Logic Design as an Entry Point for Non-engineers: Increasing Diversity in Microelectronics Education and Research

Ani Nahapetian
University of California, Los Angeles (UCLA)
Los Angeles, CA 90095
ani@cs.ucla.edu

ABSTRACT

This paper presents specific course content suggestions for creating and teaching an introductory course in logic design to non-engineering students. The primary motivation for the course is to inform and attract non-engineering students to field of microelectronics. Most students have no exposure to wireless systems, IC design, nanotechnology, etc in high school and hence they can not determine their aptitude for work in the field. With the use of logic design and its inherently shallow learning curve, students can preview many aspects of microelectronics. By appealing to a broader university student population, the course has the ability to increase the interest and the diversity of students in microelectronics education and research, overall.

Keywords

Logic Design, increasing diversity, non-majors.

1. INTRODUCTION

Microelectronics education in particular and Electrical Engineering and Computer Science in general, experience disproportionably fewer women and minority enrollment than other majors. These statistics have led to a concerted effort to improve outreach and recruitment, but the rate of improvement is still very small.

This paper proposes an alternative and orthogonal approach to increasing the diversity of the student population engaged in microelectronics education and research. A new entry point to field is proposed using a logic design course for non-engineers.

There is a great deal of literature, as well as experience, using introduction to programming classes for non-majors to increase the enrollment of students in Computer Science degree programs. A logic design course can serve a similar purpose for Computer Engineering.

Many, if not most, students have little or no exposure to Computer Engineering and microelectronics in high school. There is little mention of the topics in standard science courses, including Chemistry, Biology, Physics, and Math, that all college-bound high school students take. As a result, the students are both intimidated by and ill-informed about what the field entails. Additionally, the lack of early exposure makes it difficult for them to determine their true aptitude for the topic, which would increase the level of interest in pursuing the field as a major in college.

With the development and/or expansion of existing logic design courses, a new entry point for non-engineering students to microelectronics can be created. This would address the well-known phenomenon of the shrinking pipeline in Engineering, which is used to describe the decreasing number of women and minorities that go from undergraduate, to graduate, to academic work.

A focus on design, as opposed to tools can encourage students with a true potential to succeed to delve further into other degree courses. Secondly, if students end up taking only one course in microelectronics, a practice of design is a fundamental enhancement to their education.

The introductory course is a perfect venue for introducing cutting-edge research. This level of early exposure has the potential to encourage more students to consider and prepare themselves for research opportunities and graduate degrees.

To create initial interest in a logic design course, it is possible to consider incorporating microelectronics into programming projects for introductory programming courses. Similar efforts have been made to introduce the various fields of Computer Science to students in CS 1 courses. Specific programming projects, which have been tried and tested at UCLA, are presented, in this regard.

This paper is based on the experience of teaching logic design at the University of California, Los Angeles, as well as the experience of preparing and teaching programming to non-engineers, there.

The paper's main contributions are based on actual classroom experiences and surveys of the related work. First, the basis for suggesting a logic design course for non-engineers is presented. General considerations for course design are provided, including comments on lecture example and project selection. Finally, some detailed projects ideas are suggested and motivated.

2. COURSE INFRASTRUCTURE

2.1 Course Structure and University Conventions

The University of California, Los Angeles (UCLA) is large research focused public university in California. It has an undergraduate population of nearly 27,000 students, where over 11% of the students are enrolled in the engineering school.

Currently, there is a Logic Design for Digital Systems course, offered by both the Computer Science and Electrical Engineering departments. This lecture course serves as a prerequisite for the Digital Logic Design Laboratory course, and its enrollment is largely limited to Computer Science (CS) and Electrical Engineering (EE) students.

UCLA, also, has two types of introductory programming courses, those offered by the Computer Science department for CS and Engineering students and those offered by the Mathematics department's Program in Computing (PIC) for all other non-engineering students.

2.2 Potential Student Enrollment

The PIC courses can serve as a useful predictor for general student body interest in our logic design course. There is usually a large diversity of majors enrolled in the UCLA PIC courses, which are essentially programming courses for non-engineers.

A significant percentage of students are related to the mathematical and statistical fields, mostly due to the department graduation requirements. Additionally, roughly about 30% of the students are from the natural sciences and 10% of the students are from the social sciences, from such fields as Linguistics, Cognitive Science, Economics, and Sociology.

The remaining 20% of the class is composed of students who have not yet declared their major. This subsection of students is an extremely attractive group, who can be enticed to pursue Computer Engineering as their major.

Enrollment for a logic design for non-engineers course is difficult to predict based off of the programming courses, however there are surely to be similarities. Additionally, consider the logic courses that Philosophy departments offer. Such courses are often popular with pre-law students, and they fulfill certain general education requirements. A logic design course can be approved by university curriculum committees to apply for similar credits in general educations (GE) requirements, if they are targeted to non-engineers.

3. BASIS FOR APPROACH

The following observations, trends, and conclusions colored my perspective and became the basis for the choices made in structuring, preparing, and delivering the course content.

3.1 Lack of Previous Exposure

The lack of exposure that students have to microelectronics in high school, surely colors their enrollment in college. Although high school typically offer Chemistry, Biology, Physics, and Math to all college-bound high school students, Computer Engineering is only rarely and possibly mentioned in a Physics or a Computer Science high school course. As a result, students are both intimidated by and ill-informed about what the field entails. Additionally, the lack of early exposure makes it difficult for them to determine their true aptitude for the topic, thus steering away potentially great engineers to other fields.

An approach attempted previously involves highlighting the impact that microelectronics has on student life with technology such as mobile phones, cars, and smart cards. Although notable, this approach misses the fact that student want to evaluate their aptitude and potential to succeed in a field of study. And therefore, it is important to also highlight the specifics of study.

This is especially true, since little is understood from the names associated with the subfields of study in microelectronics. For example, the term Computer Aided Design (CAD) is often associated with drafting. Similarly, despite the multitude of embedded systems that students are surrounded by, the term is not used or understood by the general public.

A clear explanation for the definition, as well as the mention of design, algorithms, computational geometry, statistical analysis, and more must be made clear to students somehow.

3.2 Logic Design Good Match

Of all of the diverse course offerings in the Computer and Electrical Engineering departments, logic design has several unique characteristics that make it a good match for student recruiting to the field.

First, logic design has a shallow learning curve, as the course is based on a relatively small knowledge base. The large majority of the work in the course is dedicated to design, once fundamental tools and approaches are learned. For example, logic design is not about creating K-maps. Instead, once K-maps are understood, student are repeatedly asked to design various circuits, where K-maps may potentially serve as one tool they use from their toolkit.

A second and related point is that logic design requires no prerequisites. It is a well-contained course, which does not require any Physics, Math, or Electrical Engineering knowledge. This is especially attractive to social science or humanities students who, in some cases, are not even required to take one Math or Physics class in college.

Third, logic design can be a conduit for introducing other more advanced topics in microelectronics, such as FPGAs, nanotechnology, embedded systems, low-power design, and more. FPGAs can be presented with the discussion of programmable logics arrays. Embedded systems can be presented initially as a motivation for creating circuits. Low power systems can be presented along with the discussion of gate level power consumption and delay.

Finally, the fact that there is a reciprocal class in many Philosophy departments creates the strong potential for convincing university curriculum committees to approve a logic design course to serve as a GE. Additionally, the success and popularity of Philosophy logic courses demonstrates the likelihood that students from the humanities would be interested in taking a microelectronics course, if the topics was presented in a palatable way.

3.3 Increasing Diversity

A great deal of research and experience exists regarding increasing the numbers of women and minority that enroll in engineering departments [Margolis2001] [Margolis2008].

The intimidation factor of not knowing or not having first hand experience with a field has been shown to be especially detrimental to women. It has been shown that students who may have a strong aptitude from the subject but lack experience and confidence do not explore the field further [Carter2006].

By presenting logic design in a general education setting to a larger and more diverse audience, it is possible to increase the diversity of the students who explore Computer Engineering study and research.

3.4 Learning from CS 1 Courses

There is a great deal of literature, as well as experience, using introduction to programming courses (commonly referred to as CS 1 courses) for non-majors to increase the enrollment of students in Computer Science degree programs [Cortina2007] [Gimenez2009] [Goldman2004] [Hoffman2009] [Smarkusky2009] [Zhang2007].

Additionally, introductory programming courses have been used to survey more advanced Computer Science topics, including algorithms, artificial intelligence and computer graphics, to freshman [DesJardin2010] [Dodds2008], as well as application areas [Nahapetian2011].

A similar approach can be carried out for Computer Engineering, using a logic design course. As highlighted previously, a logics design course can allow a glimpse into more advanced topics, through design projects and lecture examples. Providing students with exposure to some advanced concepts encourages them to search for undergraduate research opportunities.

4. COURSE CONTENT IDEAS

The paper enumerates expansions and modifications to existing courses to inform non-engineering students about the requirements and joys of Computer Engineering.

4.1 Focus on Design

As a general rule, the focus of the course should be on design, and not acquiring knowledge of tools. Specific design tools, such as K-maps, or common modules, such generate-propagate modules, are not the main objective of a logic design course. Instead, the course should reinforce the necessary creativity and planning required to create successful and functional designs. The focus on design and problem solving is a much more accurate picture of the requirements necessary to continue with microelectronics education and research.

The homework examples from [Ercegovac1998] have served as an excellent source of ideas and inspiration, in this regard. By challenging students to create complex digital systems using simple modules, the examples flex the students design muscle. Related work has also looked at this approach, for paradigm-based teaching of logic design [Potkonjak2007].

Conversely, if students end up taking only one course in microelectronics, a practice of design is a fundamental enhancement to their general education. Outside of visual design courses and writing courses, there is little opportunity for students to practice design and creativity to this level. A logic design course enables an interesting combination of creativity, organization, and hands-on thinking, which should be a fundamental part of any university education.

4.2 Research Transfer

With the large range of cutting-edge microelectronics research going on today, in embedded systems, wireless system, CAD, and other area, non-engineering and even CS students have little exposure to them from their required undergraduate courses. This logic design course is a perfect venue to incorporate and transfer research advances and ideas to the Computer and Electrical Engineering pedagogy.

Accomplishing research transfer, by taking problems from research work and transitioning them in to undergraduate course projects and exam questions, can be both invigorating and informative for students. It creates the opportunity for students to become informed and interested in microelectronics research. Additionally, it presents the cutting-edge side of the field, along with its applicability to our everyday lives.

4.3 Using Introductory Programming as a Recruiting Tool

Introductory programming course can still serve as an entry point to Computer Engineering. By employing the advanced topics and application area survey approach [DesJardin2010] [Dodds2008] [Nahapetian2011] carried out previously for topics such as algorithms, artificial intelligence and bioinformatics, programming projects related to microelectronics can expose and excite students in a very real way.

Consider the following programming projects that introduce students to programming concepts, while also introducing them to microelectronics subfields.

- For a project on recursion, drawing HTrees [Sedgewick] and explaining their significance in CAD;
- For a project related to pseudorandom number generation, carrying out placement using simulated annealing;
- For a project practicing object oriented programming, creating classes for gates, circuits, and wires, and then simulating the power characteristics and fault tolerance of circuits. Common benchmarks, such as the ISCAS '85 benchmarks can be used in this case.

5. CONCLUSION

This paper argues for the potential that a logic design course has to increase the diversity of students pursing microelectronics education and research, by way of providing an entry point for non-engineers to explore the field. Specific and practical approach for teaching such a course were presented and motivated.

6. REFERENCES

[Carter2006] Carter, L. 2006. Why students with an apparent aptitude for computer science don't choose to major in computer science. In *Proceedings of the 37th SIGCSE Technical Symposium on Computer Science Education* (Houston, Texas, USA, March 03 - 05, 2006). SIGCSE '06.

[Sedgewick] R. Sedgewick and K. Wayne. Introduction to Programming in Java: An Interdisciplinary Approach. Addison Wesley, 2008.

[Cortina2007] Cortina, T. J. 2007. An introduction to computer science for non-majors using principles of computation. SIGCSE Bull. 39, 1 (Mar. 2007), 218-222.

[DesJardin2010] Marie desJardins , Michael Littman, Broadening student enthusiasm for computer science with a great insights course. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, (Milwaukee, Wisconsin, USA, March 10-13, 2010).

[Dodds2008] Dodds, Z., Libeskind-Hadas, R., Alvarado, C., Kuenning, G. 2008. Evaluating a breadth-first cs 1 for scientists. In *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education*, (Portland, OR, USA, March 12-15, 2008)

[Ercegovac1998] Ercegovac, M. D., Lang, T., and Moreno, J.H. (1998) *Introduction to Digital Systems.* Wiley and Sons.

[Gimenez2009] Gimenez, O., Petit, J., and Roura, S. 2009. Programacio 1: A pure problem-oriented approach for a CS1 course. In *Procedings of the 4th Informatics Education Europe* (Freiburg, Germany, November 2009).

[Goldman2004] Goldman, K. J. 2004. A concepts-first introduction to computer science. In *Proceedings of the 35th SIGCSE Technical Symposium on Computer Science Education* (Norfolk, Virginia, USA, March 03 - 07, 2004).

[Hoffman2009] Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., and Hosking, A. L. 2009. A multidisciplinary approach towards computational thinking for science majors. *SIGCSE Bull.* 41, 1 (Mar. 2009), 183-187.

[Margolis2008] Margolis, J. et. al. (2008). *Stuck in the Shallow End: Education, Race, and Computing*. Cambridge: MIT Press.

[Margolis2001] Margolis, J., Fisher, A. (2001) *Unlocking the Clubhouse: Women in Computing*. Cambridge: MIT Press.

[Nahapetian2011] Nahapetian, A. 2011. La Bella Figura: Making a Good Impression When Teaching Introduction to Programming to Non-engineers. In CCSC Southwestern Conference, 2011.

[Potkonjak2007] Potkonjak, M., Wong, J.L. 2007. Introduction to digital design: A paradigm-based approach. In IEEE International Conference on Microelectronic Systems Education, 2007, pp. 167-168.

[Smarkusky2009] Smarkusky, D. L., Toman, S. A. 2009. An interdisciplinary approach in applying fundamental concepts. In *Proceedings of the 10th ACM conference on SIG-information technology education*, (Fairfax, Virginia, USA October 22-24, 2009).

[Zhang2007] Zhang, M., Lundak, E., Lin, C., Gegg-Harrison, T., Francioni, J. 2007. Interdisciplinary application tracks in an undergraduate computer science curriculum. In *Proceedings of the 38th SIGCSE technical symposium on computer science education*, (Covington, Kentucky, USA, March 07-11, 2007).