

Various Approaches to Learn the Computer Architecture and Organization Course

Goce Armenski¹, Sasko Ristov², Marjan Gusev³, and Magdalena Kostoska⁴

Abstract—Learning the Computer and Organization course is very difficult for computer science students. Apart of the course difficulties, its "hardwaredness" discourages the students to learn the course in depth even more. The computer science students prefer to comprehend the software based courses, where they use the computers only as a black-box tool to run their software programs or web applications or services, rather than to learn how a computer works, especially today's complex multiprocessor architectures, memory hierarchy, pipelining and parallelization, hazards and so on. In this paper, we present several approaches that we have introduced in the Computer Architecture and Organization course. The increased number of students in recent years (from 200 to 1000 per course) motivated us to change the course by introducing several online learning tools and visual simulators to provoke the students' curiosity about the internal organization of the computer. The students are satisfied of these educational tools, which helped them to overcome the required material.

Index Terms—e-Assessment; Hardware; Online Learning Tools; Simulators.

I. INTRODUCTION

Increased number of the computer science students enforced universities to continually improve their programs in order to follow the rapid development of new IT and more globally ICT techniques and technologies. New e-learning trends have changed the traditional way of teaching in the classroom. This concept is excellent for software based courses, where the students can use online tools for debugging and checking the correctness of their programs [25], which also can be used for detecting plagiarism.

However, hardware based courses in computer science curricula cannot be based on e-learning only. The e-learning tools can be used as additional learning tools, but still the students need a lot of practical work, especially in the laboratory.

In this paper we present several different approaches on how to use the e-learning tools in the hardware based Computer Architecture and Organization course. Online e-Assessment system is used both for assessing a student's knowledge and online learning. It is also used for assessing through

learning during the laboratory exercises. Additionally, two visual simulators are used. These educational tools made the course more interesting.

The rest of the paper is organized as follows. Several approaches on how to organize the Computer Architecture and Organization course are presented in Section II. Section III explains the course's organization. In Section IV we present the various learning approaches that we use in the course. The final Section V concludes our work.

II. RELATED WORK

This section presents the related work in the area of learning the hardware based courses, focusing on the Computer Architecture and Organization course.

There are many different pros and cons of using visual simulators or working on real hardware. Nevertheless, the teachers must introduce appropriate hands-on exercises, homework assignments or projects, in addition to the lectures. Liang [14] surveys the proposed hands-on assignments and projects.

Hardware based courses require practical work on real hardware [11], [19], [12]. FPGA-based configurable processors are cheap and can be used in laboratory exercises [26], [13] and the students can develop, implement and monitor both hardware and software of multi-core processor systems.

On the other side, visual simulators together with appropriate hands-on exercises will lighten the learning about the computer architecture and will improve the students' interest in hardware [16], [21]. Although there are many visual simulators, such as [18], [5], [10], [4], [9], [22], [23], none of them can cover all topics in computer architecture and organization.

III. ABOUT THE COMPUTER ARCHITECTURE AND ORGANIZATION COURSE

This section briefly describes the Computer Architecture and Organization course.

A. The Course Organization

The course content is delivered to the students with five classes per week, organized as: 2 classes of theoretical lectures, 2 classes of tutorials and 1 class of practical laboratory tutorials. A student gains 6 ECTS after passing the course.

The course syllabus covers all relevant topics proposed by the ACM/IEEE recommendations [2]. The used textbook [17] covers: Computer abstractions and technology; Control including the MIPS instruction set; Computer arithmetic; Organization of the processor and datapath; Memory and storage organization; and Multi-chip multi-core multi-processors.

¹ G. Armenski is with University Ss Cyril and Methodius, Faculty of Computer Science and Engineering from 2001, Rugjer Boshkovik 16, PO Box 393, 1000 Skopje, Macedonia, Email: goce.armenski@finki.ukim.mk

² S. Ristov is with University Ss Cyril and Methodius, Faculty of Computer Science and Engineering from 2008, Rugjer Boshkovik 16, PO Box 393, 1000 Skopje, Macedonia, Email: sashko.ristov@finki.ukim.mk

³ M. Gusev is with University Ss Cyril and Methodius, Faculty of Computer Science and Engineering from 1989, Rugjer Boshkovik 16, PO Box 393, 1000 Skopje, Macedonia, Email: marjan.gusev@finki.ukim.mk

⁴ M. Kostoska is with University Ss Cyril and Methodius, Faculty of Computer Science and Engineering from 2008, Rugjer Boshkovik 16, PO Box 393, 1000 Skopje, Macedonia, Email: magdalena.kostoska@finki.ukim.mk

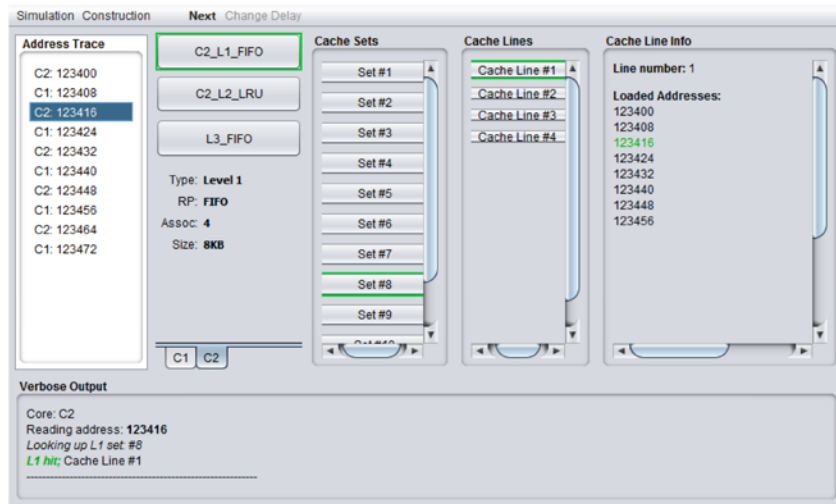


Fig. 1. A snapshot of the EDUCache simulation

The tutorials support these learning topics, including Performance parameters; Data presentation and codes; Computer arithmetic; Digital logic design; Flip-flops and memories; and Essential design of digital logic, including VHDL. Practical laboratory tutorials correspond to the theoretical tutorials.

B. Increased Number of Students

Usually, the lectures and tutorials are organized in larger groups of 80-120 students, while the practical laboratory tutorials are conducted in computer laboratories in groups of up to 20 students. About 200 students were enrolled per course until 2011 and one or two professors with several assistants covered the lectures and tutorials. Starting from 2012, the number of the students has increased even more, i.e., the course enrolled about 1000 students.

Several learning tools were introduced and the course methodology was changed towards the improvement of the course and the student success rate.

IV. LEARNING TOOLS FOR THE COURSE

This section presents the learning tools that are used in the Computer Architecture and Organization course. They are grouped into two groups: Visual simulator tools and online learning tool.

A. Visual Simulators

Since this is the course in the first year of study, we introduced visual simulators, instead of working on a real hardware. We believe that working on a real hardware will discourage even more the computer science students to learn, rather than to awake their curiosity. They prefer more software-based things, instead of hardware-based. This practical approach visualizes the concepts of low level computer digital logic and helps the students learn better and easier how the computer works.

Two visual simulators are used in the course: EDUCache [3] and HADES [8]. The following sections briefly present both visual simulators.

1) *The EDUCache Simulator*: The EDUCache simulator is developed by our students to be used for the Computer Architecture and Organization course. It is JAVA-based tool, which visually explains how a cache memory works, its organization, especially various cache mapping, cache replacement policies, cache positioning, cache lines and blocks, associativities, etc.

The students can design a CPU with an arbitrary number of cache levels, each level with unique parameters: cache size, replacement policy, cache line, and associativity. This yields different number of sets, as well.

ACM/IEEE-CS paid special attention to parallel and distributed computing by adding new knowledge area [1]. EDUCache also supports a simulation of multi-core chips and multilevel cache usage. A processor can have one or several homogeneous cores, each of which has access to a particular cache level of today's L1, L2 and L3. These multi-level caches can be private per core or shared among several or all cores.

A student can design an existing CPU with existing cache hierarchy. But the EDUCache allows the students to design their own cache hierarchy in order to achieve the benefits of different CPU architectures and organizations. For example, AMD CPUs use small cache set associativity of the L1 cache and smaller L3 caches, while Intel's CPUs are designed with higher cache set associativity of the L1 cache and bigger L3 caches.

Afterwards, a student can define a specific data access pattern and visualize its execution presenting each data access as cache hit or miss in each cache level, as depicted in Figure 1. This is the most powerful feature of the EDUCache simulator because the computer science students use their software based skills to learn hardware based courses.

Several predefined hands-on are proposed to use the EDU-

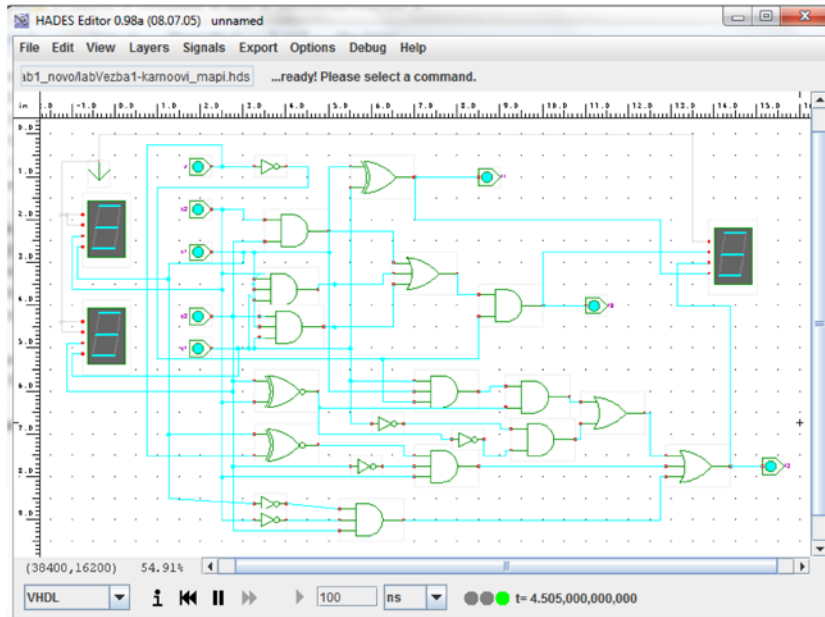


Fig. 2. A snapshot of the HADES simulator

Cache simulator in the Computer Architecture and Organization course [20].

2) *The HADES Simulator:* HADES [24] (Hamburg DEsign System) is also an interactive java based simulation tool, which is also used by the students to design and simulate basic digital systems design, as shown in Figure 2. It is a fully interactive simulator, which consists of graphical editor, discrete-event based simulation engine(s), libraries of simulation components etc.

Hades includes a lot of basic- and complex-gates with up to six inputs, several standard flip-flops, different I/O models, including animated LEDs or seven-segment displays and interactive switches, memories with graphical editors, LC-displays and so on.

The students can simulate three different processor families and architectures:

- The 32-bit R3051 microprocessor (MIPS R3000 family) from IDT;
- The 8-bit PIC16 microcontrollers from Arizona Microchip (namely the PIC16C84 and PIC16F84); and
- The 4-bit Intel MCS4 (i4004) microprocessor chip set; the first microprocessor in history.

B. Online learning and assessment tool

Our e-Assessment system has been used more than 15 years for many courses [6] and several upgrades were introduced during these years. It is used both for online learning and assessment. Additionally, Moodle Learning Management System [15] is used in the course.

1) *e-Assessment:* The e-Assessment system supports four question types:

- Multiple choice questions (MCQ);

- Single choice questions (SCQ);
- single value input questions (SVQ); and
- Essays (EQ).

MCQs are mostly used by the teachers in the question database, in front of SCQs. The other two question types are less used. The experience showed that SVQ produced several problems during the assessments and students are usually revolted. For example, presenting floating points numbers (dot or comma), rounding, writing the units, etc. EQ are not used although they are easy to define, i.e., they also don't need answer options such as SVQs. We believe that the main reasons are the increased number of students per course and possibility of unproper grading.

2) *Online learning:* The e-Assessment system has another module for online learning, which uses its e-Assessment engine to iteratively ask questions, thus enforcing the student to move from one learning objective to another. The online learning tool is organized as follows. The course is divided in several lectures (per week), each of which consists of knowledge items. Each knowledge item contains several learning objectives. Each student has to pass all knowledge items each week by answering the questions.

The system has several strategies in order to motivate the students. The navigation algorithm is the first strategy, which enables the students to move in the knowledge tree in an adaptive manner. The next strategy defines when a given knowledge item is learned. That is, each learning objective is passed after three correct answers in a row [7], while a knowledge area is passed after all learning objectives are passed individually, plus the mixed questions of the whole knowledge area (questions that cover several learning objectives of a single knowledge area).

These strategies enable a unique learning path for a student, that is, better students will move to the next knowledge area faster, while the worse students will need to spend more time on learning.

V. CONCLUSION

Learning the Computer Architecture and Organization course is difficult, especially for computer science students and the universities are making efforts to lighten it. This paper presents how several approaches for learning are used in this direction. We join the use of visual simulators and online learning tool.

Our self-developed EDUCache simulator, which covers the main part of the course material and maybe the hardest one (CPU architecture and organization), and the HADES simulator for digital circuits show the theory in practice. However, they are supplemented with online learning tools for e-Assessment and online Learning through assessment.

REFERENCES

- [1] ACM / IEEE-CS, "Computer science curricula 2013: Curriculum guidelines for undergraduate degree programs in computer science," 2013. [Online]. Available: <http://www.acm.org/education/CS2013-final-report.pdf>
- [2] ACM/IEEE-CS Joint Interim Review Task Force, "Computer science curriculum 2008: An interim revision of cs 2001, report from the interim review task force," 2008. [Online]. Available: <http://www.acm.org/education/curricula/ComputerScience2008.pdf>
- [3] B. Atanasovski, S. Ristov, M. Gusev, and N. Anchev, "Educache simulator for teaching computer architecture and organization," in *Global Engineering Education Conference (EDUCON), 2013 IEEE*, Berlin, Germany, 2013, pp. 1015–1022.
- [4] Y.-T. Chen, J. Cong, and G. Reinman, "Hc-sim: a fast and exact l1 cache simulator with scratchpad memory co-simulation support," in *Proc. of the 7-th IEEE/ACM/FIP Int. conf. on HW/SW codesign and system synthesis (CODES+ISSS '11)*. USA: ACM, 2011, pp. 295–304.
- [5] J. Edler and M. D. Hill, "Dinero iv trace-driven uniprocessor cache simulator," 2012. [Online]. Available: <http://pages.cs.wisc.edu/~markhill/DineroIV/>
- [6] M. Gusev and G. Armenski, "On-line learning and etesting," in *Information Technology Interfaces, 2002. ITI 2002. Proceedings of the 24th International Conference on*. IEEE, 2002, pp. 147–152.
- [7] —, "e-Testing usage in e-Education," in *e-Learning Paradigms and Applications, Agent-based Approach*, M. Ivanovic and L. Jain, Eds. Springer Verlag, 2014, vol. Studies in Computational Intelligence, vol. 528.
- [8] N. Hendrich, "A java-based framework for simulation and teaching: Hades the hamburg design system," in *Microelectronics Education*, B. Courtois, N. Guillemot, G. Kamarinos, and G. Stehelin, Eds. Springer Netherlands, 2000, pp. 285–288.
- [9] E. Herruzo, J. Benavides, R. Quisilant, E. Zapata, and O. Plata, "Simulating a reconfigurable cache system for teaching purposes," in *Micro-electronic Systems Education (MSE '07), IEEE International Conference on*, 2007, pp. 37–38.
- [10] A. Jaleel, R. S. Cohn, C.-K. Luk, and B. Jacob, "Cmpsim: A pin-based on-the-fly multi-core cache simulator," in *The Fourth Annual Workshop MoBS, co-located with ISCA '08*, 2008.
- [11] I. Kastelan, D. Majstorovic, M. Nikolic, J. Eremic, and M. Katona, "Laboratory exercises for embedded engineering learning platform," in *MIPRO, 2012 Proc. of the 35th Int. Conv.*, 2012, pp. 1113–1117.
- [12] D. Kehagias and M. Grivas, "Software-oriented approaches for teaching computer architecture to computer science students," *Journal of Communication and Computer*, vol. 6, no. 12, pp. 1–9, Dec. 2009.
- [13] J. H. Lee, S. E. Lee, H.-C. Yu, and T. Suh, "Pipelined cpu design with fpga in teaching computer architecture," *Education, IEEE Transactions on*, vol. 55, no. 3, pp. 341–348, 2012.
- [14] X. Liang, "A survey of hands-on assignments and projects in undergraduate computer architecture courses," in *Advances in Computer and Information Sciences and Engineering*, T. Sobh, Ed. Springer Netherlands, 2008, pp. 566–570.
- [15] Moodle, "Learning management system," [retrieved: Nov, 2013]. [Online]. Available: <https://moodle.org/>
- [16] B. Nikolic, Z. Radivojevic, J. Djordjevic, and V. Milutinovic, "A survey and evaluation of simulators suitable for teaching courses in computer architecture and organization," *Education, IEEE Transactions on*, vol. 52, no. 4, pp. 449–458, nov. 2009.
- [17] D. A. Patterson and J. L. Hennessy, "Computer organization and design, fourth edition: The hardware/software interface," MA, USA, 2009.
- [18] D. Patti, A. Spadaccini, M. Palesi, F. Fazzino, and V. Catania, "Supporting undergraduate computer architecture students using a visual mips64 cpu simulator," *Education, IEEE Transactions on*, vol. 55, no. 3, pp. 406–411, aug. 2012.
- [19] J. Qian, R. Wang, S. Shi, Y. Zhu, and Z. Xie, "Simplifying and integrating experiments of hardware curriculums," in *Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on*, vol. 9, 2010, pp. 610–614.
- [20] S. Ristov, B. Atanasovski, M. Gusev, and N. Anchev, "Hands-on exercises to support computer architecture students using educache simulator," in *Computer Science and Information Systems (FedCSIS), 2013 Federated Conference on*, Sep. 2013, p. 769–776.
- [21] S. Ristov, M. Stolikj, and N. Ackovska, "Awakening curiosity - hardware education for computer science students," in *MIPRO, 2011 Proceedings of the 34th International Convention, IEEE Conference Publications*, Opatija, Croatia, may 2011, pp. 1275–1280.
- [22] SimpleScalar LLC, "SimpleScalar tool set," [retrieved: May, 2013]. [Online]. Available: <http://www.simplescalar.com/>
- [23] University of Extremadura, "Smpcache - simulator for cache memory systems on symmetric multiprocessors," [retrieved: May, 2013]. [Online]. Available: <http://arco.unex.es/smpcache/>
- [24] University of Hamburg, "HADES - Hamburg design system," 2002, [retrieved: Aug., 2013]. [Online]. Available: <http://tams-www.informatik.uni-hamburg.de/applets/hades/html/>
- [25] B. Vogel-Heuser, S. Rehberger, T. Frank, and T. Aicher, "Quality despite quantity teaching large heterogenous classes in c programming and fundamentals in computer science," in *Global Engineering Education Conference (EDUCON), 2014 IEEE*, 2014, pp. 367–372.
- [26] X. Wang, "Multi-core system education through a hands-on project on fpgas," in *Frontiers in Education Conference (FIE), 2011*, 2011, pp. F2G–1–F2G–6.