BOB JONES UNIVERSITY
EST. 1927

College of Arts & Sciences

CpS 310

Microprocessor Architecture

Fall 2024

Instructor:	Jordan Jueckstock
Office:	AL 76 *(3rd floor, left hall)*
Office Hours:	10:00-10:50am, weekdays
Email:	jpjuecks@bju.edu

Course Description:

Study of the basic microprocessor architecture focusing on the fetch-decode-execute cycle. Project involves writing a program which simulates the workings of a microprocessor including instruction decoding, addressing techniques, interrupt processing, etc. Discussion of RISC and CISC philosophies. *Prerequisite(s): CpS 209, CpS 230. 3 Credits.*

Course Context:

This course fulfills the following objectives of the Computer Science department:

- CS1: Design and implement efficient solutions to problems in various domains.
- CS2: Demonstrate understanding of fundamental concepts in computer science, including:
  - Language translation
  - Limitations of computers
  - Stored program (a.k.a. von Neumann) architecture
  - Memory hierarchy

Course Goals:

This course will help students to develop:

- Elementary understanding of CPU implementation details
- Detailed understanding of how machine-language programs produced by compilers are loaded and executed by a CPU
- Appreciation for and the ability to measure performance impacts caused by the memory hierarchy and other significant computer system "implementation details"

Course Textbook (Required):

Computer Systems: A Programmers Perspective (3rd Edition), by Bryant & O'Hallaron. Published by Pearson, 2016. ISBN-13: 978-0134092669

Schedule:

The anticipated schedule of lecture topics, class milestones, and due dates is maintained on the CpS 310 website at https://protect.bju.edu/cps/courses/cps310/schedule/.

Assignments:

Tests: multiple-choice/short-answer exams gauging understanding of core class concepts
Quizzes: in-class electronic quizzes gauging understanding of significant/challenging topics
Homework: take-home written exercises reinforcing key non-programming concepts/skills
The Project: a large, semester-long project in which each student will demonstrate personal understanding of computer architecture, and software development competency, by designing and implementing an instruction-level ARM7TDMI CPU simulator ("ArmSim")

Since the project is the central graded artifact of the class, and since it provides a significant software design and implementation experience for students, students are required to reach specific minimal functionality milestones (e.g., the *baseline test suite* documented in the project documentation) to achieve specific letter grades.

Grading			
#	*Category*	*Pts.*	*Total*
7	Quiz / Homework	10-40	200
3	Tests	100	300
4	Project Milestones		500
	Total		**1000**

Scale	
A	900+***
B	800-899**
C	700-799*
D	600-699
F	0-599

- For an "A" (***): project ***must*** pass baseline test suite *and* include **2** extra deliverables
- For a "B" (**): project ***must*** pass baseline test suite *and* include **1** extra deliverable
- For a "C" (*): project ***must*** pass baseline test suite in addition to 700+ points earned
- **The maximum class grade for students not passing the baseline test suite is a D.**

Beyond the major thresholds mentioned above, grades are computed on a simple 10-point scale (see below) based on points earned out of 1000. Grades are not rounded up (or down—which probably should go without saying). Instead, all students are allotted 5 bonus "grace points" (which have the effect of rounding up, e.g., 695 to 700). The instructor reserves the right to confiscate these grace points, at his sole discretion and at any time, for repeated (or egregious) displays of disrespect to either the instructor or fellow students. *(Students who lose their grace points will be informed as soon as possible.)*

Deadlines / Late Work:
See the standard department late policy: https://cs.bju.edu/academics/policies/late-work-policy/. Note that a "free late" is an *earned privilege*, generally reserved for students who (a) have already established a track record of timely, quality submissions, (b) are experiencing an unanticipated/unavoidable scheduling hardship, and (c) proactively talk to the instructor about the situation before the posted due date.

The instructor reserves the right to change assignment due dates as deemed necessary. Assignments are due, electronically, by 11:59 pm of the date posted in the course schedule unless otherwise noted.

Due to grading constraints during finals week, the instructor reserves the right to shorten the late period for end of semester projects.

Accommodations:
Students needing accommodations due to a learning disability (visual, auditory, etc.) should provide an accommodation form obtained from the Academic Resource Center as soon as possible. Accommodations cannot be given without a form provided by the Academic Resource Center.

Getting Help:
Students struggling with an assignment or concepts in the class are strongly encouraged to ask the instructor for assistance either:

- in class *(usually best; someone else in class often has the same problem!)*
- before / after class
- during office hours (walk-in or by appointment)
- via email
- via Teams discussions

Academic Honesty and Integrity Policy:
See the CpS department policy: https://cs.bju.edu/academics/policies/academic-integrity-policy/.

The short version is:

- Never look at (or listen to verbal descriptions of, or touch 3D-printed representations of, or…) other students' code *unless* that code is *broken* (i.e., doesn't work) and you are the one helping them diagnose the problem
- Never write code with another student helping/coaching/shoulder-surfing you (personal help sessions are for discussing ideas and sketching diagrams, not writing production code)
- All helps received (from persons or articles or videos or whatever) must be documented (name/reference, kind of help received, time spent getting help) in your report *[caveat: standard language/library documentation, class notes/examples, or time spent with the instructor do not need to be included in "help received"]*

Students are expected to be familiar with this policy and its impact on student collaboration from experience in prior CpS classes taken at BJU. Students lacking such experience should contact the instructor *immediately* to make sure they understand the standards to which they will be held and to which they should hold other students with whom they interact.

Since the goal of the assignments in this course is to help students personally develop the skills covered and not merely to complete the tasks assigned, and since the use of AI to complete (or jumpstart) tasks effectively defeats that goal, you may not use generative AI tools (i.e. Chat GPT, Bing Chat, Google Bard, GitHub Copilot, etc.) in this course for any assignment without the professor's express permission.

Whenever granted permission to use generative AI tools for an assignment, students are required to document the help so received in the usual manner (source, purpose, and extent). They must additionally submit full transcripts of their AI tool interactions (i.e., all prompts entered, and all artifacts produced) along with the normal submission artifacts (code, report, etc.).

Copyright Policy: