

Computergrundlagen Geschichte des Computers

Axel Arnold

Institut für Computerphysik
Universität Stuttgart

Wintersemester 2010/11

1641: Rechenmaschine von B. Pascal



B. Pascal,
1623 - 1662

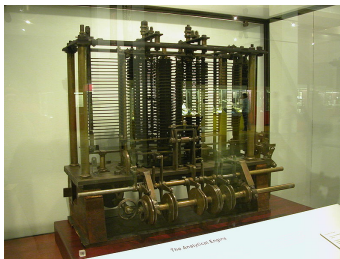


- mechanische Rechenmaschine
- addierte & subtrahierte ganze Zahlen
- nicht programmierbar

1837: Analytical Engine von C. Babbage



C. Babbage,
1791 - 1871

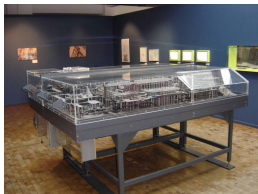


- programmierbare mechanische Rechenmaschine
- 3 Lochkartenleser für Programm, Konstanten und Eingabezahlen
- 4 Grundrechenarten, Sprünge
- nie gebaut

1941: Z3 von K. Zuse



K. Zuse,
1910 - 1995



Zuse Z1

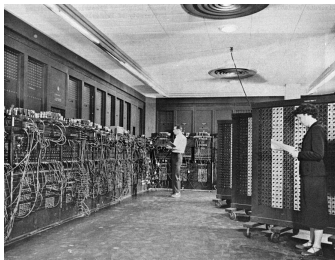


Zuse Z3

Z1, 1937: programmierbare mechanische Rechenmaschine
Z3, 1941: elektromechanischer Rechner:
 Relais und Kernspeicher

- 0,9 / 1,4 Flops (Fließkomma-Operationen / Sekunde)
- Programm und Daten über eigene Lochkartenleser
- im 2. Weltkrieg zerstört, Rekonstruktion in München

1946: ENIAC

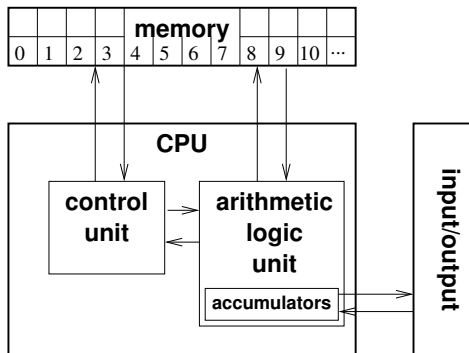


- erster Röhrenrechner
- 350 Flops
- Grundrechenarten + Wurzeln
- Programmeingabe durch Verdrahtung
- 1948 Einbau eines Befehlsspeichers

Von Neumann-Architektur (1945)



J. von Neumann,
1903 - 1957

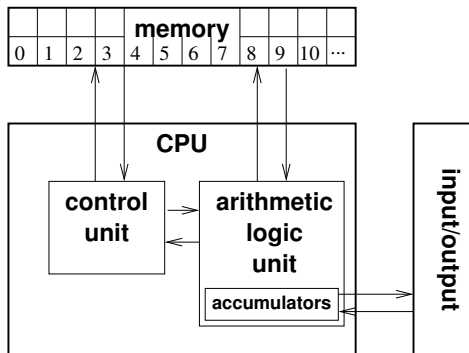


- Grundlage aller modernen Rechnerarchitekturen nach ENIAC
- *ein linearer* Ganzzahl-Speicher für Programme und Daten
- random access memory (RAM) – Speicher mit wahlfreiem Zugriff
- Befehle sind Zahlen im Speicher, können überschrieben werden

Von Neumann-Architektur (1945)



J. von Neumann,
1903 - 1957



- Befehle werden sequenziell abgearbeitet
- *Befehlszähler* enthält die Adresse des aktuellen Befehls
- wird normalerweise hochgezählt
- Sprünge und Verzweigung sind spezielle Befehle

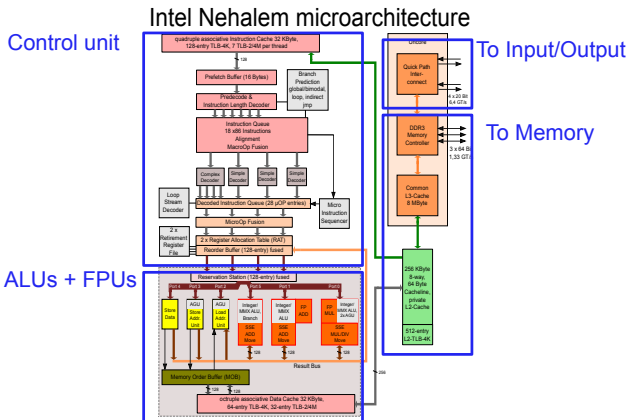
1976: Apple I / II



- 1976: Apple I, erster „Serien“-PC, 100 Exemplare
- 1977: Apple II, 2 Millionen Exemplare
- 8-bit CPU (6502), 48kB RAM
- Massenmarkt: C64 – 30 Millionen Exemplare
- Grund: zahlreiche Computerspiele
- erst ab ca 1990 setzen sich IBM-kompatible Heim-PCs durch

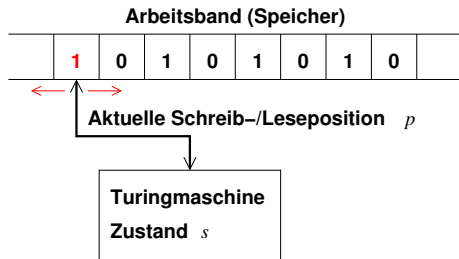
2008: Intel Nehalem

http://www.icp.uni-stuttgart.de



Auch ein Nehalem-Kern ist immer noch eine von Neumann-CPU!

Theoretische Informatik, Berechenbarkeit



A. Turing, 1912 - 1954

- ca 1920, D. Hilbert: Was ist berechnenbar, was beweisbar?
- 1931, K. Gödel: Entweder widerspruchsfrei oder vollständig!
- 1936: A. Turing: **Turingmaschine**, in „On Computable Numbers, with an Application to the "Entscheidungsproblem" “
- Nicht alles ist berechnenbar!
- Wir können nicht mal bestimmen, was nicht beweisbar ist

Turingmaschinen

Eine *Turingmaschine* ist definiert durch

- eine endliche Menge Γ , das Arbeitsalphabet, mit Leerzeichen $\sqcup \in \Gamma$
- eine endliche Menge Z von Zuständen der Maschine
- den Startzustand $s \in Z$ und den Haltezustand $h \in Z$
- eine Überföhrungsfunktion $\delta : Z \times \Gamma \rightarrow Z \times \Gamma \times \{\leftarrow, \downarrow, \rightarrow\}$

Eine *Konfiguration* ist

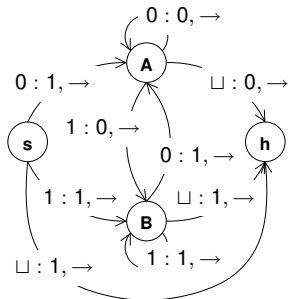
- eine Zeichenkette w über dem Arbeitsalphabet
- eine Position p in w , die aktuelle Arbeitsposition
- der aktuelle Zustand z der Turingmaschine

Eine Konfiguration (w, p, z) führt in eine andere (w', p', z') über, wenn $\delta(z, w_p) = (z', w'_p, d)$, und d p nach p' versetzt.

Startkonfiguration: Eine Eingabe w , p am Anfang von w , Zustand s .

Darstellung von Turingmaschinen

Beispiel: Eine „1“ vorne einfügen



A: „0“ gemerkt

B: „1“ gemerkt

s	0	A	1	→
s	1	B	1	→
s	□	h	1	→
A	0	A	0	→
A	1	B	0	→
A	□	h	0	→
B	0	A	1	→
B	1	B	1	→
B	□	h	1	→

Ablauf: Schreibe Zustand in das Arbeitsband vor die aktuelle Position, \vdash bezeichnet Übergang

$s101 \vdash 1B01 \vdash 11A1 \vdash 110B\sqcup \vdash 1101s\sqcup$

Turingmaschinen II

- Turingmaschinen können kombiniert werden
Beispiel: Hinter jedem Zeichen eine 0 einfügen
- *universelle Turingmaschine*: eine Turingmaschine, die berechnet, was eine Turingmaschine tut
- *Halteproblem*: es gibt keine Turingmaschine, die entscheidet, ob eine gegebene Turingmaschine bei leerer Eingabe anhält
- Turingmaschinen und von Neumann-Computer sind äquivalent
- Church-Turing-These:

Turing-berechenbare Funktionen sind genau die von Menschen intuitiv berechenbaren Funktionen