Master of Science in VLSI Design and Embedded Systems

Handbook

April 2026

Introduction

Master of Science in VLSI Design and Embedded Systems

This degree is designed for individuals who wish to deepen their knowledge of Very Large-Scale Integration (VLSI) and Embedded Systems design, along with their wide-ranging applications in industries such as semiconductors, telecommunications, consumer electronics, automotive systems, and the Internet of Things (IoT). It is particularly suitable for those who will have responsibility for planning, developing, and managing hardware and embedded solutions, as well as for individuals seeking to transition into roles in chip design, embedded software development, and hardware-software integration.

In all cases, students should be prepared to engage in substantial academic and technical studies. A prior degree in electronics, electrical engineering, or computer engineering is advantageous but not strictly required; however, the programme assumes technical aptitude, including a background in basic digital logic, computer architecture, and programming (e.g., C/C++, Python, or assembly), as well as fundamental mathematical skills relevant to hardware design and signal processing. The programme targets students with engineering, computer science, or STEM-related training or professional experience who are motivated to advance their expertise in VLSI design, verification, and embedded systems development.

Entry requirements

Education Requirements

Candidates will ordinarily hold an EQF Level 6 degree. Students without a technical background (either through their degree or through work experience) may be required to complete additional preparatory coursework. Candidates can be asked to demonstrate sufficient knowledge and skills through technical certifications or practical experience. Minimum technical expectations include: working knowledge of linear algebra, probability, and calculus; familiarity with digital design, combinational and sequential logic, microprocessors, and SoC fundamentals; prior programming experience (Python, C, or C++); and exposure to HDL (Verilog or VHDL) is beneficial.

Language Requirements

Applicants must demonstrate English proficiency at the equivalent standard of IELTS 6.5 or higher. The programme is taught in English.

Instructional design

Teaching: The programme combines asynchronous and synchronous components. Asynchronous elements include lecture videos, readings, and assignments to support diverse work-life schedules. Synchronous video sessions provide accountability and motivation, with students maintaining continuous access to teachers and peers through direct messaging and group chat. Modules are offered on a publicly advertised rolling-cohort schedule.

Assessment: Each module is assessed through two components: regular assignments (typically 50%) and a final assignment or project (typically 50%). All learning outcomes must be connected to at least one specific assignment, and students must be assessed on all learning outcomes to pass a module.

Degree structure

The degree consists of 30 available elective modules (5 ECTS each, all EQF Level 7) plus one compulsory Capstone Project (10 ECTS). Students select 18 elective modules to complete the 90 ECTS requirement.

Module	ECTS	Level	Type
VLSI Design Part 1	5	EQF 7	Elective
VLSI Design Part 2	5	EQF 7	Elective
Embedded System Design Part 1	5	EQF 7	Elective
Embedded System Design Part 2	5	EQF 7	Elective
SoC Design Part 1	5	EQF 7	Elective
SoC Design Part 2	5	EQF 7	Elective
Advanced SoC Design	5	EQF 7	Elective
Computer Architecture Part 1	5	EQF 7	Elective
Computer Architecture Part 2	5	EQF 7	Elective
Advanced Computer Architecture	5	EQF 7	Elective
Design Verification	5	EQF 7	Elective
Design for Testing	5	EQF 7	Elective
Physical Design	5	EQF 7	Elective

Advanced VLSI Design Part 1	5	EQF 7	Elective
Advanced VLSI Design Part 2	5	EQF 7	Elective
Advanced VLSI Design Part 3	5	EQF 7	Elective
Advanced VLSI Design Verification Part 1	5	EQF 7	Elective
Advanced VLSI Design Verification Part 2	5	EQF 7	Elective
Advanced VLSI Design for Testing Part 1	5	EQF 7	Elective
Advanced VLSI Design for Testing Part 2	5	EQF 7	Elective
Advanced VLSI Physical Design and Verification Part 1	5	EQF 7	Elective
Advanced VLSI Physical Design and Verification Part 2	5	EQF 7	Elective
Advanced IC Manufacturing, Packaging and Testing Part 1	5	EQF 7	Elective
Advanced IC Manufacturing, Packaging and Testing Part 2	5	EQF 7	Elective
Advanced Embedded System Design Part 1	5	EQF 7	Elective
Advanced Embedded System Design Part 2	5	EQF 7	Elective
Advanced Embedded System Design Part 3	5	EQF 7	Elective
Hands-on-Labs	5	EQF 7	Elective
Pilot Project	5	EQF 7	Elective
Design Automation	5	EQF 7	Elective
Capstone Project	10	EQF 7	Compulsory

Module Descriptions

1. VLSI Design Part 1

This course provides learners with the foundational knowledge and applied skills essential for careers in the semiconductor and embedded systems industry. The course integrates theoretical concepts with practical implementation, covering both front-end and back-end VLSI design methodologies. Learners are introduced to VLSI design principles, ASIC and FPGA workflows, and industry-standard verification processes. Building on digital logic fundamentals, the course incorporates the RISC-V Instruction Set Architecture, guiding students toward the Register Transfer Level (RTL) design and implementation of a pipelined RISC-V processor. In addition, learners explore CMOS basics, full-custom ASIC design strategies, and critical timing concepts. By connecting digital electronics theory with practical applications, the course enables students to design systems that are functionally correct, synthesizable, and optimized for performance and resource efficiency.

Learning Outcomes

1. Integrate theoretical VLSI concepts with practical workflows to design optimized digital systems for performance and area.
2. Collaborate in virtual design environments to solve VLSI implementation challenges using industry-standard tools.
3. Demonstrate the ability to critically assess and refine VLSI designs in alignment with professional standards of the semiconductor industry.

2. VLSI Design Part 2

This course provides a structured introduction to the Verilog Hardware Description Language (HDL), equipping learners with the ability to model, simulate, and design digital systems at the RTL level. The course begins with the fundamentals of Verilog, including modules, data types, procedural and continuous assignments, as well as behavioral and structural modeling. Learners will gain practical experience in hardware design environments through the use of Linux commands and the VI editor. Building on this foundation, the course advances to complex Verilog concepts such as tasks, functions, synthesizable constructs, and testbench development, with a strong emphasis on writing modular, reusable, and efficient code. Finally, learners are introduced to code coverage techniques, enabling them to evaluate verification quality and completeness effectively.

Learning Outcomes

1. Integrate advanced Verilog features to design scalable, reusable, and synthesizable RTL code.
2. Evaluate verification completeness using code coverage metrics and refine test strategies accordingly.
3. Demonstrate professional coding practices by adhering to industry standards for clarity, modularity, and efficiency in Verilog-based designs.

3. Embedded System Design Part 1

This course lays the foundation for developing and debugging modern embedded systems. The course begins with hardware system interfacing fundamentals and introduces learners to microprocessors and microcontrollers, with platforms such as STMicroelectronics serving as examples. Emphasis is placed on

embedded C programming for system-level applications. Learners also gain exposure to the ARM Instruction Set Architecture (ISA) and supporting toolchains, while exploring key quality attributes of embedded systems. Additionally, the course introduces debug interfaces and debuggers such as SWD and JTAG, providing practical insight into low-level programming, system control, and debugging practices essential for embedded system development.

Learning Outcomes

1. Integrate microcontroller platforms, embedded C, and debugging techniques to develop functional embedded applications.
2. Evaluate embedded systems for performance and quality, applying industry-relevant metrics and debugging workflows.
3. Demonstrate professional proficiency in low-level programming and system control for embedded applications.

4. Embedded System Design Part 2

This course builds on foundational concepts to provide learners with a comprehensive understanding of embedded peripheral interfaces from a programmer's perspective. The course explores the internal functionality and software-level integration of analog and digital peripherals, such as ADCs, DACs, sensors, buttons, LEDs, LCDs, and other input/output devices, using microcontroller platforms. Learners will gain practical experience in controlling and interfacing peripherals through embedded programming. The course also introduces the fundamentals of embedded networking, covering widely used communication protocols and real-time operating systems (RTOS) essentials.

Learning Outcomes

1. Integrate multiple peripheral devices with microcontrollers to design functional embedded applications.
2. Evaluate embedded networking solutions for IoT and real-time applications, considering performance, reliability, and protocol suitability.
3. Demonstrate professional proficiency in designing and debugging embedded systems that combine peripheral interfaces with communication protocols.

5. SoC Design Part 1

This course introduces learners to the architecture, design methodologies, and communication protocols central to modern System-on-Chip (SoC) development. The course begins with a detailed study of SoC architecture, emphasizing the integration of processing cores, memory subsystems, and hardware accelerators. Learners will gain an understanding of the complete SoC design flow, including specification, IP integration, and interconnect design. The course covers bus protocols (AXI, APB), on-chip communication standards, and system-level design methodologies that enable efficient, scalable hardware development.

Learning Outcomes

1. Integrate IP blocks, memory, and peripheral interfaces into a simplified SoC design using industry-relevant methodologies.
2. Evaluate processor–peripheral and wireless communication implementations for efficiency, scalability, and protocol compliance in SoC contexts.

3. Demonstrate the ability to design and refine SoC architectures that align with real-world performance, power, and area requirements.

6. SoC Design Part 2

This course provides an in-depth exploration of essential infrastructure components and software foundations in modern System-on-Chip (SoC) design. The course begins with a detailed study of memories and memory controllers, covering SRAM, DRAM, and Flash, along with interfacing methods and controller architectures designed to optimize latency, bandwidth, and power consumption. Learners will then examine Memory Management Units (MMUs) and virtual memory mechanisms such as address translation, page tables, and Translation Lookaside Buffers (TLBs), and their critical role in SoC systems. Additionally, the course explores cache architectures, coherency protocols, and storage integration within SoC designs.

Learning Outcomes

1. Integrate hardware components (memories, controllers, interrupt systems) with software layers (drivers, OS, toolchains) in SoC design workflows.
2. Evaluate SoC hardware-software interaction for efficiency, responsiveness, and power-performance trade-offs.
3. Demonstrate professional proficiency in designing SoC systems that balance hardware architecture with software abstraction layers to meet application-level requirements.

7. Advanced SoC Design

This course provides learners with a deep understanding of emerging innovations in System-on-Chip architectures, with a particular focus on multi-die and chiplet-based systems. The course begins with an overview of Systems of Chips, chiplets, and advanced integration techniques, emphasizing the industry transition from monolithic designs to modular and scalable architectures. A central theme is the Universal Chiplet Interconnect Express (UCIe), a new standard enabling seamless, high-bandwidth die-to-die communication across heterogeneous components. Learners will also engage with a case study on RISC-V SoC design, exploring open-source hardware configuration and ecosystem advantages.

Learning Outcomes

1. Integrate chiplet-based components and interconnect standards to design scalable multi-die SoC systems.
2. Evaluate trade-offs in SoC architectures with respect to performance, scalability, cost, and manufacturability in advanced semiconductor products.
3. Demonstrate the ability to apply advanced SoC design principles and open-source methodologies to real-world system architecture challenges.
4. Lead and manage iterative and agile hardware development cycles for complex SoC designs, coordinating collaborative tasks, integrating IP blocks, and managing interdependencies.
5. Demonstrate autonomy in selecting appropriate prototyping, verification, and integration strategies, including FPGA-based validation and collaborative multi-team coordination.
6. Design and justify accelerator-integrated SoC pipelines, assessing hardware–software interactions and performance trade-offs in AI-oriented workloads.

8. Computer Architecture Part 1

This course introduces learners to the fundamental and performance-driven principles of modern computer systems. The course begins with an exploration of processor instruction set architectures (ISAs), covering both RISC and CISC paradigms, as well as system-level models such as Von Neumann and Harvard architectures. Learners study essential components including address, data, and control buses, interrupt controllers, and Direct Memory Access (DMA), alongside the fundamentals of memory organization. The course then covers virtual memory concepts such as address translation, page tables, and Translation Lookaside Buffers, which are essential for modern processor design.

Learning Outcomes

1. Integrate concepts of ISAs, pipelining, memory systems, and cache design into a simplified processor model.
2. Evaluate architectural trade-offs between performance, efficiency, and scalability in processor design and memory hierarchy.
3. Demonstrate professional-level understanding by applying architectural concepts to real-world case studies and system-level challenges.

9. Computer Architecture Part 2

This course offers an in-depth study of two major processor architectures: ARM and RISC-V. It begins with the core principles of reduced instruction set computing (RISC), providing students with a solid understanding of their architectural foundations, instruction sets, and pipeline structures. Learners will then explore ARM-specific features, including software interfaces, exception handling, and peripheral programming, reinforced through practical coding exercises. The course advances into topics such as ARM TrustZone, memory protection units, and performance monitoring, highlighting security and optimization techniques. Comparative discussions of ARM and RISC-V design philosophies and ecosystem support prepare learners for embedded system development and System-on-Chip (SoC)-level integration using both architectures. This module also introduces advanced architectural features including low-level hardware virtualization concepts.

Learning Outcomes

1. Evaluate trade-offs between ARM and RISC-V architectures when selecting processors for embedded or SoC design projects.
2. Integrate ARM and RISC-V architectural concepts into embedded systems design workflows for practical applications.
3. Critically assess processor security and performance requirements and propose design solutions using TrustZone or RISC-V alternatives.

10. Advanced Computer Architecture

This course offers an advanced study of the RISC-V architecture, with a focus on system-level design and specialized extensions. Learners will develop a detailed understanding of privilege modes, memory protection, exception and interrupt handling, and system programming within RISC-V. Emphasis is placed on practical engagement with RISC-V toolchains, debugging methods, and application development for both embedded and general-purpose platforms. The course also introduces key ISA specifications such as PLIC, Debug, and Trace, alongside support for extensions including vector, atomic, floating-point, bit manipulation, and custom ISAs. By the end of the course, students will be equipped to leverage the flexibility of the open RISC-V ecosystem to design high-performance and domain-specific

applications. This course explores contemporary processor architectures with emphasis on hardware virtualization.

Learning Outcomes

1. Evaluate system design trade-offs when selecting RISC-V ISA extensions for high-performance and domain-specific applications.
2. Integrate advanced RISC-V architectural features into embedded or general-purpose system workflows.
3. Design and justify custom ISA extensions to address specific application or domain requirements within the RISC-V ecosystem.
4. Critically assess advanced processor architectures supporting virtualisation, GPU-based parallelism, and heterogeneous computing, making informed architectural recommendations.

11. Design Verification

This course offers an in-depth exploration of SystemVerilog (SV) for both RTL design and functional verification, building a strong link between hardware description and validation. The course begins with RTL design using SystemVerilog, emphasizing synthesizable constructs, hierarchical design, and best practices for writing modular and reusable code. It then shifts focus to SystemVerilog as a Hardware Verification Language (HVL), introducing advanced verification techniques such as object-oriented programming, constrained randomization, and functional coverage. Learners will also gain proficiency in writing SystemVerilog Assertions (SVA) for property checking and applying formal verification to ensure design correctness early in the development cycle. Combining theoretical insights with hands-on practice, this course equips students with the ability to design reliable hardware and build efficient, assertion-based verification environments.

Learning Outcomes

1. Evaluate design robustness by analyzing verification results using coverage metrics and assertion outcomes.
2. Integrate SystemVerilog RTL and HVL techniques to build scalable verification environments.
3. Design and apply formal verification flows to identify design flaws early in the development cycle.
4. Demonstrate professional autonomy in constructing object-oriented, constrained-random testbenches using industry-standard SystemVerilog verification methodologies.

12. Design for Testing

This course provides a thorough introduction to Design for Testability (DFT) techniques essential in modern VLSI design for ensuring manufacturability and achieving high fault coverage. Learners will study fundamental concepts such as Automatic Test Pattern Generation (ATPG), fault models (stuck-at, transition, bridging), scan chain insertion, test compression, boundary scan (IEEE 1149.1), and test coverage analysis. The course integrates both theoretical foundations and hands-on practice, allowing students to implement and verify test structures through lab-based exercises. By the end of the course, learners will be equipped with the knowledge and skills to design circuits for testability, evaluate coverage metrics, and apply industry-standard DFT methodologies to practical silicon design challenges.

Learning Outcomes

1. Evaluate trade-offs between fault coverage, design complexity, and test overhead in real-world DFT applications.
2. Integrate DFT techniques such as scan chains, ATPG, and boundary scan into digital design workflows.
3. Demonstrate professional competency in applying DFT methodologies to meet industry standards for silicon testability and fault coverage.

13. Physical Design

This course offers a comprehensive study of Physical Synthesis and Physical Design in the VLSI implementation flow, emphasizing the transformation of synthesized RTL into optimized and manufacturable layouts. Students will explore timing-driven placement, logic restructuring, and multi-objective optimization for area, power, and performance. The course further examines the complete Physical Design flow, covering floorplanning, placement, clock tree synthesis (CTS), routing, and signoff checks such as timing analysis. Through practical labs and industry-standard tool exercises, learners will develop the ability to perform both block-level and full-chip physical design, ensuring compliance with stringent fabrication and performance requirements.

Learning Outcomes

1. Evaluate different physical design methodologies for efficiency, scalability, and design closure.
2. Integrate physical synthesis and physical design techniques into an end-to-end VLSI implementation flow.
3. Design and optimize physical layouts that meet performance, power, and area targets while satisfying manufacturability constraints.
4. Demonstrate professional autonomy in executing full-chip physical design flows using industry-standard EDA tools.

14. Advanced VLSI Design Part 1

This course provides an in-depth study of front-end design challenges and solutions in advanced digital systems, with emphasis on Clock Domain Crossing (CDC), Reset Domain Crossing (RDC), low-power design methodologies, and RTL signoff techniques. Learners will explore methods to identify and resolve CDC and RDC issues for robust data transfer across asynchronous domains. The course also examines low-power design strategies, such as clock gating, power gating, multi-voltage operation, and UPF-based power intent specification. In addition, students will gain hands-on experience with RTL linting, logic synthesis, and formal equivalence checking, ensuring that RTL designs are functionally accurate, optimized, and ready for physical implementation. Through a mix of theoretical concepts and tool-driven labs, learners will build the ability to address design quality, power efficiency, and correctness in complex SoC and ASIC development.

Learning Outcomes

1. Evaluate design choices for robustness, low-power efficiency, and correctness in advanced SoC and ASIC development.
2. Integrate CDC/RDC analysis, low-power strategies, and RTL signoff techniques into advanced digital design workflows.
3. Demonstrate professional autonomy in applying advanced front-end methodologies to real-world VLSI design challenges.

15. Advanced VLSI Design Part 2

This course offers an in-depth exploration of FPGA-based digital system design, starting with the fundamentals and progressing to advanced implementation techniques. Learners will study FPGA architectures, design flows, and hardware description languages (Verilog/VHDL) to model and realize digital circuits on reconfigurable hardware. The curriculum extends to advanced topics such as IP integration, high-speed interfaces, clocking strategies, and design optimization with respect to performance, power, and area. Students will also engage with specialized FPGA features, including embedded processors, DSP blocks, and high-level synthesis (HLS) for design acceleration. Through hands-on labs and project-based activities, participants will acquire practical expertise in developing, verifying, and deploying complex FPGA-based designs, preparing them for applications in prototyping, embedded systems, and hardware acceleration.

Learning Outcomes

1. Evaluate FPGA design solutions for efficiency, scalability, and suitability in real-world applications such as prototyping and hardware acceleration.
2. Integrate advanced FPGA features (embedded processors, DSP blocks, HLS) into comprehensive digital system designs.
3. Demonstrate professional proficiency in developing and deploying FPGA-based systems through hands-on labs and project-based exercises.

16. Advanced VLSI Design Part 3

This course provides a comprehensive study of Analog and Mixed-Signal (AMS) Integrated Circuit (IC) Design, combining theoretical foundations with advanced design methodologies. Learners will explore the principles and design of key analog circuits, including operational amplifiers (op-amps), voltage references, oscillators, phase-locked loops (PLLs), analog-to-digital converters (ADCs), and digital-to-analog converters (DACs). The course further examines mixed-signal design challenges such as noise coupling, substrate interference, and layout techniques for high-performance analog circuits. Advanced topics include RF circuit design fundamentals, high-speed data converter design, and power management ICs.

Learning Outcomes

1. Evaluate analog and AMS circuit designs for trade-offs involving power, performance, and noise.
2. Integrate analog and mixed-signal design methodologies to develop complete AMS IC solutions.
3. Demonstrate professional competency in designing and analysing complex analog circuits using industry-standard simulation tools.

17. Advanced VLSI Design Verification Part 1

This course introduces students to advanced verification and debugging techniques within the digital design flow, with a particular emphasis on Gate-Level Simulation (GLS), timing analysis, and power verification. Learners will explore GLS methodologies, including back-annotated simulation and timing-aware verification, to validate designs post-synthesis and post-place-and-route. The course also covers X-state propagation analysis, glitch detection, and techniques for resolving common simulation failures.

Learning Outcomes

1. Evaluate GLS results to identify and resolve issues related to timing, glitches, and X-propagation.
2. Integrate GLS, STA, and power verification techniques into a comprehensive post-synthesis verification workflow.
3. Demonstrate professional autonomy in diagnosing and resolving complex verification challenges arising from gate-level simulation and timing closure activities.

18. Advanced VLSI Design Verification Part 2

This course delves into power-aware verification methodologies that are crucial for developing modern, energy-efficient integrated circuits. With power consumption being a critical constraint in today's semiconductor designs, this course equips learners with the tools, techniques, and methodologies needed to verify low-power designs effectively. Students will explore UPF-based power intent specification, multi-voltage verification, power domain isolation, retention strategies, and power-aware simulation.

Learning Outcomes

1. Evaluate design correctness under multiple power scenarios using UPF-based verification environments.
2. Integrate power-aware verification techniques, including isolation, retention, and multi-voltage checks, into a complete low-power design verification flow.
3. Demonstrate professional autonomy in applying UPF strategies and power verification methodologies to ensure compliance with low-power design intent.

19. Advanced VLSI Design for Testing Part 1

This course offers an in-depth study of digital design implementation and testing methodologies, blending theoretical concepts with practical training in industry-standard EDA tools. The course focuses on advanced scan-based DFT techniques, Built-In Self-Test (BIST) for logic (LBIST) and memory (MBIST), and the integration of Design-for-Debug (DfD) strategies using embedded logic analysis tools.

Learning Outcomes

1. Integrate LBIST and MBIST strategies to enhance the reliability of digital systems under real-world conditions.
2. Evaluate DFT and DfD techniques for their effectiveness in improving test coverage, debug capability, and silicon reliability.
3. Demonstrate professional autonomy in applying advanced scan, BIST, and debug methodologies to complex digital design challenges.

20. Advanced VLSI Design for Testing Part 2

This course delves into advanced Design-for-Testability (DFT) strategies tailored for low-power and complex SoC architectures. It emphasizes power-aware test techniques, hierarchical DFT for large SoCs, test data volume compression, IEEE standards compliance, and emerging test methodologies for advanced packaging and chiplet-based designs.

Learning Outcomes

1. Evaluate trade-offs between power efficiency, test coverage, and test time when applying advanced DFT techniques in SoC designs.

2. Integrate hierarchical DFT strategies and IEEE 1687/1500 standards into complex, multi-die SoC test architectures.
3. Demonstrate professional autonomy in developing and justifying DFT solutions for advanced packaging and chiplet-based semiconductor products.

21. Advanced VLSI Physical Design and Verification Part 1

This course provides an in-depth understanding of advanced Static Timing Analysis (STA) and Physical Verification processes essential for ensuring both the performance and manufacturability of modern IC designs. Students will explore advanced STA concepts, including multi-corner multi-mode (MCMM) analysis, on-chip variation (OCV), advanced OCV (AOCV), and parametric OCV (POCV), as well as the impact of process, voltage, and temperature (PVT) variations. The course also covers comprehensive physical verification flows, including DRC, LVS, and ERC checks.

Learning Outcomes

1. Evaluate design trade-offs and propose solutions to timing and physical verification challenges in complex digital IC designs.
2. Integrate advanced STA methodologies (MCMM, OCV, AOCV, POCV) into timing closure workflows for high-performance ICs.
3. Demonstrate professional autonomy in executing and interpreting physical verification results (DRC, LVS, ERC) to ensure IC manufacturability and compliance.

22. Advanced VLSI Physical Design and Verification Part 2

This course delves into advanced practices in digital IC implementation, with emphasis on power-aware physical design, AI-driven optimization, logic equivalence checking, and formal sign-off methodologies. Students will explore power intent implementation using UPF, techniques for IR drop analysis, electromigration mitigation, and power mesh optimization. The course also examines AI and machine learning applications in EDA, covering automated placement, routing optimization, and design space exploration.

Learning Outcomes

1. Evaluate the effectiveness of AI-driven optimization techniques in addressing timing closure challenges for complex SoCs.
2. Integrate power-aware physical design techniques (UPF, IR drop, EM mitigation) into end-to-end IC implementation flows.
3. Demonstrate professional autonomy in applying formal equivalence checking and sign-off methodologies to ensure logical and physical correctness of advanced IC designs.

23. Advanced IC Manufacturing, Packaging and Testing Part 1

This course offers an in-depth introduction to the hardware foundations of modern electronics, focusing on Printed Circuit Board (PCB) design, semiconductor manufacturing, and IC fabrication processes. Learners will explore PCB design principles and layout techniques, including signal integrity, EMC considerations, and design for manufacturability.

Learning Outcomes

1. Evaluate PCB design choices to balance manufacturability, performance, and signal integrity trade-offs.
2. Integrate semiconductor fabrication knowledge into IC design decisions to improve yield and reliability.
3. Demonstrate professional competency in applying PCB design and semiconductor manufacturing principles to real-world electronics engineering challenges.

24. Advanced IC Manufacturing, Packaging and Testing Part 2

This course provides a comprehensive exploration of the post-silicon phase of semiconductor development, emphasizing Packaging Design, Package Assembly, and Test Engineering. Learners will examine advanced IC packaging technologies, including flip-chip, wafer-level packaging, and 3D/2.5D integration techniques. The course also covers post-silicon validation, bring-up procedures, functional verification, and failure analysis methodologies.

Learning Outcomes

1. Evaluate trade-offs in packaging design decisions with respect to manufacturability, reliability, and performance in advanced IC products.
2. Integrate post-silicon validation and failure analysis techniques into semiconductor test and packaging workflows.
3. Demonstrate professional autonomy in selecting and justifying packaging and test strategies for complex, multi-die semiconductor products.

25. Advanced Embedded System Design Part 1

This course provides an in-depth exploration of advanced embedded system design, integrating both hardware and software perspectives. Learners will engage with Linux-based embedded programming environments, exploring kernel configuration, device driver development, and system-level software integration. The course also addresses real-time operating systems (RTOS), focusing on task scheduling, interrupt management, and inter-process communication.

Learning Outcomes

1. Develop basic embedded applications using Linux-based programming environments.
2. Integrate Linux kernel modules and device drivers into embedded hardware platforms.
3. Demonstrate professional competency in designing and deploying RTOS-based embedded systems with real-time constraints.

26. Advanced Embedded System Design Part 2

This course provides learners with the expertise to design and implement intelligent embedded solutions for edge and IoT environments. Starting with cloud and fog computing architectures, the course explores sensor fusion, data acquisition, and wireless communication protocols critical to IoT deployments. Learners will then examine machine learning deployment on embedded platforms, covering model optimization, neural network inference engines, and hardware accelerators for edge AI.

Learning Outcomes

1. Evaluate security, scalability, and performance trade-offs in designing intelligent edge and IoT systems.
2. Integrate cloud/fog computing, sensor fusion, and wireless protocols into cohesive IoT system architectures.
3. Demonstrate professional competency in deploying machine learning models on edge devices with real-time constraints and resource limitations.

27. Advanced Embedded System Design Part 3

This course emphasizes the application of embedded systems across specialized domains such as robotics, automotive, healthcare, and RF technologies. Learners will explore domain-specific embedded system requirements, safety standards, and regulatory frameworks applicable to these fields. The course covers robotics control systems, sensor integration for autonomous navigation, automotive embedded systems (AUTOSAR, functional safety), and healthcare device design.

Learning Outcomes

1. Evaluate embedded solutions for compliance, safety, and performance across healthcare, robotics, and automotive domains.
2. Integrate perception, signal integrity, and cryptographic methods into a domain-specific embedded application design.
3. Design a domain-driven embedded system prototype (simulation or project-based) addressing real-world constraints such as safety, compliance, and robustness.
4. Demonstrate responsibility for deploying and validating embedded systems on FPGA or equivalent prototyping platforms, ensuring correct functionality under domain-specific constraints.
5. Exercise professional judgement in balancing timing, power, and resource constraints during embedded system implementation and testing.
6. Evaluate and integrate NPU-based accelerator subsystems within embedded platforms, considering real-time constraints, memory hierarchies, and HW–SW co-design principles.

28. Hands-on-Labs

This course offers immersive, project-based practice in VLSI and embedded systems, allowing students to translate theoretical knowledge into real-world applications. Students will work with industry-standard EDA tools and embedded platforms to complete end-to-end design, verification, and testing projects.

Learning Outcomes

1. Evaluate end-to-end design flows for efficiency, accuracy, and industry alignment using lab projects.
2. Integrate hardware and software components in hands-on VLSI and embedded systems projects.
3. Demonstrate professional competency in using industry-standard tools to complete complex design challenges.

29. Pilot Project

This course provides an applied, integrative experience across semiconductor and embedded system design, bridging theory with real-world practice. Students undertake a structured project covering design

specification, implementation, verification, and final presentation, guided by milestones and faculty mentorship.

Learning Outcomes

1. Integrate design, verification, and testing techniques into a coherent, milestone-driven pilot project.
2. Evaluate project outcomes against industry standards, identifying strengths and areas for improvement.
3. Demonstrate professional competency in planning and executing a structured technical project from specification to final presentation.
4. Collaborate effectively with peers and mentors throughout the project lifecycle.

30. Design Automation

This course equips learners with the scripting and version control skills essential for streamlining modern VLSI design workflows. Beginning with Shell scripting and Python for automation, the course progresses to advanced topics including Makefile-driven build systems, TCL scripting for EDA tool integration, and CI/CD pipelines for hardware design.

Learning Outcomes

1. Integrate scripting and version control tools to design end-to-end automation solutions for VLSI workflows.
2. Evaluate automation strategies for efficiency, maintainability, and scalability in complex hardware design environments.
3. Demonstrate professional autonomy in applying design automation techniques to reduce manual effort and improve consistency in VLSI development pipelines.

31. Capstone Project

This course serves as the culminating experience in the VLSI and Embedded Systems programme, designed to replicate real-world industry challenges while demonstrating the student's comprehensive technical and professional competencies. Students will independently define, design, implement, verify, and present a complete project in VLSI or embedded systems, applying knowledge and skills gained throughout the programme.

Learning Outcomes

1. Integrate technical knowledge, creativity, and project management skills to deliver a complete, production-grade solution.
2. Collaborate effectively in distributed teams by engaging in milestone reviews, peer feedback, and joint problem-solving.
3. Present project outcomes through a functional prototype and a professional technical presentation, demonstrating readiness for industry roles.
4. Demonstrate full professional autonomy by planning, executing, and delivering a complex VLSI or embedded systems project using collaborative tools, version control, and structured engineering methodologies.
5. Take responsibility for making and justifying engineering trade-offs involving performance, power, cost, timing, and system reliability in a real-world project context.

Internships policy

Internships must be a genuine extension of the student's academic programme, providing opportunity to apply theoretical knowledge to substantive projects directly related to their field of study. Internships consisting primarily of administrative or routine tasks will not be approved.

Every internship must have a defined start date, end date, and formal learning plan with objectives agreed in advance by the student, the host organisation, and the relevant college. Responsibilities and task complexity should increase over time. Each student must be assigned a named supervisor within the host organisation who holds relevant expertise and is responsible for providing regular guidance and feedback.

Woolf prioritises paid internships to ensure equitable access regardless of socioeconomic background. Unpaid internships may only be approved where they constitute a genuine learning opportunity and do not displace the work of a paid employee.

Programmatic standards

Day-to-day management sits with the relevant college. Each college must have a designated Woolf contact responsible for vetting and approving all host organisations and placements before any internship may proceed. Colleges are responsible for matching students to approved positions.

Students must complete pre-internship preparation before commencing a placement, which may include CV writing, interview support, and other instruction as necessary. Virtual internships are encouraged to widen access beyond geographical constraints; support systems must address the challenges of remote work, including cross-timezone communication and fostering professional belonging.

Programme effectiveness must be evaluated on an ongoing basis. Formal evaluations will be collected from students, host supervisors, and academic advisors, and will inform curriculum design and programme improvement.

Grading Scheme

General Marking Criteria and Classification

Marking of student work keeps in view the scale of work that the student can reasonably be expected to have undertaken in order to complete the task.

The assessment of work for the course is defined according to the following rubric of general criteria:

1. **Engagement:**
    - Directness of engagement with the question or task
    - Range of issues addressed or problems solved
    - Depth, complexity, and sophistication of comprehension of issues and implications of the questions or task
    - Effective and appropriate use of imagination and intellectual curiosity
2. **Argument or solution:**
    - Coherence, mastery, control, and independence of work
    - Conceptual and analytical precision

    - Flexibility, i.e., discussion of a variety of views, ability to navigate through challenges in creative ways
    - Completion leading to a conclusion or outcome
    - Performance and success of the solution, where relevant
3. **Evidence (as relevant):**
    - Depth, precision, detail, range and relevance of evidence cited
    - Accuracy of facts
    - Knowledge of first principles and demonstrated ability to reason from them
    - Understanding of theoretical principles and/or historical debate
    - Critical engagement with primary and/or secondary sources
4. **Organisation & Presentation:**
    - Clarity and coherence of structure
    - Clarity and fluency of writing, code, prose, or presentation (as relevant)
    - Correctness of conformity to conventions (code, grammar, spelling, punctuation, or similar relevant conventions)

Definition of marks

97-100

Work will be so outstanding that it could not be better within the scope of the assignment. These grades will be used for work that shows exceptional excellence in the relevant domain; including (as relevant): remarkable sophistication and mastery, originality or creativity, persuasive and well-grounded new methods or ideas, or making unexpected connections or solutions to problems.

94-96

Work will excel against each of the General Criteria. In at least one area, the work will be merely highly competent.

90-93

Work will excel in more than one area, and be at least highly competent in other respects. It must be excellent and contain: a combination of sophisticated engagement with the issues; analytical precision and independence of solution; go beyond paraphrasing or boilerplate code techniques; demonstrating quality of awareness and analysis of both first principles or primary evidence and scholarly debate or practical tradeoffs; and clarity and coherence of presentation. Truly outstanding work measured against some of these criteria may compensate for mere high competence against others.

87-89

Work will be at least very highly competent across the board, and excel in at least one group of the General Criteria. Relative weaknesses in some areas may be compensated by conspicuous strengths in others.

84-86

Work will demonstrate considerable competence across the General Criteria. They must exhibit some essential features of addressing the issue directly and relevantly across a good range of aspects; offer a coherent solution or argument involving (where relevant) consideration of alternative approaches; be

substantiated with accurate use of resources (including if relevant, primary evidence) and contextualisation in debate (if relevant); and be clearly presented. Nevertheless, additional strengths (for instance, the range of problems addressed, the sophistication of the arguments or solutions, or the use of first principles) may compensate for other weaknesses.

80-83

Work will be competent and should manifest the essential features described above, in that they must offer direct, coherent, substantiated and clear arguments; but they will do so with less range, depth, precision and perhaps clarity. Again, qualities of a higher order may compensate for some weaknesses.

77-79

Work will show solid competence in solving problems or providing analysis. But it will be marred by weakness under one or more criteria: failure to fully solve the problem or discuss the question directly; some irrelevant use of technologies or citing of information; factual error, or error in selection of technologies; narrowness in the scope of solution or range of issues addressed or evidence adduced; shortage of detailed evidence or engagement with the problem; technical performance issues (but not so much as to prevent operation); poor organisation or presentation, including incorrect conformity to convention or written formatting.

74-76

Work will show evidence of some competence in solving problems or providing analysis. It will also be clearly marred by weakness in multiple General Criteria, including: failure to solve the problem or discuss the question directly; irrelevant use of technologies or citing of information; factual errors or multiple errors in selection of technologies; narrowness in the scope of solution or range of issues addressed or evidence adduced; shortage of detailed evidence or engagement with the problem; significant technical performance issues (but not so much as to prevent operation); poor organisation or presentation, including incorrect conformity to convention or written formatting. They may be characterised by unsubstantiated assertion rather than argument, or by unresolved contradictions in the argument or solution.

70-73

Work will show evidence of competence in solving problems or providing analysis, but this evidence will be limited. It will be clearly marred by weakness in multiple General Criteria. It will still make substantive progress in addressing the primary task or question, but the work will lack a full solution or directly address the task; the work will contain irrelevant material; the work will show multiple errors of fact or judgment; and the work may fail to conform to conventions.

67-69

Work will fall down on a number of criteria, but will exhibit some of the qualities required, such as the ability to grasp the purpose of the assignment, to deploy substantive information or solutions in an effort to complete the assignment; or to offer some coherent analysis or work towards the assignment. Such qualities will not be displayed at a high level, and may be marred by irrelevance, incoherence, major technical performance issues, error and poor organisation and presentation.

64-66

Work will fall down on a multiple General Criteria, but will exhibit some vestiges of the qualities required, such as the ability to see the point of the question, to deploy information, or to offer some coherent work. Such qualities will be substantially marred by irrelevance, incoherence, error and poor organisation and presentation.

60-63

Work will display a modicum of knowledge or understanding of some points, but will display almost none of the higher qualities described in the criteria. They will be marred by high levels of factual or technology error and irrelevance, generalisation or boilerplate code and lack of information, and poor organisation and presentation.

0-60

Work will fail to exhibit any of the required qualities. Candidates who fail to observe rubrics and rules beyond what the grading schemes allow for may also be failed.

Indicative equivalence table

US GPA	US Grade	US Percent	UK Mark	UK UG Classification	UK PG Classification	Malta Grade	Malta Mark	Malta Classification	Swiss Grade
4	A+	97 - 100	70+	First	Distinction	A	80-100%	First class honours	6.0
3.9	A	94-96				B	70-79%	Upper-second class honours	
3.7	A-	90-93							5.5
3.3	B+	87-89	65-69	Upper Second	Merit	C	55-69%	Lower-second class honours	
3	B	84-86	60-64						
2.7	B-	80-83	55-59	Lower Second	Pass				5
2.3	C+	77-79	50-54			D	50-54%	Third-class honours	
2	C	74–76	45-49	Third	Pass				
1.7	C-	70–73	40-44						
1.3	D+	67–69	39-	Fail	Fail				
1	D	64–66							
0.7	D-	60–63							
0	F	Below 60				F			

Synchronous Adjustments Template

Synch discussions may affect the mark on submitted assignments: written work is submitted in advance, and a discussion follows. This provides students an opportunity to clarify and explain their written claims, and it also tests whether the work is a product of the student’s own research or has been plagiarised.

The synchronous discussion acts to shift the recorded mark on the submitted assignment according to the following rubric:

+3

Up to three points are added for excellent performance; the student displays a high degree of competence across a range of questions, and excels in at least one group of criteria. Relative weaknesses in some areas may be compensated by conspicuous strengths in others.

+/- 0

The marked assignment is unchanged for fair performance. Answers to questions must show evidence of some solid competence in expounding evidence and analysis. But they will be marred by weakness under one or more criteria: failure to discuss the question directly; appeal to irrelevant information; factual error; narrowness in the range of issues addressed or evidence adduced; shortage of detailed evidence; or poor organisation and presentation, including consistently incorrect grammar. Answers may be characterised by unsubstantiated assertion rather than argument, or by unresolved contradictions in the argument.

- 3 (up to three points)

Up to three are subtracted points for an inability to answer multiple basic questions about themes in the written work. Answers to questions will fall down on a number of criteria, but will exhibit some vestiges of the qualities required, such as the ability to see the point of the question, to deploy information, or to offer some coherent analysis towards an argument. Such qualities will not be displayed at a high level or consistently, and will be marred by irrelevance, incoherence, error and poor organisation and presentation.

0 (fail)

Written work and the oral examination will both be failed if the oral examination clearly demonstrates that the work was plagiarised. The student is unfamiliar with the arguments of the assignment or the sources used for those arguments.

Plagiarism

Plagiarism is the use of someone else’s work without correct referencing. The consequence of plagiarism is the presentation of someone else’s work as your own work. Plagiarism violates Woolf policy and will result in disciplinary action, but the context and seriousness of plagiarism varies widely. Intentional or reckless plagiarism will result in a penalty grade of zero, and may also entail disciplinary penalties.

Plagiarism can be avoided by citing the works that inform or that are quoted in a written submission. Many students find that it is essential to keep their notes organised in relation to the sources which they summarise or quote. Course instructors will help you to cultivate professional scholarly habits in your academic writing.

Depending on the course, short assignment essays may not require students to submit a bibliography or to use extensive footnotes, and students are encouraged to write their assignments entirely in their own words. However, all essays must acknowledge the sources on which they rely and must provide quotation marks and citation information for verbatim quotes.

There are several forms of plagiarism. They all result in the presentation of someone’s prior work as your new creation. Examples include:

- Cutting and pasting (verbatim copying)
- Paraphrasing or rewording
- Unauthorised Collaboration
- Collaboration with other students can result in pervasive similarities – it is important to determine in advance whether group collaboration is allowed, and to acknowledge the contributions or influence of the group members.
- False Authorship (Essay Mills, Friends, and Language Help)
- Paying an essay writing service, or allowing a generous friend to compose your essay, is cheating. Assistance that contributes substantially to the ideas or content of your work must be acknowledged.

Complaints and appeals

Students and faculty should always seek an amicable resolution to matters arising by addressing the issue with the person immediately related to the issue. Students should handle minor misunderstandings or disagreements within a regular teaching session or by direct message, or with their College. If a simple resolution is not possible, or the matter remains unresolved for one party, the steps outlined in this section apply to all groups, colleges, and units of Woolf.

The Red Flag system

An issue with a red flag should be submitted in the case that a member of Woolf seeks to make an allegation of serious misconduct about another member, including matters of cheating, plagiarism, and unfair discrimination or intolerance.

Any member of Woolf, seeking to raise a matter of serious concern, should submit a red flag by emailing redflag@woolf.education. Provide a short, clear description of the issue.

If a student submits an issue with a red flag, or if a faculty member submits an issue about a student, it will trigger a meeting with the student's College Advisor. If the issue is not resolved, the matter will be escalated to the College Dean, or to a committee designated by the College Dean, which will have the power to clear the flag.

If an issue is submitted with a red flag by a faculty member about another faculty member, then the issue is reported directly to the College Dean.

For both students and faculty members, after the Dean's decision, the one who submits the complaint is provided the opportunity to accept or appeal the decision; if the one submitting the issue appeals the decision, it will be assigned to the Quality Assurance, Enhancement, and Technology Alignment Committee, which is a subcommittee of the Faculty Council.

Mitigating circumstances

When serious circumstances ('Mitigating Circumstances'), beyond the control of a student or faculty member, adversely affect academic performance or teaching support, a Mitigating Circumstances report must be submitted using Woolf's red flagging system. Mitigating Circumstances may include but are not limited to serious medical problems, domestic and personal circumstances, major accidents or interruptions of public services, disturbances during examination, or serious administrative or procedural errors with a material effect on outcomes.

Mitigating circumstances do not normally include a member’s personal technology problems, including software, hardware, or personal internet connection failures; employment obligations or changes in employment obligations; permanent or sustained medical conditions (unless there is a sudden change of condition); or circumstances where no official evidence has been submitted.

Mitigating circumstances are normally only considered when a red flag has been submitted for the issue before the deadline of an affected written project or assignment, or within one week of a cumulative examination. Proof of mitigating circumstances may result in an extended deadline or examination period, or the possibility to retake an examination; it will not result in any regrading of existing submissions or exams.

Grade appeals

Students who dissent from the grades they have received should follow the normal procedure for submitting a red flag.