

**Prof. Dr. Dieter Kranzlmüller**

Dr. Nils gentschen Felde

Dr. Karl Furlinger

Stephan Reiter

Christian Straube

## Grundlagen der technischen Informatik

Workshop im Rahmen des Informatik-Probestudiums 2012

- **Tag 1 (Fr):**
  - Grundlagen der technischen Informatik mit Übungen
  - Einführung in die Bedienung von Linux
  - Übungen
- **Tag 2 (Mo):**
  - Übungen
  - HPC und paralleles Rechnen
- **Tag 3 (Di): Grundlagen der Computergrafik**
  - POV-Ray
  - Ausblick und Demo
- **Tag 4 (Mi): Grundlagen der IT-Sicherheit**
  - Grundlagen der IT-Sicherheit mit Übungen
  - Demonstration "Passwörter unter Windows knacken"

## In der Informatik geht es oft darum, ...

- Informationen/Daten zu **speichern**:
  - Wie lange? → kurzzeitig (flüchtig) vs. dauerhaft (persistent)
  - Wie viel? → Datenmenge/-größe
  - Wie schnell zugreifbar? → Zugriffszeiten, Latenzen
  - Wie teuer? → Kosten für die Speicherung
  - Wie robust? → Robustheit gegen Ausfälle, Verschleiß, ...
  - Wie sicher? → Schutz vor unberechtigtem Lesen/Manipulieren/Löschen
- Informationen/Daten zu **verarbeiten**:
  - Berechnungen
  - Algorithmen
  - Programme
- Informationen/Daten zu **transportieren**:
  - Art und Ausdehnung des Netzes
  - Verwendetes Medium → Kabel (Glasfaser, Kupfer, ...), Luft





- Nach Speicherdauer:
  - Persistent
  - Flüchtig
- Nach Zugriffsart:
  - Sequenziell
  - Wahlfrei/adressierbar
- Nach Nutzungsmodus:
  - Lesen und schreiben
  - Nur lesen
- Nach verwendeter Technologie:
  - Optische Speicher
  - Magnetische Speicher
  - Halbleiterspeicher

**Aufgabe:**

Ein großes Blatt Papier ist 0,1 Millimeter dick. Wie dick wird es, wenn man es 42 mal jeweils in der Mitte faltet?

Wie hängt die Antwort mit dem Thema Datenrepräsentation/Datenspeicherung zusammen?

- Repräsentation von Zahlen
- Repräsentation von Buchstaben und Zeichen
- Repräsentation von Bildern
- Repräsentation von Videos
- Repräsentation von Audiodaten

- Beispiel: Wie lässt sich die Zahl 25 mit Hilfe von Bits (also nur durch 0 und 1) darstellen?
- Grundlegende Idee: Transformation der Dezimalzahl in eine Dualzahl
- Dezimalsystem:
  - Es gibt 10 verschiedene Ziffern (0 bis 9), daher Basis 10
  - $25_{10} = 5 \cdot 10^0 + 2 \cdot 10^1 + 0 \cdot 10^2 + \dots$
  - Führende Nullen können weggelassen werden:  $0000025 = 25$
- Dualsystem:
  - Es gibt 2 verschiedene Ziffern (0 und 1), daher Basis 2
  - $25_{10} = 1 \cdot 2^0 + 0 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4 + 0 \cdot 2^5 + \dots = 11001_2$

- Wie lassen sich negative Zahlen repräsentieren?
  - Beispiel: -25
  - Ansätze:
    - Ein Bit wird für das Vorzeichen reserviert: 0 = positiv, 1 = negativ
    - Spezielle Komplementdarstellung, bei der das Bitmuster der positiven Zahl invertiert wird → ermöglicht effiziente Addition
- Wie lassen sich Kommazahlen repräsentieren?
  - Beispiel: 25,75
  - Ansätze:
    - Feste Anzahl von Bits vor und nach dem Komma → Nachteile?
    - Gleitkommazahlen (*floating point numbers*)

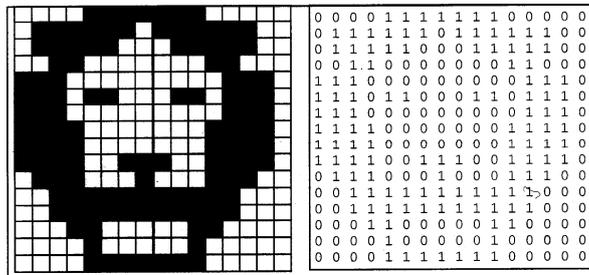
- Beispiel: Wie lässt sich der Text „Hallo Welt!“ mit Hilfe von Bits darstellen?
- Grundlegende Idee: Jedes Zeichen wird durch eine bestimmte, festgelegte Abfolge von 0 und 1 dargestellt (vgl. Morsealphabet)
- Problem: Keine mathematische Beziehung zwischen Buchstaben und deren Binärdarstellung (wie bei Zahlen)  
→ Daher: Vereinbarung von Standards

**Aufgabe:**

Wie viele Bits werden benötigt, um das gesamte Alphabet, alle Umlaute, alle Ziffern und alle Zeichen einer Standard-Tastatur darstellen zu können?

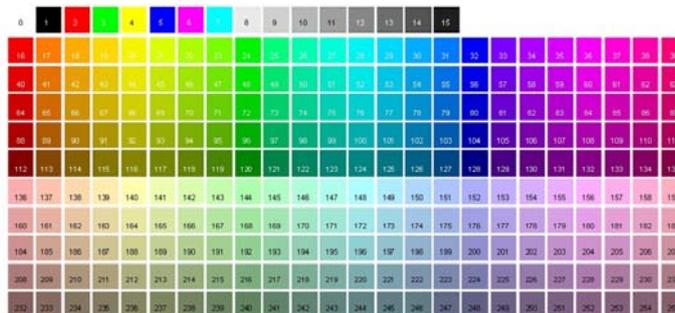
- Wichtige Standards zur Kodierung von Zeichen:
  - ASCII (American Standard Code for Information Interchange): 128 Zeichen, 7 Bits
  - ISO-8859-1 bis ISO-8859-15: Verschiedene 8-Bit-Zeichensätze (256 Zeichen) als Erweiterung zu ASCII
  - Unicode: Kodierung aller Zeichen (unterschiedlicher Zeichensysteme) mit 16 und mehr Bits (versionsabhängig)
  - UTF-8 (Unicode Transformation Format): Am weitesten verbreitete Kodierung der Unicode-Zeichen (vor allem im Web)
- Worin besteht der Unterschied zwischen der Zahl 25 und der Zeichenkette „25“?

- Frage: Wie lässt sich ein Bild als Bitfolge darstellen?
- Grundlegende Idee (Rasterbilder, schwarz-weiß):
  - Ein „Gitter“ wird über das Bild gelegt.
  - In jedem Feld wird entschieden:
    - Überwiegt weiß? → 0
    - Überwiegt schwarz? → 1

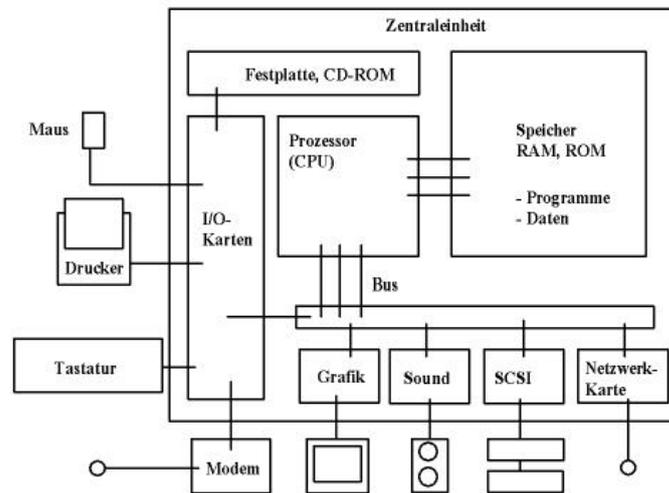


- Farbbilder:

- Qualitätsbestimmende Merkmale:
  - Auflösung (Anzahl Pixel/Bildpunkte)
  - Farbtiefe
- Komprimierung

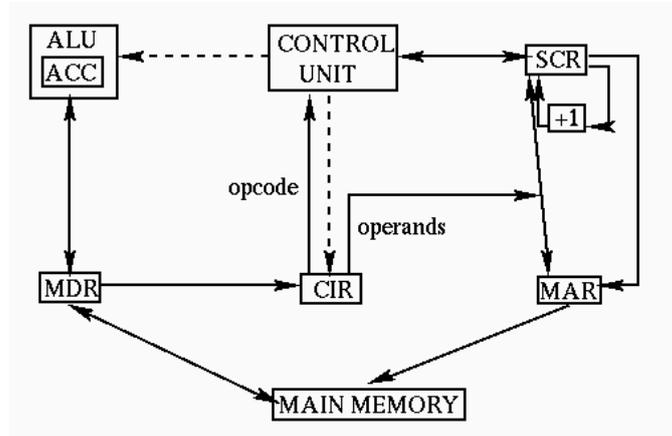


- Wir können jetzt Informationen/Daten als Bits darstellen. Das reicht aber noch nicht.
- Frage: Wie lassen sich diese Daten rechnergestützt verarbeiten?
  - Einfache mathematische Operationen auf Zahlen, z.B.:
    - Addition
    - Subtraktion
    - Multiplikation
    - Division
    - Vergleiche (<, >, =)
  - Logische Operationen auf Bitmustern



- Verarbeiten = Processing
- Prozessor ist das „Gehirn“ eines Computers
- Befehlssatz eines Prozessors umfasst i.d.R. arithmetische und logische Funktionen, die durch die **Arithmetisch-logische Einheit (ALU)** des Prozessors realisiert werden

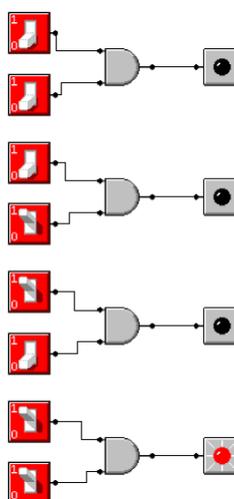




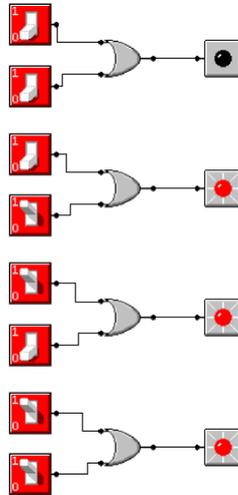
- ALU verarbeitet Bits → Für die Verarbeitung von Binärdaten wird eine spezielle Algebra benötigt, die Operationen auf der Menge  $\{0,1\}$  definiert.
- Wichtigste Boolesche Funktionen:
  - Konjunktion (AND)
  - Disjunktion (OR)
  - Negation (NOT)
  - (Implikation)
  - (Äquivalenz)

p	q	$p \cdot q$	$p + q$	$\neg p$
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

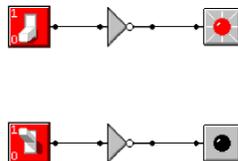
- **Schaltglied (Gatter): Realisierung einer Booleschen Operation**
  - Technische Umsetzung (Hardware) mittels Transistoren
  - Transistor: Nicht-mechanisches Halbleiterbauelement, verstärkt elektrische Signale
  - Moderne Prozessoren können über 1 Milliarde Transistoren auf einem einzigen Halbleiter-Chip besitzen.
  - Menge der verwendeten Gatter muss funktional vollständig sein.
- **Schaltnetz: Schaltungstechnische Realisierung einer Booleschen Funktion**
  - Mehrere Gatter werden über Leitungen miteinander verbunden (vernetzt), um mathematische Funktionen zu realisieren.
  - Wichtig bei Funktionen: Wie viele Argumente?  
→ Pro Argument ein Eingang ins Schaltnetz



p	q	p · q
0	0	0
0	1	0
1	0	0
1	1	1



p	q	p + q
0	0	0
0	1	1
1	0	1
1	1	1



p	-p
0	1
1	0

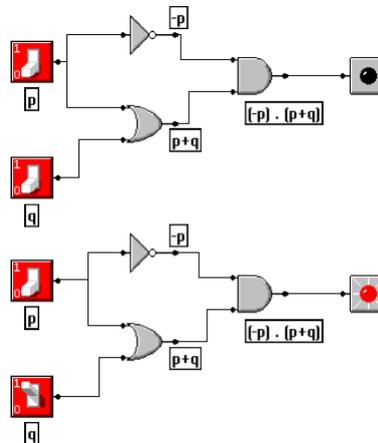
- Schaltglied (Gatter): Realisierung einer Booleschen Operation
  - Technische Umsetzung (Hardware) mittels Transistoren
  - Transistor: Nicht-mechanisches Halbleiterbauelement, verstärkt elektrische Signale
  - Moderne Prozessoren können über 1 Milliarde Transistoren auf einem einzigen Halbleiter-Chip besitzen.
- Schaltnetz: Schaltungstechnische Realisierung einer Booleschen Funktion

**Aufgabe:**

Gegeben ist die Boolesche Funktion  $f(p,q) = (\neg p) \cdot (p+q)$ .

- Wie sieht die Wahrheitstabelle zu dieser Funktion aus?
- Wie sieht das Schaltnetz aus, das diese Funktion realisiert?

Realisierung der Funktion  $f(p,q) = (\neg p) \cdot (p+q)$ :



- Addition (von Binärzahlen) wichtiger arithmetischer Befehl jedes CPU-Befehlssatzes
- Frage: Wie bildet eine CPU-ALU die Summe aus zwei n-Bit-Zahlen?
- Input:
  - Operand 1: Summand 1
  - Operand 2: Summand 2
- Output: Summe

**Aufgabe:**

Entwerfen Sie ein Addierwerk (Schaltnetz), das zwei positive 4-Bit-Zahlen addiert.

**Strukturiertes Vorgehen:**

- Wie funktioniert Addition? → Formalisierung eines „intuitiven“ Verfahrens
- Wie lässt sich die Addition von zwei einzelnen Bits realisieren? Was muss berücksichtigt werden?
- Wie lassen sich Zahlen addieren, die jeweils aus mehr als einem Bit bestehen?

