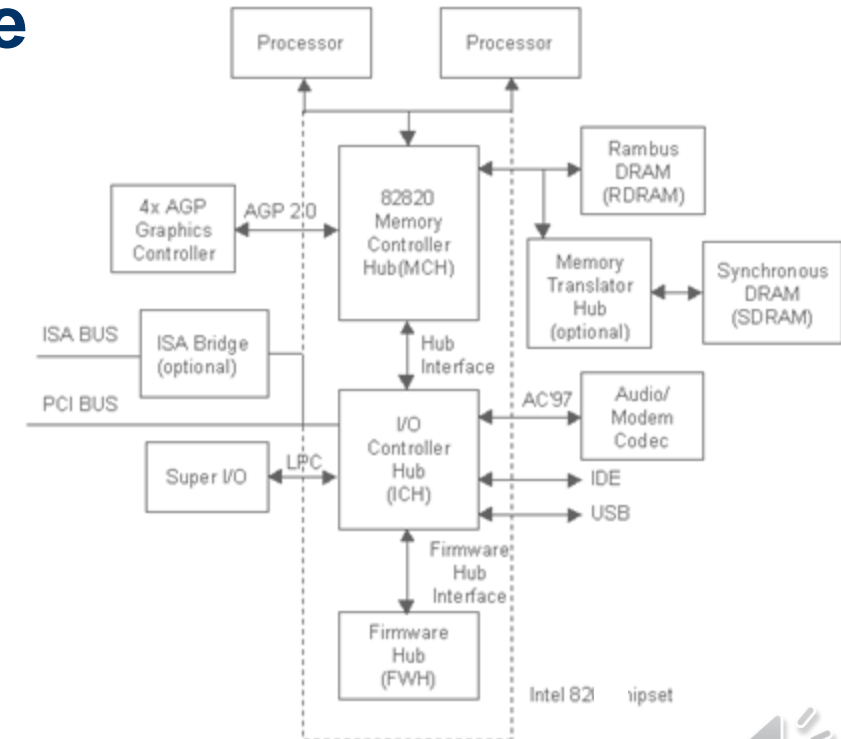
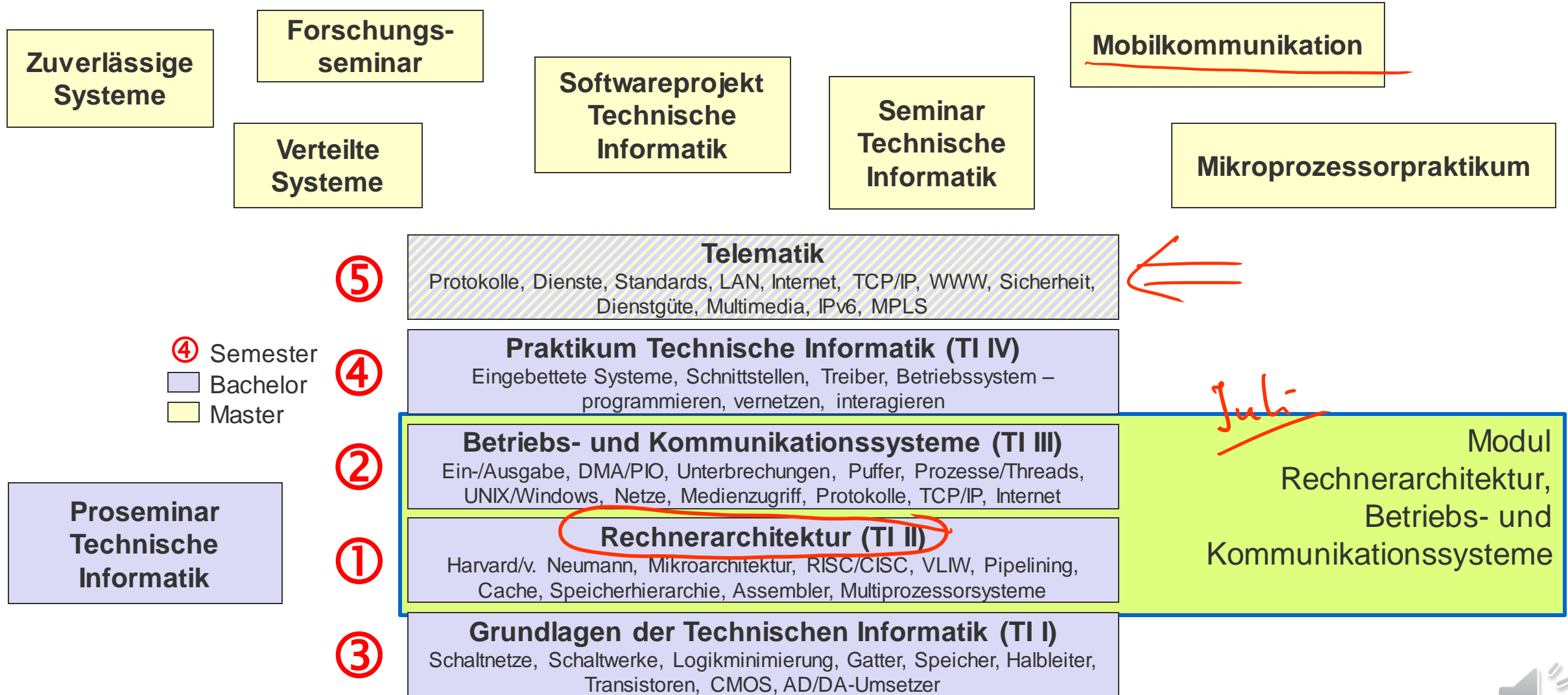


# TI II: Computer Architecture Winter 2017/2018

Prof. Dr.-Ing. Jochen Schiller  
Computer Systems & Telematics  
Freie Universität Berlin, Germany



# Veranstaltungen der Technischen Informatik



# Organisation

## Vorlesung

- Freitags, 10:00-12:00h, HS, Takustr. 9

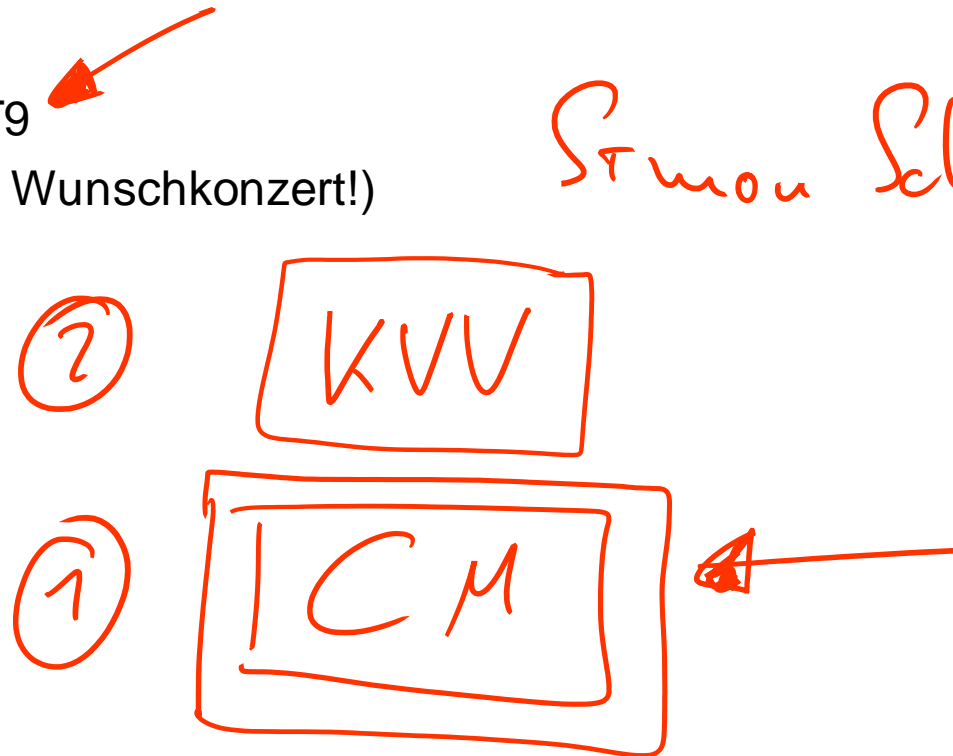
## Sprechstunde

- Dienstags, 14:00-15:00h, Raum 156, T9
- Tutoren: während der Tutorien (und im Wunschkonzert!)

*Simon Schmitt*

## Tutorien

- Max. 25 Student/innen pro Gruppe
- Zeiten siehe KVV und nächste Folie
- Registrierung im KVV



# Organisation – Tutorien

| Wochentag   | Zeit (c.t.) | Seminar # | Raum  | Tutor              |
|-------------|-------------|-----------|-------|--------------------|
| Mittwochs   | 08 – 10     | 06        | K 038 | Lilli Walter       |
|             |             | 09        | K 044 | Abraham Söyler     |
| Mittwochs   | 12 – 14     | 02        | K 044 | Lilli Walter       |
|             |             | 05        | K 046 | Niclas Schwarzlose |
| Donnerstags | 12 – 14     | 01        | K 044 | Leonard König      |
|             |             | 08        | K 046 | Andre Schröder     |
| Donnerstags | 14 – 16     | 04        | K 044 | Andre Schröder     |
|             |             | 07        | K 046 | Niclas Schwarzlose |
| Donnerstags | 16 – 18     | 03        | K 044 | Leonard König      |
|             |             | 10        | K 046 | Abraham Söyler     |

Wunschkonzert: jeden Freitag, 14 – 16 Uhr, SR 055



# Organisation - Übungszettel

## Übungen

- “Theoretischer” Anteil
- Praktischer Anteil (NASM)
- Alle an den Poolrechnern lösbar

## Ausgabe / Abgabe

- Wöchentliche Ausgabe, zweiwöchige Bearbeitung
- Ausgabe vor der VL über KVV
- Abgabe vor der VL im Tutorenfach (erster Stock, Takustr. 9), oder als PDF (!) im KVV
- Quellcode (kommentiert!) im KVV
- Verspätete Abgaben werden ignoriert

## Besprechung

- In den Tutorien
- In der Woche nach der Abgabe



## Organisation – Klausur

### Varianten

- 120min Rechnerarchitektur & Betriebs- und Kommunikationssysteme Modulprüfung
  - d.h. für die Mehrzahl keine Klausur im Februar sondern im Juli!
- Neue Studienordnung Lehramt/Erasmus: 60min Rechnerarchitektur

### Klausur

- TBA

≥ 50% der Punkte müssen in der Klausur erreicht werden

- Die Klausur bestimmt die Endnote

**Nicht vergessen: Registrierung im Campus Management!**



## Spielregeln

Aktive Teilnahme in den Tutorien wird vorausgesetzt!

- Jede/r muss in der Lage sein, die Lösung zu präsentieren
- Die Übungszettel müssen erfolgreich bearbeitet werden
  - Erfolgreich = die Aufgaben wurden vollständig mit einem eigenen Lösungsansatz bearbeitet
  - Zwei Aufgaben pro Zettel gehen in die Bewertung ein
  - N-2 Zettel müssen am Ende erfolgreich bearbeitet sein

Passive Teilnahme in den Tutorien wird vorausgesetzt!

- Mindestens n-2 mal anwesend sein

Teamwork

- Nur in 2er Gruppen abgeben



# AUFBAU DER VERANSTALTUNG





# Contents

## Organizational

- Organizational Information
- Course overview

## Introduction

- Single Processor Systems
- Historical overview
- Six-level computer architecture

## Data representation and Computer arithmetic

- Data and number representation
- Basic arithmetic

## Microarchitecture

- Microprocessor architecture
- Microprogramming
- Pipelining
- Very Long Instruction Word

## Instruction Set Architecture

- CISC vs. RISC
- Data types, Addressing, Instructions
- Assembler

## Memories

- Hierarchy, Types
- Physical & Virtual Memory
- Segmentation & Paging
- Caches

## Computer Systems

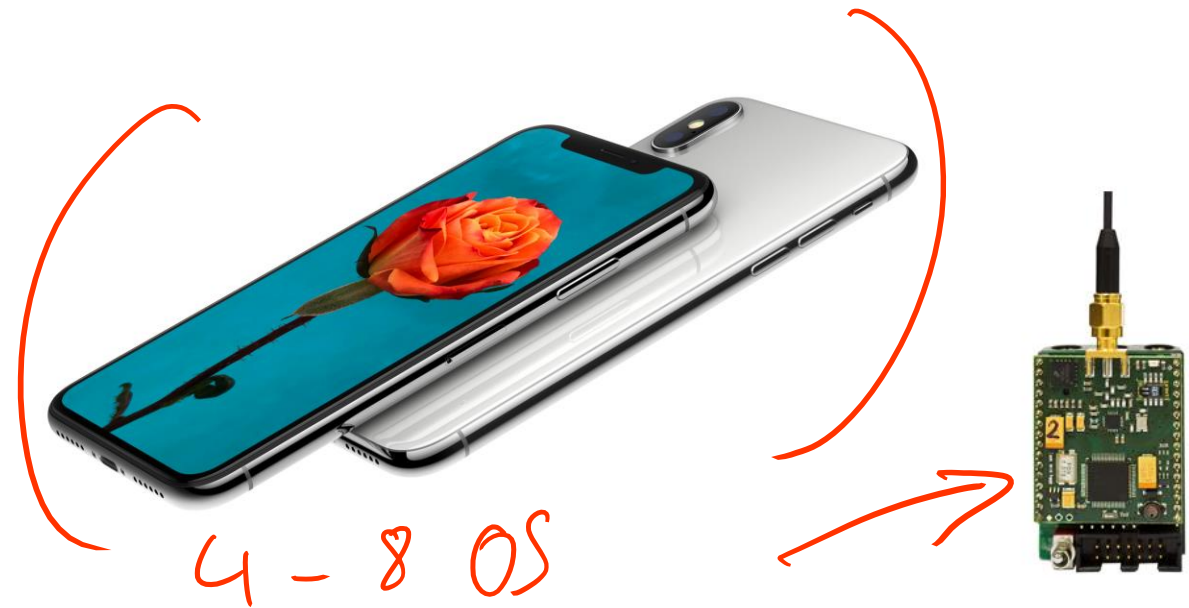
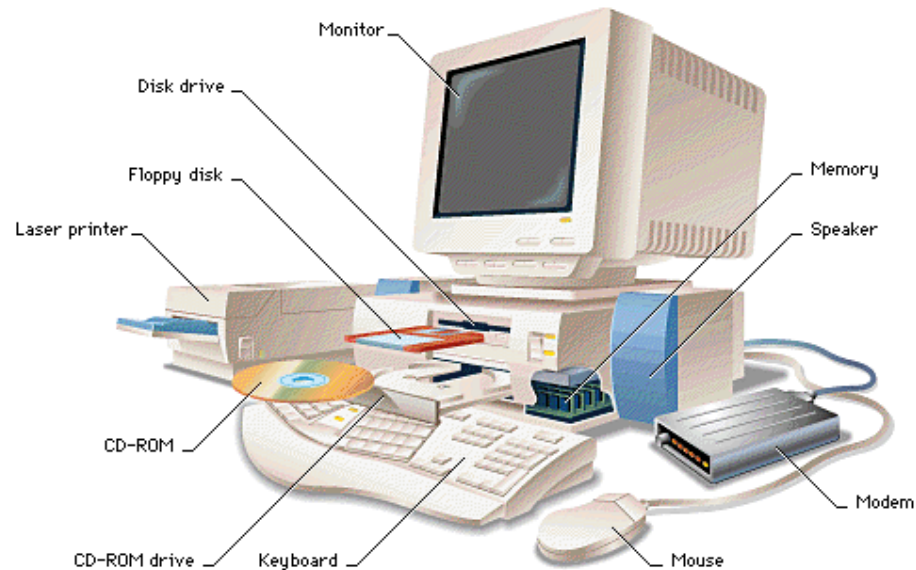
- Interconnects
- Components, Peripherals
- Classical PCs
- Parallel Systems, Clusters, Multiprocessors



# Topics of this Course

## Some questions

- Who did open the case of a computer?
- Who did assemble a computer from components?
- Who did write a program in Java?
- Who did write a program in C?
- Who did write a program in Assembler? ←



## Topics of this Course

At the end of this course, you should ...

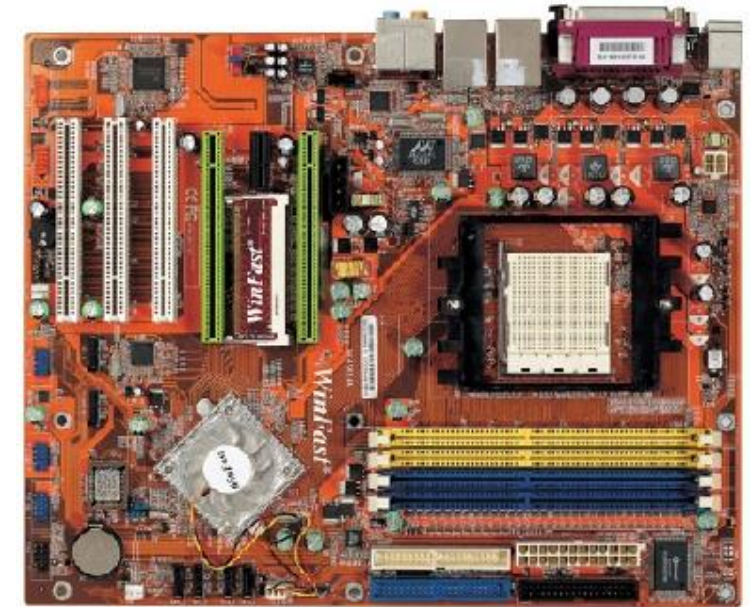
- know the different components of a computer system
- know the internals of a computer
- know how a computer stores data, i.e., text, audio, video
- know how a program is executed
- be able to write small assembler programs
- understand basic arithmetic

This should help you to

- understand computers in general
- understand how high-level programming languages are translated into machine language
- improve your programming skills



<https://de.ifixit.com/Teardown/iPhone+8+Teardown/97481>



$$C = a + b$$

↑  
?



# Literature

The course follows (roughly) the books:

- A. S. Tanenbaum, T. Austin:  
Structured Computer Organization, 6. edition,  
Pearson, 2013

Rechnerarchitektur, 6. Auflage, Pearson Studium,  
2014

- D. A. Patterson, J. L. Hennesy: Computer  
Organization and Design, 5. edition, Morgan  
Kaufmann, 2013

Rechnerorganisation und -entwurf, 5. Auflage, De  
Gruyter, 2016

