# Computer Engineering using Innovative Instructional Technologies at the University of Southern California

Timothy Mark Pinkston* Peter A. Beerel†
Computer Engineering Division
EE-Systems Dept., University of Southern California
Los Angeles, CA 90089-2562
{http://www.usc.edu/dept/ceng/}

## 1 Introduction

Our ultimate goal as university educators is to guide students through perhaps the most important stage of a life-long learning experience. The principal challenges associated with achieving this goal for computer engineering students are to

- teach them the engineering fundamentals and specializations they need to successfully develop and integrate various software and hardware components into a complex product,
- provide them with the underlying engineering ethics and economics as well as liberal arts exposure that will guide them to develop products beneficial to our society,
- empower them with the oral and written communication skills needed to thrive in typical team-based engineering environments, and
- impart to them a learning experience that is rewarding rather than frustrating by better teaching backbreaker concepts and by more finely interlacing lectures with hands-on lab experience.

In the first part of this paper, we discuss how USC's computer engineering curriculum addresses these challenges. In particular, although our curriculum incorporates a variety of degree options that allows flexibility for students to focus on a specialization of their choice, for each specialization the student is required to take courses covering a broad range of fundamental skills that are necessary in this increasingly multidisciplinary field. Our emphasis of fundamental skills is particularly important in our masters program in which some of our students come from backgrounds other than electrical engineering and computer science and must take remedial courses to catch up. In addition, our undergraduate curriculum incorporates courses in engineering ethics and engineering economics to empower students with the wisdom of how to use their knowledge for the betterment of society. Our undergraduate curriculum also requires two writing courses, one of which focuses on technical communication. In both our undergraduate and graduate classes, oral communication skills are fostered through required oral presentations of homework and term projects.

In the second part of this paper, we discuss USC's current vision for using instructional technologies to better reach our students. One of the principal cornerstones for improving teaching is the increased interaction between the student, the teaching assistant, and the professor. As a private institution, we have the luxury of offering small classes to both our upper-division undergraduate students as well as our graduate students. We believe, however, that small classes are not enough. We describe our current efforts to use new forms of instructional technology to enhance the learning experience in which the internet as well as multimedia based educational lectures and modules can be used to invigorate our students' learning experience.

## 2 Computer Engineering Curriculum

Toward achieving our ultimate goal and addressing the challenges it presents, we have developed an undergraduate computer engineering curriculum guided by several broad principles and desired objectives [1]. Among them are to (1) provide basic grounding in the fundamentals of mathematics, basic sciences, engineering and engineering economics as well as appreciable content in the humanities and social sciences, (2) make the freshman year as common as possible to that of other engineering majors to facilitate possible changes in major, (3) include a substantial design experience to give our graduates the capability to delineate and solve in a practical way engineering problems which will be encountered during his/her professional career, (4) provide flexibility for students to choose among a wide variety of degree options and specializations within the major, (5) require a certain de-

*This paper was supported in part by an NSF Career Award, grant ECS–9624251.

†Support was also provided by an NSF Career Award, grant MIP-9502386.

gree of specialization in at least one area of the major in order to prepare students adequately to enter the computer engineering profession, (6) require substantial breadth as change in technical area is a common occurrence in the career of a typical engineer, (7) keep our curriculum consistent with the latest ABET[1] accreditation criteria and with other highly-ranked departments, and (8) maintain a curriculum that contains distinctive and innovative features which set it apart from programs at other highly-ranked universities. To a large extent, (4)–(8) also apply to our graduate computer engineering curriculum.

### 2.1 Degree Offerings

In keeping with Objective (4), USC offers many degree options to students who wish to engage in a course of study concentrated on computer engineering. At the undergraduate level, students have the choice of pursuing a BS degree in electrical engineering with special emphasis on computers, *EE(Computers)*, or a joint BS degree in computer engineering and computer science, *CENG/CS*. At the graduate level, students can pursue MS and PhD degrees in computer engineering (*CENG*).

Undergraduate enrollment in *EE(Computers)* and *CENG/CS* has grown steadily over the past five years by approximately 20% and 160%, respectively. This is in stark contrast to the nearly 25% decline in enrollment in EE during the same period. While this enrollment trend may, in part, be due to the present popularity of computers, we believe it speaks to the success and excitement surrounding our undergraduate computer engineering curriculum. The overall demographics of our *EE(Computers)* and *CENG/CS* undergraduate enrollment since fall of 1993 are 20.3% Asian-American, 4.6% African-American, 13% Hispanic-American, 0.4% Native-American, 37.8% White-American, and 23.9% Non-Residents. The demographics of our *CENG* graduate enrollment are 16% Asian-American, 0.7% African-American, 1.7% Hispanic-American, 0% Native-American, 11.4% White-American, and 70.2% Non-Residents. Females make up ≈10% of our computer engineering students.

### Undergraduate *EE(Computers)* Curriculum

A total of 131 semester units are required for the BSEE(Computers) degree. Objective (1) is met by requiring students to take 70 units of fundamental and general education courses: 20 units of humanities and social sciences (including one course which must cover engineering ethics), 20 units of math (calculus thru differential equations), 12 units of physics (mechanics, electromagnetics and optics), 8 units of composition and communication, 4 units of chemistry or material science, 3 units of engineering economics, and 3 units of basic computer programming. Objective (2) is met by encouraging students to schedule these courses into the majority of the first year of study. In addition to these, 19 units of "core" courses in EE are required: Intro to Electrical Engineering, Intro to Digital Logic, Linear Circuits, Intro to Linear Systems, Electromagnetics I, and Intro to Probability and Statistics.

Objective (6) is met by requiring students to take 18 units of "entry-level" EE courses which, generally, are prerequisites for advanced technical electives. Objectives (4) and (5) are met by allowing students to choose approximately 6 units from three of the four topical areas which broadly represent modern EE (in fulfillment of the 18 unit "entry-level" requirement) and to choose one area in which to specialize: (i) Computer Engineering; (ii) Electronic Devices and Circuits; (iii) Communication, Control, and Signal Processing; or (iv) Electromagnetics and Energy Conversion. All *EE(Computers)* students choose (i) as their specialization area and most choose (ii) and (iii) as their breadth areas.

The remaining 24 units go toward fulfilling the computer engineering specialization requirement with technical elective courses such as, but not limited to, Digital and Electronic Circuit Design (or MOS VLSI Circuit Design), Systems Design Using Microprocessors, Computer Systems Organization, Computer Networks, Operating Systems, Programming Systems Design, and a Senior Design Project course. Note that Objective (3) is met by requiring students to take at least three design courses, one of which must be a capstone design course. Moreover, this curriculum meets Objective (7). Finally, we believe Objective (8) is realized in part by meeting the other seven objectives. Additional distinctive features of the *EE(Computers)* curriculum are described in [1].

### Undergraduate *CENG/CS* Curriculum

The *CENG/CS* major is more focused than *EE(Computers)* and, therefore, offers less choice of breadth but still provides substantial choice of specialization within the major. A total of 132 or 133 semester units are required for the BSCENG/CS degree. Students are required to take 66 units of fundamental and general education courses: 24 units of humanities and social sciences (including one science elective course), 24 units of math (calculus, differential equations, linear algebra and probability), 8 units of physics (mechanics, electromagnetics), 7 units of composition, and 3 units of engineering economics. In addition to these, 19 units of "core" courses in *CENG/CS* are required: Intro to Electrical Engineering or Intro to Computer Science, Intro to Digital Logic, Intro to Digital Circuits, Basic Computer Programming, Data Structures, and Principles of Software Development.

[1] Accreditation Board for Engineering and Technology.

It is also compulsory for students to take 22 units of EE courses (circuits and computer organization/systems) and 24 units of CS courses (theory and programming) covering Digital Electronics, Systems Design Using Microprocessors, Computer Systems Organization, Discrete Methods, Theory of Computation, Design and Analysis of Algorithms, and Operating Systems. In addition to these courses, either a 3 unit Senior Design Project course in EE or a 4 unit Design and Construction of Large Software Systems course in CS are required to fulfill the capstone design requirement.

The remaining 14 units go toward fulfilling the computer engineering specialization requirement in at least one of the following four areas using CS and EE technical elective courses such as, but not limited to, those in parentheses: Theory (Programming Languages, Finite Automata); Software Systems (Software Engineering, Databases, Multimedia); Graphics and Multimedia (Computer Graphics, Multimedia, Digital Media, Computer Networks); and Advanced Circuit Design (MOS VLSI Circuits, Digital and Electronic Circuit Design, Integrated Circuit Design).

### Graduate *CENG* Curriculum

The *CENG* masters program builds upon the undergraduate curriculum while still providing flexibility in depth and breadth of specialization. A minimum of 27 semester units are required, consisting of no more than 9 units of remedial graduate courses and a minimum of 18 units of advanced graduate courses. The remedial units—although not required—allow students coming from different undergraduate programs to acquire background in fundamental computer engineering courses discussed above, which is required. Six of the 18 advanced units cover "core" required courses: Computer Systems Architecture and VLSI System Design. The remaining units are used to cover an area of specialization: Computer Architecture, Networks, VLSI, or CAD. Students are required to take 6 units of depth and 6 units of corresponding breadth for their choice of specialization.

For the Computer Architecture specialization, depth courses include Parallel Processing, Interconnection Networks, Real-Time Computer Systems, Fault Tolerant Computer Systems, Multithreaded and Data-Flow Architectures, System Architecting, and Compiler Design. Breadth courses include, but is not limited to, Queuing Theory, Broadband Network Architecture, Computer Communications Networks, Logic Design and Switching Theory, Design of Reliable Digital Systems, Computer Aided Design, Advanced VLSI System Design, Integrated Circuit Analysis and Design, Analysis of Algorithms, and Analysis of Parallel Computation.

The *CENG* doctoral program provides the most flexibility by requiring that students take only one advanced course from each of the computer architecture, theory, and software systems areas within the 60 unit graduate-level coursework requirement.

## 2.2 Computer Architecture Specialization

It is important for us to give our computer architecture students opportunity to receive a sound, well-rounded exposure to the gamut of hardware and software issues. Within the EE department, the Computer Engineering Division administers many undergraduate (400-level and below) and graduate courses (400-level and above) for the computer architecture specialization. A sampling of a dozen courses comprising the nucleus of our architecture offerings is described below with the primary text cited.

**EE101** Intro to Digital Logic [3]: Boolean algebra; number systems and binary arithmetic; combinational logic and simple sequential logic design.

**EE102L** Intro to Digital Circuits [3]: Practical digital design using MSI TTL devices; system design involving finite state controllers; laboratory experiments using PLD and logic simulation.

**EE357** Basic Organization of Computer Systems [4]: Organization and operation of the processor, memory and I/O at the machine language level; assembly language programming; computer arithmetic.

**EE454L** Intro to System Design using Microprocessors [5]: Operation and timing of 8/16-bit microprocessors; design and implementation of a single-board computer based on an Intel 8088 and peripherals; microprocessor-based system design.

**EE457** Computer Systems Organization [6]: Register transfer level machine organization; CPU datapaths and control; microprogramming; computer arithmetic; memory hierarchy.

**EE459L** Senior Design Project: Design, implementation and test of a substantial computer hardware project; presentation, demonstration and project report.

**EE554** Real-Time Computer Systems: Structure of real-time systems; scheduling and synchronization; queuing applications; reliability and availability.

**EE557** Computer Systems Architecture [7]: Instruction set and microarchitecture; advanced pipelining and instruction-level parallelism; memory hierarchy and I/O systems; parallel processing; interconnection networks.

**EE653** Multithreaded and Data-Flow Architectures: programmability of general purpose multiprocessor systems; functional programming; data-flow and multithreaded computers.

**EE656** Fault Tolerant Computer Systems: Fault classification, modeling and evaluation; measures of reliability; detection and recovery; case studies of networks and distributed systems.

**EE657** Parallel Processing [8]: Parallel programming; workload-driven evaluation and case studies; distributed shared-memory and message passing multiprocessors; prefetching and multithreading.

**EE659** Interconnection Networks [9]: Theory, design and analysis of parallel processor interconnection networks; topologies, deadlock-free routing, switching and flow control; network interfaces.

We expose students to a rich tool set which has repeated use in several of our courses to reinforce students' learning. Acquiring design skills using these tools will be invaluable to the student's professional career and research career if (s)he continues in the doctoral program. Industrial-grade CAD tools (see [2]) are used for teaching combinational and sequential logic design in EE102L; simple arithmetic circuits and control units in EE357; complete CPU, microcoding, and pipeline design in EE457; memory and I/O interface design for microprocessors in EE454L; and hardware design verification in EE459L. Very high scale integrated circuit hardware description language (VHDL) tools are also used in EE457 and EE459L. University-developed microprocessor, multiprocessor, and network simulators and micro/macro benchmark suites are used in the advanced graduate-level courses to further equip our students with cutting-edge tools and methods for doing research and development in the computer architecture area.

## 3 Enhancing The Learning Experience Using Instructional Technology

We are actively pursing techniques that use instructional technology to enhance our students' learning experience. Currently, our efforts have focused on using this technology to make course-related information web-accessible, better teach backbreaker concepts using simulation and animation, and increase the interactivity and response time between students and faculty.

As a first step to integrating instructional technologies into our classroom, we have recently established the systematic development of class web pages. The required content of these class web pages include a course description, syllabus, location and time, contact information for the instructor and teaching assistants (TAs), midterm/final exam dates, and grading policies. Optional web-based materials include on-line homeworks, which is of particular benefit to our off-campus masters students because it circumvents the delay incurred by our currier service. Publishing homeworks on-line is also helpful to the faculty as they no longer need to rush to design the homework assignment before the beginning of lecture and can broadcast the assignment via the web. In addition, some faculty have reduced the number of repeated questions asked by students by maintaining a frequently asked question (FAQ) list for each homework.

The systematic establishment of web pages on a school-wide basis, although leading to an improved educational experience, is not trivial. It requires significant infrastructure that includes computer accounts to publish the web pages, class-web-page-development training, support-staff and faculty support. Faculty support is perhaps the most difficult to obtain: the time-pressures most faculty face make them hesitant, if not unable, to adopt new instructional technology that requires some amount of time and energy to learn and maintain. Motivated by this understanding, we decided to leverage off of existing staff and TAs. As an addition to our existing TA training program, we developed a 3-hour class-web-page-development class in which our TAs learn web page development techniques while actually creating the initial web page for the class with which they are assisting that semester.

We are also actively exploring using existing and developing new instructional technology to better teach so-called *backbreaker* concepts [10]. For example, in computer architecture the mechanisms for exploiting instruction level parallelism (pipelining, scoreboarding, multiple-issue, etc.) are typically the backbreaker concepts, particularly since they involve dynamic behavior which is difficult to convey using traditional instructional techniques. We believe that multimedia educational modules in which these techniques are explained using animation will help students learn faster and with less effort. To support this move towards multimedia modules we are trying to (1) collaborate with other universities to share instructional technology materials, (2) create an infra-structure in which interested faculty can generate such material and be rewarded for their efforts, and (3) build multimedia equipped classrooms that support this mode of teaching.

We are currently in the planning stages of a new building, the Engineering Academic Center, which will house a variety of state-of-the-art multimedia equipped classrooms. One innovative classroom model we are considering is the studio classroom, pioneered by RPI [11]. These classrooms typically have multiple semicircular rows of student work-areas that are tiered to promote more eye contact with the instructor. The students have desks in front of them which face the front of the classroom where the instructor stands and delivers a multimedia based lecture that is projected above or beside him/her. Behind the students is a second workspace set up with a computer and possibly lab equipment.

This studio classroom environment supports two forms of increased interactivity. First, the instructor may be able to set up in-class questions in which the students can enter the answer into their computer on-the-fly. Their answers can be immediately recorded and graded, and the percentage of students who had the correct answer can be tallied and projected onto the screen in front of the class, providing immediate feedback to the student and instructor. Secondly, the studio classroom allows for lab exercises and lectures to be interlaced in which short hands-on lab exercises and mini-lectures can be intertwined in one class session. During the lab exercise portion of the class, the instructor can view all workstations from the front and walk around the classroom giving one-on-one guidance. We believe this radically new form of increased interactivity may help invigorate our students' learning experience.

A third means of increasing interactivity between the student and faculty we are exploring is the use of advanced web-based applications to support both asynchronous and synchronous group learning. For example, the University of Illinois effectively uses software that maintains threaded discussions among students and TAs [12]. This effectively merges on-line office hours with static FAQ sheets. Moreover, they developed web-interfaced on-line quizzes that are automatically graded, providing students with immediate feedback instead of the more typical two-week turn-around time needed to hand-grade homeworks. They have done an evaluation of these techniques that demonstrates impressive improvements in student performance.

## 4 Conclusion

Many challenges face deliverers of computer engineering education. A balanced curriculum has appropriate components of liberal arts; engineering ethics, economics, and communication; engineering fundamentals; and area specialization. The University of Southern California offers a well-balanced computer engineering curriculum that contains distinctive features and promotes innovative instruction. Curricular uses of the internet are envisioned to improve our educational paradigm and to help meet the expectation that information technology will be an integral part of students' learning experiences. Many new opportunities for effective web-based teaching are emerging which we are actively, but cautiously, pursuing. Web- and multimedia-based instructional technology must be accompanied by a critical eye that distinguishes novelty from a truly improved form of pedagogy. Consequently, we believe it is critical to adopt effective evaluation mechanisms, such as those done at the University of Illinois. Finally, infrastructure and support costs of emerging instructional technologies are high and have to be considered wisely and minimized where possible.

## Acknowledgements

We acknowledge the many people who have been active in the development and adoption of new instructional technologies at USC's School of Engineering including, Stan Settles, Dick Kaplan, Ellis Horowitz, John Silvester, Rick Lacy, Lewis Johnson, and Jerry Mendel. Discussions with Gandhi Puvvada have also contributed to this paper.

## References

[1] H. H. Kuehl and J. M. Mendel. "Electrical Engineering Curriculum Review Task Force Final Report". Technical report, University of Southern California, Sept. 1996.

[2] G. Puvvada and M. Breuer. "Teaching Computer Hardware Design Using Commercial CAD Tools". *IEEE Transactions on Education*, 36(1):158–163, February 1993.

[3] J. Wakerly. *Digital Design: Principles & Practices, $2^{nd}$ Edition.* Prentice Hall, 1994.

[4] Hamacher, Vranesic, and Zakey. *Computer Organization, $4^{th}$ Edition.* McGraw-Hill.

[5] J. Uffenbeck. *The 80x86 Family: Design, Programming, and Interfacing, $2^{nd}$ Edition.*

[6] D. Patterson and J. Hennessy. *Computer Organization & Design: The Hardware and Software Interface, $2^{nd}$ Edition.* Morgan Kaufmann, 1997.

[7] J. Hennessy and D. Patterson. *Computer Architecture: A Quantitative Approach, $2^{nd}$ Edition.* Morgan Kaufmann, 1996.

[8] D. Culler, J.P. Singh, and A. Gupta. *Parallel Computer Architecture.* Morgan Kaufmann, 1998.

[9] J. Duato, S. Yalamanchili, and L. Ni. *Interconnection Networks: An Engineering Approach.* IEEE Computer Society Press, 1997.

[10] J. M. Mendel. Multimedia instruction for engineering backbreaker-concepts, 1998. Unpublished manuscript.

[11] J. R. Young. Rensellear polytechnic institute: The studio classroom. *ASEE PRISM*, page 15, January 1996.

[12] B. Oakely, II. A virtual classroom approach to teaching circuit analysis. *IEEE Transactions on Education*, 39(3):287–296, August 1996.