

ECE2020 Course Syllabus

ECE2020

Fundamentals of Digital System Design (3-0-0-3)

CMPE Degree

This course is Required for the CMPE degree.

EE Degree

This course is Required for the EE degree.

Lab Hours

0 supervised lab hours and 0 unsupervised lab hours

Course Coordinator

Keezer, David C

Prerequisites

CS 1371 [min C] or CS 1171 or CS 1301 [min C]

Corequisites

None

Catalog Description

Computer system and digital design principles. Switch and gate design, Boolean algebra, number systems, arithmetic, storage elements. Datapath, memory organization. Instruction set architecture, assembly language.

Textbook(s)

Wakerly, John, *Digital Design: Principles and Practice* (5th edition), Prentice Hall, 2017. ISBN 9780134460093 (required)

Wills & Wills, *Digital Computer Systems*, 2011. (required) (comment: See this site for electronic book: <http://www.ece.gatech.edu/academic/courses/ece2020/ComputerDesign.pdf>)

myDAQ unit, National Instruments. (required) (comment: This is required for purchase for ECE students. Non-ECE students will be loaned one through the course. This item is also required for ECE 2040 and ECE 3084)

Course Outcomes

Upon successful completion of this course, students should be able to:

1. Realize a Boolean expression with electronic switches and DeMorgan equivalent gates using a basic schematic design and simulation environment.
2. Simplify Boolean expressions in product-of-sums (POS) and sum-of-products (SOP) forms using Karnaugh Maps.
3. Implement encoders, decoders and multiplexers using basic gates (including pass gates).
4. Use basic design tools to analyze first-order combinatorial logic delay and energy dissipation.

5. Express signed, unsigned, fixed-point and floating-point values in binary representations.
6. Perform signed and unsigned addition and subtraction, observing errors.
7. Implement latches and registers and use in state machines and register files.
8. Implement toggle cells for designing divide-by-N counters.
9. Analyze sequential logic timing and power consumption using different tools.
10. Design storage systems using memory arrays composed of basic static and dynamic cells.
11. Implement a datapath using register files, functional units, and a three bus architecture.
12. Describe the instruction types and formats of a RISC instruction set.
13. Implement instruction sequences in a datapath using a RISC instruction set.
14. Describe the operation of a branching, single-cycle controller.

Student Outcomes

In the parentheses for each Student Outcome:

"P" for primary indicates the outcome is a major focus of the entire course.

"M" for moderate indicates the outcome is the focus of at least one component of the course, but not majority of course material.

"LN" for "little to none" indicates that the course does not contribute significantly to this outcome.

1. (P) An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics
2. (LN) 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
3. (LN) An ability to communicate effectively with a range of audiences
4. (LN) 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. (LN) 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. (LN) An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions
7. (M) An ability to acquire and apply new knowledge as needed, using appropriate learning strategies.

Topical Outline

Course description: Computer system and digital design principles. ECE 2020 introduces the many levels of abstraction that enable today's digital computing systems. It explores design at the layers of a computing platform from switches and wire to a programmable machine. At each layer, the design process of transforming a specification into an implementation is introduced and practiced. Design tools are used to build, evaluate and compare implementation approaches.

1. Introduction to Computing Systems
2. Switch Design
3. Boolean Algebra
4. Gate Design
5. Simplification
6. Building Blocks
7. Number Systems
8. Arithmetic
9. Latches and Registers
10. Counters

11. State Machines
12. Memory
13. Datapaths
14. Introductory Assembly Programming