Master of Science

Computer Engineering (Technische Informatik)

(MScTI)

Description of the course modules (Modulhandbuch)

Heidelberg University
Faculty of Engineering Sciences

Version 2.2

Key Information

Name of university	Heidelberg University
Name of department	Faculty of Engineering Sciences
Name of degree course	Master of Science Computer Engineering
Type of degree course	Consecutive
Acronym	MScTI
Formats of studies	Full time or part time
Prescribed duration of study	2 years, i.e. 4 semesters
Total number of credit points	120
Location	Heidelberg
University places	Unlimited
Target group	Holders of Bachelor of Science, Magister, Staatsexamen, Diploma or equivalent final degree of at least 6 semester study. Major in computer science, Mathematics, Natural Sciences or Engineering with proven knowledge in computer science (modules with $\approx$ 24 CP).
Date of version	31.01.2024

1 Qualification Goals and Profile

1.1 Präambel – Qualification Goals at Heidelberg University

In keeping with Heidelberg University's mission statement and constitution, degree programs are designed to provide a comprehensive academic education, incorporating subject-specific, cross-disciplinary, and career-related objectives that prepare students for their future professional careers. The resulting skills profile is included in the course description for all university disciplines and is implemented in each degree program's specific qualification objectives, curricula, and modules.

The main points of the competence profile are:

- Development of subject-specific skills, with a particular emphasis on research
- Development of the skills required for trans-disciplinary dialogue
- Development of practical problem-solving skills
- Development of personal and social skills
- Promotion of students' willingness to assume social responsibility on the basis of the skills acquired.

1.2 Profile of the MScTI

The research oriented master program in Computer Engineering (MScTI) at Heidelberg University is organized by the Institute of Computer Engineering and the Faculty of Engineering Sciences. Its educational objective is to deepen and broaden the student's expertise and prepare them for a research or development oriented professional career in the field of Computer Engineering as well as for participation in PhD programs. The students develop a thorough understanding of various approaches and methods and are able to assess their advantages and drawbacks in order to develop the best solution for a given problem. They are able to realize which solutions are inappropriate or suboptimal and have the necessary skills to devise novel approaches. The MScTI has a focus on practical skills. Students learn to work with state-of-the-art tools from computer engineering research and are able to apply these skills to develop working solutions for application-oriented problems efficiently.

Students can choose among one of three specializations *Emerging Computing, Microelectronics, Biorobotics*, which are sub-fields of 'computer engineering'. 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.

The master includes a research phase, consisting of a Seminar, a Study Project and the Master Thesis, where students acquire the ability to do independent research and to document and publish their own 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.

1.3 Subject Related Qualification Objectives

After completing the master program 'Computer Engineering', the obtained skills obviously depend on the chosen specialization.

Emerging Computing: Graduates are able to program parallel systems with shared and distributed memories. They know a wide range of compute architectures and specialized hardware like Multi-Core CPUs, GPUs, or FPGA coprocessors. They understand the respective strengths and performance limitations, power efficiency and bottlenecks and can address these issues for instance by programming techniques or suited data structures.

Microelectronics: Graduates can understand and design analogue and digital circuits and systems up to the practical implementation in microelectronic chips. They understand concepts about the functionality and programming of microprocessors and peripheral circuits as well as of reconfigurable architectures. They can use these concepts to design own circuits or to implement them in embedded FPGA platforms. They know about emerging hardware concepts in particular for implementing fast and efficient compute hardware.

Biorobotics: Graduates have acquired a deep understanding of nonlinear dynamical systems in order to design nonlinear control systems. The graduates have in-depth knowledge of modeling, simulation, design, control and optimization of real systems in robotics and rehabilitation and the use of the latest sensor technology.

1.4 Transdisciplinary Qualification Objectives

Graduates of the MScTI possess the skill to work independently with a variety of tools for various special applications and to choose the appropriate ones to solve specific problems. They are able to work in a structured way and can organize complex professional projects. Additionally, they can acquire a basic understanding about legal and financial aspects of creating and running a company and are able to apply marketing strategies and tools.

1.5 Employment Opportunities

Emerging Computing: The ever-growing importance of information technology asks for a detailed knowledge of the underlying compute hardware in, for instance, computing centers, cars, mobile and edge devices. Performance, power efficiency and cost targets ask for optimized hardware choices for the task at hand. The large number of companies working in the above areas are potential employers for the graduates of the MScTI.

Microelectronics: The increasing need for microelectronics devices and the wish of Europe for more independence will lead to a strongly increasing need of competence in the area of microelectronics and hardware design. Small startup companies in the field as well a large players are therefore potential employers.

Biorobotics: Graduates having acquired competences in the areas of biorobotics can for instance address challenges in the wide area of rehabilitaton and in the enhancement of the quality of life of elderly people. The general competences on sensing and control are applicable in a wide range of applications, including industrial automation.

2 Structure of the MScTI [1]

The modules in the MScTI belong to one of the categories

- *Fundamentals* (Grundlagen),
- *Specializations* (Vertiefung),
- *Free Courses* (frei wählbare Veranstaltungen),
- *Soft Skills*,
- *Seminar*,
- *Study Project* (Studienarbeit) and
- *Master Thesis*.

These categories are described in the following subsections:

Fundamentals

Albeit a broad knowledge basis is typically provided by a Bachelor degree, the modules in category Fundamentals shall contribute to establish a well defined background knowledge and competences for all students of the MScTI, who may have completed Bachelor studies in very different fields. For some students, overlap with completed BSc modules is unavoidable, and the modules should be seen as a 'refresh' courses. The two modules

- Advanced Computer Architecture
- Control Systems Design

are mandatory. They are scheduled in the 1st and 2nd semester (i.e. one in winter term, the other in summer term). Because all new students must attend, these modules also help to know each other.

Specializations

These modules treat, on an advanced level, topics from the 3 main research directions of ZITI:

- *Microelectronics*,
- *Emerging Computing* and
- *Biorobotics*.

The available modules and their assignment to one or more specializations are listed in the table on page 8. Students can chose freely among all modules in this category. A total of 6 modules must be completed. We strongly recommend, however, to concentrate on one specialization topic. Completing a significant number of modules from one specialization provides a broad knowledge base and eases the completion of the Master Thesis. The successful completion of a specialization will be documented explicitly in the Master Grade Report as a 'major' if

- 5 modules from the specialization are completed
- including the 2 modules of that specialization labeled as 'compulsory'.

[1] This version of the module handbook was created as part of a general revision of the degree program in the context of the HeiQuality quality assurance procedures. Since the examination regulations take longer to change than the module handbook, there might be a time window, where module handbook and examination regulations are out of sync. Please always check the currently valid examination regulations, as they have priority over the module handbook.

Free Courses

The aim of the Free Courses is to broaden expertise and knowledge. The prime intention is to look beyond computer science and engineering, but such modules can be chosen as well. Students can benefit from the huge lecture program of Heidelberg University and gain for instance background knowledge in fundamentals or applications of their research track. The lectures can be chosen freely from the course catalog of the Heidelberg University. Modules on Bachelor level are allowed, but should not repeat existing knowledge. The modules chosen as Free Courses must meet the following conditions:

- they are graded,
- the sum of credit points is 12 CP or more,
- they broaden expertise beyond the status quo, i.e. they do not repeat already completed modules, for instance from a previous Bachelor program.

Obviously, Bachelor modules completed in a previous study do not broaden knowledge and cannot be counted. In case of doubt, the Prüfungsausschuss decides on the approval. The application must be filed before the module is taken.

It is also possible to select modules from the MScTI for the Free Courses. This can be useful if the choice of a preferred track is not yet clear in the first semesters. In this situation, Specializations modules from several tracks can be completed in the early phase of the studies. If one track is then chosen, the modules which are not from that track can be assigned to the Free Courses.

Soft Skills

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:

- Modules from the MScTI classified as 'Soft Skill' in the table below,
- Two introductory modules from the Entrepreneurship certificate of UHD (6 CP),
- Courses from the University course program classified as soft skill courses,
- Language courses up to a maximum of 6 CP.

Note: The well established module 'Tools' (page 12) gives student a quick overview and introduction to useful software and methods. The module relies on practical, supervised, on-site participation with no significant homework, so that only 4 CP are justified. This smaller number of 'soft-skill' points also makes it more compatible to other BSc or MSc programs.

Seminar

In the seminar the students deepen their knowledge on a specific topic and present this topic to the other participants in a presentation. In parallel, a 'paper' about the topic is written by all participants. These papers are reviewed by other participants to understand and experience the steps of a real publication process. For the seminar 2 CP are allocated as Soft Skills and 4 CP for the professional contents.

Study Project

This module introduces the student to the work in a chosen research group. The topic is typically on an introductory level so that background knowledge for the specific task can be gained and tools required to complete the task can be learned. The topic can be (but does not have to) an introduction to the subject which will later be treated in the Master Thesis. The

Study Project is completed by a report. As 14 CP are assigned, the work load is roughly 50% of the 3rd semester load, i.e. significant.

Master Thesis

The master thesis spans a duration of 6 months and is usually carried through in the specialization (if one is chosen) in the 4th semester. After handing in the written thesis document, the results are presented in a final, public colloquium which enters into the grading by the reviewers.

Summary

In summary, the following number of modules have to be completed successfully in the various categories in order to reach the required 120 CP:

2 modules from Fundamentals	12 CP
6 modules from Specializations	36 CP
2 modules from Free Courses	12 CP
2 or more modules from Soft Skills	12 CP
Seminar	4(+2) CP
Study Project	14 CP
Master Thesis with final colloquium	30 CP
Sum	120 CP

Mobility Window

Because the students should be present to carry out their Study Project and Master Thesis in a research group in Heidelberg during the 3rd and 4th semester, the best choice for a semester abroad is the 2nd semester. The 12 CP of the Free Courses as well as, for instance, 6 CP in the area of Soft Skills can easily be accomplished abroad because they can be accepted with few restrictions. If the students finds typically 2 modules of 2×6 CP abroad which match the topics in the Specializations (to be agreed upon with the Prüfungsausschuß beforehand), the full 30 CP of the 2nd semester can be obtained abroad.

Modules Overview and Relationships

The following table shows all regular modules of the MScTI and their assignment to the Fundamentals, Soft Skills or one of the three Specializations. The circle ($\circ$) indicates that the module is compulsory in the respective category or track. The same module cannot be counted twice.

Note that additional modules which are held exceptionally for instance by guest lecturers can be accepted and assigned to one of the categories by the Studienkommission.

Module	Fundamentals	Soft Skills	Spec.: Microelec.	Spec.: Em. Comp.	Spec.: Biorobotics	Module Responsible	page
Advanced Computer Architecture	○					NT	10
Control Systems Design	○					LM	11
Tools		×				PF/AS	12
Entrepreneurship		×				CG	-
Components, Circuits & Simulation			○			PF	13
Digital Hardware Description and Verification			○			DK	14
Full Custom VLSI Design			×			PF	15
Advanced Analogue Building Blocks			×			PF	16
Digital Semicustom Design Flow			×			AG	17
Reconfigurable Embedded Systems			×	×	×	DK	18
Applied Quantum Computing				×		MR	19
Quantum Computing Hardware			×	×		MR	20
Performance Essentials for CPUs and GPUs				○		RS	21
GPU Computing (Architecture + Progr.)				○	×	HF	22
Embedded Machine Learning				×		HF	23
CPU Algorithm Design				×		RS	24
GPU Algorithm Design				×		RS	25
Consistency and Coherency				×		HF	26
High Perf. and Distr. Comp.				×		HF	27
Emerging Computing Paradigms			×	×		NT	28
Architecture and CAD for FPGA			×	×		DK	29
Energy Efficient Computing			×	×		DK	30
Robotics					○	LM	31
Biomechanics and Biorobotics 1					○	LM/DH	32
Computational Biomechanics					×	DH	33
Haptics and Human Robot Inter./Reha.					×	LM	34
Biorobotics 2					×	DH	35
Biosignal Processing and Machine Learning					×	AA/MH	35
Seminar							37
Research Project							38
Master Thesis							39

Table 1: Modules in the MScTI and assignment to the various categories. Compulsory modules in a category are labeled with ○.

AA = Dr. Amin Aminifar AG = Dr. Andreas Grübl, AS = Dr. Alexander Schubert, CG = Dr. Christoph Garbe, DH = Prof. Dr. Daniel Häufle, DK = Prof. Dr. Dirk Koch, HF = Prof. Dr. Holger Fröning, LM = Prof. Dr. Lorenzo Masia, MH = Dr. Mostafa Haghi, MR = JProf. Dr. Marko Rančić, NT = Prof. Dr. Nima TaheriNejad, PF = Prof. Dr. Peter Fischer, RS = Prof. Dr. Robert Strzodka.

3 Module Descriptions

The following pages contain the descriptions of all modules offered primarily by the MScTI.

Modules are open to non-MScTI students, as long as sufficient room space and infrastructure (computer seats, lab places, compute resources...) area available. In the *Seminar*, the number of participants is limited by the time available for presentations. MScTI students are accepted with priority. Free slots are available for non-MScTI students. The *Study Project* and the *Master Thesis* are only intended for students of the MScTI.

None of the modules is compulsory in other bachelor or master programs.

An optional inclusion of modules in other study programs is left to these programs.

Temporary deviations from the default semester assignment or of the lecturer may sometimes be required for organizational reasons. Longer term changes will be amendment in updated versions of the document.

In the following module descriptions

- 'ST' is summer term ('Sommersemester'), lectures starting mid April,
- 'WT' is winter term ('Wintersemester'), lectures starting mid October.

Code: MScTI_ACA	**Course Title:** Advanced Computer Architecture		
Module Coordinator: Prof. Dr. Nima TaheriNejad	**Type:** Lecture with exercise		
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** ST

Module Parts and Teaching Methods:
- Lecture (3 h / week)
- Practical exercises with homework (1 h / week)

Objectives: Students...
- By the end of this lecture, the students will be able to:
- list and discuss advantages of and challenges in parallel computing
- describe at least one method of managing performance of multiprocessor systems
- list at least two interconnect technology and a method to select the suitable solution for a system
- discuss various memory systems and their performance figures
- name at least two emerging memory technologies and elaborate on their advantages and disadvantages

Content:
- Processor Architectures
- Parallel Computing
- Programming models & architectures
- Multiprocessor architectures
- Interconnects (incl. Network-on-Chip)
- Caching in Multiprocessors
- Multi-threading
- Dataflow
- Memory & storage
- Emerging memory technologies
- Introduction to emerging computing paradigms (approximate and in-memory computing)
- Heterogeneous Multiprocessor Systems

Prerequisites: none	**Recommended Knowledge:** Basic Computer Architecture

Literature:
- Lecture Notes and Handouts
- A list of sources that will be provided in the course

Testing: Defined by the lecturer before the beginning of the course

Code: MScTI_CSD		Course Title: Control Systems Design	
Module Coordinator: Prof. Dr. Lorenzo Masia		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** WT

Module Parts and Teaching Methods:
- Lecture (2 h / week)
- Practical exercise with homework (2 h / week)

Objectives: Students...
- 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

Content:
- 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:	Recommended Knowledge:
none	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

Testing: Written exam at the end of the semester. Successful participation in the programming exercises is required to be accepted to exam.

Code: MScTI_TOOLS		**Course Title:** Tools	
Module Coordinator: Prof. Dr. Peter Fischer		**Type:** Lecture with exercise	
Credit Points: 4	**Workload:** 120 h	**Teaching Hours:** 4 / week	**Term:** ST

Module Parts and Teaching Methods:

- Lecture with on-site exercises (4 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 give introduction
- are familiar with the concepts of good scientific practice and can apply them to their own scientific work.
- know how to find scientific literature in professional library systems

Content:

- Topics are chosen from a pool of possibilities as a function of interest of students and availability of knowledgeable lecturers. The list of topics is regularly adapted to new developments. Example Topics are listed below:
- Introduction to Linux
- Version control tools (git, svn,..)
- Introduction to python
- Mathematical software (Mathematica)
- Data evaluation and plotting (gnuplot, root)
- 2D & 3D drawing, construction and visualization (PovRay, OpenSCAD, PostScript, pdf)
- styles and templates (powerpoint, word)
- introduction to Latex
- team work
- project planning and management
- Good scientific practice
- literature search, bibliometric measures, open access concepts,...

Prerequisites: none	**Recommended Knowledge:**

Literature:

- announced in lecture per topic

Testing: No testing. Regular attendance required.

Code:	Course Title:
MScTI_CCS	Components, Circuits and Simulation
Module Coordinator:	**Type:**
Prof. Dr. Peter Fischer	Lecture with exercise

Credit Points:	Workload:	Teaching Hours:	Term:
6	180 h	4 / week	ST

Module Parts and Teaching Methods:
- Lecture (2 h / week)
- Practical exercise with homework (2 h / week)

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

Content:
- Diode and transistor operation principle
- Modeling of Diode and MOS, large / small signal models
- Voltage and current sources, Thévenin 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

Prerequisites:	Recommended Knowledge:
none	Introduction to Physics

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)

Testing: Defined by lecturer before beginning of course

Code: MScTI_HDL		**Course Title:** Digital Hardware Description and Verification	
Module Coordinator: Prof. Dr. Dirk Koch		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** WT

Module Parts and Teaching Methods:
- Lecture (2 h / week)
- Practical exercise with homework (2 h / week)

Objectives: Students...
- can describe digital circuits using hardware description languages
- know typical languages like Verilog, System-Verilog, VHDL
- can meaningfully structure a large design into smaller design blocks
- have experience in designing state machines
- can set up simulation environments (test benches) for verification
- know about advanced verification methods
- realise the difference betwwen synthesiable and non-synthesizable constructs

Content:
- Syntax and semantics of Verilog, VHDL, Systemverilog
- Modules and their instantiation, simple and advanced ports
- Structural description, behavioural description
- Description of common blocks (RAM, FIFO, Register file, counter,...)
- Robust description of state machines
- Instantiation of hard macros (in FPGAs or VLSI)
- Simulation, test patterns, test benches,...
- Introduction to UVM

Prerequisites: none	**Recommended Knowledge:**

Literature:
- Defined in lecture

Testing: Defined by lecturer before beginning of course

Code: MScTI_ANADESIGN		**Course Title:** Full Custom VLSI Design	
Module Coordinator: Prof. Dr. Peter Fischer		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** WT

Module Parts and Teaching Methods:
- Lecture (2 h / week)
- Practical exercise with homework (2 h / week)

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.

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

Prerequisites: none	**Recommended Knowledge:** MScTI_CCS

Literature:
- Lecture script available online

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

Code: MScTI_AABB	**Course Title:** Advanced Analogue Building Blocks		
Module Coordinator: Prof. Dr. Peter Fischer	**Type:** Lecture with exercise		
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** WT

Module Parts and Teaching Methods:

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

Objectives: Students...

- know a variety of advanced circuit topologies
- get a deeper qualitative and quantitative understanding of the behavior of analogue circuits,
- can define performance measures, extract them from simulations and optimize circuit parameters
- are able to chose an appropriate circuit for a given problem

Content:

- Circuit families and topics that will be treated in some detail will be chosen from:
- Noise of components and circuits
- Charge amplifier with feedback and filter
- Logic Families (NMOS, Dynamic, Pass Gate, Differential)
- PLL
- 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*

Testing: Oral Examination

<table>
<tr><td colspan="2">Code:
MScTI_DIGDF</td><td colspan="2">Course Title:
Digital Semicustom Design Flow</td></tr>
<tr><td colspan="2">Module Coordinator:
Prof. Dr. Dirk Koch, A. Grübel</td><td colspan="2">Type:
Lecture with exercise, lab, project</td></tr>
<tr><td>Credit Points:
6</td><td>Workload:
180 h</td><td>Teaching Hours:
4 / week</td><td>Term:
ST</td></tr>
<tr><td colspan="4">Module Parts and Teaching Methods:
• Lecture (2 h / week)
• Practical and practical chip project (2 h / week)</td></tr>
<tr><td colspan="4">Objectives: 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.</td></tr>
<tr><td colspan="4">Content:
• 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.</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="2">Recommended Knowledge:
deeper knowledge of Digital Hardware Design</td></tr>
<tr><td colspan="4">Literature:
• A reading list will be provided in the lecture</td></tr>
<tr><td colspan="4">Testing: 30’ oral exam at the end of the semester. At least 50% of the exercises and the chip project must be passed.</td></tr>
</table>

Code: MScTI_RES		Course Title: Reconfigurable Embedded Systems	
Module Coordinator: Prof. Dr. Dirk Koch		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** ST

Module Parts and Teaching Methods:

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

Objectives: Students...

- understand elements, properties and requirements of embedded systems
- get basic understanding of reconfigurable architectures
- learn application design methodologies for microprocessors and FPGAs
- 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,
- implement and program a sample embedded FPGA platform

Content:

- 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
- High-level FPGA design methodologies (incl. HLS)
- CAD Tool basics: how is code translated into an FPGA configuration?
- System-on-Chip architecture – controller, buses and peripherals
- HW/SW co-design
- Embedded system software (stand-alone and real-time kernels)
- Reconfigurable custom Instruction Set Architecture (ISA) extensions

Prerequisites: none	**Recommended Knowledge:** none

Literature:

- Provided by the lecturer or on lecture web site

Testing: Defined by lecturer before beginning of course

Code: MScTI_AQC		**Course Title:** Applied Quantum Computing	
Module Coordinator: JProf. Dr. Marko Rančić		**Type:** Lecture with exercises	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** ST

Module Parts and Teaching Methods:

- Lecture (2-3 h / week)
- Practical exercise with homework (1-2 h / week)

Objectives: Students...

- Understand the benefits which quantum computing brings to classical computing
- Understand the main bottlenecks of modern quantum computing
- Name most common approaches to quantum computing
- Get extensive hands on experience and theoretical understanding of main quantum computing algorithms

Content:

- Introduction to quantum mechanics
- Introduction to quantum computing
- Quantum noise
- Quantum computing approaches: Universal Quantum computing, NISQ Quantum computing and Quantum Annealing,
- Main architectures: Superconducting, Photonic, Trapped Ions, Spin qubits
- Universal quantum computing algorithms: (Shor's / Grover's /HHL algorithms, quantum phase estimation)
- Noisy-intermediate scale algorithms (Variational Quantum Eigensolver, Imaginary time evolution, Quantum Approximate Optimization algorithm)
- Quantum annealing

Prerequisites: none	**Recommended Knowledge:** Basic Computer Architecture

Literature:

- Lecture Notes and Handouts
- A list of other sources will be provided in the course

Testing: Defined by lecturer before beginning of course

Code: MScTI_QCH		**Course Title:** Quantum Computing Hardware	
Module Coordinator: JProf. Dr. Marko Rančić		**Type:** Lecture with exercises	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** WT

Module Parts and Teaching Methods:
- Lecture (2-3 h / week)
- Practical exercise with homework (1-2 h / week)

Objectives: Students...
- Understand the basic hardware approaches to quantum computing
- Understand the basis of cryogenic technology involved in quantum computing
- Understand quantum computing hardware roadmaps

Content:
- Introduction to quantum mechanics
- Introduction to quantum computing
- Selected topics in condensed matter and atomic physics
- Superconducting qubits (transmons)
- Photonic qubits
- Trapped ion qubits
- Cold atoms in optical lattices
- Spin Qubits
- Nitrogen vacancy centers in diamond qubits
- Topological Qubits
- Quantum Error correction: Toric code
- Quantum Error correction: Wen plaquette model

Prerequisites: none	**Recommended Knowledge:** Basic Quantum Mechanics

Literature:
- Lecture Notes and Handouts
- A list of further sources will be provided at the beginning of the course

Testing: Defined by lecturer before beginning of course

Code: MScTI_PERF	**Course Title:** Performance Essentials for CPUs and GPUs		
Module Coordinator: Prof. Dr. Robert Strzodka	**Type:** Lecture with exercise		
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** ST

Module Parts and Teaching Methods:
- Lecture 2 h / week
- Exercise 2 h / week plus homework

Objectives: Students...
- understand performance implications of code constructs,
- design better programs following guidelines of an effective programming style,
- know how to program CPUs and GPUs with the same source code,
- can select efficient parallel algorithms from existing libraries.

Content:
- Most important dos and donts of efficient code
- Unified high level programming of CPUs and GPUs
- Efficient allocation, data access, computation
- Clear and effective programming style
- Parallel data structures and algorithms
- Libraries for dense and sparse linear algebra
- Specialized libraries

Prerequisites: none	**Recommended Knowledge:** Programming experience

Literature:
- Bjarne Stroustrup: *A Tour of C++* (2rd ed, Addison-Wesley, 2022)
- Additional material will be provided through the learning platform

Testing: 50% of points from the exercises are required for participation in the oral or written exam.

Code: MScTI_CEGPU		**Course Title:** GPU Computing - Architecture + Programming	
Module Coordinator: Prof. Dr. Holger Fröning		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** WT

Module Parts and Teaching Methods:
- Lecture (2 h / week)
- Practical exercise with homework (2 h / week)

Objectives: Students...
- know GPU architectures and associated design decisions,
- know the factors that determine the performance of GPU programs, and are able to program GPUs to solve computing problems,
- 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,
- are capable to decide when accelerators like GPUs are suitable for a given computing problem.

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

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

Literature:
- N.W. Wilt: *The CUDA Handbook: A Comprehensive Guide to GPU Programming* (Addison-Wesley, 2013)
- D.B. Kirk, W.W. Hwu: *Programming Massively Parallel Processors* (Morgan- Kaufmann, 2010)
- T.G. Mattson, B.A. Sanders, B.L. Massingill: *Parallel Patterns for Parallel Programming* (Addison Wesley, 2004)
- J.L. Hennessy, D.A. Patterson: *Computer Architecture: A Quantitative Approach* (Morgan Kaufmann, 2017)

Testing: The module is completed with a graded written or oral exam. project. The final grade of the module is determined by the grade of this exam. The requirements for the assignment of credits follows the regulations in section modalities for exams. Details will be given by the lecturer at the start of the semester.

Code: MScTI_CEEML		**Course Title:** Embedded Machine Learning	
Module Coordinator: Prof. Dr. Holger Fröning		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** ST

Module Parts and Teaching Methods:
- Lecture (2 h / week)
- Practical exercise with homework (2 h / week)

Objectives: Students...
- Know machine learning (ML) basics
- Are familiar with neural network architectures for image, signal, and speech processing
- Can design such model architectures for simple problems
- Understand the computational requirements of such architectures
- Know different processor and system architectures to execute ML models
- Are capable of finding solutions to deploy ML models on resource-constrained processors

Content:
- Basics of ML
- Neural architecture design, including basics of neural networks, automatic differentiation and optimization, regularization and generalization
- Applications including time series and computer vision
- Safe optimizations with regard to SW and HW
- Unsafe optimizations with regard to SW and HW
- Future directions of embedded ML

Prerequisites: none	**Recommended Knowledge:** Computer architecture (e.g. courses “GPU Computing”, “Introduction to HPC”, or ”Advanced Computer Architecture”), Python

Literature:
- I. Goodfellow, Y. Bengio and A. Courville: *Deep Learning* (MIT Press, 2006)
- B. Reagen et.: *Deep Learning for Computer Architects (Synthesis Lectures on Computer Architecture)* (Morgan & Claypool. 2017)
- C. M. Bishop: *Pattern Recognition and Machine Learning (Information Science and Statistics)* (Springer. 2006)

Testing: The module is completed with a graded written or oral exam. project. The final grade of the module is determined by the grade of this exam. The requirements for the assignment of credits follows the regulations in section modalities for exams. Details will be given by the lecturer at the start of the semester.

Code:	Course Title:
MScTI_CPUALG	CPU Algorithm Design

Module Coordinator:
Prof. Dr. Robert Strzodka

Type:
Lecture with exercises and project

Credit Points:	Workload:	Teaching Hours:	Term:
6	180 h	4 / week	WT

Module Parts and Teaching Methods:
- Lecture 2 h / week
- Exercise 2 h / week plus homework in first half of semester
- Project 2 h / week plus homework in second half of semester

Objectives: Students...
- are able to exploit all the parallelism and bandwidth in large CPUs,
- can make design decisions depending on tradeoffs in parallel algorithms,
- know how to apply advanced transformations for higher parallelism and data locality,
- understand parallel performance models and scalability.

Content:
- Parallel performance models and scalability
- 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

Prerequisites:
Familiarity with C++

Recommended Knowledge:
C++17 and STL (e.g. MScTI_PERF)

Literature:
- Michael McCool, Arch Robison, James Reinders: Structured Parallel Programming, Morgan Kaufmann, 2012
- Michael Voss, Rafael Asenjo, James Reinders: ProTBB, Springer Nature, 2019
- Additional material will be provided through the learning platform

Testing: 50% of points from the exercises are required for participation in the project exam, which consists of a software design with documentation and a presentation of results. Alternatively to the project exam, an oral (20-30 min) exam may be announced by the Module Coordinator.

Code: MScTI_GPUALG	**Course Title:** GPU Algorithm Design		
Module Coordinator: Prof. Dr. Robert Strzodka	**Type:** Lecture with exercises and project		
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** WT

Module Parts and Teaching Methods:

- Lecture 2 h / week
- Exercise 2 h / week plus homework in first half of semester
- Project 2 h / week plus homework in second half of semester

Objectives: Students...

- are able to exploit all the parallelism and bandwidth in large CPUs,
- can make design decisions depending on tradeoffs in parallel algorithms,
- know how to apply advanced transformations for higher parallelism and data locality,
- understand how to balance numerical efficiency and parallel efficiency.

Content:

- 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

Prerequisites: Familiarity with C++ and CUDA	**Recommended Knowledge:** C++17 and STL, GPU and CUDA

Literature:

- David B. Kirk, Wen-mei W. Hwu: *Programming Massively Parallel Processors* (3rd ed, Morgan Kaufmann, 2017)
- Additional material will be provided through the learning platform

Testing: 50% of points from the exercises are required for participation in the project exam, which consists of a software design with documentation and a presentation of results. Alternatively to the project exam, an oral (20-30 min) exam may be announced by the Module Coordinator.

<table>
<tr><td colspan="2">Code:
MScTI_CECOCO</td><td colspan="2">Course Title:
Consistency and Coherence</td></tr>
<tr><td colspan="2">Module Coordinator:
Prof. Dr. Holger Fröning</td><td colspan="2">Type:
Lecture with exercise</td></tr>
<tr><td>Credit Points:
6</td><td>Workload:
180 h</td><td>Teaching Hours:
4 / week</td><td>Term:
ST</td></tr>
<tr><td colspan="4">Module Parts and Teaching Methods:
• Lecture (2 h / week)
• Practical exercise with homework (2 h / week)</td></tr>
<tr><td colspan="4">Objectives: Students...
• know coherence and consistency principles of parallel architectures,
• know design techniques for scalable synchronization and communication primitives,
• are familiar with advanced concepts like transactional memory, relaxed consistency, and scalable coherence
• know how to design and optimize complex parallel code for particular communication 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="4">Content:
• 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 architectures</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="2">Recommended Knowledge:
Computer architecture (e.g. “GPU Computing”, “Introduction to HPC”, or ”Advanced Computer Architecture),
parallel programming principles, C, C++, OS basics</td></tr>
<tr><td colspan="4">Literature:
• M. Herlihy, N. Shavit: The Art of Multiprocessor Programming (Morgan Kaufmann, 2012)
• J.L. Hennessy, D.A. Patterson: Computer Architecture: A Quantitative Approach (Morgan Kaufmann, 2017)</td></tr>
<tr><td colspan="4">Testing: The module is completed with a graded written or oral exam. project. Details will be given by the lecturer at the start of the semester.</td></tr>
</table>

Code: MScTI_CEHPDC		**Course Title:** High-Performance and Distributed Computing	
Module Coordinator: Prof. Dr. Holger Fröning		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** WT

Module Parts and Teaching Methods:

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

Objectives: Students...

- know message passing, cluster computing architectures, 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 or energy, and scalability in terms of time and capacity

Content:

- HPC architectures and message passing
- Parallel algorithm design and Message Passing Interface (MPI)
- MPI internals
- Characterization and benchmarking
- Short introduction to parallel training of machine learning models
- Practical problems and their solutions

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

Literature:

- G. Hager and G. Wellein: *Introduction to High Performance Computing for Scientists and Engineers* (Taylor & Francis Inc, 2011)
- I. Goodfellow, Y. Bengio and A. Courville: *Deep Learning* (MIT Press, 2006)

Testing: The module is completed with a graded written or oral exam. Details will be given by the lecturer at the start of the semester.

Code: MScTI_ECP	**Course Title:** Emerging Computing Paradigms		
Module Coordinator: Prof. Dr. Nima TaheriNejad	**Type:** Lecture with exercise		
Credit Points: 8	**Workload:** 240 h	**Teaching Hours:** 4 / week	**Term:** ???

Module Parts and Teaching Methods:
- Lecture (4 h / week)
- Practical exercises with homework (2 h / week)

Objectives: Students...
- By the end of this lecture, the students will be able to:
- explain the fundamental principle of neuromorphic computing
- name and describe at least one approximate computing solution at software, architecture, and circuit level.
- discern whether approximate computing may be used for a specific application
- describe the fundamental advantage of in-memory computing
- list five emerging memory technologies
- solve (evaluate the state of) memristive circuits
- name at least one advantage and disadvantage of stochastic computing
- describe the principle of quantum computing
- name at least one advantage and disadvantage of optical computing

Content:
- Challenges in computing
- ML accelerators and neuromorphic computing
- Approximate computing
- In-memory computing
- Emerging memory technologies
- Memristive Computing
- Stochastic computing
- Quantum computing
- Optical computing

Prerequisites: None	**Recommended Knowledge:** Basic Computer Architecture. Ideally: CCS, HDL, VLSI, ACA, and Embedded Machine Learning

Literature:
- A list of sources that will be provided in the course

Testing: Defined by the lecturer before the beginning of the course

Code: MScTI_ACF		**Course Title:** Architecture and CAD for FPGAs	
Module Coordinator: Prof. Dr. Dirk Koch		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** WT

Module Parts and Teaching Methods:
- Lecture (2 h / week), Practical exercise with homework (2 h / week)

Objectives: Students...
- The overall objective of Architecture and CAD for FPGAs – ACF is to provide a deep understanding of the ecosystem that is required to make user logic working on an FPGA. This includes the FPGA hardware itself but also the tools that are required to map user logic onto an FPGA. In detail the course unit has the objectives:
- list and explain advanced components of FPGA devices
- list and explain architectural details of FPGA devices
- understand the design factors that drive FPGA development
- understand the ecosystem that is required to build FPGA chips as well as the CAD tools to program them
- designing own customized FPGAs, including the logic and special primitives, the routing fabric and the configuration logic
- understand and program tools for logic synthesis, technology mapping and place and route,
- understand test strategies of FPGAs, including factory tests
- understand Hardware verification, including the internals of a simulator

Content:
- Hardware programmability: reconfigurable logic and routing structures
- Dedicated blocks for memory and arithmetic (why and how)
- Understanding FPGA application requirements
- Performace tuning: how to improve area, speed and power consumption
- Physical implementations of FPGAs and embedded FPGAs versus other ASICs
- Logic synthesis and technology mapping (translate user logic into FPGA primitives)
- Place & Route (of user circuits)
- Bitstream Assembly, Partial reconfiguration
- Factory test and chip characterization (how do we ensure that our fabricated chip works as expected)
- Simulation: simulation techniques and accuracy versus simulation speed
- Robustness and security in FPGAs
- Outlook emerging technologies: resistive RAM, phase change memory, spintronics for reconfigurable computing

Prerequisites: none	**Recommended Knowledge:** Hardware design (Verilog or VHDL), Algorithms and data structures, FPGA Basics

Literature:
- An updated list of papers will be provided in the classes

Testing: Defined by lecturer before beginning of course

Code: MScTI_EEC		**Course Title:** Energy Efficient Computing	
Module Coordinator: Prof. Dr. Dirk Koch		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** ???

Module Parts and Teaching Methods:
- Lecture (2 h / week)
- Practical exercise with homework (2 h / week)

Objectives: Students...
- Energy-efficiency is the probably most important objective in virtually all computing systems. It not only enables the mobile revolution, it is also key to cramp billions of transistors into a single processor chip, and is consequently a performance driver. Moreover, embedded systems usually have limited power envelopes and global warming requires us to rethink our computing systems. The unit has the following objectives:
- understand the importance and impact of energy efficiency in computing systems
- learn the sources of energy usage in a computing system
- understand the similarities and differences of computing in datacenters and in embedded systems
- understand and evaluate hardware and software technologies and methods for building energy efficient systems
- develop and implement high performance and energy efficient code

Content:
- The need for energy efficient computing
- Impact of technology and system-level design choices on energy-efficiency
- Design factors impacting cost, performance and energy-efficiency
- The impact of memory on performance and energy-efficiency
- Performance optimization techniques, including frequency voltage scaling and SIMD and multi-core processing
- Specialized hardware and acceleration for performance and more energy-efficient computing
- Existing (e.g., chiplets and 3D integration) and emerging technologies for energy efficient computing
- Performing design-space exploration
- Programming and evaluation of different performance and energy-efficiency tuning techniques

Prerequisites: none	**Recommended Knowledge:** none

Literature:
- to be defined in lecture or on web page

Testing: Defined by lecturer before beginning of course

Code: MScTI_ROB		**Course Title:** Robotics - Kinematics, Dynamics and Control	
Module Coordinator: Prof. Dr. Lorenzo Masia		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** WT

Module Parts and Teaching Methods:
- Lecture (2 h / week)
- Exercises (2 h / week)

Objectives: 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

Content:
- 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: none	**Recommended Knowledge:** 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)

Testing: Written exam at the end of the semester. Successful participation in the exercises is required to be accepted to exam.

Code: MScTI_BIOM1		**Course Title:** Biomechanics and Biorobotics 1	
Module Coordinator: Prof. Dr. Lorenzo Masia		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** ST

Module Parts and Teaching Methods:

- Lecture (2 h / week) with exercises (2 h / week), Practical (block)

Objectives: 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.

Content:

- Physiological basics of the human and animal bodies, 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

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 Contro*
- 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)

Testing: Successful completion of biomechanical lab project with presentation and report

Code: MScTI_COMPBIOM		**Course Title:** Computational Biomechanics	
Module Coordinator: Prof. Dr. Daniel Häufle		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** WT

Module Parts and Teaching Methods:
- Lecture (2 h / week) with seminar/exercises (2 h / week)

Objectives: Students...
- can repeat the steps necessary to derive equations of motion for rigid body systems
- can solve simple examples of biomechanical models
- can name the input-output relations of muscle models
- can build a model of a muscle-driven biomechanical system
- can run and analyse forward-dynamic simulations
- can implement low-level reflex models

Content:
- Simulating rigid body dynamics
- Muscle models
  - Hill-type model development
  - Limitations and recent additions
- Muscle/tendon routing
  - via-point and via-elipse method
  - wrapping surfaces method
- Neuron models
- Muscle synergies
- Details models of body regions: legs, arms, hand, spine

Prerequisites: Biomechanics 1	**Recommended Knowledge:**

Literature:
- Will be suggested in lecture

Testing: Defined by lecturer before beginning of course

Code: MScTI_HAPTICS		Course Title: Haptics and Human Robot Interaction / Rehabilitation	
Module Coordinator: Prof. Dr. Lorenzo Masia		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** ST

Module Parts and Teaching Methods:
- Lecture (2 h / week), Programming Exercises (2 h / week)

Objectives: 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.

Content:
- 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: Matlab/Simulink, MScTI_Robotics, CSD

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 Module Coordinator

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

Code: MScTI_BIOROB2		Course Title: Biorobotics 2	
Module Coordinator: Prof. Dr. Daniel Häufle		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** ST

Module Parts and Teaching Methods:
- Lecture (2 h / week) with seminar/exercises (2 h / week)

Objectives: Students...
- are able to list and explain key characteristics of animal-inspired robotic systems
- are able to link biomechanical concepts to biorobotic design questions
- are able to explain differences between classical robotic concepts and bio-inspired approaches
- can implement basic biorobotics concepts in hardware demonstrators
- are able to present key biorobotics concepts in a scientific manner

Content:
- Concepts of animal locomotion
  - standing, walking, passive dynamic walkers
  - running, spring mass/SLIP model, catapult mechanism
  - hopping, jumping, raibert controller
  - biomorphology, birdbot etc.
- Comparison to classic robotic concepts, ZMP...
- Bio-inspired actuation systems
  - Virtual elasticity and damping, impedance control,
  - SEA's, PEA's,
  - Artificial muscles
- Soft robotics

Prerequisites: Robotics 1	**Recommended Knowledge:**

Literature:
- Will be suggested in lecture

Testing: Defined by lecturer before beginning of course

Code: MScTI_BIOSIG		**Course Title:** Biosignal Processing and Machine Learning	
Module Coordinator: Dr. Mostafa Haghi and Dr. Amin Aminifar		**Type:** Lecture with exercise	
Credit Points: 6	**Workload:** 180 h	**Teaching Hours:** 4 / week	**Term:** ST

Module Parts and Teaching Methods:

- Lecture (2 h / week) with seminar/exercises (2 h / week)

Objectives: Students...

- understand and describe the most contributing biosignals in biomedical applications
- perform the preprocessing and processing of biosignals such as electrocardiogram
- describe and implement the biosignal processing techniques such as discrete wavelet transform to adjust with the frequency boundary of cardiorespiratory parameters
- apply the biosignal processing techniques in cardiac abnormality detection
- identify and distinguish the well-known biosignal data formats
- explain the fundamental concepts in machine learning
- implement and use deep learning for medical applications
- describe at least one machine learning solution for addressing noise concern in biosignal processing
- apply machine learning in the context of biosignal processing

Content:

- Introduction to biosignals and signal processing (general briefing)
- Signal generation and improvement
- Signal visualization
- Basic of R wave and R wave detection in electrocardiogram
- Computer Aided Detection (CAD) – biosignal, Atrial Fibrilation detection
- Data formats for biosignals
- Introduction to the application of machine learning in the biomedical/health domain
- Classical machine learning in the biomedical domain
- Deep learning in the biomedical domain
- Reinforcement learning in the biomedical domain
- How to treat noisy data using machine learning?
- Current and future challenges for machine learning in biomedical applications

Prerequisites:	**Recommended Knowledge:** Basic Calculus and Algebra, Algorithm, and Python Programmin

Literature:

- Will be suggested in lecture

Testing: Defined by lecturer before beginning of course

<table>
<tr><td colspan="2">Code:
MScTI_SEMINAR</td><td colspan="2">Course Title:
Seminar</td></tr>
<tr><td colspan="2">Module Coordinator:
All Groups</td><td colspan="2">Type:
Seminar</td></tr>
<tr><td>Credit Points:
4+2</td><td>Workload:
180 h</td><td>Teaching Hours:
Block</td><td>Term:
ST/ WT</td></tr>
<tr><td colspan="4">Module Parts and Teaching Methods:
• Presentation
• Paper
• Review of paper of other participants</td></tr>
<tr><td colspan="4">Objectives: Students...
• can search literature for a specific subject,
• can select and structure material for a presentation,
• can make slides for a presentation,
• can give a scientific presentation,
• learn to summarize a topic in a publication-type writeup,
• are familiar with the typical reviewing process of publications.</td></tr>
<tr><td colspan="4">Content:
• Literature research
• Preparation of presentation
• Oral Presentation (≈ 45 Minutes)
• Preparation of a short summary 'paper' (≈ 10 pages)
• reviewing process of several other papers
• Active participation in other student's presentations and discussion</td></tr>
<tr><td colspan="2">Prerequisites:
none</td><td colspan="2">Recommended Knowledge:
Initial knowledge in chosen field</td></tr>
<tr><td colspan="4">Literature:
• Provided by supervisor and by own literature search</td></tr>
<tr><td colspan="4">Testing: Presentation, paper, review, regular active participation</td></tr>
</table>

<table>
<tr><td colspan="2">Code:
MScTI_SA</td><td colspan="2">Course Title:
Study Project</td></tr>
<tr><td colspan="2">Module Coordinator:
All Groups</td><td colspan="2">Type:
Practice Course</td></tr>
<tr><td>Credit Points:
14</td><td>Workload:
420 h</td><td>Teaching Hours:
n.a.</td><td>Term:
ST/ WT</td></tr>
<tr><td colspan="4">Module Parts and Teaching Methods:
• Practical Course</td></tr>
<tr><td colspan="4">Objectives: Students...
• can 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="4">Content:
• 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="2">Recommended Knowledge:
Knowledge in research field</td></tr>
<tr><td colspan="4">Literature:
• Depending on subject, provided by supervisor</td></tr>
<tr><td colspan="4">Testing: Written Report</td></tr>
</table>

<table>
<tr><td colspan="2">Code:
MScTI_THESIS</td><td colspan="2">Course Title:
Master Thesis</td></tr>
<tr><td colspan="2">Module Coordinator:
All Groups</td><td colspan="2">Type:
Practice Course</td></tr>
<tr><td>Credit Points:
30</td><td>Workload:
900 h</td><td>Teaching Hours:
n.a.</td><td>Term:
ST/ WT</td></tr>
<tr><td colspan="4">Module Parts and Teaching Methods:
• Master Thesis</td></tr>
<tr><td colspan="4">Objectives: Students...
• 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="4">Content:
• 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="2">Recommended Knowledge:
Knowledge in research field</td></tr>
<tr><td colspan="4">Literature:
• Depending on subject, provided by supervisor</td></tr>
<tr><td colspan="4">Testing: Written Thesis, Colloquium</td></tr>
</table>