Computer Organization and Design

Comp Sci 310 – Spring 2016

Instructor: Scott Summers **Office:** Halsey 220
Email: summerss@uwosh.edu **Phone:** 920-424-1324

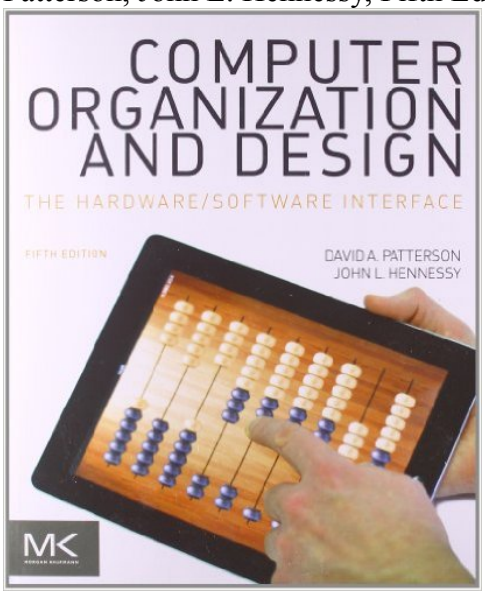
Office hours: MWF 10:20 AM – 11:20 AM
T 10:00 AM – 11:00 AM, 12:40 PM – 1:40 PM

Lectures: MWF Halsey 456 9:10 AM – 10:10 AM

Prerequisites: CS251 and Math212 with a grade of C or better.

Course website: http://www.uwosh.edu/d2l. You should check d2l on a regular basis as it will contain any lecture notes, handouts, assignments, announcements, and grades. I'll do my best to let you know when something new and important comes up, but it is your responsibility to check the web site frequently for information that you might not get otherwise.

Recommended textbook: Computer Organization and Design: The Hardware/Software Interface, David A. Patterson, John L. Hennessy, Fifth Edition, Morgan Kaufmann, 2014.



Note: If you have special needs, please come and talk to me at the end of the first class.

Course description and learning outcomes: This course focuses on the design of microprocessors, especially the Arithmetic and Logic Unit and the Control Unit, as well as the memory hierarchy, especially registers, DRAM-based main memory and caches. At the conclusion of the course, the student will be able to:

1) Outcome #1
   a) describe the various ways in which computer performance can be measured and explain the pros and cons of each measure
   b) explain the tradeoffs associated with instruction set architecture design
   c) use the CPU performance equation to compare the performance of processor architectures

d) apply Amdahl's law to understand the impact of an architectural modification on performance

2) Outcome #2
   a) understand and describe the tradeoffs between complex instruction set computers (CISC) and reduced instruction set computers (RISC)
   b) understand how fundamental mathematical operations, such as addition, subtraction, multiplication, and division, can be optimized with appropriate number representation, rounding, and digital circuit implementation schemes
3) Outcome #3
   a) understand the key principles used in creating datapaths and designing the control unit for single-cycle and multi-cycle microarchitectures
   b) appreciate how the choice of implementation strategy affects the clock rate & CPI of a computer system
   c) distinguish between microprogrammed and hardwired processor control, and describe the benefits of each approach
   d) explain the concept of instruction-level parallelism and describe the challenges associated with taking advantage of it
   e) describe the general design of a processor that implements instruction-level parallelism
   f) identify data, control and structural hazards for a given architecture/code segment pair
   g) analyze the performance of a code segment in a given instruction execution pipeline
4) Outcome #4
   a) describe how a cache works
   b) enumerate various methods for enhancing cache and memory performance
   c) distinguish among directly-mapped cache, associative cache, and set-associative cache, and describe the principal issues related to cache memory organization
   d) quantify the impact of the hit ratio on the effectiveness of the cache memory system
5) Outcome #5
   a) demonstrate the ability to work effectively in teams on small-scale digital design projects

Course grade: Your final course grade will be based on the following components:

Component	Weight
Quizzes	20%
Homework assignments	20%
Exam #1	20%
Exam #2	20%
Exam #3	20%

Grading will be on a plus/minus system. Grading *may* be done on a curve depending on the overall performance of the class. If no curve is used, your grade will be computed based on the following:

Percentage	Grade	Percentage	Grade	Percentage	Grade
>91	A	>79 and ≤ 81	B-	>67 and ≤ 69	D+
>89 and ≤ 91	A-	>77 and ≤ 79	C+	>61 and ≤ 67	D
>87 and ≤ 89	B+	>71 and ≤ 77	C	>55 and ≤ 61	D-
>81 and ≤ 87	B	>69 and ≤ 71	C-	≤55	F

Exams: Exam material will come from the lectures, quizzes and assignments. There will be more information about each exam as it approaches. The actual exam dates will be announced in class at least one week before the exam. All exams will be taken during the regular class period. If you are unable to take a scheduled exam, it may be possible to take a make-up exam provided that you do both of the following, which are then subject to my approval:

1. Make arrangements prior to the scheduled exam (for last minute emergencies, telephone me at 920-424-1324 or leave a message at the Computer Science office, 920-424-2068 **No after-the-fact notifications will be accepted.**
2. Have a written medical excuse signed by the attending physician OR have a note of justification from the Dean of Students Office.

If allowed, only one make-up exam will be given. It will be a comprehensive exam given at an arranged time during the last week of the semester.

Assignments: Late homework assignments will NOT be accepted. Extensions on assignments and labs may be granted at the discretion of the instructor if you provide a valid justification (in the form of a written excuse from a medical doctor or the Dean of Students Office) before the due date.

Late assignments are worth 0 points.

If you believe an assignment, quiz or exam was graded incorrectly or unfairly and would like to have it re-graded, please let me know about it *within one week* of having the assignment or exam graded. I will re-grade the entire assignment and you may gain or lose points.

Academic Dishonesty: Academic dishonesty of any kind will not be tolerated. All assignments (except for those designated as group assignments), quizzes and exams are to be completed individually. While discussion of ideas and problems with fellow students is encouraged, all projects and labs must be done individually. In certain circumstances, code fragments from the instructor may be provided to eliminate tedious coding or to provide a common framework for all students. All other code must be original. Online resources may be used to help you understand the material, but you may not copy online code nor can you "borrow" code from other students, past or present. For group assignments, each group must submit original work.

Any suspected academic dishonesty will be dealt with on a case-by-case basis. Any clarification of what does or does not constitute academic dishonesty must take place *before* you turn in questionable work. For clarification on what constitutes academic dishonesty, contact me or consult the printed policy in the UWO Student Discipline Code, Chapter UWS 14.