



6.004 | Spring 2017 | Undergraduate

Computation Structures

Menu

[More Info](#)

Syllabus

The following information reflected how this course was taught at MIT.

Course Meeting Times

Lectures: 2 sessions / week, 1 hour / session

Recitations: 2 sessions / week, 1 hour / session

Course Description

This course offers an introduction to the engineering of digital systems. Starting with MOS transistors, the course develops a series of building blocks – logic gates, combinational and sequential circuits, finite-state machines, computers and finally complete systems. Both hardware and software mechanisms are explored through a series of design examples.

This course is required material for any Electrical Engineering and Computer Science (EECS) undergraduate who wants to understand (and ultimately design) digital systems. A good grasp of the material is essential for later courses in digital design, computer architecture and systems. Before taking this course, students should feel comfortable using computers; a rudimentary knowledge of programming language concepts and electrical fundamentals is assumed.

The problem sets and lab exercises are intended to give students “hands-on” experience in designing digital systems; each student completes a gate-level design for a RISC processor during the semester. Students are encouraged to get help from others in understanding the material, but the designs and measurements they hand in must be their own work.

Prerequisites

[8.02 Physics II: Electricity and Magnetism](#)

Learning Objectives

On completion of this course, students will be able to

1. Understand the role of abstraction in the design of large digital systems, and explain the major software and hardware abstractions in contemporary computer systems.
2. Analyze the performance of digital systems using measures such as latency and throughput.
3. Design simple hardware systems based on a variety of digital abstractions such as ROMs and logic arrays, logic trees, state machines, pipelining, and buses. synthesize digital systems from a library of representative components and test the designs under simulation.
4. Understand the operation of a moderately complex digital system – a simple RISC-based computer – down to the gate level, and be able to synthesize, implement, and debug its components.
5. Appreciate the technical skills necessary to be a capable digital systems engineer.

Measurable Outcomes

Upon completion of this course, students will be able to

1. Identify flaws and limitations in simple systems implemented using the static discipline (noise assumptions, etc).
2. Identify flaws and limitations in simple systems implemented using clocked registers with asynchronous inputs (metastability issues).
3. Identify flaws and limitations in simple systems implemented using pipelined processors (pipeline hazards).
4. Identify flaws and limitations in simple systems implemented using semaphores for process synchronization (deadlocks).
5. Identify flaws and limitations in simple systems implemented using shared-memory multiprocessors (sequential inconsistency).
6. Characterize the logic function of combinational devices using CMOS, ROM, or PLA technologies.
7. Explain synthesis issues for combinational devices using CMOS, ROM, or PLA technologies from their functional specification.
8. Explain synthesis of acyclic circuits from combinational components.
9. Calculate performance characteristics of acyclic circuits with combinational components.
10. Explain and calculate performance characteristics of single-clock sequential circuits.
11. Design, debug, and test combinational circuits of the complexity of an arithmetic logic unit.
12. Design, debug, and test a controller for a finite-state machine.
13. Pipeline a combinational circuit for improved throughput.
14. Understand issues affecting microprocessor instruction set design.
15. Complete and debug the design of a simple CPU with a given RISC-based intruction set.
16. Measure the memory access performance of a processor, and tune cache design parameters to improve performance.
17. Analyze the operation of page-based virtual memory systems.
18. Translate simple programs from C to machine language.
19. Deduce processor state from a memory snapshot during execution.

Feedback

Problem sets

There are no weekly graded problem sets. Instead there are two types of problems with answers provided in the worksheet section of each unit that you can use to test your understanding of the material:

- 1. 6.004x on MITx problems that allow you to test yourself and check if you are correct without viewing the complete answer.
- 2. Tutorial problems which provides additional practice problems with written answers.

Collaboration

The assignments are intended to help you understand the material and should be done individually. You are welcome to get help from others but *the work you hand in must be your own*. Copying another person’s work or allowing your work to be copied by others is a serious academic offense and will be treated as such. We do spot-check submissions to the online check-off system for infractions of the collaboration policy. So please don’t tempt fate by submitting someone else’s work as your own; it will save us all a lot of grief.

Labs

There are seven lab assignments due at various times during the term and a design projects due at the end of the term. Note that you can submit your work for a lab more than once, for example, as you complete each part. To earn points for a lab, all of the problems must be completed.

Quizzes

There are four 50-minute closed-book quizzes. The questions will be similar to those on the tutorial problems, worksheets and past quizzes; you’ll be asked to provide short, written answers and/or explanations.

There is no final exam.

Grading

The final grade is determined by performance on the quizzes (120 points total), the labs (75 points total) and the design project (20 points total). In addition, *you must have a non-zero score for each of the labs and labs 1, 3, 6 and 7 must be checked off as a prerequisite for passing the course*. A missing lab will result in a failing grade.

Once your combined score has been computed as explained above, here’s how grades will be assigned:

GRADES	POINTS
A	$175 \leq \text{total points}$
B	$155 \leq \text{total points} < 175$
C	$135 \leq \text{total points} < 155$
D	$115 \leq \text{total points} < 135$
F	total points < 115, missing lab



Over 2,500 courses & materials
Freely sharing knowledge with learners and educators around the world. [Learn more](#)

[Accessibility](#)

[Creative Commons License](#)

[Terms and Conditions](#)

Proud member of: Open Education GLOBAL



© 2001–2024 Massachusetts Institute of Technology