

<% strPageTitle = "List of core computing topics for ECE" CreatePageHead %>

# List of core computing topics for EEs/CompEs

This is a minimal list of material related to computing that I believe is essential for the long-term career success of electrical and computer engineers. It is intended to be a MINIMAL list; that is, not what is *good* to know or *might possibly* be useful or is useful for *some* engineers, but knowledge that the lack of which would *very likely* hold our graduates back in their careers.

## Computing

### Programming

- Programming in a practically usable language (e.g., C (or Python?)) (CC)
- Debugging strategies and skills (CC)
- Turning equations into code (CC)
- Variables, arrays, indices, pointers, conditionals, loops (CC)
- Functions/subroutines/methods, parameters (CC)
- Structured/modular programming and basic principles of software design (CC)
- Recognizing parallelism in algorithms (CC)
- Compilation vs assembly (CC)
- Fundamental numerical methods (CC)
- Object-oriented programming concept (CC)

### Digital hardware

- Digital combinatorial logic, standard gates and digital structures (CC)
- Key MSI units (e.g., multiplexers, memory, registers, adders) (CC)
- Basic logic synthesis (Karnaugh maps [brief]) (CC)
- Sequential logic and clocked logic (CC)
- State machine (CC)
- Physical representations of 1s and 0s (voltages, magnetic, optical, spin states, qubits) (CC)
- Transistors used as switches make basic CMOS gates, 6T SRAM, 1T+1C DRAM (CC)
- Gate delay, rise time, critical path (CC)
- Concept of a glitch (CC AC)
- Basic digital logic circuit parameters/models (capacitance, channel resistance determine gate delay) (CC AC)
- Programmable logic/FPGAs (CC AC)
- Hands-on digital circuit experience (power and ground, timing) (CC)
- Hardware description language exposure (e.g. Verilog, VHDL, or RTL) (CC AC)
- Analog-to-digital and digital-to-analog conversion (CC)

### Architecture

- Basic elements of a computer architecture (ALU, PC, registers, memory, busses, sequencer, I/O) (dlj)
- Memory hierarchy, cache (dlj)
- von-Neumann architecture (dlj)

- Hardware interrupts, polling (dlj)
- Pipelining concept and purpose (dlj)
- Pipeline design (dlj AC-EE, core-CompE)
- Harvard architecture (dlj AC)
- Out-of-order execution (dlj AC)
- Speculative execution (dlj AC)
- Superscalar/VLIW (dlj AC)
- Vector processor (dlj AC)
- Multi-core (dlj AC)
- Cache/memory management and tradeoffs (dlj AC)

## Software/hardware interface

- How binary object code controls architecture and executes program on hardware (instruction fetch, decode; operand addressing and fetch; ALU operation; data store (dlj)
- Physical addresses; direct, indirect and offset addressing (dlj)
- I/O software interface (interrupts, how hardware signal triggers software action) (dlj)
- Driver design (dlj AC-EE, core-CompE)
- Stack and stack-based programming (dlj)
- Cartoon understanding of compilation and compiled vs assembly code (dlj)
- Operating systems
  - Cartoon view of what an OS does (scheduling, multitasking, I/O, interrupts (dlj)
  - Time virtualization (time slices, multiple threads on sequential machine) (dlj AC-EE, core Comp E)
  - Memory virtualization and memory hierarchy management (dlj core-CompE)
  - Threads and thread management (dlj core-CompE)
  - Multi-core issues (dlj core-CompE)
  - Real-time OS issues (dlj AC-EE, core-CompE)

## Computational theory

- Automata theory (nv)
- Turing machine (knowing the term and what it applies to, not deep) (nv)

## Networking

- Cartoon understanding of IP (CC)
- Cartoon description of TCP (CC)
- Concept of IP address and cartoon view of what a router does (CC)
- Cartoon view of how the internet works (CC)
- Know what a MAC is and does (CC)

## Applications

- Numerical methods and techniques (CC)
- Application languages (e.g., Matlab) (CC)

## Considered and rejected as not core

- Familiarity with CAD tools, simulation (/CC)
- Recursion (/CC) (But perhaps needed for access to advanced CS courses)
- Concept of thread (/CC)
- SR, D flip-flops and latches (/CC)

(CC) means Curriculum Committee discussed and accepted as core requirement

(/CC) means Curriculum Committee discussed and rejected as core requirement

(CC AC) means accepted by Curriculum Committee as an advanced core (but not core) topic

(xyz) means proposed for the core (essential) list by some faculty member, but not yet discussed by

Curriculum Committee. (/xyz) means (xyz) thinks this is NOT core material. (dlj) = Douglas L. Jones, (sl) =

Steve Levinson, (ek) = Erhan Kudeki, (jm) = Jonathan Makela, (sm) = Sean Meyn, (nv) = Nitin Vaidya,

(sp) = Sanjay Patel

(dlj AC) means (dlj) thinks this is advanced core (only) material. <% CreatePageFoot %>