Teaching And Learning Support For Computer Architecture And Organization Courses Design On Computer Engineering and Computer Science For Undergraduate: A Review

Wijaya Kurniawan
Computer Engineering, Faculty of Computer Science
Brawijaya University
Malang, East Java, Indonesia
wjaykurnia@ub.ac.id

Mochammad Hannats Hanafi Ichsan
Computer Engineering, Faculty of Computer Science
Brawijaya University
Malang, East Java, Indonesia
hanas.hanafi@ub.ac.id

***Abstract*—Understanding computer system hardware is basic for the undergraduate student which study in the computer area. System user always needs a faster, cheap and reliable process which is conducted by it computer system self. To fulfill that process is required how a computer works from their design and how they operate. That course is conducted by Computer Architecture and Organization (CAO) course. Two computer undergraduate area which studies CAO is Computer Engineering (CE) and Computer Science (CS). To support great learning at CAO both CE and CS needs to not just study which has a material course but they had to practically study computer architecture and design by the simulator. Why simulator is chosen, because to create computer such as CPU, RAM etc. needs big systems that can manufacture them. Nowadays too much freely available simulator that can be downloaded and used anytime, but not all simulators meet study requirements and didn't meet the CAO Learning Outcome. CS and CE have an international curriculum that given by ACM Computing Curricula. So this paper is discussed what kind of simulator that meets Computing Curricula requirement to implement at learning process which helps the instructor to choose simulator that can be able to grasp knowledge that delivered to the student.**

Keywords—Computer Organization and Architecture; Computer Engineering; Computer Science; Simulator;

I. INTRODUCTION

Computer System plays an integral role in all engineering profession. Users always demanding a fast, powerful but cheap at its cost[1]. Based on users demand, system designer needs to understand processor component (ACM, Computer Engineering Curricula 2016 2015), design, operation and their impact on overall performance of computer systems [1]. Advancement in computer system hardware and Software to satisfy the user, but it will difficult to meet user demands [2].

Nowadays advancement in Faculty of Computer Science, Computer Architecture and Organization (CAO) for an undergraduate student is basic of all departments inside of the faculty. Two of several departments who meet an advance of CAO is Computer Science (CS) and Computer Engineering (CE) [3] [4]. At CAO, Computer Architecture study attributes - attributes of a computer system related to a programmer. For example a set of instructions, arithmetic used, addressing technique, the mechanism of the I/O. Computer Organization while studying part associated with the operational units of computer and the relationship between the components of a computer system, for example, control signals, interfaces, memory technology. From that situation, there is too much difference between CS and CE for their body of knowledge about how they learn about CAO. CE has 60 core hours [3] but CS has 16 core hours to study CAO [4]. So it will give a huge impact to what kind and deep of course that given to the student.

To support great learning achievement, the course needs a simulator to help the student learn about CAO. Some university using a low-cost microcontroller as tools for teaching Computer Architecture and Organization[5]. But it is still lacking for designing computer system competencies because we using an already built system. Some university also using VHDL with the help of an FPGA to design a Multicore CPU[6]. But this approach assumes that the students already have the basic knowledge so it can be done at one semester. So, this approach can be implemented in Master/Magister level but is difficult in Bachelor/Undergraduate level. At the undergraduate level, visualization of different architectures enhances the learning process among students by using simulators [7].

Nowadays, there are many freely available simulators that can save time resource, ease of use, and have the capacity to learn the basic concept of computer architecture. The basic for the students is totally different in CE and CS because different course was given, for example Digital System (DS) course [3] [4]. DS is the basic of CAO and their comprehension is totally different. So in this research would learn about the requirement of students, to review the provided simulator and perform suitable instructional design to be given to the students. So we can choose which simulator that meets the requirement to study CAO both CE and CS.

II. COMPUTER ARCHITECTURE AND ORGANIZATION COURSES

For an undergraduate student, there is too many differences between Computer Engineering (CE) and Computer Science (CS). Computer Architecture and Organization (CAO) all have been proposed at Computing Curricula for CS at 2013 and for

978-1-5386-0549-3/17/$31.00 ©2017 IEEE

CE at 2016. Their learning outcome is different too. Their differences will be explained in this section and course material was given based on the book of William Stalling [8].

A. CAO For CE

Computer Engineering has 60 core hours for Computer Architecture and Organization Course based on Computing Curricula for Computer Engineering 2015 [3], which is:

1) History and Overview (1 hour); this chapter is to introduce the first computer that has been developed and their development until present. From the vacuum tube technology until Ultra Large Scale Integration Computer. Integrated Circuit, embedded system, a microprocessor that has been developing until their clock speed.

2) Relevant tools, standards, and/or engineering constraints (1 hour); this chapter is focused on knowing and measurement factor system attributes. How to do desirable benchmark strategy until Ahmdal and Little Law.

3) Instruction Set Architecture (10 hours); this chapter is to learn about one boundary where the computer designer and programmer can view the machine instruction set. Functional processor requirement to implement processor machine instruction set. The focus is to know how that instruction does. Examining the type of operand and operator that specified by machine instruction set. Two big issues that discussed here, first how operand specified the address, and second how to organize the bits of instruction gained to define operand addresses and that operation.

4) Measuring Performance (3 hours); this chapter is extended from the previous meeting. This chapter is focused in how to benchmark the systems based a common measurement method.. A common processor measurement is a rate at how long the instruction executed expressed as Millions of Instruction Per Second (MIPS). That measurement has proven inadequate to evaluating processors performance based on their architecture.

5) Computer Arithmetic (3 hours); this chapter is started with an overview of Arithmetic and Logic Unit (ALU). This chapter is focused on learning complex ALU aspect and his arithmetic.

6) Processor Organization (10 hours); this chapter introduced and discussed processor organization from the basic organization until high-level organization. Started with summary processor organization, register as a processor internal memory then analyze it. This chapter also discussed about pipelining, the example is pipeline that used in x86 and ARM. It sets the stage for the discussion of RISC and superscalar architecture.

7) Memory System Organization and Architecture (9 hours); this chapter is to learn about internal and external memory. First examining key of computer memories characteristics and a cache memory which is an essential element of all modern computer systems. It also learn about ROM, DRAM, and SRAM memories. Then it is discussing their error control technique that used to enhance memory reliability. External memory is being examined too. Then it discuss about memory devices and systems. The most important device is a magnetic disk, RAID until optical memory.

8) I/O Interfacing and Communications (7 hours); the third key of computer element is I/O modules. Each I/O module interfaces to the system bus, central switch or controls more than one peripheral devices. I/O modules contain logic to performing communication function between peripheral and bus.

9) Peripheral Subsystems (7 hours); a wide variety of peripheral can be conducted by various method operation from the processor to control range devices. This chapter will conduct the peripheral data transfer. Sometimes peripheral data transfer rate is much slower than the memory of a processor. On another hand, it can be faster. The different data format in peripheral that attached to the computer is discussed too.

10) Multicore Architecture (6 hours); this chapter provides an overview multicore systems. First with analyzing hardware performance factor that brings development to a multicore computer rather than the software challenges to improve multicore systems. After learning multicore organization, it discussed how to examine three examples of multicore product from Intel, ARM, and mainframes.

11) Distributed System Architecture (3 hours); last chapter, beginning with an overview of superscalar approaches that contrasting with superpipelining. Second key design issues that associated with superscalar implementation are their architecture. This chapter is discussed about most prominent approaches to the parallel organization about cache coherence, multithreaded processor, and chip multiprocessor. Afer that, the last is describing clusters which consist multiple independent computers that organized.

At Computer Engineering, the Learning Outcomes (LO) [3] of CAO are:

1) Explain the differences between computer architecture and computer organization.

2) Visualize the main components of the computer.

3) Explain the reasons and strategies for each different computer architecture with their strengths and weaknesses.

4) Identify many techniques to achieve high-performance computing such as parallel and distributed architectures.

5) Describe how computer engineering uses or benefits from computer architecture and organization.

B. CAO For CS

Computer Science has 16 core hours for Computer Architecture and Organization Course based on Computing Curricula for Computer Science 2013 [4], which is:

1) Digital Logic and Digital System (3 hours); different from CE, first it is discussed about understanding the basic concept and terminology of number systems. Explaining the technique for converting between digital and binary and explain the rationale that why hexadecimal notation is being used.

2) Machine-Level Representation of Data (3 hours); data machine level that conducted by integer and floating point. This chapter disucussed about how to represent it become crucial design issue and how to treat it by an arithmetic operation that commonly used in the processor.

3) Assembly Level Machine Organization (6 hours); this chapter is discussed about high-level programming languages such as Pascal or ADA, but very little architecture visible at this implementation. The designer point of view see the machine instruction set to provide functional processor requirement with machine instruction set. It also learn about machine language of the registers, memory structure until ALU functioning.

4) Memory System Organization and Architecture (3 hours); this chapter learns about internal and external memory. First, it examines key of computer memories characteristics and a cache memory that is an essential element of all modern computer systems. After that, it is to learn about ROM, DRAM, and SRAM memories. Then it discussed their error control technique that used to enhance memory reliability. The difference from CE is, it is not discussed about RAID technology Optimation, but just to introduce it.

5) Interfacing and Communication (1 hour); this chapter is totally different with CE. This chapter just introduce various method that can be practically implemented. Interfacing and Communication factor such as data transfer rate, how to use system bus, and their impact to the processor.

At Computer Science, the Learning Outcomes (LO) [4] of CAO are:

1) Understand the functionality of each computer system components, characteristics, performances, interactions, and parallelism to enhance the performance.

2) Understand computer architecture for developing a program which can achieve high performance.

3) Understand the trade-off for each computer system components for selecting a system to be used.

III. Course Design

Based on computing curricula that defined Computer Science (CS) and Computer Engineering (CE), its need classified what kind, of course, that will be conducted by the student. Before it, this research is listing about simulation software that meets minimum requirement within the course. There are too many software simulations that can be used, it is not possible to use more than one simulator at the course so we need to choose which one is better between CS and CE.

From Learning Outcome / Learning Objectives (LO) is where we started. We use course standard from the standard given by Blooms Taxonomy (Course Goal, and Program Outcome). This paper is focused on Learning Objective and Instruction only.

Assessment is to measure how much success rate will be get from this process, based on course that delivered by the teacher and learning objective that given by question, problem-solving and evaluation. The instruction is given by presentation from lectures, delivery course, and self-study by the student.

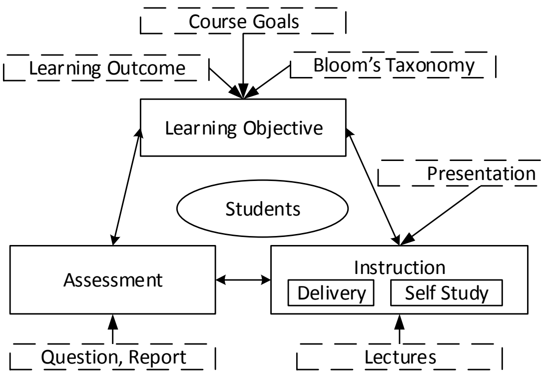

Fig. 1 Learning Method

A. Software Simulator

There are many kinds of simulation software that has a different feature and suitability [2] [7]. The simulators are:

1) Electrical Numerical Integrator and Computer: the simulator for the first computer. Using a decimal based number. The output is a printer and the input is IBM punch card.

2) The Visible Virtual Machine: simulating the fetch-execute cycle of the computer system based on Von Neumann Machine.

3) MARS: a simulator for MIPS assembly language such as how to implement addition and multiplication, for and while loop, etc in MIPS assembly language.

4) Logisim: an educational tool for designing and simulating digital logic circuits. Even though it can only simulate the circuit, it has the capacity to design and simulate an entire CPU circuit.

5) SPIM: this simulator runs and executes a MIPS assembly language in 32-bit processor and provides a debugger and minimal set of operating system services.

6) CPU-OS Simulator: this simulator can simulate the CPU based on RISC type and can also simulate process and memory management at the Operating System layer [9].

B. Simulator Requirement

Simulation software will be chosen based on Computing Curricula CS 2013 and CE 2016. The simulators must: can cover all topics in the courses, able to run on laboratory computers, freely available on the internet and have user-friendly Graphical User Interface (GUI) [7].

1) CS Student: For CS students, we can see from the computing curricula that they only need not the full concept of CAO. So we choose CPU-OS Simulator because this simulator has some features which is enough to:

a) Visualize how assembly language will be converted into machine language.

b) Visualize how data represented and moving between register inside the CPU.

c) Visualize how to make branch and looping in assembly language.

d) Visualize the five state of the process inside Operating System layer.

e) Visualize the memory management done (cache memory for example) in a computer system.

2) CE Student: For CE students, we can see from the computing curricula that they need the full concept of CAO. So we choose both Logisim and CPU-OS Simulator because what can't be done in CPU-OS Simulator can be done in Logisim. Logisim has some features that don't exist within CPU-OS Simulator which is:

a) Designing and building a logic circuit from gates such as AND gate, OR gate, etc.

b) Ability to simulate and visualize digital signal at each cycle of the clock (rising edge and falling edge).

c) Visualize connecting multiple object or component via wiring.

d) Insert and debugging the code in its "true" machine code form.

3) Course Design for CS 2013 and CE 2016: It is must be considered that the simulator must be usable to make the students can easily visualize what they learn inside the classroom.

IV. Result and Discussion

After discussing requirement analysis about differences teaching material between CS and CE, then in this chapter, we will discuss the depth of each chapter that learned and about the detailed characteristic of which simulator that selected. In this chapter will be an analysis of the suitability of the simulator device with each course either from CS or CE. Each simulator has its own advantages and disadvantages. In this chapter, we will analyze the suitability between simulator devices with each course from CS and CE.

Because each simulator has its own advantages and disadvantages, this causes the analysis which is performed will get different results. Here we will present the results of our analysis. This causes the analysis performed will get different results. We will present in the form Pros, Cons, CAO for CE and CAO for CS. Pros are a simulator that has an advantage at CAO teaching and Cons is their disadvantage at teaching material. CAO for CE is teaching material that accommodated by the simulator for Computer Engineering. CAO for CS is teaching material that accommodated by the simulator for Computer Science. Here we will present the results of the analysis that we do.

1) ENIAC: Electrical Numerical Integrator and Computer. This simulator simulating the first computer that developed [10]. Using a decimal based number. The output is a printer. Input is IBM punch card.

a) ENIAC Pros: Have Graphical User Interface

b) ENIAC Cons: Difficult for using and see the result. The user must first learn how to input the command and data because it uses decimal dial, not a keyboard like a modern computer. The user must learn to how to see the result because it uses the panel, not monitor like a modern computer. The block diagram of the computer system is also invisible because it only shows the GUI for the user who looks at the physical appearance of the computer. Can't see the circuit detail of the computer

c) CAO for CE using ENIAC: History and Overview

d) CAO for CS using ENIAC: NA

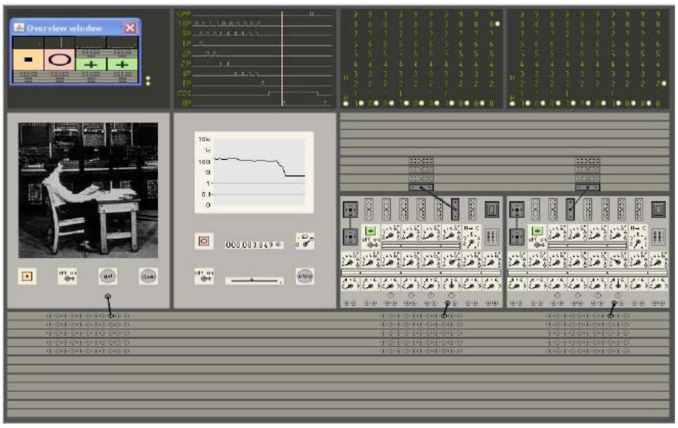

Fig. 2 ENIAC User Interface [10]

2) VVM: The Visible Virtual Machine. This simulator used to simulate the fetch-execute cycle of the computer system based on Von Neumann Machine. This simulator provides an environment that integrated for development the simplest computer architecture [11].

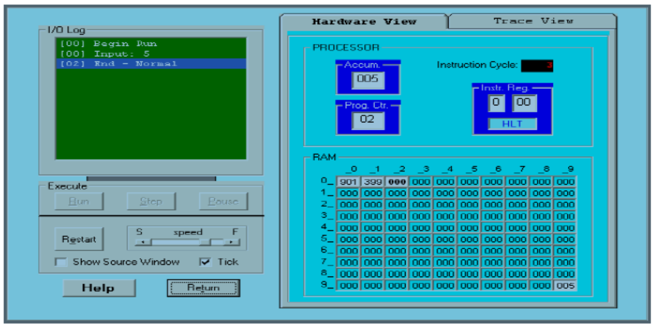

Fig. 3 VVM User Interface [11]

a) VVM Pros: Have GUI. Can see the content of the Accumulator Registers, Program Counter Register, RAM (address & data). The user can manually input the program. Can see the flow of data inside the Von Neumann Machine Architecture

b) VVM Cons: Using machine language, a user who only studies the HLL language must first learn assembly. Can't see the circuit detail of the computer, can only see the big picture of its architecture.

c) CAO for CE using VVM: History and Overview

d) CAO for CS using VVM: NA

3) MARS: a simulator for MIPS assembly language such as how to implement addition and multiplication, for and while loop, etc in MIPS assembly language [12].

a) MARS Pros: Have GUI. Good for learning machine language (Instruction Set Architecture) and how to implement it (for loop, while loop, etc). Can see how a data value is represented by hexadecimal. Can see the content of the general purpose registers

b) MARS Cons: Using machine language, a user who only studies the HLL language must first learn assembly. Not showing the flow of data and the architecture of the machine. Can't see the circuit detail of the computer

c) CAO for CE using MARS: Instruction Set Architecture

d) CAO for CS using MARS: Machine-Level Representation of Data, Assembly Level Machine Organization

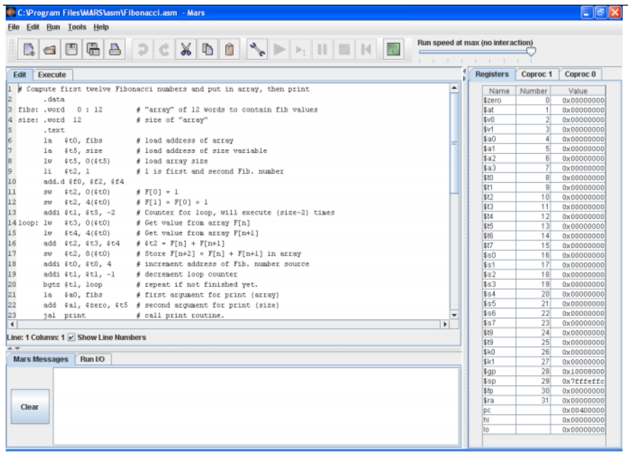

Fig. 4 MARS User Interface [12]

4) Logisim: an educational tool for designing and simulating digital logic circuits [13]. Even though it can only simulate the circuit, it has the capacity to design and simulate an entire CPU circuit.

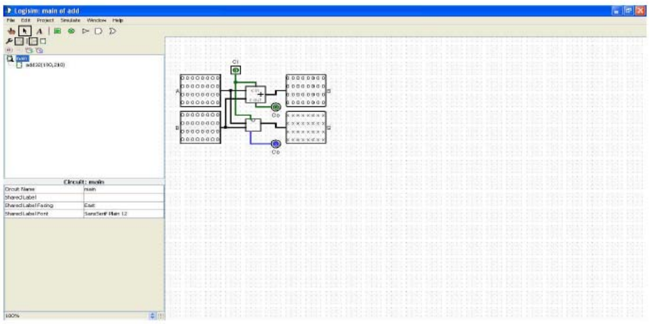

Fig. 5 Logisim User Interface [13]

a) Logisim Pros: Have GUI. Can make, see, and modify the digital logic and is capable of making and simulating your own CPU's circuit

b) Logisim Cons: Can't use and see general instruction set architecture because it can only use binary (0 & 1) for its input and output because this simulator mainly purpose is for simulating digital logic circuit

c) CAO for CE using Logisim: Measuring Performance, Computer Arithmetic, Processor Organization

d) CAO for CS using Logisim: Digital Logic and Digital System, Machine-Level Representation of Data

5) SPIM: this simulator runs and executes a MIPS assembly language in 32-bit processor and provides a debugger and minimal set of operating system services [14].

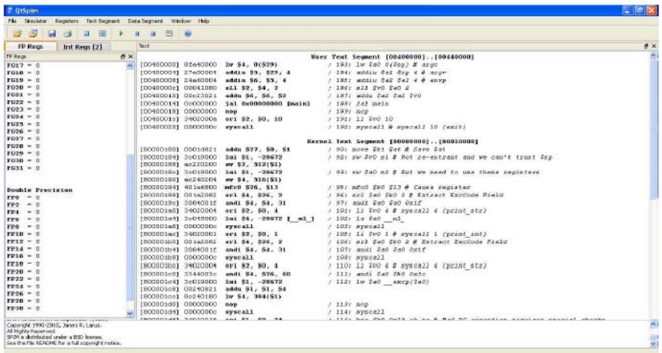

Fig. 6 SPIM User Interface [14]

a) SPIM Pros: Have GUI. Good for learning machine language (Instruction Set Architecture) and how to implement it (for loop, while loop, etc). Can see how a data value is represented by hexadecimal. Can see the content of the floating point (FP) and integer (INT) registers

b) SPIM Cons: Using machine language, a user who only studies the HLL language must first learn assembly. Not showing the flow of data and the architecture of the machine. Can't see the circuit detail of the computer

c) CAO for CE using SPIM: Instruction Set Architecture

d) CAO for CS using SPIM: Machine-Level Representation of Data, Assembly Level Machine Organization

6) CPU-OS Simulator: this simulator can simulate the CPU based on RISC type and can also simulate process and memory management at the Operating System layer [9].

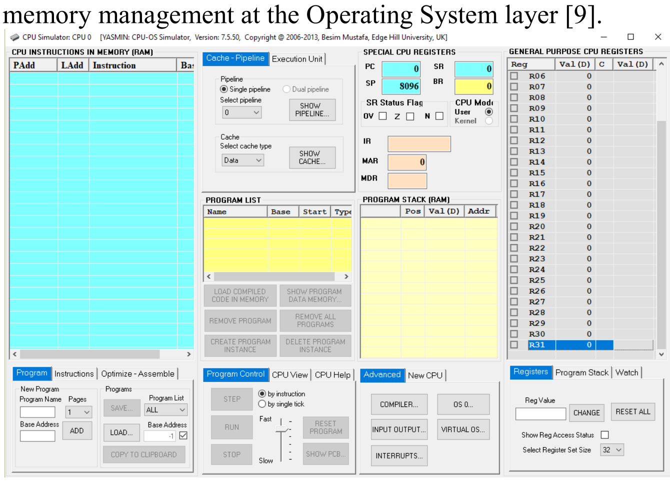

Fig. 7 CPU-OS User Interface [9]

a) CPU-OS Pros: Have GUI. Good for learning machine language (Instruction Set Architecture) and how to implement it (for loop, while loop, etc). Can see how a data value is represented by hexadecimal. Have a compiler feature so the user can see how HLL be compiled into machine language. Have an interrupt feature. Can see the memory system. Have simulator for operating system (OS)

b) CPU-OS Cons: Not showing the flow of data and the architecture of the machine. Can't see the circuit detail of the computer

c) CAO for CE using CPU-OS: Instruction Set Architecture, Measuring Performance, Computer Arithmetic, Processor Organization, Memory System Organization and Architecture

d) CAO for CS using CPU-OS: Machine-Level Representation of Data, Assembly Level Machine Organization, Memory System Organization and Architecture

V. Conclusion And Future Work

Learning process not only to talk and discuss. The development for the teaching material is also critical to conduct an educating CAO. The learning process between CE and CS is totally different, their LO about CAO is different too. Therefore in this paper, we presented the important thing about how using the simulator to support the learning process. The criteria of the selection simulators based on a simulator that can accommodate teaching materials and have the capacity to deliver lectures anywhere and anytime. So we propose when and what topics at CAO and what simulators are used which is:

1) For CS students: Machine-Level Representation of Data (CPU-OS Simulator); Assembly Level Machine Organization (CPU-OS Simulator); Memory System Organization and Architecture (CPU-OS Simulator)

2) For CE students: Computer Arithmetic (Logisim); Processor Organization (Logisim); Memory System Organization and Architecture (CPU-OS Simulator and Logisim); I/O Interfacing and Communication (CPU-OS Simulator and Logisim)

In the future, we hope we can implement the simulator as a tool to aid teaching CAO to both CS and CE students. Then we will analyze the benefit of the simulator using some pre and post-test before and after using the simulator. Also, we want to see are these simulators are suitable or we should change or add other simulator software to meet the demand of Learning Outcome for CAO body knowledge in CS and CE students.

Authors and Affiliations Acknowledgment

Thanks to Computer Engineering and Computer Science Departments, Faculty of Computer Science, Brawijaya University. This paper is proposed to deliver Computer Architecture and Organization Courses. Computer Architecture and Organization courses have been delivered in Computer Engineering and Computer Science Departments.

References

[1] A. S. Nayak and M. Vijayalakshmi, "Teaching Computer System Design and Architecture Course – An Experience.," in *Innovation and Technology in Education (MITE)*, IEEE International Conference in MOOC, 2013.

[2] R. Hasan and S. Mahmood, "Survey and Evaluation of Simulators Suitable for Teaching for Computer Architecture and Organization," in *Proceedings of 2012 UKACC International Conference on Control*, 2012.

[3] ACM, Association for Computing Machinery, "Computer Engineering Curricula 2016," IEEE, 2015.

[4] ACM, Association for Computing Machinery, "Computer Science Curricula 2013," IEEE Society, 2013.

[5] D. C. T. Lo, K. Qian and L. Hong, "The Use of Low-Cost Portable Microcontroller in Teaching Undergraduate Computer Architecture and Organization," in *IEEE Second Integrated STEM Education Conference*, 2012.

[6] M. Reichenbach, B. Pfundt and D. Fey, "Designing and Manufacturing of Real Embedded Multi-Core CPUs: A Holistic Teaching Approach in Computer Architecture," in *10th European Workshop on Microelectronics Education (EWME)*, 2014.

[7] O. A. Siddiqui, R. Hasan, S. Mahmood and A. R. Khan, "Simulators as A Teaching Aid for Computer Architecture and Organization," in *4th International Conference on Intelligent Human-Machine System and Cybernetics*, 2012.

[8] W. Stallings, Computer Organization and Architecture (Designing for Performance), Ninth Edition, Pearson Education, Inc., 2013.

[9] B. Mustafa, "Modern Computer Architecture Teaching and Learning Support: An Experience in Evaluation," in *International Conference on Information Society (i-Society)*, 2011.

[10] T. Zoppke, R. Rojas, "The virtual life of ENIAC: simulating the operation of the first electronic computer," *EEE Annals of the History of Computing,* vol. 28, no. 2, pp. 18-25, April-June 2006.

[11] C. Bruce, G. Mohay, G. Smith, I. Stoodley and R. Tweedale, Transforming IT Education: Promoting a Culture of Excellence, US: Informing Science Press, 2006.

[12] K. Vollmar and P. Sanderson, "MARS: An Education-Oriented MIPS Assembly Language Simulator," in *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, Houston, Texas, USA, 2006.

[13] Timothy Stanley, Vasu Chetty, Matthew Styles, Shin-Young Jung, Fabricio Duarte, Tin-Wai Joseph Lee, Michael Gunter, and Leslie Fife, "Teaching computer architecture through simulation: (a brief evaluation of CPU simulators)," *Journal of Computing Sciences in Colleges,* vol. 27, no. 4, pp. 37-44, 2012.

[14] Kenneth Vollmar and Pete Sanderson, "A MIPS assembly language simulator designed for education," *Journal of Computing Sciences in Colleges,* vol. 21, no. 1, pp. 95-101, 2005.