CS/EE 340 Computer Architecture
Fall 2009
Monday, 7:00pm – 9:45pm, SCIT 215

The schedule, together with assignments and lectures, is subject to change in the progress of the course. Announcements made in the class override the schedule in case of conflicts.

Instructor: Dr Victor Govindaswamy
Email: victor.govindaswamy@tamut.edu
Website: https://www.eagle.tamut.edu/faculty/victor/linkpage.html

Class schedule, lecture notes, assignments, and announcements are all posted to the class website. Please check the website before every class.

Semester Credit Hours: 3 hours

Office Location: SCIT 114
Office Hours: M 11:00-1 pm, 6-7 pm
T 7-10 pm
W 4-5 pm
R 11:00-1 pm, 6-7 pm
Phone: 903.334.6657

Required Textbook/Resources:

Computer Organization and Architecture: Designing for Performance, 8th Edition, William Stallings, Copyright: 2010, Publisher: Prentice Hall, (ISBN-10: 0136073735, ISBN-13: 9780136073734), Published: 04/03/2009.

Software & Hardware:

1) This course will be taught using the Microsoft Macro Assembler 8.0 (MASM) Tool within the WINDOWS environment. The main hardware/software platform used during class will be a Microsoft NT 4.0 Workstation accessing the MASM programming environment. Other tools might also be used such Multimedia Logic, MARS MIPS emulator, CEDAR Logic Emulator and JLS.

2) Students can get all the Lecture Slides, notes and other links at the course webpage. The link for the course web page will be provided on the day of the first class.

3) Students must keep copies of all assignments and projects sent by e-mail.

4) Assignments are to be word-processed. Continuing and regular use of e-mail is expected.

Course Description:

Tools amplify our abilities. An airplane is a tool that lets us fly. A car is a tool that lets us travel quickly from place to place on the ground. So, what is a computer? Computers are not human. But they are probably the most human tool that man has ever made.

When is it safe to rely on results from a computer? What are its limits? When we ask it to do such-and-such, how long will it take? Will the answer be right? When the program ``breaks," what might have caused the problem? If we really know cars and engines, we drive differently, with a trained ear and a sense of what is really happening. We can push the car to its limits and we know when to back off. With a computer, to really drive it well, it helps to have a similar sense for what the machine is really doing.

Computer architecture is concerned with the structure and behavior of the various functional modules of the computer; and how they interact to provide the processing needs of the user. In particular this course covers computer systems ranging from PCs through multiprocessors with respect to hardware design and instruction set architecture. This includes units and related technologies such as primary and secondary memory, caches, central processing unit (CPU), and pipelines.

A menu of "possibilities" will be presented, analyzed, and evaluated based on the technology available today. In no event should it be assumed that the architecture that looks strongest today will be the best in the new millennium. My approach will be that it is methodology, not conclusions, that must be emphasized. For while methodology is relatively timeless, conclusions are not. We will learn the simple, native language of computers. We will learn how computers think and some of the limits they have. We strive to truly understand computers, and we will make a very good start in this course. We will lay the foundation. We will gain an understanding of bits and bytes, of ands and ors and nots, of integers and floats and doubles, of addition, subtraction, multiplication, and division, of gates, latches, flip-flops, and memories. We will learn to think like a computer, and thereby realize the limitations on the thinking of all computers.

Justification:

Engineers, especially electrical and software, and computer scientists need to understand the relationship between modern computer architecture and the performance of software running on that hardware. This course is meant to provide an introduction to computer architecture, especially those aspects that have underpinned performance developments on single CPU machines over the last 20 years: namely the memory hierarchy and pipeline. It also will show how the software environment influences performance, in particular the compiler, programming language and parallel programming model.

Special Considerations:

The material in this course is far more challenging than that in your first few programming courses and the pace is faster. To have the best chance of succeeding in the course, you must plan to devote at least two hours outside of class for every hour spent in class. You must read the assignments before coming to class and work through examples and problems in the textbook. You should be aware that missing even one lecture will put you significantly behind your classmates. If you must be absent, arrange to obtain the lecture notes of someone who is an

accomplished note-taker. All of the lecture notes and slides are posted to the class website; it is highly recommended to print these and take notes directly on the lecture slides.

Course Objectives and Outcomes:

This course prepares students to work professionally in the area of computer architecture design. At the end of this course, a student will

- Develop an understanding of the architecture of small computer systems.
- Learn central processor design methodologies and evaluation techniques.
- Learn relationship between hardware design and instruction set architecture.
- Study various processor architectures and organizations.

Course Outline:

Each of the following topics will be covered in this course:

- Chapter 1 Introduction
  - 1.1 Organization and Architecture
  - 1.2 Structure and Function
  - 1.3 Key Terms and Review Questions
- Chapter 2 Computer Evolution and Performance
  - 2.1 A Brief History of Computers
  - 2.2 Designing for Performance
  - 2.3 The Evolution of the Intel x86 Architecture
  - 2.4 Embedded Systems and the ARM
  - 2.5 Performance Assessment
  - Introduction to Assembly Language Lab
- Chapter 3 A Top-Level View of Computer Function and Interconnection
  - 3.1 Computer Components
  - 3.2 Computer Function
  - 3.3 Interconnection Structures
  - 3.4 Bus Interconnection
  - 3.5 PCI
  - Assembly Language Lab
- Chapter 4 Cache Memory
  - 4.1 Computer Memory System Overview
  - 4.2 Cache Memory Principles
  - 4.3 Elements of Cache Design
  - 4.4 Pentium 4 Cache Organization
  - 4.5 ARM Cache Organization
  - Assembly Language Lab
- Chapter 5 Internal Memory Technology
  - 5.1 Semiconductor Main Memory
  - 5.2 Error Correction
  - 5.3 Advanced DRAM Organization
  - Assembly Language Lab

- Chapter 6 External Memory
  - 6.1 Magnetic Disk
  - 6.2 RAID
  - 6.3 Optical Memory
  - 6.4 Magnetic Tape
  - Assembly Language Lab
- 7 Input/Output
  - 7.1 External Devices
  - 7.2 I/O Modules
  - 7.3 Programmed I/O
  - 7.4 Interrupt-Driven I/O
  - 7.5 Direct Memory Access
  - 7.6 I/O Channels and Processors
  - Assembly Language Lab
- Chapter 8 Operating System Support
  - 8.1 Operating System Overview
  - 8.2 Scheduling
  - 8.3 Memory Management
  - 8.4 Pentium Memory Management
  - 8.5 ARM Memory Management
  - Assembly Language Lab
- Chapter 9 Computer Arithmetic
  - 9.1 The Arithmetic and Logic Unit (ALU)
  - 9.2 Integer Representation
  - 9.3 Integer Arithmetic
  - 9.4 Floating-Point Representation
  - 9.5 Floating-Point Arithmetic
  - Assembly Language Lab
- Chapter 10 Instruction Sets: Characteristics and Functions
  - 10.1 Machine Instruction Characteristics
  - 10.2 Types of Operands
  - 10.3 Intel x86 and ARM Data Types
  - 10.4 Types of Operations
  - 10.5 Intel x86 and ARM Operation Types
  - Assembly Language Lab
- Chapter 11 Instruction Sets: Addressing Modes and Formats
  - 11.1 Addressing
  - 11.2 x86 and ARM Addressing Modes
  - 11.3 Instruction Formats
  - 11.4 x86 and ARM Instruction Formats
  - 11.5 Assembly Language
  - Assembly Language Lab
- Chapter 12 Processor Structure and Function
  - 12.1 Processor Organization
  - 12.2 Register Organization
  - 12.3 The Instruction Cycle
  - 12.4 Instruction Pipelining
  - 12.5 The x86 Processor Family
  - 12.6 The ARM Processor

Course Requirements and Etiquette:

Website and Email: Students are responsible for checking the course Website frequently for updates and notices relative to class materials and schedule. Email may be used occasionally to send notices of an advisory nature, but should NOT be relied upon as the official means of communication to the class. Subject line of email should indicate "EE-CS 340" and include the student's full name. It is the student's responsibility to ensure that I have your email address in my address book, and that the provided address is available and operational.

The course schedule is achievable. However, if necessary, the schedule will be modified during the semester to allow coverage of the most critical topics. I will attempt to provide advance notice of changes, but you should check the schedule frequently on the course Website.

Participation: Participation extends beyond mere attendance. Expect your instructor to keep track of how often you contribute to class discussion (as a whole), particularly during the panel discussion section. You may miss up to three classes without penalty - your first two absences count whether you have a good excuse or not. Each absence beyond the first three will cost you points off of your participation grade. The only exceptions to this rule are severe illness (doctor's note required) and A&M-Texarkana's approved trips/activities. Appropriate documentation for absences beyond the first three is necessary the class day directly before or after the one you miss. In general: this class is intensive and interactive. Missing class could seriously affect your grade! Students are reminded not to approach the instructor for copies of the previous week's materials during immediately before, during, or immediately after class. Students are expected to collect materials from their classmates or see the instructor during office hours.

Homework/Labs: Homework in the form of the labs will be given frequently throughout the semester to measure learning progress. The schedule for these labs indicated on the course website may be adjusted at the discretion of the instructor based on the pace of topics covered. Each lab is due on the day indicated on the course schedule available on the course web site. It is usually due a week after it has been assigned. Homework assignments will be due and collected, in the form of hardcopies as well as emailed softcopies, at the beginning of the class period on the required due date. Work submitted after the end of class will be considered late. Both hardcopies as well as email softcopies need to be submitted on time, otherwise it will be considered late.

Note: Save your work frequently on different disks and start your work early. A computer crash is no excuse for a late delivery, nor is non-working printer for last minute printouts.

Exams: Two examinations will be taken, covering all course content up the week before the each exam. There will be two subject matter exams covering the key sections of the text, lectures and labs, and course content up to the week before the each exam. Exam format may combine a mixture of short answer, true/false, matching, sort answer, one or two essay questions and/or algorithmic or programming questions covering *all* readings, lecture, hand-out and class discussion content. Two-hours will be allotted for these exams, and they will begin promptly at the beginning of the class period during which they are scheduled. These exams will be open book and open class notes.

NO early exams will be given. Failure to appear for a scheduled exam at the appointed time, unless due to a dire emergency will result in the assignment of a zero grade.

Grade Challenges: Discussion or challenges of individual grades will not be entertained in the classroom before, during or immediately following class. Normal office hours are available for this purpose. Solutions/keys for assignments will be discussed in class, but will generally not be posted or made available for general distribution. In the case of dispute concerning submission/grade on an assignment, it is the student's responsibility to produce papers as proof. Tests/Exam Papers/Projects will not be returned to the students permanently.

Method of Instruction:

Each lecture will cover the same topics as the reading assignment in the textbook but often from a different perspective. The lectures will not simply repeat the material in the textbook. It is, therefore, essential that you read the assigned material and work through the problems in each chapter in advance of the lecture. Otherwise, you may find the lectures difficult. Pop quizzes over the reading material will be given at random intervals to encourage you to do the readings.

Withdrawals:

Students in good standing who withdraw before the official withdraw date (see schedule of classes: http://tamut.edu/admissions/fallschedule2008.htm) will receive a grade of W. Students withdrawing after this date will receive a grade of F unless a hardship authorization is obtained.

Class Attendance Policy:

Rolls may be taken on a regular basis. It is your responsibility to take notes, obtain assignments, and turn in work on time. Your absence from class does not relieve you of any of these responsibilities. .A grade of zero will be assigned on any missed assignments/examinations/ labs/homeworks unless the instructor is notified in advance and arrangements are made to take the quiz at another time.

Incompletes:

A grade of I will be given only in exceptional circumstances.

Grading Policy:

Name	Description	Weight
Examination No. 1	This exam will assess your understanding of the material covered in class during the first half of the semester.	15%
Examination No. 2	This exam will assess your understanding of the material covered in class during the second half of the semester. While no specific emphasis will be placed on the topics already included in the first exam, you will be expected to have and, whenever relevant, show an in-depth understanding of those topics when discussing the new material in the second half of the semester.	35%
Homeworks (Labs)	Frequent exercises will be given to assess your understanding of the material covered in class.	50%

The instructor reserves the right to make other assignments that are not part of the published schedule.

Grading Scale:

In general, semester grades will be determined as follows:

A 100 to 90
B 89 to 80
C 79 to 70
D 69 to 60
F 59 and below

Final grades **may** be curved based on overall class performance. Grades are final once submitted, and are not changed unless a grading error has been made.

Late / Makeup Policy:

There is a 100% penalty for any late submissions.

Program Grading:

1. Programs are due, hardcopies as well as emailed softcopies, at the beginning of class on the date specified. Late programs (submitted after the beginning of lecture on the due date) will NOT be accepted. Emailed assignments will not be graded. Only hardcopies will be graded.
2. A copy of your source program and its output should be submitted in hard copy form as well as an email zipped file. Program output should be labeled (by the program) so that it is understandable to the reader, with the assignment number, date, and your name. Program output and disks without proper labeling will not be graded.

3. Testing is a critical part of the programming process. The burden of proof that a program works always rests with the programmer. Whether or not test data are provided, you must adequately test your program to insure that it works correctly in all reasonable cases.
DO NOT WAIT UNTIL THE LAST MINUTE TO START THESE ASSIGNMENTS.
4. Programming assignments may specify that a particular approach be used. Read the assignment carefully to be sure that you understand how the problem should be solved. If you use the wrong approach, the fact that your program produces the correct output is immaterial.
5. Programs must be well-structured, readable, and efficient. Use meaningful names, indentation, comments, and other elements of style discussed in the course. An unreadable program is not maintainable and is worthless even if it produces the correct result.
6. Output should be neat, properly aligned and have useful headings. Requests for interactive input should be preceded by a prompting message.
7. Do not jeopardize your grade by allowing others to copy your work. The penalties for giving and receiving help are the same.

In general, the following point system will be used for grading of code:
Overall Deductions:

- 20% Not using the provided test data
- 20% Improper use of language features (see 4)
- 20% Not following the instructions regarding the required approach (see 4)
- 30% Your name and assignment number does not appear in your code and/or output

50% Performance

- 0 Program runs correctly and produces the correct output
- 10 Program has minor error (e.g. typo in formula or text)
- 20 Program has major error or many minor errors
- 30 Program does not run due to syntax error
- 40 Only program fragments turned in

30% Maintenance

- 10 Poor naming of variables/functions
- 5 Improper indentation
- 10 Poor program structure
- 20 No documentation in code

20% Input/Output

- 5 No prompting for input
- 5 No headings
- 5 Alignment problems
- 20 No proof of output provided

Cheating: You are expected to turn in your own original work. Getting help in finding an error is encouraged, but copying other students work or code from other sources is forbidden and will result in a grade of zero for that assignment or for the entire course. You need to be able to explain your program.

Academic Dishonesty:

Academic honesty is expected of students enrolled in this course. Cheating on examinations, unauthorized collaboration, falsification of research data, plagiarism, and undocumented use of materials from any source constitute academic dishonesty and may be grounds for a grade of 'F' in the course and/or disciplinary actions. For additional information, see the university catalog.

Disability Accommodations:

Students with disabilities may request reasonable accommodations through the A&M-Texarkana Disability Services Office by calling 903-223-3062.

The schedule, together with assignments and lectures, is subject to change in the progress of the course. Announcements made in the class override the schedule in case of conflicts.