

# ECE 212 Syllabus

<b>Course:</b>	ECE 212
<b>Credit Hours:</b>	3
<b>Course Title:</b>	Fundamentals of Logic Design
<b>Course Description:</b>	

Introduction to digital logic design. Boolean algebra, switching functions, Karnaugh maps, modular combinational circuit design, programmable logic, latches, flip-flops, finite state machines, synchronous sequential circuit design, datapaths, memory technologies, caches, and memory hierarchies. Use of several CAD tools for simulation, logic minimization, synthesis, state assignment, and technology mapping.

**Prerequisite(s):** C- or better in ECE 109

## **Textbook(s) and/or other required material:**

Required textbook:

Digital Design: Principles and Practices, Fourth Edition. John F. Wakerly. Prentice Hall, Upper Saddle River, NJ, 2006.

\$143.00

Hardware components that must be purchased by students: breadboard, wire cutters/strippers, and batteries.

## **Course objectives. By the end of this course, the student should be able to (use demonstrative verbs):**

1. Simplify Boolean algebraic expressions by applying Boolean algebra theorems.
2. Express functions in canonical sum-of-product and canonical product-of-sum forms.
3. Determine the function performed by a combinational- or sequential-logic circuit through analysis.
4. Design and synthesize a combinational- or sequential-logic circuit to perform a desired function.
5. Minimize the cost, complexity, power, and latency of a combinational- or sequential-logic circuit by applying various logic minimization approaches, including Boolean simplification, Karnaugh Maps, the Quine-McCluskey algorithm, and programmed minimization tools.
6. Describe and simulate a combinational- or sequential-logic circuit using the Verilog hardware description language.
7. Implement a combinational- or sequential-logic circuit using a programmable logic device.
8. Design combinational-logic building blocks including decoders, encoders, multiplexers, comparators, adders, subtractors, ALUs, and parity generators/checkers.
9. Design sequential-logic building blocks including latches, flip-flops, counters, and registers.
10. Construct complex digital subsystems using meaningful arrangements of decoders, encoders, multiplexers, comparators, adders, subtractors, ALUs, parity generators/checkers, counters, and registers.
11. Design a datapath and the control logic to orchestrate the datapath.
12. Determine the most appropriate memory technology for a particular memory application, by

# ECE 212 Syllabus

matching technology attributes to requirements.

13. Design a cache.

14. Simulate the operation of a cache given a memory address trace.

15. Determine the average memory access time given a memory address trace and a memory hierarchy.

## Topics covered:

boolean algebra (2 lectures);standard representations of logic functions, combinational-circuit analysis (1 lecture);combinational-circuit synthesis (1 lecture);logic minimization with Karnaugh Maps (1 lecture);logic minimization including don't-cares (1 lecture);logic minimization with Quine-McCluskey algorithm (1 lecture);timing hazards, hardware description languages & Verilog, Espresso minimization tool (1 lecture);programmable logic devices (PLA, PAL), tri-state buffers (1 lecture);decoders, encoders (1 lecture);cascaded decoders, multiplexers (MUX) (1 lecture);cascaded MUXes, implementing logic with MUXes, XOR, XNOR, comparators, 1-bit full-adder, adders, subtractors, ALUs (1 lecture);transient faults, parity generation and checking (1 lecture);latches and flip-flops (1 lecture);sequential-circuit analysis (1 lecture);sequential-circuit synthesis (1 lecture);state machine design (1 lecture);state minimization, state assignment, one-hot coding (1 lecture);counters, shift registers (1 lecture);datapath and control design (3 lectures);memory technologies (1 lecture);memory hierarchy, average access time, caches (1 lecture);caches (1 lecture); miscellaneous topics(1 lecture)

## Class/laboratory schedule (sessions per week and duration of each session):

Two 75-minute lectures per week

## Contribution of course to meeting the requirements of Criterion 5 - other:

## Contribution of course to meeting the requirements of Criterion 5 - math and basic sciences:

## Contribution of course to meeting the requirements of Criterion 5 - engineering topics:

3 hours

## Contribution of course to meeting the requirements of Criterion 5 - general education:

## Relationship of this course to program learning outcomes:

Learning Outcome	Level of Instruction	Related Course Content
Outcome A	Major	Students apply Boolean algebra, switching functions, combinational and sequential logic building blocks,

**Relationship of this course to program learning outcomes:**

<b>Learning Outcome</b>	<b>Level of Instruction</b>	<b>Related Course Content</b>
Outcome B	Intermediate	<p>combinational- and sequential-circuit analysis, combinational- and sequential-circuit synthesis, logic minimization, and hardware description languages, to describe problems and design solutions.</p> <p>Students design, construct, and debug digital logic circuits on a breadboard. Students experiment with these physical implementations.</p> <p>Students design, construct, and test digital logic circuits to implement required behavior (e.g., vending machine controller, elevator controller, etc.) under real-world cost, size, power, and manufacturability constraints, such as minimizing chip count to fit a complete circuit on a breadboard.</p>
Outcome C	Intermediate	
Outcome D	N/A	
Outcome E	Basic	Students formulate problems in terms of datapath and control.
Outcome F	N/A	
Outcome G	Basic	Students write reports for their breadboard projects.
Outcome H	N/A	

**Relationship of this course to program learning outcomes:**

<b>Learning Outcome</b>	<b>Level of Instruction</b>	<b>Related Course Content</b>
Outcome I	N/A	
Outcome J	N/A	
Outcome K	Major	Students represent engineering problems with formalisms of combinational and sequential logic. Students use contemporary hardware description languages (HDL), hardware simulators, and hardware synthesis and minimization tools to describe, verify, and optimize digital logic designs.

**Person who last prepared this description and date of preparation:**

- Ozturk, Hatice Orun (hoo) - Apr 1st, 2010 (09:28pm)