Master of Science

Computer Engineering (Technische Informatik)

MScTI

Description of the course modules

Heidelberg University
Department of Physics and Astronomy

Version 1.14 (V1)

Name of university	Heidelberg University
Name of department	Department of Physics and Astronomy
Name of degree course	Master of Science Computer Engineering
Format of studies	Full time, part time
Type of degree course	Consecutive
Date of version	15.01.2020
Prescribed period of study	Two years, i. e. four semesters
Location	Heidelberg
Total number of credit points	120
University places	Unlimited
Target group	Holders of Bachelor of Science, Magister, Staatsexamen, Diploma or equivalent final degree of at least 6 semester study having majored in computer science or in Mathematics, Natural Sciences or Engineering with attestable lectures (recommended 24 CP) imparting knowledge in computer science.

Preamble

Quality Objectives at Heidelberg University for Study Programmes and Teaching

Having regard to its mission statement and constitution, Heidelberg University's degree courses have subject-related, interdisciplinary and occupational objectives. They aim to provide a comprehensive academic education equipping graduates for the world of work.

Consequently, the following competence profile has been drawn up for inclusion in module handbooks as a profile of skills valid for all disciplines. It shall be applied to the specific objectives of the individual courses and then implemented in their curricula and modules.

The main points of the competence profile:

- developing subject-related skills with a pronounced research orientation
- developing the ability to engage in trans disciplinary dialogue
- developing practice-related problem-solving skills
- developing personal and social skills
- promoting the willingness to assume social responsibility on the basis of the skills acquired

Disciplinary and Interdisciplinary Quality Objectives of the Master Programme "Computer Engineering"

The research oriented master program in Computer Engineering (MScTI) at Heidelberg University is organized by the Institute of Computer Engineering and the Department of Physics and Astronomy. Its educational objective is to qualify students for a research or development oriented professional career in the field of Computer Engineering as well as for participation in PhD programs. The students should get a thorough understanding of possible approaches and solutions and should be able to assess their advantages and drawbacks so that they can chose a good solution for a given problem. They should realize when a solution is inappropriate or suboptimal and should be able to devise novel approaches / solutions. The MScTI emphasizes on practical skills so that the students can use the available tools and methods (software, mathematics) to develop working solutions efficiently.

Students can choose one of three specializations, which are sub-fields of 'computer engineering' ('Application Specific Computing', 'Microelectronics', 'Robotics, Haptics and Biomechanics'). Each specialization consists of a set of modules on an advanced level, which cover the field to a large extent. By following a sufficient number of modules in such a specialization, students reach the state-of-the art in the area to become fully competitive.

After having completed the research phase (seminar, student research project, master thesis), students have obtained the ability to do research independently, to document and publish research work. They deepen their knowledge on scientific methods, information engineering, hardware and software, interdisciplinary system thinking, experience in practical applications as well as the communication competence and the ability to work in teams.

Possible career options are in the practical development of hardware systems for data acquisition and for fast data processing, the efficient solution of compute-intensive tasks on modern, high performance, heterogeneous hardware, the design of analogue or digital microelectronics circuits, the design and operation of robotic systems, etc.

Subject-related Qualification Objectives

After completing the master program ‘Computer Engineering’ the graduates are able to program parallel systems with shared and distributed memory and use the learned structures to develop new architectures of parallel computers. They acquire a deep understanding of nonlinear dynamical systems in order to design simple nonlinear control systems. Additionally, they understand concepts about the functionality and programming of microprocessors and peripheral circuits as well as reconfigurable architectures and use these concepts to implement an own circuit respectively to implement and to program a sample embedded FPGA platform.

Transdisciplinary Qualification Objectives

Master’s graduates in Computer Engineering possess the required skills to work independently with a variety of software tools for various special applications and to choose the appropriate one to solve problems. They are able to apply structured working methods and can organize complex professional projects. Also they have a basic understanding about legal aspects of founding and running a company, financial aspects of founding and running a business and are able to apply marketing strategies and tools.

Overview of the course modules

Module	Module Coordinator	ECTS
Fundamentals		
▪ Parallel Computer Architecture (compulsory)	Brüning	6
▪ Control Systems Design (compulsory, “System Design” in PO)	Masia	6
▪ C++ Practice	Strzodka	6
▪ Electronics	Wurz	6
▪ Introduction to High Performance Computing	Fröning	6
▪ Microcontroller Based Embedded Systems	Wurz	6
▪ Reconfigurable Embedded Systems	Kugel	6
Soft Skills		
▪ Tools	all	4
▪ Entrepreneurship	extern	6
Main Subject / Specialization		
▪ Components, Basic Circuits & Simulation	Fischer	6
▪ Full Custom VLSI Design	Fischer	6
▪ Digital Hardware Design	Brüning	6
▪ Digital Semi Custom Design Flow	Brüning	6
▪ Functional Verification	Brüning	6
▪ Advanced Analogue Building Blocks	Fischer	6
▪ Silicon Sensors & Readout Electronics	Fischer	6
▪ GPU Computing	Fröning	6
▪ Accelerator Practice	Strzodka	6
▪ Parallel Algorithm Design	Strzodka	6
▪ Advanced Parallel Algorithms	Strzodka	6
▪ Advanced Parallel Computing	Fröning	6
▪ FPGA Coprocessors	Kugel	6
▪ High Performance Interconnection Networks	Brüning	6
▪ Parallel Algorithms, Application Computing	Bastian	8
▪ Parallel Solution of Large Linear Systems	Bastian	8
▪ Robotics 1 - Kinematics, Dynamics and Control	Masia	6
▪ Robotics 2 - Simulation and Optimization in Robotics	Masia	6
▪ Biomechanics and Biorobotics	Masia	6
▪ Haptics and Human Robot Interaction / Rehabilitation	Masia	6
▪ Robotics Practical	Masia	6
▪ Robotic Games	Masia	6

Structure of Courses

All in all, the following modules have to be completed successfully (120 CP):

▪ 3 modules from „Fundamentals“	18 CP
▪ 5 modules from „Main Subject / Specialization“	30 CP
▪ 2 modules from „Free Courses”	12 CP
▪ 2 or more modules from „Soft Skills“	12 CP
▪ Seminar	4 CP
▪ Student research project	14 CP
▪ Master thesis with final colloquium	30 CP

Fundamentals

The following modules from “Fundamentals” are compulsory and have to be completed (mandatory modules):

- Parallel Computer Architecture
- Control Systems Design

As an elective you can choose any other subject listed above in “Fundamentals”.

Main Subject / Specialization

In principle, advanced students can choose their in-depth modules freely according to the examination rules / regulations. We recommend, however, following one of the model curricula completing an exceptional specialization in a certain field of Computer Engineering. When completing a sufficient number of modules in such a specialization during your studies, this specialization will be documented explicitly in your Master Grade Report.

This is a list of model curricula and the modules required in each case:

1. Microelectronics

3 compulsory modules:	- Components, Basic Circuits & Simulation
	- Digital Hardware Design
	- Full Custom VLSI Design
2 elective modules from:	- Advanced Analog Building Blocks
	- Digital Semicustom Design Flow
	- Electronics
	- Functional Verification
	- Microcontroller Based Embedded Systems
	- Reconfigurable Embedded Systems
	- Silicon Sensors & Readout Electronics

2. Application Specific Computing

3 compulsory modules:	- GPU Computing
	- Parallel Algorithm Design
	- Reconfigurable Embedded Systems
2 elective modules from:	- Accelerator Practice
	- Advanced Parallel Algorithms
	- Advanced Parallel Computing
	- C++ Practice
	- Electronics
	- FPGA Coprocessors
	- High Performance Interconnection Networks
	- Introduction to High Performance Computing
	- Microcontroller Based Embedded Systems
	- Parallel Algorithms, Application Computing
	- Parallel Solution of Large Linear Systems

3. Robotics, Haptics and Biomechanics

3 compulsory modules:	- Robotics 1 - Kinematics, Dynamics and Control
	- Biomechanics and Biorobotics
	- Haptics and Human Robot Interaction / Rehabilitation
2 elective modules from:	- Robotics 2 - Simulation and Optimization in Robotics
	- Robotics Practical
	- Robotic Games
	- GPU Programming
	- Microcontroller Based Embedded Systems
	- Reconfigurable Embedded Systems

Free Courses

The aim of the Free Course is to broaden ones expertise. The lectures can be chosen from the course catalogue of the Heidelberg University. To be approved as a Free Course in the MScTI, the lecture must meet the following three conditions:

- it is graded,
- the sum of credit points awarded is 12 CP or more,
- it reasonably contributes to a broadening of the expertise for this study program.

In case of doubt, the Dean of Studies makes the decision on the approval.

It is also possible to select Fundamentals or Specializations modules from the MScTI for the Free Course. When the decision on a preferred specialization is not yet made in the first semester, it is a good choice to take a second specialization as the Free Course.

Soft Skills

All in all, 12 CP must be completed in the field of soft skills, 2 of which are integrated in the seminar. For the remaining 10 CP the following courses can be chosen:

- Tools (4 CP),
- Entrepreneurship (6 CP),
- Courses from the University course program classified as soft skill courses,
- Language Courses (6 CP maximum).

Seminar

For the seminar 2 CP are allocated as soft skills in addition to 4 CP for the professional contents.

Seminar: 4 CP (professional contents) + 2 CP (soft skills)

Overview of the course modules

Fundamentals:

<table>
<tr><td colspan="2">Code:
MScTI_PCA</td><td colspan="3">Course Title:
Parallel Computer Architecture</td></tr>
<tr><td colspan="2">Lecturer:
Prof. Dr. U. Brüning</td><td colspan="3">Type:
Lecture with exercises / lab / …</td></tr>
<tr><td>Credit Points:
6</td><td>Workload:
180h</td><td>Teaching Hours:
4 / week</td><td>Language:
German / English</td><td>Term:
WS</td></tr>
<tr><td colspan="5">Module Parts and Teaching Methods:
• Lecture (2 hours/week)
• Exercise / Lab: practical programming exercises on parallel computer system (2 hours/week)</td></tr>
<tr><td colspan="5">Objectives: The students
• understand the concepts and principles of parallel processing and the underlying hardware structures, so that they are able to program parallel systems with shared and distributed memory,
• use the learned structures to develop new architectures of parallel computers.</td></tr>
<tr><td colspan="5">Contents:
• Concepts of Parallel Processing
• SIMD-Architectures
• MIMD-Architectures
• Shared Memory
• Distributed Memory
• Communication and Synchronization
• Multithreading
• Taxonomy of Interconnection Networks
• Point-to-Point INs
• Switched INs, Shuffles, Crossbars, Routing, Latency
• Communication Protocols
• Virtual Shared Memory
• Dataflow Architectures</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="3">Recommended Knowledge:
basic knowledge of Computer Architecture</td></tr>
<tr><td colspan="5">Literature: A reading list will be provided in the script.
The script will be accessible on the web site of the Computer Architecture Chair.</td></tr>
<tr><td colspan="5">Form of Testing and Examination: 30’ oral exam at the end of the semester
At least 50% of the exercises must be passed.</td></tr>
</table>

Code: MScTI_CSD		**Course Title:** Control Systems Design		
Lecturer: Prof. Dr. L. Masia		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** SS

Module Parts and Teaching Methods:

- Lecture: 2 hours/week
- Exercise with homework: 2 hours/week

Objectives: After completing this course the students will be able to:

- describe linear phenomena and linear dynamical systems,
- analyze linear systems by using state space representation, root locus and nyquist method,
- design linear control systems based on classical PID control scheme,
- apply the methods to simple practical examples in engineering and physics.

Contents:

- Introduction to feedback control
- Modeling in the frequency and time domain
- Time response of dynamic systems
- Reduction of multiple subsystems
- Stability analysis
- Steady-state errors
- Root locus techniques
- Controller design via root locus
- Frequency response techniques
- Design via frequency response
- Design via state space
- Introduction to digital control systems

Prerequisites: none	**Recommended Knowledge:** Theory of linear systems

Literature:

- K. Ogata: “Modern Control Engineering”
- Gene F. Franklin, J. David Powell, e al.: “Feedback Control of Dynamic Systems”
- W. Bolton: “Bolton: Mechatronics”
- Basilio Bona: “Dynamic modelling of mechatronic systems”

Form of Testing and Examination: Written exam at the end of the semester.

Successful participation in the programming exercises is required to be accepted to exam.

<table>
<tr><td colspan="2">Code:
MScTI_ADVCPP</td><td colspan="3">Course Title:
C++ Practice</td></tr>
<tr><td colspan="2">Lecturer:
Prof. Dr. R. Strzodka</td><td colspan="3">Type:
Lecture with exercises and project</td></tr>
<tr><td>Credit Points:
6</td><td>Workload:
180h</td><td>Teaching Hours:
4 / week</td><td>Language:
German / English</td><td>Term:
WS</td></tr>
<tr><td colspan="5">Module Parts and Teaching Methods:
• Lecture 2 hours/week
• Exercise 1 hour/week on average plus homework
• Project 1 hour/week on average plus homework</td></tr>
<tr><td colspan="5">Objectives: Students are able to:
• apply all major features of modern C++,
• design better programs following guidelines for an effective programming style,
• use and combine different programming patterns.</td></tr>
<tr><td colspan="5">Contents:
• A tour of modern C++ from start to end
• Practical use of new functionality since C++11, e.g. constexpr, move refs and ctors, initializer lists, lambdas, variadic templates, variants, threads, regex, tuples, bindings
• How to select among the language features
• Clear and effective programming style
• Intensive exercises with practical applications of discussed C++ feature

This is an advanced programming course, not suitable for C++ beginners! Without prior knowledge and experience with C++98 there will be too much new information at once.</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="3">Recommended Knowledge:
Understanding of all basic C++ concepts such as references, classes, inheritance, overloading, templates, STL
Multi-year programming experience</td></tr>
<tr><td colspan="5">Literature:
• Bjarne Stroustrup: “A Tour of C++”, Addison-Wesley, 2014
• Bjarne Stroustrup: “The C++ programming language”, 4th ed, Addison-Wesley, 2013</td></tr>
<tr><td colspan="5">Form of Testing and Examination: 50% of points from the exercises are required for participation in the project exam, which consists of a software design, an oral presentation and a written report, including a statement of independent, unaided project work. Alternatively to the project exam, an oral (20 min) exam may be announced by the lecturer.</td></tr>
</table>

Code: MScTI_ELEC		**Course Title:** Electronics		
Lecturer: A. Wurz		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** WS

Module Parts and Teaching Methods:

- Lecture on “Electronics” (3hours/week)
- Exercise with homework (1 hours/week)

Objectives: After this course the students will be able to:

- analyze the application of active and passive components,
- understand the methods of circuit design,
- design simple electronic circuits and apply the methods to practical examples.

Contents:

- Resistors, capacitors, inductivities
- Diodes (rectifiers, switches)
- Transistors (amplifier, switches)
- Field-effect transistors (JFET, MOSFET)
- Operational amplifier (amplifier, analog filter)
- Oscillators (LC oscillators, crystal oscillators)
- Phase-locked loop, Laplace transformation
- Power supply circuits
- Transmission of analog and digital signals
- Analog to digital conversion
- Simulation of circuits
- Techniques of electronic design

Prerequisites: none	**Recommended Knowledge:** none

Literature:

- Horowitz and Hill: “The Art of Electronics”
- Herrmann Hinsch: “Elektronik”
- U. Tietze, Ch. Schenk: “Halbleiterschaltungstechnik”

Form of Testing and Examination: To be defined by lecturer before beginning of course

Code:	**Course Title:**
MScTI_INTROHPC	Introduction to High Performance Computing
Lecturer: Prof. Dr. H. Fröning	Type: Lecture with exercise

Credit Points:	**Workload:**	**Teaching Hours:**	**Language:**	**Term:**
6	180h	4 / week	German / English	WS

Module Parts and Teaching Methods:

- Lecture (2 hours/week)
- Exercise with homework (2 hours/week)

Objectives: Students

- know message passing and scalable programming
- are familiar with the most important past and present concepts for large-scale computing problems
- can design and optimize solutions for large-scale computing problems
- know how to use MPI and related software tools to implement large-scale computing problems
- are capable to solve large-scale computing problems with objectives including performance in terms of time and energy, and scalability in terms of time and capacity, and consolidation of compute resources

Contents:

- HPC architectures and message passing
- Parallel algorithm design and Message Passing Interface (MPI)
- MPI internals
- Workload characterization
- Short introduction to accelerated computing
- Practical problems and their solutions

Prerequisites:	**Recommended Knowledge:**
none	Computer architecture basics, parallel programming principles, C, C++, OS basics

Literature:

- Georg Hager, Gerhard Wellein: “Introduction to High Performance Computing for Scientists and Engineers”, Taylor & Francis Inc

Form of Testing and Examination: 15 – 30 min. oral exam or 1h written exam announced by lecturer

<table>
<tr><td colspan="2">Code:
MScTI_MES</td><td colspan="3">Course Title:
Microcontroller Based Embedded Systems</td></tr>
<tr><td colspan="2">Lecturer:
A. Wurz</td><td colspan="3">Type:
Lecture with exercises</td></tr>
<tr><td>Credit Points:
6</td><td>Workload:
180h</td><td>Teaching Hours:
4 / week</td><td>Language:
German / English</td><td>Term:
SS</td></tr>
<tr><td colspan="5">Module Parts and Teaching Methods:
• Lecture (2h)
• Exercise/ Lab work (2h)</td></tr>
<tr><td colspan="5">Objectives: After this course the students will be able to:
• analyse active and passive electronic components for practical applications,
• understand the methods of circuit design, the functionality and programming of microprocessors and peripheral circuits so that they can build a complex microprocessor circuit themselves,
• install a development environment for program development,
• apply methods for debugging in microprocessor circuits.</td></tr>
<tr><td colspan="5">Contents:
• project management
• circuit design
• microcontrollers
• mp3 decoder + Ethernet
• power supply (linear regulators +switching regulators)
• selection of components
• CAD program (schematic +layout)
• manufacturing boards
• stuffing boards
• C program development
• test program
• implementing + debugging
• electronic construction techniques</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="3">Recommended Knowledge:
none</td></tr>
<tr><td colspan="5">Literature:
• Paul Horowitz, Winfield Hill: “The Art of Electronics”</td></tr>
<tr><td colspan="5">Form of Testing and Examination: To be defined by lecturer before beginning of course</td></tr>
</table>

Code: MScTI_RES		Course Title: Reconfigurable Embedded Systems		
Lecturer: Dr. A. Kugel		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** SS

Module Parts and Teaching Methods:

- Lecture (2 hours) / practical exercise (lab, 1 hour avg) / project (lab, 1 hour avg) / homework
- Just-in-time teaching sessions (4 hours) on selected topics

Objectives: After completing this course students are able to:

- list and explain important elements and properties of embedded systems,
- describe fundamental principles and components of reconfigurable technology,
- apply elementary application design methodologies for microprocessors and FPGAs to implement, program and test a sample embedded FPGA platform.

Contents:

- Requirements and specific properties of embedded systems
- Overview on hardware components: microcontrollers, peripherals, FPGAs
- Real-time issues and scheduling
- FPGA design tools: HDL (incl. VHDL tutorial), simulator, debugger
- System-on-Chip architecture – controller, buses and peripherals
- HW/SW co-design
- Embedded system software (stand-alone and real-time kernels)

Prerequisites:	**Recommended Knowledge:**
none	none

Literature:

- Peter Marwedel: “Eingebettete Systeme“, Springer Lehrbuch, 1. Auflage 2007
- Th. Flick, H. Liebig: “Mikroprozessortechnik“, Springer Lehrbuch, 4. Auflage 2004
- H. Bähring: “Mikrorechner-Systeme“, Springer Lehrbuch, 3. Auflage 2002
- Karim Yaghmour: “Building Embedded Systems”, O’Reilly, April 2003

Form of Testing and Examination: 50% score on exercises plus either oral exam (15min) or project (see below). Available option announced at start of course.

Project exam: autonomous elaboration of project task assigned by lecturer. Successful completion requires all of: operational design/program, written report (5 pages), presentation (10 minutes) with colloquium, statement of unaided work.

Soft Skills:

Code: MScTI_TOOLS		**Course Title:** Tools		
Lecturer: several, changing		**Type:** Lecture with exercises		
Credit Points: 4	**Workload:** 120h	**Teaching Hours:** 4 / week	**Language** German / English	**Term:** WS

Module Parts and Teaching Methods:

- Lecture (2 h / week)
- Supervised Exercises (2 h / week)

Objectives: Students

- have an overview of the functionalities of various software tools suited to accomplish frequent tasks, like the creation of drawings and illustrations, programming, solving of mathematical problems, analysis and visualization of data, search for literature or working in a team.
- are able to improve their work flows by choosing an appropriate tool
- are aware that application of a suited tool improves their working quality and efficiency
- are able to deepen their knowledge and skill in the presented tools on the basis of the introductions given

Contents:

Subjects are chosen from a pool of possibilities as a function of interests and need of students. The list is regularly adapted to new developments

- introduction to Linux
- version control tools (git, svn,..)
- introduction to python
- mathematical software (Mathematica)
- data evaluation and plotting (gnuplot, root)
- software documentation tools (doxygen)
- 2D & 3D drawing, construction and visualization (PovRay, OpnSCAD, PostScript, pdf)
- styles and templates (powerpoint, word)
- introduction to Latex
- team work
- project planning
- literature search

Prerequisites: none	**Recommended Knowledge:** none

Literature: announced by lecturer

Form of Testing and Examination: Regular participation (maximum 2 courses missed without justification) and successful completion of supervised exercises

Code: MScTI_ES		**Course Title:** Entrepreneurship		
Lecturer: tbd (external)		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** SS

Module Parts and Teaching Methods:

- Lecture (2 hours/week)
- Exercise (2 hours/week)

Objectives: After completing this course students will be able to:

- use innovation methods like Design Thinking, Rapid Prototyping and Business Model Innovation to develop tangible solutions for real life problems
- present business cases in short presentations (pitches) to stakeholders and investors
- develop basic marketing strategies to find first customers for their products
- apply the legal framework to found their own company

Contents:

- Design Thinking
- Rapid Prototyping
- Business Model Innovation
- Presentation of business cases
- finance
- marketing
- patent law and copyright
- legal forms of companies

Prerequisites: none	**Recommended Knowledge:** none

Literature:

- Eric Ries: “Lean Startup”, Redline Verlag, 2012
- Alexander Osterwalder: “Business Model Generation”, Campus Verlag, 2011
- Steve Blanck: “The Startup Owners's Manual”, K & S Ranch, 2012
- Ash Maurya: “Running Lean”, O'Reilly, 2013

Form of Testing and Examination: Presentation of results

Main Subject / Specialization:

<table>
<tr><td colspan="2">Code:
MScTI_ANASIM</td><td colspan="3">Course Title:
Components, Basic Circuits & Simulation</td></tr>
<tr><td colspan="2">Lecturer:
Prof. Dr. P. Fischer</td><td colspan="3">Type:
Lecture with exercises</td></tr>
<tr><td>Credit Points:
6</td><td>Workload:
180h</td><td>Teaching Hours:
4 / week</td><td>Language:
German / English</td><td>Term:
SS</td></tr>
<tr><td colspan="5">Module Parts and Teaching Methods:
• Lecture (2h)
• Practical exercise with homework (2h)</td></tr>
<tr><td colspan="5">Objectives: Students
• can design simple analog circuits by combining elementary building blocks
• can predict the properties (gain, frequency behavior) of simple circuits and give analytical approximate expressions for gain, bandwidth, output resistance, etc.
• can use analogue simulators to analyze circuits in the time and frequency domain
• know what an operation point is, how it affects circuit behavior and how it can be set
• can relate the geometry and operation point of transistors to their small- and large signal properties</td></tr>
<tr><td colspan="5">Contents:
• Diode and transistor operation principle
• Modelling of Diode und MOS, large / small signal models
• Voltage and current sources, Thevenin equivalent
• Component and circuit description with complex variables
• Bode plot, transfer function
• Analogue simulation (dc, ac, transient)
• Basic circuits: current mirror, gain stage, cascode, source follower, differential pair
• Practical exercises with professional simulation tools</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="3">Recommended Knowledge:
Introduction to physics</td></tr>
<tr><td colspan="5">Literature:
• P. R. Gray, P. J. Hurst, S. H. Lewis, R. G. Meyer: “Analysis and Design of Analog Integrated Circuits”, Wiley & Sons, New York, 1993
• D. A. Johns, K. Martin: “Analog Integrated Circuit Design”, Wiley & Sons, 1997</td></tr>
<tr><td colspan="5">Form of Testing and Examination: To be defined by lecturer before beginning of course</td></tr>
</table>

Code: MScTI_ANADESIGN		Course Title: Full Custom VLSI Design		
Lecturer: Prof. Dr. P. Fischer		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** WS

Module Parts and Teaching Methods:

- Lecture (2h)
- Practical exercise with homework (2h)

Objectives: Students

- can carry out the complete design process from a circuit idea to a final, checked layout,
- understand how design rules are related to semiconductor properties or manufacturing issues,
- are able to practically carry out a mixed mode simulation,
- are able to extract parasitic values and perform a simulation with these parasitics,
- can program simple automatized scripts using SKILL.

Contents:

- Semiconductor manufacturing
- Technology & design rules, technology files
- Layout of components, rules, matching
- Design Rule Check
- Extraction, Layout versus Schematic Check
- ESD and Antenna rules, latchup
- Test equipment & test procedures
- Script programming using SKILL
- Parasitic extraction & simulation
- Mixed Mode simulation

Prerequisites:	Recommended Knowledge:
none	MScTI_ANASIM

Literature:

- Lecture script available online

Form of Testing and Examination: Design (schematic entry, simulation and layout) of a simple circuit with a short presentation.

Code: MScTI_DIGHD		**Course Title:** Digital Hardware Design		
Lecturer: Prof. Dr. U. Brüning		**Type:** Lecture with exercises / lab / project		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** WS

Module Parts and Teaching Methods:

- Lecture (2 hours/week)
- Exercise: design and simulation of digital hardware with EDA software (2 hours/week)

Objectives: The students

- understand the concepts and principles of hardware design and the methodology for design and verification of hardware structures that means concretely that they are able to use their acquired knowledge to design new and efficient hardware and they can simulate and verify the developed designs.

Contents:

- Introduction to the principles of hardware design
- use of Hardware Description Languages like Verilog HDL.
- design of combinational and sequential logic.
- overall design flow for Integrated Circuits
- Design descriptions
- Design elements
- Simulation
- Verification of Hardware

Prerequisites: none	**Recommended Knowledge:** basic knowledge of Digital Circuit Design

Literature: A reading list will be provided in the script.

The script will be accessible on the web site of the Computer Architecture Chair.

Form of Testing and Examination: 30’ oral exam at the end of the semester

At least 50% of the exercises must be passed.

Code:		Course Title:		
MScTI_DIGDF		Digital Semi Custom Design Flow		
Lecturer: Prof. Dr. U. Brüning		**Type:** Lecture with exercises / lab / project		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** SS

Module Parts and Teaching Methods:

- Lecture (2 hours/week)
- Exercise / Project: backend processing for ASICs with EDA software (2 hours/week)

Objectives: The students

- deepen their knowledge of the methodology for semi-custom ASIC design,
- are able to use their acquired knowledge to design very complex chips,
- can run the complete backend design process for modern chip technology.

Contents:

- Advanced methods for design of application specific ICs
- Synthesis of complex hardware systems
- Static Timing Analysis (STA)
- Place&Route of modules and standard cells
- Signal integrity analysis
- Design rule checks
- Generation of mask data
- The SEED-2002 agreement between Cadence Design Systems and the University of Heidelberg allows our students to work and learn with the most modern EDA tools that are usually only used in chip industry.

Prerequisites:	Recommended Knowledge:
none	deeper knowledge of Digital Hardware Design

Literature: A reading list will be provided in the script.

The script will be accessible on the web site of the Computer Architecture Chair.

Form of Testing and Examination: 30’ oral exam at the end of the semester

At least 50% of the exercises and the chip project must be passed.

<table>
<tr><td colspan="2">Code:
MScTI_DIGVERI</td><td colspan="3">Course Title:
Functional Verification</td></tr>
<tr><td colspan="2">Lecturer:
Prof. Dr. U. Brüning</td><td colspan="3">Type:
Lecture with exercises / lab / project</td></tr>
<tr><td>Credit Points:
6</td><td>Workload:
180h</td><td>Teaching Hours:
4 / week</td><td>Language:
German / English</td><td>Term:
SS</td></tr>
<tr><td colspan="5">Module Parts and Teaching Methods:
• Lecture (2 hours/week)
• Exercise (2 hours/week)</td></tr>
<tr><td colspan="5">Objectives: The students
• understand the concepts and principles of functional verification and the methodology,
• use the acquired for building verification environments,
• are able to verify complex hardware designs.</td></tr>
<tr><td colspan="5">Contents:
• Introduction to the principles of functional verification
• Simulation-Based Verification
• Formal Verification
• Use of Hardware Verification Languages like System Verilog
• Use of Verification Methodologies like OVM
• Verification Planning
• Coverage Models
• Assertion-Based Verification</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="3">Recommended Knowledge:
Experience in Digital Hardware Design</td></tr>
<tr><td colspan="5">Literature: A reading list will be provided in the script.
The script will be accessible on the web site of the Computer Architecture Chair.</td></tr>
<tr><td colspan="5">Form of Testing and Examination: 30’ oral exam at the end of the semester
At least 50% of the exercises must be passed.</td></tr>
</table>

Code: MScTI_ANABLOCKS		Course Title: Advanced Analogue Building Blocks		
Lecturer: Prof. Dr. P. Fischer and others		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** WS / SS

Module Parts and Teaching Methods:

- Lecture (2 hours/week)
- Practical exercise (2 hours/week)

Objectives: The students

- have a broad overview of advanced circuits so that they are able to chose an appropriate approach for a given problem
- get a deeper qualitative understanding of the behavior of analogue circuits,
- can quantitatively analyze analogue circuits and extract important figures of merit,
- know a large variety of advanced circuit topologies.

Contents:

The lecture introduces various building blocks, mathematical tools or knowledge on more advanced topics, picked from the list below as a function of student background and interest. Content in SW or SS can vary.

- Advanced transistor properties
- Feedback: properties, mathematical treatment, stability, Nyquist test
- Noise of components and circuits
- Transfer function, impulse response, poles and zeros
- Cascaded amplifiers
- Advanced current mirrors
- Differential circuits, common mode feedback
- DACs and ADCs
- Switches
- Switched Capacitor Circuits

Prerequisites: none	**Recommended Knowledge:** MScTI_ANADESIGN

Literature:

- Razavi: “Design of analog CMOS integrated circuits“
- J. Millman: “Microelectronics“

Form of Testing and Examination: To be defined by lecturer before beginning of course

Code:		Course Title:		
MScTI_DET		Silicon Detectors & Readout Electronics		
Lecturer: Prof. Dr. P. Fischer		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** WS

Module Parts and Teaching Methods:

- Lecture (2 hours/week)
- Exercise with homework (2 hours/week)

Objectives: The students

- know the basic working principles of silicon detectors, so that they are able to derive important properties (speed, resolution..)
- know different sensor types with their properties, so that they can choose the best detector type for a given application
- know the basics on how to read out the signals and which figures of merit are of importance so that they can design / chose readout concepts for a given application

Contents:

- Basics
  - Interactions of particles and photons with matter (short)
  - Semiconductors, doping, diodes, manufacturing technology
  - Spatial resolution, energy resolution, noise…
- Particle Sensors
  - PiN Diodes, Pads, Pixel, Strips
  - DEPFETs, MAPS
  - Non-silicon materials
- Photo Sensors
  - Quantum efficiency, spectral sensitivity, response time
  - APDs, SiPMs, CCDs, CMOS APS, others
- Readout circuits
  - Charge amplifier, Transimpedance amplifier, bandwidth, noise
  - Readout chips for strip- and pixel detectors
- Applications

Prerequisites:	Recommended Knowledge:
none	Basic knowledge in Electrodynamics, Quantum Mechanics and Solid State Physics

Literature:

- S. M. Sze: “Semiconductor Devices”, Wiley, ISBN 0471874248
- G. Lutz: “Semiconductor Radiation Detectors”, Springer, ISBN 3540648593
- Rossi/Fischer/Rohe/Wermes: “Pixel Detectors”, Springer, ISBN 3540283323

Form of Testing and Examination: To be defined by lecturer before beginning of course

Code: MScTI_GPU		**Course Title:** GPU Computing		
Lecturer: Prof. Dr. H. Fröning		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** WS

Module Parts and Teaching Methods:

- Lecture (2 hours/week)
- Exercise with homework (2 hours/week)

Objectives: Students

- know the factors that determine the performance of GPU programs, and are able to program GPUs to solve computing problems,
- are familiar with GPU architecture and design decisions,
- can design and optimize CUDA programs for compute- or memory-intensive problems,
- know how to use CUDA tools to aid in programming, debugging and performance tuning,
- are capable to solve compute- or memory-intensive problems using GPUs with objectives including performance in terms of time and energy, and are capable to decide when accelerators like GPUs are suitable for a given computing problem.

Contents:

- Basics of GPU architecture and programming model
- Introduction to CUDA
- Performance optimization techniques
- Consistency and coherence of GPUs
- Alternatives to CUDA and advanced GPU concepts

Prerequisites:	**Recommended Knowledge:**
none	Parallel programming, C++ programming skills

Literature:

- T.G. Mattson, B.A. Sanders, B.L. Massingill: “Parallel Patterns for Parallel Programming”, Addison Wesley 2004
- D.B. Kirk, W.W. Hwu: “Programming Massively Parallel Processors”, Morgan-Kaufmann 2010

Form of Testing and Examination: 15 – 30 min. oral exam or 1h written exam, announced by lecturer

Code: MScTI_PAD		**Course Title:** Parallel Algorithm Design		
Lecturer: Prof. Dr. R. Strzodka		**Type:** Lecture with exercises and project		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** WS

Module Parts and Teaching Methods:

- Lecture 2 hours/week
- Exercise 1 hour/week on average plus homework
- Project 1 hour/week on average plus homework

Objectives: Students are able to:

- exploit the available parallelism in modern CPUs,
- make design decisions depending on tradeoffs in parallel algorithms,
- apply and combine parallel patterns in their own programs.

Contents:

- Multiple levels of parallelism
- Parallel design patterns
- Parallel data access
- Communication vs. computation
- Latency vs. throughput
- Work efficiency vs. step efficiency
- Locality vs. parallelism
- Tools for parallel programming
- Intensive exercises

Prerequisites: none	**Recommended Knowledge:** Basic C++

Literature:

- Michael McCool, Arch Robison, James Reinders: “Structured Parallel Programming”, Morgan Kaufmann, 2012

Form of Testing and Examination: 50% of points from the exercises are required for participation in the project exam, which consists of a software design, an oral presentation and a written report, including a statement of independent, unaided project work. Alternatively to the project exam, an oral (20 min) exam may be announced by the lecturer.

Code: MScTI_ACC		**Course Title:** Accelerator Practice		
Lecturer: Prof. Dr. R. Strzodka		**Type:** Lecture with exercises and project		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** SS

Module Parts and Teaching Methods:

- Lecture 2 hours/week
- Exercise 1 hour/week on average plus homework
- Project 1 hour/week on average plus homework

Objectives: Students are able to:

- program accelerators on a high-level with parallel patterns,
- create multi-backend programs that can run on different architectures,
- select efficient parallel algorithms from existing accelerator libraries.

Contents:

- Overview of programming paradigms for accelerators
- High level accelerator programming
- Effective use of STL-like algorithm libraries
- Multi-backend programming for different architectures
- Libraries for dense and sparse linear algebra
- Specialized libraries
- Simultaneous use of multiple accelerators

Prerequisites:	**Recommended Knowledge:**
none	Basic C++

Literature:

- Will be announced by the lecturer

Form of Testing and Examination: 50% of points from the exercises are required for participation in the project exam, which consists of a software design, an oral presentation and a written report, including a statement of independent, unaided project work. Alternatively to the project exam, an oral (20 min) exam may be announced by the lecturer.

<table>
<tr><td colspan="2">Code:
MScTI_APA</td><td colspan="3">Course Title:
Advanced Parallel Algorithms</td></tr>
<tr><td colspan="2">Lecturer:
Prof. Dr. R. Strzodka</td><td colspan="3">Type:
Lecture with exercises and project</td></tr>
<tr><td>Credit Points:
6</td><td>Workload:
180h</td><td>Teaching Hours:
4 / week</td><td>Language:
German / English</td><td>Term:
WS</td></tr>
<tr><td colspan="5">Module Parts and Teaching Methods:
• Lecture 2 hours/week
• Exercise 1 hour/week on average plus homework
• Project 1 hour/week on average plus homework</td></tr>
<tr><td colspan="5">Objectives: Students are able to:
• apply advanced transformations to improve parallelism and locality,
• make detailed design decisions depending on tradeoffs in parallel algorithms,
• balance numerical efficiency and parallel efficiency.</td></tr>
<tr><td colspan="5">Contents:
• The lectures MScTI_PAD and MScTI_APA can be attended in the same semester in parallel. MScTI_PAD looks at more topics in breadth, while MScTI_APA looks at fewer topics in depth.
• Most recent developments in GPUs
• On-the-fly data transformations
• Data locality optimizations
• Hierarchical algorithms
• SIMD utilization
• Precision, accuracy and numerical schemes
• Numerical efficiency vs. parallel efficiency
• Data representation</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="3">Recommended Knowledge:
Basic C++, C++11, CUDA (e.g. MScTI_GPU)</td></tr>
<tr><td colspan="5">Literature:
• David B. Kirk, Wen-mei W. Hwu: “Programming Massively Parallel Processors”, 3rd ed, Morgan Kaufmann, 2017
• More will be announced by the lecturer</td></tr>
<tr><td colspan="5">Form of Testing and Examination: 50% of points from the exercises are required for participation in the project exam, which consists of a software design, an oral presentation and a written report, including a statement of independent, unaided project work. Alternatively to the project exam, an oral (20 min) exam may be announced by the lecturer.</td></tr>
</table>

<table>
<tr><td colspan="2">Code:
MScTI_APC</td><td colspan="3">Course Title:
Advanced Parallel Computing</td></tr>
<tr><td colspan="2">Lecturer:
Prof. Dr. H. Fröning</td><td colspan="3">Type:
Lecture with exercises</td></tr>
<tr><td>Credit Points:
6</td><td>Workload:
180h</td><td>Teaching Hours:
4 / week</td><td>Language:
German / English</td><td>Term:
SS</td></tr>
<tr><td colspan="5">Module Parts and Teaching Methods:
Lecture (2 hours/week)
Exercise with homework (2 hours/week)</td></tr>
<tr><td colspan="5">Objectives: The students
know principles of parallel architectures, including synchronization, consistency, and coherence,
are familiar with advanced concepts like transactional memory, relaxed consistency, and multi-threading,
know how to design and optimize complex parallel code for particular compute and synchronization problems,
are capable of solving complex computing problems using massively parallel processors, understanding the implications of architectural design decisions on performance in terms of time and energy, and reasoning about the suitability of certain processor architectures for a given computing problem.</td></tr>
<tr><td colspan="5">Contents:
Principles of parallel computing
Shared memory architectures
Programming paradigms, communication and synchronization concepts and algorithms
Consistency models and scalable cache coherence
Multi-/many-core and multi-threading architectures
Emerging topics in parallel computing</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="3">Recommended Knowledge:
MScTI_PCA, MScTI_INTROHPC, C++, OS basics</td></tr>
<tr><td colspan="5">Literature:
John L. Hennessy, David A. Patterson: “Computer Architecture: A Quantitative Approach” (The Morgan Kaufmann Series in Computer Architecture and Design)
Maurice Herlihy, Nir Shavit, “The Art of Multiprocessor Programming”, Morgan Kaufmann</td></tr>
<tr><td colspan="5">Form of Testing and Examination: 15 – 30 min. oral exam or 1h written exam, announced by lecturer</td></tr>
</table>

Code: MScTI_FPGA		Course Title: FPGA Coprocessors		
Lecturer: Dr. A. Kugel		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** WS

Module Parts and Teaching Methods:

- Lecture (2 hours) / practical exercise (lab, 1 hour avg) / project (lab, 1 hour avg) / homework
- Just-in-time teaching sessions (4 hours) on selected topics

Objectives: After completing this course students are able to:

- list and explain advanced components of FPGA devices,
- list and explain coprocessor architectures and communication types,
- select, configure and program FPGA IP library elements,
- create custom IP cores using structural data-flow and FSM based control-flow design techniques,
- use IP cores to create hybrid applications for processor and reconfigurable coprocessor with appropriate interface mechanisms,
- program and test coprocessor applications.

Contents:

- Reconfigurable Computing Hardware
    - FPGA Device Architecture and Features
    - Reconfigurable Computing Architectures
    - (Re-)Configuration Management
- Programming Reconfigurable Systems
    - Compute Models and System Architectures
    - Programming FPGA Applications in VHDL
    - Data- and Control- Flow Graphs
    - High-Level Synthesis Tools
- Mapping Designs to Reconfigurable Platforms
    - Technology Mapping
    - Datapath Optimizations
- Projects: Implementing Applications with FPGAs
- Computation, Image-processing, I/O oriented

Prerequisites: none	Recommended Knowledge: FPGA and HDL fundamentals (e.g. from MScTI_RES)

Literature:

- Scott Hauck & André Dehon: “Reconfigurable Computing”, Morgan Kaufmann, 2008

Form of Testing and Examination: 50% score on exercises plus either oral exam (15min) or project (see below). Available option announced at start of course.

Project exam: autonomous elaboration of project task assigned by lecturer. Successful completion requires all of: operational design/program, written report (5 pages), presentation (10 minutes) with colloquium, statement of unaided work.

Code: MScTI_HPNET		**Course Title:** High Performance Interconnection Networks		
Lecturer: Prof. Dr. U. Brüning		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** SS

Module Parts and Teaching Methods:

- Lecture (2 hours/week)
- Exercise with homework (2 hours/week)

Objectives: The students

- understand the concepts and principles of interconnection networks,
- will be able to configure and use interconnection networks for given demands,
- can use the learned structures to develop new high performance interconnection networks.

Contents:

- Topologies, Switching, Routing, Flow Control
- Fault tolerance and Deadlocks
- Collective Communications
- Congestion Management
- Network Interfaces
- On-Chip Networks
- Performance Evaluation and Simulation

Prerequisites:	**Recommended Knowledge:**
none	MScTI_PCA, MScTI_APC

Literature: A reading list will be provided in the script.

The script will be accessible on the web site of the Computer Architecture Group.

Form of Testing and Examination: 30‘ min. oral exam or 2h written exam, announced by lecturer

The following module is imported from the master program in computer science:

<table>
<tr><td colspan="2">Code:
IPHR</td><td colspan="3">Course Title:
Parallel Algorithms, Application Computing (Paralleles Höchstleistungsrechnen)</td></tr>
<tr><td colspan="2">Lecturer:
Prof. Dr. P. Bastian/ Dr. S. Lang</td><td colspan="3">Type:
Lecture with exercises</td></tr>
<tr><td>Credit Points:
8</td><td>Workload:
240h</td><td>Teaching Hours:
6 / week</td><td>Language:
German</td><td>Term:
WS</td></tr>
<tr><td colspan="5">Module Parts and Teaching Methods:
• Lecture 4 SWS
• Exercises 2 SWS</td></tr>
<tr><td colspan="5">Objectives: The student
• knows different architectures for high-performance computers,
• knows synchronization mechanisms in parallel systems including performance aspects,
• can handle the most important programming paradigms for parallel systems,
• is able to solve basic synchronization tasks,
• understands the parallelization of linear algebra algorithms,
• is able to assess the performance of a parallel program.</td></tr>
<tr><td colspan="5">Contents:
• Systems with global address space
• Cache coherence
• Systems with local address space and message passing
• critical sections, condition synchronization, semaphore
• posix threads
• programming of graphics cards
• message passing theory, MPI
• client server model, remote procedure call
• Assessment of parallel algorithms
• load balancing
• dense linear algebra algorithms, solution of sparse linear systems
• particle methods
• parallel sorting</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="3">Recommended Knowledge:
knowledge of a higher-level programming language (preferably C, C++), knowledge of data structures and algorithms</td></tr>
<tr><td colspan="5">Literature:</td></tr>
<tr><td colspan="5">Form of Testing and Examination: Minimum 50% successful exercises and final examination.</td></tr>
</table>

The following module is imported from the master program in computer science:

<table>
<tr><td colspan="2">Code:
IPLGG</td><td colspan="3">Course Title:
Parallel Solution of Large Linear Systems
(Parallele Lösung großer Gleichungssysteme)</td></tr>
<tr><td colspan="2">Lecturer:
Prof. Dr. P. Bastian</td><td colspan="3">Type:
Lecture with exercises</td></tr>
<tr><td>Credit Points:
8</td><td>Workload:
240h</td><td>Teaching Hours:
6 / week</td><td>Language:
German</td><td>Term:
WS</td></tr>
<tr><td colspan="5">Module Parts and Teaching Methods:
• Lecture 4 SWS
• Exercises 2 SWS</td></tr>
<tr><td colspan="5">Objectives: The student
• knows the discretization of scalar elliptic partial differential equations with the finite element method,
• understands the abstract concept of subspace correction methods,
• is able to apply this to domain decomposition and multigrid methods,
• understands the convergence theory for these methods,
• is able to implement these methods on a parallel system and can judge the performance of the methods.</td></tr>
<tr><td colspan="5">Contents:
• Basis of Finite Element methods for elliptic partial differential equations
• Subspace correction methods
• Overlapping and non-overlapping domain decomposition methods with convergence theory
• Geometric multigrid methods with convergence theory
• Algebraic multigrid methods</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="3">Recommended Knowledge:
knowledge of a higher-level programming language (preferably C++), knowledge of numerical methods for differential equations</td></tr>
<tr><td colspan="5">Literature:</td></tr>
<tr><td colspan="5">Form of Testing and Examination: Minimum 50% successful exercises and final examination.</td></tr>
</table>

Code: MScTI_ROB1		Course Title: Robotics 1 - Kinematics, Dynamics and Control		
Lecturer: Prof. Dr. L. Masia		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** WS

Module Parts and Teaching Methods:

- Lecture (2 hours / week)
- Exercises (2 hours / week)

Objectives: The students

- can apply principles of mechanics to mechanisms and robotics problems,
- can explain theory and solve problems using appropriate algorithms of robot kinematics and dynamics,
- can give an overview on state of the art robotics applications in various fields,
- can explain the function of robotics hardware such as actuators, sensors in a robotic system,
- can understand the different control solution in industry and human-robot interface.

Contents:

- State of the art robot types (Humanoid robots, manipulators, wearable robots and assistive devices, swarm robots, unmanned land/sea/aerial vehicles, etc.)
- State of the art robot applications in (Industry, Medicine, Care, Rescue/Humanitarian, Space, Transport etc.)
- Actuators and sensors in robotics
- Mechanical concepts, rigid body motions and homogeneous transformations
- Forward and Inverse kinematics of open chains
- Differential kinematics and statics
- Trajectory generation in joint and cartesina workspace
- Motion planning
- Dynamics
- Robot control

Prerequisites:	Recommended Knowledge:
none	Basic knowledge in Mechanics and Linear Algebra

Literature:

- B. Siciliano, et al: “Robotics - Modeling, Planning and Control”
- F. Park & K. Lynch: “Modern Robotics – Mechanics, Planning and Control”
- Mark W. Spong, Seth Hutchinson and M. Vidyasagar: “Robot Dynamics and Control”, second edition

Form of Testing and Examination: Written exam at the end of the semester.

Successful participation in the exercises is required to be accepted to exam.

Code: MScTI_ROB2		**Course Title:** Robotics 2 - Simulation and Optimization in Robotics		
Lecturer: Prof. Dr. L. Masia		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** SS

Module Parts and Teaching Methods:

- Lecture (2 hours / week)
- Programming Exercises (2 hours / week)

Objectives: The students

- can explain and apply advanced principles of modeling, optimization and control of dynamic processes, in particular mechanical systems,
- can apply nonlinear optimization and optimal control methods and can compare and evaluate different mathematical approaches,
- can model, classify and analyze complex motions of mechanical systems, e.g. in robotics or biomechanics, and investigate specific properties such as stability,
- know how to use software tools based on C++ and Lua for modeling, simulation, optimization and visualization of humanoid and robotic systems,
- are capable of solving optimal control problems numerically and to evaluate the quality of the solution.

Contents:

- Dynamic process modeling
- Modeling of complex mechanical systems (e.g. humanoids)
- Simulation of mechanical Systems (Integrators and Initial Value Problems)
- Boundary value problems
- Nonlinear optimization problems
- Optimal control problems in robotics
- Direct and indirect methods for optimal control problems
- Stability of dynamical systems
- Simulation and visualization of mechanical systems
- Modeling multi body systems with RBDL (Rigid Body Dynamics Library)
- Visualization of motions of mechanical systems with Puppeteer
- Solution of optimal control problems with MUSCOD-II for different mechanic al examples
- Modeling bipedal walking and running motions

Prerequisites:	**Recommended Knowledge:**
Robotics 1 - Kinematics, Dynamics and Control or Theoretical Mechanics or similar knowledge	Programing skills in C/C++ Knowledge in Matlab/Octave Introduction to Numerical mathematics, Algorithmic Optimization 1, Numerical mathematics 1;

Literature:

- J. T. Betts: “Practical Methods for Optimal Control Using Nonlinear Programming”
- J. Nocedal, S. Wright: “Numerical Optimization”

Form of Testing and Examination: Written exam at the end of the semester.

Successful participation in the programming exercises is required to be accepted to exam.

Code: MScTI_BIOMECH		Course Title: Biomechanics and Biorobotics		
Lecturer: Prof. Dr. L. Masia		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week + block	**Language:** German / English	**Term:** WS

Module Parts and Teaching Methods:

- Lecture (2 hours / week) with exercises (2 hours / week)
- practical (block)

Objectives: The students

- can explain the basics of human physiology,
- can distinguish between different concepts of biological motion,
- can model different aspects of biological motion generation (neural control, muscle activity, reflexes),
- understand the function and are familiar with the use of devices for motion analysis such as marker-based and IMU based motion capture systems and electromyography,
- can explain the concept of human machine interaction and biorobotics
- can understand the theory behind control of interacting system for measuring human biometric and biomechanical signals
- are able to independently plan and execute a biomechanical study, possibly in a team,
- can analyze motion capture data with respect to a specific biomechanical question,
- can write code for analysis or visualization of biomechanical data,
- can present project results in a scientific way using posters, presentations or other media,
- are able to formulate a documentation for the project including the created code.

Contents:

- Physiological basics of the human body and of animals
- Body proportions and anthropometric data
- Muscle physiology and muscle models
- Neural control of biological motion and interaction
- Human sensor systems and sensor-based motion control
- Human motion/interaction measurements: camera and marker based (sparse) motion capture, IMU based motion capture, electromyography, force plates, pressure soles, markerless motion capture
- Biorobotics and human-robot interaction
- Control of interactive robotic devices
- Methodological principles of control and experimental design using robotics
- Human motion and interaction performance analysis
- Design and execution of a problem specific biomechanical lab study

Prerequisites: none	Recommended Knowledge: Robotics 1- Kinematics, Dynamics and Control

Literature:

- Robert McNeill Alexander: “Exploring Biomechanics - Animals in Motion”
- David A. Winter: “Biomechanics and Motor Control of Human Movement”
- Etienne Burdet , David W. Franklin, e al. Human Robotics: “Neuromechanics and Motor Control”
- Reza Shadmehr, Steven P. Wise: “The Computational Neurobiology of Reaching and Pointing: A Foundation for Motor Learning” (Computational Neuroscience Series)

• Reza Shadmehr, Sandro Mussa-Ivaldi: “Biological Learning and Control: How the Brain Builds Representations, Predicts Events, and Makes Decisions” (Computational Neuroscience Series)
Form of Testing and Examination: Successful completion of biomechanical lab project with presentation and report

Code: MScTI_HAPTICS		**Course Title:** Haptics and Human Robot Interaction / Rehabilitation		
Lecturer: Prof. Dr. L. Masia		**Type:** Lecture with exercises		
Credit Points: 6	**Workload:** 180h	**Teaching Hours:** 4 / week	**Language:** German / English	**Term:** SS

Module Parts and Teaching Methods:

- Lecture (2 hours / week)
- Programming Exercises (2 hours / week)

Objectives: The students

- can understand the design principles behind assistive technology,
- can run CAD program and design basic interactive systems,
- know the different technological solutions for haptics and robotic rehabilitation,
- can explain and apply principles of modeling and control of dynamically interacting mechanical systems,
- can apply control methods for human-robot interaction devices ,
- can model, actuators and mechanical systems, in robotics or biomechanics, and investigate stability robustness and metrological performance,
- know how to use software tools based on Matlab Simulink for modeling, simulation, and data visualization in rehabilitation devices,
- know how to implement a stable controller for haptic,
- are capable of analyzing data collected by means of rehabilitation devices and running statistical analysis.

Contents:

- Dynamically interacting mechanical systems (e.g. haptic devices)
- Sensing and motor specialization in human physiology
- Haptics and human robot interaction
- Actuation, sensors and controllers for haptics
- Mechanical design solutions of interacting Robots
- End Effector robots, exoskeletons and exosuits
- Introduction to CAD for mechanical systems and haptic devices
- Control problems in rehabilitation robotics
- Admittance and impedance controllers
- Stability of dynamically interacting systems
- Foundation of prosthetics and orthotics
- Mechanical measurement for human machine interactions
- Clinical data analysis and statistics

Prerequisites: none	**Recommended Knowledge:** Knowledge in Matlab/Simulink Robotics 1 - Kinematics, Dynamics and Control Control Systems Design (or System Theory)

Literature:

- Thorsten A. Kern. Engineering “Haptic Devices: A Beginner's Guide for Engineers”
- Ming C. Lin e Miguel Otaduy. “Haptic Rendering: Foundations, Algorithms, and Applications” (English Edition)”
- Lynette Jones. “Haptics” (MIT Press Essential Knowledge series)
- Material provided by the Instructor and Lecturer

Form of Testing and Examination: Successful completion of working groups lab project, using available setup with final presentation and report.

<table>
<tr><td colspan="2">Code:
MScTI_ROGA</td><td colspan="3">Course Title:
Robotic Games</td></tr>
<tr><td colspan="2">Lecturer:
Prof. Dr. L. Masia</td><td colspan="3">Type:
Project oriented course</td></tr>
<tr><td>Credit Points:
6</td><td>Workload:
180h</td><td>Teaching Hours:
4 / week</td><td>Language:
German / English</td><td>Term:
WS / SS</td></tr>
<tr><td colspan="5">Module Parts and Teaching Methods:
• Supervised project groups in the Laboratory
• Theoretical foundations in lectures form</td></tr>
<tr><td colspan="5">Objectives: After this course the students will be able to:
• design an autonomous mobile robot
• solve a two-player non-cooperative game and implement the solution in real-time</td></tr>
<tr><td colspan="5">Contents:
• Behavioral-based recursive, nested control structure (RNBC)
• Kinematics of wheeled mobile robots
• Non-holonomic control
• Foundations of game theory in particular non-cooperative two-player games, e.g. Cat and Mouse
• Practical implementations and experimental testing of the methods and algorithms used</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="3">Recommended Knowledge:
Knowledge of C/C++</td></tr>
<tr><td colspan="5">Literature:
• Badreddin, E.: "Control and System Design of Wheeled Mobile Robots", Habilitationsschrift, 1997
• Dudek, G., Jenkin, M.: “Computational Principles of Mobile Robotics”, Cambridge University Press, 2000
• Y. C. Shin and C. Xu: “Intelligent Systems: Modeling, Optimization, and Control”, CRC Press, 2008
• J. Engwerda: “LQ Dynamic Optimization and Differential Games“, J. Wiley, 2005</td></tr>
<tr><td colspan="5">Form of Testing and Examination:
Experimental demonstration, oral presentation and short written report</td></tr>
</table>

<table>
<tr><td colspan="2">Code:
MScTI_ROBP</td><td colspan="3">Course Title:
Robotics Practical</td></tr>
<tr><td colspan="2">Lecturer:
Prof. Dr. L. Masia</td><td colspan="3">Type:
Practical course</td></tr>
<tr><td>Credit Points:
6</td><td>Workload:
180h</td><td>Teaching Hours:
4 / week</td><td>Language:
German / English</td><td>Term:
WS / SS</td></tr>
<tr><td colspan="5">Module Parts and Teaching Methods:
• Practical course in groups of 2 persons</td></tr>
<tr><td colspan="5">Objectives: The students
• can use different types of robotics hardware and explain the respective applications and challenges related to them,
• are able to independently plan and execute robotics projects in a team,
• can apply theoretical knowledge in robotics to implement solutions on real platforms,
• can present results of a robotics project in a scientific way.</td></tr>
<tr><td colspan="5">Contents:
• Challenges of real hardware vs. model calculations
• Working principles and practical implementation of sensors and actuators
• Development or modification of robotic hardware
• Code development for specific hardware
• Robotic projects on different kinds of hardware such as (but not limited to)
o Robot arm
o Mobile platform
o Unmanned aerial vehicle
o Humanoid robot</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="3">Recommended Knowledge:
Basic knowledge in C/C++
Robotics 1 - Kinematics, Dynamics and Control</td></tr>
<tr><td colspan="5">Literature:
• Script</td></tr>
<tr><td colspan="5">Form of Testing and Examination: Oral colloquium and written documentation.</td></tr>
</table>

Code: MScTI_SEM		Course Title: Advanced Seminar		
Lecturer: all groups		**Type:** Seminar with presentation		
Credit Points: 4 + 2 (soft skills)	**Workload:** 180h	**Teaching Hours:** 2 / week	**Language:** German / English	**Term:** WS / SS

Module Parts and Teaching Methods:

- Seminar

Objectives: After this course the students will be able to:

- search literature for a specific subject,
- select subject / material for a presentation,
- prepare material (slides) for a presentation,
- give a scientific presentation.

Contents:

- Literature research
- Preparation of presentation
- Oral Presentation (~45 Minutes)
- Preparation of a short summary report (~10 pages)
- Active participation in other student's presentations & discussion

Prerequisites:	**Recommended Knowledge:**
none	General knowledge about the chosen field

Literature:

Partially provided by lecturer

Form of Testing and Examination: Presentation, written summary, regular active participation

<table>
<tr><td colspan="2">Code:
MScTI_SA</td><td colspan="3">Course Title:
Student Research Project</td></tr>
<tr><td colspan="2">Lecturer:
all groups</td><td colspan="3">Type:
Practice course</td></tr>
<tr><td>Credit Points:
14</td><td>Workload:
420h</td><td>Teaching Hours:
n.a.</td><td>Language:
German / English</td><td>Term:
WS / SS</td></tr>
<tr><td colspan="5">Module Parts and Teaching Methods:
• Practical course</td></tr>
<tr><td colspan="5">Objectives: After this course the students will be able to:
• dig into scientific and technical aspects of a selected topic,
• manage and carry through a small research project,
• write a medium length report.</td></tr>
<tr><td colspan="5">Contents:
• Research work on a specific topic.
• Management of work.
• Preparation of a medium length report.</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="3">Recommended Knowledge:
Knowledge in research field</td></tr>
<tr><td colspan="5">Literature:
Depending on subject, provided by supervisor</td></tr>
<tr><td colspan="5">Form of Testing and Examination: Written report</td></tr>
</table>

<table>
<tr><td colspan="2">Code:
MScTI_THESIS</td><td colspan="3">Course Title:
Master Thesis</td></tr>
<tr><td colspan="2">Lecturer:
all groups</td><td colspan="3">Type:
Practice course</td></tr>
<tr><td>Credit Points:
30</td><td>Workload:
900h</td><td>Teaching Hours:
n.a.</td><td>Language:
German / English</td><td>Term:
WS / SS</td></tr>
<tr><td colspan="5">Module Parts and Teaching Methods:
• Master Thesis</td></tr>
<tr><td colspan="5">Objectives: After this course the students will be able to:
• manage and carry through a large research project,
• write an extended thesis,
• report on own scientific work in an oral presentation.</td></tr>
<tr><td colspan="5">Contents:
• Research work on a specific topic.
• Management of work.
• Preparation of a longer written thesis.
• Oral presentation in the colloquium.</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="3">Recommended Knowledge:
Knowledge in research field</td></tr>
<tr><td colspan="5">Literature:
Depending on subject, provided by supervisor</td></tr>
<tr><td colspan="5">Form of Testing and Examination: Written thesis, colloquium</td></tr>
</table>