

# 64-040 Modul IP7: Rechnerstrukturen

## 0. Info zur Organisation

Norman Hendrich

Universität Hamburg  
MIN Fakultät, Fachbereich Informatik  
Vogt-Kölln-Str. 30, D-22527 Hamburg  
hendrich@informatik.uni-hamburg.de

<https://tams.informatik.uni-hamburg.de/lectures/2015ws/vorlesung/rs>

WS 2015/2016

# Inhalt

## 1. Übersicht

Themen

Terminplanung

## 2. Organisatorisches

Vorlesung

Übungen

Tutorium

Praktikum

Klausur

## 3. Literaturempfehlungen

## 4. Software



## Modul Rechnerstrukturen: Motivation

*Das Pflichtmodul Rechnerstrukturen vermittelt ein begrenztes und wohl ausgewähltes Theorie- und Methodenrepertoire für Konfigurierung, Entwurf, Realisierung und angemessene Nutzung von Rechnern und Kommunikationsnetzen sowie ihrer Basiskomponenten unter Berücksichtigung technologischer, ökonomischer und anwendungsspezifischer Randbedingungen.*

*... Grundlegende Konzepte, Organisationsformen und Entwurfsmethoden von Rechnerarchitekturen und deren Vernetzung, einschließlich der Betriebssoftware. . .*

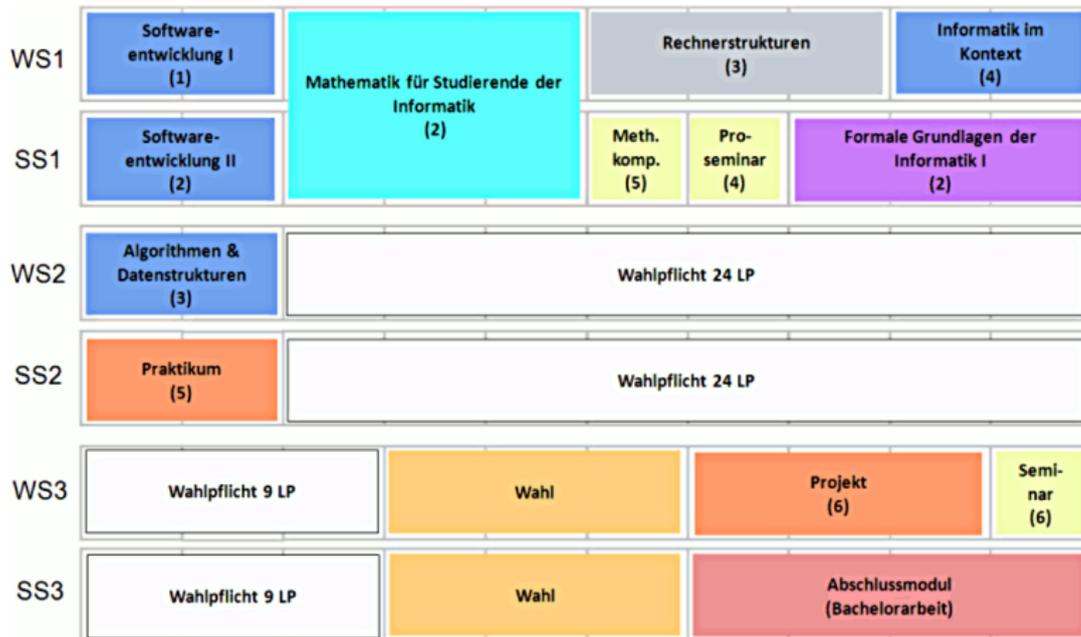
<https://www.inf.uni-hamburg.de/studies/orga/mhb/modulhandbuch-2013.pdf>



## Modul Rechnerstrukturen: Lernziele

- ▶ Leitbild: *eine der tragenden Säulen der Informatik*
- ▶ Faktenwissen: *Grundkenntnisse über digitale Rechner*
- ▶ Methodenwissen: *Analyse und Synthese von technischen Systemen*
- ▶ Transferkompetenz: *Anwendungen der Methoden der technischen Informatik, Verständnis von zeitlichen Abfolgen, Zusammenspiel von Software- und Hardwarekomponenten*
- ▶ Normativ-bewertende Kompetenz: *Urteilsvermögen zur Analyse von Rechnerarchitekturen und deren Komponenten*
- ▶ Schlüsselqualifikationen: *Kooperations- und Teamfähigkeit, Problemlösungskompetenz, Abstraktionsvermögen, Anwendung von Entwurfsmethoden, Befähigung zur Analyse und Synthese von technischen Systemen*

# Informatik-BSc: Übersicht



# Software-System-Entwicklung-BSc: Übersicht

WS1	Software-entwicklung I (1)	Informatik im Kontext (3)	Rechnerstrukturen (3)	Mathematik für Studierende der Informatik (2)	
SS1	Software-entwicklung II (2)	Meth. komp. (5)	Pro-seminar (4)		
WS2	Software-entwicklungs-praktikum (5)	Grundlagen von Datenbanken (5)	Projekt-manag. (5)	Wahlpflicht Informatik	Wahlpflicht Informatik
SS2	Softwaretechnik (4)	Wahlpflicht Informatik	Wahl / Anwendungsgebiet	Projekt (6)	
WS3	Wahl / Anwendungsgebiet	Seminar (6)	Industriepraktikum (6)		
SS3	Wahlpflicht Informatik	Wahl / Anwendungsgebiet	Abschlussmodul (Bachelorarbeit)		



# Rechnerstrukturen: Themen

- ▶ Einführung
- ▶ Grundprinzip des von-Neumann Rechners
- ▶ Abstraktionsschichten, virtuelle Maschinen
- ▶ Hardware-/Software-Schnittstelle
  
- ▶ Informationsbegriff und -theorie, Codierung
- ▶ Entwurf von digitalen Schaltungen
- ▶ Grundkomponenten des Rechners, Realisierungsaufwand
- ▶ Rechnerarchitektur
- ▶ Assemblerprogrammierung

# Themenübersicht

## Teil 1

- ▶ Information und Repräsentation
- ▶ Zahldarstellung und Arithmetik
- ▶ Boole'sche Algebra und -Funktionen
- ▶ Schaltnetze und Schaltwerke
- ▶ Entwurf digitaler Schaltungen
- ▶ Komponenten der Register-Transfer Ebene

## Teil 2

- ▶ Rechnerarchitektur
- ▶ Befehlssätze, CISC- und RISC-Architektur
- ▶ Maschinen- und Assemblerprogrammierung
- ▶ Speicherhierarchie und Virtueller Speicher
- ▶ I/O-Operationen, Interrupts
- ▶ Leistungsbewertung, Pipelining, Parallelrechner

# Terminübersicht Teil 1

14.10	16.10	Einführung, von-Neumann Konzept
21.10	23.10	Information, Zahldarstellung
28.10	30.10	Arithmetik, Textcodierung
04.11	06.11	Informationstheorie, Optimalcodes
11.11	13.11	Boole'sche Algebra, bitweise Operationen
18.11	20.11	Schaltfunktionen, Normalformen
25.11	27.11	Schaltnetze, Gatter, Rechenwerke
02.12	04.12	Schaltwerke, Flipflops, Entwurf von Schaltungen

## Terminübersicht Teil 2

09.12	11.12	Register-Transfer-Ebene, ISA, Befehlsformate
16.12	18.12	x86-Architektur, Assemblerprogrammierung
06.01	08.01	Funktionsaufrufe, Stack
13.01	15.01	Datenstrukturen, Speicherhierarchie
20.01	22.01	Speicherhierarchie, Virtueller Speicher
27.01	29.01	Pipeline, Parallelrechner



# Vorlesung

- ▶ Mi. 16:15-17:45 ESA Hörsaal B, Edmund-Siemers-Allee 1  
 Fr. 12:15-13:45 Audimax II , von-Melle Park (4 SWS)
- ▶ Folien (pdf) und Materialien werden gestellt
- ▶ Lecture2Go: [lecture2go.uni-hamburg.de](http://lecture2go.uni-hamburg.de)
- ▶ diverse gute Lehrbücher verfügbar — Empfehlungen s.u.
- ▶ eingestreute Hinweise auf aktuelle Themen und Vertiefung
  
- ▶ Assemblerprogrammierung: Linux x86 mit GNU-Toolchain
- ▶ optional unter Windows: Cygwin mit GNU-Toolchain
  
- ▶ Informationen und Downloads auf der Webseite:

# Übungen

- ▶ mehrere Gruppen, Details siehe KVV/Webseite/STiNE
- ▶ Anmeldung über STiNE
- ▶ 1 SWS, wöchentlich
  
- ▶ Diskussion und Nachbereitung der Vorlesung
- ▶ Gruppenarbeit erwünscht (max. 3 Teilnehmer pro Gruppe)
  
- ▶ Übungsaufgaben zum Vertiefen und Erarbeiten des Stoffes
- ▶ Aufgabenblätter jeweils Mittwoch zum Download verfügbar
- ▶ Abgabe der Lösungen schriftlich bis nächsten Mittwoch 24:00 beim Gruppenleiter oder im TAMS-Sekretariat

## Übungen: Scheinkriterien

- ▶ regelmässige aktive Teilnahme
- ▶ mindestens zweimal an der Tafel vorrechnen
- ▶ höchstens zweimal (unentschuldigt) gefehlt
  
- ▶ alle Aufgabenblätter bearbeitet
- ▶ jeweils mindestens 30 % der Punkte pro Aufgabenblatt
- ▶ und mindestens 50 % der Gesamtpunktzahl

## Übungen: Gruppeneinteilung und Wechselwünsche

- ▶ Übungen beginnen in der zweiten Vorlesungswoche  
Achtung: erstes Übungsblatt ist bereits online!
- ▶ Anmeldung und Gruppenzuordnung über STiNE
- ▶ Auslastung derzeit noch nicht bekannt
- ▶ jeweils zwei Gruppen parallel: einfache Wechselmöglichkeit
- ▶ Ausgleich der Gruppengröße in der ersten Woche
- ▶ Wechselwünsche: bitte zur gewünschten Gruppe erscheinen, vor Ort in Warteliste eintragen
- ▶ späterer Wechsel bei freien Plätzen nach Rücksprache mit den Gruppenleitern möglich (Übertragung der Punkte, etc.)

# Tutorium

- ▶ freiwillig: Nacharbeiten von Stoff und Übungen
- ▶ mehr Zeit für Diskussion und Fragen als in der Übung
  
- ▶ ab zweiter Woche, wöchentlich
- ▶ Do. 18:30 Informatikum, F-132
  
- ▶ Paniktutorium: Vorbereitung auf die Klausur
- ▶ Januar 2016, Ort und Zeit wird bekanntgegeben

# Praktikum

- ▶ Blockkurse á 4 Termine zu je 3 h (1 SWS)
- ▶ Vorbereitung **unbedingt** erforderlich
- ▶ Erarbeiten und Programmieren eines einfachen Prozessors
- ▶ Komponenten auf der Register-Transfer-Ebene
- ▶ Zeitverhalten, Speicheransteuerung
- ▶ Mikroprogrammierung
- ▶ Assemblerprogrammierung
- ▶ I/O-Operationen und Interrupts

<http://tams.informatik.uni-hamburg.de/lectures/2015ws/praktikum/rechprak/>

# Klausur

- ▶ Abschlussprüfung des Moduls Rechnerstrukturen
  - ▶ Note geht in BSc-Zeugnis ein
  
  - ▶ insgesamt zwei Klausurtermine in den Semesterferien
  - ▶ Do 04/02/2016, 10:30-12:30, Audimax-I
  - ▶ Mi 02/03/2016, 09:30-11:30, ESA Hörsaal A
- pünktlich kommen!**

Tipp: möglichst bereits die erste Klausur mitschreiben, die zweite Klausur fällt regelmäßig schlechter aus.



# Probeklausur

- ▶ Umfang und Fragen wie „echte“ Klausur
- ▶ anschließende gemeinsame Korrektur/Besprechung
  
- ▶ eventuell im Rahmen des Tutoriums (2-stündig)
- ▶ Termin: vorletzte Semesterwoche?



## Feedback erwünscht

Hinweise auf Fehler und Ungereimtheiten in den Folien und Materialien sind ausdrücklich erwünscht! Nur keine Scheu, und bitte auch rechtzeitig Rückmeldung bei Unklarheiten in den Übungsaufgaben.

Außerdem: Vorschläge und Hinweise auf nützliche Tools oder schöne Lehrmaterialien zum Thema der Vorlesung sind immer willkommen!

[hendrich@informatik.uni-hamburg.de](mailto:hendrich@informatik.uni-hamburg.de)  
[maeder@informatik.uni-hamburg.de](mailto:maeder@informatik.uni-hamburg.de)

# Kontakt

Dr. Norman Hendrich  
 hendrich@informatik.uni-hamburg.de  
 +4940 42883 2399  
 Informatikum, Haus F314

Dr. Andreas Mäder  
 maeder@informatik.uni-hamburg.de  
 +4940 42883 2502  
 Informatikum, Haus F317

## Literatur: Empfohlene Lehrbücher

- ▶ Randal E. Bryant and David O'Hallaron,  
*Computer Systems — A programmers perspective*,  
 Pearson Prentice Hall, 2nd. Ed., 2014

Rechnerarchitektur mit Schwerpunkt Software und Systeme, leider nicht ganz billig. Viele C-Programme und Systemprogrammierung. Beispiele anhand Intel x86 Architektur.

- ▶ Andrew S. Tanenbaum,  
*Structured Computer Organization / Computerarchitektur*,  
 6th. edition, Pearson Prentice Hall, 2013

Guter Überblick, klares didaktisches Konzept. Java VM, Intel x86, SPARC.



## Literatur: weitere Lehrbücher

- ▶ David A. Patterson and John L. Hennessy, *Computer Organization and Design — the hardware/software interface*, 5th edition, Morgan Kaufmann, 2014

Schönes Lehrbuch von den Entwicklern der RISC/MIPS Prozessoren.

- ▶ David A. Patterson and John L. Hennessy, *Rechnerorganisation und -entwurf — die Hardware/Software-Schnittstelle* 4. Auflage, Spektrum Verlag, 2011

Die deutsche Übersetzung, leider eine Auflage zurück: die aktuellen Ergänzungen wie z.B. Multi-Core Maschinen fehlen. Mehrere Exemplare in der Informatik-Bibliothek.

## Literatur: weitere Lehrbücher

- ▶ Schiffmann und Schmitz,  
*Technische Informatik 1, Grundlagen der digitalen Elektronik*  
*Technische Informatik 2, Grundlagen der Computertechnik*  
Springer Verlag, 2004, 2005
- ▶ Schiffmann und Schmitz,  
*Übungsbuch zur Technischen Informatik 1 und 2*  
Springer Verlag, 2003  
  
Dutzende von Übungsaufgaben mit detailliert entwickelten Lösungen.
- ▶ Miles Murdocca and Vincent Heuring,  
*Computer Architecture and Organization, An Integrated Approach* John Wiley and Sons, 2007



## Literatur: Assemblerprogrammierung

- ▶ Randy Hyde,  
*Art Of Assembly Language Programming*,  
online zum Download verfügbar unter  
<http://homepage.mac.com/randyhyde/webster.cs.ucr.edu/www.artofasm.com/index.html>

## Literatur: Vertiefung

- ▶ John L. Hennessy and David A. Patterson,  
*Computer Architecture — A Quantitative Approach*,  
 5th. edition, Morgan Kaufmann, 2012  
 Die Bibel zum Thema Rechnerarchitektur
- ▶ Donald E. Knuth, *The Art of Computer Programming:*  
*Volume 1 Fascicle 0: MMIX*  
*Volume 4 Fascicle 0: Boolean Functions*  
*Volume 4 Fascicle 1: Bitwise Tricks and Techniques, Binary  
 Decision Diagrams* Addison-Wesley, 2006-2009
- ▶ Giovanni de Micheli *Synthesis and Optimization of Digital  
 Circuits*, McGraw-Hill, 1994

# Software

- ▶ Java-VM für diverse Programmierbeispiele
- ▶ Hades Schaltungssimulator

<http://tams.informatik.uni-hamburg.de/applets/hades/>

- ▶ Assembler und Tools, Debugger
- ▶ C-Compiler
  
- ▶ GNU-Toolchain empfehlenswert: gcc, binutils, gdb
- ▶ Insight-Debugger als Frontend zu gdb
- ▶ Unter Windows: Cygwin mit Development-Tools installieren

<http://www.cygwin.com/>

- ▶ Links und weitere Infos auf der Webseite zur Vorlesung