Teaching a Laboratory Intensive Class in a Distance Education Mode

J. Moore*, M.A. Thornton*, and R.W. Skeith**
Southern Methodist University*, University of Arkansas**

Abstract

A unique set of obstacles occur when administering a distance education class that has a significant laboratory component. A class in Digital Systems Design targeting fourth-year and introductory graduate students that is offered both in conventional and distance education modes is described. Because the course is design oriented, students must engage in several design projects that would ordinarily be carried out in a laboratory to satisfactorily achieve the course learning outcomes. The Digital Systems Design course has been offered in dual conventional and distance modes for several years and methods for dealing with laboratory assignments and design projects have been developed and are described here.

Introduction

Distance education is becoming more common and is very popular for students with fulltime professional careers seeking to obtain an advanced degree. Most universities that offer distance education courses usually restrict their offerings to classes that are largely theoretical and have the traditional homework, report, and examination assignments. In engineering fields, this restriction imposes severe limitations to overall degree programs since class work that involves a "hands-on" component is not included. Furthermore, there is some concern that the quality of the education received by traditional on-campus students with access to laboratories may be better than that received by distance education students. This has become an issue for both Universities in the US and the UK and in fields as diverse as Language and Engineering. Some of these universities have begun using simulator packages to try to create the laboratory experience for distance stories [1,2]. Some of the tools used by these universities are LabView, Virtual Network Computing (VNC), or providing the needed software to perform the labs [3].

Due to the proliferation and common use of personal computers, it can be safely assumed that all distance students have access to some type of computing facility. In fact, the current trend among distance education is to offer recorded lecture content as streaming video media from a server connected to the Internet or by providing DVDs. In circuit design courses, it is common to use Electronic Design Automation (EDA) tools to specify, analyze, and simulate circuit designs. Unfortunately, many EDA tools are very expensive, and although they are usually donated to universities, require a license checkout procedure and are required to run on university computers.

Recently, some EDA companies and Field Programmable Gate Array (FPGA) vendors have been offering free downloads of their tools for student usage that typically have some restrictions incorporated as compared to their professional versions. This has made it possible to have a design intensive course offering in the distance education mode. The only requirement is that the

distance student must have access to a computer and be capable of corresponding with class instructors via email.

The remainder of this paper will describe the Digital systems Design course offered within this framework and will describe some of the lessons learned and obstacles associated with this approach. In the next section, we will provide an overview of the course including the learning objectives and major topics. Next, we will describe the laboratory exercises that are performed by both the traditional on-campus and the distance education students. Finally, we provide a summary of this experience.

Course Description

The catalog description of the Digital Systems Design course is as follows:

Modern topics in digital systems design including the use of HDLs for circuit specification and automated synthesis tools for realization. Programmable logic devices are used throughout the course. This course has heavy laboratory assignment content and a design project.

The main objective of the course is to provide electrical and computer engineering students with enough background and experience to allow them to gain entry-level employment as a digital design engineer. The only prerequisite for the course is successful completion of an entry-level digital logic course. Such entry level courses are very common and exist in some form in almost all electrical and computer engineering curricula but they often provide only limited exposure to modern EDA tools and their usage in circuit design due to the amount of time that is used to introduce concepts such as Boolean algebra, logic minimization, and finite state machine design.

The learning objectives of this second course in digital systems design are as follows:

- Digital Logic Design Review
- HDL (Discrete Event) Simulators
- Verilog Language
- Combinational Logic Synthesis using Verilog
- Programmable Logic Architecture and LPMs
- Timing Constraints and Timing Models in Programmable Logic
- Pipelining for Increased Throughput
- Sequential Logic Synthesis using Verilog
- FSM State Assignment
- High Level Synthesis

A key element of the course is the ability to use a Hardware Description Language (HDL) to describe a synthesizable circuit. In the example discussed here, the Verilog HDL is used although prior offerings of the course utilized VHDL. Furthermore, most introductory digital logic courses focus on the ability to design circuits that are functionally correct with perhaps some area optimization through gate reduction techniques. In this second course, more emphasis is placed on timing and delay thus introducing the student to the use of timing analyzers and design optimization techniques for increasing clock frequency such as pipelining. Prior to taking

this course, most students that have exposure to pipelining have received it from a study of RISC processor instruction pipelines in a computer architecture course and do not realize that this is in fact a general digital circuit delay optimization method.

Although the design experiments and course project focus on programmable logic, many of the techniques are equally applicable to designs targeting other technology such as standard cell or full custom. The importance is the use of HDLs in conjunction with automated EDA synthesis, simulation, and timing analysis tools that are commonly used regardless of the ultimate implementation technology. FPGAs and their associated EDA tools are particularly convenient for this course since they require a minimal amount of maintenance, are generally available for running in Microsoft OS environments, and usually are packaged as a self-contained suite of tools comprised of a synthesizer, place and route, simulation (both functional and timing), timing analysis, and others. Additionally, these tools allow for design entry to be performed in both schematic capture and HDL specification modes. This is particularly convenient for students who are just beginning to learn HDLs since they can immediately begin using the tools in a schematic capture mode.

In the first week of the class, all students both traditional and distance are required to go through a tutorial on the usage of the tools. In order to complete this relatively easy but critical assignment, distance students must download and install the student version of the EDA tools on their own computer. To make it convenient for distance students, a link to the tool download webpage is provided directly on the class webpage. In fact, many of the traditional students also download and install the student version of the tools on their own computers as well as this offers them the capability to work on the design assignments at home.

Laboratory Exercises and Design Project

The course assignments consist of ten "laboratory design experiments" and a class design project. Each of the design experiments is described on a separate webpage all linked to the main class webpage. Some of the design experiments later in the course require results of earlier design experiments to be completed. In the modern era of IP reuse, this is an important aspect of the course. A summary of the design experiments is provided in Table 1.

Table 1: Summary of Design Experiments

Experiment Number	Title	Concept	Description
1	Tutorial	Tool Installation and Use	Vendor Supplied Tutorial
2	Introduction to Schematic Capture	Design Entry and Simulation	A 1-bit register is designed and used as a building block for a 4-bit register
3	2×8 Multiplier using Verilog	Describe Combinational Logic in HDL	A schematic of a 2×8 Multiplier is given with simulation input/output and students must implement this in Verilog
4	Introduction to LPMs	Use of Existing Cores	Students are given the basic Newton-Raphson iteration equation and asked to implement it using existing cores (adders, multipliers) in schematic capture.
5	Using LPMs in HDL	Use of Existing Cores in HDL	Students re-implement the 2×8 Multiplier but this time they replace their custom Verilog with instantiations of existing cores
6	Timing Analysis	Use of timing analyzer and introduction to pipelining	Students use the 2×8 Multiplier schematic that was given in experiment 3 and perform various timing analyses. They then insert DFFs into the circuit and analyze the increased predicted clock frequency.
7	Bit-level Pipeling	More experience with pipelining	Students are given a fairly aggressive required clock speed and asked to further pipeline the 2×8 Multiplier to reach the specification. They are also asked to use the 2×8 Multiplier as a basic block and build an 8×8 pipelined Multiplier
8	Sequential Logic in HDL	Describe a synchronous sequential circuit in HDL	Students must implement a register that has serial and parallel I/O with an ability to have selectable up/down count capability as well as left/right shift capability.

Experiment Number	Title	Concept	Description
9	Datapath/Controller Design	Design a circuit consisting of a datapath with a synchronous controller	Students reuse the Newton-Raphson iteration datapath design in experiment 4 with a synchronous controller and memory unit that computes the multiplicative inverse of an integer.
10	Integer Division	Design moderately complex circuit with memory unit, synchronous controller and arithmetic datapath	Reuse the numerical inverse circuit in experiment 8 with the pipelined multiplier in experiment 7 to create an integer division circuit.

Each of these design experiments require students to implement their designs and then to simulate them and analyze them with the EDA tools and report their results back in a written report. Traditional students demonstrate their designs in the on-campus laboratory at a designated laboratory section time to a Teaching Assistant (TA). Distance students are required to submit their design files and reports to the TA in an email message.

Additionally, every student is required to complete an end of semester design project. This project is a moderately complex arithmetic circuit that has fairly demanding area and timing constraints. It is convenient to impose area constraints when FPGAs are the target technology by simply requiring that all designs target a specific device. Since different devices have varying amounts of on-chip resources available, a device can be specified that ensures a specific area limitation. Also, the final project requires internal pipelining and overlapped computation to occur in order to meet the timing requirements.

A recently assigned final project required students to design a circuit that had a single 8-bit input bus and an 8-bit output bus. Input was assumed to be twelve 8-bit values comprising a matrix, A, and the computed output are the nine values comprising a matrix B where $B=A^TA$. While this seems fairly simple computationally, students were given performance specifications that included an initiation rate of 15 clock cycles and a latency of 23 clock cycles. This means that every 15 clock cycles, a new set of matrix A values is present and that after 23 clock cycles from the beginning of the input matrix A values, the output values must be produced. Clearly, there is not enough time to load all twelve input values, then perform the computation, followed by outputting the resultant nine values; thus students must use overlapped computation. Furthermore a minimal clock speed of 65 MHz was specified for the Altera Cyclone EP1C3T100C6 device. Due to the resource constraints of this device, students could only use a maximum of three internal multipliers and had to carefully allocate and schedule available resources.

Issues Encountered for Distance Students

Several issues were encountered with respect to distance students and their performance of the experiments. One of the recurring issues is that different versions of student versions and professional versions of the tools support different sets of devices and even more importantly produce slightly different timing and simulation analysis results. This can be especially discomforting when a distance student has just barely met timing requirements as specified by their version of the tool only to find out that when the TA tried to reproduce their results, the timing analyzer reported a slower speed.

Because we are required to use the professional versions of the tools in the university laboratories and distance students always use the free student versions, such incompatibilities always exist. Furthermore, the professional versions of the tools support the latest devices while student versions generally support devices that are one or more generations older. The best way to deal with this issue is to maintain an installation of each version of the tool being used. It is fair to insist that distance students use the latest version of the free tool with the result being that the TA only needs use a dedicated computer that contains the free tool.

Another issue is that the EDA tools produce a large number of intermediate design files during processing. Distance students often attempt to send all of these files to the TA in an email message. In the later designs and the project this can be tens of megabytes of data resulting in email accounts overflowing. Distance students must clearly understand that they need to only send the initial design specification files. This causes some extra work for the TA since every distance student's design files must be re-synthesized, re-analyzed, and re-simulated. The amount of extra work is sufficient enough that TAs assigned to this course are given credit for teaching two sections, one for the on-campus students and one for the distance students.

Often distance students will want to send analysis files to support their claims that they have met the specifications. It can be the case, whether intentionally or through accident, that the TA cannot recreate the results claimed by the distance students. It is very important that distance students understand that all the results they claim in their reports will necessarily be recreated by the TA before they are accepted. This also underscores the need for maintaining an installation of the same versions of the tools that the distance students use.

Conclusions and Summary

Although some extra work is required in administering a laboratory intensive course via the distance education mode, it is possible to do so when student versions of EDA tools are available. Our experience has shown that careful recreation of student claimed results and requiring detailed design reports to be submitted can allow for a successful course. The amount of interaction required with distance students and remote submission of designs is increased and can require significant time on the part of the course instructor or TA if a large number of distance students are enrolled.

References

[1] Forger, G., Reynaert, J. and Penfield, S., "An Online Language Lab: Reaching and Teaching at a Distance. In G. Richards (Ed.), Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education", Chesapeake, VA: AACE 2003, pp. 472-473.

[2] Blanchard, R.E., Moron-Garcia, S.D., and Bates, M.R., "Converting the physical to the virtual: providing a laboratory experience for distance learners in engineering", Current Developments in Technology-Assisted Education, 2006, pp. 1208-1213.

[3] Elliott, Stephen J. and Kukula, Eric P., "The Challenges Associated with Laboratory-Based Distance Education", EDUCAUSE Quarterly Vol. 30, November 2007, pp. 37-42.

Biographical Information

Jason Moore
Jason Moore is currently a PhD candidate in computer engineering at Southern Methodist University. His research interests are in high-performance digital arithmetic circuits and he has extensive experience as a TA and instructor in the course described in this paper and others. Prior to coming to SMU, Jason also worked as a TA at the University of Arkansas where he received the B.S. degree in computer engineering.

Mitchell A. Thornton
Mitch Thornton is a professor in the Department of Computer Science and Engineering and the Department of Electrical Engineering at Southern Methodist University. His research includes EDA algorithm design, digital systems design, and computer arithmetic. Mitch is a licensed professional engineer.

Ronald W. Skeith
Ron Skeith is a professor in the Department of Computer Science and Computer Engineering at the University of Arkansas in Fayetteville. His research includes software engineering and computer engineering. Ron is a licensed professional engineer.