MASSIVE DIGITAL DESIGN EDUCATION

for Large Amount of Undergraduate Students

MILOŠ BEČVÁŘ, HANA KUBÁTOVÁ, MARTIN NOVOTNÝ
Czech Technical University in Prague, Department of Computer Science and Engineering,
Karlovo nám. 13, 121 35 Prague 2, Czech Republic

1. INTRODUCTION

Our university has relatively great experience with teaching digital design fundamentals for large amount of students. In past, the number of labs in digital design courses was minimized due to their complicated organization and low effectiveness. Moreover, due to the relative absence of labs, students had weak comprehension of problems and topics.

Rapid development of programmable devices and EDA tools has enabled development of new, technology-driven learning processes in education, thus facilitating the learning process (in sense of faster and better comprehension of presented knowledge), making it more attractive and less time consuming.

In this paper we present our experience with usage of programmable technology, as well as relevant EDA tools for massive education. We were inspired by courses at top universities, e.g. [2], but the composition of labs we adapted for 10 times larger amount of students. In introductory course students get knowledge of digital design principles as well as an experience with design kits and design tools. This knowledge and experience are then utilized in consecutive courses.

This paper is organized as follows: Section 2 describes used technology, Section 3 introduces our hardware courses - Logical Circuits (3rd semester), Computer Units (4th semester) and Architectures of Computer Systems (5th semester) and Section 4 concludes the paper.

2. TECHNOLOGY AND DESIGN TOOLS

Xilinx and Altera are two most significant companies in the market of programmable devices. Their products, including circuits, design kits and development systems, are comparable. As Xilinx provides better university support, we have decided to use Xilinx products. ISE 7.1i is the development system in which students design their circuits. In introductory course, the designs are downloaded into the design kits Digilent XCRP [3], equipped with CPLD Xilinx CoolRunner XCR3064 [4]. In intermediate course the design kits equipped with FPGA Xilinx Spartan2E 200 are used.

ModelSim simulator is used for simulation. In introductory and intermediate courses students simulate designs created in Xilinx ISE. Designs in advanced courses are created in HDL Designer and then again simulated in ModelSim.

3. DIGITAL DESIGN COURSES

Courses Logical Circuits (3rd semester), Computer Units (4th semester) and Architectures of Computer Systems (5th semester) are obligatory for all students of Informatics and Computer Science branch of our bachelor study program. Each of these courses is attended by more than 300 (currently 350) students.

3.1 LOGICAL CIRCUITS

This course covers knowledge of basic principles and methods: combinational circuits, logic minimization methods, sequential circuits, FSMs, typical blocks used in processors (adders, multiplexers, decoders, registers, counters, etc.), analysis (hazards, maximum frequency estimation) and introduction to the digital testing methods.

Previously, students attended 14 lectures, 12 seminars and 2 labs. In labs students realized logic circuits on solderless breadboards with TTL integrated circuits and wires. Such circuits were untransparent and unreliable mainly due to the complicated wiring. The construction of the functional circuit spent a plenty of time. Moreover, if the simulation was necessary, the circuit had to be redrawn into the simulation system.

Currently, students attend 14 lectures, 7 seminars and 7 labs. In labs they “construct” their circuits in schematic editor that is the part of EDA tool. When finished, they translate the circuit into the bitstream and download the circuit into the design kit equipped with CPLD. The circuit in a schematic form is transparent. If students make some mistake, correction is simple and fast. Moreover, if simulation is needed, the scheme (together with stimuli) is used as an input to the embedded simulator.

The productivity has increased about 4 times in comparison with former education. In past, during one lab students were able to realize 1-2 functions of 4 input variables. Currently, typical tasks solved during one lab are:

- Two-bit adder a) constructed from two one-bit full adders, then b) realized as three 5-input functions; both versions are also simulated.
- Electronic code-lock realized as both Mealy and Moore FSM. This helps students to better understand the difference between these two basic types of FSMs (in past, students designed FSMs “on paper” only). Both versions are also simulated.

All circuits are designed, simulated and then downloaded into design kits. Note that the complexity of logic circuits is increasing proportionally to the student’s experience. Some blocks, realized in earlier labs, can be and are reused in further labs, similarly to software construction methods where the final design is typically composed from previously debugged blocks.

Students may download limited versions of design tools (ISE WebPACK and ModelSim XE Starter) from Internet [4]. Therefore, students may prepare their labs and projects at home. Experience with design tools is reused in further courses.

3.2 COMPUTER UNITS

Computer Units course deals with construction of more complex digital circuits which create a basis of a computer. It reuses knowledge from previous courses and teaches how the datapath arithmetic circuitry and several types of controllers can be designed.

In the beginning of course, the simple 16-bit processor DOP is used as a learning tool to demonstrate how the simple sequential processor can be constructed. Second goal of this example processor is to teach microprogramming and students are required to develop and verify a new instruction on microprogrammed controller [5]. This task is performed in the first half of semester.

In the remaining part of semester, theoretical seminars about construction of datapath and controller were held in classroom. Efficiency of these seminars was low and we have decided to add a new laboratory project at the end of semester.

Students are required to design, verify and implement a simple datapath application and controller on Xilinx Digilent D2E board. To decrease the pressure on students at the end of semester, this task is performed in groups of two. Datapath designs are various types of adder/subtracter, multiplier/divider and shifter circuitry, overall 10 different datapath designs are given. Every datapath can be combined with one of 3 types of a controller: binary-encoded, one-hot encoded or microprogrammed. It means that we have about 30 different topics for 300 students. On average, only 5 groups are solving the same task.

Result of these new laboratory projects was a notable improvement of knowledge about computer arithmetic and controller design issues during examination. We have also successfully reused the knowledge of EDA tool Xilinx ISE gained in previous course.

3.3 ARCHITECTURES OF COMPUTER SYSTEMS

Undergraduate computer architecture course completes the sequence of obligatory hardware courses. It covers more advanced topics from the processor architecture, the memory subsystem and parallel systems. This course is on the higher level of abstraction and detailed projects on FPGAs are not feasible. One of the obvious topics is an instruction pipelining. Traditionally students are introduced into this topic by using functional simulator of DLX 32-bit RISC processor [6]. To cover also the real circuit implementation of the pipeline and its hazard detection and avoiding logic, the visual VHDL model HDLDLX was created [7]. Students are required to write the simple VHDL parallel statements defining the condition when data hazards are detected and forwarding or a pipeline stall is enforced. Comparing to old years the students after completing new Logical Circuits and Computer Units courses completed this task easier. This shows that previous courses provided students with more background and understanding of synchronous circuits which is essential to complete this task.

Future enhancement of this course will include the demonstration of implemented 5-stage pipelined DLX processor inside Xilinx FPGA.

Table 1: Summary of labs content, hardware kits and design tools

Course	Logical Circuits	Computer Units	Architectures of Computer Systems
Topics taught	Basic principles	Building blocks of processors	Advanced processor architecture - pipelining
Labs	Combinational and sequential circuits, FSMs	Microprogram for DOP processor; unit with controller	Editing visual VHDL model of DLX processor
Hardware kits	Digilent XCRP (CPLD CoolRunner)	Digilent D2E-DIO2 (FPGA Spartan2E)	Not used. Planned to use Digilent D2E-DIO2 (FPGA Spartan2E)
Design tools	Xilinx ISE, ModelSim	DOP processor simulator, Xilinx ISE, ModelSim	WinDLX simulator, HDL Designer, ModelSim

4. FINAL REMARKS

The relatively great amount of students is taught the fundamentals of the digital design using contemporary design tools. Two main effects have been achieved: used technology allows performing more experiments and the experiments can be more complicated. Moreover, as students use state-of-the-art design tools, circuits and design methods, they are better prepared for nowadays demands of industrial practice.

REFERENCES

[1] Gajski, D. D.: „Principles of Digital Design”, 447 pp. Prentice-Hall International, Inc., 1997

[2] MIT OpenCourseWare: http://ocw.mit.edu/OcwWeb/Electrical-Engineering-and-Computer-Science/6-111Spring2004/CourseHome/

[3] Digilent, http://www.digilentinc.com

[4] Xilinx, http://www.xilinx.com/

[5] Bečvář, M., Pluháček A., Daněček J.: DOP - a CPU Core for Teaching Basics of Computer Architecture In: *Proceedings of the Workshop on Computer Architecture Education (WCAE 2003)*

[6] Hennessy,J.L., Patterson, D.A.: Computer Architecture : A Quantitative Approach, second edition, Morgan Kaufmann Publishers, 1995

[7] Bečvář, M.: Teaching Basics of Instruction Pipelining with HDLDLX In: *Proceedings of the Workshop on Computer Architecture Education (WCAE 2004)*

ACKNOWLEDGEMENTS

This work has been partially supported by the grants CTU0418613 (Support of the new laboratory courses of structured study of Informatics and Computer Science), GA102/04/0737 (Modern methods of digital systems design) of the Czech Grant Agency and the MSM6840770014 research program.