Course Syllabus: Computer Architecture

Course Information
CS/SE 3340 Computer Architecture – Section 003
Time: M/W 1:00 pm – 2:15 pm
Location: ECSS 2.203

Class Resources
We will use Piazza for class discussion, announcements, exam reminders, course materials, and homework instructions. Rather than emailing questions to me or the TA, I encourage you to post your questions on Piazza so that everyone has the same information.
Sign up here: https://piazza.com/utdallas/fall2017/cs3340

Piazza also has a free android/iphone app.

Professor Contact Information
Dr. Karen Mazidi
Karen.Mazidi@utdallas.edu
Office: ECSS 3.203
Office phone: 972-883-3868
Office hours: M/W 4:00 – 5:00 pm

TA Information
TBD

Course Pre-requisites
CS/CE/TE 1337 (Computer Science I) with a grade of C or better
and
CS/CE/TE 2305 (Discrete Math I) 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

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.

Topics and Sequence

- Week 1 – 8/21 – Read: 1.1 – 1.5
  - course intro
  - numbering and coding systems
  - signed and unsigned integer representation
  - getting started with MIPS and MARS
- Week 2 – 8/28 – Read: 2.1 – 2.6
  - MIPS programming
  - using C as pseudocode
  - machine language instructions
  - arithmetic/logic instructions
  - MIPS syscalls
  - **Homework 1: Getting started with MIPS**
- Week 3 – 9/4 – Read: 2.7 - 2.8, Appendix A
  - Monday 9/4 Labor Day no classes

    - branch instructions
    - address calculation
    - loops
- Week 4 – 9/11 – Read: 3.1 – 3.4
    - addressing modes
    - mul/div instructions
    - **Homework 2: Basic MIPS**
- Week 5 – 9/18 – Read: 2.7 – 2.8, 1.6-1.11
    - subroutines
    - the stack
    - putting it all together into MIPS programs
    - evaluating computer performance
    - **Homework 3: MIPS control structures**
- Week 6 – 9/25
    - Exam 1 Review
    - Exam 1
- Week 7 – 10/2 – Read: 3.5
    - floating-point representation
    - floating-point operations
    - **Homework 4: Floating point operations**
- Week 8 – 10/9
    - dynamic memory
    - exceptions and interrupts
    - advancing coding techniques
    - **Homework 5: Iteration and Recursion**
- Week 9 – 10/16
    - Exam 2 Review
    - Exam 2
- Week 10 – 10/23 – Read: Appendix B, 4.1 – 4.4
    - digital logic basics
    - MIPS non-pipelined CPU
    - control signals
    - **Homework 6: Advanced architecture presentation**
- Week 11 – 10/30 – Read: 4.5 – 4.8
    - hazards
    - MIPS pipelined implementation
    - pipeline hazards
- Week 12 – 11/6 – Read: 5.1 – 5.10
    - MIPS Memory Hierarchy
    - advanced memory concepts
    - **Homework 7: Memory optimization**
- Week 13 – 11/13
    - Exam 3 Review
    - Exam 3
    - **Homework 8: File I/O and parity simulation**

- Week 14 – 11/20
    - Fall break – no classes
- Week 15 – 11/27 – Read: Chapter 6
    - Parallelism, new programming paradigms
    - Summary: MIPS v. ARM v. x86
    - Project Presentations
- Week 16 – 12/4
    - Project Presentations
- Final Exam Week: Fri 12/8 – Thurs 12/14

Grading

Course grade is composed of:

- 25% - homework; approximately 8 assignments
- 45% - 3 exams; 15% each
- 25% - project
- 5% - attendance (1 - %unexecused absences)

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

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/
- Laptop computers should only be open during coding time. We will practice coding in class, which you can either do on your laptop or paper if you don't want to bring a laptop to class. During coding time when laptops are allowed, please be mindful 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 confirmation 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.
- 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***

UT Dallas 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.