Zürcher Hochschule
für Angewandte Wissenschaften

zh aw School of Engineering

Modul Informatik-II

Kurs Informatik-3: Teil-1

www.engineering.zhaw.ch/de/engineering/studium/bachelor/informatik/studium-zurich.html

Prof. Dr. Olaf Stern
Leiter Studiengang Informatik
+41 58 934 82 51
olaf.stern@zhaw.ch

Lernziele: (Allgemein)

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

- **Die Studierenden kennen die *grundlegende Architektur von Rechnern* und die wichtigsten *Architekturelemente*.**
- **Sie sind vertraut mit der *elementaren Arbeitsweise eines Computers* und der *hardwarenahen Programmierung*. Sie können diese an einfachen Beispiel erläutern.**
- **Die Studierenden kennen die grundsätzlichen Aufgaben eines *Betriebssystems*. Sie können die *typischen* Verfahren und *Algorithmen*, die bei der *Entwicklung* von *Betriebssystemen* zur Anwendung gelangen, beschreiben.**

Lernziele: (Allgemein)

Zürcher Hochschule für Angewandte Wissenschaften
zh aw School of Engineering

- **Die Kurse der Module Informatik I und Informatik II (der Modulgruppen "Grundlagen der Informatik I+II") vermitteln den Studierenden die *Grundlagen der Informatik, die jede / jeder Studierende unabhängig von der Wahl der Wahlpflichtmodule im Fachstudium erlangen sollte*.**
- **Die vermittelten Grundlagen *werden in den Modulen im Fachstudium vorausgesetzt*.**

Lernziele: Spezifisch Teil 1

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

- **Die Studierenden kennen die Themen des Kurses Informatik-3 und sind vertraut mit dem geplanten Ablauf und den organisatorischen Belangen** (inklusive der Leistungsnachweise)**.**
- **Die Studierenden können den *grundlegenden Aufbau eines Computers* erläutern und sind insbesondere vertraut mit der *Von-Neumann-Architektur*.**

  Sie kennen die Vor- und Nachteilen sowie grundlegende Erweiterungen der *Von-Neumann-Architektur*.

Themenüberblick:

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

Technische Informatik / Rechnerarchitektur

[Olaf Stern, 16 Lektionen: 17. September - 12. November 2013]

- **Einführung / Übersicht**
- **Grundlegende Rechnerarchitektur**
  - **Von-Neumann-Rechner**
  - **Erweiterungen: Harvard-Architektur …**

Themenüberblick:

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

Technische Informatik / Rechnerarchitektur

[Olaf Stern, 16 Lektionen: 17. September - 12. November 2013]

- Einführung / Übersicht
- Grundlegende Rechnerarchitektur
- **Prozessoren**
  - **Leistungsmessung**
  - **Schematischer Aufbau**
  - **Prinzipielle Arbeitsweise (vereinfacht)**
  - **Pipelining**

Themenüberblick:

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

Technische Informatik / Rechnerarchitektur

[Olaf Stern, 16 Lektionen: 17. September - 12. November 2013]

- Einführung / Übersicht
- Grundlegende Rechnerarchitektur
- Prozessoren
- **Befehle – die „*Wörter*“ des Rechners**
  - **Befehl: Aufbau und Arten**
  - **Direkte und indirekte Adressierung**
  - **Assembler-Sprache / mnemonische Symbole**
  - **Programm-Konstrukte:** Einfache Operationen, Schleifen, Unterprogramme
  - **Einfache Programmbeispiele**

Themenüberblick:

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

Technische Informatik / Rechnerarchitektur

[Olaf Stern, 16 Lektionen: 17. September - 12. November 2013]

- Einführung / Übersicht
- Grundlegende Rechnerarchitektur
- Prozessoren
- Befehle – die „Wörter“ des Rechners
- ***„Mini-Power-PC“***
  - **Schema**
  - **Befehlssatz**

Themenüberblick:

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

Technische Informatik / Rechnerarchitektur

[Olaf Stern, 16 Lektionen: 17. September - 12. November 2013]

- Einführung / Übersicht
- Grundlegende Rechnerarchitektur
- Prozessoren
- Befehle – die „Wörter“ des Rechners
- „Mini-Power-PC“
- **Speicher**
  - **Speicherarten und Speicheraufbau**
  - **Speicherhierarchie**
  - **Cache (Puffer, Zwischenspeicher)**

Themenüberblick:

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

Technische Informatik / Rechnerarchitektur

[Olaf Stern, 16 Lektionen: 17. September - 12. November 2013]

- Einführung / Übersicht
- Grundlegende Rechnerarchitektur
- Prozessoren
- Befehle – die „Wörter“ des Rechners
- „Mini-Power-PC“
- Speicher
- **„Mini-Power-PC“** (Fortsetzung)
  - **Emulationsprogramm**
  - **Simulation einfacher Programme in Pseudo-Assembler- und Maschinen-Code**

Themenüberblick:

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

Grundlagen: Betriebssysteme

[Daniel Aebi, 14 Lektionen: 19. Nov. 2013 - 14. Jan. 2014]

- **Einführung**
- **Interrupt-Verarbeitung**
- **Prozesse und Threads**
- **(CPU-)Scheduling: SJF, LIFO, FIFO, …**
- **Synchronisation: Semaphoren, Monitore, Deadlocks**
- **Hauptspeicherverwaltung: Speicherhierarchie, Paging**
- **Geräte- und Dateiverwaltung: DMA, Memory-mapped I/O, Gerätetreiber, Dateisysteme**
- **Fallstudie(n): Windows 7, Linux**

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

Zu Technische Informatik / Rechnerarchitektur

- **Helmut Herold, Bruno Lurz, und Jürgen Wohlrab: *Grundlagen der Informatik*, 2. aktualisierte Auflage, Pearson Studium, August 2012**

  Anmerkungen: Gut und leicht zu lesen, nicht Formalismen im Vordergrund, viele Beispiele und Übungen, deckt tatsächlich fast alle Grundlagenthemen ab und ist auch gutes Nachschlagewerk für Grundlagen (nicht Vertiefung bzw. Detail), nicht immer formal ganz korrekt, ca. 70,- CHF

Literatur

Zu Technische Informatik / Rechnerarchitektur

- **David A. Petterson, John L. Hennessy: *Rechnerorganisation und Rechnerentwurf*, Oldenburg Verlag, März 2011**

  Aktuelle Deutsche Ausgabe des englischen Originals «*Computer Organization and Design*», sehr umfangreich, einfach zu lesen, auch für Vertiefung teilweise geeignet, ca. 60,- EUR

- **John L. Hennessy, David A. Petterson: *Computer Architecture. A Quantitative Approach*, Academic Press, Sep. 2006**

  Standardwerk für Vertiefung geeignet, ca. 50,- EUR
  (auch als Taschenbuch für ca. 10,- EUR erhältlich!)

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

Zu Betriebssystemen

- **Peter Mandl: *Grundkurs Betriebssysteme*, Vieweg + Teubner, ISBN 978-3-8348-0809-7, 2. Auflage, 2010**

  Primärliteratur, ca. 30,- EUR (=> muss besorgt werden !)

- **A. Tanenbaum: *Moderne Betriebssysteme*, Pearson Studium, 3. Auflage, ISBN 978-3-8273-7342-7, 2009**

  Ergänzende Literatur

Organisatorisches

„Lernteams“ / „Arbeitsgruppen“

- **2 oder max. 3 Studierende**
- **Lösen der Übungsaufgaben gemeinsam (Abgabe pro Team); nicht aufteilen!**
- **Empfehlungen:**
  - **Regelmässig** gemeinsam auch **Stoff auf- und nachbereiten**
  - Nicht erst am Sonntagabend mit den Aufgaben beginnen
    (u. a. keine Garantie seitens der IT, dass Moodle immer verfügbar ist)

Zürcher Hochschule für Angewandte Wissenschaften
zh aw School of Engineering

Leistungsüberprüfungen

- **„Abgesetzte“ Modulprüfung, in KW26/27, 60 Min., Anteil 50%**
- **Semesterklausuren (1-2), während regulärer Lektionen, ca. 60 Min., Anteil 35%**
- **Übungsaufgaben, wöchentlich, Anteil 15%**
  - Ausgabe der Aufgabenblätter jeweils am Ende einer Veranstaltung
  - Abgabe jeweils zu Beginn der folgenden Veranstaltung
  - Korrigierte Rückgabe jeweils wieder ca. eine Woche später
  - **NEU: Aufgaben beinhalten auch höhere Anforderungen** (z. B. Recherche, zusätzliche neue Lerninhalte, …)
  - Korrektur der Übungen: **Basil Moser (**mosb@zhaw.ch**)**

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

Leistungsüberprüfungen (Forts.)

- **...**
- **Übungsaufgaben: Bewertungs-Schema (0 bis 3 Punkte je Aufgabenserie)**
  - **0 Punkte:** Nicht abgegeben / weitgehend falsch
  - **1 Punkt:** Aufgaben grundsätzlich gelöst, gröbere Unzulänglichkeiten / Fehler
  - **2 Punkte:** Aufgaben weitgehend gelöst, nur kleine Unzulänglichkeiten / Fehler
  - **3 Punkte:** Sehr gute vollständige und fehlerfreie Lösung aller Aufgaben inkl. der optionalen Aufgaben

    (3 Punkte nur möglich, wenn optionale Aufgaben gestellt werden)

Bewertungsraster: Werden alle Aufgabenserien mit 2 Punkten gelöst, entspricht dieses einer 6 als Gesamtnote für die Übungsaufgaben

Organisatorisches

Allgemeines

- **Verhinderung**

  => Abmeldung per E-Mail / SMS beim Klassenchef

- **Mobile Phones** (und andere „*Spielzeuge*“)
- **E-Mail (dienstlich)**
- **Surfen, Facebook & Co**

- **Fragen ⇔ Feedback**

Themenüberblick:

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

Technische Informatik / Rechnerarchitektur

- **Einführung / Übersicht**
- **Grundlegende Rechnerarchitektur**
  - **Von-Neumann-Rechner**
  - **Erweiterungen: Harvard-Architektur …**
- **Prozessoren**
- **Befehle – die „Wörter“ des Rechners**
- **„Mini-Power-PC“**
- **Speicher**
- **„Mini-Power-PC“ (Fortsetzung)**

Lerninhalte Teil 1

Zürcher Hochschule für Angewandte Wissenschaften
School of Engineering

- **Einführung / Übersicht**
- **Grundlegende Rechnerarchitektur**
  - **Von-Neumann-Rechner**
    - Ansatz, Entwicklung, Komponenten
    - Programmablauf
    - Vor- / Nachteile
  - **Erweiterungen**
    - Harvard-Architektur
    - Super-Harvard-Architektur
    - Mischformen

Von-Neumann-Rechner

- **Konzept der Steuerung eines Rechners:**
  - **Das Programm wird mit den anderen Daten gleichberechtigt im Rechner in einem wiederbeschreibbaren Speicher abgelegt, auf den beliebig zugegriffen werden kann (*RAM - Random Access Memory*)**

    Somit kann auch nicht zwischen Programmdaten und sonstigen Daten als solches unterschieden werden und Programmdaten können wie andere Daten verändert werden

    Ursprünglich auch:

    „*Model for a stored-program digital computer*“
  - **Alle Daten sind binär codiert**

Von-Neumann-Rechner

- **Konzept der Steuerung eines Rechners: (Forts.)**
  - **Die Steuerung erfolgt über einzelne Befehle[+], die in einem Programm abgespeichert sind**
  - **Der Rechner ist universell**
    - Das **Programm** ist **nicht *„fest verdrahtet“***!
    - Der **Rechner** kann **beliebige Programme**, d. h. **Berechnungen** ausführen (Universalrechner; ***TM*-mächtig**)
  - **Ohne korrektes Programm ist der Rechner nicht funktionsfähig**

[+]Was ein Befehl konkret im Sinne eines Rechners ist, wird später aufgegriffen

Von-Neumann-Rechner

- **Konzept der Steuerung eines Rechners: (Forts.)**
  - **Das Programm ist eine sequentielle Abfolge (logischer) Anweisungen bzw. Entscheidungen**
  - **Ein besonderer Speicher (bzw. Register), der sogenannte *Befehlszähler* (oder *Programmzähler*), gibt die Adresse des nächsten Befehls an**
  - **Bedingte Programmbefehle erlauben Entscheidungen über den Fortgang des Programms**

[Die Bezeichnung ***Von-Neumann-Rechner*** leitete sich aus dem Namen des (österreichisch-ungarischen) Mathematikers **John von Neumann** ab, dessen wesentliche Arbeit 1945 veröffentlicht wurde]

Von-Neumann-Rechner

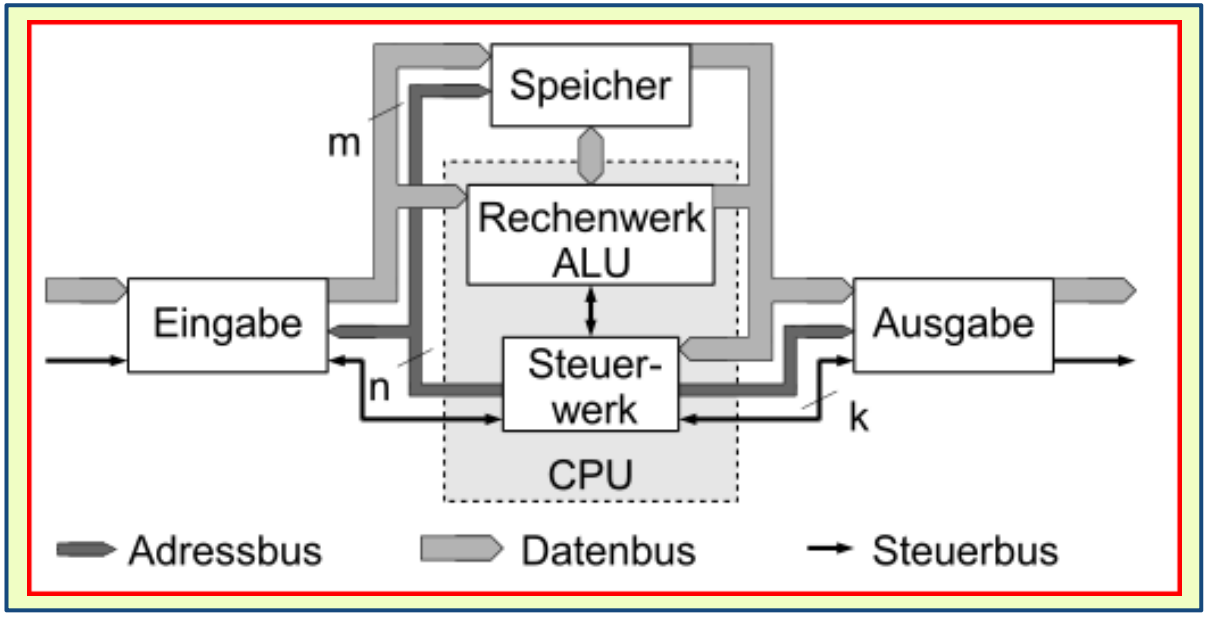
- **Komponenten (Grundelemente):**
  - ***Steuerwerk***
  - ***Rechenwerk***
  - ***Speicher(werk)***
  - ***Eingabewerk***
  - ***Ausgabewerk***

  Darin spiegeln sich auch die zwei weiteren Grundideen eines *Von-Neumann-Rechners* wieder:

  - **einfach in Technik und Funktion sowie**
  - **realisierbar**

Von-Neumann-Rechner

- **Komponenten (Grundelemente): (Forts.)**



Quelle: Wikipedia „http://de.wikipedia.org/wiki/Von-Neumann-Architektur“, Stand Aug. 2011

Von-Neumann-Rechner

- **Komponenten (Grundelemente): (Forts.)**
  - ***Steuerwerk*** **(auch als *Leitwerk* oder *Control Unit* bezeichnet)**

    Aufgabe: „*Herzstück*“ eines Rechners

    - **Interpretiert** den **Programmcode**
    - **Erstellt** die entsprechenden **Verbindungen** zwischen den **Daten** (Speicher) und dem **Rechenwerk**
    - **Steuert** die **Reihenfolge** der einzelnen **Programmbefehle**

    Es beinhaltet u. a. das ***Befehlsregister***, den ***Befehlsdecoder*** sowie den ***Befehlszähler***

Von-Neumann-Rechner

- **Komponenten (Grundelemente): (Forts.)**
  - ***Rechenwerk*** **(auch als *Arithmetische-Logische-Einheit*, *Arithmetic Logic Unit* oder kurz *ALU* bezeichnet)**

    Aufgabe: Stellt **logische** und **Rechen-Operationen** zur Verfügung

    Dieses können z. B. sein: Addieren, Multiplizieren, logische Vergleiche (NICHT-, UND-, ODER-Verknüpfungen), Verschieben von Werten / Stellen, ...

Anmerkung: Im Laufe der Entwicklung wurden Steuerwerk und Rechenwerk(e) zusammengefasst und als ***CPU*** (***Central Processing Unit***) bzw. **Hauptprozessor** bezeichnet (umfasst heute häufig noch mehr)

Von-Neumann-Rechner

- **Komponenten (Grundelemente): (Forts.)**
  - ***Speicherwerk*** **(oder *Memory*)**

    Aufgabe: Speicherung aller Daten

    - **Programme**, **Daten**, **Zwischen-** und **Endergebnisse** werden in **demselben Speicher** abgelegt
    - Der **Speicher** ist in fortlaufend **durchnummerierte** (gleichgrosse) **Zellen** unterteilt. Über diese „***Nummer***“, die als „***Adresse***“ bezeichnet wird, kann auf jede Speicherzelle zugegriffen werden:
      - **Inhalt einer Zelle auslesen: → Wert lesen**
      - **Inhalt einer Zelle verändern: → Wert schreiben**

Von-Neumann-Rechner

- **Komponenten (Grundelemente): (Forts.)**
  - ***Eingabe- / Ausgabewerk*** **(oder *I/O-Unit*)**

    Aufgabe: Steuert die **Ein-** und **Ausgabe aller Daten** des **Rechners**

    (D. h., das Einlesen bzw. Ausgeben von Zeichen jeglicher Art von/auf beliebigen Ein-/Ausgabegeräten)

Von-Neumann-Rechner

- **Komponenten (Grundelemente): (Forts.)**
  - ***Bus-Systeme*: *Daten*-, *Adress*- und *Steuerbus***

    Aufgabe: Verknüpft die **einzelnen Komponenten** miteinander zum **Austausch** von **Daten** und **Steuersignalen**

    (z. B. Rechenoperanden, Programmdaten, Steuersignale, Statusmeldungen, …)

Von-Neumann-Rechner

- **Komponenten (Grundelemente): (Forts.)**
  - ***Register*** **(als Teil des Steuerwerks)**

    Aufgabe: Zwischenspeicherung von **Daten**; dies sind **ausgezeichnete Datenspeicher**, auf die **sehr schnell zugegriffen** werden kann.

    Steuer- und Rechenwerk arbeiten in der **Regel Befehle ausschliesslich mit den Werten dieser ausgezeichneten Register ab** (d. h. Werte **lesen** und Ergebnisse **schreiben**)

    Register sind physikalisch im **Steuer- und Rechenwerk** integriert

Von-Neumann-Rechner

- **Komponenten (Grundelemente): (Forts.)**
  - ***Register*** **(Forts.)**

    Die übliche Register sind:

    - ***Befehlsregister*** (=> im ***Steuerwerk***)
    - ***Befehlszähler*** / ***Programmzähler*** (=> im ***Steuerwerk***)
    - ***Akkumulator*** (=> im ***Rechenwerk***)
    - ***Zustandsregister*** (=> im ***Steuerwerk***)
    - ***Statusregister*** (=> im ***Rechenwerk***)
    - ***Interrupt-Register*** (=> im ***Steuerwerk***)
    - [***Hilfs-*** und ***Arbeitsregister***, häufig auch als ***General Purpose Register*** bezeichnet (=> im ***Steuerwerk*** und ***Rechenwerk***)]

Von-Neumann-Rechner

- **Komponenten (Grundelemente): (Forts.)**
  - ***Register*** **(als Teil des Steuerwerks)**

    Die Grösse der *Registers* eines Rechners – i. d. R. ein Vielfaches von *8* Bit – charakterisiert wesentliche Eigenschaften des Rechners:

    - Die **Grösse** des ***Akkumulators*** und der ***Arbeitsregister*** beschränkt die Grösse der in einem ***Zyklus*** **bearbeitbaren Zahlen** (z. B. bei ***32*** **Bit** => ergibt 2^{32} differenzierte Darstellungen)
    - Die **Grösse** des ***Befehlszählers*** legt die Grösse des (direkt) **ansprechbaren** („***adressierbaren***“) **Speicherbereichs** fest
    - Die **Grösse** des ***Befehlsregisters*** beschränkt die **Anzahl** der **möglichen Befehle**

Von-Neumann-Rechner

- **Komponenten (Grundelemente): (Forts.)**
  - ***Register*** **(als Teil des Steuerwerks)**

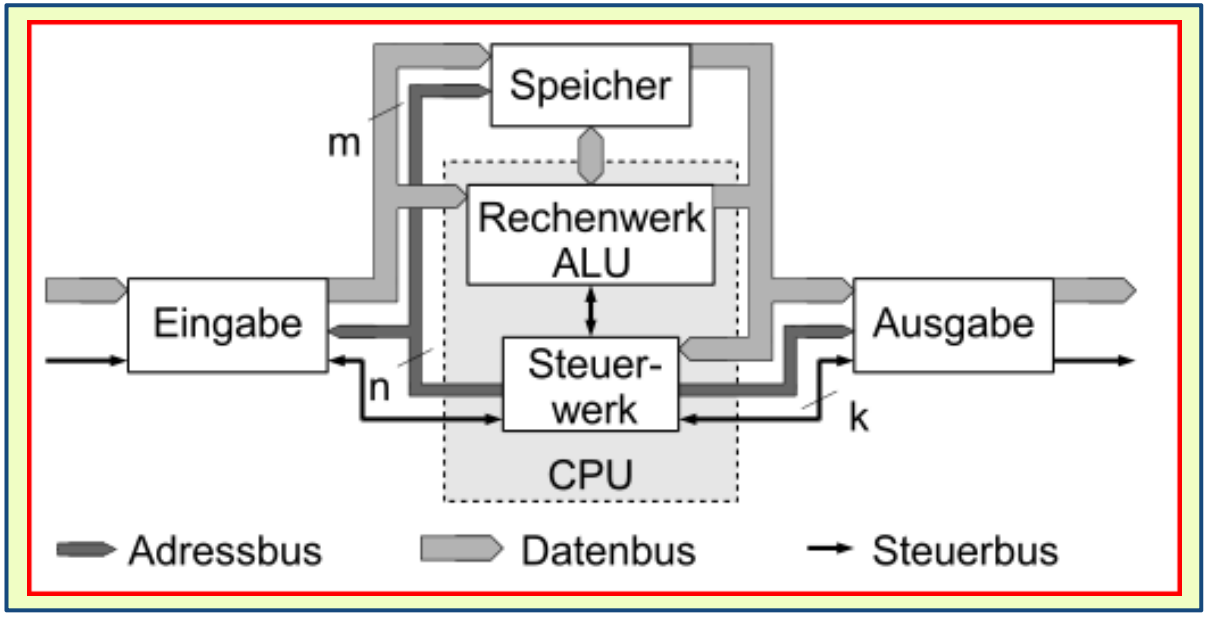
    Kleinere Register (mit den zugehörigen Rechen- und Steuerwerken) sind **einfacher / kostengünstiger** zu realisieren), ggf. jedoch **nicht ausreichend** bzw. **nicht effizient** für **komplexe Berechnungen**

    Aktuelle Rechner haben i. d. R. ***32*-Bit-** oder ***64*-Bit-Register**; kostengünstige Mikroprozessoren haben häufig ***8*-Bit-Register** (mit ***16*-Bit Adressregistern**)

    [Die Realisierung von Registern wird im Kurs Elektronik aufgezeigt]

Von-Neumann-Rechner

- **Komponenten (Grundelemente) - Zusammenspiel:**



Quelle: Wikipedia „http://de.wikipedia.org/wiki/Von-Neumann-Architektur“, Stand Aug. 2011

Von-Neumann-Rechner

- **Komponenten (Grundelemente): (Forts.)**
  - **„*Programmablauf*“:**

    1. Der **aktuelle Befehl** wird aus der **Speicherzelle**⁺, auf die der **Befehlszähler zeigt**, **ausgelesen** und in das **Steuerwerk übertragen**
    2. Das **Steuerwerk dekodiert den Befehl** und schaltet die entsprechenden **Signale auf den Steuerleitungen**

⁺Mit **„***Speicherzelle***“** ist hier ein Wort gemeint, d. h. eine adressierte Speicherzelle und ggf. die folgenden (i. d. R. 8, 16, 32 oder 64 Bit lang)

Von-Neumann-Rechner

- **Komponenten (Grundelemente): (Forts.)**
  - **„*Programmablauf*“:**

    3. Die für den **Befehl erforderlichen Operanden** werden aus dem **Speicherwerk gelesen** und in das **Rechenwerk** bzw. die **festgelegten Register** übertragen

    4. Die **dekodierte Operation** wird ausgeführt; das Ergebnis in ein **Register** (oder den **Speicher**) **geschrieben**

Von-Neumann-Rechner

- **Komponenten (Grundelemente): (Forts.)**
  - **„*Programmablauf*“:**

    5. Der **Befehlszähler** wird **um eins erhöht** oder auf Grund eines **Sprung-Befehls** um einen anderen Wert verändert

    [In **aufeinanderfolgenden Speicherzellen** stehen entsprechend **aufeinanderfolgende Programmbefehle**]

    Der Zyklus starten von vorne!

Von-Neumann-Rechner

- **Eigenschaften:**
  - Ein ***Von-Neumann-Rechner*** ist ein **universeller Rechner**, der **gleichmächtig** zu einer ***TM*** ist
  - Die **systematische Struktur** ermöglicht eine **effiziente Strukturierung** der vielfältigen **Operationen**
  - Die **sequentielle Arbeitsweise** **garantiert** einen **einfachen, deterministischen Ablauf** der **Programme**

    Nichtdeterministisches Verhalten durch unterschiedliche **Laufzeiten** in der realen Hardware im **Betrieb** ist **prinzipiell ausgeschlossen**

Von-Neumann-Rechner

- **Eigenschaften: (Forts.)**
  - Die Möglichkeit, den **Programm-Code modifizieren** zu können, **kann** auch dazu führen, dass dieser **unbeabsichtigt** oder **gezielt verändert** wird und das **Programm** und/oder **Daten beschädigt** werden … oder eventuell der **Rechner vollständig abstürzt!**

    Zugriffskontrollen und **Memory Protection** können dieses verhindern – **oder auch nicht …**

Von-Neumann-Rechner

- **Eigenschaften: „*Von-Neumann-Flaschenhals*“**
  - Die **strikte Trennung** von **Recheneinheit** und **Speicher** **bedingt**, dass **Daten sehr häufig übertragen** **werden müssen.**

!

=> Damit wird der **Datenbus**, der für den **Transport der Daten** (**Programm-** und **sonstige Daten**) zuständig ist, **zum Eng-pass** – dem sogenannten „***Von-Neumann-Flaschenhals***“ („***Von Neumann Bottleneck“***).

Von-Neumann-Rechner

- **Eigenschaften: „*Von-Neumann-Flaschenhals*“ (Forts.)**
  - Dieses wird insofern **verschärft**, als dass das **Von-Neumann-Konzept** den **expliziten Sequenzialismus** vorsieht (**ein Schritt nach dem anderen, keine Parallelität)** und …

    … der **Datenbus** für die **Übertragung** von **Programmbefehlen** und **Daten** genutzt wird, also die **übertragbare Datenmenge (Kapazität) aufgeteilt** wird.

    [Bei den ersten Rechnern stellte dieses keine echte Einschränkung dar, da die ***CPU*** (beinhaltet Steuer- und Rechenwerk) die langsamste Einheit darstellte; seit Jahrzehnten wächst jedoch die Leistungsfähigkeit der ***CPUs*** deutlich schneller als die der **Datenbusse** und **insbesondere die der Speicher**!]

Von-Neumann-Rechner

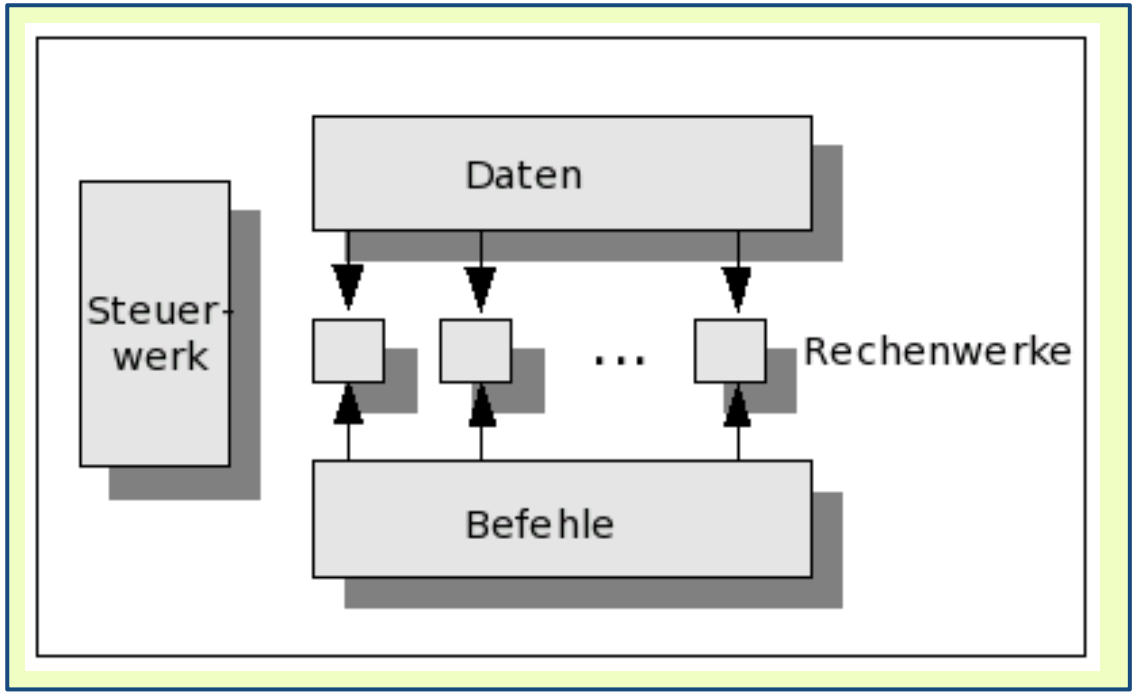
- **Eigenschaften: „*Von-Neumann-Flaschenhals*“ (Forts.)**
  - **Schon früh wurden grundlegende Konzepte zur Verringerung des *Von-Neumann-Flaschenhals* vorgeschlagen (und auch umgesetzt):**
    - Einfügen eines **schnellen Zwischenspeichers zwischen CPU** und **Speicher** (***Cache***, siehe später)
    - **Getrennte Zwischenspeicher** und **Datenbusse** für **Daten** und **Programm-Code**
    - **Vorhersage** von **bedingten Programmverzweigungen / -sprüngen** (***branch prediction***, siehe später)
    - **Parallele Strukturen** (Fachstudium)
    - …

Harvard-Architektur

- **Konzept-Differenzen (gegenüber *Von-Neumann-Rechner*):**
  - **Trennung** von **Befehlsspeicher (Programmdaten)** und **Daten-speicher**
  - **Nutzung** von **getrennten Datenbussen** für den **Zugriff** auf **Programm-Code** (im **Befehlsspeicher**) und **Daten** (im **Datenspeicher**)
  - [Parallele Nutzung von mehreren Rechenwerken]
  - [Getrennte Zwischenspeicher für Befehle (Programm-Code) und Daten]

Harvard-Architektur

- **Konzept-Differenzen (gegenüber *Von-Neumann-Rechner*):**



Quelle: Wikipedia „http://de.wikipedia.org/wiki/Harvard-Architektur“, Stand Aug. 2011

Harvard-Architektur

- **Eigenschaften (Vorteile):**
  - **Befehle (Programm-Code)** und **Daten können gleichzeitig geladen werden**
  - **Programm-Code** kann **weder beabsichtigt noch unbeabsichtigt (durch Daten) überschrieben** werden

    [Z. B. sind ***Puffer-Überläufe*** **einfacher kontrollierbar**]
  - **Befehlswortbreite** und **Datenwortbreite** können **unterschiedlich gross** sein:

    => typischerweise kleiner für **Befehle** und **grösser** für **Daten**

Harvard-Architektur

- **Eigenschaften (Nachteile):**
  - **Wie allgemein immer bei Parallelität möglich, kann es zu nichtdeterministischem Verhalten durch unterschiedliche Laufzeiten im Betrieb kommen**
  - **Nicht benötigter Speicherplatz für das Programm kann nicht für Daten genutzt werden – und umgekehrt**

Harvard-Architektur

- **Erweiterungen:**
  - **„*Nicht benötigter Speicherplatz für das Programm kann nicht für Daten genutzt werden – und umgekehrt*“**

    => Bei der ***Super-Harvard-Architektur*** kann auf **einen gemeinsamen Speicher** über **verschiedene Daten-Busse parallel zugegriffen** werden

    (Speicher kann wieder **gemeinsam genutzt** werden)

Harvard-Architektur

- **Auch wenn viele der aktuellen Rechner Elemente der *Harvard-Architektur* umsetzen, Erweiterungen wie *Pipelining* und mehrere *Speicher- / Cache-Hierarchien* nutzen, so arbeiten sie (fast) alle im Grundsatz nach wie vor nach dem *Von-Neumann-Konzept* !**
- **Ausnahmen davon sind i. d. R. Spezialrechner, wie Signal-prozessoren oder einfache Mikroprozessoren, die z. B. ein festes Programm implementiert haben**