Informal Computer Engineering Overview for Peer Advisors and Incoming Students

Robert Dick
31 May 2013

This document briefly summarizes topics of interest to peer advisors and students who are considering joining the Computer Engineering major. It has a few embedded URLs, so if you want to follow the links, read it in a PDF viewer.

1 What is Computer Engineering?

Computer Engineering students learn the theory and practice of designing and analyzing complex digital hardware/software systems. They learn how to design computers from bottom to top, from transistors to logic gates, processors, operating systems, and applications.

2 How Does Computer Engineering Relate to Electrical Engineering?

Computer Engineers focus on digital systems, i.e., systems in which discrete values are used to represent state (on and off, integer numbers). Although the transistors they work with are analog components, they use simplified models that treat them as either on or off. Simplifying the models of individual components allows Computer Engineers to design working systems with many components (millions). Computer Engineers must be good with digital circuits, logic, microprocessors, and software. This allows them to design complete hardware-software computer systems.

Electrical Engineers generally use more complex component models than Computer Engineers. For example, they may represent the current through a transistor as a non-linear, gradually varying function of gate voltage while Computer Engineers represent it with a step function. Working with more complex, often analog, component models generally means that Electrical Engineers are limited to designing systems with fewer components. Electrical Engineers are not required to be good with each part of computer system design. This allows them to focus on a particular topic such as signal processing, control theory, or analog circuits.

3 How Does Computer Engineering Relate to Computer Science?

Many concepts learned by Computer Engineers and Computer Scientists overlap. For example, both learn about algorithms, computational complexity, and software design. However, they learn about these things for different reasons. There are many reasons a Computer Scientist might learn these things. A Computer Engineer learns them to use in designing and building complex hardware-software computer systems. Computer Engineers must also learn about circuits, logic, and microprocessor architecture. If you want to design, analyze, and build complex hardware-software systems, pick Computer Engineering. If you want to focus on the theory or practice of analyzing, using, or designing parts of computer systems, e.g., designing...
artificial intelligence algorithms without necessarily focusing on the details of hardware used to run them, pick Computer Science.

4 What do Computer Engineers do After Graduating?

Many go to the many companies working on designing embedded systems, i.e., special-purpose computers such as those in smartphones, cars, and robots. Many go into software engineering. Some go into microprocessor design. A few go into management consulting. It is a fairly flexible major because recruiters looking for software engineers generally interview Computer Scientists and Computer Engineers and recruiters looking for digital hardware engineers generally interview Computer Engineers and Electrical Engineers. Pay and job opportunities for all three majors fluctuate a bit from year to year, but are generally similar. In short, it is probably best to pick the major (among these three) you are most interested in because the differences in pay and opportunities usually aren’t huge.

5 What Classes Should Computer Engineering Students Take First?

- EECS 270: This class covers logic design. Many other Computer Engineering classes have this as a prerequisite. There is also a decent correlation between liking EECS 270 and liking Computer Engineering. Take it as soon as possible.

- EECS 215: This class covers electronic circuits. It is lower priority than EECS 270, but is quite useful to take early, especially for Computer Engineering students interested in circuits and signal processing.

- EECS 280: This class covers data structures and programming. It is lower priority than EECS 270, but is quite useful to take early, especially for Computer Engineering students who want to focus on software.

EECS 215 and EECS 280 are similar in priority, and other classes can also be considered. See Section 8 for details.

6 When Should Students Interested in Computer Engineering Talk With Computer Engineering Advisors?

As soon as they have significant interest in the major. That way, we can explain the major and make sure they are taking appropriate classes to be on track for later declaration. It is easy to schedule an appointment here.

7 The Computer Engineering Program Guide

The Computer Engineering Program Guide is a terse description of the major. Students who are seriously considering the major should read it. Pages 1–4 and 10 cover essential administrative information (what you need to do to graduate). Pages 5–9 provide advice to make it easy for students to determine their main interests and career goals and select appropriate courses.

8 How Should Computer Engineering Students Select Classes in Later Semesters?

1. Look at Page 6 of The Computer Engineering Program Guide You will eventually need to take three of the upper-level courses in the bottom row, and one of these will likely be an MDE course (in
a dark box). Read the descriptions of all of these courses online and pick the three that look most interesting and relevant to your goals. If your goals are fuzzy, then read pages 5–9 of the The Computer Engineering Program Guide or set up an appointment with a Computer Engineering advisor. It is fine if you aren’t totally sure about your three upper-level courses. You can make adjustments later. Just pick three that look interesting for now.

2. Work backward in the dependency graph on Page 6 of the guide. Each solid arrow indicates that the source course must be taken before the destination course. Determine which courses you can take now to get closer to your upper-level picks. For example, if you want to eventually take EECS 470, starting with EECS 270 and EECS 280 is a good idea. If you want to eventually take EECS 427, starting with MA 116 or EECS 215 is a good idea. Why worry about upper-level courses so early? Because some of them take 40 hours per week. They are great courses, but you won’t want to take multiple very heavy courses in the same semester. Clearing prerequisites early will allow you to space out the most demanding courses over time.

3. Check the anonymous Course Workload Surveys of EECS students to see how much time each of the courses you are considering requires and pick a subset that won’t bore or overload you. If you don’t like taking entirely technical courses in a particular semester, take an Intellectual Breadth or General Elective to avoid that situation.

Please send suggestions of improvements to this overview to Robert Dick <ceadvisor@umich.edu>