



## Course Syllabus

---

### Course Information

#### **CS/SE 3340.502 – Computer Architecture**

Term:

**Spring 2017**

Days & Time and Location:

TTh 5:30PM - 6 :45PM @ ECSS 2.412

---

### Instructor Contact Information

*Nhut Nguyen, Ph.D.*

Phone: 972-883-4521

Email: [nhutnn@utdallas.edu](mailto:nhutnn@utdallas.edu)

Office hours: TTh 10:00AM – 11:00AM, also by appointment

Office: ECSS 3.607

---

### Course Pre-requisites, Co-requisites, and/or Other Restrictions

CE/CS/TE 1337 or equivalent, and CE/CS/TE 2305 or equivalent.

---

### Course Description

This course introduces the concepts of computer architecture by going through multiple levels of abstraction, 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. Students that have credit for CS 2310 or CS/SE4340 cannot receive credit for this course (3 semester hours).

---

### Student Learning Objectives/Outcomes

After successful completion of this course, the student should

- be able to write a fully functional, stand-alone medium size assembly language program (e.g. a basic Telnet client)
- have an ability to represent numbers in and convert between decimal, binary, and hexadecimal and perform calculations using 2's complement arithmetic
- understand the basic model of a computer including the datapath, control, memory, and I/O components
- be able to program efficiently in an assembly level instruction set, including the use of addressing modes and data types
- understand the role of compilers, assemblers, and linkers and how programs are translated into machine language and executed

- be able to demonstrate comprehension of a pipelined architectures including datapaths and hazards
- understand the memory hierarchy including caches and virtual memory
- be able to demonstrate comprehension of computer performance measures and their estimation

---

**Required Textbook:**

**"Computer Organization and Design - The Hardware/Software Interface – 5<sup>th</sup> Edition"**, Patterson and Hennessey, Morgan-Kaufmann, 2013. ISBN-13: 978-0124077263. **Note: there are several editions of the same title, make sure that you get the correct edition (for MIPS).**

**Required Course Materials:***RISC ASSEMBLER/SIMULATOR*

It is assumed you are familiar with the PC environment, can create and edit text files, run programs, etc. The programs will be in assembly language for the MIPS processor. This course uses the MARS MIPS assembler and simulator. MARS is available, free, for download from the Internet through the site:

<http://courses.missouristate.edu/kenvollmar/mars/>.

The MARS simulator can assemble MIPS assembly language source files, load and run them with a users console window for input/output, and debug them if they do not work properly.

---

**Assignments & Academic Calendar**

**Exams:** There will be three exams during the course: two midterms and a final exam. The exams will be open notes: notes taken during the sessions can be used in the exams but no books or slides are allowed. The midterm exams will be limited to material covered during the immediate unit but the final exam is comprehensive. Test material will be taken mainly from classroom lectures.

**Assignments:** There will be regularly assigned reading and homework. Reading assignments should be done before the class session. Homework will require the student to spend time programming a computer outside of class. It includes a program to demonstrate the correct operation of the assigned tasks.

There will be regularly assigned in-class exercises that will be used to assess student's participation.

Assignments should be submitted using your eLearning account. Each programming assignment must contain:

1. A copy of the final working assembly language source code with comments and documentation. The file should be "text-only" and the extension must be ".s" or ".asm".
2. A screenshot showing keyboard input and displayed output from the console.

**Project:** A team programming project will be assigned. Details will be announced in the class.

## Class Schedule

<b>Week</b>	<b>Date</b>	<b>Topic</b>	<b>Reading Assignment</b>	<b>Assignments</b>	<b>Due</b>
1	Jan 10	Introduction			
	Jan 12	Intro to computer organization	Ch 1		
2	Jan 17	Introduction to Assembly Language Programming	Appendix A	HW #1	
	Jan 19	Performance evaluation, Amdahl's law	Ch 1.6,1.9		
3	Jan 24	Data Representations, Bin/Oct/Hex	Ch.2.3		HW #1
	Jan 26	Number Representations: signed, floating point	Ch.2.4	HW #2	
4	Jan 31	Instructions Representation	Ch 2.5		
	Feb 2	Assembly Ops: Load/Store/Add/Sub/etc	Ch 2.2		HW #2
5	Feb 7	Comparing, Branching and Looping	Ch 2.7	HW #3	
	Feb 9	Bits and bytes manipulation & other instructions	Ch 2.6		
6	Feb 14	Subroutines in Assembly language	Ch 2.8, A.6		HW #3
	Feb 16	Exam I review		HW #4	
7	<b>Feb 21</b>	<b>Exam I</b>			
	Feb 23	Comparing ISAs	Ch. 2.16-7		HW #4
8	Feb 28	Addressing modes & System software	Ch 2.10, 2.12-13	HW #5	<b>Project CHKP</b>
	Mar 2	Integer Arithmetic	Ch 3.1-3.4		
9	Mar 7	Floating Point Arithmetic	Ch 3.5		HW #5
	Mar 9	Input & Output		HW #6	
10	<b>Mar 13-18</b>	<b>Spring Break</b>			
11	Mar 21	Interrupts and Exceptions	Ch 4.9, A.7		
	Mar 23	Exam II review			HW #6
12	<b>Mar 28</b>	<b>Exam II</b>			
	Mar 30	Processor: Datapath & Control	Ch 4.1-4	HW #7	
13	Apr 4	Processor: Pipelining	Ch 4.5		
	Apr 6	Processor: Pipelined Datapath	Ch 4.6-8		HW #7
14	Apr 11	Advanced Instruction Level Parallelism	Ch 4.10		
	Apr 13	Introduction to memory hierarchy	Ch 5.1-3		
15	Apr 18	Virtual Memory	Ch 5.4-7		
	Apr 20	<i>Contingency</i>			
16	Apr 25	Exam III Review			<b>Project</b>
	<b>Apr 27</b>	<b>Exam III</b>			

---

## Grading Policy

The grade each student will earn from this class will be based on a weighted score calculated by using the following table:

		Weighted Score	Grade
		93.0 - 100	A
		90.0 - 92.9	A-
		87.0 - 89.9	B+
		83.0 - 86.9	B
		80.0 - 82.9	B-
		77.0 - 79.9	C+
		73.0 - 76.9	C
		70.0 - 72.9	C-
		67.0 - 69.9	D+
		60.0 - 66.9	D
		Below 60.0	F

Exam I	10%
Exam II	15%
Exam III	25%
Assignments	20%
Project	30%
	100%

Grades will be assigned according to the scale on the right:

### *Programming assignments grading:*

Code Development	30%	(compile w/o error)
Program Execution	20%	(run successfully)
Program Design	25%	(conform to spec)
Documentation	15%	(program, comments)
Coding Style	10%	(clear, efficient)

---

## Course & Instructor Policies

- **Attendance policy:** missing **four** in-class exercises leads to **one letter grade drop**, missing **five** in-class exercises leads to **an F grade**.
- There will be no makeup exams under normal circumstances.
- No late homework or assignment will be accepted!
- I do not read e-Learning e-mails. Please use my UTD e-mail account above for any communications.

---

## 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.

*These descriptions and timelines are subject to change at the discretion of the Instructor.*