AC 2011-2257: DOES LOGIC SYNTHESIS VISUALIZATION EXCITE EETS?

MADDUMAGE KARUNARATNE, University of Pittsburgh at Johnstown, PA

Dr. Maddumage Karunaratne is an Associate Professor and the Head of the Electrical Engineering Technology department at the University of Pittsburgh at Johnstown, PA. The department offers two undergraduate degrees in Electrical Engineering Technology and Computer Engineering Technology. Dr. Karunaratne earned a Bachelor of Science degree from the University of Moratuwa (Sri Lanka), a Master of Science from the University of Mississippi (Oxford), and a Ph.D. from the University of Arizona (Tucson).

Before joining academia, he gained fourteen years of extensive industry experience working in the semiconductor industry in software development, application engineering, design, testing and verification of digital integrated circuits. He has taught electrical and general engineering technology classes at Pitt-Johnstown for the last seven years.

His research and teaching interests include Semiconductor circuit Testing and Verification, Low Power Design Analysis, Digital and Embedded Systems, Electromagnetic Wave Scattering, and IC Design Automation Software development.

He can be reached at maddu@pitt.edu or at 225 Engineering and Science Building University of Pittsburgh at Johnstown Johnstown, PA 15904

©American Society for Engineering Education, 2011
Does Logic Synthesis Visualization Excite EETs?

Abstract:

Based on industry trends and recent advances in low cost silicon manufacturing technology, it is becoming apparent that in the future, electrical and computer engineers will most likely implement their digital designs, whether large or small, using programmable logic devices such as CPLDs and FPGAs rather than discrete electronic components like integrated circuit chips, transistors, resistors, etc. This is becoming a reality in both small scale and medium scale production volumes where quantities can run into thousands. On the other hand, even those who are not focused on building careers based on digital circuit design will probably encounter electronic systems built on such devices in their professional work. Therefore, it has become necessary to introduce related courses at undergraduate level along with a considerable number of hands on laboratory sessions as well.

This paper discusses the teaching and enhancements made to such courses in digital design to undergraduates majoring in Electrical Engineering Technology (EET). The author will elaborate the attempts taken in promoting a certain level of excitement in students during the digital design course. The paper also describes several considerations taken into account in the adaptation of Verilog Hardware Description Languages (HDL) and automation based digital design flow to the EET curriculum several years ago. The demographics of the particular student population and their immediate careers suggested that most students would not pursue graduate studies in computer engineering, nor would they seek employment related to design and manufacture of Integrated Circuit components. As a result, a much broader and deeper study of modern digital design methodologies has not been a necessity. The informal feedback received from recent graduates in the industry has validated the adopted approaches while those pursuing graduate studies appreciate the depth of the curriculum as well.

At undergraduate level, digital designs described in HDL can start at two main different abstraction levels: at the schematic level, or at the register transfer level. Traditional design courses are geared toward schematic based implementation flows which are tedious but less challenging for novice students than higher abstraction levels which demand comprehension of advanced design. However, incentives have been incorporated in to student work to describe hardware designs at higher textual levels, and to use software packages for graphically viewing their low level implementations on computer. This circuit visualization was to provide a level of excitement among students to obtain variations in final implementations of their designs on programmable logic hardware boards. The paper will also present results from a student survey taken at the end of the course to gauge the effectiveness of HDL based design flow, and how the graphical viewing of the final circuits that students designed at a higher textual level excited and incentivized students in the undergraduate EET program at this university.

Part I: introduction:

Until recent years, if not still, student assignments in digital design courses at undergraduate level consisted primarily of paper designs, or at best such as in senior design projects they were a large morass of SSI and MSI (Small and Medium Scale Integration) silicon devices plugged on to circuit
wiring (bread) boards. A higher percentage of time and effort were devoted on debugging the connections and wirings than on actual design or in optimizing the functions of the circuit. Even after HDLs such as VHDL have become standard and widely being used in commercial integrated circuit chip making industry, undergraduate academic curriculums in Electrical and Computer engineering were very slow to adopt them\textsuperscript{[1]} because of the cost of the hardware and associated Computer Aided Design (CAD) tools coupled with lack of faculty experience on such design flows required for digital designs.

For many years, at college level courses that utilized HDL, the designs were limited to use them only for design and simulation steps. Cost of programmable hardware development boards and associated software packages kept them away from most digital design courses. This scenario has changed with the advent of cheaper computers, inexpensive Programmable Logic Devices (PLDs), and the associated development environments. Also, tool vendors such as Xilinx, Altera, Cypress, etc., have realized that they need to support the teaching community to train future engineers to utilize their wares for it offers them a competitive advantage over the purchasing choices that such a trained graduate would make in the future. Such openness and education discounts have created other third party vendors to manufacture and support very inexpensive hardware development boards targeting mainly the college education market. The flexibility of PLDs and the availability of inexpensive CAD tools with huge academic discounts, if not free, from the industry that support these devices make them ideal for a wide range of digital design courses. Developing designs in hardware languages and associated methodologies teach students the skills to implement large and complex circuits using high level abstraction and adds marketability of engineering graduates.

Prior to adaptation of HDL based flows, digital designs were described on paper or on graphical schematics CAD tools by connecting the vendor supplied icons of building blocks named technology cells. Then the databases created for the design became proprietary and third party tools could not read them. Also, the process of entering a design in that manner was tedious and error prone. Compared to direct schematic-based designs, HDL based design flows provide the following advantages \textsuperscript{[2 ~ 6]} which students like:

- Designs can be transcribed at various levels of abstraction. Designers can write their register transfer language (RTL) descriptions without choosing a specific fabrication technology. Logic synthesis tools can automatically convert the design to any fabrication technology. This was particularly attractive to students who always seek ways to complete their work as quickly as possible. The students clearly preferred writing most of circuit descriptions at RTL level.

- Describing designs in an HDL allows functional verification of the design early in the design cycle. However, the highest behavioral level modeling does not provide sufficient details to verify the correctness of the final implemented circuit. By working at the RTL level, they can optimize and modify the RTL description until it meets the desired functionality. Most design bugs are eliminated at this point. This significantly decreases design effort.

- Designing with an HDL is analogous to computer programming. A textual description with comments is an easier way to develop and debug circuits. Gate-level schematics are
cumbersome and functionally opaque for complex digital designs. All the students have taken C programming course and therefore are comfortable in writing software.

- Drawing the circuit diagram by hand becomes time consuming and messy on paper when it involves close to a hundred gates. In the HDL flow with a synthesis target of a PLD, one can easily obtain a navigational hierarchy based schematic sheets for any size circuit. Students like this feature since they see a clean, optimized, block based diagram of their design.

Historical industry trends show that future electrical and computer technology engineers will implement their digital circuits and systems using programmable logic devices such as CPLDs and FPGAs, rather than discreet IC components. With the rapidly rising costs for the creation of silicon lithography masks succeeded by actual device fabrication at nano-fabs, only a very minute fraction of digital designs would be made in either Application Specific Integrated Circuit (ASIC) technology since such a cost would be justifiable with ultra high volume of millions of IC chips. Teaching colleges should also prepare their future graduates to be able to function effectively in the real world. As expected, it is apparent \[4-8, 16-23]\ that the academic departments have been adding new courses or new technical content to teach HDL based design flows at some level to undergraduate EET students.

Therefore, to help the graduates to be competent and competitive in their major disciplines, the EET majors are exposed to the HDL based digital design techniques at the undergraduate level. There were several key considerations taken into account during the adaptation of Hardware Description Languages (HDL) and automation based digital design flow to the particular curriculum. The demographics of the student population and their immediate careers suggest that most graduates do not pursue graduate studies in computer engineering, nor do they seek employment related to design and manufacture of Integrated Circuit components. However, there have been a few select students in recent years who gained employment in areas specialized in development of digital systems overlapping into programmable logic devices. The limited skill set instilled upon them was adequate for them to further their expertise confidently.

The primary objective of EET program at this institution has been to provide a broad skill set to graduates to work in various local and regional industries including the defense sector, and therefore the EET curriculum does not provide formally declared specialization areas. The core curriculum schedule accommodates only one digital design course namely EET 1161 “Digital Electronics” in the junior year. In the senior year an elective is offered looking deeper in to system level design using software/hardware co design, design verification, testing, design for test, and advanced design concepts. The EET 1161 course is accompanied by its associated laboratory course, EET 1061. The academic year provides 15 weeks for the courses and about three hours contact time a week for each course.

Part II of the paper describes the Course objectives and Content along the description of selected steps of the typical HDL based design flow to suit the particular EET curriculum; Part III elaborates on laboratory experiments and project assignments which are done at various levels of HDL language. They also include logic synthesis targeting PLDs just for viewing schematics generated from the RTL. Part IV ponders upon the results of a survey done on students in the course at the end
of the semester to determine the excitement *cool factor* of just writing a circuit and viewing it on computer auto generated by synthesis software. Part V states concluding remarks with the future treads.

Part II: course content and objectives

The objective of the course and the lab were to provide students with a skill set so they would be able to analyze, develop, design and implement a reasonably large and complex digital circuit using either discrete components or a PLD. When wiring the circuit in discrete parts seems tedious students are advised to use a CPLD development board for lab work. Table 1 shows the lecture topics and the relevant lab topics for the course.

The theme of the course may be viewed as consisting of the following four sections:

- **Section A**: Digital logic foundations such as number systems and arithmetic, Boolean algebra and expressions;
- **Section B**: Combinational logic design, verification, and implementation
- **Section C**: Sequential logic design, verification, and implementation;
- **Section D**: Designs involving hierarchy and widely used circuit blocks.

The course topics at this institution are not partitioned in to above four sections; rather the course is organized into topics in the sequential order as shown in TABLE 1 to provide a progressive learning flow.

<table>
<thead>
<tr>
<th>Lecture Topics</th>
<th>Lab Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Information representation in Digital Systems: decimal, binary, and hexadecimal systems;</td>
<td>Lab 1: introduction of logic families, digital trainers, threshold voltages, etc.</td>
</tr>
<tr>
<td>2 <em>Boolean Algebra</em> and function minimizations using Karnaugh Maps, etc</td>
<td>Lab 2~3: Boolean functions and Boolean algebraic manipulations, minimizations</td>
</tr>
<tr>
<td>3 <em>Combinational Logic elements</em>: Basic and complex logic gates; high impedance output gates; partitioning for single and multi level optimizations.</td>
<td>Lab 4~5: Combinational logic design, design partitioning, Boolean manipulations</td>
</tr>
<tr>
<td>4 <em>Combinational Logic Design and Functions</em>: Top-down and bottom-up design and hierarchy; Design automation and implementation methods; Role of Hardware Description Languages in design automation; Technology libraries; propagation delays; simulation and verification of circuits; decoders; multiplexors; encoders; Logic circuit</td>
<td>Lab 6: Combinational logic design with gate level Verilog HDL, testbench creation using Verilog procedural blocks, Verilog models of standard hardware components to emulate hardware for functionally verifying the designed circuit.</td>
</tr>
</tbody>
</table>
synthesis using pre-designed function blocks.

Lab 7: Parameterized designs, hierarchical designs, Verilog testbench developments, and the need for non-exhaustive testing

5 Arithmetic Circuits: Iterative combinational circuits; adders; subtractors; arithmetic overflow; multiplication by constants and vectors; increments and decrements.

Lab 8: Sequential adder using iterative adder blocks implemented in hardware LSI/MSI silicon parts.

Lab 9: Combinational logic arithmetic circuits simulated and implemented on CPLD boards.

6 Sequential Circuit Design: sequential logic elements and timing diagrams; sequential circuit analysis and design; state diagrams; state assignments; one-hot encoding; Different abstract levels of Verilog HDL language, simulation and verification of sequential designs.

Lab 10: Design of finite state machines in HDL and verification by simulation using testbenches.

7 Sequential Design Blocks: Analysis and design of data registers, universal shift registers, ripple and synchronous binary counters; Sequential design partitions as data and control sections.

Lab 11: Design of two sequential circuits in hierarchical partitions based on control and data processing modules; their implementation on CPLD.

Table 1: Digital Electronics Course Curriculum

Students learn the materials in Section A in the first few classes of the course so that they know the algebra of the foundation of digital designs rather than merely relying on CAD tools for logic optimization. It is followed by learning the basic logic gates, and Verilog HDL is first introduced at that time to describe combinational primitive logic gates. Students who take the course have already done C-programming and the similarity between Verilog HDL and C language helps significantly as does the prior exposure to number systems, logical and relational operations, decisions statements, and loops.

The inventors of Verilog language [11] wanted a language with syntax similar to the C programming language so that it would be familiar to engineers and readily accepted. The language is case-sensitive, has a preprocessor like C, and the major control flow keywords, such as "if" and "while", are similar. The formatting mechanism in the printing routines and language operators and their precedence are also similar. Verilog HDL was selected as the language over VHDL primarily because all the students will have gained the prerequisite experience in C programming for one semester.

After the theory and techniques in Combinational Logic Design (Section B) and Functions have been covered, Verilog is introduced more formally with the concepts of abstraction and encapsulation (of modules). As Table 1 outlines, students learn hierarchical design and design
reuse; technology libraries and ASIC design flow; delay modeling at gate level, role of Hardware Description Languages in design automation; simulation and verification of circuits; decoders; multiplexors; encoders; and other typical functional blocks. Verilog HDL is introduced via examples for gate level designs already completed on paper by drawing the diagrams. In the Verilog HDL itself, students learn gate level and RTL level circuit modeling, concurrent assignments for combinational designs, and procedural blocks for testbenches.

More semantics complexities arise in Verilog when Sequential Circuits (Section C) are described at RTL level which is the preferred level of abstraction for sequential designs when targeting Complex PLDs (CPLDs). Students learn classical sequential circuit design and then use Verilog to describe them rather than directly coding from the problem specifications. However they are introduced to the RTL coding style of finite state machines which follows the switch-case statements in C language. For Section D, the sequential circuit blocks such as counters, shift registers, and linear feedback shift register (LFSR) modules are taught along with Verilog RTL coding styles.

Part III: class projects and laboratory experiments

Lab assignments typically reflect on the material covered in the lecture class. Students use ModelSim\textsuperscript{[9]} for simulation and validation of their design work. They develop testbenches for verification of their designs. As the table 1 shows, lab assignment 7 introduces parameterized designs to allow resizing of the same module to build larger circuits. It also demonstrates that a circuit may not be tested exhaustively even when it seems simple and easily renders itself for exhaustive testing.

About ten weeks into the course, after students have practiced writing RTL code, they designed two combinational circuits in a laboratory assignment. One circuit is to control a traffic light junction while the other was to evaluate a polynomial function when all the coefficients are also external inputs. After the circuits have been verified via the Modelsim simulations, students used Xilinx CPLD based boards with ISE CAD software\textsuperscript{[13]} to synthesize the circuits. Students were to view different levels of hierarchy and navigate the schematics created by ISE software. From the observed reactions it was apparent that they really enjoyed seeing the schematics of the textual design they created.

The follow up lab assignment was to expand the traffic light problem to design the sequencing counter and the sequential control logic to make it a standalone digital system. The other problem was to redesign the polynomial circuit to evaluate it step by step sequentially in order to optimize for circuit size. At the same time students were to learn how to partition a design from a verbal specification into data and control sections to be designed as individual modules with hierarchy.

In this course, the Verilog language is treated as a tool, and the effort and emphasis placed on teaching Verilog can be estimated to be about 25% of the course load, which can not be increased. Since it is the only digital design core course that these EET students take and any increase in Verilog content would adversely affect the teaching of design principles. Also adding another core course simply to expand the amount of Verilog language covered in the curriculum is not needed.
because such a broad spectrum is required only for those who would be working in design for manufacture of integrated circuit components, and not for system designers or system integrators of logic devices. On the other hand those who have gone to graduate studies have not found a lack of the depth in certain areas adversely affect pursuance of their desired area of study in graduate programs. As a result, a much broader and deeper study of modern digital design methodologies is not a necessity for the majority of the EET undergraduates here.

Since the teaching of Verilog HDL is not the major emphasis nor it is taught heavily in the course, students do not show lack of interest or boredom in the course. Although both students and the instructor like to increase the amount of time in learning Verilog, the time limits and core learning required prevents it. Both parties however think that students are exposed to a sufficient amount of Verilog HDL in design abstraction levels, coding styles, and implementation options that students would be able to self-learn and become effective in implementing large and complex designs on programmable devices or other means.

Part IV: student feedback on visualization

Students were surveyed at the end of the course to gauge the effectiveness of HDL and associated tools in learning digital electronics, although the real benefit would occur when they graduate and find employment in computer related fields. The familiarity and exposure to a HDL based design methods and programmable logic devices would enable them for life-long learning in changing technologies.

The survey was conducted on a small class (a typical electrical engineering class has less than 20 students) at the end of the EET1161 lecture course and EET1061 labs, and all of the students participated. The results are summarized qualitatively in Table 2. The responses were given based on the typical rubrics of 1 to 5 (1-strongly disagree, 2-disagree, 3-neutral, 4-agree, 5-strongly agree). Column 2 of Table 2 lists what each question was trying to assess from students’ perception and knowledge. The 3rd column shows the average rubric score given by students for each question. Final column in Table 2 attempts to give a qualitative meaning to the data in column 3 so that whether the Verilog HDL content its delivery methods need to be changed for better learning of HDL based design flows.

Almost all the questions posed to students received an affirmative answer ranging from “ok” to “yes” in the qualitative score indicating that the HDL flow and visualization approach gave them a great outcome on the two courses taught. The question 7 in Table 2 carries the qualitative measure of “yes” which states that future students would benefit by increased Verilog HDL content in the course, at least which is the message from the last semester EET students in EET1161/1061 course. Looking at the responses to the questions, it is an encouragement to see students showing a desire to learn new technologies, new methods and approaches, which in turn encourages addition of more modern practices into the curriculum. The obvious disadvantage of introducing HDL based flow would be that it reduces the time available to teach certain topics in depth and prevents some more topics being added. With the C-programming background students have, a major disadvantage of having to learn programming techniques anew was not present in this case.
<table>
<thead>
<tr>
<th>Essence of the question asked</th>
<th>Tag</th>
<th>Average</th>
<th>Qualitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Students were knowledgeable at low level Verilog</td>
<td>V1</td>
<td>4.4</td>
<td>yes</td>
</tr>
<tr>
<td>2 Students were knowledgeable at RTL level Verilog</td>
<td>V2</td>
<td>3.9</td>
<td>yes</td>
</tr>
<tr>
<td>3 Students were glad/excited to see schematics of random logic synthesized from RTL –combinational</td>
<td>V4</td>
<td>3.9</td>
<td>yes</td>
</tr>
<tr>
<td>4 Students were happy to be able to use a range of arithmetic operators on vector signals</td>
<td>V5</td>
<td>4.3</td>
<td>yes</td>
</tr>
<tr>
<td>5 Students were glad/excited to see schematics of arithmetic functional blocks synthesized from RTL -combinational</td>
<td>V7</td>
<td>4.2</td>
<td>yes</td>
</tr>
<tr>
<td>6 Students were glad/excited to see schematics of sequential blocks &amp; arithmetic functional blocks synthesized from RTL -sequential</td>
<td>V9</td>
<td>3.7</td>
<td>yes</td>
</tr>
<tr>
<td>7 Student would have <em>benefited</em> more in the course, if student knew <em>more</em> Verilog</td>
<td>Q7</td>
<td>3.5</td>
<td>ok</td>
</tr>
</tbody>
</table>

Table 2: Student Survey Results

Part V: conclusion

With the current complexity levels of functionality in digital circuits and the abundance of inexpensive programmable logic devices with a wide range of circuit characteristics, HDL based design techniques are rapidly replacing the traditional circuits built using discrete integrated circuit parts. Using an HDL, designs can be described at any abstract level, and designers can write their circuit descriptions without choosing either a specific fabrication technology in the case of Application Specific ICs or a particular programmable logic device. CAD tools can easily convert designs to future technologies; functional verification of the design can be done very early in the design cycle; and a significant time reduction in the design cycle can be accomplished by using an effective HDL[^5][^10][^12].

It is evident that it is becoming past due to have introduced HDL training into the undergraduate EET curricula at every education institution. As this survey paper discovered that students like to describe their digital designs in HDL using higher abstraction levels which saves them time and effort for their course assignments. Also, most of the students like to visually see the final form of their text based circuit description displayed on the computer monitor in color showing the major functional blocks. ISE software also does a great job of providing exploration within the schematics although the circuit has not even been implemented on the hardware board. In conclusion, the incorporation of Verilog HDL based design implementation techniques helps students learn the course material, and also makes the EET graduates more marketable. By providing electrical engineering technology graduates skills associated with advanced and efficient design methods, they will be trained to see the need to implement and promote technological changes at their work place throughout their careers as a lifelong practice.

[^5]: \(5\)
[^10]: \(10\)
[^12]: \(12\)
References:


