

Praktische Informatik 1

Aufbau und Funktionsweise eines Computers

Thomas Röfer

- Hardware und Software
- von-Neumann-Architektur
- Schichtenmodell der Software
- Zahlssysteme
- Repräsentation von Daten im Computer
- Benutzung des Repositories

Erhalt eines SBLN

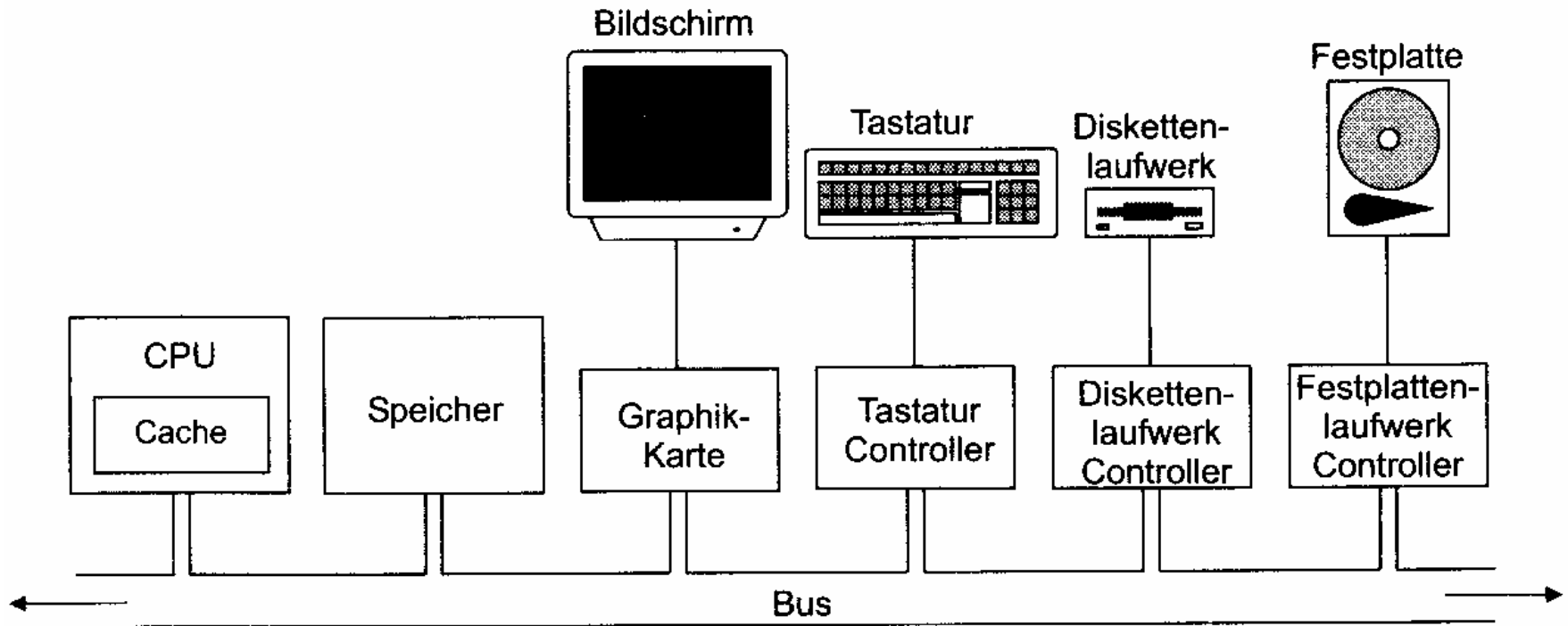
- Bedingungen

- Summe der erreichten Punkte der 12 Übungsblätter muss mindestens 50% der erreichbaren Punkte sein
- Präsentation der eigenen Lösung im Tutorium
- Fachgespräch bestehen

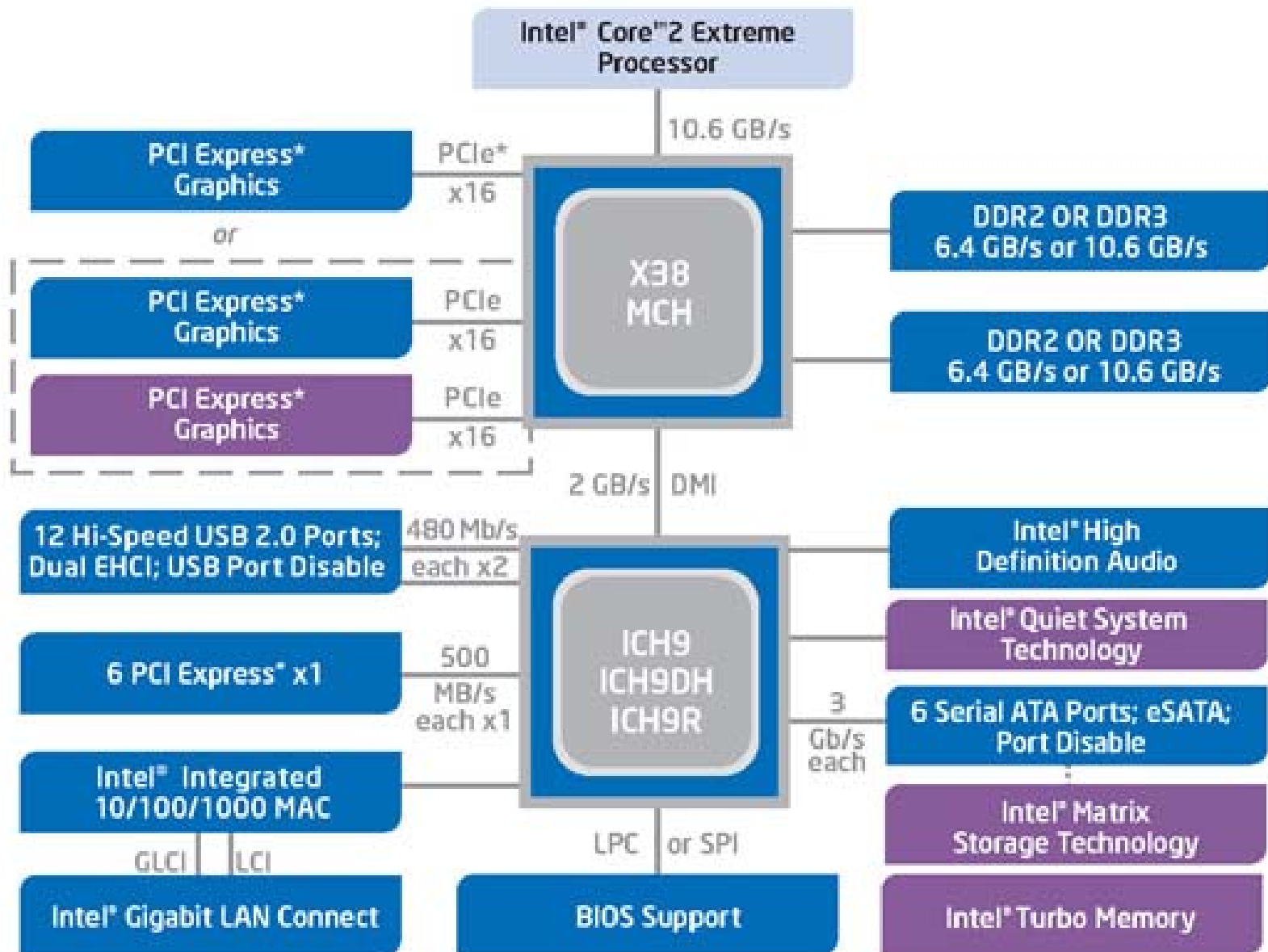
- Benotung

- $\geq 95\% \rightarrow 1,0$ $\geq 90\% \rightarrow 1,3$
- $\geq 85\% \rightarrow 1,7$ $\geq 80\% \rightarrow 2,0$ $\geq 75\% \rightarrow 2,3$
- $\geq 70\% \rightarrow 2,7$ $\geq 65\% \rightarrow 3,0$ $\geq 60\% \rightarrow 3,3$
- $\geq 55\% \rightarrow 3,7$ $\geq 50\% \rightarrow 4,0$
- $< 50\% \rightarrow$ nicht bestanden

Hardware

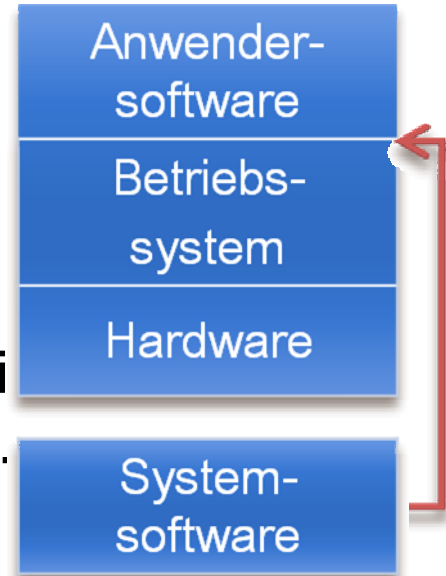


- Anbindung über Bus
- Peripherie hat eigene Steuereinheit (Controller)



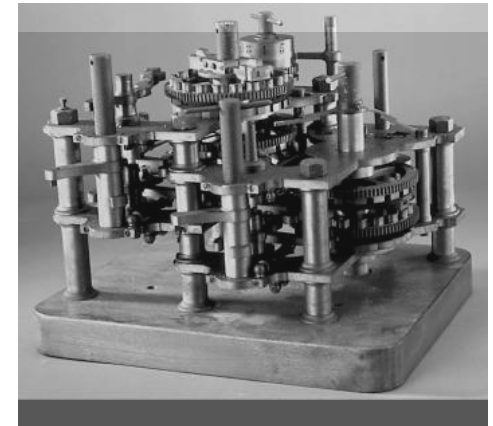
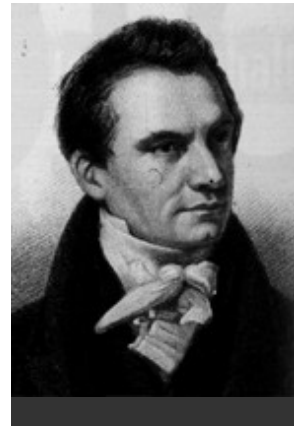
Software

- Betriebssystem
 - Isoliert Anwendersoftware von der Hardware
 - Verwaltet Ressourcen des Rechners
 - Firmware/BIOS
 - MSDOS, MS Windows, Linux, MacOS, Solaris
 - PalmOS, MS Windows Mobile, SymbianOS, ...
- Systemsoftware
 - Lösung von Aufgaben im Rechner
 - z.B. Entwicklungswerkzeuge: Compiler, Linker, Debugger
- Anwendersoftware
 - Lösung von Problemen der externen Welt der Anwender
 - z.B. Textverarbeitung, Tabellenkalkulation, ..., Spiele
- Zuordnung zunehmend schwierig



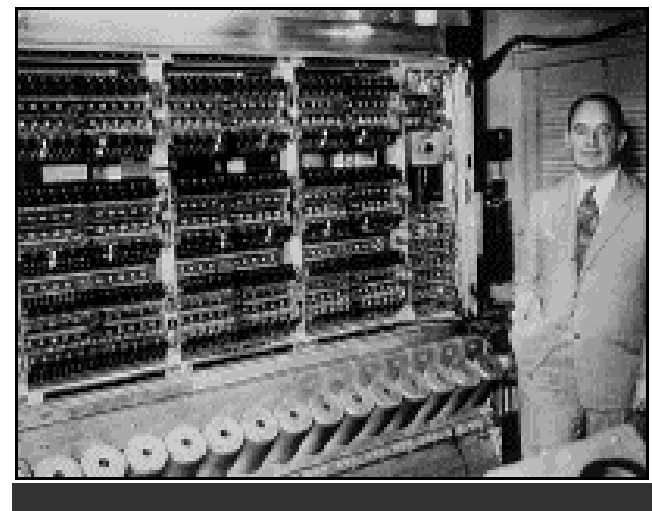
Grundlagen Analytical Engine

- Von Charles Babbage (1834)
- Aufbau
 - Prozessoreinheit
 - Speichereinheit
 - Programmsteuerung
- Programme
 - sind Sequenzen von Befehlen,
 - werden nacheinander vom Prozessor verarbeitet,
 - werden auf Daten im Speicher angewendet



Grundlagen

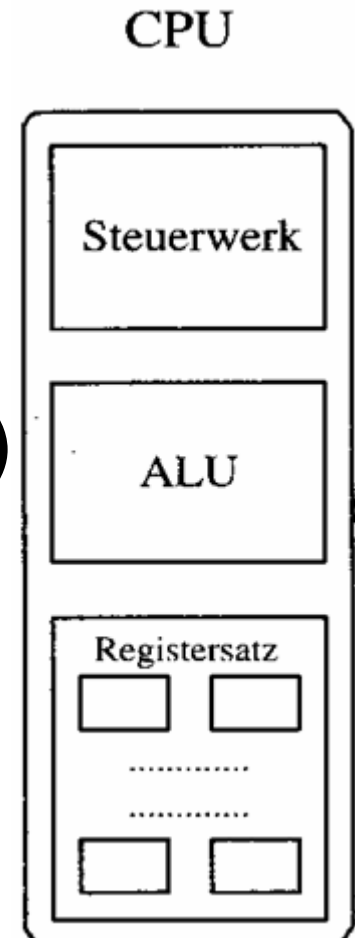
- Erste Rechner
 - Zuse Z3 (1941)
 - Mark 1 (1944)
 - ENIAC (1946)
 - turingmächtig
- von-Neumann-Architektur (1945)
 - *“First Draft of a Report on the EDVAC”*



von Neumann Architektur

Prozessor (CPU)

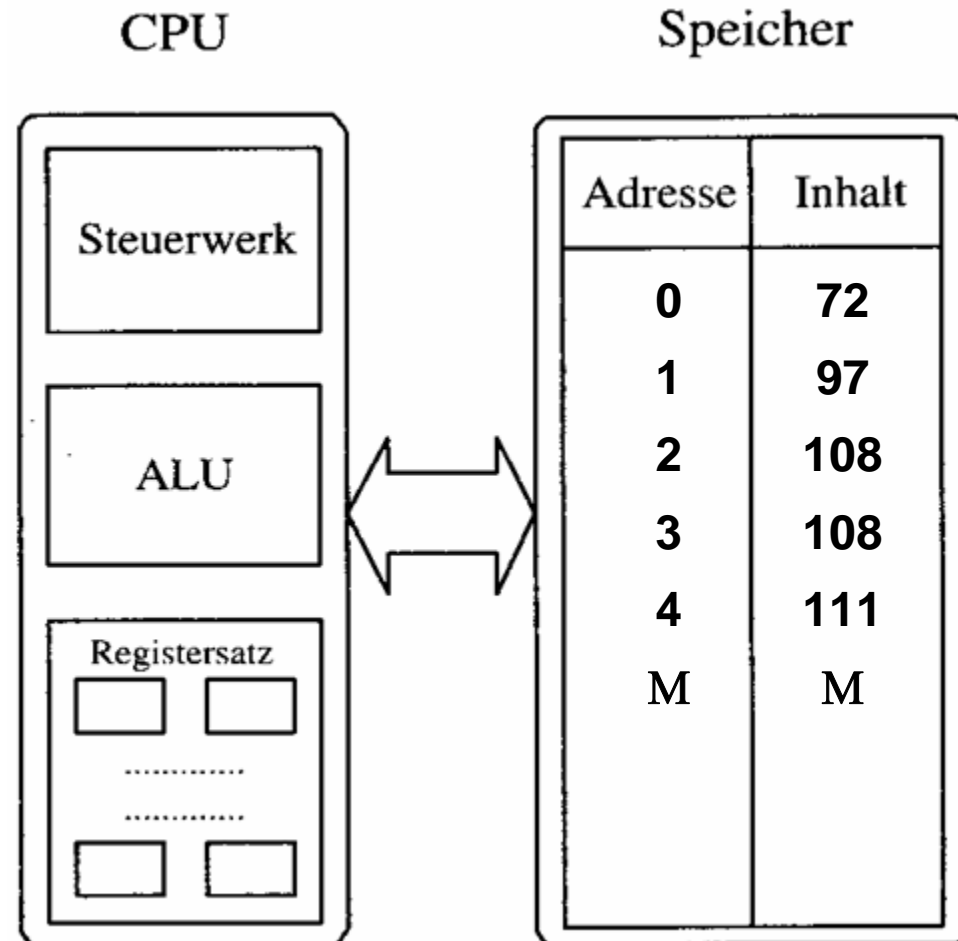
- Steuerwerk
 - Führt Programme aus
- Arithmetisch-logische Einheit (ALU)
 - Rechnet
- Registersatz
 - Speichert Daten im Prozessor



von Neumann Architektur

Speicher

- Wahlfreier Zugriff (RAM)
- Jede Speicherzelle hat
 - eine Adresse
 - einen Inhalt
- Beispiel: ein PC mit 512 MB hat 536.870.912 Speicherzellen
- Programm und Daten im selben Speicher



Bits und Bytes

- 1 Bit kann 2 Zustände einnehmen: 0 oder 1
- Mehrere Bits: 2^{Anzahl} Zustände
- Bei n Bits
 - Bit 0 = least significant bit (LSB)
 - Bit $n-1$ = most significant bit (MSB)
- 1 Byte = 8 Bits
 - 2^8 Zustände = 256 Zustände, d.h. 0 ... 255
 - Speicherzellen sind 1 Byte groß
- 1 Nibble = 4 Bits

Bit 2	Bit 1	Bit 0	
0	0	0	Zustand 0
0	0	1	Zustand 1
0	1	0	Zustand 2
0	1	1	Zustand 3
1	0	0	Zustand 4
1	0	1	Zustand 5
1	1	0	Zustand 6
1	1	1	Zustand 7

Bits und Bytes

1 Wort = ?


- Anzahl Bits ergibt sich aus der Breite des Datenbusses
- Aktuell: IA32: 32 Bit, Intel 64/AMD64: 64 Bit
- Adresse eines Wortes ist immer die Adresse des ersten Bytes des Wortes
 - Zugriff auf Wortgrenzen heißt *aligned* (ausgerichtet)
- Little Endian
 - LSB ist Teil des ersten Bytes
- Big Endian
 - LSB ist Teil des letzten Bytes
- Beispiel: die Zahl 3
 - Little Endian: 03 00 00 00
 - Big Endian: 00 00 00 03

Adresse	Little end.	Big endian
n	Bits 0-7	Bits 24-31
n+1	Bits 8-15	Bits 16-23
n+2	Bits 16-23	Bits 8-15
n+3	Bits 24-31	Bits 0-7

Größenangaben

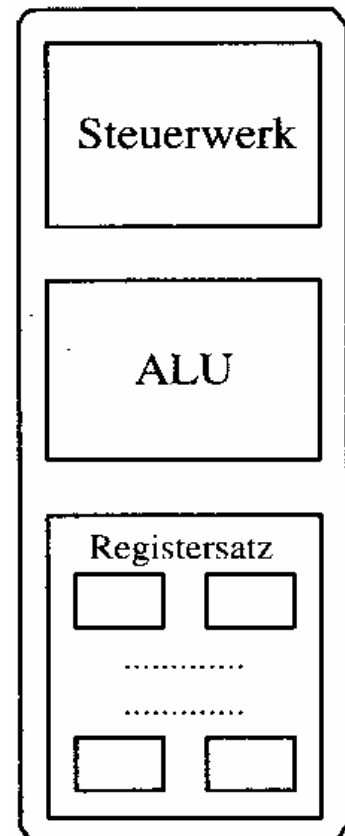
- Üblich in der Informatik
 - Kilobyte, KB: $2^{10} = 1.024$ Bytes
 - Megabyte, MB: $2^{20} = 1.048.576$ Bytes
 - Gigabyte, GB: $2^{30} = 1.073.741.824$ Bytes
 - Terabyte, TB: $2^{40} = 1.099.511.627.776$ Bytes
 - Eigentlich KiB, MiB, GiB, TiB
- Im normierten Sprachgebrauch
 - Kilobyte, KB: $10^3 = 1.000$ Bytes
 - Megabyte, MB: $10^6 = 1.000.000$ Bytes
 - Gigabyte, GB: $10^9 = 1.000.000.000$ Bytes
 - Terabyte, TB: $10^{12} = 1.000.000.000.000$ Bytes

Zustände und ihre Interpretationen

- Bits im Speicher repräsentieren Zustände
- Ihre Bedeutung hängt davon ab, wie sie interpretiert werden
- 4 Bytes (z.B. 65 117 116 111) können z.B.
 - als ganze Zahl gedeutet werden (integer, z.B. 1.869.903.169)
 - als kurze Gleitkommazahl (float, z.B. $7,56561 \cdot 10^{28}$)
 - als Folge von vier Zeichen (string, z.B. „Auto“)
 - als Folge von Befehlen für den Prozessor
 - INC CX
 - JNZ +116
 - DB 111
 - als Farbe (vom Grafikchip, z.B. )
- In Programmiersprachen wird die Interpretation durch Datentypen festgelegt

Programmausführung

- Programme sind in Maschinensprache kodiert
- Prozessortyp-abhängig
 - z.B. PowerPC vs. IA32
- Spezielles Register: Befehlszähler
- Fundamentaler Instruktionszyklus einer CPU



Programmausführung

Instruktionszyklus einer CPU

- Fetch
 - Hole Befehl, dessen Adresse im Befehlszähler steht, in das Instruktionsregister
- Increment
 - Erhöhe den Befehlszähler, damit er danach auf die nächste auszuführende Instruktion verweist
- Decode
 - Dekodiere die Instruktion, damit klar wird, was zu tun ist
- Fetch Operands
 - Falls nötig, hole die Operanden aus den im Befehl bezeichneten Stellen im Speicher
- Execute
 - Führt die Instruktion aus, ggf. durch die ALU
 - Bei einem Sprung wird ein neuer Wert in den Befehlszähler geschrieben
- Loop
 - Beginne von vorne

Programmausführung

- Typische Befehle
 - laden (Speicher → Register)
 - schreiben (Register → Speicher)
 - verknüpfen von Registern (z.B. addieren)
 - (bedingter) Sprung
- Takt
 - Befehle werden getaktet ausgeführt
 - Anzahl Takte/Befehl hängt vom Befehl ab
 - Beschleunigung durch Fließbandverarbeitung (Pipelining)

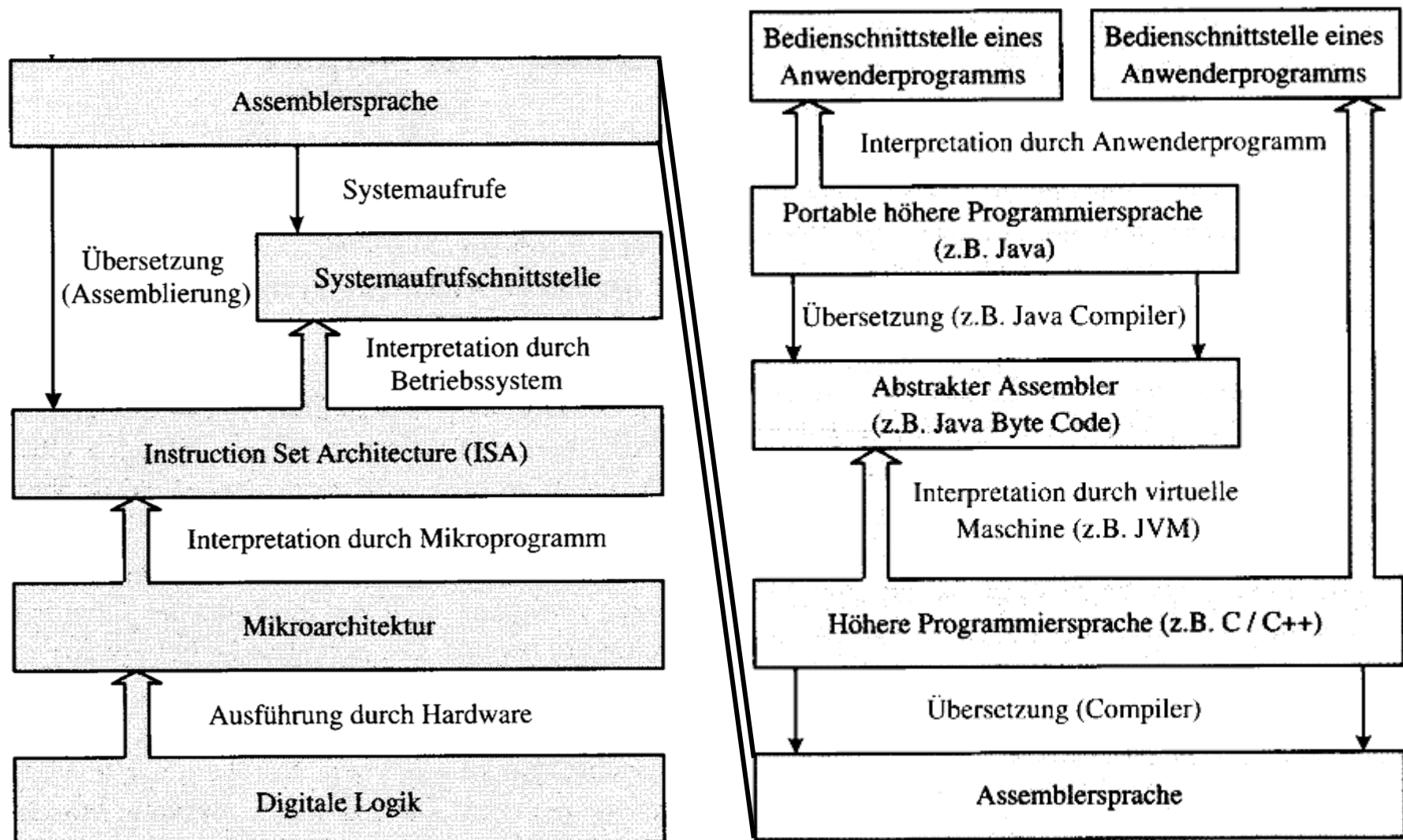
von-Neumannscher-Flaschenhals

- Prozessor greift sequenziell auf Speicher zu
 - Durch einen Kanal (Daten-Bus)
- Prozessor (viel) schneller als Speicher
 - Dadurch muss Prozessor oft auf Speicher warten
- Gebräuchliche Lösungsansätze
 - viele CPU-Register
 - schnelle Zwischenspeicher (Cache)

Moores Gesetz

- Gordon Moore (Intel):
 - Die Anzahl der Transistoren pro Chip verdoppelt sich alle 18 Monate
- Folgen
 - Verdopplung der Speichergröße alle 18 Monate
 - Verdopplung der Geschwindigkeit alle 18 Monate
- Beispiel
 - 1981: IBM PC, 4,77 MHz, 64 KB RAM, 160 KB Floppy
 - 2007
 - Intel Core 2 Duo, 2 · 3 GHz (nur $2^{10,3}$ · 4,77MHz, aber Core 2 braucht weniger Takte pro Instruktion)
 - 2 GB RAM (2^{15} · 64 KB)
 - 320 GB Festplatte (2^{21} · 160 KB)

Schichtenmodell der Software



Betriebssystem

- Verwaltet Ressourcen eines Rechners
 - Systemaufrufchnittstellen (oft APIs – Application Programming Interfaces)
- Verwaltet Prozesse und ihre Ressourcen
 - Prozessorzeit (oft in Zeitscheiben)
 - Speicher
 - Dateien
 - Kommunikationspfade
- Enthält Gerätetreiber
 - bzw. bietet Schnittstellen für Gerätetreiber
- Bietet Ein- und Ausgabeströme
 - Dateisystem
 - TCP/IP



Programmierparadigmen

- Imperativ (z.B. Pascal, C, C++, Java)
 - Abfolge von Befehlen verändert den Zustand des Programms (d.h. Wert der Variablen bzw. Inhalt des Speichers)
- Deklarativ (Haskell, Prolog)
 - Programmausführung durch Anwendung von Regeln
 - Es gibt keinen sich ändernden Zustand
- Funktionsgetrieben (z.B. Pascal, C)
 - Programm wird anhand der Funktionalität strukturiert
- Objektorientiert (z.B. C++, Java)
 - Programm wird anhand der verwalteten Daten strukturiert

```
int fac(int n) {  
    int r = 1;  
    for (int i = 1; i <= n; ++i) {  
        r = r * i;  
    }  
    return r;  
}
```

```
fac 0 = 1  
fac (n + 1) = (n + 1) * fac n
```

Programmiersprachen

Vom Quelltext zur Ausführung

