

A Hands-on approach in Teaching Computer Organization & Architecture through Project Based Learning

Aruna S. Nayak¹, Namrata D. Hiremath², Umadevi F.M.³, Vishwanath G. Garagad⁴

School of Computer Science & Engineering, KLE Technological University, Hubballi, India

¹arunan@kletech.ac.in

²namrata_hiremath@kletech.ac.in

³uma_devi_fm@kletech.ac.in

⁴vishwanath_gg@kletech.ac.in

Abstract: Computer Organization and Architecture (COA) is a fundamental core course in the curriculum of undergraduate engineering course in Computer Science. It plays an important role in the academic life of a computer science graduating student, to specialize in the verticals of System Architecture, Embedded System Design, or pursue research in the domains of system engineering for Masters as well as PhD degrees. In this paper, the authors present their teaching experiences through hands-on exposure using Project Based Learning (PBL) to improve the efficacy of learning concepts of Computer Organization and Architecture. Special emphasis is given on imparting the analytical, critical thinking, life-long learning and team working skills. It is evidenced in the literature that team activities promote enhanced learning experiences and value added skills to students, since students tend to engage actively in the group based academic activities. Theoretical concepts of the course are dealt at an abstract level, hence the instructors chose to use LOGISIM, a simulation tool to impart experiential learning. The experiment is observed to be productive, since students exhibited improvements realizing the concepts and were also skilled with analytical and critical thinking.

Keywords: Project based learning, course activity, datapath design, Logisim.

1. Introduction

COA is a pre-requisite course for other higher semester core courses such as Microcontrollers, Operating Systems, System Software, Parallel Computing, Quantum, Computing and so on.

Aruna S. Nayak

School of Computer Science & Engineering,

KLE Technological University, Hubballi, India

This course requires a thorough understanding of the fundamentals of Digital and Basic Electronics, both of which are hardware related courses. Studies have shown that generally CSE students show a lack of interest in these courses as their inclination is more towards programming

courses. Hence it has always been a challenge for course instructors to motivate students and create an interest in hardware courses. The experiment conducted on second year students for a course on COA is presented here. This PBL approach was conducted on II year undergraduate CSE students of a Technological University, in Karnataka, India. The total student strength across five divisions was three hundred and fifty three, with around seventy students per division. This strength also comprises roughly 10% lateral entry students. Lateral entry students are those who have completed their 3-year diploma in Computer Science and Engineering and have joined the undergraduate degree course in its second year. These students have good practical knowledge in their core courses, but at the same time, they are at a distinct disadvantage as compared to their peers since their admission process into the mainstream is delayed due to Government policies. This situation calls for an additional challenge to the course instructors to elevate these students to the same level as other students, which compelled the course instructors to think along the following lines:

- 1) How to motivate and encourage CSE undergraduate students to study hardware courses?
- 2) How to improve the performance of the students in these courses?
- 3) Does introduction of laboratory experience in COA course enhance student learning?

In the year 2017-18 the two courses on Digital System Design (DSD) and COA were taught in the same semester during the II year of the four year degree course. This attempt however failed to impart the concepts of COA to students as DSD is a prerequisite. A novel Project Based Learning (PBL) approach was devised in 2018-19 to improve the performance of students in hardware related courses, where the two courses DSD and COA were merged as one course with higher credits[15]. Moreover today's 21st century graduates need to acquire additional skills like critical thinking, design skills, problem solving, team leadership, communication and social skills[2]. IEEE/Association for Computing Machinery (ACM) Computer Curricula 2001 [3] and accreditation institutions [4], [5] encourage these objectives to be introduced in the engineering curriculum. COA syllabus was framed

following ACM guidelines where digital computer design and program execution using it is the main objective. This necessitated the need to introduce an approach called Project Based Learning which was first introduced at the University of Aalborg (Denmark) since 1974[6]. This awareness led to a rigorous 3-day workshop conducted for faculty in our university to support these learning processes. PBL approach is needed for adding lab experiences, complex problem solving, create awareness among students to demonstrate the use of various available techniques and practices. This experience promotes a spirit of team work and professionalism among them. Thus it was thought appropriate to bring in a PBL approach as an integral part of the curriculum [1].

2. Background

The course on COA involves many abstract concepts which are taught for CSE undergraduate students at an early stage[7]. Patterson and Hennessy highlight a traditional approach to teach COA but limit practical work to using assembly language programming and associated assembler and simulation tools[8]. Some recent works have emphasised the need to provide integrated laboratories to enable students to gain hands-on experience in COA[13]. Laboratory experiences are crucial to developing essential skills among students. Hands-on laboratory experience strengthens students' ability to understand the fundamental concepts, by effectively implementing a hardware system to attain the set objectives for the activity. This experience is also an add-on to students in terms of exposure to state-of-the-art methodologies and tools which in turn will prepare them for real engineering work after graduation[16]. These practices are adopted at Colorado University at Boulder [9] and at Auburn University [10]. The paper [11] describes a strategy to guide students in a step-by-step manner to construct their own basic processor. This approach seeks to help students go beyond simply understanding of computer architecture and guide them to apply the concepts to the design of a simple computer architecture.

3. Methodology

The merger of DSD and COA resulted in an L-T-P of 4-0-1.5, where Lecture duration was of 4 hours/week (4 credits), and practical duration was of 3 hours/week (1.5 credit). A project based team activity was conducted where team formation was done based on the previous grades of students. Students were grouped into teams, each comprising of four students with a proportional mix of gender and previous grades. Each team was assigned a course project to work on, whose problem statement was framed by the concerned course instructor. Relevant theory concepts to achieve the said objectives were taught in class and students started with construction of computer building blocks using basic logic gates in the laboratory which further served as inputs to better understand the

working of computing systems. The course design is focused to cover the design and operation of underlying hardware blocks in a computer, interface of hardware blocks to perform the complex computations/ operations, memory block designs, factors contributing to the computer's performance, instruction encoding, fetching and execution in computers (RISC in specific and RISC and CISC in general), role of control circuit to co-ordinate the data flow in the process of executing an instruction.. The organization and architecture of computers being at an abstract level are best understood through the design of datapath which explains the flow of data within the processor, from processor to memory and vice versa when an instruction is executed on a processor[12]. This necessitates the use of a suitable simulation tool. Considering the constraints of complexity and learning overhead of students, a basic survey conducted by instructors to identify a simple yet powerful simulation tool, LOGISIM a logic simulator was chosen. LOGISIM is equipped with a powerful library of hardware blocks that help students to design, simulate and analyze circuit using an easy to use graphical user interface. The activity given to students focuses on design and simulation of datapath for a given problem statement. The step-by-step design and simulation of datapath is influenced and derived from the concepts covered by Hamacher et al. in their text Computer Organization and Embedded Systems[14]. The activity involves design of datapath for arithmetic, logical, data transfer, and branch instructions. Students were given the following guidelines for implementing the activity:

1. Analyze the given problem statement for its various functionalities and relate it to specific topics in the syllabus.
2. Map the topics of learning to each of the features of the processor.
3. Represent the given problem in the form of block diagram.
4. Design complete Datapath diagram (Include animation if possible) along with relevant control signals activated for each clock pulse (wherever applicable).
5. Simulate the designed solution using Logisim.

The teams worked on their projects for the entire duration of the semester of 15 weeks, with continuous assessment done as per predefined rubrics. The progress of each team was continually monitored, and timely guidance was given by course instructors

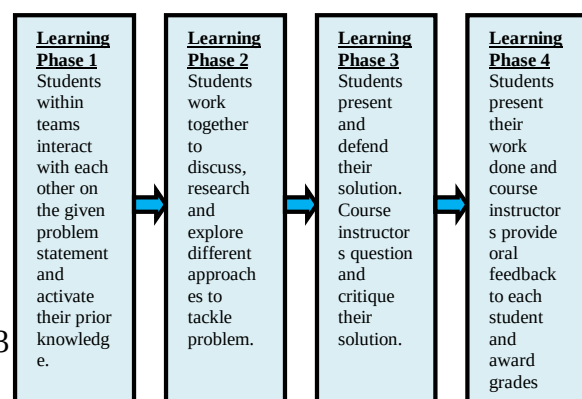


Fig. 1. Project based learning phases

Fig.1 shows the PBL expectations for the conduct of the course activity. The entire course activity planning and assessment was spread across 15 weeks. The details of which are discussed in Table 1 below.

Table 1. Activity planning and assessment

Week 1	Introduction to Logisim
Week 2 - 4	Design, simulation and implementation of combinational circuits
Week 5	Assessment 1(Combinational circuits)
Week 6 - 8	Design, simulation and implementation of sequential circuits
Week 9	Assessment 2(Sequential circuits)
Week 10	Design, simulation and implementation of n-bit ALU
Week 11	Design, simulation and implementation of M x N memory
Week 12&13	Course activity
Week 14	Intermediate assessment
Week 15	Final assessment

For the final assessment of the course activity each student in the team was required to present the details and was assessed for datapath design, individual contribution, innovation and simulation as per the rubrics shown in Table 2.

Table 2. Assessment rubrics

Datapath Design (8 M)	Able to design data path for the given instruction and show output for all possible inputs. (6-8M)	Able to design datapath, but works for only few inputs. (2-5M)	Unable to design. (0-1 M)
Individual Contribution (6M)	Actively participates as an individual and in a team for successful completion of structured enquiry. (4-6M)	Poor participation in a team. (2-3M)	Doesn't participate in a team. (0-1M)
Innovation (4M)	Able to solve the problem with innovative ideas and arrive at appropriate conclusion. (3-4M)	Able to solve the problem by adopting the existing methods. (1-2M)	Unable to solve the problem even with existing methods. (0M)
Simulation (2M)	Able to simulate designed circuit and the simulation	Unable to simulate the circuit properly and hence the	Unable to simulate the circuit. (0M)

	shows output for all possible inputs. (2M)	simulation is partial. (1M)	
--	--	-----------------------------	--

Fig.2 depicts the hardware within a processor and is termed as datapath for a RISC based machine. It also represents the way in which data generally flows during instruction execution [14]. This served as a base for students to design/simulate their own hardware for execution of a given instruction.

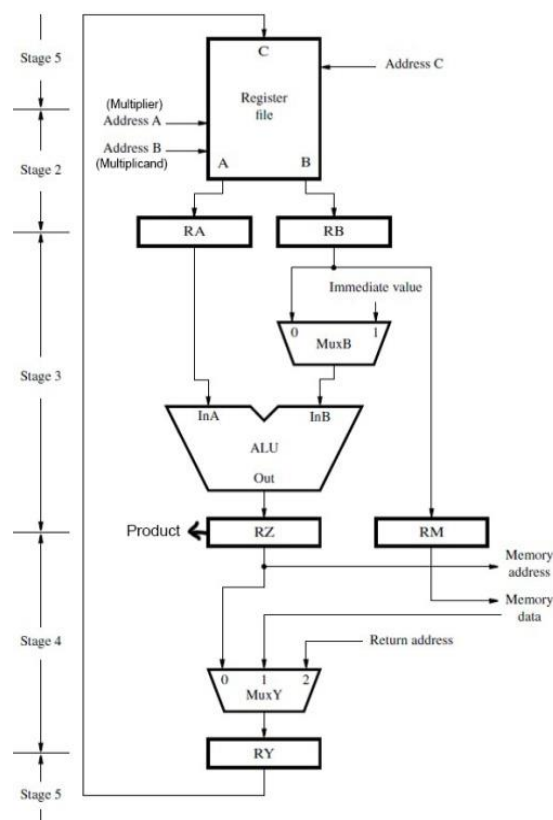


Fig.2. Datapath in a processor



Fig. 3. Discussion on the activity



Fig. 4. Brainstorming session in progress

Fig. 3 & 4 show the active participation of students in their course projects

4. Results and Discussions

Fig. 5 shows a given sample example of Datapath component for a binary sequential multiplier to multiply two 4-bit unsigned numbers done by one of the teams. To multiply two binary numbers a 4 bit adder, a temporary 4 bit register A, 1-bit register carry C and a 2:1 multiplexer are used

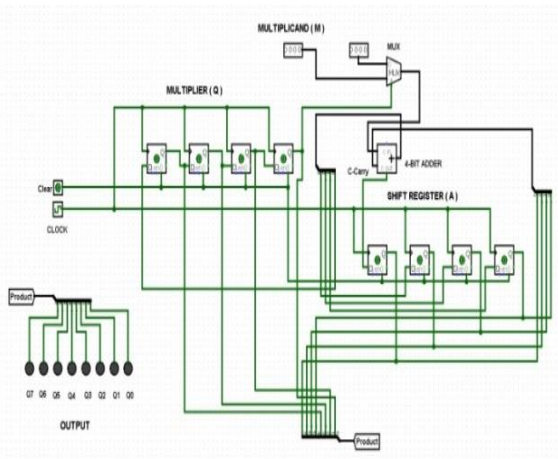
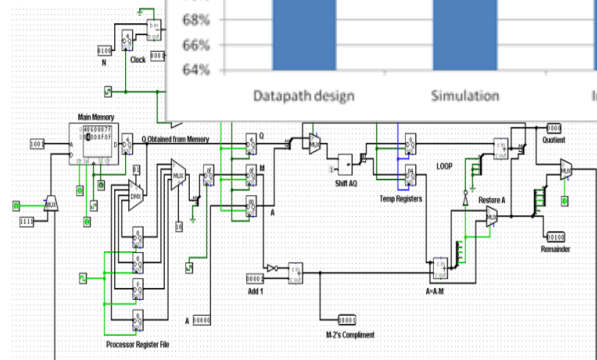
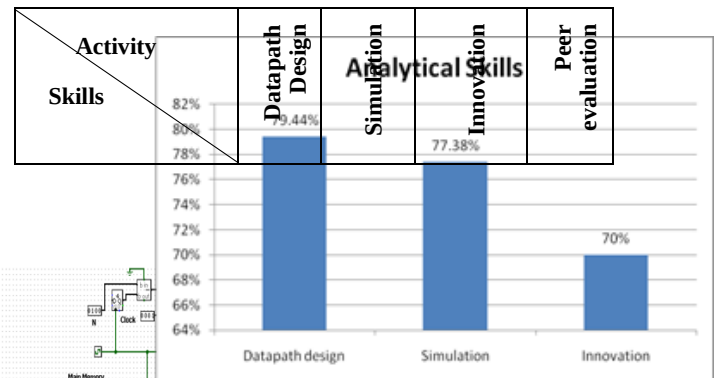


Fig. 5. Datapath component for sequential multiplier

Fig. 6 shows a RISC based datapath component designed by one of the teams, to divide two 4-bit numbers one of which is in memory location A and the other in processor register R6 using restoring division method. It stores the result back in memory location B

Fig. 6. Datapath component for 4-bit restoring division

Table 3. Weightage for Activity vs Skills



Analytical skills	60%	30%	10%	-
Tool usage	-	100%	-	-
Critical-thinking skills	80%	-	20%	-
Teamwork	-	-	-	100%

The scheme for assessing the activity is depicted in Table 3 where the activities datapath design, simulation, innovation and peer evaluation are mapped to analytical skills, tool usage, critical-thinking skills and teamwork. Datapath design and simulation involves analysis of how assembly instructions are decoded and executed on the underlying hardware of the system and hence contribute to the analysis skill building of students. Since the hardware is at an abstract level, simulation is the only means to

Fig.7. Attainment of Analytical skills

visualize the flow of data within the system and hence the student learns the usage of suitable simulation tool. In addition to analytical skills the critical-thinking ability among students is also honed through datapath design and innovation. The peer rating of each student in a team was averaged and the scores so obtained were used to assess the team quality of each of them.

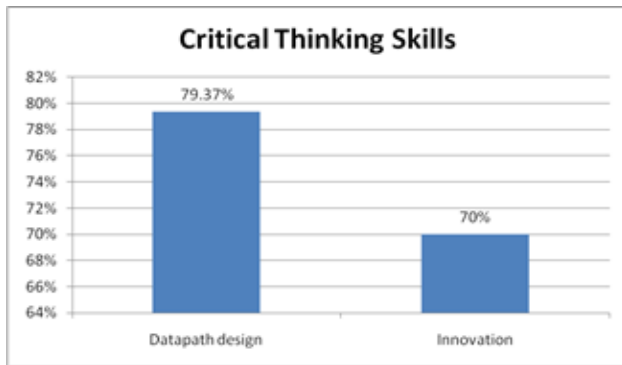


Fig. 8. Attainment of Critical Thinking skills

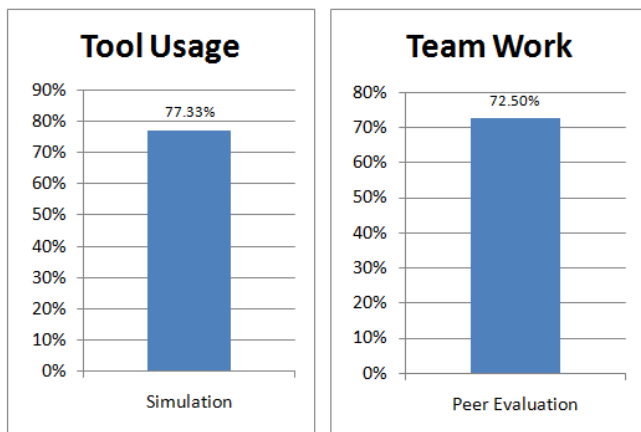


Fig. 9. Attainment of Teamwork & Tool usage

The average attainment of analytical skills through datapath design, simulation and innovation is 75.6% as shown in Fig.7. Similarly the critical thinking skill acquired by students stand at 74.7% as shown in Fig.8. Fig. 9 shows the attainment for teamwork and tool usage which are both above 70%.

5. Conclusion

The consistent efforts of the course teachers who were continuously engaged in imparting design skills to students resulted in around 75% of students obtaining analytical and critical thinking skills which improves their chances of a better performance in not only in higher semester courses but also enhances their placement opportunities. However the assessment of individual student contribution in a team is subject to error in judgment by the course teachers since it is difficult to evaluate each student's involvement and performance. The evaluation parameter, innovation stands at a good 70% score, as many students were keen on developing innovative alternate solutions for the same problem statement given to the team. Introduction of PBL into the course has motivated the students to understand the fundamentals of COA and associated laboratory helped to visualize the same. This in turn benefitted students to perform well in theory along with laboratory, which

encourages students in developing critical thinking, collaborative learning and leadership qualities.

References

- [1] Breiter, Andreas, Görschwin Fey, and Rolf Drechsler. "Project-based learning in student teams in computer science education." *Facta universitatis-series: Electronics and Energetics* 18, no. 2 (2005): 165-180.
- [2] Martínez-Monés, Alejandra, Eduardo Gómez-Sánchez, Yannis A. Dimitriadis, Iván M. Jorrín-Abellán, Bartolomé Rubia-Avi, and Guillermo Vega-Gorgojo. "Multiple case studies to enhance project-based learning in a computer architecture course." *IEEE Transactions on Education* 48, no. 3 (2005): 482-489.
- [3] ACM/IEEE computer society computing curricula (2004). [Online]. Available: <http://www.computer.org/education/cc2001>
- [4] (ABET) Accreditation Board of Engineering and Technology (2003). [Online]. Available: <http://www.abet.org/>
- [5] R. Felder and R. Brent, "Designing and teaching courses to satisfy the ABET Engineering Curricula," *J. Eng. Educ.*, vol. 91, no. 1, pp. 7-25, Jan. 2003.
- [6] Clausen, Helene Balslev, and Vibeke Andersson. "Problem-based learning, education and employability: a case study with master's students from Aalborg University, Denmark." *Journal of Teaching in Travel & Tourism* 19, no. 2 (2019): 126-139.
- [7] Calazans, Ney Laert Vilar, and Fernando Gehm Moraes. "Integrating the teaching of computer organization and architecture with digital hardware design early in undergraduate courses." *IEEE Transactions on Education* 44, no. 2 (2001): 109-119.
- [8] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, 2nd ed. San Mateo, CA: Morgan Kaufmann, 1998, p. 964.
- [9] J. P. Avery, J. L. Chang, M. J. Piket-May, J. F. Sullivan, L. E. Carlson, and S. C. Davis, "The integrated teaching and learning lab," in *Proc. 1998 Frontiers Educ. Conf.*, Tempe, AZ, Nov. 1998, <http://fairway.ecn.purdue.edu/~fie/>.
- [10] J. Y. Hung and S. M. Wentworth, "An integrated approach for electrical engineering laboratories," in *Proc. 1998 Frontiers Educ. Conf.*, Tempe, AZ, Nov. 1998.
- [11] Schuurman, Derek C. "Step-by-step design and simulation of a simple CPU architecture." In *Proceeding of the 44th ACM technical symposium on Computer science education*, pp. 335-340. ACM, 2013.
- [12] Al-Aubidy, Kasim M. "Teaching computer organization and architecture using simulation and FPGA applications." *J. Comput. Sci* 3, no. 8 (2007): 624-632.
- [13] Hiremath, Namrata D., F. M. Umadevi, and S. M. Meena. "Tutorial on Computer Organization and Architecture-Advantages and Challenges." *Journal of Engineering Education Transformations* (2018).
- [14] Hamacher, Carl. *Computer organization and embedded systems*/by Carl Hamacher...[et al.]. New York: McGraw-Hill Higher Education; London: McGraw-Hill [distributor], 2011.
- [15] Nayak, Aruna S., and M. Vijayalakshmi. "Teaching Computer System Design and Architecture course—An experience." In 2013 IEEE International Conference in MOOC, Innovation and Technology in Education (MITE), pp. 21-25. IEEE, 2013.
- [16] Nayak, Aruna S., G. G. Vishwanath, and F. M. Umadevi. "Effective teaching of Course on 8051 Microcontrollers through course Project." *Journal of Engineering Education Transformations* (2015): 192-195.