

Constructivist Multi-Access Lab Approach in Teaching FPGA Systems Design with LabVIEW

<http://dx.doi.org/10.3991/ijep.v3iS3.2768>

Walid BALID¹, Mahmoud Abdulwahed² and Imad Alrouh¹

¹Aleppo University, Aleppo, Syria

²Qatar University, Doha, Qatar

Abstract—Embedded systems play vital role in modern applications [1]. They can be found in autos, washing machines, electrical appliances and even in toys. FPGAs are the most recent computing technology that is used in embedded systems. There is an increasing demand on FPGA based embedded systems, in particular, for applications that require rapid time responses. Engineering education curricula needs to respond to the increasing industrial demand of using FPGAs by introducing new syllabus for teaching and learning this subject. This paper describes the development of new course material for teaching FPGA-based embedded systems design by using ‘G’ Programming Language of LabVIEW. A general overview of FPGA role in engineering education is provided. A survey of available Hardware Programming Languages for FPGAs is presented. A survey about LabVIEW utilization in engineering education is investigated; this is followed by a motivation section of why to use LabVIEW graphical programming in teaching and its capabilities. Then, a section of choosing a suitable kit for the course is laid down. Later, constructivist closed-loop model the FPGA course has been proposed in accordance with [2-4; 80,86,89,92]. The paper is proposing a pedagogical framework for FPGA teaching; pedagogical evaluation will be conducted in future studies. The complete study has been done at the Faculty of Electrical and Electronic Engineering, Aleppo University.

Index Terms—Constructivism, Embedded FPGA Systems, Engineering Education, LabVIEW, Hands-on, Remote Lab.

I. INTRODUCTION

Field-Programmable Gate Arrays (FPGAs) can be defined as a user programmable integrated circuits that offer reasonably high level of integration, negligible prototyping cost, and instantaneous manufacturing capability. Embedded FPGA systems have emerged rapidly during the last decade from expensive and complicated products, dedicated mainly for military and space applications, to modest cost products that are used in wide range of commercial applications [7-9]. Nowadays, embedded FPGA systems are the first, and mainly the only available, choice for products that need rapid signal processing and computing such as video graphics and video encoders [10,11]. FPGAs are at the heart of many systems and ideal for rapid prototyping of embedded systems designs. FPGA chips have grown in logic capacity while maintaining affordable cost for many applications [12]. It is evident that FPGAs are becoming popular for designating the most of embedded systems [13,14]. FPGA is a mature technology for the implementation of digital systems [15-17]. In fact, many FPGA chips achieve 100x speedups and

100x performance gains per unit of area as compared to a similar microprocessor [18-20], and 41x performance gains per unit of area as compared to a similar DSP.

FPGAs are becoming increasingly commonplace in digital design laboratories at all course levels in engineering higher education. FPGAs have been shown to afford a number of new opportunities for classroom learning, such as: FPGA-based robotics laboratory experiments [21-23]. FPGAs have also been successfully used in the classroom to study system on programmable chip (SoPC) design, hardware/software co-design, computer architecture, and signal processing hardware implementations [22-25]. The flexibility and ease of use of FPGAs provide an opportunity for students to work on more meaningful projects with tens of thousands of gates while still learning the fundamentals of digital design [26,27]. According to published experiences in literature [28-33], many universities around the world have adopted FPGA educational systems to teach their digital systems design courses. The first interests in embedding FPGA into engineering curriculum can be traced to 1997 [34]; since then, the development of new courses dedicated to Embedded Systems courses have been increasingly a trend in main Universities across the US [35], Europe [36], China [37], Taiwan [38] and South Korea [39]; interesting comparative studies on embedded systems curriculum development have also been recently investigated in literature [40,41].

II. HARDWARE PROGRAMMING LANGUAGES

There has been a number of different programming languages can be adopted for programming embedded FPGA systems, such as: Hardware Description Languages (HDL), e.g. VHDL [42], Verilog [43], C/C++ based (high-level programming languages), such as: JHDL [44], SystemC [45], Catapult-C [46], Impulse-C [47], Streams-C [48,49], Mittrion-C [50], and the Graphical programming languages (‘G’) e.g. LabVIEW [51].

Recent literatures [52-54] emphasize on the importance of using graphical programming for embedded FPGA systems development; it is noted that the life cycle of graphical programming is five times faster than textual programming; furthermore, graphical programming is rapid to learn and does not need particular programming experience [55]. On the other hand, the development of embedded FPGA systems using traditional programming languages need prospective in-depth knowledge in C/C++ or HDL in order to perform bit-level operations. This could be a valid approach for low-level optimization purposes. However, most engineering students do not have such in-depth knowledge in programming; moreo-

ver, the concept of embedded programming is alien to most. One of the ways to solve this conundrum is to implement technologies that provide a higher level of abstraction. Such an approach would enable students to focus more on the design and less on the individual bits. Literature [56] indicates that in order to program embedded systems, a major shift in paradigm is required. Literatures [53,54] show that the graphical programming languages are better suited for embedded design because they are based on the dataflow paradigm. Dataflow programming paradigm [57] uses the imperative programming model wherein the program is modeled as a series of operations with the data seemingly invisible. One advantage, for example, is that this enables dataflow programming paradigm to be as applicable to multi-core and multi-processor systems as it is to single-core systems just be duplicating a process loop/structure as illustrated in Figure 1.

III. LABVIEW GRAPHICAL PROGRAMMING ENVIRONMENT

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) is a visual programming language was developed by National Instruments (www.ni.com) in the early of 1980s with the vision of building an effective programming tool that reduces the required time to develop instrumentation systems software [58]. LabVIEW uses a dataflow programming model in which the output of each computation node is calculated when all the inputs are determined for that node. LabVIEW considers a suitable and cost-effective tool for implementing engineering education applications and help in introducing embedded systems design at earlier stages in the engineering curriculum. During recent years LabVIEW has seen a significant increase in industry, as well as academia with applications ranging from hands-on [59] to simulation [60]. LabVIEW has become a vital tool for engineers and scientists in research throughout academia, industry, and government labs. LabVIEW has been used in educational settings for implementing capstone design projects in the undergraduate curriculum. It has been used in data acquisition and control for a variety of systems including electrical and mechanical systems [61]. In addition, it has been used in data acquisition for monitoring a renewable energy based power system [62]. LabVIEW was taught in an interdisciplinary course on data acquisition [63] and integrated in a laboratory course to provide students with realistic, industry-based simulation experiences in the laboratory section of an upper division electronics course [64]. LabVIEW has been used as well to address the needs of various courses in a technology and science curriculum [65-70]. LabVIEW has also been integrated into a freshman engineering curriculum [71,72]. In [73], the authors reported how students developed programs in LabVIEW in a freshman engineering course to learn basic feedback control concepts. Choi [74] used LabVIEW as a tool for teaching a mechatronics course. Eckhoff [75] used LabVIEW for developing a remote control for a smart truss bridge rig offered for a class on smart materials and sensors. Lauterburg [76] enumerates several examples in which LabVIEW simulations have been effectively used in teaching physics and engineering in a classroom setting or in the lab, such as tracking motions with a laser distance sensor, the demonstration and analysis of heat conduction and visualizing acoustic signals. LabVIEW has been reported as a more suitable programming

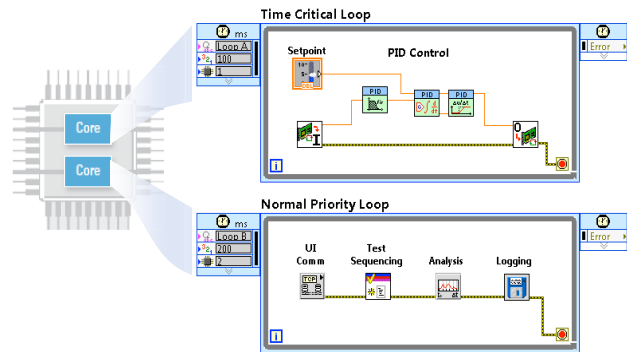


Figure 1. Programming Dual-core processor in LabVIEW

environment for engineering students than textual programming languages such as C or Java [77].

LabVIEW have many features that make it a favorable tool for developing engineering education software tools, some of these are:

1. LabVIEW is inherently a Virtual Instrumentation (VI) platform and development environment for a visual programming language that effectively enables building user friendly interfaces for any engineering application.
2. LabVIEW has comprehensive and wide-range libraries for all engineering applications, such as: measurements, data processing, various kinds of embedded systems, industrial systems, and data analysis.
3. LabVIEW has advanced simulation implementation for a wide-spectrum of engineering applications/problems.
4. LabVIEW toolkits and models offer a comprehensive library sets of off-the-shelf functions.
5. LabVIEW has many interfacing libraries that enable integrating other engineering files, such as: Matlab m file codes, Solidworks 3D models, IP codes, and this is particularly an enabler for using legacy codes.
6. LabVIEW also has connectivity tools with other applications such as MS Office or SQL database.
7. LabVIEW offers multimedia connectivity where audio/video files can be embedded in the developed application.
8. LabVIEW offers a software application development kit that enables the creation of standalone executable applications from LabVIEW VIs, similar to other programming languages such as C++ /Basic/HDL.

These features and others have made LabVIEW an increasingly used platform for developing engineering education applications.

IV. LABVIEW APPLICATION AREAS

LabVIEW applications are spread in various systems. LabVIEW has proven high results in industrial applications [78]. Some of the application areas of LabVIEW are: Simulation, Data Acquisition, Embedded Systems (MCUs, DSPs, FPGAs, and MPUs), Industrial Systems, and Data Processing. The built-in library of LabVIEW has a number of VIs that can be used to design, simulate, and develop any system. These features of LabVIEW will help provide an interdisciplinary integrated teaching and learning experience that integrates team-oriented, hands-

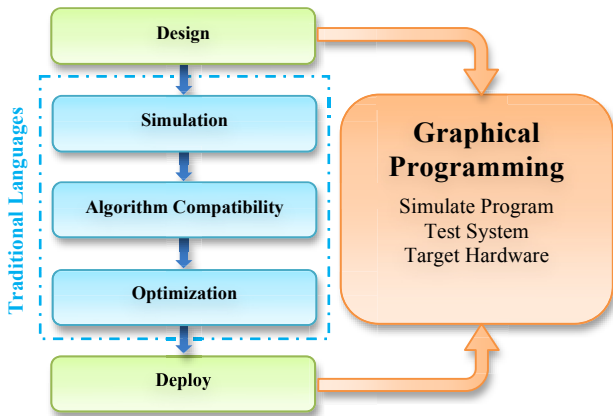


Figure 2. Comparison between steps taken in conventional methods to graphical programming to program hardware target; the “G” programming for embedded systems has turned the advantage of higher level of abstraction from a novel idea to a practical implementation possibility.

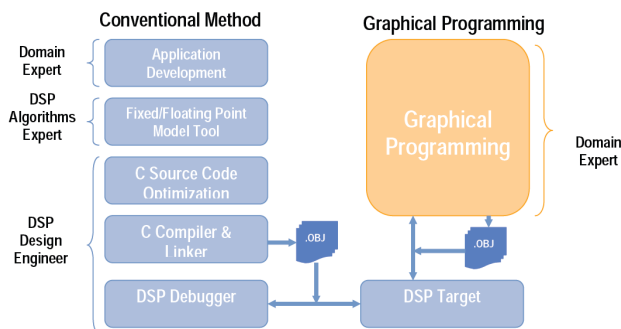


Figure 3. Comparison between steps taken to program DSP in conventional methods to graphical programming

on learning experiences throughout the engineering technology, and sciences curriculum that engaging students in the design and analysis process beginning with their first year.

LabVIEW has the ability to target hardware ranging from desktop systems, handheld computers, real-time systems, FPGA-based boards, 32-bit MPUs and DSPs. One of the strength points of graphical programming languages is that users have the ability to port the same design to multiple targets with little to no code changes. But the power of graphical programming lies in the fact that the designer does not need to interface with the low-level code that is generated for the corresponding targets. The designer can always stay at the higher level of abstraction which makes it easy to develop sophisticated systems that can then be downloaded easily onto the desired target, including DSPs. Figure 3 illustrates a comparison between steps taken to program DSP in conventional methods to ‘G’ programming.

Figure 4 shows a comparison between steps taken to program FPGA in conventional method to ‘G’ programming.

Inherent Parallelism One of the biggest advantages of dataflow paradigm on which graphical programming is based on is that it is inherently parallel. With FPGAs being one of the most common hardware platforms, this becomes very critical. Figure 5 and 6 illustrate a comparing ‘G’-based with Textual-based code.

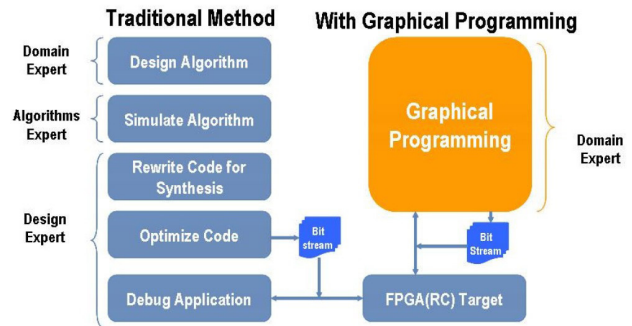


Figure 4. Comparison between steps taken to program FPGA in conventional methods to graphical programming



Figure 5. Comparing ‘G’-based with Textual-based code for two parallel DAQ timed-loops

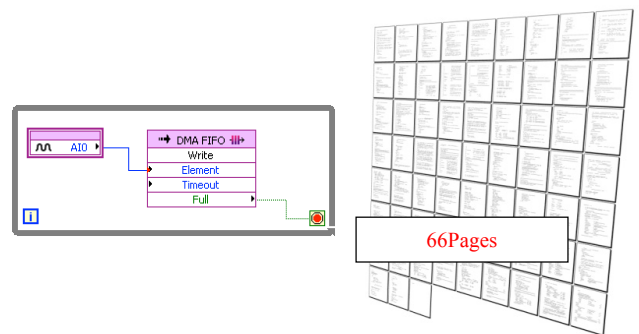


Figure 6. Comparing ‘G’-based with Textual-based code for reading and storing data loop from FPGA analog input.

V. THE SPARTAN-3E LAB KIT

The Spartan-3E starter Board has been adopted as a platform for the FPGA laboratory course. The Board is based-on Xilinx FPGA chip with up to 1.6million gate, 376 I/Os, 32 bit RISC processor, and DDR interfaces. The board features a Xilinx Platform Flash, USB and JTAG parallel programming interfaces with numerous FPGA configuration options via the onboard Intel StrataFlash and ST SPI Flash. It also contains a plenty of peripherals such as: VGA port, PS/2 port, 2x16 LCD, RS232 dual-interface. The kit is sold with a very appealing price for academic; USD149.

The board is also compatible with the MicroBlaze Embedded Development Kit (EDK) and PicoBlaze from Xilinx. Further technical specifications for this board can be found at [79]. Figure 7 illustrates the Xilinx SPARTAN-3E XUP board.

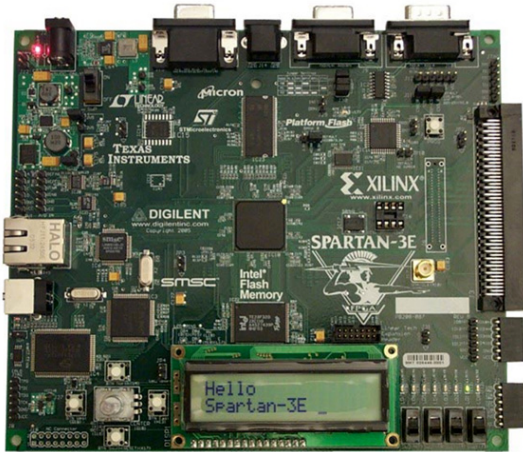


Figure 7. Xilinx Spartan-3E Starter Board

Some of the factors that were considered in choosing the right hardware platform are: The cost to the students; Influential hardware properties of the Spartan-3E starter board, where Xilinx XC3S300E Spartan-3E FPGA provides sufficient "elbow-room" to hold multiple 32-bit processor cores along with a long list of peripherals. It contains a variety of standard interfaces (multiple SPI devices, variety of memory interfaces, serial ports, VGA, PS/2, etc...), all these features provide the students with opportunity to make meaningful experiments. A sufficient number of human I/O devices available via expansion connectors allowing instant gratification for the lower-level and introductory experiments (switches, rotary knob, push buttons, LEDs, and an LCD display); fast downloading of Bitstream file, operating system, and responsive debugging over high-bandwidth USB port. PHY 10/100 Ethernet interface enables inter- and intra-group projects; retaining multiple FPGA configurations in a non-volatile flash memory.

VI. EXPANDING THE KIT CAPACITY

Although the kit has a variety of peripherals' on board, a number of other peripherals (20-module) have been developed in order to enrich the boards functionality for teaching and learning purposes of the laboratory course; these extra peripherals include units for light intensity measurement, DC motor control, temperature measurement, etc... The reason for designing these peripherals is that the students don't have in-depth knowledge to program the available on board modules. By these modules students can be prepared to be able to develop the on board modules. Some of the designed expansion card and modules are illustrated in Figure 8.

VII. THE FPGA LABORATORY COURSE DESIGN METHODOLOGY

A system engineering approach with strong emphasis on formative assessment and feedback has been utilized in developing the pedagogical content of the course. Theoretical developments of feedback systems design for pedagogical processed has been originally proposed in form of mathematical dynamical linear differential models in [86,87,89,92]. Implementation and evaluation of findings have been disseminated in [2-4]. The empirical investigation indicated significant performance in utilizing closed-loop systems approach in the process of teaching

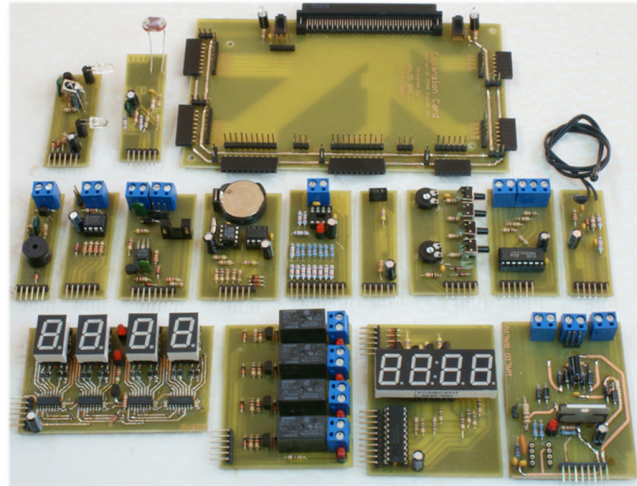


Figure 8. The Designed Expansion Card and Modules

and learning, with gains in learning ranged between 70% to 100% [3,4]. Impact of utilization of virtual experiments in preparation mode for hands-on labs have been investigated in [5,90,94]; Statistical analysis results have shown a strong evidence that virtual lab preparation contributes positively to learning experience in hands-on lab sessions.

Based-on the previous theoretical developments [86,87,89,92] and positive empirical findings [1-5], a closed-loop FPGA course has been designed. The concept of the Trilab [6,86,91], blended pedagogically with Kolb's Experiential Learning theory [80] were utilized in realizing the closed-loop model, together with other elements. The Trilab concept indicates the utilization of three laboratory access modes for enriching experiential learning, via virtual, hands-on, and remote experiments.. The three lab components is proposed to be integrated with other innovative teaching and learning approaches [4, 93] such as frequent sequential assessment, frequent feedback, and experiential activities via projects, such as shown in Figure9; Further details are provided in the next section.

VIII. A MODEL OF CONSTRUCTIVIST FPGA LAB

The design methodology of the proposed FPGA course is centered on the proposed comprehensive and constructivist educational model of learning/teaching in closed-loop that illustrated in Figure 9. The course would have first clear objectives; then the Kolb-learning style (KLS) is applied; based on KLS inventory and the course objectives, the teaching methodology in the classroom to be set; then the first session, will start. Each session would also have clear objectives and it consists of classroom session and laboratory session; the audience electronic response Systems (ARS) in the classroom session is to stimulate the students' attention every 10min and to provide a formative assessment and evaluation of the students understanding (as a closed-loop).

The remote lab is also can be used in classroom for giving deeper understanding of the theoretical aspects in the real world (e.g. applying a PID controller parameter and evaluate the system real-time response). After the classroom session starts the laboratory session with a pre-lab session which can be a simulation for a given homework and can be done separately or in cooperative groups. After that the hands-on session will be conducted to apply the simulation results on the real rig and evaluate the findings. Since the available time for the hands-on lab is

SPECIAL FOCUS PAPER
CONSTRUCTIVIST MULTI-ACCESS LAB APPROACH IN TEACHING FPGA SYSTEMS DESIGN WITH LABVIEW

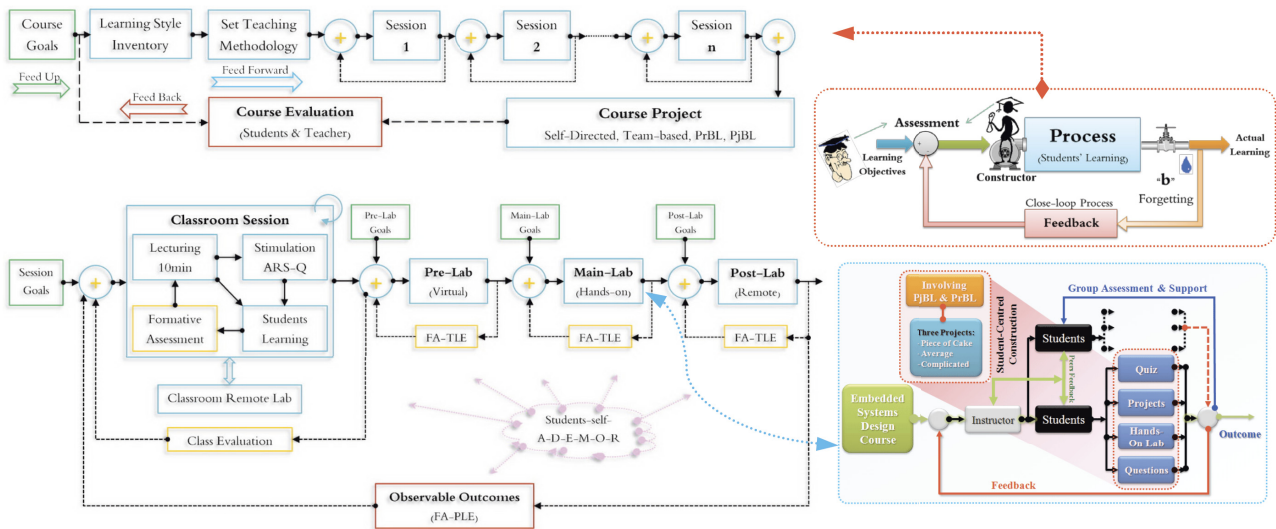


Figure 9. The scheme of the proposed comprehensive educational model of learning/teaching in closed-loop process.

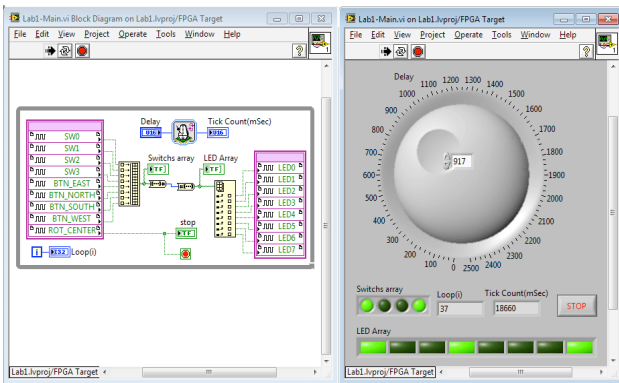


Figure 10. The Graphical code and interface of Lab.01

not adequate for the students, a remote lab is advised to be conducted after the hands-on session. All sessions are to be evaluated by peers and teacher. At the end of the course, a self-directed team-based project is proposed; the project has meet a real problem has a direct relationship with the course contents.

Proceeding of the developed model of learning/teaching, a set of hands-on laboratory activities based on LabVIEW, Spartan Kit, and the extra peripherals have been designed; the lab manual was designed and systematized in a constructivist student-centric process so that the students can follow the manual and apply the steps with a minimum supervision. It's very important to boost every single idea by a graphical model or picture such as: functional diagram, flowchart, structure chart, logic chart, etc. which can help the students to more clearly gathering the information needed to design the experiment, doing the planned process, gaining hardware and software knowledge, and fully understand the objectives of the experiments.

IX. FPGA REMOTE LAB IMPLEMENTATION

Figures 11-13 show the remote lab platform which has been implemented by using a remote desktop sharing method. LogMeIn-Hamachi was used for proving a VPN and private IP for the experiment rig. The experiment rig and its components connection are illustrated in Figure14.

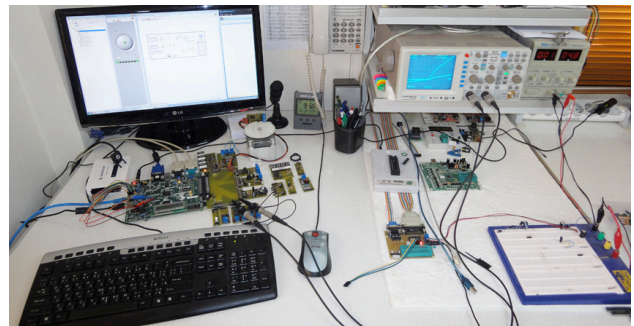


Figure 11. Remote Lab Platform

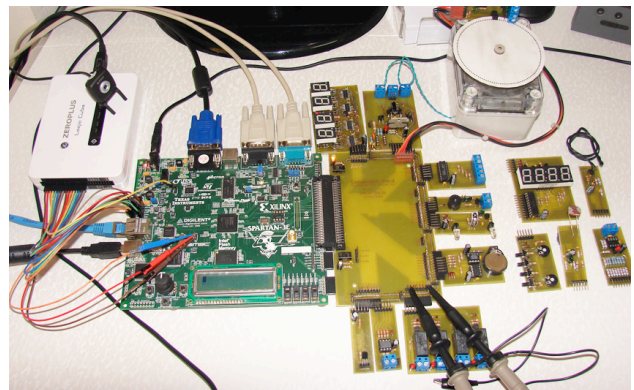


Figure 12. Remote Lab Platform, interfacing the Kits

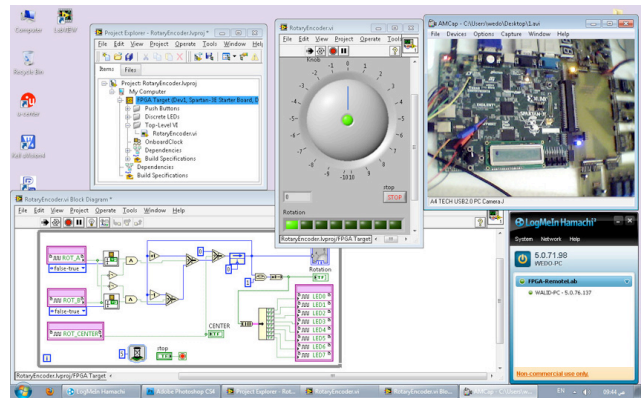


Figure 13. Remote access to the Lab from distance PC.

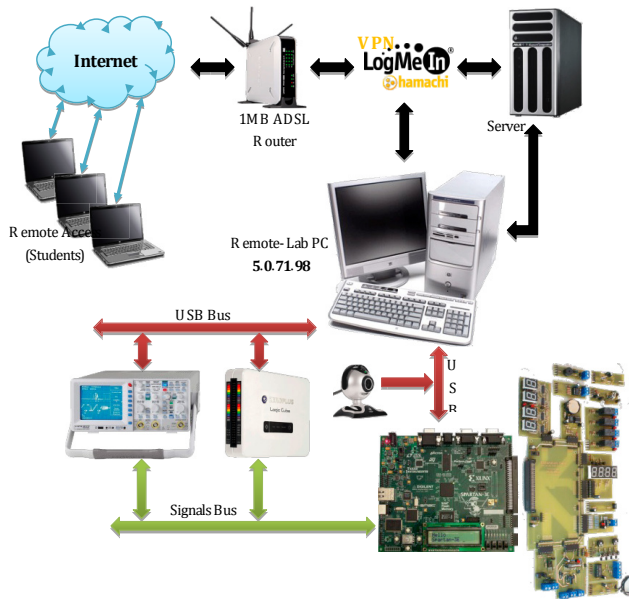


Figure 14. The components of the Remote Lab scheme

X. FUTURE PLANS

The Learning/Teaching process is considered as a dynamical, non-linear, process with numerous, dissimilar, and unknown states (parameters); thus describing the model of such system is very complicated. In such cases, fuzzy logic control approach considers as the optimal solution for modeling such system that the fuzzy logic requires describing the system inputs and outputs, no matter the internal states model are defined or not. Therefore, we suggest modeling the learning/teaching process using fuzzy logic or adaptive control approach.

XI. CONCLUSIONS

The paper reports on the development of a hands-on curriculum for teaching and learning of Embedded FPGA Systems design. To avoid complications in VHDL languages and syntax learning and to have more focus on systems design concepts, LabVIEW has been adopted as a programming environment for the ESs FPGA course. SPARTAN-3E starter board for FPGA chips have been found reasonably adequate (economically and technically) for the course objectives, however a number of extra peripheral modules have been developed in order to enrich the boards functionality. A system engineering approach with strong emphasis on formative assessment and feedback has been utilized in developing the pedagogical content of the course. The capacities of LabVIEW enabled implementing a triple access mode whereby students can perform FPGA experiments in three formats: virtual (simulation), hands-on and remote lab modes. The course with its technical and pedagogical models will be taught and the outcomes will be evaluated next academic semester.

REFERENCES

[1] W. Balid and M. Abdulwahed, "Multi-Purpose Open Source Embedded Systems Laboratory Kit for Engaging Students Towards Experiential Education", The 9th Annual American Society for Engineering Education (ASEE) Global Colloquium on Engineering Education Budapest, Hungary October 12-15, 2009

- [2] Balid W., I. Al-Rouh , M. Alhaj Dibo, "YES, Constructivist Project Based Learning Could be the Magical Cure for Engaging Future Engineers! Case of an Embedded Systems Course", Researches Journal at Aleppo University, Engineering Sciences Series, ver. 55 / (2010).
- [3] M. Abdulwahed and W. Balid, "A Constructivist PBL Approach in Teaching Embedded Systems Hands-on Course, Comparative Study. The 9th Annual American Society for Engineering Education (ASEE) Global Colloquium on Engineering Education Budapest, Hungary October 12-15, 2009.
- [4] M. Abdulwahed and W. Balid, "An Assessment Rich PBL vs. Classical Teaching approach. The 2nd international research symposium on PBL (IRSPBL09), 2-4 December, Victoria University, Melbourne, Australia, 2009
- [5] M. Abdulwahed and W. Balid, "The Impact of Different Pre-Lab Preparation Modes on Embedded Systems Hands-on Lab. The 9th Annual American Society for Engineering Education (ASEE) Global Colloquium on Engineering Education Budapest, Hungary October 12-15, 2009
- [6] M. Abdulwahed and ZK. Nagy, Developing the TriLab, a Triple Access Mode Laboratory of a Process Control Rig Using LabVIEW and Joomla. Computer Applications in Engineering Education. <http://dx.doi.org/10.1002/cae.20506>
- [7] K. Compton and S. Hauck, "Reconfigurable Computing: A Survey of Systems and Software" ACM Computing Surveys, vol. 34, no. 2, pp. 171–210, June 2002.
- [8] N. Ohba and K. Takano, "An SoC Design Methodology using FPGAs and Embedded Processors" in Proc. of the 41st Annual Conference on Design Automation, June 2004, pp. 747–752.
- [9] S. A. Edwards, "Experiences Teaching an FPGA-based Embedded System Class," in Proc. of the 1st Workshop on Embedded System Education, vol. 2, October 2005, pp. 56–62.
- [10] T. Wiantong, P. Y. K. Cheung, and W. Luk, "Hardware / software Codesign: A Systematic Approach Targeting Data-intensive Applications" IEEE Signal Processing Magazine, vol. 22, pp. 14–22, May 2005. <http://dx.doi.org/10.1109/MSP.2005.1425894>
- [11] I. Amer, W. Badawy, and G. Jullien, "A Design Flow for an H.264 Embedded Video Encoder," in Proc. of the 3rd International Conference on Information and Communications Technology, November 2005, pp. 178–181.
- [12] K. Compton and S. Hauck, "Reconfigurable Computing: A Survey of Systems and Software" ACM Computing Surveys, vol. 34, no. 2, pp. 171–210, June 2002. <http://dx.doi.org/10.1145/508352.508353>
- [13] N. Ohba and K. Takano, "An SoC Design Methodology using FPGAs and Embedded Processors," in Proc. of the 41st Annual Conference on Design Automation, June 2004, pp. 747–752. <http://dx.doi.org/10.1145/996566.996769>
- [14] S. A. Edwards, "Experiences Teaching an FPGA-based Embedded System Class," in Proc. of the 1st Workshop on Embedded System Education, vol. 2, October 2005, pp. 56–62.
- [15] D. B. Stephen, J. F. Robert, Jonathan Rose, and G. V. Zvonko, Field-Programmable Gate Arrays. Kluwer Academic Publishers, 1992.
- [16] R. C. Cofer and Benjamin Harding. Rapid System Prototyping with FPGAs: Accelerating the Design Process. Elsevier, 2005.
- [17] Esma Alaer, Ali Tangel, and Mehmet Yakut. MIB-16 FPGA based design and implementation of a 16-bit microprocessor for educational use. WSEAS Transactions on Advances in Engineering Education, 5(5), May 2008.
- [18] D.A. Buell, K.L. Pocek, Custom computing machines: an introduction. J. Supercomput. 9(3), 219–29 (1995) <http://dx.doi.org/10.1007/BF01212869>
- [19] A. DeHon, Comparing computing machines, in SPIE-Int. Soc. Opt. Eng. Proceedings of SPIE—the International Society for Optical Engineering, vol. 3526, pp. 124–33, 1998
- [20] J.E. Vuillemin, P. Bertin, D. Roncin, M. Shand, H.H. Touati, P. Boucard, Programmable active memories: reconfigurable systems come of age. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 4(1), 56–69 (1996)
- [21] K. Newman, J. O. Hamblen, and T. S. Hall, "An introductory digital design course using a low-cost autonomous robot," IEEE

- Transactions on Education, vol. 45, no. 3, pp. 289–296, Aug. 2002. <http://dx.doi.org/10.1109/TE.2002.1024623>
- [22] J. O. Hamblen, "Rapid prototyping using field-programmable logic devices," *IEEE Micro*, vol. 20, no. 3, pp. 29–37, May/June 2000. <http://dx.doi.org/10.1109/40.846307>
- [23] J. O. Hamblen and T. S. Hall, "Engaging undergraduate students with robotic design projects" in Proceedings of the Second IEEE International Workshop of Electronic Design, Test and Applications, Jan. 2004.
- [24] T. S. Hall and J. O. Hamblen, "System-on-a-programmable-chip development platforms in the classroom," *IEEE Transactions on Education*, vol. 47, no. 4, pp. 502–507, Nov. 2004. <http://dx.doi.org/10.1109/TE.2004.825926>
- [25] T. S. Hall and D. V. Anderson, "A framework for teaching real-time digital signal processing with field-programmable gate arrays," *IEEE Transactions on Education*, vol. 48, no. 3, Aug. 2005. <http://dx.doi.org/10.1109/TE.2005.853069>
- [26] M. A. Soderstrand, "Role of FPGAs in undergraduate project courses," in 1997 IEEE International Conference on Microelectronic Systems Education Proceedings, April 1997, pp. 109–110.
- [27] M. S. Nixon, "On a programmable approach to introducing digital design," *IEEE Transactions on Education*, vol. 40, no. 3, pp. 195–206, Aug. 1997. <http://dx.doi.org/10.1109/13.618030>
- [28] Don Bouldin. Impacting education using FPGAs. In Proceedings of the 18th International Symposium on Parallel and Distributed Processing, Santa Fe, USA, April 2004.
- [29] A. E. Stephen, Experiences teaching an FPGA-based embedded systems class. In Proceedings of the Workshop on Embedded Systems Education (WESE), New Jersey, USA, 2005.
- [30] Yong-K. Jung. Work in progress – a rapid design methodology for FPGA-based processor platform design education. In Proceedings of the 35th. ASEE/IEEE Frontiers in Education Conference, Indianapolis, USA, October 2005.
- [31] Dong-Soo Kang, Soo Yun Hwang, Kyoung-Son Jhang, and Kang Yi. A low cost and interactive rapid prototyping platform for digital system design education. In Proceedings of the 2007 International Conference on Microelectronic Systems Education, San Diego, CA, USA, June 2007. <http://dx.doi.org/10.1109/MSE.2007.9>
- [32] Thilo Pionteck. Teaching informatics students the secrets of hardware design. In Proceedings of the 2007 International Conference on Microelectronic Systems Education, San Diego, CA, USA, June 2007. <http://dx.doi.org/10.1109/MSE.2007.82>
- [33] Michael J. Wirthlin. Senior-level embedded system design project using FPGAs. In Proceedings of the 2005 International Conference on Microelectronic Systems Education, Anaheim, CA, USA, June 2005.
- [34] H. Ochi, "ASaver.1: An FPGA-based education board for computer architecture/system design," in Proc. Asia and South Pacific Design Automation Conf., Makuhari, Japan, Jan. 1997, pp. 157–165. <http://dx.doi.org/10.1109/ASPDAC.1997.600102>
- [35] Hall, Tyson, Bruckner, and Halterman, "A Novel Approach to an Embedded Systems Curriculum", 36th ASEE/IEEE Frontiers in Education Conference, San Diego, CA, Oct. 28 – 31, 2006..
- [36] A. Sangiovanni-Vincentelli and Alessandro Pinto, "Embedded system education: a new paradigm for engineering schools?" *SIGBED Rev.*, vol.2, no.4, pp.5–14, 2005.
- [37] Kang Yimei, "A Graduate Program on Embedded Software Engineering in China", 20th Conference on Software Engineering Education & Training (CSEET'07), 2007.
- [38] Chi-Sheng Shih, etc. "Toward HW/SW Integration: a Networked Embedded System Course in Taiwan", *ACM SIGBED Review*, Vol.4, No.1, January 2007, pp36-54. <http://dx.doi.org/10.1145/1217809.1217815>
- [39] Suehee Pak, Eunha Rho, Juno Chang, and Moon Kim, "Demand-Driven Curriculum for Embedded System Software in Korea", [www.cs.virginia.edu/sigbed/archives/2005-10/03-wese2005%20\(Pak\).pdf](http://www.cs.virginia.edu/sigbed/archives/2005-10/03-wese2005%20(Pak).pdf)
- [40] Minaiae, Afsaneh and Reza Sanati-Mehrziy, "An International Study of Embedded Systems Design in the Computer Science & Engineering Curriculums", Proceedings of Annual ASEE Conference, 2007.
- [41] Wang Ping. "Research on the Embedded System Teaching", 2008 International Workshop on Education Technology and Training, 978-0-7695-3563-0/08 2008 IEEE, DOI 10.1109.
- [42] IEEE. IEEE standard VHDL language reference manual, IEEE Std 1076-1987, March 1988.
- [43] D. E. Thomes and P. Moorby. The Verilog Hardware Description Language. Kluwer Academic, 3rd edition edition, 1996.
- [44] P. Bellows, BL Hutchings. JHDL—An HDL for reconfigurable systems. Proceedings of IEEE Workshop on FPGAs for Custom Computing Machines, 1998
- [45] The Open SystemC Initiative, www.systemc.org
- [46] Mentor Graphics, Catapult-C Synthesis: Full-Chip High-Level Synthesis, http://www.mentor.com/products/esl/high_level_synthesis/catapult_synthesis, 2010
- [47] Impulse Accelerated Technologies, Inc., Impulse Co-Developer, <http://www.impulseaccelerated.com>, 2010
- [48] Los Alamos National Laboratory, <http://www.streams-c.lanl.gov>
- [49] D. Pellerin, "Streams-Based Programming Accelerates FPGA-Based Embedded Applications," *Embedded Magazine*, pp.29-31, Nov. 2006.
- [50] Mittrion product brief, 2005, http://www.mittrion.com/press/Mittrion_product_brief_051017.pdf
- [51] National Instruments Inc., <http://www.ni.com/>
- [52] N. Halbwachs, P. Caspi, P. Raymond, and D. Pilaud. The synchronous data flow programming language lustre. Proceedings of the IEEE, 79(9):1305–1319, 1991. <http://dx.doi.org/10.1109/5.97300>
- [53] U. Banerjee, R. Eigenmann, A. Nicolau, and D. Padua. Automatic program parallelization Proceedings of the IEEE, 81(2):211–243, 1993. <http://dx.doi.org/10.1109/5.214548>
- [54] G. Berry. The effectiveness of synchronous languages for the development of safety-critical systems. White paper, Esterel Technologies, 2003.
- [55] M. Young, D. Argiro, and S. Kubica. Cantata: Visual programming environment for the khoros system. *Computer Graphics*, 29(2):22–24, May 1995. <http://dx.doi.org/10.1145/204362.204367>
- [56] Lee, E.A., "What's Ahead for Embedded Software", Vol 33, Issue 9, *IEEE Computer* [4] N. Halbwachs, P. Caspi, P. Raymond, and D. Pilaud. The synchronous data flow programming language lustre. Proceedings of the IEEE, 79(9):1305–1319, 1991.
- [57] WM Johnston, JRP Hanna, RJ Millar, "Advances in dataflow programming languages ", *ACM Computing Surveys (CSUR)*, 2004
- [58] Travis, J. & Kring, J., *LabVIEW for everyone: Graphical programming made easy and fun*. Prentice Hall: Upper Saddle River, NJ, 2007
- [59] N.Kiritisis, Y.Huang, D.Ayrapetyan, "A Multi-Purpose Vibration Experiment using LabVIEW." *American Society for Engineering Education Annual Conference & Exposition*, 2003.
- [60] C. Neuman, D. Lieberman, D. Engelberg, A. Flamholz, P.Marchese, G. Tremberger, "LabVIEW Graphical Programming in an Introductory Engineering Physics Course" *American Society for Engineering Education Annual Conference & Exposition*, 2005.
- [61] H. Abu-Mulaweh, "The Use of LabVIEW and Data Acquisition Unit to Monitor and Control Air-Conditioning Processes", Proceedings of the 2007 American Society for Engineering Education Annual Conference and Exposition
- [62] R. Pecen, M.D. Salim, and A. Zora, "A LabView Based Instrumentation System for a Wind-Solar HybridPower Station", *Journal of Industrial Technology*, Volume 20, Number 3, 2004
- [63] J. Globig, "An Interdisciplinary, LabVIEW Based, Data Acquisition and Measurements Course", Proceedings of the 2003 American Society for Engineering Education Annual Conference & Exposition
- [64] J. Garcia and P. Backer, "Assessment of LabVIEW and Multisim in the delivery of electronics laboratory content", Proceedings of the 2007 American Society for Engineering Education Annual Conference and Exposition
- [65] J. A. Anderson, R. B. Korrapati, & N. K. Swain., "Digital signal processing using virtual instrumentation." Proceedings of SPIE, Vol. 4052, April 2000.
- [66] R. B. Korrapati, & N. K. Swain., "Study of Modulation Using Virtual Instruments." Proceedings of National Conference on Allied Academies, Spring 2000.

- [67] N. K. Swain, J. A. Anderson, & R. B. Korrapati. "Computer-based Virtual Engineering Laboratory (CBVEL) and Engineering Technology Education." 2000 Annual ASEE Conference Proceedings.
- [68] L. Wells and J. Travis, *LabVIEW for Everyone*, Graphical Programming Even Made Easier, Prentice Hall, NJ 07458, 1997.
- [69] R. Pecan, M. Salim, & A. Zora, "A LabView Based Instrumentation System for a Wind-Solar Hybrid Power Station." *Journal of Industrial Technology*, Vol. 20, June 2004–August 2004.
- [70] J. Vinicius & S. Osvaldo "Using LabVIEW in a Mini Power System Model Allowing Remote Access and New Implementations." International Conference on Engineering Education, 2007.
- [71] J. Hrynuik, M. Pennington, D. Illig, J. P. Dempsey, "Freshman Engineering: An Introductory Computer Course Teaching MATLAB and LabVIEW", Proceedings of the 2008 American Society for Engineering Education Annual Conference and Exposition
- [72] D. Illig, J. Hrynuik, M. Pennington, J. P. Dempsey, "The Influence of Student Feedback and Involvement on a Course Teaching MATLAB and LabVIEW", Proceedings of the 2008 American Society for Engineering Education Annual Conference and Exposition
- [73] S. Daniels, D. Harding, M. Collura, "Introducing Feedback Control to First Year Engineering Students using LabVIEW", Proceedings of the 2005 American Society for Engineering Education Annual Conference and Exposition.
- [74] S. J. Choi, D. Shetty, J. Poduraev, W. Y. Lee, & J. H. Lee, (2003) International Collaboration and Improvement of Mechatronics Education Based on Simulation and Virtual Instruments. Available: http://www.ni.com/academic/journal_asee.htm
- [75] E. C. Eckhoff, V. M. Eller, S. E. Watkins, & R. H. Hall, (2002) Interactive virtual laboratory for experience with a smart bridge test. In proceedings of the ASEE Annual Conference & Exposition, Montre' al, Quebec, Canada, June 16-19
- [76] U. Lauterburg, (2001) LabVIEW in Physics Education. [Online] A white paper about using LabVIEW in physics demonstration and laboratory experiments and simulations. Available online at: www.clab.unibe.ch/labview/whitepaper/LV-PhysicsWPScreen.pdf.
- [77] P. J. Moriarty, B. L. Gallagher, C. J. Mellor, & R. R. Baines, (2003) Graphical computing in the undergraduate laboratory. Teaching and interfacing with Lab-View. *American Journal of Physics*, 71(10), 1104–1114 <http://dx.doi.org/10.1119/1.1582189>
- [78] N. A. Muhamad, S. Ali, "LabVIEW with fuzzy logic controller simulation panel for condition monitoring of oil and dry type transformer", *IEEE Trans. on Engineering, Computing and Technology*, Vol. 14, pp. 187-193, 2006.
- [79] Spartan-3E Starter Kit Board User Guide UG230 (v1.0) March 9, 2006, http://www.xilinx.com/bvdocs/user_guides/ug230.pdf
- [80] M. Abdulwahed, Z. K. Nagy, 2009. Applying Kolb's Experiential Learning for Laboratory Education. *Journal of Engineering Education*, Vol. 98, no 3, pp 283-294. <http://dx.doi.org/10.1002/j.2168-9830.2009.tb01025.x>
- [81] M. Abdulwahed, Z. K. Nagy, R. Blanchard, 2009. Constructivist Project Based Learning Design, a Cybernetics Approach. *Journal of Education, Information and Cybernetics*, 1(2), 2009, pp 1-8
- [82] B. Alhalabi; M. K. Hamza; S. Hsu; and N. Romance, "Virtual Labs VS Remote Labs: Between Myth & Reality", Florida Higher Education consortium Statewide Conference, 1998, ERIC document reproduction Service No. RIE ED 437 026
- [83] Heinz-Dietrich Wuttke, Karsten Henke, Nadine Ludwig, "Remote Labs versus Virtual Labs for Teaching Digital System Design", International Conference on Computer Systems and Technologies - CompSysTech' 2005, pg. IV 2.1-2.6
- [84] Giuliano Donzellini and Domenico Ponta, "The electronic laboratory: traditional, simulated or remote?" in "Advances on remote laboratories and e-learning experiences", Bilbao, 2007, pg. 223-245
- [85] LogMeIn VPN, <https://secure.logmein.com/products/hamachi>
- [86] M. Abdulwahed, Z. K. Nagy, R. Blanchard 2008a. Using Feedback Control Engineering for Analysing and Designing Effective Lecturing Model. *The 38th Annual Frontiers in Education (FIE) Conference (ASEE/IEEE Conference on Education)*, 22-25.Oct, New York, USA.
- [87] M. Abdulwahed, Z. K. Nagy, R. Blanchard 2008b, The TriLab, a Novel View of Laboratory Education. *International conference on innovation, good practice and research in engineering education, EE2008, Engineering Subject Centre, Loughborough, UK, 14-16 July 2008.*
- [88] M. Abdulwahed, Z. K. Nagy, R. Blanchard, 2009. Constructivist Project Based Learning Design, a Cybernetics Approach. *Journal of Education, Information and Cybernetics*, 1(2), 2009, pp 1-8.
- [89] M Abdulwahed, 2010. Towards Enhancing Laboratory Education by the Development and Evaluation of the TriLab: A Triple Access Mode (Virtual, Hands-on, and Remote) Laboratory. *PhD Thesis, Loughborough University, UK*
- [90] M. Abdulwahed and Z. K. Nagy (2011a). The Impact of Different Preparation Modes on Laboratory Education Learning Outcomes. Available online in the *Journal of Computer Applications in Engineering Education*. <http://dx.doi.org/10.1002/cae.20536>
- [91] M. Abdulwahed and Z. K. Nagy, 2011c. The TriLab, a Novel ICT Based Model of Laboratory Education. *Computers & Education*, 56 (1), 262–274 <http://dx.doi.org/10.1016/j.compedu.2010.07.023>
- [92] M. Abdulwahed and Z. K. Nagy, 2012. A Control Systems Engineering Approach to Designing an Effective Lecturing Model: The Implication of Feedback and Self-Construction of Knowledge. In *Innovations 2012: World Innovations in Engineering Education and Research*, ed. W. Aung, et al., iNEER, Potomac, MD, USA, pp. 29-48.
- [93] M. Abdulwahed, B. Jaworski, & A. R. Crawford, (2012). Innovative approaches to teaching mathematics in higher education: a review and critique. *Nordic Studies in Mathematics Education*, 17(2), 49–68.
- [94] M. Abdulwahed and Z. K. Nagy, 2009. The Impact of the Virtual Lab on the Hands-on Lab Learning Outcomes, a Two Years Empirical Study. The Twentieth Annual Conference of The Australian Association For Engineering Education, Adelaide, Australia 6-9 December 2009.

AUTHORS

Walid Balid is with Aleppo University, Aleppo, Syria.

Mahmoud Abdulwahed is with Qatar University, Doha, Qatar.

Imad Alrouh is with Aleppo University, Aleppo, Syria.

This article is an extended and modified version of a paper presented at the EDUCON2013 conference held at Technische Universität Berlin, Berlin, Germany from March 13-15, 2013. Received 14 May 2013. Published as resubmitted by the authors 21 May 2013.