AC 2009-827: PROJECT G2: CIRCUIT DESIGN IN THE UNDERGRADUATE CLASSROOM

Steven Kurtz, University of Notre Dame

Jay Brockman, University of Notre Dame

Ramzi Bualuan, University of Notre Dame

Project G2:
Circuit Design in the Undergraduate Classroom

Abstract

This paper examines the efforts of undergraduate students to design and construct control circuitry for a Lego® robot. The work focused on following two different design paradigms for implementing circuits. The goal of the study was to determine how well suited the two approaches are for undergraduate VLSI Design class projects. The first implementation focused on logic layout at the mask level to produce an Application Specific Integrated Circuit (ASIC). The second implementation involved Verilog code being mapped to a Field-Programmable Gate Array (FPGA). Both methods provided students with different insights into the design process while exposing them to a variety of CAD tools used in manufacturing today.

This project grew out of Project G (short for Godzilla), an earlier endeavor in which a group of undergraduates constructed a Lego® robot using the Lego Mindstorms™ tool kit. This work looked for alternative ways to control the robot and was performed as a mixture of class projects and faculty directed undergraduate research. The project has resulted in successfully driving the Godzilla robot via an FPGA Board. The ASIC design is nearly complete and a working test chip has been fabricated on MOSIS. In addition, the observations from this work have helped provide insight into different ways VLSI Design can be taught.

In this paper we examine the two design approaches and their results in detail, highlighting the challenges encountered along each path as well as the major benefits. We look at how those two approaches correlate to changing trends in modern computer technology. Finally, we analyze how this work can be translated into the classroom as undergraduate senior design capstone projects.

Introduction

In previous work at the University of Notre Dame, a group of undergraduate students from various engineering backgrounds produced a Lego® version of the monster Godzilla[1]. The robot (Figure 1), standing about 13" tall, drove around wagging its tail and spouting real flames from its mouth, all at the touch of a Lego® remote control. A compilation of tidbits arising from numerous different engineering backgrounds, Godzilla was a diverse robot. With light in his eyes, gear driven legs, fire on his breath, and a computer program for brains, Godzilla brought to life the disciplines of his creators.

In the original robot, all of the programming was done using the Lego Mindstorms™ toolkit[2]. The Lego® tools provide a simplistic programming language Not Quite C (NQC) which is compiled and loaded into a microprocessor contained in the robot's control unit. This control unit, known as an RCX, interacts with external motors and sensors via special Lego® wire contacts built into its sides. In the semester following

Godzilla's creation, a group of undergraduate students in Notre Dame's VLSI Design course took on the challenge of designing a structured ASIC which would replace the RCX-based control system as their senior design project. Thus began Project G2.

Figure 1: Godzilla Robot Breathing Fire

In the semesters that followed, Project G2 morphed from a quest to replace Godzilla's Lego® RCX units, into a broader examination of different design methods suitable for accomplishing that task. This study of design methodologies served as an extension of previous efforts undertaken at the university[3]. Under faculty direction, students explored two such design paths, which were structured to highlight different aspects of VLSI Design while exposing students to many of the various design tools and practices that are used in industry.

Below we share our observations of each methodology in terms of its: learning objectives, design tools, project outcomes, and overall learning experience.

Design Methodology One - ASICs

The first design approach was the VLSI Design class project in which students worked in groups with the goals of designing a Lego® robot and producing an ASIC capable of controlling the robot. Each group of students was expected to design a robot (in this case Godzilla), define the logic required to control the robot, implement that logic in an ASIC using an IC layout tool, and then perform tests to ensure that the design was functional.

The end result would be a chip that could be fabricated on the MOSIS[4] process as well as in the facilities at the University of Notre Dame.

ASIC - Learning Objectives

The ASIC approach to chip design focused heavily on the layout and fabrication process side of chip design. Students studied the properties of transistors as well as the process steps required to fabricate them. During the course of the semester students were introduced to several new computer design tools which they used to layout and test the designs. The tools used for this exercise were from the Mentor Graphics tool suite[5]. Introducing students to tools like IC Station and Mach TA (MTA) allowed students to apply the concepts about chip design that they learned in class while becoming acquainted with CAD software used in industry. In addition, students became familiar with design rules used for real processes as they applied the MOSIS design rules to their circuit. An added benefit of designing everything in sync with the design rules used at MOSIS was that the chips could then really be fabricated allowing students to test their finalized circuits for real.

ASIC - Design Tools

To enable students to work with design at such a low level, many different tools from the Mentor Graphics suite were used. Among the most commonly used tools were IC Station and Mach TA. IC Station was used to specify the areas covered by such materials as metal, poly, active, and the varying diffusions required to create p and n-type transistors. This involved students not only with the various layers of an IC, but also with the design specifications required for fabrication. For example, the minimum wire thicknesses and limitations in the proximity of transistors must abide by certain design rules. Once circuits were assembled in IC Station, Mach TA was used to test the design. MTA allowed students to apply a series of test vectors to the chip inputs, which the tool used to generate output values. By comparing MTA generated outputs to the expected output values, students were able to test and debug their ICs. While students gained a greater appreciation for the numerous design considerations taken into account when creating an ASIC, the learning curve for these tools did result in a limited complexity of chip design.

ASIC – Project Outcomes

The results of this project were quite encouraging. Students broke down Godzilla's NQC code into several self-contained finite state machines which could be implemented via a series of Programmable Logic Arrays (PLAs), Look Up Tables (LUTs), and Delay Flip-Flops (DFFs). Because of the complexity of constructing ASICs at the mask level, each group of four to six students worked to design and to test one of the above three key building blocks. Next, after combining the circuits generated by other groups, each portion of the chip was laid out and tested individually using IC Station and MTA. All of the individual finite state machines implemented worked perfectly when tested on their own.

However, when all placed on the same chip (Figure 2), errors arose in parts of the chip that had been demonstrated to work. As a result the entire chip was never fabricated. In its place, a chip consisting of several of the individual parts of Godzilla was fabricated in the MOSES 0.6µm process. This test chip included a full-adder PLA, one of the counters used for timer interrupts in Godzilla's finite state machines, and an individual inverter. Unfortunately the time required for chip fabrication resulted in this portion of the project spanning a second semester. Upon return from fabrication, students tested the chip and found it to be functioning successfully at clock speeds approaching 10MHz. While a completed ASIC has yet to be constructed to replace the RCX in Godzilla, the project was considered a success as the major blocks of student designs functioned correctly when actually fabricated.

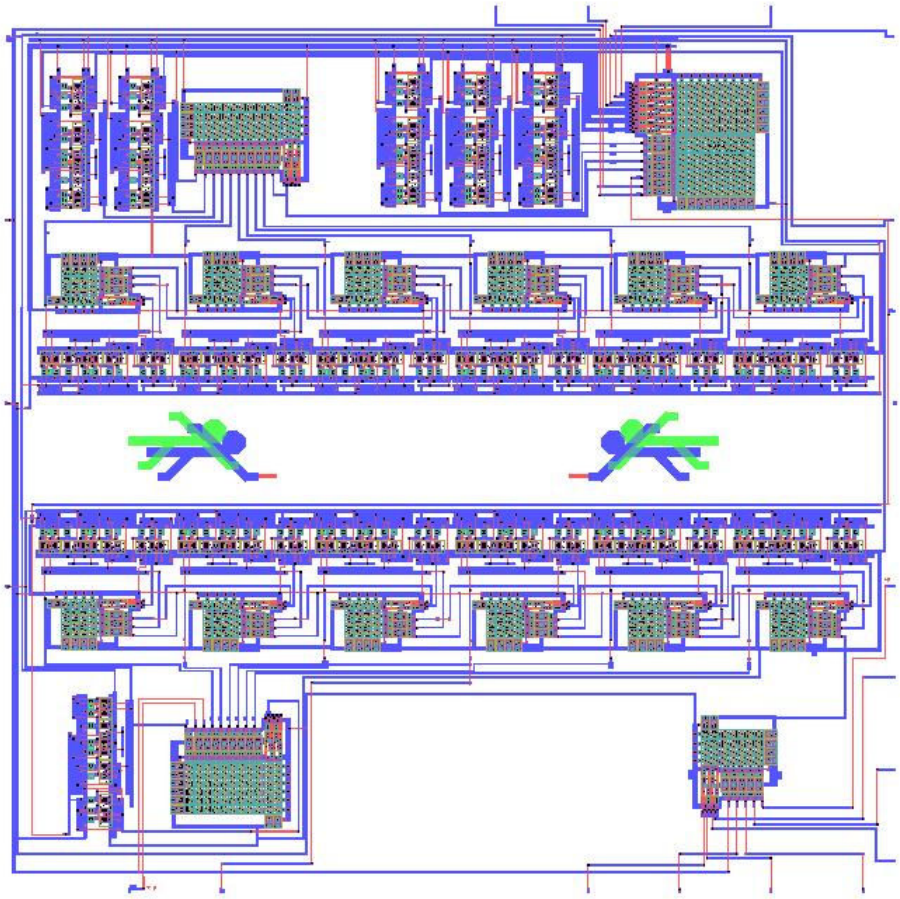

Figure 2: Student designed Godzilla ASIC

ASIC – Overall Learning Experience

The ASIC approach to circuit design highlighted many key ideas in the computer industry. Students obtained first hand experience demonstrating both the advantages and disadvantages of customized designs. By seeing how factors such as transistor sizing, placement, and organization can produce a faster and smaller chip, they gained an appreciation for custom ASICs. While the Godzilla chip was relatively simple and did not

require advanced optimization, the method still provided a valuable learning experience. This design approach also gave students an impression of how complicated custom design is, and why design costs can be so large. Students also received hands-on practice working with tools used in the industry while also being exposed to the intricacies of transistor function and fabrication. Overall, everyone learned a lot, and the result of their efforts yielded a successfully fabricated structured ASIC.

Some of the disadvantages of this design implementation are what led us to our second attempt to control Godzilla. One problem with designing ASICs is the large amount of time that goes into each design. Many hours were spent just to design circuits as simple as an inverter. Also, the high cost of fabricating each ASIC is problematic. For applications with millions of chips an ASIC is a good idea, but for the one and only Godzilla, a faster and cheaper method should be explored. One such methodology can be found in the use of FPGAs.

Design Methodology Two - FPGAs

The second implementation also fit into the context of a VLSI Design project with the goal of a different approach to controlling Godzilla, highlighting different aspects of chip design. However, this project was conducted outside of the classroom as an independent research project. Students used the Verilog hardware design language to formulate the logic required to control the robot, and an FPGA board to actually implement the circuit. Once tested with the robot and an FPGA board, automatic placement and routing synthesis tools were used to generate ASICs similar to those produced in the first project. Once again, students strove to bring Godzilla to life.

FPGA - Learning Objectives

The FPGA oriented approach to controlling Godzilla focused on a much more abstract synthesis of the circuit. Instead of developing the logic by aligning transistors, students used Verilog to describe the system and an FPGA board to perform the actual layout. Design work done with Verilog taught students good methods for planning, structuring and describing circuits. Running code on an FPGA board allowed students to bridge the gap between the software description of their circuit to the actual Lego® robot. It also emphasized the versatility of FPGA boards due to their reprogrammable nature.

Overall the work focused on a top down approach to studying circuits by moving from ASM charts to Verilog code, and then allowing a computer to automatically configure the circuit on the FPGA board. Students were able to follow their project from conception to realization in the course of a semester.

FPGA – Design Tools

The main design tools used in this research were Mentor's HDL Simulator ModelSim and the Windows distribution of the Xilinx[6] software package. ModelSim was used to test the Verilog modules by applying test vectors to the circuit's inputs. By observing

changes in signals, or the lack thereof, testing the Verilog code was easy. The use of Verilog to package everything into small, simple modules also made development flow smoothly. Once the Verilog modules had been tested thoroughly with ModelSim, the Xilinx software enabled students to save the circuit on the FPGA board.

Using the Xilinx tools, students recompiled and tested the Verilog code before placing and routing it on the FPGA hardware. The FPGA board hosted a variety of input and outputs including buttons, switches, LEDs, and pins. This allowed students to vary the form of the I/O signals which facilitated easy testing and debugging. In addition to on-board I/O the FPGA board also interfaced with both an external breadboard and with an H-Bridge. The H-Bridge was required to step voltages up from the FPGA's 5V supply to the 9V range expected by the Lego® motors. The breadboard provided extra space to interface with the Lego® touch sensors.

The last tool used in this approach was Mentor's IC Station which provided students with a quick glimpse at the process used to create transistor layouts. IC Station was also used to AutoPlace and Route the finalized design to produce an ASIC similar to those constructed in the first design.

FPGA – Project Outcomes

The best result of this design strategy was the speed of the design cycle. In the course of one semester, the circuit's logic was defined, coded, and tested resulting in the Godzilla robot running on the FPGA board in place of the RCXs. The original embedded software written in NQC was redesigned as a dedicated state machine, implemented in Verilog and synthesized into hardware. Using the inherent modularity of Verilog, code could be written and tested incrementally, reducing time debugging the chip. In addition, students were able to add features like a clock divider that had not been feasible in the scope of the previous design. The speed with which the program could be modified and reloaded into the FPGA allowed for many changes in code to account for differences between the RCX and the FPGA board. For example, there are slight differences in the operation speed of the motors between when they are being run via RCX and FPGA board. Thus, the timing scheme had to be recalibrated to ensure consistent performance.

Once the Verilog code had been tested with the robot using the FPGA board, Mentor tools were used to automatically place and route the chip. The Verilog code was translated into a schematic which was then mapped to a library of standard cells. The tool generated a layout of these standard cells and wired them all together. The resulting chip (Figure 3) could have been fabricated on the MOSIS process. This whole process was accomplished via a series of scripts, so students did not spend significant time learning how to use the Mentor tools. However, after a simple tutorial in IC station, they were able to at least understand what the shapes and colors in their final circuit corresponded to.

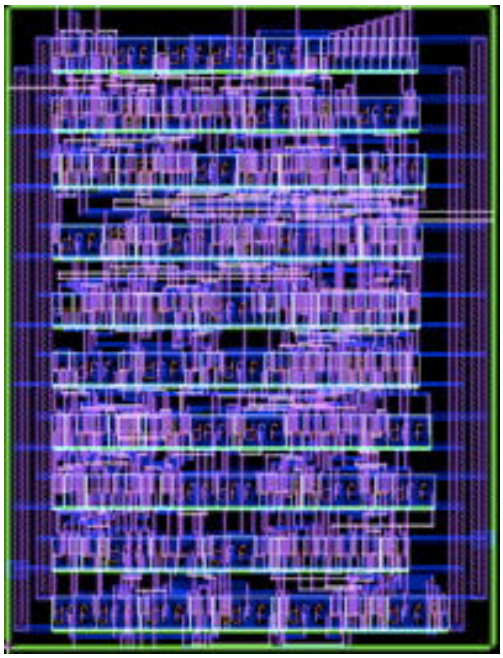

Figure 3. Computer generated Godzilla ASIC

FPGA – Overall Learning Experience

The FPGA approach to building and testing the circuit emphasized different aspects of chip design than creating the original ASIC. While design was faster and easier, there were many low-level aspects of chip design that were brushed over. For example, when describing the circuit, logic was constructed using key words like "xor" and "nand" rather than physically connecting transistors in a pull-up and pull-down network. Furthermore, everything in the FPGA was routed automatically. Students did not experience first hand the intricacies of laying out a circuit. However, after producing a simple layout of an inverter in IC Station, students were able to appreciate some of the complexities involved in custom design.

This method was generally a great success. The design flow was well suited to follow a project from the concept phase to a physically realized circuit in a single semester. A lot of time was saved by using standard cell libraries, AutoPlace and Route, and an FPGA board to execute the circuit. A simple module can be written, debugged, and tested on a breadboard in an hour allowing student designs to take on greater levels of difficulty. Students learned how design languages like Verilog and routing tools like Xilinx can significantly reduce engineering design times and costs. Finally, through this process students were exposed to a wide variety of design tools and practices that are used in the industry.

ASIC vs. FPGA

The two design methodologies presented in this paper approach circuit design from opposite directions. In the ASIC example, students start by learning how transistors are

implemented individually. From there, they learn how to leverage different transistor properties when implementing logic functions. Assembling these logic functions, students implement their designs in a hands-on layout process.

The FPGA approach begins with a very abstract, high-level logical description of the circuit. This description is then expressed as Verilog code and CAD software translates it down to the gate level on an FPGA board. When generating an ASIC using this approach the CAD software once again hides the translation to the gate/transistor level. Most of a student's time is spent well above the transistor level disregarding the intricacies inherent to the transistor level of circuits.

In the classroom, circuit design using FPGAs is appealing due to its flexibility and low design cost. The design process is significantly faster and allows for later updates or modification. For small scale endeavors, purchasing and programming FPGA boards is significantly less expensive than purchasing mask sets required for ASICs. As mask costs continue to grow, reprogrammable logic becomes increasingly appealing in the commercial market. As students work or their projects, they experience the benefits of FPGA boards allowing them to appreciate why FPGAs have become the industry standard for many time-critical applications.

On the other hand, designing ASICs offers many perks as well. With an ASIC the engineer can customize the circuit to a degree that is impossible to reach using generic reprogrammable logic found in FPGAs. When performance really matters, whether it is speed, power, or size, a well designed ASIC promises significant advantages. In addition, if large numbers of the chips are to be made, the design costs of ASICs become less significant.

When considering Project G2, or other comparable VLSI Design projects, the FPGA approach appears to be a better fit. The process is better suited to fitting within a single semester time frame as students can expect to get concrete results. Financially, purchasing several FPGA boards is much less expensive than having an ASIC fabricated. The FPGA design is also more forgiving to student mistakes as it is easily reprogrammed. With an ASIC, determining timing schemes for a robot would become very difficult and any mistakes would require external circuitry to fix them. While the FPGA approach is less focused on circuit-level design and analysis, simple tutorials can be introduced to the class to present these types of design considerations to students. In the end, in our ASIC vs. FPGA comparison, it appears that the latter methodology used to reprogram Godzilla was the better one.

Classroom Application

In VLSI Design at Notre Dame, we aim to impart each student with an appreciation for the complexity of designing circuits and the many tradeoffs that must be made along the way to a complete chip. When outlining senior design projects, we feel it is important that the project highlights and reinforces the concepts that students have learned in class

while also engaging the students' interest. Through our case study, we learned numerous lessons at both the teacher and student levels.

Ideally, both methods presented here would be incorporated into a design project providing students with a great understanding of the advantages to each approach. However, there is not enough time to do both, so professors must simply be aware of the tradeoffs required when selecting a particular design methodology. For example, when teaching a class designed to focus on circuit level optimization, the ASIC approach is the ideal method. Our experience has shown that this method gives students a great appreciation for the intricacies of custom design. However the design process is complex and requires professors to get the projects moving very early in the semester. In addition, it is unreasonable to expect a fabricated chip in one semester so stopping with a completed mask set may be the only option. For schools whose senior design courses span two semesters, students can spend the first semester designing the chip, and the second semester analyzing the finished product and interfacing it with the structure it was designed to control.

When teaching a higher-level, single-semester course, the FPGA design flow integrates itself nicely with general VLSI Design concepts. Students begin by learning to identify, describe, and define logic in a well-organized fashion using Verilog. As the class's focus shifts from high-level logic toward the intricacies of the transistor and low-level design, students convert their circuits to FPGA boards and auto-place and routed ASICs. By mixing in simple transistor layout tutorials using tools like IC station, students also gain an appreciation for the mask layers used in chip fabrication. Throughout the project, in addition to reinforcing ideas taught in the classroom, students learn to work within their groups to divide up the workload.

From a student perspective, it is critical that the project is interesting and enjoyable. A successful method for accomplishing this is to allow students to pursue creative designs with concrete goals. The end product should be something the student can visualize and appreciate on its own. In our experience using a Lego® robot such as Godzilla for the design target worked very well. Lego® robots enable students to cater their projects to their own particular interests as they apply their advanced education to childhood toys. The Lego Mindstorms™ toolkits provide students a wide range of design opportunities. The parts are easy to build, modify, and reuse thus reducing the time required to build the robot. In addition, the RCX unit provides an easy way to test the robot's functionality prior to completing the circuit on the FPGA board. In the end, we feel that Project G2 is representative of a very good design target which can be used as a model for other senior design projects.

Conclusion

Project G2 has resulted in success on several levels. The success of the ASIC test chip fabricated on MOSIS demonstrates the feasibility of the first design process. Better yet, the robot is currently operated via an FPGA board running Verilog code written by undergraduate students at Notre Dame. Throughout the process of reworking the control

structure for Godzilla, undergraduate students have been exposed to numerous different CAD tools, design strategies, and circuit properties. Based on all of our experiences, we have learned that while both approaches offer different benefits, the FPGA approach is more conducive to a capstone project for a VLSI Design class.

Acknowledgments

Special thanks goes to visiting professor Dr. Michael Ciletti for his contributions to our understanding and application of Verilog to this project.

Bibliography

1. Bualuan, Ramzi et al. "AC 2007-2671: Project G: Multidisciplinary Teamwork at its Best," American Society for Engineering Education, Honolulu, Hawaii, 2007.
2. http://mindstorms.lego.com/
3. Bernstein, Gary H. et al. "From Bits to Chips: A Multidisciplinary Curriculum for Microelectronics System Design Education," IEEE International Conference on Microelectronic Systems Education, 2003.
4. http://www.mosis.com/
5. http://www.mentor.com/
6. http://www.xilinx.com/