Instructor: Henry Owen  
E-Mail: owen@gatech.edu

**Course Objective:** 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 wires to programmable machines. At each layer, the design process of transforming a specification into an implementation is introduced and practiced. Completion of this course will enable students to understand fundamentally how computers work and are applied to domains such as robotics and smart phones.

**Pre-requisites:** CS1371 Computing for Engineers or CS1171 Computing in Matlab or CS1301 Intro to Computing

**Text:** Class Notes and the web site [http://www.ece.gatech.edu/academic/courses/ece2020/index.html](http://www.ece.gatech.edu/academic/courses/ece2020/index.html)

**Class Web Site:** Lecture notes, assignments, solutions, etc. will be posted on T-Square: [https://t-square.gatech.edu/portal](https://t-square.gatech.edu/portal)

**Grading:**

- Class Attendance and Class participation: 20%
- Weekly mini-Quizzes: 50%
- Final Exam: 30%

**Quizzes:** The quizzes will be closed book, closed notes.

**Attendance:** Students are responsible for all material covered in class, including changes in exam schedules announced in class. Make-up quizzes will be considered only if the student informs the instructor with a valid excuse prior to the exam.

**Homework:** Practice problems will be assigned on a regular basis. If you do not regularly work the assigned practice problems you cannot expect to do well on the exams. Practice problems will not be graded.

**Academic Honesty:** Although students are encouraged to work together to learn the course material, all students are expected to complete quizzes and exams individually, following all instructions stated in conjunction with the exam. All conduct in this course will be governed by the Georgia Tech honor code. Any suspected cases of academic dishonesty will be reported to the Dean of Students for further action. A copy of the Georgia Tech Honor Code can be found at [http://honor.gatech.edu/content/2/the-honor-code](http://honor.gatech.edu/content/2/the-honor-code) and the Georgia Tech student code of conduct at [http://www.policylibrary.gatech.edu/student-affairs/code-conduct](http://www.policylibrary.gatech.edu/student-affairs/code-conduct).
Topical Outline

1. Introduction to Computing Systems
   a. building complex systems out of simple elements
   b. examples in today's products
   c. architecture block diagram

2. Switch Design
   a. behavior vs. implementation truth tables
   b. switch combinations: series and parallel
   c. semiconductor switches: n-type & p-type
   d. implementing logical functions
   e. implementing basic gates
   f. introduction to VLSI technology

3. Boolean Algebra
   a. Boolean expressions & algebra
   b. DeMorgan's square & DeMorgan's theorem
   c. standard forms: SOP/POS using min/max terms

4. Gate Design
   a. designing with gates vs. switches
   b. decoupling behavior and implementation using mixed logic
   c. implementing SOP and POS expressions
   d. gate delay and energy dissipation
   e. pass gates and floating outputs

5. Simplification
   a. expression simplification
   b. 2, 3 and 4 variable Karnaugh maps
   c. negative logic and don't cares

6. Building Blocks
   a. powers of two, working with binary
   b. encoders/decoders
   c. pass gates and tri-state outputs
   d. multiplexers/demultiplexers
   e. programmable logic arrays

7. Number Systems
   a. notations: decimal, binary, hexadecimal
   b. representations: unsigned vs. two's complement
   c. representations: integer, fixed point and floating point
   d. symbolic representations

8. Arithmetic
   a. addition and subtraction
   b. ranges and resolutions
   c. error and overflows
   d. adder/subtractor implementation
9. Latches and Registers
   a. combinatorial vs. sequential logic
   b. bistable element using basic gates
   c. RS latch and transparent latch
   d. shift register and register
   e. two-phase non-overlapping clocking
   f. edge vs. level triggering; read/write enables
   g. energy and power in a clocked system

10. Counters
    a. basic toggle cell operation
    b. building binary counters
    c. building divide-by-N counters
    d. cascading multi-digit counters

11. State Machines
    a. state machine operation
    b. transition diagrams and tables
    c. state machine implementation: Moore and Mealy
    d. state machine operation in behavioral HDL
    e. in-class mini-lab: state machine

12. Memory
    a. memory cell behavior and protocol
    b. static random access memory (SRAM) cell
    c. dynamic RAM (DRAM) cell
    d. memory chip organization
    e. building memory systems
    f. bit, byte and word addressing
    g. alignment, byte order

13. Datapaths
    a. operands: register file and immediate values
    b. three bus architecture
    c. execution units: arithmetic, logical, shift
    d. memory interface

14. Introductory Assembly Programming
    a. basic computer organization
    b. instruction formats
    c. datapath operations: arithmetic, logical, shift, memory
    d. conditional execution (if-then-else)
    e. basic loops (while)