Course Syllabus

Course Information
CS/SE/TE 3340 Computer Architecture -- Section: 0U2
Time: MW 12:30 – 2:45 pm
Location: ECSS 2.203

This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza.

Find our class page at: https://piazza.com/utdallas/summer2017/csse3340/home

---

Professor Contact Information
Dr. Karen Mazidi
Email: Karen.Mazidi@utdallas.edu
Office: ECSS 3.203
Office phone: 972-883-3868
Office hours: M/W 5:30 – 6:30 pm

Mentoring Center Information
ECSS 4.415 Main Room / Walk-in Tutoring
login at: https://csmc.utdallas.edu/login.php to see hours of operation and other information

TA Information
TA information will be posted in Coursebook as soon as it is available

---

Course Pre-requisites, Co-requisites, and/or Other Restrictions
CE 1337 or CS 1337 or TE 1337 with a grade of C or better
and
CE 2305 or CS 2305 or TE 2305 with a grade of C or better

Course Description
This course introduces the concepts of computer architecture by going through multiple levels of abstraction, and the numbering systems and their basic computations. It focuses on the instruction-set architecture of the MIPS machine, including MIPS assembly programming, translation between MIPS and C, and between MIPS and machine code. General topics include performance calculation, processor datapath, pipelining, and memory hierarchy. Credit cannot be received for both courses, (CS 3340 or SE 3340 or TE 3340) and (CE 4304 or EE 4304).

Student Learning Objectives/Outcomes

After successful completion of this course, students will:

1. Be able to write a fully functional, stand-alone medium size assembly language program
2. Have an ability to represent numbers in and convert between decimal, binary, and hexadecimal and perform calculations using 2's complement arithmetic
3. Understand the basic model of a computer including the datapath, control, memory, and I/O components
4. Be able to program efficiently in an assembly level instruction set, including the use of addressing modes and data types
5. Understand the role of compilers, assemblers, and linkers and how programs are translated into machine language and executed
6. Be able to demonstrate comprehension of a pipelined architectures including datapaths and hazards
7. Be able to demonstrate comprehension of computer performance measures and their estimation
8. Understand the memory hierarchy including caches and virtual memory

---

Required Textbooks and Materials

Computer Organization and Design, Fifth Edition: The Hardware/Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design) Patterson and Hennessy, ISBN 9780124077263.

There are many editions of this book, make sure you get the 5th edition with MIPS.

> This course will involve coding in the MIPS assembly language using MARS (MIPS Assembler and Runtime Simulator). Get it free here:
> http://courses.missouristate.edu/KenVollmar/mars/index.htm
> The home page of MARS also has links for tutorials. See the Download link for download instructions. This should work on any computer with Java installed.

Suggested Course Materials

Any additional materials will be available from the course webpage.

Topics and Sequence

- Week 1 – 5/31
  - Course introduction
  - Number and coding systems
  - Signed and unsigned integer representations
  - Getting started with MARS
- Week 2 – 6/5 and 6/7
  - MIPS coding
  - Conversion C->MIPS->machine
  - R instructions (arithmetic/logic)
  - MIPS syscalls

- Week 3 – 6/12 and 6/14
  - I instructions
  - address calculation
  - loops
  - branching
  - J instruction
- Week 4 – 6/19 and 6/21
  - Addressing modes
  - Subroutines
  - The stack
- Week 5 – 6/26 and 6/28
  - Mul, Div instructions
  - FP representation
  - FP instructions
- Week 6 – 7/3 and 7/5
  - Dynamic memory
  - Exceptions and Interrupts
  - Exam 1
- Week 7 – 7/10 and 7/12
  - Digital logic basics
  - MIPS non-pipelined CPU
  - Control signals
- Week 8 – 7/17 and 7/19
  - Hazards
  - MIPS Pipelined CPU
- Week 9 – 7/24 and 7/26
  - MIPS memory hierarchy
  - Virtual memory
- Week 10 – 7/31 and 8/2
  - Exam 2
  - Virtual machines
  - Parallelism
- Week 11 – 8/7 and 8/9
  - Project presentations
- Week 12 - ? (8/11 or 8/12)
  - Project Presentations

Grading Policy

- Assignments (30% of the course grade): There will be approximately 8 programming/homework assignments.
- Exams (2 of them) (each 15% of the course grade).
- Project (25% of the course grade).
- Quizzes and class attendance/participation (15%).

Letter grades will be assigned as follows according to the UTD +/- conventions.

Project Overview

- Students will work singly or with *one* partner to develop a substantial coding project
- Choice 1: assembly language application (MIPS, x86, ARM, etc.)
- Choice 2: parallel/concurrent application (Go, Java, etc.)
- project must be pre-approved by instructor
- more details will be posted in eLearning

Course & Instructor Policies

- Students should take notes by hand using either physical or electronic paper. Here is why: http://www.scientificamerican.com/article/a-learning-secret-don-t-take-notes-with-a-laptop/ Also, you can use your hand-written notes on quizzes.
- Laptop computers are allowed only during "demo" time. During demo time when laptops are allowed, please be mature, be respectful of the distractions that you may cause others if you are off task. Perhaps you think you can multi-task, but science says otherwise: http://www.npr.org/2013/05/10/182861382/the-myth-of-multitasking
- Assignments must be turned in on the due date, by midnight. It's a good idea to screen-shot your upload in eLearning. This is the only way you can prove that you really did turn in the homework.
- Late assignments are deducted by 10% on the first and second days late. After two days, the assignment will not be accepted.
- There are no make-up quizzes but one quiz will be dropped at the end of the semester.
- Makeup exams are not given unless prior permission has been granted due to extenuating circumstances.
- Do not turn in programming code or other work that is not your own. This will result in a zero for all parties. Discussion and collaboration are good things, turning in someone else's work as your own is not.
- **If you do not agree with a grade you have been given, you must make your case within a week of receiving the grade or the grade stands as is.**

Comet Creed

This creed was voted on by the UT Dallas student body in 2014. It is a standard that Comets choose to live by and encourage others to do the same:

"As a Comet, I pledge honesty, integrity, and service in all that I do."

Attendance Policy

Per Computer Science administration guidelines, please be aware that ***3 consecutive absences leads to one letter grade drop. Four consecutive absences lead to an F.***

UT Dallas Syllabus Policies and Procedures

The information contained in the following link constitutes the University's policies and procedures segment of the course syllabus.

Please go to http://go.utdallas.edu/syllabus-policies for these policies.

The descriptions and timelines contained in this syllabus are subject to change at the discretion of the Professor.