2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics

Simulators as a Teaching Aid for Computer Architecture and Organization

Osama Ahmed Siddiqui
Computer Engineering Department
Sir Syed University of Engineering & Technology
Karachi, Pakistan
oahmed@ssuet.edu.pk

Salman Mahmood
Computer Engineering Department
Sir Syed University of Engineering & Technology
Karachi, Pakistan
salmanm@ssuet.edu.pk

Raza Hasan
Computer Engineering Department
Sir Syed University of Engineering & Technology
Karachi, Pakistan
raza_6@hotmail.com

Asim Rasheed Khan
Computer Engineering Department
Sir Syed University of Engineering & Technology
Karachi, Pakistan
arkhan@ssuet.edu.pk

Abstract— **Advancement in computer system hardware makes it difficult to meet the demands of teaching computer architecture and organization. Visualization of different architectures enhances the learning process among students by using simulators. Rather making a comprehensive simulator that can cater the requirement for the course, freely available simulators can serve the purpose as they can save time resource, ease of use, capacity to learn the concepts to fullest, can be accessed anywhere and anytime by the students. Finding the appropriate simulator for the course is a difficult task. This paper discusses about the simulators that can be used for the course *Computer Architecture and Organization* which can help instructors in making students able to grasp the knowledge delivered in lectures.**

Keywords-component; Computer Architecture and Organization; Simulators; Electrical Numerical Integrator and Computer; The Visible Virtual Machine; MARS; LOGISIM; SPIM;

I. Introduction

At Sir Syed University of Engineering & Technology the course Computer Architecture and Organization is offered to Computer Engineering students in their third year. The prerequisite of this course is Logic Design and Switching Theory and Microprocessor Based System Design which is being taught in their second year.

Computer Architecture and Organization is the fundamental course in designing curricula for Computer Engineering. These courses are usually organized in such a way that students gain not only a purely theoretical experience, but also practical understanding of the topics covered in lectures [1].

The course is used to provide students with the knowledge of basic architecture and organization and Basic functions of a computer system and its components and organization and structuring of the major components of a modern day computer. Also Processor logic, Arithmetic Logic Unit, Shifters, Accumulators and Registers design, Hardware implementation of multiplication and division, Floating point representation. Memory systems, Interleaving techniques and Memory organization, Virtual Memory and its Organization, advance technique of Virtual Memory implementation, segmentation and paging techniques, algorithms of page memory, Direct Memory Access, I/O, I/O interfaces and Peripherals are studied.

There was a lack of giving students the exposure of different architectures visualization through graphical representation with the help of simulators. Instead, they were receiving the theoretical aspect of internal structure and components of the computer, how the instructions are being executed and handled by different architectures. This problem can be dealt by the use of simulators [2].

For the batch 2010, currently enrolled in their 5th Semester, the course Computer Architecture and Organization was restructured to two hours of lectures and two hours of laboratory work per week whereas before it was only three hours of lectures per week. This way the gap between theoretical and practical work was removed.

The paper is organized as follows: Section II describes the course overview. Section III describes the criteria on which the simulators were tested. Section IV describes the different simulators that were included in the course. Section V concludes the paper and gives future work that has not been done due to time limitations.

II. Course Overview

After the restructuring of the course the topics were updated and introduction of the laboratory work was included. Below are the details:

A. Theory

The course Computer Architecture and Organization includes two credit hours of theory lectures giving students the theoretical background. The lectures are conducted with the help of multimedia projector making it easy for the instructor

978-0-7695-4721-3/12 $26.00 © 2012 IEEE
DOI 10.1109/IHMSC.2012.33

to demonstrate different computer components according to the architectures. The lectures are conducted using the following course books:

- Computer Organization and Architecture Eighth Edition [3].
- Computer Organization and Design Fourth Edition [4].

B. *Laborotary*

Laboratory consists of two credit hours. The laboratory is equipped with thirty HP Pro 2000 Micro tower computers. The specifications of the computers are as follows:

- Operating System WinXP.
- Intel Core 2 Duo.
- 3.6 GHz.
- 250 GB HDD.
- DVD Combo Rom.

Each batch consists of 250 students, which are divided into five sections having 50 students in each section in order to accommodate the students into the laboratory. Each section is divided into two groups Practical Group I and Practical Group II having 25 students each.

III. CRITERIA FOR SELECTING SIMULATORS

Meetings were held among the course instructors and they came up to the conclusion that the simulator that which will be taught in the course should possess certain properties and should be tested by the instructors before it is included in the laboratory work. The characteristics that a simulator should posses are given below:

- Topics covered in the lectures. (capacity of learning the concepts to fullest)
- Able to run on laboratory computers. (easy installation on laboratory computers)
- Freely available on the internet. (anywhere and anytime accessibility and free of cost)
- User friendly Graphical User Interface. (ease of use)

Several simulators were tested and afterwards it was unanimously decided which simulators should be taught in the course of Computer Architecture and Organization. Also the topics were rearranged so the topics that are to be taught in the lecture room match the laboratory work. Work done in the laboratory should give students the clear idea on the topics covered.

IV. SIMULATORS

Laboratory work consists of 15 Laboratory experiments. Below is the laboratory break down according to the topics covered in response to the simulators.

A. *Electrical Numerical Integrator and Computer*

The first laboratory experiment is to give students the flavor of the world's first electronic digital computer and ancestors of all contemporary modern computers. The simulator uses an enhanced graphical user interface providing the original look [5]. It can be used as Java Applet, Java Web start and executable jar file and is freely available on the internet [6].

This gives students the idea on how the world's first computer works and gives students the idea on how the accumulator registers works using the decimal dials mounted in the transmitter panels. As well as an input using IBM punched card, printer as output device for numerical results and read/write memory cells [6] are shown in figure 1.

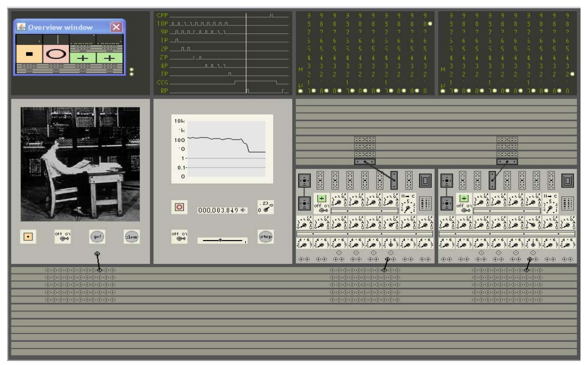

Figure 1. Overview of the ENIAC user interface

B. *The Visible Virtual Machine*

The Visible Virtual Machine is a visual based, virtual computing environment that allows the student user to create a program, and then to actually view the execution of the program within the (virtual) CPU of The Visible Virtual Machine. The system is based on the well-known *Little Man Computer* model of computing [7].

In Visible Virtual Machine the most important aspect of the computing process is maintained, while the needlessly complex details are ignored. For example, both data and instructions are represented and stored as decimal integers rather than as binary strings. Also, the programming language for the hypothetical machine comprises of only the most fundamental operations for computing activity [7].

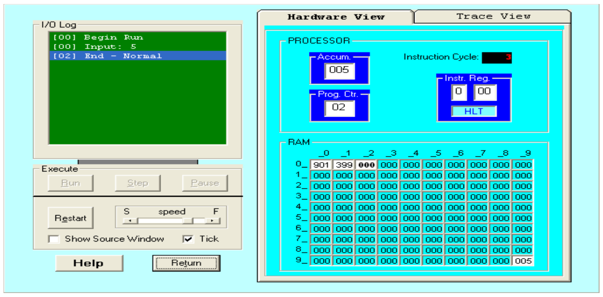

Figure 2. Overview of The Visible Virtual Machine user interface

This gives students the idea about how the von Newmann architecture works and how the data is stored in RAM and how the data is loaded in the different registers such as Accumulator register, Program Counter and Instruction Register. Trace view gives students the working of instruction cycle for the given code.

This application is provided free to faculty and students. The Graphical User Interface is very easy to adopt as well and it serves the purpose for the basic architecture.

In the second Laboratory experiment students learn to explore the simulator. Later, it's Programming and simulate The Visible Virtual Machine program.

In the third Laboratory experiment students learn conditional expressions loops using The Visible Virtual Machine.

In the fourth Laboratory experiment students learn conditional branch instruction The Visible Virtual Machine.

C. *MARS*

The MARS simulator implements the educationally important portions of the Microprocessor without Interlocked Pipeline Stages instruction set utilized by Computer Organization and Design Third Edition [8].

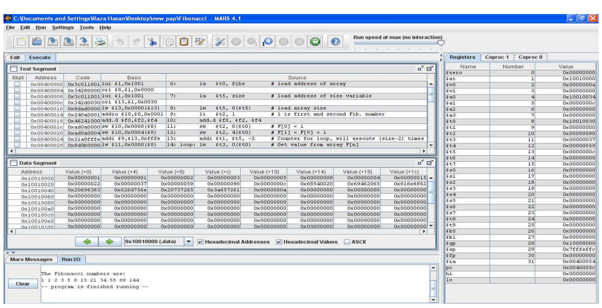

Figure 3. Overview of MARS user interface

MARS is a lightweight interactive development environment for programming a Microprocessor without Interlocked Pipeline Stages (MIPS) assembly language, intended for educational-level use with Patterson and Hennessy's Computer Organization and Design. MARS has been written for the benefit of the computer science educational community, and MARS is open-source software [9]. MARS laboratory outlined was derived from the course ICS233 [12].

In the fifth Laboratory experiment students learn the general structure of MIPS assembly language, writing the assembly language programs in MARS and simulating the program using MARS.

In the sixth Laboratory experiment students learn to implement conditional expressions and loops.

In the seventh Laboratory experiment students learn to use MIPS bit manipulation instructions.

In the eighth Laboratory experiment students are introduced with the implementation of arrays as an abstract data structure in MIPS.

In the ninth Laboratory experiment students learn how to write a procedure in Microprocessor without Interlocked Pipeline Stages and learn how to perform integer multiplication and division operations in MIPS.

In the tenth Laboratory experiment students learn to carry out arithmetic operations using a floating-point representation of real numbers and also learn to use logical operations to mask fields within a word.

In the fourteenth Laboratory experiment students learn the implementation of instructions for a CPU and learn how caches affect program performance.

D. *Logisim*

Logisim is an educational tool for designing and simulating digital logic circuits. It is simple enough to facilitate learning the most basic concepts related to logic circuits. With the capacity to build larger circuits from smaller sub circuits, and to draw bundles of wires with a single mouse drag, Logisim can be used (and is used) to design and simulate entire CPUs for educational purposes [10].

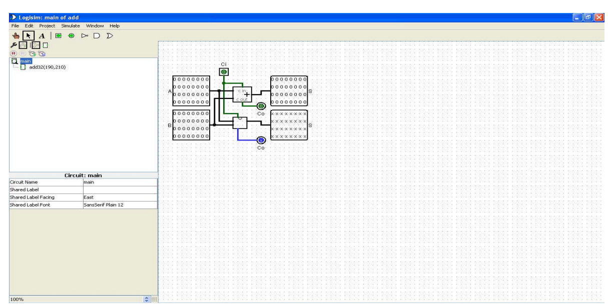

Figure 4. Overvie of LOGISIM user Interface

Logisim is used by students at universities in computer organization courses, to full-semester courses on computer architecture. Logisim is open-source and runs on *any* machine supporting Java 5 or later; special versions are released for MacOS X and Windows. The cross-platform nature is important for students who have a variety of home computer systems [10].

In eleventh Laboratory experiment students learn to implement combinational logic circuits.

In the twelfth Laboratory experiment students learn to implement a simple 16-bit single-stage non-pipeline ALU circuits

In the thirteenth Laboratory experiment students learns to implement a simple 16-bit single-stage Registers File and an Instruction memory.

E. SPIM

SPIM is a self-contained simulator that runs MIPS32 programs. It reads and executes assembly language programs written for this processor. *SPIM* also provides a simple debugger and minimal set of operating system services [11].

The newest version of *SPIM* is called *QtSPIM*, and unlike the other version, it runs on Microsoft Windows, Mac OS X, and Linux. *SPIM* is copyrighted by James Larus and distributed under a BSD license [11].

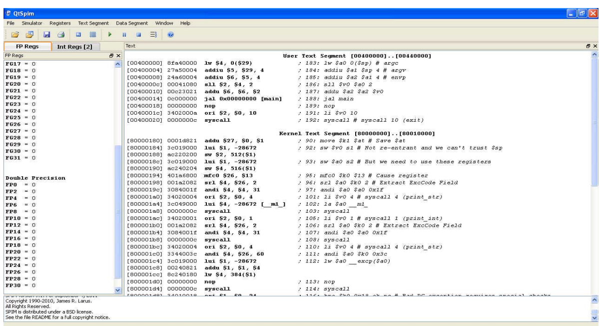

Figure 5. Overview of QTSPIM user interface

In the fifteenth Laboratory experiment students learn how to observe pipelining mechanism.

V. CONCLUSION AND FUTURE WORK

The simulators discussed above will help to enhance the level of understanding of students by virtue of using simulators. Advancement in computer architecture may make it tough for students to understand the latest technology and this can be solved with the help of simulators. To fill this gap different simulators are used and behaviour of the students was positive towards it. The criteria for the selection of simulators is based on simulator which is freely available over the internet and easy to use, capacity to learn the concepts delivered in lectures, anywhere and anytime. The above mentioned simulators serve the purpose of this course.

In future, survey on simulators would be conducted and it can be seen what are the difficulties faced by the students and how students can be benefited from use of simulators.

Later evaluation will be conducted on the knowledge grasped by students from the use of simulators by comparing the past results of students when there was no laboratory work conducted in comparision to the current result of the students and will see if there is any significant difference in the behavior of the students learning process.

ACKNOWLEDGMENTS

The authors acknowledge the encouragement, support and guidance given by **Chairman** *Computer Engineering Department* **Dr. Syed Misbahuddin** of *Sir Syed University of Engineering & Technology* to prepare the Lectures notes Laboratory manual and also giving input in selecting the simulators.

Special thanks to **Mr. Masood Hussain** *Assisstant Professor* in *Computer Engineering Department* of *Sir Syed University of Engineering & Technology* for reviewing of the research paper and giving his valuable suggestions.

REFERENCES

[1] Nikolic, B. Radivojevic, Z. Djordjevic, J. Milutinovic, V. , "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 DOI:10.1109/TE.2008.930097 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4967893&isnumber=5291647.

[2] Cecile Yehezkel, William Yurcik, Murray Pearson, and Dean Armstrong. 2001. Three simulator tools for teaching computer architecture: Little Man computer, and RTLSim. *J. Educ. Resour. Comput.* 1, 4 (December 2001), 60-80. DOI=10.1145/514144.514732 URL:http://doi.acm.org/10.1145/514144.514732.

[3] William, S., "Computer Organization and Architecture Designing for Performance, 8th edition", Pearson press, 2009.

[4] Patterson, D. and Hennessy, J., *Computer Organization and Design: The Hardware/Software Interface*, 3rd edition, San Francisco, CA: Morgan Kaufmann, 2004.

[5] Zoppke, T., Rojas, R.; , "The virtual life of ENIAC: simulating the operation of the first electronic computer," *Annals of the History of Computing, IEEE* , vol.28, no.2, pp. 18- 25, April-June 2006 DOI:10.1109/MAHC.2006.40 URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1631908&isnumber=34215.

[6] Zoppke, T. Electrical Numerical Integrator and Computer Simulator, URL: http://www.zib.de/zuse/Inhalt/Programme/eniac/index.html, retrieved 21 December 2011.

[7] Dr. Westin, S. The Visible Virtual Machine, URL: http://www.cba.uri.edu/vvm/, retrieved 23 December 2011.

[8] Kenneth Vollmar and Pete Sanderson, 2006. MARS: an education-oriented MIPS assembly language simulator. *SIGCSE Bull.* 38, 1 (March 2006), 239-243. DOI=10.1145/1124706.1121415 URL: http://doi.acm.org/10.1145/1124706.1121415

[9] Kenneth Vollmar, MARS, URL: http://courses.missouristate.edu/KenVollmar/MARS/, retrieved 22 December 2011.

[10] Carl Burch, LOGISIM, URL: http://ozark.hendrix.edu/~burch/logisim/index.html, retrieved 21 December 2011.

[11] James Larus, QTSPIM, URL: http://pages.cs.wisc.edu/~larus/spim.html, retrieved 23 December 2011.

[12] IPutu D Raharja, ICS 233 - Computer Architecture and Assembly Language, URL: http://faculty.kfupm.edu.sa/ICS/raharja/teaching.html, retrieved 28 December 2011.