CSCE 211: Digital Logic Design

1. Course number and name: CSCE 211: Digital Logic Design

2. Credit: 3-hrs; Contact: 3 lectures of 50 minutes each or 2 lectures of 75 minutes each per week

3. Instructor: Fall 2010: Larry M. Stephens, Chin-Tser Huang, and Manton Matthews
   Spring 2011: Chin-Tser Huang

4. Text book: Alan B. Markovitz, *Introduction to Logic Design, Third Edition,* McGraw Hill, 2010.

   Circuit kits provided at no cost.

5. Specific course information
   a. Catalog description: Number systems, Boolean algebra, logic design, sequential machines.
   b. Prerequisites: MATH 141
   c. Required in CE and CS curricula

6. Specific goals for the course
   a. Specific outcomes of instruction:
      - Represent numbers and perform arithmetic in bases 2, 8, 10, and 16.
      - Encode symbols and numbers in binary codes.
      - Add and subtract using 2's complement code.
      - Evaluate and simplify logical functions using Boolean algebra.
      - Represent logical functions in Canonical form and with AND, OR, NOT, XOR, NAND, NOR logic gates.
      - Analyze and design combinatorial circuits.
      - Simplify combinatorial circuits using Karnaugh maps.
      - Implement functions with NAND-NAND and NOR-NOR logic.
      - Analyze and design modular combinatorial logic circuits containing decoders, multiplexers, demultiplexers, 7-segments display decoders and adders.
      - Use the concepts of state and state transition for analysis and design of sequential circuits.
      - Use the functionality of flip-flops for analysis and design of sequential circuits. Introduce computational problem-solving techniques
   b. Relation of course outcomes to Student Outcomes: CE: see page 2; CS & CIS: see page 4

7. Topics covered and approximate weight (14 weeks, 3 hours/week, 42 hours total)
   1. Chapter 1, Introduction
   2. Chapter 2, Combinational Systems

3. Chapter 3, The Karnaugh Map
4. Chapter 5, Designing Combinational Systems with Medium Scale Integrated Circuits
5. Chapter 6, Analysis of Sequential Systems
6. Chapter 7, Design of Sequential Systems

c.

Computer Engineering

Relation of Course Outcomes to EAC Student Outcomes*

Course Outcomes (CE)	Student Outcomes											
	(a) apply knowledge of mathematics, science, and engineering	(b) design and conduct experiments, … interpret data	(c) design a system, component, or process to meet desired needs …	(d) function on multidisciplinary teams	(e) identify, formulate, and solve engineering problems	(f) an understanding of professional and ethical responsibility	(g) communicate effectively	(h) the broad education to understand the impact of engineering solutions …	(i) a recognition of the need for, and an ability to engage in life-long learning	(j) a knowledge of contemporary issues	(k) use the techniques, skills, and modern engineering tools ….	(CE) demonstrate knowledge of discrete mathematics [CE]
Criteria	a	b	c	d	e	f	g	h	i	j	K	CE
1. Represent numbers and perform arithmetic in bases 2, 8, 10, and 16.	2											
2. Encode symbols and numbers in binary codes.	1											
3. Add and subtract using 2's complement code.	2											
4. Evaluate and simplify logical functions using Boolean algebra.	2										3	2
5. Represent logical functions in Canonical form and with AND, OR, NOT, XOR, NAND, NOR logic gates.	2										3	1
6. Analyze and design combinatorial circuits.	2	2	3		2						3	
7. Simplify combinatorial circuits using Karnaugh maps.	2										3	
8. Implement functions with NAND-NAND and NOR-NOR logic.	2		3								3	

9. Analyze and design modular combinatorial logic circuits containing decoders, multiplexers, demultiplexers, 7-segments display decoders and adders.	1	2	3	2	2						3	
10. Use the concepts of state and state transition for analysis and design of sequential circuits.	2		3		2						3	1
11. Use the functionality of flip-flops for analysis and design of sequential circuits.	2		3		2						3	

* 3 = major contributor, 2 = moderate contributor, 1 = minor contributor; blank if not related

d.

Computer Science & Computer Information Systems

Relation of Course Outcomes to CAC Student Outcomes*

Course Outcomes (CS & CIS)	Student Outcomes											
	All									CS		CIS
	(a) apply knowledge of computing and mathematics appropriate to the discipline	(b) analyze a problem, and identify and define the computing requirements …	c) design, implement, and evaluate a computer-based system, …	(d) function effectively on teams to accomplish a common goal	(e) An understanding of professional, ethical, legal, … responsibilities	(f) communicate effectively with a range of audiences	(g) analyze the local and global impact of computing on … society	(h) Recognition of the need for … continuing professional development	(i) current techniques, skills, and tools necessary for computing practice	(j) apply mathematical foundations, algorithmic principles, and CS theory …	(k) apply design and development principles	(j) An understand processes that support the information systems environment.
Criteria	a	b	c	d	e	f	g	h	i	j	K	J
1. Represent numbers and perform arithmetic in bases 2, 8, 10, and 16.	2											
2. Encode symbols and numbers in binary codes.	1											
3. Add and subtract using 2's complement code.	2											
4. Evaluate and simplify logical functions using Boolean algebra.	2								3	2		
5. Represent logical functions in Canonical form and with AND, OR, NOT, XOR, NAND, NOR logic gates.	2								3	2		
6. Analyze and design combinatorial circuits.	2	2	3						3		3	
7. Simplify combinatorial circuits using Karnaugh maps.	2								3	2		

8. Implement functions with NAND-NAND and NOR-NOR logic.	2		3						3		3	
9. Analyze and design modular combinatorial logic circuits containing decoders, multiplexers, demultiplexers, 7-segments display decoders and adders.	1	2	3	2					3		3	
10. Use the concepts of state and state transition for analysis and design of sequential circuits.	2	2	3						3	2	3	
11. Use the functionality of flip-flops for analysis and design of sequential circuits.	2	2	3						3		3	

* 3 = major contributor, 2 = moderate contributor, 1 = minor contributor; blank if not related