San José State University

Science/Computer Science
CS 147, Computer Architecture, Section 4, Spring 2026

Course and Contact Information

Instructor:	Sriram Rao
Office Location:	DH 282
Telephone:	
Email:	sriram.rao@sjsu.edu
Office Hours:	Tuesdays 12-1 and by appointment.
Class Days/Time:	Tue/Thu: 10:30-11:45p
Classroom:	Dudley Moorehead Hall 161
Prerequisites:	CS 47 or CMPE 102 or equivalent (with a grade of C-" or better).

Grader and Labs Information

Labs will be published here (https://github.com/Tiparium/CS_147_Handouts) .

Grader	Nainoa-Faulkner Jackson

Discord Handle	Tiparium
Email	nainoa.faulkner-jackson@sjsu.edu
Lab Hours	
Lab Location	Variable (Typically MLK Library)
Office Hours	To Be Determined
Class Discord Link	**Link to Discord Server (https://discord.gg/gd32VJykX2)**

Labs will require the following in order to function:

- Docker
- Github account (For getting the labs.)
- **Lab Files (https://github.com/Tiparium/CS_147_Handouts)**

Course Description

Introduction to the basic concepts of computer hardware structure and design, including processors and arithmetic logic units, pipelining, and memory hierarchy.

Course Learning Outcomes (CLO)

Computer architecture is the science and art of selecting and interconnecting hardware components to create a computer that meets functional, performance and cost goals. In this course, students will learn how to completely design a correct single processor computer, including processor datapath, processor control, memory systems, and I/O.

It describes how computers operate at a fairly low level of abstraction. For example:

- What are the components of a computer and how do they fit together?
- How do computers do arithmetic?
- Understand the functionality and operation of the basic elements of a computer system

including processor, memory and input/output
- Reason about first-order performance
- Understand the hardware/software interface.
- Understand MIPS processor design as it relates to the MIPS ISA.
- Understand and design components for a single processor in Verilog.

Understanding these fundamentals thoroughly is absolutely essential to your future success in computer science. The material of this course is quite detailed and requires careful and diligent study.

Upon successfully completing this course, students will have strong systems foundation about how microprocessors are designed and how they execute programs on a single computer.

Course Requirements and Assignments

Please see **Syllabus (https://sjsu.instructure.com/courses/1626056/assignments/syllabus)**

The **University Policy S16-9 (http://www.sjsu.edu/senate/docs/S16-9.pdf)** , Course Syllabi (http://www.sjsu.edu/senate/docs/S16-9.pdf) requires the following language to be included in the syllabus:

“Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction, preparation/studying, or course related activities, including but not limited to internships, labs, and clinical practica. Other course structures will have equivalent workload expectations as described in the syllabus.”

Grading Information

Please see **Syllabus. (https://sjsu.instructure.com/courses/1626056/assignments/syllabus)**

Classroom Protocol

This is your class. Please ask questions. Please come prepared. Do not engage in activity that may distract other students.

I do not take attendance except for the first two classes. Students not attending either of the first two classes will be dropped to make room for students on the waiting list. Attempting to get

marked as present (by have someone else attend in your place or using technological deceptions) will be considered academic dishonesty and at a minimum will result in you getting dropped from the course.

We will use discord channel/canvas discussion for announcements about programming labs as well as discussion topics. For faster turnarounds, please use the discord channel.

Check canvas on a weekly basis for schedules/reading assignments/due dates.

Course material developed by the instructor is the intellectual property of the instructor. Students ***should not*** publicly share or upload instructor generated material for this course such as exam questions, lecture notes, hands-on exercises or homework solutions without instructor permission.

This is my first time teaching this class. The schedule listed in the syllabus is tentative and may change based on student needs. Furthermore, the course project is new and we will learn by doing :-).

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' **Syllabus Information web page (http://www.sjsu.edu/gup/syllabusinfo/)** at http://www.sjsu.edu/gup/syllabusinfo/" Make sure to review these policies and resources.

Cheating/Academic Dishonesty

I take issues of Academic Dishonesty very seriously. ***Do not cheat***. ***Do not share code***. If we detect cheating in a programming assignment or an exam, you will get a **0** for that lab/exam. Repeat offense will likely lead to a **F** grade in the course.

All exams will be in-class. You are not allowed to use any phones during the exam.

Artificial intelligence (AI) tools like ChatGPT, Google Gemini, and GitHub Copilot are not permitted to be used as a replacement for the writing or problem-solving components of this class. SJSU's subscription to Turnitin has an AI-detection feature, and assignments that have been determined by that application or by other convincing evidence to have been written by AI in substantial fractions will receive an automatic zero. The incident will also be reported to the University as academic misconduct.

Acknowledgements

This course adapts (and also uses) materials from courses taught by Karu Sankaralingam, David Sinclair, Kaushik Patra. They have been adapted and modified by us with permission. As such please do not post any materials from this course on public websites.

Course Syllabus

Edit

Science/Computer Science

CS 147-04, Computer Architecture, Spring 2026

Required Texts/Readings

Textbook

The **required** text book for this class is **Computer Organization and Design MIPS Edition: The Hardware/Software Interface, 6th Edition (https://www.amazon.com/dp/0128201096?ref=ppx_yo2ov_dt_b_fed_asin_title)**. We will refer to this book as H&P in the schedule below. **(http://www.mypearsonstore.com/bookstore/computer-systems-a-programmers-perspective-plus-mastering-0134123832)**

Optional Reading

- Shing Kong, Revised by Milo Martin, The Elements of Logic Design Style
  Online: **http://www.cis.upenn.edu/~milom/elements-of-logic-design-style/ (http://www.cis.upenn.edu/~milom/elements-of-logic-design-style/)**
- Building a functional processor for a MIPS ISA: **Pedagogically Motivated and Composable Open-Source RISC-V Processors for Computer Science Education (https://arxiv.org/pdf/2509.20514v1)**

References for Verilog

- **Verilog Cheat Sheet (https://pages.cs.wisc.edu/~karu/courses/cs552/spring2021//handouts/misc/Verilog_cheat.pdf)**

Reading

To get the most out of this class, ***prior to each lecture***, students are encouraged to read the relevant chapter from the textbook.

Other technology requirements / equipment / material

Programming assignments will be a significant part of this course. We will build programs and use run them on Linux (such as, Ubuntu 22.04). Hence, access to a computer that runs Linux will be required.

Course Requirements and Assignments

I do not grade on a curve. The exams and projects measure what you are expected to have learned.

Course Project: There will be an individual course project which entails a complete functional design of a microprocessor. All components of your design will be written in Verilog.

Programming Labs: We will be doing **individual** programming assignments**. Individual programming assignments are not group projects.** If students get help on assignments, even to resolve a seemingly trivial problem, it must be documented in the code with the name of the person rendering the help and a brief description of the help provided. Extensive help on a project will result in a reduced grade. Failure to document help, or any other forms of cheating will result in a failing grade on the assignment at a minimum and may result in failure of the course. See **Integrity (http://info.sjsu.edu/static/schedules/integrity.html)** for more information. Even in open source, you cannot copy code from one open source project to another without attribution.

Programming assignments will be distributed via Github. Please create a Github account (if you don't already have one). We will use Gradescope for assignment submission and grading.

Help on Labs: Nainoa-Faulkner Jackson (course grader) will have weekly **in-person** "**lab hours (https://sjsu.instructure.com/courses/1595954)** " in to help you with the labs. Please seek his help whenever needed.

The **University Policy S16-9 (http://www.sjsu.edu/senate/docs/S16-9.pdf)** , Course Syllabi (http://www.sjsu.edu/senate/docs/S16-9.pdf) requires the following language to be included in the syllabus:

“Success in this course is based on the expectation that students will spend, for each unit of credit, a minimum of 45 hours over the length of the course (normally three hours per unit per week) for instruction, preparation/studying, or course related activities, including but not limited to internships, labs, and clinical practica. Other course structures will have equivalent workload expectations as described in the syllabus.”

Final Examination or Evaluation

This course will have a cumulative final exam given during exam week. Per **University schedule (https://www.sjsu.edu/classes/final-exam-schedule/fall-2024.php)** this is on Tuesday, May 14 from 10:45a-12:45p.

There will be two in-class exams given in the semester (the last being the final exam).

Grading Information

Determination of Grades

Grades will be calculated based on the individual project grades, the two mid-semester exams, the final, discussion participation. Briefly, the weighted distribution is,

Course Project	40%
Midterm 1	15%
Midterm 2	15%
Final	20%
Homework/Class Participation	10%

The grade distribution will be:

Percentage	**Grade**
93 and above	A
90-93	A-

87-90	B+
84-87	B
80-84	B-
77-80	C+
74-77	C
70-74	C-
67-69	D+
64-67	D
60-64	D-
59 and below	F

Classroom Protocol

This is your class. Please ask questions. Please come prepared. Do not engage in activity that may distract other students.

I do not take attendance except for the first two classes. Students not attending either of the first two classes will be dropped to make room for students on the waiting list. Attempting to get marked as present (by have someone else attend in your place or using technological deceptions) will be considered academic dishonesty and at a minimum will result in you getting dropped from the course.

Course material developed by the instructor is the intellectual property of the instructor. Students ***should not*** publicly share or upload instructor generated material for this course such as exam

questions, lecture notes, hands-on exercises or homework solutions without instructor permission.

This is my first time teaching this class. The schedule listed below is tentative and may change based on student needs. Furthermore, the programming project is new and we will learn by doing :-).

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' **Syllabus Information web page (http://www.sjsu.edu/gup/syllabusinfo/)** at http://www.sjsu.edu/gup/syllabusinfo/" Make sure to review these policies and resources.

Course Assignments

There are two types of assignments for this course:

1. **Weekly Written Homework:** These will be assigned on Tuesday and will be due before class the following Tuesday. Answers should be submitted on canvas. Late submissions will not be accepted. These will count towards class participation. The work you submit must be your work and your work alone.
2. **Programming Labs:** (More on this below). There will be 5 programming labs and a course project during the semester. You will be writing Verilog programs. Get familiar with the toolchain :).

Programming Labs/Project

This course project builds upon **individual** programming assignments**. Individual programming assignments are not group projects.** The programs you submit must be your work and your work alone.

Caveats:

1. A requirement is that you **do not** publish solutions on public GitHub repos.
2. The labs are known to be demanding and debugging can be time consuming. So, please get started early for each lab.

Early submission:

- Submissions that are at least 2 or more days ***earlier*** than the deadline will get a 10% bonus.

Late Submissions:

1. Submissions that are up to a week late will be penalized 30% of the score.
2. Submissions that are up to two weeks late will be penalized 50% of the score.
3. I will not accept submissions that are more than two weeks beyond the submission date. This will be considered as a non-submission.
4. At my discretion, non-submissions will likely result in a grade penalty.

Cheating/Academic Dishonesty

I take issues of Academic Dishonesty very seriously. ***Do not cheat***. ***Do not share code***. If we detect cheating in a programming assignment or an exam, you will get a **0** for that lab/exam. Repeat offense will likely lead to a **F** grade in the course.

Artificial intelligence (AI) tools like ChatGPT, Google Gemini, and GitHub Copilot are not permitted to be used as a replacement for the writing or problem-solving components of this class. SJSU's subscription to Gradescope has an AI-detection feature, and assignments that have been determined by that application or by other convincing evidence to have been written by AI in substantial fractions will receive an automatic zero. The incident will also be reported to the University as academic misconduct.

Preparing for the course

Start preparing for the course by doing the tutorials:

- Learn UNIX command line tools: **UNIX Tutorial (https://info-ee.surrey.ac.uk/Teaching/Unix/)**
- Learn C: **C tutorial (https://www.cs.hmc.edu/~rhodes/courses/cs134/sp19/readings/pointers.pdf)**

Course Schedule

(Tentative) Course Schedule

Week	Date	Topic/Theme	Reading(s) from H&P
1	1/22	Overview	1-1.5
2	1/27	Performance: Moore's law, Dennard Scaling, Amdahl's law	1.6-1.8, 1.11 and Moore's law paper

	1/29	Verilog Tutorial	Appendix B
3	2/3	MIPS ISA	2.1-2.10
	2/5	Arithmetic and Logic	3.1-3.2
4	2/10	Processor: Single cycle data path	4.1-4.3 and Appendix D
	2/12	Processor: Single cycle control path	4.4 and Appendix D
5	2/17	Processor Pipelining	4.6-4.7
	2/19	Processor Pipelining	4.6-4.7
6	2/24	Pipeline hazards	4.8-4.9
	2/26	**Midterm-I**	
7	3/3	Exceptions	4.10
	3/5	Cache Concepts	5.1-5.4, 5.8
8	3/10	Cache Design	
	3/12	Cache Performance	
9	3/17	Virtual Memory	5.6-5.7
	3/19	Virtual Memory	
10	3/24	Virtual Memory (review)	
	3/26	Main Memory & ECC	5.5, 5.11
11	3/31	**Spring break**	No class
	4/2	**Spring break**	No class
12	4/7	I/O	6.10 and Appendix A.8
	4/9	**Midterm-II**	
13	4/14	Processor (Superscalar)	4.11 and Appendix A.7
	4/16	Processor (Superscalar)	**MIPS R10K paper (http://ece552.ece.wisc.edu/mipsr10000.pdf)**

14	4/21	Parallel Processors	6.1-6.3 and 6.6
	4/23	Multithreading/Multicore	6.4-6.5 and 6.11
15	4/28	Advanced Arithmetic	3.3-3.5
	4/30	GPUs, Accelerators, etc.	6.6
16	**5/5**	**TBD**	
	5/7	**TBD**	
17	**5/14/26**	**Final Exam: 10:45a-12:45p**	