The summer team

- Instructor: Alexandra Carey
  - fire@soe.ucsc.edu
- Tutor: To be announced in lab

- Classes meet here (Physical Sciences 136)
- Labs meet in BE 150 (for now)

Our online presence

- Our online presence...
  - On the Web
    - http://www.soe.ucsc.edu/classes/cmpe012/Summer08
  - Forum
    - http://forums.soe.ucsc.edu
  - Google calendar...

What we will cover in this class

Part 1: Introduction to Logic Design
- Brief History of Computers (Notes)
- Logic Gates and Functions (Ch 3)
- Integer Number Representation (Ch 2)
- Computing Systems & Abstraction (Ch 1)
What we will cover in this class

Part 2: The LC-3 computer system
- LC-3 Architecture (Ch 4)
- LC-3 Instruction Set Architecture (Ch 5)
- LC-3 Machine Language Programming (Ch 6)
- LC-3 Assembly Language Programming (Ch 7)
- LC-3 Input and Output (Ch 8)
- LC-3 TRAPS and subroutines (Ch 9)
- LC-3 Stack (Ch 10)

Part 3: The 68HC11 microcontroller (not in book)
- Microcontrollers and embedded systems
- HC11 microcontroller
- HC11 assembly
- HC11 I/O and interrupts

Part 4: Fractional numbers
- Fixed and floating-point numbers and arithmetic

What we will NOT cover in this class

Part 5: The C programming language
- Intro to the C programming language (Ch 11)
- C variables (Ch 12)
- C control structures (Ch 13)
- C functions (Ch 14)
- C pointers and arrays (Ch 15, 16)
- C recursive functions (Ch 17)

BUT we will use C as our reference high-level programming language, with examples throughout

Extended course description

Required skills to pass the course.

1. Number representations, including
   a. arbitrary base conversion  b. binary, hex, decimal, 2's C
   c. bitwise operators    d. Binary fixed point numbers
   e. single-precision floating-point format

2. Binary Arithmetic, including
   a. Signed magnitude add/sub  b. Unsigned add/sub/mul
   c. Two's compliment add/sub/mul  d. IEEE floating point add/sub/mul

3. Computing Systems
   a. Basic logic gates (and, or, not, xor)
   b. Determining the function of simple combinational circuits
   c. Adder and mux logic blocks

4. Assembly language programming
   a. Arithmetic and bitwise operations  b. Procedure calls
   c. Stack & memory operations  d. Assembly implementation of C control structures

5. An understanding of acceptable and unacceptable collaboration, the need to ensure permission to collaborate in a class, and an automatic urge to acknowledge collaborators and others who have assisted in a project.
Extended course description

Core topics:
1. Assembly language programming including
   a. Arithmetic and bitwise operations
   b. Arrays, stacks
   c. Procedure calls
   d. Addressing modes
   e. Both CISC and RISC architectures
2. An understanding of basic computing systems including
   a. Basic logic gates and/or/xor/not
   b. Basic logic blocks (adder, mux)
   c. Registers, memory, CPU, I/O
   d. Steps to execute an instruction
   e. Data structures
3. Binary arithmetic
   a. Signed magnitude add/sub
   b. Two's compliment add/sub/mul
   c. Floating point add/sub/mul
4. Number representations, including
   a. Arbitrary base conversion
   b. Binary, hex, decimal, 2s C
   c. Bitwise operators
   d. Binary fixed point numbers
   e. Arbitrary bases (e.g., 3, 60)
   f. Biased representation
   g. IEEE Floating point format
5. An understanding of basic system software including
   a. Assembly and compilation
   b. Loading and linking
   c. The basic functions of the operating system
6. Interrupts and I/O
   a. Causes of interrupts
   b. Interrupt service routines
   c. Memory mapped I/O

The textbook

Patt, Patel

Introduction to Computing Systems: From Bits and Gates to C and Beyond

McGraw-Hill, Second Ed.

Course work

- Class (CMPE12) Requirements
  - Attending lectures
  - Doing the weekly homework
    * Posted online, collected in class
    * Optional homework from the book (not graded)
- Lab (CMPE12L) Requirements
  - Going to your lab section
  - Weekly lab assignments
    * Posted online, submitted online

Lab work

- Part 1: Logic design with Multimedia Logic
  - Simulated logic gates
- Part 2: Programming assignments in LC-3
  - Simulated architecture
- Part 3: Programming assignments in HC11
  - Real hardware kit
Lab rules

- Each lab assignment consists of two things
  - Lab work (code, design file, etc.)
  - Lab write-up
- Lab assignment score = code + write-up
- Assignments are turned in using the UNIX `submit` command
- Must submit both the lab code and the write-up
- Make sure to make the deadline, since even 1 second late will stamp your assignment as…

LATE

Lab rules

- On time labs
  - Eligible for 100% of points, including extra credit
  - Eligible for 100% of points on the write-up
- Late labs
  - Lose all extra credit
  - Every week late, max score decreases by 25%
- Ex: Lab worth 20 points + 5 extra credit + 12 write-up
  - 1 late week: 0 extra credit, max points are 15 and 9
  - 2 late weeks: 0 extra credit, max points are 10 and 6
  - 3 late weeks: 0 extra credit, max points are 5 and 3
  - 4 late weeks: 0 extra credit, max points are 0 and 0
  - Never turned in: -1 and -1

Done early? Get a check-off

- Go through the grading checklist in lab with your tutor in person
- Lock in your score with your tutor
  - Your tutor only
- Do the write-up
- Submit the lab assignment by the deadline
  - Both the lab file(s) and the write-up file
  - Submission into the locker is what really counts
- No difference in grading between in-person check-off and normal grading with file submission
- Only the submitted files will be considered “official,” regardless of what you showed in class

No time? Submit it late

- Lab re-submission is not allowed
- Make sure you like what you submit
- If you are not done in time, don’t submit, complete the lab, and just submit late
**Extreme programming**

- New-ish trend in the industry
- Program as a team, together at one terminal
- Reduces typos and bugs
- Improves morale and effectiveness
- Check the class website for papers
- You can try it out in some of the labs (if you want)

**Attendance**

- Mandatory for the class
  - How will I know if you cut?
- Mandatory for the lab
  - In lab, we may take roll
- Attending both lab sections are required
  - First and second day each week

**Grades for CMPE12 (the class)**

- 20% Weekly homework and pop quizzes
  - Lowest homework is dropped
- 20% Midterm 1
- 20% Midterm 2
- 40% Comprehensive final

- Things you don’t turn in score -1
- Cannot make up missed exams
- No curve

**Grades for CMPE12L (the lab)**

- You must turn in all lab assignments (none are dropped from the final grade)
- All grades for code and write-up add up to 100%

- Things you don’t turn in score -1
- Make up missed labs with late penalty
- No curve
Special needs

- Students with special needs should refer to the Disability Resource Center
- We will accommodate your needs

Final Grades Breakdown

<table>
<thead>
<tr>
<th>Grade</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>95.0 and up</td>
<td>A+</td>
</tr>
<tr>
<td>90.0 to 95</td>
<td>A</td>
</tr>
<tr>
<td>85.0 to 90</td>
<td>A-</td>
</tr>
<tr>
<td>80.0 to 85</td>
<td>B+</td>
</tr>
<tr>
<td>75.0 to 80</td>
<td>B</td>
</tr>
<tr>
<td>70.0 to 75</td>
<td>B-</td>
</tr>
<tr>
<td>65.0 to 70</td>
<td>C+</td>
</tr>
<tr>
<td>60.0 to 65</td>
<td>C</td>
</tr>
<tr>
<td>40.0 to 60</td>
<td>D</td>
</tr>
<tr>
<td>Under 40</td>
<td>F</td>
</tr>
</tbody>
</table>

Lab/Class Pass Rule

- The class and lab are fully-coupled
  - You must pass both the class and the lab to receive a passing score
- You must pass the class to pass the lab
- You must pass the lab to pass the class

Academic Dishonesty (Cheating)

- What is cheating?
- How does it happen?
- What happens if I get caught?
Homework from this section

- Check out our Web page
  - http://www.soe.ucsc.edu/classes/cmpe012/Summer08/
- Register on the forums
  - http://forums.soe.ucsc.edu
- Read Chapter 3 of the textbook

DO NOT CHEAT!