2006-1146: A NEW RAPID MICROPROCESSOR SYSTEM DESIGN LABORATORY DEVELOPMENT FOR DIGITAL DESIGN EDUCATION

Yong-Kyu Jung, Texas A&M University

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 R4 Computing that is Rapid, Reconfigurable and Reliable computing systems design and implementation for future Reengineering.

A New Rapid Microprocessor System Design Laboratory Development for Digital Design Education

Abstract

This paper presents a new rapid microprocessor system design laboratory to be used in the early stages of digital design education. To reduce a gap between current digital fundamentals and computer design courses, a register-transfer level (RTL) microprocessor design, which provides both functional and structural features and implementation options of the design, is taught in the new laboratory. In addition, this rapid RTL microprocessor system design laboratory offers a closer pre-industrial, real-world design experience, because an RTL design is considered as one of the preferable forms of the silicon chip design.

Introduction

Because technology has evolved drastically, introducing design practices that are similar to real-world projects into the classroom is one of the crucial issues facing engineering education societies. Before discussing detailed activities, an examination of the “pros and cons” of current digital design practices in the classroom [1] is worthwhile. In industry, delivering the most marketable and typically large-scale, high-quality products in the shortest amount of time is critical. Engineering education, on the other hand, usually deals with more general subjects that may not often, if ever, be used during the post-academic engineering career, as well as with practices involving small design regardless of quality. In particular, students take a series of digital design courses – such as Digital Fundamentals, Logic Circuits and Computer Architectures – to learn microprocessor system design. Few intensive courses can compensate for the drawbacks of many courses taught in the current curricula. These intensive courses are typically coupled with laboratory exercises that include hardware design language (HDL)-based implementations. Because laboratory work is one of the most effective learning tools for novice students, providing intuitive, easy-to-use and expendable instruments is another important factor in the eventual delivery of better outcomes.

Since programmable devices and associated tool sets have been introduced in academia, digital system designs using field-programmable gate arrays (FPGAs) have often been added to existing curricula to enhance the practical learning process [2]. For instance, FPGA-based platforms along with configurable processor cores have been successfully used for various large classroom projects [3]. We are continuously encouraged to reduce the time spent teaching topics related to digital system design. We also have to adapt a rapid digital system design process for smoother migration toward the next level of courses. A customized processor platform and design environment for classroom instruction was used to achieve more efficient teaching and smoother migration [4]. In addition, this RTL processor platform implemented in HDLs accelerated the design process in classroom projects.

Objectives of the New Rapid Microprocessor System Design Laboratory

The main laboratory objective is to increase individual design opportunities and hands-on experiences for each student, so that they are motivated and prepared for team projects. Consequently, team performance is improved. The other objectives are to 1) actively engage more students in more complex designs than typical, less complicated design projects done on a smaller scale; 2) efficiently complete system-level design within a limited time period; and 3) swiftly achieve greater levels of achievement in digital system design.

Development of the New Rapid Digital System Design Laboratory

Since relatively large system-level design requires careful guidance if inexperienced students are to confidently complete assigned tasks, a new integration-oriented RTL design methodology including various VHDL design examples was developed to expedite learning and implementation of processor system design using Verilog HDL. As a result, students not only learn processor design and both HDLs concurrently through their implementation, they also obtain knowledge on rapid design procedures including design reuse techniques such as design-for-reuse (DfR) and design-with-reuse (DwR) [9].

New integration-oriented RTL design methodology

In order to teach rapid processor design within a single semester, selective contemporary computer architectures were taught prior to instruction in design instruction set architecture (ISA) and followed by behavioral implementation in the classroom. In the end, the individual student can develop an RTL processor system by using FPGAs in conjunction with verification tools under the new integration-oriented design methodology. In the new methodology, “rapid design completion” is one of the key aspects. The others – “continuous design improvement” and “various design diversity” – are considered necessary to leverage this course continuously, as well as to allow seamless migration to higher level courses.

Rapid Design Completion

HDL-based design and reuse constitute one feasible approach to achieve rapid digital system design in RTL. Since every design must be verified, students perform both simulation and emulation. The combined use of electronics design automation (EDA) tools for simulation and the flexible emulation tool can be preferred for FPGA-based laboratory projects. RTL design inherently obliges hierarchical, structural design; thus, how to integrate the components used in the large design becomes much more complicated than issues involving design encountered in small scale practices usually found in the early stages of laboratory work. Because there is insufficient time to practice such complex design integration by the traditional trial-and-error method, a pseudo pipeline is built to initiate large-scale integration. This pseudo pipeline is turned into a target RTL processor in the final stages of the project.

Integration of a Pseudo Pipeline

Since pipelined structures became essential in contemporary microprocessors, implementation and verification techniques of pipelined designs have been continually introduced to both academia and industry. Unlike other approaches [5, 6, 7], a pseudo pipeline is constructed by interconnecting the almost identical pipeline stages. Each pipeline stage follows the common interfacing rules illustrated in figure 1 for simplifying the entire pipeline integration procedure. For example, the nth stage, S_n, synchronizes its operation at both the rising and falling edges of its system clock. At the rising edge of the clock, the stage S_n communicates with the previous stage, S_{n-1}; reads valid input and stores it into a temporary storage, "S_n_Stg_bf"; and then the S_n stage acknowledges its successful reception to the S_{n-1} by transmitting a signal, "S_n_Valid". After acknowledged, the S_{n-1} stage withdraws its control signal, "S_{n-1}_CMD". This synchronized interface is repeated at the falling edge of the clock, but performed internally for executing the desired operations of the same stage.

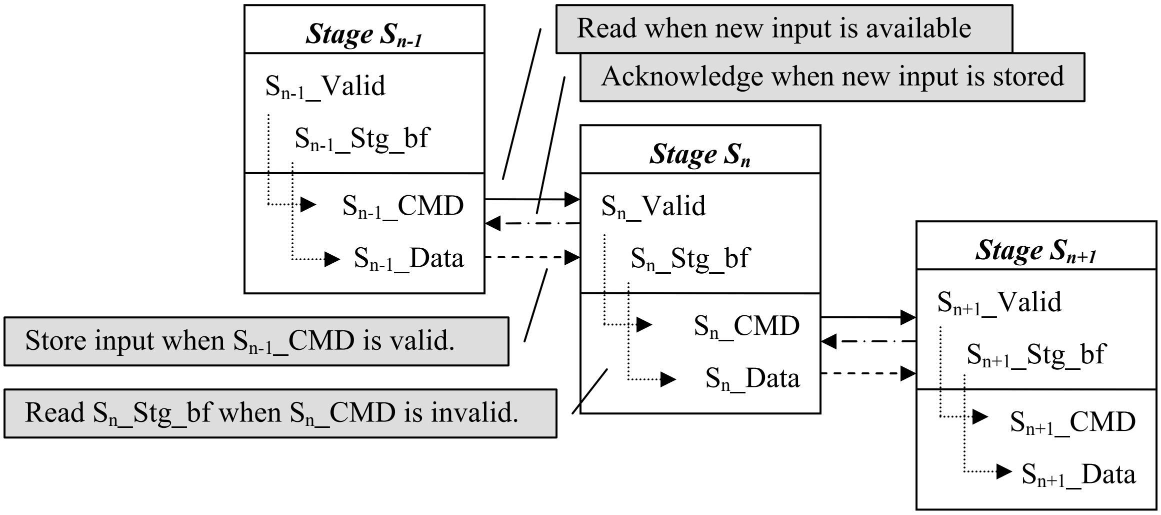

Figure 1: Intuitive, Expendable Integration for the Construction of a Pseudo Pipeline

In order to complete an entire processor system, a function for each stage in the pseudo pipeline is implemented and integrated one at a time. A sample laboratory assignment in [10] explains details of implementation and verification works of a stage in the pipeline. In this lab assignment, students are asked to implement an instruction decoder, which is the second stage of their pseudo pipelines.

Continuous Design Improvement and Design Diversity

In order to continuously enhance a digital design course, the HDL codes that students developed were reviewed with the existing codes written in different HDL for the next implementation. A hierarchical, modular HDL design library containing the HDL codes will impact effective dissemination, especially for accurate usage. The library structure also permits systematically adding advanced design for the future rapid design. Because students do not always possess equal design capability, various design tasks can motivate them to higher achievement. Simplified and abridged system integration continues to populate different types of processor design tasks for different levels of students. Therefore, the methodology facilitates advancement to the next level of digital system design courses. As a result, students can acquire knowledge and enrich their design experience while designing different architectures in the higher level courses.

Development of the Flexible Emulation tool for the Laboratory Practice

In order to take the laboratory practice beyond simulation to actual hardware implementation, FPGA-based debugging tools are necessary [8]. As illustrated in figure 2, the RTL emulation tool was developed for the new laboratory exercises, so that students can explore contemporary design environments including vendors and in-house simulation and emulation tools. Eventually, they will be able to verify their processor systems quickly and precisely. An RTL processor system is configured in an FPGA (a). Both instruction/data buses and control signals in FPGA are interfaced to the emulation engine (b) through the noise rejecting, shielded I/O connection boxes (c).

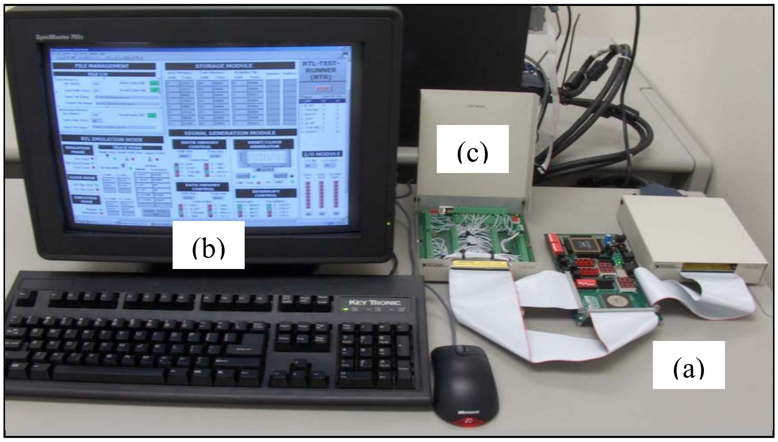

Figure 2: A Flexible RTL Emulation Setup

RTL simulations – functional and timing – kick off RTL verification in the new laboratory. Further RTL emulation is recommended, depending on the complexity and scale of the design. At the beginning of the system-level emulation, a group of tests can be executed either manually or automatically. These tests represent the same or similar types of operations or instructions for screening errors. Finally, after screening and debugging trivial errors, a full-scale extensive verification using entire benchmarks is undertaken. The emulator runs an enormous number of instructions on the real-time environment, and traces the instructions tested, which is a daunting task. In addition, it significantly improves evaluation procedures in terms of the accuracy and the amount of time and effort required to judge each student's laboratory works.

Organization of the New Laboratory for Rapid Microprocessor System Design

The new laboratory organization assists students with large and complicated system-level design, especially for its integration and verification. It also supports an efficient procedure to assess and evaluate both individual and team designs. The new laboratory consists of three sections: 1) digital logic component design; 2) pseudo pipeline design; and 3) RTL processor system design and verification. Table 1 summarizes the subjects and the detailed contents taught in each section. Many different design input formats have been prepared for various real-world implementations in the laboratory.

So that maintenance of laboratory works closely followed the lecture, most of the subjects taught in the classroom were used as individual projects. Most of students successfully demonstrated their design and implementation of a soft processor core within the time given. They verified their six-stage pipelined RISC cores by both functional and timing simulations. Extended emulation was recommended for the term group project. In the team project, a parallel execution unit consisting of the following units was developed: a functional, a branch processing, and a memory-access. Therefore, students had collaboration opportunities and opportunities to use their works as cross references; thus, every student in the team was actively involved the team activities.

Laboratory Subjects	Contents
HDL (Verilog) and FPGA/Tools	FPGA; RTL Synthesis; and Simulation/Emulation
	Verilog HDL and 16-bit Barrel Shifter
Microprocessor Design I *(Digital Logic Components Design in RTL)*	4-bit CLA Adder and CLA Unit
	16-bit CLA Adder/Subtractor and ALU
	Dual-port register and Memory system
Microprocessor Design II *(RTL 6-stage Pipeline using a Pseudo Pipeline: Fetch; Decode; Dispatch; Issue; Execution-FU/BPU/MAU; and Write-back)*	Pseudo Pipeline (6-stage) and Instruction Fetch
	Instruction Decoder
	Register Access (Dispatch/Write-back) and Issue
	Instruction Execution (ALU, B-Shifter, Multiplier, etc.)
Microprocessor Design III *(RTL Processor Design/Verification)*	6-stage RISC Pipelined Processor
	Verification of the RISC Pipelined Processor System

Table 1: Organization of the New Laboratory: Subjects and Contents

Conclusions

The integration-oriented RTL processor design methodology was successfully used throughout the laboratory projects. In particular, individual projects such as regular laboratory practices and a term team project allowed each student to perform her/his own tasks, as well as to collaborate with other team members. RTL emulation provided an accurate, effective way for evaluating complex designs by each student. Consequently, students were not only motivated and engaged the laboratory practices more enthusiastically, but they also achieved higher expectation of their design implementation in less time.

References

[1] Yong-Kyu Jung, “An Innovative Rapid Processor Platform Design for Early Engineering Education,” Proc. of ASEE Annual Conf., June 2005.

[2] J. Schneider, M. Bezdek, Z. Zhang, Z. Zhang, and D. Rover, “A Platform FPGA-based Hardware-Software Undergraduate Laboratory,” Proc. of IEEE Int. Conf. on Microelectronic Systems Education, June 2005.

[3] T. Hall and J. Hamblen, “System-on-a-Programmable-Chip Development Platforms in the Classroom,” IEEE Trans. on Education, 2004.

[4] Yong-Kyu Jung, “WIP – A Rapid Design Methodology for FPGA-based Processor Platform Design Education,” Proc. of 35th ASEE/IEEE Frontiers in Education Conf., Oct. 2005.

[5] J. R. Burch, and D. L. Dill, “Automated Verification of Pipelined Microprocessor Control,” CAV, June 1994.

[6] J. Levitt and K Olukotun, “Verifying Correct Pipeline Implementation for Microprocessors,” Proc. of International Conference on Computer Aided Design (ICCAD) Conf., Nov. 1997.

[7] J. Hamblen, H. Owen and S. Yallamanchili, “An Undergraduate Computer Engineering Rapid Systems Prototyping Laboratory,” *IEEE Trans. on Education*, Feb. 1999.

[8] M. Holland, J. Harris, S. Hauck, “Harnessing FPGAs for Computer Architecture Education,” Proc. of IEEE Int. Conf. on Microelectronic Systems Education, June 2003.

[9] J. Chang and S. Agun, “On Design-For-Reusability in Hardware Description Languages,” Proc. of IEEE Computer Society Annual Workshop on VLSI (WVLSI'00), 2000.

[10] Yong-Kyu Jung, Rapid Digital System Deisgn Laboratory Assignment: Instruction Decoder Design and Implementation in Verilog, http://etidweb.tamu.edu/classes/entc249/ENTC%20249%20lab7_handout.pdf, Sep. 2005