Paper ID #25925

An Integrated Learning Approach Used in Computer Architecture

Prof. Chaomin Luo, Mississippi State University

Dr. Chaomin Luo received his Ph.D. in Department of Electrical and Computer Engineering at University of Waterloo, in 2008, his M.Sc. in Engineering Systems and Computing at University of Guelph, Canada, and his B.Eng. in Electrical Engineering from Southeast University, Nanjing, China. He is currently Associate Professor in the Department of Electrical and Computer Engineering, at the Mississippi State University. He was panelist in the Department of Defense, USA, 2015-2016, 2016-2017 NDSEG Fellowship program and panelist in 2017 NSF GRFP Panelist program. He was the General Co-Chair of 2015 IEEE International Workshop on Computational Intelligence in Smart Technologies, and Journal Special Issues Chair, IEEE 2016 International Conference on Smart Technologies, Cleveland, OH. Currently, he is Associate Editor of International Journal of Robotics and Automation, and International Journal of Swarm Intelligence Research. He was the Publicity Chair in 2011 IEEE International Conference on Automation and Logistics. He was on the Conference Committee in 2012 International Conference on Information and Automation and International Symposium on Biomedical Engineering and Publicity Chair in 2012 IEEE International Conference on Automation and Logistics. He was a Chair of IEEE SEM - Computational Intelligence Chapter; a Vice Chair of IEEE SEM- Robotics and Automation and Chair of Education Committee of IEEE SEM. He has extensively published in reputed journal and conference proceedings, such as IEEE Transactions on Neural Networks, IEEE Transactions on SMC, IEEE-ICRA, and IEEE-IROS, etc. His research interests include engineering education, computational intelligence, intelligent systems and control, robotics and autonomous systems, and applied artificial intelligence and machine learning for autonomous systems. He received the Best Paper Award in the IEEE International Conference on Information and Automation (IEEE ICIA2017). He is an ASEE, INFORMS, and IEEE member. He is currently an Associate Editor of The 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IEEE-IROS 2019) .

An Integrated Learning Approach Used in Computer Architecture

Chaomin Luo, Ph.D.
Department of Electrical and Computer Engineering
Mississippi State University
Mississippi State, MS 39762, USA
email: Chaomin.Luo@ece.msstate.edu

Abstract

Student-centered learning has been discovered as the most effective pedagogy for teaching and learning science process skills and content. However, in some difficult curriculum, use of student-centered learning pedagogy only for enhancement of engineering student skills has its drawback. This paper presents a ***back-and-forth*** based pedagogy integrated with the student-centered learning for engineering and computer science student curriculum enhancement in Computer Architecture course. In the back-and-forth based learning, course materials are logically decomposed into interconnected pieces. The previous section will be frequently reviewed by the instructor later, on a back and forth basis, while some assignments are assigned to students for enhancement of their learning quality. A series of well-prepared review problems, examples, and assignments were assigned to students to cover various previous topics in this course, which assist in student learning enhancement of hardware and programming skills in Computer Architecture course. Satisfactory performance was evaluated by various milestone review sessions, assignments, in-class exercises, exams and other activities. Results of learning outcomes and assessment indicate that this integrated learning pedagogy is effective and efficient in student learning and improving the quality of computer design and organization.

1. Introduction

In higher education, especially, in engineering education, all the effort of educators is focused on educating students to be qualified for their future professionals. To achieve these learning outcomes in engineering education, a variety of pedagogical considerations have been implemented and experimented, such as inquiry-based learning (Behrouzi and Kuchma, 2016), project-based learning (Khorbotly, 2015; Wang *et. al*., 2017; Zhao *et. al*., 2017; Luo, 2015), cooperative learning (Akili, 2012), active learning (Luo, 2015; Akili, 2014; Oliveira, 2015), divide-and-conquer learning (Kos and Miller, 2017; Sullivan-Green, *et. al*., 2017), student-centered learning (Grigg and Stephan, 2018), and problem-based method (Oliveira, 2015), etc.

The project-based learning approach is one of the innovative methods promoted in engineering education. Khorbotly developed and taught a computer vision curriculum in the undergraduate electrical and computer engineering program using a project-based learning pedagogy. Some issues implemented in project-based approach are addressed (Khorbotly, 2015). Luo presented an on-going multiple-project-based pedagogy in electrical and computer engineering program. In this course, a sequence of well-prepared projects was assigned to students to cover various topics to help student learning for enhancement of research skills (Luo, 2015). Behrouzi and Kuchma addressed an inquiry-based learning pedagogy used in a freshman civil and structural engineering curriculum with an equipment-light laboratory course (Behrouzi and Kuchma, 2016).

Active-based learning is a learning protocol, in which teaching strives to involve students in the learning process more directly than in other methods (Luo, 2015). It emphasizes learning without the burden of assignments and without assessment through intimidating exams and tests unlike a traditional classroom environment. Oliveira adopted active learning approaches to encourage active learning and engagement among students in face-to-face electrical engineering technology courses. The assessment results demonstrated that the active learning strategies have successfully met the teaching requirements (Oliveira, 2015). Cooperative learning activities promote peer interaction and assist the development of engineering course in terms of better learning of concepts and content. Akili developed a cooperative learning method in a large-scale engineering education, in which the cooperative learning has been proven to be effective for all sorts of students (Akili, 2012). Kos and Miller utilized a divide-and-conquer learning scheme to teach a large freshmen engineering course by decomposing the course work into two types of assignments, weekly homework and a final report (Kos and Miller, 2017). Grigg and Stephan used a student-centered learning pedagogy in activities for large-enrollment undergraduate programs setting to foster a quality learning experience for engineering students through delivering foundational knowledge and facilitating skills development (Grigg and Stephan, 2018). Their engineering course was taught by addressing the distinct learning needs, interests, aspirations, or cultural backgrounds of individual students and groups of students.

Some educators integrate two or more teaching strategies for advancement of education quality. For instance, Khorbotly combined a project-based pedagogy and a traditional lecture-base teaching method that better develop and teach a computer vision class in the undergraduate electrical and computer engineering program (Khorbotly, 2015). Oliveira implemented active learning, and cooperative learning associated with a problem-based learning protocol in electrical engineering technology hands-on courses (Oliveira, 2015).

Student-centered learning has been discovered as the most effective pedagogy for teaching and learning science process skills and content. However, in some difficult curriculum, use of student-centered learning pedagogy only for enhancement of engineering student skills has its drawback.

This paper addresses a ***back-and-forth*** based pedagogy integrated with the student-centered learning for engineering and computer science student curriculum enhancement in Computer Architecture course. The objective of this Computer Architecture course offered for electrical engineering, computer engineering, software engineering and computer science students is to cultivate an understanding of modern computing technology through an in-depth study and learning of the interface between hardware and software. This paper describes a new course curriculum development that dedicates to enhancing the quality of student learning by such an integrated learning pedagogy. In the back-and-forth based learning, course materials are logically decomposed into interconnected pieces. The previous section will be frequently reviewed by the instructor later, on a back and forth basis, while some assignments are assigned to students for enhancement of their learning quality. This paper describes preliminary evidence that this hybrid back-and-forth and student-centered methodology can effectively improve student learning in the senior-level and graduate level semester-long Computer Architecture course by learning assessment.

2. Description of the Course

The ECE4713/6713 Computer Architecture course is a required course for computer engineering, software engineering, and computer science students. It is also a Technical Elective course for electrical engineering students. It delivers two 75-minute lectures per week. This course addresses basic structures in modern microprocessor and computer system architecture design. Topics include computer organization, instruction set design, memory system design, pipelining, memory hierarchy, and other techniques to exploit parallelism. System level topics are also covered such as storage subsystems and fundamental multiprocessor systems. It discusses quantitative evaluation of design alternative using design metrics such as caches performance and power dissipation. In this course, basic and advanced concepts of computer organization and architecture including CPU design, performance analysis, memory systems, and I/O interfacing, alternative design and evaluation of the control unit, pipelines, the arithmetic and logic unit, cache performance, and memory hierarchy. This course covers instruction formats and construction, addressing modes, computer performance measurement, particularly as they relate to hardware, virtual memory, and computer I/O concepts.

3 Back-and-Forth Learning Pedagogy Fused with Student-centered Learning

This course is a really challenging one in senior level and graduate level. It addresses basic structures in modern microprocessor and computer system architecture design. The course is complicated for students with various majors. Obviously, teaching the topics in order enables student to be tough to follow the course materials. Therefore, a back-and-forth learning pedagogy is developed in the sense that unlike course materials are covered in order, learning materials with *interconnection* in the back-and-forth pedagogy are particularly arranged to be covered repeatedly in a back-and-forth mode, Back and Forth. This technique allows instructors to explain a concept or idea in various stages and share thoughts with students repeatedly.

This curriculum is mainly focused on a set of rules and methods that describe the functionality, organization, and implementation of computer systems and architecture. Students learn MIPS assembly language, fundamentals of hardware technologies, computer arithmetic, pipelining, memory hierarchies, and I/O., while corresponding review questions, assignments, and in-class exercises were assigned to students frequently in a back-and-forth mode to apply the models they have learned to deeply understand computer architecture and organization. For evolution of outcomes, milestone questionnaires were utilized to analyze and interpret results to meet requirements of application and design of computer architecture and organization. The teaching quality was effectively assessed by student self-assessment and course assessment. Results of learning outcomes and assessment indicate that this integrated learning pedagogy is effective and efficient in student learning and improving the quality of computer design and organization.

In the student-centered learning, we implemented our learning methods as follows.

- We assigned the review and preview materials to students. Usually, we posted some preview questions related to the next lecture of materials, before the classes. Students can download the materials to review some past course materials and preview some materials that can help them to better understand the course materials.

- We assigned "In-class exercise assignments" in the classroom. Aligned to the current learning materials, we well-prepared some in-class exercise questions such as pipelined datapath questions, in which students are allowed to practice questions on group or individual basis with prescriptive time. After the class, the solutions are posted online.
- We prepared some examples to assist students in better understanding the course materials. The sample questions are properly prepared to associate with both knowledge of past materials and current materials. In this mode, students had 'hands-on' practice with the hard-to-understand materials. The instructor can teach students the materials by using the sample questions they just completed to help them to smoothly transit to new contents.

There are some topics as examples summarized in Table 1. We describe the difficult-to-be-learned topics and list some materials that can help students to actively prepare for the tough contents.

Table 1 The difficult topics and their reviewed materials

Topics	Difficult-to-be-learned topics (Back and Forth mode)	Previous knowledge to be reviewed
Sign-extend	○ Logical AND immediate and logical OR immediate put 0s into the upper 16 bits to form a 32-bit constant, unlike add immediate, which does sign extension. ○ MIPS has no subtract immediate instruction, and negative numbers need sign extension, so the MIPS architects decided to sign-extend the immediate field. ○ The binary bit pattern representing a number hides leading bits to fit the width of the hardware in Sign Extension Shortcut; sign extension simply restores some of them ○ To compute the branch target address, the branch datapath includes a sign extension unit ○ The two units needed to implement loads and stores, in addition to the register file and ALU, it needs the data memory unit and the **sign extension** unit. ○ ID/EX pipeline register can supply them from the immediate field since sign extension leaves these bits unchanged, therefore, these 6 bits are also the 6 least significant bits of the immediate field in the instruction.	○ Logical *AND* immediate and logical OR immediate – *andi, ori* ○ Branch hardware and the sign extension hardware. The datapath modified to resolve hazards via forwarding, in full datapath with sign extension ○ Sign extend – *addi, addiu, slti, sltiu*
Control Hazards	○ Control hazards tend to have higher branch frequencies as well as less predictable branches. ○ Delayed decision is used to resolve the control hazard ○ Control hazards remain important in both simple pipelines and more sophisticated ones. ○ In a five-stage pipeline, it can make the control hazard a feature by redefining the branch.	○ Control Hazards ***- beq, bnq, lw*** ○ Computers indeed use prediction to handle branches (Chapter 1) ○ MIPS software will place an instruction immediately after the delayed branch instruction that is not affected by the branch, and a taken branch changes the address of the instruction that follows this safe instruction.

Data Hazards	o Data hazards occur when the pipeline must be stalled because one step must wait for another to complete. o Data hazards arise from the dependence of one instruction on an earlier one that is still in the pipeline o Branch prediction and forwarding assist in making a computer fast o One complicated case, potential data hazards cause between the result of the instruction in the WB stage, the result of the instruction in the MEM stage, and the source operand of the instruction in the ALU stage. o Data hazards can be eliminated by adding nops to the code	o Data Hazards ***- lw, add, nop***, o In any MIPS instructions with one instruction that awaits another to complete. o MIPS instruction sets can simplify for pipeline designers, who must already cope with structural, control, and data hazards. o Load-use data hazard A specific form of data hazard in which the data being loaded by a load instruction has not yet become available when it is needed by another instruction.

Table 2 The interconnects among the topics

	a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	Remarks
a	N.A	_	_	_	_	_	_	_	_	Fundamental topic
b	√	N.A	_	_	_	_	_	_	_	Programming related
c	√	√	N.A	_	_	_	_	_	_	Performance
d	√	√√√	_	N.A	_	_	_	_	_	Datapath
e	√	√	√	_	N.A	_	_	_	_	Memory Hierarchy
f	√	√√	√	√	_	N.A	_	_	_	Processors
g	_	√√	√√√	√√√	_	√√√	N.A	_	_	Pipelined datapath
h	√	√√	_	√	√	√	√	N.A		Multiprocessors
i	√	√√	√	√	√	√	√	√	N.A	A comprehensive topic

Topics covered in this course are summarized as follows. In the learning pedagogy, we well analyzed the following topics and discovered their correlations so that we are aware which topics need to be reviewed and pre-reviewed in preparation of the course materials. In Table 2, the interconnections and correlations among topics to be covered are analyzed and summarized. It indicates, in Table 2, the extent to which one topic depends on another topic. "-" means there is no rigorous interconnection; "√" represents there is connection; "√√√" represents there is very high connection.

(a) Computer Abstractions and Technology: Interaction between computer hardware and software, performance, advances in speed, capability, and cost of processors and memory
(b) Instructions: Language of the Machine, Operations and operands, MIPS addressing
(c) The Role of Performance: Program execution and performance, cost/performance, and MIPS.
(d) The Processor: Datapath and Control Datapath, multicycle implementation of the MIPS instruction set, execution steps and control, hardwired vs. Microprogrammed control unit
(e) Large and Fast: Exploiting Memory Hierarchy, General principles, main memory, caches, and virtual memory.
(f) Interfacing Processors and Peripherals: Types and characteristics of I/O devices, I/O performance measures, buses, I/O interface to the memory, processor, and operating system, I/O interrupts, DMA, and handshaking.

(g) Enhancing Performance with Pipelining, Basic pipelining principles, pipelined datapath and control, hazards, pipeline implementation problems, performance of pipelined systems.
(h) Multiprocessors, Programming multiprocessors, multiprocessors connected by a single bus, multiprocessors connected by a network, clusters.
(i) Design I/O subsystem.

4. The Self-Assessments to Gauge Learning Outcomes

A series of well-prepared review assignments were assigned to students to cover a variety of topics in this course, Self-Assessments were carried out by various milestone review sessions. Aligned to ABET outcomes, the questionnaires are used as the self-assessments of for ABET assessment. Three questionnaires intended to obtain feedback from students to assist student's in improvement of learning quality, were carried out (summarized in Table 3). The results of averages in three questionnaires are 3.91, 4.04, and 4.12 out of 5, respectively. It implies that the learning quality of students has been improved as the developed strategy is implemented. In Table 3, it depicts that average of Questionnaire 2 is *3.32%* better than Questionnaire 1, whereas average of Questionnaire 3 has improved by *5.37%* from Questionnaire 1. The average of Questionnaire 3 has improved by *1.98%* in comparison with Questionnaire 2, which obviously shows the trend of improvement as the new fused pedagogy of the student-centered learning and back-and-forth scheme is implemented.

Table 3 Three questionnaires and improvement due to new pedagogy

	Questionnaire 1	Questionnaire 2	Questionnaire 3
	3.91	4.04	4.12
Improvement based on Questionnaire 1	—	3.32%	5.37%
Improvement based on Questionnaire 2	—	—	1.98%

These self-assessments are recommended at the end of the semester that form the basis for instructors to improve the teaching considering students' experience. Students in this course respond to the following five questions, corresponding to the ABET outcomes.

- Question 1 - "I can understand how to evaluate the performance of different computer architectures, and understand advances in speed, capability, and cost of processors and memory." (Outcome b: An ability to design and conduct experiments, as well as to analyze and interpret data relating to electrical systems).
- Question 2 - "I understand fundamental building blocks to design a memory, control unit, processor system of computer." (Outcome c: An ability to design electrical systems, components, or processes to meet desired needs).
- Question 3 - "I can understand, write, and execute C, as well as translate C into MIPS Assembly Language programs, and covert MIPS into machine codes of computer organization and architecture." (Outcome e: An ability to identify, formulate, and solve electrical engineering problems).
- Question 4 - "I have effective communication skills in the context of a collaborative, multi-disciplinary (CpE, EE, SE, and SE CS) class environment". (Outcome g: An ability to communicate effectively).

The self-assessment questionnaire results are summarized in Table 4 and Figure 1. It is clear that all students ‘strongly agree’, ‘agree’, or “neutral” with the statements aligned to the ABET outcomes (b) and (e). In the instructor’s experience of teaching multiple courses, the percentages for ‘strongly agree’ are relatively high in this course, pointing to a possible effect of the pedagogies we applied. Particularly, the percentage of ‘strongly agree’ on Question 4 is much higher at *67.90%* than usual. Question 4 is related to an ability to communicate in a collaborative context. The percentage of ‘strongly agree’ and ‘agree’ together on Question 1 is high at *85.80%*. Question 1 is related to an ability to design and conduct experiments, as well as to analyze and interpret data relating to electrical systems. Additionally, Question 3 is related to an ability to identify, formulate, and solve electrical engineering problems, which students have *50.00%* ‘strongly agree’ and *39.30%* ‘agree’ of statement. Question 2 is “I understand fundamental building blocks to design a memory, control unit, processor system of computer, alighted to an ability to design electrical systems, components, or processes to meet desired needs, reflects at *7.14%* of disagree. It is because that this class has no lab sessions associated with it. Students have no opportunity to perform lab and course project. Moreover, in comparison with the instructor’s previous experience teaching this course with a traditional project-based method (*i.e.* no student-centered and no back-and-forth module for reflection and adjustments), the percentages for ‘strongly agree’ and ‘agree’ in the current course are also higher. It is concluded that the hybrid version of the back-and-forth and student-centered is more effective.

Table 4 The questionnaire of students for assessment of education quality

Questions and Outcome	Survey				
	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Q1-b	42.90%	42.90%	14.30%	0%	0%
Q2-c	28.60%	42.90%	21.40%	7.14%	0%
Q3-e	50.00%	39.30%	10.70%	0%	0%
Q4-g	67.90%	25.00%	0%	3.60%	3.60%

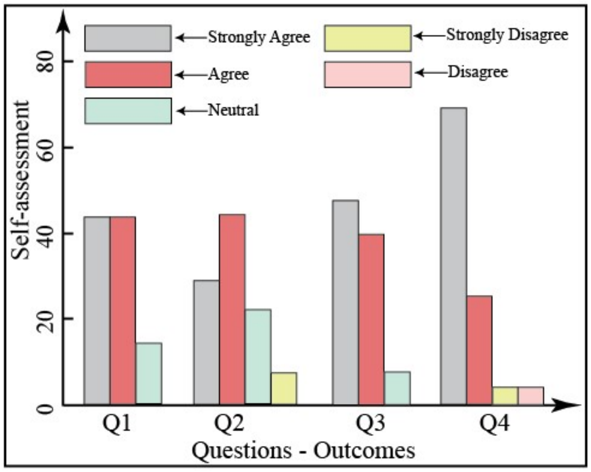

Figure 1 The illustration of the self-assessment results

5. Conclusion

Some efforts taken in approaching the instruction of a computer architecture course with a hybrid model of the student-centered learning methodology fused with a proposed ***back-and-forth*** pedagogy have been described in this paper. The preliminary evidence has been described that the developed back-and-forth methodology can effectively improve student learning in the senior-level and graduate level semester-long computer architecture course by learning assessment. Satisfactory performance was evaluated by various milestone review sessions, assignments, in-class exercises, exams and other activities.

References

1. A. Behrouzi, and D. Kuchma (2016, June), Inquiry-Based Learning to Explore the Design of the Built Environment Paper, *2016 ASEE* (American Society for Engineering Education) *Annual Conference & Exposition*, New Orleans, Louisiana. 10.18260/p.25725.
2. S. Khorbotly (2015, June), A Project-based Learning Approach to Teaching Computer Vision at the Undergraduate Level Paper, *2015 ASEE Annual Conference & Exposition*, Seattle, Washington. 10.18260/p.23432.
3. J. Wang, C. Luo, W. Zhao, and X. Li (2017, June), Empowering Students with Self-Regulation in a Project-Based Embedded Systems Course, *124th ASEE Annual Conference & Exposition* (ASEE'2017), June 25 - 28, 2017, Columbus, Ohio.
4. W. Zhao, X. Luo, C. Luo, and Y. Peng (2017, June), Design and Implementation of Project-Based Courses on Cutting-Edge Computer Technologies, *124th ASEE Annual Conference & Exposition* (ASEE'2017), June 25 - 28, 2017, Columbus, Ohio.
5. C. Luo (2015, June), Enhancing the Quality of Student Research by an On-going Multiple-Project-Based Course, *122nd ASEE Annual Conference & Exposition* (ASEE'2015), Seattle, WA, USA, June 14-17, 2015.
6. W. Akili (2012, June), Cooperative Learning Protocols: Varieties and Obstacles, *2012 ASEE Annual Conference & Exposition*, San Antonio, Texas.
7. C. Luo (2015, June), Hands-On Active Learning via Development of FPGA-based Intelligent Microwave Oven Controller, *122nd ASEE Annual Conference & Exposition* (ASEE'2015), Seattle, WA, USA, June 14-17, 2015.
8. A. M. Oliveira (2015, June), Simple Ways so Facilitate Active Learning in Hands-on Electrical Engineering Technology Courses, *2015 ASEE Annual Conference & Exposition*, Seattle, Washington. 10.18260/p.24709
9. W. Akili (2014, June), On Implementation of Classroom-Based Pedagogies of Engagement: Relevant Measures and General Outcomes, *2014 ASEE Annual Conference & Exposition*, Indianapolis, Indiana. https://peer.asee.org/22882.
10. L. E. Sullivan-Green, R. Mathur, and A. H. Feinstein (2017, June), Flipping STEM Classrooms Collaboratively Across Campuses in California, *2017 ASEE Annual Conference & Exposition*, Columbus, Ohio. https://peer.asee.org/28370.
11. S. J. Grigg, and E. A. Stephan (2018, June), (PREP)ARE: A Student-centered Approach to Provide Scaffolding in a Flipped Classroom Environment, *2018 ASEE Annual Conference & Exposition*, Salt Lake City, Utah. https://peer.asee.org/29651.