Bridging the gap between discrete and programmable logic in introductory digital logic laboratories

Kevin Nickels, Farzan Aminian, J. Paul Giolma
Department of Engineering Science
Trinity University

Abstract

Most contemporary introductory digital logic design laboratories utilize discrete small-scale integrated (SSI), medium-scale integrated (MSI), and programmable logic such as field programmable gate arrays (FGPAs) or complex programmable logic devices (CPLDs). These more complex programmable devices (CPLDs and FPGAs) exhibit a superset of SSI and MSI functionality, enabling instructors to completely eliminate construction of circuits using SSI and MSI chips if they so choose.

This paper describes the approach taken by the authors for introductory digital design laboratories in the Engineering Science Department of Trinity University, and explains some of the reasons for the given approach. The approach begins with construction of SSI and MSI circuits, includes a bridging laboratory where a complex SSI/MSI circuit is reimplemented in a CPLD, and finally moves to more complex designs utilizing no SSI/MSI constructs. The student base for introductory digital logic in our engineering program is quite broad, with all engineering students utilizing digital design in portions of two courses and a third optional course.

This approach allows beginning students to benefit from the direct mapping from gate level design to implementation afforded by construction of circuits using SSI components. This is important, particularly to students of the “sensor” learning style. The bridging activity allows assimilation of concepts involving complex programmable logic without having confounding design concepts. Finally, the use of CPLDs with graphical or hardware description language design entry allows the construction of more complex designs than would be possible using only SSI/MSI components, and exposes the students to these useful modern tools. The mixture and progression of these techniques allows us to serve the various audiences in the Engineering Science program at Trinity, and allows us to incorporate complex programmable logic devices into the curriculum without losing several important features of the traditional introductory digital design laboratories.

Departmental Background

Trinity University is a primarily undergraduate liberal arts and science institution of about 2300 students. The Engineering Science Department is a small and intellectually diverse department, with 9 faculty members (4 mechanical engineers, 2 chemical engineers, and 3 electrical engineers) and approximately 120 students. The department features a broad-based engineering curriculum devoted to a liberal and integrative engineering education in the context of the University's tradition of the liberal arts and sciences. This mission of the department is explained in the mission statement of the department.

> The Engineering Science curriculum emphasizes an in-depth understanding of the fundamentals of the physical sciences, mathematics, and engineering sciences, which form the foundation for technical work in all fields of engineering. Some specialization is available through elective courses in chemical, electrical and mechanical engineering, taken during the junior and senior years. The program provides significant hands-on experience through engineering laboratories and participation in engineering projects required in eight semester-long design courses. The emphasis on fundamentals is intended to prepare students for dealing with the rapid pace of technology and the interdisciplinary nature of engineering practice. The laboratory and design portions of the program provide the students with a balanced perspective on the theory and practice of the engineering profession.[9]

One impact of this mission is the delineation of courses into fundamentals and elective courses. The fundamentals and design courses are required of all students, and lay the foundation for work in electrical, chemical, and mechanical engineering. The elective courses round out the groundwork for a quality engineering science degree with some specialization in a student's chosen field. As a result, the student base for fundamentals and design courses contains not only engineering students who consider themselves primarily devoted to the study of electrical engineering, but also of mechanical and chemical engineering. The electrically oriented elective courses serve an audience that might be considered more traditional for these primarily introductory courses.

The department offers five courses that utilize digital logic in some fashion. Table 1 summarizes the basic information about these courses.

- In the spring of the sophomore year, all engineering students take Electronics I. This course gives an introduction to both analog and digital electronic circuits. Design and analysis of basic combinational and sequential circuits (e.g. arithmetic circuits, multiplexers, decoders, counters) are covered in depth. The laboratory for this course utilizes SSI chips and breadboards for the implementation of simple combinatorial and sequential circuits.
- In the fall of the junior year, all engineering students take Junior Design. This laboratory oriented design course builds on the Electronics I course and its laboratory to design an electronic circuit that meets certain specifications. In recent years, the course projects have utilized CPLDs exclusively for the design of simple finite state machines. Two recent course projects have been the design of the electronics of a vending machine and the design of a traffic light controller for a major intersection.

- A more traditional course in digital logic design is taught in alternate years. Students take this Digital Logic Design either in conjunction with Junior Design or the following year, as appropriate. This course reviews and expands upon combinatorial and sequential design, as covered in Electronics I, then moves to a more in-depth study of the design of complex combinatorial and sequential circuits up to the level of a simple computer architecture. The laboratory for this course utilizes SSI/MSI logic with breadboards as well as CPLDs on vendor-supplied training boards to implement projects.

Table 1: Summary of Courses Utilizing Digital Logic Design

Course		**Frequency**	**Course Type**	**Number**
ENGR 2364	Electronics I	Every Year	Required of all engineers	30-35
ENGR 2164	Electronics I Laboratory	Every Year	Required of all engineers	30-35
ENGR 3181	Junior Design	Every Year	Required of all engineers	30-35
ENGR 4365	Digital Logic Design	Alternate Years	Electrical Elective	12-15
ENGR 4165	Digital Logic Design Laboratory	Alternate Years	Electrical Elective	12-15

Student Learning Styles

It is well known that attention to students' learning styles can increase learning gains.[2,3] Therefore, one question that should be asked when revising a curriculum, as many departments are doing with their digital logic design curricula, is "what impact will this change have on students with each learning style?" There are several useful definitions for the term *learning style*. In this paper, we will utilize Felder and Silverman's five-dimensional categorization.[3] The five dimensions given in this work include

- perception, on a scale ranging from sensory to intuitive,
- input, on a scale ranging from visual to auditory,
- organization, on a scale ranging from inductive to deductive,
- processing, on a scale ranging from active to reflective, and
- understanding, on a scale ranging from sequential to global.

We give particular attention to the perception dimension. Felder and Silverman's perception scale mirrors the sensing-intuition scale of the Myers-Brigs Type Indicator[1] (MBTI) for personality types.

Carl Jung described sensing and intuition as different ways in which people perceive the world around them.[5] According to Felder and Silverman, a student who is operating in the sensing mode[*] preferentially perceives new information through external stimuli --- sights, sounds, and physical sensations. In contrast, a student who is operating in the intuitive mode preferentially perceives new information indirectly --- through speculation, imagination, or hunches.

---

[*] The mode in which a particular learner is operating can change from moment to moment. Preferences for learning styles can also change over longer periods of time.

Pedagogic Features of Discrete and Programmable Logic

Both discrete logic (the construction of circuits from SSI or MSI chips) and programmable logic (the construction of circuits from more complex chips, such as FPGAs or CPLDs) have pedagogical advantages[8]. A discrete logic implementation is the closest in "feel" to a gate level design, and is thus somewhat more pleasing to "sensors" (students who prefer to learn in the sensing mode) than "intuitors" (students who prefer to learn in the intuitive mode). . The importance of modular design and debugging techniques are more apparent in discrete logic. Intermediate signals can be easily monitored. Programmable logic, on the other hand, offers the opportunity to design and implement much more complex designs. Simulation offers an easy way to monitor all signals, group busses, and validate designs. Wiring mistakes and hardware faults are all but eliminated in CPLD-based designs. It is possible for a student to test several iterations of design in a short time.

However, both discrete and programmable logic have distinct disadvantages as well. Even moderate designs can cause many hours spent debugging hardware and wiring errors. The number of chips required for a design grows quickly with the complexity of a design, and each new iteration of a design is almost as much work to implement as the first. However, the total immersion in the virtual world of the CAD package removes digital circuits from the "constructed" to the realm of the "simulated." If a design is programmed into the CPLD at all, the functional checkout of the circuit is a disappointment to the student. Access to on-chip intermediate signals is limited and slow relative to discrete logic if simulation is not used.

Impact on Student Learning

One impact of disciplinary education is the ability to tailor pedagogical approaches to a narrow target audience. Kolb, in explaining his experiential learning model, describes some aspects of teaching and learning styles that may be attributable to a student's chosen discipline[6]. However, in a diverse environment such as our fundamentals courses, there may be more variety in learning style than in a "standard EE curriculum," if such a thing exists. Our design-oriented program may also draw more "sensors" than "intuitors." It's possible that students who always saw themselves as mechanical engineers haven't had as much previous experience with circuitry as electrical engineering students. McDermott and Shaffer[7] cite the lack of concrete experience with relevant circuits as one of the difficulties students experience in introductory courses. For whatever reason, we have found through informal observation that for a significant part of our student population, the move from physical construction of a circuit on a breadboard to virtual construction of the same circuit in a CPLD design environment inhibits learning[8].

Bridging the gap – moving from discrete to programmable logic

The approach taken by the Engineering Science Department includes both implementation styles at different levels of the curriculum. In Electronics I, when engineering students begin to work with digital circuits, discrete logic is exclusively used. In this course, students design and build two-level combinatorial and up to two-bit sequential circuits using discrete logic. This course serves a general engineering audience, with students that will later select primarily mechanical and chemical engineering elective courses in our curriculum as well as those who gravitate toward the traditionally electrically oriented elective courses.

In Junior Design the following semester, the students familiarize themselves with the use of the CPLD as a "black-box" design tool by completing a simple combinatorial circuit and a four-state traffic light controller before embarking on a more complex sequential design. The students have designed (on paper) both introductory circuits in Electronics I.

Digital Logic Design begins with a review of combinatorial and sequential design, as covered in Electronics I (which may have taken up to three semesters before this course.) During this portion of the course, discrete logic is utilized in the laboratory. At the end of this section, programmable logic devices and memories are introduced. The architecture and uses of Programmable Logic Arrays, Programmable Read-Only Memories, Field-Programmable Gate Arrays, and Complex Programmable Logic Devices are studied. By this point in the course, all students have been introduced to the use of CPLDs in Junior Design. In Digital Logic Design Laboratory, the same sequential design that was done with discrete logic is reimplemented and simulated in the CPLD programming environment. This allows the student to concentrate for a while on learning the mechanics of design entry and simulation, and not on the digital design process. From this point in the course, only CPLDs are used for implementation.

The aggregate result of these courses is that the students begin working with the most direct representation of the outcome of the digital design process, a circuit built from discrete SSI components and debugged on a breadboard. Once reasonably complex circuits have been mastered, including modular debugging techniques and proper construction documentation, the student begins to work with CPLDs as "black-box" implementations of digital designs. Those students that wish to delve deeper into the world of digital logic design then study the architecture and construction of these CPLDs as "white-box" implementations of programmable logic, as well as utilizing the modern design tools (such as hardware description languages) available today.

In this manner, we can introduce engineering students to the modern tools with which they are expected to have familiarity while satisfying the "sensors" in our community that the implementation of digital designs follow logically from the gate-level abstractions they have learned.

Conclusion

By recognizing the pedagogic benefits of both discrete and programmable logic, we have constructed a program that attempts to teach to the varying learning styles of our students. Teachers need to remain sensitive to the needs of our diverse audiences when embarking on curricular reform, and to identify what we lose as well as what we gain when we replace the old with the new.

Bibliography

1. K. Briggs, I. Briggs-Myers. *Myers-Briggs Type Indicator: Test Booklet: Abbreviated Version*, Palo Alto, Calif.: Consulting Psychologists Press, 1983.
2. T. Hein, D. Budny. Teaching to Students' Learning Styles: Approaches That Work. In *Proceedings – ASEE/IEEE Conference on Frontiers in Education* 1999.
3. R. Felder and L. Silverman. Learning and teaching styles in engineering. *Journal of Engineering Education*, 77(2), February 1988.
4. R. Felder, G. Felder, E. Dietz. A Longitudinal Study of Alternative Approaches to Engineering Education: Survey of Assessment Results. In *Proceedings – ASEE/IEEE Conference on Frontiers in Education* 1997.
5. C. Jung, *Psychological Types*, Princeton University Press, Princeton, N. J., 1971 (Originally published in 1921.)
6. K. Kolb. Learning Styles and Disciplinary Differences. In A. Chickering and Associates (Eds.), *The Modern American College*, 1981. San Francisco: Jossey-Bass Publishers.
7. L. McDermott and P. Shaffer. Research as a guide for curriculum development: An example from introductory electricity. Part I: Investigation of student understanding. *American Journal Physics* 60(11), November 1992.
8. K. Nickels. Pros and Cons of replacing discrete logic with programmable logic in introductory digital logic courses. In *Proceedings -- American Society for Engineering Education Annual Conference,* 2000.
9. Trinity University Courses of Study Bulletin, 2000-2001
10. K. Wetzel, K. Harmeyer. Success in Low-Level and High-Level Mathematics Courses in Undergraduate Engineering College as a Correlate to Individual Learning Style. In *Proceedings – ASEE/IEEE Conference on Frontiers in Education* 1997.

KEVIN M. NICKELS
Dr. Nickels is an assistant professor in the Department Engineering Science at Trinity University. He received the B.S. degree in Computer and Electrical Engineering from Purdue University (1993), and received the M.S. degree (1996) and the Ph. D. (1998) in Electrical Engineering from The University of Illinois at Urbana-Champaign. He is currently working in the areas of computer vision, pattern recognition, and robotics. Dr. Nickels has been a member of ASEE since 1998, and a member of IEEE (Robotics and Automation, Computer, and Education Societies) since 1994.

FARZAN AMINIAN
Dr. Aminian is an associate professor in the Department of Engineering Science at Trinity University and a registered Professional Engineer in the State of Texas. He received his B.S. degree in electrical engineering from the University of Oklahoma in 1983 and his M.S. and Ph.D. degrees in electrical engineering from The Ohio State University in 1984 and 1989. Dr. Aminian joined Trinity in 1989 and is currently serving as the chairman of the Central Texas Section of IEEE.

J. PAUL GIOLMA
Dr. Giolma is an associate professor in the Department of Engineering Science at Trinity University and a registered Professional Engineer in the State of Texas. He received his BSEE degree from the University of Florida (1969), the MSEE degree (1971) and the Ph.D. (EE/Biotechnology, 1975) from Carnegie-Mellon University. Dr. Giolma has been a member of ASEE since 1981 and has served as chair of the New Engineering Educators, chair of the Gulf Southwest Section, and as a member of the Board of Directors as chair of Zone III. He is a member of IEEE (Signal Processing, Engineering In Medicine and Biology, and Education Societies), Sigma Xi, Tau Beta Pi and NSPE.