2. Computer (Hardware)

Computeraufbau: nur ein Überblick

➔ Genauer: Modul „Digitale Systeme" (2. Semester)
➔ Jetzt: Grundverständnis für Abarbeitung von Programmen in der Maschinensprache

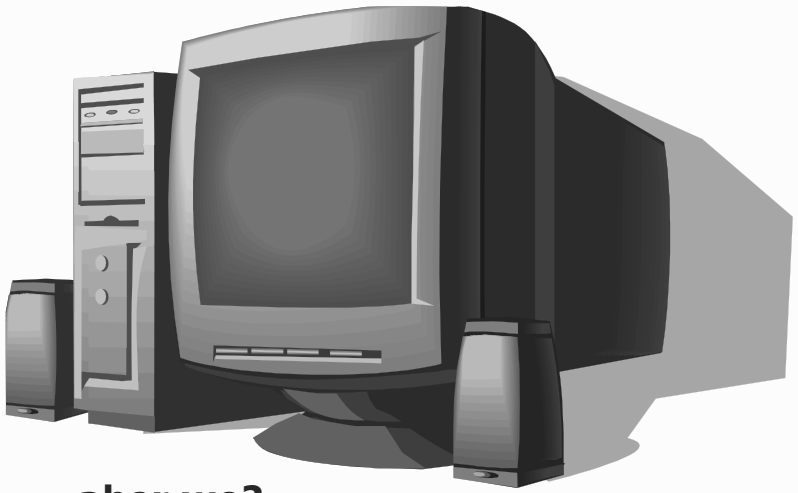
Grundlegendes Architektur-Prinzip:

„Von-Neumann-Rechner"

John von Neumann, 1945 konzipiert, alle heutigen Rechner davon geprägt

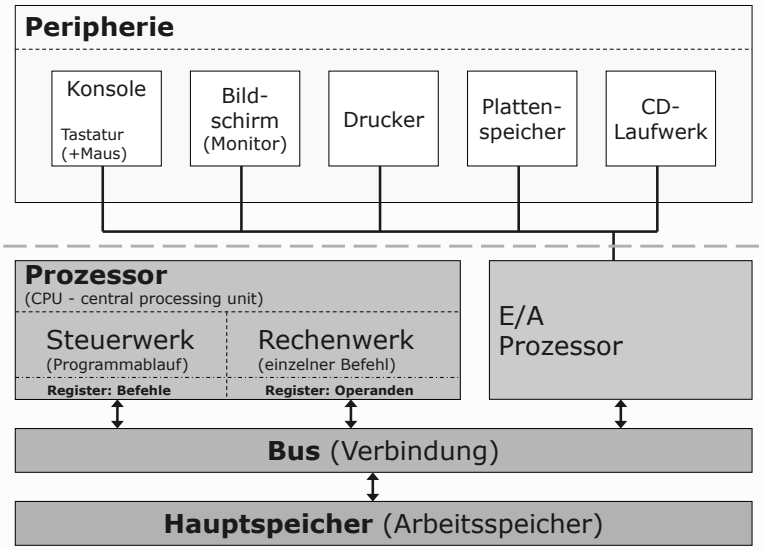
Johann von Neumann Haus /
Österreichisch-ungarischer Mathematiker
geb. Budapest 1903, gest. Washington 1957,
1926-29 Dozent an Berliner Universität

Von-Neumann-Rechner



... aber wo?

Einfacher Von-Neumann-Rechner mit Peripherie



Bestandteile des Von-Neumann-Rechners

- **Prozessor (CPU)**
  - Steuerwerk (Ablauf der Befehle) mit Befehlszähler und Befehlsregister
  - Rechenwerk (einzelne Befehle) mit Registern für Operanden: ALU Arithmetic Logic Unit
    (*Register:* spezielle Speicher, extrem kurze Zugriffszeiten)
- **Hauptspeicher (Arbeitsspeicher)**
  - Daten *und* Programme (!)
  - unterteilt in *adressierbare Speicherzellen* konstanter Länge (1 Byte = 8 Bit, 1 Wort = 2, 4 o. 8 Byte)
- **Verbindungen (Bus)**
- **Ein-/Ausgabe-Prozessoren**
  - Spezialprozessoren
  - Beschleunigung durch Parallelisierung

Beispiel: Inhalt Hauptspeicher (reales Beispiel)

Hier: Wortadresse (1 Wort = 4 Byte)

Wer erkennt dargestellte Informationen?

Adresse	Inhalt
60225	...
60226	00110011 01000110 01110000 10000001
60227	10110011 11100110 01110000 00011001
60228	00110011 00000110 01110000 11000001
60229	00000011 00000110 01110000 10000001
60230	10110011 00000110 01110000 01100001
60231	01110011 00000110 01110000 00000101
60232	01100111 00000001 01101011 01000101
60233	00111011 00000110 01110000 11100001
...	...
...	...
280461	00000000 00000000 00000000 00000001
...	...
...	...
440204	00000000 00000000 00000000 00000101
...	...

Speicherinhalt: Befehle und Daten

Befehl →

Daten →

Adresse	Inhalt
...	...
60231	01110011 00000110 01110000 00000101
60232	01100111 00000001 01101011 01000101
60233	00111011 00000110 01110000 11100001 **dekodiert: Maschinenbefehl JUMP 60229**
60234	...
...	...
...	...
440204	00000000 00000000 00000000 00000101 **dekodiert: ganze Zahl 5**

Probleme:
- Adresse 60233: könnte auch Zahl repräsentieren: Welche?
- Wer entscheidet: Zahl oder Befehl gemeint?

Speicherinhalt dekodiert

Speicher: Programm

Adresse	(Dekodierter) Inhalt
60225	...
60226	STORE ''1'' 280461
60227	LOAD A 280461
60228	LOAD B 440204
60229	TEST-EQU-0 B
60230	JUMPCOND 60234
60231	A := 2 * A
60232	B := B – 1
60233	JUMP 60229
60234	STORE A 280461
60235	...
60239	...
60240	...

Was passiert?

A, B: Register
MR A 2: A := 2 * A
SR B 1: B := B - 1

Speicher: Daten

Adresse	Inhalt
280460	...
280461	1 (y)
280462	...
280463	...
440202	...
440203	...
440204	5 (x)
440205	...
440209	...
440210	...
440211	...
440212	...

Originalprogramm

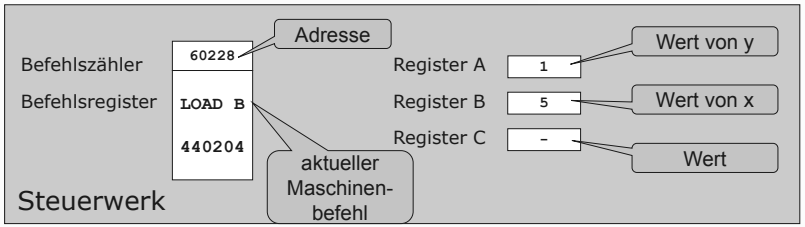
Prozedur in Pascal

entspricht obigem Maschinenprogramm mit

y in Zelle 280461
x in Zelle 440204

```
FUNCTION zweiHoch(x:INTEGER):INTEGER;
   VAR y: INTEGER;
BEGIN
   y := 1 ;
   WHILE x > 0 DO
     BEGIN
       y := 2 * y ;
       x := x - 1 ;
     END;
   RETURN y;
END zweiHoch;
```


Prozessor



Arbeit des Computers:

Ausführen der festgelegten Folge von Befehlen

→ Speicherinhalte ändern sich:

Genau so?

	Speicherzelle: y 280461	Speicherzelle: x 440204
Initialisierung	1	5
1. Schritt	2	4
2. Schritt	4	3
3. Schritt	8	2
4. Schritt	16	1
5. Schritt	32	0

Statt Hauptspeicherzellen in Registern

Arbeitsweise „Von-Neumann-Rechner"

sequentielle Arbeit (Befehle nacheinander)

Steuerwerk der CPU:
Sequentielle Ausführung der Befehlsfolge:

- Befehl des Befehlsregisters ausführen
- nächsten Befehl des Programms laden
  (neuer Inhalt: Befehlszähler, Befehlsregister)
  (→ fortlaufend bzw. programmierter Sprung)

Rechenwerk der CPU:
Aktuelle arithmetische und logische Operation ausführen:

- Arithmetik (z.B. ADD, SUB, MUL ...)
- Logik (z.B. Test auf Gleichheit: TEST-NOT-0)

Maschinensprache - Assemblersprache

Maschinensprache: kodiert, nicht lesbar

00111011 00000110 01110000 11100001

Assemblersprache: "lesbare" Variante

JUMP 60229

Assembler (Übersetzerprogramm):

Assemblersprache → Maschinensprache

Computer

13