A Practical One-Semester "VLSI Design" Course for Computer Science (and Other) Majors

Robert A. Walker

Kent State University
Department of Math and Computer Science
Kent, OH 44242
walker@mcs.kent.edu

Abstract

This paper describes the development and content of a "VLSI Design" course. We had two main goals for the course: to develop a one-semester course for computer science (and other) majors, and to give the students practical experience with real industrial tools. To meet those goals, we provided only enough material on logic design and IC operation to orient the students, focused on FPLD-based design, and used high-quality design tools. We also found a wealth of free material available on the web, including course information, entire textbooks, and extensive technical data.

1 Introduction

This paper describes the development and content of a "VLSI Design" course for computer science majors. *VLSI Design* is the process of designing a large computer chip (more specifically, an *integrated circuit*, or IC), using semi-automated *computer-aided design* (CAD) tools on a workstation or personal computer (PC).

Most Computer Engineering and some Computer Science programs include a senior / graduate level course on this topic. However, such computer-engineering-oriented VLSI Design courses usually have a large number of prerequisites, while one of our main goals was to develop a self-contained one-semester course for computer science (and other) majors. Furthermore, those courses tend to emphasize IC fabrication and detailed design at the transistor level, while another of our goals was to avoid those details and give our students practical, real-world experience that would let them use real VLSI design tools to program an actual chip in the final design project.

The result was a one-semester course that blended theory with practice and gave the students a useful skill. This paper describes that course as follows. Sections 2 and 3 examine the development of the course and briefly summarize the final content. Section 4 explains our problems in choosing a textbook, and our innovative solutions to those problems. Sections 5 and 6 describe the VLSI design tools we used and the hardware in our VLSI design lab (and make some interesting observations regarding the use of "old" PCs), and Section 7 describes the final student projects. Finally, Sections 8 and 9 evaluate our experience and present our conclusions. For those interested, all course materials are available online at http://www.mcs.kent.edu/~walker/classes/vlsi.s98.

2 Developing a One-Semester VLSI Design Course

For a student who wants to design "computer chips", the typical plan of study would involve a sequence of electrical engineering courses. At the University of California of Berkeley, for example [2], the student would take an introductory course in analog and digital circuits (EE40), followed by courses in electronics (EE105), the internal operation of ICs (EE130), digital ICs (EE141), and the processing and design of ICs (EE143). Finally, at the graduate level, the student could take a course in computer-aided design of ICs (EE244)—the long awaited "chip design" course[1]. At the end of this course sequence, the student would be well versed in logic design[2] and the device physics and internal operation of an IC, and would have at least one full semester of using CAD/VLSI tools.

In early 1998, we decided to develop a new VLSI Design course for CS majors at Kent State that would offer practical, real-world skills. The course would be available at both the senior and beginning graduate level, and the

---

[1] This plan of study may also include a microprocessor design course [5, 6], which typically provides additional training in digital circuit design and interfacing.

[2] The terms *logic design* and *digital circuit design* are generally considered synonymous. This paper will usually use logic design.

Proceedings of SIGCSE'99, pp. 237-241

only prerequisites that we would assume were courses in computer organization and computer architecture. We specifically did not assume the students had much prior exposure to logic design or IC operation.

The planning stage for the course was fairly extensive, and drew on a variety of sources. First, of course, was the web. The excellent list of computer science course web pages at http://www.cis.temple.edu/courses.html, and the less-extensive list at http://lamar.colostate.edu/~beaty provided a good overview of VLSI design and related courses offered throughout the world. We also browsed the web pages of the various computer engineering publishers to look at their textbooks, and visited their booths at the trade show accompanying the Design Automation Conference in June 1998. Finally, we talked to professors at other universities that were teaching VLSI design courses.

The key insight that arose during this planning was that there was no way we could expose students to all aspects of VLSI design in a single semester, even by including only the "necessary" prerequisite information needed for our goals. Instead, we decided to concentrate on one particular aspect of VLSI design—design using *field-programmable logic devices* (FPLDs), and cover only enough material on logic design and IC fabrication to orient the students.

Integrated circuits are implemented in a variety of fabrication technologies, depending on performance needs, economics of scale, etc. The old, traditional VLSI design course usually focuses on *full-custom application-specific ICs* (*ASICs*). These full-custom ASICs are built from scratch, and are used mainly for high-performance designs. Other VLSI design courses focus on *standard-cell ASICs*. These ASICs are built from pre-designed components called *standard cells*, and are used for high-volume (>20,000 chips) designs.

Yet a third set of VLSI design courses (and even some logic design courses) focus on *field-programmable logic devices* (FPLDs). As with standard-cell ASICs, the designer builds using pre-designed components. However, instead of sending the resulting design to a foundry to be fabricated into a chip, the chip can be "programmed" on blank ICs by the designer in a matter of minutes (hence the term "field programmable"). FPLD technology is usually used for low-volume (<2,000 chips) designs.

The bottom line is that teaching FPLD-based design is a perfect choice for a one-semester VLSI design course for computer science majors. First, the use of pre-designed building blocks is a familiar abstraction for computer science students, who are used to working with libraries of functions when programming. Second, the tools can be used without a deep understanding of electronics, IC device physics, etc. Finally, many designs are actually built in small quantities, so teaching students FPLD-based design gives them a realistic and useful skill.

How does this VLSI Design course compare to those previously described at SIGCSE? In 1996, Concepcion and Millican [3] argued that VLSI Design courses where the students actually fabricate chips are prohibitively expensive, and that it is sufficient to use powerful CAD tools and simulations. We agree with their advocacy of powerful design tools and simulations, but with recent changes in technology, we advocate the use of FPLD-based design instead of full-custom ASIC design. Two years before, Gschwind [7] also advocated the use of reprogrammable hardware, but the technology has changed so much since then (for example, they used only schematic capture and simulation tools) that his paper serves today more as an advocate of FPLD-based design than as a model for a current VLSI Design course.

3 Course Outline

The first quarter of our course introduced the students to VLSI design. Four lectures discussed the various implementation technologies, so the students got a clear picture of what techniques they would and would not learn. Three lectures briefly reviewed combinational, sequential, and state machine logic design techniques—not in great detail, but enough to orient the students. Finally, five lectures described the use of a CAD tool (VIEWlogic's Workview Office, described later) for schematic capture and simulation—one which performed various "bookkeeping" and analysis functions, but which still required the students to manually design the desired circuit.

Given this background, the second quarter of the course had two main goals. First, it introduced IC fabrication techniques and discussed various alternatives for FPLD-based design (e.g., the products available from the three major FPLD vendors). While the students were learning this new material in lecture, they were also going through tutorials on their own to learn VIEWlogic's Workview Office, and then implementing a project of their choice.

After the general introduction to FPLD-based design in the second quarter of the course, the third quarter focussed on the use of one specific set of FPLD-based design tools (Altera's MAX+PLUS II, described later). Unlike Workview Office, which required the student to actually design the circuit, MAX+PLUS II let the student specify the circuit at a higher level, and then it took care of the details of actually synthesizing the design.

Finally, in the fourth quarter of the course, the students used MAX+PLUS II to design a final project, while at the same time the lectures introduced them to high-level synthesis, the next higher level of abstraction in the process of automated chip design.

4 Choosing Textbooks

One of the major problems we faced in designing such a diverse course was choosing a textbook. We wanted to cover logic design, IC technologies, FPGA-based design, and high-level synthesis, and we needed tutorials showing the students how to use the two VLSI design tools, but we didn't want to require the students to buy a large number of expensive books. We found that most "VLSI Design" books focused too much on full-custom design, and required too much prerequisite knowledge. Most "logic design" books covered only the basics of logic design, and little on ICs. Most books were very general and omitted details (including our much-needed tutorials) on specific design tools. In the end, we chose a unique approach unlike any we had used previously.

For the first quarter of the course (the introduction to VLSI design), we used several different references. Neither of our two primary texts ([4] and [9], described later) gave a good overview of the various implementation technologies, but we were fortunate enough to discover that the author of an excellent new book on VLSI design ([10]) was generous enough to put the entire book online! Similarly, most of the lecture material on logic design was written locally, but students wanting more information could refer to the online version of a well-known text on that topic ([8]).

Finally, for a tutorial introduction to our first CAD tool (VIEWlogic's Workview Office, described later), we used our first primary text—[4]—for two reasons. First, it was the only tutorial we were able to locate in a commercially available textbook! Second, it included a copy of the software on CDROM, so those students who wanted to work on their home PCs (as opposed to those in our VLSI Design Lab) could do so.

For the second quarter of the course, we again used [10] for the introduction to IC fabrication technologies, and we used our second primary text—[9]—for an overview of the various FPLD products available from different vendors. We heavily supplemented this information with the full data sheets on those products, available online from the vendors' web sites. We were pleasantly surprised by the wealth of information on those sites, which also included product comparison guides, application notes describing the use of the products, etc.

For the third quarter of the course, which was the detailed introduction to our second CAD tool (Altera's MAX+PLUS II, described later), we again used our second primary text—[9]. Most of that book was devoted to an overview of that tool and the various design languages it supported, but it also had two chapters with extensive large examples. We supplemented the overview with a detailed tutorial from Altera's web site [1], and the information on the design languages with detailed information available from the tool's help facility (which is quite extensive). Like [4], [9] also included a CDROM, making it possible for those students who wanted to work at home to do so.

Finally, for the fourth quarter of the course (the introduction to high-level synthesis), we used course notes written locally for a previous course on that topic.

In summary, one of the major challenges in developing this course was choosing the textbooks. We eventually settled on two primary texts, each tied to a specific CAD tool and containing a copy of that CAD tool on an accompanying CDROM. We supplemented heavily from materials freely available on the world-wide-web: two textbooks that were available in their entirety, and extensive product information available from the FPLD vendors.

5 The VLSI Design Tools

Two VLSI design tools were chosen for the course. The first, VIEWlogic's Workview Office, is a schematic capture and simulation tool, and the second, Altera's MAX+PLUS II, is a full-featured logic synthesis tool. Both were installed in our department's VLSI Design Lab (see the next section), and were available in the two primary textbooks for the students who wanted to use them at home on their own PCs. The tools are briefly described here, and our experiences using them are summarized in Section 9.

VIEWlogic's Workview Office is a schematic capture and simulation tool. By supporting schematic capture, it allows the student to graphically enter a design by placing and connecting logic gates and higher level components. It supports hierarchy, so students can design their own components, each of which consists of other lower-level components, and it performs design consistency checks, warning the student of invalid design practices. Once the design is entered, it supports simulation, meaning the student can provide binary waveforms for each input to the circuit, and the tool will simulate the circuit and supply the output waveforms that will result. Using the simulator, the student can "test" the circuit under a variety of inputs to verify that it actually performs as expected.

In contrast, Altera's MAX+PLUS II is a full-featured logic synthesis tool. Instead of making the student design the circuit by hand and then use schematic capture to enter the various gates and flip-flops, it allows the student to "design" by specifying the behavior at a higher level (e.g., as a textual specification of a state machine). Then it automatically synthesizes the design, selecting and interconnecting the necessary gates and flip-flops.

6 The VLSI Design Lab

The VLSI Design tools chosen were available in different formats. A severely limited student version of VIEWlogic's Workview Office came on a CDROM in [4], and ran on PCs under Microsoft Windows. A slightly limited student version of Altera's MAX+PLUS II came on a CDROM in [9], and also ran on PCs under Microsoft Windows. Altera

Corporation also donated a copy of the full commercial Unix version of that software to our VLSI Design.

Since this was the first time a VLSI Design course was taught at Kent State in over 5 years, none of this software had been previously installed at Kent State, and even worse, there was no “VLSI Design Lab” available for the course. However, we did have some research lab space available, containing an HP C180 workstation and a Gateway P6/300 PC, so that lab was designated the “VLSI Design Lab” and used for the course.

Furthermore, we found several old “broken” PCs of questionable value that were donated for use in the lab. These included 2 Gateway P5-60’s, 2 Cybermax DX4/100’s, and one Elite Micro DX2/66. Only one of these PCs worked when we first tried them, one had a broken disk drive, all had 16MB of memory or less, and none had any software newer than Windows 3.1. (In contrast, our Gateway P6/300 was much more powerful, had 64 MB of memory and a 4MB disk, and ran Windows 95.) However, with the investment of some new memory chips, new disk drives, and one new monitor, we were able to put 3 Windows 95 machines and 2 Linux machines into service, all with 1MB drives and at least 32 MB of memory.

Although the resulting systems still had severely limited memory and disk space, and obsolete CPUs most high school students would laugh at, they worked surprisingly well in the lab! Compiling large designs ran slower on those systems than on the Gateway P6/300, but not slow enough that everyone chose the “fast” machine as their first choice. Memory and disk space was limited, but more than sufficient to install and run the design tools. In an age where we’re constantly exhorted by the media to buy “bigger, better, faster” machines, we were very pleasantly surprised to find that these old, obsolete machines we rescued from the scrap heap were perfectly adequate machines for our purposes.

7 The Projects

There were two main projects handed out in the course. The first was a three-part project using VIEWlogic’s Workview Office, and the second was a three-part project using Altera’s MAX+PLUS II.

The first project began by introducing the students to VIEWlogic’s Workview Office. The students were asked to do two of the lab exercises in [4], going through the tutorial to design a four-bit ripple carry adder, and then designing a four-bit arithmetic unit. After these preliminary design experiences, they were given their choice of three projects in that book, or an equivalent project of their choice.

The second project began by introducing the students to Altera’s MAX+PLUS II, with the students asked to do the tutorial available online. Then they were asked to propose a final project for approval, and to work on that final project during the last few weeks of class.

One of the more interesting parts of this project was that not only were the students required to design, simulate, and test their project using MAX+PLUS II, but they were required to actually program a FPLD chip and demonstrate that their design “really” worked. For these projects, each PC in the VLSI Design Lab was connected to an Altera UP1 Education Board. Each of these boards contained two different FPLDs, and a variety of switches, push buttons, LEDs, and seven-segment displays.

The final projects took each student several days or more to design, and then anywhere from a few minutes to several days to get working on the UP1 Education Boards (depending on how well the student had tested the design during simulation).

Some of the more interesting student projects were the following, in no particular order:

- Prime number detector (computer science graduate student)—one of the few projects that “worked” the first time due to good use of modular testing and debugging; an interesting implementation of a fairly complex arithmetic algorithm (tested in advance using a C program)
- 3-state electronic lock (computer science undergraduate student)—excellent overview of the design and major design decisions; good description of the need to “debounce” the push-button to avoid glitches
- Programmable door alarm (computer science undergraduate student)—challenging design that pushed the limits of the AHDL design language; also a design that found an error in the design tools
- Tic-tac-toe game (computer science graduate student)—complex state machine broken down into a number of smaller components
- 4x4 multiplier (computer science graduate student)—designing a complex module from scratch; good understanding of when to use binary and when to use binary-coded-decimal
- Robot controller (computer science graduate student)—only project designed, debugged, and downloaded to hardware without any help from the instructor
- Look-up table (technology graduate student)—the only design that use the RAM / ROM capabilities of the FPLDs; good description of the need to “debounce” the push-button to avoid glitches
- “Chorded” switches (mathematics graduate student)—complex state machine, many hand-designed components
- Vending machine (computer science graduate student)—complex state machine

We definitely achieved our goal of developing a course accessible to computer science (and other) majors. Most of the students in the course were computer science majors, and two were in technology and in mathematics.

8 Evaluation

In general, the course was quite successful. On the student evaluations at the end of the course, the students:

- rated the course highly (58% excellent, 25% very good, 17% good),
- thought the work load was appropriate (33% strongly agree, 50% agree, 17% tend to agree), and
- rated the instructor's teaching highly (67% excellent, 25% very good, and 17% good).

However, there were some problems as well. VIEWlogic's Workview Office was definitely less "polished" and professional than Altera's MAX+PLUS II. The student edition was severely limited, there was almost no documentation available, the help menus that explained the toolbars didn't work, and the program was plagued with numerous minor bugs and glitches. In contrast, there were almost no problems with Altera's MAX+PLUS II, and its online help facility was quite comprehensive (although to be fair, that software is much more expensive).

9 Conclusions

The most important conclusion that we reached is that it is definitely possible to design a useful, practical, one-semester course in VLSI design, and one that can be taken by students other than computer engineering majors. The key is to provide only enough material on logic design and IC operation to orient the students, to focus on FPLD-based design, and to use high-quality design tools.

A second, more general, conclusion is that there is a wealth of free material available on the web. Although most computer science instructors are familiar with the benefits of placing their course materials on the web, they may not be familiar with the benefits of searching the web for materials to improve their courses or their own skills as educators. We found that lists of courses are available, which make it possible for one educator to learn from more experienced peers. We also found that entire textbooks can sometimes be found online, and for those courses that use commercial hardware or software, extensive technical data is often available on the company's web site.

Finally, a minor but surprising conclusion was that an educational lab doesn't always need the latest and most expensive PCs—old, "obsolete" machines may be perfectly adequate, given a small investment in additional memory and new disk drives. Given the extensive media hype over new CPUs, clock speed, and cache size, it was surprising just how little such things matter for all but the most computationally oriented tasks.

10 Acknowledgements

The author thanks Kent State University for providing him with the startup funds necessary to build the VLSI Design Lab. He also thanks Altera Corporation for donating the full commercial version of the MAX+PLUS II software and several UP1 Education Boards to the lab, and for providing the students in the course with CDROMs of technical literature and hardcopy manuals describing the AHDL and VHDL design languages.

11 References

[1] Altera Corporation. *MAX+PLUS II Getting Started*. http://www.altera.com/html/literature/lsoft.html.

[2] http://www.eecs.berkeley.edu/EE/Students/classes.shtml

[3] Concepcion, Arturo I., and Millican, David R. "Developing the VLSI Laboratory for the Computer Architecture Course". In *Proc. of SIGCSE'96*, pp. 47–52.

[4] Duckworth, R. James. *Workview Office Student Edition: Schematic Entry and Digital Analysis*. Prentice Hall, 1997.

[5] Estell, John K., and Owen, Thomas A. "Experiencing the Codesign Process: Microcomputer Systems II Laboratory". In *Proc. of SIGCSE'95*, pp. 34–38.

[6] Ewing, Donald J. "Microcomputer Systems I: A Computer Science and Engineering Capstone Course". In *Proc. of SIGCSE'93*, pp. 155–159.

[7] Gschwind, Michael. "Reprogrammable Hardware for Educational Purposes". In *Proc. of SIGCSE'94*, pp. 183–187.

[8] Katz, Randy H. *Contemporary Logic Design*. Benjamin Cummings / Addison-Wesley, 1993. http://www.cs.berkeley.edu/~randy/CLD/CLD.html

[9] Salcic, Zoran, andSmailagic, Asim. *Digital Systems Design and Prototyping Using Field-Programmable Logic*. Kluwer Academic Publishers, 1997.

[10] Smith, Michael John Sebastian. *Application-Specific Integrated Circuits*. Addison-Wesley, 1997. http://spectra.eng.hawaii.edu/~msmith/ASICs/HTML/ASICs.htm