CMPE325 - Computer Architecture and Organization

Department: Computer Engineering

Instructor: Prof. Dr. Omar Ramadan, **E-mail:** omar.ramadan@emu.edu.tr, **Office:** CMPE115, **Tel.:** 1194

Assistant: TBA

Meeting times and places
Monday: 14:30-16:20 (CMPE128), Monday: 16:30-18:20, (CMP227), Thursday: 14:30-16:20 (CMPE128)

Program Name: Computer Engineering		**Program Code:** 25
Course Number: CMPE325	**Credits:** 4 Cr	**Year/Semester:** 2019-2020 Fall

☒ Required Course ☐ Elective Course

Prerequisite(s): CMPE 224 Digital Logic Systems

Catalog Description:
The main concern of this course is to provide a comprehensive overview of computer architecture with specific emphasis on design of reduced instruction set computers, helping the students understand the principles and tradeoffs such as cost/performance, or speed/flexibility, behind the design of modern computer systems. This course provides a foundation for bridging the gap between programming and the inner complexities of the computer.

Course Web Page:
http://cmpe.emu.edu.tr/courses/cmpe324

Textbook(s):
Computer Organization and Design: The Hardware/Software Interface, D.A. Patterson and J.L. Hennessy, 5th, Morgan Kaufmann, 2014. (ISBN: 978-0-12-374750-1)

Indicative Basic Reading List :
Computer Architecture: A Quantitative approach, J.L. Hennessy and D.A. Patterson, 3rd Ed., Morgan Kaufmann, 2003.

Topics Covered and Class Schedule (tentative):
(4 hours of lectures per week)

Weeks 1-2	Basic of Computer Organization; Design Principles of RISC Processors; Arithmetic Operators; data representation; Language of the Machine I; Instruction formats, register organization, memory access using load and store instructions, accessing arrays, memory addressing, assembly language conventions.
Weeks 3-4	Language of the Machine II; control instructions, looping in MIPS, comparison instructions, logical operators, pseudo instructions, instruction encoding, supporting procedures in computer hardware, passing the arguments to a procedure, register saving conventions, how to use stacks, nested calls.
Weeks 5	Arithmetic for computers; positive and negative binary numbers, addition, subtraction, shifting, logic operations and overflow detection; Designing ALU for basic MIPS instructions, Ripple carry adder and Carry look ahead adder implementations
Week 6	Designing ALU for basic MIPS instructions, Ripple carry adder and Carry look ahead adder implementations
Week 7	Floating point arithmetic algorithms; addition, subtraction, multiplication, and division, MIPS floating point instruction; Logic conventions and clocking for MIPS data-path.
Weeks 8,9	(Midterm Exam)
Weeks 11-12	MIPS single clock cycle implementation; Building a data-path for R-type, Immediate memory-register transfer instructions and control instruction, Designing ALU control, Comparing the performance of Single and Multi-cycle implementations; Control unit design of Single Cycle Data-path.
Week 13-14	The multiple clock cycle implementation, Designing the control unit for the multiple clock cycle implementation: Finite state machines (FSM) and Microprogramming.

Laboratory Schedule:

1.	Introduction to PCSpim (MIPS R2000 Simulator

2.	Memory Referenced MIPS Instructions Used In Accessing The Arrays
3.	Modular Programming in MIPS Using Jump-and-Link (jal) and Jump-Return (jr) Instructions
4.	Introduction to Circuit Synthesis Using ALTERA MAX-PLUS-II VHDL Tools.
5.	Single Clock Data Path in ALTERA MAX-PLUS-II VHDL Environment.
6.	Single Clock Data Path for 16-bit R-type Instructions in ALTERA MAX-PLUS-II VHDL Environment.

Course Learning Outcomes:

By the end of the course students should able to do following:

(1) Study the fundamentals of computer instruction set architecture, including machine-level instruction formats and addressing modes
(2) Describe the difference between RISC and CISC instruction sets
(3) Implement binary number representations and binary arithmetic.
(4) Ability to write, to encode, and to run a simple assembler program on a MIPS processor.
(5) Solve basic algorithms and hardware structures for performing integer computer arithmetic (addition, subtraction, multiplication, and division).
(6) Compute floating point format representation of real numbers in a computer
(7) Construct the basic algorithms and hardware structures for addition and multiplication of floating point numbers.
(8) Perform the basic structure and organization of a MIPS processor data-path.
(9) Learn the function of the control unit in a processor and will be able to design a control unit for a simple MIPS and Multi-clock based MIPS processor.

	Method	**No**	**Percentage**
Assessment (Tentative)	Quizzes	2	20%
	Midterm Exam	1	30%
	Labs	6	10 % (5% attendance, 5 % performance)
	Final Examination	1	40%

Policy on makeups: For eligibility to take a makeup exam, the student should bring a medical report within 3 working days of the missed exam. You will have only one make-up for Midterm or Final. Make-up will be organized at the end of the semester. No makeup will be done for quizzes.

Policy on cheating and plagiarism: Any student caught cheating at the exams or assignments will automatically fail the course and may be sent to the disciplinary committee at the discretion of the instructor.

Policy on NG grades: NG grade will be given in case of missing any exam without an acceptable excuse. NG will also be given in case of poor Lab attendance (<50%). NG will also be given in case of very poor attendance.

Contribution of Course to Criterion 5

Credit Hours for:
Mathematics & Basic Science : 0
Engineering Sciences and Design : 4
General Education : 0

Relationship of Course to Program Outcomes

The course has been designed to contribute to the following program outcomes:
1) an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
6) an ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions.

Prepared by: Prof. Dr. Omar Ramadan	**Date Prepared:** Sep. 26th , 2019