

## EE 260 Introduction to Digital Design

**Credits:** 4

**Categorization of credits:** engineering topic

**Instructor(s):** Christopher Manloloayo. Yingfei Dong, Revised Jan. 15th, 2021.

**Textbook and Other Required Materials:** Springer started a free Textbook initiative. We are using the following textbooks

- [Introduction to Logic Circuits & Logic Design with Verilog](#)
- [Quick Start Guide to Verilog](#)
- [Quick Start Guide to VHDL](#)

**Supplemental Textbooks** (Not Required): Digital Design, Principles & Practices. by John F. Wakerly, 5th Edition

**Designation:** Required for CENG.

**Catalog Description:** Introduction to the design of digital systems with an emphasis on design methods and the implementation and use of fundamental digital components.

**Pre- and Co-requisites:** Pre-requisites: EE 160 Programming for Engineers or consent.

**Class/Lab Schedule:** three 1-hour lectures and one 3-hour lab session per week.

**Topics Covered:** The lecture component of the course covers the following topics:

- State diagrams and ASM charts (3 hrs)
- Programmable logic such as PROMs, PALs, and PLAs, (2.5 hrs)
- Conversion between ASM chart (or state diagram) to sequential circuits (5 hrs)
- Conversion between logic diagrams and combinational circuits made up of logic gates such as NANDs, NORs, voltage inverters. Mixed logic. (2 hrs)
- Boolean algebra, K-maps (5 hrs)
- Representations using bits: binary numbers, twos complement, sign magnitude, ASCII (1.5 hr)
- Simple arithmetic including addition, subtraction, negation, bit-wise logic operations, carry, borrow, overflow. (1.5 hr)
- Implementation of circuits with simple switches, modeling CMOS. (2 hrs)
- Commonly used medium sized integrated (MSI) circuits including multiplexers, demultiplexers, counters, registers, static RAM. (3.5 hrs)
- Modular design techniques including iterative partitioning, hierarchical partitioning, functional partitioning, busses. (3 hrs)
- A simple computer, its architecture, and how to program it. (7 hrs)
- Latches, flip flops, timing, and glitches. (4 hrs)

The laboratory component covers the following topics:

- *Assignments 0, 1, and 2:* Introduction to lab procedures and CAD tools (3 weeks)

- *Assignments 3 and 4:* Logic probes and programming EPROMS: Introduction to circuit parts, protoboards, and EPROM programming. Measuring behavior of gates. (2 weeks)
- *Assignment 5 and 6:* Combinational circuits: Simple combinational circuit design given boolean expression and a word problem. Use of K-maps for optimization. Simple sequential circuit design. (2 weeks)
- *Assignment 7:* Sequential circuit design (scrolling door sign for the professor). (2 weeks)
- *Assignment 8:* Moderate sized sequential circuit design (electronic racketball). Modular and hierarchical design. (4 weeks)
- *Assignment 9:* If time permits, another design assignment. Examples include programming a RAM using serial to parallel conversion or an application of microcontrollers, e.g., PIC microcontroller (2 weeks)

All assignments involve using a computer aided design (CAD) tool (e.g., Logicworks) to design and simulate circuits, and then the design is implemented using discrete parts on a protoboard.

**Course Objectives and Relationship to Program Objectives:** This is an introductory course to digital circuits. The objective of the course is to understand and apply the fundamental underpinnings and practice of digital circuit design. Design tools, techniques, and methodologies should be mastered so that small to moderate sized circuits can be designed, debugged, and tested. [The following Program Objectives are addressed: 1, 2, 3, 4, 5.]

#### **Course Outcomes and Their Relationship to Program Outcomes**

The following are the course outcomes. The Program Outcomes (numbered 1-8) that they address are in square braces (“[ ]”).

- Understand state machines and how to describe them using formalisms such as state diagrams and ASM charts. [1,7]
- Know how to design, build (with circuit parts on a protoboard), test, and debug a digital circuit from formal descriptions including state diagrams, ASM charts, truth tables, Boolean expressions. [1,2,7]
- Understand modern digital circuit technologies, including logic gates, programmable logic, medium scale integrated (MSI) circuits, CMOS, and how to apply them. [1,2]
- Know the mathematical underpinnings of digital circuits including number representations, arithmetic, logic operations, Boolean algebra, algorithms for arithmetic computation, and K-maps for optimization. [1,7]
- Understand how digital circuits are built from transistors, resistors, and capacitors. [1]
- Know how to apply modular and hierarchical design. [1,2]
- Understand important timing concepts and issues in digital circuits. [7]
- Learn CAD tools for simulation and optimization as part of the design process. [2,7]
- Understand a simple computer: its architecture, the machine instructions it executes, and how to program it. [1]
- Understand and practice the process of digital circuit design, from word problem, to formal description of a system, to implementation, testing, and debugging. [1,2,7]
- Know how to work in a team (Laboratory assignments require group work). [5]
- 

**Computer Usage:** Students use computer aided design tools: LogicWorks (circuit design and simulation software tool), truth\_table software tool to program EPROMs, and word processing for

lab reports. The students use LogicWorks for most of their lab experiments and around 20-25% of their homework.

**Design Credits and Features:** There are two design credits due to (i) a laboratory component, where 75% is on design and 25% is on learning equipment, circuits, and tools; and (ii) a lecture component, where 30% of the assignments are design problems, techniques and methodologies.