ECE 485 - Computer Organization and Design

Credits: 3, **Contact Hours:** One 160 minute lecture session per week.

Coordinator: E. Oruklu, Associate Professor of ECE

Textbook: Computer Organization and Design MIPS Edition The Hardware/Software Interface 5th Edition, by David Patterson, John Hennessy; ISBN: 978-0-12-407726-3

2019 Catalog Data: ECE 485: Computer Organization and Design. Credit 3.
This course provides the students with understanding of the fundamental concepts of computer architecture, organization, and design. It focuses on relationship between hardware and software and its influence on the instruction set and the underlying Central Processing Unit (CPU). The structural design of the CPU in terms of datapath and control unit is introduced. The technique of pipelining and hazard management are studied. Advanced topics include instruction level parallelism, memory hierarchy and cache operations, virtual memory, parallel processing, multiprocessors and hardware security. The end to end design of a typical computer system in terms of the major entities including CPU, cache, memory, disk, I/O, and bus with respect to cost/performance trade-offs is also covered. Differentiation between ECE 485 and ECE 585 is provided via use of projects / case studies at differing levels. Satisfies ECE Professional Elective (P).

Prerequisites by topic:

1. Boolean algebra, Combinational logic designs
2. Basic programming

Enrollment: Required course for CPE majors. Elective course for EE majors.

Specific outcomes of instruction:
After completing this course, the student should be able to do the following:

1. Use the performance / complexity tradeoffs for defining the RISC instruction set
2. Translate a high-level program into RISC instruction set
3. Write a RISC assembler level program including use of subroutines for repetitive tasks
4. Design an Arithmetic and Logic Unit (ALU) Hardware for RISC instruction set
5. Identify the single cycle datapath for execution of RISC instructions
6. Identify the multi cycle datapath on how a typical RISC instruction goes through its five stages
7. Develop the pipelining model and identify the hazards associated with its operation
8. Define the control unit and the associated control signals
9. Implement a control unit in various forms including sequential circuits and microprogram
10. Describe the hierarchical memory system and the cache operation
11. Describe the operation of the non-volatile storage system
12. Describe the basic operation of the I/O and the interconnecting bus
13. Develop and test a VHDL program to capture the processor module operation

Relationship of ECE 485 specific outcomes of instruction to student outcomes:

	Student Outcomes:	**Course Goals**
1	An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics	1, 2, 3, 4, 9, 13
2	An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors	4, 5, 6, 8, 9, 13
3	An ability to communicate effectively with a range of audiences	
4	An ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts	
5	An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives	
6	An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions	4, 7, 9, 10, 11, 12, 13
7	An ability to acquire and apply new knowledge as needed, using appropriate learning strategies	3, 7, 9,13

Topics:

- Intro to Computer Architecture (0.5week)
- Computer Performance (0.5 week)
- Introduction to VHDL (0.5 week)
- MIPS Instruction Set Arch (1.5 week)
- Computer Arithmetic (0.5 week)
- Arithmetic Logic Unit Design (0.5 week)
- Single Thread Behavior (0.5 week)
- Data Path and Control (0.5 week)
- Single Cycle Operation (1 week)
- ALU Control and Control Logic (0.5 week)
- Control Design/Microprogramming (0.5 week)
- Multi Cycle Operations (1 week)
- Pipelining (1 week)
- Structural and Data Hazards (0.5 week)
- Branch Hazards and Exceptions (0.5 week)
- Intro to Instruction Level Parallelism (0.5 week)
- Introduction to Memory Systems (0.5 week)
- Cache Fundamentals (0.5 week)
- Cache Perf Improvements (0.5 week)
- Intro to Virtual Memory (0.5 week)
- Disk Storage and reliability (0.5 week)
- Introduction to I/O and buses (0.5 week)
- Introduction to Hardware Security (1 week)

Prepared by: E. Oruklu **Date:** March 2nd, 2020