Bulletin of Electrical Engineering and Informatics
Vol. 9, No. 2, April 2020, pp. 747~754
ISSN: 2302-9285, DOI: 10.11591/eei.v9i2.1972

CPU implementation using only logisim simulator to achieve computer architecture learning outcome

Mochammad Hannats Hanafi Ichsan, Wijaya Kurniawan
Computer Engineering, Faculty of Computer Science, Brawijaya University 8th Veteran Road, Indonesia

Article Info

Article history:

Received Aug 10, 2019
Revised Oct 19, 2019
Accepted Dec 10, 2019

Keywords:

8-bit CPU
Computer architecture and organization (COA)
Computer engineering
Learning outcome
Logisim

ABSTRACT

This paper proposes the design, implementation, and evaluation of an 8-bit CPU architecture in computer organization and architecture (COA) course for Undergraduate Computer Engineering using only Logisim simulator. The main advantage of using one simulator is the elimination of inefficiency so that students can concentrate more on the course's content without spending their time just to learn how to use each different type of simulators. To achieve that, a simple CPU architecture is pre-designed and be implemented in Logisim simulator. According to previous researches, Mic-1 CPU architecture is chosen because it is the simplest one and can be easily built using many simple logic gates already existed in the Logisim. To evaluate the desired outcome, students are separated into two different groups. Each group used different type of learning media and material and then their examination scores and satisfaction are compared to each other. Every student belongs to the group that only used Logisim simulator obtained higher score and more than 50% are satisfied with the new learning process and material. Thus, it envisaged that this method will made the delivering learning process in COA course be more efficient than what has happened so far.

Corresponding Author:

Wijaya Kurniawan,
Computer Engineering, Faculty of Computer Science,
Brawijaya University,
8th Veteran Road, Malang, Indonesia.
Email: wjaykurnia@ub.ac.id

1. INTRODUCTION

One course that becomes a basic knowledge in the field of computation education is Computer Organization and Architecture (COA) course. This course is the basis of the study programme in Computer Science (CS) and Computer Engineering (CE) [1]. Association for Computing Machinery (ACM) in an association that defines the Learning Outcome (LO) for the course. On CE, much more core hours is needed in COA course to achieve the LO [2]. Learning media which is selected must have certain capabilities that ensure the LO can be achieved [3-5]. The media in COA course are mostly in the type of simulator software [6]. All this time, because of each simulator uniqueness, many types of simulators need to be used in the delivering learning process of COA's course, particularly in Computer Engineering Programme [7, 8]. The first study that has been done is to determine the most suitable simulator software that has many features needed to deliver most of the topics covered in this COA course [9]. There existed many software that can be used to describe how CPU works. The previous research showed that Logisim is a simple simulator that is easy to use because it is based on Graphical User Interface (GUI) to create a digital logic circuit [10-12].

Usually, there are many chapters in many COA's handbook [13]. There is also exists many simple architecture of of an 8-bit CPU architectures such as Mic-1, Mic-2, [14-17], SAP-1, SAP-2, etc [18-20]. They are based on the architecture of Von Neumann machine [21]. The election for the best and simplest architecture to be implemented in the simulators for learning process have also been conducted in the previous research which results in the choice of Mic-1 CPU architecture [22]. This research is a continuation from two previous researches. It focused on how to evaluate the Computer Organization and Architecture's (COA) Learning Outcome (LO) achievement by the students implementing the Mic-1 CPU Architecture using Logisim simulator.

2. LITERATURE REVIEW

Computer organization and architecture involves many concepts where the books used for the learning process are related to the number of theories and little to practice [23]. Currently, most undergraduate students, who take the Computer Engineering undergraduate programme, have already made a contact with computers as a user [24]. More over, there is a very large gap to be filled to teach the organization and architecture of computers adequately [25]. This gap is caused by the increasing number of abstraction layers between real hardware and end-user-oriented applications. The old approach, to introduce studies on computer organization and architecture, is proposed by Patterson and Hennessy [26]. It reviewed the hardware section to understand computer organization and architecture, and used assembly language and scheduling [14], while the Learning Outcome for Computer Engineering programme demands how the undergraduate student can design the CPU so that the computer can run in accordance with its concept of architecture and organization.

The needs of learning outcomes for computer engineering undergraduate students are about four levels of the CPU area: processor abstraction, CPU organization, microcode, and its implementation [27]. Some free softwares are recommended as another alternative to improve the quality of the teaching. Several studies have been conducted to select software that is compatible with COA's Learning Outcomes, but different chapters need a different software for simulation. Thus, the learning process during the current semester will run out just to recognize many different simulators. Some usually used simulators are: ENIAC (Electrical Numerical Integrator and Computer) [28], VVM (Visible Virtual Machine) [29], MARS [30], Logisim [10], SPIM [31], CPU-OS simulator [32], DRAM SIM 2 [33], gem5-gpu [34] etc. Therefore, a simulator with a specific design is needed in accordance with the Learning Outcome so that students are more focused on the content of the learning material.

3. RESEARCH METHOD

This section will explain the methods of this conducted research. Firstly, Mic-1 CPU Architecture is pre-designed and implemented in Logisim simulator to reach the Learning Outcome. Secondly, instruction and assesment design processes based on Bloom Taxonomy is examined. Division of learning participants in classes into two clusters which used different learning materials is discussed too.

3.1. Mic-1 CPU architecture

Based on Von Neumann architectures, the inside of the computer consists of three main parts which is Central Processing Unit (CPU), Main Memory, and Input/Output (I/O) [35, 36]. CPU can be said as the brain for the computer. All of calculation to process data and also all of the data flow that happens inside of computer is controlled by this CPU. In the inside of this CPU, there exists Arithmetic Logic Unit (ALU), Control Unit (CU), general purpose registers, specific purpose registers, and datapath that connects all of those things. In this research, the CPU to be implemented in simulator using Logisim is designed in the following order: 1. ALU, 2. general and specific purpose register, 3. Datapath design, and 4 CU. In this research, the designed CPU which is called Mic-1 will be chosen. It is chosen because of its simplicity and convenience. So, it is also expected that this design will make students able to understand and implement it easily.

To design the ALU, some specification, about what type of calculation that our CPU can do, has to be defined. In this research, we will use the simplest things of calculation's type which is increment, adder, AND logic operation, and OR logic operation. Beside of those calculation, we will also need to implement a data forwarding in this ALU. It is done because in the Mic-1 architecture, the ALU become the central part in the CPU, which means that all data flow will be going through this part. In specific, we will use two 8 bits of data that will go through one multiplexer which has three selector bits to choose one from eight combination of data forwarding and calculation in the following order: 1st data forwarding, 2nd data forwarding, 1st data increment, 2nd data increment, 1st and 2nd data addition, 1st and 2nd data subtraction,

1st and 2nd data AND Logic Operation, and lastly, 1st and 2nd data OR Logic Operation. This design can be seen in the Figure 1.

The second thing to be designed is about general purpose register and specific purpose register. In this CPU architecture, we will only use two general purpose register which is A register and B register. These two register is used to store two 8 bits of data that become the input to the ALU as shown in Figure 1. A register is also functioned as an accumulator, the register that is used to store temporary results from the calculation made by ALU. Specific purpose register is also designed to meet the operating requirement for the CPU. Program Counter (PC) is a specific purpose register that functions as a reference for the sequence of instruction that wants to be done. Memory Buffer Register (MBR) is a specific purpose register that functions as a gate and to store the instruction or data that comes from the main memory unit going to inside of the CPU. Register Out is a specific purpose register that functions as a gate and to forward the final result of calculations to the monitor or to HEX display so it can be seen by the users. The last two specific purpose register is the micro-instruction register and micro-program-counter that functions to break the user's instruction from main memory to become micro-instructions that is understandable by computer.

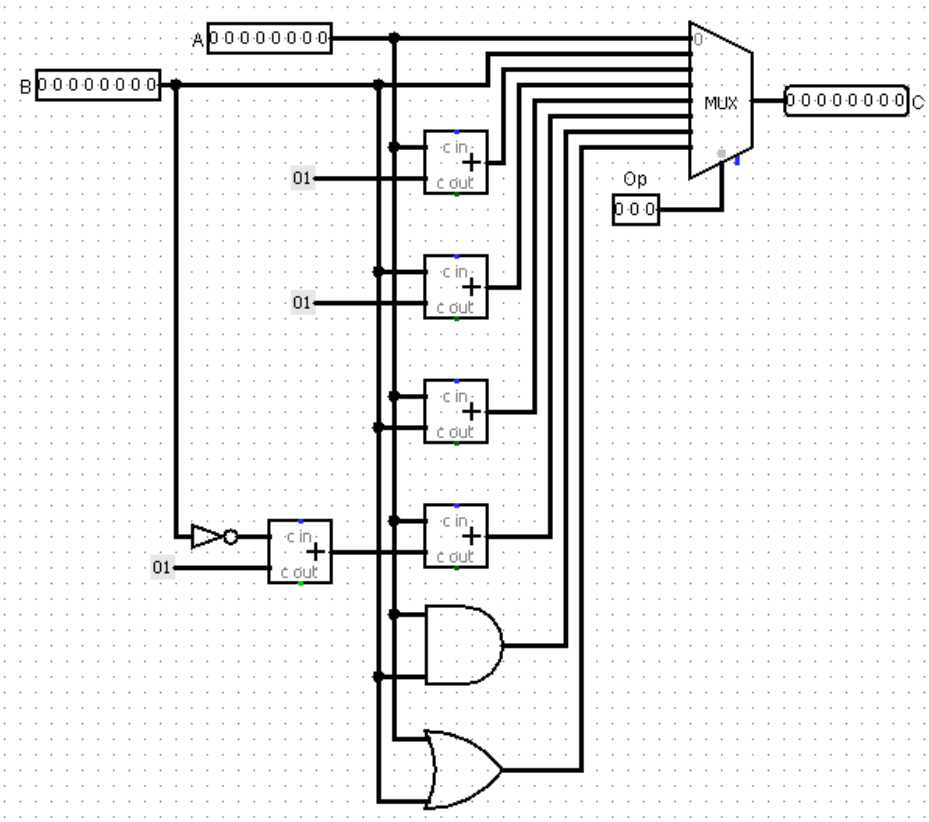

Figure 1. ALU design using logisim

The third thing to be designed is about the numbering of control lines and the circuit diagram that is used to connect ALU, general purpose register, specific purpose register, and CU. In this architecture, we declare 16 control lines as follows: 5 bits for next address, 1 bit for jump instruction's condition, 3 bits for ALU selector operations, 5 bits for write lines to general purpose and specific registers, and lastly 2 bits for multiplexer that forwards B register to ALU. The ALU is placed as the central part of the circuit.

The fourth and the last thing to be designed is the Control Unit (CU). This CU consist of binary number written as a hexadecimal number that represents the condition of zero (0) and one (1) in each control lines described above [37]. As it name suggests, these zero-one condition indicating the active and non-active condition for the register and also for the operations that will be made in the ALU and B register multiplexer. By arranging these conditions according to some instructions that is made by the CPU designer, we can make the circuit operates to process the data from the main memory to produce some expected results from the desired user's instructions. The hexadecimal numbers are stored to Read Only Memory (ROM) as shown in the Figure 2. Finally, the 8-bit CPU circuit implemented in Logisim such as in the Figure 3.

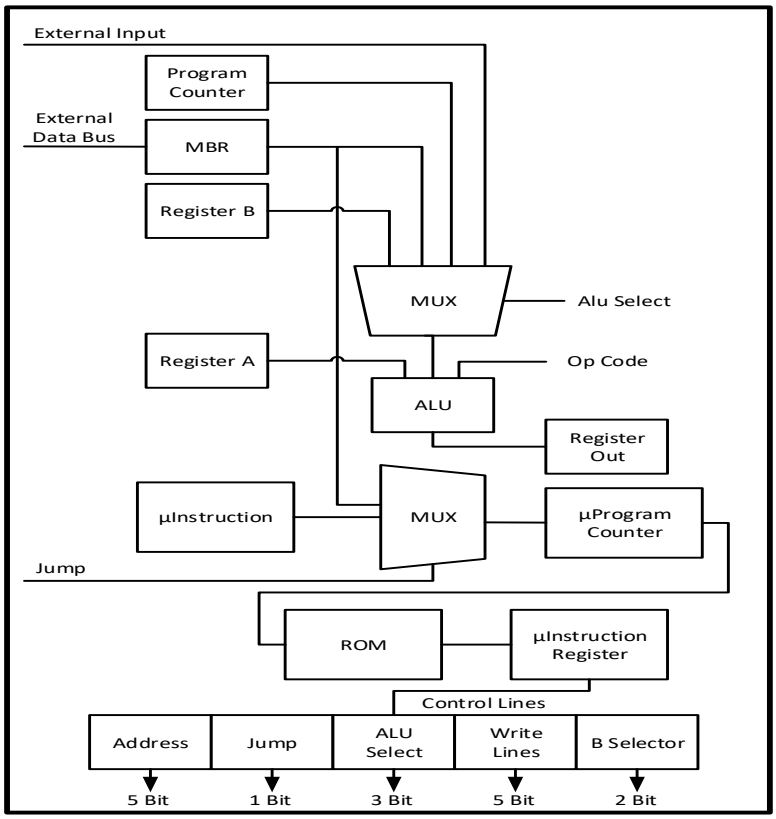

Figure 2. 8-bit CPU Mic-1 architecture

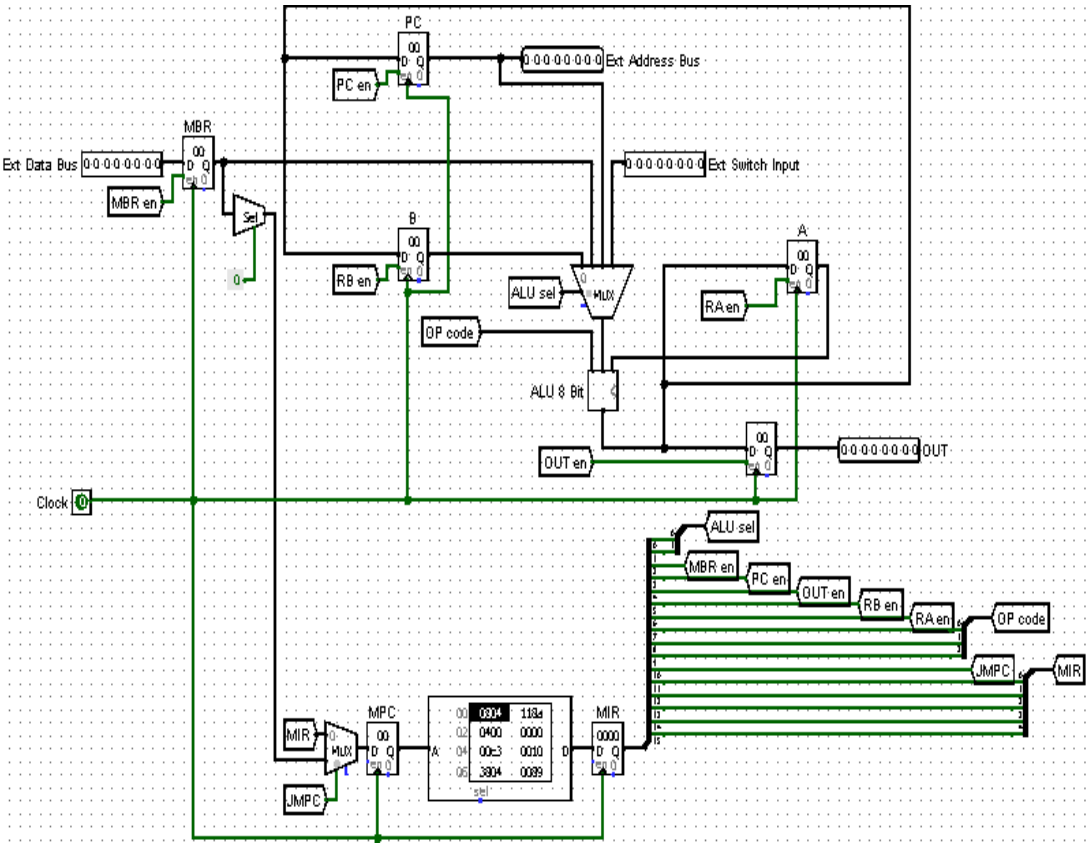

Figure 3. Implemented CPU 8-bit in Logisim

3.2. Instruction and assessment processes in bloom taxonomy

Based on the theory of a bloom taxonomy, an algorithm to improve the quality of the students is shown as in Figure 4. There are three areas which is 1. Learning Objective (LO), 2. Instruction, and 3. Assessment [38-40]. This research is focused on the Instruction and Assessment because LO is already referred in the ACM Computing Curricula Computer Engineering. This chapter presents Instruction and Assessment process to evaluate the LO achievement by undergraduate CE students in COA course which used an 8-bit CPU simulation architecture, Mic-1, using Logisim as its learning support tool.

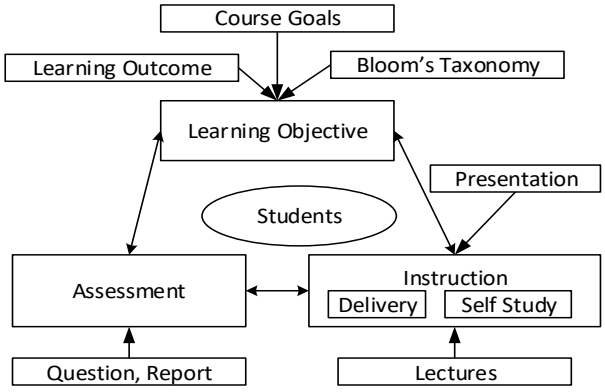

Figure 4. Learning objective algorithm

Instruction has been made on the design of the Mic-1 simulated electronic circuit on Logisim. The simulation adapted to the LO is showed in the previous studies. Instruction process is conducted face to face in the classroom by Delivery and Self Study. Later, the result will be carried out directly against the student by comparison between two different groups using different learning material. There will be an assessment report and questionnaire which is used to assess the success rate and satisfaction associated to the media teaching/created simulator.

The material is presented during the 60 core hours course. There are two kinds of students cluster that are divided into 4 classes. The first cluster is divided again into two classes of 40 students each with a total of 80 students. The first cluster is using materials with general teaching materials (books, Slideshow, and e-learning). The second cluster is also divided into two classes, each containing 40 students with a total of 80 students who use the Logisim simulator to implement an 8-bit CPU architecture. So the total tests are conducted on 160 students. This cluster differentiation is made to find out about the success rate of the simulator and simulation usage compared with the general teaching materials. Examination on the entire cluster of students is using multiple choice questions as many as 100 questions which is done in 2 hours. The material contains a variety of topics that reflect the overall material.

On the side of the level of student satisfaction, the satisfaction rate questionnaire is granted to students who have used the Logisim learning module. Assessment was conducted on several aspects of 20 questions divided into five topics, include: the relevance of the material, features in Logisim, the suitability of the material, clarity of information and terminology, and understanding of the material.

4. RESULTS AND DISCUSSION

This section will explain how the mechanism of the test and the results are obtained. In the first test, all students either receive the material using common CPU Modules or Logisim simulator will get the same exam. The exam contains definitions, problems related to the analysis of the material in accordance with some points in the COA's LO. In this case, the student score is divided into eight categories. Table 1 shows how the categories are made. The results from the final exam from both cluster is shown in Figures 5 and 6, respectively.

Because there are more students that achieved "A" score in cluster-2, it can be concluded that the students in cluster-2 has achieved better performance compared to other students in cluster-1. To get more insight, the results of both cluster is then analysed based on the level of minimum, maximum, average, and the standard deviation of the test's score that can be seen in Table 2.

Table 1. Scoring categoris

Symbol	Quantity
A	>80 points
B+	75-80 points
B	69-75 points
C+	60-69 points
C	55-60 points
D+	50-55 points
D	44-50 points
E	<44 points

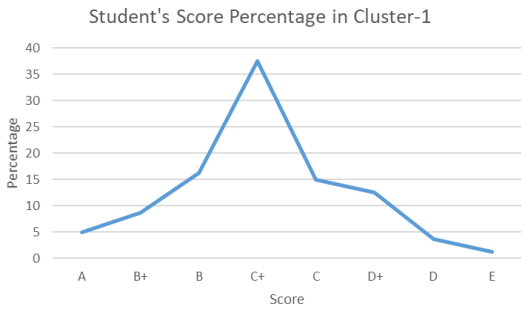

Figure 5. Student's score in cluster-1

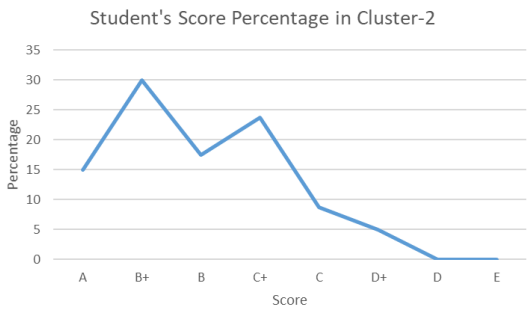

Figure 6. Student's score in cluster-2

Table 2. First testing result

	Common Modules	Logisim CPU Simulator
Min	42	52
Max	90	98
Average	65.175	73.2625
stddev	10.88231	10.6221

From the first experiment results, it showed that the minimum and the maximum of examination results is increased by ten. It is true that not all of them has those same results, but from the three values: minimum, maximum, and average, it can be concluded that most of them have been experiencing some enhancement. It also can be seen that the standard deviation is not changing very much. Thus, it can be said that the performance for every individual has been increased with the new learning modules.

The second test is performed only to the students on Cluster 2 who received the material by use of Logisim to implement an 8-bit CPU architecture. The tests are carried out using a questionnaire with four (4) criteria: less good, fairly good, good, and excellent. The results of a second test can be seen in Table 3.

Table 3. Second testing result

No.	Criteria	not good	pretty good	well	very good
1	The relevance of the material with Learning Objective	1%	8%	57%	34%
2	CPU Features On Logisim	2%	14%	52%	33%
3	The suitability of the material to the evaluation	1%	13%	59%	27%
4	Clarity of information and terms	0%	14%	65%	21%
5	An understanding of the material	1%	11%	65%	23%
	Average percentage	1%	12%	60%	28%

From the second experiment results, specific measurement is being taken according to the usage of this new learning module related to the learning process in class. More than 50% from those 80 students argued that they are more satisfied from many objects significant to the learning process: relevance with learning objectives, features in Logisim, material suitability, information clarity, and understanding of the material. Connecting with the result from the first experiment, this new learning module is already satisfied to fulfil the objective to improve the learning objective achievement in COA course. While some previous researches showed that many different simulators are needed in COA teaching activities [6, 7], these experiments results demonstrated that by pre-designing and implementing Mic-1 CPU architecture in Logisim simulator, it can be used as a single tools to enhance and obtain the desired Learning Outcomes for undergraduate students in Computer Engineering Programme.

5. CONCLUSION

In the previous research, a Mic-1 CPU architecture using Logisim simulator have been made. In this research, some undergraduate students in Computer Engineering department have already use that architecture as a foundation to design and implement their own CPU circuit using Logisim simulator. On the test that has been done, it is shown that those students have a better understanding about Instruction Set Architecture, Measuring Performance, Computer Arithmetic, Processor Organization, Memory System Organization and Architecture, I/O Interfacing and Communications, and Peripheral Subsystems. The abstract nature and complexity of the CPU circuit were also solved by hands-on and making a direct observation using Logisim simulator.

In the future, it is planned to implement this new learning module by embedding the Logisim circuit to Field Programmable Gate Array (FPGA) by VHDL language. Logisim is very satisfying to help students imagining about how bits of the data and instruction flow from one part to other parts in the computer system. But because it is still only a simulator, a few steps need to be taken to bring this logic circuit design to be realized in a hardware. It is difficult to require that every education institution who teaches COA course must have equipment to making an Integrated Circuit (IC) or processor. So the realizable plan is to implement the design in an FPGA. This hardware has the ability of flexible programmable logic gate which is enough to satisfy the needs to implement logic circuit design of the CPU.

REFERENCES

[1] Association for Computing Machinery (ACM), "Computer Science Computing Curricula," IEEE Computer Society, 2016.

[2] Association for Computing Machinery (ACM), "Computer Engineering Curricula 2016," IEEE Computer Society, 2015.

[3] E. A. Davis and J. S. Krajcik, "Designing Educative Curriculum Materials to Promote Teacher Learning," *Sage Journal*, vol. 34, no 3, pp. 3-14, 2005.

[4] A. Setiawan, A. Handojo and R. Hadi, "Indonesian Culture Learning Application based on Android," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 1, pp. 526-535, 2017.

[5] M. Ally, "Mobile Learning: Transforming the Delivery of Education and Training," Edmonton: AU Press, 2009.

[6] P. W. C. Prasad, A. Alsadoon, A. Beg and A. Chan, "Using simulators for teaching computer organization and architecture," *Journal of Computer Application in Engineering Education*, vol. 24, no. 2, pp. 215-224, 2016.

[7] Shine, V.J., &. Sathish, P.K., "Teaching Computer Architecture Using Simulation Tools," *International Journal of Computer Science and Information Technologies*, vol. 5, no. 2, pp. 1411-1413, 2014.

[8] I. Damaj, A. Zaher and J. Yousafzai, "Assessment and Evaluation Framework with Successful Application in ABET Accreditation," *International Journal of Engineering Pedagogy*, vol. 7, no. 3, pp. 73-91, 2019.

[9] W. Kurniawan and M. H. H. Ichsan, "Teaching and learning support for computer architecture and organization courses design on computer engineering and computer science for undergraduate: A review," *2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, Yogyakarta, pp. 1-6, 2017.

[10] V. Chetty, M. Styles, S. -Y. Jung, F. Duarte, T.-W. J. Lee and M. Gunter, "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.

[11] T. d. S. Almeida, P. H. d. C. Lima, R. L. d. Carvalho and W. G. d. Silva, "Improvement in Logisim to Digital Systems Simulation in Higher Levels of Abstraction and Synthesis," *International Journal of Applied Information Systems (IJAIS)*, vol. 12, no. 13, pp. 1-7, 2018.

[12] A. Borodzhieva, Y. Aliev and G. Ivanova, "Modeling And Simulation Of Convolutional Encoders Using Logisim For Training Purposes In The University of Ruse," *International Scientific Journals of Scientific Technical Union of Mechanical Engineering*, vol. 2, no. 6, pp. 275-278, 2017.

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

[14] N. Srilatha, M. Sravani and Y. Divya, "Optimal Round Robin CPU Scheduling Algorithm using Manhattan Distance," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 6, pp. 3664-3668, 2017.

[15] G. Teodoro, T. Kurc, G. Andrade, J. Kong, R. Ferreira and J. Saltz, "Application performance analysis and efficient execution on systems with multi-core CPUs, GPUs and MICs: a case study with microscopy image analysis," *The International Journal of High Performance Computing Application*, vol. 31, no. 1, 2015.

[16] Y. Hu,Q. Li, Z. Cao, and J. Wang, "Parallel simulation of high-dimensional American option pricing based on CPU versus MIC," *Journal of Concurrency and Computation Practice and Experience*, vol. 27, no. 5, pp. 1110-1121, 2015.

[17] S. Hurn, K. Lindsay and D. J. Warne, "A heterogeneous computing approach to maximum likelihood parameter estimation for the Heston model of stochastic volatility," *Australian and New Zealand Industrial and Applied Mathematics Journal*, vol. 57, pp. C364-C381, 2016.

[18] C. Llorente, "Outcome-based Approach in Teaching Digital Systems Design for Undergraduate Computer and Electronics Engineering Programs," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 9, pp. 113-118, 2017.
[19] A. Sarosh, B. Zafar e A. A. Khan, "A Computationally Intelligent Expert System for Design Parameterization of Sub Orbital Carrier Vehicles," *Journal of Space Technology*, vol. 6, no. 1, pp. 2-7, 2016.
[20] H. Calvo, M. A. Moreno-Armendáriz and S. Godoy-Calderón, "A practical framework for automatic food products classification using computer vision and inductive characterization," *Journal Neurocomputing*, vol. 175, pp. 911-923, 2016.
[21] A. S. Tanenbaum, "Structured Computer Organization, Fifth Edition, Amsterdam," The Netherlands: Pearson Prentice Hall, 2005.
[22] M. H. H. Ichsan and W. Kurniawan, "Design and implementation 8 bit CPU architecture on Logisim for undergraduate learning support," *2017 International Conference on Sustainable Information Engineering and Technology (SIET)*, pp. 132-137, 2017.
[23] B. Newman and P. Newman, "Development through life: A psychosocial approach," Boston, USA: Cengage Learning, 2017.
[24] S. Card, "The psychology of human-computer interaction," Boca Raton: Taylor Francis, 2018.
[25] K. Hwang e N. Jotwani, Advanced Computer Architecture, 3e, India: McGraw Hill, 2016 .
[26] D. A. Patterson and J. L. Hennessy, "Computer Organization and Design: The Hardware/Software Design Fifth Edition," Oxford, USA: Elsevier, 2014.
[27] G. T. Hicham, E. A. Chaker and E. Lotfi, "Comparative Study of Neural Networks Algorithms for Cloud Computing CPU Scheduling," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 6, pp. 3570-3577, December 2017.
[28] N. Chandran, D. Gangodkar and A. Mittal, "Real-Time Implementation and Performance Optimization of Local Derivative Pattern Algorithm on GPUs," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 6, pp. 5457-5471, December 2018.
[29] Y. –S. Jeong, H. –W. Kim, N. Y. Yen, J. H. Park, "Multi-WSN Simulator with Log Data for Efficient Sensing on Internet of Things," *International Journal of Distributed Sensor Networks*, vol.1, pp. 1-11, 2015.
[30] B. Mustafa, "YASS: A System Simulator for Operating System and Computer Architecture Teaching and Learning," *European Journal of Science and Mathematics Education*, vol. 1, no. 1, pp. 34-42, 20132.
[31] Y. -C. Chung, W.-C.Hsu, S.-H. Hung, D. Kaeli, T. B. Jablin, Y. Sun and R. Ubal, "HSA Simulators," *Journal Heterogeneous System Architecture*, pp. 159-183, 2016.
[32] S. N. Deshpande and S. M. Dol, "FBL: An Activity to Improve Students' Conceptual Understanding of the Course Operating System," *Journal of Engineering Education Education Transformation*, vol. 30, no. 4, 2017.
[33] Y. Kim, W. Yang and O. Mutlu, "Ramulator: A Fast and Extensible DRAM Simulator," in *IEEE Computer Architecture Letters*, vol. 15, no. 1, pp. 45-49, 1 Jan.-June 2016.
[34] J. Ma, L. Yu, J. M. Ye and T. Chen, "MCMG simulator: A unified simulation framework for CPU and graphic GPU," *Journal of Computer and System Sciences*, vol. 81, no. 1, pp. 57-71, 2015.
[35] Kang W., Deng E., Wang Z., Zhao W, "Spintronic Logic-in-Memory Paradigms and Implementations,". *In: Suri M. (eds) Applications of Emerging Memory Technology*, *Springer Series in Advanced Microelectronics*, vol. 63, pp. 215-229, 2020.
[36] Kasabov N.K., "From von Neumann Machines to Neuromorphic Platforms," *In: Time-Space, Spiking Neural Networks and Brain-Inspired Artificial Intelligence, Springer Series on Bio- and Neurosystems*, vol. 7, pp. 661-677, 2019.
[37] T. Lioris, G. Dimitroulakos and K. Masselos, "An early memory hierarchy evaluation simulator for multimedia applications," *Elsevier, Journal Microprocessors and Microsystems*, vol. 38, no. 1, pp. 31-41, 2014.
[38] N. B. Zaidi, C. Hwang, S. Scott, S. Stallard, J. Purkiss and M. Hortsch, "Climbing Bloom's taxonomy pyramid: Lessons from a graduate histology course," *Journal Anatomical Sciences Education*, vol. 10, no 5, pp. 456-464, 2017.
[39] I. F. Sari, A. Rahayu, D. I. Apriliandari, and D. Sulisworo, "Blended Learning: Improving Student's Motivation in English Teaching Learning Process," *International Journal of Languages Education and Teaching*, vol. 6, no. 1, pp. 163-170, 2018.
[40] V. Jayashree, A. D. Kadage, S. A. Patil and G. S. Joshi, "OBE Module for Placement of Undergraduate Program Students," *Journal of Engineering Education Transformations (JEET)*, vol. 26, no. 9, pp. 49-54, 2016.