CS 351: Computer Architecture, Spring 2017

Instructor	Dr. Suzanne Rivoire
Meeting times	MoWe 2:00–3:50 PM, Ives 79
Drop-in office hours	MoWe 4–5 PM We 12–1:45 PM *Office hours are in Darwin 116F. Please knock if the door to 116 is closed.*
Textbook	[required] Patterson and Hennessy, *Computer Organization and Design: ARM Edition*, ISBN 9780128017333 **The edition is really important!**
Prerequisites	Grades of C- or better in CS 215 and CS 252, or consent of instructor.

Catalog description

(4 units) Lecture, 4 hours. Instruction set design; stages of instruction execution, data and control path design; CISC, RISC, stack architectures; pipelining; program optimization techniques, memory hierarchy: cache models and design issues, virtual memory and secondary storage; I/O interfacing; advanced topics to include some of the following: parallel architectures, DSP or other special purpose architecture, FPGA, reconfigurable architecture, asynchronous circuit design.

Course Goals

The major goals of this course are for you to

1. Understand the mechanics of how hardware and system software execute the programs that you write.
2. Understand software and hardware's contributions to the performance, reliability, and energy efficiency of your programs and systems.

For a list of detailed objectives that will be used to assess whether or not you have met these goals, visit http://rivoire.cs.sonoma.edu/cs351/objectives.html. You can also use that list as an exam study guide.

Prerequisites

Grade of C- or better in both CS 215 and 252.

Students who do not meet these prerequisites will need instructor consent to remain in the course.

Consolidated Syllabus

You may download the course description, objectives, syllabus, and schedule in a consolidated pdf: http://rivoire.cs.sonoma.edu/cs351/syllabus_consolidated.pdf

Exam dates

Exam 1:	Feb. 27 (Mon.)	In lecture
Exam 2:	Apr. 5 (Wed.)	In lecture

Exam 3 (final): May 15 (Mon.) 2:00–3:50 PM

Students who have scheduling conflicts on these dates should contact the instructor at the beginning of the semester.

Coursework and Grading

Course Activities

Lecture and Reading

The tentative course schedule shows the topics to be covered. Students are expected to attend all lectures and to get the notes from another student if absent. Students are also expected to skim the assigned reading material before each lecture and read more fully after the lecture.

In-class Activities

In-class activities, including quizzes, will be given almost every lecture. Students' lowest 3 scores on these activities will be dropped from the grade calculation. These activities cannot be made up.

Homework problem sets

Approximately 6 homework problem sets will be assigned. You may work in groups of up to three students and submit a single solution set for the group.

Projects

In addition to the problem sets, students will complete projects to further explore the topics covered in class. You may work in groups of up to three students on the projects.

You are not required to work with the same group for the entire semester. You are permitted to do each homework set/project with a different group. However, you are strongly encouraged to work with at least one other student on each assignment.

Exams

Three exams will be given, with the third during the scheduled final exam time. The exams cover the material from lecture, homework, projects, and the textbook. Exams will emphasize recent material, although you are responsible for knowing previous material as well. You may bring one 8.5 by 11-inch handwritten sheet of notes to all exams.

Makeup exams will be given only in extraordinary circumstances.

Grading Policies

Grade breakdown

Exams	45%
Homework problem sets	30%
Projects	15%
Class activities	10%

Your final semester grade will be rounded to the nearest integer.

Cutoffs for letter grades (after rounding)

93	90	87	83	80	77	73	70	67	63	60	0
A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F

CS majors must take this course for a letter grade.

Up to 3% may be added to your final grade at the instructor's discretion for constructive participation in the class. Constructive participation includes in-class participation; asking good questions via email or during office hours; and doing outstanding or extra work on assignments. No other adjustments of borderline grades will be considered.

Late policy

Late homework problem sets: No late problem sets will be accepted. This policy allows solutions to be distributed in time for you to study for exams.

Late projects: If you miss a project due date, you may submit the project by the beginning of the next class session with no penalty. This is the only extension that will be given for minor emergencies.

Regrade policy

Regrade requests will be accepted up to 7 days after an assignment or exam is returned. The reason for the regrade request must be explained *in writing* and submitted as a hard copy along with the assignment or exam to be regraded. Note that all regrade requests, except for those pointing out mistakes in the totaling of points, will cause the *entire* assignment or exam to be regraded. The adjusted grade may therefore be higher or lower than the initial grade.

Attendance Policy

Your attendance is highly encouraged, and absence from class can affect your grade in the following ways:

- You may miss valuable material in lecture and will need to get notes from another student.
- You may miss graded activities or exams, which can only be made up under extraordinary circumstances.
- A pattern of poor attendance will make it difficult to earn the constructive participation bonus on your final semester grade.

Collaboration Policies

Special note for group work

Your work is the collective responsibility of your group: you will all get the same grade for the assignment, and you will all be held responsible for any violation of the course collaboration policy in the work you submit.

If you start working with a group on a particular assignment but are no longer comfortable sharing this credit or responsibility with one or more of your groupmates, please let me know as soon as possible.

Project and Homework Assignment Collaboration Policy

Academic misconduct is taken very seriously in this course. For each homework assignment or class project, you must work with at most *one group* of up to 3 students.

The work you turn in must be the sole work of your group members. You may discuss ideas and approaches with other students and the instructor, but you should work out all details and write up all solutions with your group.

The following actions will be penalized as academic dishonesty:

- Copying part or all of another group's assignment
- Copying old or published solutions
- Looking at another group's work or discussing another group's work in great detail. You will be penalized if your solution matches another group's solution too closely.
- Showing your group's work or describing your work in great detail to anyone other than your group members or the instructor.

Exam Collaboration Policy

Exams must be your own work. You are allowed to consult only your own brain, your 8.5x11" handwritten cheat sheet, and other materials specifically permitted by the instructor. Quiz policies will vary and will be announced when the quiz is given. On both exams and quizzes, giving or receiving unpermitted aid will be penalized as academic dishonesty.

Penalties for Academic Dishonesty

Academic dishonesty will be severely penalized; at a *minimum*, you will receive a grade of 0 on the assignment. For more information, see SSU's cheating and plagiarism policy (http://www.sonoma.edu/UAffairs/policies/cheating_plagiarism.htm) and the Dispute Resolution Board website (http://www.sonoma.edu/senate/committees/drb/drb.html).

Course and University Resources

Online Resources

Website

- The course homepage is http://rivoire.cs.sonoma.edu/cs351/.
- The schedule page (http://rivoire.cs.sonoma.edu/cs351/schedule.html) will be regularly updated with links to assignments.
- The resources page (http://rivoire.cs.sonoma.edu/cs351/resources.html) will be updated with links to software tools and helpful resources.

Moodle Gradebook

The course gradebook will be kept on Moodle (http://moodle.sonoma.edu) so that you can check your grades and compute your average at any time. Grades will be posted to Moodle shortly after assignments are returned.

Email List

Course announcements will be sent to your SSU email address, so you should check your email frequently.

University Resources

Disability Accommodations

If you are a student with a disability and you think you may require accommodations, please register with the campus office of Disability Services for Students (DSS), located in Salazar Hall - Room 1049, Phone: (707) 664-2677, TTY/TDD: (707) 664-2958. DSS will provide you with written confirmation of your verified disability and authorize recommended accommodations. This authorization must be presented to the instructor before any accommodations can be made. Visit http://www.sonoma.edu/dss for more information.

University Policies

There are important University policies that you should be aware of, such as the add/drop policy, cheating and plagiarism policy, grade appeal procedures, accommodations for students with disabilities, and the diversity vision statement. Go to this URL to find them: http://www.sonoma.edu/uaffairs/policies/studentinfo.shtml.

CS 351 Learning Objectives

Note: this course's goals and objectives are based on the Association for Computing Machinery (ACM)'s Computing Curricula 2013, mostly in the knowledge area of Architecture and Organization.

Exam 1: Introduction; Metrics; Assembly language

Introduction (Sec. 1.1–1.5)

- Identify the major domains of computing (embedded, desktop, server, etc.).
- Based on the requirements of each of these domains, analyze the appropriateness of different hardware and technologies for each domain.
- Define and explain vocabulary related to program translation and execution, such as *compiler*, *assembler*, *assembly language*, and *machine language*.
- Define vocabulary related to data storage, such as *bit*, *byte*, *KB*, *MB*, *GB*, and *TB*.
- Explain (at a high level) the interactions among I/O devices, the processor, and memory.
- Classify specific devices as I/O, processor, or memory.

Metrics (Sec. 1.6–1.10)

Metrics: Processor-specific metrics

- Apply the classic (single-core) processor performance formula.
- Apply the classic processor power and energy consumption formulas.
- Explain the implications of the "power wall" in terms of further processor performance improvements and the drive towards harnessing parallelism.

Metrics: Evaluation

- Evaluate performance in terms of latency and throughput/bandwidth, and determine which metrics are appropriate for different uses.
- Express relative performance in terms of *speedup*.
- Explain the circumstances in which a given system performance metric is useful.
- Explain the inadequacies of benchmarks as a measure of system performance.
- Assess systems' performance, power, and cost in a given situation.
- Critically examine claims about systems' performance, power, and cost.

Assembly (Ch. 2.1–2.9, 2.12–2.14)

- Summarize how instructions are represented at both the machine level and in the context of a symbolic assembler.
- Show how fundamental high-level programming constructs are implemented at the machine-language level.
- Explain different instruction formats, such as addresses per instruction and variable-length vs. fixed-length formats.
- Explain how subroutine calls are handled at the assembly level.
- Write simple assembly language program snippets using...
    - arithmetic operations
    - logical operations
    - memory operations
    - conditional operations
    - functions and the call stack
    - string processing and manipulation

Exam 2: Processor implementation (basic and pipelined); AMAT and locality

Basic Processor Implementation (Ch. 4.1–4.4)

- Explain the organization of the classical von Neumann machine and its major functional units.
- Describe how an instruction is executed in a classical von Neumann machine.
- Trace the execution of instructions and programs on a simple processor implementation.
- Modify the simple processor implementation to accommodate additional instructions.
- Compare alternative implementations of datapaths.

Pipelined Processors (Ch. 4.5–4.9)

- Trace the execution of instructions and programs on pipelined processor implementations.
- Quantitatively compare the performance of programs on pipelined and non-pipelined processors.
- Explain basic instruction level parallelism using pipelining and the major hazards that may occur.
- Explain the concept of branch prediction and its utility.
- Quantitatively evaluate the performance of programs on systems that use mechanisms like forwarding and branch prediction to mitigate these hazards.

Exam 3: Memory hierarchy and virtual memory; I/O; parallelism

Caches (Ch. 5.1–5.4)

- Describe how the use of memory hierarchy is used to reduce the effective memory latency.
- Analyze the spatial and temporal locality of different programs.
- Compute Average Memory Access Time under a variety of cache and memory configurations and mixes of instruction and data references.
  Quantitatively and qualitatively reason about the effects of cache block size, mapping scheme, replacement policy, and write policy on the performance and complexity of the hardware.
- Trace the cache state, including the mapping of memory addresses to cache blocks, for a pattern of memory references.

Virtual Memory (Ch. 5.6–5.8)

- Explain the workings of a system with virtual memory management.
- Trace the state of the page table and TLB, including the mapping of virtual to physical addresses, for a pattern of memory references.
- Calculate the required size of the page table for a given set of design decisions.
- Trace the steps in handling a page fault.
- Explain how VM allows protection and isolation.

I/O and Reliability (Ch. 4.9, 5.2, 5.5, 5.11, online readings)

- Evaluate I/O throughput and latency.
- Apply the classic MTTF equation.
- Describe data access from a magnetic disk drive.
- Apply the classic equation for magnetic disk performance.
- Explain (at a high level) how flash memory works.
- Compare the pros and cons of different storage types, and evaluate which is better for a given purpose.
- Explain how interrupts are used to implement I/O control and data transfers.
- Define *interrupts* and *polling*, compare their pros and cons, and evaluate which is better in a given situation.
- Define *DMA* and explain how it works.
- Quantitatively compare the performance and fault-tolerance of different approaches to RAID for different I/O access patterns.

Parallel Hardware and Software (Ch. 4.10, 5.10, 6)

- Define *speedup* and *efficiency*, and explain the notion of an algorithm's scalability in this regard.
- Describe the relevance of scalability to performance.
- Apply Amdahl's Law and Gustafson's Law and analyze their limitations and implications.
- Explain the need for cache coherence and trace a simple cache coherence protocol.
- Explain (at a high level) how GPUs work and what types of problems they are good at.

CS 351: Computer Architecture – Spring 2017 Course Schedule

Except for exam dates, all schedule information is tentative. The most recent version of the schedule is online at http://rivoire.cs.sonoma.edu/cs351/schedule.html.

	Monday	Wednesday
Week 1 Jan 23–Jan 27	Intro and syllabus CS 252 review "quiz"	Domains of computing Performance metrics *Project 1 assigned*
		Reading: Ch. 1.1-1.4
Week 2 Jan 30–Feb 03	Performance metrics Amdahl's Law	CPU performance Power and energy metrics *HW 1 assigned*
	Reading: Ch. 1.6, 1.10	*Reading: Ch. 1.7, 1.8*
Week 3 Feb 06–Feb 10	LEGv8 ISA intro Arithmetic operations Machine code	Machine code, continued Logical operations Conditionals
	Reading: Ch. 2.1-2.3; data representation notes; Ch. 2.5	*Reading: Ch. 2.5, 2.6, 2.7*
Week 4 Feb 13–Feb 17	Conditional operations Memory operands and the stack *HW 1 due; HW 2 assigned*	Memory
	Reading: Ch. 2.3, 2.7, 2.9; Optional: 2.14	
Week 5 Feb 20–Feb 24	Functions	LEGv8 catchup Processor implementation intro *HW 2 due*
	Reading: Ch. 2.8	*Reading: Ch. 4.1-4.2; Skim: 4.3*
Week 6 Feb 27–Mar 03	**EXAM 1**	The datapath *HW 3 assigned*
		Reading: Ch. 4.4
Week 7 Mar 06–Mar 10	Datapath and control path	Processor implementation review
	Reading: Ch. 4.4	
Week 8 Mar 13–Mar 17	*Spring break - no class*	
Week 9 Mar 20–Mar 24	Pipelining intro *HW 3 due; HW 4 assigned*	Pipelined implementation Data hazards and forwarding *Project 2 assigned*
	Reading: Ch. 4.5	*Reading: Ch. 4.7*
Week 10 Mar 27–Mar 31	Control hazards and branch prediction	Processor implementation review Memory hierarchy intro
	Reading: Ch. 4.8	*Reading: Ch. 5.1-5.2; pp. 412-416*
Week 11 Apr 03–Apr 07	Cache mapping schemes *HW 4 due; Project 2 statement of interest due*	**EXAM 2**
	Reading: Ch. 5.1-5.2; pp. 412-416	
Week 12 Apr 10–Apr 14	Cache block sizing *HW 5 assigned*	Cache write policies; cache review
	Reading: Ch. 5.3-5.4	*Reading: Write policy notes*
Week 13 Apr 17–Apr 21	Virtual memory intro	Virtual memory, continued
	Reading: Ch. 5.7; Virtual memory notes	
Week 14 Apr 24–Apr 28	Memory hierarchy catchup *HW 5 due; HW 6 assigned*	I/O: disks and flash
	Reading: Ch. 5.8	*Reading: Ch. 5.5, 5.2*
Week 15 May 01–May 05	RAID	RAID wrap-up Parallelism: introduction and metrics
	Reading: Ch. 5.11 (online)	*Reading: Ch. 6.1-6.2*
Week 16 May 08–May 12	Data-level parallelism and GPUs Thread-level parallelism *HW 6 due*	Cache coherence *Project 2 due*
	Reading: Ch. 6.3, 6.6	*Reading: Ch. 5.10*
Finals May 15–May 19	**EXAM 3: Monday 2:00 AM–3:50 PM**	