University of Nevada, Reno

COMPUTER SCIENCE & ENGINEERING DEPARTMENT
CS 219 Computer Organization
Spring 2018

Course Information

- **Credits:** 3.0
- **Lecture hours:** Monday, Wednesday, 4:00 - 5:15 pm
- **Instructor:** Dwight Egbert, Professor of Computer Science & Engineering (egbert@cse.unr.edu)
- **Office Hours:** 1:00 - 2:30PM Monday & Wednesday (or by appointment); Room 322 SEM;
- **Final Exam:** May 14, Monday, 2:45 - 4:45PM, in the classroom.
- **MidTerm Exam:** March 15, Thursday, 4:00-5:15 PM, in the classroom.

Description

Introduction to organization and integration of computer components. Topics include: computer abstractions and performance, arithmetic operations, instruction set architecture, assembly programming, datapath, pipelining, memory hierarchy, I/O, and parallel architectures.

Prerequisites

CS 202 and CPE 201

Textbooks, Hardware, and Software

Required Textbook

- WilliamStallings(2015) Computer Organization and Architecture: Designing for Performance, 10th Edition. Pearson. (ISBN: 13: 9780134101613) - or International Edition (ISBN:13: 978-1292096858) - or CourseSmart eTextbook (ISBN: 13: 9780134102061)

Required hardware and software

- Each student must have access to a laptop or PC. All of the software required for this course is either open source or free to download. There may be some software that is only available for Windows, although we are trying to find suitable software that will run on Linux and MAC PCs as well.

Recommended Reference

- M. M. Mano and C. R. Kime. (2008) Logic and Computer Design Fundamentals, 4th Edition. Pearson. (ISBN: 0-13-600158-0).

Academic Success Services

- Your student fees cover usage of the Math Center (784-4433 or www.unr.edu/mathcenter/), Tutoring Center (784-6801 or www.unr.edu/tutoring/), and University Writing Center (784-6030 or www.unr.edu/writing_center/. These centers support your classroom learning; it is your responsibility to take advantage of their services. Keep in mind that seeking help outside of class is the sign of a responsible and successful student.

- If you have a disability for which you will need to request accommodations, please contact me or Mary Zabel at the Disability Resource Center (Pennington Student Achievement Center, Suite 230, 784-6000, or www.unr.edu/drc/), as soon as possible to arrange for appropriate accommodations.

- Surreptitious or covert video-taping of class or unauthorized audio recording of class is prohibited by law and by Board of Regents policy. This class may be videotaped or audio recorded only with the written permission of the instructor. In order to accommodate students with disabilities, some students may have been given permission to record class lectures and discussions. Therefore, students should understand that their comments during class may be recorded.

The University Math Center (UMC) is focused on helping students with mathematical and statistical concepts. While mathematics is used extensively in engineering, the UMC does not have the resources to help students with engineering courses. Engineering students are encouraged to use the UMC for help in their math classes, and they are welcome to use its computer lab and study area any time –regardless of course. However, UMC tutors cannot answer questions regarding engineering courses.

To obtain tutoring for specific college of engineering courses please see: [https: //www.unr.edu/engineering/student-resources/tutoring](https://www.unr.edu/engineering/student-resources/tutoring) or drop in at Edmund J. Cain Hall 108H.

Textbook Chapters and Topics **Approximate Calendar**

```
Chapter 1 Basic Concepts and Computer Evolution                  Week 1
1.1   Organization and Architecture
1.2   Structure and Function
1.3   A Brief History of Computers
1.4   The Evolution of the Intel x86 Architecture
1.5   Embedded Systems
1.6   Arm Architecture
1.7   Cloud Computing

Chapter 2 Performance Issues                                     Week 2
2.1   Designing for Performance
2.2   Multicore, Mics, and GPGPUs
2.3   Two Laws that Provide Insight: Ahmdahl's Law and Little's Law
2.4   Basic Measures of Computer Performance
2.5   Calculating the Mean
2.6   Benchmarks and Spec

Chapter 9 Number Systems (CPE 201 Review)                              Week 3
9.1   The Decimal System
9.2   Positional Number Systems
9.3   The Binary System
9.4   Converting Between Binary and Decimal
9.5   Hexadecimal Notation

Chapter 10 Computer Arithmetic
10.1  The Arithmetic and Logic Unit
10.2  Integer Representation
10.3  Integer Arithmetic
10.4  Floating-Point Representation
10.5  Floating-Point Arithmetic

Chapter 11 Digital Logic (CPE 201 Review)
11.1  Boolean Algebra
11.2  Gates
11.3  Combinational Circuits
11.4  Sequential Circuits
11.5  Programmable Logic Devices


Chapter 12 Instruction Sets: Characteristics and Functions        Week 4
12.1  Machine Instruction Characteristics
12.2  Types of Operands
12.3  Intel x86 and ARM Data Types
12.4  Types of Operations
12.5  Intel x86 and ARM Operation Types
Appendix 12 A Little-, Big-, and Bi-Endian
```

Chapter 13 Instruction Sets: Addressing Modes and Formats
13.1 Addressing Modes
13.2 x86 and ARM Addressing Modes
13.3 Instruction Formats
13.4 x86 and ARM Instruction Formats
13.5 Assembly Language

Appendix B Assembly Language and Related Topics Weeks 5 - 8
B.1 Assembly Language
B.2 Assemblers
B.3 Loading and Linking
B.4 Key Terms, Review Questions, and Problems

Mid-Term Exam Week 9

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 Point-to-Point Interconnect
3.6 PCI Express

Chapter 14 Processor Structure and Function Week 10
14.1 Processor Organization
14.2 Register Organization
14.3 Instruction Cycle
14.4 Instruction Pipelining
14.5 The x86 Processor Family
14.6 The ARM Processor

Chapter 15 Reduced Instruction Set Computers
15.1 Instruction Execution Characteristics
15.2 The Use of a Large Register File
15.3 Compiler-Based Register Optimization
15.4 Reduced Instruction Set Architecture
15.5 RISC Pipelining
15.6 MIPS R4000
15.7 SPARC
15.8 RISC versus CISC Controversy

Chapter 4 Cache Memory Week 11
4.1 Computer Memory System Overview
4.2 Cache Memory Principles
4.3 Elements of Cache Design
4.4 Pentium 4 Cache Organization
Appendix 4A Performance Characteristics of Two-Level Memories

Chapter 5 Internal Memory Week 12
5.1 Semiconductor Main Memory
5.2 Error Correction
5.3 DDR DRAM
5.4 Flash Memory
5.5 Newer Nonvolatile Solid-State Memory Technologies

Chapter 6 External Memory
6.1 Magnetic Disk
6.2 RAID
6.3 Solid State Drives
6.4 Optical Memory
6.5 Magnetic Tape

Chapter 7 Input/Output Week 13
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 Direct Cache Access
7.7 I/O Channels and Processors
7.8 External Interconnection Standards
7.9 IBM zEnterprise EC12 I/O Structure

Chapter 8 Operating System Support Week 14
8.1 Operating System Overview
8.2 Scheduling
8.3 Memory Management
8.4 Intel x86 Memory Management
8.5 Arm Memory Management

Chapter 16 Instruction-Level Parallelism and Superscalar Processors
16.1 Overview
16.2 Design Issues
16.3 Intel Core Microarchitecture
16.4 ARM Cortex-A8 596 16.5 ARM Cortex-M3

Chapter 17 Parallel Processing Week 15
17.1 Multiple Processor Organizations
17.2 Symmetric Multiprocessors
17.3 Cache Coherence and the MESI Protocol
17.4 Multithreading and Chip Multiprocessors
17.5 Clusters
17.6 Nonuniform Memory Access
17.7 Cloud Computing

Chapter 18 Multicore Computers
18.1 Hardware Performance Issues
18.2 Software Performance Issues
18.3 Multicore Organization
18.4 Heterogeneous Multicore Organization
18.5 Intel Core i7-990X
18.6 ARM Cortex-A15 MPCore
18.7 IBM zEnterprise EC12 Mainframe

Student Participation and Course Policies:

The course will contain two basic and interrelated blocks. First, the textbook will provide the framework for the course. Second, as material is reached in the textbook it will be related to supplementary material covering advanced computer topics.

Students are expected to attend all classes and read all of the assigned sections of the textbook. Often, material will not be covered in both lectures and reading assignments. Thus, both are essential to a full understanding of the course content.

During many classes a short example problem (class exercise) related to the current topic will be assigned. Students will spend a few minutes working alone on this problem followed by a few minutes discussing their solutions with two or three other students.

These solutions will be collected and used as a basis for up to 5% extra credit (5% for all class exercises and 2.5% for 50% submission etc.) for the course grade.

Also, completion of homework is essential. Homework will be due on WebCampus in .pdf format on (OR BEFORE) the date and time posted with the homework assignment. If your HW is one second late for whatever reason – it is late.

NOTE: All Homework must be submitted as a .pdf file made from a text or WORD file. Complex equations and diagrams can be submitted as hand drawn figures (or scanned images) inserted in spaces in the print out. However, everything else must be typed. Please be sure to keep a copy of anything you submit for homework or Lab.

LATE HOMEWORK WILL BE ACCEPTED FOR AT MOST 50% CREDIT.

Late homework will normally be accepted for up to **two weeks** after the original due date unless special circumstances justify more time. However, **all work must be submitted no later than May 1**.

Students are expected to attend, and be on time, for every class. This demonstrates professionalism and consideration for your fellow students and your Instructor. While the course does not have an attendance policy, students who miss class and/or are late for class may experience an impact on their grade by missing classroom activities. Likewise, if you attend a class you are expected to remain for the entire class period unless you obtain prior approval to leave early.

UNR Athletics: If you are involved with any university-sponsored athletic activities that will have an impact on your attendance, please provide your Instructor with a letter from your coach and/or the IINR Athletic Department as soon as possible, but no later than the end of the second week of classes. This should include the official schedule of your activities which will impact your attendance throughout the semester.

The instructor reserves the right to add to, and/or modify any of the above policies as needed to maintain an appropriate and effective educational atmosphere in the classroom and the laboratory. in the case that this occurs, all students will be notified in advance of implementation of the new and/or modified policy.

Students are encouraged to study together, but each person must prepare his or her solutions and have a firm understanding of any work turned in. When you put your name on your homework you are stating that it is your own work and not the work of another person. As a reminder of UNR academic standards, please read from the ***UNR on-line Catalog, University Code of Conduct and Policies, POLICIES AND GUIDELINES, ACADEMIC STANDARDS***
http://catalog.unr.edu/content.php?catoid=13&navoid=3708 defining these standards.

Specifically, the following: "*Plagiarism is defined as submitting the language, ideas, thoughts or work of another as one's own; or assisting in the act of plagiarism by allowing one's work to be used in this fashion*." This means that if another student asks to borrow your work to copy - JUST SAY NO - or you are participating in plagiarism. The following sanctions are listed in the above reference.

"Subsection B: Sanctions for Violation of Academic Standards

- An undergraduate or graduate student found responsible for violating this policy may not withdraw from the course in question and may not utilize the "grade replacement or grade appeals policies" for that course.
- Sanctions for violations of University academic standards for academic dishonesty may include academic and/or disciplinary sanctions. Academic sanctions for both undergraduate and graduate students may include: filing a final grade of "F"; reducing the student's final course grade one or two full grade points; awarding a failing mark on the coursework in question; or requiring the student to retake or resubmit the coursework.
- Sanctions for violations of University academic standards for class misconduct may include disciplinary sanctions. Academic sanctions may include temporary removal from the classroom by the faculty member or being dropped from the class. Dropping a student from a class must be approved by the Dean.
- An undergraduate or graduate student may also be subject to discipline for academic dishonesty pursuant to the provisions of the University's Student Code of Conduct. Disciplinary sanctions for both undergraduate and graduate students may include the following: warning, reprimand, restitution, probation, suspension, expulsion, or revocation of degree."

Course Grade Structure

Each course activity will contribute to the course grade as shown below. All activities will be graded on a scale of 0-100 points, and the final course grade will be determined as shown below.

STUDENTS MUST HAVE AN AVERAGE EXAM SCORE OF 50 OR GREATER IN ORDER TO PASS THE COURSE

All exams given in this course will be closed notes and closed books. Only calculators and materials handed out at the time of the exam may be used. Normally, plus/minus grades are not given in this class. The instructor reserves the right to assign plus/minus grades under special circumstances involving borderline grades based upon class participation. Your grade will never be lower than defined here unless you have an excessive number of un-excused absences from class, however, positive class participation can be used as a basis for raising your grade.

HOMEWORK	30%
MID TERM EXAM	30%
COMPREHENSIVE FINAL EXAM	40%
= COURSE GRADE	100%

90 - 100 points = A | 80 - 89.9 points = B | 65 - 79.9 points = C | 50 - 64.9 points = D | 00 - 49.9 points = F

Learning Outcomes

Learning Outcome A: Students will be able to describe the structure and functioning of a digital computer, including its overall system architecture, operating system, and digital components.

Learning Outcome B: Students will be able to explain the generic principles that underlie the building of a digital computer, including data representation, digital logic and processor programming.

Learning Outcome C: Students will be able to apply some fundamental coding schemes.

Learning Outcome D: Students will be able to present and discuss simple examples of assembly language appropriate for an introductory course.

Course Outcomes:

The course outcomes are skills and abilities students should have acquired by the end of the course. These outcomes are defined in terms of the Computer Science and Engineering ABET Accreditation Program outcomes which are relevant to this course. All outcomes are listed below and those relevant to this course are identified in the following Table.

1. an ability to apply knowledge of computing, mathematics, science, and engineering.
2. an ability to design and conduct experiments, as well as to analyze and interpret data.
3. an ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs, within realistic constraints specific to the field.
4. an ability to function effectively on multi-disciplinary teams.
5. an ability to analyze a problem, and identify, formulate and use the appropriate computing and engineering requirements for obtaining its solution.
6. an understanding of professional, ethical, legal, security and social issues and responsibilities.
7. an ability to communicate effectively with a range of audiences.
8. the broad education necessary to analyze the local and global impact of computing and engineering solutions on individuals, organizations, and society.
9. a recognition of the need for, and an ability to engage in continuing professional development and life-long learning.
10. a knowledge of contemporary issues.
11. an ability to use current techniques, skills, and tools necessary for computing and engineering practice.
12. an ability to apply mathematical foundations, algorithmic principles, and computer science and engineering theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices.
13. an ability to apply design and development principles in the construction of software systems or computer systems of varying complexity.

Program Outcomes	Course Outcomes	Course Strategies & Actions
3	Students demonstrate that they understand the building blocks of modern computers, including the feature trade-offs involved with different modern architectures.	Study of both Princeton and Harvard Architectures. Examination of components within the CPU. Homework and exam questions covering these topics.
9	Students demonstrate that they understand how computer architectures change and evolve as the technology and market change.	Study of how different CPU features support different computer applications. Homework requiring Web searches for details of current implementations.
10	Students demonstrate that they can learn the assembly language for more than a single CPU architecture.	Homework requiring specific program functions. Examination of assembly language for both Intel 8086 CISC and ARM 7 RISC processors.