EECE.3170: Microprocessor Systems Design I

Summer 2017

Syllabus

Course Meetings

Section 011: MWTh 10:30 AM-12:50 PM, Falmouth 205

Course Website

Main site: http://mjgeiger.github.io/eece3170/sum17/

Schedule: http://mjgeiger.github.io/eece3170/sum17/schedule.htm

Course Discussion Group

All course announcements will be posted on the discussion group—you are responsible for checking the board regularly or enabling direct e-mail updates from Piazza.

Sign up link: https://piazza.com/uml/summer2017/eece3170

Instructor

Dr. Michael Geiger
E-mail: Michael_Geiger@uml.edu
Office: Perry 118A / Ball 301A *(moving during term)*
Phone: 978-934-3618 (x43618 on campus)
Office hours: TBD

During my office hours, student questions are my top priority. I am available by appointment at other times.

Feel free to stop by my office, e-mail me questions, or schedule a one-on-one appointment. Office hours are subject to change.

Textbook

No textbooks are required for this course. Relevant reading material will be posted on the course schedule page on a regular basis.

Course Overview

Description: This course provides an introduction to microprocessors. It uses assembly language to develop a foundation on the hardware, which executes a program. Memory and I/O interface design and programming. Study of microprocessor and its basic support components, including CPU architecture, memory interfaces and management, coprocessor interfaces, bus concepts, serial I/O devices, and interrupt control devices. Laboratories directly related to microprocessor functions and its interfaces. 3 credits.

Prerequisites: EECE.2160 (ECE Application Programming) and EECE.2650 (Logic Design)

Course Overview (cont.)

Course Objectives: By the end of this course, you should understand and/or be able to use all of the following:

1. **Microprocessor Software Architecture:** Data formats, types, and alignment. Memory addressing and organization. Stack operation.
2. **Microprocessor Instructions:** Instruction formats and types: data transfer, arithmetic, logical, shift/rotate, conditional execution, program control, subroutines.
3. **Assembly Language Programming:** Ability to write, modify, and debug programs written in assembly language. Translation of high-level code to assembly language. Programs that integrate assembly and high-level code.
4. **Microprocessor Interfacing:** Memory and I/O interfacing. Bus cycles.
5. **Interrupt Processing:** Hardware and software interrupts.
6. **Microcontroller-based Systems:** Microcontroller architecture and instruction set. Microcontroller programming using both assembly language and high-level code. Design and debug microcontroller-based circuits.

Grading: Grades will be computed on an A to F scale; no A+ grades will be assigned, in accordance with UMass Lowell policy. The weights assigned to the various items are:

Homework/lab assignments	55%
Exam 1	15%
Exam 2	15%
Final	15%

Incomplete grades will only be given in exceptional situations, and the student must be passing the class at the time the grade is requested.

The following rubric describes how grades will be assigned if no grading curve is applied. A grading curve may be used at the instructor's discretion, depending on the overall course average at the end of the term. Grades will not be curved down, meaning that the table below describes the minimum letter grade you will earn for a final average in each of the ranges shown:

Range	Grade	Range	Grade
> 92	A	78-79	C+
90-92	A-	73-77	C
88-89	B+	70-72	C-
83-87	B	68-69	D+
80-82	B-	60-67	D
		< 60	F

Your grade is based strictly on the work you do during the semester. Please do not ask if you can have extra credit work to improve your grade—the answer will be no.

Class participation: You are responsible for all material discussed or announced in class. You are expected to attend class regularly and participate in any in-class discussions, as such exercises are essential to your learning. Although lecture attendance is not explicitly required, regular attendance will improve your understanding of the course concepts.

Course Overview (cont.)

Exams: Make-up exams will only be offered in exceptional circumstances. You must notify Dr. Geiger as early as possible in order to determine an appropriate make-up date.

Assignment policies: Your assignments will be a mix of typical homework problems, programming assignments, and labs that involve both programming and hardware interfacing. All assignments will be posted on the course web page.

Assignment policies include the following:

- All assignments must be completed individually unless explicitly specified. You may be allowed to work in groups for some assignments.
- Late assignments are penalized at a rate of 10% per day.
- Some programming assignments may require an instructor to check off the completion of one or more milestones within the assignment.

Academic Honesty

All assignments and exams must be completed individually unless otherwise specified. You may discuss concepts or material covered in class, but may not share any details of your solutions to assigned problems, including algorithms and code. Plagiarism (copying solutions from an outside source) is also unacceptable and will be treated as an instance of cheating.

Students are allowed to discuss assignments in general terms and to help one another fix specific errors. In this case, students are required to note that they received assistance from a classmate by listing that person's name and the nature of their assistance as part of their lab report or homework solution.

Any assignment or portion of an assignment that violates this policy will receive a grade of zero for all parties concerned. Depending on the severity of the infraction, or in cases of repeat violations, additional penalties may be given at the instructor’s discretion, up to and including a failing grade in the course.

Further information on the university Academic Integrity policy can be found at:

http://www.uml.edu/Catalog/Undergraduate/Policies/Academic-Policies/Academic-Integrity.aspx

Course Schedule

This schedule contains a tentative schedule of topics we will cover throughout the term; the course website will contain the most up-to-date version. The web page will also describe suggested reading for each lecture. This schedule is subject to change.

The exams are currently scheduled as shown below. The first exam will be held on **Thursday, May 25 in class**, the second exam will be held on **Monday, June 12 in class**, and the third exam will be held **Monday, June 26 in class**.

Lecture	Date	Lecture Topics
1	M, 5/15	Course introduction; role of ISA Data types Data storage and addressing
2	W, 5/17	x86 introduction & assembly basics Data transfer instructions
3	Th, 5/18	Arithmetic instructions
4	M, 5/22	Logical instructions Shift and rotate instructions
5	W, 5/24	Bit test/scan instructions Exam 1 Preview
	Th, 5/25	**EXAM 1**
No Monday lecture—Memorial Day		
6	W, 5/31	Conditional execution Jump/loop instructions Exam 1 Review
7	Th, 6/1	Subroutines HLL and x86 assembly
8	M, 6/5	HLL and x86 assembly (continued)
9	W, 6/7	PIC introduction PIC instruction set
10	Th, 6/8	PIC instruction set (continued) Exam 2 Preview
	M, 6/12	**EXAM 2**
11	W, 6/14	PIC assembly programming
12	Th, 6/15	PICkit basics Working with delay
13	M, 6/19	Interrupts Analog to digital conversion
14	W, 6/21	Practice problems
15	Th, 6/22	Topics TBD Exam 3 Preview
	M, 6/26	**EXAM 3**