
M.Sc. Informationssystemtechnik Basis-Module (PO 2015)

Modulhandbuch
Stand: 01.03.2021



TECHNISCHE
UNIVERSITÄT
DARMSTADT

Studienbereich Informationssystem-
technik

Dieses Modulhandbuch listet nur die Grundlagen. Für die Wahlbereiche Vertiefungen und Anwendungen gibt es ein eigenes Modulhandbuch.

Modulhandbuch: M.Sc. Informationssystemtechnik Basis-Module (PO 2015)

Stand: 01.03.2021

Studienbereich Informationssystemtechnik
Email: studienberatung@ist.tu-darmstadt.de

Inhaltsverzeichnis

1	Vertiefungen - Grundlagen	1
1.1	Wahlkatalog KTS: Kommunikationstechnik und -systeme	1
	Kommunikationsnetze II	1
	Kommunikationstechnik I	3
	TK3: Ubiquitous / Mobile Computing	5
1.2	Wahlkatalog SES: System on Chip und Eingebettete Systeme	7
	Algorithmen für Hardware-Entwurfswerkzeuge	7
	Praktikum zu Algorithmen für Hardware-Entwurfswerkzeuge	9
	Advanced Digital Integrated Circuit Design	10
	Architekturen und Entwurf von Rechnersystemen	12
	Rechnersysteme II	14
1.3	Wahlkatalog SWE: Software-Engineering	15
	Software-Engineering - Wartung und Qualitätssicherung	15
	Einführung in den Compilerbau	17
	Echtzeitsysteme	18
	Compiler Tooling	19
2	Vertiefungen / Anwendungen	21
3	Studium Generale	22

1 Vertiefungen - Grundlagen

1.1 Wahlkatalog KTS: Kommunikationstechnik und -systeme

Modulname Kommunikationsnetze II					
Modul-Nr. 18-sm-2010	Kreditpunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1	Angebotsturnus WiSe
Sprache Englisch			Modulverantwortliche Person Prof. Dr.-Ing. Ralf Steinmetz		
1	<p>Lerninhalt</p> <p>Die Vorlesung Kommunikationsnetze II umfasst die Konzepte der Computervernetzung und -telekommunikation mit dem Fokus auf dem Internet. Beginnend mit der Geschichte werden in der Vorlesung vergangene, aktuelle und zukünftige Aspekte von Kommunikationsnetzen behandelt. Zusätzlich zu bekannten Protokollen und Technologien wird eine Einführung in Neuentwicklungen im Bereich von Multimedia Kommunikation (u.a. Video Streaming, P2P, IP-Telefonie, Cloud Computing und Service-orientierte Architekturen) gegeben. Die Vorlesung ist als Anschlussvorlesung zu Kommunikationsnetze I geeignet.</p> <p>Themen sind:</p> <ul style="list-style-type: none"> • Grundlagen und Geschichte von Kommunikationsnetzen (Telegrafie vs. Telefonie, Referenzmodelle, ...) • Transportschicht (Adressierung, Flusskontrolle, Verbindungsmanagement, Fehlererkennung, Überlastkontrolle, ...) • Transportprotokolle (TCP, SCTP) • Interaktive Protokolle (Telnet, SSH, FTP, ...) • Elektronische Mail (SMTP, POP3, IMAP, MIME, ...) • World Wide Web (HTML, URL, HTTP, DNS, ...) • Verteilte Programmierung (RPC, Web Services, ereignisbasierte Kommunikation) • SOA (WSDL, SOAP, REST, UDDI, ...) • Cloud Computing (SaaS, PaaS, IaaS, Virtualisierung, ...) • Overlay-Netzwerke (unstrukturierte P2P-Systeme, DHT-Systeme, Application Layer Multicast, ...) • Video Streaming (HTTP Streaming, Flash Streaming, RTP/RTSP, P2P Streaming, ...) • VoIP und Instant Messaging (SIP, H.323) 				
2	<p>Qualifikationsziele / Lernergebnisse</p> <p>Die Vorlesung Kommunikationsnetze II umfasst die Konzepte der Computervernetzung und -telekommunikation mit dem Fokus auf dem Internet. Beginnend mit der Geschichte werden in der Vorlesung vergangene, aktuelle und zukünftige Aspekte von Kommunikationsnetzen behandelt. Zusätzlich zu bekannten Protokollen und Technologien wird eine Einführung in Neuentwicklungen im Bereich von Multimedia Kommunikation (u.a. Video Streaming, P2P, IP-Telefonie, Cloud Computing und Service-orientierte Architekturen) gegeben. Die Vorlesung ist als Anschlussvorlesung zu Kommunikationsnetze I geeignet.</p>				
3	<p>Empfohlene Voraussetzung für die Teilnahme</p> <p>Grundlegende Kurse der ersten 4 Semester werden benötigt. Die Vorlesung Kommunikationsnetze I wird empfohlen. Das Theoriewissen aus der Vorlesung Kommunikationsnetze II wird in praktischen Programmierübungen vertieft. Gundelegende Programmierkenntnisse sind daher hilfreich.</p>				
4	<p>Prüfungsform</p> <p>Modulabschlussprüfung:</p> <ul style="list-style-type: none"> • Modulprüfung (Fachprüfung, Klausur, Dauer: 120 min, Standard BWS) 				

5	Benotung Modulabschlussprüfung: • Modulprüfung (Fachprüfung, Klausur, Gewichtung: 100 %)		
6	Verwendbarkeit des Moduls MSc ETiT, MSc iST, Wi-ETiT, CS, Wi-CS		
7	Notenverbesserung nach §25 (2)		
8	Literatur Ausgewählte Kapitel aus folgenden Büchern: • Andrew S. Tanenbaum: Computer Networks, 5th Edition, Prentice Hall, 2010 • James F. Kurose, Keith Ross: Computer Networking: A Top-Down Approach, 6th Edition, Addison-Wesley, 2009 • Larry Peterson, Bruce Davie: Computer Networks, 5th Edition, Elsevier Science, 2011		
Enthaltene Kurse			
	Kurs-Nr. 18-sm-2010-vl	Kursname Kommunikationsnetze II	
	Dozent Prof. Dr.-Ing. Ralf Steinmetz, M.Sc. Philipp Achenbach, M.Sc. Tobias Meuser, M.Sc. Christoph Gärtner		Lehrform Vorlesung
			SWS 3
	Kurs-Nr. 18-sm-2010-ue	Kursname Kommunikationsnetze II	
	Dozent Prof. Dr.-Ing. Ralf Steinmetz, M.Sc. Philipp Achenbach, M.Sc. Tobias Meuser, M.Sc. Christoph Gärtner		Lehrform Übung
			SWS 1

Modulname Kommunikationstechnik I					
Modul-Nr. 18-kl-1020	Kreditpunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1	Angebotsturnus WiSe
Sprache Deutsch			Modulverantwortliche Person Prof. Dr.-Ing. Anja Klein		
1	Lerninhalt Signale und Übertragungssysteme, Basisbandübertragung, Detektion von Basisbandsignalen im Rauschen, Bandpass-Signale und -Systeme, Lineare digitale Modulationsverfahren, digitale Modulations- und Detektionsverfahren, Mehrträgerübertragung, OFDM, Bandspreizende Verfahren, CDMA, Vielfachzugriff				
2	Qualifikationsziele / Lernergebnisse Die Studierenden können nach Besuch der Lehrveranstaltung: <ul style="list-style-type: none"> • Signale und Übertragungssysteme klassifizieren, • Grundlegende Komponenten einfacher Übertragungssysteme verstehen, modellieren, analysieren und nach verschiedenen Kriterien optimal entwerfen. • Übertragungssysteme über ideale, mit weißem Gauß'schen Rauschen behaftete Kanäle verstehen, bewerten und vergleichen, • Basisband-Übertragungssysteme modellieren und analysieren, • Bandpass-Signale und Bandpass- Übertragungssysteme im äquivalenten Basisband beschreiben und analysieren, • lineare digitale Modulationsverfahren verstehen, modellieren, bewerten, vergleichen und anwenden, • Empfängerstrukturen für verschiedene Modulationsverfahren entwerfen • Linear modulierte Daten nach der Übertragung über ideale, mit weißem Gaußschen Rauschen behaftete Kanäle optimal detektieren, • OFDM verstehen und modellieren, • CDMA verstehen und modellieren, • Grundlegende Eigenschaften von Vielfachzugriffsverfahren verstehen und vergleichen. 				
3	Empfohlene Voraussetzung für die Teilnahme Elektrotechnik und Informationstechnik I und II, Deterministische Signale und Systeme, Mathematik I bis IV				
4	Prüfungsform Modulabschlussprüfung: <ul style="list-style-type: none"> • Modulprüfung (Fachprüfung, Klausur, Dauer: 90 min, Standard BWS) 				
5	Benotung Modulabschlussprüfung: <ul style="list-style-type: none"> • Modulprüfung (Fachprüfung, Klausur, Gewichtung: 100 %) 				
6	Verwendbarkeit des Moduls BSc ETiT, BSc Wi-ETiT, BSc CE, MSc iST, BSc MEC				
7	Notenverbesserung nach §25 (2)				
8	Literatur Gemäß Hinweisen in der Lehrveranstaltung				
Enthaltene Kurse					
	Kurs-Nr. 18-kl-1020-vl	Kursname Kommunikationstechnik I			
	Dozent Prof. Dr.-Ing. Anja Klein			Lehrform Vorlesung	SWS 3

Kurs-Nr. 18-kl-1020-ue	Kursname Kommunikationstechnik I		
Dozent Prof. Dr.-Ing. Anja Klein, Dr. rer. nat. Sabrina Klos, M.Sc. Kilian Kiekenap	Lehrform Übung	SWS 1	

Modulname TK3: Ubiquitous / Mobile Computing					
Modul-Nr. 20-00-0120	Kreditpunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1	Angebotsturnus Jedes 2. Sem.
Sprache Deutsch			Modulverantwortliche Person Prof. Dr. rer. nat. Eberhard Max Mühlhäuser		
1	Lerninhalt <ul style="list-style-type: none"> - Kenntnis technischer Grundlagen der Mobilkommunikation - Kenntnis wichtiger Herausforderungen, Thesen und Modelle des Ubiquitous Computing - Methodenwissen über aktuelle Ansätze des Ubiquitous Computing Stoffplan: <ul style="list-style-type: none"> - Einführung in Ubiquitous Computing - Definitionen und Bedeutung - Herausforderungen und Klassifikation - Wichtiges zur historischen Entwicklung (Mark Weiser u.a.) - Von Terminologie zu Taxonomie - Referenzarchitekture - Mobilkommunikation als 'Enabling Technology' - Einordnung und physikalische Grundlagen - Elementare Mehrfachzugriffs- und Modulationsverfahren - Zellulare Weitverkehrsnetze: von GSM bis LTE - Drahtlose lokale Netze: WLAN, Bluetooth und ZigBee - Internet-of-Things: RFID und Smart Items - Grundlagen von RFID-Systemen - EPC und Smart Items - NFC: Nahfeld-Kommunikation - Service Discovery und Cloudlets - Grundlagen der Skalierbarkeit im Ubiquitous Computing - Service Discovery: Grundlagen - Service Discovery: konkurrierende Ansätze - Cloudlets: Forschungsansätze für Ubiquitous Cloud Computing - Context- und Location Aware Computing - Grundlagen der Adaptivität in Ubiquitous Computing - Kontext-Modelle und Ansätze für Context-Aware Computing - Technische Grundlagen der Ortsbestimmung und Location Awareness - Mensch-Maschine-Interaktion für Ubiquitous Computing - Einführung: Ease-of-Use und Post-Desktop-Interaktion - Interaction Design und Multitmodale Interaktion - Grundlagen von Multitouch-Systemen - Pen-and-Paper-Interaktion und Tangible Interaction - UI Design: Evaluationstechniken - Systematisches UI Engineering - Privatsphäre und Vertrauen im Ubiquitous Computing - Einführung in Privacy und rechtliche Grundlagen - Zum Wesen personenbezogener Daten - Privacy-Enhancing Technologies (PETs) und Anonyme Kommunikation - Einführung in Vertrauen und Reputation - Vertrauensmodelle und Computational Trust - Trust-Management-Systeme 				
2	Qualifikationsziele / Lernergebnisse				

	Studierende kennen nach erfolgreichem Besuch der Veranstaltung die technische Grundlage mobiler Kommunikation. Sie verstehen die grundlegenden Herausforderungen von Ubiquitous Computing. Sie kennen aktuelle Ansätze um diese Herausforderungen zu lösen. Sie sind außerdem in der Lage ihre Kenntnisse auf aktuelle Probleme anzuwenden.		
3	Empfohlene Voraussetzung für die Teilnahme Computer Netzwerke und verteilte Systeme		
4	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> [20-00-0120-iv] (Fachprüfung, Mündliche/Schriftliche Prüfung, Standard BWS) 		
5	Benotung Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> [20-00-0120-iv] (Fachprüfung, Mündliche/Schriftliche Prüfung, Gewichtung: 100%) 		
6	Verwendbarkeit des Moduls B.Sc. Informatik M.Sc. Informatik M.Sc. Wirtschaftsinformatik B.Sc. Psychologie in IT Joint B.A. Informatik B.Sc. Sportwissenschaft und Informatik M.Sc. Sportwissenschaft und Informatik Kann im Rahmen fachübergreifender Angebote auch in anderen Studiengängen verwendet werden.		
7	Notenverbesserung nach §25 (2) In dieser Vorlesung findet eine Anrechnung von vorlesungsbegleitenden Leistungen statt, die lt. §25 (2) der 5. Novelle der APB und den vom FB 20 am 30.3.2017 beschlossenen Anrechnungsregeln zu einer Notenverbesserung um bis zu 1.0 führen kann.		
8	Literatur Literaturempfehlungen werden kontinuierlich aktualisiert, Beispiele für verwendete Literatur könnten sein: A Primärliteratur: Handbook of Research: Ubiquitous Computing Technology for Real Time Enterprises edited by Prof. Dr. Max Mühlhäuser, Dr. Iryna Gurevych, 2008, Information Science Reference, ISBN-10: 1599048329 B Sekundärliteratur: 1. F. Adelstein, S. Gupta et al.: Fundamentals of Mobile & Pervasive Computing McGraw Hill 2004, 2. Stefan Poslad: Ubiquitous Computing, Wiley 2009, ISBN 978-0-470-03560-3 3. Kapitel Mobilkommunikation: M. Sauter: Grundkurs Mobile Kommunikationssysteme: UMTS, HSDPA und LTE, GSM, GPRS und Wireless LAN; Vieweg-Teubner Studium 2010 4. J. Krumm (Ed.): Ubiquitous Computing Fundamentals, CRC Press 2010 D. Cook, S. Das (Ed.): Smart Environments, Wiley 2005		
Enthaltene Kurse			
	Kurs-Nr. 20-00-0120-iv	Kursname TK3: Ubiquitous / Mobile Computing	
	Dozent	Lehrform Integrierte Ver- anstaltung	SWS 4

1.2 Wahlkatalog SES: System on Chip und Eingebettete Systeme

Modulname Algorithmen für Hardware-Entwurfswerkzeuge					
Modul-Nr. 20-00-0183	Kreditpunkte 3 CP	Arbeitsaufwand 90 h	Selbststudium 60 h	Moduldauer 1	Angebotsturnus Jedes 2. Sem.
Sprache Deutsch und Englisch			Modulverantwortliche Person Prof. Dr.-Ing. Andreas Koch		
1	Lerninhalt <ul style="list-style-type: none"> - The VLSI design problem - Fundamental graph representations and algorithms - Representations for hierarchical circuits - Fabrication technologies for integrated circuits - Layout compaction - Timing analysis - Heuristical optimization techniques - Placement problems, algorithms, and cost functions - Exact optimization techniques - Partitioning and its use in placement - Floorplanning problems, representations, and techniques - Routing problems, algorithms, and cost functions 				
2	Qualifikationsziele / Lernergebnisse Studierende kennen nach erfolgreichem Besuch der Veranstaltung verschiedene Technologien für die Realisierung von integrierten Schaltungen. Sie können aus den verschiedenen Technologien die Anforderungen an Automatisierungswerkzeuge für verschiedene Teilaufgaben des Entwurfs- und Realisierungsprozesses herleiten. Sie sind vertraut mit der Modellierung technologischer Probleme durch formale Konzepte wie Graphen, Gleichungssysteme etc. Sie verstehen grundlegende Verfahren zur Lösung auch von harten Problemen und können aufbauend auf Erfahrungen mit verschiedenen Basisalgorithmen neue bzw. verfeinerte Implementierungen zur Erledigung der Entwurfsaufgaben entwickeln.				
3	Empfohlene Voraussetzung für die Teilnahme Empfohlen wird der erfolgreiche Besuch der Veranstaltungen "Digitaltechnik" sowie "Algorithmen und Datenstrukturen" und "Funktionale und objektorientierte Programmierung".				
4	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> • [20-00-0183-vl] (Fachprüfung, Mündliche/Schriftliche Prüfung, Standard BWS) 				
5	Benotung Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> • [20-00-0183-vl] (Fachprüfung, Mündliche/Schriftliche Prüfung, Gewichtung: 100 %) 				
6	Verwendbarkeit des Moduls B.Sc. Informatik M.Sc. Informatik B.Sc. Computational Engineering M.Sc. Computational Engineering M.Sc. Wirtschaftsinformatik B.Sc. Psychologie in IT Joint B.A. Informatik B.Sc. Sportwissenschaft und Informatik M.Sc. Sportwissenschaft und Informatik Kann im Rahmen fachübergreifender Angebote auch in anderen Studiengängen verwendet werden.				
7	Notenverbesserung nach §25 (2)				

8	Literatur Literaturempfehlungen werden kontinuierlich aktualisiert, Beispiele für verwendete Literatur könnten sein: Gerez: Algorithms for VLSI Design Automation Wang/Chang/Cheng: Electronic Design Automation		
Enthaltene Kurse			
	Kurs-Nr.	Kursname	
	20-00-0183-vl	Algorithmen für Hardware-Entwurfswerkzeuge	
	Dozent		Lehrform
			Vorlesung
			SWS
			2

Modulname Praktikum zu Algorithmen für Hardware-Entwurfswerkzeuge					
Modul-Nr. 20-00-0571	Kreditpunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1	Angebotsturnus Jedes 2. Sem.
Sprache Deutsch und Englisch			Modulverantwortliche Person Prof. Dr.-Ing. Andreas Koch		
1	Lerninhalt - Realisieren von Hardware-Entwurfswerkzeugen aus dem Bereich Layout-Synthese, speziell zu Themen wie Timing Analyse, Platzierung und Verdrahtung - Evaluieren der Ergebnisqualität und Rechenzeit- und Speicheranforderungen der eigenen Werkzeuge im Vergleich zu existierenden Implementierungen				
2	Qualifikationsziele / Lernergebnisse Nach erfolgreichem Absolvieren der Veranstaltung können die Studierenden eigenständig Hardware-Entwurfswerkzeuge für eine vorgegebene Zieltechnologie von integrierten Schaltungen erstellen. Sie können ihre Werkzeuge bezüglich verschiedener Gütemaße evaluieren und mit anderen existierenden Implementierungen vergleichen.				
3	Empfohlene Voraussetzung für die Teilnahme Der erfolgreiche Besuch bzw. die aktive parallele Teilnahme an der Veranstaltung "Algorithmen für Hardware-Entwurfswerkzeuge" ist dringend empfohlen.				
4	Prüfungsform Bausteinbegleitende Prüfung: • [20-00-0571-pr] (Studienleistung, Mündliche/Schriftliche Prüfung, Standard BWS)				
5	Benotung Bausteinbegleitende Prüfung: • [20-00-0571-pr] (Studienleistung, Mündliche/Schriftliche Prüfung, Gewichtung: 100%)				
6	Verwendbarkeit des Moduls B.Sc. Informatik M.Sc. Informatik B.Sc. Computational Engineering M.Sc. Computational Engineering M.Sc. Wirtschaftsinformatik B.Sc. Psychologie in IT Joint B.A. Informatik B.Sc. Sportwissenschaft und Informatik M.Sc. Sportwissenschaft und Informatik Kann im Rahmen fachübergreifender Angebote auch in anderen Studiengängen verwendet werden.				
7	Notenverbesserung nach §25 (2)				
8	Literatur Bereitgestellte wissenschaftliche Arbeiten zu den vorgeschlagenen Basisverfahren.				
Enthaltene Kurse					
	Kurs-Nr. 20-00-0571-pr	Kursname Praktikum zu Algorithmen für Hardware-Entwurfswerkzeuge			
	Dozent Prof. Dr.-Ing. Andreas Koch			Lehrform Praktikum	SWS 4

Modulname Advanced Digital Integrated Circuit Design					
Modul-Nr. 18-ho-2010	Kreditpunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1	Angebotsturnus WiSe
Sprache Englisch			Modulverantwortliche Person Prof. Dr.-Ing. Klaus Hofmann		
1	Lerninhalt Modelle von MOS-Transistoren, CMOS-Logikschaltungen, Chip-Layout und Entwurfsregeln, Statisches und Dynamisches Verhalten von CMOS-Schaltungen, Synchrone CMOS-Schaltungen, Performanz- und Leistungscharakterisierung, Entwurfstechniken und CAD-Werkzeuge, FPGA- und Gate Array Technologien, Speichertechnologien, Chip-Test				
2	Qualifikationsziele / Lernergebnisse Ein Student kann nach Besuch der Veranstaltung <ul style="list-style-type: none"> • die Kurzkanaleigenschaften von CMOS-Transistoren in einer modernen Halbleitertechnologie aufzeigen, • die Schaltungsprinzipien digitaler Gatter basierend auf CMOS-Transistoren aufzeigen und bezüglich ihrer Eigenschaften analysieren, • den durchgängigen Schaltungsentwurf digitaler ASICs basierend auf Standardzellen (Design, Layout, Simulation/Verifikation) aufzeigen, • die Vor- und Nachteile von synchroner und asynchroner Logik, Mehrphasentaktsystem usw. aufzeigen, • die unterschiedlichen Entwurfsstile integrierter elektronischer Systeme (ASIC, ASIP, Full-custom/Semicustom, PLA, PLD, FPGA) unterscheiden und kennt deren wichtigste Unterscheidungsmerkmale, • Basisschaltungen für logische und arithmetische Blöcke (Summierer, Multiplizierer, DLL, PLL) analysieren und kennt wichtige Eigenschaften, • Halbleiterspeicher (DRAM, SRAM, Flash, MRAM, FeRAM) nach ihrem Speicherprinzip unterscheiden und kennt deren Eigenschaften und Anwendungsgebiete. 				
3	Empfohlene Voraussetzung für die Teilnahme Vorlesung „Elektronik“				
4	Prüfungsform Modulabschlussprüfung: <ul style="list-style-type: none"> • Modulprüfung (Fachprüfung, Klausur, Dauer: 90 min, Standard BWS) 				
5	Benotung Modulabschlussprüfung: <ul style="list-style-type: none"> • Modulprüfung (Fachprüfung, Klausur, Gewichtung: 100 %) 				
6	Verwendbarkeit des Moduls MSc ETiT, MSc Wi-ETiT, MSc iCE, MSc iST, MSc MEC, MSc EPE				
7	Notenverbesserung nach §25 (2)				
8	Literatur Skriptum zur Vorlesung; John P Uyemura: Fundamentals of MOS Digital Integrated Circuits; Neil Weste et al.: Principles of CMOS VLSI Design				
Enthaltene Kurse					
	Kurs-Nr. 18-ho-2010-vl	Kursname Advanced Digital Integrated Circuit Design			
	Dozent Prof. Dr.-Ing. Klaus Hofmann			Lehrform Vorlesung	SWS 3

Kurs-Nr. 18-ho-2010-ue	Kursname Advanced Digital Integrated Circuit Design		
Dozent Prof. Dr.-Ing. Klaus Hofmann	Lehrform Übung	SWS 1	

Modulname Architekturen und Entwurf von Rechnersystemen					
Modul-Nr. 20-00-0012	Kreditpunkte 5 CP	Arbeitsaufwand 150 h	Selbststudium 105 h	Moduldauer 1	Angebotsturnus Jedes 2. Sem.
Sprache Deutsch			Modulverantwortliche Person Prof. Dr. phil. nat. Marc Fischlin		
1	Lerninhalt <ul style="list-style-type: none"> - Technologische Grundlagen und Trends der Mikroelektronik - Entwurfsflüsse für mikroelektronische Systeme - Beschreibung von Hardware-Systemen - Charakteristika von Rechnersystemen - Architekturen für parallele Ausführung - Speichersysteme - Heterogene Systems-on-Chip - On-Chip und Off-Chip Kommunikationsstrukturen - Aufbau eingebetteter Systeme, z.B. im Umfeld von Cyber-Physical Systems 				
2	Qualifikationsziele / Lernergebnisse Studierende kennen nach erfolgreichem Besuch der Veranstaltung funktionale und nichtfunktionale Anforderungen an heterogene diskrete und integrierte Rechnersysteme. Sie verstehen Techniken zum Aufbau solcher Systeme und können Entwurfsverfahren und -werkzeuge anwenden, um selbständig mit Hilfe der Techniken Rechner(teil)systeme zu konstruieren, die gegebene Anforderungen erfüllen. Sie können die Qualität der Systeme in verschiedenen Gütemaßen bewerten.				
3	Empfohlene Voraussetzung für die Teilnahme Empfohlen: Erfolgreicher Besuch der Vorlesungen „Digitaltechnik“ und „Rechnerorganisation“ bzw. entsprechende Kenntnisse aus anderen Studiengängen				
4	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> • [20-00-0012-iv] (Fachprüfung, Mündliche/Schriftliche Prüfung, Standard BWS) 				
5	Benotung Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> • [20-00-0012-iv] (Fachprüfung, Mündliche/Schriftliche Prüfung, Gewichtung: 100%) 				
6	Verwendbarkeit des Moduls B.Sc. Informatik B.Sc. Wirtschaftsinformatik B.Sc. Psychologie in IT Joint B.A. Informatik B.Sc. Sportwissenschaft und Informatik B.Sc. Informationssystemtechnik Kann im Rahmen fachübergreifender Angebote auch in anderen Studiengängen verwendet werden.				
7	Notenverbesserung nach §25 (2)				
8	Literatur Literaturempfehlungen werden kontinuierlich aktualisiert, Beispiele für verwendete Literatur könnten sein: Nikhil/Czeck: Bluespec by Example Arvind/Nikhil/Emer/Vijayaraghavan: Computer Architecture: A Constructive Approach Hennessy/Patterson: Computer Architecture – A Quantitative Approach Crockett/Elliott/Enderwitz/Stewart: The Zynq Book Flynn/Luk: Computer System Design Sass/Schmidt: Embedded Systems Design				

Enthaltene Kurse			
	Kurs-Nr. 20-00-0012-iv	Kursname Architekturen und Entwurf von Rechnersystemen	
	Dozent	Lehrform Integrierte Ver- anstaltung	SWS 3

Modulname Rechnersysteme II					
Modul-Nr. 18-hb-2030	Kreditpunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1	Angebotsturnus SoSe
Sprache Deutsch			Modulverantwortliche Person Prof. Dr.-Ing. Christian Hochberger		
1	Lerninhalt <ul style="list-style-type: none"> • Konfigurierbare Technologien • FPGA-Architekturen und Eigenschaften • System-On-Chip, HW-Komponenten, SW-Tool-Chain, Support-SW • Coarse Grained Reconfigurable Architectures, PE-Architektur, Modulo-Scheduling 				
2	Qualifikationsziele / Lernergebnisse Nach Abschluss des Moduls kennen die Studierenden rekonfigurierbare Technologien und Chip-Architekturen, die diese verwenden (FPGAs und CGRAs). Sie können die passende Technologie für konkrete Anwendungen auswählen. Sie wissen, welche Komponenten zu einem System-on-Chip gehören, und können ein anwendungsspezifisches SoC konfigurieren und programmieren. Studierende können rechenintensive Anwendungen auf ein CGRA abbilden und kennen die Einschränkungen und Hürden bei der Abbildung.				
3	Empfohlene Voraussetzung für die Teilnahme Solide Grundkenntnisse der Digitaltechnik und der Rechnerarchitektur (wie sie z.B. in den Vorlesungen "Logischer Entwurf" und "Rechnersysteme I" erworben werden. Grundkenntnisse in der Programmiersprache C sollten vorhanden sein.				
4	Prüfungsform Modulabschlussprüfung: <ul style="list-style-type: none"> • Modulprüfung (Fachprüfung, mündliche Prüfung, Dauer: 30 min, Standard BWS) 				
5	Benotung Modulabschlussprüfung: <ul style="list-style-type: none"> • Modulprüfung (Fachprüfung, mündliche Prüfung, Gewichtung: 100 %) 				
6	Verwendbarkeit des Moduls MSc ETiT, MSc iST, MSc iCE, MSc Wi-ETiT				
7	Notenverbesserung nach §25 (2)				
8	Literatur Die Folien zur Vorlesung können über Moodle heruntergeladen werden.				
Enthaltene Kurse					
	Kurs-Nr. 18-hb-2030-vl	Kursname Rechnersysteme II			
	Dozent Prof. Dr.-Ing. Christian Hochberger, M.Sc. Johanna Rohde			Lehrform Vorlesung	SWS 3
	Kurs-Nr. 18-hb-2030-ue	Kursname Rechnersysteme II			
	Dozent Prof. Dr.-Ing. Christian Hochberger, M.Sc. Johanna Rohde			Lehrform Übung	SWS 1

1.3 Wahlkatalog SWE: Software-Engineering

Modulname Software-Engineering - Wartung und Qualitätssicherung					
Modul-Nr. 18-su-2010	Kreditpunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1	Angebotsturnus SoSe
Sprache Deutsch			Modulverantwortliche Person Prof. Dr. rer. nat. Andreas Schürr		
1	Lerninhalt Die Lehrveranstaltung vertieft Teilthemen der Softwaretechnik, welche sich mit der Pflege und Weiterentwicklung und Qualitätssicherung von Software beschäftigen. Dabei werden diejenigen Hauptthemen des IEEE „Guide to the Software Engineering Body of Knowledge“ vertieft, die in einführenden Softwaretechnik-Lehrveranstaltungen nur kurz angesprochen werden. Das Schwergewicht wird dabei auf folgende Punkte gelegt: Softwarewartung und Reengineering, Konfigurationsmanagement, statische Programmanalysen und Metriken sowie vor allem dynamische Programmanalysen und Laufzeittests. In den Übungen wird als durchgängiges Beispiel ein geeignetes Open SourceProjekt ausgewählt. Die Übungsteilnehmer untersuchen die Software des gewählten Projektes in einzelnen Teams, denen verschiedene Teilsysteme des betrachteten Gesamtsystems zugeordnet werden.				
2	Qualifikationsziele / Lernergebnisse Die Lehrveranstaltung vermittelt an praktischen Beispielen und einem durchgängigen Fallbeispiel grundlegende Software-Wartungs- und Qualitätssicherungs-Techniken, also eine ingenieurmäßige Vorgehensweise zur zielgerichteten Wartung und Evolution von Softwaresystemen. Nach der Lehrveranstaltung sollte ein Studierender in der Lage sein, die im Rahmen der Softwarewartung und -pflege eines größeren Systems anfallenden Tätigkeiten durchzuführen. Besonderes Augenmerk wird dabei auf Techniken zur Verwaltung von Softwareversionen und -konfigurationen sowie auf das systematische Testen von Software gelegt. In der Lehrveranstaltung wird zudem großer Wert auf die Einübung praktischer Fertigkeiten in der Auswahl und im Einsatz von Softwareentwicklungs- Wartungs- und Testwerkzeugen verschiedenster Arten sowie auf die Arbeit im Team unter Einhaltung von vorher festgelegten Qualitätskriterien gelegt.				
3	Empfohlene Voraussetzung für die Teilnahme Grundlagen der Softwaretechnik sowie gute Kenntnisse objektorientierter Programmiersprachen (insbesondere Java).				
4	Prüfungsform Modulabschlussprüfung: <ul style="list-style-type: none"> • Modulprüfung (Fachprüfung, fakultativ, Standard BWS) 				
5	Benotung Modulabschlussprüfung: <ul style="list-style-type: none"> • Modulprüfung (Fachprüfung, fakultativ, Gewichtung: 100 %) 				
6	Verwendbarkeit des Moduls MSc ETiT, MSc iST, MSc Wi-ETiT, Informatik				
7	Notenverbesserung nach §25 (2)				
8	Literatur www.es.tu-darmstadt.de/lehre/se_ii/				
Enthaltene Kurse					
	Kurs-Nr. 18-su-2010-v1	Kursname Software-Engineering - Wartung und Qualitätssicherung			
	Dozent Prof. Dr. rer. nat. Andreas Schürr, M.Sc. Sebastian Marvin Ruland			Lehrform Vorlesung	SWS 3

Kurs-Nr. 18-su-2010-ue	Kursname Software-Engineering - Wartung und Qualitätssicherung		
Dozent Prof. Dr. rer. nat. Andreas Schürr, M.Sc. Sebastian Marvin Ruland	Lehrform Übung	SWS 1	

Modulname Einführung in den Compilerbau					
Modul-Nr. 20-00-0904	Kreditpunkte 5 CP	Arbeitsaufwand 150 h	Selbststudium 105 h	Moduldauer 1	Angebotsturnus Jedes 2. Sem.
Sprache Deutsch			Modulverantwortliche Person Prof. Dr. phil. nat. Marc Fischlin		
1	Lerninhalt - Aufbau von Compilern - Kontextfreie Grammatiken zur Beschreibungen der Syntax von Programmiersprachen - Lexing- und Parsingverfahren - Zwischendarstellungen - Semantische Analyse - Laufzeitorganisation - Code-Erzeugung - Software-Werkzeuge für den Compilerbau - Implementierungstechniken für Compiler				
2	Qualifikationsziele / Lernergebnisse Studierende kennen nach erfolgreichem Besuch der Veranstaltung den Aufbau von Compilern. Sie verstehen formale Konzepte zur Beschreibung von Syntax und Semantik von Programmiersprachen. Sie können diese Konzepte mit algorithmischen Verfahren kombinieren, um selbständig zu einer spezifizierten Programmiersprache einen passenden Compiler zu implementieren, der die Sprache auf die gewünschte Zielmaschine abbildet. Sie kennen Software-Werkzeuge zur Unterstützung des Compilerbaus und können diese zusammen mit manuellen Techniken bei der Implementierung von Compilern einsetzen.				
3	Empfohlene Voraussetzung für die Teilnahme Empfohlen: Erfolgreicher Besuch der Vorlesungen "Algorithmen und Datenstrukturen", "Funktionale und objektorientierte Programmierung" sowie "Rechnerorganisation", bzw. entsprechende Kenntnisse aus anderen Studiengängen				
4	Prüfungsform Bausteinbegleitende Prüfung: • [20-00-0904-iv] (Studienleistung, Mündliche/Schriftliche Prüfung, Standard BWS)				
5	Benotung Bausteinbegleitende Prüfung: • [20-00-0904-iv] (Studienleistung, Mündliche/Schriftliche Prüfung, Gewichtung: 100 %)				
6	Verwendbarkeit des Moduls B.Sc. Informatik B.Sc. Informationssystemtechnik Kann im Rahmen fachübergreifender Angebote auch in anderen Studiengängen verwendet werden.				
7	Notenverbesserung nach §25 (2)				
8	Literatur Literaturempfehlungen werden kontinuierlich aktualisiert, ein Beispiel für verwendete Literatur könnte sein: Watt/Brown: Programming Language Processors in Java				
Enthaltene Kurse					
	Kurs-Nr. 20-00-0904-iv	Kursname Einführung in den Compilerbau			
	Dozent Prof. Dr.-Ing. Andreas Koch			Lehrform Integrierte Veranstaltung	SWS 3

Modulname Echtzeitsysteme					
Modul-Nr. 18-su-2020	Kreditpunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1	Angebotsturnus SoSe
Sprache Deutsch			Modulverantwortliche Person Prof. Dr. rer. nat. Andreas Schürr		
1	Lerninhalt Die Vorlesung Echtzeitsysteme befasst sich mit einem Softwareentwicklungsprozess, der speziell auf die Spezifika von Echtzeitsystemen zugeschnitten ist. Dieser Softwareentwicklungsprozess wird im weiteren Verlauf während der Übungen in Ausschnitten durchlebt und vertieft. Der Schwerpunkt liegt dabei auf dem Einsatz objektorientierter Techniken. In diesem Zusammenhang wird ein echtzeitspezifisches State-of-the-Art CASE-Tool vorgestellt und eingesetzt. Des weiteren werden grundlegende Charakteristika von Echtzeitsystemen und Systemarchitekturen eingeführt. Auf Basis der Einführung von Schedulingalgorithmen werden Einblicke in Echtzeitbetriebssysteme gewährt. Die Veranstaltung wird durch eine Gegenüberstellung der Programmiersprache Java und deren Erweiterung für Echtzeitsysteme (RT-Java) abgerundet.				
2	Qualifikationsziele / Lernergebnisse Studenten, die erfolgreich an dieser Veranstaltung teilgenommen haben, sollen in der Lage sein, modellbasierte (objektorientierte) Techniken zur Entwicklung eingebetteter Echtzeitsysteme zu verwenden und zu bewerten. Dazu gehören folgende Fähigkeiten: <ul style="list-style-type: none"> • Systemarchitekturen zu bewerten und Echtzeitsysteme zu klassifizieren • selbständig ausführbare Modelle zu erstellen und zu analysieren • Prozesseinplanungen anhand üblicher Schedulingalgorithmen durchzuführen • Echtzeitprogrammiersprachen und -Betriebssysteme zu unterscheiden, zu bewerten und einzusetzen. 				
3	Empfohlene Voraussetzung für die Teilnahme Grundkenntnisse des Software-Engineerings sowie Kenntnisse einer objektorientierten Programmiersprache				
4	Prüfungsform Modulabschlussprüfung: <ul style="list-style-type: none"> • Modulprüfung (Fachprüfung, fakultativ, Standard BWS) 				
5	Benotung Modulabschlussprüfung: <ul style="list-style-type: none"> • Modulprüfung (Fachprüfung, fakultativ, Gewichtung: 100 %) 				
6	Verwendbarkeit des Moduls MSc ETiT, BSc iST, MSc Wi-ETiT, BSc Informatik				
7	Notenverbesserung nach §25 (2)				
8	Literatur www.es.tu-darmstadt.de/lehre/es/				
Enthaltene Kurse					
	Kurs-Nr. 18-su-2020-vl	Kursname Echtzeitsysteme			
	Dozent Prof. Dr. rer. nat. Andreas Schürr			Lehrform Vorlesung	SWS 3
	Kurs-Nr. 18-su-2020-ue	Kursname Echtzeitsysteme			
	Dozent Prof. Dr. rer. nat. Andreas Schürr			Lehrform Übung	SWS 1

Modulname Compiler Tooling					
Modul-Nr. 20-00-1013	Kreditpunkte 6 CP	Arbeitsaufwand 180 h	Selbststudium 120 h	Moduldauer 1	Angebotsturnus Jedes 2. Sem.
Sprache Deutsch und Englisch			Modulverantwortliche Person Prof. Dr.-Ing. Andreas Koch		
1	Lerninhalt Moderne Compiler zielen primär darauf ab, effizienten Code für eine bestimmte Plattform zu generieren und hierfür nutzen sie fortgeschrittene Analysis- und Transformationswerkzeuge. Eine solche Infrastruktur ist aber auch nützlich für Quellcodetransformation, z.B. für Werkzeuge, die Codes annotieren, instrumentieren, oder in eine kanonische Form bringen. Die Entwicklung solcher Werkzeuge ist für die C++ Sprache aufgrund ihrer Komplexität eine Herausforderung. Eine offene Compiler Infrastruktur, die in einer Vielzahl von Forschungs- und Produktionscompilern genutzt wird, ist die LLVM Infrastruktur (www.llvm.org). Ein vielgenutztes Front-End für C, C++ und objective C ist Clang, welches mächtige Mechanismen für die Extraktion von Information aus dem abstrakten Syntaxbaum zur Verfügung stellt, und so Modifikationen des Quellcodes wie auch die Generierung der Zwischenrepräsentation von LLVM ermöglicht. Die Studierenden arbeiten mit verschiedenen Komponenten und Techniken des Clang/LLVM Frameworks und implementieren praktische Übungen für Quelltransformationen. Die Clang/LLVM Techniken beinhalten insbesondere die Handhabung und Matching Techniken auf dem abstrakten Syntaxbaum von Clang. Beispiele von Quelltransformationen werden verschiedene Facetten von Code-Erweiterung und -Refactoring beinhalten, z.B. für die Instrumentierung paralleler Codes, für die Übermittlung von Information zwischen der statischen Analyse und der Laufzeitumgebung von (parallelen) Codes, oder für Code Refactoring um bestimmte Coding Standards einzuhalten.				
2	Qualifikationsziele / Lernergebnisse Nach dem Besuch dieses Kurses kennen die Studierenden grundlegende und fortgeschrittene Konzepte der syntaktischen und semantischen Code Analyse und Quelltext-Transformation, basierend auf der Clang/LLVM Technologie. Insbesondere können sie auf spezielle Aufgaben zugeschnittene statische Analyse- und Code- Transformations Werkzeuge entwerfen und implementieren, das geeignete Abstraktionsniveau für die zu lösende Aufgabe reflektieren und entscheiden, und weitere Nutzungsszenarien für Compiler Technologie erstellen.				
3	Empfohlene Voraussetzung für die Teilnahme Vorlesung Einführung in Compilerbau (EiCB), Vorlesung System- and Parallel Programming (SPP), Kenntnisse von C++				
4	Prüfungsform Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> [20-00-1013-pr] (Studienleistung, Mündliche/Schriftliche Prüfung, Standard BWS) 				
5	Benotung Bausteinbegleitende Prüfung: <ul style="list-style-type: none"> [20-00-1013-pr] (Studienleistung, Mündliche/Schriftliche Prüfung, Gewichtung: 100 %) 				
6	Verwendbarkeit des Moduls B.Sc. Informatik M.Sc. Informatik Kann in anderen Studiengängen verwendet werden.				
7	Notenverbesserung nach §25 (2)				
8	Literatur				
Enthaltene Kurse					

Kurs-Nr. 20-00-1013-pr	Kursname Compiler Tooling		
Dozent Prof. Dr.-Ing. Andreas Koch		Lehrform Praktikum	SWS 4

2 Vertiefungen / Anwendungen

Für die Wahlbereiche Vertiefungen und Anwendungen gibt es ein eigenes Modulhandbuch (Modulhandbuch: Wahlbereiche B.Sc. / M.Sc. Informationssystemtechnik (PO 2015)).

3 Studium Generale

Die Module für das Studium Generale finden Sie in einem gesonderten Modulhandbuch für das Studium Generale (Modulhandbuch Studium Generale).