

# ITG



**Gesellschaft für Informatik e. V. (GI)**  
**Informationstechnische Gesellschaft im VDE (ITG)**

## **Curriculum für Bachelor- und Masterstudiengänge Technische Informatik**

Fachbereich  
Technische Informatik

**GI/ITG-Empfehlungen**  
Stand: März 2018

Curriculum für Bachelor- und Master-Studiengänge Technische Informatik  
Empfehlungen der Gesellschaft für Informatik e. V.  
und der Informationstechnischen Gesellschaft im VDE

## **Arbeitskreis Curriculum Technische Informatik**

Mladen Berekovic, Universität zu Lübeck  
Lars Hedrich, Goethe-Universität Frankfurt am Main  
Christian Hochberger, TU Darmstadt  
Wolfgang Karl, Karlsruher Institut für Technologie (KIT)  
Reinhold Kröger, Hochschule RheinMain Wiesbaden  
Volker Lohweg, Hochschule Ostwestfalen-Lippe, Lemgo  
Erik Maehle, Universität zu Lübeck (Stellv. Sprecher FB Technische Informatik)  
Peter Marwedel, TU Dortmund  
Ulf Schlichtmann, TU München  
Uwe Schmidtman, FH Oldenburg-Ostfriesland-Wilhelmshaven  
Klaus Schneider, TU Kaiserslautern  
Djamshid Tavangarian, Universität Rostock  
Klaus Waldschmidt, Goethe-Universität Frankfurt am Main  
Norbert Wehn, TU Kaiserslautern

Das Präsidium der GI hat in seiner Sitzung vom 26. Januar 2018 diese Empfehlungen zustimmend zur Kenntnis genommen.

Der Vorstand von ITG und VDE hat diese Empfehlungen am 3. Juli 2017 verabschiedet.

Gesellschaft für Informatik e. V. (GI)  
Wissenschaftszentrum  
Ahrstraße 45  
D-53175 Bonn  
[www.gi-ev.de](http://www.gi-ev.de)

Informationstechnische  
Gesellschaft im VDE (ITG)  
Stresemannallee 15  
D-60596 Frankfurt  
[www.vde.com](http://www.vde.com)

Kontakt: Fachbereich Technische Informatik, AK Curriculum TI  
[maehle@iti.uni-luebeck.de](mailto:maehle@iti.uni-luebeck.de)

## Inhaltsverzeichnis

### Vorwort

<b>1. Einleitung.....</b>	<b>1</b>
1.1 Motivation und Herausforderung.....	1
1.2 Berufsbild.....	1
1.3 Bildungsziele.....	2
1.4 Zielgruppe .....	4
<b>2. Aufbau der Studiengänge.....</b>	<b>5</b>
2.1 Struktur.....	5
2.2 Definition Themengebiete.....	6
2.3 Lehrformen.....	6
<b>3. Bachelorstudiengang .....</b>	<b>7</b>
3.1 Kompetenzen und Themengebiete Technische Informatik .....	7
3.2 Kompetenzen und Themengebiete Informatik .....	9
3.3 Kompetenzen und Themengebiete Elektrotechnik und Informationstechnik.....	10
3.4 Kompetenzen und Themengebiete Mathematik und Physik.....	11
3.5 Überfachliche Schlüsselkompetenzen und Abschlussarbeiten .....	12
<b>4. Masterstudiengang .....</b>	<b>14</b>
4.1 Kompetenzen und Themengebiete methodischer Bereich.....	14
4.2 Kompetenzen und Themengebiete anwendungsbezogener Bereich.....	15
<b>Literatur .....</b>	<b>17</b>
 <b>ANHANG: Beschreibung der Kompetenzen und Themengebiete.....</b>	 <b>19</b>
I. BACHELOR.....	20
A. Pflichtbereich Technische Informatik .....	20
B. Wahlpflichtbereich Technische Informatik .....	31
C. Informatik.....	42
D. Pflichtbereich Elektrotechnik.....	43
E. Wahlpflichtbereich Elektrotechnik .....	47
II. MASTER .....	50
A. Methodischer Bereich .....	50
B. Anwendungsorientierter Bereich .....	59

## Vorwort

Mit der fortschreitenden Digitalisierung werden mehr und mehr Geräte und Gegenstände mit eigener Rechenleistung ausgestattet und zunehmend lokal oder global vernetzt. Sie werden damit immer autonomer und intelligenter und eröffnen innovative neue Anwendungen und Dienstleistungen wie z. B. das „Internet der Dinge“. Dies betrifft insbesondere auch die industrielle Produktion, wo unter der Bezeichnung „Industrie 4.0“ eine neue digitalisierte und vernetzte Generation von Automatisierungssystemen entsteht.

All diese Systeme verfügen neben der erforderlichen Software auch über geeignete, oft spezialisierte Hardware wie Sensoren und Aktoren, um mit ihrer Umwelt interagieren zu können. Für ihre Entwicklung sind daher interdisziplinäre Kenntnisse sowohl aus der Informatik als auch der Elektrotechnik und Informationstechnik erforderlich. Die Technische Informatik bewegt sich als Fachdisziplin genau in diesem Bereich, indem sie sich sowohl mit Hardware als auch systemnaher Software befasst. In den Fachgesellschaften GI und VDE-ITG wird sie durch den gemeinsamen Fachbereich Technische Informatik (FB TI) mit seinen Untergliederungen repräsentiert.

In der Ausbildung ist Technische Informatik ein fester Bestandteil sowohl im Studium der Informatik als auch der Elektrotechnik und Informationstechnik. An etlichen Hochschulen gibt es auch bereits eigenständige Bachelor- und Master-Programme in Technischer Informatik. Aufgrund der oben beschriebenen Entwicklung ist zu erwarten, dass das Studium der Technischen Informatik immer mehr an Bedeutung gewinnen wird. Der Arbeitskreis „Curriculum Technische Informatik“ des GI/ITG FB TI hat sich vor diesem Hintergrund zur Aufgabe gestellt, Empfehlungen für ein modernes Curriculum in Technischer Informatik an Universitäten und Hochschulen für angewandte Wissenschaften zu erarbeiten, wobei auf früheren Empfehlungen des Arbeitskreises zur Technischen Informatik in Bachelor- und Masterstudiengängen in Informatik aufgebaut werden konnte. Der Arbeitskreis setzte sich dabei aus Vertretern der Informatik sowie Elektrotechnik und Informationstechnik an Universitäten und Hochschulen für angewandte Wissenschaften zusammen.

Ausgehend vom Berufsbild der Technischen Informatik werden die Bildungsziele genannt und daraus die zu vermittelnden Kompetenzen und die zugehörigen Themengebiete abgeleitet. Die Struktur des Studiengangs richtet sich nach den GI-Empfehlungen für einen interdisziplinären Typ-2-Studiengang. Im Anhang findet sich eine detaillierte Beschreibung der Kompetenzen und Themengebiete, die sich an der in vielen Modulhandbüchern heute üblichen Darstellung orientiert.

Es sei besonders betont, dass es sich um curriculare Empfehlungen handelt, die keinerlei bindenden Charakter haben, sondern nur als Orientierungshilfe dienen sollen. Die einzelnen Hochschulen werden hier je nach Profilbildung eigene Schwerpunkte setzen. Die genaue Stoffauswahl liegt ohnehin in der Verantwortung der jeweiligen Lehrenden.

Die Fakultäten- und Fachbereichstage Informatik sowie Elektrotechnik und Informationstechnik wurden um Stellungnahmen gebeten, die in dem Text berücksichtigt sind. Weiterhin wurden die Stellungnahme des GI-Fachbereichs Informatik und Ausbildung/Didaktik der Informatik (IAD) sowie wertvolle Hinweise aus dem Präsidium der GI und dem Vorstand der ITG mit einbezogen. Allen Beteiligten gilt unser Dank, dass sie zur Verbesserung der Empfehlungen beigetragen haben.

Wir hoffen, dass die vorliegenden Empfehlungen zur Ausbildung in Technischer Informatik in eigenständigen Studiengängen hilfreich sein werden und zur Qualität der Lehre einen Beitrag leisten mögen.

Prof. Dr. Hannes Federrath  
Präsident der Gesellschaft  
für Informatik e.V. (GI)

Dr.-Ing. Volker Schanz  
Geschäftsführer der Informationstechnischen  
Gesellschaft (VDE-ITG)

# 1. Einleitung

## 1.1 Motivation und Herausforderung

Der internationale Maschinen- und Anlagenbau, Fahrzeugtechnik, Gebäudetechnik, Medizintechnik und weitere technologiegetriebene Branchen profitieren zunehmend von neuen Konzepten und Entwicklungen, welche stark von der Informatik und den Informationstechnologien getragen werden. Die allumfassende zuverlässige Vernetzung von Sensoren, Aktoren und massiv softwaregestützten informationsverarbeitenden Systemen [1] wird die genannten Branchen zukünftig stark dominieren. Der Forschungs- und Lehrbedarf in diesem Feld motiviert sich aus der Tatsache, dass Embedded Systems [2] Industrieanwendungen mit steigender Tendenz beherrschen. Auf Basis intelligenter und digital vernetzter Systeme werden industrielle Produktion und Informations- und Kommunikationstechnik zunehmend verzahnt (,Industrie 4.0‘ [3]).

Weiterhin spielt die Modellierung und Diagnose von Maschinen und Anlagen eine bedeutende Rolle. Hinzu kommen beispielsweise die Fernwartung von Maschinen und die Sensor- und Informationsfusion für die zustandsorientierte und vorausschauende Maschinenwartung sowie Industrielle Bildverarbeitung und Mustererkennung als eine Schlüsseltechnologie für die Qualitätssicherung in produzierenden Unternehmen [4]. Interdisziplinäre Ansätze aus Technik, Biologie und Psychologie können dabei neue zukunftsweisende Lösungen ermöglichen [1].

Der genannte Sachverhalt zeigt auf, dass die Komplexität heutiger Anlagen nur mithilfe massiv vernetzter Eingebetteter Systeme beherrscht werden kann. Diese müssen einerseits eine gewisse Autonomie aufweisen, andererseits muss die Komplexität im Sinne von Datenkonsistenz und -plausibilität selbst beherrscht werden. Hier spielt die Fähigkeit zur Selbstorganisation, die sich in den sog. Selbst-X-Eigenschaften wie selbst-konfigurierend, selbst-optimierend, selbst-heilend oder selbst-schützend widerspiegelt, eine wichtige Rolle. Derartige industrielle Informations- und Kommunikationssysteme müssen in der Lage sein, die Anforderungen der heutigen Anwenderbranchen (s. o.) zu berücksichtigen. Hierzu zählen unter anderem: erforderliche Prozessechtzeitfähigkeit und hohe Verfügbarkeitsanforderungen von Produktionsprozessen; beschränkte Ressourcen (Speicher, Prozessorleistung) der eingesetzten Gerätetechnik, notwendige Systemintegration der IuK-Technologien in bestehende Strukturen; hohe Anforderungen an IT-Sicherheit und funktionale Sicherheit, Konflikt zwischen langfristiger Verfügbarkeit von Investitionsgütern und der Änderungsgeschwindigkeit von IuK-Technologien, Verwendung von vorhandener Sensorik, Adaptionfähigkeit von Systemen im Lebenszyklus [5].

## 1.2 Berufsbild

Die Technische Informatik beschäftigt sich mit der Architektur, dem Entwurf, der Realisierung, der Bewertung und dem Betrieb von Rechner-, Kommunikations- und Eingebetteten Systemen sowohl auf der Ebene der Hardware als auch der systemnahen Software. Sie ist ein interdisziplinäres Gebiet, in dem ingenieurwissenschaftliche Fragestellungen, insbesondere aus der Elektrotechnik und Informationstechnik, mit der Informatik zusammenwirken. Diese Kenntnisse werden nahezu überall in Forschung und Entwicklung in der Industrie und an Hochschulen benötigt. Durch die weitgehende Automation von Abläufen und Prozessen mit dem Trend zu immer autonomer und intelligenter werdenden Systemen bearbeiten die Technischen InformatikerInnen Bereiche wie Intelligente Technische Systeme (Embedded Systems, Cyber-Physical Systems), Smart

Grids, Robotik, Bildverarbeitung und Mustererkennung etc. Insbesondere umfasst ihr Aufgabenfeld typische informatikorientierte Facetten wie u. a. mathematisch-methodische und systemanalytische Vorgehensweisen sowie algorithmische und datenstrukturierende Aspekte. Darüber hinaus entwickeln die Technischen InformatikerInnen ein besonderes Verständnis in Bezug auf die Interaktion von Mensch und technischer Anlage, da diese Voraussetzung zur Realisierung komplexer Systeme ist.

Aufgrund der zukünftigen Herausforderungen in Wissenschaft und Wirtschaft stehen den Technischen InformatikerInnen blendende Berufsaussichten bevor [6].

### **1.3 Bildungsziele**

Um dem oben dargestellten Berufsbild gerecht zu werden, müssen die Technischen InformatikerInnen die zugehörigen Kompetenzen in ihrem Studium erwerben. Zur systematischen Beschreibung und Taxonomie von Kompetenzen sind in der Lehr-Lern-Forschung entsprechende Kompetenzmodelle entwickelt worden.

Eine viel verwendete allgemeine Lernzieltaxonomie wurde von Anderson/Krathwohl [7] eingeführt. Sie basiert auf der sog. Anderson-Krathwohl-Taxonomie-Matrix (AKT-Matrix), die als Spalten kognitive Prozessstufen und als Zeilen Wissensdimensionen enthält.

Die kognitiven Prozessstufen sind in Niveaustufen eingeteilt: Erinnern (P1), Verstehen (P2), Anwenden (P3), Analysieren (P4), Beurteilen (P5), (Er-)Schaffen (P6). Die Wissensdimension beschreibt die Art des Wissens und unterscheidet dabei zwischen Faktenwissen (W1), konzeptuellem Wissen (W2), prozeduralem Wissen (W3) und Metakognition (W4). In die Felder der Matrix können dann die entsprechenden Kompetenzen je nach kognitiver Prozessstufe und Art des Wissens eingetragen werden.

Um den Anforderungen des Informatikstudiums besser gerecht zu werden, hat der GI-Fachbereich ‚Informatik und Ausbildung / Didaktik der Informatik (IAD)‘ ein reduziertes AKT-Modell entwickelt, das insbesondere die Kontextualisierung stärker berücksichtigt. Die kognitive Prozessdimension sieht die vier Stufen Verstehen, Anwenden und Übertragen, Analysieren und Bewerten sowie Erzeugen (Spalten der Matrix) vor. Bei der Kontextualisierung wird zwischen geringer Kontextualisierung und Komplexität sowie starker Kontextualisierung und hoher Komplexität unterschieden (Zeilen der Matrix). Diese reduzierte AKT-Matrix wird in [8] dafür eingesetzt, die Inhaltsbereiche der Informatik für das Bachelorstudium kompetenzorientiert zu beschreiben. Weiterhin werden die Typen wissenschaftlichen Arbeitens durch die Art und Komplexität des im Studiengang verwendeten Anwendungskontextes und dem Niveau der vermittelten kognitiven Prozesse und deren Wissensdimensionen beschrieben. Es werden sechs Typen unterschieden, die vom Kennen und Verstehen von Konzepten der Informatik ohne Kontextualisierung (T1) bis zur eigenständigen Erweiterung von Konzepten und Methoden der Informatik eines aktuellen Forschungsgebiets um anspruchsvolle Beiträge und deren Verteidigung in der Fachöffentlichkeit in Wort und Schrift (Promotion) (T6) reicht. Damit lässt sich insbesondere das Profil eines Studiengangs charakterisieren.

Die folgende Beschreibung der Bildungsziele orientiert sich an den Kognitiven Prozessdimensionen und Wissensdimensionen aus den Kompetenzmodellen. Die einzelnen Themenbereiche werden im Anhang ebenfalls kompetenzorientiert beschrieben, allerdings

nicht in Form einer AKT-Matrix, sondern in der in vielen Modulhandbüchern üblichen Form mit der Auflistung von Kompetenzen und Inhalten.

Die Technischen InformatikerInnen müssen über ein breites interdisziplinäres Faktenwissen verfügen. Dies umfasst neben den Kerngebieten der Technischen Informatik selbst und denen der Praktischen und Theoretischen Informatik vor allem die Elektrotechnik und Informationstechnik. Außerdem werden Grundlagen der Mathematik, vor allem der diskreten Mathematik und der Ingenieurmathematik, benötigt. Hinzu kommen die für die spätere Berufsausübung sehr wesentlichen fächerübergreifenden Schlüsselkompetenzen.

Zur Strukturierung und Einordnung der Fakten ist konzeptuelles Wissen erforderlich. Dies beinhaltet die Klassifizierung und das Erkennen von Zusammenhängen und Beziehungen und schließt vor allem die Befähigung zur Abstraktion und Modellbildung ein. Eine wichtige Rolle spielt hier die Systemarchitektur auf den verschiedenen Ebenen eines Rechners.

Aufbauend auf dem Faktenwissen und konzeptuellen Wissen müssen die Technischen InformatikerInnen prozedurales Wissen besitzen. Sie müssen hierbei Kompetenz über ein weites Spektrum von Verfahren und Methoden erwerben, auf deren Basis dann die Fähigkeit beruht, komplexe Hardware- und Softwaresysteme zu verstehen und mittels geeigneter Verfahren zu entwerfen und zu implementieren. Hierzu werden heute komplexe und leistungsfähige Werkzeuge (Tools) eingesetzt. Die Technischen InformatikerInnen müssen daher die entsprechende Werkzeugkompetenz aufweisen, um diese Tools sinnvoll und effektiv nutzen zu können. Dabei ist weniger die detaillierte Kenntnis eines konkreten Tools als die Fähigkeit zur Einarbeitung und Nutzung einer Klasse von Tools relevant.

Typische Projekte in Wirtschaft oder Wissenschaft werden heute fast ausschließlich in oft interdisziplinären Teams bearbeitet. Für die Technischen InformatikerInnen ist daher die für eine erfolgreiche Teamarbeit erforderliche Sozialkompetenz unverzichtbar. Auch Grundkenntnisse betriebswirtschaftlicher Zusammenhänge und rechtlicher Vorgaben sind wünschenswert.

Die Technische Informatik unterliegt einem rasanten technologischen Fortschritt und Wandel sowohl bei den Fakten und Konzepten als auch bei den Methoden und Werkzeugen. Daher müssen die Technischen InformatikerInnen metakognitives Wissen besitzen und u. a. die Fähigkeit erwerben, sich selbständig fortzubilden und sich ständig in neue Sachverhalte einzuarbeiten.

Die Bachelor-AbsolventInnen sollen auf Basis eines breiten und soliden Grundlagenwissens in der Lage sein, unter Anwendung wissenschaftlicher Methoden selbständig und im Team Lösungen für komplexe Aufgabenstellungen der Technischen Informatik zu finden und diese auch auf neue, ähnliche Projekte übertragen zu können.

Die Master-AbsolventInnen sollen darüber hinaus über vertiefendes Fakten- und konzeptuelles Methodenwissen verfügen und befähigt sein, anspruchsvolle Problemstellungen mit wissenschaftlichen Methoden systematisch zu analysieren, zu bewerten und neue Lösungen zu finden. Zudem sollen sie eigenständig Projekte leiten, präsentieren und neue Projekte initiieren können.

## 1.4 Zielgruppe

Hauptzielgruppe des Curriculums sind Studierende in eigenständigen Bachelor- und Masterstudiengängen in Technischer Informatik an Universitäten und Hochschulen für angewandte Wissenschaften. Dies schließt sowohl konsekutive als auch nicht-konsekutive Studiengänge, z. B. zur Weiterbildung, mit ein. Für Bachelor- und Masterstudiengänge werden einerseits Mobilität zwischen Hochschulen im In- und Ausland sowie Beachtung von Akkreditierungsrichtlinien, andererseits aber auch Profilbildung, Flexibilität und die Anpassung an sich wandelnde fachliche Entwicklungen gefordert. Die Curricula müssen daher stetig weiterentwickelt und angepasst werden. Dabei spielen neben wissenschaftlicher Qualität und Aktualität Interdisziplinarität und Internationalität eine immer wichtigere Rolle.

In diesem Prozess möchte die vorliegende Empfehlung eine Orientierungshilfe geben, welche Themengebiete in einem modernen Studiengang Technische Informatik im Pflichtbereich enthalten sein sollten, und exemplarisch wichtige Wahlpflichtbereiche nennen. Erarbeitet wurden diese Empfehlungen von einem Arbeitskreis des Fachbereichs Technische Informatik (FB TI) der Gesellschaft für Informatik (GI) und der Informationstechnischen Gesellschaft (ITG). Dieser Arbeitskreis hat auf Grundlage der damaligen Version der Empfehlungen der GI für Bachelor- und Masterprogramme an Hochschulen [8] und des Fakultätentags Informatik zur Einrichtung von konsekutiven Bachelor- und Masterstudiengängen in Informatik an Universitäten [9] bereits im Jahr 2011 GI-Empfehlungen zum Curriculum Technische Informatik in Bachelor- und Masterstudiengängen Informatik erarbeitet [10], die in die hier vorliegenden Empfehlungen eingeflossen sind. Da es sich jetzt um eigenständige Bachelor- und Masterstudiengänge in Technischer Informatik handelt, ist insbesondere die Elektrotechnik und Informationstechnik als weitere Disziplin mit integriert worden. Hierzu wurde der Arbeitskreis um Experten aus diesem Fach erweitert.



## 2. Aufbau der Studiengänge

### 2.1 Struktur

Entsprechend der Klassifikation der GI [8] handelt es sich bei der Technischen Informatik um sog. Typ-2-Studiengänge, die neben einem überwiegenden Informatik-Anteil einen speziellen Anwendungsbereich, hier die Elektrotechnik und Informationstechnik, enthalten. Im Einzelnen werden folgende Anteile empfohlen:

Kategorie	Bachelor	Master
Informatik	40 – 50 %	40 – 50 %
Elektrotechnik und Informationstechnik	20 – 30 %	17 – 30 %
Mathematische und naturwissenschaftlich/technische Grundlagen	10 – 20 %	
Fachübergreifende Vertiefungen		9 – 22 %
Fachübergreifende Grundlagen		
Überfachliche Schlüsselkompetenzen	10 – 20 %	9 %

Hinzu kommen jeweils noch die Bachelor- bzw. Masterarbeit sowie ggf. ein Praktikum.

Eine ähnliche Aufteilung wird vom Fakultätentag für Elektrotechnik und Informationstechnik (FTEI) vorgeschlagen:

- ~20% Grundlagen
- 30 - 50% Informatik
- 30 - 50% Elektrotechnik und Informationstechnik
- Rest Vertiefung, um den verschiedenen Fakultäts- und Hochschulprofilen Rechnung zu tragen.

Nach dem Vorschlag des FTEI kann der Anteil an Elektrotechnik und Informationstechnik somit etwas höher und der Informatikanteil etwas niedriger gewählt werden als nach der GI-Empfehlung.

Die jeweilige Gewichtung der Anteile nach den obigen Empfehlungen von GI und FTEI wird sich in der Regel stark am Profil der Hochschule orientieren.

Die einzelnen Anteile können wiederum in Pflicht- und Wahlpflichtbereiche unterteilt sein. Bachelorstudiengänge besitzen in der Regel einen relativ hohen Anteil von Pflichtveranstaltungen, in denen die fachlichen Grundlagen vermittelt werden, und verfügen nur über einen relativ kleinen Wahlpflichtbereich, in dem die Studierenden sich weiter spezialisieren können. Masterstudiengänge bieten dagegen üblicherweise sehr viele Wahlmöglichkeiten, die sich an dem Profil der jeweiligen Hochschule orientieren und den Studierenden sehr viele Freiheiten lassen, sich nach ihren Neigungen und Vorlieben zu vertiefen.

Derzeit wird in Deutschland Technische Informatik größtenteils allein als Bachelor-Studiengang angeboten, vorwiegend an Hochschulen für angewandte Wissenschaften. Die Regelstudienzeit beträgt 6 bis 8 Semester, wobei bei mehr als 6 Semestern meist ein Praxissemester integriert ist. Der Berufsqualifizierung durch diese Studiengänge kommt somit eine besondere Bedeutung zu. Es gibt auch einige wenige konsekutive Studiengänge mit entweder 6 + 4 oder 7 + 3 Semestern, sowohl an Universitäten als auch Hochschulen für

angewandte Wissenschaften. Reine Masterstudiengänge sind ebenfalls nur sehr wenig vertreten, teilweise konzentriert auf ein Teilgebiet der Technischen Informatik.

## **2.1 Definition Themengebiete**

Aus den zu vermittelnden Kompetenzen werden Inhalte abgeleitet, die zu Themengebieten zusammengefasst werden. Die vorliegenden curricularen Empfehlungen beziehen sich bewusst nicht auf konkrete Lehrveranstaltungen oder Module, sondern auf inhaltlich zusammengehörige Themengebiete. Die Umsetzung dieser Inhalte in einzelne Lehrveranstaltungen und deren Gruppierung zu Modulen soll den einzelnen Hochschulen überlassen bleiben, wobei mehrere Themengebiete zu einem größeren Modul oder auch Teile verwandter Themengebiete zu einer gemeinsamen Veranstaltung zusammengefasst werden können. Auch die Unterscheidung zwischen dem Pflicht- und Wahlpflichtbereich ist fließend zu verstehen. Während grundlagenorientierte Themengebiete in der Regel im Pflichtbereich angesiedelt sein werden, können weiterführende Themengebiete je nach Hochschule und Profil des Studiengangs entweder im Pflicht- oder im Wahlpflichtbereich vorgesehen werden. Ganz entsprechend ist auch die Grenze zwischen Bachelor- und Masterstudiengängen nicht scharf zu ziehen. Allerdings sollte vermieden werden, die gleiche Veranstaltung in einem konsekutiven Studiengang sowohl im Master als auch im Bachelor anzubieten.

## **2.3 Lehrformen**

Vermittelt werden die Inhalte der Technischen Informatik, wie an Hochschulen üblich, in Form von Vorlesungen, Übungen, Seminaren, Praktika und Projekten, die zu Modulen zusammengefasst werden, wobei ein Modul eine oder mehrere Vorlesungen, Übungen, Seminare, Praktika und ggf. Projekte enthalten kann. Eine wichtige Rolle spielen in der Technischen Informatik Werkzeuge, die z. B. zum Hardware- und Softwareentwurf oder zur Bewertung von charakteristischen Eigenschaften wie Leistung und Zuverlässigkeit eingesetzt werden. Besonders in den Übungen, Praktika und Projekten sollten diese Werkzeuge Verwendung finden, damit die Studierenden Kompetenzen mit deren praktischem Einsatz erwerben können. Oft bietet es sich daher an, die Übungen als praktische Übungen im Labor, in Form von Praktika oder als Projekte durchzuführen. In den Seminaren sollen die Studierenden sich unter Anleitung anhand der Literatur in ein Thema einarbeiten und dieses vor Zuhörern präsentieren und Fragen in einer Diskussion beantworten. Neben den fachlichen Kompetenzen werden hier auch Sozialkompetenzen vermittelt. Werden die Seminare in englischer Sprache durchgeführt, kann auch die Fremdsprachenkompetenz gefördert werden.

Zu den Themengebieten existiert eine Vielzahl von Fach- und Lehrbüchern, die zur Vertiefung des Lehrstoffes herangezogen werden sollten. Methoden des E-Learning, multimediale Lehrformen und andere neue Lehr- und Lernformen können an geeigneten Stellen integriert werden. Inhalte können dabei sowohl in induktiver als auch in deduktiver Lehrform vermittelt werden.

### **3. Bachelorstudiengang**

Um die in Abschnitt 1.3 aufgeführten Bildungsziele zu erreichen, werden die im Folgenden geschilderten Themengebiete empfohlen. Dabei wird die in Abschnitt 2.1 eingeführte Struktur zugrunde gelegt. Es wird jeweils nach Pflicht- und Wahlpflichtbereich unterschieden. Der recht umfangreiche Informatik-Anteil ist noch einmal in Technische Informatik und allgemeine Informatik untergliedert.

In jedem der genannten Themengebiete werden prinzipiell die in unter den Bildungszielen aufgeführten Kompetenzen vermittelt, allerdings mit unterschiedlicher Gewichtung. Veranstaltungen im Pflichtbereich vermitteln vorwiegend Faktenwissen und grundlegendes konzeptuelles und prozedurales Wissen. Vielfach wird in Laborübungen und Praktika zudem elementare Werkzeugkompetenz erworben. Die Projektarbeit dient vor allem auch dem Erwerb sozialer Kompetenz. Im Wahlpflichtbereich kann besonderes Gewicht auf bestimmte Kompetenzarten, z. B. Methoden- und Werkzeugkompetenz, gelegt werden.

Die Kompetenzen und Inhalte der einzelnen Themengebiete sind ausführlich im Anhang dargestellt. Es wurde dabei besonderer Wert auf eine kompetenzorientierte Beschreibung der Lernergebnisse [11] gelegt. Im Folgenden wird zunächst ein grober Überblick über die zu erwerbenden Kompetenzen und die dazu gehörigen Themengebiete gegeben.

#### **3.1 Kompetenzen und Themengebiete Technische Informatik**

Die Technischen InformatikerInnen müssen fundierte technologische sowie Analyse-, Entwurfs- und Realisierungskompetenzen zur Hardware und systemnahen Software besitzen.

Eine zentrale Rolle spielt die Digitaltechnik, da sie die Basis der technischen Realisierung heutiger Rechner-, Kommunikations- und eingebetteter Systeme bildet. Die Technischen InformatikerInnen müssen in der Lage sein, digitale Schaltungen auf verschiedenen Abstraktionsebenen (z. B. Gatter- und Registertransferebene) formal zu beschreiben, zu entwickeln und zu analysieren. Darauf aufbauend müssen sie grundlegende Komponenten von Rechnern wie Prozessoren, Speicher, Ein-/Ausgabeeinheiten erläutern und daraus einfache Rechner komponieren und bewerten können. Ebenfalls müssen sie die maschinennahe Programmierung von Prozessoren beherrschen und einfache Assemblerprogramme erstellen können. Es sollten dabei nicht nur theoretische Kenntnisse erworben werden, sondern die praktische Umsetzung in reale Systeme unter Laborbedingungen muss ebenfalls beherrscht werden.

Neben der Hardware ist systemnahe Software ein wichtiges Gebiet in der Technischen Informatik. Hierzu zählen Betriebssysteme und Rechnernetze, die direkt auf der Hardware aufsetzen. Die Technischen InformatikerInnen müssen die Konzepte und Strukturen moderner Betriebssysteme erläutern können und in der Lage sein, kleinere Anwendungen unter der Nutzung von Betriebssystem-APIs zu programmieren. Aufgrund der großen Bedeutung der Vernetzung heutiger Rechensysteme, insbesondere dem Internet, müssen sie in der Lage sein, die verschiedenen Schichten des Kommunikationssystems und der zugehörigen wichtigsten Protokolle beschreiben und für die Programmierung einfacher Anwendungen nutzen können.

Neben den oben beschriebenen grundlegenden Kompetenzen müssen in den höheren Semestern fortgeschrittene Kenntnisse und Kompetenzen erworben werden, die auf den Grundlagen aufbauen. Hierzu zählt die Fähigkeit, komplexere Rechnerarchitekturen,

insbesondere Konzepte der Parallelarbeit, zu beschreiben, zu bewerten und zu vergleichen. Mit der wachsenden Bedeutung von Rechensystemen, die mit ihrer physikalischen Umwelt interagieren und als solche gar nicht mehr sichtbar sind, müssen die Technischen InformatikerInnen derartige Eingebettete bzw. Cyber-Physikalische Systeme charakterisieren, modellieren und entwerfen können. Hierzu müssen sie auch die Modellierung und Simulation analog-digitaler Cyber-physikalischer Systeme, die durch Verschmelzung von Eingebetteten Systemen mit ihrer physikalischen Umgebung entstehen, beherrschen. Sie müssen den Entwurf komplexer Hardware/Software-Systeme nicht nur theoretisch verstehen, sondern auch in der Lage sein, reale Systeme in Teamarbeit zu entwickeln und zu implementieren.

Die obigen Kernkompetenzen sind für alle Bachelorstudierenden der Technischen Informatik erforderlich. Darüber hinaus sollten aber auch noch weiterführende Kompetenzen in Bereichen erworben werden, welche die Studierenden aus einem breiteren Angebot auswählen können. Die Auswahl kann insbesondere durch die spätere Ausrichtung im Beruf oder die Schwerpunkte in einem anschließenden Masterstudium motiviert sein. Hierbei kann es sich um vertiefende methodische Kompetenzen z. B. zum Entwurf oder Test von Digitalschaltungen, zum Einsatz von Systembeschreibungssprachen, zur Virtualisierung, zur Evaluation von Systemen oder der Programmierung von Eingebetteten Systemen handeln, aber auch um technologische Kompetenzen z. B. zur Rechnertechnologie, zur Energieeffizienz oder zu vernetzten Systemen. Ein weiterer wichtiger Bereich sind Kompetenzen und Kenntnisse zu typischen Anwendungsbereichen der Technischen Informatik wie z. B. Medizintechnik, Robotik, Mechatronik oder der Automatisierung technischer Prozesse.

Mit dem Vordringen von Rechnersystemen in immer weitere, vor allem auch kritische Bereiche der modernen Industriegesellschaft gewinnen die IT-Sicherheit und die Zuverlässigkeit immer mehr an Bedeutung. Dies spiegelt sich auch in Artikel 32 der Datenschutzgrundverordnung (General Data Protection Regulation, GDPR) wider, wo u. a. die Resilienz von informationstechnischen Systemen und Diensten als neuer Begriff eingeführt wird. IT-Sicherheit und Zuverlässigkeit spielen daher in den einzelnen Themengebieten des Pflichtbereichs bereits eine wichtige Rolle. Vertiefende Kompetenzen zur Verlässlichkeit, Robustheit und Fehlertoleranz, insbesondere im Fall von Hardwarefehlern, können zudem im Wahlpflichtbereich erworben werden. Kompetenzen zur IT-Sicherheit gegen Angriffe von außen (Security) sind für alle InformatikerInnen erforderlich und werden daher im Pflichtbereich zur Informatik vermittelt (siehe 3.2).

Zur Vermittlung der oben beschriebenen Kompetenzen dienen die im Folgenden aufgeführten Themengebiete, untergliedert in einen Pflichtbereich für die Kernkompetenzen und einen Wahlpflichtbereich für die vertiefenden Kompetenzen. Während die Pflichtveranstaltungen an allen Hochschulen in etwa gleiche Inhalte und Kompetenzen vermitteln und damit auch einen Wechsel erleichtern, orientieren sich die Wahlpflichtveranstaltungen in der Regel an dem Profil der Hochschule. Die aufgeführte Liste ist daher als exemplarisch und offen zu verstehen und sollte um jeweils neue aktuelle Themengebiete leicht erweiterbar sein.

Im Anhang sind die Empfehlungen für den Pflichtbereich recht detailliert, ähnlich wie in einem Modulhandbuch, wiedergegeben, gegliedert nach Kompetenzen, Inhalten und Übungen/Praktika. Letztere geben exemplarisch an, wie der gelehrte Stoff vertieft und gefestigt werden kann. Insgesamt soll eine Orientierung gegeben werden, wie die einzelnen Themengebiete inhaltlich ausgestaltet werden können, um die geforderten Kompetenzen zu vermitteln.

### **Pflichtbereich**

- Digitaltechnik
- Rechnerorganisation
- Betriebssysteme
- Rechnernetze
- Rechnerarchitektur
- Eingebettete Systeme
- Modellierung und Simulation analog-digitaler Systeme
- Projektarbeit

### **Wahlpflichtbereich**

- Verlässlichkeit, Robustheit und Fehlertoleranz
- Synthese Digitaler Schaltungen
- Test von Hardwareschaltungen
- Programmierung Eingebetteter Systeme
- Rechnertechnologie
- Robotik
- Medizintechnik
- Bildverarbeitung/Mustererkennung
- Usability Engineering für technische Systeme
- Evaluation von Systemen
- Automatisierung Technischer Prozesse (Steuerungs-/Regelungstechnik)
- Systembeschreibungssprachen
- Energieeffizienz
- Virtualisierung
- Vernetzte Systeme
- ...

## **3.2 Kompetenzen und Themengebiete Informatik**

Neben der Technischen Informatik müssen die Studierenden auch über Grundkompetenzen in den anderen Kernbereichen der Informatik verfügen. Hierzu gehören formale, algorithmische Kompetenzen, die zum Entwurf und zur Analyse von Algorithmen und den zugehörigen Datenstrukturen benötigt werden, sowie Realisierungskompetenz zu deren Implementierung in gängigen höheren Programmiersprachen. Darauf aufbauend muss die Fähigkeit erworben werden, größere Softwaresysteme unter Auswahl geeigneter Methoden systematisch zu analysieren, zu entwickeln und zu bewerten. Hierzu sind auch Projektmanagement-Kompetenzen erforderlich, da derartige Systeme in Teams entworfen und implementiert werden müssen. Zu den grundlegenden formalen Kompetenzen gehören zudem Kenntnisse zu formalen Sprachen und Automaten, der Berechenbarkeit und der Komplexität von Algorithmen.

Aufgrund ihrer großen praktischen Bedeutung müssen Konzepte und Strukturen von Datenbanken erläutert und deren praktischer Einsatz beherrscht werden. Von zunehmender

Bedeutung ist die IT-Sicherheit [12], zu der die wichtigsten Methoden und Verfahren beschrieben und geeignete Maßnahmen eingesetzt werden können müssen.

Neben diesen obligatorischen Kernkompetenzen sollte weiterhin die Möglichkeit zur Vertiefung durch Auswahl aus einer Liste von Veranstaltungen bestehen. Diese kann z. B. fortgeschrittene Kenntnisse zu algorithmischen oder auch Analyse-, Entwurfs- oder Realisierungskompetenzen umfassen.

Da die zugehörigen Themengebiete üblicherweise aus dem Angebot der Informatik übernommen werden, sei auch auf die GI-Empfehlungen für Bachelorprogramme im Studienfach Informatik [8] hingewiesen, wo eine genaue kompetenzorientierte Beschreibung in Form der reduzierten AKT-Matrix enthalten ist.

#### **Pflichtbereich**

- Programmiersprachen und -methodik
- Algorithmen und Datenstrukturen
- Software-Engineering
- Grundlagen der Theoretischen Informatik
- Datenbanken und Informationssysteme
- IT-Sicherheit

#### **Wahlpflichtbereich**

- Programmierung paralleler und verteilter Systeme
- Mensch-Computer-Interaktion
- Künstliche Intelligenz
- Informatik und Gesellschaft
- ...

### **3.3 Kompetenzen und Themengebiete Elektrotechnik und Informationstechnik**

Technische InformatikerInnen müssen solide Kenntnisse und Kompetenzen aus der Elektrotechnik und Informationstechnik erwerben, da sie sich schwerpunktmäßig mit Hardware oder hardwarenaher Software beschäftigen und oft im Team mit ElektrotechnikerInnen und anderen IngenieurInnen zusammenarbeiten.

Die Basis bildet die Fähigkeit, elektrische und magnetische Felder zu erklären sowie einfache elektrische Gleich- oder Wechselstromnetzwerke und Schaltkreise zu berechnen und zu analysieren. Darauf aufbauend sollten elektronische Bauelemente, insbesondere Dioden und Transistoren erläutert und die zugehörigen Grundsaltungen analysiert, berechnet und erzeugt werden können. Den Schwerpunkt sollten dabei analoge Schaltungen bilden, da digitale Grundsaltungen schon in den Themengebieten aus der Technischen Informatik abgedeckt werden.

Viele technische Systeme verarbeiten digitale oder analoge Signale. Die Technischen InformatikerInnen müssen daher die theoretischen Grundlagen zu kontinuierlichen und diskreten Signalen und der zugehörigen Systemtheorie erläutern und die zugehörigen

formalen Methoden wie Fourier-, Laplace- oder Z-Transformationen anwenden können. Zudem sollten sie in der Lage sein, komplexe gemischt analog-digitale Systeme zu analysieren, zu entwerfen und zu bewerten.

Vertiefende Kompetenzen z. B. zur Kommunikations- oder Messtechnik können dann wieder aus einem größeren Angebot ausgewählt werden.

Die zugehörigen Themengebiete ergeben sich dann wie folgt. Eine genauere Beschreibung ist wieder im Anhang zu finden.

#### **Pflichtbereich**

- Grundlagen der Elektrotechnik
- Elektronische Schaltungstechnik, Bauelemente
- Signale und Systeme
- Analog-digitale Systeme

#### **Wahlpflichtbereich**

- Kommunikationstechnik
- Drahtlose Kommunikation
- Messtechnik
- Grundlagen der Mechatronik
- ...

### **3.4 Kompetenzen und Themengebiete Mathematik und Physik**

Wie in allen technischen Disziplinen sind mathematische Kompetenzen für die Technische Informatik unbedingt erforderlich. Während in der Informatik vor allem Kompetenzen in linearer Algebra und diskreter Mathematik, insbesondere der Graphentheorie und Kryptographie gefordert sind, benötigen die Technischen InformatikerInnen für die elektrotechnischen Themengebiete auch gute Grundkenntnisse aus der Analysis, insbesondere zu Integraltransformationen und zur Lösung von Differentialgleichungen. Da für viele Gebiete stochastische Methoden eine große Rolle spielen, sind zudem Kompetenzen in der Wahrscheinlichkeitstheorie erforderlich. Weitere Bereiche der Mathematik wie z. B. die Numerik oder Statistik können bei den vertiefenden Kompetenzen angesiedelt sein.

In vielen Bereichen der Technischen Informatik spielen auch die physikalischen Grundlagen eine wichtige Rolle. Beispielhaft seien hier die elektrischen und magnetischen Felder, Halbleiter-Bauelemente oder Cyber-physikalische Systeme genannt. Allerdings handelt es sich hierbei in der Regel um spezielle Teilbereiche der Physik, die am besten innerhalb der jeweiligen Themengebiete behandelt werden sollten, um die erforderlichen Kenntnisse auf diese zugeschnitten zu vermitteln. Eine allgemeine Einführung, die einen Überblick über die gesamte Physik gibt, ist zwar nützlich, wird aber nicht als unbedingt erforderlich angesehen. Sie sollte aber im Wahlpflichtbereich angeboten werden.

Nähere Angaben zu den mathematischen Kompetenzen und zugehörigen Inhaltsbereichen für InformatikerInnen finden sich in [8]. Für Technische InformatikerInnen bietet sich bzgl.

Lineare Algebra und Analysis vor allem die Ingenieurmathematik an, wie sie an vielen Hochschulen für die ingenieurwissenschaftlichen Studiengänge angeboten wird. Entsprechendes gilt für die Physik. Da hier keine besonderen Anforderungen seitens der Technischen Informatik bestehen, sei auf detaillierte Angaben zu den Themengebieten Ingenieurmathematik und Physik im Anhang verzichtet.

Einen Überblick über die einzelnen Themengebiete enthält die folgende Liste.

#### **Pflichtbereich**

- Ingenieurmathematik (Lineare Algebra, Analysis)
- Logik und Diskrete Strukturen
- Wahrscheinlichkeitstheorie

#### **Wahlpflichtbereich**

- Numerik
- Physik
- Statistik
- ...

### **3.5 Überfachliche Schlüsselkompetenzen und Abschlussarbeit**

Wie bei der Struktur in Abschnitt 2.1 vorgesehen, gehören auch überfachliche Schlüsselkompetenzen z. B. aus den Bereichen Betriebswirtschaft, Jura, Existenzgründung oder Projektmanagement mit zum Curriculum. Die Technischen InformatikerInnen müssen in der Lage sein, über ihre technischen Fachkenntnisse hinaus, die ökonomischen, ökologischen, rechtlichen und sozialen Aspekte in den einzelnen Anwendungsfeldern zu beurteilen und zu beachten. Weiterhin spielt vor allem in internationalen Teams Fremdsprachenkompetenz eine wichtige Rolle.

Sie müssen die Sozialkompetenz besitzen, im Team, insbesondere auch mit Fachleuten anderer Disziplinen und AnwenderInnen, zu kommunizieren und zu kooperieren. Hierzu gehören auch Selbstkompetenz und die Fähigkeit zum Konfliktmanagement. Ein genauerer Überblick über nicht-kognitive Kompetenzen für InformatikerInnen wie z. B. Selbststeuerkompetenz, Kooperationskompetenz oder Lernkompetenz, die primär im Zusammenhang mit kognitiven Kompetenzen implizit erworben werden und nicht expliziter Lerngegenstand sind, findet sich in [8].

Der Erwerb dieser Schlüsselkompetenzen kann sowohl in eigenen Veranstaltungen erfolgen als auch in anderen Veranstaltungen wie Praktika oder Seminaren integriert werden.

Eine Liste von möglichen Themengebieten ist im Folgenden aufgeführt. Auf eine genaue Beschreibung im Anhang wird verzichtet, da es sich um nicht fachspezifische Veranstaltungen handelt, die für einen breiten Hörerkreis aus verschiedenen Disziplinen konzipiert sind.



- Einführung in die Wirtschaftslehre
- StartUp and New Business
- Qualitätsmanagement
- Ethik der Forschung
- Wissenschaftstheorie
- Projekt- und Teamkompetenz
- ...

Schließlich ist jeweils zum Abschluss der Studiengänge eine Bachelorarbeit anzufertigen, deren Ergebnisse in einem Kolloquium präsentiert werden. Diese vertieft vor allem die Fachkompetenz, trägt aber auch zu den überfachlichen Schlüsselkompetenzen bei.

## 4. Masterstudiengang

Da Masterstudiengänge in der Regel weniger streng strukturiert sind als Bachelorstudiengänge und sich stark an den Forschungsschwerpunkten der jeweiligen Hochschulen orientieren, werden hier nur relativ allgemeine Empfehlungen gegeben. Oft haben die Studierenden eine Vielzahl von Wahlmöglichkeiten und können sich das Curriculum nach ihren Neigungen und Vorlieben zusammenstellen. Im Folgenden wird zwischen zwei Bereichen unterschieden, aus denen jeweils Themengebiete ausgewählt werden können: der *methodische* und der *anwendungsbezogene* Bereich. Dabei sind nicht nur Themengebiete aus dem Kernbereich der Technischen Informatik aufgeführt, sondern auch solche, in denen Technische Informatik eine wesentliche Rolle spielt. Damit wird dem Zusammenwachsen verschiedener interdisziplinärer Gebiete, vor allem im Anwendungsbereich, Rechnung getragen.

Die nachfolgende Zusammenstellung ist exemplarisch gemeint, d. h. je nach Profil der Hochschule können noch weitere Themengebiete in beiden Bereichen hinzugefügt bzw. weggelassen werden.

### 4.1 Kompetenzen und Themengebiete methodischer Bereich

Im Masterstudium werden aufbauend auf einem abgeschlossenen Bachelorstudium oder einer beruflichen Tätigkeit tiefere Kompetenzen der Technischen Informatik vermittelt. Es soll insbesondere die Fähigkeit erworben werden, anspruchsvolle Aufgaben mit Konzepten und Methoden der Technischen Informatik eigenständig zu lösen und neue Ideen zu entwickeln.

In den methodischen Bereich gehören Kompetenzen, die eher generischen Charakter haben und grundlegende Konzepte und Verfahren der Technischen Informatik auf hohem wissenschaftlichem Niveau behandeln. Hierzu gehören u. a. Entwurfsmethoden wie Hardware/Software Co-Design, nicht-funktionale Eigenschaften wie Verlässlichkeit und Robustheit, aber auch Systemklassen mit besonderen Anforderungen wie Echtzeitsysteme.

Die einzelnen, im Folgenden aufgeführten und im Anhang näher beschriebenen Themengebiete können dabei jeweils auch in mehrere Veranstaltungen, die einzelne Aspekte vertiefen, aufgeteilt sein.

- Parallelrechner
- Hardware/Software Co-Design
- EDA (Electronic Design Automation)
- Optimierung
- Rekonfigurierbare Architekturen
- Echtzeitsysteme
- Debugging und Validierung von Software in Eingebetteten Systemen
- Formale Verifikation
- Technische Zuverlässigkeit
- Organic Computing
- Physik-/Bio-/IT-Interaktion
- Nano-Elektronik
- ...

## **4.2 Kompetenzen und Themengebiete anwendungsorientierter Bereich**

Der anwendungsbezogene Bereich vermittelt Kompetenzen zum Einsatz von fortgeschrittenen Methoden der Technischen Informatik in konkreten Anwendungsfeldern und Applikationen. Beispiele sind Multimedia-Systeme, Roboter, Hochleistungsrechner und Systeme der Verkehrs-, Fertigungs- oder Energietechnik.

Mögliche Themengebiete, die wieder im Anhang ausführlicher beschrieben werden, sind:

- Systemarchitekturen für Multimedia
- Ambiente Systeme
- Software Ubiquitärer Systeme
- Drahtlose Sensornetze
- Mobile Roboter
- Medizinische Roboter
- (Mikro-)Sensoren und Aktoren
- Informationsfusion
- Hochleistungsrechner
- Nichtkonventionelles Rechnen (Quanten-Computing, DNA-Computing etc.)
- Energieinformatik
- Verkehrsinformatik
- Fertigungsautomatisierung
- Computational Intelligence
- Cloud Computing

...

## **Danksagung**

Bei der Beschreibung der Kompetenzen und Themengebiete wurde teilweise auch auf Angaben und die Mitarbeit von Fachkolleginnen und -kollegen aufgebaut, die nicht Mitglied des Arbeitskreises sind. Ihnen sei an dieser Stelle ganz herzlich für ihre Hilfe gedankt.

## Literatur

- [1] Manfred Broy (Hrsg.): CYBER-PHYSICAL SYSTEMS – Innovation durch software-intensive eingebettete Systeme, Acatech DISKUTIERT, Springer-Verlag, Berlin 2010.
- [2] Stephan Ziegler, Anne Müller (Ed.): Eingebettete Systeme – Ein strategisches Wachstumsfeld für Deutschland (Anwendungsbeispiele, Zahlen und Trends), BITKOM Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e. V., Berlin 2010.
- [3] Bundesministerium für Wirtschaft und Energie, Bundesministerium für Bildung und Forschung: Plattform Industrie 4.0, <http://www.plattform-i40.de/I40/Navigation/DE/Home/home.html>, abgerufen am 13.03.2018.
- [4] Holtmannspötter, D. et al.: VDE Technologieprognosen – Internationaler Vergleich 2010, Zukünftige Technologien, Consulting der VDI Technologiezentrum GmbH, Nr. 88, Düsseldorf, September 2010.
- [5] Artemis Industry Association, [https://artemis-ia.eu/about\\_artemis.html](https://artemis-ia.eu/about_artemis.html), abgerufen am 13.03.2018.
- [6] Computerwoche – Überall begehrt: Technische Informatiker, <http://www.computerwoche.de/karriere/hp-young-professional/2484821/>, online publiziert am 23.05.2011, abgerufen am 13.03.2018.
- [7] Anderson, L.W.; Krathwohl, D.: A Taxonomy for Learning, Teaching, and Assessing. A Revision of Bloom’s Taxonomy of Educational Objectives. Addison Wesley 2001.
- [8] Empfehlungen für Bachelor- und Masterprogramme im Studienfach Informatik an Hochschulen. Empfehlungen der Gesellschaft für Informatik 2016, <https://dl.gi.de/handle/20.500.12116/2351>, abgerufen am 13.03.2018.
- [9] Empfehlungen zur Einrichtung von konsekutiven Bachelor- und Masterstudiengängen in Informatik an Universitäten. Fakultätentag Informatik 2004, [https://www.ft-informatik.de/fileadmin/user\\_upload/uploads/tx\\_sbdownloader/bachelor\\_master\\_empfehlungen.pdf](https://www.ft-informatik.de/fileadmin/user_upload/uploads/tx_sbdownloader/bachelor_master_empfehlungen.pdf), abgerufen am 13.03.2018.
- [10] Curriculum Technische Informatik in Bachelor- und Masterstudiengängen Informatik. Empfehlungen der Gesellschaft für Informatik 2011, <https://dl.gi.de/handle/20.500.12116/2349>, abgerufen am 13.03.2018.
- [11] Lernergebnisse praktisch formulieren. nexus Impulse für die Praxis, Nr. 2, Hochschulrektorenkonferenz (Hrsg.), Bonn 2013, <http://www.hrk-nexus.de/fileadmin/redaktion/hrk-nexus/07-Downloads/07-02-Publikationen/nexus-Impuls-2-Lernergebnisse.pdf>, abgerufen am 13.03.2018.

- [12] IT-Sicherheit in der Ausbildung. Empfehlungen der Gesellschaft für Informatik zur Berücksichtigung der IT-Sicherheit in der schulischen und akademischen Ausbildung 2006,  
<https://dl.gi.de/handle/20.500.12116/2343>,  
abgerufen am 13.03.2018.

## **ANHANG: Beschreibung der Kompetenzen und Themengebiete**

Im folgenden Anhang wird eine ausführliche Beschreibung der im Hauptteil kurz aufgeführten Themengebiete gegeben. Die Darstellung orientiert sich dabei an der heute in vielen Modulhandbüchern üblichen Form mit einer Unterteilung in Lernziele/Kompetenzen und Inhalte. Bei Themengebieten im Pflichtbereich werden auch Vorschläge zu möglichen Inhalten der Übungen gemacht.

Die Beschreibungen sind exemplarisch gemeint und sollen lediglich zur Orientierung dienen. Die konkrete Ausgestaltung bleibt natürlich den jeweiligen Lehrenden überlassen. Es kann daher auch zu Überschneidungen der einzelnen Themengebiete kommen, auch vor dem Hintergrund, dass vor allem im Wahlpflichtbereich nicht alle Themengebiete angeboten bzw. gehört werden.

# I. BACHELOR

## A. Pflichtbereich Technische Informatik

### Digitaltechnik

Die Informatik basiert ganz wesentlich auf digitalen Schaltungen zur technischen Realisierung konkreter Systeme der Datenverarbeitung. Ein fundiertes Verständnis der Digitaltechnik und darauf aufbauender Schaltungen zur Datenverarbeitung ist daher unabdingbar für eine vollständige Grundausbildung in Technischer Informatik.

### *Lernziele/Kompetenzen*

Die Studierenden sind in der Lage,

- verschiedene Darstellungsformen von Zahlen und Alphabeten in Rechnern zu beschreiben,
- eine formale und programmiersprachliche Schaltungsbeschreibung anzufertigen,
- technische Realisierungsformen von Schaltungen zu erläutern,
- basierend auf dem Verständnis für Aufbau und Funktion aller wichtigen Grundsaltungen und Rechenwerke, unbekannte Schaltungen zu analysieren,
- einfache eigene Schaltungen zu entwickeln,
- die digitaltechnische Umsetzung der in anderen Veranstaltungen (Rechnerorganisation, Rechnerarchitektur und Eingebettete Systeme) behandelten Systeme anzugeben.

### *Inhalte*

- Theoretische Grundlagen
  - Informationsdarstellung, Zahlensysteme
  - Binärdarstellungen negativer Zahlen, Gleitkomma-Zahlen
  - Alphabete, Codes
- Elektronikgrundlagen der Technischen Informatik
  - einfache elektrische Zusammenhänge wie Ohmsches Gesetz, Kondensatorgleichung
  - MOS-Transistoren, CMOS-Schaltungen, Energieeffizienz
  - Tri-State und Open Drain Schaltungen
- Gatterebene
  - Boolesche Algebra/Schaltalgebra, Normalformen
  - Schaltfunktionsoptimierung (zweistufig und mehrstufig, z. B. KV-Diagramme, Quine-McCluskey, Notwendigkeit heuristischer Verfahren)
  - Schaltnetze, Grundsaltungen (Multiplexer, Halb/Voll-Addierer)
  - Schaltwerke, Automatentypen (FlipFlops, Mealy und Moore)
  - Automatenrealisierungen, Zustandscodierung
  - Realisierungsformen von digitalen Schaltungen: Gatter, PLDs, FPGAs, ASICs
- Entwurf auf Register-Transfer-Ebene
  - Hardware-Beschreibungssprachen
  - Schaltungssimulation, Delay-Modelle
  - Synthese
- Rechenwerke
  - Addierer-Varianten (Ripple-Carry, Carry-Look-Ahead, Carry-Save), Realisierung der Subtraktion
  - Multiplizier-Schaltungen (parallel, seriell, Block-Multiplizierer), Divisionsschaltungen
- Steuerwerke
  - Binär codiert
  - 1-aus-n codiert



- Speichertechnologien
  - Halbleiterspeicher
  - magnetische und optische Speicherung

## ***Übungen***

- Umwandlung verschiedener Zahlensysteme ineinander
- Rechnungen mit binär dargestellten Zahlen
- Analyse einfacher Transistorschaltungen: Logische Funktion, Schaltzeitpunkt, Stromverbrauch
- Minimierung von Schaltfunktionen, am besten rechnergestützt
- Verschiedene Realisierungen von Schaltnetzen: mit Gattern, mit PLDs, mit FPGAs
- Analyse einfacher Digitalschaltungen bzgl. Funktion und Zeitverhaltens
- Entwurf von Digitalschaltungen
- Simulation einfacher Schaltungen mit interaktiven graphischen Schaltungssimulatoren
- Darstellung und Simulation einfacher Schaltungen mit Hardware-Beschreibungssprachen (z. B. serieller oder Block-Multiplizierer)

## **Rechnerorganisation**

Kenntnisse grundlegender Prozessorkonzepte und der Assemblerprogrammierung sind die Voraussetzungen für eine sinnvolle und ressourcenbewusste Nutzung von Rechnerstrukturen.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- einen einfachen Rechner aus Grundkomponenten zu komponieren, dies insbesondere am Beispiel Eingebetteter Systeme. Dazu werden folgende Kompetenzen benötigt:
  - verschiedene Realisierungsformen komplexer Schaltungen erläutern können,
  - die relevanten Speichertechnologien beschreiben können,
  - Aufbau und Programmierung von Prozessoren analysieren können,
  - einfache Assemblerprogramme erstellen können,
- spezifische Eigenschaften des Rechners bei der Programmierung zu berücksichtigen.

### ***Inhalte***

- Steuerwerke
  - Fest verdrahtete Steuerwerke, mikroprogrammierte Steuerwerke
- Grundlegende Rechnerorganisation
  - Von-Neumann-Rechnermodell, Assemblerprogrammierung, Befehle, Adressierungsarten
  - Grundlegende Befehlssatzarchitekturen: Akkumulator oder Einadressmaschine, Register-Maschine (mit zwei oder drei Adressen), Stackmaschinen (Null-Adressmaschine)
  - Befehlsabarbeitung, Interrupts
  - Einfache Ein-/Ausgabe Schaltungen (z. B. binäre Ports, serielle Schnittstellen und Timer, Analog-Digital-Umsetzer)
  - Fallstudie (z. B. Mikrocontroller)
- Diskussion RISC/CISC-Architektur
- Pipelining des Maschinenbefehlszyklus
  - Pipeline-Hemmnisse
  - Methoden zur Auflösung von Pipeline-Konflikten

- Speicherhierarchien, Speicherorganisation
  - Cache-Speicher, Cache-Organisation, Aktualisierungsstrategien
- Unterstützung von Betriebssystemfunktionen
  - Speicherverwaltung
  - Schutzmechanismen
  - Virtualisierung
- Ein-/Ausgabe-System, Schnittstellen, Interrupt-Verarbeitung
- Bus-Systeme

## ***Übungen***

- Realisierung eines Steuerwerkes als Hardware-Steuerwerk mittels One-Hot oder binärer Codierung und als Mikroprogramm-Steuerwerk
- Erstellen einfacher, exemplarischer Assemblerprogramme (möglichst für alle verschiedenen Befehlssatzarchitekturen, auf dem Papier oder anhand entsprechender Simulatoren)
- Vergleich des Verhaltens und quantitative Bewertung verschiedener Cache-Organisationsformen
  - Berechnen des Zugriffsverhaltens von Cache-Speichern
  - Berechnen von Fehlzugriffsraten bzw. Trefferraten
- Pipeline-Organisation
  - Berechnen von Beschleunigungsfaktoren beim Einsatz von Pipelines
  - Erkennen und Auflösen von Pipeline-Konflikten in Beispielpogrammen
- Sprungvorhersage
  - Vergleich des Verhaltens verschiedener Sprungvorhersagetechniken
- Optimierung von RISC-Programmen zur Vermeidung von Pipeline-Konflikten
- Ermitteln physikalischer Adressen anhand gegebener segmentierter virtueller Adressen und der Seitentabelle

## **Laborübungen zu Digitaltechnik und Rechnerorganisation**

Der Stoff aus den Themengebieten Digitaltechnik und Rechnerorganisation sollte möglichst durch Laborübungen, entweder integriert in den Übungsbetrieb oder als eigenständiges Praktikum, vertieft werden. Durch die praktische Umsetzung von Aufgaben wird die tatsächliche Relevanz des in der Vorlesung vermittelten Stoffes noch einmal betont. Dabei können auch bereits weiterführende Themengebiete anklingen.

### ***Lernziele/Kompetenzen***

Generelles Lernziel von Praktika und Laborübungen ist die Elaboration des in Vorlesungen vermittelten Wissens anhand einer praktischen Umsetzung des Stoffes. Studierende sollen speziell zu den Themengebieten Digitaltechnik und Rechnerorganisation folgende Kompetenzen erwerben:

Die Studierenden sind in der Lage,

- den Entwurf und die Integration einfacher Hardware-Komponenten in ein bestehendes System durchzuführen,
- ein Gefühl für die Komplexität und die Probleme bei der Umsetzung einfacher Programmieraufgaben in Assembler zu entwickeln,
- Interrupts und Ausnahmen zu behandeln,

- die Nutzung einfacher Ein-/Ausgabeschnittstellen z. B. zur Implementierung von Steuerungen/Regelungen zu beherrschen.

### ***Inhalte***

Die hier geplanten Laborübungen bzw. Praktika können grundsätzlich in zwei verschiedenen Konstellationen ablaufen.

In der einen Konstellation wird für alle Aufgabenteile eine einheitliche Plattform zur Realisierung von Hardware-Komponenten, zur Assemblerprogrammierung und zur Ein-/Ausgabe verwendet. In diesem Fall bietet es sich an, als Plattform FPGAs einzusetzen, da diese sowohl zur Realisierung vollständiger Systeme geeignet sind als auch die nachträgliche individuelle Realisierung von zusätzlichen Hardware-Komponenten ermöglichen.

In der anderen Konstellation verwendet man zur Realisierung der einzelnen Aufgabenteile verschiedene Realisierungsplattformen, um unterschiedliche Schwerpunktsetzungen zu ermöglichen. Dabei können Standardschaltkreisbausteine, FPGAs und fertige Mikrocontroller-Systeme zum Einsatz kommen. Es sollte auch die Gelegenheit geboten werden, mit Entwurfswerkzeugen und typischen Messgeräten aus dem Hardware-Entwurfsbereich zu arbeiten (einfache Messgeräte, Oszilloskop, Logikanalysator).

In beiden Fällen sind nur elementare Kenntnisse der Elektrotechnik erforderlich, wie sie in der Vorlesung Digitaltechnik leicht vermittelt werden. Der Schwerpunkt liegt auf der digitalen Schaltungstechnik und der Assemblerprogrammierung unter Nutzung der zugehörigen Entwurfsverfahren und -werkzeuge.

### ***Aufgabenteile***

Konkret werden folgende Aufgabenteile empfohlen, aus denen sich einzelne Laborübungen oder ein eigenständiges Praktikum zusammensetzen sollten:

- Entwurf und Realisierung einfacher Schaltnetze und Schaltwerke
- Entwurf und Integration von Hardware-Komponenten, z. B. UART, ALU, Befehlsdekoder, Adress-Rechenwerk, Interrupt-Controller, ...
  - Entwurf und Simulation einer reinen Verhaltensbeschreibung einer Prozessorkomponente
  - Synthese und Post-Layout-Simulation
  - Inbetriebnahme und Test der realisierten Schaltung
- Assemblerprogrammierung
  - Einfache Assemblerprogramme zur Realisierung von z. B. arithmetischen Funktionen, Textverarbeitung, Display-Steuerung, einfache Eingebettete Systeme
  - Interruptprogrammierung
- Ein-/Ausgabe-Programmierung
  - Bedienung einer E/A-Komponente per Polling
  - Realisierung einer einfachen Steuerung/Regelung
  - Kooperation einer Kontrollschleife und einer interruptgesteuerten Eingabeverarbeitung

## **Betriebssysteme**

Ein Betriebssystem umfasst die Software und Firmware, die den Betrieb eines Rechners ermöglicht. Es verwaltet einerseits die Betriebsmittel eines Rechners möglichst effizient und offeriert andererseits den ProgrammiererInnen geeignete, langlebige Abstraktionen der

zugrunde liegenden Hardware. Die Kenntnis zentraler Betriebssystemkonzepte ist für ein solides Gesamtverständnis moderner IT-Systeme unerlässlich und eine wesentliche Voraussetzung für viele Berufsfelder. Zudem liefern sie die Grundlage für weiterführende Veranstaltungen etwa zu Betriebssystemkonstruktion, Echtzeitverarbeitung, Verteilten Systemen.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Konzepte und Strukturen moderner Betriebssysteme für Ein- und Mehrprozessorsysteme im Zusammenhang zu erklären,
- Prozesse und Threads mit Synchronisationsprimitiven auf konkrete kleinere Aufgaben mit Nebenläufigkeit korrekt anzuwenden,
- die Wirkungsweise verschiedener Scheduling-Verfahren in Hinblick auf verschiedene Güteigenschaften für gegebene kleinere Anwendungsfälle zu analysieren und zu beurteilen,
- verschiedene Formen der nachrichten- und speicherbasierten Interprozesskommunikation zu vergleichen und für Anwendungsfälle zu beurteilen,
- Dateisysteme zu klassifizieren und in Hinblick auf Nutzungsmuster auszuwählen,
- Sicherheitsprobleme zu identifizieren, Schutzmechanismen zu benutzen und Sicherheitsvorgaben zu beurteilen,
- kleinere Anwendungen unter Nutzung großer Teile eines Betriebssystem-APIs zu programmieren, zu testen und deren Leistung zu beurteilen bzw. Engpässe experimentell zu bestimmen.

### ***Inhalte***

- Geschichte der Betriebssysteme, Arten und Einsatzbereiche von Betriebssystemen
- Systemkonzepte und -strukturen
- Prozesse und Threads
- Scheduling
- Prozesssynchronisation
- Interprozesskommunikation
- Speicherverwaltung, Virtual Memory Management
- I/O-Management
- Dateisysteme
- Sicherheit und Schutzmechanismen
- Virtuelle Maschinen
- Beispiele, Fallstudien

### ***Übungen/Praktikum***

Die Übungen sollten, wenn möglich, praktische Anteile enthalten.

- Kennenlernen der Programmierschnittstelle (API) eines Betriebssystems, z. B. UNIX
- Praktische Erfahrungen in der Nutzung externer Schnittstellen von Betriebssystemen sammeln, z. B. durch Programmieraufgaben unter Nutzung von Systemdiensten
- Praktische Erfahrungen in der Nutzung interner Schnittstellen von Betriebssystemen sammeln, z. B. Gerätetreiber und Interrupt Handler, Dateisystemtypen entwickeln
- Leistungsmessungen zu Betriebssystemfunktionalitäten durchführen
- Inkrementelle Entwicklung eines elementaren Betriebssystems auf einem PC-Emulator

## **Rechnernetze**

Fast alle heutigen Rechnersysteme sind vernetzt, wobei das Internet eine wichtige Rolle spielt. Dieses Themengebiet gibt eine Einführung in das Gebiet der Rechnernetze, wobei alle Schichten des Kommunikationssystems betrachtet werden.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die Bedeutung von Schichtenmodellen, die Aufgaben und Funktionsweise der verschiedenen Schichten sowie die wichtigsten Protokoll- und Dienstvertreter in jeder Schicht zu erklären,
- für ein gegebenes Anwendungsproblem zu entscheiden, welche Netztechnologien in den verschiedenen Schichten eingesetzt werden sollten,
- zu erklären, wie das Internet im Kern und in den Endsystemen funktioniert,
- eigene kleine Anwendungen zu programmieren,
- die Konzepte der Protokolle TCP, IP, HTTP und SMTP zu beschreiben.

### ***Inhalte***

Das Themengebiet beschäftigt sich mit folgenden Inhalten:

- Computernetzwerke und das Internet
- Anwendungsschicht
- Transportschicht
- Vermittlungsschicht
- Sicherungsschicht und Bitübertragung
- Zusammenfassung und Ausblick

### ***Übungen/Praktikum***

In den Übungen sollen die Studierenden das in der Vorlesung vermittelte Wissen sowohl anhand von Papierübungsaufgaben nachvollziehen als auch in kleineren Programmieraufgaben einüben. Beispiele für typische Aufgaben:

- Bandbreitenberechnungen für Medien
- Datenratenberechnungen für bestimmte Protokolle
- Nachvollziehen von Protokollen mit Message Sequence Charts
- Verwendung von telnet für verschiedene höherwertige Dienste wie Versenden von Emails
- Socket-Programmierung
- Programmieren eines kleinen http-Servers und -Clients
- Verwendung von Mail-APIs z. B. von Java

## **Rechnerarchitektur**

Die Rechnerarchitektur ist eine allgemeine Strukturlehre mit deren Hilfsmitteln, die als ingenieurwissenschaftliche Disziplin die Aufgabe hat, bestehende und zukünftige Rechnersysteme zu beschreiben, zu bewerten, zu beurteilen und zu entwerfen. Sie betrachtet den Aufbau und die Eigenschaften des Ganzen (Rechnersystem) sowie seiner Teile (Komponenten) und seiner Verbindungen (Globalstruktur, Infrastruktur). Die grundlegenden Kenntnisse der Hardwarestruktur eines Rechners auf einer konzeptionellen Darstellungsebene, die Sicht der AnwenderInnen auf einen Rechner sowie das

Operationsprinzip bilden die Voraussetzung für den effizienten Einsatz von Rechnern in den verschiedenen Anwendungsgebieten.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- den Aufbau, die Organisation und das Operationsprinzip von Rechnersystemen zu erörtern,
- den Zusammenhang zwischen Hardware-Konzepten und den Auswirkungen auf die Software zu analysieren, um effiziente Programme erstellen zu können,
- aus dem Verständnis über die Wechselwirkungen von Technologie, Rechnerkonzepten und Anwendungen die grundlegenden Prinzipien des Entwurfs nachzuvollziehen und anzuwenden,
- Rechnerkonzepte zu bewerten und zu vergleichen.

Insbesondere soll dieses Themengebiet die Voraussetzung liefern, vertiefende Veranstaltungen über moderne Mikroprozessorarchitekturen, Parallelrechner, Fehlertoleranz und Leistungsbewertung zu besuchen und aktuelle Forschungsthemen zu verstehen.

### ***Inhalte***

- Einführung in die Rechnerarchitektur, Klassifikation
- Grundprinzipien des Rechnerentwurfs: Kompromissfindung zwischen Zielsetzungen, Randbedingungen, Gestaltungsgrundsätzen und Anforderungen, Leistungskriterien
- Superskalartechnik, VLIW-Prinzip, spekulative Ausführung, Sprungvorhersage
- Mehrfädige Befehlsausführung
- Speichersysteme
  - RAID
  - Network-Attached Storage
  - Storage-Area-Networks (SAN)
- Parallelrechnerkonzepte, speichergekoppelte Parallelrechner (symmetrische Multiprozessoren, Multiprozessoren mit verteiltem gemeinsamem Speicher), nachrichtenorientierte Parallelrechner, Multicore/Manycore-Architekturen, parallele Programmiermodelle, Konsistenzmodelle und Kohärenzprotokolle
- Verbindungsnetze (Topologien, Routing)
- Grundlagen der Vektorverarbeitung, SIMD, Multimedia-Verarbeitung
- Leistungsbewertung
- Energie-effizienter Entwurf
- Grundlagen der Fehlertoleranz, Zuverlässigkeit, Verfügbarkeit, Ausfallsicherheit
- Sicherer Betrieb von Rechnersystemen und deren Vernetzung

### ***Übungen/Praktikum***

Die Übungen sollten, wenn möglich, auf parallelen Rechensystemen bzw. mit Hilfe geeigneter Simulationswerkzeuge (z. B. für Prozessorarchitekturen, Speicherhierarchien, Verbindungsnetze) durchgeführt werden.

- Grundlagen des Aufbaus von Rechnern: Herausarbeiten der Ausgewogenheit von Rechnerentwürfen
  - Prozessorgeschwindigkeit vs. Bus-Bandbreite/Latenz vs. Speicherkapazität/Speicherzugriffszeiten vs. E/A-System
- Parallelverarbeitung
  - Berechnen der Parameter des Amdahl-Gesetzes und Ableiten von Parallelitätsparametern
  - Berechnung von Speedup- und Effizienzwerten

- Erstellen einfacher paralleler Programme (MPI, OpenMP) und Ausführen auf Simulator, parallelem System und Berechnen von Leistungsparametern
- Verbindungsstrukturen
  - Berechnen der Leistungsparameter verschiedener Verbindungsnetze und Vergleich der Ergebnisse
  - Wegeberechnungen für verschiedene Routing-Verfahren
- Rechnerbewertung, Leistungsbewertung
  - Berechnen von Parametern der CPI-Formel mit einfachen Beispielprogrammen
  - Berechnen von SPEC-Parametern an Fallstudien
  - Vergleich von Rechnern anhand von Benchmark-Daten
- Fehlertoleranz
  - Erstellen von Zuverlässigkeitsgraphen und Ermittlung der Systemfunktion
  - Berechnen von Funktions-, Fehler- und Ausfallwahrscheinlichkeiten

## **Eingebettete Systeme**

Eingebettete Systeme sind informationsverarbeitende Systeme, die in ein umgebendes technisches System eingebettet sind. In der Regel interagieren Eingebettete Systeme mit einer physikalischen Umgebung. Zusammen mit ihrer Umgebung bilden sie so genannte cyber-physikalische Systeme (CPS). Eingebettete Systeme müssen effizient mit Ressourcen umgehen, u. a. auch mit den Ressourcen „Zeit“ und „Energie“. Beispiele Eingebetteter Systeme finden sich u. a. im Verkehrswesen, in der Telekommunikation, in der Produktion, in der Medizintechnik und im Consumerbereich.

### ***Lernziele/Kompetenzen***

Dieser Teil des Curriculums soll die Studierenden in die Lage versetzen, einfache Eingebettete Systeme zu entwickeln und bei der Entwicklung von Cyber-Physical Systems zu kooperieren. Sie sind insbesondere in der Lage,

- Unterschiede zwischen Desktop- und Eingebetteten Systemen zu erklären und zu berücksichtigen,
- Spezifikations- und Modellierungstechniken für Eingebettete und Realzeitsysteme miteinander zu vergleichen und geeignete Techniken auszuwählen,
- die Prinzipien einschlägiger Basistechnologien im Bereich der Hardware und Systemsoftware zu erläutern und einzusetzen,
- Entwicklungstechniken theoretisch und praktisch zu beherrschen,
- Eingebettete Systeme zu beurteilen und zu optimieren,
- Risiken in besonderen Bereichen zu beurteilen,
- die Zuverlässigkeit zu bewerten.

### ***Inhalte***

- Spezifikations- und Modellierungstechniken Eingebetteter Systeme und Cyber-Physical Systems, insbesondere
  - Modell-basierter Entwurf
  - Models of Computation
  - Techniken für daten- und kontrollflussdominierte Systeme
  - Techniken für lokale und heterogene verteilte Systeme
- Prinzipien von Hardware-Plattformen, insbesondere von
  - Hardwareklassen (applikationsspezifische Schaltkreise (ASICs), Field-Programmable Gate Arrays (FPGAs), programmierbare Prozessoren)
- Hardwarekomponenten (Analog/Digital- und Digital/Analogumsetzer einschl.

Abtasttheorem und Verbindung zur physikalischen Umgebung, Prozessoren, Speicher, Peripheriebausteine)

- Kommunikationssysteme
  - Netzwerkanbindung
  - Echtzeitverhalten
  - Vernetzung Eingebetteter Systeme
- Prinzipien der Basissoftware
  - Echtzeit-Betriebssysteme
  - Middleware
- Abbildung von Anwendungen auf Verarbeitungsplattformen
  - Software-Engineering und Software-Entwicklung für Eingebettete Systeme
  - Grundzüge des Echtzeit-Scheduling
  - Abbildung auf (heterogene) Mehrprozessorsysteme
  - Werkzeugunterstützung
  - Debugging
  - Echtzeitverhalten von Anwendungsschichten
- Evaluation
  - Simulative und analytische Verfahren
  - Bewertung von Systemen z. B. hinsichtlich der Latenzzeit, des Energieverbrauchs
  - Laufzeitgarantien
- Optimierung
  - Anwendung von Optimierungstechniken auf den Entwurf Eingebetteter Systeme
  - Mehrkriterien-Optimierung
  - exemplarische Betrachtung spezieller Optimierungstechniken, z. B. von Codekompressionstechniken oder des Energiemanagements
- Grundlagen verlässlicher Eingebetteter und cyber-physikalischer Systeme (Sicherheit, Zuverlässigkeit, Verfügbarkeit)
- Sichere Softwareentwicklung
- Test von Eingebetteten Systemen (z. B. Scan Design, JTAG)
- Exemplarische Betrachtung von Anwendungen und Einbettungen in cyber-physikalische Systeme oder von Beispielen zum Internet der Dinge

### ***Übungen/Praktikum***

- Spezifikation hierarchischer Zustandsautomaten und deren Umsetzung in Hardware oder Software
- Spezifikation von Systemen unter Nutzung von Echtzeit-Profilen z. B. von UML und Java
- Programmierung einfacher Beispielsysteme
- Programmierung mobiler Eingebetteter Systeme
- Datenerfassung physikalischer Daten (z. B. von Sensoren der Medizintechnik)
- Realisierung der Steuerung von technischen Systemen (z. B. Modelleisenbahnen, Kfz-Systeme)
- Ermittlung des Energiebedarfs
- Realisierung von Hardware-in-the-Loop-Systemen in cyber-physikalischen Systemen
- Praktische Demonstration von Auswirkungen der Abtastung
- Nutzung eines Echtzeit-Betriebssystems
- Lösung von Echtzeit-Scheduling-Aufgaben
- Erprobung der Effizienzsteigerung durch Programm-Optimierungen
- Konzeption und Durchführung von Tests Eingebetteter und cyber-physikalischer Systeme
- Demonstration der Auswirkungen der Verlässlichkeitsanforderungen
- Wechselwirkung zwischen lokaler Autonomie und globaler Vernetzung



## **Modellierung und Simulation analog-digitaler Systeme**

Cyber-physikalische Systeme entstehen durch eine ganzheitliche Betrachtung von eingebetteten Rechnersystemen mit ihrer physikalischen Umgebung. Während die Informationsverarbeitung im eingebetteten Rechnersystem mit diskreten Werten zu diskreten Zeitpunkten erfolgt, hat dessen physikalische Umgebung typischerweise ein kontinuierliches/analogenes Verhalten, das z. B. eher durch Differenzialgleichungen beschrieben werden kann. Daher erfordert die Modellierung von Eingebetteten Systemen mit ihrer Umgebung analog-digitale Modelle, die durch diskrete Zustandswechsel jeweils zustandsabhängig unterschiedliche Differenzialgleichungen aktivieren. Aus diesen Modellierungen können Anforderungen hinsichtlich des zeitlichen Reaktionsverhaltens des Eingebetteten Systems abgeleitet werden, die für die weitere Entwicklung des eigentlichen Eingebetteten Systems erforderlich sind. Verwandte Modellierungen sind aus der Elektrotechnik als „mixed-signal“-Systeme bekannt. Das Themengebiet führt in die Modellierung und die Simulation analog-digitaler Systeme ein, die auch aus der Entwicklung von „mixed-signal“-Systemen im Hardwareentwurf bekannt sind. Dabei werden neben formalen Modellierungen wie hybride Automaten und deren Analyseverfahren auch generelle Probleme bei der Modellierung und der Simulation wie Zeno- oder Zero-Crossing-Probleme betrachtet, die zum Verständnis dieser Modellierungen von essenzieller Bedeutung sind.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- analog-digitale Systeme in gängigen Sprachen zu modellieren und zu entwickeln,
- solche Systeme durch Simulation zu analysieren,
- dabei Zeno- oder Zero-Crossing-Probleme zu identifizieren und zu beheben,
- den praktischen Umgang mit Simulations- und Verifikationswerkzeugen zu erlernen.

### ***Inhalte***

- hybride Automaten und Systeme; Analyseverfahren und Entscheidungsfragen
- Beschreibungssprachen und Werkzeuge wie Modelica, Matlab/Simulink oder Dymola
- Simulation und Validierung von analog-digitalen Systemen
- praktischer Umgang mit Werkzeugen wie Matlab/Simulink, AMS-Designer, Eldo

## **Projektarbeit**

In einem Projektpraktikum soll die Bearbeitung von praxisnahen Projekten und die Lösung von umfangreichen Problemen exemplarisch demonstriert werden. Dabei soll die Aufgabenstellung so gestaltet werden, dass eine ganzheitliche, Software und Hardware umfassende Methodik eingesetzt werden muss und dass Teamarbeit erforderlich ist. Dabei soll auch eine Weiterentwicklung der sozialen Kompetenzen erfolgen. Damit sollen insgesamt Lernziele verfolgt und Kompetenzen vermittelt werden, die mittels Vorlesungen und klassischen, an kurzfristig lösbaren Problemen orientierten Praktika nicht vermittelt werden können.

## ***Lernziele/Kompetenzen***

Die Studierenden sind in die Lage,

- komplexe, Hardware und Software umfassende Projekte strukturiert durchzuführen,
- mit anderen Projektbeteiligten zielgerichtet zur Lösung der Gesamtaufgabe beizutragen,
- sowie die notwendigen außerfachlichen Kompetenzen zu entwickeln.

Insbesondere soll dieser Teil die Voraussetzung liefern, um erfolgreich an einem Softwareentwicklungspraktikum teilzunehmen sowie weitergehende Lehrveranstaltungen zu nicht-funktionalen Eigenschaften von Software erfolgreich zu besuchen und aktuelle Forschungsthemen zu verstehen.

## ***Inhalte***

Im Rahmen dieses Moduls werden die grundlegenden Kenntnisse zur Realisierung von Projekten demonstriert und erprobt:

- Phasen der Realisierung komplexer Hardware-/Software-Projekte
- Aufwandsschätzung
- Performance-Vorhersage
- Finanzielle Aspekte
- Konzeption und Realisierung von Teamarbeit
- Auswahl von Hardwareplattformen
- Simulation von Komponenten
- Test von gemischten Hardware-/Software-Systemen
- Projektmanagement
- Konfliktmanagement
- Regelungen zum geistigen Eigentum

## **B. Wahlpflichtbereich Technische Informatik**

### **Verlässlichkeit, Robustheit und Fehlertoleranz**

Mit der zunehmenden Abhängigkeit unserer Gesellschaft vom Einsatz von Rechnern steigen auch die Anforderungen an deren Verlässlichkeit und Robustheit. Gleichzeitig nimmt aufgrund zunehmender Komplexität die Fehleranfälligkeit zu. Fehlertoleranz erlaubt es, dass das System trotz interner Fehler noch das spezifizierte Verhalten aufweist, und ist damit eine wichtige Methode zur Erhöhung der Verlässlichkeit und Robustheit. Schlüssel hierzu ist der gezielte Einsatz nützlicher Redundanz.

#### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die wichtigsten Fehlertypen in Hardware und Software und deren Abstraktion zu Fehlermodellen darzustellen,
- die grundlegenden Redundanztechniken (statische und dynamische Redundanz, Mischformen etc.) zu erörtern,
- einzelne Verfahren der Fehlerdiagnose, der Rekonfiguration, des Wiederanlaufs und der Fehlermaskierung zu erklären,
- typische Anwendungsbeispiele und Beispiele für fehlertolerante Rechner zu beschreiben,
- Fehlertoleranztechniken anhand von mathematischen Zuverlässigkeitsmodellen quantitativ zu analysieren,
- geeignete Fehlertoleranztechniken vergleichend zu beurteilen und für ein gegebenes Anwendungsgebiet auszuwählen.

#### ***Inhalte***

- Grundlegende Begriffe
- Kriterien zur technischen Verlässlichkeit und Robustheit von Rechnern
- Allgemeine Redundanztechniken
- Fehlerdiagnose
- Rekonfiguration und Fehlerbehebung
- Fehlermaskierung
- Beispiele für fehlertolerante Systeme

### **Synthese digitaler Schaltungen**

Die Synthese digitaler Schaltungen ist ein mächtiges Werkzeug zur Erzeugung hochkomplexer Systeme. Hier werden die Methoden und Algorithmen zur automatisierten Synthese von solchen Schaltungen auf abstrakten Ebenen vorgestellt. Im Bereich High-Level-Synthese wird die Software bzw. das Hardware/Software Co-Design von Systemen mit einbezogen.

#### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- durch das grundlegende Verständnis der Synthesevorgänge von digitalen Schaltungen bis hinauf zur Systemebene auf allen Ebenen einen selbständigen Entwurf solcher Systeme und eine Analyse eines Design-Flows oder einzelner Schritte aus diesem durchzuführen,
- in Hardwarebeschreibungssprachen digitale Schaltungen und Systeme zu entwerfen und die Auswirkungen auf eine automatische anschließende Synthese zu erkennen.

## ***Inhalte***

- Y-Diagramm, Entwurfsraum
- Hardwarebeschreibungssprachen
- VHDL-/Verilog
- Logiksynthese
  - Synthese 2- und mehrstufiger Logik
  - Logikoptimierung
  - BDDs
- Codeanalyse und -optimierung
- Codegenerierung
- High-Level-Synthese
  - Scheduling
  - Allocation
- Hardware/Software Co-Design

## **Test von Hardwareschaltungen**

Der Test von Hardwareschaltungen prüft die korrekte Funktion nach der Fertigung. Als sehr kostenintensiver Schritt in der Produktion sind Built-In-Tests sowie automatisch generierte Testmuster essentiell für Schaltungen heutiger Komplexität. Neben dem eigentlichen Testproblem und den zugehörigen Algorithmen vermittelt dieses Themengebiet auch Kenntnisse über Fehlerabdeckung und Ausfallmechanismen, die heute auch die Langzeitzuverlässigkeit berühren können.

## ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Testmuster und Tests eigenständig zu erzeugen,
- die zugrundeliegenden Verfahren (z. B. Testmusterberechnung) zu verstehen und algorithmisch zu beurteilen,
- Built-In- und externe Test-Hardware zu analysieren und zu entwerfen.

## ***Inhalte***

- Verteilungsfunktionen
- Berechnung von Funktions- und Ausfallwahrscheinlichkeiten
- Fehlermodelle für elektronische Schaltungen
- Zuverlässigkeit
- Testmusterberechnung
  - D-Algorithmus
  - SAT-basierte Verfahren
- Sequentielles Testen
- Fehlersimulation
- Design for Test
  - Funktionstestverfahren
  - Scan-Pfad
  - Built-In-Test
- Test-Hardware

## **Programmierung Eingebetteter Systeme**

Ein großer Teil der Funktionalität Eingebetteter Systeme wird mit Hilfe von Software realisiert. Die Entwicklung derartiger Software unterscheidet sich aber wesentlich von klassischer Desktop-/Serversoftware, u. a. wegen der geforderten Verlässlichkeit, der Beachtung der nur begrenzt vorhandenen Ressourcen sowie wegen der Variationsbreite der Hardwareplattformen und der Lösungen.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- charakteristische Unterschiede zwischen Desktop- /serverbasierter Software und der Software Eingebetteter Systeme zu erkennen und Konsequenzen für die Softwareentwicklungsmethoden abzuleiten,
- Ausführungsplattformen, insbesondere Betriebssysteme für Eingebettete Systeme und Middleware-Plattformen, hinsichtlich ihrer Eignung für die Gesamtfunktionalität zu bewerten und auszuwählen,
- quantitative Bewertungen der verfügbaren Ressourcen (Speicher, Energie, Rechenleistung, eingeschränkte Netzwerkkonnektivität) vorzunehmen und mit der geforderten Softwarefunktionalität abzugleichen,
- hardwarenahe Software modellgetrieben zu entwickeln und systematisch zu testen,
- Techniken zur Realisierung verlässlicher Software, zur Realisierung einer großen Vielfalt von Software-Varianten (Produktlinien) und zur Realisierung von vernetzten Eingebetteten Systemen anzuwenden.

### ***Inhalte***

- Modellierung Eingebetteter Systeme
- Hardwarenahe Programmierung von Eingebetteten Systemen, Entwicklung zeitgetriebener und reaktiver Anwendungen, Randbedingungen der Cross-Entwicklung
- Metriken zur quantitativen Bewertung von Anforderungen und von Ausführungsplattformen sowie deren Abgleich
- Software-Produktlinien, Konfigurierbarkeit von Software, aspektorientierte Programmierung
- Betriebssysteme für Eingebettete Systeme
- Vernetzung von Eingebetteten Systemen, Kommunikations-Middleware
- Ausnahmebehandlung und fehlerresistente Software, resiliency
- Zuverlässige Software: Standards und Normen, Zertifizierung
- Modell-basierter Entwurf von Software für Eingebettete Systeme, Software-Synthese (z. B. MDS-Techniken, Matlab/Simulink)
- Systematisches Testen von Software Eingebetteter Systeme, Regressionstests, Automatisierung)
- Verifikationstechniken für Software Eingebetteter Systeme

## **Rechnertechnologie**

Die Rechnertechnologie oder auch Computertechnologie bildet zusammen mit der Rechnerarchitektur die Grundlage moderner Rechnerkomponenten. Sie wird häufig auch einfach als Hardware bezeichnet und kann durchaus als das Reitpferd aller modernen IT-Anwendungen angesehen werden, da sie die Last der Software zu tragen hat. Prozessoren, Halbleiterspeicher und weitere Rechner- und Interfacekomponenten sowie die anwendungs-

spezifischen Schaltungen werden heute überwiegend als hochintegrierte Chips entworfen. Die Komplexität integrierter Schaltungen (VLSI) ist in den vergangenen Jahrzehnten exponentiell gewachsen (Moore'sches Gesetz). In zunehmendem Maße werden auch Logik- und Speicherfunktionen sowie analoge und digitale Funktionen gemeinsam auf einem Chip integriert. Dadurch ist es gelungen, mikroelektronische Implementierungen ganzer Systeme (Silicon on Chip) auf dem Chip durchzuführen.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- den Herstellungsprozess integrierter Schaltungen darzustellen und zu erklären und die technologischen Randbedingungen zu beurteilen,
- die Ausbeute integrierter Schaltungen qualitativ und quantitativ zu analysieren,
- die Grundlagen analoger und digitaler (MOS-)Schaltungstechnik darzustellen,
- die Zeitmodellierung und die Laufzeit digitaler Schaltungen zu erörtern, abzuschätzen und zu bewerten,
- die Grundlagen von FPGA-Schaltungen zu erörtern,
- zu entscheiden, ob und wann eine FPGA-Schaltung sinnvoll eingesetzt werden kann und welche FPGA-Architektur sich zur Implementierung eines speziellen Problems eignet,
- die Speicherhierarchien digitaler Rechensysteme darzustellen und zu beurteilen,
- die Technologien und Organisationsformen von Halbleiter- und Massenspeichern darzustellen,
- Prozessoren und digitale Rechenkomponenten in den Grundlagen darzustellen,
- Prozessoren zu vergleichen, zu evaluieren und für ein Anwendungsgebiet auszuwählen.

### ***Inhalte***

- MOS-Integrationstechnik (VLSI)
  - Planarprozess
  - Ausbeute integrierter Schaltungen
- Schaltungstechnik
  - Logikkomponenten (CMOS Gatter, DOMINO)
  - Programmierbare Schaltungen (PLD, FPGA)
  - Zeitmodellierung
- Speichertechnologien
  - Speicherhierarchie
  - Halbleiterspeicher (SRAM, DRAM, Flash)
  - Zugriffsarten (DDR)
  - CD-ROM (EFM Codierung)
  - Festplatten
- Prozessortechnologien für:
  - Superskalar
  - VLIW
  - Multicores
  - Heterogene Systeme

## **Robotik**

Roboter sind in der industriellen Fertigung heute fest etabliert und ermöglichen eine vollautomatische Produktion mit hoher Qualität und Präzision. Durch ihre Softwaresteuerung lassen sie sich flexibel an die jeweiligen Anwendungen anpassen. Dabei müssen die Bewegungen des Roboters genau festgelegt und geplant werden.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- den Aufbau, die Funktion und Anwendungen von Industrierobotern zu beschreiben,
- die Kinematik von Industrierobotern zu modellieren und zu berechnen,
- Verfahren zur Bahn- und Aufgabenplanung einzuordnen und anzuwenden,
- Industrieroboter zu programmieren.

### ***Inhalte***

- Robotik-Grundlagen
- Komponenten und Subsysteme eines Industrieroboters
- Kinematik
- Bahnplanung
- Aufgabenplanung
- Roboter-Programmierung

## **Medizintechnik**

In der modernen Medizin werden heute eine Vielzahl von medizintechnischen Geräten eingesetzt, die die Ärztin oder den Arzt bei der Diagnose, der Therapie und dem Monitoring von Patientinnen und Patienten unterstützen. Ein großer Teil dieser Geräte ist softwaregesteuert mit stark zunehmender Tendenz. Hierbei sind strenge Richtlinien und Genehmigungsverfahren einzuhalten, um eine Gefährdung der Patientinnen und Patienten sicher zu vermeiden.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die Grundlagen der medizinischen Messtechnik zu beschreiben,
- komplexe Zusammenhänge bei der Messtechnik physiologischer Parameter zu erörtern,
- mit Messunsicherheiten umzugehen,
- die Randbedingungen für den Einsatz medizintechnischer Geräte und die wichtigsten Anwendungsgebiete zu erläutern.

### ***Inhalte***

- Grundlagen der Anatomie und Physiologie
- Verfahren der Funktionsdiagnostik
- Bildgebende Systeme
- Therapiesysteme
- Monitoring
- Medizinische Informationsverarbeitung
- Wichtige gesetzliche Vorschriften
- Medizintechnische Anwendungen

## **Bildverarbeitung/Mustererkennung**

Die Bildverarbeitung und Mustererkennung behandelt die Verarbeitung von Signalen und Bildern sowie die Erkennung und Klassifikation von Objekten aus Signalen und Bildern. Dazu werden Transformationsmethoden, Klassifikatoren und Methoden des maschinellen Lernens benötigt und in problemangepassten Verfahren eingesetzt.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- wichtige Grundbegriffe der digitalen Bildverarbeitung und Mustererkennung zu verstehen,
- Bildverarbeitungs- und Mustererkennungssysteme zu analysieren und zu beurteilen,
- Klassifikatoren problemangepasst auszuwählen und einzusetzen.

### ***Inhalte***

- Grundlagen der Bildverarbeitung
- 2-dimensionale Signale und Spektren, Filter, Transformationen
- Stochastische Prozesse
- Merkmalsreduktion, Merkmalsselektion
- Klassifikatoren
- Neuronale Netze
- Support-Vektor Maschine
- Bayessche Klassifikation

## **Usability Engineering für technische Systeme**

Die Entwicklung und Ausgestaltung technischer Systeme erfordert die Einbeziehung unterschiedlicher Agierender im Entstehungs- und Lebenszyklus eines Produktes. Usability Engineering verfolgt dabei das Ziel, die Gebrauchstauglichkeit und die Nutzerqualität von komplexen technischen Systemen zu erhöhen. Hierzu werden Prinzipien und Konzepte der Mensch-Maschine-Interaktion, Messung von Gebrauchstauglichkeit im Lebenszyklus und Aspekte des Systems Engineerings genutzt.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- grundlegende Prinzipien und Methoden der multimodalen Mensch-Maschine-Interaktion zu beschreiben, zu charakterisieren und zu erklären,
- Methoden der Nutzerqualitäts- und Gebrauchstauglichkeitsmessung darzustellen und zu erörtern,
- Schnittstellenkonzepte in technischen Systemen zu diskutieren,
- Systems Engineering für Informatiksysteme zu erläutern,
- zu entscheiden, wann verschiedene Praktiken eingesetzt werden können und welche Methoden sich für die praktische Erprobung eignen.

### ***Inhalte***

- Basistechniken der multimodalen Mensch-Maschine-Interaktion
  - Psychologische und physiologische Aspekte
  - Multi-modale Perzeption



- Rechnergestützte Verarbeitungsmechanismen in der Interaktion (haptisch, optisch, verbal)
- Methoden der Nutzerqualitäts- und Gebrauchstauglichkeitsmessung
- Schnittstellen
  - physiologisch
  - technisch
- Systems Engineering
  - Grundlagen der Anforderungsanalyse, der Anforderungsdefinition und des Anforderungsmanagements
  - Grundlagen der Modellbildung, Simulation und Bewertung
  - Systemintegration, Verifikation und Validierung

## **Evaluation von Systemen**

Evaluation ist die zentrale Technik zur Beurteilung von Systemen hinsichtlich verschiedener Kriterien. Evaluation ist notwendig, um neben den funktionellen Systemeigenschaften auch die nicht-funktionellen zu beurteilen.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Verfahren der Evaluation von Systemen zu erläutern,
- ein geeignetes Evaluationsverfahren zu wählen,
- Evaluationsexperimente praktisch durchzuführen,
- die Systembeeinflussung durch die Evaluation zu beurteilen,
- die Grenzen der Evaluation zu beurteilen.

### ***Inhalte***

- Simulation, Messung und Modellbildung als Basistechniken der Evaluation
- Evaluation im Hinblick auf folgende Kriterien
  - Evaluation von Systemen im Hinblick auf Performance-Evaluation
  - Beurteilung der Service-Qualität
  - Worst-Case Execution Times
  - Energie- und Leistungsevaluation
  - Thermische Evaluation
  - Zuverlässigkeits-Evaluation

## **Automatisierung Technischer Prozesse (Steuerungs-/Regelungstechnik)**

Die Steuerungs- und Regelungstechnik legt die Grundlagen, um technische Systeme zu automatisieren. Sowohl analoge als auch digitale Regelungsprinzipien werden in diesem Zusammenhang vorgestellt.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Steuerungs- und Regelungssysteme auszuwählen, zu beurteilen und zu planen,
- Automatisierungssysteme zu analysieren, zu simulieren und zu entwerfen,

- den Unterschied zwischen kontinuierlichen und diskreten Regelkreisen zu erklären und zu nutzen.

### ***Inhalte***

- Systembeschreibungen mit Differentialgleichungen und im Frequenzbereich mit Übertragungsgliedern
- Regelkreise
  - Aufbau geschlossener Regelkreise
  - Stabilität
  - Entwurf
  - Zustandsraum
- Diskrete Steuerungen
  - Prozessmodelle
  - Steuerungsentwurf
  - Speicherprogrammierbare Steuerungen
  - Echtzeitverarbeitung
- Fuzzy-Regelung, Software-Synthese FIR, IIR
- Matlab Simulink Code-Generierung

## **Systembeschreibungssprachen**

Um nicht-funktionale Eigenschaften wie geringen Energieverbrauch oder Echtzeitfähigkeit überhaupt erst erfüllen zu können, müssen typischerweise anwendungsspezifische Hardwareplattformen eingesetzt werden. Daher müssen solche Eingebettete Systeme als verteilte Systeme mit parallelen Prozessen modelliert werden, wobei im Gegensatz zu reiner paralleler Software die Modellierung der Zeit eine wichtige Rolle spielt. Zur Beschreibung solcher Systeme wurden unterschiedliche Modellierungen entwickelt, die im Englischen als „models of computation“ (MoC) bezeichnet werden. Ein solches MoC wird dadurch charakterisiert, indem beschrieben wird, wann, warum, welche atomaren Aktionen des Systems ausgeführt werden. Dabei kann man grundsätzlich zwischen daten-, takt- und ereignisgesteuerten Modellen unterscheiden. Dieses Themengebiet des Curriculums behandelt die grundlegenden MoCs anhand von ausgewählten Systembeschreibungssprachen. Zu jedem MoC werden ferner semantische Probleme wie der Abhängigkeit des Verhaltens von zeitlichen Verzögerungen der Eingaben, Kausalität, Sicherstellung endlichen Speicherbedarfs, sowie Maßnahmen bei der Modellierung zur Verbesserung der Abschätzung von Echtzeitschranken erörtert, und es werden entsprechende Analyseverfahren hierzu vorgestellt. Ferner werden Algorithmen zur Synthese von Hard- und Software aus diesen Beschreibungen diskutiert.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- parallele Ausführungsmodelle (MoCs) zu erklären und weiter zu entwickeln,
- für gegebene Anwendungen passende MoCs auszuwählen und hiermit zu modellieren,
- gegebene Modelle hinsichtlich semantischer Inkonsistenzen zu analysieren,
- Modelle eines MoCs in Modelle eines anderen MoCs zu übertragen,
- vorhandene MoCs miteinander zu kombinieren bzw. neue MoCs zu entwickeln.

## ***Inhalte***

- datenflussgesteuerte Prozessnetzwerke (Kahn-Netzwerke, MPI etc.)
- takt-/zeitgesteuerte MoCs (synchrone Sprachen, time-triggered Architectures etc.)
- ereignisgesteuerte MoCs (SystemC, Verilog, VHDL etc.)
- transaktionsgesteuerte MoCs (SystemC etc.)

## **Energieeffizienz**

Die Nutzung der Informationsverarbeitung erfordert je nach eingesetzter Verarbeitungs-Plattform einen gewissen Aufwand an Energie. In der Regel wird dabei elektrische Energie in Wärmeenergie gewandelt. Durch geschickten Einsatz optimierter Techniken können energetisch effiziente Lösungen gestellter Aufgaben erreicht werden. Daraus ergeben sich positive Effekte hinsichtlich der Kosten der Informationsverarbeitung, des Benutzerkomforts, der Auswirkungen auf die Umwelt sowie auch der realisierbaren Anwendungen.

## ***Lernziele/Kompetenzen***

Studierende sind in der Lage,

- die Auswirkungen des Einsatzes informationstechnischer Komponenten auf den Energieeinsatz zu beurteilen,
- Maßnahmen zur Reduktion des Energieeinsatzes zu planen, durchzuführen und zu überprüfen.

## ***Inhalte***

- Begriff der Energie
  - Formen der Energie, Zusammenhang zwischen Leistung und Energie
  - Energieerhaltung
- Quellen von Energie für die Informationsverarbeitung
  - Erzeugung elektrischer Energie (einschl. energy harvesting)
  - Energiespeicherung
- Einsatz von Energie in Systemen der Informationsverarbeitung
  - Ursachen des Energieverbrauchs bzw. der Wandlung in Wärmeenergie bei Rechen- und Kommunikationsvorgängen
  - Vorhersagen zur Weiterentwicklung des Energieverbrauchs (z. B. ITRS-Roadmap)
  - Modellierung des Energieverbrauchs (einzelne Komponenten und life cycle analysis (LCA) für ganze Systeme)
  - Auswirkungen des Energieverbrauchs (u. a. dark silicon, Umwelteinflüsse)
  - Wechselwirkungen mit dem thermischen Verhalten und der Zuverlässigkeit
  - Tradeoff zwischen Energieeinsatz und Dienstqualität
- Techniken zur Reduktion des Energieverbrauchs
  - Reduktion des Energieaufwandes über die vollständige Lebenszeit von informationstechnischen Komponenten (Entwurf, Fertigung, Transport, Nutzung und Entsorgung)
  - Reduktion des Energieaufwandes auf den unterschiedlichen Ebenen (Layout, Schaltkreis, Systemhardware, Software): low power modi, voltage scaling, Task-Migration u. a.
  - Modellierung des Nutzerverhaltens

## **Virtualisierung**

Die Virtualisierung hat in vielen Bereichen der Informatik sehr an Bedeutung gewonnen. Insbesondere im Umfeld der Rechnerarchitektur wird die Virtualisierung in nahezu allen Ebenen eingesetzt, um durch Abstraktion eine vereinfachte Nutzung bestimmter Rechnerressourcen zu ermöglichen. Den AnwenderInnen wird hierfür eine Abstraktionsschicht auf Software- oder Hardwareebene zur Verfügung gestellt. Diese logische Schicht wird zwischen AnwenderIn und Ressource eingeführt, um die physischen Gegebenheiten der zu virtualisierenden Ressource zu verstecken. Die Virtualisierung ermöglicht auf diese Weise, die Ressourcen des Computersystems transparent zusammenzufassen oder aufzuteilen.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Konzepte der vielfältigen Virtualisierungslösungen in den verschiedenen Architekturfeldern zu beschreiben,
- Beispiele einzuordnen und zu beurteilen,
- vorhandene Lösungen für den Betrieb virtueller Maschinen anwendungsbezogen auszuwählen und in der Praxis einzusetzen,
- Effekte der Virtualisierung zu beurteilen und durch Messungen quantitativ zu bewerten,
- Virtualisierungstechniken in eigenen Hard- und Softwareentwicklungen anzuwenden.

### ***Inhalte***

- Einführung (Grundbegriffe, Bezug Abstraktion, abstrakte virtuelle Maschinen)
- Zusammenfassende Wiederholung Betriebssysteme (CPU-Virtualisierung, Virtual Memory Management, Hintergrundspeicher-Virtualisierung)
- Virtuelle Maschinen (VM) (Betriebssystemvirtualisierung, Hosted VM-Virtualisierung, Hardware-Layer-Virtualisierung)
- Speichervirtualisierung
- Netzwerkvirtualisierung
- I/O-Virtualisierung
- Maskierung von Verteiltheit
- Virtuelle Rechensysteme (Private/Community/Public Cloud Computing, IaaS/PaaS/SaaS, Mandantenfähigkeit, Abgrenzung Grid Computing)

## **Vernetzte Systeme**

Vernetzte Systeme stellen heute den Standard in komplexen Anlagen, Systemen oder Fahrzeugen dar. Die Herausforderungen, die an derartige Systeme gestellt werden, basieren im Wesentlichen auf ihrer Prozessechtzeitfähigkeit, entsprechenden Betriebssystemen, Middleware-Konzepten und Lastverteilung und Datenmanagement. Beispiele vernetzter Systeme sind u. a. in Fahrzeugen (vernetzte Steuergeräte) oder in großen Produktions- und Fertigungsanlagen (Maschinen- und Anlagenbau, Prozess- und Chemieindustrie), aber auch im Bereich IT-Infrastruktur (Gebäude, Unternehmen) zu finden.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Grundlagen vernetzter Systeme zu erklären,
- Netze und Protokolle zu beschreiben und zu charakterisieren,

- Eigenschaften und Verhalten vernetzter Systeme zu beschreiben, zu charakterisieren und deren Einsatz zu erklären,
- Prinzipien der Modellierung vernetzter Systeme zu erläutern,
- Betriebssysteme und Middleware zu bewerten und deren Einsatz festzulegen,
- zu entscheiden, wann verschiedene Netztopologien eingesetzt werden.

### ***Inhalte***

- Netzwerke und Kommunikationsprotokolle
- Betriebssysteme, Datenbanken und Lastmanagement
- Web Services und softwareorientierte Architekturen
- Middleware-Konzepte
- Komplexitätsanalyse in Systemen

## **C. Informatik**

Bzgl. der Themengebiete der Informatik sei auf die Empfehlungen der GI für Bachelor- und Masterprogramme im Studienfach Informatik an Hochschulen [8] verwiesen.

## **D. Pflichtbereich Elektrotechnik**

### **Grundlagen der Elektrotechnik**

Die Grundlagen der Elektrotechnik führen über die physikalischen Grundlagen in die Elektrotechnik ein. Damit sollen alle weiterführenden Vorlesungen in die Lage versetzt werden, auf einheitliche Kenntnisse zurückzugreifen.

#### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Felder qualitativ für einfache Anordnungen zu analysieren und zu erzeugen,
- den Zusammenhang zwischen Feld und Bauelement zu erklären/analysieren,
- einfache Gleichstromnetzwerke und Wechselstromnetzwerke zu berechnen,
- einfache Schaltkreise im Frequenzbereich zu analysieren.

#### ***Inhalte***

- Elektrisches Feld
- Magnetfeld
- Bauelemente
  - Widerstand
  - Kapazität
  - Induktivität
- Kirchhoffsche Sätze
- Gleichstromkreise
- Wechselstromrechnung
  - Zeigerdiagramme
  - Übertragungsfunktionen
  - Laplace-Transformation
- Struktur und Funktion von einfachen Filtern
- Schaltvorgänge
  - Einschwingvorgänge
  - Differentialgleichungen

#### ***Übungen/Praktikum***

- Netzwerkberechnung
- Wechselstromberechnung
- Simulation von Netzwerken möglichst am Rechner

### **Elektronische Schaltungstechnik, Bauelemente**

Dieser Themenbereich baut auf den elektrotechnischen Grundlagen auf, legt die Grundlagen in den Bauelementen und führt in verschiedene Zweige der Schaltungstechnik ein. Auf Energietechnik sowie HF-Technik kann verzichtet werden. Zumindest eines von beiden sollte im Wahlpflichtbereich hörbar sein.

## ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Lineare Netzwerke zu analysieren und zu erzeugen,
- Transistor-Modelle, Gleichungen, Ersatzschaltbilder zu erläutern und anzuwenden sowie den Aufbau von Dioden und MOS-Transistoren zu erklären,
- Grundsaltungen sowie Transistorschaltungen zu verwenden, zu analysieren, zu berechnen und zu erzeugen.

## ***Inhalte***

- Bauelemente
  - Diode
  - Bipolar
  - MOS-Transistor
- CMOS-Prozess – Fertigungstechnologie
- Nanotechnologie
  - Memristor
  - CNT
- Analoge Schaltungstechnik
  - Eintransistorschaltungen
  - Mehrtransistorblöcke
    - Stromspiegel
    - Differenzverstärker
- Verstärker
  - Operationsverstärker
  - Komparatoren
  - Endstufen
- Mischer Modulation/Demodulation
  - Mischer
- Digitale Schaltungstechnik, Gatter
  - Einfache Logikgatter
  - Energiebetrachtungen (dark silicon)
  - Laufzeitberechnung
  - FPGAs, PLAs

## ***Übungen/Praktikum***

- Entwurf und Berechnung von einfachen Transistorschaltungen
- Entwurf und Berechnung von Signalverarbeitungsketten
- Analyse von gegebenen Layouts, Schaltplänen
- Simulationen mit Schaltungssimulatoren

## **Signale und Systeme**

Der Themenbereich Signale und Systeme legt die theoretischen Grundlagen für die Beschreibung, Verarbeitung und Transformation von Signalen für eine Vielzahl von technischen Systemen. Die Einführung der Fourier- bzw. Laplace-Transformation ermöglicht eine mächtige systemtheoretische Beschreibung und erlaubt eine systematische Analyse und einen systematischen Entwurf, die in vielen weiterführenden Modulen gebraucht werden.



## ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- zwischen Zeit- und Frequenzbereich zu unterscheiden und Transformationen zwischen diesen anzugeben und durchzuführen,
- Systeme zu analysieren und zu klassifizieren,
- die systemtheoretischen Konzepte für LTI-Systeme zu erklären und zur Analyse und zum Entwurf von realen Systemen zu übertragen.

## ***Inhalte***

- LTI-Systeme
- Fourier-Transformation
- Laplace-Transformation
- Bode-Diagramm
- Faltung
- Übertragungsfunktion, Impuls-, Sprungantwort
- Systemanalyse im Frequenzbereich
- Regelkreise
- DFT, FFT
- Filter
- Klassifikation
  - Kontinuierliche
    - Tiefpass
    - Bandpass ...
- Diskrete Signale
  - Abtastung
  - Abtasttheorem
  - Z-Transformation
- Diskrete Filter
  - FIR
  - IIR
- Analoge Übertragungsverfahren: AM und FM

## ***Übungen/Praktikum***

- Ausführlicher, vertiefter Umgang/Berechnungen mit Frequenzbereichsdarstellungen und Transformationen
- Konstruieren und Interpretieren von Bode-Diagrammen
- Entwurf und Analyse von Filtern sowohl im Kontinuierlichen als auch Diskreten

## **Analog-digitale Systeme**

Dieser Themenbereich legt die Grundlagen im Systementwurf von komplexen Hardwaresystemen. Er bietet eine Systematik zur Einordnung und zum Entwurf von komplexen analogen und digitalen Systemen. Die Einordnung in Eingebettete Systeme sollte vorbereitet werden.

## ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Mixed-Signal-Systeme zu analysieren und zu entwerfen,
- Verhaltensmodelle von Mixed-Signal-Systemen zu erzeugen,
- analoge und digitale Teilsysteme auszuwählen, zu bewerten und zu optimieren,
- ein gewünschtes System in ein Systemmodellierungs/-simulationswerkzeug zu übertragen und zu simulieren/bewerten.

## ***Inhalte***

- Y-Diagramm
- Entwurfshierarchie
- Beschreibungssprachen
  - SystemC-AMS
  - VHDL/Verilog – AMS
- Wandler
  - A/D, D/A
  - Flash-Wandler
  - $\Sigma\Delta$ -Wandler
- Analoge Filter/Aliasing
- Transceiver
  - Bussysteme
  - Modulation/Demodulation im Frequenz/Zeitbereich
  - Heterodyne/Image Reject
- Sensorik
- Aktorik
- Regelschleifen
- Modellierung komplexer Regelschleifen mit Sensoren, Aktoren
- Fallstudie, z. B. Handy, Quadcopter

## ***Übungen/Praktikum***

- Entwurf und Berechnung von Mixed-Signal-Systemen
- Simulationen mit Schaltungssimulatoren und/oder Systemsimulatoren wie z. B. Matlab/Simulink, SystemC
- Praktische Aufbauten und Messungen
- Behandlung der Fallstudie mit Auslegung und Umsetzungen von Teilsystemen

## E. Wahlpflichtbereich Elektrotechnik

### Kommunikationstechnik

Die Kommunikationstechnik beinhaltet die Theorie und praktische Ausführung von Informationsübertragung über Kanäle, die in ihrer technischen Ausführung auf verschiedenste physikalische Medien aufsetzen kann. Ein wichtiges Element ist die theoretische Herleitung und Modellierung von Übertragungskapazitäten sowie geeignete Codierungen, um über gestörte Kanäle übertragen zu können.

#### *Lernziele/Kompetenzen*

Die Studierenden sind in der Lage,

- die grundlegenden Zusammenhänge der Informationsübertragung über Kanäle zu erklären,
- gegebene Kanalcodierungen und Übertragungsverfahren darzustellen, zu beurteilen und zu optimieren,
- Komponenten auszuwählen und ein Kommunikationssystem über einen Kanal zu erzeugen.

#### *Inhalte*

- Modulation/Demodulation
- Quellen und Kanäle:
  - Entropie und Kanalkapazität
  - einfache Kanalmodelle Binärkanal, Gauß-Kanal
- Quellencodierung: Diskrete und kontinuierliche Nachrichtenquellen
  - Rate Distortion Funktion
  - Entropiecodierung – Quantisierung und Kompression
  - Prädiktive Codierung – Transformationscodierung
- Kanalcodierung:
  - Blockcodes
  - Faltungscodes
  - Algorithmen zur Decodierung
- Modulationsarten:
  - Amplitude Shift Keying (ASK)
  - Phase Shift Keying (PSK), DPSK, QPSK, QAM
  - Frequency Shift Keying (FSK) – kohärenter und inkohärenter Empfang
- Demodulation und Störverhalten
- Multiplex- und Vielfachzugriffsverfahren:
  - Zeitmultiplex
  - Frequenzmultiplex
  - Code Division Multiple Access (CDMA)
  - Orthogonal Frequency Division Multiplex (OFDM)

**Bemerkung:** Die Themengebiete „Signale und Systeme“ sowie „Kommunikationstechnik“ könnten zu „Nachrichtentechnik“ oder „Systemtheorie“ zusammengelegt werden.

## **Drahtlose Kommunikation**

Die drahtlose Kommunikation ist im Hinblick auf das nachrichtentechnische Verständnis von Funkkommunikation für die Technische Informatik von großer Bedeutung. Das Themengebiet soll ein Verständnis des Funkkanals, der Signalmodellierung sowie der wichtigsten Übertragungs- und Empfangsprinzipien in der Funkkommunikation vermitteln. Des Weiteren soll ein Überblick über Aufbau und Technik ausgewählter aktueller Funkstandards vermittelt werden.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- grundlegende Eigenschaften drahtloser Kommunikationssysteme darzustellen,
- mathematisch zu beschreiben,
- Entwurfsentscheidungen methodisch fundiert zu treffen,
- in diesem Zusammenhang die dargestellten Inhalte zu erklären, abzuschätzen und zu bewerten.

### ***Inhalte***

- Eigenschaften und Modellierung des Funkkanals
- Detektion und Diversität
- Algorithmen und Signalverarbeitung im Physical Layer
- Algorithmen und Signalverarbeitung im Data Link Layer
- Mobilfunksysteme und drahtlose Netzwerke

## **Messtechnik**

Der Themenbereich Messtechnik vermittelt Grundlagen zur elektrischen Messtechnik und geht auf Messfehler und theoretische und praktische Signalverarbeitung zur Messtechnik ein.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- messtechnische Probleme zu analysieren,
- messtechnische Methoden auszuwählen und anzuwenden,
- Messabweichungen zu beurteilen,
- Messaufbauten zu erzeugen.

### ***Inhalte***

- Statistische Grundlagen
- Einheitensystem
- Messverfahren
  - Resistive
  - Kapazitive, induktive
  - Optische...
- Messabweichungen/Fehlerrechnung
- Messketten
- Messaufnehmer
- AD-Umsetzer
- Signalverarbeitung
- Digitale Messeinrichtungen

# **Grundlagen der Mechatronik**

Mechatronische Systeme sind heute sehr komplexe, vernetzte und hochgradig adaptive Systeme. Sie enthalten maschinenbauliche, elektrotechnische und informationstechnische Komponenten und bewirken in ihrem Zusammenspiel eine gewünschte Gesamtfunktion. Dieser gesamtheitliche Blickpunkt sollte daher Grundlage eines Themengebietes sein, bei dem diese gesamtheitliche Sicht mit Detailwissen aus verschiedenen Wissensdomänen angereichert wird. Es ist selbstverständlich, dass dabei die Systemtheorie eine zentrale Rolle spielt.

## ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die Grundlagen der Mechatronik in ihren etwa gleichgewichtigen Anteilen aus Elektrotechnik, Maschinenbau und Informatik zu erklären,
- die Gesetze der Mechanik und die Prinzipien des maschinenbaulichen Konstruierens darzustellen und anzuwenden,
- neben der Theorie und den Gesetzmäßigkeiten verschiedener physikalischer Realisierungen die spezifischen Lösungsmethoden zu erörtern, abzuschätzen und zu bewerten,
- die Austauschbarkeit der Realisierungen auf der Basis von Randbedingungen zu erkennen und zu beurteilen,
- die Systemtheorie (in der Regelungstechnik) anzuwenden,
- die Prinzipien der reaktiven Software und ihre Umsetzung auf der Basis geeigneter Systemsoftware darzustellen und zu beurteilen,
- Grundlagen der Mess- und Sensortechnik darzustellen,
- Grundlagen der Aktorik darzustellen,
- das Wissen in die berufliche Praxis umzusetzen.

## ***Inhalte***

- Grundlagen der Mechatronik
- Modellbildung in der Mechatronik
- Signal- und Systemtheorie
- Werkstoffe der Mechatronik
- Reaktive Systemsoftware
- Mess- und Sensortechnik
- Aktorik

## II. MASTER

### A. Methodischer Bereich

#### Parallelrechner

Parallelrechner bestehen aus mehreren parallel arbeitenden Prozessoren oder Recheneinheiten, die zur Lösung einer gemeinsamen Aufgabe miteinander kooperieren. Dies setzt entsprechende Verbindungsnetzwerke zwischen den Prozessoren oder Recheneinheiten voraus und erfordert eine parallele Programmierung. Wichtige Aspekte sind dabei die Leistungsbewertung, Fehlertoleranz und Energieeffizienz. Im Zuge von Grid und Cloud Computing sind Parallelrechner über das Internet verbunden und werden damit virtualisiert aus der Ferne nutzbar.

#### *Lernziele/Kompetenzen*

Die Studierenden sind in der Lage,

- Modelle für parallele Verarbeitung zu bestimmen und zu motivieren,
- unterschiedliche Parallelrechnerarchitekturen zu beschreiben und zu charakterisieren,
- die Problematik paralleler Programmierung und mögliche Lösungsansätze zu erklären,
- die grundlegenden Konzepte und die Funktionsweise von parallelen Betriebs- und Speichersystemen darzustellen,
- die Konzepte zur Fehlertoleranz in Parallelrechnern zu erörtern,
- die Problematik der Energieeffizienz zu diskutieren,
- die grundlegenden Ansätze für Grid und Cloud Computing zu erläutern,
- die Leistungsfähigkeit (z. B. Speedup, Effizienz) von Parallelrechnern anhand mathematischer Modelle zu analysieren und zu bewerten,
- zu entscheiden, ob und wann ein paralleles Rechnersystem sinnvoll eingesetzt werden kann, welche Parallelrechnerklasse sich zur Lösung eines speziellen Problems eignet und wie viele Prozessoren sinnvoll einsetzbar sind.

#### *Inhalte*

- Motivation und Grenzen für parallele und verteilte Verarbeitung
- Modelle der Parallelverarbeitung
- Klassifikation von Parallelrechnern und Prinzipien der parallelen Verarbeitung
- Multi/Manycore-Prozessoren
- Multiprozessorsysteme und Multicomputersysteme (Cluster)
- Vektorrechner und Feldrechner
- Betriebssystemunterstützung für parallele Rechner
- Parallele Speichersysteme (Storage)
- Leistungsbewertung paralleler Systeme
- Fehlertolerante Parallelrechner
- Energieeffizienz
- Virtualisierung
- Grid und Cloud Computing

#### Hardware/Software Co-Design

Die hohe Rechenleistung bei gleichzeitig möglichst geringem Energieverbrauch heutiger Eingebetteter Systeme lässt sich oft nur durch die Verwendung spezieller Hard- und Software

erreichen. Da der Aufwand zur Konstruktion solcher spezieller Entwürfe sehr hoch ist und eine Integration somit erst spät erfolgen kann, werden hierfür oft virtuelle Prototypen eingesetzt. Das Themengebiet führt in den gemeinsamen Entwurf von Hard- und Software ein und zeigt deren Unterschiede hinsichtlich nicht-funktionaler Eigenschaften. Dabei werden Hardware-Entwürfe auf der Basis von FPGAs und Software-Entwürfe mit Standard- und anwendungsspezifischen Prozessoren miteinander verglichen.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- verschiedene Entwurfsmethodiken zu beschreiben,
- geeignete HW/SW-Partitionierungen problemangepasst auszuwählen,
- die Simulation und Integration von Systemteilen durchzuführen,
- verschiedene Entwurfsalternativen zu vergleichen.

### ***Inhalte***

- HW-Entwicklung mit FPGAs
- vertiefte Kenntnisse der Prozessorarchitekturen
- anwendungsspezifische Prozessoren und retargierbare Compiler
- WCET-Analyse und Echtzeitfähigkeit

## **EDA (Electronic Design Automation)**

Die Vorlesung vermittelt Kenntnisse und Kompetenzen über die Werkzeuge und Methoden des rechnergestützten Schaltungsentwurfs. In systematischer Form sollten die Design-Flows und ihre Werkzeuge im Einzelnen mit Bezügen zur Informatik, aber auch zur Halbleitertechnologie und zur Schaltungstechnik vorgestellt werden. Hier ist gesondertes Augenmerk auf die Komplexität der Entwürfe und die komplexitätsreduzierenden Abstraktionen zu richten.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- einen Entwurfsablauf aus Automatisierungs- und Komplexitätssicht zu beurteilen,
- den Zusammenhang zwischen informatischen Fragestellungen und ihrer Anwendung auf die Schaltungstechnik zu analysieren und zu übertragen,
- einfache eigene EDA-Werkzeuge zu entwerfen,
- anforderungsspezifische Entwurfsabläufe zu erzeugen.

### ***Inhalte***

- Überblick über den System- und IC-Entwurf
- Entwurfsebenen, Entwurfsstile
  - Y-Diagramm
- Werkzeuge für den funktionellen Entwurf
  - Digitale Synthese
  - Formale Verifikation
  - Simulation, Emulation
  - Timing-Analyse
  - Logik-Optimierung

- Werkzeuge für den physikalischen Entwurf
  - Platzierung
  - Verdrahtung
  - DRC, Extraktion, LVS

## **Optimierung**

In einer Vielzahl von Anwendungsgebieten der Informationstechnik spielt die Optimierung von Systemen im Hinblick auf Zielfunktionen eine große Rolle. Der mögliche Optimierungsspielraum wird dabei durch Variable und deren mögliche Werte modelliert. Es gilt, unter vorhandenen Randbedingungen Variablenbelegungen derart zu finden, dass Zielfunktionen Extremwerte (Maxima oder Minima) annehmen.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Standardoptimierungsverfahren (ganzzahlige Programmierung, Simulated Annealing, evolutionäre Algorithmen) darzustellen,
- Optimierungsprobleme zu modellieren,
- die Komplexität eines Optimierungsproblems zu beurteilen,
- ein geeignetes Optimierungsverfahren auszuwählen,
- die Güte eines Lösungsverfahrens zu beurteilen.

### ***Inhalte***

- Modellierung von Optimierungsaufgaben
- Standardverfahren der Optimierung
  - Ganzzahlige Programmierung
  - Branch-and-bound
  - Simulated Annealing
  - Neuronale Netze
  - Evolutionäre Algorithmen
- Anwendungen
  - Exemplarische Anwendungen obiger Verfahren

## **Rekonfigurierbare Architekturen**

FPGAs sind eine seit langem im Einsatz befindliche Technologie zur Realisierung digitaler Systeme. Sie bieten auch die Möglichkeit, Systeme zur Laufzeit („Online“) neu zu konfigurieren, um sie den Bedürfnissen individueller Anwendungen anzupassen. Neben FPGAs gibt es weitere Schaltungstechniken, die eine Konfigurierbarkeit beinhalten (z. B. Coarse-Grain Arrays). Zukünftige Technologien werden noch in viel stärkerem Maße konfigurierbar sein, so dass ein grundsätzliches Verständnis der Möglichkeiten und Erfordernisse rekonfigurierbarer Architekturen eine große Bedeutung erlangen wird. High-Level-Compiler für adaptives Rechnen erleichtern den Einsatz rekonfigurierbarer Systeme.



## ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die Folgen des Einsatzes einzelner konfigurierbarer Techniken abzuschätzen,
- kritische/relevante Code-Stellen in Anwendungen zu erkennen,
- in Abhängigkeit von der Anwendung eine geeignete Systemarchitektur festzulegen,
- eine geeignete Hardware-Software-Partitionierung vorzunehmen,
- aus der Kenntnis der Kompilationstechniken heraus nützliche Veränderungen am Code vorzunehmen.

## ***Inhalte***

- Konfigurierbare Technologien (Fuses, LUTs, Antifuses)
- Technologie und Architektur von FPGAs
- Entwurfswerkzeuge
- Eingebettete Prozessoren in FPGAs
- Hardware/Software Co-Design
- Management der Rekonfiguration in FPGAs
- Modells of Computation, Systemarchitekturen, Wechselspiel Prozessor ↔ Rekonfigurierbarer Datenpfad
- Memory Management, Banking, Interleaving, Caching
- Profiling, Code-Analyse
- High-Level-Synthese, Adaptive Compiler
- Abbildung von Kernen auf rekonfigurierbare Ressourcen, Kompilationstechniken, Modulo-Scheduling
- Anwendungsbeispiel eines adaptiven Rechners aus der Bildverarbeitung

## **Echtzeitsysteme**

Echtzeitbetrieb ist ein Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind, derart, dass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind (DIN 44330). Viele Echtzeitsysteme sind heute Eingebettete Systeme. Sie sind daher Teil eines größeren Hardware/Software-Systems. Sie interagieren mit externen Umgebungen, sind häufig ereignisgetrieben (reaktiv) oder zeitgesteuert und arbeiten häufig nebenläufig und verteilt. Herausragende Eigenschaften von Echtzeitsystemen sind von den Anwendungen vorgegebene Zeitbedingungen. Über die klassischen technischen Anwendungen hinaus hat die Echtzeitverarbeitung auch in vielen anderen Anwendungsbereichen, wie etwa der Verarbeitung von Multimedia-Strömen, Einzug gehalten.

## ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Anforderungen von Echtzeitanwendungen zu analysieren und diese zu beurteilen,
- Modellierungsmethoden für Echtzeitanwendungen anzuwenden,
- die verbreitetsten Echtzeit-Planungsverfahren zu erläutern, deren Einsatz für Echtzeitanwendungen zu planen und deren Wirkung vorherzusagen,
- Struktur, Schnittstellen und Kenngrößen von Echtzeitbetriebssystemen in Hinblick auf die Nutzung für Echtzeitanwendungen zu beurteilen,
- die Programmierung von Echtzeitanwendungen ereignisgesteuert oder zeitgesteuert auch unter Einsatz von Echtzeitbetriebssystemen zu konzipieren und durchzuführen,

- die Programmierung verteilter Echtzeitanwendungen unter Einsatz einer Echtzeit-Middleware zu planen und durchzuführen,
- Kenngrößen von Echtzeitsystemen und -Anwendungen experimentell zu bestimmen und zu beurteilen.

### ***Inhalte***

- Grundlagen der Echtzeitverarbeitung (Grundbegriffe, Vorhersagbarkeit, Verlässlichkeit, Klassifizierung, Wertfunktionen, ereignis- und zeitgesteuerte Verarbeitung)
- Modellierung von Echtzeitanwendungen (z. B. SysML, UML MARTE)
- Planungsverfahren/Scheduling
- Überblick zu Echtzeitsprachen und ihren Programmiermodellen
- Architektur und Schnittstellen von Echtzeit-Betriebssystemen
- Verteilte Systeme und Echtzeitkommunikation, Echtzeit-Middleware
- Beispiele

## **Debugging und Validierung von Software in Eingebetteten Systemen**

Die Software-Entwicklung ist heutzutage für einen wesentlichen Teil der Kosten bei der Entwicklung von Eingebetteten Systemen verantwortlich. Zunehmend spielen dabei insbesondere das Auffinden von Fehlern und damit auch der Nachweis der korrekten Funktion eine entscheidende Rolle. Gerade der Trend zu Multi-Core-Systemen verstärkt die Probleme, die sich aus der eingeschränkten Beobachtbarkeit von Prozessoren in Eingebetteten Systemen ergeben. Dieses Themengebiet soll den Studierenden Kenntnisse und Techniken vermitteln, die zur Fehlersuche und zur Validierung der korrekten Funktion in Eingebetteten Systemen eingesetzt werden können.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die Beobachtbarkeit von Software in Eingebetteten Systemen darzustellen,
- verschiedene Debug-Architekturen zu beschreiben,
- Verfahren zum Aufspüren von Fehlern in eingebetteter Software zu beherrschen und anzuwenden,
- Technologien zum Gewinnen von Traces aus Eingebetteten Systemen zu erklären.

### ***Inhalte***

- Ausführung von strukturierten Programmen, Aufrufhierarchie, Stack-Layout
- Debug-Information in binären Programmdateien
- Kontrolle des beobachteten Systems
  - Realisierung von Haltepunkten
  - Kommunikation mit der Außenwelt
- Debug-Schnittstellen, BDI, JTAG, NEXUS
- Zugriff auf Stack und Heap von externen Systemen aus
- Trace-Schnittstellen (ETM, MCDS, NEXUS, Aurora)
- Limitierungen der Trace-Schnittstellen
- Konfigurationsoptionen für Trace-Schnittstellen (Instr.-Trace, Data-Trace)
- Rechtliche Aspekte von Traces: DO 178B, ISO 26262
- Architektur von Trace-Werkzeugen

## **Formale Verifikation**

Viele Eingebettete Systeme werden in sicherheitskritischen Anwendungen eingesetzt und müssen daher frei von Entwurfsfehlern sein. Der fehlerfreie Entwurf von Software ist allerdings bereits bei Standardsoftware ein großes Problem, welches bei Eingebetteten Systemen aufgrund der Verwendung von spezieller Hardware, parallelen Abläufen und einzuhaltenden Echtzeitschranken wesentlich erschwert wird. Aus diesem Grund wurden in den letzten Jahren verstärkt Verfahren der formalen Verifikation für diese Systeme entwickelt, mit denen man nachweisen kann, dass ein modelliertes System die angegebenen temporalen und funktionalen Spezifikationen erfüllt. Dieses Themengebiet behandelt die Modellierung von zustandsbasierten Eingebetteten mit Transitionssystemen sowie deren symbolische Repräsentation, die von allen gängigen Verifikationsverfahren vorausgesetzt wird. Ferner werden Logiken zur Spezifikation zeitlichen Verhaltens behandelt, wobei Prädikatenlogiken und temporale Logiken eine besondere Rolle spielen. Als Verifikationsverfahren werden neben klassischen fixpunktbasieren Model-Checking-Verfahren auch neue unvollständige Verfahren besprochen.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Systeme als Transitionssysteme zu modellieren,
- deren Äquivalenz nachzuweisen,
- zeitliche Eigenschaften formal zu spezifizieren,
- vorhandene Spezifikationslogiken weiterzuentwickeln,
- Verifikationswerkzeuge zu bedienen,
- neue Verifikationsverfahren für neue oder eingeschränkte Spezifikationslogiken zu entwickeln.

### ***Inhalte***

- Transitionssysteme und deren Äquivalenzbegriffe (wie Bisimulation)
- symbolische Repräsentation von Transitionssystemen u. a. auch BDDs
- Aussagenlogik und Prädikatenlogik als allgemeine Spezifikations Sprachen
- temporale Logik zur Spezifikation zeitlicher Abläufe
- klassisches Model-Checking durch Fixpunktiterationen
- SAT- und SMT-basierte Verfahren

## **Technische Zuverlässigkeit**

Die Zuverlässigkeit technischer Systeme ist eine nicht-funktionale Eigenschaft, die für den Einsatz des Systems vor allem in kritischen Bereichen, aber auch allgemein für einen wirtschaftlichen Betrieb von großer Bedeutung ist. Sie lässt sich qualitativ oder quantitativ mit stochastischen Methoden beschreiben und sollte bereits beim Entwurf des Systems berücksichtigt werden. Einen wichtigen Einfluss auf die Zuverlässigkeit haben die Qualitätssicherung und die Wartbarkeit des Systems.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die Grundkonzepte der Technischen Zuverlässigkeit zu diskutieren,
- die Zuverlässigkeit von technischen Systemen mit mathematischen Modellen zu analysieren,
- Qualitätstests auszuwählen und anzuwenden,
- eine Wartbarkeitsanalyse durchzuführen,
- Entwurfsrichtlinien beim Entwurf zuverlässiger und wartbarer Systeme zu befolgen.

### ***Inhalte***

- Grundkonzepte
- Zuverlässigkeitsanalyse
- Qualifikationstests
- Wartbarkeitsanalyse
- Entwurfsrichtlinien für Zuverlässigkeit, Wartbarkeit und Software-Qualität

## **Organic Computing**

Organic Computing bezeichnet robuste, fehlertolerante und emergente Systeme. Sie weisen sogenannte Selbst-X-Eigenschaften wie z. B. Selbst-Organisation, Selbst-Konfiguration, Selbst-Optimierung, Selbst-Heilung, etc. auf und können sich damit u. a. an Umwelteinflüsse dynamisch anpassen. Als Vorbild dieser Systeme dienen Strukturen und Methoden biologischer Systeme.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Eingebettete Systeme mit Selbst-X-Eigenschaften zu analysieren und zu entwerfen,
- emergente Eigenschaften solcher Systeme darzustellen und zu analysieren,
- Konzepte auf einfache Systeme zur Erzeugung robuster Systeme zu übertragen.

### ***Inhalte***

- Grundideen des Organic Computing
- Selbstorganisation und Emergenz
  - Definition und Grundlagen der Selbstorganisation
  - Adaptive und selbstorganisierende Systeme
  - Selbst-X-Eigenschaften
  - Definition von Emergenz
  - Negative und positive Emergenz
- Architekturen und Entwurf
  - Observer/Controller-Architekturen
  - Learning Classifier Systems
  - Evolutionäre und Genetische Algorithmen
  - Organic Computing und Middleware
  - Multi-Agenten-Systeme und Auktionen
  - Künstliche Hormonsysteme (AHS)
- Anwendungen
  - Robuste Systeme

## **Physik-/Bio-/IT-Interaktion**

Informationsverarbeitung wird zunehmend in die reale Welt mit ihren physikalischen, chemischen und biologischen Aspekten integriert. Dadurch entstehen Wechselwirkungen, die über die Grenzen der einzelnen Disziplinen hinausgehen. Gerade durch die Überwindung der Grenzen der einzelnen Disziplinen entstehen sehr spannende Forschungsaufgaben, deren Lösungen umfangreiche Beiträge zum Verständnis unserer Umwelt liefern können.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- auf hoher Ebene grob naturwissenschaftliche Zusammenhänge zu überblicken, diese einzuordnen und deren Grenzen zu erkennen,
- im Team mit Spezialisten aus anderen Fächern eine Integration der Informationsverarbeitung mit der realen Welt zu planen, zu realisieren, einzusetzen und die Ergebnisse auszuwerten,
- sich in benötigtem Maße in naturwissenschaftliche Gebiete einzuarbeiten.

### ***Inhalte***

- Einführung
    - Grundkonzepte der Physik, Chemie und Biologie (überblicksmäßig)
  - Sensoren (exemplarisch)
    - Physikalische Sensoren
    - Biophysikalische Sensoren
    - Biochemische Sensoren
  - Aktuatoren
  - Informationsverarbeitung und Raum
    - Koordinatensysteme, Approximation im Rechner
  - Informationsverarbeitung und Zeit
    - Realzeitsysteme, Approximation von Zeit im Rechner
    - Maximale Ausführungszeiten
  - Vernetzte Eingebettete Systeme
    - Internet of things
    - Logistische Systeme
    - Moderne Energieverteilungssysteme
    - IT-basierte Hilfsmittel in der Medizin
  - Akzeptanz-, Zuverlässigkeits- und Sicherheitsfragen
- Die Behandlung dieser Themen erfordert zusätzliche Kenntnisse außerhalb der Informatik, zum Beispiel in der Ingenieurmathematik und Statistik.

## **Nano-Elektronik**

Die weiteren Fortschritte bei der Verwendung von traditioneller CMOS-Schaltungstechnik sind überschaubar. Alternativ werden neuartige elektronische Bauelemente in Nanotechnologie erforscht. Ihre Funktionsweise, Herstellungsverfahren sowie elektronische Grundschaltungen sind Gegenstand dieses Themengebiets.

## ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die quantenmechanischen Grundsätze von Nano-Bauelementen zu erläutern,
- die Funktionsprinzipien der wichtigsten Transistor-Bauelemente in Nano-Strukturen zu beschreiben,
- elementare Grundsaltungen abzuleiten,
- die statistische Schaltungstechnik zu erklären.

## ***Inhalte***

- Nanotechnologie
- Boltzmann-Transportgleichung, Transportmechanismen (insbesondere Tunnel- und ballistischer Transport)
- Nanotransistoren: Molecular Switches, Spin FETs, Single Electron Transistoren, Nanotube FETs, Nanowire Transistoren
- Molecular Electronics
- Coulomb Blockade und Single-Electron Transistoren
- Spintronics
- Nanotubes & Nanowires
- Elektronische Grundsaltungen mit Nano-Devices
- Statistischer Schaltungsentwurf & Design Tools
- Alternative Realisierungen: chemische Berechnungen

## **B. Anwendungsorientierter Bereich**

### **Systemarchitekturen für Multimedia**

Die digitale Verarbeitung multimedialer Daten wie Sprache, Bilder oder Video stellt hohe Anforderungen an die Rechenleistung und Speicherkapazität der zugehörigen Rechnersysteme. Oft sind zudem Echtzeitbedingungen einzuhalten. Es sind daher spezielle, auf multimediale Verarbeitung zugeschnittene Systemarchitekturen erforderlich.

#### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Verfahren zu Medienverarbeitung und -übertragung zu erläutern,
- Bezüge zum Rechnersystem, insbesondere der Gestaltung multimediatebezogener Hardware-Architekturen herzustellen,
- Codierungen zur Audio- und Video-Übertragung und Speicherung, insbesondere deren Wechselwirkung mit spezialisierten Hardware-Architekturen, zu beschreiben,
- moderne Prozessorarchitekturen, z. B. Prozessoren für Digital-TV, Spielekonsolen zu diskutieren,
- skalierbare und zuverlässige Speichersysteme, z. B. für Multimedia-Server zu erörtern,
- QoS-Prinzipien, Anwendung von Kryptographie-Techniken im Kontext multimedialer Informationsverarbeitung zu beschreiben.

#### ***Inhalte***

- Multimediasysteme, Medien, Leistungsanforderungen an Rechner und Systeme
- Netz- und Rechnerarchitektur für Multimedia
- Massenspeichersysteme für multimediale Daten
- Verfahren für die Medienverarbeitung und -verteilung
- Betriebssystem-Aspekte
- Schutz und Authentizität multimedialer Daten

### **Ambiente Systeme**

Ambiente Systeme realisieren intelligente Lebens- und Arbeitsumgebungen unter Verwendung von ubiquitären (verfügbaren) und pervasiven (integrierten) Technologien. Ambient Computing erweitert dies um Aspekte der räumlichen Interaktion (Ambient Interfaces), der maschinellen Verarbeitung kontextsensitiver Informationen (Ambient Intelligence) und der Gestaltung raumbezogener Medien (Ambient Multimedia). Anwendungsgebiete umfassen z. B. technische Unterstützungssysteme (Ambient Assisted Living), räumliche Dienstleistungen (SmartHome, SmartBuilding, SmartCity), Entertainment (Ambient Gaming) und Edutainment (Ambient University).

#### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- den Stand der Entwicklung ambienter Systeme zu erörtern,
- ambiente Systeme zu konzipieren und zu modellieren,
- kleinere ambiente Anwendungen zu entwickeln.

## ***Inhalte***

- Technische Komponenten Ambienter Systeme
- Ambiente Interfaces
- Kontextsensitivität
- Nutzermodellierung und -management
- Ambiente Interaktion
- Ambient Intelligence
- Anwendungen
- Security/Privacy/Datenschutz/Ethik

## **Software Ubiquitärer Systeme**

*Ubiquitous Computing* ist einer der Begriffe, die moderne Anwendungen der Informationsverarbeitung kennzeichnen. Der Begriff bezieht sich auf die Möglichkeit, per *Computing* überall und jederzeit auf jede (nicht vertrauliche) Information zugreifen zu können. Schlüsselemente dazu sind mobile, vernetzte Geräte. Hierzu sind geeignete Kommunikationsstandards wie auch geeignete Geräte-Hard- und Software erforderlich.

## ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Standardverfahren der Entwicklung ubiquitärer Systeme zu erklären,
- Netzwerkprotokolle zu modellieren,
- Ressourcen zu modellieren,
- kleine vernetzte Anwendungen zu entwickeln.

## ***Inhalte***

- Grundlagen und Konstruktionsprinzipien der Software ubiquitärer Systeme
- Notwendige Systemkomponenten
  - Betriebssysteme
  - Middleware, einschl. Kommunikations-Software
  - Datenhaltungssysteme
  - Kontextabhängigkeit des Verhaltens
  - Varianten- und Aspekt-Management
- Ressourcenknappheit und Ressourcenmodellierung
  - Speicher
  - Rechenleistung
  - Energie
  - Netzwerkkonnektivität

## **Drahtlose Sensornetze**

Drahtlose Sensornetze bestehen aus einer Ansammlung hunderter oder tausender „Sensorknoten“, welche die weiträumige Beobachtung von Phänomenen der realen Welt erlauben, ohne dabei die physischen Prozesse wesentlich zu beeinflussen. Ein Sensorknoten ist dabei ein mit Sensoren, Prozessoren und drahtlosen Kommunikationseinheiten ausgestatteter Miniaturcomputer, der per Batterie gespeist über längere Zeiträume Sensordaten erfassen, verarbeiten und zu anderen Knoten bzw. zu einer Basisstation übertragen kann.



### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die Besonderheiten von Sensornetzen sowie der damit verbundenen Herausforderungen und Konzepte darzustellen,
- den Aufbau und Entwurf zur Analyse von Protokollen für Sensornetzwerke methodisch zu beherrschen,
- die aktuellen Forschungsaktivitäten zu Sensornetzen zu deuten und weiterzuverfolgen.

### ***Inhalte***

- Grundlagen der Sensornetze
- Architektur der Sensorknoten und Sensornetze
- Bitübertragung
- Sicherungsschicht, Vermittlungs- und Transportschicht
- Identität und Adressierung
- Zeitsynchronisation, Lokalisation
- Datenhaltung und Topologiekontrolle
- Anwendungen

## **Mobile Roboter**

Mobile Roboter können sich im Gegensatz zu stationären Industrierobotern in ihrer Umgebung frei bewegen, um ihre Aufgaben zu erfüllen. Wesentliche Herausforderungen sind dabei neben der Sensorik und Aktorik die Selbstlokalisierung und die Navigation. Beispiele sind Serviceroboter wie z. B. Putz- oder Rasenmäherroboter, autonome Transportroboter und Roboterfahrzeuge sowie Unterwasser- und Flugroboter. Weiterhin zählen biologisch inspirierte Roboter, wie Kletter- und Laufroboter, humanoide Roboter und Schwarmroboter zu den mobilen Robotern.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die Sensorik und Aktorik mobiler Roboter darzustellen und zu erklären,
- die Kinematik von Roboterantrieben abzuleiten,
- die Grundlagen der Mensch-Roboter-Interaktion exemplarisch darzustellen,
- Konzepte für Multi-Robotersysteme (Teams, Schwärme etc.) zu erörtern,
- grundlegende KI-Paradigmen für mobile Roboter (reaktiv, deliberativ, hybrid) zu beschreiben und zu vergleichen,
- Verfahren zur Missionsplanung einzusetzen,
- Methoden des Roboterlernens zu beurteilen,
- Verfahren zur Navigation, insbesondere Selbstlokalisierung und Wegplanung sowie SLAM-Ansätze kritisch zu vergleichen, zu evaluieren und praktisch einzusetzen,
- einfache mobile Roboter selbst zu konzipieren und zu programmieren.

### ***Inhalte***

- KI-Paradigmen für mobile Roboter
- Sensorik
- Aktorik und Kinematik
- Navigation (Lokalisation, Wegplanung, SLAM)
- Roboter-Lernen

- Multi-Robotersysteme
- Mensch-Roboter-Interaktion
- Beispielroboter und aktuelle Trends

## **Medizinische Roboter**

Roboter werden zunehmend auch zur Unterstützung der Ärztin oder des Arztes in der Medizin eingesetzt. Beispiele sind Operationsroboter in der Chirurgie oder Bestrahlungsroboter in der Nuklearmedizin. Häufig sind diese Roboter mit entsprechenden Planungs- und Navigationssystemen auf der Basis bildgebender Verfahren gekoppelt.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die Zusammenhänge von Robotik, Bildverarbeitung, Navigation in der Medizin darzustellen,
- Verfahren zur mathematischen Beschreibung von Robotern zu beherrschen,
- die Grundlagen der medizinischen Bildverarbeitung zu erörtern,
- die Zusammenhänge der Navigation und Robotik zu beschreiben.

### ***Inhalte***

- Kinematik, Bewegungsplanung von Robotersystemen
- Roboterprogrammierung
- Medizinische Navigation
- Sensorik in medizinischen Anwendungen
- Medizinische Bildverarbeitung
- Operationsplanung

## **(Mikro-)Sensoren und Aktoren**

Mikrosysteme interagieren durch Sensoren und Aktoren mit ihrer Umgebung. Sensoren erfassen Daten, welche anschließend analog und/oder digital verarbeitet werden. Die Verarbeitungsergebnisse werden von Aktoren umgesetzt. Beispiele für Mikrosysteme mit Sensoren und Aktoren sind z. B. Inertialsysteme zur Navigation, welche Mikrogyroscopie enthalten, die zur Bestimmung von Drehraten eingesetzt werden, oder Airbagsysteme in der Automobiltechnik, welche mikromechanische Beschleunigungssensoren enthalten.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die Herstellungsprozesse für Mikrosysteme grundlegend zu erörtern,
- grundlegende Sensorkonzepte zu beschreiben,
- grundlegende Aktorkonzepte darzustellen,
- die emergenten Eigenschaften solcher Systeme zu analysieren.

### ***Inhalte***

- Überblick Anwendungsgebiete von Mikrosensoren und -aktoren
- Werkstoffe und Herstellung
  - Sensor- und Aktorwerkstoffe

- Fertigungsverfahren (Galvanotechnik, LIGA, Maskentechnik, verschiedene Prozessschritte, Mikromontage)
- Grundlegende Wandlungsprinzipien von Sensoren und Aktoren
  - Piezoresistive Wandler
  - Kapazitive Wandler
  - Piezoelektrische Wandler
  - Elektromagnetische Wandler
  - Elektrodynamische Wandler
  - Thermo-elektrische Wandler
  - Seebeck-, Peltier-Elemente
- Optische Grundelemente
  - Lichtwellenleiter
  - Refraktive/diffraktive Optiken
- Mikrofluidik
  - Mikrokanäle
  - Aktive/passive Mikroventile
  - Mikropumpen
  - Mikroreaktionskomponenten

## **Informationsfusion**

Informationsfusion beschreibt die Methodik, Daten aus verschiedenen Informationsquellen, z. B. Sensoren, Datenbanken aber auch Humaninformation mit dem Ziel der Informationsergänzung zu kombinieren. Oftmals ist es nicht möglich, Information robust und präzise durch nur eine Informationsquelle zu erfassen. In diesen Fällen erlaubt die Fusion von Informationen aus heterogenen Quellen eine ganzheitliche Erfassung und Betrachtungsweise einer Aufgabenstellung. Hierzu müssen Daten in eine Abstraktionsebene transformiert werden, die eine Fusion heterogener Quellen erlaubt. Anschließend findet die Verknüpfung auf einer Merkmals- oder Szenenebene statt. Informationsfusion findet eine breite Anwendung in den Bereichen Robotik, Kognitive Systeme, Expertensysteme, intelligente technische Systeme im Maschinen- und Anlagenbau, Smart Grids, Medizintechnik usw.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- unterschiedliche Sensoren zu beschreiben und zu charakterisieren sowie deren typische Verwendung zu erklären,
- grundlegende Prinzipien der Sensorsignal-Repräsentation zu erläutern und darzustellen,
- die Grundlagen der Informationsfusion und Fusionsprinzipien zu erläutern und zu diskutieren,
- verschiedene Fusionsprinzipien und deren Vor- und Nachteile zu analysieren und zu bewerten,
- zu entscheiden, wann verschiedene Fusionsprinzipien eingesetzt werden und wie diese in heterogenen Informationsfusionssystemen genutzt werden, um ein spezielles Problem zu lösen.

### ***Inhalte***

- Sensorische Prinzipien
- Elementare messtechnische Grundlagen
- Repräsentanz von Wissen und Unsicherheit

- Fusionskonzepte, basierend auf
  - Wahrscheinlichkeitstheorie inkl. Bayes-Netze
  - Evidenztheorie (Dempster-Shafer-Theorie)
  - Fuzzy-Mengen-Theorie
  - Möglichkeitentheorie (Possibility Theorie)
  - Neuronale Netze
- Datenregistrierung und Synchronisation
- Exemplarische Darstellung verschiedener Systeme

## **Hochleistungsrechner**

Die rechnergestützte Simulation bildet heute neben der Theorie und dem Experiment die dritte Säule der Wissenschaft. Die Entwicklung von Rechnern mit hoher Rechenleistung insbesondere für Gleitkommaarithmetik ermöglicht die Berechnung komplexer mathematischer Modelle für Anwendungen in den Naturwissenschaften, der Medizin und Pharmakologie und in den Ingenieurwissenschaften. Neben den Hochleistungsrechnern, bei denen vornehmlich die Rechenleistung bezüglich eines Programms eine Rolle spielt (Latenz), werden Hochleistungsrechner auch im kommerziellen Bereich mit Anwendungen aus dem Datenbankbereich oder der Transaktionsverarbeitung bzw. für Suchmaschinen eingesetzt. Kennzeichnend ist für die Klasse von Hochleistungsrechnern ein hoher Durchsatz von Programmen. Neben der Rechenleistung werden an Hochleistungsrechner hohe Anforderungen bezüglich der Energieeffizienz und der Zuverlässigkeit gestellt.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Architekturmerkmale von Hochleistungsrechnern zu beschreiben,
- Hochleistungsrechner zu bewerten und bezüglich einer Anwendung auszuwählen,
- Hochleistungsrechner zu programmieren,
- die Verwaltung von Hochleistungsrechnern zu erörtern.

### ***Inhalte***

- Einsatzgebiete und Anwendungsbereiche für Hochleistungsrechner
- Bewertung von Hochleistungsrechnern
  - Rechenleistung
  - Energieeffizienz
  - Zuverlässigkeit
- Grundlagen paralleler Architekturen
  - Multiprozessoren mit verteiltem Speicher
  - Multiprozessoren mit gemeinsamem Speicher
  - Cluster-Systeme
  - Vektorrechner
  - Multi-/Many-core-Architekturen
  - Verbindungsnetze
  - Speicherarchitekturen
- Grundlagen paralleler Programmierung
  - Parallele Programmiermodelle
  - MPI, OpenMP
- Compiler und Werkzeuge für Hochleistungssysteme
  - Parallelisierende Compiler

- Werkzeuge zur Leistungsanalyse und Leistungsverbesserung
- Verwaltung von Hochleistungsrechnern
- Fallstudien
  - Höchstleistungsrechner aus der TOP-500-Liste
  - Mainframe-Rechner

## **Nichtkonventionelles Rechnen (Quanten-Computing, DNA-Computing etc.)**

Unter Nicht- oder auch Unkonventionellem Rechnen versteht man den Einsatz von alternativen und innovativen Methoden und Technologien für den Bau neuer Rechner, die sich stark von der Architektur des von-Neumann-Rechners und aktuellen Rechnerarchitekturen unterscheiden. Zum Unkonventionellen Rechnen zählt man u. a. DNA-Computing (Rechnen mit DNA-Sequenzen), Optisches Rechnen (Rechnen mit Photonen statt mit Elektronen), Quantencomputing (Rechnen mit Qubits oder mit dem Spin eines Elektrons), Nanocomputing (Rechnen mit Bauelementen aus der Nanotechnik) oder Reversible Logiken. Jede dieser Richtungen besitzt unterschiedliche Funktionsweisen und derzeit auch unterschiedliche Realisierungschancen.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- verschiedene neue technologische Entwicklungen im Hinblick auf deren Einfluss auf zukünftige Rechner- und auch Netzwerkarchitekturen richtig einzuschätzen,
- durch Anwendung der Methoden alternativer Berechnungsverfahren und des Einsatzes der Technologien des nichtkonventionellen Rechnens, zukünftige Rechensysteme zu planen, zu entwerfen und deren Funktionsweise zu beschreiben.

### ***Inhalte***

- Quantencomputing und Quantenkryptographie
  - Qubits, Qubit-Register, Superposition, Verschränkung, Quanten-Algorithmus von Deutsch
  - Herstellen von Superpositionszuständen (Ionenfallen, Kernspin-Techniken)
  - Quantenverschlüsselung, BB84-Algorithmus, EPR-Algorithmus
- DNA-Computing
  - Grundprinzipien des DNA-Computings (Hybridisierung von Basen)
  - Adleman-Algorithmus zur Lösung des Travelling-Salesman-Problems mit DNA
  - DNA-gesteuerte Mikroroboter im Körper
- Memristor-Technologie
  - Speichern mit Memristoren
  - Rechnen mit Memristoren (Neuronale Netze mit Memristoren, Memristor-3-Phasen-Digitallogik, Einsatz von Memristoren in Mikroprozessoren)
  - Hybride Systeme (Kopplung von CMOS-Logik mit rekonfigurierbaren Verbindungen auf Memristor-Basis)
- Optische Informationsverarbeitung und Photonik in der IT
  - Frühere Idee des Optischen Computers
  - Optische Network-on-Chips, Silicon Photonics
  - Wellenlängen-Multiplex-Netzwerke
- Nanocomputing
  - Funktionsweise und Aufbau von Quantum-Cellular-Automaten (QCA)
  - Arithmetische Schaltkreise mit QCA

- Alternative Berechnungsmethoden
  - Mehrwertige Logik, Redundante Zahlensysteme
  - Reversible Logik
  - Umsetzung dieser Berechnungsmethoden mit oben genannten Technologien

## **Energieinformatik**

Die Energieinformatik greift die speziellen Anforderungen an Methoden der Informatik sowie Steuerungs-, Informations- und Kommunikationstechnologien in vernetzten Energiesystemen auf, die zunehmend durch fluktuierende und dezentrale Energieeinspeisung geprägt sind. Während im klassischen Energiesystem Strom aus großen Kraftwerken zentral im Höchstspannungsnetz eingespeist und dann über das Mittelspannungsnetz zum Niederspannungsnetz geleitet und verteilt wird, erfolgt die Erzeugung zukünftig zu einem großen Teil dezentral im Verteilnetz (also Niederspannungsnetz). Das Prinzip der verbrauchsgesteuerten Erzeugung muss aufgrund der volatilen und kaum steuerbaren Stromeinspeisung aus Photovoltaik und Wind ersetzt werden durch das neue Prinzip des erzeugungsgesteuerten Verbrauchs. Dies erfordert den Aufbau eines Energieinformationsnetzes mit verteilter Systemintelligenz, das an den richtigen Orten im Netz Informationen über Erzeugung und Verbrauch sammelt und zeitgerecht dort zur Verfügung stellt, wo sie verarbeitet werden als Grundlage für Steuerungsentscheidungen der unterschiedlichen Akteure im Netz. Die Energieinformatik muss geeignete Architekturen und Methoden bereitstellen, um gemäß den vielfältigen Anforderungen des Energiesystems eine verlässliche Versorgung mit elektrischer Energie gewährleisten zu können.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die Anforderungen an die Energieversorgung angemessen zu beurteilen,
- den Bedarf an Informationsverarbeitung und Kommunikation in Energienetzen sachgerecht zu beurteilen,
- Maßnahmen zur Effizienzsteigerung des Energieeinsatzes (auch übergreifend über Strom-, Gas- und Wärmenetze) zu planen, durchzuführen und zu überprüfen,
- zu erörtern, wie Energieflexibilität zu erschließen und bestmöglich auszuschöpfen ist, um eine dynamische Anpassung an veränderliche Lastprofile zu erreichen.

### ***Inhalte***

- Begriff der Energie
- Formen der Energie, Zusammenhang zwischen Leistung und Energie
- Energieerhaltung
- Erzeugung elektrischer Energie (klassische, erneuerbare Energiequellen; einschl. energy harvesting)
- Prinzipien der Energieverteilung
- Stabilität des Energiesystems, Regelenergie, Bilanzkreise
- Modellierung und Simulation von Energiesystemen, Lastschätzung
- Rollen im deregulierten Energiesystem, wesentliche Anforderungen der Regulierung
- Energiespeicherung/-wandlung
- Virtualisierung (virtuelle Kraftwerke und virtuelle Speicher)
- Energieinformationssysteme mit verteilter Systemintelligenz (Multiagentensysteme, Organic Computing, Autonomic Computing)
- Intelligente Zähler, Sicherheit und Schutz der Privatsphäre

- Verlässlichkeit des Energiesystems
- Anreizverfahren und Marktmechanismen im Energiemanagement (Märkte EEX, Merit Order, ...)
- Nutzungsschnittstellen und Nutzerverhalten
- Zertifizierung von Energiesystemen

## **Verkehrsinformatik**

Bei wasser-, straßen-, schienen- und luftgestütztem Verkehr wird zunehmend die Informationsverarbeitung eingesetzt, um die Sicherheit zu erhöhen, Energie einzusparen, die Kosten insgesamt zu senken. Dabei handelt es sich alles in allem um einen wichtigen Spezialfall von Cyber-Physical Systems. Beim Entwurf, der Optimierung und der Wartung von solchen Systemen sind fachübergreifende Kompetenzen erforderlich, die in ausreichender Breite und in ihrem Zusammenspiel eine entsprechende Hochschulausbildung nahe legen.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- gemischte informatische/verkehrstechnische Systeme zu entwickeln und zu analysieren,
- Verkehrsströme zu modellieren,
- Fahrzeugdynamik zu modellieren,
- sicherheitstechnische Standards zu erklären.

### ***Inhalte***

- Grundlagen der Fahrzeugphysik
- Grundlagen informationstechnischer Einrichtungen in Fahrzeugen
- Modellierung von Verkehrsflüssen
- Verkehrslenkung
- Sicherheitstechnik von Fahrzeugen und Verkehrswegen, Sicherheitsnachweise
- Kommunikation in und zwischen Fahrzeugen
- Fahrer-/Piloten-/Kapitäns-Assistenzsysteme
- Autonomes Fahren/Fliegen

## **Fertigungsautomatisierung**

Automatisierung von Fertigungsprozessen stellt ein Schlüsselement in allen Bereichen des Maschinen- und Anlagenbaus und der fertigenden Industrie dar. Eine wesentliche Rolle spielen dabei Steuerungskonzepte, die, von der Sensorik bis hin zur Leitebene, sämtliche Einsatzgebiete der Automation abbilden. Strategien und Konzepte dezentraler Steuerungen, Plug-and-Play-Fähigkeit, Wandlungsfähigkeit von Anlagen und Mechanismen der Selbstadaption werden zukünftige Fertigungssysteme in ihrer Funktionalität entscheidend erweitern.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- Grundbegriffe, Konzepte und Strategien der Fertigungstechnik zu beschreiben und darzustellen,

- wirtschaftliche Einordnung verschiedener Fertigungssysteme darzustellen und zu diskutieren,
- Methoden der Fertigungsoptimierung und -automation zu erläutern,
- Grundlagen der Optimierungstheorie darzustellen,
- Steuerungs- und Leittechnik sowie speicherprogrammierbare Steuerungen (SPS) zu analysieren und zu bewerten,
- zu entscheiden, wann IT-basierte dezentrale Kommunikationssysteme in der Fertigungstechnik vorteilhaft eingesetzt werden können.

### ***Inhalte***

- Fertigungstechnik
  - Einführung
  - Grundbegriffe
  - Fertigungssysteme, Struktur und Aufbau
  - Wirtschaftlichkeit und Kostenoptimierung
- Fertigungsoptimierung
  - Prinzipien der Automation
  - Mathematische Optimierungstheorie
  - Self-X-Strategien
- IT-basierte Grundgebiete
  - Speicherprogrammierbare Steuerungen
    - Einführung, Aufbau, Funktionsweise
  - Programmierung DIN EN 61 131.3
  - Verteilte Netzwerke und Systeme

## **Computational Intelligence**

Unter Computational Intelligence (CI) ist die rechnergestützte Informationsverarbeitung in Anlehnung an biologische, biochemische und neurowissenschaftliche Prozesse und Mechanismen zu verstehen. Diese werden unter Zuhilfenahme mathematisch-ingenieurwissenschaftlicher Konzepte auf technische Fragestellungen adaptiert. Dabei stehen Verfahren im Vordergrund, die die menschliche Fähigkeit nachahmen, Wissen selbst unter vagen und unvollständigen Bedingungen zu erlangen und zu interpretieren. CI-Methoden spielen in der Bildverarbeitung, Sprachverarbeitung, Data Mining, Mustererkennung, Selbst-Optimierung usw. eine bedeutende Rolle.

### ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- grundlegende Prinzipien der Wissensrepräsentation zu erklären und zu charakterisieren,
- grundlegende Ansätze der Beschreibung von Unsicherheiten darzustellen und zu erörtern,
- Zusammenhänge zwischen Wissensmodellierung und Unsicherheit darzustellen und zu diskutieren,
- Grundlagen verschiedener Modellierungskonzepte der CI zu erläutern und zu bewerten,
- verschiedene CI-Prinzipien und deren Vor- und Nachteile zu analysieren und zu bewerten,
- zu entscheiden, welche CI-Ansätze für konkrete Aufgabenstellungen geeignet sind und welche Methoden sinnvoll einsetzbar sind.



## ***Inhalte***

- Repräsentanz von Wissen und Unsicherheit
- Fuzzy-Systeme
- Neuronale Netze
- Evolutionäre Algorithmen
- Schwarmintelligenz
- Künstliche Immunsysteme
- Exemplarische Darstellung verschiedener Systeme

## **Cloud Computing**

Das Cloud Computing ist ein neuartiges Konzept für ein Verteiltes System, das das Potential hat, den Charakter komplexer IT-Infrastrukturen fundamental zu verändern. Ein Verteiltes System ist dabei ein Zusammenschluss mehrerer unabhängiger Rechner, die über ein Netzwerk kommunizieren. In der Informatik existieren verschiedene Architekturen für so ein Verteiltes System, beispielsweise das klassische Client-Server-Modell, Cluster- und Grid-Computing, Peer-to-Peer-Netze (P2P) und Web-Computing oder Utility-Computing. Beim Cloud Computing werden Rechen- und Speicherressourcen, Netzwerkkapazitäten und Softwaredienste in großen IT-Zentren (Cloud-Providern) konsolidiert. Diese Ressourcen lassen sich dynamisch an wechselnde Nutzungsbedingungen anpassen und werden den NutzerInnen (Cloud-User) in unterschiedlichen Diensten (Cloud-Services) zur Verfügung gestellt. Die NutzerInnen haben dabei große Freiheiten bezüglich der konkreten Wahl der physikalischen Infrastruktur (*Infrastructure as a Service, IaaS*), des Betriebssystems, der Datenbanken und der Programmiersprache (*Platform as a Service, PaaS*) oder der Software (*Software as a Service, SaaS*), die sie verwenden möchten.

## ***Lernziele/Kompetenzen***

Die Studierenden sind in der Lage,

- die Konzepte des Cloud Computing zu erörtern sowie seine Anwendungen zu beurteilen,
- die Basistechnologien des Cloud Computing zu beschreiben und zu charakterisieren,
- aktuelle technologische Entwicklungen zu erklären,
- zu entscheiden, ob und wann Cloud Computing sinnvoll eingesetzt werden kann und welche Cloud-Technologie sich zur Lösung eines speziellen Problems eignet.

## ***Inhalte***

- Evolution des Cloud Computing
  - Grundlagen verteilter Rechnersysteme
  - Entwicklung vom Client-Server-Modell über Cluster-Computing, Grid-Computing, P2P, Utility-Computing bis hin zum Cloud-Computing
- Basistechnologien für das Cloud Computing
  - Virtualisierungskonzepte
    - Anwendungsvirtualisierung (z. B. JVM)
    - Vollständige Virtualisierung (z. B. VMM, VMware/VirtualBox/KVM)
    - Paravirtualisierung (z. B. Xen oder VMware ESX)
    - Hardware-, Betriebssystem-, Netzwerk-, Speicher-Virtualisierung
- Web-Sprachen (z. B. Java Script, XML, AJAX, GWT – Google Web Toolkit)
- Web-Services (z. B. RPC, SOAP mit WSDL, REST, JSON)
- Effiziente verteilte Daten-Speicherung und -Verarbeitung (z. B. MapReduce, Hadoop (HDFS))

- Funktionale Kategorien von Cloud-Systemen (und entsprechende Beispiele)
  - Software as a Service (z. B. Google Docs, Adobe Photoshop Express, Apple iWork.com)
  - Platform as a Service (z. B. Google App Engine, AppScale, typhoonAE)
  - Infrastructure as a Service (z. B. Amazon Elastic Compute Cloud (EC2), Eucalyptus)
- Fehlertoleranz, Monitoring, Sicherheit und „Privacy“ beim Cloud Computing
  - Cloud-Federation
  - Risiko-Management und Disaster Recovery
  - Service Level Agreements (SLA) und Quality of Service (QoS)
  - Daten-Sicherheit, Daten-Compliance und Digital Rights Management (DRM)
  - Portal-Technologien, Identitäten, Identity Federation