max planck institut
informatik

Ideen und Konzepte der Informatik

Rechner

Kurt Mehlhorn

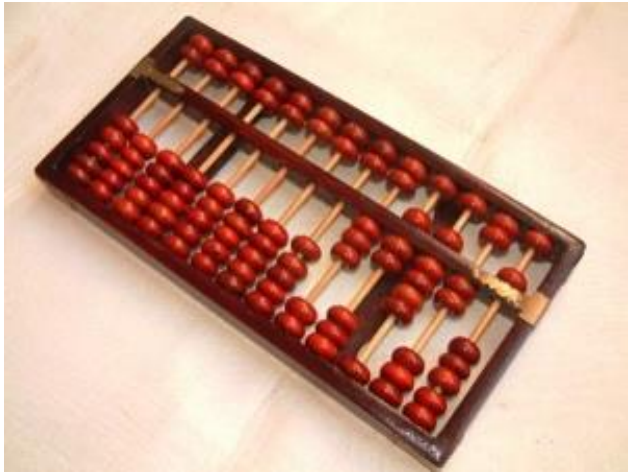
Übersicht

1. Geschichtlicher Rückblick
2. Wie funktionieren Computer?
   - Der Von-Neumann-Rechner
3. Universalität von Rechnern (Basis für Siegeszug der Informatik)
   - Was bedeutet Universalität ?
4. Laufzeit, Rechenzeit
5. Alan Turing
   - Person / Turingmaschine / Turingthese
6. Quantencomputer

Geschichtlicher Rückblick

- Charles Babbage (1791 – 1871) + Ada Lovelace: Maschine zur Auswertung von Polynomen, Logarithmentafeln, nie fertig
- Alan Turing (1936) entwirft einfachen universellen Rechner als Gedankenexperiment.
- Konrad Zuse (1941): erster funktionierender programmierbarer Rechner
- Mauchly und Presper (43/44) bauen ENIAC
- Grace Hopper (53): erste Programmiersprache (Cobol)

Frühe “analoge” Rechner

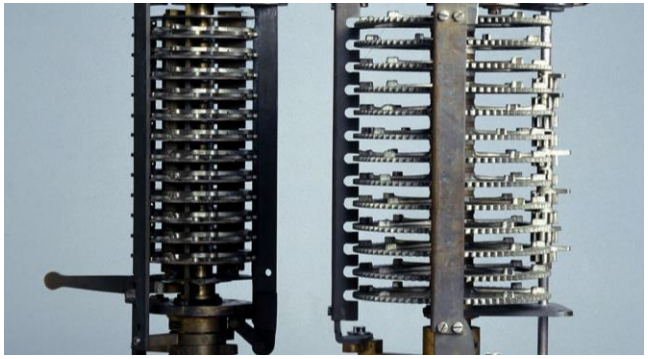


Abacus 2400 BC

Mechanismus von Antikythera 100 BC

Astronomische Uhr

Programmierbare Rechner



Charles Babagge (1791-1871), Ada Lovelace (1815-1852)

„Analytical Machine“

Maschine zur Auswertung von Polynomen, Logarithmentafeln

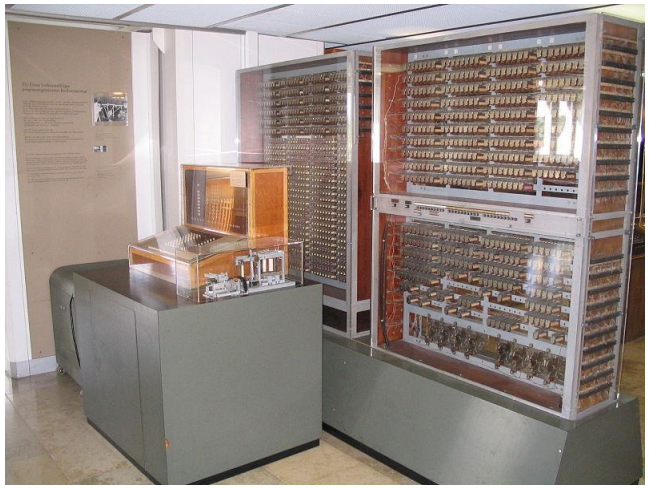
nie fertig geworden

Konrad Zuse (1910-1995)

Z1, 1938

Mechanisch

Frühe Computer (Konrad Zuse)



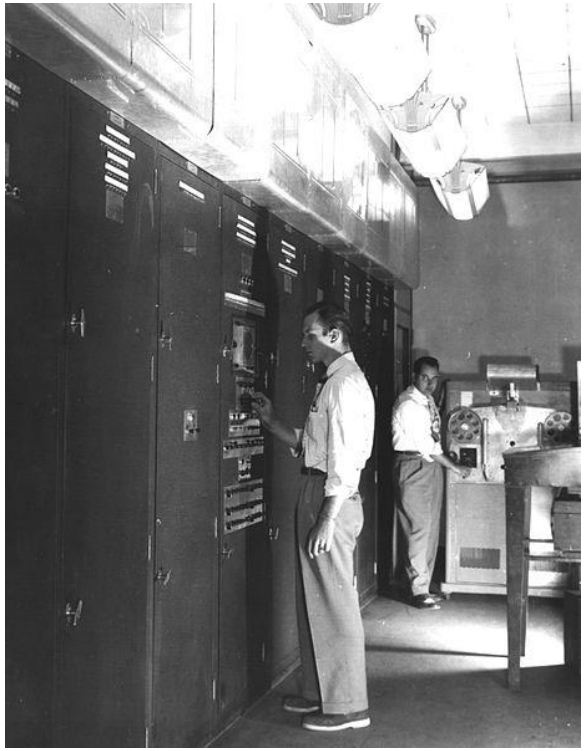
Zuse Z3 (1941)

Zuse Z4 (1942 - 45)

- Z3 und Z4 arbeiten mit Relais (elektromagnetische Schalter)
- Z3 und Z4 sind programmierbar (Programm extern) und universell

ENIAC *(1946)*

EDVAC (Electronic Discrete Variable Automatic Computer)



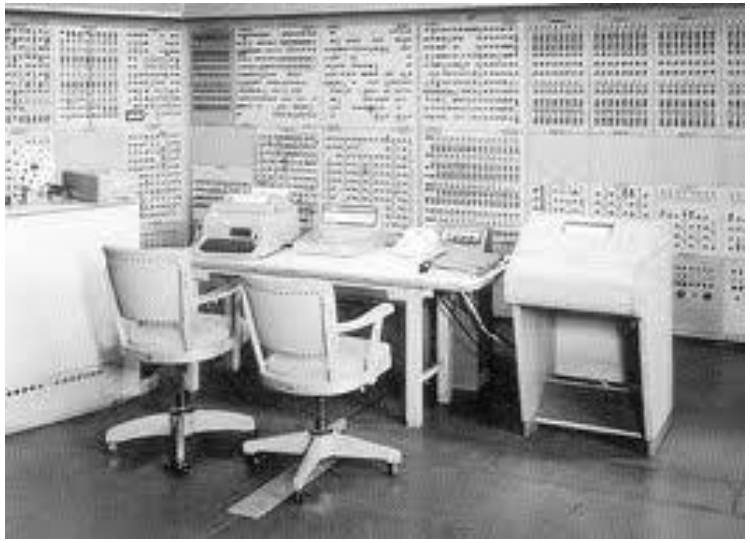
“First Draft of a Report on the EDVAC”
by John von Neumann, 1945

EDVAC, fertiggestellt im Jahr 1951

- Programme im Speicher
- Speicher: 5,5 kilobytes = 44,000 Bits
- Eine Multiplikation in 2,9 Millisekunden
- 6000 Vakuumröhren
- Stromverbrauch 56 kW
- 45,5 m² Bodenfläche und 7850 kg Gewicht
- Betriebspersonal 30 Personen für jede 8-Stunden-Schicht
- Kosten: 500000 Dollar (entspricht etwa 7 Millionen in 2018)

Das Vorbild für alle modernen Rechner

Spätere Computer



Hardware und Software

Graphische Oberfläche

- Und alles hinter schönen Bildern und mit bequemer Benutzung
- So nutzen Sie Ihr Telefon, Ihren Rechner

Anwendungen

- Mailprogramm, Browser, Whatsapp
- Hauptinhalt der Vorlesung

Betriebsystem

- Windows, Unix, Android, …
- Verwaltet die Resourcen des Rechners
- Kommt irgendwann

Hardware

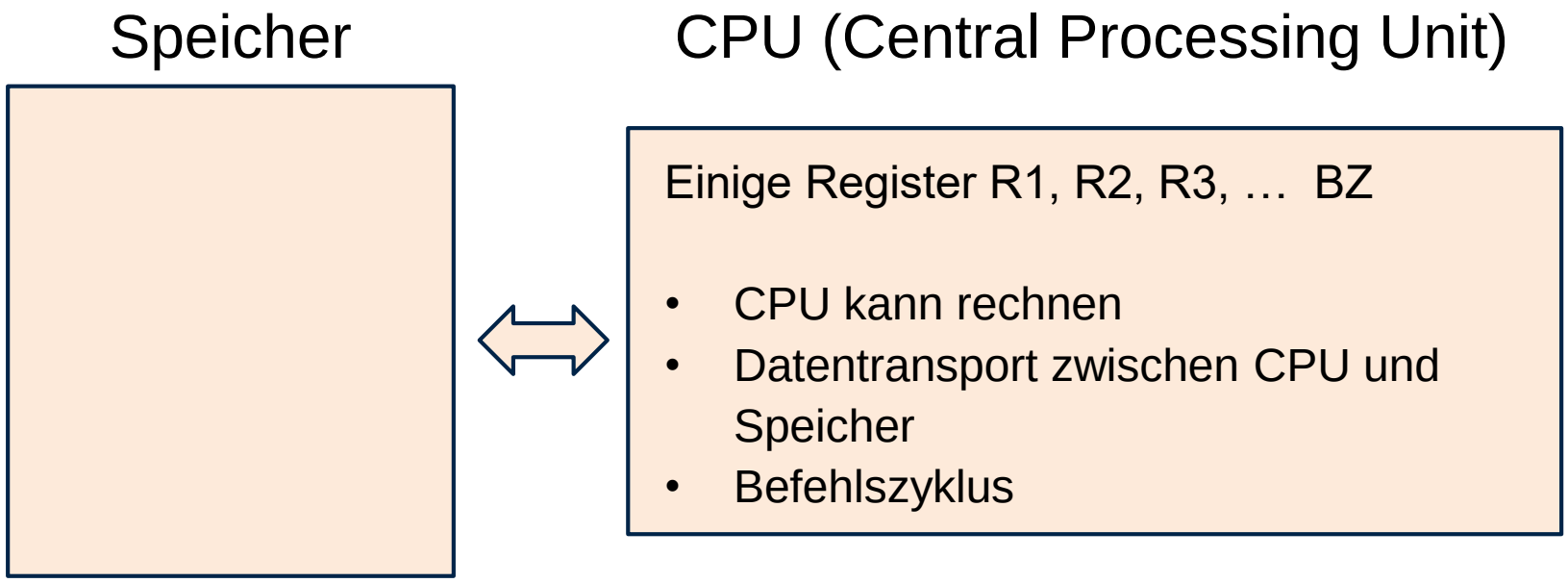
- Intel, Lenovo, HP,
- Darüber heute, sie ist gut versteckt.

Aufbau von Rechnern

Bitstrings und Speicher

- Bitstring = Folge von Nullen und Einsen, z.B. 00101000
- Bitstring kann man interpretieren als
  - Zahl in Binärdarstellung, z.B. 1011 ≡ 1x8 + 0x4 + 1x2 + 1x1 = 11, siehe Video Binärzahlen.
  - Buchstabe, z.B. 10110111 ≡ @
    00001101 ≡ a
- Speicher besteht aus Speicherzellen
  - Sind nummeriert 0, 1, 2, 3, 4, ….
  - Jede enthält einen Bitstring der Länge 64, früher 32, 16.

Von-Neumann-Rechner



- Rechner = Speicher + CPU
- Speicherzellen in der CPU heißen Register
- CPU kann rechnen (Befehlsumfang = billiger Taschenrechner)
- Speicher enthält Daten und Programm
- Befehlszyklus sorgt für die Ausführung des Programms, BZ = Befehlszähler

Typische Befehle

Transport	R3 ← M[5] M[5] ← R2	R1 ← M[R4] M[R2] ← R1
Rechnen	R1 ← 0	R1 ← R2 + R3
Sprung	BZ ← 7	BZ ← R1
Bed. Sprung	if R1 > 0, BZ ← n, else BZ ← BZ + 1	
Stop	STOP	

Programme und Befehlszyklus

Programm ist eine Folge von Befehlen

- Befehl 1
- Befehl 2
- Befehl 3
- Befehl 4
- Befehl 5
- Befehl 6

Befehlszyklus,

BZ = Befehlszähler

1. BZ ← 1
2. Führe Befehl mit der Nummer BZ aus. Falls STOP, halte an.
3. Erhöhe BZ um eins (außer bei Sprungbefehl, der BZ setzt)
4. Gehe nach 2.

In M[1] steht eine Zahl n ≥ 1, berechne 1 + … + n.

1. **R1 ← 0**
2. **R2 ← M[1]**
3. **R1 ← R1 + R2**
4. **R2 ← R2 - 1**
5. **IF R2 > 0, BZ ← 3**
6. **STOP**

Am Anfang steht in M[1] eine natürliche Zahl n ≥ 1.

Wenn das Programm stoppt, dann steht in R1 die Summe 1 + … + n.

Ausführung für n = 4

BZ

R1

R2

M[1]

Laufzeit =

Korrektheit des Programs

1. **R1 ← 0**
2. **R2 ← M[1]**
3. **R1 ← R1 + R2**
4. **R2 ← R2 - 1**
5. **IF R2 > 0, BZ ← 3**
6. **STOP**

Am Anfang steht in M[1] eine natürliche Zahl $n \geq 1$.

Wenn das Programm stoppt, dann steht in R1 die Summe $1 + \ldots + n$.

Jedes Mal bevor Befehl 3 ausgeführt wird, gilt:

- In R2 steht eine natürliche Zahl i mit $n \geq i \geq 1$.
- In R1 steht $n + \ldots + (i + 1)$

Höhere Programmiersprachen

1. **R1 ← 0**
2. **R2 ← M[1]**
3. **R1 ← R1 + R2**
4. **R2 ← R2 - 1**
5. **IF R2 > 0, BZ ← 3**
6. **STOP**

```
sum ← 0;
i ← n;
while (i > 0)
    sum ← sum + i;
    i ← i – 1;
```

Produktivitätsgewinn

Java, C, C++, Python,

Compiler übersetzen

Hardware

- Der Speicher, die CPU (Central Processing Unit), die Peripherie (Bildschirm, Tastatur, Maus, Netzanbindung, …)
- Führt Befehle aus und realisiert den Befehlszyklus.
- Reagiert auf Peripherie.
- Kauft man im Laden.
- Kann jedes Programm ausführen.

Software

- Die Summe der installierten Programme.
- Programme sind im Speicher abgelegt und werden durch die Hardware ausgeführt.
- Der Fantasie sind kaum Grenzen gesetzt.
- Lädt man aus dem Netz.
- Erstellen von guter Software ist teuer.
- Vervielfältigen ist billig, Grenzkosten ≈ Null

Kenngrößen und Neuerungen

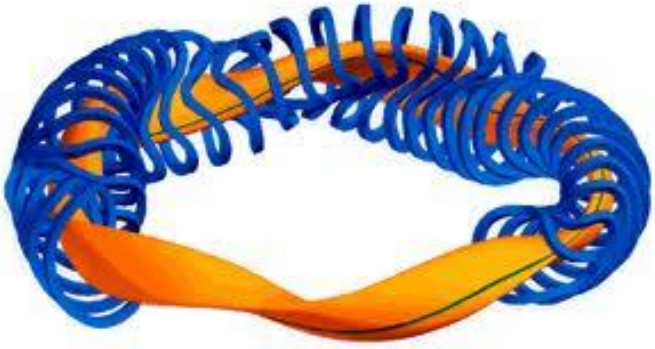
- Hauptspeicher: 10^9 Worte a 64 Bit
- Befehlszyklus: 10^9 Befehle pro Sekunde
- Eine Million mal leistungsfähiger (Geschwindigkeit, Speicher, Größe), Tausend mal billiger als 1950
- Neuerungen seit 1950
  - Interrupts (Unterbrechungen), mehrere Programme gleichzeitig
  - Speicherhierarchie: Cache, Main, Disk
  - Bildschirme, Grafik, Maus, Sound, Touch, Mikro
  - Netze
  - Preis und Leistung
  - **Software, Nutzerfreundlichkeit**

Bildschirme und Graphik

- Dieser Schirm: 1920 x 1080 Bildpunkte (Pixels)
- Die CPU kann für jedes Pixel Farbe und Helligkeit einstellen

Supercomputer

- 72 Schränke
- 73000 PowerPCs mit je 2 Gbyte RAM
- Simulation: Physik, Klima, Chemie, Strömung
- 13 Mio Euro



Universalität von Rechnern

Universalität von Rechnern

- Rechner sind **universell,** d.h., sie können **jedes** Programm ausführen, sofern es nur in ihre Maschinensprache übersetzt ist und es die Ressourcen des Rechners nicht sprengt.

- Universalität von Rechnern ist wesentlich für den Erfolg der Informatik:
  - Das gleiche Programm auf vielen Rechnern.
  - Viele Programme auf einem Rechner.

Rechner sind universell

- Normale Werkzeuge sind nicht universell: Hammer, Feile, Zange, Auto, ….
- Viele Programme auf einem Rechner: Ein Smartphone ist Telefon, aktiver Kalender, Fitnesstrainer, Bankterminal, Wetterauskunft, Browser, Suchmaschine, Musik- und Filmspieler, Spielzeug, …
- Ein Programm auf vielen Rechnern: Word läuft auf Rechnern von IBM, Lenovo, Toshiba, Samsung, Apple, …

Laufzeit von Programmen
Effiziente Programme

Laufzeit von Programmen

- Ausführungszeit in Sekunden
  - $n = 10^8$, 0.19 sec $n = 10^9$, 1.23 sec
- Für theoretische Überlegungen: Anzahl der ausgeführten Befehle/Operationen
- Unser Summenprogramm: Laufzeit = 3 + 3n = O(n)
- O( ) = Landausymbol für asymptotisches Wachstum: gibt nur den am schnellsten wachsenden Anteil wieder und ignoriert konstante Faktoren

Effiziente Programme

- Ein Programm heißt effizient, wenn seine Laufzeit an Eingaben der Größe *n* beschränkt ist durch Cn^k für Konstanten *C* und *k.*
- Eingabe der Größe *n*:
  - Graph mit *n* Knoten und Kanten
  - Zahl mit *n* Ziffern
- P = Menge aller Probleme für die es ein effizientes Programm gibt (polynomzeitberechenbar).

Beispiele

Kein effizientes Programm bekannt:

- Primfaktorzerlegung
- Problem des Handlungsreisenden
- 3-Färbung von Graphen
- Erfüllbarkeitsproblem der Aussagenlogik

Effizientes Programm bekannt:

- Sortieren
- Suchen
- Kürzeste Wege
- Multiplikation von Zahlen
- Lösen von linearen Gleichungen
- Test, ob eine Zahl Primzahl ist.

- Obige Probleme sind in P.

Zusammenfassung

- Rechner sind (im Prinzip) recht einfach aufgebaut: Recheneinheit und Speicher
- Befehlzyklus: wiederhole bis STOP-Befehl
  - Führe Befehl aus und erhöhe Befehlszähler um 1
  - Bei Sprungbefehl setze BZ auf die genannte Adresse
- Übersetzung von höherer Programmiersprache in Maschinensprache erfolgt maschinell
- Rechner sind universell: viele Programme auf einem Rechner, das gleiche Programm auf vielen Rechnern
- Zukunft??: Quantenrechner