (*Random Access*) auf die einzelnen Speicherzellen möglich
  - ▣ Es kann auf jede Speicherzelle gleich schnell zugegriffen werden
- Im Gegensatz dazu:
  - ▣ Sequenzieller Zugriff (die Daten sind aufeinanderfolgend abgespeichert)
  - ▣ Beispiel: Magnetband

# Der Cache

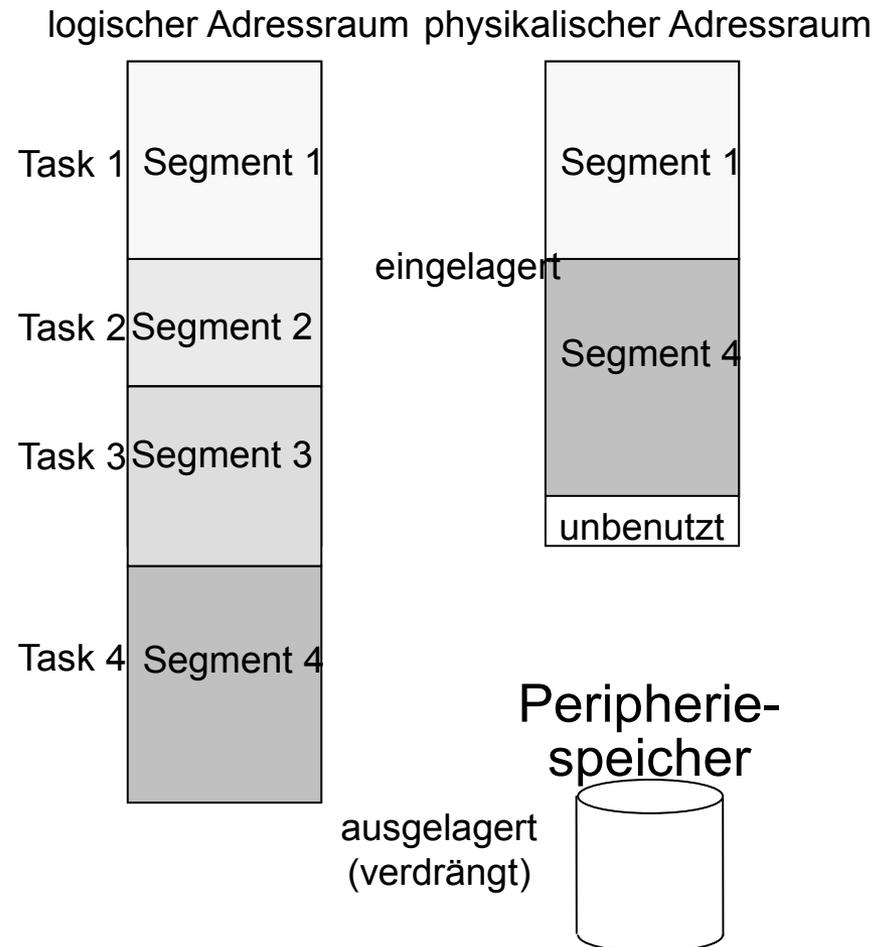
32



Der Cache ist ein Pufferspeicher zwischen der CPU und dem Hauptspeicher  
Physisch befindet er sich in der CPU

# Virtueller Speicher

- Reicht der Hauptspeicher nicht aus, kann auf die Festplatte ausgelagert werden.
- Damit ist es möglich einen sehr großen „virtuellen“ Adressraum zu bekommen: (z.B. 4 GByte bei 32 Bit Adressen).
- Reicht der Hauptspeicher nicht aus, kann auf die Festplatte ausgelagert werden.
- Dieser Vorgang wird z.B. durch Segmentierung oder Paging realisiert.



# Aufbau einer Festplatte

34

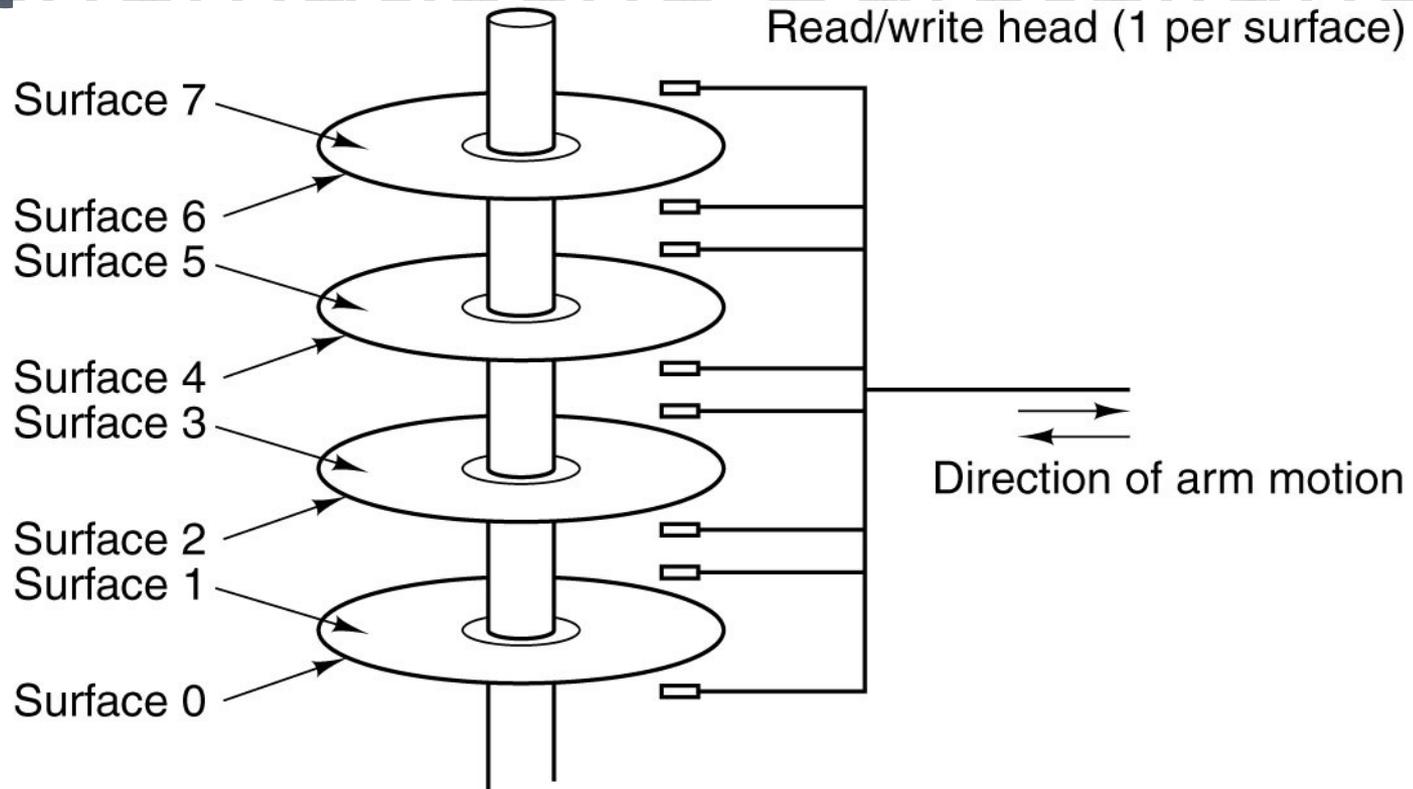
- Eine Festplatte besteht aus
  - ▣ mehreren runden Aluminiumscheiben mit magnetisierbarer Beschichtung
  - ▣ zwei Schreib-/Leseköpfe pro Scheibe mit je einer Induktionsspule
  - ▣ die Schreib-/Leseköpfe schweben auf einem Luftkissen über den Oberflächen der Scheiben



# Festplatte

## (schematische Darstellung)

35



# Schreiben/Lesen

36

## □ Schreiben

- Je nach Polarisierung des Stroms, der durch einen Schreib-/Lesekopf fließt, werden die magnetischen Partikel auf der Platte nach rechts oder links ausgerichtet

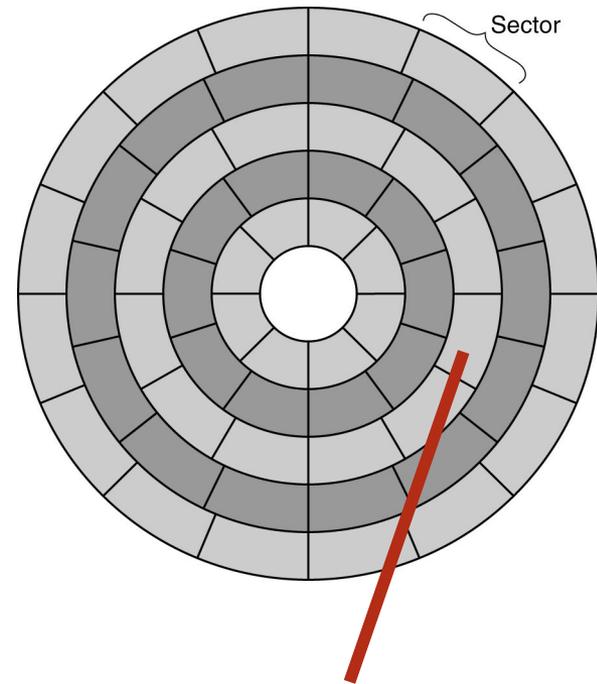
## □ Lesen

- Läuft unter dem Kopf ein magnetisierter Bereich vorbei, wird ein positiver oder negativer Strom indiziert

# Prinzip der

# Datenspeicherung

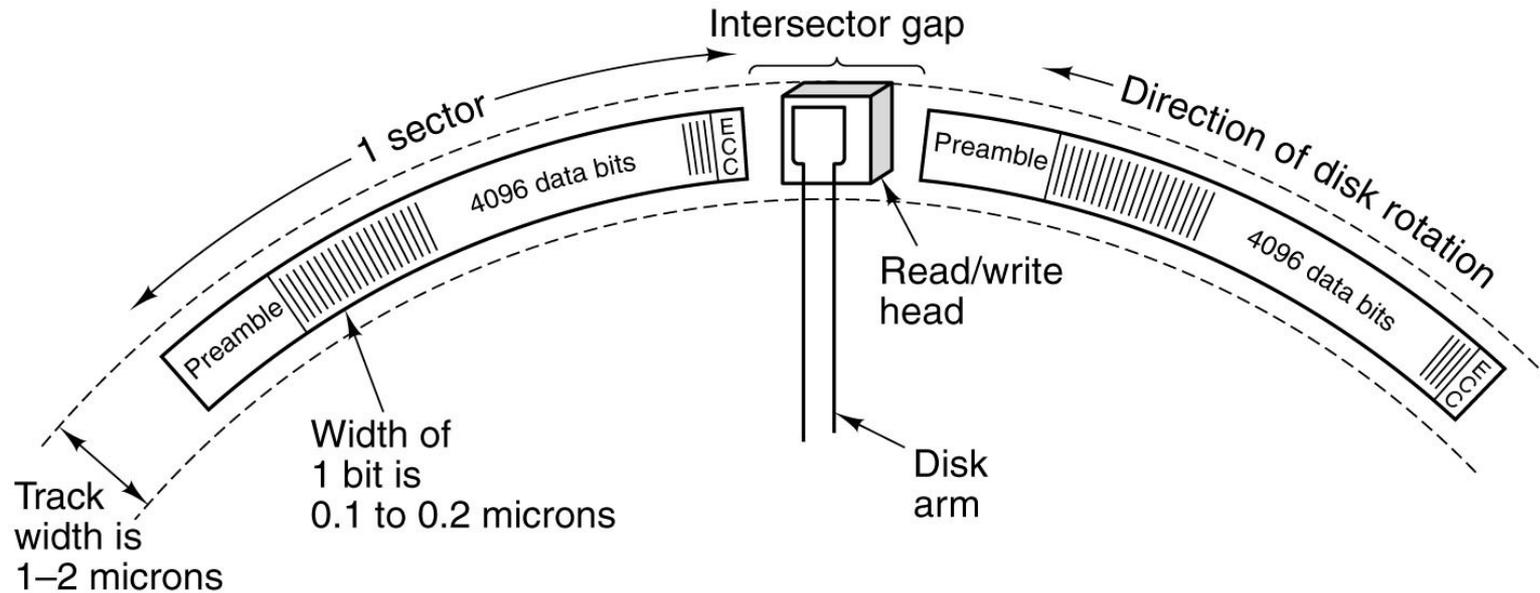
- Einteilung in Spuren und Sektoren (Formatieren)
- Spuren liegen konzentrische Kreise in gleichmäßigem Abstand um den Mittelpunkt (äußerste Spur = 0 innerste = N)
- Sektoren unterteilen jede Spur in einzelne Abschnitte gleicher



Spur 2, Sektor 3

# Aufteilung einer Spur

38

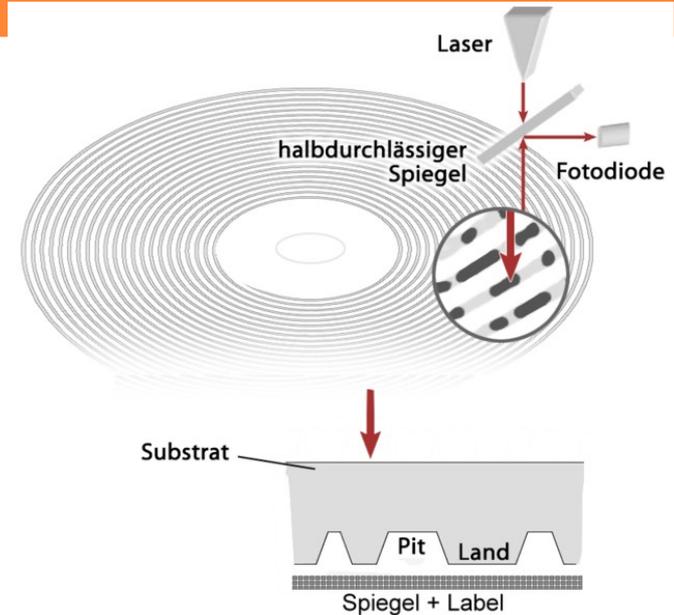


Ausschnitt aus einer Plattenspur.  
Dargestellt sind zwei Sektoren.

# CD-ROMs

39

- Daten mit Hilfe einer von innen nach außen laufenden Spiralspur gespeichert
- Spiralspur besteht aus *Pits* (Gruben) und *Lands* (Flächen)
- Abstastung des Daten durch Laserstrahl und Fotodiode



([http://commons.wikimedia.org/wiki/Image:CD\\_Prinzip.p](http://commons.wikimedia.org/wiki/Image:CD_Prinzip.p))

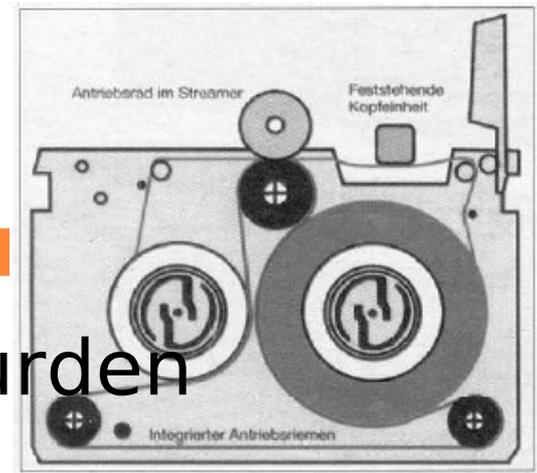
# DVDs

40

- Grundsätzlich gleicher Aufbau wie CD
- Unterschiede
  - ▣ Kleinere Pits
  - ▣ Engere Spirale
  - ▣ Laser mit geringerer Wellenlänge
  - ▣ Zwei Schichten möglich
  - ▣ Höhere Datendichte und Transfer rate



# Magnetband



41

- Erste Bandspeichergeräte wurden 1949/50 ausgeliefert
- Gegenwärtige Nutzung hauptsächlich als Backup- und Archiv-Medium für große Datenbeständen
- Information wird durch Magnetisierung festgehalten und wieder ausgelesen
- Die Zugriffsart ist sequentiell, d.h. die Daten werden aufeinanderfolgend kontinuierlich geschrieben oder gelesen

# RAID (**R**edundant **A**rray of

## 42 **I**nexpensive **D**isks)

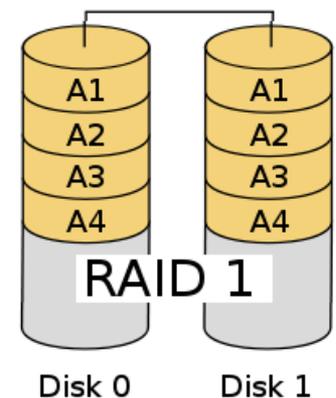
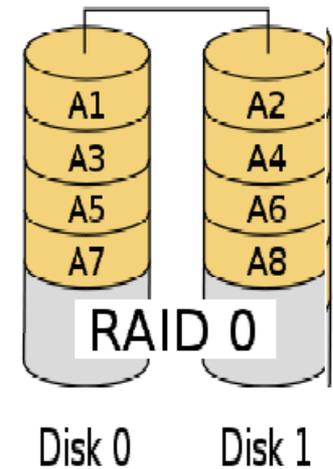
- Grundidee:
  - Zusammenschliessen vielen kleineren Festplatten zu einen großen logischen Laufwerks
  - Aus Sicht des Benutzers unterscheidet sich ein logisches RAID-Laufwerk nicht von einer einzelnen Festplatte.
- Ziele:
  - Erhöhung der Ausfallsicherheit
  - Steigerung der Transferraten
  - Aufbau großer logischer Laufwerke

# RAID (1)

## Redundant Array of Inexpensive Disks

43

- RAID 0 (Striping)
  - ▣ blockweises Verteilen der Daten auf verschiedenen Festplatten
  - ▣ Steigerung der Transferraten (insbesondere bei Parallelität)
  - ▣ Keine Steigerung der Ausfallsicherheit
  - ▣ Nutzkapazität =  $N * \text{Festplattenkapazität}$
- RAID 1 (Mirroring)
  - ▣ Spiegelung der Daten
  - ▣ Steigerung der Ausfallsicherheit
  - ▣ Nutzkapazität =  $N/2 * \text{Festplattenkapazität}$



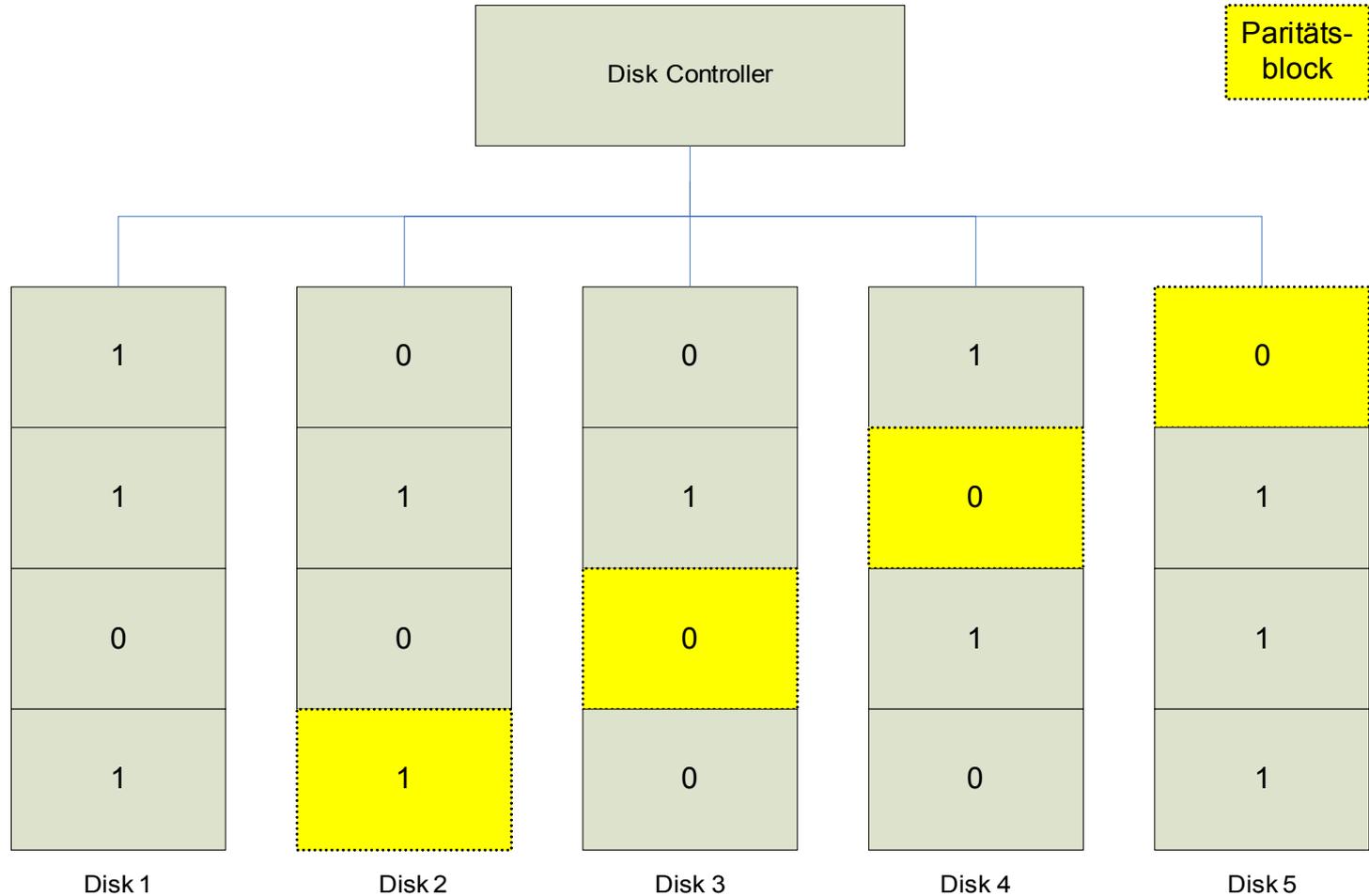
# Kombination von RAID 0 und RAID 1

- um Geschwindigkeit von RAID 0 und Datensicherheit von RAID 1 zu erhalten
- je nach Hersteller: RAID 0+1, RAID 0/1 oder RAID 10
- zwei Varianten sind denkbar
  - ▣ zunächst stripen, dann spiegeln
  - ▣ erst spiegeln, dann stripen

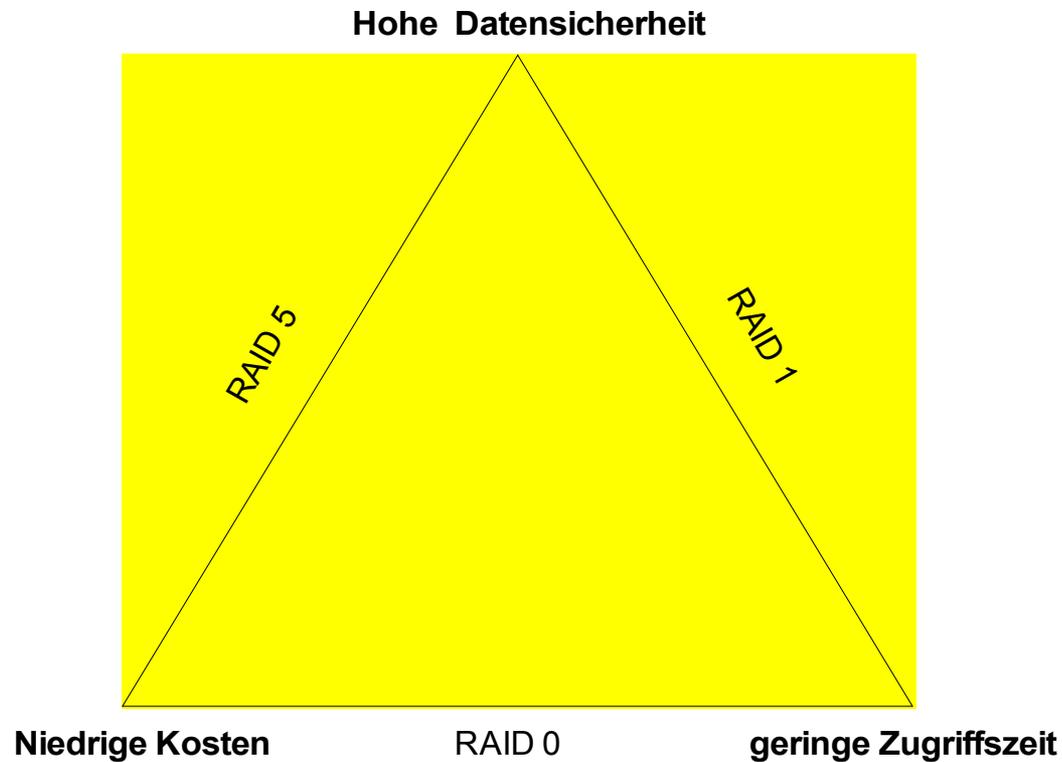
# RAID 5

- blockweises Verteilen der Daten auf verschiedenen Festplatten
- zusätzliche Speicherung von Parity-Informationen für die Fehlerkorrektur
  - ▣ Berechnung der Paritätsdaten erfolgt durch XOR-Verknüpfung der jeweiligen Datengruppe
- Nutzkapazität =  $(N-1) * \text{Festplattenkapazität}$
- Datenverlust beim Ausfall von 2 Platten
- Zusätzliche Kosten beim Schreiben zur Berechnung der Paritätsdaten

# RAID 5



# Zusammenfassung: RAID im Vergleich





Busse



# Bus (Sammelschiene)

- Verbindung mehrere Komponenten eines Rechners über dieselben Leitungen; dabei darf zu einem Zeitpunkt immer nur eine Komponente Informationen auf die Leitungen legen
- Busarten:
  - ▣ Adreßbus, Datenbus, Steuerbus
  - ▣ als Gesamtheit: Systembus

# Bus (Sammelschiene)

Die Informationen werden binär codiert über den Bus **seriell** oder **parallel** übertragen

## SERIELL

- Daten werden 1-bitweise nacheinander übertragen.
- Transportdauer hoch
- Nur geeignet, wenn wenig Daten zur Übertragung anfallen, z. B. beim Anschluss der Maus

## PARALLEL

- Daten werden 8/16/32-bitweise über parallele Datenleitungen übertragen.
- Größere Datenmenge in der gleichen Zeit übertragbar
- Geeignet, wenn viele Daten zur Übertragung anfallen, z. B. beim Anschluss des Druckers

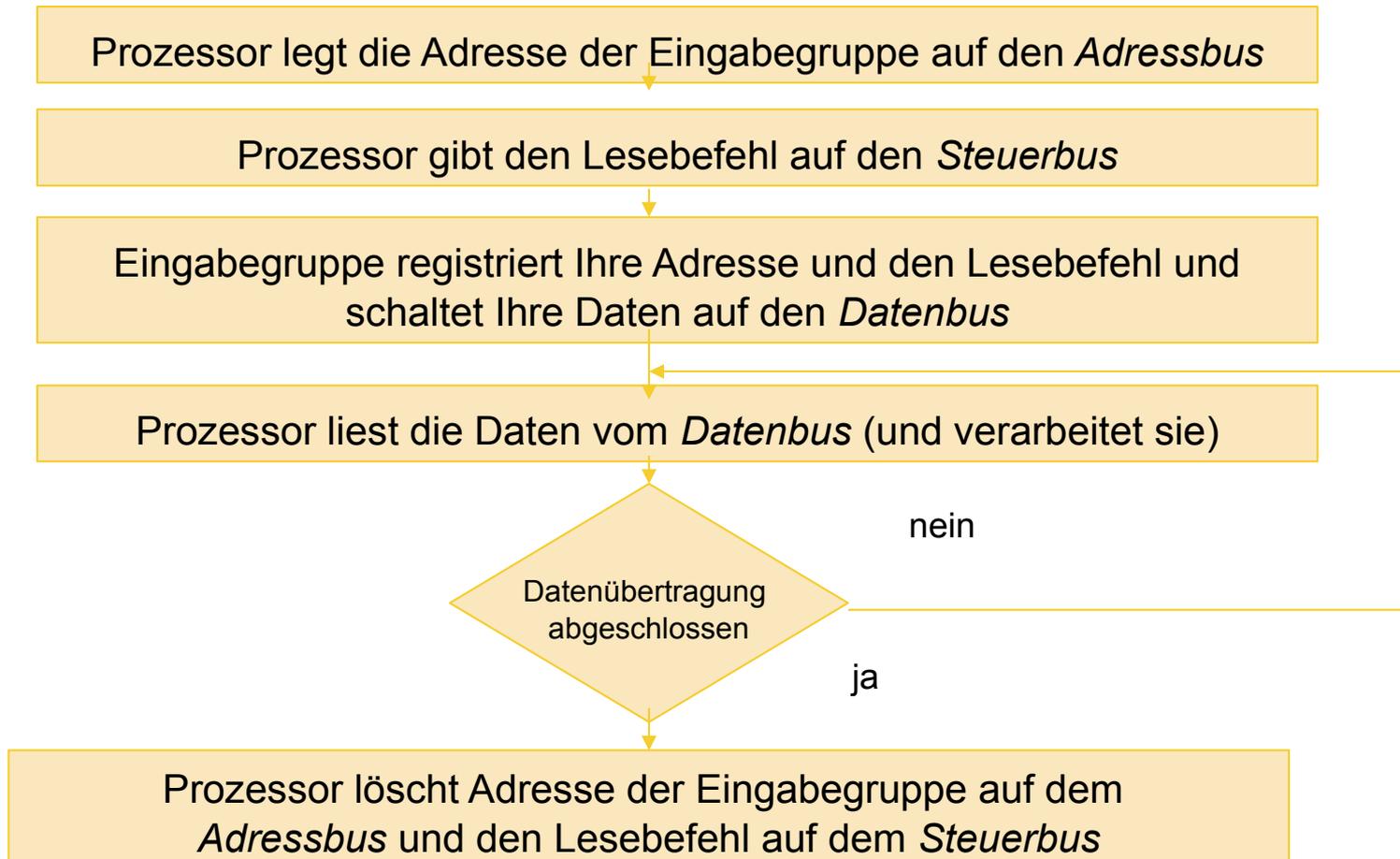
# Bus

- Adreßleitungen
  - ▣ Diejenigen Leitungen, auf denen die Adreßinformation transportiert wird (unidirektional).
- Datenleitungen
  - ▣ Transportieren Daten und Befehle von/zum Prozessor (bidirektional).
- Steuerleitungen
  - ▣ Geben Steuerinformationen von/zum Prozessor (uni- oder bidirektional).

# Bus

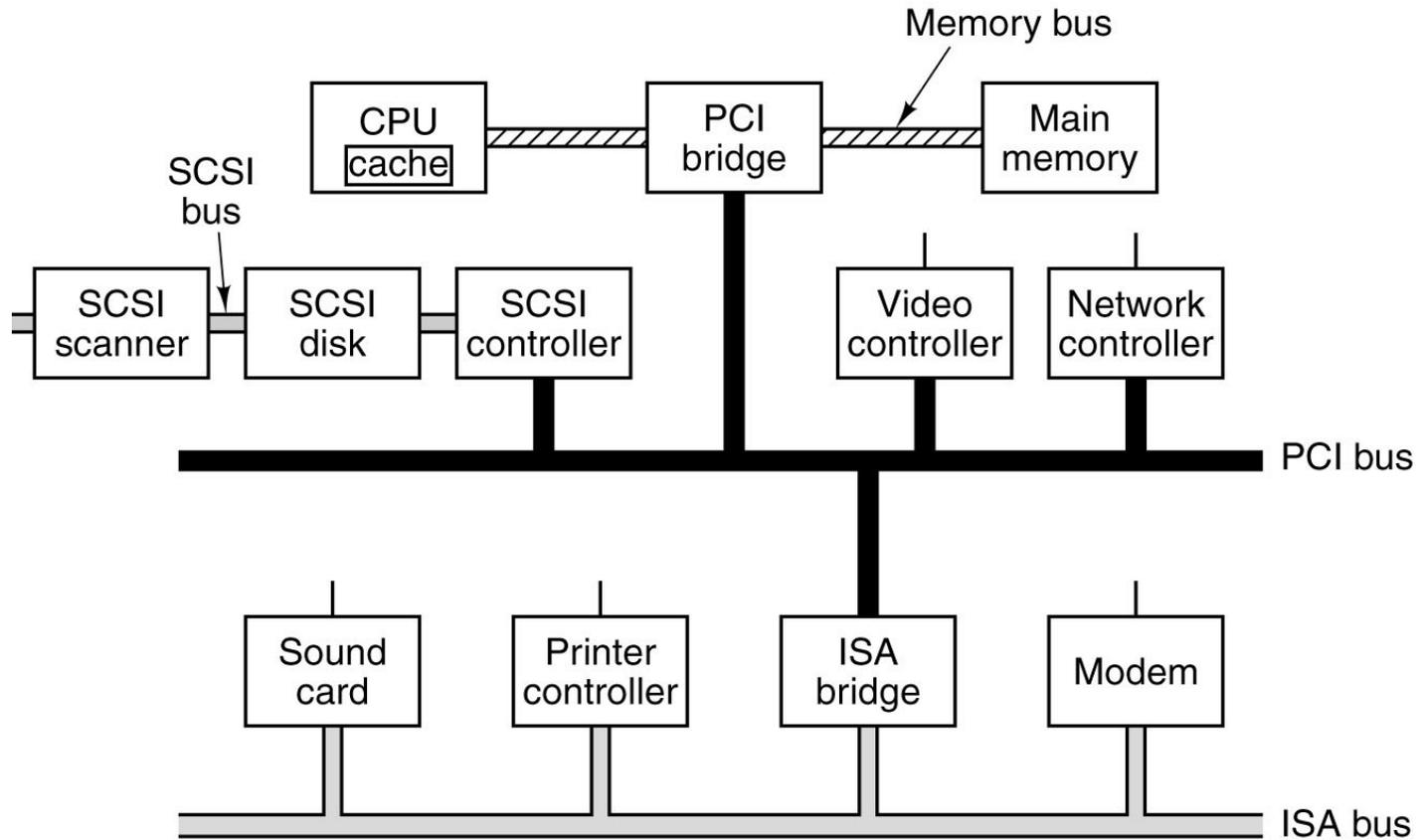
- Adreßleitungen
  - ▣ Diejenigen Leitungen, auf denen die Adreßinformation transportiert wird (unidirektional).
- Datenleitungen
  - ▣ Transportieren Daten und Befehle von/zum Prozessor (bidirektional).
- Steuerleitungen
  - ▣ Geben Steuerinformationen von/zum Prozessor (uni- oder bidirektional).

# Der Chipsatz - Das Bussystem

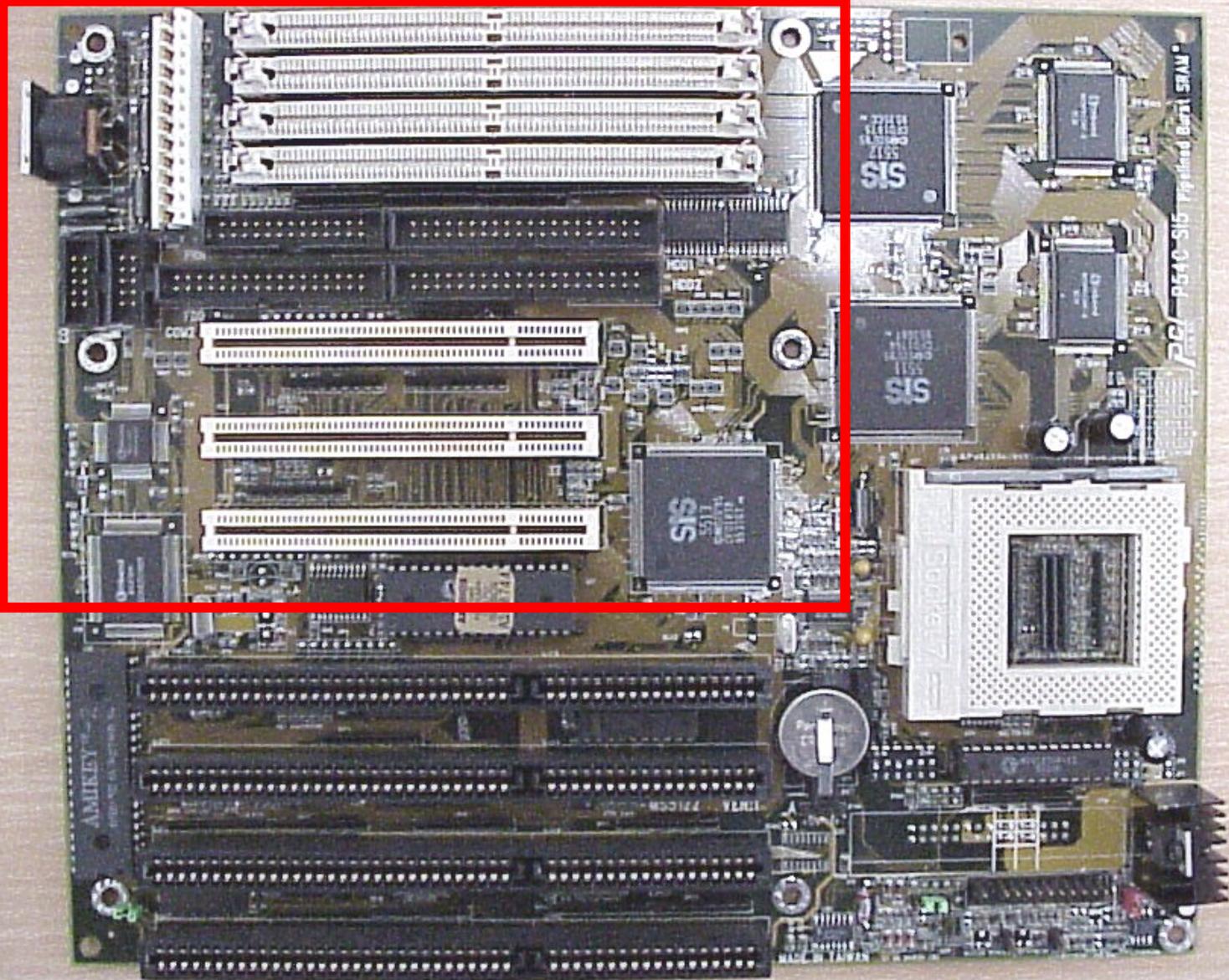


# Ein- und Ausgabebusse

54



# Motherboard



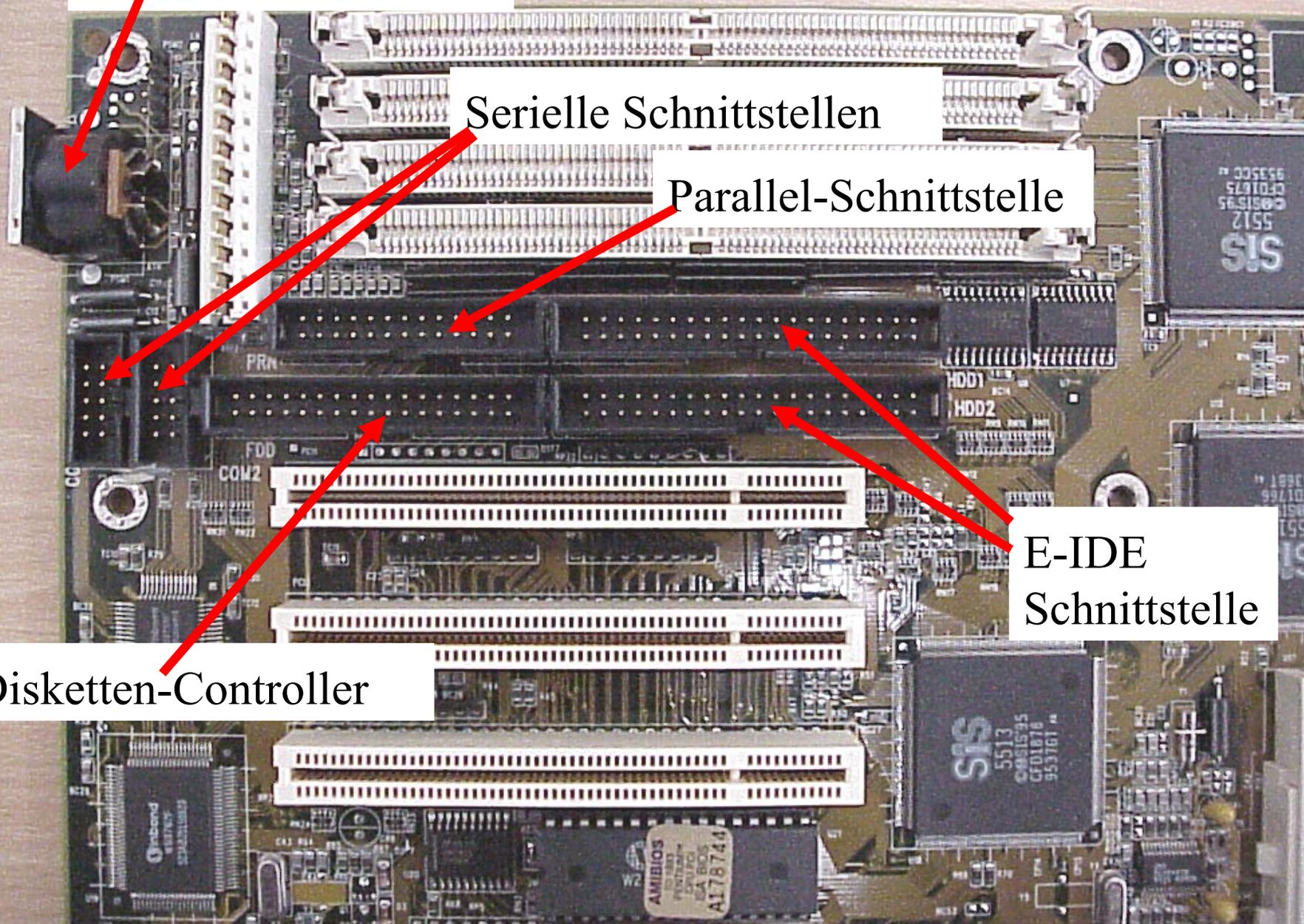
Tastatur Schnittstelle

Serielle Schnittstellen

Parallel-Schnittstelle

E-IDE  
Schnittstelle

Disketten-Controller



# PC von hinten

Parallel-Schnittstelle

Serielle Schnittstellen

Soundkarte

Monitor  
Netzwerk

# Peripherie

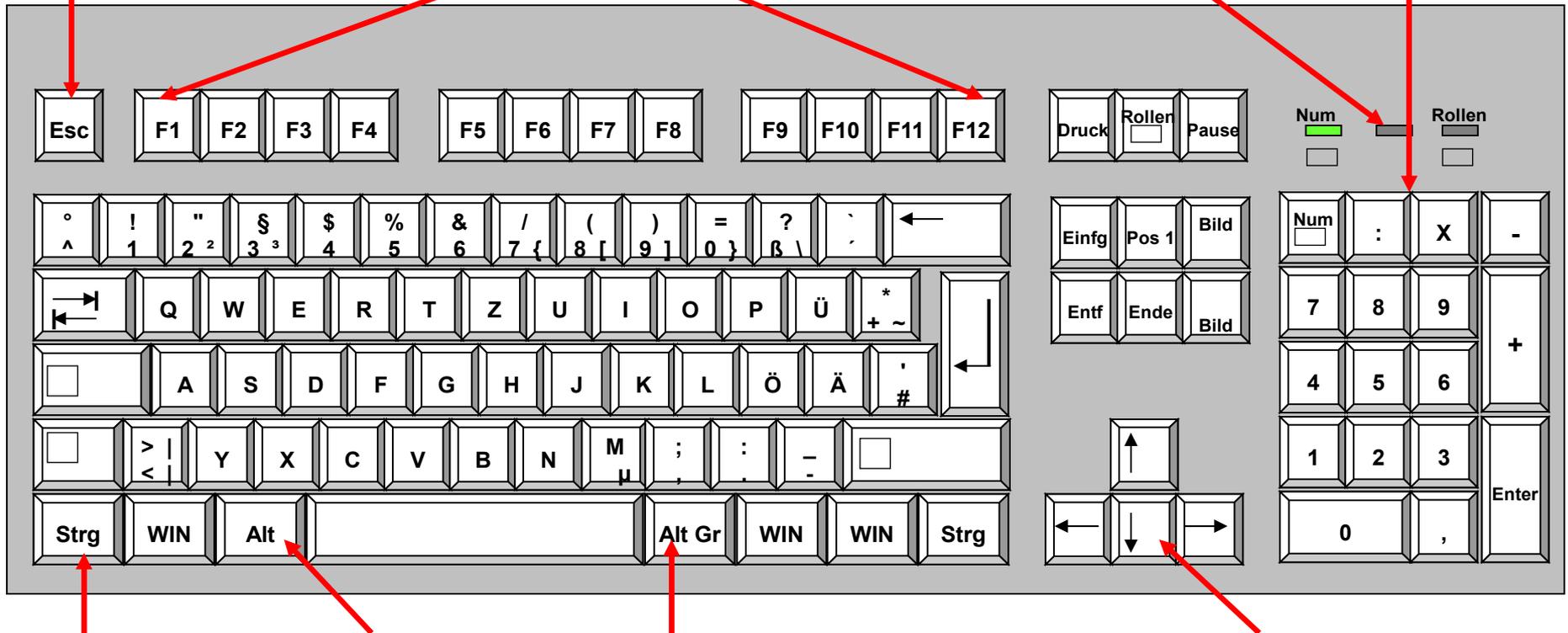
# Die Tastatur

[ESC]-Taste

Funktionstasten

LED-Anzeige

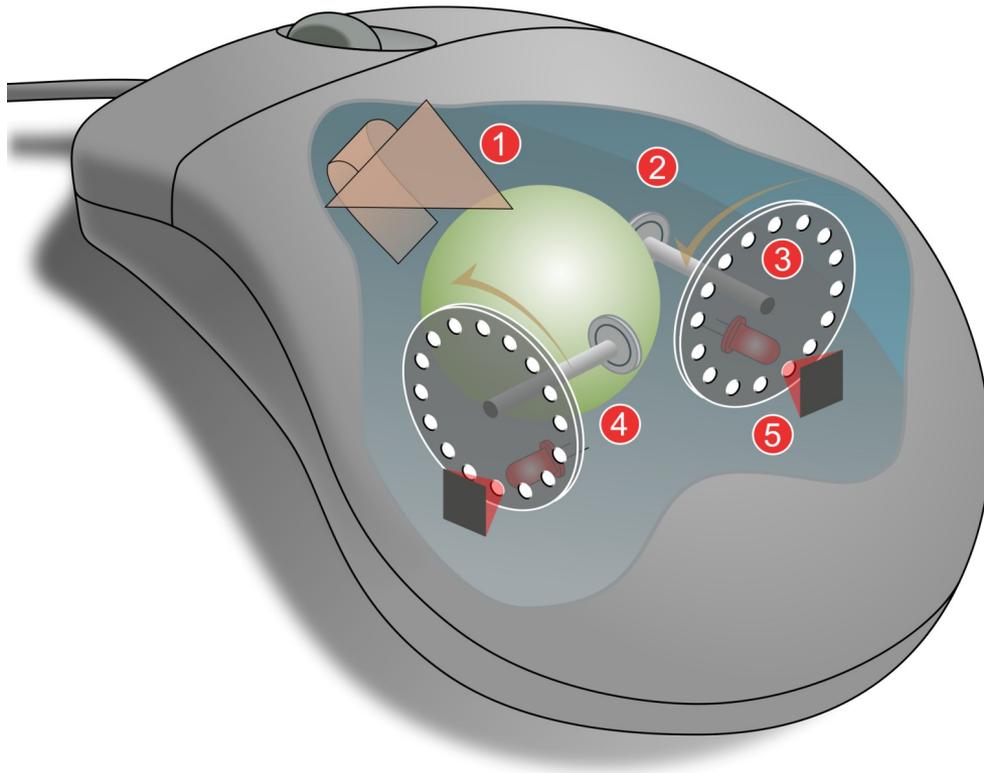
Ziffernblock



[STRG]-Taste [ALT]-Taste [ALTGR]-Taste [Return]-Taste Cursortasten

# Kugelmaus

60



1. Kugel
2. X- Y-Achsen
3. Lochscheiben
4. Infrarot LEDs
5. Sensoren

# Scanner als Eingabegerät

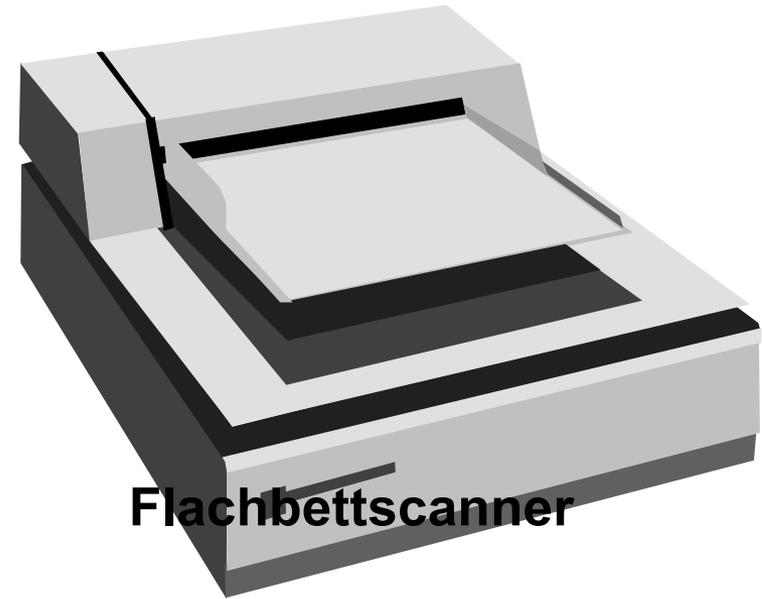
61

## □ Scanner

- Optisches Einlesen von Texten und Grafiken



**Handscanner**



**Flachbettscanner**

# Weitere Eingabegeräte

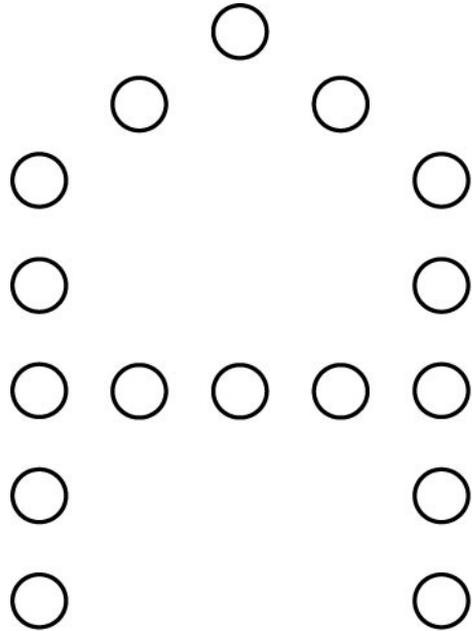
62

- Joystick
  - ▣ Steuerknüppel für Computerspiele
- Lesegriffel
  - ▣ Zum Lesen von Strich- bzw. Barcode (z.B. an Kassen)
- Mikrofon
  - ▣ Aufzeichnung von Sprache und Musik
- Digitale Kamera
  - ▣ Aufnahmen von Bildern und Filmen mit

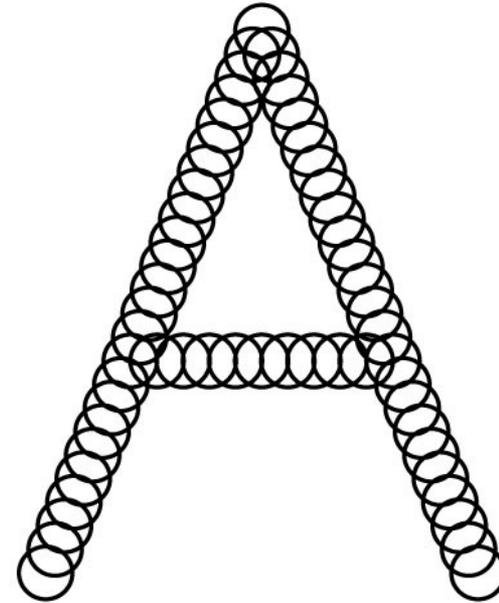
Übertragungsmöglichkeit zum PC

# Nadeldrucker

63



(a)



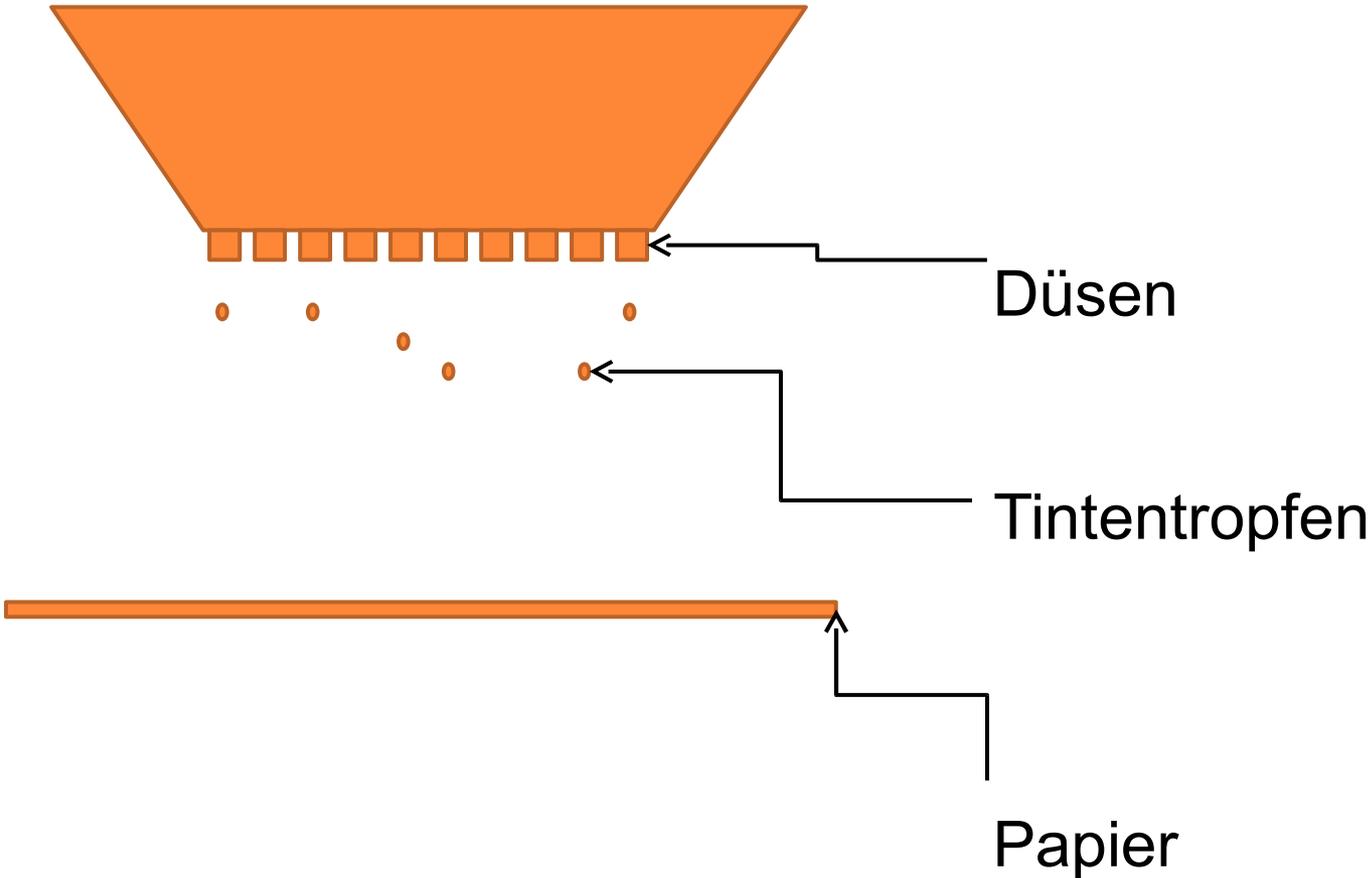
(b)

(a) Der Buchstabe „A“ als 5x7 Matrix

(b) Der Buchstabe „A“ mit überlappenden

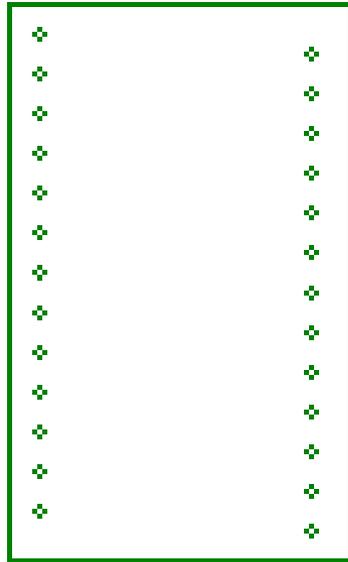
# Tintenstrahldrucker

64



# Tintenstrahldrucker(2)

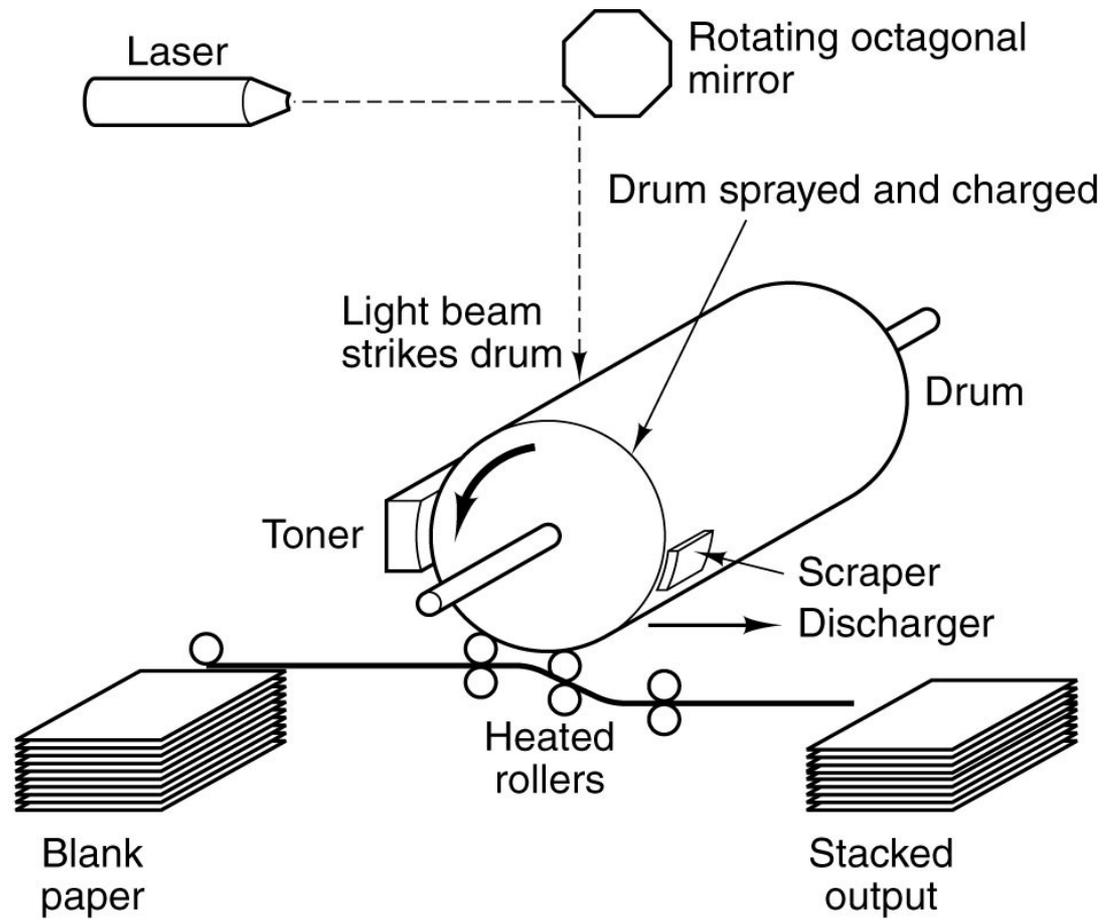
65



Quelle: [de.wikipedia.org](http://de.wikipedia.org)

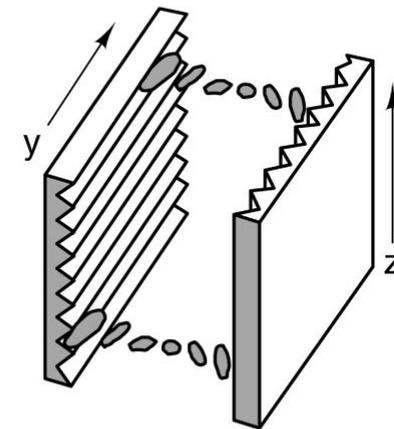
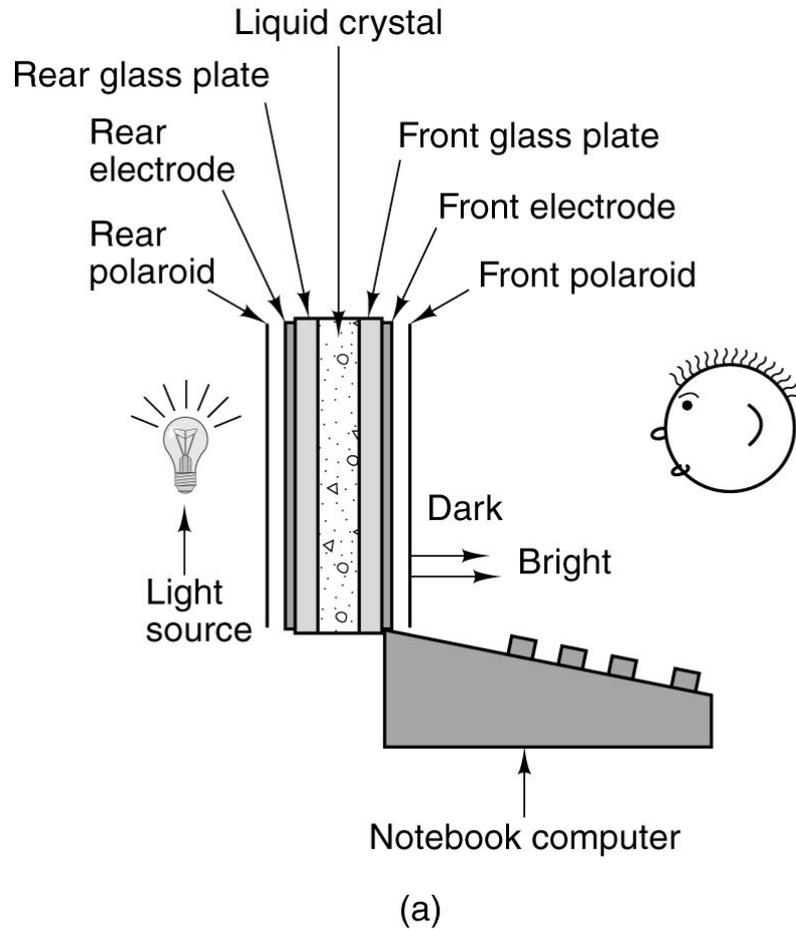
# Laserdrucker

66



# Flachbildmonitore

67



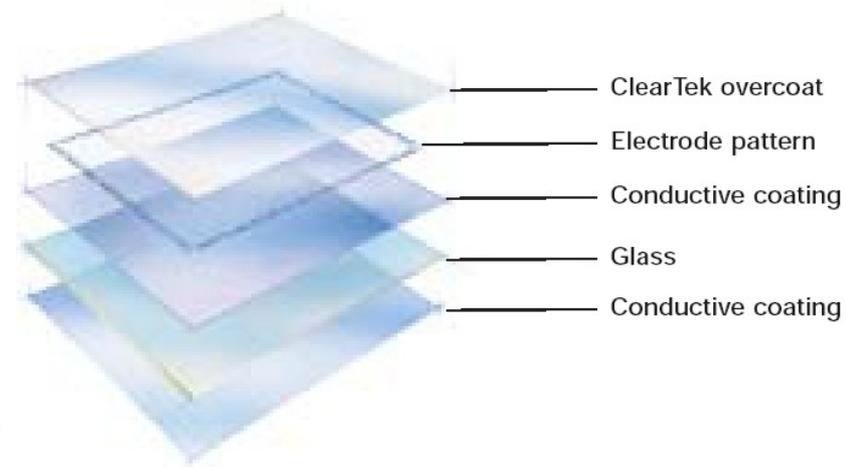
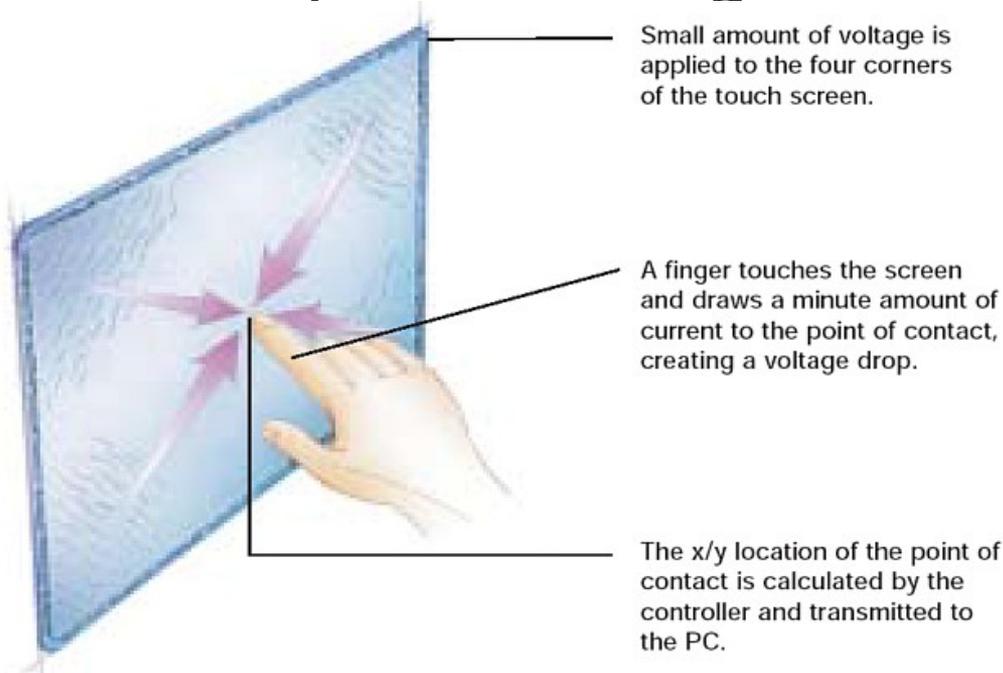
(b)

# Berührungsbildschirm (Touchscreen)

- Technische Realisierung
  - ▣ mit allen Bildschirmarten möglich
    - Kathodenstrahlröhre
    - Flüssigkristallbildschirm
    - Plasmabildschirm
    - Projektion
  - ▣ verschiedene Arten der Berührungsauswertung
    - kapazitiv
    - resistiv
    - ...

# Kapazitiver Touchscreen (1)

## □ Prinzipzeichnung



ClearTek kapazitiver MicroTouch Berührbildschirm von 3M Touch Systems  
[www.3m.com/us/electronics\\_mfg/touch\\_systems/EWCD/Korea/downloads/PDFs/Capacitive-OV.pdf](http://www.3m.com/us/electronics_mfg/touch_systems/EWCD/Korea/downloads/PDFs/Capacitive-OV.pdf)

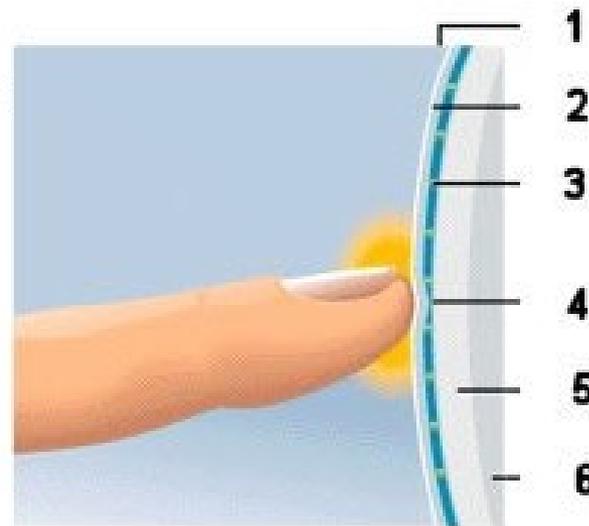
# Kapazitiver Touchscreen (2)

- Funktionsweise
  - leitfähige Beschichtung der Glasscheibe
  - über die Schicht verteilte Spannung
  - Berührung leitet Spannung ab
  - Ermittlung des Berührungspunktes durch Spannungsmessung
- Vorteile
  - empfindlich
  - schnell
  - gut zum Ziehen
- Nachteile
  - nicht mit Handschuhen bedienbar
  - Schicht kann zerkratzt werden

# Resistiver Touchscreen (1)

## □ Prinzipzeichnung

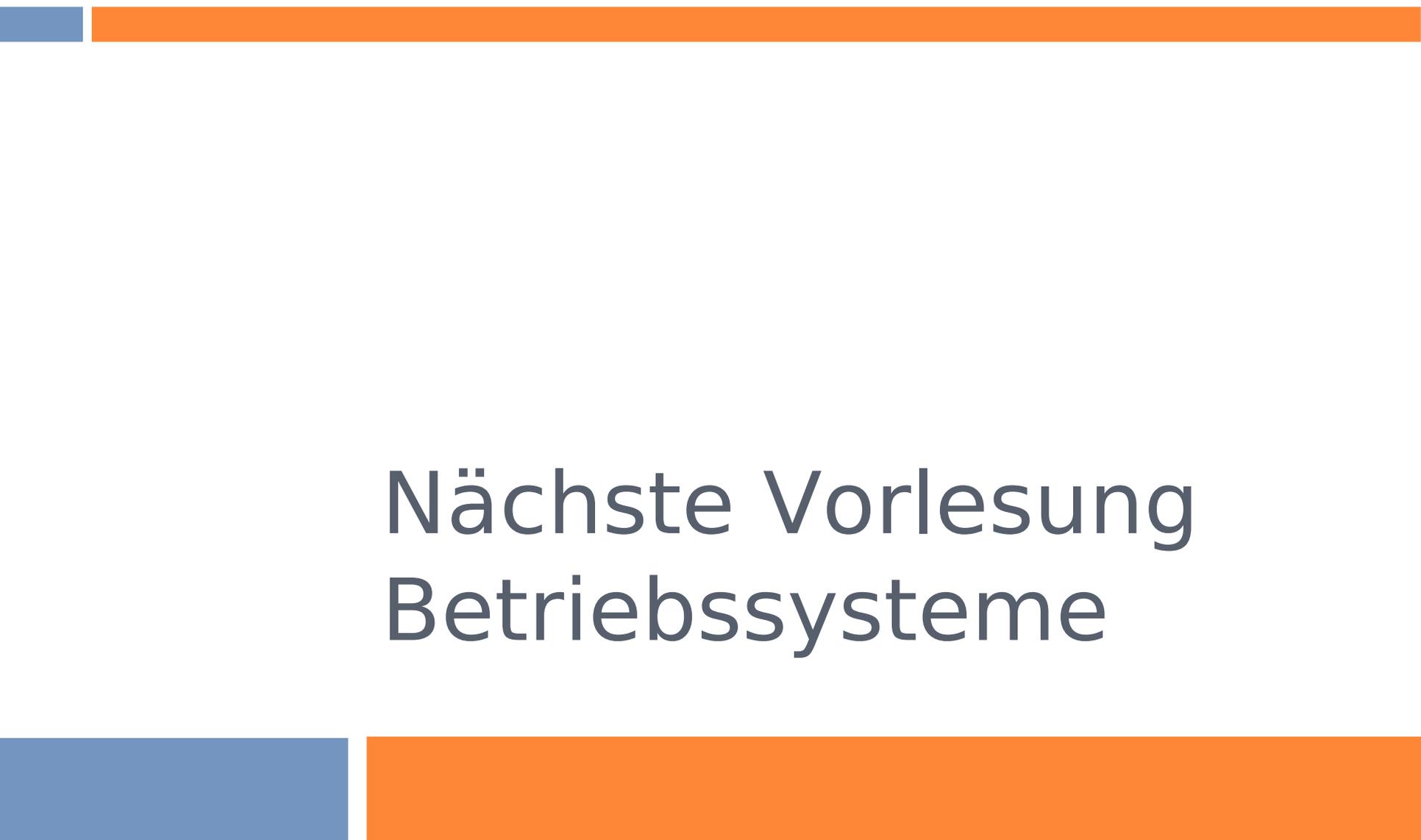
- 1 Widerstandsfähige Beschichtung
- 2 Leitende Oberschicht
- 3 Abstandshalter
- 4 Resistive Beschichtung
- 5 Glasscheibe
- 6 CRT



AccuTouch Berührungsbildschirm von elo Touchsystems  
[www.elotouch.de/Produkte/Touchscreens/AccuTouch/accworks.asp](http://www.elotouch.de/Produkte/Touchscreens/AccuTouch/accworks.asp)

# Resistiver Touchscreen (2)

- Funktionsweise
  - Matrix aus Widerstandsdrähten, leitende Folie, dazwischen nicht leitende Abstandshalter
  - Berührung drückt Folie gegen die Drahtmatrix
  - Ermittlung des Berührungspunktes durch Spannungsmessung
- Vorteile
  - hohe Auflösung
  - funktioniert auch mit Stift oder ähnlichem Gegenstand
  - preisgünstig
  - gewisse haptische Rückmeldung
- Nachteile
  - Folie kann aufgerissen werden



# Nächste Vorlesung Betriebssysteme