Welcome to CMPE 12
Computing Systems and
Assembly Language Programming
Fall 2011
Joel Ferguson

Based on slides from
Cyrus Bazeghi and Andrea Di Blas
The team (syllabus)

Instructor: F. Joel Ferguson  fjf@ucsc
Office Hours: Mon 3:30 to 4:30 Crown Provost office and by appointment

TAs: Aimen Al-Refai  aalrefai@soe.ucsc
     Christopher Hoseit  cphoseit@soe.ucsc
     Seokjoong Kim  seokjkim@soe.ucsc

Undergraduate Lab Tutors

MSI tutor: Evan Wegley  ewegley@ucsc
Outline of CMPE12

• Beginning logic design (and circuits)
• Computer organization
• Assembly language programming
• C programming language
The website:
http://www.soe.ucsc.edu/classes/cmpe012/Fall11(under development)

The forum:
http://forums.soe.ucsc.edu/
Class schedule:

- Lectures: CLU1 MWF 2:00-3:10
- Labs
  - Section 01: Baskin 109 MW 9-11PM
  - Section 02: Baskin 109 TTh 8-10PM
  - Section 03: SS1 135 MF 9-11AM
  - Section 04: SS1 135 MW 6-8PM
Ordering of class material

Part 1: introduction and logic design

- Computing Systems and Abstraction (Ch 1)
- Number Bases and Arithmetic (Ch 2.1)
- Logic Functions and Gates (Ch 3)
- Boolean Algebra and logic minimization (NITB)
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)
  - Integers and Bit Operations (Ch 2.2-2.6)
- 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)
What we will cover in this class

Part 3: The C programming language
- Fixed- and floating-point numbers and arithmetic (Ch 2.6-2.7)
- 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)
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 complement 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 (must be taught)

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. Unsigned add/sub/mul/div
   c. Two's complement add/sub/mul
   d. Floating point add/sub/mul

4. Number representations, including
   a. Arbitrary base conversion
   b. Binary, hex, decimal, 2s Complement
   c. Bitwise operators
   d. Binary fixed point numbers
   e. Arbitrary bases (e.g., 3, 60)
   f. Biased representation
   g. IEEE Floating point format

5. HLL language - basic coverage of C.
   a. Syntax
   b. Data types
   c. Procedure calls
   d. Arithmetic/bitwise operations

6. An understanding of basic system software including
   a. Assembly and compilation
   b. Loading and linking
   c. The basic functions of the operating system

7. Interrupts and I/O
   a. Causes of interrupts
   b. Interrupt service routines
   c. Memory mapped I/O
Course Work and Grading

- Lectures
  - Quizzes and Participation (5%-10% extra credit)
- Weekly Homework but worst grade dropped (15% of grade)
  - Cannot be turned in late, why worst grade is dropped
- Midterms (25% apiece)
- Final Exam (35% of grade)
- Both homework (and solutions) and lab assignments will be posted on our web page every week
Homework

• Two parts:
  – Recommended exercises from the textbook, NOT GRADED!
  – GRADED homework will be posted on Thursday or Friday and due ON PAPER, in class, by the following Wednesday at beginning of class.
Lab Work

- Logic design with Multimedia Logic
- Programming assignments in LC-3 (using the simulator)
- Maybe another Assembly language
- Programming assignments in C

NOTE: Labs start MONDAY October 3rd!
Team Labs

Team Programming

- Program as a team, together at one terminal
- Reduces typos and bugs
- Improves moral and effectiveness
- Increases learning speed
- Develops engineering skills
- Must join/maintain team all quarter.

Just in: Change in syllabus – You must work as a team with a different partner each week.
Attendance

- Mandatory for both class and lab
- Use iclickers for in-class quizzes and feedback, please purchase
- Both lab sections are required until the lab is turned in
- You must pass the class to pass the lab, you can pass the class but fail the lab
Special needs

• If you qualify for classroom accommodations because of a disability, please get an Accommodation Authorization from the Disability Resource Center (DRC) and submit it to me in person outside of class (e.g., office hours) within the first two weeks of the quarter. Contact the DRC at 459-2089 (voice), 459-4806 (TTY), or http://drc.ucsc.edu for more information on the requirements and/or process.
Grades for CMPE12L

- You must turn in ALL lab assignments (working or not)
- Use **submit** to the course locker: **cmpe012-fjf.f11**
- All lab assignments have the same weight
- Read the web page about rules for late submissions and re-submissions
- Friday Lab time reserved as extra time for students in all sections.
Academic Integrity

- What is a violation? *Turning in work as your own, that is not your own.*

- Consequences
  - Academic Sanction: *Fail Class*
  - Disciplinary Sanction: *From warning to expulsion.*

See [http://www.ucsc.edu/academics/academic_integrity/undergraduate_students/](http://www.ucsc.edu/academics/academic_integrity/undergraduate_students/)
Academic Dishonesty

- Damages trust,
- Wastes time that can be spent teaching and learning,
- Erodes value of degree and other’s work, and
- Personal integrity is weakened for future temptations.
Assignments

- Check out the class website: www.soe.ucsc.edu/classes/cmpe012/Fall11
- Check out the textbook’s website: www.mhhe.com/patt2
- Check out the course discussion forum: http://forums.soe.ucsc.edu
- Read Chapters 1 and 2.1 of the textbook by Monday.