



# Technische Grundlagen der Informatik 2

## Rechnerorganisation

### Kapitel 1: Grundlegende Ideen, Technologien und Komponenten

Prof. Dr. Ben Juurlink

Fachgebiet: Architektur eingebetteter System  
Institut für Technische Informatik und Mikroelektronik  
Fak. IV – Elektrotechnik und Informatik

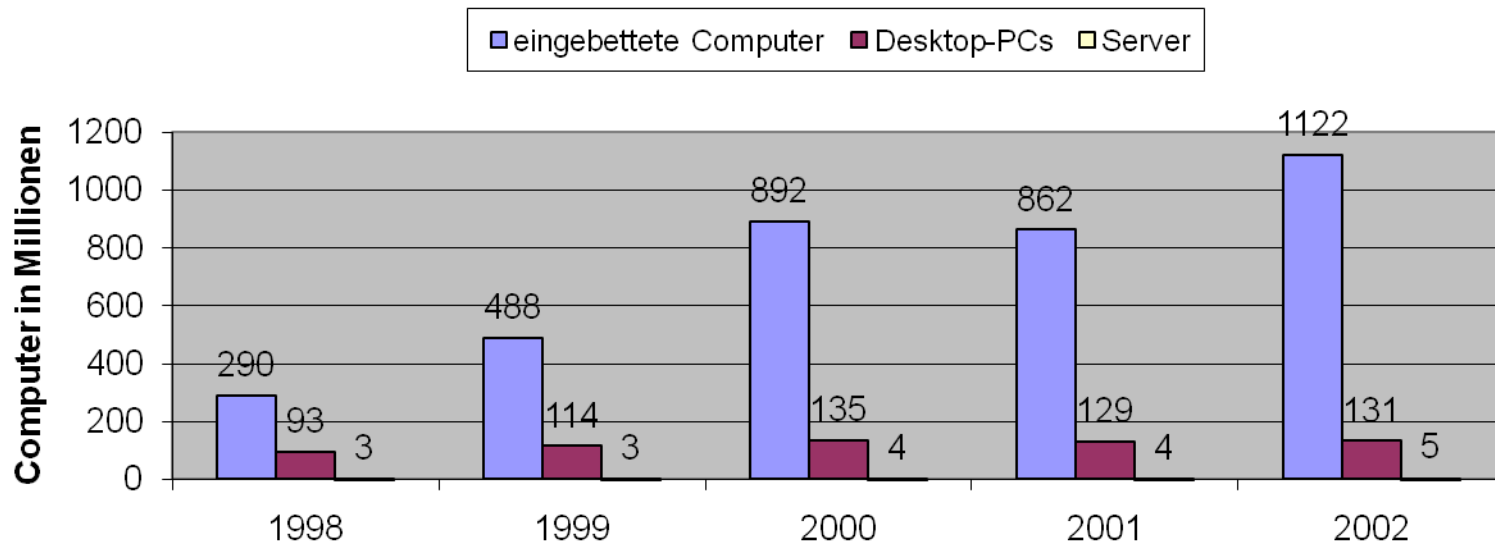
SS 2014

- Die Computerrevolution
- Klassen von Computersystemen
- Hardware- und Software-Ebenen
- Von einer Hochsprache zur Sprache der Hardware
- Klassische Computerkomponente
- Technologien

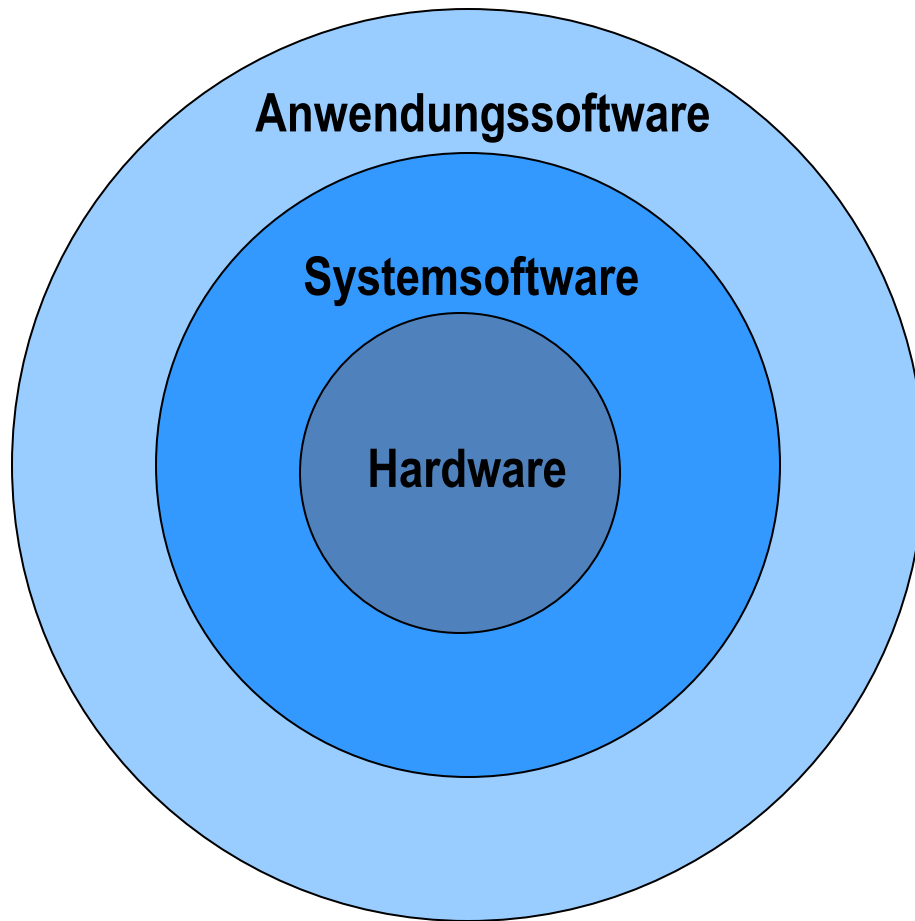
- Computer haben die Welt verändert.
- Vor 10-20 Jahren waren folgende Anwendungen Science Fiction:
  - Laptops
  - World Wide Web (WWW, Web)
  - Handys
  - Digitales Fernsehen/Kamera
  - Forschung am menschlichen Genom
  - ...
- Diese und andere Anwendungen sind nur möglich Dank der Entwicklung der Rechnertechnologie

- **Arbeitsplatzrechner** (*desktop computers*) (100 M/Jahr)
  - der auf/unter/neben ihrem Schreibtisch
  - gute Leistungen zu akzeptablen Preisen
  - führen Software von Drittanbietern aus
- **Server** (5 M/Jahr)
  - bewältigen große Lasten (Anwendungen aus dem technisch-/wissenschaftlichen Bereich, Verarbeitung vieler kleiner Jobs (Web Server))
  - gleiche Technologie wie Arbeitsplatzrechner, jedoch höheres Maß an Erweiterbarkeit
- **Eingebettete Rechner** (*embedded computers*) (1000 M/Jahr)
  - größte Klasse und größte Bandbreite an Anwendungen und Leistungen
  - in Waschmaschinen und KFZs, Handys und PDAs, Videospielsystemen und digitalen TV-Geräten, . . .

Anzahl von verkauften Prozessoren (1998 - 2002)



- Für Arbeitsplatzrechner und Server komplette Rechnersysteme, können jedoch mit mehreren Prozessoren (*cores*) ausgestattet sein.
- Für eingebettete Rechner tatsächliche Anzahl der Prozessoren



- Vereinfachte Darstellung der Hardware (HW) und Software (SW) als hierarchische Ebenen in Form von konzentrischen Kreisen
- Komplexe Anwendungen bestehen häufig aus mehreren SW-Ebenen
- **Systemsoftware**: SW, die allgemein nützliche Dienste bereitstellt (z. B. Betriebssysteme, Compiler und Assembler)

```
void swap(int v[], int k)
{
    int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

Hochsprachenprogramm (in C)  
[*High-level language Program*]

C Compiler

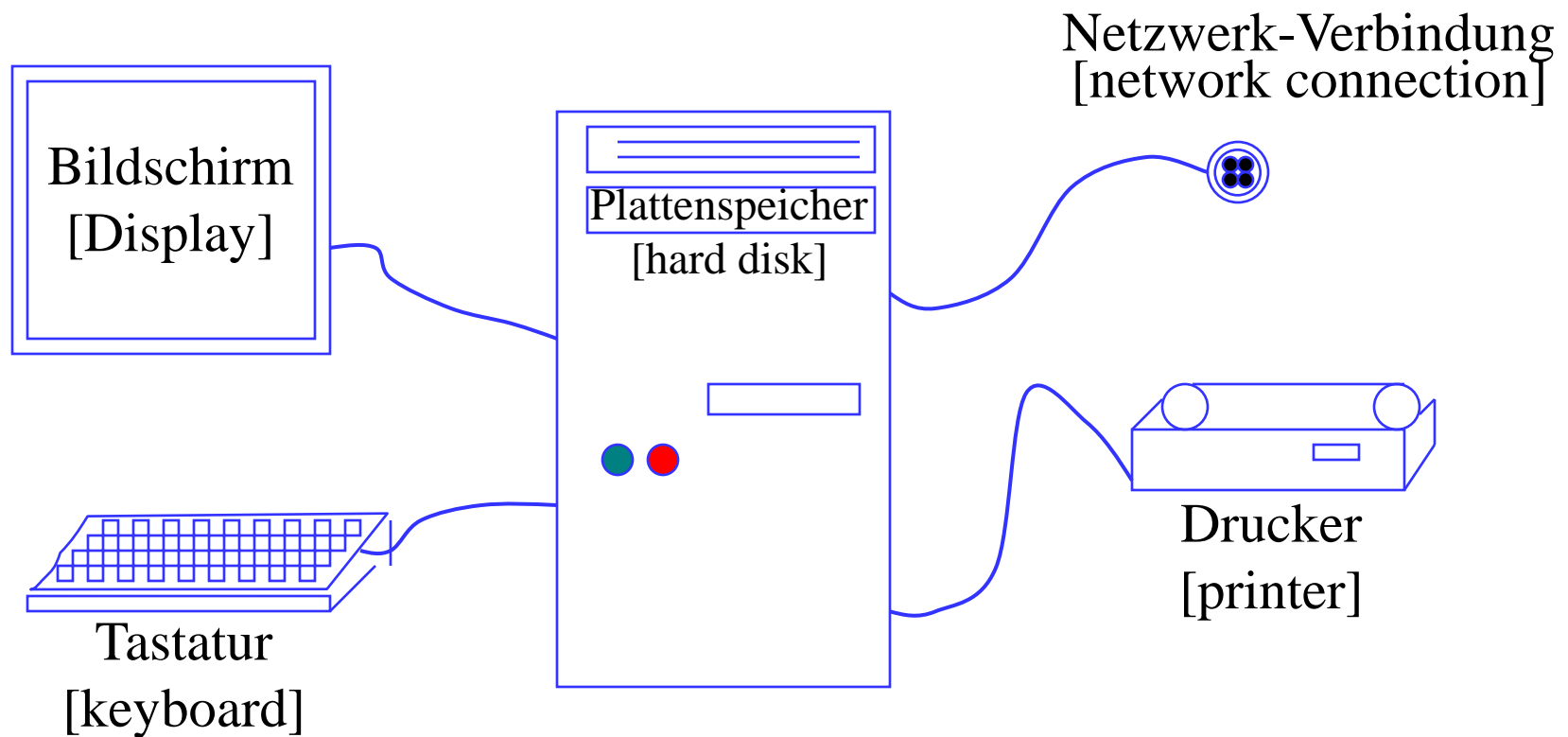
```
swap:
    muli    $2, $5, 4
    add     $2, $4, $2
    lw      $15, 0($2)
    lw      $16, 4($2)
    sw      $16, 0($2)
    sw      $15, 4($2)
    jr      $31
```

Assemblerprogramm für MIPS  
[*Assembly language program*]

Assembler

```
00000000101000010000000000011000
00000000100011100001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
101011001111001000000000000000...
```

Binärer Maschinencode für MIPS  
[*Binary machine language program*]





Die 5 klassischen Komponente eines Computers:

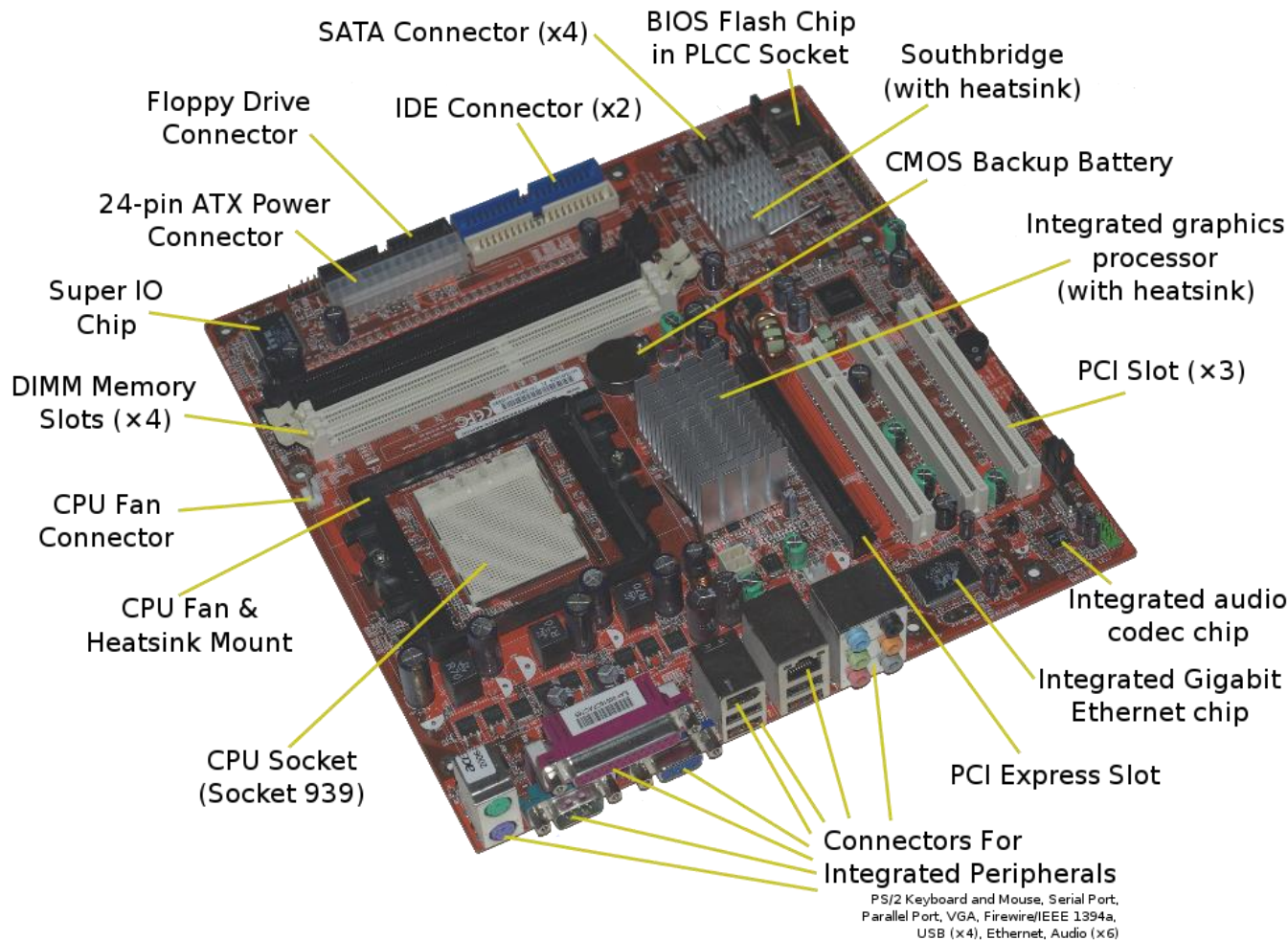
- Eingabegeräte (*input devices*) (Maus, Tastatur, ...)
- Ausgabegeräte (*output devices*) (Bildschirm, Drucker, ...)
- Speicher:
  - Intern: DRAM, SRAM [flüchtig]
  - Extern: Festplatte, CD, Diskettenlaufwerk [nicht flüchtig]
- Datenpfad
  - führt Operationen aus
  - die **Muskeln** eines Prozessors
- Leitwerk / Steuerung (*control*)
  - sendet Signale, welche die Operationen bestimmen
  - **Gehirn** eines Prozessors

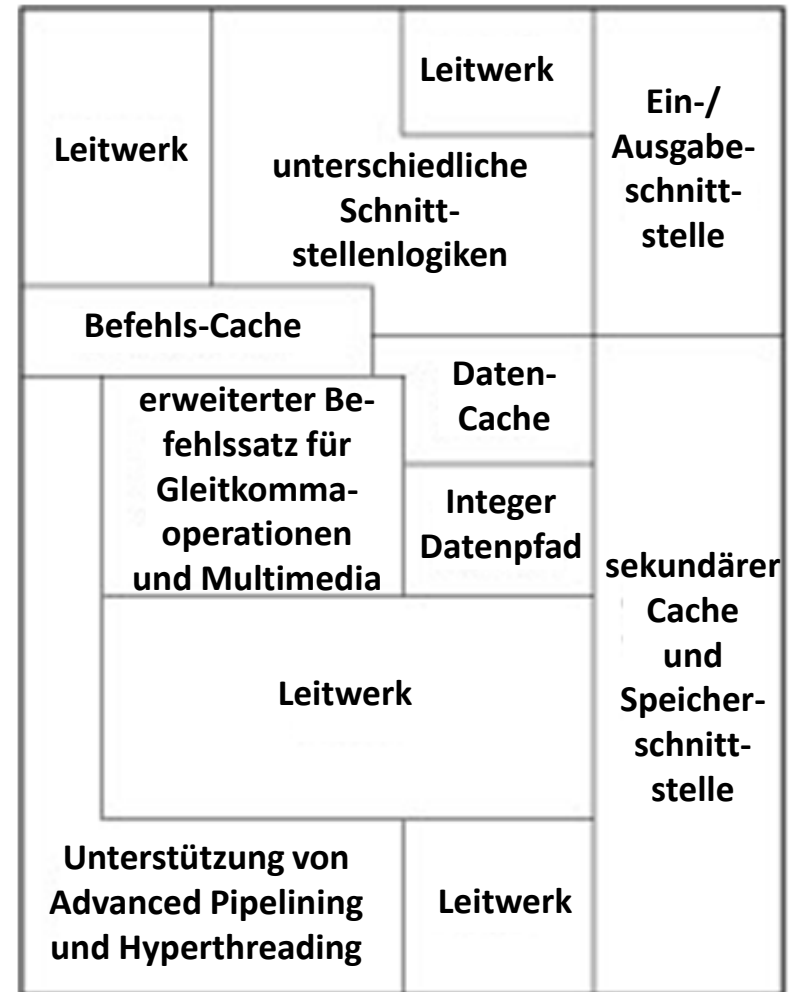
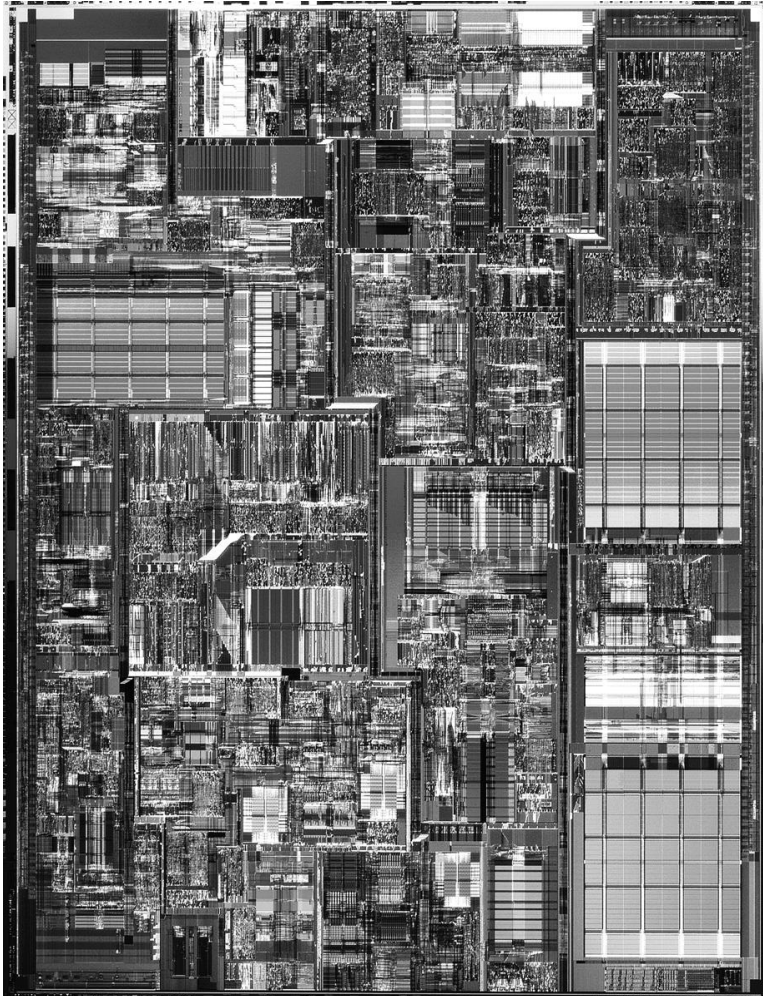
Prozessor  
oder  
*Central  
Processing  
Unit (CPU)*



## Hauptplatine (motherboard)

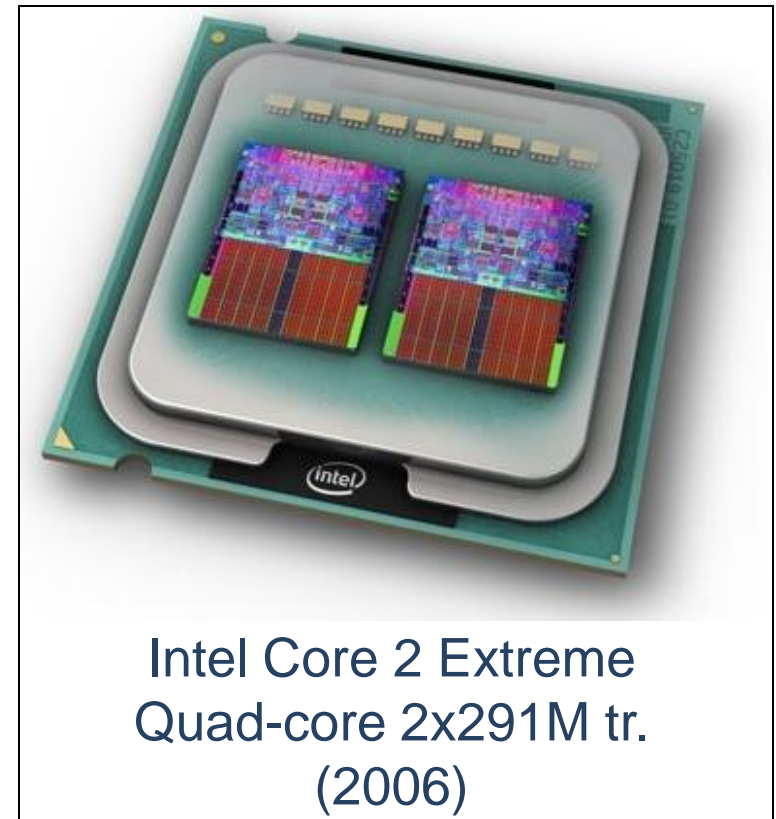
- DIMM = Dual In-line Memory Module
- PCI = Peripheral Component Interconnect
- IDE = Integrated Drive Electronics (bus for hard disk drives)
- SATA = Serial Advanced Technology Attachment



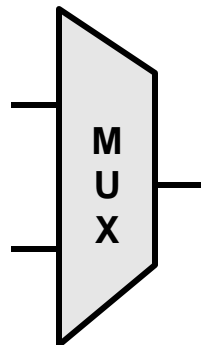




- Unser Fokus: Der Prozessor (CPU = *Central Processing Unit*)
  - Datenpfad
  - Steuerwerk
- Realisiert mit Billionen von
- Unmöglich zu verstehen, wenn man die Transistoren einzeln betrachtet
- Wir benötigen **Abstraktionen** auf viele Ebenen.



- **Abstraktion:** Ein Modell, bei dem Details der unteren Ebenen eines (Computer-)Systems vorübergehend ausgeblendet werden, um die Entwicklung komplexer Systeme zu erleichtern
- Beispiele:
  - Schaltkreise (MUX statt Schaltkreis)
  - Befehlssatzarchitektur (digitaler Rechner „=“ der Satz von Befehlen, den er ausführen kann)
  - Programmabstraktion (Funktionen, Klassen, Objekte)
  - Datenabstraktion (Sätze, Queues)





- Eine sehr wichtige Abstraktion!
- Schnittstelle zwischen HW und SW
- Standardisierung von Befehle, Bitfolgen, u. s. w.
- Vorteil:
  - verschiedene Implementierungen einer Architektur möglich
- Nachteil:
  - verhindert manchmal neue Innovationen

***Wahr oder nicht wahr?***

*Binäre Kompatibilität ist äußerst wichtig*

- Computer haben die Welt verändert
- Die 3 Klassen von Computersystemen sind *Arbeitsplatzrechner*, *Server* & *eingebettete Rechner*
  - eingebettete Rechner höchste in Anzahl, Arbeitsplatzrechner in \$\$\$
- 5 klassischen Komponenten eines Computers sind:
  - Eingabegeräte (*input devices*), Ausgabegeräte (*output devices*)  
Speicher, Datenpfad, Leitwerk/Steuerung (*control*)
- Unser Fokus ist der Prozessor (CPU = *Central Processing Unit*)
- Brauchen *Abstraktion* um den Prozessor verstehen zu können