A Computer Architecture Education Curriculum through the Design and Implementation of Original Processors using FPGAs

Yutaka Sugawara, Kei Hiraki

Department of Computer Science
University of Tokyo
Bunkyo-ku, Tokyo, Japan
{sugawara, hiraki}@is.s.u-tokyo.ac.jp

Abstract

In this paper, we present the new curriculum of the processor laboratory of the Department of Computer Science at the University of Tokyo. This laboratory is a part of the computer architecture education curriculum. In this laboratory, students design and implement their own processors using field-programmable gate arrays (FPGAs), and write the necessary software. In 2003, the curriculum of the laboratory was changed, the main change being that the FPGA was changed to a large one to increase the range of design trade-offs. As a result, students have been enabled to implement the techniques used in modern processors such as FPU, cache, branch prediction, and superscalar architecture. In this paper, we detail the new curriculum and note the educational results of the year following the changes. Especially, we focus on the educational advantages of the large FPGA size.

1 Introduction

In architecture education, understanding existing architectures and acquiring skills to design new architectures are important goals for the students. Typical architecture education curriculums include both lectures and laboratories to achieve the goals. Concerning the lectures, architecture educations based on quantitative approaches are established using popular text books such as one written by Hennessy and Patterson [1]. However, students cannot acquire skills to design new architectures from just lectures. Laboratories are necessary to train students how to design new architectures.

For that reason, the processor laboratory [2] was introduced in the Department of Computer Science at the University of Tokyo. The laboratory started in 1992 as a part of the computer architecture education curriculum. In the laboratory, junior students design and implement processor systems using FPGAs. They build entire computer systems including processor architecture and software. The laboratory's main goals for the students are:

1. To precisely and concretely learn the internal structure and behavior of processors
2. To acquire a sense of trade-offs in processor architecture design
3. To experience the trade-offs involved in an entire system including software and hardware

The first goal is important because being familiar with basic processor structure is important for a good understanding of architecture techniques. The second goal is important because selecting an optimal architecture under a given condition is the most important topic in designing computer architecture. The third goal is important for understanding how the performance of a computer system is affected by each of its various parts.

The first goal cannot be completely achieved from just lectures because omitted structures of processors are explained in typical cases. Even when all the signals in a processor are shown, it is hard to explain changes of signals when each instruction is executed. On the other hand, when students implement their own processors in the laboratory, they can more precisely and concretely understand the behavior of all the internal signals of processors.

In lectures, a limited part of the second goal is achieved when the targets of trade-offs are detailed by the lectures. Furthermore, some parts of the trade-off conditions are often ignored to simplify the problem. However, when the students design their own architectures in the laboratory, they can experience a wide range of real trade-offs.

The third goal is not achieved in lectures because it is hard to precisely model the trade-offs of an entire system. On the other hand, when students build whole systems of their own including software and hardware, they experience real trade-offs in respect to entire systems. Especially, they can learn how to divide functions between hardware and software.

Because FPGAs are used in the laboratory, students can immediately run the processors they have designed.

Therefore, students can try many design alternatives to experience trade-offs of the architecture and of the entire system.

The curriculum of the processor laboratory was changed in 2003, the main purpose being to increase the range of architecture design trade-offs students can experience. Therefore, we changed the FPGA used in the laboratory to a large one.

In the previous curriculum, 5K-gates FPGAs were used. Therefore, though students learnt many architecture techniques in lectures, most of the techniques could not be implemented in the laboratory because of the FPGA size limitation. For example, it was impossible to implement techniques used in modern processors such as FPU, cache, branch prediction and superscalar architecture. 1M-gates FPGAs are used in the new curriculum; therefore, students can implement most of the techniques learnt in lectures as long as they have enough development power. In this paper, we present the details of the new processor laboratory curriculum and the educational results of year following the introduction of the changes. Especially, we focus on the educational advantages of the large FPGA size.

In Section 2, we explain both the previous and the new curriculum in the processor laboratory of our department. Section 3 shows the educational results of the past year in the laboratory. In Section 4, we present related works, and Section 5 concludes the paper.

2 The Processor Laboratory Curriculum

2.1 Previous Curriculum

Until 2002, Xilinx XC4005, a 5K-gate FPGA, was used in the processor laboratory. Students were divided into groups of five or six members. The goal of the laboratory was to run a ray tracer on students' original processor systems as fast as possible.

Each group builds a processor system board, an example is shown in Figure 1. Wrapping wires are used for connecting the components. The processor was implemented on an XC4005 FPGA. Total of 256KB SRAMs and 128KB ROMs were available to implement the memory system. The processor board could communicate with workstations via uPD71051 serial I/F. This serial I/F was used for scene data input and image data output of the ray tracer running on the processor board. The students were entirely responsible for the design of the architecture of their processor, the memory system, and the I/O system on the board.

The software tools were also made by the students themselves. Each group developed a runtime library, a cross assembler, a simulator, and a cross compiler for the processor system. The runtime library included communication routines of the serial I/F and floating point calculation primitives.

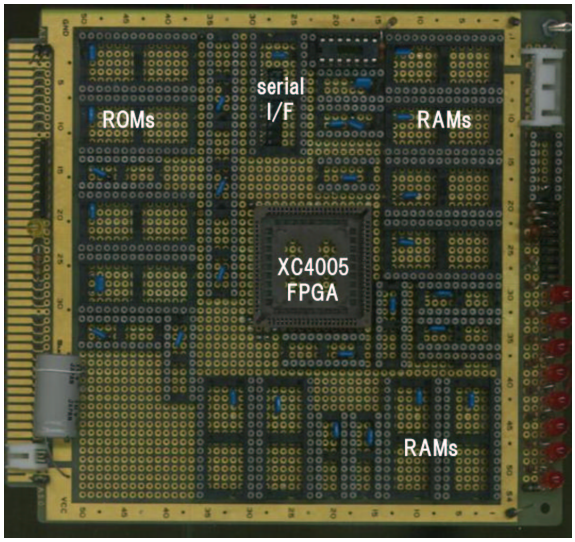

Figure 1: Example of a processor board used in the previous curriculum

The main advantages of the previous curriculum were (1) students could build and understand the entire system, and (2) they could learn what functions should be implemented using hardware when hardware size is limited. However, because of the FPGA size limitation, students could not implement most of the architecture techniques they learnt in lectures.

2.2 New Curriculum

The curriculum of the processor laboratory was changed in 2003; the main purpose being to enable students to implement most of the architectures they learnt in the lectures, something that was not achieved in the previous curriculum. The main change was to increase the FPGA size; thus, the Xilinx XC2V1000, a 1M-gates FPGA was introduced.

However, the change of the FPGA meant that building "a whole system" was no longer possible. This is because dedicated system boards are used for the laboratory and students are not required to wire components. When the components are wired using wrapping wires, the resulting circuit cannot operate at enough clock speed for the new FPGA. Furthermore, pin pitches of modern chips are too fine to be wired by hand. Therefore, we gave up making the students wire the boards by themselves.

The new system board is shown in Figure 2. The FPGA board includes an XC2V1000 FPGA, total 4MB of synchronous SRAMs, 128MB PC100 SDRAM, and USB I/F. An extension board is used for implementing additional I/O circuits by hand. In the laboratory last year, most groups implemented 7-segment LED arrays on the extension boards for debugging.

In a contest at the end of the semester, students present the processors they have made in the laboratory, and the performances of the processors are evalu-

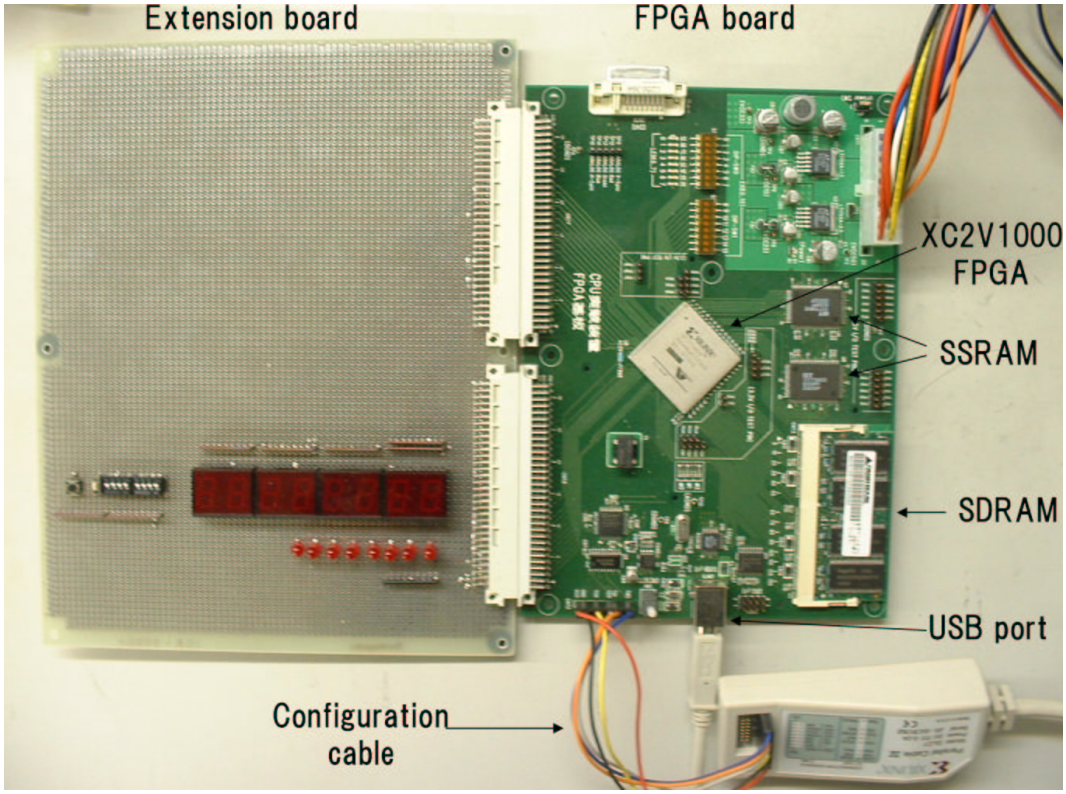

Figure 2: The new curriculum processor board

Figure 3: Output image of the new ray tracer

ated using a benchmark program. The benchmark program is an extended version of the ray tracer used in the previous curriculum. The output image of the new ray tracer is shown in Figure 3. Xilinx ISE6.1i tools are used for processor design. Table 1 shows comparisons between the previous and new curriculums.

	previous	new
FPGA	XC4005 (5K gates)	XC2V1000 (1M gates)
memory	SRAM (256KB, 100ns), ROM (128KB, 100ns)	SSRAM (4MB, 100MHz), SDRAM (128MB, PC100)
I/O	uPD71051	FTDI245(USB), etc.

Table 1: Comparisons between the previous and new curriculums

3 Educational Results

3.1 Design Result

In 2003 in the processor laboratory, the first year of the new curriculum, 6 groups designed processor systems. Table 2 shows the results of each group. The score is the execution time of the ray tracer measured in the end of semester contest.

As shown in Table 2, all groups implemented floating point units, and two groups implemented caches, important for the performance of modern processors. Implementing these techniques would have been impossible in the previous curriculum. Figure 4 shows a block diagram of the processor designed by group 1 [3] in Table 2.

group No.	Clock (MHz)	score (sec.)	features
1	50	35	FPU, pipeline I-cache, D-cache
2	50	45	FPU, pipeline I-cache, D-cache
3	40	178	FPU
4	12.5	548	FPU
5	50	641	FPU
6	50	N/A	FPU, pipeline

Table 2: Contest results of each group

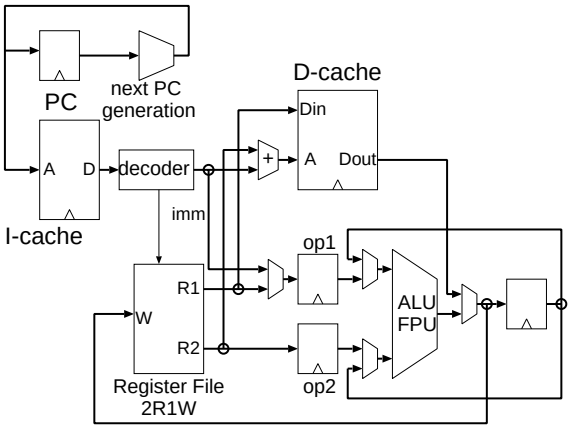

Figure 4: Block diagram of group 1's processor

In addition, 2 senior students voluntarily designed processors during the past year. One student implemented a processor with dynamic instruction scheduling using Tomasulo's algorithm. The other student designed a 2-way chip multiprocessor architecture. Furthermore, another two students are now voluntarily designing 4-way superscalar, and SIMD architectures, respectively, by extending processors they designed in the laboratory. Since these processors require many resources, they could not have been implemented using the FPGA of the previous curriculum.

3.2 Achievement of Educational Goals

Understanding Processor Structure In both the previous and the new curriculums, most of the students succeeded in implementing complete processors. This result shows that they gained an understanding of the structure of operational processors. Therefore, both curriculums successfully achieved this goal.

Acquiring a Sense of Architecture Trade-offs As detailed in Table 2, students successfully implemented FPUs and caches under the new curriculum. Further, the large FPGA enabled some eager students to implement more challenging techniques such as Tomasulo's algorithm, chip multiprocessor architecture, superscalar architecture, and SIMD architecture. None of these techniques could have been implemented using the FPGA used in the previous curriculum. Therefore, the new curriculum enabled students to experience wider ranges of trade-offs than in the regime of the previous curriculum.

Learning Trade-offs of Entire System As described in Section 2, board wiring is unnecessary in the new curriculum because a dedicated system board is used. Therefore, in the new curriculum, some of the trade-off conditions of the processor system are fixed; whereas in the previous curriculum, the processor board was fully designed and wired by students. In this respect, the previous curriculum was better than the new one.

4 Related Works

To the best of our knowledge, the processor laboratory of our department [2] has instituted the first curriculum in which students design and implement their own processors using FPGAs. Though there are many ideas using FPGAs for computer architecture education, most curriculums fully or partially specify the architecture that the students learn [4][5][6]. Our curriculum, though, allows students themselves to decide the architecture they will implement. Because all the necessary software is also made by the students themselves, instruction sets and execution models are not restricted. Therefore, students can experience a wider range of design trade-offs than usually possible.

In an idea described by Gray [7], students learn architecture trade-offs through modifying a given processor to improve the performance. However, because the baseline architecture is specified in this situation, the range of trade-offs is limited.

The CITY-1 framework [8] is similar to our curriculum in that students design their own processor architecture. However, some reference implementations are presented to induce students' ones. On the other hand, our curriculum encourages students to construct their own architecture without any guiding model.

5 Concluding Remarks

In this paper, we presented our department's newly introduced processor laboratory curriculum. The laboratory is a part of students' computer architecture education; whereby in the processor laboratory they design and implement their own processors using FPGAs. Students also write the software necessary for the processors.

The main purpose of the curriculum change was to enable students to implement most of the architecture techniques they learned in lectures; thus the increased FPGA size. In the new curriculum, 1M-gates FPGAs are now used, compared to the 5K-gates FPGAs used

previously. The result is that students can now experience a wide range of trade-offs. In the laboratory, students really implemented modern techniques such as caches and FPUs which could not have been implemented under the previous curriculum. The large FPGA size is certainly useful for teaching architecture design trade-offs.

The main drawback in the new curriculum is that students cannot design the whole system because the system board has been pre-wired. In this respect, the previous curriculum was better; though to improve this situation, from this year, we will have students construct serial I/O circuits.

References

[1] J. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach, Second Edition.* 1995.

[2] T. Matsumoto and K. Hiraki, "Laboratory of Designing Original Processors using FPGAs," in *Proceedings of the 2nd Japanese FPGA/PLD Design Conference & Exhibit(in Japanese)*, pp. 289–302, June 1994.

[3] http://www.nvaca.com/cpu/.

[4] Murray Pearson, Dean Armstrong and Tony McGregor, "Using Custom Hardware and Simulation to Support Computer Systems Teaching," *Workshop on Computer Architecture Education*, 2002.

[5] Daniel Ellard, David Holland, Nicholas Murphy, Margo Seltzer, "On the Design of a New CPU Architecture for Pedagogical Purposes," *Workshop on Computer Architecture Education*, 2002.

[6] Ross Brennan and Michael Manzke, "On the Introduction of Reconfigurable Hardware into Computer Architecture Education," *Workshop on Computer Architecture Education*, 2003.

[7] J. Gray, "Hands-on Computer Architecture - Teaching Processor and Integrated Systems Design with FPGAs," *Workshop on Computer Architecture Education*, June 2000.

[8] Ryuichi Takahashi, Noriyoshi Yoshida, "Diagonal Examples for Design Space Exploration in an Educational Environment CITY-1," in *Proc. of IEEE International Conference on Microelectronic Systems Education*, pp. 71–73, July 1999.