

# 64-040 Modul IP7: Rechnerstrukturen

[http://tams.informatik.uni-hamburg.de/  
lectures/2012ws/vorlesung/rs](http://tams.informatik.uni-hamburg.de/lectures/2012ws/vorlesung/rs)

– Kapitel 14 –

Andreas Mäder



Universität Hamburg  
Fakultät für Mathematik, Informatik und Naturwissenschaften  
Fachbereich Informatik

**Technische Aspekte Multimodaler Systeme**

Wintersemester 2012/2013

# Kapitel 14

## Rechnerarchitektur

Motivation

Beschreibungsebenen

Wie rechnet ein Rechner?

Literatur



# Was ist Rechnerarchitektur?

## Definitionen

1. *The term architecture is used here to describe the attributes of a system as seen by the programmer, i.e., the conceptual structure and functional behaviour, as distinct from the organization and data flow and control, the logical and the physical implementation. [Amdahl, Blaauw, Brooks]*
2. *The study of computer architecture is the study of the organization and interconnection of components of computer systems. Computer architects construct computers from basic building blocks such as memories, arithmetic units and buses.*

## Was ist Rechnerarchitektur? (cont.)

*From these building blocks the computer architect can construct anyone of a number of different types of computers, ranging from the smallest hand-held pocket-calculator to the largest ultra-fast super computer. The functional behaviour of the components of one computer are similar to that of any other computer, whether it be ultra-small or ultra-fast.*

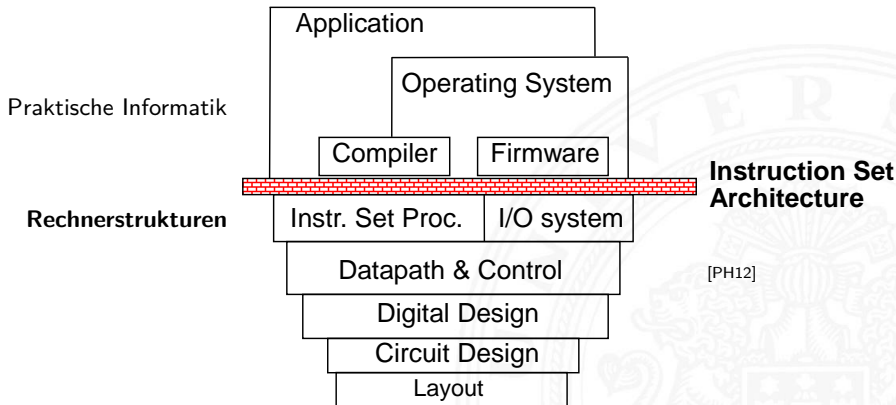
*By this we mean that a memory performs the storage function, an adder does addition, and an input/output interface passes data from a processor to the outside world, regardless of the nature of the computer in which they are embedded. The major differences between computers lie in the way of the modules are connected together, and the way the computer system is controlled by the programs. In short, computer architecture is the discipline devoted to the design of highly specific and individual computers from a collection of common building blocks. [Stone]*

## Was ist Rechnerarchitektur? (cont.)

Zwei Aspekte der Rechnerarchitektur

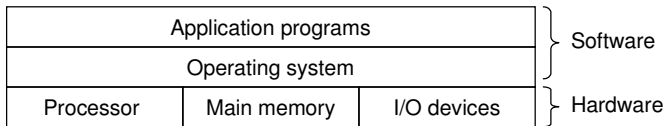
1. Operationsprinzip: das funktionelle Verhalten der Architektur
  - = Programmierschnittstelle
  - = ISA – **I**nstruction **S**et **A**rchitecture  
Befehlssatzarchitektur
  - = Maschinenorganisation
2. Hardwarestruktur: beschrieben durch Art und Anzahl der Hardware-Betriebsmittel sowie die sie verbindenden Kommunikationseinrichtungen
  - = Implementierung: welche Einheiten, wie verbunden. . .
  - = beispielsweise „von-Neumann“ Architektur

# Schnittstelle zur praktischen Informatik



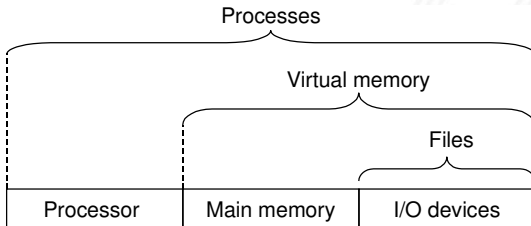
# Beschreibungsebenen

## ► Schichten-Ansicht: Software – Hardware



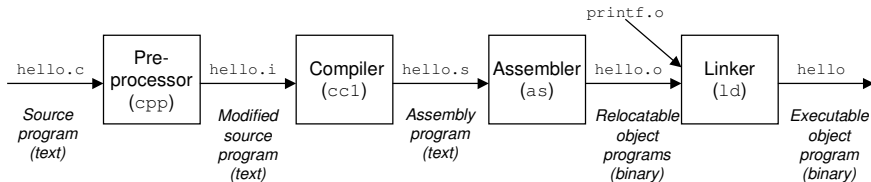
[BO11]

## ► Abstraktionen durch Betriebssystem



[BO11]

# Das Kompilierungssystem



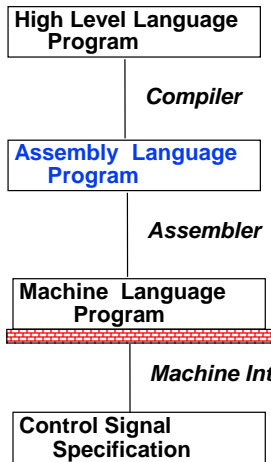
[BO11]

⇒ verschiedene Repräsentationen des Programms

- ▶ Hochsprache
- ▶ Assembler
- ▶ Maschinensprache



# Das Kompilierungssystem (cont.)



```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

```
lw $15, 0($2)
lw $16, 4($2)
sw $16, 0($2)
sw $15, 4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

ALUOP[0:3] <= InstReg[9:11] & MASK

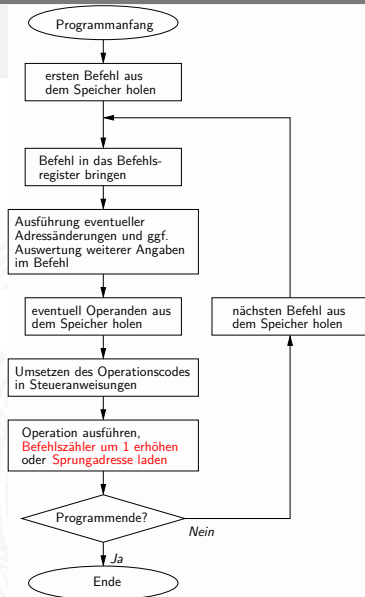
[PH12]

## Wie rechnet ein Rechner?

- ▶ Architektur im Sinne von „Hardwarestruktur“  
beispielsweise als von-Neumann Architektur
  - ▶ „Choreografie“ der Funktionseinheiten?
  - ▶ wie kommuniziert man mit Rechnern?
  - ▶ was passiert beim Einschalten des Rechners?
- ▶ Erweiterungen des von-Neumann Konzepts
  - ▶ *parallele*, statt sequentieller Befehlsabarbeitung  
Stichwort: superskalare Prozessoren
  - ▶ dynamisch veränderte Abarbeitungsreihenfolge  
Stichwort: „*out-of-order execution*“
  - ▶ getrennte Daten- und Instruktionsspeicher  
Stichwort: *Harvard-Architektur*
  - ▶ *Speicherhierarchie*, Caches etc.

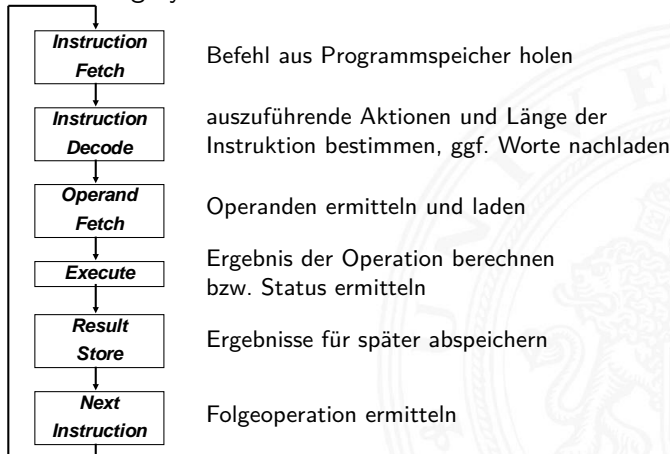
# Programmverarbeitung

- ▶ von-Neumann Architektur
  - ▶ Programm als Sequenz elementarer Anweisungen (Befehle)
  - ▶ als Bitvektoren im Speicher codiert
  - ▶ Interpretation (Operanden, Befehle und Adressen) ergibt sich aus dem Kontext (der Adresse)
  - ▶ zeitsequenzielle Ausführung der Instruktionen

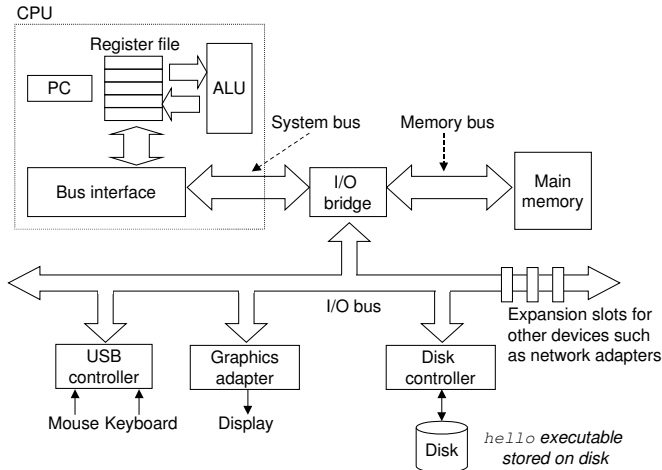


## Programmverarbeitung (cont.)

### ► Ausführungszyklus

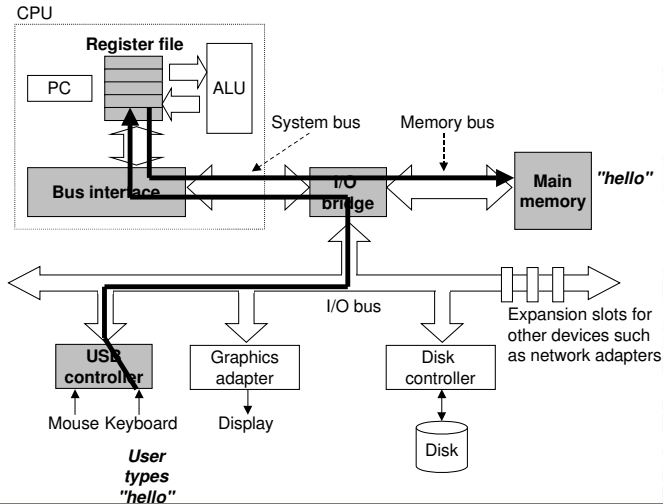


# Hardwareorganisation eines typischen Systems



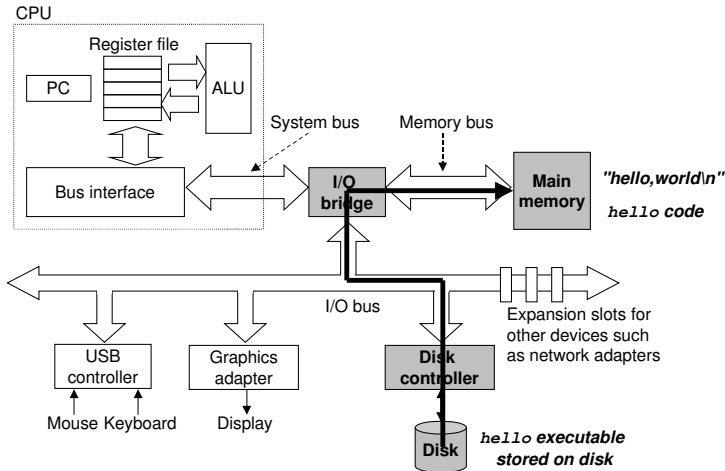
[BO11]

# Programmausführung: 1. Benutzereingabe



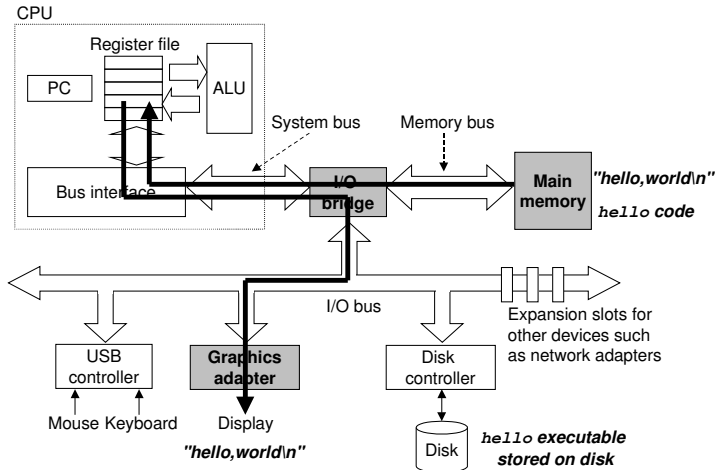
[BO11]

## Programmausführung: 2. Programm laden



[BO11]

## Programmausführung: 3. Programmlauf



[BO11]



# Boot-Prozess

## Was passiert beim Einschalten des Rechners?

- ▶ Chipsatz erzeugt Reset-Signale für alle ICs
- ▶ Reset für die zentralen Prozessor-Register (PC, ...)
- ▶ PC wird auf Startwert initialisiert (z.B. 0xFFFF FFEF)
- ▶ Befehlszyklus wird gestartet
- ▶ Prozessor greift auf die Startadresse zu  
dort liegt ein ROM mit dem Boot-Programm
- ▶ Initialisierung und Selbsttest des Prozessors
- ▶ Löschen und Initialisieren der Caches
- ▶ Konfiguration des Chipsatzes
- ▶ Erkennung und Initialisierung von I/O-Komponenten
- ▶ Laden des Betriebssystems

# Literatur

- [BO11] R.E. Bryant, D.R. O'Hallaron:  
*Computer systems – A programmers perspective.*  
2nd edition, Pearson, 2011. ISBN 0–13–713336–7
- [Tan06] A.S. Tanenbaum: *Computerarchitektur – Strukturen, Konzepte, Grundlagen.* 5. Auflage, Pearson Studium, 2006.  
ISBN 3–8273–7151–1
- [Tan09] A.S. Tanenbaum: *Structured Computer Organization.*  
5th rev. edition, Pearson International, 2009.  
ISBN 0–13–509405–4

## Literatur (cont.)

[PH11] D.A. Patterson, J.L. Hennessy: *Rechnerorganisation und -entwurf – Die Hardware/Software-Schnittstelle*.  
4. Auflage, Oldenbourg Verlag, 2011.  
ISBN 978-3-486-58190-3

[PH12] D.A. Patterson, J.L. Hennessy: *Computer Organization and Design – The Hardware/Software Interface*.  
4th rev. edition, Morgan Kaufmann Publishers, 2012.  
ISBN 978-0-12-374750-1  
Grafiken aus: Foliensatz