An Innovative Rapid Processor Platform Design for Early Engineering Education

Yong-Kyu Jung

Texas A&M University-College Station

Abstract

A new course for the early stage of digital design education is demonstrated. An innovative usage of technologies and an effective organization of numerous subjects to leverage the current classroom practices are presented. In particular, the new course objectives and organization are explained to provide an overall view and details of our rapid design process, as well as to achieve engineering educational goals for reducing a gap between the technologies used in industry and in academia. Expected impacts on realistic, pre-industrial experiences at the early stage of engineering education are also discussed.

Introduction

To teach embedded processor system design, students generally take a series of digital design courses. For instance, courses on Digital Fundamentals and Logic Circuits [1] followed by Computer Architectures [2] are typically taught in the classroom. To accelerate the practical learning process in the classroom, rapid digital system design [3] using field-programmable gate arrays (FPGAs) is often added to the existing curricula. Using FPGA-based prototyping requires learning at least one hardware description language, such as VHDL or Verilog HDL. However, we are continuously challenged not only to reduce the time for teaching the majority of these topics to a single semester, but also to adapt a rapid digital system design process for smoother migration toward the next level of the course. To do so, we have started to develop a customized processor platform for our classroom instruction, instead of using commercial platforms [4], such as FPGA-based platforms along with configurable processor cores [5]. Since reusing existing designs is one of the keys for the rapid design process, we have taught both VHDL and Verilog HDL as primary implementation languages. In this paper, we will discuss the development of this new course including course objectives, organization and evaluation.

Pros and Cons of Current Digital Design Practices in the Classroom

What is the difference between the classroom and industry?

This question has been asked many times. In this paper we consider only the differences in the goals, the quality and the scale of work between industry and academia. Because delivering the most marketable products in the shortest amount of time is critical in industrial projects, working engineers are usually more interested in learning application-specific knowledge and skills as quickly as possible. On the other hand, engineering education usually deals with more general

subjects that are necessary from an educational point of view. Many of those topics, however, may not often, or ever, be used for the post-academic engineering career. To leverage classroom practices, rapid design approaches using FPGA-based synthesizable register-transfer level (RTL) implementations must be added to the current curricula.

Since classroom projects are generally of smaller scale and lower quality than industrial projects, functional simulation followed by FPGA-based verification may be enough. However, industrial projects usually require accurate functionality and timing as well. Because of that requirement, both functional and timing simulations with logic synthesis and optimization are necessary. For realistic, real-world experiences, an efficient rapid design method must be developed to enhance quality and scale of classroom projects.

What do we usually do in the classroom?
We usually offer a series of courses such as Digital Logic Circuits and Computer Architectures coupled with laboratory exercises that include HDL-based implementations. Since technologies and tools are continuously evolving, digital design education must catch up with the pace of the technology evolution in industry. Furthermore, engineering education should predict and lead the direction of the future technology paradigm shift. We should offer a few intensive courses rather than do a series of many courses taught in the current curricula!

Is a commercial FPGA-configurable processor platform appropriate for classroom projects?
A typical or even an advanced hardware design flow starts with the conversion of a paper design into HDLs or schematic design entries. This is followed by logic synthesis including placing and routing for programming the target FPGA. According to the current design flow, FPGA-based design has played an important role in rapid hardware design. Even though performance of FPGA-based design tools has been improved dramatically, most students are inexperienced and will only use a few limited functions or features of the tools. To educate lower or junior level undergraduate students using this type of environment, we need to develop a new customized platform to deliver viable solutions to the engineering education society.

Objectives of the New Course

Our primary objectives in developing the new course are summarized as follows:
Reduction of a gap between the technologies used in industry and in academia:
"What to do" is often more important than "how to do" in academia, so that behavioral implementation may be sufficient. How to implement a design, however, is often much more critical in industry, so RTL implementation and synthesis using the commercial electronics design automations (EDAs) must be performed. Since a soft hardware core can be quickly turned into a physical chip using EDAs, the soft core becomes one of the most preferable hardware types. We expect a soft RTL synthesizable processor core as the final output of our classroom projects. Eventually, this project helps us reconstruct and improve our current effort to meet the future industry's demands.

Efficient assimilation and migration of technologies:
We use the technologies usually available in the classroom as preliminary requisites for the new course. For instance, the use of FPGA tools is a basic. In addition, we concentrate on

customizing a processor platform [6] to classroom projects for efficient migration toward the future digital design education. To develop and properly use an educational processor design platform, we have not only taught VHDL and Verilog HDL to develop components, but also have introduced composition rules for the components.

Pre-industrial experiences at the early stage of engineering education:
To meet this objective, we use various different input formats for classroom projects. In addition, RTL design, implementation and verification have been fulfilled. Laboratory works in particular are designed along the lines of industrial internship so that students may more confidently take their future assignments in industry.

Exploration of digital fundamentals and advanced systems within a single semester:
The new course uses a hierarchical structure in order to deal with a wide range of subjects within a limited time period. The detail design hierarchy consists of a digital component design, a datapath and control design, a pipeline design, and a processor platform design. Students are eventually able to complete a pipelined RISC processor system through the design hierarchy within a semester.

Organization and Maintenance of the New Course

The new course consists of three sections: digital fundamentals; computer architectures and HDLs; and rapid development of an embedded processor system platform. The subjects and the detail contents taught in each section are summarized in Table 1. Many different design input formats have been prepared for various real-world implementations in the classroom.

Sections	Subjects	Contents
Digital Fundamentals	Logic Elements	Binary logic and gate/function Simplification
	Components Design	logic minimization/FSM
	Systems Design	Combinational logic/ Sequential circuit
Comp Arch. /HDLs	HDLs	VHDL/Verilog HDL
	Components Design	Digital design methods; Sim./Syn.; Memory & interface in HDLs
	Computer Architecture	CISC/RISC/Superscalar/VLIW arch.; Pipelining and hazards
Rapid Design of Processor Platform	ISA Design	ISAs; Instruction set design
	RTL Design in HDLs	Func./timing sim. & synth.; FSM design and synthesis in HDLs
	JJ249-04 RISC Core Design	Overview/Design components/ pipeline stages of JJ249-04
		Integration of a pipeline; Design a RISC system
	JJ249-04 RISC Core Verification	Core verification; testbenches & verification environment
		Sim.-based ver.; FPGA-based ver.; RISC core design analysis
Summary		Rapid prototyping of soft RTL RISC core

Table 1: Organization of the new course: Subjects and Contents

Section 1: Digital Fundamentals
Subjects on digital design of logic elements, components, and systems have been taught as digital fundamentals. In particular, the subjects taught are focused on improving students' design analysis and optimization capability for the small designs.

Section 2: Computer Architectures and Hardware Description Languages

Selective contemporary computer architectures have been overviewed and essential techniques of those architectures, especially for instruction set architecture (ISA) and pipelining, have been taught. In detail, ISA design including instruction set and micro-architecture have also been covered. In addition, both VHDL and Verilog HDL have been used as primary development languages for components, functional logic, datapath and control. Verilog HDL has been particularly used for developing the entire processor platform. VHDL, however, has been taught not only as an alternative, but also as a powerful reference to understand and reuse numerous other hardware modeling efforts.

Section 3: Rapid Design of an Embedded Processor System Platform

A 16-bit pipelined embedded RISC processor (JJ249-04) core has been developed. First, the JJ249-04 ISA and its micro-architecture have been designed. As illustrated in Figure 1, four pipeline stages, which can be dynamically configured as a 3- or a 4-stage pipeline depending on the instructions decoded, have been defined. Different shapes of the stage buffers have been used for delivering information to the next stages. We anticipate the reuse and modification of this pipeline scheme for an out-of-order issue/execution and a superscalar design in upcoming semesters. For the parallel memory operations, Harvard architecture with un-unified memory system has been interfaced by two different address generators for instruction fetch and operand load/store.

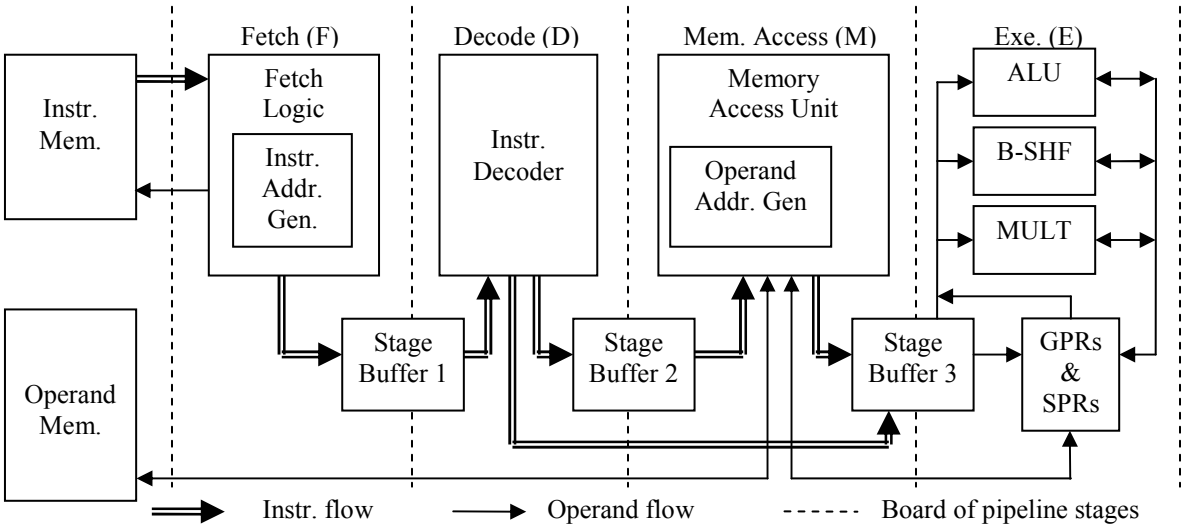

Figure 1: A block diagram of the pipelined embedded RISC processor (JJ249-04) core

For maintenance of laboratory exercises to follow the lecture more closely, some subjects taught in the classroom were used as individual or team projects. Most of the components and the small blocks, such as memories and functional units, have been assigned as personal projects. The large blocks have been assigned to at least two small teams. Each team has implemented the same task. Therefore, these two implementations have been used as cross references during verification. Since for novice designers such complicated integration requires extensive collaboration, early collaboration and feedback are key aspects in successful implementation of the entire design. Finally, two groups have been able to complete their embedded RISC processor core assigned as a final project.

Evaluation and Future Upgrade of the New Course

Based on the subjects listed in Table 1, an evaluation method has been developed, allowing us to measure the outcomes as well as to monitor the status of the student's progression as both an individual and a team member. In order to measure this progression, we have gradually increased complexity of subjects and projects until the students had a certain level of confidence in their digital design using HDLs. Therefore, the remaining subjects and design projects can be completed faster and are of better quality as team projects than the individual projects previously assigned.

For more accurate evaluation, we have quantified detail works in each project by classifying it according to its required verification feature. For instance, functional verification is considered as a primary class. Timing verification is an intermediate class. Further synthesis effort, design analysis, or optimization is evaluated as a matured class. Based on our experience, most projects have been done as the primary or the intermediate class, because design efforts including implementation and verification still required a great deal of extra effort to familiarize students with correct usage of HDLs, debugging strategies and skills, and so on.

Tasks	Issues on the current practice	Future upgrade
Execution stage	Time limitation	Task will be given earlier (about 2 weeks)
	A single large team task	At least two teams will work on the task
Pipeline	Integration	Provide a pseudo pipeline as a reference
	Verification	Use func./timing sim. and FPGA emulation
Platform	FPGA (UP2): FLEX10K (3.7 LEs)	Large FPGA (UP3): Cyclone EP1C20 (20K LEs)
	UART & FPGA-based emulation	Labview & larger FPGA-based emulation

Table 2: Issues of our current practice and direction of the future upgrade

Since we discovered that a large project took longer than we expected, the material would be taught early enough to meet system integration schedule. The large project was assigned to a team twice the size of the others. A comparison of the performance results between the large team and the other smaller teams, however, indicated that two small teams performed better, because the small teams discussed each other's work and used each other's work as a cross reference. System integration will be improved by using a reference pseudo code so that the integration process can be done more smoothly, accurately and quickly.

Adding other features for the future design requires larger FPGA. Introducing more complex architectures and techniques needs more sophisticate verification environment than our current setup. Since synthesizing a processor platform with relatively large testbenches also requires larger FPGA than the one currently used, a larger FPGA board, such as Altera UP3, will be used in the future. Since our current processor platform emulation setup has limited capability for the further emulation, it will be upgraded to a new NI Labview-based emulation. In addition, enhancements of the course materials, including detailed diagrams and functional flow descriptions of each design task, is necessary to the novice designers, so as not to lead them in the wrong direction. We will add these issues to leverage our current practice so that it will be possible to do more matured design and verification in the same amount of time. The issues

chosen from our practice and the prospective solutions and the direction of the future upgrade are described in Table 2.

Conclusions

Although it was quite challenging to deliver so many subjects and materials in a single semester, the course derived the following benefits: lessons on digital hardware design from fundamentals to computer architectures; a hands-on experience on an entire processor design process using a customized rapid design method; a learning opportunity on HDLs; an experience on the FPGA-based design tools including simulation, emulation and synthesis; and a pre-industrial experience in structural design and RTL implementation of a soft RISC processor core.

We learned valuable lessons from our practices. In particular, precisely balanced laboratory workloads and seamless or smooth transition from classroom lecture to laboratory practices are the important non-technical aspects required to complete this course more efficiently and successfully. As technologies evolve, this course will be continuously supplemented with new technologies and educational materials. As an extended effort, we plan to develop a set of the graphical user interface (GUI)-based tools that will eventually assist a small group of students in designing their application-specific industrial quality of soft RTL synthesizable processor platform cores by clicking their computer mouse.

Acknowledgements

I would like to acknowledge Joshua W. Jones for his help in organizing laboratory works. Ralene Marcoccia -University Program Coordinator at Altera- donated UP2 boards and software.

References

[1] M. Mano and C. Kime, Logic and Computer Design Fundamentals, 3rd Edition, Prentice Hall, Oct 6, 2003.
[2] J. Hennessy and D. Patterson, Computer Architecture A Quantitative Approach, 2nd Edition, San Francisco, CA: Morgan Kaufmann, 1996.
[3] J. Hamblen and M. Furman, Rapid Prototyping of Digital Systems - Second Edition, Kluwer Academic Publishers, August 2001.
[4] R. Goering, "Platfrom-based design: A choice, not a panacea", EETIMES, Sep. 2002.
[5] T. Hall and J. Hamblen, "System-on-a-Programmable-Chip Development Platforms in the Classroom", *IEEE Trans. on Education*, 2004.
[6] A. Sangiovanni-Vincentelli, et. al, "Benefits and Challenges for Platform-Based Design", *Proc. of 41st Design Automation Conf.*, pp. 409-414, June, 2004.

YONG-KYU JUNG is an Assistant Professor at Texas A&M University in College Station. He received his B.S.E.E. from Korea University (Seoul) in 1985, an M.S.E.E. and a Ph.D. from Georgia Institute of Technology (Atlanta) in 1997 and in 2001 respectively. He was a vice president of VLSI technology (2001-2004) at VP Technologies Inc. His research interest is processor platform design automation for wireless communication.