Syllabus
CSc 342/343 Computer Organization/Computer Systems Design Laboratory

Spring Semester 2021

Instructor: Professor Isidor Gertner
igertner@ccny.cuny.edu

Formal office Hours: M,W 2:00-3:00 Pm, and 24/7 via email or Slack

Public course communications, assignments will be disseminated via Slack

Students will use private and direct communications channel to submit their reports

Catalog Descriptions

Cs342 This course provides computer science and engineering students with an in-depth look at computer architecture and the hardware/software interface. The major topics are: computer abstractions and technology; the role of performance and measuring performance; SPEC, computer arithmetic; machine language; a comparative analysis of instruction sets of current processors using debuggers, simulators and by the partial reverse engineering of executables. The processor: datapath and control; RISC vs CISC; design, implementation (using VHDL), and verification (in simulation) of a simplified RISC processor using CAD tools. Enhancing performance with pipelining. Memory hierarchy, cache, virtual memory, performance issues, interfacing processors and peripherals; PCI chipset. Overview of multiprocessors, grid computing.
Cs343 This course provides hands-on experience designing computers using commercially available CAD tools such as simulators and hardware description languages. Using programmable chips, students produce running machines which can be made part of a computer system using special brands.

Course Major Topics: CS342- This course provides computer science and engineering students with an in-depth look at computer architecture, ISA and the hardware/software interface. The major topics are: computer abstractions and technology; the role of performance and measuring performance; computer arithmetic; machine language; a comparative analysis of instruction set architectures of current processors such as from Intel, AMD, ARM, MIPS using debuggers, simulators, compilers, OS and by the partial reverse engineering of executables. The processor: datapath and control; RISC vs CISC; design, implementation (using VHDL), and verification (in simulation) of a simplified RISC processor using CAD tools. Implementation of CPU components, and ultimately "CPU LITE" on a Field Programmable Gate Array (FPGA) device hosted on a printed circuit board. Enhancing performance with pipelining. Vector instructions, Memory hierarchy, cache, virtual memory, performance issues, interfacing processors and peripherals. Overview of multiprocessors, vector processors, vector programming (SIMD-Single Instruction Multiple Data) with SSE, AVX, NEON instructions, intrinsic functions; principles for writing vectorizable, parallelizable code and performance gains will be discussed.

CS343-Labs provide hands-on experience designing computer components using commercially available CAD tools such as simulators and VHDL -hardware description languages. Students will design, implement (using VHDL), and test (in simulation and on FPGA device) a simplified RISC processor and its components using CAD tools. Implement CPU components (e,g adders, subtractors, multipliers, dividers, memory), and ultimately "CPU LITE" on a Field Programmable Gate Array (FPGA) device hosted on a printed circuit board. Students will have to write a testbench program in VHDL to test their design. A testbench program is used for testing the design and making sure it works according to specifications prior to programming FPGA device. *You will design and build, using FPGA technology, "MIPS LITE" processor.*

Required Texts: Computer Organization and Design: The Hardware/So, **Authors:** David Patterson John Hennessy ,Publisher: Elsevier Science & Technology, **eBook ISBN:** 9780124078864 **Paperback ISBN:** 9780124077263
Edition: 5, Year Published: 2014, Price: from $72.61 USD.

Any Edition can be used: MIPS Edition, ARM Edition, RISC-V Edition.*https://www.elsevier.com/books/computer-organization-and-design-mips-edition/patterson/978-0-12-407726-3*

Required Software:

1. **What is an FPGA? Programming and FPGA Basics - INTEL® FPGAS.** FPGA is an acronym for field programmable gate array (Quartus, ModelSim). **https://www.intel.com/content/www/us/en/products/programmable/fpga/new-to-fpgas/resource-center/overview.html**
2. **Intel® 64 and IA-32 Architectures Optimization Reference Manual** https://software.intel.com/content/www/us/en/develop/download/intel-64-and-ia-32-architectures-optimization-reference-manual.html
3. **Visual Studio Community Edition- Free https://visualstudio.microsoft.com/vs/community/**
4. **Any 64 bit LINUX OS on Intel X64 with gcc, gdb installed.**
5. **MARS-MIPS Simulator, https://courses.missouristate.edu/KenVollmar/MARS/download.htm** Installation and usage instructions will be provided during the course.

Catalog Enrollment Requirements: PRE: CSC 21100 OR (CSC 21000 AND EE 2100). CO: CSC34300.

Prerequisites: Experience with object-oriented programming, some knowledge of assembly, data structures, digital logic design experience, number representations.

Academic Integrity Statement

Academic dishonesty is unacceptable and will not be tolerated. Cheating, forgery, plagiarism, and collusion in dishonest acts undermine the college's educational mission and the students' personal and intellectual growth. College students are expected to bear individual responsibility for their work, to learn the rules and definitions that underlie

the practice of academic integrity, and to uphold its ideals. Ignorance of the rules is not an acceptable excuse for disobeying them. Any student who attempts to compromise or devalue the academic process will be sanctioned.

For students who violate Academic Integrity, my policy is to give a failing grade to any assignment, project presentation, project report, that has been plagiarized, or partially copied or an exam in which you have cheated. **A**cademic sanctions are at the discretion of the Professor, including a D or F for the course.

In addition, I am required by College policy to submit a report of suspected academic dishonesty to the Office of the Dean of Students. ***This report becomes part of your permanent file.***

HomeWorks and Grading

I will assign three major take home tests for cs342 Take-Home Test 1 (THT1), THT2, THT3. I will assign two quizzes on the same material in Take_Home_Tests.

I will use the same approach for the CS343 labs. I will assign 3 major projects, and allocate 1 month for each project. There will be one required summarizing midterm small project.

To prepare students for the projects, I will assign smaller review projects, exercises. I will review your working project files with you.
No reports will be required for these, and no grades will be given, just a checkmark.

Final Grades
For Cs342 grade breakout is:
Quiz1 (10%),Quiz2(10%), THT1(20%),THT2(30%),FinalTHT3(30%)
For CS343
Lab1 (25%),Lab2(25%),MidTermLab(25%), FinalLab(25%)
Please note, you will not qualify for a good grade both Cs342 and Cs343 if your labs are not satisfactory.

Prevention of Academic Dishonesty

For each major report, THT students must submit all ***three deliverables, as explained below:***

1. ***Comprehensive Report (For each project I will specify what content should be in the report)***
2. ***Short video presenting your project for my review and evaluation***
3. ***Project, test files for verification***

Participation and Activities in the course

1. Attending the lectures, and reading the textbook

2. Submit reports, 2 min video, and working code files (as described above) for all Take-home project tests assignments

3. Take all in-class tests.

Technical Objectives CS 342

1. Understand machine Language.
2. Compare ISA for processors Intel, AMD, X86, MIPS, ARM, CORTEX, and CISC vs RISC
3. Different compilers generate different assembly code! GCC. MS Visual Studio
4. Operating system matters. LINUX, WINDOWS, 32 bit, 64 bit
5. Compilers not always create optimal code?
6. You will have to OPTIMIZE (improve) compiler generated code.
7. Understand how program executable, variables, are stored in memory
8. Use debugger in LINUX and Windows environments.
9. Understand and use (write program using C++ and vector machine instructions (AVX, SSE, NEON), use of intrinsic functions, how to write vectorizable code.
10. SISD versus SIMD programming

Technical Objectives Cs 343

To design all components using VHDL needed to build a simple processor.

Learn to use CAD tools: QUARTUS and ModelSim. Test your designs on a Field Programmable Gate Array (FPGA) device hosted on a printed circuit board such as DE-2_70, DE-2-115. Or other boards (if available) to you.

Due to pandemic, we will not use boards this semester.

Our aim in cs342/343 is to help you become a better programmer by teaching you the basic concepts underlying all computer systems. We want you to learn what really happens when your programs run, so that when things go wrong (as they always do) you will have the intellectual tools to solve the problem.

Why do you need to understand computer systems if you do all of your programming in high level languages? Finite representations of numbers have significant limitations, representation of numbers as integer, float, double have limited range, and precision you have to understand. You need to know assembly language to understand the effects of bugs and to optimize compiler generated code. You need to understand how memory is allocated to objects. How to measure performance of a program. Performance improvement with vector processing.

In the lab you will learn to use commercial computer-aided design tools (Model_Sim, Quartus) to design, simulate, and to perform functional verification of digital circuits used to build computers in implement on a FPGA device.

1. OUTCOMES

	342 COURSE OUTCOMES	Excellent (4)	Good (3)	Average (2)	Poor (1)	None (0)
1	Knowledge of representing data, instructions in the computer: addressing modes; control instruction; procedure call instructions, RISC vs. CISC instruction; pointers vs. arrays; recursive procedures					
2	Knowledge of SIMD instructions/performance; reverse engineering (disassembling) binary executable code into assembly					
3	Ability to optimize compiler-generated assembly code to achieve best performance					
4	Ability to perform timing analysis of compiler-generated code vs. the optimized assembly code					
5	Ability to obtain information on the processor and the features it supports					
6	Ability to compare instruction set architectures of MIPS, x86, ARM on different platforms					
7	Knowledge of design and verification in simulation of various computer blocks such as ALU, IO, controller, registers and memory					

	343 COURSE OUTCOMES	Excellent (4)	Good (3)	Average (2)	Poor (1)	None (0)
1	Ability to use commercial computer-aided design tools (Model_Sim, Quartus) to design, simulate, and to perform functional and timing verification of digital circuits used to build computers					

2	Ability to program and test all designs on a Field Programmable Gate Arrays (FPGA) device installed on a board					
3	Basic Knowledge of VHDL and use of libraries					
4	Ability to prepare and defend a report that demonstrates correct implementation of the design					
5	Ability to exchange design ideas with other students and make a presentation in support of outcomes1 through 4					