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

EE 379K, Fall, 2000
Yale Patt, Instructor
TAs: Katherine Buchheit, , Chandra Jain, Onur Mutlu,
Kameswar Subramanian, Francis Tseng, Brian Ward
Course Outline
August 30, 2000

August 30: Lecture 1. Overview of EECS 100. 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.

September 4: Labor Day. No class.
September 6: Lecture 2: Administrative details.

September 11: Lecture 3: Bits and operations on bits.

-- The bit as a unit of information.
-- Encoding of bits: Binary numbers (integer data type, ASCII characters.
-- Negative numbers, 2’s complement representation, sign-extension.
-- hex representation of binary numbers.
-- Arithmetic operations on numbers. ADD, SUB. [Note that x+x = left shift]
-- Logical operations on bits. AND, OR, NOT.

September 13: Lecture 4. Bits and operations on bits (continued).

September 18: Lecture 5. Basic Logic Structures.

-- The transistor as a switch
-- Basic Gates (AND, OR, NOT)
-- Truth table representations
-- Any arbitrary function can be built out of these gates
(no attempt at minimization. Just an awareness exercise)
-- full ADDER, MUX, DECODER

September 20: Lecture 6. Storage elements.

-- Basic storage element (Gated RS latch)
-- A register
-- a logic circuit to implement a small piece of memory (perhaps 2**2 x 3)
-- concept of memory: address space, addressibility

September 25: Lecture 7. Introduction to Von Neumann model. Emphasis on memory.

-- 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, addressibility revisited (MAR, MDR)

September 27: Lecture 8. ISA Specification of the LC-2

-- instruction formats
-- operates
-- LD/ST
-- indirects
-- control (condition codes: N,Z,P)
-- The datapath necessary to implement the LC-2
-- How I/O works at a very simple level [Keyboard in, screen out]
-- KBDR, KBSR, CRTDR, CRTSR (ready bit, interrupt enable bit)
-- Conversion

October 2: Lecture 9. ISA Specification of the LC-2 (continued)

October 4: Lecture 10. Exam 1.

October 9: No formal class. Review Exam 1, study LC-2 Simulator.

October 11: Lecture 11. Problem Solving and On-Line Debugging.

-- Elements of Problem Solving (stepwise refinement, systematic decomposition, etc.)
-- Fundamentals of Debugging (setting breakpoints, single-step, deposit, examine, etc.)

October 16: Lecture 12. A stored program in the LC-2 ISA

-- the control structure of a stored program (sequential, conditional, iteration)
-- a detailed example in machine language -- counting the number of "?"
-- example will use keyboard input, crt output.
-- example will include entering data via the keyboard and outputting on
the monitor (search a file counting occurrences
of a particular character -- in detail, using LC-2)

October 18: Lecture 13. A stored program in the LC-2 ISA (continued).

October 23: Lecture 14. Moving up a level, Assembly Language and the Assembler.

-- going from higher to lower level: interpretation vs. translation
-- translation: what do assemblers and compilers do?
-- hand assemble programs from earlier lectures.

October 25: Lecture 15. Detailed example of Lecture 11, in Assembler.

October 30: Lecture 16. Physical I/O.

November 1: Lecture 17. The TRAP instruction and I/O Service Routines

-- Keyboard and Monitor Data and Status Registers
-- Polling and Interrupt driven processing
-- ASCII/binary conversion

November 6: Lecture 18. Subroutines (JSR/RET mechanism)

November 8: Lecture 19. Stacks. Parameters. How are they passed?

November 13: Lecture 20. Review or catch up!

November 15: Lecture 21. Exam 2.

November 20: Lecture 22. Applications of stacks. (Interrupt processing, data conversion)

November 22: Lecture 23. Review.

November 27: Lecture 24. More applications of stacks.

November 29: Lecture 25. The Calculator Example (pulling it all together).

December 4: Lecture 26. The Calculator Example (pulling it all together).

December 6. Lecture 27. Review or Catch up! Prepare for Final exam.

Programming Assignments

1st programming assignment (machine language) -- Due: October 15, 11:59pm.

2nd programming assignment (machine language) -- Due: October 22, 11:59pm.

3rd programming assignment (assy lang) -- Due: November 5, 11:59pm.

4th programming assignment (assy lang) -- Due: November 21, 11:59pm.

5th programming assignment (assy lang) -- Due: December 3, 11:59pm.

6th programming assignment (assy lang) -- Due: December 8, 11:59pm.

Problem Sets

1st problem set -- Due: just before class, September 18.

2nd problem set -- Due: just before class, September 25.

3rd problem set -- Due: just before class, October 2. (Note: exam on Oct 4)

4th problem set -- Due: just before class, October 30.

5th problem set -- Due: just before class, November 13. (Note: exam on Nov 15)