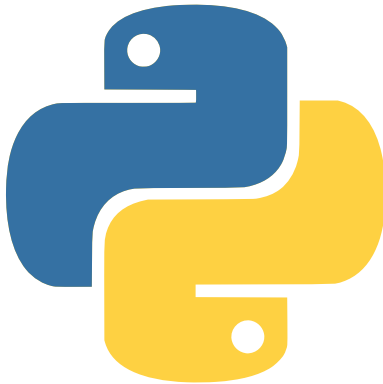




Universität Stuttgart

Projekt digit@L – BOOST. SKILLS. SUPPORT.



Dominik
Göddeke

Programmierkurs Python

Aufbau und Arbeitsweise von
Computern

Aufbau und Arbeitsweise von Computern

Aufbau und Arbeitsweise von Computern

- Ziel: Programmieren lernen

Aufbau und Arbeitsweise von Computern

- Ziel: Programmieren lernen
- Dazu: **Gedankenmodelle** (mental models) als allgemeines Abstraktionskonzept

Aufbau und Arbeitsweise von Computern

- Ziel: Programmieren lernen
- Dazu: **Gedankenmodelle** (mental models) als allgemeines Abstraktionskonzept
 - Etabliertes Konzept in der Didaktik von Grundschule bis Berufsleben

Aufbau und Arbeitsweise von Computern

- Ziel: Programmieren lernen
- Dazu: **Gedankenmodelle** (mental models) als allgemeines Abstraktionskonzept
 - Etabliertes Konzept in der Didaktik von Grundschule bis Berufsleben
 - Nutzung zulässiger Vereinfachungen und zielführender Analogien

Aufbau und Arbeitsweise von Computern

- Ziel: Programmieren lernen
- Dazu: **Gedankenmodelle** (mental models) als allgemeines Abstraktionskonzept
 - Etabliertes Konzept in der Didaktik von Grundschule bis Berufsleben
 - Nutzung zulässiger Vereinfachungen und zielführender Analogien
- Hier: **schnelle Fortschritte im Verständnis** der prinzipiellen Arbeitsweise von Computern ohne Eintauchen in Hardware-Details

Aufbau und Arbeitsweise von Computern

- Ziel: Programmieren lernen
- Dazu: **Gedankenmodelle** (mental models) als allgemeines Abstraktionskonzept
 - Etabliertes Konzept in der Didaktik von Grundschule bis Berufsleben
 - Nutzung zulässiger Vereinfachungen und zielführender Analogien
- Hier: **schnelle Fortschritte im Verständnis** der prinzipiellen Arbeitsweise von Computern ohne Eintauchen in Hardware-Details
- Können später bei Bedarf oder Interesse nachgetragen werden

Von-Neumann-Architektur

Von-Neumann-Architektur

- Bekanntestes Gedankenmodell für Computer: **Von-Neumann-Architektur**

Von-Neumann-Architektur

- Bekanntestes Gedankenmodell für Computer: **Von-Neumann-Architektur**
 - John von Neumann, 1945

Von-Neumann-Architektur

- Bekanntestes Gedankenmodell für Computer: **Von-Neumann-Architektur**
 - John von Neumann, 1945
 - Einer der bekanntesten und produktivsten Mathematiker des 20. Jahrhunderts

Von-Neumann-Architektur

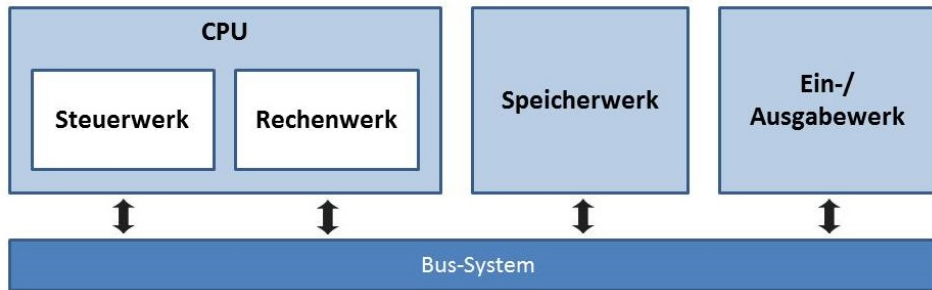
- Bekanntestes Gedankenmodell für Computer: **Von-Neumann-Architektur**
 - John von Neumann, 1945
 - Einer der bekanntesten und produktivsten Mathematiker des 20. Jahrhunderts
 - Einer der Wegbereiter der Informatik als eigenständige Wissenschaftsdisziplin

Von-Neumann-Architektur

- Bekanntestes Gedankenmodell für Computer: **Von-Neumann-Architektur**
 - John von Neumann, 1945
 - Einer der bekanntesten und produktivsten Mathematiker des 20. Jahrhunderts
 - Einer der Wegbereiter der Informatik als eigenständige Wissenschaftsdisziplin
- Vorteil der von-Neumann-Architektur: **Einfachheit dank des starken Abstraktionsgrads**

Von-Neumann-Architektur

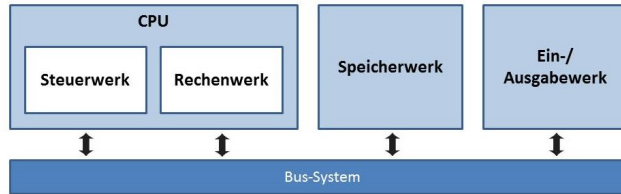
Klassische Darstellung:



Andreas Wilkens, <https://vfhcab.eduloop.de/loop/Von-Neumann-Architektur>, CC-BY

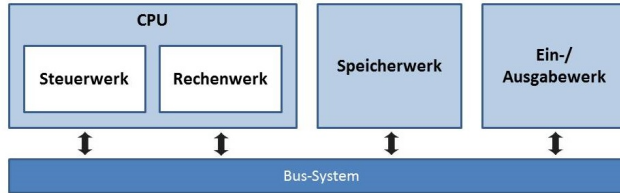
Rechen- und Steuerwerk

Rechen- und Steuerwerk



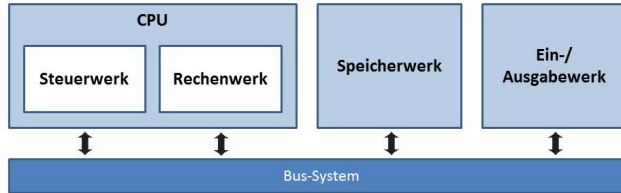
- **CPU (central processing unit)** arbeitet Befehle ab, produziert Ergebnisse

Rechen- und Steuerwerk



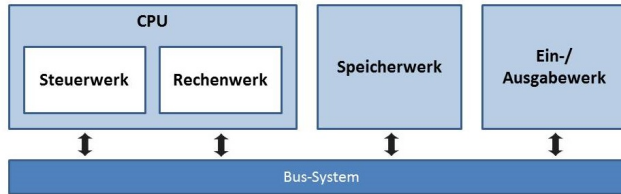
- **Rechenwerk (arithmetic logic unit, ALU)** „rechnet“

Rechen- und Steuerwerk



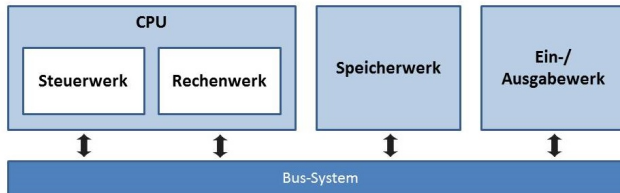
- **Rechenwerk (arithmetic logic unit, ALU)** „rechnet“
 - Abarbeiten von arithmetischen und logischen Operationen

Rechen- und Steuerwerk



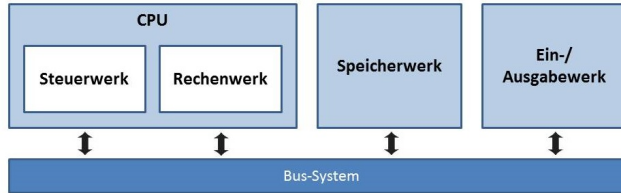
- **Rechenwerk (arithmetic logic unit, ALU)** „rechnet“
 - Abarbeiten von arithmetischen und logischen Operationen
 - Im Gedankenmodell unerheblich wie im Detail: Transistoren, kleine grüne Männchen, ...

Rechen- und Steuerwerk



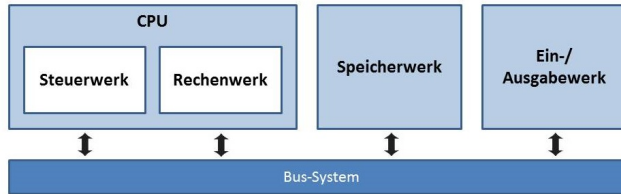
- **Rechenwerk (arithmetic logic unit, ALU)** „rechnet“
 - Abarbeiten von arithmetischen und logischen Operationen
 - Im Gedankenmodell unerheblich wie im Detail: Transistoren, kleine grüne Männchen, ...
 - Ebenso unerheblich: Befehle und Daten in Binärcodierung (100100011111010)

Rechen- und Steuerwerk



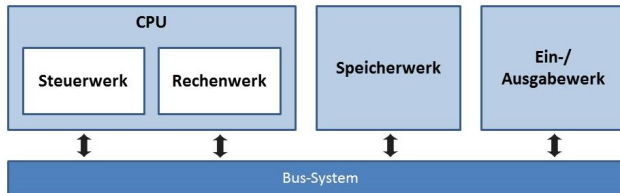
- **Steuerwerk (control unit)** „koordiniert“

Rechen- und Steuerwerk



- **Steuerwerk (control unit)** „koordiniert“
 - Füttern des Rechenwerks mit abzuarbeitenden Instruktionen (Befehlen, dem auszuführenden Programm)

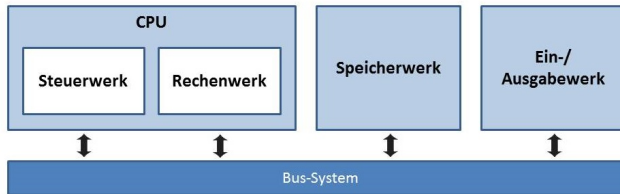
Rechen- und Steuerwerk



- **Steuerwerk (control unit)** „koordiniert“
 - Füttern des Rechenwerks mit abzuarbeitenden Instruktionen (Befehlen, dem auszuführenden Programm)
 - Und mit Eingabedaten für die Operationen

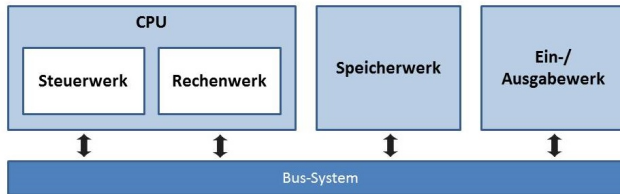
Speicherwerk

Speicherwerk



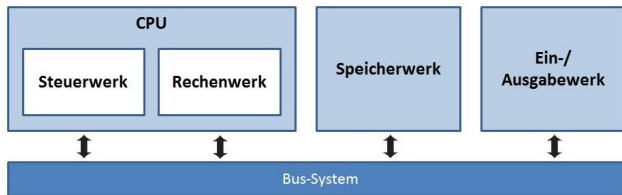
- **Speicherwerk** enthält Daten

Speicherwerk



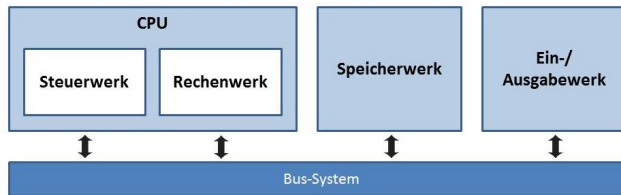
- **Speicherwerk** enthält Daten
- Wichtig: Eingabe- und Ausgabedaten des bearbeiteten Programms, aber auch das Programm selbst

Speicherwerk



- **Speicherwerk** enthält Daten
- Wichtig: Eingabe- und Ausgabedaten des bearbeiteten Programms, aber auch das Programm selbst
- Essentiell im von-Neumann-Modell: CPU darf jede Stelle im Speicher modifizieren, also auch das Programm selbst

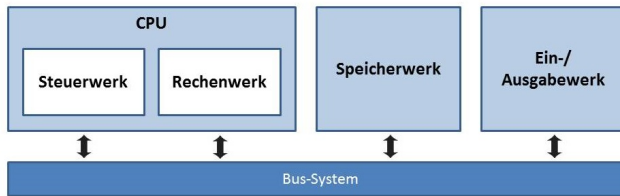
Speicherwerk



- **Speicherwerk** enthält Daten
- Wichtig: Eingabe- und Ausgabedaten des bearbeiteten Programms, aber auch das Programm selbst
- Essentiell im von-Neumann-Modell: CPU darf jede Stelle im Speicher modifizieren, also auch das Programm selbst
- Programme können selbst Ein- oder Ausgaben von Programmen sein

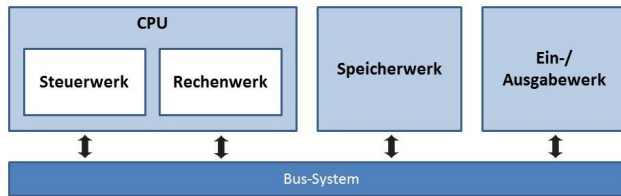
Ein-/Ausgabewerk und Bus-System

Ein-/Ausgabewerk und Bus-System



- **Bus-System** (lateinisch omnibus, Kommunikationssystem) regelt Kommunikation zwischen einzelnen Komponenten

Ein-/Ausgabewerk und Bus-System



- **Bus-System** (lateinisch omnibus, Kommunikationssystem) regelt Kommunikation zwischen einzelnen Komponenten
- **Ein- und Ausgabewerk:** Lochkartenleser, Tastaturen, Bildschirme, USB-Sticks, SD-Karten, irgendeine Cloud, ...

Diskussion

Diskussion

- Für **moderne Mehrkern-CPUs**

Diskussion

- Für **moderne Mehrkern-CPUs**
 - Duplizieren von Teilen des Modells ohne fundamentale Änderungen

Diskussion

- Für **moderne Mehrkern-CPUs**
 - Duplizieren von Teilen des Modells ohne fundamentale Änderungen
 - Mehr Rechen- und Steuerwerke in einer CPU etc.

Diskussion

- Für **moderne Mehrkern-CPUs**
 - Duplizieren von Teilen des Modells ohne fundamentale Änderungen
 - Mehr Rechen- und Steuerwerke in einer CPU etc.
- Rückschau: **enorm mächtige Abstraktion**, trotz ihrer Einfachheit

Diskussion

- Für **moderne Mehrkern-CPU**s
 - Duplizieren von Teilen des Modells ohne fundamentale Änderungen
 - Mehr Rechen- und Steuerwerke in einer CPU etc.
- Rückschau: **enorm mächtige Abstraktion**, trotz ihrer Einfachheit
- Alle modernen „Computer“ funktionieren nach diesem Prinzip

Diskussion

- Für **moderne Mehrkern-CPU**s
 - Duplizieren von Teilen des Modells ohne fundamentale Änderungen
 - Mehr Rechen- und Steuerwerke in einer CPU etc.
- Rückschau: **enorm mächtige Abstraktion**, trotz ihrer Einfachheit
- Alle modernen „Computer“ funktionieren nach diesem Prinzip
 - PCs, Laptops, Tablets, Telefone, Supercomputer, Armbanduhren

Diskussion

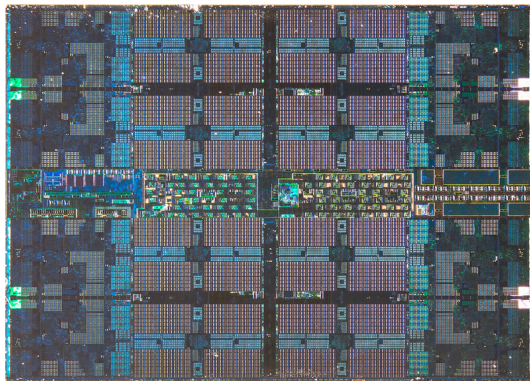
- Für **moderne Mehrkern-CPU**s
 - Duplizieren von Teilen des Modells ohne fundamentale Änderungen
 - Mehr Rechen- und Steuerwerke in einer CPU etc.
- Rückschau: **enorm mächtige Abstraktion**, trotz ihrer Einfachheit
- Alle modernen „Computer“ funktionieren nach diesem Prinzip
 - PCs, Laptops, Tablets, Telefone, Supercomputer, Armbanduhren
 - CPUs und GPUs von AMD, ARM, Intel, NVIDIA, Qualcomm, . . .

Diskussion

- Für **moderne Mehrkern-CPU**s
 - Duplizieren von Teilen des Modells ohne fundamentale Änderungen
 - Mehr Rechen- und Steuerwerke in einer CPU etc.
- Rückschau: **enorm mächtige Abstraktion**, trotz ihrer Einfachheit
- Alle modernen „Computer“ funktionieren nach diesem Prinzip
 - PCs, Laptops, Tablets, Telefone, Supercomputer, Armbanduhren
 - CPUs und GPUs von AMD, ARM, Intel, NVIDIA, Qualcomm, . . .
- **Ausnahme** die die Regel bestätigt: Quantencomputer

Diskussion

Kurzer Ausflug in die Realität: die-shot AMD Ryzen5



Fritzchens Fritz, <https://www.flickr.com/photos/130561288@N04/48350029156/in/photostream/>, CC-0

Verständnis des Steuerwerks

Verständnis des Steuerwerks

- Essentiell im Folgenden: **konzeptionelles Verständnis des Steuerwerks**

Verständnis des Steuerwerks

- Essentiell im Folgenden: **konzeptionelles Verständnis des Steuerwerks**
- Erinnerung: Programm ist Aneinanderreihung von Befehlen im Speicher

Verständnis des Steuerwerks

- Essentiell im Folgenden: **konzeptionelles Verständnis des Steuerwerks**
- Erinnerung: Programm ist Aneinanderreihung von Befehlen im Speicher
- Erinnerung: Rechenwerk arbeitet einzelne elementare Befehle sequentiell ab

Verständnis des Steuerwerks

- Essentiell im Folgenden: **konzeptionelles Verständnis des Steuerwerks**
- Erinnerung: Programm ist Aneinanderreihung von Befehlen im Speicher
- Erinnerung: Rechenwerk arbeitet einzelne elementare Befehle sequentiell ab
- Realisierung der „**Regieführung**“ dazwischen **im Steuerwerk**

Verständnis des Steuerwerks

- Essentiell im Folgenden: **konzeptionelles Verständnis des Steuerwerks**
- Erinnerung: Programm ist Aneinanderreihung von Befehlen im Speicher
- Erinnerung: Rechenwerk arbeitet einzelne elementare Befehle sequentiell ab
- Realisierung der „**Regieführung**“ dazwischen **im Steuerwerk**
- **Befehlszähler (program counter, PC)** für Reihenfolge der einzelnen Anweisungen

Verständnis des Steuerwerks

- Befehlszähler verweist auf **Stelle im Speicher**, an der nächste Anweisung „steht“, d.h. in Maschinen-lesbarer Form vorliegt

Verständnis des Steuerwerks

- Befehlszähler verweist auf **Stelle im Speicher**, an der nächste Anweisung „steht“, d.h. in Maschinen-lesbarer Form vorliegt
- Befehl wird ausgeführt, Befehlszähler wird inkrementiert

Verständnis des Steuerwerks

- Befehlszähler verweist auf **Stelle im Speicher**, an der nächste Anweisung „steht“, d.h. in Maschinen-lesbarer Form vorliegt
- Befehl wird ausgeführt, Befehlszähler wird inkrementiert
- Befehl besitzt **Operanden**, d.h. Daten, auf denen er operiert: a und b im Befehl $c = \text{add}(a, b)$

Verständnis des Steuerwerks

- Befehlszähler verweist auf **Stelle im Speicher**, an der nächste Anweisung „steht“, d.h. in Maschinen-lesbarer Form vorliegt
- Befehl wird ausgeführt, Befehlszähler wird inkrementiert
- Befehl besitzt **Operanden**, d.h. Daten, auf denen er operiert: a und b im Befehl $c = \text{add}(a, b)$
- Operanden verweisen ebenfalls auf Stelle im Speicher, an der sie vorliegen

Verständnis des Steuerwerks

- Befehlszähler verweist auf **Stelle im Speicher**, an der nächste Anweisung „steht“, d.h. in Maschinen-lesbarer Form vorliegt
- Befehl wird ausgeführt, Befehlszähler wird inkrementiert
- Befehl besitzt **Operanden**, d.h. Daten, auf denen er operiert: a und b im Befehl $c = \text{add}(a, b)$
- Operanden verweisen ebenfalls auf Stelle im Speicher, an der sie vorliegen
- Bus-System stellt sicher, dass Befehle und ihre Operanden zum Rechenwerk, sowie Ergebnisse zurück in den Speicher gelangen

Impressum, Danksagung und Quellen



Stiftung
Innovation in der
Hochschullehre



Gefördert durch die Stiftung Innovation in der Hochschullehre im Rahmen des Projekts digit@L, <https://stiftung-hochschullehre.de>

Gefördert mit Mitteln der Deutschen Forschungsgemeinschaft (EXC 2075 - 390740016) im Rahmen der Exzellenzstrategie

Autor: Dominik Göddeke, IANS, Universität Stuttgart



Weitere Quellen:

- Logos Universität Stuttgart, IANS, SimTech: Universität Stuttgart, alle Rechte vorbehalten
 - Logo Python: <https://freesvg.org/387>, CC-0
 - Logo Stiftung: Stiftung Innovation in der Hochschullehre, alle Rechte vorbehalten
 - Logo ZOERR: Universität Tübingen, alle Rechte vorbehalten
-



Veröffentlicht auf dem Zentralen OER Repositorium Baden-Württemberg, <https://www.zoerr.de>
