EXPERIENCES TEACHING SYNTHESIS OF FPGAS AND TESTABLE ASICS

Donald W. Bouldin
Electrical Engineering
University of Tennessee
Knoxville, TN 37996-2100
dbouldin@utk.edu

ABSTRACT

Microelectronic system designers are increasingly capturing their designs using hardware description languages such as VHDL and Verilog. The designs are then most often synthesized into programmable logic devices such as field-programmable gate arrays (FPGAs). This approach places the emphasis on high-level design which reduces time to market by relying on synthesis software and programmable logic to produce working prototypes rapidly. These prototypes may then be altered as requirements change or converted into high-volume mask gate arrays or other application-specific integrated circuits (ASICs) when the demand is known to be sufficient. These ASICs, however, must be designed to be testable to screen out those with manufacturing defects. Hence, scan logic must be inserted, test vectors generated and fault grading performed to ensure a high level of testability. Experiences encountered from teaching a two-semester graduate sequence on these topics will be summarized.

1. INTRODUCTION

Microelectronic system courses generally emphasize both fundamental issues and technology-dependent skills. The fundamentals are basic concepts which are likely to be applicable for several years after graduation, whereas the technology-dependent skills may last only a few years. Therefore, it is prudent to teach state-of-the-art skills that enable graduates to contribute soon after employment begins.

Since microelectronic system designers in industry are increasingly using synthesis to produce systems based on FPGAs and/or ASICs, a two-course sequence has been developed and taught at the University of Tennessee to provide students with exposure to this style of design. This paper summarizes the objectives of this sequence, the prerequisities, an outline of the lecture topics, descriptions of the laboratory assignments and software/hardware support. The extensive use of email and the world-wide web is described along with experiences learned from teaching this sequence over a ten-year period.

2. COURSE OBJECTIVES

The philosophy guiding this two-semester sequence corresponds to the vision espoused by those attending the NSF-sponsored 1993 Workshop on Rapid Prototyping of Microelectronic Systems for Universities: "Educate students who can use the paradigm of design, simulate, design-for-test, build and test (as opposed to just design, build and test) to create microelectronic systems, not just integrated circuits, of sufficient quality that the global competitiveness of U.S. industry will be continued and enhanced." [1].

Hence, the objectives of these courses [2] are:

- To present fundamental techniques of microelectronic systems design including design synthesis, simulation, testing, prototyping and measurement.
- To reinforce the lectures and discussions with laboratory experience using state-of-the-art computer-aided design tools.
- To develop human communication skills via a team project requiring both written and oral reports.
- To illustrate capturing a design in a technology-independent means using a mix of levels (behavior and structure) and then to map the synthesized result into several technologies which can be compared.
- To provide in-depth projects using FPGAs and testable ASICs that involve architectural tradeoffs, simulation and prototyping.

3. COURSE CONTENT

This sequence is intended for senior undergraduate or beginning graduate students in electrical engineering who have completed introductory digital logic design and microprocessor interfacing courses. Each student is expected to spend about ten hours per week on this sequence with 2.5 hours of lectures per week and the remaining time consisting of laboratory and project assignments performed in an open lab.

Lecture topics include:

- Design Methodology
- Design Capture
- Role of Synthesis
- Hardware Description Languages
- Partitioning System Functions and I/O
- VHDL for Combinational Logic
- VHDL for Controllers
- Role of Simulation
- Technology Choices
- Technology-Independent Design
- FPGA Floorplans and Interconnect
- Unique Attributes of FPGAs
- Choosing an FPGA
- Physical Placement and Routing

0-8186-7996-4/97 $10.00 © 1997 IEEE

- Post-Layout Timing Simulation
- ASIC Testing
- Fault Grading
- Technology Comparisons
- Rapid Prototyping
- Interaction with CAD Suppliers
- Interaction with ASIC Foundries
- Technical Report Writing
- Group Presentations

The laboratory and project assignments for the first semester include:

- LOGIN, UNIX, File Editing and Email
- Schematic Entry and Simulation (Viewlogic)
- Capturing Structure Using VHDL; Downloading
- Capturing Behavior Using VHDL
- Capturing Behavior Using ABEL
- FPGA Placement and Routing (Xilinx, Actel and Altera)
- Post-layout Simulation
- Retargeting
- Project Proposal, Presentation, Demo (Xilinx) and Report
- Final Exam

The laboratory and project assignments for the second semester include:

- ASIC Design Validation
- ASIC Design Synthesis and Simulation
- Making an ASIC Design Testable (Synopsys)
- Physical Placement and Routing (Epoch)
- Post-Layout Timing Simulation
- Reconfigurable Computing (EVC Demo)
- Retargeting
- Project Proposal, Presentation and Report
- Final Exam

4. LABORATORY SUPPORT

In keeping with the goal of providing students with exposure and experience with state-of-the-art CAD tools, we have joined the university programs of several CAD vendors. Thus, the software used in this course sequence includes synthesis tools from Viewlogic and Synopsys and physical placement and routing tools from Xilinx, Actel, Altera and Cascade Design Automation (Epoch).

For prototyping of FPGAs, we use demonstration boards supplied by Xilinx and the Engineer's Virtual Computer (EVC) supplied by the Virtual Computer Corporation. All of these units contain reconfigurable logic so the equipment is available for reuse each year. We have 12 Xilinx boards to support 12 two-person projects. We have only one EVC but it is easily shared by all of the students in the class since it is accessible from any workstation.

For ASIC prototyping, we originally submitted designs to MOSIS but have stopped doing this since we almost always found no design errors. The use of NSF funds for this purpose seemed superfluous. We do, however, continue to submit to MOSIS manual layout designs from another course not described here.

We use a cluster of six UNIX workstations (Sparcs) for this sequence. This provides a ratio of 4 persons per seat, which has proved to be very adequate. In fact, the machines are shared by another course which has 8-16 students performing custom layouts. Each workstation has 64 MBytes of internal RAM and a 19-inch color monitor. Since several of the CAD tool packages require 1-3 GBytes of disk storage for the executables and we have several toolsets, we have allocated 12 GBytes for tools and 2 GBytes total for all (not each) user files. For microelectronic systems research projects, we have additional machines and storage capacity so that the machines for this course sequence are not overloaded.

5. EXPERIENCES

Except for the initial handout in the first course, all syllabi, assignments and tutorials are disseminated via the the world-wide web [2]. Students generally utilize a split-screen approach with one window containing tutorial instructions and a second window exercising the appropriate CAD tool needed for the design. Email is used for lecture followup questions and answers and to report completion of assignments. These practices have facilitated communication and enhanced learning.

During the first semester, Viewlogic is used as the initial schematic capture and VHDL synthesis and simulation toolset because of its simplicity. During the second semester, Synopsys is used because of its ability to insert scan logic, automatically generate manufacturing test vectors and to perform fault grading for ASICs. Projects during the first semester are performed by two-person teams. During the second semester, some moderately complex designs are performed on an individual basis with large designs being conducted by four-person teams.

During the first semester, only the basics of the VHDL language are introduced. These include structural techniques which use "components" much like schematics and behaviorial constructs such as "case" and "if-then-else" statements. Thus, only a few of the language concepts need to be mastered but the students are being prepared for the second semester in which more advanced features are presented.

A variety of projects have been undertaken. Generally, each contains a mixture of computation and control. Emphasis is placed on **finishing** all of the project design steps (synthesis, simulation, testing, prototyping and measurement) and not on the complexity of the project itself. Students are encouraged to complete every step for a base level of functionality before adding additional features.

In summary, this sequence has met its goals of providing exposure to state-of-the-art tools and design techniques. However, it is a continuing struggle to achieve this high level of quality.

REFERENCES

[1] Bouldin, D. (Editor), "Report of the 1993 Workshop on Rapid Prototyping of Microelectronic Systems for Universities", University of Tennessee Report, April 1994, http://microsys6.engr.utk.edu/ece/nsf_ic93.html.

[2] Bouldin, D., "Designing Microelectronic Systems", http://microsys6.engr.utk.edu/ece/bouldin_courses.