

# 64-040 Modul IP7: Rechnerstrukturen

[http://tams.informatik.uni-hamburg.de/  
lectures/2011ws/vorlesung/rs](http://tams.informatik.uni-hamburg.de/lectures/2011ws/vorlesung/rs)

Andreas Mäder



Universität Hamburg  
Fakultät für Mathematik, Informatik und Naturwissenschaften  
Fachbereich Informatik

**Technische Aspekte Multimodaler Systeme**

Wintersemester 2011/2012



# Inhalt

## Übersicht

Themen

Terminplanung

## Organisatorisches

Vorlesung

Übungen

Praktikum

Tutorium

Klausur

## Literaturempfehlungen

## Software





## Modul Rechnerstrukturen: Motivation

*Die Ausbildung im Pflichtmodul Rechnerstrukturen vermittelt im Rahmen der Vorlesung ein begrenztes und wohl ausgewähltes Theorie- und Methodenrepertoire für die Konfigurierung, den Entwurf, die Realisierung, und die angemessene Nutzung von Rechnern- und Kommunikationsnetzen sowie ihrer Basiskomponenten, u. a. unter Berücksichtigung technologischer, ökonomischer und anwendungsspezifischer Randbedingungen. . . .*

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

[www.informatik.uni-hamburg.de/Info/Studium/BSc/Module/Rechnerstrukturen.shtml](http://www.informatik.uni-hamburg.de/Info/Studium/BSc/Module/Rechnerstrukturen.shtml)

## 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

WS1	Softwareentwicklung I (1)	Mathematik für Studierende der Informatik (2)			Rechnerstrukturen (3)		Informatik im Kontext (4)	
SS1	Softwareentwicklung II (2)	Mathematik für Studierende der Informatik (2)			Meth. komp. (5)	Proseminar (4)	Formale Grundlagen der Informatik I (2)	
WS2	Algorithmen & Datenstrukturen (3)	Wahlpflicht						
SS2	Praktikum (5)	Wahlpflicht						
WS3	Wahlpflicht		Wahl		Projekt (6)		Seminar (6)	
SS3	Wahlpflicht		Wahl		Abschlussmodul (Bachelorarbeit)			

# 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)	Formale Grundlagen der Informatik I (2)		
WS2	Software-entwicklungs-praktikum (5)	Grundlagen von Datenbanken (5)	Projekt-manag. (5)	Wahlpflicht Informatik	Wahlpflicht Informatik	
SS2	Software-technik (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

# Themenübersicht

- ▶ Information und Repräsentation
- ▶ Zahldarstellung und Arithmetik
- ▶ Boole'sche Algebra und -Funktionen
- ▶ Schaltnetze und Schaltwerke
- ▶ Entwurf digitaler Schaltungen
- ▶ Komponenten der Register-Transfer Ebene
- ▶ Rechnerarchitektur
- ▶ Befehlssätze, CISC- und RISC-Architektur
- ▶ Maschinen- und Assemblerprogrammierung
- ▶ Speicherhierarchie und Speicherverwaltung
- ▶ I/O-Operationen, Interrupts
- ▶ Leistungsbewertung, Parallelrechner



# Terminübersicht

19.10	21.10	Einführung, von-Neumann Konzept
26.10	28.10	Information, Zahldarstellung
02.11	04.11	Arithmetik, Textcodierung
09.11	11.11	Informationstheorie, Optimalcodes
16.11	18.11	Boole'sche Algebra, bitweise Operationen
23.11	23.11	Schaltfunktionen, Normalformen
30.11	02.12	Schaltnetze, Gatter, Rechenwerke
07.12	09.12	Schaltwerke, Flipflops, Entwurf von Schaltungen
14.12	16.12	Register-Transfer-Ebene, ISA, Befehlsformate
21.12	23.12	x86-Architektur, Assemblerprogrammierung



## Terminübersicht (cont.)

11.01	13.01	Funktionsaufrufe, Stack
18.01	20.01	Datenstrukturen, Speicherhierarchie
25.01	27.01	Speicherhierarchie, Virtueller Speicher
02.02	05.02	Pipeline, Parallelrechner



## Feedback erwünscht

Bitte alle Fehler und Ungenauigkeiten in den Folien und Materialien melden. Ebenso bitte Feedback bei Unklarheiten etc.!

Vorschläge und Hinweise auf Tools, schöne Lehrmaterialien etc. sind immer willkommen!

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



# Kontakt

Dr. Andreas Mäder  
maeder@informatik.uni-hamburg.de  
+49 40 42883 2502  
Informatikum, Haus F-317



# Vorlesung

[tams.informatik.uni-hamburg.de/lectures/2011ws/vorlesung/rs](https://tams.informatik.uni-hamburg.de/lectures/2011ws/vorlesung/rs)

- ▶ Mi. 16:15-17:45 Erzwiss H, VMP-8  
Fr. 14:15-15:45 Phil D, VMP-6 (4 SWS)
- ▶ Skript (Folien) und Materialien werden gestellt
- ▶ Überarbeitung während des Semesters
- ▶ diverse gute Lehrbücher verfügbar — Empfehlungen s.u.
- ▶ Assemblerprogrammierung: x86 mit GNU-Toolchain
- ▶ unter Windows: Cygwin mit GNU-Toolchain
- ▶ eingestreute Hinweise auf aktuelle Themen und Vertiefung
- ▶ Informationen und Downloads auf der Webseite



# Übungen

[tams.informatik.uni-hamburg.de/lectures/2011ws/vorlesung/rs/uebungen](https://tams.informatik.uni-hamburg.de/lectures/2011ws/vorlesung/rs/uebungen)

- ▶ sechzehn Gruppen geplant, Details siehe KVV/Webseite
- ▶ 1 SWS, wöchentlich
  
- ▶ Diskussion und Nachbereitung der Vorlesung
- ▶ Gruppenarbeit erwünscht (max. 3 Teilnehmer pro Gruppe)
  
- ▶ Übungsaufgaben zum Vertiefen und Erarbeiten des Stoffes
- ▶ Aufgabenblätter jeweils Freitag zum Download verfügbar
- ▶ Abgabe der Lösungen schriftlich/email bis nächsten Freitag 12:00 beim Gruppenleiter oder im TAMS-Sekretariat



## Übungen: Scheinkriterien

- ▶ [tams.informatik.uni-hamburg.de/lectures/2011ws/vorlesung/rs/uebungen/scheinkriterien.pdf](https://tams.informatik.uni-hamburg.de/lectures/2011ws/vorlesung/rs/uebungen/scheinkriterien.pdf)
- ▶ regelmäßige aktive Teilnahme
- ▶ mindestens zweimal an der Tafel vorrechnen
- ▶ höchstens zweimal (entschuldigt) gefehlt
  
- ▶ alle Aufgabenblätter bearbeitet
- ▶ jeweils mindestens 30 % der Punkte pro Aufgabenblatt
- ▶ und mindestens 50 % der Gesamtpunktzahl



## Übungen: Gruppeneinteilung / Wechselwünsche?

- ▶ Übungen beginnen in der zweiten Vorlesungswoche
- ▶ Anmeldung und Ranking letzte Woche über STiNE
- ▶ Auslastung derzeit ca. 11 Gruppen
- ▶ 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.)



# Übungen: Gruppeneinteilung / Wechselwünsche? (cont.)

## Derzeitige Belegung

19.10. 11:00

Mo	12-13	F334	F009	40	×2
	13-14	F334	F009	35	×2
	14-15	F334	F534	40	×2
	15-16	F334	F534 *	15	×1
Di	10-11	F334	F009 *	12	×1
	11-12	F334 *	F009 *	1	×0
	12-13	F334	F534	26	×2
	13-14	F334	F534 *	21	×1

\* streichen ?



# Praktikum

[tams.informatik.uni-hamburg.de/lectures/2011ws/praktikum/rechprak](https://tams.informatik.uni-hamburg.de/lectures/2011ws/praktikum/rechprak)

- ▶ im laufenden Semester  
in der vorlesungsfreien Zeit (Februar, März) **empfohlen**
- ▶ 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



# Tutorium

[tams.informatik.uni-hamburg.de/lectures/2011ws/vorlesung/rs/tutorium](https://tams.informatik.uni-hamburg.de/lectures/2011ws/vorlesung/rs/tutorium)

- ▶ freiwillig, Nacharbeiten von Stoff und Übungen
- ▶ Diskussion, Fragen: mehr Zeit als in der Übung

## 1. ab nächster Woche, wöchentlich — Alternativen:

- ? Do. 16-17 – nach SE1-Übungen
- ? Di. 14-15 – parallel zu SE1-Übungen
- ? Mo. 14-15 – parallel zu RS-Übungen
- Mo. 15-16 – –"–
- ▶ Di. 12-13 – parallel zu RS-Übungen
- Di. 13-14 – –"–

Präferenz in Vorlesung 19.10.

wahrscheinlicher Termin

## 2. Vorbereitung auf die Klausur: „Paniktutorium“

# Klausur

[www.informatik.uni-hamburg.de/StB/klausurtermine.shtml](http://www.informatik.uni-hamburg.de/StB/klausurtermine.shtml)

- ▶ Abschlussprüfung des Moduls Rechnerstrukturen
- ▶ Note geht in BSc-Zeugnis ein
- ▶ insgesamt zwei Klausurtermine in den Semesterferien
- ▶ 13. Feb. 2012 10:00-13:00 Audimax 1
- ▶ 22. März 2012 12:00-15:00 Audimax 1

**pünktlich kommen!**

- ▶ Tipp: möglichst bereits die erste Klausur mitschreiben
- ▶ zweite Klausur fällt oft schlechter aus (leider)



## Meinungsbild: Probeklausur?!

- ▶ häufig gewünscht, aber aufwändig...
- ▶ Entscheidung fällt während des Semesters
- ▶ Umfang und Fragen wie „echte“ Klausur
- ▶ vermutlich im Rahmen des Tutoriums (2-stündig)
- ▶ Korrektur durch Übungsgruppenleiter?
- ▶ Besprechung in den Übungen
- ▶ Termin: vorletzte Semesterwoche?



## Literatur: empfohlene Lehrbücher

- ▶ Randal E. Bryant, David R. O'Hallaron,  
*Computer systems : A programmers perspective*,  
2nd edition, Pearson, 2011  
Rechnerarchitektur mit Schwerpunkt Software und Systeme,  
leider nicht ganz billig. Viele C-Programme und Systemprogrammierung. Beispiele anhand Intel x86  
Architektur. Keine wesentlichen Änderungen gegenüber der Erstauflage von 2003.
- ▶ Andrew S. Tanenbaum,  
*Computerarchitektur: Strukturen, Konzepte, Grundlagen*,  
5th edition, Pearson, 2006  
Guter Überblick, klares didaktisches Konzept. Java VM, Intel x86, SPARC. Mit jeder Auflage komplett  
überarbeitet und aktualisiert.
- ▶ Andrew S. Tanenbaum,  
*Structured Computer Organization*,  
5th revised edition, Pearson, 2009



## Literatur: weitere Lehrbücher

- ▶ David A. Patterson, John L. Hennessy,  
*Computer Organization and Design :  
the hardware/software interface*,  
4th edition, Morgan Kaufmann, 2009
- ▶ David A. Patterson, John L. Hennessy,  
*Rechnerorganisation und -entwurf :  
die Hardware/Software-Schnittstelle*,  
3. Auflage, Elsevier / Spektrum Verlag, 2005

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 (cont.)

- ▶ Wolfram Schiffmann, Robert Schmitz,  
*Technische Informatik 1, Grundlagen der digitalen Elektronik,*  
*Technische Informatik 2, Grundlagen der Computertechnik,*  
5. Auflage, Springer Verlag, 2004, 2005
- ▶ Wolfram Schiffmann, Robert Schmitz, Jürgen Weiland,  
*Übungsbuch zur Technischen Informatik 1 und 2,*  
3. Auflage, Springer Verlag, 2004  
Dutzende von Übungsaufgaben mit detailliert entwickelten Lösungen.
- ▶ Miles Murdocca, Vincent Heuring,  
*Computer Architecture and Organization :*  
*an integrated approach,*  
John Wiley and Sons, 2007



## Literatur: aus Hamburg

- ▶ Klaus Lagemann,  
*Rechnerstrukturen*, Springer Verlag, 1987
- ▶ Dietmar Möller,  
*Rechnerstrukturen, Grundlagen der Technischen Informatik*,  
Springer Verlag, 2003
- ▶ Dietmar Möller, Martin Lehmann,  
*Vorlesung: Rechnerstrukturen*,  
Universität Hamburg, FB Informatik, 2008  
[www.informatik.uni-hamburg.de/TKRN/world/abro/RS/RS.html](http://www.informatik.uni-hamburg.de/TKRN/world/abro/RS/RS.html)
- ▶ Andreas Mäder,  
*Vorlesung: Rechnerarchitektur und Mikrosystemtechnik*,  
Universität Hamburg, FB Informatik, 2010  
[tams.informatik.uni-hamburg.de/lectures/2010ws/vorlesung/ram](http://tams.informatik.uni-hamburg.de/lectures/2010ws/vorlesung/ram)



## Literatur: Assemblerprogrammierung

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



## Literatur: Vertiefung

- ▶ John L. Hennessy, David A. Patterson,  
*Computer Architecture : A Quantitative Approach*,  
4th. edition, Morgan Kaufmann, 2007  
Die Bibel zum Thema Rechnerarchitektur
- ▶ Donald E. Knuth,  
*The Art of Computer Programming, Volume 1, Fascicle 1:  
MMIX – A RISC Computer for the New Millenium*,  
Addison-Wesley, 2005



## Literatur: Vertiefung (cont.)

- ▶ Donald E. Knuth,  
*The Art of Computer Programming, Volume 4, Fascicle 0: Introduction to Combinatorial Algorithms and Boolean Functions*,  
Addison-Wesley, 2008
- ▶ Donald E. Knuth,  
*The Art of Computer Programming, Volume 4, Fascicle 1: Bitwise Tricks & Techniques, Binary Decision Diagrams*,  
Addison-Wesley, 2009
- ▶ Giovanni de Micheli  
*Synthesis and Optimization of Digital Circuits*,  
McGraw-Hill, 1994

## Literatur: über den Tellerrand hinaus

- ▶ Reiner Hartenstein,  
*Standort Deutschland: Wozu noch Mikro-Chips*,  
IT-Press Verlag, 1994 (vergriffen)
- ▶ Tracy Kidder,  
*The soul of a new machine*,  
diverse Verlage, 1981
- ▶ Jeff Hawkins,  
*On Intelligence*,  
Times Books, 2004
- ▶ Raul Rojas,  
*Neural Networks : A Systematic Introduction*,  
Springer, 1996  
[page.mi.fu-berlin.de/rojas/neural](http://page.mi.fu-berlin.de/rojas/neural)



# Software

- ▶ Java-VM für diverse Programmierbeispiele
- ▶ HADES Schaltungssimulator — HAMBURG DEsign System  
[tams.informatik.uni-hamburg.de/applets/hades](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  
[www.cygwin.com](http://www.cygwin.com)
- ▶ Links und weitere Infos auf der Webseite zur Vorlesung