CE100/L: Digital Design  

Extended Course Description

Catalog copy

CE100. Logic Design. F,W,S Boolean algebra, logic minimization, finite state machine design, sequential circuits, common logic elements, programmable logic devices, and an introduction to system design. Simplified electrical behavior of circuits including three state outputs, propagation delay, logic levels and fanout. Prerequisite: Course 12. J. Ferguson

CE100L. Logic Design Laboratory (1 credit). F,W,S Laboratory sequence illustrating topics covered in course 100. One two-hour laboratory session per week. Weekly laboratory assignments that require the use of the oscilloscope, TTL circuits, computer-aided design and simulation tools, and programmable logic. Prerequisite: course 12; previous or concurrent enrollment in course 100 is required. J. Ferguson.

Explanation of prerequisites

CMPE12: students must be familiar with and be able to reason about discrete logic with an introduction to Boolean logic. They must be able to manipulate logic expressions, understand truth tables, and be familiar with number representations.

Required Skills to pass the course

1. Minimize logic using Boolean Algebra (taught in CMPE16)
3. Use elementary logic blocks such as multiplexers, decoders and flip-flops.
4. Translate a problem description to a state table or state diagram.
5. Realize a finite state machine from a state table or state diagram, using elementary gates and D flip-flops.
6. Read and draw schematic diagrams of logic circuits fluently, with good understanding of the inverter-bubble convention.
7. Read timing diagrams.
8. Understand concepts of propagation delay and fanout
9. Understand cMOS logic circuits at the switch level.
10. Design, breadboard, and debug combinational and sequential circuits.
11. Maintain a thorough lab notebook while working in the lab.
12. Explain a lab in a professional lab report.

Core topics (must be taught)

1. Limitations of physical gates including fanout and logic levels.
2. Two-level logic minimization using Karnaugh Maps.
3. Use of elementary logic blocks such as multiplexers and flip-flops.
4. Translate a problem description to a state table or state diagram.
5. Realize a finite state machine from a state table or state diagram, using elementary gates and D flip-flops.
6. Design, function and limitations of latches and flip-flops, including set-up and hold times, asynchronous and synchronous inputs, advantages and disadvantages of latches and flip-flops.
7. Use of flip-flops or latches as synchronizers to interface asynchronous inputs to a synchronous system.
8. Design of counters
9. Reading timing diagrams.
10. Tristate gates and how they are used.
11. Programmable logic.
12. Use of software for simulation and design.

Optional topics.

1. Asynchronous finite state machine analysis and design.
2. Design of adders
3. Design of multipliers
4. Quine-McCluskey algorithm for logic minimization.
5. More detailed electrical explanations, including TTL and open-collector circuits.

Core Lab Exercises

1. Use of design tools to design, build, and document circuits.
2. Design, breadboard, debug, and document combinational logic circuits.
3. Use of programmable logic.

Optional Lab Exercises

1. Design sequential systems with multiple, interacting finite-state machines.
2. Use an digital-to-analog converter (required in CMPE121)