An Approach of Project-Based Learning: Bridging the Gap Between Academia and Industry Needs in Teaching Integrated Circuit Design Course

Xiaokun Yang, *Member, IEEE*

***Abstract*—In teaching an integrated circuit (IC) design course, many benefits can be gained by offering an industry-relevant project associated with a training to a bundle of electronic design automation (EDA) tools and design methodologies. However, few open-source projects were able to cover the key qualifications needed by industry. Therefore, this article proposes an approach of project-based learning (PBL), aiming at bridging the gap between the industry needs and the learning outcomes from academia. Specifically, this article first conducts an investigation on basic qualifications necessary for entry-level IC designers. By summarizing those results as a specification, the development, implementation, and assessment to an open source project is presented to include the latest EDA tools and methodologies needed by IC design companies, as well as the fundamental knowledge and skills of the course outcomes. The effectiveness of this work is evaluated by the analysis of students' final exam results using *t*-tests. It shows that students who had participated in the project achieved higher levels of acquired knowledge to the design on ICs. Student survey and evaluation also demonstrate positive effect on student achievement with the PBL approach. Further, the public availability of this project has a big potential to offer a framework to practical courses and improve students' knowledge and skills in many topics, such as computer architecture and micro-systems.**

***Index Terms*—Industry-relevant project, integrated circuit (IC), project-based learning (PBL), system-on-chip (SoC).**

I. Introduction

PROJECT-BASED learning (PBL) is a significant requirement in teaching engineering courses, providing opportunities to learn and apply knowledge and skills tied to college and career readiness [1], [2]. Many practical projects and methodologies thus have been developed to allow students to tackle real-world challenges and encourage critical-thinking [4], [11]. And some of the projects were directly from industry [5]. However, few PBL focused on exploring the key qualifications needed by chip design companies, resulting in a gap between knowledge and skills gained in universities and the real practical experiences needed by industry.

On the industry side, the interaction between academia and electronic design automation (EDA) tool vendors has proven to be successful in the field of integrated circuit (IC) design. Traditionally, companies provide special topics lectures for universities to keep them aware of new aspects and targets in industry. For example, Synopsys provides universities with access to comprehensive curricula for Bachelor and Master Programs in IC design and EDA development [6]. Another two leading EDA vendors—Cadence and Mentor Graphics—offer the similar University Programs called Cadence University Software Program [7] and Higher Education Program for Mentor Graphics [8]. However, all teaching resources from such University Programs are available for members only. The projects are mainly based on licensed EDA tools.

Another two leading field-programmable gate array (FPGA) vendors—Altera (now Intel) and Xilinx—also provide a friendly service to academics [9], [10]. For example, Xilinx University Program (XUP) collaborates with academics to develop and deliver many lab materials. To encourage instructors and researchers use their tools and technologies, XUP also provide workshop materials which are freely accessible to academics to use in classes. The provided experiments are fundamental logic designs, target at training students to use their EAD tools, and FPGAs, which are not suitable for teaching system-level designs on an IC, or system-on-chip (SoC).

On the university side, there are two main challenges to provide PBL in line with industry requirements. First, it is difficult for academics to obtain direct experience in designing an application-specific IC (ASIC), particularly in the SoC level. Instructors would be struggling with the complicated industrial design flow involving register-transfer level (RTL) design rules and coding style, automatic verification/simulation environment, synthesis results of behavioral models, timing issues, design constraints, and so on. Second, due to the time limitation of a semester-long course, it is often not feasible to expect students to finish a project through the entire FPGA design flow involving a bundle of tools and methodologies.

To overcome the aforementioned challenges, this article proposes the development, implementation, and assessment with an open source project in teaching an IC design course. As a design-focused curriculum, the system-level verification environment can be shared to students to rapidly validate and demonstrate the SoC design. Specifically, the main contributions of the proposed work are below.

1) Conducts an investigation to the key qualifications needed for entry-level IC designers. By summarizing those results as a specification, an approach of PBL using an open source project is further proposed, enables

Manuscript received June 1, 2020; revised August 10, 2020 and November 30, 2020; accepted January 3, 2021.

The author was with the Silicon 3D Engineering, Beijing Research and Development Center, Advanced Micro Devices Inc., Beijing 100190, China. He is now with the Department of Engineering, University of Houston-Clear Lake, Houston, TX 77059 USA (e-mail: yangxia@uhcl.edu).

Digital Object Identifier 10.1109/TE.2021.3050450

to cover the fundamental knowledge, and skills needed by industry.
2) Presents the details of the implementation, including the design schedule, pedagogical approaches, and the experimental results in terms of simulation, area–speed–power estimation, and the final FPGA demonstration.
3) The assessment is based on an evaluation of the final exam using *t*-tests, showing significantly better performance for students had participated in the project. The project is publicly available in order to serve more practical courses related to the design on digital systems.

The organization of this article is as follows. In Section II, the related works are discussed. Section III conducts the investigation on basic qualifications of industry needs. Seeing the investigation result as a specification, Sections IV and V introduce the proposed PBL containing the project, schedule, and pedagogical approaches. In Section VI, the experimental results are detailed and in Section VII, the student survey and evaluation are described. Finally, Section VIII presents the concluding remarks and Section VIII-A discusses the future work of this article.

II. Related Works

Prior researches to PBL have been dedicating to effectively narrow the gap between industry and the engineering education [15], [16]. As an example, Oguz and Oguz [15] compiled a list of skills that are sought by the software companies, composed by an inspection of 50 advertisements for the software engineering position. Together with many collected information like interviews to students and recent graduates, it concluded that project experience is the best cure to narrow the gap.

A case study for using PBL combined with collaborative learning (CL) and industry best practices was presented in [17]. Through building a modular management system in an embedded systems course, it showed an improvement of teaching, learning, and student assessment processes. To the specific IC design course, in [18], an analog IC design flow was introduced with a major project—an RFIC integrated by mixer, power amplifier, and low noise amplifier using 65-nm technology.

In contrast, the IC design in the digital world is a bit different in terms of a bundle of EDA tools and FPGA demonstration. Exiting research to the digital IC design courses mainly focused on either fundamental logic level [21], [22] or software–hardware co-design platforms which are most likely for microcontroller programming and computer architecture courses [19], [20]. For instance, a flexible VHDL framework for simulation and synthesis was presented in [22], providing a simple and intuitive method to implement basic logic circuits. In [19], an FPGA-based education platform was presented to support the experiments from hardware systems to vision algorithms. It was equipped with the Xilinx PYNQ-Z2 board where a CPU host and an FPGA in the same chip are integrated. The flexibility to use the ARM processor can significantly reduce the design complexity on hardware; nonetheless, the programming experience on an ARM core is most likely the teaching outcomes of a microcontroller programming course.

Therefore, this article proposes an approach of PBL by carrying out a pure RTL design on FPGA. By researching and seeing the industry needs as a specification, the proposed PBL learning is able to cover most key qualifications necessary by industry.

TABLE I
Key Qualifications of Industry Needs

Requiremnets	Basic Qualifications
Design Flow	1. ASIC and/or FPGA Design Flow
HDL Programming	2.1 Verilog/VHDL
	2.2 Editor - VIM/Emacs/WinEdt
	2.3 Linux OS
	2.4 Scripts such as Perl, tcl, and Makefile
Simulation (Verification)	3.1 Development/implementation of test plans, verification (simulation) environments, validation components, and tests
	3.2 Simulators - Synopsys VCS/Cadence NC/ Mentor Graphic ModelSim
FPGA Demo	4.1 FPGA tool - Xilinx Vivado/Intel Quartus
	4.2 FPGA prototype and debug
Design Related	5.1 Bus architecture such as AXI and Wishbone
	5.2 SoC peripherals such as I2C, VGA, Timer
	5.3 RTL IP, Vivado/Quartus IP, and design reuse
	5.4 Basic design rules such as FSM, clock domain crossing, and FIFO
	5.5 Tradeoff between quality and design cost

III. Research on Basic Qualifications

This section conducts an investigation to the specific requirements for an entry-level chip designer, including the FPGA Hardware Engineer from Mentor Graphics, Digital IC Design (PHY) Engineer from Apple, Design Verification Engineer from AMD, and many more. The research is based on the information on Indeed.com and Linkedin.com by filtering the relevance of full-time job type and entry-level experience. After compiling the details of 20 posted job opportunities, the key qualifications needed by industry are summarized in Table I.

The ASIC/FPGA design flow is necessary as shown in the second row of this Table (or item 1). Apart from the hardware description language (HDL) (item 2.1) and one of the editor software (item 2.2), the basic Linux command (item 2.3) and one of the script languages (item 2.4) are needed because the chip design environment is usually Linux/Unix based and multiple scripts involved.

In the front-end, simulation experience, including development of test plans, simulation environment, bus function models (BFMs), scoreboards/monitors, and test cases, is required (item 3.1). In order to run simulation in RTL or netlist level, one of the simulators is a must (item 3.2). The dominant simulators in industry include Mentor Graphics ModelSim/Questa, Synopsys VCS, and Cadence NC. Additionally, FPGA demonstration is needed where students can rapidly exercise and validate their design of interest (item 4.2). The FPGA tools include Intel Quartus and Xilinx Vivado (item 4.1).

To bring real-life context and technology to the curriculum, it is imperative to devise a project with industry standards and

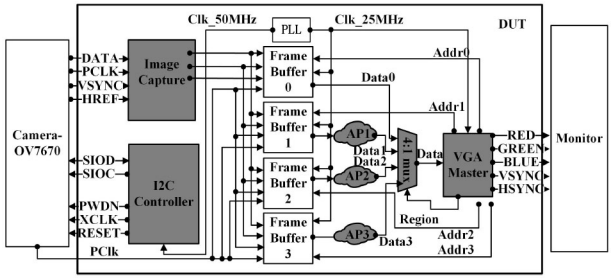

Fig. 1. Design architecture.

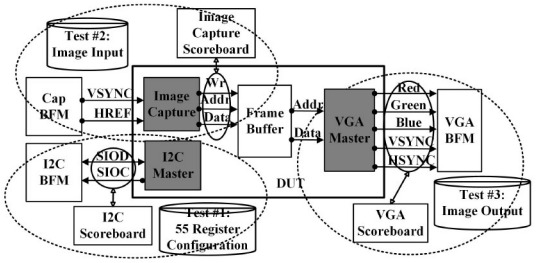

Fig. 2. Verification (simulation) environment.

protocols. As an entry-level position, first, the project experience should cover basic design methods and design rules, such as the clock domain crossing (CDC), finite-state machine (FSM), synchronous/asynchronous FIFOs (item 5.4), and the design constrains in terms of area–speed–power (item 5.5). Further, some widely used industrial protocols like bus architectures (item 5.1) and bus peripherals, such as I2C and UART, should be studied (item 5.2). The submodule implementation is usually named intellectual property (IP) level design, compared with system-level integration. All the IPs, including third-party designs and self-designs, would be integrated into an SoC (item 5.3).

IV. Overview of the Open Source Platform

As per the investigation mentioned above, this section presents an approach of PBL aiming to cover all the items listed in Table I. The experiments have run for three semesters, with 10 graduate students in Spring 2017, 12 in Fall 2017, and 13 in Spring 2019. One instructor and one teaching assistant (TA) worked to provide tutorials and training, and helped in the HDL programming and performance evaluation. The details of the project and schedule are introduced in this section.

A. Project Description

Generally, the project is based on an open-source platform published in [12], capable of capturing images by interfacing an OV7670 camera and displaying both the original images and the results of grayscale images on a VGA-interfaced monitor. The design architecture is shown in Fig. 1, mainly containing three interfaces: 1) an I2C master; 2) a VGA master; and 3) an Image Capture slave. The I2C master is used to set up the OV7670 camera. After being configured, the camera enables to send 320×240 size of images into the Frame Buffer #0 via the Image Capture interface.

To study the design tradeoff between quality of results and resource cost on FPGA, in Fig. 1, three approximate designs—AP1, AP2, and AP3, on color-to-grayscale converter are integrated into the SoC. In theory, the higher the approximation levels, the more hardware resource can be saved. To find the minimum design cost corresponding to the quality bound is the basic idea of IC design. Further, Frame Buffers 1, 2, and 3 are used to store the results from approximate design 1, 2, and 3, respectively. And then the VGA master reads data out from the four Frame Buffers and display the data pixel by pixel on a 640×480 size of monitor.

TABLE II
Project Schedule

Course	Topic	WKs	HRs
Lectures	Introduction on IC design flow, Verilog HDL, Design rules and constraints, etc.)	1-5	15
Laboratory (Mini Project)	Training: VIM, ModelSim, and Vivado; simulation env, Nexys 4 FPGA	6-7	6
Lecture	Industry specifications: AMBA AXI, I2C, VGA, and OV7670	8	3
Laboratory (Final Project)	IP design and verification; SoC integration and verification; FPGA demo and evaluation	9-13	15
Laboratory (Final Project)	Discussion and presentation, project assessments	14	3

The verification/simulation environment is shown in Fig. 2, including design-under-test (DUT), bus function models (BFMs), scoreboards, and test vectors. Specifically, each design interface is equipped with a bus function model for driving/responding the interface and a scoreboard for testing the functionalities. The environment is controllable by a tcl script to compare data between design interfaces and golden models.

B. Project Schedule

In addition to the midterm week, each full-semester course contains 14 weeks, including lectures and projects. The curriculum schedule of the Spring 2019 semester is shown in Table II. Six weeks—from the first week to the fifth week, plus the eighth week—are spent to introduce all the theories and principles. In the eighth week, specifically, the industry standards, including I2C, VGA, and OV7670 data sheet, are explained by the instructor. The interfaces and timing diagrams of theses standards are the design specification of the final project. The rest of eight weeks focus on conducting the final project. All the programming language, EDA tools, and design methodologies introduced in Section III are taught and applied by carrying out the project for a thorough and practical understanding of theoretical concepts.

V. Pedagogical Approaches and Project Implementations

In this section, the pedagogical approaches related to the PBL is discussed. Then, the details of the project implementation is further introduced.

A. Pedagogical Approaches

The final project can be divided into three phases: 1) the design phase; 2) implementation phase; and 3) testing phase. The design phase emphasizes learning industrial standards and creating the design specification; the implementation phase focuses on describing hardware and integrating SoC by Verilog HDL; and the testing phase concentrates on the functional verification and FPGA demonstration.

The main challenge of the design phase is lack of experience to create the SoC architecture with multiple industry specifications. In this phase, the instructor gives an overview to the system architecture, as well as explains the block and timing diagrams to each submodule. Much interactive discussion of the main concepts should be involved in this phase.

In the implementation phase, the big challenge for students is to think in hardware when programming with Verilog HDL. Not only the industry design rules but also good coding style should be emphasized by the instructor. Additionally, two training classes are offered to help students utilize the EDA tools, build a typical test bench, and demonstrate basic combinational and sequential circuits on FPGA.

The testing phase consists of simulation on IPs and SoC, as well as FPGA demonstration. The simulation has always been neglected to teach such courses, which is actually crucial to the front-end development of IC design flow. IP-level test bench and the final FPGA demonstration should be completed by students with instructor and TA's support. However, to build a system-level verification environment is not feasible to a semester-long and design-focused course. Therefore, the open-source test bench will be explained by the instructor and then provided to students to rapidly verify their SoC integration.

B. Project Implementations With Learning Method

The main contribution of this article is to present a PBL approach to cover items of basic qualifications in Table I. Hence, this section discusses the details of the project implementation in Table III to check all the items one by one.

In the sixth and seventh weeks, a mini project—design on a timer—is given to students to take practice for FPGA demonstration and using the EAD tools, including VIM, Mentor Graphic ModelSim, and Xilinx Vivado. Through the two-week training, students learn how to build a simple simulation environment and run the FPGA design flow. The instructor coordinates and supports students' work by offering the training tutorials and relevant references of the EDA tools.

Starting at the ninth week, the instructor helps students create the block diagram of the SoC architecture containing three submodules or IPs—the I2C master, Image Capture slave, and the VGA master. In what follows, students are split into three teams, four to five students per group. Each group completes design and simulation of one submodule (item 5.2 covered as shown in the second column). Students should finish the submodule design specification in the ninth week. And in the tenth week, students focus on the HDL programming and simulation (items 2.1–2.4, 3.1–3.2, 5.4 covered). During the submodule implementation, the teams continuously report and present their progress and results to the instructor who provides coordination and help. Within the team, the students' teamwork improves as the project progresses.

TABLE III
PROJECT IMPLEMENTATION AND KNOWLEDGE AND SKILLS GAINED FOR THE THREE GROUPS

Group	WKs & Items	Tasks	Knowledge and Skills
1, 2, 3	6-7	Mini Project: Design and verify the functionality of a Timer	Basic design with Verilog, Use of VIM, ModelSim, Vivado, and Nexys 4 FPGA
1	9-10 (items 2.1-2.4 3.1-3.2 5.2, 5.4)	Final Project (IP): Design and verify the functionality of the I2C master	Studying the I2C spec, basic combinational and sequential circuits design including FSM, counter, tri-state buffer, MUX, etc.
2		Final Project (IP): Design and verify the functionality of the Image Capture slave	Studying the Image Capture spec, basic combinational and sequential circuits design including concatenation, dff, counter, etc.
3		Final Project (IP): Design and verify the functionality of the VGA master	Studying the VGA spec, basic combinationa and sequential circuits design including FSM, counter, comparator, etc.
1, 2, 3	11-12 (items 5.1, 5.3)	Final Project (SoC): SoC Integration	System-level design, integration, and verification
1, 2, 3	13-14 (items 4.1-4.2)	Final Project (FPGA) FPGA prototype	Synthesis, placement & routing, and FPGA demonstration

In the 11th week, each team shares their submodule design to others, then integrates all the three IPs into an SoC (items 5.1 and 5.3 covered). The system-level integration and verification are performed in two weeks. The system test bench is given to students to verify the SoC interfacing and functionality. In the following 13th week, the synthesis and placement and route are implemented to obtain the final FPGA netlist (items 4.1 and 4.2 covered). Meantime, the system performance can be evaluated in terms of slice count, latency, and power consumption (item 5.5 covered). Finally, the whole project implementation, activities, and the results should be presented and discussed at the end of the semester.

VI. Results of Simulation, FPGA Demonstration, and Design Performance

In this section, the design simulation and FPGA demonstration are discussed. The results in real-life hardware schematic and resource cost can inspire and encourage students to think in hardware when coding Verilog HDL.

A. Verification/Simulation

Students test the functionality of their designs in two steps—simulation and FPGA demonstration. The IP-level simulation should be mainly completed by students by group. After that, instructor and TA involve in the system-level verification to the SoC integration. Fig. 3 shows the simulation results by using ModelSim. Through monitoring commands sent by the I2C

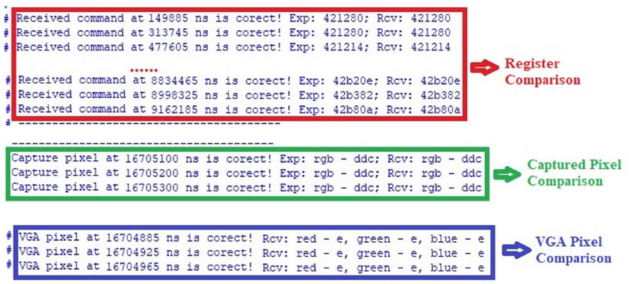

Fig. 3. Simulation results.

TABLE IV
SoC Resource Cost on FPGA

Resource Cost	LUTs	Regs	RAM	IOs	TP (mW)
SoC	774	297	106	34	220

master, the I2C interface design can be verified shown in the first box. In the second box, the design on the Image Capture slave is verified by comparing the data being stored into frame buffers and the data from golden files. As an example, the first pixel being checked is hexadecimal "ddc", including 4-bit red pixel (hexadecimal "d"), 4-bit green (hexadecimal "d"), and 4-bit blue (hexadecimal "c"). Similarly, in the third box, the design on the VGA master can be verified.

B. Area–Speed–Power Evaluation

In what follows, the HDL design is synthesized into gate-level netlist using Xilinx Vivado. Then the design performance in terms of slice count and power consumption is evaluated in Table IV. As depicted, the SoC design spends 774 slice LUTs, 297 slice registers, 106 block RAMS, and 34 IOs. Additionally, the total power (TP) consumption is 220 mW, including 102-mW static power and 118-mW dynamic power. During the back-end operation, students can learn how to estimate the chip size and power cost on FPGA. More important, it can help students match the behavioral model level design with the real hardware component, which is one of the biggest challenges for the beginners learning HDL.

C. Design Constrains

To find the optimal design corresponding to different quality bound is an important skill for IC designers. The knowledge, skills, and experience of industry design rules and coding style are thus crucial to students to describe a hardware system. As a case study, therefore three approximations of design on color-to-grayscale image converter are given to students to test the quality-resource tradeoff [13]. Table V shows the synthesis results of the exact design (EX) and approximate design 1, 2, and 3 (AP1, AP2, and AP3). It can be observed that the higher approximation of the design achieves higher maximum operational frequency (MOF) and more resource saving in terms of slice count and power dissipation.

TABLE V
Resource Cost on Color-to-Grayscale Converter

Designs	LUTs	Regs	MOF (MHz)	Power (mW)
EX	264	382	230.264	21
AP1	228	342	292.466	17
AP2	106	172	353.788	15
AP3	22	23	899.402	5

D. FPGA Demonstration

Finally, the SoC design and integration are demonstrated and presented in the last laboratory session of the semester, with a documentation report, including project design, implementation, and experimental results.

VII. Assessment

The theoretical and practical knowledge and skills gained via PBL learning are mainly estimated using the exam results. The exam was taken by 35 graduate students, 22 students in Spring 2017 and Fall 2017, and 13 students in Spring 2019. The exam results are used to evaluate the following hypothesis: after completing the project, students have increased their theoretical and practical knowledge on the key qualifications of the industry needs. The exam results of students who did or did not participate in the project are statistically compared by t-tests to test the hypothesis. Additionally, a student survey was taken in Spring 2019 to show the students' response. Finally, the student evaluations over the three semester are analyzed to depict the improvement of the course by using PBL.

A. Exam on Knowledge and Skills

The final exam can be divided into two parts: the knowledge and skills on the design of ICs. The first part is used to evaluate the most important knowledge needed by an entry-level IC designer, and the second part is created for testing the practical skills by using the theory. The first part is 40% of the final grade and the second part is 60%.

As one of the most important learning outcomes, the timing issue for designing digital circuits is tested in the first part. Actually, the timing constraint is also one of the most common interview questions in industry, which somehow determines success or failure of the chip tape-out. For example, the true/false path and the setup/hold time constraint are tested in Fig. 4(a) and (b). Students should understand that the B-D-E-F-G path is the longest path in Fig. 4(a) but not the true path. Hence, the critical path for finding the MOF in Fig 4(b) should be the C-E-F-G path.

One of the biggest challenges for the beginners learning HDL is to map the RTL design with a specific piece of hardware. As an example in the second part, in Fig. 4(c), there are three similar Verilog descriptions using conditional signal assignment to describe hardware. However, the first assignment description would be synthesized as a two-input multiplexer, the second design would be converted into a latch, and the third design would be a tri-state buffer.

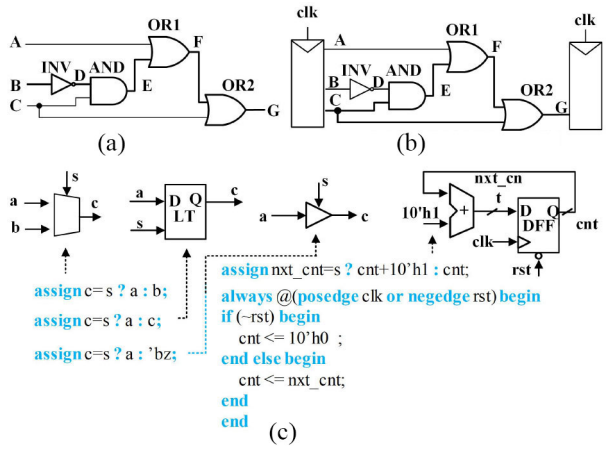

Fig. 4. Sample exam questions—fundamental knowledge. (a) False path. (b) Setup and hold contraints.

TABLE VI
EXAM RESULT (SPRING 2017 VERSUS FALL 2017)

Exam Studies / Weights	Semester	Grade Av.	Grade S.D.	Effect size (d)	t
Konwledge(40%)	S 2017	3.24	0.56	0.15	0.33 ($p<.05$)
	F 2017	3.15	0.67		
Skill(60%)	S 2017	3.10	0.23	0.26	0.55 ($p<.05$)
	F 2017	3.04	0.24		

The fourth example in Fig. 4(c) is to test the Verilog coding style. In this case, the combinational design using the "assign" block would be converted into a ten-bit accumulator, and the sequential circuit using the "always" block would be specified as a 10-bit register. Together the design is a ten-bit counter by clock cycles. The design on counters is suggested to separate the combinational and sequential descriptions in order to make the HDL design readable and understandable.

B. Exam Results

Final grades are ranked on a scale of 1–5, where 1 is unsatisfactory, 2 satisfactory, 3 neutral, 4 good, and 5 excellent. The overall student grade of the two-part test is the simple average of the results. Students' background knowledge is assessed based on the results of previously taken courses. *t*-tests show that the initial knowledge of students enrolled the course in Spring 2017 and Fall 2017 is similar as seen in Table VI. The maximum t value is 0.55 and the effect size is 0.26.

This article focuses on the comparison between results of students taking the PBL (in Spring 2019) and students not taking the PBL (in Spring 2017 and Fall 2017). The grade average and standard deviation of the exam are shown in the third and fourth columns in Tables VII and VIII. Further, the result of *t*-tests is shown in the sixth column.

First, in the second and third rows in Table VII, it shows that the average grades for knowledge testing are 3.24 (S.D. 0.56) and 3.51 (S.D. 0.59), respectively, in Spring 2017 and Spring 2019. There is a 0.27 difference between the two groups of students due to the practical skills achieved by the approach of PBL. The major difference occurs at the test on practical skills

TABLE VII
EXAM RESULT (SPRING 2017 VERSUS SPRING 2019)

Exam Studies / Weights	Semester	Grade Av.	Grade S.D.	Effect size (d)	t
Konwledge(40%)	S 2017	3.24	0.56	0.47	0.96 ($p<.05$)
	S 2019	3.51	0.59		
Skills(60%)	S 2017	3.10	0.23	2.46	4.74 ($p<.05$)
	S 2019	3.67	0.24		

TABLE VIII
EXAM RESULT (FALL 2017 VERSUS SPRING 2019)

Exam Studies / Weights	Semester	Grade Av.	Grade S.D.	Effect size (d)	t
Konwledge(40%)	F 2017	3.15	0.67	0.58	1.22 ($p<.05$)
	S 2019	3.51	0.59		
Skills(60%)	F 2017	3.04	0.24	2.68	5.43 ($p<.05$)
	S 2019	3.67	0.24		

which is shown in the fourth and fifth rows. It can be observed that the average grades for students in Spring 2017 and Spring 2019 are 3.10 (S.D. 0.23) and 3.67 (S.D. 0.24), respectively, showing a significant 0.57 average grade increase between the two groups ($p < 0.05$) in favor of students that participated in the PBL in Spring 2019.

Likewise, the performance between students in Fall 2017 and Spring 2019 is evaluated in Table VIII. It shows similar differences between students participated in the PBL and did not participate. In the knowledge tests, the average grade of students in Fall 2017 is 3.15 (S.D. 0.67) and that of students in Spring 2019 is 3.51 (S.D. 0.59). The significant difference is in the practical skills test. The average grade of students in Fall 2017 is 3.04 (S.D. 0.24), whereas that of students in Spring 2019 is 3.67 (S.D. 0.24), showing a notable difference of 0.63 ($p < 0.05$). Therefore, it can be stated that the students who worked on the project achieve better results than those who did not ($p < 0.05$).

According to the *t*-test results, the significance probability (p-value) is less than the significance level (0.05), indicating that there is a statistically significant difference in the results between those who participated in the PBL and those who did not, particularly to the test on practical skills.

C. Student Survey

A student survey was taken in Spring 2019 to evaluate the PBL effectiveness. In the first column of Table IX, it shows the eight questions to students, and in the first row five answers can be chosen on a scale of A–E, where A is strongly agree, B is agree, C is unsure, D is disagree, and E is strongly disagree.

Specifically, for the first question, 84.62% students strongly agree that the course design is helpful to bridge the gap between academy and industry needs. For the second and third questions, 61.54% and 84.62% students strongly agree that the lectures and training are helpful to the final project. Additionally, for the fourth and fifth questions, more than 75% students strongly

TABLE IX
STUDENT SURVEY

Survey Questions	A (Strongly agree)	B (Agree)	C (Unsure)	D (Disagree)	E (Strongly Disagree)
Q1: Overall, do you think the design of the class is helpful to bridge the gap between academy and industry needs in integrated circuit design?	84.62%	15.38%	0	0	0
Q2: Do you think the five-week lecture is helpful to know about the latest technologies to carry out the final project?	61.54%	38.46%	0	0	0
Q3: Do you think the two-lecture training with the integrated circuit design flow, EDA tools, simulation environment is helpful to the final project?	84.62%	15.38%	0	0	0
Q4: Did the project work, including the IP tree and system-on-chip (SoC) tree, help you understand and learn the basics of design and implementation of a SoC?	76.92	23.08			
Q5: Did the Project-Based Learning (PBL) during the class help your understanding the industrial FPGA design flow?	84.62%	15.38%	0	0	0
Q6: Did you find the final project interesting?	92.30%	7.70%	0	0	0
Q7: Did the project help you to extend your multidisciplinary knowledge?	61.54%	38.46%	0	0	0
Q8: Was the project work easy to finish?	23.08%	30.77%	23.08%	15.38%	7.70%

agree that the project is helpful to understand the FPGA design flow. For the sixth question, 92.3% students strongly found the project interesting, and for the seventh question, 61.54% students strongly agree that the project helps them to extend their multidisciplinary knowledge. The objective of the PBL is to make less than 25% students feel neither too easy nor too hard to complete the project. For the last question, it can be observed that the difficulty of the project is appropriate to graduate student in a semester-long course.

D. Student Evaluation to Academic Abilities

In this section, the difference between the courses over the three semesters are discussed. In general, the teaching materials, assignments, and exams are equally performed through the three semesters. In Spring 2019, however, a major project is added and executed spanning over eight weeks to carry out the entire FPGA design flow.

At the end of each semester, students evaluate the course including the difficulty of the course and the level of knowledge/understanding gained from the course. As shown in the second to the fourth rows in Table X, 38% students in Spring 2019 feel that the course is hard due to the additional work to the project, comparing to 29% and 25% students in Fall 2017 and Spring 2017. Though it is the most difficult semester, 92% students in Spring 2019 rate the level of knowledge/understanding gained from this course to be high as shown in the sixth row, compared to the percentage of 86% and 90% students in the Fall 2017 and Spring 2017. It is a solid evidence to demonstrate that the improvement of the course is principally due to the PBL learning industrial flow.

E. Google Analytics Results

In order to serve researchers, engineers, and students in IC design field, the project has been made publicly available on https://sceweb.sce.uhcl.edu/xiaokun/OpenIC/ since December 2019. As the results on Google Analytics [14], shown in Fig. 5, the open-source Web page has resulted in 300 new users and 1759 page views until March 2020. It has been widely browsed

TABLE X
STUDENT EVALUATION TO THE KNOWLEDGE/UNDERSTANDING GAINED FROM THE COURSE

Compared to my expectations, this course was	Hard	As expected	Easy	
	38%	62%	0	S 2019
	29%	71%	0	F 2017
	25%	60%	15%	S 2017
How would you rate the level of knowledge or understanding you gained from this course?	High	Adequate	Low	
	92%	8%	0	S 2019
	86%	14%	0	F 2017
	90%	10%	0	S 2017

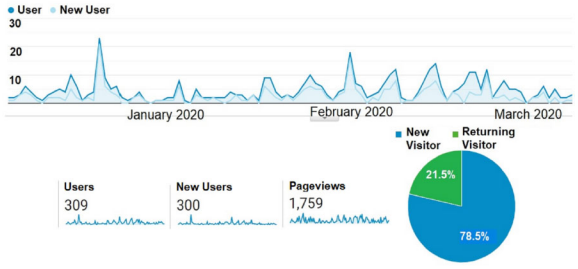

Fig. 5. Results from Google Analytics.

and downloaded from users in more than 20 countries during the last three months. The preliminary result demonstrates that this project has a potential to be widely used in teaching many advanced level courses in the future, such as advanced digital system design, verification methodology, FPGA design and verification, and many more.

VIII. CONCLUSION

In academia, to provide engineering students opportunities to engage with industry-relevant projects is a big challenge due to a lack of research to the real-world requirements in industry and the time constraint to set up a similar design environment to industry counterparts. Therefore, this article presents

a PBL approach by using an open source platform, targeted at covering basic qualifications for industry needs in the field of IC design. The project is appropriate for graduate students in computer engineering program. It provides an opportunity to discuss and solve practical IC design problems, work in teams, and achieve a joint result by PBL. The exam results show significantly better grades for students who participated in the project. More important, the presented PBL platform is made publicly available, offering a good framework to implement practical courses related to the multidisciplinary field of computer engineering and improve students' knowledge and skills in topics of computer architecture, digital circuit/system design, embedded system, etc.

A. Future Work

Recently, online learning platforms and open source projects are becoming increasingly popular for teaching and managing engineering courses [23], [24]. For example, Rodriguez-Sanchez *et al.* [23] presented some international collaborative projects that students online from Indian and Spanish are able to work together with open source tools. Similarly, in [24], new approaches with e-learning material, including Web-based animations, lecture videos, tools for graphical simulation, and remote labs, were developed and described in the literature.

Under this context, future work of this article will focus on creating a network laboratory with online-available platforms and tools, as well as effective remote communication and teamwork. For example, the Amazon Elastic Compute Cloud (Amazon EC2) can be used to develop FPGA applications with Amazon Web Services (AWS) cloud [26], and many free-registration platforms like FPGA Accelerator Research Infrastructure Cloud (FAbRIC) [27] and open-source projects and tools on Github.com like Digital simulator [25] can be used to expand the online platform.

REFERENCES

[1] D. Tsybulsky and Y. Muchnik-Rozanov, "The development of student-teachers' professional identity while team-teaching science classes using a project-based learning approach: A multi-level analysis," *Teach. Teach. Educ.* vol. 79, pp. 48–59, Mar. 2019.

[2] J. Larson, S. S. Jordan, M. Lande, and S. Weiner, "Supporting self-directed learning in a project-based embedded systems design course," *IEEE Trans. Educ.*, vol. 63, no. 2, pp. 88–97, May 2020.

[3] K. D. Pham, M. Vesper, D. Koch, and E. Hung, "EFCAD—An embedded FPGA CAD tool flow for enabling on-chip self-compilation," in *Proc. IEEE 27th Annu. Int. Symp. Field-Program. Custom Comput. Mach. (FCCM)*, San Diego, CA, USA, Jun. 2019, pp. 5–8.

[4] M. Mykhailova and K. M. Svore, "Teaching quantum computing through a practical software-driven approach: Experience report," in *Proc. 51st ACM Techn. Symp. Comput. Sci. Educ. (SIGCSE)*, Feb. 2020, pp. 1019–1025.

[5] N. A. M. Salleh and K. M. Yusof, "Industrial based final year engineering projects: Problem based learning (PBL)," in *Proc. 7th World Eng. Educ. Forum (WEEF)*, Kuala Lumpur, Malaysia, Sep. 2018, pp. 782–786.

[6] *Synopsys Electronic Design University Program*. Accessed: Apr. 18, 2019. [Online]. Available: https://www.synopsys.com/community/university-program

[7] *Cadence University Software Programs*. Accessed: Apr. 18, 2019. [Online]. Available: https://ip.cadence.com/about/university-program

[8] *Mentor Graphics Higher Education Program*. Accessed: Apr. 18, 2019. [Online]. Available: https://www.mentor.com/company/higher_ed

[9] *Xilinx University Program*. Accessed: Apr. 18, 2019. [Online]. Available: https://www.xilinx.com/support/university.html

[10] *Intel FPGA University Program*. Accessed: Apr. 18, 2019. [Online]. Available: https://software.intel.com/en-us/fpga-academic

[11] M. Božanić and S. Sinha, "A survey of current trends in master's programs in microelectronics," *IEEE Trans. Educ.*, vol. 61, no. 2, pp. 151–157, May 2018.

[12] X. Yang, Y. Zhang, and L. Wu, "A scalable image/video processing platform with open source design and verification environment," in *Proc. 20th Int. Symp. Qual. Electron. Design (ISQED)*, Santa Clara, CA, USA, Apr. 2019, pp. 110–116.

[13] Y. Zhang, X. Yang, L. Wu, and J. H. Andrian, "A case study on approximate FPGA design with an open-source image processing platform," in *Proc. IEEE Comput. Soc. Annu. Symp. VLSI (ISVLSI)*, Miami, FL, USA, Sep. 2019, pp. 372–377.

[14] M. Yamba-Yugsi, S. Luján-Mora, and H. Pacheco-Romero, "Using Google analytics to analyze users of a massive open online course," in *Proc. Int. Conf. Inf. Syst. Comput. Sci. (INCISCOS)*, Quito, Ecuador, Apr. 2020, pp. 280–285.

[15] D. Oguz and K. Oguz, "Perspectives on the gap between the software industry and the software engineering education," *IEEE Access*, vol. 7, pp. 117527–117543, 2019.

[16] D. Akdur, "The design of a survey on bridging the gap between software industry expectations and academia," in *Proc. 8th Mediterr. Conf. Embedded Comput/(MECO)*, Budva, Montenegro, Jul. 2019, pp. 1–5.

[17] M. C. Rodriguez-Sanchez, A. Torrado-Carvajal, J. Vaquero, S. Borromeo, and J. A. Hernandez-Tamames, "An embedded systems course for engineering students using open-source platforms in wireless scenarios," *IEEE Trans. Educ.*, vol. 59, no. 4, pp. 248–254, Nov. 2016.

[18] L. Albasha and O. Hammi, "Introducing industrial design flow of an RFIC chip to a graduate course: Building the ecosystem and bridging the gap between industry and academia," *IET Circuits Devices Syst.*, vol. 11, no. 4, pp. 299–303, Jul. 2017.

[19] T. Wu, Y. Wang, W. Shi, and J. Lu, "HydraMini: An FPGA-based affordable research and education platform for autonomous driving," in *Proc. Int. Conf. Connected Auton. Driving (MetroCAD)*, Detroit, MI, USA, Jul. 2020, pp. 45–52.

[20] T. McGrew and E. Schonauer, "Framework and tools for undergraduates designing RISC-V processors on an FPGA in computer architecture education," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Las Vegas, NV, USA, Apr. 2019, pp. 778–781.

[21] L. Shanshan and Y. Shiqiang, "Digital logic experiment teaching based on experimental platform," in *Proc. 12th Int. Conf. Comput. Sci. Educ. (ICCSE)*, Houston, TX, USA, Oct. 2017, pp. 33–37.

[22] J. Durre and H. Blume, "SF^3: A scalable and flexible FPGA-framework for education and rapid prototyping," in *Proc. IEEE Int. Conf. Microelectron. Syst. Educ. (MSE)*, Jun. 2017, pp. 35–38.

[23] M. C. Rodriguez-Sanchez, P. Chakraborty, and N. Malpica, "International collaborative projects on digital electronic systems using open source tools," *Comput. Appl. Eng. Educ.*, vol. 28, no. 4, pp. 792–802, Jul. 2020.

[24] M. Winzker and A. Schwandt, "Open education teaching unit for low-power design and FPGA image processing," in *Proc. IEEE Front. Educ. Conf. (FIE)*, Covington, KY, USA, Mar. 2020, pp. 1–9.

[25] H. Neemann. (2014). *Digital*. [Online]. Available: https://github.com/hneemann

[26] (2020). *Amazon EC2*. [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

[27] (2018). *FPGA Accelerator Research Infrastructure Cloud (FAbRIC)*. [Online]. Available: https://wikis.utexas.edu/display/fabric/Home

Xiaokun Yang (Member, IEEE) received the Ph.D. degree from the Department of Electrical and Computer Engineering, Florida International University, Miami, FL, USA, in Spring 2016.

From 2007 to 2012, he has worked as an ASIC Design and Verification Engineer with Advanced Micro Devices, China Electronic Corporation, and PowerLayer MicroSystems, Beijing, China. He has more than 15 years of experience in integrated circuit design and verification. He is currently an Assistant Professor with the College of Science and Engineering, University of Houston-Clear Lake, Houston, TX, USA. As the first author and corresponding author, he has published more than 50 papers, including three patents, more than 15 peer-review journals, and more than 30 prestigious international conferences. His research interests include HW/SW co-design on FPGA, FPGA acceleration on AI/ML, and system-on-chip architecture.

Dr. Yang has served on several editorial boards and journal reviewers, including the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, and the IEEE TRANSACTIONS ON EDUCATION, and numerous conference committees and section chairs, such as ISVLSI and ISQED.