CMPE223/CMSE222 - Digital Logic Design

Department: Computer Engineering

Instructor information
Name: Prof. Dr. Omar Ramadan
E-mail: omar.ramadan@emu.edu.tr
Office: CMPE115

Meeting times and places
Tuesday: 16:30-18:20, Room: CMPE227 (Lab)
Wednesday: 10:30-12:20, Room: CMPE033
Thursday: 10:30-12:20, Room: CMPE033

Program Name: Computer Engineering	**Program Code:** 25	
Course Number: CMPE223/CMSE222	**Credits:** 4 Cr	**Year/Semester:** 2017-2018/ Spring

☒ Required Course ☐ Elective Course

Prerequisite(s):
MATH163 Discrete Mathematics

Catalog description:
Binary Systems (Binary Numbers, Octal and Hexadecimal Numbers, Number Base Conversions, Complements, Signed Binary Numbers, Binary Codes, Binary Logic). Boolean Algebra and Logic Gates (Basic Definitions, Basic Properties of Boolean Algebra, Boolean Functions, Canonical and Standard Forms, Other Logic Operations, Digital Logic Gates, ICs). Simplification of Boolean Functions (The Map Method, Tow-and Three- Variable maps, Four- and Five- Variable Maps, Product of Sums Simplification, NAND and NOR Implementation, Other Two-Level Implementations, Don't-Care Condition, The Tabulation Method, Determination of Prime Implicants, Selection of Prime Implicants). Combinational Logic (Design Procedure, Adders, Subtractors, Code Conversion, Analysis Procedure, Multilevel NAND Circuits, Multilevel NOR Circuits, Exclusive-OR Functions). MSI and PLD Components (Binary Adder and Subtractor, Decimal Adder, Decoders and Encoders, Multiplexers, PLA and PAL).

Course web page:
http://cmpe.emu.edu.tr/courses/cmpe223

Textbook(s):
J. F. Weakerly, "*Digital Design: Principles and Practices*", Prentice-Hall, 2006.

Indicative basic reading list :
1. S. Brown and Z. Vranesic, "Fundamentals of Digital Logic with VHDL Design", McGraw-Hill, 2009
2. Richard S. Sandige, "*Digital Design Essentials*," Prentice-Hall 2002.

Topics covered and class schedule (tentative):
(4 hours of lectures per week)

Week 1, 2	Binary, Octal, and Hexadecimal Numbers, Number Base Conversions, Signed Binary Numbers and Complements, Binary Addition, Subtraction, and Overflow, Binary Codes, and Binary Logic.
Weeks 3-8	Simplification of Boolean Functions, SOP and POS Simplifications, NAND and NOR Implementations, Other Two-Level Implementations, Multilevel NAND and NOR Circuits, Exclusive-OR Functions, Don't-Care Conditions, The Tabulation Method, Determination and Selection of Prime Implicants.
Weeks 9, 10	Midterm
Week 11	Introduction to VHDL, Entities and architectures, Identifiers, spaces and comments, Netlists, Signal assignments. Generics. Constant and open ports, Test benches and configurations.
Weeks 12, 13	Combinational Logic, Analysis Procedure, Design Procedure, Adders/Subtractors, Code Conversion, and VHDL Implementations.
Weeks 14, 15	MSI Components, Binary Adder and Subtractor, Decimal Adder, Decoders and Encoders, Multiplexers, VHDL Implementations.
Weeks 16-17	Final

Laboratory schedule (tentative):
(2 hours of laboratory per week)

Week 1, 2	Lab preparations and groups arrangements.

Week 3	Getting familiar with the tools
Week 4	Intro. Quartus II Design Environment & A Review Of VHDL Basics
Week 5	Intro. Quartus II Design Environment: Compiling And Simulating The Schematic Entry
Week 6	Intro. To Quartus II Software: Let's Review Our Mind
Week 7	Quartus II Design Environment: VHDL Basics
Week 8	Basic VHDL Prog. Of Combinational Circuits: Structural Programming Through NETLSTS
Week 9, 10	Midterm
Week 11	Basic VHDL Prog. Of Combinational Circuits …/ Continuation
Week 12	No Lab.
Week 13	Design and Implementation of Combinational Circuits
Week 14	Lab Quiz
Week 15	No Lab.

Course learning outcomes:
Upon successful completion of the course, students are expected to have the following competencies:
1. Use the Karnaugh map technique to simplify Boolean functions (SOP/POS) with/without don't care conditions (e1,e2,e3)
2. Design combinational logic circuits using AND, NOT, OR, NOR, NAND, XOR and XNOR logic gates (b1,b2,b3)
3. Use binary, octal, and hexadecimal number systems and apply techniques for number base conversions (a1,a2)
4. Perform the mathematical operations using signed and unsigned binary numbers (a1,a2)
5. Use algebraic manipulations associated with Boolean variables to build and evaluate Boolean expressions and functions (a1,a2,a3)
6. Find complements of Boolean functions using generalized DeMorgan's and/or duality methods (a2,a3)
7. Use the Tabulation technique to simplify Boolean functions (SOP/POS) with/without don't care conditions (e1,e2,e3)
8. Analyze combinational circuits and find their functions (e1,e2,e3)
9. Simulate the behavior of logic devices and combinational circuits using the VHDL language (k1,k2,k3)
10. Use functional combinational units such as adders/subtractors, comparators, decoders, multiplexers, to design larger size combinational logic systems (b1,b2,b3)

	Method	**No**	**Percentage**
Assessment (tentative)	Midterm Exam	1	35%
	Final Examination	1	45%
	Quiz and/or HW		5%
	Lab	~6	15% (5% attendance, 5% performance, 5% Lab Quiz)

Policy on makeup: There is no makeup for the quizzes. Only one makeup exam can be given for one of the missed exams (midterm or final). In order to be able to enter a makeup exam, you MUST submit a written report to your instructor stating your excuse within 3 days of that examination.

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 the following cases:
Lab attendance < 50% or Missing any exam without an acceptable excuse.

Contribution of course to ABET criterion 5
Credit Hours for:
Mathematics & Basic Science : 0
Engineering Sciences and Design : 4
General Education : 0

Relationship of the course to program outcomes
The course has been designed to contribute to the following program outcomes:
a) an ability to apply knowledge of mathematics, science, and engineering.
b) an ability to design and conduct experiments, as well as to analyze and interpret data
e) an ability to identify, formulate, and solve engineering problems.
k) an ability to use the techniques, skills, and modern engineering tools necessary for engineering practice.

Prepared by: Assoc. Prof. Dr. Muhammed Salamah / Prof. Dr. Omar Ramadan	**Date Prepared:** Feb. 12, 2018