Institute of Cyber-physical Systems							
Name of the subject:	Code of the subject:	Credits:	Weekly hours:				
					lec	sem	lab
Advanced Computer Architectures I	NIXKA1EBNE	2	full-time		2	0	0
Responsible person for the subject: Dr. Mehdi Taassori			Classification:				
Subject lecturer(s):							
Prerequisites:	Introduction to Computer Architectures			NIESA1EBNE			
Way of the assessment:	Exam						

Course description

Goal:	This class provides an approach to Advanced Computer Architecture. Learn how a modern computer works and implement a principled design.
Course description:	Fundamentals of quantitative design and analysis. Principles in processor design. Parallelism and its exploitation. The focus is on fundamental techniques to improve the performance in computer architecture.

Lecture schedule

Education week	Topic
1.	• Fundamentals of Quantitative Design and Analysis ○ Introduction ○ Classes of Computers ○ Defining Computer Architecture
2.	• Fundamentals of Quantitative Design and Analysis ○ Trends in Technology ○ Trends in Power and Energy in Integrated Circuits ○ Quantitative Principles of Computer Design
3.	• Processor (Pipeline) ○ An Overview of Pipelining ○ Pipelined Datapath and Control ○ Data Hazards: Forwarding versus Stalling
4.	• Processor (Pipeline) ○ Control Hazards ○ Exceptions ○ Parallelism via Instructions
5.	• Instruction-Level Parallelism and Its Exploitation ○ Instruction-Level Parallelism: Concepts and Challenges ○ Basic Compiler Techniques for Exposing ILP ○ Reducing Branch Costs With Advanced Branch Prediction
6.	• Instruction-Level Parallelism and Its Exploitation ○ Overcoming Data Hazards With Dynamic Scheduling ○ Dynamic Scheduling: Examples and the Algorithm ○ Hardware-Based Speculation ○ Exploiting ILP Using Multiple Issue and Static Scheduling
7.	• Instruction-Level Parallelism and Its Exploitation ○ Exploiting ILP Using Dynamic Scheduling, Multiple Issue, and Speculation ○ Advanced Techniques for Instruction Delivery and Speculation ○ Multithreading: Exploiting Thread-Level Parallelism to Improve Uniprocessor Throughput
8.	• Data-Level Parallelism in Vector, SIMD, and GPU Architectures ○ Introduction ○ Vector Architecture

9.	• Data-Level Parallelism in Vector, SIMD, and GPU Architectures o SIMD Instruction Set Extensions for Multimedia
10.	• Data-Level Parallelism in Vector, SIMD, and GPU Architectures o Graphics Processing Units o Detecting and Enhancing Loop-Level Parallelism
11.	• Thread-Level Parallelism o Introduction o Centralized Shared-Memory Architectures
12.	• Thread-Level Parallelism o Performance of Symmetric Shared-Memory Multiprocessors o Distributed Shared-Memory and Directory-Based Coherence
13.	• Thread-Level Parallelism o Synchronization: The Basics o Models of Memory Consistency: An Introduction
14.	• Warehouse-Scale Computers to Exploit Request-Level and Data-Level Parallelism

Mid-term requirements

Conditions for obtaining a mid-term grade/signature	Quizzes, Homeworks, Project A minimum of 51% must be achieved in each part to receive a signature.

Assessment schedule

Education week	Topic

Method used to calculate the *mid-term grade* (to be filled out only for subjects with mid-term grades)

Type of the replacement

Type of the replacement of written test/mid-term grade/signature	

Type of the exam (to be filled out only for subjects with exams)

Written and multiple-choice exam

Calculation of the exam mark (to be filled only for subjects with exams)

- Homework 5%
- Quiz 0 - 10%
- Project 0 - 10%
- Exam 75% - 95%
- The submission of homework and project by the designated deadline is mandatory for all students.
- Attendance for quizzes, and the exam is mandatory.
- Conducting the quiz and delivering the project depends on the class schedule.

Final grade calculation methods:

0-59 points - Fail
60-69 points - Pass
70-79 points – Satisfactory
80-89 points - Good
90-100 points – Excellent

References	
Obligatory:	Hennessy, John L., and David A. Patterson. *Computer architecture: a quantitative approach*. Elsevier, Sixth edition.
Recommended:	Hennessy, John L., and David A. Patterson. *Computer architecture: a quantitative approach*. Elsevier, Fifth edition.
Other references:	Sorin, Daniel J., Mark D. Hill, and David A. Wood. A primer on memory consistency and cache coherence. *Synthesis lectures on computer architecture* (2011): 1-212.