

Course Syllabus

Course Information (Winter 2023)

For specific questions related to your work/grade and to schedule office hours appointments, please email eeecs31@eeecs.uci.edu. This will send an email to the entire instruction team (instructor & TA's), so we can answer your question as quickly as possible.

Code	Instructor	Time	Location	Final
	Quoc-Viet Dang qpdang@uci.edu	TBD	TBD / Zoom	TBD
	Quoc-Viet Dang	Asynchronous / TuTh 9:30am - 10:50am	TBD / Zoom	TBD

Code	Instructor	Office Hours	Location	Notes
16420	Dang, Quoc-Viet	Fridays 10am-10:50am	Zoom	

Course Objective

The goal of this course is to learn the basic principles of digital design. The course aims at enabling a student to design small digital systems for different application starting from abstract executable specifications and referring it to a manufacturable design for the given library of components.

Course Textbook(s)

Any of the following (one is mandatory; we will primarily be using Frank Vahid's book) are okay for this course:

Frank Vahid, [Digital Design with RTL Design, VHDL, and Verilog](#), SECOND EDITION, John Wiley, 2010. *First Edition also okay. Referred to as the “Blue Book” in class. *Official course book*.

Sarah Harris and David Harris, [Digital Design and Computer Architecture: ARM Edition](#), FIRST EDITION, Morgan Kaufmann, 2015. Referred to as the “ARM Book” in class.

D. D. Gajski, [Principles of Digital Design](#), Prentice Hall, 1997.

Lecture Format

The lectures are based on videos and slides as well as board work. The lectures will be mostly based on the material covered in the specified textbook. However, there may be

some additional material covered. In order to come to grips with the course, it is essential that you view videos, attend the lectures and ask questions in the class. It is the responsibility of the students to note down any additional material covered in lectures and discussed in class. It is in the best interest of the student to come to class after viewing the relevant videos and reading the relevant chapter(s) of the book. **The lectures will be fast paced.**

Homework

The student will be given homework for practice at home. Discussion sections will demonstrate procedure and results for some of the homework problems.

Homework Grading Policy

1. Late submissions get 0 points.
2. If the deadline for homework is 6:50pm, it means 6:50:00pm. Submitting at 6:50:01pm is considered late.
3. If you submit an unreadable file (Canvas/Gradescope won't load it) or blank file, you may not get any credit. You can check your own submissions.
4. If you submit in the wrong folder, you may not get any credit.
5. Extenuating circumstances are handled on a case-by-case basis
 1. an example of an extenuating circumstance that is excused:
 - being sick & having a doctor's note to verify that you have been sick during the time period of the homework/assignment
 2. an example of an unfortunate circumstance that is not excused:
 - falling off your skateboard at 6:45pm and scraping your knee; you couldn't submit your homework because you had to clean up. You should not be waiting until the last minute to submit your homework.

Evaluation Strategy

The evaluation will be based on the final exam (~40%), two midterms (~20% each; one midterm during Summer Session), and homework/participation (~20% total). Some adjustments may be made depending on the overall progress of the class as a whole. Alternative test arrangements can only be made under extenuating circumstances (i.e. doctor's note, etc). Final grade*: this class mostly follows the default UCI grading scale. *specific -'s & +'s will be adjusted during final grading depending on overall performance of the class.

EEE Grading Scale (UCI)

Name:	Range:
A	100% to 93.5%
A-	< 93.5% to 90.0%

EEE Grading Scale (UCI)

Name:		Range:
B+	< 90.0%	to 86.5%
B	< 86.5%	to 83.5%
B-	< 83.5%	to 80.0%
C+	< 80.0%	to 76.5%
C	< 76.5%	to 73.5%
C-	< 73.5%	to 70.0%
D+	< 70.0%	to 66.5%
D	< 66.5%	to 63.5%
D-	< 63.5%	to 60.0%
F	< 60.0%	to 0.0%

Academic Integrity Policy

If you are not already familiar with the department's cheating and plagiarism policies and procedures, please visit the [UCI Office of Academic Integrity and Student Conduct](#), available from the Engineering/UCI Undergraduate Student Affairs Office.

Add and Drop Policy

Adding students will depend on class size and TA availability. No adds will be signed after the end of the first week of instruction. Drops may be permitted until the end of the second week of class. Add/Drop cards will be signed by the instructor in the class.

Tentative Lecture Schedule

Week 1: Introduction to digital systems, binary numbers and arithmetic.

In this week students will learn relationships between software and hardware and different levels of abstraction for representing design. They will also learn about representing numbers in binary representation and how to perform arithmetic with binary numbers.

Week 2: Boolean Algebra and Boolean Functions.

This is a refresher on knowledge obtained in previous classes with emphasis on Boolean algebra axioms and theorems as well as different forms for describing Boolean functions.

Week 3: Boolean Simplifications and Logic Gates.

Basic techniques for Boolean optimization for performance cost and power. Also in this week students will learn different cost functions for optimization.

Week 4: Digital Technology and Technology Optimizations.

In this week students will learn how to manipulate and optimize Boolean functions for different libraries of logic gates and different technologies.

Week 5: Design of RTL Combinatorial Components: ALU, Shifters, Comparator, etc.

These lectures teach students how to implement basic programming-language operations such as addition, multiplication, comparison and other more complex functions.

Week 6: Sequential Design.

In this week student learn basic concepts and components of information storage such as latches and flip-flops and how to specify storage components using finite-state-machine(FSM) model.

Week 7: FSM Model, Analysis and Synthesis Techniques for Sequential and Storage Components.

In this week student learn basic concepts and components of information storage and techniques for design and optimization of components described by FSM models.

Week 8: Design of RTL Storage Components: Register, Memories, Register Files, RAMs, FIFOs, Stacks, etc.

This week students learn how to design component for storing variables and perform simple sequential operations. These lectures are geared toward learning organization and design of storage components intended for storing a variety of data structures found in programming languages.

Week 9: From C to RTL.

Methodology for converting executables to RTL. Step by step explanation of basic tasks of designing datapath and controller components.

Week 10: Processor Organization and Design.

Design principles for custom hardware units executing fixed code that is specified with FSM models. Also, techniques for design of standard processors from a given instruction-set (IS). Discussion of IS choice on performance of processor design.