CENG 309 - Introduction to Computer Architecture

Course Code	Course Name	Semester	
CENG 309	Introduction to Computer Architecture	Fall ☒ Spring ☐ Summer ☐	
Hours		**Credit**	**ECTS**
Theory	**Practice** / **Lab**	3	7
3	0 / 0		

Course Details	
Department	Computer Engineering
Course Language	English
Course Level	Undergraduate ☒ Graduate ☐
Mode of Delivery	Face to Face ☒ Online ☐ Hybrid ☐
Course Type	Compulsory ☒ Elective ☐
Course Objectives	This course introduces the fundamental concepts of computer architecture. It covers the structure, functionality, and design of modern computer systems, focusing on the hardware-software interface, performance optimization, and system organization. Key topics include instruction set architecture (ISA), pipelining, memory hierarchies, input/output systems, and parallelism.
Course Content	The course covers an in-depth exploration of computer architecture, beginning with an overview of computer systems and their evolution, including the fundamental structures of Von Neumann and Harvard architectures. It discusses measuring computer performance through various metrics such as execution time, throughput, and latency, and explores benchmarking using SPEC, MIPS, and FLOPS, along with Amdahl's Law. The course examines Instruction Set Architecture (ISA) and machine language, comparing CISC and RISC architectures, addressing instruction formats, and exploring addressing modes. It dives into the design and operation of the Arithmetic Logic Unit (ALU), covering binary arithmetic and logical operations, and details hardware implementation. Pipelining, including stages, benefits, and performance optimization, is explained with real-world examples. The course also addresses pipeline hazards and methods for hazard resolution, including data forwarding and branch prediction. Memory hierarchy, cache design, and virtual memory are studied for their impact on performance. The curriculum includes input/output systems, buses, and Direct Memory Access (DMA). Parallel computing concepts like shared and distributed

	memory systems, multithreading, and synchronization issues are introduced, along with the architecture of GPUs and their applications. Power efficiency, energy-efficient design techniques, and advancements in computing such as quantum and neuromorphic computing are covered, with discussions on current trends in CPU and GPU development, cloud computing, and future challenges in computer architecture.
Course Method/ Techniques	Anlatım ☒ Soru-Cevap ☒ Sunum ☒ Müzakere ☒
Prerequisites/ Corequisites	
Work Placement(s)	

Textbook/References/Materials

- **Required Text:** *Computer Organization and Design: The Hardware/Software Interface* by David Patterson and John Hennessy (5th Edition)
- **Supplementary Readings:** Selected research papers and articles

Course Category				
Mathematics and Basic Sciences	☐		Education	☐
Engineering	☒		Science	☐
Engineering Design	☐		Health	☐
Social Sciences	☐		Profession	☐

Weekly Schedule		
No	**Topics**	**Materials/Notes**
1	Introduction to Computer Architecture • Overview of Computer Architecture and its importance • Evolution of computer systems • Basic structure and function of a computer Von Neumann architecture vs. Harvard architecture	Reviewing examples of related topics from the source book.
2	Computer Performance and Benchmarks • Measuring performance: execution time, throughput, latency	Reviewing examples of related topics from the source book.

	• Factors affecting performance: clock cycles, cycle time, CPI • Benchmarking and performance metrics: SPEC, MIPS, FLOPS Amdahl's Law	
3	Instruction Set Architecture (ISA) and Machine Language • Introduction to ISA: what it is and why it's important • Types of ISAs: CISC vs. RISC • Instruction formats, addressing modes Machine language and assembly programming	Reviewing examples of related topics from the source book.
4	Arithmetic Logic Unit (ALU) Design • Introduction to ALU and its role in CPU • Binary arithmetic: addition, subtraction, multiplication, division • Logical operations (AND, OR, XOR) Hardware implementation of basic operations	Reviewing examples of related topics from the source book.
5	Pipelining and Performance Enhancement • Introduction to pipelining: stages, benefits, and challenges • Instruction pipeline and pipeline hazards • Clock cycle optimization Real-world examples of pipelined processors	Reviewing examples of related topics from the source book.
6	Hazards in Pipelining and Data Forwarding • Structural, data, and control hazards • Methods for resolving hazards: data forwarding, branch prediction • Stalling and its impact on performance Example: pipeline implementation in modern processors	Reviewing examples of related topics from the source book.
7	Midterm Exam Review and Recap • Review of Weeks 1–6 topics: ISA, ALU, pipelining, and performance • Q&A session and sample problems Study guide and practice test	Reviewing examples of related topics from the source book.
8	**Midterm Exam**	Reviewing examples of related topics from the source book.
9	Memory Hierarchy: Caches, Virtual Memory • Introduction to memory hierarchy: registers, cache, main memory	Reviewing examples of related topics from the source book.

	• Cache design: cache mapping techniques, cache hits/misses • Virtual memory and paging Performance implications of memory hierarchies	
10	Input/Output Systems and Buses • Basics of I/O devices and their role in computer systems • I/O performance metrics and optimization • I/O interfacing and buses (PCI, USB) Direct Memory Access (DMA) and its role in improving performance	Reviewing examples of related topics from the source book.
11	Graphics Processing Units (GPUs) • Introduction to GPUs and their architecture • Difference between CPUs and GPUs • Applications of GPUs in parallel computing and AI Performance and energy efficiency comparisons	Reviewing examples of related topics from the source book.
12	Power Efficiency and Advanced Architectures • Power consumption in modern computer systems • Energy-efficient processor design: clock gating, dynamic voltage scaling Modern advancements in computer architecture: quantum computing, neuromorphic computing	Reviewing examples of related topics from the source book.
13	Trends in Computer Architecture • Current trends in CPU and GPU design • Emerging technologies: cloud computing, edge computing, serverless architectures • Advances in processor design for AI and machine learning workloads Future challenges in computer architecture	Reviewing examples of related topics from the source book.
14	Parallelism: Multiprocessors and Multithreading • Introduction to parallelism and its importance in modern computing • Shared memory multiprocessors vs. distributed memory • Multithreading: hardware vs. software-level threads	Reviewing examples of related topics from the source book.

	Synchronization issues in parallel computing	
15	Final Exam Review	
16	**Final Exam**	

Assessment Methods and Criteria		
In-term studies	**Quantity**	**Percentage**
Attendance		%10
Lab		
Practice		
Fieldwork		%5
Course-specific internship		
Quiz/Studio/Criticize		
Homework		
Presentation / Seminar		%5
Project		
Report		
Seminar		
Midterm Exam		%20
Final Exam		%60
	Total	**%100**
Contribution of Midterm Studies to Success Grade		
Contribution of End of Semester Studies to Success Grade		
	Total	

ECTS Allocated Based on Student Workload			
Activities	**Activities**	**Activities**	**Activities**
Course Hours	16	3	48
Lab			
Practice			
Fieldwork			
Course-specific Work Placement			
Out-of-class study time	15	5	75
Quiz/Studio/Criticize			
Homework			
Presentation / Seminar	3	2	6
Project			
Report			
Midterm Exam and Preparation for Midterm	1	16	16
Final Exam and Preparation for Final Exam	1	30	30
Total Workload			**175**
Total Workload / 25			**175/25**
ECTS Credit			**7**

Program Competencies	
No	**Outcome**
Ö1	Understand the fundamental concepts and architecture of computer systems including the distinctions between Von Neumann and Harvard architectures.
Ö2	Apply various performance measurement techniques and analyze computer performance using standard benchmarks such as SPEC, MIPS, and FLOPS.
Ö3	Demonstrate an understanding of instruction set architectures (ISA), including differences between RISC and CISC, and explain how instruction formats and addressing modes affect CPU operation.
Ö4	Design and implement basic arithmetic operations and logical circuits, focusing on binary arithmetic (addition, subtraction, multiplication, division) and logical operations (AND, OR, XOR) for CPU design.
Ö5	Identify, analyze, and resolve data, control, and structural pipeline hazards to optimize processor throughput and performance.

Contribution of Course Learning Outcomes to Program Competencies/Outcomes

Contribution Level: 1: Very Slight, 2: Slight, 3: Moderate, 4: Significant, 5: Very Significant

	P1	**P2**	**P3**	**P4**	**P5**	**P6**	**P7**	**P8**	**P9**	**P10**	**P11**	**P12**	**P13**	**P14**	**P15**	**Toplam**
Ö1	5	5	5	5												20
Ö2	5	5	5	5												20
Ö3	5	5	5	5												20
Ö4	5	5	5	5												20
Ö5	5	5	5	5												20
															Toplam	100