

CSE3666: Introduction to Computer Architecture

Spring 2024

Course information

Course Number and Title: CSE 3666 Introduction to Computer Architecture

Credits: 3

Format: In person for both lectures and labs

Prerequisites: CSE 2050 or 2100; open only to students in the School of Engineering and declared Computer Science minors. Not open for credit after passing CSE 2304, 4302, or 4950.

Course description

Structure and operation of digital systems and computers. Instruction sets and assembly language. Integer and floating-point arithmetic. Machine organization, control and data paths, pipeline, and the memory hierarchy.

Textbook

David A. Patterson and John L. Hennessy. Computer Organization and Design **RISC-V Edition**: The Hardware Software Interface (The Morgan Kaufmann Series in Computer Architecture and Design) **2nd Edition**, Morgan Kaufmann. Dec 2020. ISBN-13: 978-0128203316.

Course Goals and Objectives

1. Represent data (e.g., integers, floating-point numbers, instructions.) with bits.
2. Program with RISC-V instructions (computing with instructions).
3. Design an instruction set architecture (ISA). For example, describe types of instructions, encode/decode instructions, and identify and apply good design principles.
4. Design digital circuit (e.g., arithmetic-logic unit) with logic gates.
5. Design a processor that supports a specified ISA.
6. Design a processor pipeline and handle data and control hazards.
7. Design memory cache in a computer system.
8. Evaluate the performance of a computer system.

The course schedule will be posted online separately.

Syllabus information may be subject to change, with the exception of materials for purchase.

Instructors and TAs

Caiwen Ding, caiwen.ding@uconn.edu

Z. Jerry Shi, zshi@uconn.edu

TAs

Amogh Garuda Dwajan,	Giovanni Romano,
Jacob Gerow,	Joan Tejera,
Maksym Haydamakha,	Michael Swan,
Sakeena Aimandi,	Chinmaey Sharad Shende,
Dhruv Rao,	Hongwu Peng,
Kiran Thorat,	and Sheetal Sudhir Srilatha.

They can be found on Discord.

Office Hours:

TBD. The times will be posted in an announcement in HuskyCT.

Discord Q&A:

TAs will be monitoring questions on Discord and try their best to have all questions answered promptly.

Lectures and labs

Lectures and labs are in person. Students are expected to attend lectures and labs in person.

Lectures are not streamed nor recorded.

Lecture Section 001-LEC: AUST 108, Tue and Thur, 3:30PM - 4:45PM, Caiwen Ding

Lab 001L, Mon 2:30PM - 3:20PM, ITE 134

Lab 002L, Mon 3:35PM - 4:25PM, ITE 134

Lab 003L, Mon 1:25PM - 2:15PM, ITE 134

Lab 004L, Mon 12:20PM - 1:10PM, ITE 134

Lab 005L, Mon 11:15AM - 12:05PM, ITE 134

Lecture Section 021-LEC: AUST 110, Tue and Thur, 12:30PM - 1:45PM, Jerry Shi

Lab 021L, Mon 1:25PM - 2:15PM, E2 306

Lab 022L, Mon 2:30PM - 3:20PM, E2 306

Lab 023L, Mon 3:35PM - 4:25PM, E2 306

Lab 024L, Mon 12:20PM - 1:10PM, E2 306

Lab 025L, Mon 11:15AM - 12:05PM, E2 306

Grade

Students' final grade is based on the following components:

- Labs: 10%.
- Homework: 15%.
- Participation (labs and homework): 5%. Students who receive at least 70% on ALL mandatory assignments/questions are eligible to receive up to five points (5% of the weighted total) based on their performance on labs and homework assignments. Mandatory assignments/questions will be clearly marked and may have separate submission links.
 - 2%, if they receive 70% or more of the total Lab points.
 - 3%, if they receive 70% or more of the total Homework points.
- Exams: 70%. There are two midterm exams and the final exam. The lower of the midterm exam: 15%. The higher of the midterm exams: 20%. The final exam: 30%. The highest of three exams: 5%.

The final grade is based on the weighted total of components in the course. You can **estimate** your grade using the following table. The instructors reserve the right to adjust the thresholds. There is no “curve” in this course.

Weighted total	Letter Grade
≥ 93.00	A
Below A and ≥ 90.00	A-
Below A- and ≥ 87.00	B+
Below B+ and ≥ 83.00	B
Below B and ≥ 80.00	B-
Below B- and ≥ 77.00	C+
Below C+ and ≥ 73.00	C
Below C and ≥ 70.00	C-
Below C- and ≥ 67.00	D+
Below D+ and ≥ 63.00	D
Below D and ≥ 60.00	D-

Extra credit

Students who participate in lectures by answering iClicker questions can earn up to 2 points. Students earn 2 points if they have earned 60% of the points in iClicker and 1 point if they have earned 35% of the points. **There is no makeup for iClicker questions.** Students who have answered any iClicker questions remotely while not attending in-person lectures will receive 0 points.

Attendance

All lectures and labs are in person. **The lectures are not streamed nor recorded.**

Although attendance is not required to pass the course, students are strongly encouraged to attend lectures and labs. Students who miss lectures/labs should study provided materials themselves and get help through office hours or discussions on Discord.

Lab and homework assignments

Lab and homework assignments are submitted on HuskyCT or on Gradescope. In general, students must justify their answers, write concise comments in code, and make handwriting legible.

Some questions/assignments are graded automatically. The results are available shortly after submission. Students are advised to make attempts early and correct mistakes. **We do not adjust scores from the auto grader.**

For manually graded questions, we will make every effort to provide grades in one or two weeks. We may need more time for some assignments. If you have questions regarding the grading, you **MUST** contact either instructors or TAs within **ONE WEEK** after the graded work is returned to you (or to the class). Please check your grades often in HuskyCT.

Late Policy

Students should check HuskyCT frequently for open assignments and due dates. We have a lab and a homework assignment almost every week.

All due dates are specified in the assignments. The actual deadline is 11:55 PM Eastern Time on due dates. The deadline time set on submission pages is 11:59 PM. If the deadline on the submission page is not set accurately, we will correct them promptly.

The instructors reserve the right to change deadlines accordingly as the semester progresses. All changes will be communicated in an appropriate manner.

Occasionally some students may need an extension. We allow an automatic extension of two days on some lab and homework assignments. Deadlines without automatic extension will be marked as **firm deadlines** on an assignment. There is no limit on how many times a student uses the “automatic extension” and there is no penalty on submission made during “automatic extension”. The submission page is closed after the deadline (firm deadline or deadline plus automatic extension). **All lab tests deadlines are firm deadlines.**

Late submissions are not accepted, unless students have university accepted reasons, e.g., medical emergency, and provide evidence. Students must contact the instructor or TA before the deadline, or as soon as they can if they have an emergency. There may be a penalty for late submissions, for example, 10% of the maximum points for each additional day. There may be additional requirements, for example, presenting their work to instructors.

Please note that late submissions cannot be arranged sometimes, especially when doing so would slow down the progress of the class.

No submissions are accepted if solutions to the assignment have been posted.

Midterm exams

Two midterm exams are scheduled in the evening hours. Students who cannot take the exam at the scheduled time should contact instructors and TAs early to make arrangements.

In Spring 2024, Exam 1 is scheduled on Wed, 2/21/2024 and Exam 2 is on Wed, 4/10/2024.

Both exams start at 6:30pm in MCHU 102. Please mark your calendar.

Students who miss a midterm exam due to medical emergency should provide strong evidence or get approval from the Dean of Students Office (DOS) for makeup exams. The makeup exam is arranged only after students are fully recovered. It is not guaranteed that the makeup exam takes place before the semester ends. The assessment form may also change.

Important Information on Final Exams

Final exams are scheduled by the registrar's office, not by instructors. The final exam dates are posted in the student administration system around October 1 for the fall semester and around March 1 for the spring semester. Please check the dates when they are available.

Students are required to be available for exams during the scheduled time. Students must submit requests to the DOS Office before the deadlines if they need to reschedule their final exams. The DOS office will provide instructions thereafter.

According to the University policy, vacations, previously purchased tickets or reservations, weddings (unless part of the wedding party), and other large or small scale social events, are not viable excuses for missing a final exam. Contact the DOS office with any questions.

Weekly Time Commitment

Students should expect to dedicate at least six hours a week to this course, in addition to lectures and labs. This expectation is based on the various course activities, assignments, and assessments and the University of Connecticut's policy regarding credit hours. (More information related to hours per week per credit can be accessed at the Online Student website).

Students with Disabilities

The University of Connecticut is committed to protecting the rights of individuals with disabilities and assuring that the learning environment is accessible. Students who require accommodations should contact the Center for Students with Disabilities, Wilbur Cross Building Room 204, (860) 486-2020 or <http://csd.uconn.edu/>.

Students are strongly encouraged to contact CSD in the first three weeks of the semester.

Software/Hardware/Technical Requirements

The software/Hardware/technical requirements for this course include:

- HuskyCT/Blackboard
- Gradescope
- Github
- Discord, for discussions
- Adobe Acrobat Reader (or other PDF readers)
- Google Apps
- Microsoft Office (free to UConn students through uconn.onthehub.com)
- RARS (RISC-V Assembler and Runtime Simulator)
- Python 3.6+ and packages, for example, MyHDL
- A calculator
- A cell phone or other devices that can scan documents into PDF files
- iClicker, remote or subscription, for participating in lectures
- Internet access with a minimum speed of 1.5 Mbps (4 Mbps or higher is recommended)

NOTE: This course has NOT been designed for use with mobile devices.

For information on managing your privacy at the University of Connecticut, visit the [University's Privacy page](#).

Student Responsibilities and Resources

As a member of the University of Connecticut student community, you are held to certain standards and academic policies. In addition, there are numerous resources available to help you succeed in your academic work. Review these important [standards, policies and resources](#), which include:

- The Student Code, including academic integrity and resources on avoiding cheating and plagiarism
- Copyrighted Materials
- Credit Hours and Workload
- Netiquette and Communication
- Adding or Dropping a Course
- Academic Calendar
- Policy Against Discrimination, Harassment and Inappropriate Romantic Relationships
- Sexual Assault Reporting Policy

Academic Integrity

Academic integrity is a fundamental expectation of all students in this course. Cheating, plagiarism, and other forms of academic misconduct are not allowed in this course. Below is a list of common misconducts. Please notice that this is **not** a complete list. It is your responsibility to be familiar with the Student Code of Conduct and conduct yourself according to the standards that are described in the code.

- Represent the work from other people as your own. For example, copy answers from another student's work (e.g., examination sheet, homework, quiz, etc.), or obtain solutions from other people.
- Make use of notes during a closed book/closed notebook examination.
- Make use of electronic devices, e.g., cell phones, which are not allowed in an examination.
- Allow another student to take an examination in your place.
- Assist another student to violate academic integrity. For example, allow other students to copy your work, or share your answers with other students.

You may discuss the homework with anyone and use any reference material, provided you do not copy any other person's work or solutions from any sources. Appropriate reference or credit must be acknowledged if you use the results from other people.

For the programming questions, it is expected that you write every line of code that you submit (with the exception of code given out in class and simple online examples, typically a few lines, that demonstrate the usage of library functions or system calls). The following are examples of activities that are prohibited:

- Posting assignments in the course on the Internet and then copying answers.
- Giving code to another student (via email, printouts, photos, voice, etc.).
- Posting code in a publicly accessible location.

Evaluation of Course Experience

Students will be given an opportunity to provide feedback on their course experience and instruction using the University's standard procedures, which are administered by the [Office of Institutional Research and Effectiveness](#) (OIRE).

The University of Connecticut is dedicated to supporting and enhancing teaching effectiveness and student learning using a variety of methods. The Student Evaluation of Teaching (SET) is just one tool used to help faculty enhance their teaching. The SET is used for both formative (self-improvement) and summative (evaluation) purposes.

Additional informal formative surveys and other feedback instruments may be administered within the course.

Copyright

The following is the standard copyright language provided by UConn to instructors.

My lectures, notes, handouts, and displays are protected by state common law and federal copyright law. They are my own original expression and I have recorded them prior or during my lecture in order to ensure that I obtain copyright protection. Students are authorized to take notes in my class; however, this authorization extends only to making one set of notes for your own personal use and no other use. I will inform you as to whether you are authorized to record my lectures at the beginning of each semester. If you are so authorized to record my lectures, you may not copy this recording or any other material, provide copies of either to anyone else, or make a commercial use of them without prior permission from me.

Students should be aware that instructors' materials are protected by copyright regardless of whether such a statement appears in the syllabus.

More information is available at [Copyright and Teaching | UConn Library](#).

Success

UConn has many resources to help students succeed. Success in this course and in your major program depends heavily on your personal health and well-being. Recognize that stress is an expected part of the college experience, and it often can be compounded by unexpected setbacks or life changes outside the classroom. Your teaching assistants and instructors strongly encourage you to reframe challenges as an unavoidable pathway to success. Reflect on your role in taking care of yourself throughout the semester, before the demands of exams and projects reach their peak. Please feel free to reach out to us about any difficulty you may be having that may impact your performance in your courses or campus life as soon as it occurs and before it becomes too overwhelming. In addition to your academic advisor, we strongly encourage you to contact the many other support services on campus that stand ready to assist you.

- [Dean of Students Office](#)
- [Academic Achievement Center](#)
- [Student Health and Wellness](#)

CSE3666 Major Topics

Numbers, bits, bytes, and ASCII characters

Numbers in different representations: binary, two's complement, hexadecimal, and decimal numbers.

Conversion between different number systems.

Addition/subtraction of numbers in different number systems.

ASCII characters.

Two's complement numbers

Sign extension. Negate a two's complement number.

Bits

Bitwise logical operations.

RISC-V ISA

RISC-V instructions sets.

Main objectives

Describe how RISC-V supports operations in high-level programming languages: arithmetic, logical, memory access, control flow (if-else and loop). Particularly, describe how RISC-V supports if-else, loops, and functions.

Access data (as immediate, in register, and in memory) properly. Particularly, access data in word arrays and strings.

Describe how RISC-V instructions are encoded (transformed into machine code).

Describe how processors execute instructions in machine code (decoding machine code first).

Write programs with RISC-V instructions and in RISC-V assembly code. Memorize a set of core RISC-V instructions.

Describe limitations of individual RISC-V instructions (e.g., number of registers, immediate, branch offset, and offset in jump) and how RISC-V overcomes limitations with multiple instructions.

Describe the design principles of RISC (compared to CISC) and explain how it affects the design of RISC-V ISA.

Logical and arithmetic operations

Write RISC-V instructions to perform common operations.

Accessing Data

Describe how data and arrays (e.g., immediate, integers, addresses, characters, word arrays, byte arrays, and strings) are stored in computers.

Describe the range of immediate in RISC-V instructions.

Describe register file and register numbers/names.

Write RISC-V instructions to access data in registers and in memory. Properly specify addresses. Use proper instructions for data of different types (size and sign).

Explain how endianness affects the byte order when data are stored in memory.

Explain how data size, type (signed/unsigned), and endianness affect the result of load instructions.

Control flow

Program counter.

RISC-V support for if-else and loops.

Branch instructions.

RISC-V support for function calls.

RISC-V calling convention. Passing parameters to / returning values from functions. Caller-saved / callee-saved registers.

Stack. Push/pop. Save/restore registers on stack. Allocate storage space on stack. Explain how stack is used in functions.

Instruction Encoding/decoding

Six instruction formats (R, I, S, SB, U, UJ types).

Fields in different instruction formats.

Describe the placement of immediate bits in machine code, for different types of formats.

Describe how assembler place bits in immediate in machine code and how the processor construct immediate when executing instructions.

Given enough information, encode RISC-V instruction with 32 bits and read machine code (encoded instructions).

Assembly code

Explain directives for RISC-V assembly code (e.g., .data, .align, .text)

Describe and explain memory layout of a program.

Translate pseudocode or C code to RISC-V assembly code.

Read/debug RISC-V code.

Computer Arithmetic

Digital logic design

Construct truth table from specification. Write logic expressions from truth table. Some truth tables have don't care items, in either input or output.

Describe the general structure of a state machine.

Describe the function of commonly used modules (e.g., decoder, multiplexor, ALU, registers).

Implement digital circuits from logic expressions or diagrams, starting from basic gates or using existing modules.

Given a design of digital circuit, describe its behavior/function.

Analyze the timing of digital circuit.

Implement digital circuit in MyHDL.

Integer multiplication and division

Design multiplier and division module.

Describe the behavior of the circuit.

Explain how multiplication and division are done in hardware.

Code with instructions in RISC-V M-extension.

Floating point numbers

Convert real numbers between decimal and binary representation.

Normalized representation of decimal/binary numbers.

Represent numbers in half-precision, single-precision and double-precision formats.

Read half-precision, single-precision and double-precision numbers and represent the values in normalized representation or in decimal.

Describe RISC-V F/D-extension and write assembly code with instructions in the extensions. Load/store floating-point numbers. Compute with floating-point numbers (in floating-point registers). Convert between word and single/double with RISC-V instructions.

Processor

Computer performance

CPU Execution Time and the classic equation.

CPI. Clock cycle time. Clock rate. Speedup. Amdahl's law.

Compare the processor performance.

Single-cycle processor

Design a single-cycle RISC-V processor. Describe the design of every module in the diagram.

Describe how instructions are executed in the single-cycle design. Describe the hardware components required to support each kind of instructions.

Given an instruction and its address, find the signal values in the single-cycle design.

Improve the design to support more instructions. Describe changes in the design and explain why.

Pipeline

What, why, how questions. What is a processor pipeline, why, and how does it work (how can it be done? Particularly, describe the design of the 5-stage pipeline. Explain the data and control signals in pipeline registers. Describe how the pipelined processor executes instructions.

Describe the hazards in the pipeline.

Describe the method for dealing structural hazards in the 5-stage pipeline.

Describe the method for dealing data hazards in the 5-stage pipeline. Explain how the processor determines forwarding.

Explain the methods for handling control hazards. Always flush. Static prediction.

Explain how the processor detects hazards and stalls the pipeline.

Analyze data/control dependency in code.

Write pipeline diagrams.

Analyze the performance of a pipelined processor of different designs (e.g., with forwarding, without forwarding).

Schedule instructions to reduce hazards.

Cache

Concept of cache. What, why, and how. Temporal locality and spatial locality.

Direct-mapped cache

Fields in an address for cache access. The relation of cache configuration and the number of bits in each field.

Cache lookup. How bits in addresses are used in cache lookup.

Diagram of cache. Hardware cost of cache. Total number of bits in cache. Size of comparators (the number of bits). Number and size of MUXes.

Handling read and write. Write-through. Write-back. Write allocate. No Write allocate. Write buffer.

Given a sequence of memory references(requests) and cache contents, determine the outcome of each reference and update cache.

Cache performance

Memory stall cycles. Instruction cache. Data cache. Average memory access time (AMAT).

Impact of memory system on the performance of processors. Compare performance of processors under different configurations of cache.

Types of cache misses.

Methods for improving cache performance (cache size, block size, set associativity, and multi-level).
Impact cache size on block size on AMAT.