

1. Computer Abstraktionen u. Technologie

- **Gebiet ändert sich schnell**

- Relais → Vakuum-Röhre → Transistor → IC

- VLSI → μ P → SoC → NoC

- Moore'sches Gesetz

- alle 1,5 Jahre verdoppeln sich

- Speicherkapazität

- Prozessorgeschwindigkeit

Systems on Chip

Networks on Chip

durch Fortschritte in Technologie
und Organisation (Architektur)

Was ist ein Computer?

- **Komponenten**

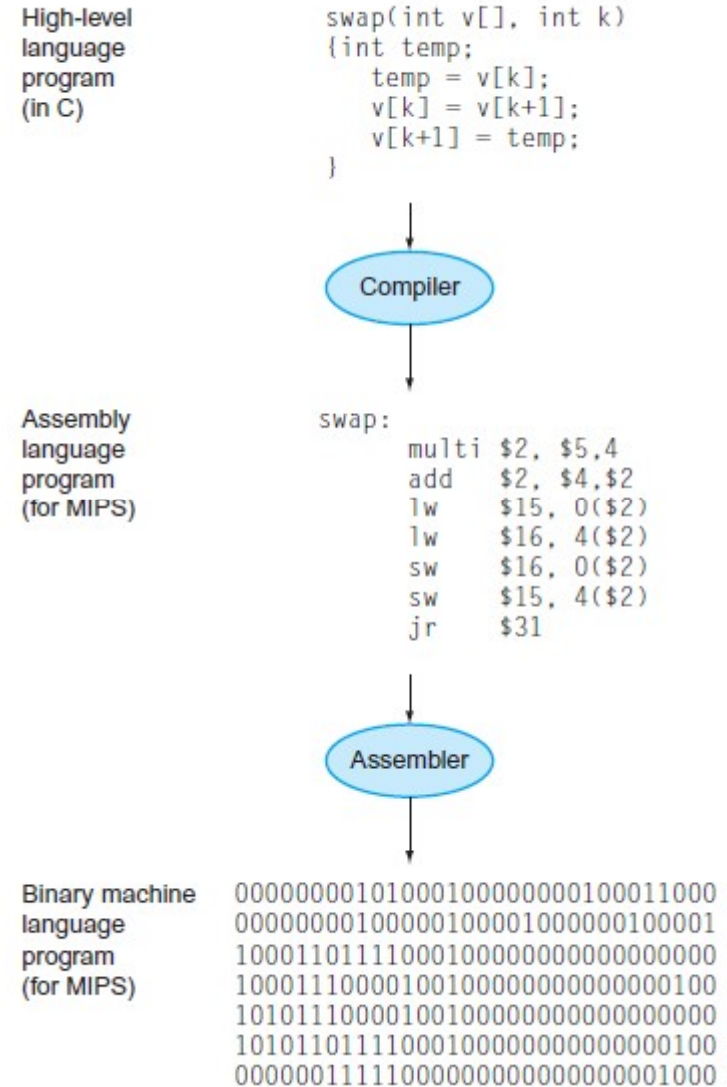
- Prozessor
- Eingabe (Maus, Tastatur, ...)
- Ausgabe (Bildschirm, Drucker, ...)
- Speicher (Festplatte, DRAM, SRAM, CD, ...)
- Netzwerk

- **Hauptaugenmerk: Prozessor**

- Datenpfad und Steuerung
 - Milliarden von Transistoren
 - unmöglich zu verstehen, indem man sich jeden einzelnen Transistor ansieht

Abstraktion

- Abstraktion
 - unterdrückt unnötige Details
 - lässt uns daher besser mit der Komplexität umgehen
- Höhere Programmiersprachen
 - Leichter zu verstehen als Assembler
 - Und Assembler ist leichter zu verstehen als binäre Maschinensprache
- Welche verwenden wir im Prozessor-Design?
 - ISA (Instruction Set Architecture)



ISA (*Instruction Set Architecture*)

- **Sehr wichtige Abstraktion**
 - definiert den Befehlssatz eines Computers
 - Schnittstelle zwischen Hardware und low-level Software
 - standardisiert Instruktionen, Struktur der Maschinen Sprache, Funktionsweise der Register, etc.
 - Vorteil
 - verschiedene Implementierungen können dieselbe Architektur realisieren
 - Nachteil
 - Binärkompatibilität
 - Neue Rechner können die alten Programme weiter ausführen.
 - Ist sehr wichtig, sagt der Markt.
 - verhindert manchmal den Fortschritt durch Innovationen
- **Moderne ISA's**
 - 80x86/Pentium/Core2/K6, PowerPC, MIPS, SPARC, ARM, RISC-V, ...

Regeln für gutes Computer-Design

- **Beachte Moore's Law**

- Alle 12-18 Monate verdoppeln sich die Leistung und Speicherplatz
- Wenn man ein Design beginnt, sollte man das berücksichtigen.
 - Design-Prozess kann sich über Jahre hinziehen.
 - Man sollte Technologie antizipieren, für die man am Ende designt.

- **Benutze Abstraktionen um das Design zu vereinfachen**

- Größere Produktivität durch Verstecken von Details.

- **Mache den Normalfall schnell**

- Effektiver, als seltene Fälle zu beschleunigen.
- Ist oft sogar einfacher zu realisieren, als die seltenen, komplizierten Fälle.

- **Höhere Leistung durch Parallelisierung**

- 10 Eier in einem großen Topf geht schneller als alle nacheinander kochen.

Regeln für gutes Computer-Design (2)

- **Höhere Leistung durch Pipelining**
 - Ist auch eine Form von Parallelisierung.
 - Sehr verbreitet in Computern.
- **Höhere Leistung durch Vorhersage**
 - „Manchmal ist es besser nachher um Verzeihung zu bitten, als vorher um Erlaubnis zu fragen.“
 - Im Durchschnitt kann es schneller sein zu raten und anzufangen, als darauf zu warten, bis völlig klar ist, was zu tun ist.
- **Speicher-Hierarchie**
 - Speicher soll schnell, groß und billig sein.
 - Geht nicht alles gleichzeitig, aber eine Speicher-Hierarchie kann helfen.
- **Zuverlässigkeit durch Redundanz**
 - Hardware, um Fehler erkennen.
 - Redundante Komponenten, um im Fehlerfall Aufgaben zu übernehmen.