Master of Science

Technische Informatik

(MScTI)

Modulhandbuch

Universität Heidelberg
Fakultät für Ingenieurwissenschaften

Version 3.0[1]

[1]Dieses Modulhandbuch wird kontinuierlich weiterentwickelt, um die Lehre im Studiengang an aktuelle Entwicklungen und Anforderungen aus Wissenschaft, Gesellschaft und Industrie anzupassen. Insbesondere können in zukünftigen Versionen des Modulhandbuchs innerhalb des Vertiefungsbereichs Module in Übereinstimmung mit dem in 1.2 bis 1.5 beschriebenen Profil des Studiengangs bzw. dem in der Studien- und Prüfungsordnung beschriebenen Absolvent*innenprofil hinzugefügt, verändert oder entfernt werden. Entsprechende Änderungen sind durch minor oder micro release Versionsnummern gekennzeichnet. Sollte eine Änderung dieses Modulhandbuchs eine Änderung des Absolvent*innenprofils zur Folge haben, wird im Sinne einer wesentlichen Änderung des Studiengangsprofils gleichzeitig eine Änderung der Studien- und Prüfungsordnung notwendig. In diesem Fall wird eine neue major release Versionsnummer vergeben.

Allgemeine Informationen

Universität	Universität Heidelberg
Fakultät	Fakultät für Ingenieurwissenschaften
Abschluss	Master of Science Technische Informatik
Studiengangtyp	konsekutiver Masterstudiengang
Akronym	MScTI
Studienform	Vollzeit oder Teilzeit
Regelstudienzeit	2 Jahre / 4 Semester
Leistungspunkte	120
Studienstandort	Heidelberg
Stand	11.06.2025
zugehörige Version der SPO	XX.XX.XXXX

1 Qualifikationsziele, Profil und Besonderheiten des Studiengangs

1.1 Präambel – Qualifikationsziele der Universität Heidelberg

Anknüpfend an ihr Leitbild und ihre Grundordnung verfolgt die Universität Heidelberg in ihren Studiengängen fachliche, fachübergreifende und berufsfeldbezogene Ziele in der umfassenden akademischen Bildung und für eine spätere berufliche Tätigkeit ihrer Studierenden. Das daraus folgende Kompetenzprofil wird als für alle Disziplinen gültiges Qualifikationsprofil in den Modulhandbüchern aufgenommen und in den spezifischen Qualifikationszielen sowie den Curricula und Modulen der einzelnen Studiengänge umgesetzt:

- Entwicklung von fachlichen Kompetenzen mit ausgeprägter Forschungsorientierung;
- Entwicklung transdisziplinärer Dialogkompetenz;
- Aufbau von praxisorientierter Problemlösungskompetenz;
- Entwicklung von personalen und Sozialkompetenzen;
- Förderung der Bereitschaft zur Wahrnehmung gesellschaftlicher Verantwortung auf der Grundlage der erworbenen Kompetenzen.

1.2 Profil des Studiengangs

Das forschungsorientierte Masterprogramm in Technischer Informatik an der Universität Heidelberg wird vom Institut für Technische Informatik und der Fakultät für Ingenieurwissenschaften organisiert. Ziel der Ausbildung ist es, die Fachkenntnisse der Studierenden zu vertiefen und zu erweitern und sie auf eine forschungs- oder entwicklungsorientierte berufliche Laufbahn im Bereich der Technischen Informatik oder auf die Teilnahme an Promotionsprogrammen vorzubereiten. Die Studierenden entwickeln ein fundiertes Verständnis für verschiedene Lösungsansätze und Methoden und sind in der Lage, deren Vor- und Nachteile abzuwägen, um die bestmögliche Lösung für ein gegebenes Problem zu erarbeiten. Sie erkennen, welche Ansätze ungeeignet oder suboptimal sind, und verfügen über die notwendigen Fähigkeiten, neue Wege und Methoden zu entwickeln. Der Studiengang legt besonderen Wert auf praktische Fertigkeiten. Die Studierenden lernen, mit modernen Werkzeugen aus der Praxis der Technischen Informatik zu arbeiten, und können diese Fähigkeiten anwenden, um effiziente und praxisorientierte Lösungen zu entwickeln.

Die Studierenden können vertiefende Lehrveranstaltungen aus zwei Schwerpunkten – *Emerging Computing* und *Chip Design* – wählen, welche die am Institut vertretenen Forschungsthemen widerspiegeln. Jeder Schwerpunkt besteht aus einer Reihe von Modulen auf fortgeschrittenem Niveau.

Der Masterstudiengang beinhaltet eine Forschungsphase, bestehend aus einem Seminar, einer Studienarbeit und der Masterarbeit, in der die Studierenden die Fähigkeit erwerben, eigenständig zu forschen und wissenschaftliche Ergebnisse zu dokumentieren und zu publizieren. Dabei vertiefen sie ihr Wissen über wissenschaftliche Methoden, Informationstechnik, Hardware und Software, interdisziplinäres Systemdenken, praktische Anwendungserfahrung sowie ihre Kommunikationskompetenz und Teamfähigkeit.

1.3 Fachliche Qualifikationsziele des Studiengangs

Absolventinnen und Absolventen sind in der Lage, zentrale Konzepte der Mikroelektronik und moderner Rechnersysteme zu identifizieren und deren technische Grundlagen gezielt zu erfassen. Sie können konkrete Problemstellungen analysieren, technische Lösungswege bewerten und

deren Wirkungsweise deuten, etwa im Hinblick auf Mikroprozessoren, Peripheriekomponenten oder rekonfigurierbare Architekturen wie FPGAs. Dabei konzentrieren sie sich auf zentrale Aspekte wie Energieeffizienz, Leistung und Skalierbarkeit. Im Laufe des Studiums lernen sie, selbst analoge und digitale Hardwarekomponenten zu entwerfen und diese gezielt zur Lösung technischer Aufgabenstellungen einzusetzen. Sie sind mit modernen Entwurfsmethoden und verschiedenen dazu verwendeten Software-Tools vertraut und können diese auf hohem Niveau bedienen und einsetzen. Sie sind fähig, komplexe Zusammenhänge zwischen Hardware und Software verständlich zu darstellen und technische Entscheidungen fachlich fundiert zu argumentieren. Ihr Wissen versetzt sie in die Lage, theoretisches Wissen auf neue Fragestellungen zu transferieren und innovative Lösungen praktisch zu implementieren, auch unter Einbezug aktueller Entwicklungen in der Computertechnik.

1.4 Überfachliche Qualifikationsziele des Studiengangs

Absolventinnen und Absolventen des MScTI beherrschen den selbstorganisierten und zielgerichteten Einsatz verschiedenster Werkzeuge für spezialisierte technische Anwendungen und können reflektiert entscheiden, welche Methoden sich zur Lösung spezifischer Problemstellungen eignen. Sie arbeiten strukturiert und sind in der Lage, komplexe fachliche Projekte eigenverantwortlich zu planen, zu steuern und erfolgreich umzusetzen. Im Rahmen ihres Studiums wenden sie wissenschaftliche Arbeits- und Präsentationstechniken sicher an und entwickeln dabei ihre Team- und Diskussionsfähigkeit in interdisziplinären und interkulturellen Kontexten weiter. Sie erwerben grundlegende Kenntnisse zu rechtlichen und finanziellen Rahmenbedingungen unternehmerischer Tätigkeit und setzen sich kritisch mit gesellschaftlichen Deutungsmustern im Spannungsfeld von Technologie, Markt und Verantwortung auseinander. Darüber hinaus sind sie in der Lage, Marketingstrategien zu beurteilen und anzuwenden.

1.5 Berufsfelder, die den Absolventinnen und Absolventen offenstehen

Der wachsende Bedarf an fortschrittlicher Informationstechnologie erfordert ein tiefgehendes Verständnis der zugrunde liegenden Rechenhardware – sei es in Rechenzentren, Systemen der Automobilindustrie, mobilen und Edge-Geräten oder in spezialisierten Höchstleistungsrechnern. Die Optimierung von Leistung, Energieeffizienz und Kosten ist entscheidend, um für jede Anwendung die passende Hardware auszuwählen. Absolventinnen und Absolventen mit Fachkenntnissen in Mikroelektronik und Rechnersystemen sind bestens darauf vorbereitet, diesen Herausforderungen zu begegnen, und stellen daher eine wertvolle Ressource für eine Vielzahl von Branchen dar.

Mögliche Berufsfelder umfassen unter anderem die praxisnahe Entwicklung von Hardwaresystemen zur Datenerfassung und schnellen Datenverarbeitung, die effiziente Lösung rechenintensiver Aufgaben auf moderner, leistungsfähiger und heterogener Hardware, sowie den Entwurf analoger oder digitaler mikroelektronischer Schaltungen.

Absolventinnen und Absolventen können Karrieren in der Halbleiterentwicklung, im Design eingebetteter Systeme und der Hardwarebeschleunigung verfolgen – sowohl bei etablierten Technologieunternehmen als auch bei innovativen Start-ups. Zudem führt der zunehmende europäische Fokus auf Unabhängigkeit in der Mikroelektronik zu einer steigenden Nachfrage nach Fachkräften im Hardwaredesign und in der Hardwareentwicklung. Dadurch eröffnen sich den Absolventinnen und Absolventen vielfältige Berufsperspektiven – sowohl bei aufstrebenden Unternehmen als auch bei großen Industriekonzernen, die an der Entwicklung der nächsten Generation von Rechentechnologien arbeiten.

1.6 Besonderheiten des Studiengangs

1.6.1 Begründung für kumulative Prüfungen

Da die zu erwerbenden Kompetenzen in manchen Modulen sehr heterogen und differenziert sind, empfiehlt es sich, diese in spezifischen Einzelprüfungen und nicht in Modulabschlussprüfungen zu prüfen. Daher sind in manchen Modulen verschiedene Prüfungsformate (z. B. Klausur und erfolgreiche Teilnahme an den Übungen; Projektdokumentation und mündliche Prüfung; etc.) vorgesehen, um verschiedene Kompetenzen abzuprüfen.

1.6.2 Begründung für Module mit weniger als 5 Leistungspunkten

Das etablierte Modul Tools (Seite 16) bietet den Studierenden im Bereich *Übergreifende Kompetenzen* einen schnellen Überblick und eine Einführung in nützliche Software und Methoden. Das Modul basiert auf praktischer, betreuter Teilnahme vor Ort und beinhaltet keine nennenswerten Hausaufgaben, sodass lediglich 4 LP gerechtfertigt sind. Die geringere Anzahl an Soft-Skill-Punkten macht es zudem besser kompatibel mit anderen Bachelor- oder Masterstudiengängen.

2 Musterstudienpläne/Musterstudienverläufe

Die folgende Grafik zeigt einen beispielhaften Studienverlauf bei Studienbeginn im Wintersemester. Präzisere Vorschläge für individuelle Studienverlaufspläne je nach Studienbeginn und beabsichtigtem Studienschwerpunkt werden regelmäßig aktualisiert und finden sich auf den Webseiten des Studiengangs: www.ingwiss.uni-heidelberg.de/de/studium/technische-informatik-master/

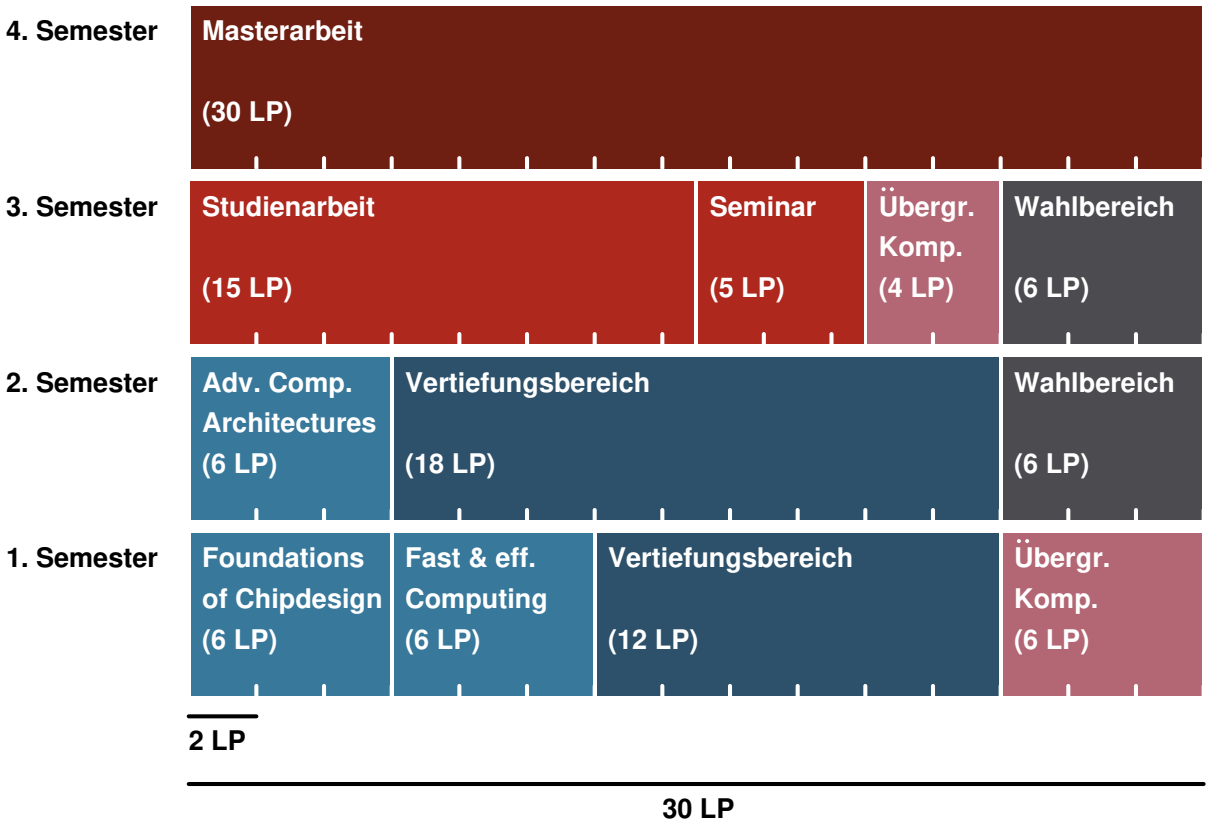

Abbildung 1: Beispielhafter Studienverlauf

Mobilitätsfenster

Die besten Zeitpunkte für ein Auslandssemester sind je nach Studienbeginn im Sommer oder Wintersemester das zweite oder dritte Fachsemester. Die 12 LP des Wahlbereichs sowie 6 LP im Bereich Schlüsselqualifikationen (insbesondere Sprachkurs) können problemlos im Ausland erbracht werden, da ihre Anerkennung in der Regel ohne Einschränkungen möglich ist. Finden die Studierenden im Ausland typischerweise zwei Module zu je 6 LP, die thematisch zur Vertiefung passen können die vollen 30 LP des jeweiligen Semesters im Ausland erworben werden.

Aktuelle Informationen zu Auslandsaufenthalten und dem zuständigen Ansprechpartner finden sich auf den Webseiten des Studiengangs: www.ingwiss.uni-heidelberg.de/de/studium/technische-informatik-master/auslandsaufenthalt

Zur Anerkennung von Leistungen siehe auch: www.ingwiss.uni-heidelberg.de/de/studium

3 Erläuterungen zur Studienstruktur

Die Module des Studiengangs fallen in die Kategorien

- Pflichtbereich,
- Vertiefungsbereich,
- Wahlbereich,
- Übergreifende Kompetenzen,
- Studienarbeit und
- Masterarbeit.

Diese Kategorien werden in den folgenden Unterabschnitten beschrieben:

Pflichtbereich

Die Module im Pflichtbereich sollen dazu beitragen, allen Studierenden ein gemeinsames Grundlagenwissen für die folgenden Vertiefungsvorlesungen zu vermitteln, unabhängig davon, in welchen unterschiedlichen Fachrichtungen sie ihr Bachelorstudium absolviert haben. Die folgenden Module sind verpflichtend zu absolvieren:

- Advanced Computer Architecture (6 LP)
- Foundations of Chip Design (6 LP)
- Methods for fast and efficient Computing (6 LP)
- Seminar: Aktuelle Themen der Technischen Informatik (5 LP)

Da alle Studierenden daran teilnehmen müssen, tragen diese Module auch dazu bei, sich gegenseitig kennenzulernen.

Vertiefungsbereich

Im Vertiefungsbereich[2] sind 30 LP zu erbringen. Die Module des Vertiefungsbereichs behandeln auf fortgeschrittenem Niveau Themen aus den zentralen Forschungsrichtungen des ZITI

- *Chip Design* und
- *Emerging Computing*.

Die verfügbaren Module und ihre Zuordnung zu einem oder mehreren der obigen Schwerpunktsbereiche sind in der Tabelle auf Seite 10 aufgeführt. Es müssen insgesamt 5 Module abgeschlossen werden. Das Institut stellt sicher, dass im Vertiefungsbereich in jedem Semester genug Module zur Verfügung stehen, um die vorgesehene Menge an Leistungspunkten zu erwerben.

Die Studierenden können frei aus allen Modulen des Vertiefungsbereichs wählen. Wir empfehlen jedoch nachdrücklich, sich auf einen Schwerpunktbereich zu konzentrieren, da das dann breite Fachwissen die Anfertigung der Masterarbeit erleichtert.

Wenn Studierende sich in einem der Schwerpunktbereiche spezialisieren, wird in den Studienabschlussdokumenten eine entsprechende Spezialisierung vermerkt, sofern 5 Module aus dem gewählten Schwerpunktbereich abgeschlossen wurden.

Beispielhafte Studienpläne zu den verschiedenen Schwerpunktbereichen sind auf der Webseite des Studiengangs zu finden.

[2] In zukünftigen Versionen des Modulhandbuchs können innerhalb des Vertiefungsbereichs Module hinzugefügt, verändert oder entfernt werden. Vergleiche Anmerkung zur Versionsnummer auf Seite 1.

Wahlbereich

Das Ziel des Wahlbereichs ist es, Fachwissen und Kenntnisse zu erweitern oder einen Einblick in andere Forschungsrichtungen zu erhalten, wie es z.B. bei interdisziplinären Forschungsthemen sinnvoll ist. Die Hauptintention besteht darin, während des Studiums auch über den Tellerrand der Informatik und Ingenieurwissenschaften hinauszublicken, wobei jedoch auch Module aus diesen Bereichen gewählt werden dürfen. Die Studierenden können vom umfangreichen Lehrangebot der Universität Heidelberg profitieren und sich beispielsweise Grundlagen- oder Anwendungswissen passend zu ihrem Forschungsschwerpunkt aneignen. Die Lehrveranstaltungen können frei aus dem Vorlesungsverzeichnis der Universität Heidelberg gewählt werden, vorausgesetzt, die anbietende Einrichtung stimmt zu. Module auf Bachelor-Niveau sind zulässig, sollten jedoch keine bereits vorhandenen Kenntnisse wiederholen. Die im Wahlbereich gewählten Module müssen folgende Bedingungen erfüllen:

- sie sind benotet,
- die Summe der Leistungspunkte beträgt mindestens 12 LP,
- sie erweitern das bestehende Fachwissen, d.h. sie wiederholen keine bereits absolvierten Module, beispielsweise aus einem vorhergehenden Bachelorstudium.

Im Zweifelsfall entscheidet der Prüfungsausschuss über die Genehmigung. Der Antrag ist vor dem Besuch des Moduls zu stellen.

Es ist auch möglich, Module aus dem MScTI im Wahlbereich zu wählen. Dies kann sinnvoll sein, wenn in den ersten Semestern noch keine eindeutige Entscheidung für einen bestimmten Schwerpunkt getroffen wurde. In diesem Fall können zunächst Module aus verschiedenen Schwerpunkten belegt werden. Sobald ein Schwerpunkt gewählt wurde, können die Module, die nicht diesem Schwerpunkt zugeordnet sind, dem Wahlbereich zugewiesen werden.

Übergreifende Kompetenzen (Soft Skills)

Im Bereich der Schlüsselkompetenzen müssen 10 LP erworben werden. Es können folgende Veranstaltungen gewählt werden:

- Module aus dem Studiengang, die in der untenstehenden Tabelle als Soft Skill gekennzeichnet sind,
- Module, die von der Fakultät für Ingenieurwissenschaften als Soft-Skill-Kurse in HeiCo ausgewiesen sind,
- Zwei Einführungsmodule aus dem Entrepreneurship-Zertifikat der Universität Heidelberg (6 LP),
- Veranstaltungen aus dem HeiSkills-Programm, die als Soft-Skill-Kurse klassifiziert sind,
- Sprachkurse bis zu einem Maximum von 6 LP.

Weitere Veranstaltungen können auf Antrag vom Prüfungsausschuss anerkannt werden.

Studienarbeit

Dieses Modul führt die Studierenden in die Arbeit einer gewählten Forschungsgruppe ein. Das Thema befindet sich in der Regel auf einem einführenden Niveau, sodass das erforderliche Hintergrundwissen sowie die zur Bearbeitung notwendigen Werkzeuge im Rahmen des Moduls erlernt werden können. Das Thema kann – muss aber nicht – eine Einführung in das spätere Thema der Masterarbeit darstellen. Die Studienarbeit wird durch einen Bericht abgeschlossen. Da dem Modul 15 LP zugeordnet sind, entspricht der Arbeitsaufwand in etwa 50% der Gesamtbelastung des dritten Semesters, also einem erheblichen Umfang.

Masterarbeit

Die Masterarbeit erstreckt sich über einen Zeitraum von 6 Monaten und wird in der Regel im gewählten Schwerpunkt (sofern vorhanden) im vierten Semester durchgeführt. Nach Abgabe der schriftlichen Arbeit werden die Ergebnisse in einem abschließenden öffentlichen Kolloquium präsentiert, das in die Bewertung durch die Gutachter*innen mit einfließt.

Modulübersicht

Die folgende Tabelle zeigt alle Module des Studiengangs und deren Zuordnung zu den Kategorien Pflichtbereich, Übergreifende Kompetenzen oder zu einem der zwei Schwerpunktbereiche des Vertiefungsbereichs.

Modul	Pflichtbereich	ÜK	Chip Design	Emerg. Comp.	Verantwortlich	Seite
Advanced Computer Architecture	○				NT	12
Foundations of Chipdesign	○				PF	13
Fast and Efficient Computing	○				HF	14
Seminar: Aktuelle Themen der TI	○				AS	15
Tools		×			PF/AS	16
Components, Circuits & Simulation			×		PF	17
Digital Hardware Description and Verification			×		DK	18
Full Custom VLSI Design			×		PF	19
Advanced Analogue Building Blocks			×		PF	20
Digital Semicustom Design Flow			×		AG	21
Reconfigurable Embedded Systems			×	×	DK	22
Current topics in Chip Design			×		Wechselnd	23
GPU Computing (Architecture + Progr.)				×	HF	25
Performance Essentials for CPUs and GPUs				×	RS	24
Embedded Machine Learning				×	HF	26
Scalable and Robust Embedded Machine Learning				×	HF	27
CPU Algorithm Design				×	RS	28
GPU Algorithm Design				×	RS	29
Consistency and Coherency				×	HF	30
High Performance & Distributed Computing				×	HF	31
Emerging Computing Paradigms			×	×	NT	32
Architecture and CAD for FPGA			×	×	DK	33
Energy Efficient Computing			×	×	DK	34
Memory-Centric Computing			×	×	NT	35
Biosignal Processing				×	NT	37
Brain-Inspired Computing			×	×	JS	36
Current topics in Emerging Computing				×	Wechselnd	38
Studienarbeit	○				Alle	39
Masterarbeit	○				Alle	40

Tabelle 1: Module des Studiengangs und deren Zuordnung zu den verschiedenen Kategorien.

4 Modulbeschreibungen

Auf den folgenden Seiten werden alle Module beschrieben, die vorrangig im Studiengang "Technische Informatik" angeboten werden.

Im Masterstudiengang Technische Informatik werden in den verschiedenen Lehrveranstaltungsarten vorwiegend folgende Lehr- und Lernformen verwendet:

- *Vorlesung:* Vortrag der Lehrenden, Vor- und Nachbereitung durch Selbststudium
- *Übung:* Selbststudium, Bearbeiten von Übungsblättern, Diskussionen der erarbeiteten Lösungen in der Gruppe, Nachbereitung von Vorlesungsstoff gemeinsam mit dem/der Übungsleiter*in
- *Praktische Übung:* praktische Programmier- oder Hardwareübung, in der Regel unter Anleitung von Übungsleiter*innen
- *Praktikum:* Eigenständige Durchführung eines Hard- oder Softwareprojekts, Verfassen von technischen Reports, Begleitung durch Praktikumsbetreuer*innen
- *Seminar:* Selbststudium/Literaturrecherche, Vorträge der Studierenden, aktive Fragen und Diskussionen, Feedback

Die Prüfungsmodalitäten der einzelnen Prüfungen werden zu Beginn der Lehrveranstaltungen bekannt gegeben.

Grundsätzlich gilt für alle Module des Studiengangs, dass die Dauer des Moduls ein Semester beträgt. Dies ist daher nicht in jeder einzelnen Modulbeschreibung ausgewiesen.

Ebenfalls für alle Module gilt, dass sich das empfohlene Semester nach den individuellen Studienverlaufsplänen der Studierenden richtet. Ein beispielhafter Verlauf ist in Bild 1 im Abschnitt 2 aufgezeigt. Weitere Studienverlaufspläne für Studienbeginn im Sommer und Wintersemester sowie mit verschiedenen Schwerpunkten, aus denen sich das empfohlene Semester für die individuellen Module ergibt, finden sich auf der Webseite. Einheitlich empfohlen sind nur die Module *Studienarbeit* im dritten und *Masterarbeit* im vierten Semester.

Keines der Module ist in anderen Bachelor- oder Masterstudiengängen verpflichtend. Eine optionale Einbindung der Module in andere Studiengänge obliegt diesen selbst. Die Module stehen Studierenden außerhalb des Studiengangs offen, sofern ausreichend Raumkapazitäten und Infrastruktur (Computerarbeitsplätze, Laborplätze, Rechenressourcen usw.) zur Verfügung stehen. Studierende des Studiengangs "Technische Informatik" werden bevorzugt aufgenommen. Freie Plätze können von Studierenden anderer Studiengänge belegt werden. Die Studienarbeit und die Masterarbeit sind ausschließlich für Studierende des Studiengangs "Technische Informatik" vorgesehen.

Die Module *Current Topics in...* dienen als flexible Module, um nicht-regelmäßige oder spontane Lehrveranstaltungen (z.B. einmalige Angebote durch Gastprofessor*innen) einzubinden. Die Liste möglicher Veranstaltungen, die in diesem Modul belegt werden können, ist auf der ZITI-Webseite veröffentlicht.

Vorübergehende Abweichungen von der standardmäßigen Semesterzuordnung oder von der vorgesehenen Lehrperson können aus organisatorischen Gründen erforderlich sein. Die Studierenden werden in einem solchen Fall ausreichend vorab informiert.

Code: MScTI_ACA	**Form:** P	**Modulname:** Advanced Computer Architecture	
Modulverantwortlicher: Prof. Dr. Nima TaheriNejad		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung (3 h / Woche)
- Praktische Übung mit Hausarbeiten (1 h / Woche)

Lernziele: Die Studierenden...
- können Vorteile und Herausforderungen des (parallelen) Rechnens aufzählen und diskutieren,
- können mindestens eine Methode zur Leistungssteuerung von Mehrprozessorsystemen beschreiben,
- können mindestens zwei Datenübermittlungstechnologien nennen und eine Methode zur Auswahl der geeigneten Lösung für ein System erläutern,
- sind in der Lage, verschiedene Speichersysteme und deren Leistungskennzahlen zu analysieren und zu diskutieren,

Lerninhalte:
- Prozessorarchitekturen,
- Paralleles Rechnen,
- Programmiermodelle und Architekturen,
- Mehrprozessorarchitekturen,
- Verbindungsnetze (inklusive Network-on-Chip),
- Caching in Mehrprozessorsystemen,
- Multithreading,
- Heterogene Mehrprozessorsysteme.
- Speicher und Speichertechnologien
- Einführung in neuartige Rechenparadigmen (Approximate und In-Memory Computing),

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** Grundlagen der Rechnerarchitektur

Literatur:
- Wird von den Dozenten bekannt gegeben.

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_FCD	**Form:** P	**Modulname:** Foundations of Chip Design	
Modulverantwortlicher: Prof. Dr. Peter Fischer		**Typ:** Vorlesung, praktische Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** WiSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- verstehen den groben Ablauf der Chip-Herstellung,
- kennen den Design-Flow für analoge und digitale Schaltungen,
- besitzen eine grundlegende Vorstellung von Widerständen, Kondensatoren und MOS-Transistoren,
- können einfache analoge Simulationen durchführen und Netzlisten mit einem stark vereinfachten MOS-Modell verwenden,
- verstehen den Zusammenhang zwischen Schaltung und Layout im analogen und digitalen Bereich,
- kennen elementare digitale Gatter und Flipflops,
- können einfache digitale Schaltungen beschreiben und digital simulieren,
- führen Messungen an einfachen Strukturen durch, die mit dem Mikroskop sichtbar sind, wie Kennlinien, Ringoszillator, Pseudozufallssequenz, Differenzverstärker und RC-Oszillator,
- erhalten einen Einblick in die kommerzielle Seite des Chip-Designs,
- verstehen die Marktstruktur und wirtschaftlichen Rahmenbedingungen der Halbleiterindustrie.

Lerninhalte:
- Überblick über den Herstellungsprozess von Halbleiterchips,
- Vorstellung des Entwurfsprozesses für analoge und digitale Schaltungen,
- Einführung in die Grundlagen von Widerständen, Kondensatoren und MOS-Transistoren,
- Durchführung einfacher analoger Simulationen anhand von Netzlisten mit vereinfachten MOS-Modellen,
- Erläuterung des Zusammenhangs zwischen Schaltung und Layout im Bereich der analogen Elektronik,
- Grundlegende digitale Bausteine wie Gatter und Flipflops,
- Erklärung des Verhältnisses zwischen Schaltkreisentwurf und Layout bei digitalen Schaltungen,
- Einführung in die Beschreibung und Simulation einfacher digitaler Schaltungen,
- Praktische Messungen an einfachen, makroskopisch sichtbaren Strukturen, darunter Kennlinien, Ringoszillatoren, Pseudozufallssequenzen, Differenzverstärker und RC-Oszillatoren,
- Einblick in wirtschaftliche Aspekte des Chip-Designs
- Vorstellung der Marktmechanismen und wirtschaftlichen Rahmenbedingungen der Halbleiterindustrie.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** keine

Literatur:
- Literatur wird in der Veranstaltung zur Verfügung gestellt

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_FEC	**Form:** P	**Modulname:** Fast and Efficient Computing	
Modulverantwortlicher: Prof. Dr. Holger Fröning		**Typ:** siehe Modulberschreibungen der wählbaren Module	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** WiSe oder SoSe

Modulteile:
- siehe Modulberschreibungen der wählbaren Module

Lernziele: Die Studierenden...
- verstehen und können die Kernaspekte schneller und effizienter Berechnungen für bestimmte Hardwaretypen anhand eines konkret ausgewählten Beispiels anwenden. Weitere Ziele sind in der entsprechenden Modulbeschreibung aufgeführt.

Lerninhalte:
- kann entweder durch den Abschluss des Moduls MScTI PERF (Performance Essentials for CPUs and GPUs) oder des Moduls MScTI CEGPU (GPU Computing) belegt werden. Die Leistungspunkte des hier gewählte Modul können nicht zusätzlich im Wahlpflichtbereich angerechnet werden. Das gewählte Modul zählt aber als absolviert für den Schwerpunktbereich Ëmerging Computing".

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** keine

Literatur:
- siehe Modulberschreibungen der wählbaren Module

Voraussetzung für die Vergabe von Leistungspunkten:
siehe Modulberschreibungen der wählbaren Module

Code: MScTI_SEMINAR	**Form:** P	**Modulname:** Äktuelle Themen der Technischen Informatik	
Modulverantwortlicher: All Groups		**Typ:** Seminar	
ECTS-Punkte: 5	**Workload:** 150 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe und WiSe

Modulteile:
- Recherche eines Themas der Technischen Informatik und Präsentation im Rahmen eines Blockseminars

Lernziele: Die Studierenden...
- können Literatur zu einem spezifischen Thema recherchieren,
- können Material für eine Präsentation auswählen und strukturieren,
- können Präsentationsfolien erstellen,
- können eine wissenschaftliche Präsentation halten.

Lerninhalte:
- Literaturrecherche,
- Vorbereitung der Präsentation,
- Mündliche Präsentation (ca. 45 Minuten),
- Aktive Teilnahme an den Präsentationen und Diskussionen der Kommiliton:innen.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** Grundkenntnisse im gewählten Fachgebiet

Literatur:
- Vom Betreuer bereitgestellt und durch eigene Literaturrecherche ergänzt

Voraussetzung für die Vergabe von Leistungspunkten:
Präsentation, aktive Teilnahme

Code: MScTI_TOOLS	**Form:** ÜK	**Modulname:** Tools	
Modulverantwortlicher: Prof. Dr. Peter Fischer		**Typ:** Vorlesung, Präsenzübung	
ECTS-Punkte: 4	**Workload:** 120 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung 2 h / Woche)
- Präsenzübung (2 h / Woche)

Lernziele: Die Studierenden...
- haben einen Überblick über die Funktionalitäten verschiedener Softwaretools, die häufige Aufgaben wie das Erstellen von Zeichnungen und Illustrationen, Programmierung, Lösung mathematischer Probleme, Datenanalyse und -visualisierung, Literaturrecherche oder Teamarbeit unterstützen,
- sind in der Lage, ihre Arbeitsabläufe durch die Auswahl geeigneter Werkzeuge zu verbessern,
- sind sich bewusst, dass der Einsatz passender Tools die Arbeitsqualität und Effizienz steigert,
- können ihr Wissen und ihre Fähigkeiten in den vorgestellten Tools auf Basis der gegebenen Einführung vertiefen,
- sind mit den Konzepten guter wissenschaftlicher Praxis vertraut und können diese auf ihre eigene wissenschaftliche Arbeit anwenden,
- wissen, wie sie wissenschaftliche Literatur in professionellen Bibliothekssystemen finden.

Lerninhalte:

Die Themen werden aus einem Pool von Möglichkeiten entsprechend dem Interesse der Studierenden und der Verfügbarkeit fachkundiger Dozierender ausgewählt. Die Liste der Themen wird regelmäßig an neue Entwicklungen angepasst. Beispielthemen sind:
- Einführung in Linux,
- Versionskontrollsysteme (git, svn usw.),
- Einführung in Python,
- Mathematische Software (Mathematica),
- Datenauswertung und Diagrammerstellung (gnuplot, root),
- 2D- und 3D-Zeichnung, Konstruktion und Visualisierung (PovRay, OpenSCAD, PostScript, PDF),
- Stile und Vorlagen (PowerPoint, Word),
- Einführung in LaTeX,
- Teamarbeit,
- Projektplanung und -management,
- Gute wissenschaftliche Praxis,
- Literaturrecherche, bibliometrische Maßnahmen, Open-Access-Konzepte usw.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** keine

Literatur:
- Wird von den Dozenten bekannt gegeben.

Voraussetzung für die Vergabe von Leistungspunkten:
Regelmäßige Teilnahme

Code: MScTI_CCS	**Form:** WP	**Modulname:** Components, Circuits and Simulation	
Modulverantwortlicher: Prof. Dr. Peter Fischer		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- können einfache analoge Schaltungen durch Kombination elementarer Bausteine entwerfen,
- können die Eigenschaften (Verstärkung, Frequenzverhalten) einfacher Schaltungen vorhersagen und analytische Näherungsformeln für Verstärkung, Bandbreite, Ausgangswiderstand usw. angeben,
- können Analogsimulatoren nutzen, um Schaltungen im Zeit- und Frequenzbereich zu analysieren,
- wissen, was ein Arbeitspunkt ist, wie dieser das Schaltungsverhalten beeinflusst und wie er eingestellt werden kann,
- können die Geometrie und den Arbeitspunkt von Transistoren mit deren Klein- und Großsignal-Eigenschaften in Beziehung setzen.

Lerninhalte:
- Betriebsprinzip von Diode und Transistor,
- Modellierung von Diode und MOSFET, Großsignal- und Kleinsignalmodele,
- Spannungs- und Stromquellen, Thévenin-Äquivalent,
- Beschreibung von Bauelementen und Schaltungen mit komplexen Variablen,
- Bode-Diagramm, Übertragungsfunktion,
- Analoge Simulation (DC, AC, transient),
- Grundschaltungen: Stromspiegel, Verstärkerstufe, Kaskode, Source Follower, Differenzverstärker,
- Praktische Übungen mit professionellen Simulationswerkzeugen.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** Physikalische Grundkenntnisse

Literatur:
- P. R. Gray, P. J. Hurst, S. H. Lewis, R. G. Meyer: *Analysis and Design of Analog Integrated Circuits* (Wiley & Sons, New York, 1993)
- D. A. Johns, K. Martin: *Analog Integrated Circuit Design* (Wiley & Sons, 1997)

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_HDL	**Form:** WP	**Modulname:** Digital Hardware Description and Verification	
Modulverantwortlicher: Prof. Dr. Dirk Koch		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** WiSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- können digitale Schaltungen mithilfe von Hardwarebeschreibungssprachen beschreiben,
- kennen typische Sprachen wie Verilog, SystemVerilog und VHDL,
- können ein großes Design sinnvoll in kleinere Entwurfsblöcke strukturieren,
- haben Erfahrung im Entwurf von Zustandsautomaten,
- können Simulationsumgebungen (Testbenches) zur Verifikation einrichten,
- kennen fortgeschrittene Verifikationsmethoden,
- erkennen den Unterschied zwischen synthesefähigen und nicht-synthesefähigen Konstrukten.

Lerninhalte:
- Syntax und Semantik von Verilog, VHDL und SystemVerilog,
- Module und deren Instanziierung, einfache und fortgeschrittene Schnittstellen (Ports),
- Strukturierte Beschreibung und Verhaltensbeschreibung von Schaltungen,
- Beschreibung typischer Funktionsblöcke (RAM, FIFO, Registerfile, Zähler usw.),
- Robuste Beschreibung von Zustandsautomaten,
- Instanziierung von Hard Macros (in FPGAs oder VLSI),
- Simulation, Testmuster, Testbenches usw.,
- Einführung in UVM (Universal Verification Methodology).

Voraussetzungen:	**Empfohlene Vorkenntnisse:**
keine	keine

Literatur:
- Literatur wird in der Veranstaltung zur Verfügung gestellt

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_FCVD	**Form:** WP	**Modulname:** Full Custom VLSI Design	
Modulverantwortlicher: Prof. Dr. Peter Fischer		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** WiSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- können den gesamten Entwurfsprozess von der Schaltungsidee bis zum finalen, geprüften Layout durchführen,
- verstehen den Zusammenhang zwischen Designregeln und Halbleitereigenschaften bzw. Fertigungsproblemen,
- sind in der Lage, eine Mixed-Mode-Simulation praktisch durchzuführen,
- können parasitäre Werte extrahieren und eine Simulation unter Berücksichtigung dieser Parasiten durchführen,
- können einfache automatisierte Skripte mit SKILL programmieren.

Lerninhalte:
- Halbleiterfertigung,
- Technologie- und Designregeln, Technologiedateien,
- Layout von Bauelementen, Regeln, Matching,
- Design Rule Check (DRC),
- Extraktion, Layout-gegen-Schematic-Check,
- ESD- und Antennenregeln, Latch-up,
- Testgeräte und Testverfahren,
- Skriptprogrammierung mit SKILL,
- Parasitärextraktion und Simulation,
- Mixed-Mode-Simulation.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** Grundkenntnisse im Design analoger Schaltungen

Literatur:
- Vorlesungsskript (verfügbar online)

Voraussetzung für die Vergabe von Leistungspunkten:
Entwurf (Schaltplanerfassung, Simulation und Layout) einer Schaltung mit einer kurzen Präsentation.

Code: MScTI_AABB	**Form:** WP	**Modulname:** Advanced Analogue Building Blocks	
Modulverantwortlicher: Prof. Dr. Peter Fischer		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** WiSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- kennen eine Vielzahl fortgeschrittener Schaltungstopologien,
- erwerben ein vertieftes qualitatives und quantitatives Verständnis des Verhaltens analoger Schaltungen,
- können Leistungskennwerte definieren, diese aus Simulationen gewinnen und Schaltungsparameter optimieren,
- sind in der Lage, eine geeignete Schaltung für ein gegebenes Problem auszuwählen.

Lerninhalte:
- Schaltungsfamilien und Themen, die detailliert behandelt werden, werden ausgewählt aus:
- Rauschen von Bauelementen und Schaltungen,
- Ladungsverstärker mit Rückkopplung und Filter,
- Logikfamilien (NMOS, Dynamic, Pass Gate, Differential),
- Phasenregelkreis (PLL),
- Kaskadierte Verstärker,
- Fortgeschrittene Stromspiegel,
- Differenzielle Schaltungen, Gleichtakt-Rückkopplung,
- Digital-Analog-Wandler (DAC) und Analog-Digital-Wandler (ADC),
- Schalter,
- Schaltkapazitätsschaltungen (Switched Capacitor Circuits).

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** Vertiefte Kenntnisse im Design analoger Schaltungen

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

Voraussetzung für die Vergabe von Leistungspunkten:
Mündliche Prüfung

Code: MScTI_DIGDF	**Form:** WP	**Modulname:** Digital Semicustom Design Flow	
Modulverantwortlicher: Prof. Dr. Dirk Koch, A. Grübl		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten und Chip-Projekt (2 h / Woche)

Lernziele: Die Studierenden...
- erlernen die Methodik für den semi-custom ASIC-Entwurf,
- können das statische Timing digitaler Schaltungen verstehen, constrainen und analysieren,
- können ihr erworbenes Wissen zum Entwurf sehr komplexer Chips anwenden,
- können den vollständigen Backend-Designprozess für moderne Chiptechnologien durchführen,
- kennen relevante Methoden zur Integration von Design-for-Test Strukturen und relevante Layout-Verifikationsmethoden.

Lerninhalte:
- Fortgeschrittene Methoden zum Entwurf anwendungsspezifischer integrierter Schaltungen (ASICs),
- Synthese komplexer Hardwaresysteme,
- Statische Timing-Analyse (STA) und Erstellung entsprechender Constraints,
- Floorplanning, Platzierung und Verdrahtung (Place & Route) von Modulen und Standardzellen,
- Clock Tree Synthese
- Analyse der Signalintegrität,
- Design Rule Checks (DRC),
- Design-for-Test.

  Die im Rahmen des EURPORACTICE-Projekts bereitgestellten Lizenzen ermöglichen es unseren Studierenden, mit modernsten EDA-Tools zu arbeiten und zu lernen, die normalerweise nur in der Chipindustrie verwendet werden. Wir verwenden in den Tutorien Software von Cadence Design Systems.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** Vertiefte Kenntnisse im Digitalen Hardwaredesign

Literatur:
- Literatur wird in der Veranstaltung zur Verfügung gestellt

Voraussetzung für die Vergabe von Leistungspunkten:
30-minütige mündliche Prüfung am Ende des Semesters. Mindestens 50% der Übungen sowie das Chip-Projekt müssen bestanden sein.

Code: MScTI_RES	**Form:** WP	**Modulname:** Reconfigurable Embedded Systems	
Modulverantwortlicher: Prof. Dr. Dirk Koch		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- verstehen die Elemente, Eigenschaften und Anforderungen eingebetteter Systeme,
- erhalten ein grundlegendes Verständnis rekonfigurierbarer Architekturen,
- lernen Entwurfsmethoden für Anwendungen auf Mikroprozessoren und FPGAs kennen,
- entwerfen eigene IP-Cores mithilfe struktureller Datenfluss- und zustandsautomatengestützter Kontrollfluss-Designtechniken,
- nutzen IP-Cores zur Erstellung hybrider Anwendungen für Prozessoren und rekonfigurierbare Coprozessoren mit geeigneten Schnittstellenmechanismen,
- implementieren und programmieren eine beispielhafte eingebettete FPGA-Plattform.

Lerninhalte:
- Anforderungen und spezielle Eigenschaften eingebetteter Systeme,
- Überblick über Hardwarekomponenten: Mikrocontroller, Peripherie, FPGAs,
- Echtzeitproblematiken und Aufgabenplanung (Scheduling),
- FPGA-Entwurfswerkzeuge: HDL (inkl. VHDL-Tutorial), Simulator, Debugger,
- Hochstufige FPGA-Entwurfsmethoden (inkl. High-Level Synthesis, HLS),
- Grundlagen der CAD-Werkzeuge: Wie wird Code in eine FPGA-Konfiguration übersetzt?,
- System-on-Chip-Architektur – Controller, Busse und Peripherie,
- Hard- und Software-Co-Design,
- Software für eingebettete Systeme (Stand-alone-Systeme und Echtzeitbetriebssysteme),
- Rekonfigurierbare, benutzerdefinierte Erweiterungen der Befehlssatzarchitektur (ISA).

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** keine

Literatur:
- Literatur wird in der Veranstaltung zur Verfügung gestellt

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_CD_Wahl	**Form:** WP	**Modulname:** Current topics in Chip Design	
Modulverantwortlicher: Verschieden		**Typ:** Vorlesung oder praktischer Kurs	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe oder WiSe

Modulteile:
- Wird von den Dozenten vor Beginn des Moduls bekannt gegeben

Lernziele: Die Studierenden...
- zeigen vertiefte Kenntnisse in einem spezifischen Bereich des Chipdesigns,
- analysieren fortgeschrittene Forschungsfragen in einem spezifischen Bereich des Chipdesigns,
- nutzen Konzepte aus Mathematik, Physik, Informatik und Ingenieurwissenschaften, um Lösungen für Forschungsfragen in einem spezifischen Bereich des Chipdesigns zu entwickeln.

Lerninhalte:
- Experimentelle, theoretische und/oder praktische Methoden in einem spezifischen Bereich des Chipdesigns.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** keine

Literatur:
- Wird von den Dozenten bekannt gegeben.

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_PERF	**Form:** WP	**Modulname:** Performance Essentials for CPUs and GPUs	
Modulverantwortlicher: Prof. Dr. Robert Strzodka		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- verstehen die Leistungsauswirkungen verschiedener Code-Konstrukte,
- entwerfen bessere Programme unter Beachtung effektiver Programmierstil-Richtlinien,
- wissen, wie sie CPUs und GPUs mit demselben Quellcode programmieren,
- können effiziente parallele Algorithmen aus vorhandenen Bibliotheken auswählen.

Lerninhalte:
- Wichtigste Dos and Donts für effizienten Code,
- einheitliche Hochsprachenprogrammierung von CPUs und GPUs,
- Effiziente Speicherverwaltung, Datenzugriff und Berechnung,
- klarer und effektiver Programmierstil,
- parallele Datenstrukturen und Algorithmen,
- Datenlayouts und deren Performanz
- parallele Bibliotheken.

Voraussetzungen: Kenntnisse in C++	**Empfohlene Vorkenntnisse:**

Literatur:
- Bjarne Stroustrup: *A Tour of C++* (3rd ed, Addison-Wesley, 2022)
- Zusätzliches Material wird über die Lernplattform bereitgestellt.

Voraussetzung für die Vergabe von Leistungspunkten:
Für die Teilnahme an der mündlichen oder schriftlichen Prüfung sind mindestens 50% der Punkte aus den Übungen erforderlich.

Code: MScTI_GPU	**Form:** WP	**Modulname:** GPU Computing	
Modulverantwortlicher: Prof. Dr. Holger Fröning		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** WiSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- kennen GPU-Architekturen und die damit verbundenen Designentscheidungen,
- kennen die Faktoren, die die Leistung von GPU-Programmen bestimmen, und sind in der Lage, GPUs zur Lösung von Rechenproblemen zu programmieren,
- können CUDA-Programme für rechen- oder speicherintensive Probleme entwerfen und optimieren,
- wissen, wie CUDA-Werkzeuge zur Unterstützung bei Programmierung, Fehlersuche und Leistungsoptimierung eingesetzt werden,
- sind in der Lage, rechen- oder speicherintensive Probleme mit GPUs zu lösen – mit dem Ziel hoher Leistungsfähigkeit in Bezug auf Zeit- und Energieeffizienz,
- sind in der Lage zu beurteilen, wann Beschleuniger wie GPUs für ein gegebenes Rechenproblem sinnvoll einsetzbar sind.

Lerninhalte:
- Grundlagen der GPU-Architektur und des zugehörigen Programmiermodells,
- Einführung in CUDA,
- Techniken zur Leistungsoptimierung,
- Konsistenz und Kohärenz bei GPUs,
- Alternativen zu CUDA und fortgeschrittene GPU-Konzepte.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** Grundlagen der Rechnerarchitektur, Prinzipien der Parallelen Programmierung, C, C++, OS Grundlagen

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

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_EML	**Form:** WP	**Modulname:** Embedded Machine Learning	
Modulverantwortlicher: Prof. Dr. Holger Fröning		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- kennen die Grundlagen des maschinellen Lernens (ML),
- sind mit neuronalen Netzwerkarchitekturen für Bild-, Signal- und Sprachverarbeitung vertraut,
- können solche Modellarchitekturen für einfache Problemstellungen entwerfen,
- verstehen die rechentechnischen Anforderungen dieser Architekturen,
- kennen verschiedene Prozessor- und Systemarchitekturen zur Ausführung von ML-Modellen,
- sind in der Lage, Lösungen zur Implementierung von ML-Modellen auf ressourcenbeschränkten Prozessoren zu entwickeln.

Lerninhalte:
- Grundlagen des maschinellen Lernens (ML),
- Entwurf neuronaler Architekturen, einschließlich Grundlagen neuronaler Netze, automatischer Differentiation und Optimierung, Regularisierung und Generalisierung,
- Anwendungen, einschließlich Zeitreihenanalyse und Computer Vision,
- Sichere Optimierungen im Hinblick auf Software und Hardware,
- Unsichere Optimierungen im Hinblick auf Software und Hardware,
- Zukünftige Entwicklungen im Bereich Embedded ML.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** Rechnerarchitektur (z.B. “GPU Computing”, “High Performance & Distributed Computing”, oder ”Advanced Computer Architecture”), Python

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

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_SREML	**Form:** WP	**Modulname:** Scalable and Robust Embedded Machine Learning	
Modulverantwortlicher: Prof. Dr. Holger Fröning		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- kennen fortgeschrittene Methoden des maschinellen Lernens (ML) sowie neuartige Hardwaretechnologien,
- sind mit fortgeschrittenen neuronalen Architekturen für verschiedene Aufgaben vertraut,
- können solche neuronalen Architekturen für komplexe Problemstellungen entwerfen,
- verstehen die rechentechnischen Anforderungen dieser Architekturen,
- kennen konventionelle und unkonventionelle Prozessor- und Systemarchitekturen zur Ausführung fortgeschrittener neuronaler Netze,
- sind in der Lage, Lösungen zur Implementierung fortgeschrittener neuronaler Architekturen auf ressourcenbeschränkten Prozessoren zu entwickeln.

Lerninhalte:
- Einführung in die Schnittstelle zwischen neuartigen Hardwaretechnologien und fortgeschrittenen neuronalen Architekturen,
- Skalierbare Methoden des maschinellen Lernens,
- Umgang mit Unsicherheiten im Lernprozess,
- Sichere und unsichere Optimierungen für fortgeschrittene neuronale Architekturen sowie konventionelle und unkonventionelle Prozessor- und Systemarchitekturen,
- Abwägung von Kosten und Qualität (Cost-Quality Trade-offs),
- Zukünftige Entwicklungen, einschließlich probabilistischer Prozessoren und Skalierungsregeln.

Voraussetzungen:	**Empfohlene Vorkenntnisse:**
Grundlagen tiefer neuronaler Netzwerke, des maschinellen Lernens, sowie der Modellkomprimierung (Quantisierung, Pruning) und Mapping auf Hardware	Rechnerarchitektur (z.B. “GPU Computing”, “High Performance & Distributed Computing”, oder ”Advanced Computer Architecture”), Python

Literatur:
- Richard McElreath: *Statistical Rethinking* (2020, Chapman & Hall)
- Kevin Patrick Murphy: *Probabilistic Machine Learning – An Introduction* (MIT Press, 2022)
- Kevin Patrick Murphy: *Probabilistic Machine Learning: Advanced Topics* (MIT Press, 2023)

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_CPUALG	**Form:** WP	**Modulname:** CPU Algorithm Design	
Modulverantwortlicher: Prof. Dr. Robert Strzodka		**Typ:** Vorlesung, Übung, Projektarbeit	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung 2 h / Woche
- Übung 2 h / Woche mit Hausarbeiten in der ersten Semesterhälfte
- Projekt 2 h / Woche mit Hausarbeiten in der ersten Semesterhälfte

Lernziele: Die Studierenden...
- sind in der Lage, das gesamte Parallelismus- und Bandbreitenpotenzial großer CPUs auszunutzen,
- können CPU-Designentscheidungen bezüglich Tradeoffs in parallelen Algorithmen treffen,
- wissen, wie fortgeschrittene CPU-Transformationen zur Erhöhung von Parallelismus und Datenlokalität angewendet werden,
- verstehen Modelle zur parallelen Leistung und Skalierbarkeit.

Lerninhalte:
- Modelle zur parallelen Leistung und Skalierbarkeit,
- mehrere Ebenen des Parallelismus,
- parallele Entwurfsmuster,
- paralleler Datenzugriff,
- Kommunikation vs. Berechnung,
- Latenz vs. Durchsatz,
- Arbeitseffizienz vs. Schritteffizienz,
- Lokalität vs. Parallelismus,
- Werkzeuge für paralleles Programmieren.

Voraussetzungen:	**Empfohlene Vorkenntnisse:**
Kenntnisse in C++	C++23 und STL (z.B. aus MScTI_PERF)

Literatur:
- Michael McCool, Arch Robison, James Reinders: Structured Parallel Programming, Morgan Kaufmann, 2012
- Michael Voss, Rafael Asenjo, James Reinders: ProTBB, Springer Nature, 2019
- Zusätzliches Material wird über die Lernplattform bereitgestellt.

Voraussetzung für die Vergabe von Leistungspunkten:
Für die Teilnahme an der Projektprüfung, die aus einem Software-Projekt mit Dokumentation sowie einer Präsentation der Ergebnisse besteht, sind mindestens 50% der Punkte aus den Übungen erforderlich. Alternativ zur Projektprüfung kann vom Modulkoordinator eine mündliche Prüfung angesetzt werden.

Code: MScTI_GPUALG	**Form:** WP	**Modulname:** GPU Algorithm Design	
Modulverantwortlicher: Prof. Dr. Robert Strzodka		**Typ:** Vorlesung, Übung, Projektarbeit	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** WiSe

Modulteile:
- Vorlesung 2 h / Woche
- Übung 2 h / Woche mit Hausarbeiten in der ersten Semesterhälfte
- Projekt 2 h / Woche mit Hausarbeiten in der ersten Semesterhälfte

Lernziele: Die Studierenden...
- sind in der Lage, das gesamte Parallelismus- und Bandbreitenpotenzial großer GPUs auszunutzen,
- können GPU-Designentscheidungen bezüglich Tradeoffs in parallelen Algorithmen treffen,
- wissen, wie fortgeschrittene GPU-Transformationen zur Erhöhung von Parallelismus und Datenlokalität angewendet werden,
- verstehen, wie numerische Effizienz und parallele Effizienz ausbalanciert werden können.

Lerninhalte:
- Neueste Entwicklungen bei GPUs,
- Datentransformationen simultan zum Datentransport,
- Optimierungen der Datenlokalität,
- hierarchische Algorithmen,
- Nutzung von SIMD,
- Präzision, Genauigkeit und numerische Verfahren,
- numerische Effizienz vs. parallele Effizienz,
- Datenrepräsentation.

Voraussetzungen: Kenntnisse in C++ und CUDA	**Empfohlene Vorkenntnisse:** C++23 und STL, GPU und CUDA

Literatur:
- David B. Kirk, Wen-mei W. Hwu: *Programming Massively Parallel Processors* (3rd ed, Morgan Kaufmann, 2017)
- Zusätzliches Material wird über die Lernplattform bereitgestellt.

Voraussetzung für die Vergabe von Leistungspunkten:
Für die Teilnahme an der Projektprüfung, die aus einem Software-Projekt mit Dokumentation sowie einer Präsentation der Ergebnisse besteht, sind mindestens 50% der Punkte aus den Übungen erforderlich. Alternativ zur Projektprüfung kann vom Modulkoordinator eine mündliche Prüfung angesetzt werden.

Code: MScTI_COCO	**Form:** WP	**Modulname:** Consistency and Coherence	
Modulverantwortlicher: Prof. Dr. Holger Fröning		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- kennen Kohärenz- und Konsistenzprinzipien paralleler Architekturen,
- kennen Entwurfsverfahren für skalierbare Synchronisations- und Kommunikationsmechanismen,
- sind mit fortgeschrittenen Konzepten wie Transaktionalem Speicher, Relaxierter Konsistenz und skalierbarer Kohärenz vertraut,
- wissen, wie komplexer Parallelcode für spezifische Kommunikations- und Synchronisationsprobleme entworfen und optimiert werden kann,
- sind in der Lage, komplexe Rechenprobleme mit massiv parallelen Prozessoren zu lösen, die Auswirkungen architektonischer Entscheidungen auf Zeit- und Energieverbrauch zu verstehen und die Eignung bestimmter Prozessorarchitekturen für gegebene Rechenprobleme fundiert zu beurteilen.

Lerninhalte:
- Shared-Memory-Architekturen,
- Programmierparadigmen, Kommunikations- und Synchronisationskonzepte sowie zugehörige Algorithmen,
- Konsistenzmodelle und skalierbare Cache-Kohärenz,
- Multi-/Many-Core- und Multithreading-Architekturen,
- Neue und aufkommende Architekturen.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** Rechnerarchitektur (z.B. “GPU Computing”, “High Performance & Distributed Computing”, oder ”Advanced Computer Architecture”), Prinzipien der parallelen Programmierung, C, C++, OS Grundlagen

Literatur:
- M. Herlihy, N. Shavit: *The Art of Multiprocessor Programming* (Morgan Kaufmann, 2012)
- J.L. Hennessy, D.A. Patterson: *Computer Architecture: A Quantitative Approach* (Morgan Kaufmann, 2017)

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_HPDC	**Form:** WP	**Modulname:** High-Performance and Distributed Computing	
Modulverantwortlicher: Prof. Dr. Holger Fröning		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- kennen Message Passing, Cluster-Computing-Architekturen und skalierbare Programmierung,
- sind mit den wichtigsten früheren und aktuellen Konzepten zur Lösung großskaliger Rechenprobleme vertraut,
- können Lösungen für großskalige Rechenprobleme entwerfen und optimieren,
- wissen, wie MPI und verwandte Softwarewerkzeuge zur Umsetzung großskaliger Rechenprobleme eingesetzt werden,
- sind in der Lage, großskalige Rechenprobleme unter Berücksichtigung von Zielen wie Laufzeit-, Energieeffizienz sowie Skalierbarkeit in Bezug auf Zeit und Kapazität zu lösen.

Lerninhalte:
- HPC-Architekturen und Message Passing,
- Entwurf paralleler Algorithmen und Message Passing Interface (MPI),
- Interna von MPI,
- Charakterisierung und Benchmarking,
- Kurze Einführung in das parallele Training von Modellen des maschinellen Lernens,
- Praktische Probleme und deren Lösungsansätze.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** Rechnerarchitektur (z.B. “GPU Computing”, “High Performance & Distributed Computing”, oder ”Advanced Computer Architecture”), Prinzipien der parallelen Programmierung, C, C++, OS Grundlagen

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

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_ECP	**Form:** WP	**Modulname:** Emerging Computing Paradigms	
Modulverantwortlicher: Prof. Dr. Nima TaheriNejad		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** WiSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- können das grundlegende Prinzip des neuromorphen Rechnens erklären,
- können mindestens eine Approximate-Computing-Lösung auf Software-, Architektur- und Schaltungsebene benennen und beschreiben,
- können beurteilen, ob Approximate Computing für eine spezifische Anwendung eingesetzt werden kann,
- sind in der Lage, den grundlegenden Vorteil des In-Memory Computing zu beschreiben,
- kennen mindestens vier neuartige Speichertechnologien,
- können den Zustand memristiver Schaltungen analysieren und bewerten,
- können mindestens einen Vor- und Nachteil des stochastischen Rechnens nennen,
- können das Prinzip des anderen Rechnens wie Quantenrechnens und Optischen Rechnens beschreiben

Lerninhalte:
- Herausforderungen im Rechnen,
- ML-Beschleuniger und neuromorphes Rechnen,
- Approximate Computing,
- In-Memory Computing,
- Neue Speichertechnologien,
- Memristives Rechnen,
- Stochastisches Rechnen,
- Andere neuartige Rechnenbeispiele

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** Grundlagen der Rechnerarchitektur. Idealerweise: CCS, HDL, VLSI, ACA, und Embedded Machine Learning

Literatur:
- Wird von den Dozenten bekannt gegeben.

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_ACF	**Form:** WP	**Modulname:** Architecture and CAD for FPGAs	
Modulverantwortlicher: Prof. Dr. Dirk Koch		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / woche	**Semester/Turnus:** WiSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- benennen und erläutern fortgeschrittene Komponenten von FPGA-Bausteinen,
- benennen und erklären architektonische Details von FPGA-Geräten,
- verstehen die Entwurfsfaktoren, die die FPGA-Entwicklung beeinflussen,
- verstehen das notwendige Ökosystem zur Entwicklung von FPGA-Chips sowie die dafür benötigten CAD-Werkzeuge,
- entwerfen eigene, angepasste FPGAs einschließlich Logik, spezieller Primitiven, Routing-Fabriken und Konfigurationslogik,
- verstehen und nutzen Werkzeuge zur Logiksynthese, Technologiemapping sowie Platzierung und Verdrahtung (Place & Route),
- verstehen Teststrategien für FPGAs, einschließlich der Tests auf Werksebene,
- verstehen Verfahren der Hardware-Verifikation, einschließlich der internen Funktionsweise von Simulatoren.

Lerninhalte:

Ziel des Moduls ist ein vertieftes Verständnis des gesamten Ökosystems, das erforderlich ist, um benutzerdefinierte Logik auf einem FPGA lauffähig zu machen - sowohl der Hardware selbst als auch der benötigten Werkzeuge zur Abbildung der Logik
- Programmierbarkeit der Hardware: rekonfigurierbare Logik- und Routing-Strukturen,
- Spezielle Blöcke für Speicher- und Arithmetikfunktionen - deren Nutzen und Implementierung,
- Verständnis für die Anforderungen typischer FPGA-Anwendungen,
- Performance Optimierung: Flächenverbrauch, Geschwindigkeit und Energieeffizienz,
- Physikalische Implementierungen von (eingebetteten) FPGAs im Vergleich zu anderen ASICs,
- Logiksynthese/Technologiemapping: Umsetzung benutzerdefinierter Logik in FPGA-Primitiven,
- Platzierung und Verdrahtung (Place & Route) von Schaltungen,
- Bitstream-Erzeugung und partielle Rekonfiguration,
- Werkstest und Chip-Charakterisierung: Wie wird sichergestellt, gefertigte Chips funktionien?,
- Simulationstechniken: Genauigkeit vs. Geschwindigkeit bei der Simulation,
- Robustheit und Sicherheit in FPGAs,
- Ausblick auf neue Technologien: resistive RAM, Phasenwechsel-Speicher, Spintronik.

Voraussetzungen:	**Empfohlene Vorkenntnisse:**
keine	Verilog/VHDL, Algorithmen und Datenstrukturen, FPGA Basics

Literatur:
- Eine aktuelle Liste von Fachartikeln wird in der Lehrveranstaltung bereitgestellt.

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_EEC	**Form:** WP	**Modulname:** Energy Efficient Computing	
Modulverantwortlicher: Prof. Dr. Dirk Koch		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- verstehen die Bedeutung und Auswirkungen der Energieeffizienz in Rechensystemen,
- lernen die Ursachen des Energieverbrauchs in einem Rechensystem kennen,
- verstehen Gemeinsamkeiten und Unterschiede zwischen dem Rechnen in Rechenzentren und in eingebetteten Systemen,
- verstehen und bewerten Hard- und Softwaretechnologien sowie Methoden zum Aufbau energieeffizienter Systeme,
- entwickeln und implementieren leistungsfähigen und energieeffizienten Code.

Lerninhalte:

Energieeffizienz ist vermutlich das wichtigste Ziel in nahezu allen Rechensystemen. Sie ermöglicht nicht nur die mobile Revolution, sondern ist auch entscheidend dafür, Milliarden von Transistoren auf einem einzigen Prozessorchip unterzubringen und wirkt somit als Leistungstreiber. Darüber hinaus verfügen eingebettete Systeme in der Regel nur über begrenzte Energiebudgets, und die globale Erwärmung zwingt uns dazu, unsere Rechensysteme neu zu denken.

- Notwendigkeit energieeffizienten Rechnens,
- Einfluss von Technologie- und Systementwurfsentscheidungen auf die Energieeffizienz,
- Entwurfsfaktoren, die Kosten, Leistung und Energieeffizienz beeinflussen,
- Einfluss des Speichers auf Leistung und Energieeffizienz,
- Techniken zur Leistungsoptimierung, einschließlich Frequenz- und Spannungs-Skalierung, SIMD- und Multi-Core-Verarbeitung,
- Spezialisierte Hardware und Beschleuniger für leistungsstärkeres und energieeffizienteres Rechnen,
- Bestehende (z. B. Chiplets und 3D-Integration) und neu entstehende Technologien für energieeffizientes Rechnen,
- Durchführung von Designraumexplorationen,
- Programmierung und Bewertung verschiedener Techniken zur Leistungs- und Energieeffizienzoptimierung.

Voraussetzungen:	**Empfohlene Vorkenntnisse:**
keine	keine

Literatur:
- Literatur wird in der Veranstaltung zur Verfügung gestellt

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_MCC	**Form:** WP	**Modulname:** Memory-Centric Computing	
Modulverantwortlicher: Prof. Dr. Nima TaheriNejad		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- beschreiben die grundlegenden Ideen und die Entwicklung der Speichertechnologien,
- nennen mindestens drei architektonische Entwürfe, die im speicherzentrierten Rechnen verwendet werden,
- erläutern die Prinzipien und Vorteile speicherzentrierter Rechenparadigmen sowie deren Auswirkungen auf Energieeffizienz und Leistung,
- wenden eine oder mehrere etablierte speicherzentrierte Rechenlösungen an,
- benennen mindestens drei neuartige speicherzentrierte Rechenlösungen sowie deren Vor- und Nachteile,
- ordnen mindestens zwei spezifische Speichertechnologien den jeweils am besten geeigneten Anwendungen zu, wie z.B. KI, maschinelles Lernen oder Edge Computing,
- nennen drei wesentliche Herausforderungen im speicherzentrierten Rechnen und schlagen jeweils eine potenzielle Lösung vor,
- fassen zwei zentrale Fortschritte zusammen, die in Fallstudien dargestellt werden.

Lerninhalte:
- Überblick über verschiedene Speichertechnologien, einschließlich SRAM, DRAM, Flash-Speicher, PCM, RRAM, MRAM usw.,
- Grundlagen der Entwurfsprinzipien und Architekturen des speicherzentrierten Rechnens,
- energieeffiziente und leistungsstarke Lösungen durch speicherzentriertes Rechnen,
- systemweite Integration von Speichertechnologien und deren Anwendungen in KI, maschinellem Lernen und Edge Computing,
- Herausforderungen und offene Forschungsfragen im Bereich des speicherzentrierten Rechnens,
- eine oder mehrere Fallstudien zu fortgeschrittenen speicherzentrierten Anwendungen.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** Advanced computer architecture und/oder Emerging computing paradigms.

Literatur:
- Wird von den Dozenten bekannt gegeben.

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_BIC	**Form:** ???	**Modulname:** Brain Inspired Computing	
Modulverantwortlicher: Prof. Dr. Johannes Schemmel		**Typ:** Lecture with exercise	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / week	**Semester/Turnus:** WiSe

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

Lernziele: Die Studierenden...
- erhalten ein grundlegendes Verständnis der Informationsverarbeitung im biologischen Gehirn,
- erwerben Basiswissen über die zugrunde liegenden mathematischen Modelle,
- entwickeln ein vertieftes qualitatives und quantitatives Verständnis analoger elektronischer Schaltungen, die zum Modellieren biologischer Neuronen erforderlich sind,
- verstehen, wie sich einfache künstliche neuronale Netze aus analogen Neuronen mithilfe des *Lui*-Neuronmoduls aufbauen lassen.

Lerninhalte:
- Membrandynamik, Ionenkanäle und das Hodgkin-Huxley-Modell,
- vereinfachte Neuronenmodelle,
- neuronale Verschaltung und Synapsen,
- einfache neuronale Schaltkreise,
- analoge Schaltungen zur Modellierung des Gehirns,
- Schaltungssimulation mit KiCad,
- Spannungsquellen, Grundschaltungen und Verstärker,
- Neuronenschaltungen,
- das physikalische Neuronenmodul *Lui*,
- Aufbau kleiner Schaltungen mit physischen Neuronenmodulen.

Voraussetzungen: Fundierte Kenntnisse in Mathematik, Biologie und Physik auf Abiturniveau.	**Empfohlene Vorkenntnisse:** MScTI_ANADESIGN oder Experimentalphysik II, Mathematische Methoden für Physik und Ingenieurwissenschaften oder vergleichbare Mathematikkenntnisse

Literatur:
- Wird von den Dozenten bekannt gegeben.

Voraussetzung für die Vergabe von Leistungspunkten:
Klausur oder mündliche Prüfung, Details für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_BIOSIG	**Form:** WP	**Modulname:** Biosignal Processing and Machine Learning	
Modulverantwortlicher: Dr. Mostafa Haghi and Dr. Amin Aminifar		**Typ:** Vorlesung, Übung	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe

Modulteile:
- Vorlesung (2 h / Woche)
- Praktische Übung mit Hausarbeiten (2 h / Woche)

Lernziele: Die Studierenden...
- verstehen und beschreiben die wichtigsten Biosignale in biomedizinischen Anwendungen,
- führen die Vorverarbeitung und Verarbeitung von Biosignalen wie dem Elektrokardiogramm durch,
- beschreiben und implementieren Biosignalverarbeitungstechniken wie die diskrete Wavelet-Transformation zur Anpassung an die Frequenzgrenzen kardiopulmonaler Parameter,
- wenden Biosignalverarbeitungstechniken zur Erkennung kardialer Anomalien an,
- identifizieren und unterscheiden bekannte Datenformate für Biosignale,
- erklären die grundlegenden Konzepte des maschinellen Lernens,
- implementieren und nutzen Deep Learning für medizinische Anwendungen,
- beschreiben mindestens eine Lösung des maschinellen Lernens zur Behandlung von Rauschproblemen in der Biosignalverarbeitung,
- wenden maschinelles Lernen im Kontext der Biosignalverarbeitung an.

Lerninhalte:
- Einführung in Biosignale und Signalverarbeitung (allgemeine Übersicht),
- Signalgenerierung und Signalverbesserung,
- Signalvisualisierung,
- Grundlagen der R-Welle und deren Erkennung im Elektrokardiogramm,
- Computergestützte Diagnose (CAD) – Biosignale und Detektion von Vorhofflimmern,
- Datenformate für Biosignale,
- Einführung in den Einsatz von maschinellem Lernen im biomedizinischen und gesundheitsbezogenen Bereich,
- Klassisches maschinelles Lernen im biomedizinischen Kontext,
- Deep Learning im biomedizinischen Kontext,
- Reinforcement Learning im biomedizinischen Kontext,
- Umgang mit verrauschten Daten mithilfe von maschinellem Lernen,
- Aktuelle und zukünftige Herausforderungen für maschinelles Lernen in biomedizinischen Anwendungen.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** Grundlagen der Mathematik, Programmierung mit Python

Literatur:
- Wird in der Veranstaltung bekannt gegeben

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_EC_Wahl	**Form:** WP	**Modulname:** Current topics in Emerging Computing	
Modulverantwortlicher: Verschieden		**Typ:** Vorlesung oder praktischer Kurs	
ECTS-Punkte: 6	**Workload:** 180 h	**Lehrstunden:** 4 / Woche	**Semester/Turnus:** SoSe oder WiSe

Modulteile:
- Wird von den Dozenten vor Beginn des Moduls bekannt gegeben

Lernziele: Die Studierenden...
- zeigen vertiefte Kenntnisse in einem spezifischen Bereich neuartiger Rechentechnologien,
- analysieren fortgeschrittene Forschungsfragen in einem spezifischen Bereich neuartiger Rechentechnologien,
- nutzen Konzepte aus Mathematik, Physik, Informatik und Ingenieurwissenschaften, um Lösungen für Forschungsfragen in einem spezifischen Bereich neuartiger Rechentechnologien zu entwickeln.

Lerninhalte:
- Experimentelle, theoretische und/oder praktische Methoden in einem spezifischen Bereich neuartiger Rechentechnologien.

Voraussetzungen: keine	**Empfohlene Vorkenntnisse:** keine

Literatur:
- Wird von den Dozenten bekannt gegeben.

Voraussetzung für die Vergabe von Leistungspunkten:
Voraussetzungen für die Vergabe von Leistungspunkten sowie die Prüfungsformen werden zu Beginn der Lehrveranstaltung bekannt gegeben.

Code: MScTI_SA	**Form:** P	**Modulname:** Studienarbeit	
Modulverantwortlicher: Alle Gruppen		**Typ:** Praktische Studienarbeit	
ECTS-Punkte: 15	**Workload:** 450 h	**Lehrstunden:** 2 / Woche	**Semester/Turnus:** SoSe/ WiSe
Modulteile: • Praktische Studienarbeit			
Lernziele: Die Studierenden... • können sich in wissenschaftliche und technische Aspekte eines ausgewählten Themas vertiefen, • ein kleines Forschungsprojekt planen und durchführen, • einen Bericht von angemessener Länge verfassen.			
Lerninhalte: • Forschungsarbeit zu einem spezifischen Thema, • Organisation und Durchführung der Arbeit, • Erstellung eines mittelumfangreichen Berichts.			
Voraussetzungen: keine		**Empfohlene Vorkenntnisse:** Kenntnisse im entsprechenden Fachgebiet	
Literatur: • Material wird vom Betreuer bereitgestellt.			
Voraussetzung für die Vergabe von Leistungspunkten: schriftlicher Bericht			

Code: MScTI_THESIS	**Form:** P	**Modulname:** Masterarbeit	
Modulverantwortlicher: Alle Gruppen		**Typ:** Masterarbeit	
ECTS-Punkte: 30	**Workload:** 900 h	**Lehrstunden:** n.a.	**Semester/Turnus:** SoSe und WiSe
Modulteile: • Masterarbeit			
Lernziele: Die Studierenden... • planen und führen ein umfangreiches Forschungsprojekt durch, • verfassen eine ausführliche Abschlussarbeit, • präsentieren ihre eigene wissenschaftliche Arbeit in einer mündlichen Vorstellung.			
Lerninhalte: • Forschungsarbeit zu einem spezifischen Thema, • Organisation und Management der Arbeit, • Anfertigung einer längeren schriftlichen Abschlussarbeit, • Mündliche Präsentation im Kolloquium.			
Voraussetzungen: keine		**Empfohlene Vorkenntnisse:** Kenntnisse im entsprechenden Fachgebiet	
Literatur: • Material wird vom Betreuer bereitgestellt.			
Voraussetzung für die Vergabe von Leistungspunkten: schriftliche Masterarbeit, Kolloquium			