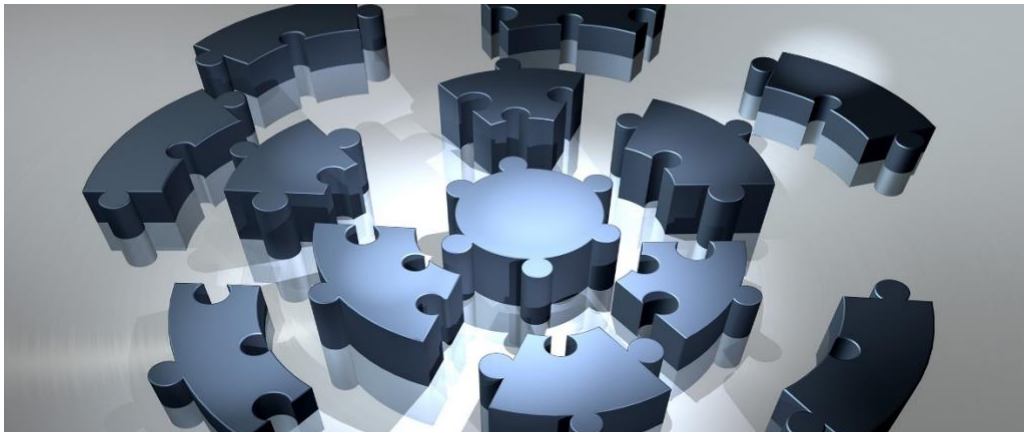

CSCE 312

All Sections

Spring 2022

Syllabus: Computer Organization

Course Description

The course objective is to integrate key notions from algorithms, computer architecture, compilers, and software engineering in one unified framework. This will be done constructively, by building a general-purpose computer system from the ground up. In the process, we will explore many ideas and techniques used in the design of modern hardware and software systems, and discuss major trade-offs and future trends. Throughout this journey, you will gain many cross-section views of the computing field, from the bare-bone details of switching circuits to the high-level abstraction of software design. The course consists of materials on the following topics: introduction to computer systems, data representation, machine language, processor architecture, memory hierarchy, and assembler. Several laboratory assignments will provide hands-on experience to many of the above topics.

Catalog Description

Introduction to computer systems from programmer's perspective: simple logic design, data representation and processor architecture, programming of processors, memory, control flow, input/output, and performance measurements; hands-on lab assignments.

Americans with Disabilities Act (ADA) Policy Statement

The Americans with Disabilities Act (ADA) is a federal anti-discrimination statute that provides comprehensive civil rights protection for persons with disabilities. Among other things, this legislation requires that all students with disabilities be guaranteed a learning environment that provides for reasonable accommodation of their disabilities. If you believe you have a disability requiring an accommodation, please contact Disability Services, located at the Department of Disability Resources at 471 Houston Street or call 979-845-1637. For additional information, visit http://disability.tamu.edu.

IN THIS SYLLABUS:

Prerequisites

All the computer science knowledge necessary for completing this course is given in the course lectures, projects, and textbook. Lab assignments will require basic computer programming skills. Some co-requisite basic background in data structures and algorithms will be helpful.

Getting Started

1. Review the entire syllabus.
2. Log into Canvas using your TAMU NetID and password.
3. Sign on to Microsoft Teams (instructions provided on pp. 3 of this document)

Important Announcement

Course runs from Jan 18, 2022 – Apr 27, 2022.

- **Exams:** There will be **2 exams** (midterm and final) to test knowledge of key topics.
    - **Midterm Exam:** Wednesday March 9 during respective LAB sessions.
    - **Final Exam:** Wednesday April 27 during respective LAB sessions.
- **Projects:** There will be **7 projects** spread evenly through the course content.

Resources

TEXTBOOK

CSCE 312: The Elements of Computing Systems, ***Second Edition*** *Building a Modern Computer from First Principles, By Noam Nisan, Shimon Schocken.*

Publisher: MIT Press, 2021.

Online Link (1st half of the book is free): https://www.nand2tetris.org/

The first half of the textbook is available free online. This is what we cover in the course in addition to advanced concepts that are covered from other sources. Printed copy of the book may be available in Campus Bookstore as well as other book suppliers like Amazon, Barnes & Noble, MIT Press, etc. Purchase of course textbook is **NOT REQUIRED**.

REFERENCE TEXTBOOKS – PURCHASE NOT REQUIRED

Computer Systems: A Programmer's Perspective, Randal E. Bryant and David R. O'Hallaron, Prentice Hall, 2015.

Digital Design, 2nd Ed, Frank Vahid, Wiley Publication, 2010

The reference from Bryant and O'Hallaron is useful for advanced CS course like CS-313. I do not recommend purchasing the Vahid book because it will be referenced in a very limited capacity.

COMPUTER

You are required to have a computing device on which you have permission to install software. Please contact the instructor if you do not have a computing device.

Bring Your Own Device (BYOD) is an initiative in the college of engineering where students are required to bring their own computing device to class. The following link explains the program and provides information about approved devices: https://engineering.tamu.edu/easa/areas/academics/byod. Approved BYOD devices fulfill the requirements for this class, although many other devices are sufficient (i.e. you probably don't have to buy a new computer for this class.)

WEB

CANVAS: https://canvas.tamu.edu/courses/138081

Primarily used for course material and publishing of grades. *In the past, we have also hosted online exams on Canvas.*

The recommended browsers for Canvas access are Safari (MAC), Mozilla Firefox or Google Chrome.

MICROSOFT TEAMS: All course communication will occur on **Microsoft Teams**. This is provided free to all students. The primary benefit is that for many questions everyone can see the answer and other students can answer as well. Teams has excellent mobile and desktop support, and has direct messaging and video conferencing capabilities. We believe the platform is fairly well suited to promote a healthy interaction between the students, as well as with the teaching staff. **Instructions on signing up to TEAMS are posted on Canvas in *Student Resources Module* → *How do I setup Microsoft TEAMS* document.**

Course Logistics:

LECTURES

The course lectures will be offered in "asynchronous" pre-recorded format. Barring exceptions, the lecture materials will be released on Canvas by 11:59pm on Saturday for the week ahead (typically two sets) in accordance with the master schedule (see pp. 6 of this document).

LABS

The Labs will be held F2F. See HOWDY for LAB timings of your section.

People

Instructor

Aakash Tyagi, PhD
Professor of Practice

- **Email**: tyagi@cse.tamu.edu
- **Web**: http://faculty.cse.tamu.edu/tyagi/

Office Hours

To be announced on Course Teams Site

Teaching Assistants:

Mayukh RoyChowdhury: mayukhrc95@tamu.edu
Sagar Adhikari: sagar0073@tamu.edu

Peer Teachers

Kaleb Dickerson kaleb.dickerson2001@tamu.edu
Caleb Terry calebterry@tamu.edu
Senhe Hao senhehao@tamu.edu

Office Hours

To be announced on Course Teams Site

Methodology

This is mostly a hands-on course, evolving around building a series of hardware and software modules. Each module development task is accompanied by a design document, an API, an executable solution, a test script (illustrating what the module is supposed to do), and a detailed implementation plan (proposing how to build it). The projects are spread out evenly and tightly coupled with the lecture material. Each lecture will start by reviewing the work that was done thus far, and giving guidelines on what to do next.

Resources

All the baseline course materials – lecture notes, book chapters, simulators, software tools, tutorials and test programs – can be downloaded freely from the course web site **http://www.nand2tetris.org**. Your teaching staff will guide you through the early phase of software installation and Do's and Don'ts The supplied software can run as is on Linux or Windows or Mac OS. The lecture notes presented in class are modified with Instructor's own thoughts and inputs as deemed necessary.

Programming

The hardware projects will be done in a simple Hardware Description Language (HDL) that can be learned in a few hours. It is covered in the Appendix in our Textbook and will also be covered by the TA's in the LAB. The resulting chips (as well as the topmost computer-on-a-chip system) will be tested and simulated on a supplied hardware simulator, running on the student's computer. The software projects can be done in Java or Python or C++. Any exceptions to this will be noted in the project documents.

Communication

The teaching staff will do its best to communicate relevant administrative information (deadlines, information about posted material, details about projects, locations of tutorials, and so on) in an effective and timely manner on *Teams*. E-mails listed on Exchange are used for communication. If you are not receiving updates, please contact the instructor. **Please use *Teams* for all correspondence related to the course.** This is a very convenient and potentially effective way to communicate with the teaching staff.

Order of Topics (subject to change)

BLOCK	TOPIC (PROJECT RELEASE)
1	Introduction and Setup, Boolean Algebra, Combinational Logic Design (Project 1)
2	Combinational Logic Design, Boolean Arithmetic (Project 2), Sequential Logic Design (Project 3)
3	Machine Language (Project 4)
4	CPU Architecture (Project 5), Assembler (Project 6)
5	Advanced Topics: Memory Hierarchy, Virtual Memory, Cache Memory (Project 7)

Grading*

% total	>=90	80-89	70-79	60-69	<60
Letter Grade	A	B	C	D	F

Exams

36%

There will be a total of 2 exams. Exams WILL NOT be comprehensive. Date and Time of the exams are shown on page 2 of this document. The mode of exams and other logistical advice will be provided ahead of the exams.

- 20% Midterm, 16% Final
  Exams will be a combination of problem solving and multiple-choice questions.

Quizzes

8%

There will be a bunch of take-home online quizzes aligned with lecture content. These are key opportunities to build and apply fundamental concepts to problem solving.

Projects

56%

Projects are aligned lock-step with the course lecture content and help us reinforce the in-class learnings and vice-versa. Some portion of the project may be assigned as "online lab-quiz".

There will be a total of 7 projects

All projects carry equal weight unless specified otherwise.

Master Plan – SUBJECT TO CHANGE

LECTURES							LABS AND PROJECTS						
WORK WEEK	Lecture Topic	Slides	4-Pager	Video	Practice	Quiz	LAB Sched	Lab Topic	Proj ID	Deploy Date	Due Date	Grading Start	Grading Finish
WEEK1	Course Syllabus and Logistics	L0	N/A	N/A	N/A	N/A							
	Course Introduction	L1	N/A	V1	N/A	N/A							
	Computer Organization and Architecture	L2	N/A	V2	N/A	N/A							
	Digital vs Analog, Number Systems, Conversions	L3	N3	V3	PQ3	Q3	19-Jan	NO LAB	Proj1	31-Jan	9-Feb	10-Feb	16-Feb
WEEK2	Logic Expressions, Boolean Algebra	L4	N4	V4	PQ4	Q4	24-Jan	LAB1: Introductions, Software Installation					
	Implementation - Switch, Logic Gates	L5	N5	V5	PQ5	Q5	26-Jan	LAB2: Simple Logic Design					
WEEK3	Combinational Logic Design	L6	N6	V6	PQ6	Q6	31-Jan	LAB3: Realizing Basic Logic Functions with NAND					
	Computer Arithmetic - Binary Addition Concepts and Building Blocks	L7	N7	V7	PQ7	Q7	2-Feb	LAB4: Multi-Way and Multi-Bit MUX and DMUX					
WEEK4	Computer Arithmetic - Multi-Bit and Multi-Input Adders	L8	N8	V8	PQ8	Q8	7-Feb	LAB5: End-to-End Combinational Logic Design	Proj2	9-Feb	16-Feb	17-Feb	28-Feb
	Computer Arithmetic - 2s complement and Subtractors, ALU	L9	N9	V9	PQ9	Q9	9-Feb	LAB6: How to design a multi-bit ripple carry adder					
WEEK5	Sequential Design - Notion of time and memory, Sequential Design, Basic Element of Memory	L10	N10	V10	PQ10	Q10	14-Feb	LAB7: 2s complement and detecting overflow in subtraction					
	Sequential Design - Modular Memory Implementation (Registers, ROM, RAM)	L11	N11	V11	PQ11	Q11	16-Feb	LAB8: How to design a Serial and Parallel Register	Proj3	16-Feb	28-Feb	1-Mar	11-Mar
WEEK6	Machine Language - Big Picture	L13	N13	V13	N/A	Q13	21-Feb	LAB9: How to design a Multi-Bit Multi-Entry RAM					
	Machine Language - Anatomy of a Machine Language	L14	N14	V14	N/A	Q14	23-Feb	LAB10: HACK assembly overview, A and C instructions					
WEEK7	Machine Language - HACK Computer Machine Language	L15	N15	V15	PQ15	Q15	28-Feb	LAB11: Register and Memory based assembly programming					
	Machine Language - Writing Basic Hack Programs	L16	N16	V16	PQ16	Q16	2-Mar	LAB12: Conditional/Unconditional constructs to realize If-Then-Else, FOR/WHILE loops	Proj4	28-Feb	11-Mar	12-Mar	30-Mar
WEEK8	NO NEW LECTURE CONTENT THIS WEEK						7-Mar	LAB13: Input/Output interactions (screen and keyboard)					
							9-Mar	MIDTERM EXAM WED 9-MAR IN LAB					
WEEK9 - SPRING BREAK													
WEEK10	Machine Language - Writing Basic Hack Programs	L16	N16	V16	PQ16	Q16	21-Mar	LAB15: HACK CPU Datapath Design					
	Computer Architecture - Basic Concepts	L17	N17	V17	PQ17	Q17	23-Mar	LAB16: HACK CPU Controlpath Design	Proj 5	21-Mar	30-Mar	31-Mar	11-Apr
WEEK11	Computer Architecture - Building the HACK Computer	L18	N18	V18	PQ18	Q18	28-Mar	LAB17: HACK Computer Design					
	Assembler - Basic Concepts	L19	N19	V19	N/A	Q19	30-Mar	Project 5 Help Session					
WEEK12	Building an Assembler	L20	N20	V20	N/A	Q20	4-Apr	LAB18: Concept and Implementation - Symbol Table and Instruction Lookup Table	Proj 6	30-Mar	11-Apr	12-Apr	20-Apr
	Memory - High Level Concepts	L21	N21	V21	N/A	Q21	6-Apr	LAB19: Overall construction of the HACK Assembler					
WEEK13	Memory - Cache Memory - Design	L23	N23	V23	PQ23	Q23	11-Apr	Project 6 Help Session					
	Memory - Cache Memory - Policies	L24	N24	V24	PQ24	Q24	13-Apr	LAB20: Cache Basic Concepts and Memory Lookup	Proj 7	11-Apr	20-Apr	21-Apr	2-May
WEEK14	Memory - Virtual Memory - Basics	L22-1	N22-1	V22-1	PQ22-1	Q22-1	18-Apr	LAB21: Cache friendly matrix multiplication					
	Memory - Virtual Memory - Advanced	L22-2	N22-2	V22-2	PQ22-2	Q22-2	20-Apr	Project 7 Help Session					
WEEK15	NO NEW LECTURE CONTENT THIS WEEK						25-Apr	Final Exam Prep QnA					
							27-Apr	FINAL EXAM WED 27-APR IN LAB					

Make Up & Late Work

Please review Texas A&M student rule 7:
http://student-rules.tamu.edu/rule07

Participation is expected.

It is *your* responsibility to keep up with the class, even when unexpected events interfere.

Exam Make Up

Missed exams will only be rescheduled for university excused absences. Note that if advanced notice is not feasible, you have 2 business days to provide notification. See *student rules*. A zero will be assigned for exams due to an unexcused absence. Documentation must be submitted **prior** to making up a missed exam.

Late Quiz/Project Submission

- **Quiz**: Late submissions not accepted without prior approval.
- **Projects**: Note details as described below:
- Submission time is determined by the timestamp recorded for your submission on Canvas. If submitted late, projects will receive a grading penalty. The number of minutes late the work is turned in (m) will be used to compute the penalty. Your overall grade for the project will be multiplied by 0.9998^{m}. ***Everyone will be given 5 credit days for the semester which translates to 5*24*60 = 7200 minutes. Penalty will be charged only for late minutes that exceed 7200.***
- Adjusted Project Total = (Raw Total Project Score * $0.9998^{(m-7200)}$)for m>7200
- How turning in late work can affect your grade:

Minutes Late	Max Grade
5	99.9%
60	98.8%
1440 (1 day)	75%
2880 (2 days)	56.2%
4320 (3 days)	42.1%

Course Support

In addition to contacting the instructor or graduate assistant for course content related questions, there are a variety of campus resources for course support.

Technology Support:

For technological issues related to Canvas and CSE S/W, contact the TAMU Help Desk:

- Student Canvas Help Website, https://lms.tamu.edu/Training-Support
- TAMU IT Help Desk:
  - Website: *http://hdc.tamu.edu/index.php* (Online Chat is available)
  - Phone: (979) 845-8300
  - Email: *helpdesk@tamu.edu*
- CSE Help Desk
  - Website: *https://wiki.cse.tamu.edu/index.php/Main_Page*
  - Phone: (979) 845-5550
  - Email: *helpdesk@cse.tamu.edu*

The TAMU Help Desk is open 24 hours a day 7 days a week. If your technical problems are unable to be resolved within 48 hours, please contact instructors or TAs for additional assistance.

Technology issues are not an excuse for missing a course requirement – make sure your computer is configured correctly and address issues well in advance of deadlines.

Student Rules

Each student has the responsibility to be fully acquainted with and to comply with the Texas A&M University Student Rules. More specific rules, information and procedures may be found in various publications pertaining to each particular service or department. For more information, please visit *http://student-rules.tamu.edu/*

Academic Integrity

"An Aggie does not lie, cheat or steal, or tolerate those who do."

Upon accepting admission to Texas A&M University, a student immediately assumes a commitment to uphold the Honor Code, to accept responsibility for learning, and to follow the philosophy and rules of the Honor System. Students will be required to state their commitment on examinations, research papers, and other academic work. Ignorance of the rules does not exclude any member of the TAMU community from the requirements or the processes of the Honor System.

Aggie Honor System Office

You should be familiar with the Aggie Honor System Office. Their website provides more information on academic integrity, plagiarism, etc. *http://aggiehonor.tamu.edu/*

- Definitions of academic misconduct, including plagiarism
- Potential sanctions (LINK)

Acknowledgement

By submitting anything for grading, you are essentially saying "On my honor, as an Aggie, I have neither given nor received unauthorized aid on this academic work. In particular, I certify that I have listed above all the sources that I consulted regarding this assignment, and that I have not received or given any assistance that is contrary to the letter or the spirit of the collaboration guidelines for this assignment."

Plagiarism

Individual programming **MUST** be done on your own. You must write assignments in your own words. Plagiarism will not be tolerated and violations will be reported to the Honor Council.

To help identify possible instances of plagiarism, we may use automated systems for plagiarism detection. Students found to have engaged in plagiarism will be referred to the Aggie Honor System.

Collaboration

Collaboration is important for facilitating learning, and your peers can be a great resource. So you are encouraged to discuss problems and general approaches with each other (but not actual solutions). Regardless, unless stated otherwise, all assignments must be done ***on your own***. It is okay to share general approaches, directions, and so on. You may not look at someone else's code (online or in person). If you have an issue that needs clarification, contact an instructor or TA. Additional overriding instructions and grading policies on permitted collaboration may be provided in the project documents.

Course Acknowledgments

The course owes its existence to the Nand2Tetris organizers Prof. Nisan and Schocken (http://www.nand2tetris.org). Additional material taught in this course is taken from the 312 course taught by Prof. E.J. Kim (http://faculty.cse.tamu.edu/ejkim/), Professor Milos Prvulovic from Georgia Tech (http://www.cc.gatech.edu/~milos/), and Professor Luis Ceze from University of Washington (https://www.cs.washington.edu/people/faculty/luisceze).

The instructor wishes to express sincere appreciation to the following notable students from prior semesters for making lasting contributions and impact on the teaching and content of this course:

1. Jyoti "JD" Dass (Graduate PhD Student and CSCE-312 TA SU'16, FA'17, SP'18, SP'19, FA'19,Su'21). He is the biggest difference-maker and inspiration for this course through his several key initiatives.
2. Jiayao (Amy) Li, Student CSCE-312 Su'17, Typewrote all of Instructor's handwritten 4-Pager notes.
3. Kevin Kuriachan, TA Sp'20, FA'20, SP'21. Created all lab exercise recordings.
4. Teddy Heinen, TA SP'20, FA'20, SP'21. Contributed to significant automation for projects and grading.
5. Jerry Yiu (Graduate PhD Student and CSCE-312 TA FA'15, FA'16). Designed several project exercises.
6. Samira Mirbagher (Graduate PhD Student and CSCE-312 TA SP'19). Introduced several innovative ideas incl. Disassembler and HackerRank for Assembler.
7. Austin Hamilton '16 (Peer Teacher CSCE-312 FA'15, SP'16, FA'16). Responsible for the first all-around course grading automation.
8. Rui Liu (Graduate Student and CSCE-312 grader SU'17, FA'17). Created several project exercises.
9. Yang Yang (Student CSCE-312 SP'16). Introduced new HACK instructions and created new exercises.
10. Muhammad Ashfaq (Student CSCE-312 SP'16). Carried out project on pipelining HACK CPU.
11. Zac Steer and Bobby Hodge '19 (Peer Teachers CSCE-312 FA'17, SP'18, FA'18)
12. Matthew Sheffield (Student CSCE-312 SP'18). Created a new game exercise in HACK.
13. Kyle Coffey (Student CSCE-312 SP'19). Offered several creative ideas for course improvement.
14. Mayukh RoyChowdhury (Graduate PhD Student and CSCE-312 TA SU'19, SU'20, FA'20, Sp'21). Contributed to several exercises in Projects 3, 4, 5.
15. Undergraduate TA's – Maitreyi Ramaswamy, Lekhitha Ammaresh, Shravan Kumaran, Edgar Martinez, Feras Khemakhem, Jose 'Gera' Garza, Carolyn Nguyen, Jeremy Spotts, Ezekiel Blevins, Evelyn Crowe, Neel Pochareddy, Melissa Zhang, Hanson Yu, Alex Labbane. Contributed several lab exercises and productivity improvements.
16. Fatma Elsheimy, TA Su'21. Drove initiative to create Verilog interface with the objective of hardware realization in FPGA
17. Pranav Jain, SP'21. Drove the development of HDL Visualizer.