Funded by
the European Union

Joint Education for Advanced Chip Design in Europe (Edu4Chip)

Deliverable Report D4.1 Module Descriptions

Author(s)	Ahmed Hemani (KTH), Saul Rodriguez Duenas (KTH), Per-Erik Hellström (KTH), Michael Pehl (TUM), Pengcheng Xu (TUM), Robert Wille (TUM), Hussam Amrouch (TUM), Yushen Zhng (TUM), Stefan Pechmannn (TUM), Luca Pezzarossa (DTU), Matthias Bo Stuart (DTU), Martin Schoeberl (DTU), Jean-Max Dutertre (IMT), Jean-Baptiste Rigaud (IMT), Olivier Potin (IMT), Raphael Viera (IMT), Alexandre Menu (IMT),Agnès Roussy (IMT), Timo Hämäläinen (TAU), Arto Oinonen (TAU), Sakari Lahti (TAU)
Date:	September 30, 2025
Version:	1.0
Due Date:	September 30, 2025
Dissemination Level:	Public
Work Package No:	4
Deliverable Number:	D4.1
Lead Beneficiary:	KTH

Disclaimer: *Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Health and Digital Executive Agency (HADEA). Neither the European Union nor the granting authority can be held responsible for them.*

Abstract

The Edu4Chip partner universities ‐ the Technical University of Denmark, the KTH Royal Institute of Technology, the École des Mines de Saint-Étienne of IMT, Tampere University, and the Technical University of Munich ‐ aim to setup aligned Master programmes and specialization directions in the domain of chip design. This effort is funded by the European Union through the program Edu4Chip. This document provides a brief description of the modules or courses that the partner universities plan to give as part of the Chip Design Curriculum to be introduced from summer and fall 2025.

1 Introduction

This deliverable presents the module descriptions as they stood at the time of writing, offering an overview of the newly developed or revised courses within the Edu4Chip Master's program/track at the partner universities. As academic programs are subject to continuous development, the content and focus of individual courses may evolve, and the descriptions provided here should therefore be regarded as indicative. This document is to be read in conjunction with Deliverable D2.1 Program Concept and Deliverable D2.2 Curriculum, which it complements.

2 Modules at DTU

DTU is integrating the Edu4Chip initiative into its existing MSc program in Computer Science and Engineering, with a primary focus on the Digital Systems specialization. This integration includes both adapting existing courses and developing new ones to strengthen the chip design curriculum. While the Digital Systems specialization is the main pathway for Edu4Chip activities, participation is open to all students enrolled in the MSc program, as specializations at DTU are considered recommendations rather than mandatory tracks.
The following presents the description of the adapted new courses. These descriptions are adapted from the official DTU course database for the academic year 2025/2026. The course database contents result from an approval and quality assurance process involving the course responsible, which beside the co-authors of this document at DTU include Assoc. Prof. Flemming Stassen, Asst. Prof. Ole Richter, and Prof. Paul Pop.

2.1 Computer Architecture and Engineering

Course Credits: 5

Course Objectives and Contents:

The course provides the participants with in-depth knowledge of computer organization, and an associated understanding of issues that influence the runtime of programs as well as of the interplay between hardware and software.

The content covered by the course includes instruction sets, organization of the processor in a modern RISC computer (control unit, datapath, pipelining), the relationship between processor organization and instruction set, buses and I/O units, relation between processor organization and operating system, memory hierarchies (including caches and virtual memory), and novel architectures which exploit different forms of parallelism.

In parallel with the lectures, students work on problems and exercises, and towards the end of the course, students work on a final small project. For example, the project can consist of analyzing and optimizing the runtime of a given program by considering the processor pipeline structure and memory hierarchy using a simulator.

Today, processors are used in all forms of electronic equipment, and hence the course is relevant for software developers as well as (electrical) engineers designing embedded systems. For these cases, it is often important to make efficient use of the resources of the processor in order to optimize performance, cost, or energy consumption.

Learning Outcomes:

- Describe the internal structure of a processor, including pipeline, cache memory, memory hierarchy, virtual memory, and buses.
- Explain the function of these elements and explain how they influence the runtime of programs, and the complexity of the underlying hardware (area, speed, and energy).
- Calculate and optimize the runtime of programs: both small program fragments and large programs for which key parameters like cache-miss rates have been established.

- Explain the throughput and latency parameters and the relationship between them.
- Explain how hardware (e.g., technology and architecture) and software (e.g., instruction set, compiler, and operating system) interact, and how the interface between the two is a key element in all systems.
- Give examples of trade-offs and optimizations involving both hardware and software.
- Explain the instruction set architecture of a typical processor.
- Document project work in the form of a technical report addressing engineers in the field. The report must conform to usual requirements for form, content, and level of abstraction.

Prerequisites (recommended):

- Basic programming skills including fundamentals of algorithms and data structures corresponding to, e.g., DTU course 02100 - Introductory Programming and Software Technology.
- Introductory knowledge of computing systems including Boolean algebra, binary number representation, combinatorial circuits, and design of simple digital circuits corresponding, to e.g., DTU courses 02132 - Computer Systems or 02135 - Introduction to Cyber Systems.
- Introductory knowledge of digital electronics including Boolean algebra, binary number representation, combinatorial circuits, and design of simple digital circuits corresponding to, e.g., DTU courses 02132 - Computer Systems or 02135 - Introduction to Cyber Systems or 02139 – Digital Electronics 2.

2.2 Hardware/Software Codesign

Course Credits: 5

Course Objectives and Contents:

The course covers codesign methods for various types of embedded systems, including System-on-Chip (SoC), Multi-Processor System-on-Chip (MPSoC), and introduces system-of-systems, also presenting chiplet-based systems. The relation of codesign to quality-management systems is introduced, as well as real-time embedded systems and the worst-case execution time metric. Other metrics, such as latency, throughput, area, and power consumption, are also presented. The concept of Pareto-optimality is introduced and used to evaluate points in the design space against each other. Principles of multivariable optimization are presented and used in the design-space exploration process.

Students will learn aspects of system design and system engineering in relation to hardware and software elements. Both document- and model-based systems engineering principles are covered with an in-depth focus on HW/SW codesign. The various aspects of a codesign process are introduced and explored through exercises and project work. This includes topics such as hardware/software design-space exploration and high-level synthesis.

Learning Outcomes:

- Describe the background and principles of HW/SW codesign.
- Contrast HW/SW codesign with other design methodologies.
- Describe the components of an HW/SW codesign process.
- Apply codesign techniques to develop a simple embedded system.
- Translate a written specification to a system-level model.
- Perform design-space exploration using optimization strategies.
- Design an embedded system with both hardware and software components.
- Evaluate the trade-offs represented by different points in the design space.

Prerequisites (recommended):

- Basic programming skills including fundamentals of algorithms and data structures corresponding to, e.g., DTU course 02100 - Introductory Programming and Software Technology.
- Knowledge of computing systems including the memory hierarchy, memory mapped devices, microcontrollers and fundamental computer architecture corresponding to, e.g. DTU courses 02155 – Computer Architecture and Engineering, 02132 – Computer Systems
- Knowledge of digital electronics including Boolean algebra, binary number representation, combinatorial circuits, and design of simple digital circuits corresponding to DTU course 02139 – Digital Electronics 2.
- Knowledge of a hardware description language such as VHDL, Verilog, SystemVerilog, or Chisel corresponding to ,e.g., DTU courses 02138 - Digital Electronics 1, 02139 - Digital Electronics 2, or 02203 - Design of Digital Systems.

2.3 Research Topics in Computer Architecture

Course Credits: 5

Course Objectives and Contents:

Computer architecture is a fast-changing and dynamic research and development field. The objective of the course is to provide knowledge of selected advanced topics in computer architecture and to let participants practice investigating topics in the literature and producing oral and written presentations that distill the essential features of a topic based on a literature study.

The topics considered in this course will change from year to year. Typical topics include instruction-level parallelism, chip-multiprocessing, network-on-chip, time-predictable computer architectures, AI accelerators, hardware for security, and approximate computing.

Students will work in groups of two or three on a hardware or software lab project. A hardware project can be building a pipelined design in an FPGA. Software projects will explore embedded chip-multiprocessor programming and programming models. Besides a working solution, students will prepare a conference-style paper on their project. The project and paper will be presented in class. Exceptional projects may lead to publications in computer architecture conferences.

Course literature consists of: *J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, current edition, Morgan Kaufmann Publishing Co., Menlo Park, CA.* In addition to the textbook, this course includes a number of readings from research papers.

Learning Outcomes:

- Explain advanced computer architecture aspects.
- Search the literature for relevant information on recent advances in computer architecture.
- Evaluate the quality and reliability of information on computer architecture found from various information sources.
- Explain the architecture and limitations of chip-multiprocessing.
- Explain network-on-chip architectures.
- Explain time-predictable computer architecture.
- Evaluate a scientific paper (from conference or journal).
- Prepare a conference-style paper.
- Give a coherent oral presentation of a chosen topic, based on a study of relevant literature.

Prerequisites (recommended):

- Basic programming skills including fundamentals of algorithms and data structures corresponding to, e.g., DTU course 02100 - Introductory Programming and Software Technology.
- Knowledge of computer organization covering processor structure, pipelines, caches, memory hierarchies, virtual memory, buses, instruction set architectures, hardware–software interplay, performance analysis, and optimization trade-offs corresponding to, e.g., DTU course 02155 - Computer Architecture and Engineering.

2.4 Design of Digital Systems

Course Credits: 5

Course Objectives and Contents:

The course enables students to design large digital circuits (e.g., hardware accelerators) in a systematic way, to implement these in FPGA and chip technology using typical CAD tools, as well as analyze and optimize the speed and area of a given digital circuit. The course addresses the design of digital systems at the RTL level, with a main focus on writing efficient RTL-code and on the relationship between HDL-constructs and the corresponding synthesized hardware implementations.

The topics covered by the course include the VHDL/Verilog languages, finite-state machine implementation, pipelining, performance measures (throughput and latency), simulation and synthesis, FPGA technology, design flow (specification, stepwise refinement, simulation, synthesis, implementation, and test), and timing analysis of sequential circuits (including metastability and clock skew).

A sequence of exercises supplements the lectures and provides hands-on experience using VHDL/Verilog and the associated CAD tools. A project consisting of developing a small hardware accelerator is included in the course.

The course literature consists of the textbook: *Pong Chu, "RTL-Hardware Design Using VHDL - Coding for Efficiency, Portability, and Scalability," Wiley, 2006*.

Learning Outcomes:

- Design a digital circuit that performs a given computation as a sequential circuit consisting of a finite state machine and a data-path.
- Systematically pipeline a given circuit and identify errors in a given circuit where pipelining has been attempted.
- Explain the performance parameters throughput and latency (and their relationship) and optimize the performance of a given circuit.
- Analyze the timing-correctness of a given sequential circuit (including circuits suffering from clock skew).
- Explain the internal organization of a typical FPGA component and explain how its resources are best exploited when implementing a given circuit.
- Explain how a given VHDL/Verilog description syntesizes to hardware (i.e., a structure of registers and combinational logic blocks) and estimate the hardware resources used for the implementation. Optimize the circuit by modifying the VHDL/Verilog code.
- Explain metastability and describe how to correctly design a circuit comprising multiple clock domains.
- Write efficient register transfer level VHDL/Verilog code that synthesize to the intended circuit implementation.

- Develop and apply testbenches for simulation and verification of the functional correctness of a digital circuit.
- Design complex digital circuits by stepwise refinement from a specification down to a level from which a hardware implementation can be synthesized.
- Organize and plan the design of a large digital circuit in a group comprising several students.
- Document this work (design, implementation, and test) in the form of a technical report addressing engineers in the field. The report must conform to usual requirements for form, content, and level of abstraction.

Prerequisites (recommended):

- Basic programming skills including fundamentals of algorithms and data structures corresponding to, e.g., DTU course 02100 - Introductory Programming and Software Technology.
- Solid knowledge of basic digital electronics, including MOS transistors and CMOS logic, Boolean algebra, number representation, combinatorial circuits, arithmetic and memory elements, propagation delay, finite state machines, and introductory use of a hardware description language corresponding to, e.g., DTU course 02138 - Digital Electronics 1 and 02139 - Digital Electronics 2.
- Introductory knowledge of computing systems covering processor architecture, instruction set architecture, memory, and performance and energy analysis corresponding to, e.g., DTU course 02132 - Computer Systems or 02135 - Introduction to Cyber Systems.

2.5 Introduction to Chip Design

Course Credits: 5

Course Objectives and Contents:

This course introduces the design of digital integrated circuits, equipping students with theoretical foundations and hands-on experience in modern chip design. The course covers fundamental digital circuit concepts, design methodologies, and the use of open-source tools for chip development.

Students will explore key topics such as transistor-level design, logic synthesis, floorplanning, verification, and System-on-Chip (SoC) integration. The course also emphasizes practical skills by guiding students through a complete chip design flow, leading to a virtual tapeout with potential access to a real tapeout.

The topics include fundamentals of digital circuit design (transistors, logic gates, combinational and sequential circuits), introduction to ASIC design and PDK concepts, memory design (types, organization, register files, and memory macros in open-source PDKs), SoC design (partitioning, floorplanning, interconnects, and networks-on-chip), verification methodologies and continuous integration, open-source tools and design flows

(OpenLane2, Tiny Tapeout, Chisel), RTL synthesis, place-and-route, timing closure, GDSII generation, and an overview of the tapeout process (virtual tapeout with SkyWater PDK, real tapeout with Tiny Tapeout).

The course is divided into lecture-based theoretical sessions and lab-based practical exercises and includes a team-based final project. Students will apply the concepts learned throughout the course and utilize open-source tools for tapeout.

Learning Outcomes:

- Explain the basic operation of semiconductor devices, including transistors, and apply this knowledge to design basic digital circuits.
- Use open-source design tools (OpenLane2, Tiny Tapeout, Chisel) to develop and simulate a digital circuit through a complete design flow.
- Experience the full chip design process, including RTL synthesis, place-and-route, timing closure, and a virtual tapeout, with an opportunity for real tapeout.
- Apply the principles of System-on-Chip (SoC) design, including interconnects, to develop a basic SoC and to evaluate multicore SoC architectures.
- Implement hardware verification techniques, including simulation-based verification and testbench design, to validate a simple SoC.
- Analyze different memory architectures, including register files and memory macros, and design simple memory subsystems.
- Demonstrate an understanding of ASIC design and process design kits (PDKs) by applying standard-cell-based design principles.
- Gain practical experience in the design, simulation, and implementation of a simple SoC, and experience a virtual tapeout process.
- Develop critical thinking and problem-solving skills through the design and analysis of medium complexity digital systems.

Prerequisites (recommended):

- Basic programming skills including fundamentals of algorithms and data structures corresponding to, e.g., DTU course 02100 - Introductory Programming and Software Technology.
- Solid knowledge of basic digital electronics, including MOS transistors and CMOS logic, Boolean algebra, number representation, combinatorial circuits, arithmetic and memory elements, propagation delay, finite state machines, and introductory use of a hardware description language corresponding to, e.g., DTU course 02138 - Digital Electronics 1 and 02139 - Digital Electronics 2.

2.6 VLSI Design

Course Credits: 5

Course Objectives and Contents:

The course enables the students to understand the basics of MOS technology, including the MOS field effect transistor, and to choose a proper production technology with due consideration of technical and economical terms. It provides the students with a complete understanding of the design process for digital systems and enables comprehensive design-space exploration using modeling, evaluation techniques, and strategies targeting both low power and high speed.

The topics covered in the course include MOS technology (masks, design rules and electrical parameters), basic components of MOS circuits (MOSFETs, resistors, capacitors, switches), technology scaling, timing and power dissipation of combinational and sequential components, levels of abstraction, design flow for ASICs, simulation, synthesis, place-and-route, design for low-power (high-level, RT-level, and gate-level techniques), memory families (static, dynamic, flash), and system-level issues (clock distribution, packaging, signal integrity).

Learning Outcomes:

- Explain the basics of MOS technology, including the MOS field effect transistor.
- Explain basic methods and conditions in digital integrated circuit design.
- Describe combinational and sequential digital systems and list their properties and characteristics (timing, area, power dissipation).
- Explain, construct, and analyze parts of a digital system, including interconnections, with due consideration of technical and economical terms.
- Illustrate the main tasks to be performed in the design of a digital system: simulation, logic synthesis and place-and-route.
- Apply suitable techniques to the design of systems for high-speed and low-power.
- Design a digital system, or part of it, based on given specifications and the methods learned.
- Analyze a system design, compare the results with a reference design, and evaluate the system's performance.

Prerequisites (recommended):

- Basic programming skills including fundamentals of algorithms and data structures corresponding to, e.g., DTU course 02100 - Introductory Programming and Software Technology.

- Solid knowledge of basic digital electronics, including MOS transistors and CMOS logic, Boolean algebra, number representation, combinatorial circuits, arithmetic and memory elements, propagation delay, finite state machines, and introductory use of a hardware description language corresponding to, e.g., DTU course 02138 - Digital Electronics 1 and 02139 - Digital Electronics 2.

2.7 Agile Hardware Design

Course Credits: 5

Course Objectives and Contents:

Agile hardware design with Chisel (Constructing Hardware in a Scala Embedded Language) is a specialized course that focuses on using the Chisel hardware description language (HDL) as part of the agile hardware design process. Chisel is a modern hardware description language that enables designers to create digital circuits with concise, high-level code that is easier to write and maintain than traditional HDLs.

The topics covered in the course include an overview of agile design (in general and for hardware), the Chisel HDL (syntax, features, and benefits), agile hardware design process using Chisel (requirements gathering, design, prototyping, testing, and deployment), design patterns and idioms in Chisel (generators, parametrization, and functional abstraction), and verification with ChiselTest (test-driven development, continuous integration, formal verification).

The course includes a hardware design project that uses Chisel and agile methodologies, providing students with practical examples of how to apply these concepts in their work.

Learning Outcomes:

- Explain the principles of agile design and how they apply to hardware development using Chisel.

- Acquire familiarity with the Chisel HDL, including its syntax, features, and benefits and use it to describe hardware.

- Explain and apply the agile hardware design process, including requirements gathering, design, prototyping, testing, and deployment.

- Apply Chisel design patterns and idioms, such as generators, parametrization, and functional abstraction.

- Use Chisel libraries and tools, such as the Rocket Chip generator, the Chisel test harness, and the Verilog emitter.

- Design and implement complex digital circuits using Chisel and agile methodologies.

- Examine and analyze real-world case studies of agile hardware design with Chisel presented as practical examples of how to apply these concepts in professional contexts.

- Write a technical report that documents the project work.

Prerequisites (recommended):

- Basic programming skills including fundamentals of algorithms and data structures corresponding to, e.g., DTU course 02100 - Introductory Programming and Software Technology.

- Solid knowledge of basic digital electronics, including MOS transistors and CMOS logic, Boolean algebra, number representation, combinatorial circuits, arithmetic and memory elements, propagation delay, finite state machines, and introductory use of a hardware description language corresponding to, e.g., DTU course 02138 - Digital Electronics 1 and 02139 - Digital Electronics 2.

- Introductory knowledge of computing systems including processor architecture, instruction set architecture, memory, and performance and energy analysis corresponding to e.g., DTU courses 02132 - Computer Systems or 02135 - Introduction to Cyber Systems.

2.8 Verification of Digital Systems

Course Credits: 5

Course Objectives and Contents:

This course provides knowledge of methods and tools for verifying integrated circuits by covering the principal elements of coverage-driven constrained random verification. This includes mapping design architecture requirements into verification plans, implementing object-oriented testbench environments, creating constrained random test cases with intended random stimuli distribution, implementing functional checks on the RTL as well as the transactional level, building coverage models for measuring verification completeness, and mapping tests, checks, and coverage models to requirements in the verification plan.

The course presents the fundamental theory necessary to understand and implement these principal elements. Students will learn how to use simulation tools to debug the design, the constrained random test cases, as well as the implemented design checks, but also how to use such tools to measure the obtained coverage metrics using the implemented coverage models.

As a practical aspect, students work in small teams on a practical implementation of a testbench in PyUVM using Python for verifying the functionality of an RTL design. In the final project, each student team will integrate their complete PyUVM verification environment comprised of tests, checks, and coverage models. The final project will be evaluated based on the theoretical correctness of each element, as well as the aggregated capability of the verification environment for finding complex RTL design bugs.

By the end of this course, students will have gained both a thorough theoretical understanding and practical experience in employing modern methods for verifying integrated circuits, using PyUVM and Python. Students will become ready to employ the extensive industry-scale environments used to verify modern complex chip designs.

Topics covered in the course include: principles of verification of integrated circuits, constrained random verification, verification planning, transaction-based testbench architecture, coverage models, functional checks, assertions, PyUVM, Python tests, scoreboards, and requirement-centric verification closure.

Learning Outcomes:

- Explain the role of different tools and languages in digital system verification and their application in verification processes.
- Apply the basic theory of constrained random verification to academic and real-world situations, including principles of functional verification of RTL designs, state space exploration, transaction concepts, verification quality, and verification plans.
- Design directed and constrained random test cases, ensuring effective stimuli distribution to verify RTL designs for real-world situations.
- Implement and/or use functional checkers, reference models, scoreboards, and assertions using temporal logic to validate design correctness for real-world situations.
- Develop and apply coverage models and coverage model closure techniques to measure verification completeness.
- Implement testbenches in PyUVM using Python to verify the functionality of an RTL design.
- Use simulation tools to debug RTL designs, test cases, functional checks, and coverage models.
- Analyze the effectiveness of a verification environment by evaluating verification metrics and bug-finding capabilities.
- Integrate different verification components, including tests, checks, and coverage models, into a complete verification environment.
- Assess the final quality of a verification environment based on its ability to detect complex RTL design bugs.

Prerequisites (recommended):

- Knowledge in imperative and object-oriented programming (including programming in C/C++), along with basic concepts in algorithms and data structures corresponding to, e.g., DTU courses 02100 - Introductory Programming and Software Technology and 02132 - Computer Systems.
- Knowledge in digital electronics, including Boolean algebra, binary number representation, combinatorial circuits, design of simple digital circuits using HDLs, simulation tools, timing, FSM-based design, and describing large digital systems at RTL with VHDL/Verilog corresponding to, e.g., DTU courses 02138 - Digital Electronics 1, 02139 - Digital Electronics 2, and 02203 - Design of Digital Systems.

2.9 Test of Digital Systems

Course Credits: 5

Course Objectives and Contents:

Testing is a critical part of the design and manufacturing process for electronic systems, ensuring that circuits function correctly, reliably, and efficiently before deployment. The course covers the fundamental theory for testing of electronic circuits with specific emphasis on the design of highly testable digital integrated circuits. It introduces the testing problem for integrated circuits, enabling the participants to design and test integrated circuits.

Topics covered by the course include physical failure mechanisms and logic fault models, testability analysis (controllability and observability), algorithms for test pattern generation, fault simulation (fault coverage), design for test, boundary scan techniques, test planning in the design phase, test systems.

Learning Outcomes:

- Explain basic methods and concepts in test and design for test of integrated circuits (e.g., stuck-at faults, D-algorithm, built-in self-test).
- Describe what a testable system implies and analyze, whether a digital circuit or parts hereof are testable.
- Analyze smaller parts of a testable digital circuit, using well-known methods for test and design for test.
- Choose and optimize the test resources that are useful under certain circumstances.
- Given a digital circuit, describe the circuits test properties (e.g., overhead, complexity) and suggest and compare different options to improve the test properties.
- Investigate a technical problem and communicate the results of the examination in a clear and precise manner, using a standard format for technical reports.
- Assess different methods to solve a selected technical problem and evaluate their advantages and disadvantages as solutions to the problem.
- Constructively criticise a written report (on an academic problem), provide eventual missing parts in the report, and conclude on the entirety of the report.
- Discuss the methods presented in technical papers and relate these methods to similar findings, indicating similarities and differences.

Prerequisites (recommended):

- Knowledge in imperative and object-oriented programming (including programming in C/C++), along with basic concepts in algorithms and data structures corresponding to, e.g., DTU course 02100 - Introductory Programming and Software Technology and 02132 - Computer Systems.

- Basic knowledge of computer organization covering processor structure, pipelines, caches, memory hierarchies, virtual memory, buses, instruction set architectures, hardware–software interplay, performance analysis, and optimization trade-offs corresponding to, e.g., DTU course 02155 - Computer Architecture and Engineering.

- Knowledge in digital electronics, including Boolean algebra, binary number representation, combinatorial circuits, design of simple digital circuits using HDLs, simulation tools, timing, FSM-based design, and describing large digital systems at RTL with VHDL/Verilog corresponding to, e.g., DTU courses 02138 - Digital Electronics 1, 02139 - Digital Electronics 2, and 02203 - Design of Digital Systems.

- Foundational knowledge of MOS technology and digital IC design covering MOSFETs, technology scaling, timing and power analysis, ASIC/FPGA/SoC design flows, simulation and synthesis, place-and-route, low-power and high-speed design techniques, and system-level considerations such as memory, clocking, and signal integrity corresponding to, e.g., DTU course 02205 - VLSI Design.

3 Modules at IMT

IMT is integrating the Edu4Chip programme concept in Chip Design into its existing ISMIN – Master's Degree in Microelectronics and Computer Science from Mines Saint-Etienne (a French engineering diploma). IMT is also implementing in Edu4Chip a new one-year post-Master programme (a French Master Spécialisé) – Advanced Master in Microelectronics Circuit Design (in french, Mastère Spécialisé Concepteur de Circuits Microélectroniques, or MS CCM).

3.1 Basics of Digital and Analog Electronics, Reminders

Course Credits: 2

Course Objectives and Contents:

This course covers basic knowledge in analog and digital electronics. In the analog part of the course, fundamental components are reviewed and analytical methods are described to predict the behavior of such components in AC and DC modes. In the digital part of the course, students learn to design digital systems with data and control paths. Discussed topics include binary number representation, design of basic digital blocks, timing analysis in combinational logic and finite state machines.

Learning Outcomes: learning the basics of analog and digital circuits.

- Analog models of passive and active components.
- Static circuit analysis using node laws,mesh laws, and Thévenin-Norton models.
- Dynamic circuit analysis (filter, transfer function).
- Representation of binary numbers and Boolean algebra.
- Design of basic combinational blocks.
- Sequential memory and temporal analysis of RTL/Design of finite state machines.

Prerequisites (recommended):

- none

3.2 Semiconductor Physics

Course Credits: 2

Course Objectives and Contents:

Introduction to semiconductor physics. Understanding the behaviour and properties of semiconductors for use in microelectronic devices.

Understanding the physical phenomena related to the properties of semiconductors.

Learning Outcomes:

- Crystallography and energy bands in semiconductors.
- Intrinsic and extrinsic semiconductors, doping.
- Influence of temperature on doping.
- Transport in semiconductors: drift and diffusion.

- Generation and recombination of charges in semiconductors.

Prerequisites (recommended): none.

3.3 Semiconductor Devices

Course Credits: 3

Course Objectives and Contents:

This course provides participants with the understanding of the physical phenomena present in basic microelectronic devices. It aims at familiarize the students with the main basic components in microelectronics in order to better understand how circuits work.

Learning Outcomes:

- Metal/semiconductor interface in transistors.
- Semiconductor/semiconductor interface.
- Metal/insulator/semiconductor interface.
- CMOS transistors.
- FD-SOI transistors.
- Memory gates.

Prerequisites (recommended):

- Course on semiconductor physics.

3.4 Process and Manufacturing

Course Credits: 2

Course Objectives and Contents:

Gain a better understanding of the microelectronics industry's complexity through the perspective of chip manufacturing. Learn the fundamentals of manufacturing and apply them in a course dedicated to fabricating NMOS transistors in a clean room.

Learning Outcomes:

- Presentation of the microelectronics industry, technological developments, clean room concept.
- Photolithography and etching
- Implantation, diffusion.
- Deposit, oxidation.

Prerequisites (recommended):

- Courses on semiconductor physics and devices.

3.5 Digital Design

Course Credits: 4

Course Objectives and Contents:

This module combines theoretical lectures and practical laboratory work to introduce students to the design of digital integrated circuits. The course begins with a review of MOSFET operation and modelling, followed by an analysis of the analog properties of CMOS combinational logic, including static transfer characteristics, dynamic behavior, and power consumption. Students will also study sequential logic elements, and memory hierarchies found in microcontrollers and application processors. The course includes an introduction to standard cell design in CMOS technology. Through hands-on laboratory sessions using the Cadence EDA tools, students will develop practical skills in sizing, designing layouts, and simulating CMOS inverters and standard cells, while also conducting design rule checks (DRC) and layout versus schematic (LVS) verification.

Learning Outcomes: learning the basics of complex digital circuit design.

- Introduction to VLSI circuits. Review of MOS transistors.
- CMOS inverter functioning.
- Combinational gates in CMOS logic.
- Dynamic logic gates/sequential cells.
- Memory technologies and arithmetic operators.
- Lab work: Getting started with the Cadence Virtuoso environment (schematic design, analog simulation).
- Lab work: design of a CMOS inverter (sizing and layout).
- Lab work: design of a standard cell.
- Lab work: introduction to DRC and LVS tools.

Prerequisites (recommended):

- Courses on the fundamentals of digital gates, semiconductor devices, and the process and manufacturing of MOS transistors.

3.6 Analog Design

Course Credits: 4

Course Objectives and Contents:

This course provides a comprehensive introduction to analog integrated circuit design, with a focus on techniques and paradigms used in modern low-power electronic systems. Students will learn to analyze and design circuits intuitively, and then reinforce this understanding with rigorous theoretical study and hands-on laboratory sessions using industry-grade CAD and simulation tools.

Learning Outcomes: identify and explain analog circuits, select appropriate design methodologies, analyse performance trade-offs, evaluate analog circuit architectures, use industrial CAD tools, and follow basic analog design flows. The course consists of 10 modules, each combining theory with practical labs:

- MOS Recap + Basic Amplifiers – MOS models, common-source/gate/follower stages, biasing, gain, swing.

- Advanced Single-Stage Amplifiers – active loads, gain boosting, noise considerations.
- Current Mirrors – simple, Widlar, Wilson, cascodes.
- In-Circuit Current References – simple, cascode, BMR.
- Differential Pairs I – operation, biasing, CMRR.
- Differential Pairs II – active loads, mismatch effects, common-mode feedback.
- Cascode Circuits – cascode amplifiers, gain, wide-swing cascodes.
- Op-Amp Design I – two-stage architecture, slew rate, output swing.
- Op-Amp Design II – compensation, stability, phase margin.
- Layout for Analog ICs – matching, symmetry, guard rings, trade-offs.

Prerequisites (recommended):

- Courses on the fundamentals of electronic circuits, semiconductor devices, and the process and manufacturing of MOS transistors.

3.7 Processor Architecture

Course Credits: 2

Course Objectives and Contents:

This course covers the study of computer architecture and processor design based on the open-source RISC-V instruction set. It includes microarchitecture design choices, data and control dependency management in pipeline architecture, memory organisation and optimised memory access.

Learning Outcomes: Study of the architecture of an open-source RISC-V instruction set and simulate its micro-architectural implementation.

- RISC-V instruction set and single-cycle architecture.
- Pipeline architecture.
- Data and control dependency management.
- Memory organisation and cache memory.
- Application Binary Interface (ABI).

Prerequisites (recommended):

- Digital design course.

3.8 HDL Synthesis

Course Credits: 4

Course Objectives and Contents:

Introduction to hardware description languages, with a particular focus on the System Verilog HDL. This level of abstraction enables the modelling of any digital system at the register transfer level (RTL). The design methodology aspects associated with the use of such languages will also be addressed. The course covers logic simulation, the logic synthesis stage and post-synthesis simulation of the synthesised system. All concepts will be illustrated through the design of a circuit of significant complexity.

Learning Outcomes: Modelling, simulating and synthesising a hierarchically designed circuit.

- Introduction to the System Verilog language.
- Structural modelling.
- Data flow/generic modelling.
- Behavioural/machine modelling with a finite number of states.
- Introduction to digital circuit synthesis.
- Combinational circuit synthesis and post-synthesis simulation.
- Sequential circuit constraints.
- Cryptographic-accelerator design and synthesis project.

Prerequisites (recommended):

- Digital design course.

3.9 Advanced Simulation and Verification

Course Credits: 3

Course Objectives and Contents:

Circuit verification concepts will be presented in this course to raise students' awareness of the issues involved in verifying a circuit during its design. Various aspects will be covered, such as code and test coverage, random test vector generation, adding assertions, and an introduction to UVM (Universal Verification Methodology).

Learning Outcomes: understanding the verification of a complex digital system, taking into account the environment (communication protocol, etc.).

- Introduction to integrated circuit verification, use of System Verilog for verification.
- String management and introspection: simulating a system using text files.
- Verification of a circuit using a C/Python/DPI model.
- Code coverage/test coverage.
- Assertion.
- Introduction to UVM, practical work.

Prerequisites (recommended):

- HDL synthesis and Digital gates courses.

3.10 Co-Design and FPGA Prototyping

Course Credits: 3

Course Objectives and Contents:

The course covers mastering an FPGA design flow for developing software and hardware applications. The topics covered will be: software/hardware partitioning of the application according to performance objectives, hardware implementation of a component: from simulation to implementation on an FPGA target, communication between the processor and programmable logic using the AXI bus, and the complete design of a system comprising a C programme supported by a hardware brick.

Learning Outcomes: acquiring a methodology for designing a software application supplemented by a hardware coprocessor.

- Methodology and software/hardware approach.
- Hardware design flow (synthesis and placement/routing).
- Hardware design on FPGA.
- Insertion of IP from the Xilinx library.
- Software and hardware design flow.
- Design of hardware IP connected to the ARM processor.
- Communication with the ARM processor via the AXI bus.
- Software and hardware integration of the application.

Prerequisites (recommended):

- HDL synthesis course.

3.11 Low Power and Energy Harvesting

Course Credits: 4

Course Objectives and Contents:

This course addresses two topics. It covers the main techniques used to reduce the power consumption of integrated circuits (clock gating, power gating, etc.). It also offers a systemic view of energy consumption control. The study is based on an autonomous system for which an energy balance analysis is carried out to balance the power produced by an energy harvesting source and the power consumed by the connected object. This module implements theoretical approaches, simulation, and energy performance characterisation. It aims at teaching the basics of low-power circuit design.

By the end of this module, students will be able to devise a design strategy for autonomous objects and define/determine their energy balance.

Learning Outcomes:

Part 1 – Design of low power integrated circuits

- Study of digital integrated circuit design techniques for very low power consumption.
- Simulation of the power consumption of an integrated circuit.

Part 2 – Integration of low-power integrated circuits into a low-power system

- Energy balance methodology.
- Principles and uses of energy sources: photovoltaic, thermal, vibratory.
- Principles and sizing of batteries.
- Modelling, simulation and characterisation of energy sources.
- Study of a multi-source power management circuit.

- Energy consumption of system components: microcontroller, sensor, radio transmitter, calculation of key parameters.
- Evaluation of energy quantum (consumption of an elementary cycle of operation).
- Measurement and analysis of the connected object's consumption.
- Integration of the autonomous system and energy optimisation.

Prerequisites (recommended):

- Digital design and Processor Architecture courses.

3.12 NMOS fabrication lab

Course Credits: 4

Course Objectives and Contents:

This course provides a comprehensive, hands-on introduction to the fabrication processes of silicon integrated circuits using NMOS technology within a cleanroom environment. Over the course of one week at the Inter-university Workshop of Micro-Nano-Electronics (CNFM platform) on the INSA-Toulouse campus, students will engage in the complete fabrication workflow starting from a bare silicon wafer. The curriculum encompasses all critical process steps, including photolithography, wet and dry etching, thermal oxidation, thin-film deposition, doping via diffusion or ion implantation, and metallization. Emphasis is placed on the physical characterization of process parameters—such as layer thickness, resistivity, and junction depth—as well as the electrical testing of fabricated structures. These structures include fundamental components (diodes, polysilicon and aluminum resistors, MOS capacitors, and both long- and short-channel MOS transistors) and integrated circuits (decoders, flip-flops, Schmitt triggers, and sequencers). Through this immersive experience, students will acquire practical expertise in the core techniques and analytical methods essential to microelectronics fabrication.

Learning Outcomes:

- Day 1: Deposition of masking oxide, Photolithography 1 (channel opening), and characterization (measurement of masking oxide thickness, R-squared analysis).
- Day 2: RCA cleaning, gate oxide deposition, polysilicon deposition, Photolithography 2 (polysilicon etching), and characterization (polysilicon thickness measurement).
- Day 3: Doping (diffusion or ion implantation), SiO_2 deposition, Photolithography 3 (contact openings), and metallization.
- Day 4: Characterization (junction depth measurement), Photolithography 4 (metal etching).
- Day 5: Functional electrical testing, packaging, and final electrical testing.

Prerequisites (recommended):

- Semiconductor Physics, Semiconductor Devices, and Process and Manufacturing courses.

4 Modules at KTH

The master’s program at KTH has been running for a few decades. A new electronics track has been created with some new courses to bring the chip-design education as the focus. These new courses are listed below. Some old courses are also listed that have been upgraded to align with the objectives of the EDU4Chip.

4.1 Introduction Integrated Circuits

Course Credits: 7.5

Course Contents:

The course is an introduction to modern integrated circuits. It covers basic topics common to analogue and digital integrated circuits built with CMOS technology. The performance and function of basic circuits and their electrical connections are studied. System architecture is introduced followed by hierarchical design, circuit simulation, physical implementation and verification. The course aims to make students acquainted with industrial EDA tools.

Learning Outcomes:

After passing the course, the student should be able to

- describe the operation, fabrication and scaling of CMOS technology
- explain and estimate the influence of electrical couplings on delay and power consumption in CMOS technology
- explain how complex integrated circuits are designed
- design standard cells, such as logic gates and flip-flops.
- describe how memories are designed and compare their performance
- use Electronic Design Automation (EDA) tools for the design, verification and characterisation of simple integrated circuits
- describe the impact of fabrication and use of integrated circuits on sustainable development objectives.

Pre-requisites:

Knowledge of digital circuits covering 3 higher education credits, equivalent to completed course IE1204/IE1205/IL2246.

Knowledge of electromagnetism covering 7.5 higher education credits, corresponding to completed course IF1330/EI1110/IE1206 or knowledge of analogue electronics covering 4.5 higher education credits, corresponding to completed course IL2246.

4.2 Embedded Hardware Design in ASIC and FPGA

N.B: This is an old course that has been enhanced in EDU4Chip project with extra lectures and labs for DFT and ATPG.

Course Credits: 7.5

Course Contents:

- Essential concepts for logic/FSM and Algorithm implementation using automated design flows.
- Use of HDL coding styles for efficiency, simulation, timing, clock domain crossing and power.
- Constraints of Technology and Optimisation, and interface to foundry and back-end physical synthesis flow.
- Optimizing Designs for area, performance and power in Logic/FSM synthesis.
- Static Timing Analysis.
- Design for test/manufacturing and ATPG

Learning Outcomes:

- explain the concepts of Abstraction, Domain, Synthesis and Analysis and classification of Synthesis Tools
- explain implementation methods such as Full Custom, Std Cells, Mask Programmable Gate Arrays and FPGA and comparison between them
- use logic/FSM coding styles and Algorithms in HCD for efficient implementation and reuse
- account for the logic/FSM architectural design space and meaning of the HDL code
- optimize Area, Performance and Power with respect to logic/FSM on algorithmic level
- account for the limitations of Technology and Optimization, their implications and use in logic/FSM
- describe the libraries used in Logic Synthesis
- calculate and analyze performance and power for logic/FSM at the algorithmic level
- explain methods for Logic Synthesis and place-and-route

Pre-requisites:

- Knowledge in electrical circuit analysis, 6 credits, corresponding to completed course EI1110/EI1120/IE1206.
- Knowledge in digital design, 6 credits, corresponding to completed course IE1205/IE1204.
- Knowledge in digital design and validation using hardware description languages, including experience with HDL simulators such as ModelSim or Xcelium, 6 credits, corresponding to completed course IL2203.
- Basic experience in Linux environments
- Basic programming and scripting knowledge
- Basic usage of the command line

4.3 Analog Integrated Circuits

Course Credits: 7.5

Course Contents:

The course is an introduction to analog integrated circuits. The course focuses on negative feedback amplifiers which are central building blocks in analog, digital and radio frequency circuits. Synthesis of amplifiers based on the asymptotic model is introduced. In detail, component modelling, low noise and low distortion design, bandwidth and frequency compensation, rest point setting and analog layout are studied. The course includes practical amplifier design from specification to physical verification, using the latest EDA tools.

Learning Outcomes:

After passing the course, the student should be able to

- explain how noise, signal power and bandwidth relate to amplifier performance
- use the asymptotic model to synthesise amplifiers with feedback
- apply different techniques to minimize noise and distortion
- apply bandwidth optimisation and frequency compensation based on phantom zeros
- implement rest-point setting circuits using current and voltage sources
- apply analog layout techniques for physical implementation
- use Electronic Design Automation (EDA) tools for the design, verification and characterisation of simple integrated circuits
- write design documentation

Pre-requisites:

Knowledge in analog electronics covering 4,5 higher education credits, equivalent to completed course IE1202/IE1207.

Knowledge of integrated circuits covering 3 higher education credits, corresponding to completed course IL2241 or completed module LAB1 in IL2241.

4.4 Design of radio frequency integrated circuits

Course Credits7.5 credits

Course Contents:

This course is an introduction to radio frequency integrated circuits and systems. Firstly, basic concepts of RF design and wireless communication are covered. Then, the course introduces transceiver architectures and their building blocks: low noise amplifier, mixer, oscillator, phase locked loops and power amplifier. The implementation of each building block in modern CMOS processes is studied in detail. Finally, current trends in RF design are described including MIMO and beamforming, direct RF sampling, envelope tracking and digitally assisted RF circuits. The course includes practical design of RF blocks from specification to physical verification using the latest EDA tools.

Learning Outcomes:

After passing the course, the student should be able to:

- describe how transceiver architectures are built and explain their advantages and limitations
- describe the modelling of active and passive integrated components at radio frequencies (RF) and the influence of parasitic electrical components
- describe performance measures of RF blocks, such as noise figure, third order intercept (IP3), gain, bandwidth and phase noise
- given a set of specifications, compare different circuit topologies for RF blocks and select a suitable topology for implementation
- analyse basic RF blocks, dimension active and passive components, and design bias point setting circuits
- discuss modern trends in RF design such as MIMO and beamforming, direct RF sampling, envelope tracking and digitally assisted RF circuits
- use Electronic Design Automation (EDA) tools to design and verify RF blocks.

Pre-requisites:

Knowledge of Introduction to Integrated Circuits covering 7.5 higher education credits, equivalent to the completed course IL2241.

Knowledge of analog integrated circuits covering 3 higher education credits, corresponding to the completed course IL2242 or the completed module LAB1 in IL2242.

4.5 Project Course in Application Specific Integrated Circuits

Course Credits: 15.0

Course Contents:

The course is project-based and focuses on designing an application-specific integrated circuit (ASIC) covering all design steps from specification to submission of GDSII manufacturing masks ("tape-out"). A few selected designs will be sent for fabrication and circuit performance will be measured in course IL2237. Students can choose to do a project involving either the use of the full digital design flow or the use of the analog/digital design flow. The course is delivered in collaboration with industrial partners who will provide guidance during several of the design phases.

Learning Outcomes:

- describe what design specifications are and how they are used in the design flow
- based on a set of specifications, propose a functional description level architecture that includes hierarchical building blocks
- create a project plan that includes tasks, milestones, deliverables, timeline and resource allocation
- use the analog/digital design flow to implement each block in the architecture
- evaluate the performance of each block at each design step and propose and make changes if necessary

- connect all building blocks in a hierarchy using either the Digital-on-Top (DoT) or the Analog-on-Top (AoT) method
- present and discuss their own designs to colleagues and experts
- write design documentation.

Pre-requisites:

Knowledge of Introduction to Integrated Circuits covering 7.5 higher education credits, equivalent to the completed course IL2241.

Knowledge of analogue integrated circuits covering 3 higher education credits, corresponding to the completed course IL2242 or the completed module LAB1 in IL2242.

Knowledge in digital system design and verification with hardware description language covering 9 higher education credits, equivalent to the completed course IL2234.

Knowledge of hardware design for embedded systems covering 4.5 higher education credits, corresponding to the completed course IL2225 or the completed module PROA in IL2225.

5 Modules at TAU

Tampere University implements the Edu4Chip programme into Master's Programme in Computing Sciences and Electrical Engineering as a specialization module Advanced Studies in System-on-Chip Design. The new and improved courses available in the module are listed below.

5.1 Digital Design

Course Credits: 5

Course Contents:

Core content

- Specification, design and analysis of synchronous logic. Different levels of description abstractions in digital systems.
- Combinational gate networks. Two and multi-level gate networks. Critical path and maximum clock frequency.
- Sequential networks. Mealy and Moore state machines. Timing analysis and determination of clock speed.
- Standard modules. Connecting modules using bus structures. Data and control paths. Register Transfer Level.
- FPGA architecture and usage in digital design. Design tools and methodologies.

Complementary knowledge

- The Y-model: behaviour, architecture and mapping. Hierarchical design.
- Propagation delay. Fan-in and fan-out. Loading of the gates.
- Extended state machines with memory. Registered state machine.
- Two's complement representation. Binary addition, subtraction, and multiplication.
- FPGA boards with peripherals. Practical designs.
- Understanding data sheets.

Learning Outcomes:

This course gives knowledge of specification, design, implementation and analysis of digital systems. Students learn the theory of combinational and sequential systems, especially the design of state machines. Students master state-of-the art design tools, and can select the best specification and implementation description for typical design tasks. FPGA platforms are used to carry out real implementations. After the course, students can implement real FPGA-based digital systems.

Prerequisites:

Recommended:

- COMP.CE.100 Introduction to Embedded Systems

5.2 Logic Synthesis

Course Credits: 5

Course Contents:

Core content

- Main phases in implementing a digital circuit.
- Basics of VHDL language and how it is synthesized into circuit.
- Component verification and reuse. Principles of RTL simulators.
- Systems with multiple clock signals. Synchronization interfaces.

Complementary knowledge

- System realization in FPGA.
- Tri-state logic. Latches.
- Data sheets

Learning Outcomes:

After this course, a student can implement a working digital system according to specification, i.e. convert a natural language specification into VHDL language hardware description, verify it, and synthesize it into FPGA chip.

Moreover, students learn the relation between VHDL description and logic realization, can determine its behavior with given stimulus, and understand clock synchronization principles.

Prerequisites:

Basic knowledge of digital logic is required, e.g. AND, OR, DFF, state machines, timing diagrams

Compulsory:

- COMP.CE.200 Digital Design

5.3 System-on-Chip Design

Course Credits: 5

Course Contents:

Core content

- The structure of a general System-on-Chip
- SoC interconnects and interfaces
- SystemVerilog as a design language
- The IP-XACT standard

Complementary knowledge

- Power, performance, and area considerations
- Clock, voltage, and power domains

Specialist knowledge

- FPGA prototyping
- Low-power design
- Modern design tools

Learning Outcomes:

After completing the course, the student can work as part of a SoC subsystem design team. They know the common internal interconnects and external interfaces, and the usual phases of the design project. They can analyze their design choices in terms of power, performance, and area.

Prerequisites:

Basic knowledge of digital design with VHDL or Verilog is mandatory.

Compulsory:

- COMP.CE.240 Logic Synthesis

5.4 High-level Synthesis

Course Credits: 5

Course Contents:

Core content

- Fundamentals of high-level synthesis
- Scheduling of IO and memories
- Sequential and combinational hardware
- Memory architecture
- Hierarchical design

Complementary knowledge

- General coding style, bit-accurate data types, pipelining, unrolling
- Unconditional/conditional IO
- Shift registers, multiplexers, shifters, accumulators, adder trees, lookup tables
- Memory organization, caching
- Sharing arrays, sharing control variables, reconvergence, channels, arbitration, feedback
- FIR filter, FFT examples

Learning Outcomes:

After the course, the student can implement digital systems using C/C++ based high-level synthesis (HLS). The student understands the whole HLS design flow from specification to implementation on FPGA. He/she can write the source code for the design, the test bench for it, and apply various optimizations with the HLS tool. The student will also understand the capabilities and limitations of HLS with respect to traditional RTL design flow.

Prerequisites:

The student should know the basics of digital design and C++ programming before attending this course. Knowledge of VHDL/Verilog HDL is also useful, but not mandatory.

Compulsory:

- COMP.CE.200 Digital Design

Recommended:

- COMP.CE.240 Logic Synthesis
- COMP.CS.110 Programming 2: Techniques

5.5 System Design

Course Credits: 5

Course Contents:

Core content

- Introduction to embedded, multi-processor system-on-chip (SoC) technologies.
- Importance of design reuse and application portability between HW and SW implementations.
- Principles of model based design with abstraction and orthogonalization of concerns.
- Layered system model and interfaces.
- Related HW and SW standards and best practises.
- Hands-on exercise work implementing a parallel, real-time application on heterogeneous HW/SW platform

Complementary knowledge

- International Technology Roadmap for Semiconductors (ITRS)
- Modularization of HW and SW components.
- Y-model based design process. Specification, synthesis, verification.
- Hardware dependent software, APIs, RTOS
- IP-XACT, OCP-IP, MCAPI, POSIX, SystemC
- Performance estimation, measurement, analysis and optimization.

Specialist knowledge

- Complexity of HW and SW and their trends
- XML-based metadata for HW/SW component integration.
- Design automation and tools. Management of configurations.
- Driver implementation.
- Embedded Linux
- Development tools chosen for the course implementation.

Learning Outcomes:

After completing the course, the student knows the design process of complex embedded systems including several processors, HW IP-blocks, SW platform components and SW applications with real-time constraints. In the exercise work, students practice HW/SW co-design and system integration, verification, and prototyping. Students implement a small video encoding system on FPGA. The first step is creating a behavioral model that is simulated on PC. From it, a set of implementations are created on FPGA board by varying the number of processors and HW IP-blocks. Performance and costs are analysed for the implementations. All this gives a realistic embedded product development experience.

Prerequisites:

Basic knowledge of digital design and C/C++ programming is mandatory.

Recommended:

- COMP.CS.110 Programming 2: Techniques
- COMP.CE.200 Digital Design

5.6 System-on-Chip Verification

Course Credits: 5

Course Contents:

Core content

- Fundamental concepts in SoC verification
- Object-oriented properties of SystemVerilog language
- Universal Verification Methodology (UVM)

Complementary knowledge

- UVM on integration level
- Formal verification

Specialist knowledge

- UVM Register Abstraction Layer
- SystemVerilog DPI

Learning Outcomes:

After completing the course, students are able to run the verification flow for a single subsystem in a SoC using a modern standard methodology. They understand how to integrate the subsystem verification environment on a higher level. They know the key metrics used for tracking the verification project.

Prerequisites:

Basic knowledge of digital design with VHDL or Verilog is mandatory. Experience of object-oriented programming is recommended.

Recommended:

- COMP.CS.110 Programming 2: Techniques
- COMP.CE.200 Digital Design
- COMP.CE.240 Logic Synthesis

5.7 Chip Implementation

Course Credits: 5

Course Contents:

Core content

- ASIC synthesis and physical design

- Standard cell technologies and libraries
- ASIC sign-off tools

Complementary knowledge

- Chip IO planning and constraints
- Design for testability
- Power, performance, and area analysis

Specialist knowledge

- Physical-aware synthesis flow
- Multi-mode Multi-Corner timing closure

Learning Outcomes:

After completing the course, the student can perform the tape out for a simple chip. They understand the contents of standard cell libraries and know the challenges in ASIC backend flow.

Prerequisites:

Basic knowledge of digital design with VHDL or Verilog is mandatory.

Recommended:

- COMP.CE.240 Logic Synthesis

6 Modules at TUM

The Technical University of Munich implements the Edu4Chip programme via a new Master Program in Microelectronics and Chip Design. Many courses, which can be selected in this program, were and are still available in other master programs, like the Master of Communications and Electronics and the Master of Electrical and Computer Engineering. The following list reflects the status of the modules developed or improved with the support of the Edu4Chip program at the time of writing this report. Some of these modules are also open to the other mentioned master programs. The current versions of the module descriptions, including responsible persons and further details as well as a list of all courses in the new master program is available through https://campus.tum.de.

6.1 Chip Design Test and Evaluation Laboratory Course

Course Credits: 5

Semester: Winter Semester

Description of achievement and assessment methods:

Achievement of the learning objectives is demonstrated by a project work which is supplemented by documentation of the results (approx. 3 pages per student) and a presentation. The project work includes the commissioning of a chip and the performance of measurements according to a catalog of requirements. The individual results are documented. The ability to assess and explain the results is demonstrated in the form of a group presentation lasting approx. 30 minutes (approx. 5 minutes per student followed by questions), whereby questions can also be asked regarding the individually documented results. The grade is based entirely on this oral examination.

(Recommended) requirements:

Before taking this module, students should complete the Research Laboratory Physical Design of Integrated Analog and Mixed-Signal Circuits or the Research Laboratory Physical Design of Integrated Digital Circuits.

Contents:
This internship is offered in the form of a block course and builds on the Digital Backend Design or Analog Backend Design research internship. A chip developed there or equivalent is put into operation by students. To this end, students work in small groups to develop the necessary control for testing a sub-circuit that they have usually developed themselves in a previous practical course; the specific functionality of the sub-circuit may vary; examples include AI accelerators, cryptographic algorithms, co-processors, or phase-locked loops. The students carry out metrological investigations and characterizations for this subcircuit, e.g., with regard to power consumption or temporal behavior, compare the results achieved with previously defined specifications and thus determine the quality of the developed chip. The results are presented in the form of a seminar.

Learning Outcomes:

After successfully completing this module, students will be able to

- Realise the comimissioning of an Application Specific Integrate Circuit (ASIC),
- Independently evaluate the functionality of an integrated circuit, and

- Evaluate the quality of an integrated circuit with regard to typical criteria such as power consumption or time behavior by measurement.

Teaching and learning methods:

This practical course has the character of group-oriented project work in which students carry out experiments but also have to work out solutions independently. The objectives are discussed in an introductory event; in accompanying seminars and feedback sessions, students receive feedback on individual difficulties.

Media formats:

In the introductory course, slides and blackboard notes are used to convey content. Students also receive digital access to program code and materials for carrying out the internship tasks and individual feedback in discussions.

6.2 Circuit Design for Security

Course Credits: 5

Semester: Summer Semester

Description of achievement and assessment methods:

Knowledge and theoretical understanding of basic elements of digital circuits and of typical circuit design problems, as well as the ability to select solutions for typical circuit design problems, particularly in the context of security, will be examined in the form of open and closed questions in a written exam (60 minutes). The questions are to be answered without additional documents and in the form of short text or bullet points. Calculations may be necessary to answer the questions.

The practical skills to independently implement, synthesize, and simulate essential elements of digital circuits are demonstrated by solving 3-5 implementation tasks. A grade bonus of 0.3 is awarded to the final grade for the successful completion of all tasks. The practical tasks are a voluntary mid-term assignment.

(Recommended) requirements:

Basic knowledge in VHDL

Basic knowledge of IT security as taught,e.g., in the lectures Grundlagen der IT-Sicherheit and Angewandte Kryptographie

Contents:

In this module, the development of digital circuits in the security context is examined starting from basic circuits. First of all, the knowledge of implementing state machines will be refreshed and extended. The implementation of important components of digital circuit design such as FIFOs and LFSRs is introduced. The implementation of pseudo-random number generators based on LFSRs and their application in the security context will be discussed. The implementation of multipliers is essential in many cryptographic applications. For this purpose, fundamental implementations as well as special variants for modulo-multiplication are discussed in the module. Also, concepts are discussed to test cryptographic circuits appropriately; Implementation strategies are introduced to optimize circuits in terms of area, power consumption, and performance.

In addition to the theoretical teaching of the content in lectures, the practical relevance is clarified in exercises and in a lab course. The students also learn to implement and synthesize digital circuits with a commercial design tool.

Learning Outcomes:

After successful completion of the module, students have the following qualifications:

- They know essential elements of digital circuits such as state machines, FIFOs, LFSRs and multipliers and can implement them independently.
- They know typical circuit design issues such as clock domain crossing, the choice of synchronous and asynchronous reset or the testability of circuits and can choose suitable solutions.
- They know application scenarios of multipliers and LFSRs in the security context, can understand the theoretical foundations of the concepts discussed, and can name the advantages and disadvantages of these.
- They are able to describe digital circuits in VHDL and perform initial synthesis steps.

Teaching and learning methods:

Lecture and exercise content is provided through slides and blackboard. The learning process of students is supported in the exercises by interactively solving tasks, group work and program demonstrations of the advisor which show implementation tasks and security scenarios. Programming exercises during the semester give students the opportunity to practice and to demonstrate their skills in implementing and synthesizing digital circuits in a security context using a commercial design tool.

Media formats:

In lecture and exercise, slides, blackboard, example code and interactive online tools, e.g. Question Catalogs, are used. The offer is complemented by a forum on Moodle, where teachers answer questions about the content and the lab course tasks.

6.3 Design of Digital Circuits

Course Credits: 5

Semester: Winter Semester

Description of achievement and assessment methods:

In a written exam (60 min), students demonstrate by answering text questions and calculating tasks that (i) they can correctly reproduce the concepts of digital circuit technology and their physical relationships and that (ii) they can apply the knowledge they have learned to solve typical problems in the field of digital design. The assignments test the achievement of the learning objectives, in particular with regard to the topics of time and power behavior of circuits, memory technologies, fault tolerance, and current concepts for the implementation of basic components of digital circuits.

In addition, students can achieve a voluntary grade bonus of 0.3 on the final grade by successfully completing the homework assignments (3 to 5 tasks). With these tasks, students demonstrate in particular their knowledge of the design process of digital circuits and their ability to transfer theoretical concepts into practice.

(Recommended) requirements:

Basic knowledge of working principals of transistors and Boolean algebra at Bachelor level is required.

Contents:

This module explains key concepts that are used in current digital design for integrated circuits. Based on fundamental physical relationships, the properties of digital circuits with regard to power consumption and time behavior are investigated and design methods for these purposes are discussed. Based on known basic circuits, current concepts for the implementation of typical components of digital circuits are discussed and their implementation variants are derived. The relationship between the description of these concepts in a hardware description language and the implementation in an integrated circuit is also discussed, whereby the various steps of the design process are also considered.

In addition to the theoretical part of the content in lectures, the practical relevance is illustrated with exercises and homeworks.

Learning Outcomes:

After successfully completing this module, students will have acquired the following qualifications:

- They understand causes of power dissipation in digital circuits and techniques to reduce power dissipation and will be able to apply these techniques in the design of digital circuits.
- They understand the differences between synchronous and asynchronous implementations of digital circuits, can determine the timing behavior of digital circuits, and can apply methods for synchronization between clock domains.
- They understand the structure of different memory technologies and can apply this knowledge to make a suitable choice when developing circuits.
- They know the principles of fault tolerance and can evaluate them.
- They know the basic elements of digital circuits such as state machines, FIFOs, adders and multipliers, understand current concepts for their implementation, and can evaluate which concepts are suitable in a given scenarios.
- They know the steps of the design process and understand the relationships between a description in a hardware description language and the resulting circuit.

Teaching and learning methods:

Knowledge is conveyed by means of slides and blackboard notes. The students' learning process is supported in the exercises by interactive solutions to tasks. Practical homework during the semester helps to deepen and illustrate the knowledge acquired.

Media formats:

Slides, blackboard work, sample code, and interactive online tools, e.g. questionnaires, are used in lectures and exercises. The offer is supplemented by a forum on Moodle, where lecturers answer questions about the content and practical tasks.

6.4 Fundamentals of CMOS Technology for Analog Design and Standard Cell Libraries

Course Credits: 6

Semester: Winter Semester

Description of achievement and assessment methods:

The exam is in a written form and it lasts 75min. The exam includes questions about the principles of CMOS technology and the basic operations of transistors as well as concepts of analog circuits and standard cells. Different CMOS technologies will be covered. In addition, problems to test the understanding of how changes in operating conditions such as voltage and temperature impact the figure of merits of circuits like power, performance, and delay. These problems cover extensive topics related to how standard cells can be optimized and how reliability degradation like variation impact analog circuits.

(Recommended) requirements:

- Basic knowledge in circuits
- Basic knowledge in transistors

Contents:

- Fundamentals in CMOS technology.
- Fundamentals in standard cell design from both analog and digital perspective.
- Basic principles of MOSFET operations along with the key electrical parameters.
- Fundamentals in variability and temperature effects on transistor's operation and performance of analog circuits.
- Basics in layout for standard cells.
- Basics of building standard cell libraries

Learning Outcomes:

At the end of the module students know and understand the fundamental principles of CMOS technology. How different technologies (e.g., planer MOSFET, FDSOI, FinFET, nanosheet, etc.) impact the figure of merits of circuits. Students obtain a comprehensive knowledge and solid understanding of how standard cells work from analog and digital perspectives. Students know and understand the impact of variability, temperature, voltage on the reliability and performance of analog circuits and standard cells. In addition, student know in detail the key electrical parameters of transistors and how they impact circuits.

Teaching and learning methods:

The module includes lectures and tutorials. In the lectures, basic concepts and fundamental of CMOS technology and the relation to analog and standard cells. Visual aids such as PowerPoint slides, overhead projections and handwriting on black board are used. In each lecture, some concepts in CMOS technology and different challenges will be presented and discussed. Then discussions are held to explain and understand the transistor operation. In the tutorials, practical exercises of how transistors operate, SPICE simulations, standard cells design, basic concepts of standard cell libraries. Students also need to read scientific papers to discuss and challenge them for more comprehensive understanding. Students practice with small examples and thus get familiar with analog circuit simulations and standard cell design.

Media formats:

PowerPoint, black board, online materials, book chapter, scientific papers

6.5 HDL Chip Design Laboratory

Course Credits: 5

Semester: Winter and Summer Semester

Description of achievement and assessment methods:

The achievement of the module's learning outcomes is assessed through practical exercises consisting of four programming tasks. Additionally, students have the opportunity to participate in an optional midterm exam.

Programming Tasks:

Through the programming tasks, students demonstrate their ability to design and develop an embedded system on FPGA. For this purpose, students write HDL code for a given system, perform simulations of the system, test their system on an FPGA, debug their code, and document the components they have developed.

Optional Midterm Exam (graded written exam, 60 minutes):

In the exam, students demonstrate their ability to understand design concepts for integrated and embedded systems, as well as to analyze and evaluate these systems by answering questions. Tasks may include interpreting results from a design simulation or theoretical questions about the concept of an embedded system.

The final grade is based 100% on the programming tasks. An improvement through the midterm exam is possible. In this scenario, the final grade is calculated based on a combination of "75% programming task grade" and "25% midterm grade," as long as the student achieves a passing grade of at least 4.0 in the programming tasks and the weighting results in an overall grade improvement.

(Recommended) requirements:

Fundamentals of digital logic design Fundamentals of programming

Contents:

Concept of an integrated electronic system; build an example of an integrated system with a microcontroller, bus, and peripherals; first implement an encryption algorithm using a standard hardware description language; then wrap the security module as a peripheral attached to the bus; design an interface between peripheral and bus; apply an FPGA design flow for embedded systems, and embedded software for testing the encryption algorithm.

Learning Outcomes:

By the end of the module, students will be able to analyze and evaluate concepts of integrated and embedded systems (“chips”). They will be capable of designing and creating integrated and embedded systems (particularly an embedded system with FPGA, bus, and peripherals) with their complex system components using a standardized hardware description language (HDL) and an FPGA design flow.

Teaching and learning methods:

Learning method:

In addition to the students' individual methods, consolidated knowledge is acquired by providing subtasks of increasing complexity and difficulty in the laboratory notes.

Teaching method:

Students are free to work on the laboratory tasks independently, according to their schedule. Students can work in the laboratory either in institute rooms or at home. An adviser is available to support them in case of significant difficulties.

Media formats:

The following kinds of media are used:

- Introductory lectures
- Lecture slides available
- Laboratory notes with detailed descriptions of tasks and tool environments
- Individual discussions with advisor

6.6 Lab Analog Chip Design

Course Credits: 5

Semester: Winter Semester

Description of achievement and assessment methods:

The examination is completed in the form of a project. This consists of a project assignment with a practical project part that demonstrates a basic understanding of the practical side of developing an integrated analog circuit, the ability to dimension the transistors it contains in a design tool, the ability to optimize such a circuit and an understanding of the interaction between the design tool and PDK. The project part is worked on by the students in small groups of approx. two students. The individual contribution to the individual work steps in the project part is documented in writing by the students (the scope of the documentation is 2 to 5 pages per student). The ability to justify and defend their own design decisions on the basis of the basic understanding they have acquired and to present them in an illustrated manner, summarized to the essential core and communicate them to a knowledgeable audience is demonstrated in a group presentation (10 minutes per student, i.e. 20 minutes for a group of two) followed by questions. The documentation (25% of the overall grade) and the presentation (75% of the overall grade) are used for the individual grading of the internship.

(Recommended) requirements:

Basic knowledge of circuit design and MOSFET operation

Contents:

The students will learn the basics of analog integrated circuit design using Cadence Virtuoso. This includes:

- MOSFET operation and biasing
- Simple Amplifier Circuits
- Basic and advanced Current Mirrors
- OTA design and amplifier metrics
- Miller OpAmp
- Advanced OTA configurations and design

The students will gain hands-on experience in designing integrated circuits with state of the art EDA software and gain a basic knowledge of analog design techniques.

Learning Outcomes:

After successfully completing the lab, the students have a basic understanding of the design of analog integrated circuits. They are able to explain to analyze science-based their design decisions of analog integrated circuits and critically reflect on them. They can bias a transistor according to its function in an analog circuit and are able to optimize their behavior. Additionally, the are able to use the basic functionalities of Cadence Virtuoso and know the connections between tooling and PDK. Students can illustrate using suitable media and summarize the steps involved in the development of an integrated analog circuit and communicate them competently.

Teaching and learning methods:

Independent simulation according to course instructions under supervision. Own presentation at the end of the course.

Five guided practical experiments with theory discussion and practical simulation of circuits and circuit concepts with Cadence Virtuoso. The simulations are carried out in groups of two or three with on-site supervision. After each experiment there will be a discussion and explanation with the supervisor.

Media formats:

Course script, Simulation software

6.7 Logic Synthesis and Physical Design

Course Credits: 6

Semester: Summer Semester

Description of achievement and assessment methods:

The examination is performed in form of an oral exam (30mins; applied when the course has less than 20 students) or a written exam (120mins; applied when the course has 20 or more students). The exam will cover tasks conducted before in the hands-on sessions/exercises (which in turn, cover the main content of the lecture in a practical fashion). The tasks/exercises allow to evaluate how well the students understood the respective concepts and how well they can implement corresponding algorithms for logic synthesis and physical design.

(Recommended) requirements:

Programming skills

Contents:

Modern computer chips consist of billions of transistors, making them some of the most complex systems ever created by humans. How does one design such intricate architectures? The answer is algorithms developed and fine-tuned over decades. In this course, students will learn about the techniques that automatically obtain computer chip designs from specifications. To this end, we will explore logic synthesis and optimization as well as partitioning, floorplanning, placement, and routing. Many of these algorithms are meta-heuristics that can be applied in completely different fields, too, like resource allocation, city planning, logistics, compilers, etc. Additionally, students will gather hands-on experience with state-of-the-art tools in logic synthesis and physical design, with the opportunity to participate in an international contest.

The students will get to know data structures and algorithms, and will be able to implement them, e.g.,

- AIGs, MIGs, XAGs, kLUT networks,
- logic optimization and technology mapping,
- combinational equivalence checking,
- floor planning,
- global and detailed placement,
- global and detailed routing,
- legalization.

Additionally, the students will learn to operate open-source industrial-strength tools in the field, like ABC, Yosys, OpenROAD, or iEDA.

Learning Outcomes:

At the end of the module, students will have

- a detailed insights into how synthesis, optimization, and physical design algorithms work,
- an understanding of different data structures and core methods for software tools,
- the ability to properly apply synthesis, optimization, and physical design tasks,
- the capability to extend software tools with novel or existing algorithms, and
- be able to use tools considered in the lecture (e.g., ABC, Yosys, OpenROAD, or iEDA).

Teaching and learning methods:

The module will be held in the form of presentations about the topics covered above followed by corresponding hands-on sessions/exercises. Using slides presentations and whiteboard sketches, the main concepts of the respectively considered topics are provided. In addition, the students will have the opportunity to deepen their knowledge through individual hands-on experiences with corresponding software tools.

Media formats:

Lecture slides, software tools.

6.8 Machine Learning for Electronic Design Automation and Manufacturing

Course Credits: 5

Semester: Winter and Summer Semester

Description of achievement and assessment methods:

The examination is performed in form of an oral exam (30mins; applied when the course has less than 20 students) or a written exam (60mins; applied when the course has 20 or more students). The exam will cover tasks conducted before in the hands-on sessions/exercises (which in turn, cover the main content of the lecture in a practical fashion). The tasks/exercises allow to evaluate how well the students understood the respective concepts and how well they can implement corresponding machine learning algorithms for design automation and manufacturing.

(Recommended) requirements:

Basic machine learning knowledge, basic programming knowledge, basic circuit design knowledge

Contents:

The complexity of modern chips significantly impacts the cost and capabilities of design and manufacturing for traditional Design Automation Toolkits. This issue is exacerbated by the increased relevance of software and applications running on modern SoCs and their co-design. Current Design Automation methodologies often struggle to fully capture and optimize complicated designs or reduce them to the initial specification. However, advancements in data-driven algorithms, particularly in Machine Learning, can address these shortcomings. This course teaches how to apply Machine Learning to enhance and improve the chip design process.

This module provides an in-depth exploration of machine learning for design automation, including:

- Theory and application of machine learning algorithms
- In-depth exploration of recent machine learning methods suitable for design automation tasks
- Overview of topics, areas, and current industrial pain points in the semiconductor chain where a great availability of data exists
- Detailed coverage of data structures and state-of-the-art tools for tackling design automation
- Hands-on experiences of machine learning algorithms applied to design automation tasks

Learning Outcomes:

By the end of the module, students will

- gain detailed insights into how machine learning can aid in the development, design, testing, and manufacturing of chips,
- understand which problems in chip design/manufacturing can be approached by data-driven methodologies and how to do so, and

- have learnt the essential requirements and best practices for introducing machine learning in a chip design/manufacturing industrial setting.

The course will also cover different data structures and core methods that may form the basis for AI-based software methodologies. Through hands-on experience, students will implement and adapt machine learning algorithms for design automation and manufacturing tasks.

Teaching and learning methods:

The module consists of presentations on the topics followed by hands-on sessions/exercises. Main concepts are delivered through slide presentations and whiteboard sketches, and students will have the opportunity to deepen their knowledge through individual hands-on experiences with corresponding algorithms and tasks.

Media formats:

Lecture slides, software tools.

6.9 Phase Locked Loop/Clocked Circuits

Course Credits: 5

Semester: Winter Semester

Description of achievement and assessment methods:

If the number of exam participators is below 40, we will have a final oral exam (20 min/student). Otherwise, we will have a final 60 min written exam. The students should demonstrate their insight into the basic concepts and system structures of phase locked loop and clocked circuits, as well as their ability to apply them to practical circuit design.

(Recommended) requirements:

Solid-State and Semiconductor Device Physics, Analog and Mixed-Signal Electronics.

Contents:

a) Principle of Clocked Circuits
b) Frequency Domain versus Time Domain
c) Classes of Oscillators
d) Clock Generation
e) Clock Multiplication
f) Phase Locked Loop
    i. Modelling of Transfer Functions
    ii. Noise Analysis
    iii. System Considerations
    iv. Digital & Analog
    v. Integer & Fractional
g) Phase Control
    i. Digital Locked Loop
    ii. Interpolators

Learning Outcomes:

Upon successful completion of the module, students are able to understand the concepts and structures of PLL and Clocked Circuits. The students have an insight in the connection between theoretical modeling and the behavior of circuits. Students are able to describe basic interaction between circuit specifications and the sizing and structure of the circuit.

Teaching and learning methods:

The course will discuss the concepts and theory in weekly lectures with Q&A session at the beginning of each lecture. Exercises for self study are given weekly and will be discussed in a separate session. Here we will also use MATLAB to support better understanding of the interaction between theoretical modeling and the behavior of circuits build for practical applications.

Media formats:

MS PowerPoint Slides, Moodle, LTspice, Matlab

6.10 Power Management for Integrated Circuits

Course Credits: 5

Semester: Winter Semester

Description of achievement and assessment methods:

Students will be expected to demonstrate: A solid understanding of the fundamental concepts, architectures and design trade-offs in power-management integrated circuits (PMICs), including voltage regulators, switches, power sequencing and protection circuits. The ability to apply these concepts to the analysis and design of practical PMIC blocks (e.g., LDOs, switching converters, charge pumps) and system-level power-management strategies. The exam will consist of a mix of short-answer questions and design problems that test both theoretical insight and practical circuit-design skills in the context of power-management ICs. Short questions (definitions, conceptual explanations, etc.)Schematic analysis (critical evaluation of given PMIC topologies, etc.)Design tasks (sizing of components, stability considerations, efficiency calculations, etc.)

(Recommended) requirements:

- Electronic circuits.
- Analog and mixed-signal electronics.

Contents:

Introduction & Use Cases

- The role of PMICs in electrical systems: placed between the energy source and the electronics
- Key requirements: size, efficiency, reliability, noise performance, cost
- Fundamental topologies:
  - Linear regulator (resistive approach)
  - Inductor-based DC/DC converter (switch + inductor)
  - Switched-capacitor converter (switch + capacitor)
  - Hybrid converters (switch + capacitor + inductor)

- Power-management system architectures: multiple rails (e.g., Li-ion battery → DC/DC → analog LDO + digital LDO)

Power Devices & Components

- Power-stage configurations: low-side switch, high-side switch, half-bridge, full-bridge
- Non-ideal effects: RDSon, dropout, parasitic capacitances, body diode
- Loss mechanisms: conduction losses, switching losses
- Passive components: MOS, MOM, MIM capacitors; planar inductors
- Power transistors: DEMOS, DMOS; SOI technologies; latch-up; safe-operating area (SOA); dead-time generation

Linear Regulators – Basics

- Principle: voltage-controlled resistance
- Architecture: power transistor, sense resistor, reference voltage, error amplifier
- Dropout voltage (~100 mV)
- DC metrics: power efficiency, current efficiency, line regulation, load regulation
- Error amplifier specs: gain, slew rate, PSR
- Transient response: step changes in input voltage and load current

Linear Regulators – Advanced Topics

- Voltage-mode vs. current-mode control
- PMOS vs. NMOS power devices
- Stability analysis: poles/zeros, Miller compensation, zero-canceling resistor
- Slew-rate enhancement & dynamic biasing
- Noise & power-supply rejection considerations
- Over-charge protection, capacitor-less LDO designs

Protection & Reference Circuits

- Overvoltage, undervoltage, and overtemperature protection
- Bandgap voltage & current references
- Start-up circuits and power-on-reset
- Short-circuit and over-current protection

Switching Regulators – Fundamentals

- Comparison of switching vs. linear regulation
- Converter topologies: LC vs. capacitive-only
- Synchronous vs. non-synchronous DC/DC
- Voltage-mode vs. current-mode, Buck vs. Boost, PFM vs. PWM, CCM vs. DCM
- Inductive vs. inductorless operation
- Stability considerations

LC Buck Converter

- Inductor sizing & current ripple
- Schematic & operating principle
- Line & load regulation
- Efficiency, stability, noise & PSRR analysis

- Compensator design (Type I/II/III)

LC Boost Converter

- Inductor sizing & current ripple
- Schematic & operating principle
- Line & load regulation
- Efficiency, stability, noise & PSRR analysis
- Comparison to Buck topology

Capacitive Buck Converters (SCVR)

- Series-parallel topologies (Dickson, ladder, Fibonacci)
- Equivalent output resistance
- Flying-capacitor and switch sizing
- Efficiency and regulation methods

Capacitive Boost Converters (Charge Pumps)

- Diode-based vs. transistor-based charge pumps
- Stage cascading & closed-loop control

Near-Field Wireless Power Transfer

- Principles: electromagnetic vs. electrostatic induction
- Basic architectures & operating frequencies
- Near-field vs. far-field applications

Far-Field Wireless Power Transfer & RF Energy Harvesting

- RF energy harvesting for low-power IoT: impedance matching, rectification, MPPT– RF path loss and power budgeting
- Full chain: RFEH → Boost → MPPT → Buck → LDO
- Far-field application examples

Learning Outcomes:

Upon successful completion of the module, students will be able to:

- Understand fundamental PMIC concepts and architectures: Grasp the operating principles and block-level structures of power-management ICs, including low-dropout regulators(LDOs), switching converters, charge pumps, power sequencing and protection circuits.
- Connect theoretical models to real-world behavior: Develop insight into how small-signal and large-signal models predict regulator dynamics, stability margins andtransient response, and relate these models to measured circuit performance.
- Describe the interplay between specifications, sizing and topology: Explain how key requirements—output accuracy, load/regulation transients, efficiency and noise—drive choices in transistor sizing, passive component selection (inductors, capacitors) and overall PMIC topology.

Teaching and learning methods:

Throughout the semester, core power-management IC principles and theoretical frameworks are introduced in weekly lectures. Each week, students will receive self-study exercises—drawn from the lecture materials and selected reference papers—to deepen their understanding. These exercises are then reviewed and discussed in dedicated problem-solving sessions with Q&A. In addition, students are expected to engage in independent study of the provided lecture notes, tutorials and research articles to fully master the course content.

Media formats:

MS PowerPoint Slides, Moodle, LTspice, Matlab

6.11 Research Laboratory Functional Design of Integrated Analog and Mixed-Signal Circuits

Course Credits: 10

Semester: Summer Semester

Description of achievement and assessment methods:

The module is examined in the form of a project work (in a group of about 5 people) in which students demonstrate that they can develop the circuit diagram of an analog/mixed-signal circuit based on a given specification. They must also develop tests for the design and successfully carry them out according to their function. The individual design steps and the tests must be documented. The written documentation of the design and tests must be submitted at the end of the module and comprises approximately 5 pages per student, in which the individual contributions to the project are presented. Successful completion of the individual subtask must be demonstrated in order to pass the coursework.

The project work also includes a presentation in which the students demonstrate that they can present and discuss the project progress and results of the front-end design or the design and optimization strategies used in a clear and knowledgeable manner (in groups, approx. 20 - 25 minutes or approx. 5 minutes per student with subsequent discussion).

(Recommended) requirements:

Lab Analog RF Chip Design

Contents:

In this module, students work in groups to develop an analog/mixed-signal circuit at transistor level based on a specification. The students realize a circuit from a typical subject area (examples can be a PLL, an ADC or a DC/DC converter). Depending on the specific topic, students implement specific objectives in the development of the hardware. Examples may include the development of a design with the smallest possible surface area or the lowest possible power consumption.

The circuit is first described at transistor level using a design tool such as those offered by Cadence or Synopsis. The transistor netlist of the circuit is developed and the transistors are suitably dimensioned. Among other things, methods of mathematical modeling of circuits are used to arrive at an initial transistor level description from the specifications; in addition, knowledge acquired in lectures is applied to carry out an initial dimensioning of the circuits. In addition, students develop tests in the form of simulations (here, for example, simulation methods such as transient analysis or periodic steady-state analysis must be suitably selected and stimuli specified) to ensure the functional correctness of their circuit. As part of the test, students check compliance with specifications and optimize their circuit if necessary.

So-called Process Design Kits (PDKs) are used in the development of the circuits, which may be subject to special confidentiality and licensing regulations for commercial manufacturers and technologies currently used in industry. Students who wish to take advantage of the opportunity to have the developed chip manufactured after the second part of the internship may have to sign a corresponding agreement. Students who do not wish to sign this agreement can alternatively use a cell library adapted for teaching or an open source alternative; however, it will probably not be possible to offer industrial production at a later date. The intended learning outcomes (see above) can be achieved in both cases, regardless of the tools used.

In addition to the technical aspects mentioned, the seminar teaches principles and techniques for project planning and project work (e.g. project structure planning, creation of requirements catalogs and specifications) as well as group work, such as the structural prerequisites for group work or the method of moderation.

Learning Outcomes:

After successfully completing this module, students will have acquired the following skills:

- They are able to develop the circuit diagram of a sophisticated analog/mixed-signal circuit at transistor level based on given specifications.
- They are able to analyze the developed circuit with regard to its technical improvement potential and use the results of this analysis to improve the circuit.
- They are able to develop tests in the form of simulations and can use these to check the functional correctness and compliance with the specifications for the circuit at transistor level.
- They can apply the principles of group work (e.g. moderation methods) and project planning (e.g. project structure planning, creation of requirements catalogs and specifications) to design a project in the field of circuit design.
- They can clearly present the design and optimization strategies used in the process of logical design to a specialist audience.

Teaching and learning methods:

The basics for the lab are taught in approximately two introductory seminars. In particular, the work steps to be carried out, the specifications to be achieved and the design tools used are presented. In approximately three further seminars, content relating to group work and project planning is taught. The internship project is carried out independently in small groups of approx. 5 students in the scope of self-study in free time allocation. There is a regular exchange with a supervisor in the form of approx. weekly seminars and individual support and tutorials in the lab room (e.g. personal support in solving difficult problems in individual design); in this context, problems of the groups with the design and project organization are discussed and solved together.

Media formats:

Digital presentations and blackboard notes are used in the seminars. Students are also provided with documentation for the design tools used and the design tools themselves.

6.12 Research Laboratory Functional Design of Integrated Digital Circuits

Course Credits: 10

Semester: Summer Semester

Description of achievement and assessment methods:

The academic achievement is shown in form of a project work (in a group of around 5 people) in which students demonstrate that they can develop a suitable RTL description based on an algorithmic description and a given specifications and, from this, a netlist description of a digital circuit. In addition, functional tests for the design must be developed and successfully carried out. The design steps and the tests must be documented. The written documentation of the design and tests must be submitted at the end of the module and comprises approximately 5 pages per student, in which the individual contributions to the project are presented. Successful completion of the individual subtask must be demonstrated in order to pass the coursework.

The project work also includes a presentation in which the students demonstrate that they can present and discuss the progress and results of the project in a clear manner and in way showing their expertise (in groups, approx. 20 - 25 minutes or approx. 5 minutes per student with subsequent discussion).

(Recommended) requirements:
HDL Chip Design Laboratory

Contents:
In this module, students work in groups to create anregister transfer level (RTL) description of a circuit based on an algorithm and from that a gatelevel netlist of this circuit. As a starting point, the students receive a textual description of an algorithm from a current topic area (for example from the field of hardware security or hardware acceleration for artificial intelligence) as well as literature to familiarize themselves with the task. Depending on the relevant subject areas, students are also given specific objectives for the development of the hardware for these algorithms. Examples here could be the development of a design with the smallest possible surface area, the lowest possible power consumption or the lowest possible latency; objectives such as avoiding side-channel failure in circuits from the field of security can also be an objective.

Following the familiarization, the students first partition their design with regard to the division into hardware and software components and divide the complex design into manageable logical blocks, for example with the help of the Kactus 2 tool. For these blocks, the students develop an RTL description of the circuit. This is done using a hardware description language (VHDL, Verilog or SystemVerilog). In addition, students develop tests to ensure the functional correctness of their circuit and its sub-modules. These are developed in accordance with the Universal Verification Methodology (UVM), for example. Simulation tools from commercial providers such as Cadence, Siemens or Synopsys are used in the process of testing.

After successful testing, students synthesize their circuit into a gatelevel netlist with the help of a commercial design automation tool, such as those offered by Cadence or Synopsys, and use the corresponding tools from the same providers to prove that the circuit is functionally correct even after this step. The development or adaptation of scripts, e.g. in the language TCL, which is widely used in electronic design automation, is necessary for both steps. Students also check the synthesis reports of their circuit for compliance with specifications and optimize their circuit if necessary.

When synthesizing the RTL description, a so-called cell library is used, which may be subject to special confidentiality and licensing agreements in the case of commercial manufacturers and technologies used in today's industry. Students who wish to take advantage of the opportunity to have the developed chip manufactured after the second part of the practical course may have to sign a corresponding agreement. Students who do not wish to sign this agreement can alternatively use a cell library adapted for teaching or an open source alternative; the same learning outcomes are achieved; however, subsequent production cannot then be offered.

In addition to the technical aspects mentioned, students learn principles and techniques for project planning and project work (e.g. project structure planning, creation of requirements catalogs and specifications), as well as group work, such as the structural requirements for group work or the method of moderation.

Learning Outcomes:

After successfully completing this module, students will have acquired the following skills:

- They are able to develop a hardware description for a complex digital circuit starting from an abstract description and considering given specifications and apply the necessary steps to translate it into a description at netlist level.
- They are able to analyze the algorithmic description as well as their derived hardware description with regard to potential improvements and use the results of this analysis to improve their circuit.
- They are able to develop tests and can use them to check the functional correctness and compliance with the specifications for the circuit at RTL and netlist level.
- They understand the principles (e.g. methods of moderation) of group work and project planning (e.g. project structure planning, creation of requirements catalogs and specifications) and can apply these to design a project in the field of circuit design.
- They can clearly present the design and optimization strategies used in the process of logical design to a specialist audience.

Teaching and learning methods:

The basics for the lab are taught in approximately two introductory seminars. In particular, the work steps to be carried out, the specifications to be achieved and the design tools used are presented. In approximately three further seminars, content relating to group work and project planning is taught. The internship project is carried out independently in small groups of approx. 5 students in the scope of self-study in free time allocation. There is a regular exchange with a supervisor in the form of approx. weekly seminars and individual support and tutorials in the lab room (e.g. personal support in solving difficult problems in individual design); in this context, problems of the groups with the design and project organization are discussed and solved together.

Media formats:

Digital presentations and blackboard notes are used in the seminars. Students are also provided with documentation for the design tools used and the design tools themselves.

6.13 Research Laboratory Physical Design of Integrated Analog and Mixed-Signal Circuits

Course Credits: 5

Semester: Winter Semester

Description of achievement and assessment methods:

The academic achievements of this module are demonstrated in form of a project work (in a group of about 5 people) in which the students demonstrate that they can develop a GDSII description of a circuit based on the circuit diagram of an analog/mixed-signal circuit and specifications, which, in addition to the given specifications, takes into account all design rules necessary for a successful tape-out. In addition, tests for the implemented design must be successfully carried out and the design steps and tests must be documented. The written documentation of the design and tests must be submitted at the end of the module and comprises approximately 5 pages per student, in which the individual contributions to the project are presented. Successful completion of the individual subtask must be demonstrated in order to pass the coursework.

The project work also includes a presentation in which the students demonstrate that they can present and discuss the progress and results of the project regarding physical circuit design in a clear and informed manner (in groups, approx. 20 - 25 minutes or approx. 5 minutes per student with subsequent discussion).

(Recommended) requirements:

Research Laboratory Functional Design of Integrated Analog and Mixed-Signal Circuits

Contents:

The content of this module is the creation of a physical design from the schematic of an analog/mixed-signal circuit. Current design methods and design tools (e.g., tools for place and route), such as those offered by the companies Cadence and Synopsis, are used. Students develop the scripts required to carry out the design process (for example in the TCL scripting language commonly used in design automation) and work on the individual steps of the physical design flow, in particular floorplanning, analog placement and routing, physical verification and signoff. Students come into contact with design rules (e.g., regarding spacing of transistors) for the technology used and learn to control the design flow in such a way that the design rules are adhered to.

Manufacturer-specific process design kits (PDKs) are used for the steps required to create a chip suitable for production. For current industry-relevant technology nodes, one of which is also to be used in the planned chip production in the practical course, special license and non-disclosure agreements must be observed for access to this data, which must be signed by students. Students who do not wish to sign this agreement can alternatively work with a provided open source PDK or a PDK modified for teaching purposes, for which no signature is required. In this case, however, it will likely not be possible to produce the chip (in a fab). The targeted learning objectives can be achieved in both cases and independent from the used PDK.

Learning Outcomes:

After successfully completing this practical course, students will have acquired the following skills:

- They are able to develop a tape-out-capable description of a complex analog/mixed-signal circuit (GDSII) starting from a schematic and considering given specifications, applying technology-dependent design rules.
- They are able to check the functional correctness and compliance with the specifications of the developed hardware according to the physical design.
- You will be able to present the results of the back-end design steps clearly to a specialist audience and discuss them competently.

Teaching and learning methods:

In this module, students work in groups to develop from the schematic of a circuit that was designed by the same group in the "Research Laboratory Functional Design of Integrated Analog and Mixed-Signal Circuits", for example, into a description that can then be taped out. Typical specifications such as area or power consumption must be adhered to. The students deal with a sub-area of a more complex circuit and consider the design rules for a given production technology for this sub-area. The students also develop tests to ensure the functionality of the circuit and compliance with the specifications after the back-end design.

As an introduction, the basics for the practical course are taught in approximately two seminars. In particular, the work steps to be carried out, the specifications to be achieved and the design tools used are presented. The practical course is carried out independently in small groups of approx. 5 students with free time allocation. Support is provided in the form of regular exchanges with a supervisor, in the form of weakly seminars or individual support ant tutor sessions in the lab room (e.g., individual support for solving difficult problems in the student specific design); in this framework, problems of the groups with the design as well as with project organization will be discussed and jointly solved.

Media formats:

Digital presentations and blackboard notes are used in the seminars. Students are also provided with documentation for the design tools used and the design tools themselves.

6.14 Research Laboratory Physical Design of Integrated Digital Circuits

Course Credits: 5

Semester: Winter Semester

Description of achievement and assessment methods:

The academic achievement is shown in form of a project work (in a group of around 5 people) in which students demonstrate that they can develop a GDSII description of a circuit based on a netlist description of a digital circuit and specifications which, in addition to the given specifications, takes into account all design rules necessary for a successful tape-out. In addition, tests for the implemented design must be carried out successfully and the design steps and tests must be documented.The written documentation of the design and tests must be submitted at the end of the module and comprises approximately 5 pages per student, in which the individual contributions to the project on physical design are presented. Successful completion of the individual subtask must be demonstrated in order to pass the coursework.

The project work also includes a presentation in which the students demonstrate that they can present and discuss the progress and results of the project in a clear and informed manner (in groups, approx. 20 - 25 minutes or approx. 5 minutes per student with subsequent discussion).

Possibility of re-taking:
In the next semester: Yes
At the end of the semester: No

(Recommended) requirements:
Research Laboratory Functional Design of Integrated Digital Circuits

Contents:

The content of this module is the creation of a physical design from the gate netlist of a digital circuit. Current design methods and design tools (e.g. for automated placement and routing) are used, such as those provided by the commercial providers Cadence or Synopsis. Students develop the scripts required to carry out the design process (for example in the TCL scripting language commonly used in design automation) and work through the individual steps of the physical design flow, which essentially consist of the following sub-steps: Floorplanning, Placement and Routing, Clock Tree Synthesis, Timing Closure, Power Optmization, Design for Manufacturability and Physical Verification and Signoff. Students come into contact with design rules (e.g., regarding placement and alignment of logic cells) for the technology used and learn to control the design process in such a way that these design rules are adhered to.

Manufacturer-specific standard cell libraries and process design kits (PDKs) are used for the steps required to create a chip suitable for production. For current industry-relevant technology nodes, one of which is also to be used in the planned chip manufacturing in the practical course, special license and non-disclosure agreements must be accepted for access to this data and must be signed by students. Students who do not wish to sign such agreements can alternatively work with a provided open source PDK or a PDK modified for teaching purposes, for which no signature is required. In this case, however, it will likely not be possible to produce the chip. The targeted learning objectives can be achieved in both cases and independent from the used PDK.

Learning Outcomes:

After successfully completing this practical course, students will have acquired the following skills:

- They are able to develop a tape-out-capable description of a complex digital circuit (GDSII) based on a netlist description and taking into account given specifications, applying technology-dependent design rules.
- They are able to check the functional correctness and compliance with the specifications of the developed hardware according to the physical design.
- You will be able to present the results of the back-end design steps clearly to a specialist audience and discuss them competently.

Teaching and learning methods:

In this module, students working in groups are given the task of converting a netlist, which was developed by the same group in the “Research Laboratory Functional Design of Integrated

Digital Circuits", for example, into a description that can then be taped out. Typical specifications such as area or target frequency must be fulfilled. The students deal with a sub-area of a more complex circuit and consider the design rules for a given production technology for this sub-area. The students also develop tests to ensure the functionality of the circuit and compliance with the specifications after the back-end design.

As an introduction, the basics for the practical course are taught in approximately two seminars. In particular, the work steps to be carried out, the specifications to be achieved and the design tools used are presented. The practical course is carried out independently in small groups of approx. 5 students with free time allocation. Support is provided in the form of regular exchanges with a supervisor, in the form of weakly seminars or individual support ant tutor sessions in the lab room (e.g., individual support for solving difficult problems in the student specific design); in this framework, problems of the groups with the design as well as with project organization will be discussed and jointly solved.

Media formats:

Digital presentations and blackboard notes are used in the seminars. Students are also provided with documentation for the design tools used and the design tools themselves.