The Visual Simulators For Architecture And Computer Organization Learning

Bosko Nikolić, Nenad Grbanović and Jovan Đorđević

***Abstract*— The paper proposes a method of an effective distance learning of architecture and computer organization. The proposed method is based on a software system that is possible to be applied in any course in this field. Within this system students are enabled to observe simulation of already created computer systems. The system provides creation and simulation of switch systems, too.**

***Index Terms*—visual simulators, architecture and computer organization, laboratory.**

I. INTRODUCTION

RAPID development of digital and computer technologies and systems contributed to their significant usage. Nowadays, significant improvements in any field of modern science cannot be conceived without a large participation of these systems. Different concepts, technologies and complexities of the systems are applied within computer systems. It is necessary to educate experts in several different fields of computer technique, as many as possible, to maintain achieved development. One of the basic fields of this science is computer architecture and organization, which gains a great importance in modern educational and scientific systems.

The way of organizing the course in computer architecture and organization is defined by the organizations IEEE Computer Society and ACM Computer Engineering Task Force [1]. Themes and fields which would be presented to the students are cited within this proposal. They are divided into several major groups: basic principles, processor organization, memory system, input/output subsystems and communication between modules.

Proposal is pointed out that it is very important that students attending these courses have the possibilities of studying and researching the characteristics and the behavior of different appliances, systems and processes. It is required that they design, implement and test hardware and software components, create experiments and examples for analyzing designed systems, and in some cases, have the possibility of altering the design itself according to the results. All these operations are more efficiently conducted as a part of laboratory practices, than as an integrated part of the lectures or as a separate course [2].

Introductory laboratory practice, explained within oral lectures and exercises, should instruct students about the techniques of designing various systems. Purpose of these exercises is demonstration of the characteristics and behavior of the system components, as well as designing and analyzing characteristics of the smaller systems. Laboratory exercises on the middle and finally levels should include more complex techniques and concepts. Within these practices, complex systems, their components and interconnection of those components should be studied. Finally, the students should be able to design and simulate entire computer system which would carry out real tasks.

Lessons in computer architecture and organization from the Faculty of Electrical Engineering, University of Belgrade are described in the paper. These lessons are organized according to afore mentioned principles using visual simulator of digital modules [3]. The paper is organized as follows. Section II provides related work, Section III gives short description of the visual simulator, Section IV describes laboratory, and Section V concludes the paper.

II. THE ANALYSIS OF EXISTING SIMULATORS

In open literature can be found a large number of simulators of computer systems that can be used as support to the courses in computer architecture and organization [4]. By analysis of available projects it can be concluded that the existing simulators are designed for different purposes and various courses. That is why it is hard to subdivide them according to a criterion. One of the more important characteristic is the question whether the simulators contain graphic presentation or not. Furthermore they can be divided according to the way the simulation is performed – a clock, an instruction or a complete program in advance, and according to question whether they are started up interactively or only the batch program is completed. Simulators are also different depending on the fact whether they show implementation details or not. Furthermore, the basic subdivision of already existing simulators can be made according to the mode of implementation of computer systems that are simulated. In that way, simulators can be divided into two groups, those that simulate fixed computer systems and those with configurable computer systems.

The first group of simulators is based upon a concrete computer system, which is realized to a certain level of abstraction. With these simulators, a user is given a possibility of running offered or his own test programs on a fixed computer system. A computer system can be realized on the block system, without implementation details for certain components, and even the most detailed, RTL level. Using these simulators students can comprehend the idea of

Boško Nikolić, Jovan Đorđević, Elektrotehnički fakultet, 3218392, nbosko@etf.bg.ac.rs
Nenad Grbanović, Telekom Srbija, 3200-168, nenadgr@telekom.rs

DOI: 10.2298/JAC0901031N

an entire system and how a similar system reacts on different kinds of problems. Computer architecture and organization depends on the course for which the simulator is intended. To implement a system like this, it is necessary to design the computer system with intended features first. After that, it is necessary to simulate entire design system. Typical representatives of this group of simulators are SimpleCPU, ESCAPE and DLXview.

The second group includes those simulators which enable the user to design planned computer system prior to simulation, and then to run the simulation. It is a custom with these simulators that there exist modules prepared in advance, the possibility of their interconnection and simulation of the system designed in that manner. Also, there are different ways of analyzing obtained results. A realized system is monitored on a block level and the simulation is possible with various time units. During implementation of this kind of simulators, it is necessary, to design all modules, their features and ways of connection first. Then, the defined objects are simulated and the user interface is defined. Typical representatives of this group of simulators are HASE and DINERO-HASE.

On the basis of above mentioned facts the most important criteria on basis of which the analysis of the simulators used within computer architecture and organization courses are distinguished (Table 1)

Based on the analysis, authors come to the conclusion that none of the simulators satisfies the criterion so it can be used in all computer architecture and organization courses. Ideal simulator should satisfy existing criterion in the following manner: 100% coverage of suggested topics, graphic interface simulator, interactive execution, possibility of completing simulation on the program, instructions and clock level, with implementation details.

III. THE BASIC FUNCTIONS OF THE SIMULATOR

At the Faculty of Electrical Engineering, University of Belgrade, it has being attempted to realize an adequate software system which would be compatible with above defined requirements for a long period of time. Basic components of the user interface for designed software systems are: digital structure viewer (middle-right), circuit and module library and navigational system on the base of the module hierarchy (left-middle), navigation according to the digital structure viewer (left-bottom), part for the parameter adjusting and simulator results inspection (bottom- middle), part for viewing simulator messages and digital structure graphic editor (right-bottom), as well as system of textual and graphics menus (up). In addition to the displayed components, unique dialog window is at disposal and it is adjustable by software to the activated dialogue type and to the system of drop-menus for the work with digital structure graphic editor.

The user has at disposal simulated components, which can be positioned to the desired position on the working panel by mouse.

Beside the basic modules which the user can use and interconnect, there are also projected and complete computer systems within the simulator. There are 4 realized systems: the system with the processor type CISC, systems with the RISC processor with or without pipeline processing, and hierarchy memory system. Architecture and organization of the designed systems have been chosen so that they include demonstration of all concepts mentioned in the text above. All systems are displayed on several levels - from the most abstract, block level to the lowest RTL level.

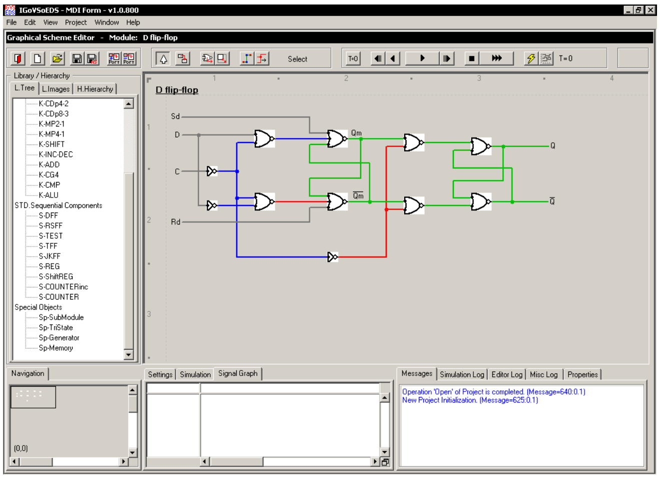

Figure 1. D flip flop

The system enables creation of structures with ten thousand standard circuits and signals, placed on the maximum 2048 screens for viewing the structure parts.

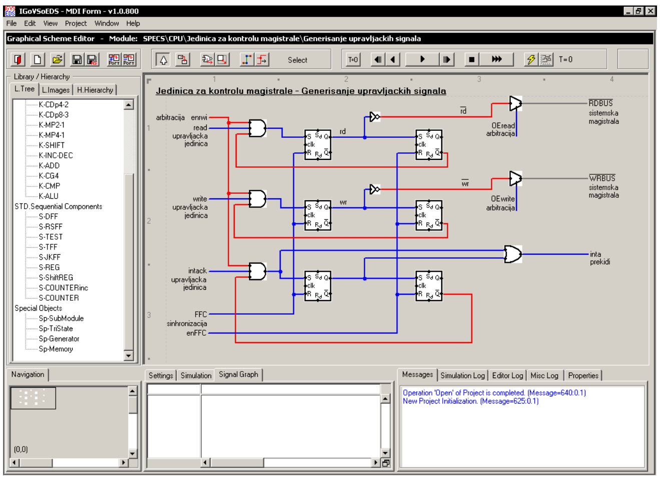

Figure 2 Bus interface

The main module is on the top of the hierarchy tree. It is possible to appoint independent clock signals for the main module and all modules subordinate to the main module. All modules subordinate to the module with an independent clock signal receive the same clock. An informational dialogue in which all clock signals nominated to the modules are presented.

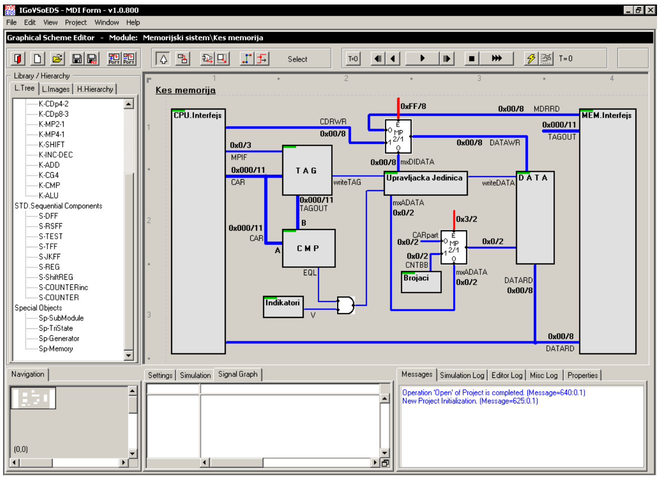

Figure 3 Hierarchy memory system

IV. USING THE SOFTWARE SYSTEM

At the Faculty of Electrical Engineering, University Belgrade students from different years of study attend several courses that belong to the area of computer architecture and organization. The students of the first year study the elementary notions about computer technique, along with the descriptions of simple configurations, that applicable in all areas of computer science. This course covers essentials in binary arithmetic and logical design (design of combinational, sequential and combinational–sequential circuits), various ways of addressing, instruction coding and same basis in architecture. The elementary principles of standard structures of computer systems designing are presented within the following course, such as processor, memory, input/ output subsystem and system bus. Advanced courses include topics that cover architecture and organization of CISC and RISC processors, pipeline processor organization, memory system organization, as well as the ways of connecting the computer system modules through one or more buses.

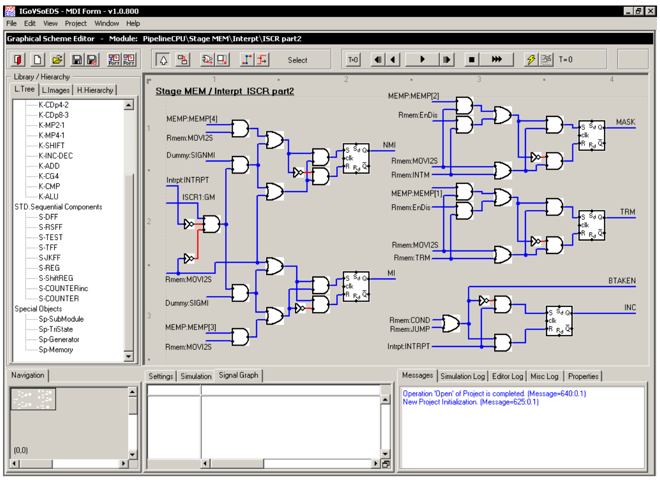

Figure 4 RISC processor with pipeline processing

Beside quoted ones, there are courses in similar areas which help in better understanding of modern computer systems, from the complex architecture and computer organization, microprocessor and multiprocessor systems, computer networks to operating systems and program compilers, distributed systems, data structure, modern programming languages and areas connected with database and information systems.

The purpose of realization of the described software system is its usage within an arbitrary course in computer architecture and organization area. As a practical confirmation of the conclusions made in the previous part of the text can be the fact that the system is used in all courses mentioned.

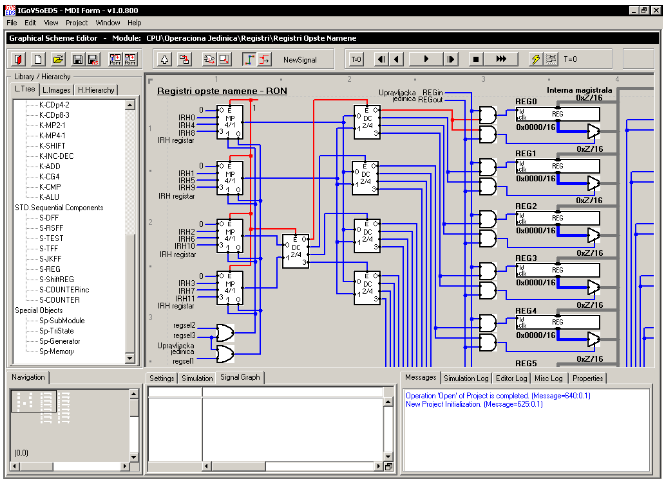

Figure 5 Part of the designed processor- General purpose registers

During the introductory exercises the system covers the actual realizations of the combinational and sequential modules most frequently used (Figure 1). From the group of the combinational modules the following are included: the multiplexer, demultiplexer, decoder, coder, adder, arithmetical-logical unit and comparator, and from the group of sequential modules the flip-flops, register and counter are processed. All mentioned modules are projected and simulated and the serviceability is checked using the system software in the laboratory.

In the following years of studying the students use the modules which represent an implemented computer system. The first system is based on the RISC processor (Figure 2). In advanced courses the students get informed about non-conventional instructions, such as the string instructions, flow control instructions, instructions for working with float point numbers through a number of laboratory exercises. The arbitration for memory access is also considered, in the case that there is more than one module for the memory cycle initiation. Three different cash memories are considered as well as three types of virtual memory with three types of the translation look-aside buffer (TLB) realization for translating virtual into physical address (Figure 3). The system of operating memory that contains 8 memory modules and supports five possible ways of interleaving of the access to the memory modules is considered later. The last topic is the work of processor with pipeline organization (Figure 4).

Students' work results in the final course with the creation of a computer system with defined characteristics. At the beginning of the course every student is given a project that contains description of the computer system architecture (Figure 5). During the course a software system which is supposed to satisfy the required conditions is created.

V. CONCLUSION

Within this paper the structure of computer architecture and organization courses at the Faculty of Electrical engineering University Belgrade is described. The courses are organized in compliance with the references of IEEE and ACM organizations. Every course beside the lectures and computer practices on the board, also encloses the group of laboratory exercises in compliance with the material covered on the lectures. As a support to the laboratory drills the software system which is used in compliance with all mentioned topics is developed.

The software system enables the analysis of existing computer systems by simulated modules. Computer systems based on CISC processor, on RISC processor with and without pipelines, and hierarchy memory system are created. Different concepts applied within these topics are illustrated to the students by the means of posed questions and assignments through prepared tasks are illustrated for the students. The simulator enables the visual display of the system throughout multiple levels, running clock simulation, an instruction or a program in advance, displaying of the time form of the selected signals, and a large number of supporting options for the detailed observation of the computer system performance.

Beside the analysis of the existing systems, there is a possibility of designing different complexity digital systems within the software system. Students have simulated modules at their disposal, which can be connected and so to create a circuit, which can be simulated and in that manner be tested, accuracy of its functions. Depending on the course, the complexity of the realized circuit can be different. At the introductory courses elementary modules are created, like adder, ALU units, registers, counters, … Processor units are created later, so that the complete computer system can be designed during the final course.

REFERENCES

[1] IEEE Computer Society and ACM, "Computer Engineering: Computing Curricula 2001," February 2003.

[2] Clements, A., "The Undergraduate Curriculum in Computer Architecture," IEEE Micro, vol. 20, pp. 13-22.

[3] J. Djordjevic, B. Nikolic, A. Milenkovic, Flexible Web-based Educational System for Teaching Computer Architecture and Organization, IEEE, Transactions on Education, 2005

[4] J. Penfold, J. K. Flanagan, A First Year Computer Organization Course on the Web: Make the Magic Disappear, IEEE Computer Society, Technical Commitee on Computer Architecture Newsletter, September 2000.

Legend: GS-Graphic Interface, SL-Simulation Level, SM- Simulation mode, ID- Implementation Details, %-percentage of the usage of suggested topics in the field of AOR, CL- Clock Level, IL-Instruction Level, PL-Program Level, IC-Interactive execution, BC-Batch execution.

System	GS	SL	SM	ID	%	Web site
SimpleCPU	Da	CL	BC	Da	40	csis.pace.edu/~bergin/iticse99/simplecpu/index.html
ESCAPE	Da	CL	IC	Da	70	www.dec.ctu.edu.vn/cit/tailieu/books/archi_courses/index
DLXview	Da	CL	IC	Da	75	yara.ecn.purdue.edu/~teamaaa/dlxview
HASE	Da	CL	IC	Ne	75	www.dcs.ed.ac.uk/home/hase/
Dinero-HASE	Da	CL	IC	Ne	85	www.dcs.ed.ac.uk/home/hase/
ASF	Ne	PL	BC	Ne	50	www.lri.fr/archi/micro
Easy CPU	Da	CL	IC	Ne	50	www.cteh.ac.il/departments/education/cpu.htm
ANT	Ne	IL	BC	Ne	40	ant.eecs.harvard.edu/guide.shtml
SimpleScalar	Ne	CL	BC	Ne	60	www.simplescalar.com/
RSIM	Ne	CL	BC	Ne	80	rsim.cs.uiuc.edu/rsim/dist.html
SimOS	Ne	PL	BC	Ne	60	simos.stanford.edu/
CacheSim	Ne	PL	BC	Ne	30	www.ece.Koloch.edu/research/labs/reveng/cachesim
LDA	Ne	PL	BC	Ne	70	www.zib.de/schintke/ldasim/index.en.html
Simics	Ne	IL	BC	Ne	75	www.simics.com/
CPU Sim	Ne	CL	IC	Da	50	www.cs.colby.edu/~djskrien/CPUSim
SIMCA	Ne	PL	BC	Ne	60	www-mount.ee.umn.edu/~lilja/SIMCA/index.html
JASPer	Da	CL	IC	Da	60	www.cs.stir.ac.uk/~kjt/software/comms
SMOK	Da	CL	IC	Da	50	www.cs.washington.edu/homes/zahorjan/ homepage/Tools/SMOK
DigLC2	Da	IL	IC	Da	70	www-personal.engin.umich.edu/~postiffm/lc2/lc2.html
JCachesim	Da	IL	BC	Ne	60	www.dii.unisi.it/~giorgi/jcachesim/

Table 1 Critical analysis of existing simulators used in computer architecture and organization courses