

Syllabus: ECE 2020 Fundamentals of Digital Design

Course Objective

Course description: 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.

Prerequisite

CS 1371 or CS 1171

Text Book

1. Digital Design: Principles and Practices, John Wakerly, Pearson Press

Grading

Grading will be based on homework (4%), 3 midterms (20%), in-class lab (6%), and final exam (30%). Check Canvas frequently to obtain up-to-date homework grades.

Late Submission

15% penalty per day is applied to late homeworks. No submission is allowed after the solutions are posted.

In-class Lab

There are two in-class labs offered in each semester. The goal is to build simple combinational and sequential circuits using software on a laptop. The first lab is on mixed logic method, and the second on finite state machine. Students will be provided with lab writeups and pre-lab assignments prior to the lab dates.

Make-up Exams

Students are responsible for all material covered in class, including changes in exam schedules announced in class. Make-up exams will be considered only if a student misses an exam due to medical reasons (***medical certificate will be required***) or if he/she misses an exam for extenuating reasons. The make up test for the latter must be arranged in consultation with the instructor ***prior to the date of the in-class exam***. Make-up exams are not guaranteed to be the same as the exam given in class.

Academic Honesty

Although students are encouraged strongly to work together to learn the course material, all students are expected to complete homeworks, labs and exams individually, following all instructions stated in conjunction with the exam. Homework and laboratory material may be discussed between students, but the problems and writeups must be completed individually. Assignments which are substantially identical will be treated as violations of the honor code. All conduct in this course will be governed by the Georgia Tech honor code. Additionally, it is expected that students will respect their peers and the instructor such that no one takes unfair advantage of anyone else associated with the course. Any suspected cases of dishonesty will be handled as per Georgia Tech's policies.

Topical Outline

Introduction to Computing Systems

- building complex systems out of simple elements; examples in today's products; architecture block diagram

Switch Design

- behavior vs. implementation truth tables; switch combinations: series and parallel; semiconductor switches: n type & p-type;
- implementing logical functions; implementing basic gates; introduction to VLSI technology

Boolean Algebra

- Boolean expressions & algebra; DeMorgan's square & DeMorgan's theorem; standard forms: SOP/POS using min/max terms

Gate Design

- designing with gates vs. switches; decoupling behavior and implementation using mixed logic; implementing SOP and POS

- expressions; gate delay (including RC) and energy dissipation; pass gates and floating outputs

Simplification

- expression simplification; 2, 3 and 4 variable Karnaugh maps; negative logic and don't cares

Building Blocks

- powers of two, working with binary; encoders/decoders; pass gates and tri-state outputs; multiplexers/demultiplexers;
- programmable logic arrays

Number Systems

- notations: decimal, binary, hexadecimal; representations: unsigned vs. two's complement; representations: integer, fixed
- point and floating point; symbolic representations

Arithmetic

- addition and subtraction; ranges and resolutions; error and overflows; adder/subtractor implementation

Latches and Registers

- combinatorial vs. sequential logic; bistable element using basic gates; RS latch and transparent latch; shift register and
- register; two-phase non-overlapping clocking; edge vs. level triggering; read/write enables; energy and power in a clocked system

Counters

- basic toggle cell operation; building binary counters; building divide-by-N counters; cascading multi-digit counters

State Machines

- state machine operation; transition diagrams and tables; state machine implementation: Moore and Mealy; state machine operation in behavioral HDL; in-class mini-lab: state machine

Memory

- memory cell behavior and protocol; static random access memory (SRAM) cell; dynamic RAM (DRAM) cell; memory chip
- organization; building memory systems; bit, byte and word addressing; alignment, byte order

Datapaths

- operands: register file and immediate values; three bus architecture; execution units: arithmetic, logical, shift; memory interface

Introductory Assembly Programming

- basic computer organization; instruction formats; datapath operations: arithmetic, logical, shift, memory; conditional
- execution (if-then-else); basic loops (while)

GRADING:

TEST 1: 20%

TEST 2: 20%

TEST3: 20%

LAB: 6%

HW: 4%

FINAL: 30% (mandatory)

Dates for tests will be announced at least one week in advance.

GTAs: TBD