Department of Electrical and Computer Engineering
The University of Texas at Austin

ECE 306, Fall, 2025
Yale Patt, Instructor
TAs: Luke Mason, Evan Lai, Madeleine Dreher
Course Outline (Syllabus)
August 25, 2025

August 25: Lecture 1. Overview of ECE 306, Levels of Transformation
--The computer -- a complex system organized in levels of interpretation.
--The computer -- a universal computational device; given enough time and space it can do anything any other computational device does.
--Hardware or software
--With just a few simple constructs, we can design very complex things

August 27: Lecture 2: Chapter 2: Bits and operations on bits.
Functions, NOT hardware.
-- The bit as a unit of information.
-- Encoding of bits: Binary numbers (integer data type, ASCII codes
-- Negative numbers, 2's complement representation, sign-extension.
-- Overflow
-- Floating point data type (Normalized numbers, Subnormal, infinities, NaNs)
-- hex representation of binary numbers.
-- Arithmetic operations on numbers. ADD, SUB. [Note that x+x = left shift]
-- Logical variables
-- Logical (Bit-wise) operations on bits. AND, OR, NOT, XOR
-- Truth table representation
-- DeMorgan's Laws

August 29: Discussion Session. Orientation to our computing environment, email, tools, LINUX commands, other logistics

September 1: Labor Day, no class

September 3: Lecture 3. Chapter 2, Bits and operations on bits (continued).

September 5: Discussion Session. Chapter 2

September 8: Lecture 4. Chapter 3. Starting with transistors, moving to Basic Logic Structures

-- Introducing digital logic hardware
-- The transistor as a switch
-- Basic Gates (AND, OR, NOT)
-- Any arbitrary function can be built out of these gates
(no attempt at designing minimal circuits.
-- full ADDER, MUX, DECODER, ALU

September 10: Lecture 5. Storage Structures. Latches, Memory.
-- Basic storage element (Gated RS latch)
-- a logic circuit to implement a small piece of memory (perhaps 2**2 x 3)
-- concept of memory: address space, addressability

September 12: Discussion Session. Emphasis on structures needed from Chapter 3 to build logic circuits and memory.

September 15: Lecture 6. Chapter 3: Finite State Machines, starting with asynchronous (no clock) finite state machines.
-- The notion of STATE (one of the most important concepts in engineering)
-- State diagram, Next State table, State Assignment
-- Implementation examples: sequential machines

September 17: Lecture 7. Implementation of a danger sign. Our first encounter with a synchronous finite state machine.
-- Why latches won't work, so we need flip-flops (registers)

September 19: Discussion Session. Emphasis on Chapter 3 and Problem sets 1,2

Problem set 1, due before class, September 22.

September 22: Lecture 8. Chapter 4. The von Neumann model.
-- the basic structure of the Von Neumann model, showing the basic flow.
-- instruction = opcode, operands
-- encoding of instructions and data
-- instruction cycle (Fetch, Decode, EA, Fetch data, Execute, Store result)
-- organization of memory
-- address space, addressability revisited (MAR, MDR)
-- instruction formats
-- our first program: Multiplying two numbers if your computer can't multiply

September 24: No class.

September 26: Discussion Session. Emphasis on the Simulator/Debugger, PL0, PL1.

Programming Lab 0 due, 11:59pm, September 28.

September 29: Lecture 9. Writing a program to solve a more sophisticated problem: Counting the number of specific characters in a file. This will require Chapter 5: The LC-3 instruction set architecture (ISA) and Chapter 6: Introduction to Structured Programming and Debugging.

Chapter 5: The LC-3 ISA
-- operate, data movement, and control instructions
-- LD/ST (also, indirects)
-- control (condition codes: N,Z,P)
-- I/O via the TRAP instruction [Keyboard input, monitor output]
-- a detailed example in machine language

Chapter 6: Structured Programming and Debugging
-- Problem Solving (stepwise refinement, systematic decomposition, etc.)
-- Debugging (setting breakpoints, single-step, deposit, examine, etc.)
-- Control structures (sequential, conditional, iteration)

October 1: Lecture 10. Use of the data path, state machine, and control signals to implement every instruction in the LC-3 ISA.

October 3: Discussion Session: Prepare for Midterm exam, PL1

Programming Lab 1 (in machine language), Due: October 5, 11:59pm.

October 6: Lecture 11. Review or catch up.

Problem set 2 (Chapters 4,5,6), due before the exam, October 8.

October 8: Lecture 12. **Midterm EXAM 1.** Chapters 1-6

October 10: Discussion sessions. Go over the exam.

October 11: Texas vs. Oklahoma -- enjoy the weekend. Drive carefully.

October 13: Lecture 13. Chapter 7: Up one level: Assembly Language
-- translation: what assemblers and compilers do.

-- hand assemble programs from earlier lectures.
-- revisiting the character count problem

October 15: Lecture 14. Subroutines, JSR/RET
-- saving/restoring state
-- success/failure mechanisms

October 17: Discussion Session, Emphasis on Subroutines and PL2

October 20: Lecture 15, The Stack

October 22, Lecture 16. Other data structures.
-- queues, character strings,
-- sequential and linked storage.
-- tradeoffs between sequential allocation and linked lists for sorted information

October 24: Discussion Session. Data Structures

Programming Lab 2, Due: October 26, 11:59pm.

Problem set 3 (Chapters 7,8), due before class, October 27.

October 27: Lecture 17. Physical I/O Basics
-- memory mapped vs. special I/O instructions
-- asynchronous activity
-- program control vs device driven
-- Load instructions for input, Store instructions for output
-- device registers (KBDR, KBSR, DDR, DSR)
-- Synchronization via the ready bit.
-- interrupt enable bit
-- I/O Service Routines

October 29: Lecture 18. Physical I/O (continued)

October 31: Discussion Session: Problem set 4, Programming Lab 3

November 3: Lecture 19. Interrupts and Exceptions
-- Privilege, Priority

November 5: Lecture 20. Trap routines.

November 7: Discussion Session: I/O, Prepare for Exam 2. PL3

Programming Lab 3, Due: November 9, 11:59pm.

November 10: Lecture 21. Review or Catch up.

Problem set 4, due before the exam, November 12.

November 12: Lecture 22. **Midterm EXAM 2.** Chapters 7-9

November 14: Discussion Session: Go over Exam 2, Emphasis on Programming Lab 4

November 17: Lecture 23. The Calculator Simulation (Stack arithmetic)

November 19: Lecture 24. The Calculator (continued). ASCII/binary conversion.

NOTE: November 19 is the last day to drop a course without special permission, and to change grading to pass/fail.

Programming Lab 4, Due: Before you leave for Thanksgiving break.

November 24-29: No class, Thanksgiving.

December 1: Lecture 25. Recursion, Maintaining ordered structures

December 3: Lecture 26. The future

December 5: Discussion Session: Prepare for Final exam, PL5.

December 8: Lecture 27. Last lecture. Free-for-all. Any OTHER questions!

Problem set 5, NOT to be handed in, use for final exam preparation.

Programming Lab 5, Due: December 8, 5pm.

December 12. **Likely date of the Final Exam, 7pm.**

(Note: the Registrar may change the date of the final exam at his discretion.

Please do not make plans to leave campus for the semester break until after the date of our final exam is confirmed by the Registrar's office.

Programming Labs:

0th programming Lab Due September 28, 11:59pm.
1st programming Lab (machine language) -- Due: October 5, 11:59pm.
2nd programming Lab (assembly language) -- Due: October 26, 11:59pm.
3rd programming Lab (assembly language) -- Due: November 9, 11:59pm.
4th programming Lab (assembly language) -- Due: November 24, 11:59pm.
5th programming Lab (assembly language) -- Due: December 8, 5pm

Problem Sets:

1st problem set, (emphasis on Chapters 1,2,3). Due: just before class, September 22.
2nd problem set, (emphasis on Chapter 4,5,6). Due: just before exam, October 8
3rd problem set, (emphasis on Chapter 7,8). Due: just before class, October 27.
4th problem set, (emphasis on Chapter 9). Due: just before exam, November 12.
5th problem set, (emphasis on Entire course). Not to be turned in.

Final exam: 12/12