Fall 2012

CEG 3310-01: Computer Organization

Michael L. Raymer
Wright State University - Main Campus, michael.raymer@wright.edu

CEG 3310 – Computer Organization

Course Overview

Understanding how a computer works is essential to writing efficient, bug-free code. In this course you will learn about the basic organization of computing systems. From digital logic and devices to instruction set architecture, we will explore the organization and control of a CPU at multiple levels of detail. You will learn how high level languages are compiled into assembly language and converted to machine code, and how that machine code runs on a CPU. We will explore how understanding computer organization is essential for writing efficient code, writing error-free code, and protecting your code from malicious use and misuse.

Learning Outcomes

After completing this course, students will be able to:

1. Perform calculations using binary, hexadecimal, and 2's complement number systems
2. Interpret floating point numbers in IEEE-754 representation
3. Explain the function of the key components of a modern CPU, including the ALU, MUXs, buses, register files, and the memory interface
4. Use the run-time stack to pass parameters between assembly language and high level code
5. Explain the key concepts of polled, memory-mapped I/O, interrupt-driven I/O, and DMA
6. Determine the effects of cache and memory hierarchy design decisions on overall memory performance

Course Details

Instructor: Dr. Michael Raymer
Office: 391 Joshi Phone: 775-5107
Email: michael.raymer@wright.edu

Textbook: Patt, Yale and Patel, Sanjay (2004). "Introduction to Computing Systems, 2nd ed.", McGraw Hill, ISBN 978-0072467505. *Required.*

For course lecture time & place, and instructor office hours, check the course web page at http://pilot.wright.edu (http://pilot.wright.edu).

Lecture Schedule

Week(s)	Topic(s)	Reading
1	Introduction, representing data	Chs. 1 & 2
2	Digital logic, the Von Neumann model	Chs. 3 & 4

3 & 4	The LC-3 machine & simulator	Ch. 5
5	LC-3 assembly language	Chs. 6 & 7
6	LC-3 data path & control signals	Chs. 5 & 6
7	Low level I/O, traps	Chs. 8 & 9
8	C programming	Chs. 11 – 13
9	The run time stack	Ch. 10 & 11
10	Parameter passing and recursion	Ch. 14 & 17
11	Interrupts	Ch. 10
12	Intel x86 assembly language	Notes
13	x86, Cache memory	Notes
14	Cache, virtual memory, final exam review	—

Grading

Quizzes (best 5 of 6): 30%
Final exam: 35%
Lab Assignments (6): 35%

Final grades will be assigned according to the university standard grading scheme: 90-100% = A, 80-89% = B, 70-79% = C, 60-69% = D, 0-59% = F. Depending on the overall distribution of final grades, the boundaries *may* be adjusted by the course instructor to ensure consistency from term to term.

Final Exam Exemption

Students that achieve a grade of A (32 or more points out of 35) on *all six quizzes* may choose to skip the final exam. If you choose to do so, the average of your quiz grades (as a percentage) will be used as your final exam grade for the overall grade calculation.

Course Web Page

The course web page is at http://pilot.wright.edu (http://pilot.wright.edu). Log in using your CaTS username (w000xxx) and password. *It is your responsibility to check this page often for announcements, assignments, and other important information.*

Course Policies

Late Assignments

Lab assignments are due by 11:59 pm on the due date. Late lab assignments will be accepted, but 10% of the total available points will be deducted for each day late. Labs are considered one day late after

11:59 pm on the due date. At 11:59 pm of each successive day (including weekends) the lab is considered an additional day late until turned in. *No points will be awarded for labs turned in more than one week (7 days) late.*

Academic Integrity

Discussion of course contents with other students is an important part of the academic process and is encouraged. However, it is expected that course programming assignments, homework assignments, and other course assignments will be completed *on an individual basis*. Students may discuss general concepts with one another, but may not, under any circumstances, work together on the actual implementation of any course assignment. If you work with other students on "general concepts" be certain to *acknowledge the collaboration* and its extent in the assignment. Unacknowledged collaboration will be considered dishonest. "Code sharing" (including code from previous quarters) is strictly disallowed. "Copying" or significant collaboration on any graded assignments will be considered a violation of the university guidelines for academic honesty.

If the same work is turned in by two or more students, *all parties involved will be held equally accountable for violation of academic integrity*. You are responsible for ensuring that other students do not have access to your work: do not give another student access to your account, do not leave printouts in the recycling bin, pick up your printouts promptly, do not leave your workstation unattended, etc. If you suspect that your work has been compromised notify me immediately.

Failure to attend the first day of class, during which time I will explain these academic honesty policies in detail, does not excuse you from following these policies. If you have any questions about collaboration or any other issues related to academic integrity, please see me immediately for clarification.

In addition to the policy stated in this syllabus, students are expected to comply with the Wright State University Code of Student Conduct (http://www.wright.edu/students/judicial/conduct.html) and in particular the portions pertaining to Academic Integrity (http://www.wright.edu/students/judicial/integrity.html) at all times.

Additional Needs

Students with disabilities or any additional needs are encouraged to set up an appointment with me at their convenience to discuss any classroom accommodations that may be necessary.