Application of Open-Hardware-Based Solutions for Rapid Transition From Stationary to the Remote Teaching Model During Pandemic

A. Bekasiewicz, *Senior Member, IEEE*, B. Pankiewicz, M. Wojcikowski, M. Klosowski, and S. Koziel, *Senior Member, IEEE*

Abstract*—*Contribution: **While research indicates usefulness of remote laboratories in teaching of digital hardware systems, their main application is to complement stationary classes. This work presents a low-cost, scalable architecture that supports rapid transformation of teaching to a model based solely on remote access mechanisms.**

Background: **Adaptation of online laboratory solutions from the literature to en-masse teaching of digital circuits is time-consuming and expensive. Solutions that permit cheap and rapid conversion of courses to remote environments seem to be of high value, especially when social distancing renders direct teaching impossible.**

Intended Outcomes: **Demonstration of a flexible and cheap architecture that permits rapid transformation of digital circuits laboratories to the remote environment. Validation of system's performance and usefulness on a large group of students.**

Application Design: **Remote digital circuits laboratories from the literature are designed to complement stationary classes. For successful teaching during pandemic, a low-cost, flexible, and efficient solution to online laboratory based on commonly available technologies is required.**

Findings: **The proposed remote laboratory architecture enables rapid conversion of on-site teaching to online model, while supporting *in-situ* upgrades and functionality enhancements. The presented solution proved to be a convenient substitute for conventional laboratories during pandemic.**

***Index Terms*—Digital circuits design, low-cost remote architecture, open-hardware, remote laboratories, teaching during pandemic.**

I. Introduction

The spread of COVID-19 pandemic across the world [1], although not unexpected, exposed insufficient preparation of many universities for conducting en-masse teaching using the remote environments and tools. According to EUROSTAT, the number of European households with broadband access to the Internet exceeds 89% [2]. From this perspective, suspension of classes realized in a stationary model cannot be considered as a serious limitation for sustained teaching. In fact, for lectures, or exercises—that involve rather one-directional communication between the lecturer and the students—a rapid transition to teaching at the distance can be achieved with the aid of readily available communication tools [3]–[7]. Moreover, the same solutions can be used to convert conventional courses into a more interactive ones in order to increase students' engagement but also to improve understanding of the material [8]–[11].

In the case of laboratory classes, transition from stationary to the remote teaching model is much more challenging. One of the main difficulties stems from the specific goal of the labs, i.e., a learn-by-doing training based on working with a real, often specialized, hardware. Hence, lack of direct access to the hardware challenges the entire concept of laboratory classes. Development of remotely controlled labs is a popular topic in the research community [12]–[19]. To date a number of solutions oriented toward enhancement of the learning experience through remotely accessed laboratories have been discussed [12]–[27]. Creative approaches to control the hardware not intended to work in the remote environments have also been reported [20]–[27]. For the field-programmable gate array (FPGA) courses, the successful laboratory should provide access to the required computer-aided design (CAD) packages, as well as to the programming interface and the input/output (I/O) components of the evaluation hardware. In [12], a single-user architecture that supports online validation of locally preimplemented hardware description layer (HDL) codes has been proposed. The method provides the remote access to peripherals using a virtual block component. A multiuser system which enables FPGA programming through a dedicated server has been discussed in [13]. The access to hardware I/O and the user feedback have been provided using a combination of custom HDL modules and a Webcam interface. In [14], the concept of controlling the hardware using the dedicated HDL codes is discarded. Instead, the I/O components are remotely controlled through a custom hardware connected to the evaluation board. Another approach, where a microcontroller is used to provide nearly seamless interface with the FPGA hardware, has been considered in [18]. However, this system provides only the emulation of buttons, which narrows down its applicability to relatively simple experiments.

Manuscript received July 13, 2020; revised November 26, 2020; accepted December 6, 2020. Date of publication January 8, 2021; date of current version August 4, 2021. *(Corresponding author: A. Bekasiewicz.)*

A. Bekasiewicz, B. Pankiewicz, M. Wojcikowski, and M. Klosowski are with the Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, 80-233 Gdansk, Poland (e-mail: bekasiewicz@ru.is).

S. Koziel is with the Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, 80-233 Gdansk, Poland, and also with the Department of Engineering, School of Technology, Reykjavik University, 101 Reykjavik, Iceland.

Digital Object Identifier 10.1109/TE.2020.3043479

The above discussed architectures are unsuitable for rapid conversion of stationary FPGA laboratories to the remote environment. One of the reasons is that these systems do not rely on off-the-shelf components, or already existing solutions. Instead, they are optimized to support customized Web-based interfaces, specific FPGA hardware, and complex I/O interfaces [12]–[15]. Their development normally involves laborious debugging/testing which contradict the whole concept of making the system ready to work in a matter of days, rather than weeks, or months. Furthermore, the mentioned solutions shift the burden of CAD setup to the user (here, student) while reducing the role of the remote platform merely to the environment for experimental validation of HDL codes [12]–[18]. Finally, the laboratories from the literature significantly provide access to the hardware only for a short period of time [12]–[14]. This makes the identification of errors—resulting from differences between behavior of simulated programs and codes implemented on the FPGA chip—very difficult and, consequently, challenges the entire concept of working with the hardware [12]–[14]. Mentioned limitations and difficulties stem from the nature of the discussed architectures. Their main objective is to complement stationary teaching model by increasing the involvement of students in realization of the course. In this regard, the development of an end-to-end system supporting a learn-by-doing concept in a fully remote environment remains an open problem.

User-friendly and reliable system for full-time remote FPGA labs should be based on well tested, widely available, and easy to set-up solutions. Moreover, it should provide remote access to both the evaluation boards and the software necessary for realization of experiments. Such a setup ensures that students can focus on realization of the experiments rather than the preparation of the environment. Furthermore, to maintain low cost and compatibility with the existing teaching programs, the remote laboratory should support the same hardware that is used for stationary courses. Finally, the system should provide a reliable emulation of I/O interfaces for a wide-variety of evaluation boards and across multiple experiments.

In this work, a low-cost remote FPGA laboratory was presented. The proposed system consists of laboratory equipment used for stationary courses that was modified to support learning-at-a-distance. The goal of the lab is to provide possibly seamless experience to the users. This was achieved using widely available solutions, such as online calendars, remote desktop protocols, open-hardware microcontroller-based I/O interfaces, and Webcams. The cost of the hardware required for conversion of the laboratory was below $40 per stand. The system was implemented on a total of 10 stands and used in a 24/7 mode for over two months of the summer semester of the year 2020. The presented architecture was used for realization of four courses for over 200 undergraduate students. Qualitative and quantitative evaluation of the system was performed based on over 180 returned surveys, as well as the system-access logs. As indicated in the assessment results, the proposed remote laboratory was not only essential to carry on classes during the COVID-19 pandemic but also well suited for students' needs. Furthermore, the extensive feedback obtained from the students provides important information on possible improvements that can be introduced to the future solutions of remote laboratories.

II. Remote FPGA Laboratory: Architecture

Design of the remote FPGA laboratory is a subject to multiple requirements concerning flexibility, efficiency, and low-cost. Another factor—important for rapid conversion of the lab to the remote environment—involves compatibility of the system with the existing teaching programs. Finally, both the access and interactions with the hardware should be possibly seamless. In this section, the architecture oriented toward addressing the mentioned design

A. Background and Assumptions

It has been assumed that the architecture of the proposed laboratory should support: 1) 24/7 access to a large number of stands; 2) heterogenic evaluation boards; 3) all FPGA experiments assumed in the teaching programs for undergraduate students; 4) remote access to all of the required software packages; 5) easy reconfiguration of the stands; and 6) simple feedback mechanism for observation of boards' behavior. Furthermore, the laboratory should provide flexible access to the hardware based on a simple scheduling mechanism, whereas the remote execution of the experiments should be possible with no-to-minimal modifications of HDL codes (as compared to stationary labs). Finally, the lab should be organized in a way that allows for assessment of exercises realized by students without the need of having a real-time contact with the teacher.

B. Architecture of the Laboratory

The proposed remote laboratory was designed according to the above-mentioned requirements. The latter promote simple and cheap solutions that can be quickly implemented. To reduce the cost and development time, the laboratory was optimized toward utilization of already available hardware. The conceptual illustration of the considered architecture is shown in Fig. 1. It consists of three main components: 1) clients; 2) gateway; and 3) servers.

The client (or student's machine) plays the role of a terminal that connects through the gateway to the server (i.e., laboratory stand) located at the University. The connection can be established using a personal computer (PC), tablet, or even a smartphone. At first glance, the use of the smartphone as the client seem dubious due to small screen and lack of the keyboard or mouse. However, handhelds can easily support external hardware such as input devices or monitors though the universal serial bus (USB) interface which makes them comparable in functionality with PCs.

The main goal of the gateway is to provide a port forwarding mechanism that facilitates the connection with the laboratory stands. The gateway is based on an open-source Linux system and implements the forwarding using a set of standard libraries. From user's perspective, the connection to the laboratory boils down to specification of gateway's Internet address, selection of the appropriate port (that redirects the client to the selected server machine), and authentication.

The server machine represents an actual hardware used for realization of the laboratory tasks. Each server machine is a Windows-based system that contains the necessary CAD packages and other software required for realization of the experiments. Moreover, it is connected with at least one evaluation board (the architecture supports heterogenic FPGA hardware) along with an external interface required to mimic the behavior of boards' I/O ports. A detailed description of the I/O interface is provided in Section II-C. The selected evaluation boards are connected to the external hardware required for realization of more complex experiments. Furthermore, each laboratory stand is equipped with the Webcam that monitors behavior of the board and/or connected peripheral components. The camera can controlled using a dedicated applet, or a Windows-based app [33].

The interaction with the server machine is realized using a remote desktop protocol (RDP) [34], [35]. RDP is an off-the-shelf technology that can establish a graphical-interface-based connection with the Windows-based remote machine. The RDP clients are not only available for free but also well supported by all popular operating systems, including Windows, Linux, macOS, Android, or iOS. Owing to widespread availability, RDP can be used for connection with the laboratory stands using various client machines. Such flexibility is crucial for making remote labs accessible to a possibly large group of students, regardless of their computer skills or available hardware. It is worth noting that, compared to other remote access solutions, such as, e.g., Chrome remote desktop [36], simple protocol for independent computing environments (SPICE) [37], virtual network computing (VNC) [38], or TeamViewer [39], RDP offers a variety of well supported and easy to configure clients/servers which make it particularly useful, even as a long-term solution to the remote laboratories.

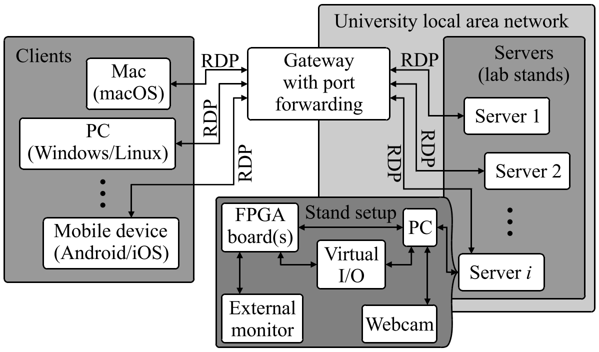


Fig. 1. Architecture of the presented remote laboratory.

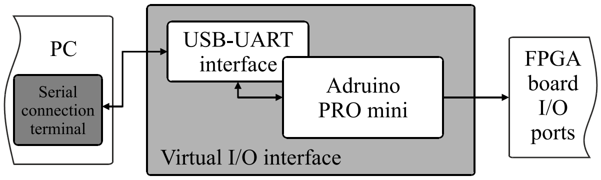


Fig. 2. Conceptual illustration of the virtual I/O interface. The device represents a transparent middleware between the FPGA and the PC.

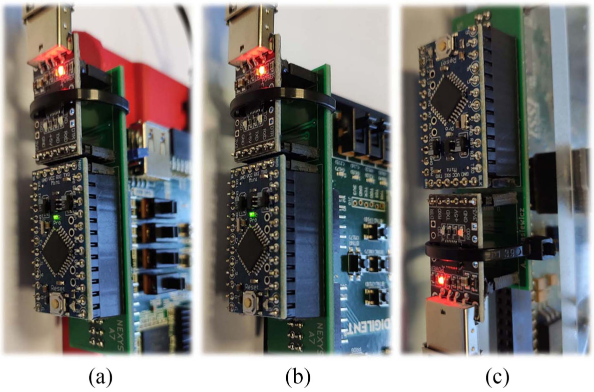


Fig. 3. Photographs of the remotely controlled I/O interfaces connected to: (a) Zybo Z7, (b) Nexys A7, and (c) Spartan 3 boards.

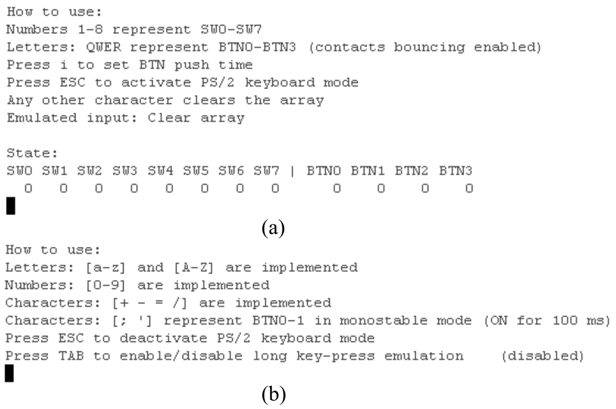
C. Open-Hardware-Based Virtual I/O

Conceptual illustration of the virtual I/O interface to the FPGA boards is shown in Fig. 2. The device consists of a 3.3 V Arduino PRO mini, an open-source hardware board based on the Atmega 328P microcontroller with 8-MHz clock [28]. The I/O ports of the device are directly connected to the multipurpose extension ports of the FPGA. Communication with the microcontroller is realized through a USB to universal asynchronous receiver–transmitter (UART) interface based on a Silicon Labs CP2102 chip [29]. The main reason for selecting open-hardware-based solutions is their low price driven by availability of many compatible devices (so-called clones). Application of the microcontroller as an external hardware for virtualization of boards' I/O interfaces is crucial for maintaining high similarity between the experiments conducted in the remote and stationary environments. Eventually, the only difference between the HDL codes validated locally and at-a-distance is the definition of FPGA I/O ports in constraint files. The latter, however, are provided in the description of exercises and thus the change does not affect the complexity of experiments.

The microcontroller is compatible with Digilent's evaluation boards, such as Nexys A7, Zybo Z7-20, and Basys 3. Older hardware with Spartan 3 FPGA chips is also supported [30]. For robust connection, the I/O interfaces are connected to the FPGA through the custom printed circuit boards (for photographs, see Fig. 3). The device can be used to control the programmed FPGA boards in a variety of ways, including emulation of switches, buttons, and personal system/2 (PS/2) keyboard [32], all with a range of adjustable settings, such as mono- and bi-stable operation, emulation of connectors bouncing [31], or variable typematic rates [32]. The interface is controlled through the UART terminal executed at the server machine. The terminal window for the virtual I/O interface is shown in Fig. 4.

III. Implementation of the Laboratory

The important advantage of the presented laboratory architecture is its versatility in terms of supported experiments.



```
How to use:
Numbers 1-8 represent SW0-SW7
Letters: QWER represent BTN0-BTN3 (contacts bouncing enabled)
Press i to set BTN push time
Press ESC to activate PS/2 keyboard mode
Any other character clears the array
Emulated input: Clear array

State:
SW0 SW1 SW2 SW3 SW4 SW5 SW6 SW7 | BTN0 BTN1 BTN2 BTN3
  0   0   0   0   0   0   0   0      0    0    0    0
█
```

(a)

```
How to use:
Letters: [a-z] and [A-Z] are implemented
Numbers: [0-9] are implemented
Characters: [+ - = /] are implemented
Characters: [; '] represent BTN0-1 in monostable mode (ON for 100 ms)
Press ESC to deactivate PS/2 keyboard mode
Press TAB to enable/disable long key-press emulation    (disabled)
█
```

(b)

Fig. 4. Terminal windows used for controlling the virtual I/O interface: (a) buttons/switches mode and (b) keyboard emulator mode.

The FPGA program realized at the University in the summer semester comprises of a total of four undergraduate courses at various advancement levels. This section is devoted to discussion of the exercises, assessment methods, and the concept of working in the environment.

A. Exercises

The FPGA courses conducted at the University include nearly 30 different laboratory exercises. Collectively, their goal is to familiarize students with the concepts related to design of combinatorial and sequential circuits but also the implementation of serial data transmission standards, interfaces to the PS/2 keyboard, ways of controlling various displays, or generating signals for a video graphics array (VGA) monitor. Other exercises involve programming of PicoBlaze and MicroBlaze processors, application of FPGA for testing of advanced encryption standard (AES)-based ciphering, or dynamic generation of graphics on the monitor. Furthermore, the laboratory supports exercises that introduce the students into implementation of Linux-based servers on the FPGA boards [30], or modeling of the hardware components using a SystemC environment [40]. Some of the exercises are oriented toward design of embedded systems composed of intellectual property (IP) cores and development of C-based applications that make use of them.

B. Assessment

One of the main challenges related to development of the presented laboratory was conception of a simple and robust, yet fair system for remote assessment of the realized exercises. The presented assessment process is based on a simple system where communication between the student and the instructor is realized either through an email or a videoconference. The system works as follows. To obtain the grade, the student submits a report containing: the source codes in HDL format, the description of the exercise and a video file containing the demonstration of the realized experiment.

The video file is generated using the Open Broadcast Studio. The program hardcodes information required for identification of the student and the identification number of the laboratory stand into the video. The video must show the source codes used for demonstration, the process of uploading a binary file to the FPGA board, as well as demonstration of the simulation results and functionality of the program on the actual hardware. During the presentation, the student can switch the view between the code and the behavior of the FPGA hardware by turning the Webcam ON or OFF.

The role of the teacher in the assessment process involves cross-analysis of the HDL codes submitted by the students, their comparison against the source files from the video, as well as evaluation of the simulation results and demonstrated functionality of the circuit. In the case of any doubts or deficiencies in the submitted report, the teacher contacts the student in order to discuss the results and/or suggest the improvements required to obtain a positive grade. The approach is simple, transparent, and limits the time required for assessment of the experiments to the necessary minimum.

C. Remote Realization of Laboratory Exercises: Summary

From student's perspective, realization of the FPGA exercise in the remote environment is as follows.

1) Book the available time slot on the selected remote laboratory stand (appropriate with respect to experiment type).
2) Login to the machine within the reserved time slot and realize the task.
3) If the task is finished within the selected timeframe, go to 4; otherwise, save the developed codes on a local computer, logout, and go to 1.
4) Record the demonstration of the experiment, send all the required files for assessment and logout.
5) If the grade is received END; otherwise, discuss the required changes with the instructor and go to 1.

The reservation of time slots is realized using a free of charge version of a TeamUp calendar [41]. TeamUp provides an application programming interface (API), which—among others—can be used to download the information on the booked time slots for further processing.

It should be emphasized that the development and implementation of the remote laboratory based on the architecture and task realization scheme discussed above has been accomplished in a timeframe of roughly two weeks. Without a doubt, the transition of the teaching paradigm from the stationary to the remote model in such amount of time can be considered rapid.

IV. Results and Discussion

The architecture of the remote laboratory presented in this work was implemented on a total of ten stands, all of which were equipped with the virtual I/O interface of Section II-C. The remote labs have been used by nearly 210 students for over two months of the summer semester of 2020.

A. Evaluation of the Laboratory—Survey

Evaluation of the laboratory was oriented toward providing the qualitative and quantitative information about the effect of working in the remote environment on the learning experience.

TABLE I
SURVEY—STATISTICAL RESULTS

Question	μ	σ^2
Remote laboratory assessment		
usefulness as a replacement of stationary labs	4.40	0.61
usefulness as a supplement to stationary labs	4.54	0.58
setup of the remote laboratory stands	4.67	0.38
reliability of the virtual I/O interface	4.33	0.54
user interface of the virtual I/O interface	4.19	0.67
assessment mechanism of the exercises	4.37	0.65
Technical documentation		
usefulness of the lectures for realization of exercises	3.87	0.73
clarity of user manual (connection setup)	4.38	0.62
clarity of user manual (CAD software)	4.53	0.45
clarity of instructions (experiments)	4.02	0.75
clarity of assessment rules	4.49	0.50
Availability/reliability of the remote laboratory		
usefulness of 24/7 access to the lab	4.79	0.41
availability of time slots when required	4.27	0.97
reliability/stability of RDP connection	4.42	0.60
readiness of the reserved machine	4.47	0.71
User experience		
connection setup	4.48	0.81
reservation of the time slots	4.58	0.63
availability of the necessary data on the webpage	3.93	0.97
Communication		
questions – teacher's response time*	4.04	1.14
exercise assessment time	3.86	1.65
Learning experience		
difficulty of exercises	3.01	0.91
usefulness of teacher's hints*	3.98	1.08
remote lab as a tool for understanding FPGA	4.39	0.54
availability of the remote labs after pandemic	4.36	0.89
availability of all the required software on remote PCs	3.31	0.42

* 46 of 184 students that filled the survey (25 percent) did not have any requests to teachers other than assessment of the exercises

The first source of data was an anonymous 25-question survey based on a Likert-type scale [26], [42]. The students were asked to answer the questions by selecting one of the available answers enumerated from 1 to 5, where 1 and 5 meant the worst and the best, respectively. The questionnaire was appropriately filled by 184 of 201 undergraduate students, which gives the participation rate of 91.5%. The questions, along with their assessment in the form of means and variances, are gathered in Table I.

The obtained results indicate not only a high satisfaction of the students with the technical aspects of working on the remote stands but also a decent quality of the documentation. The setup of the remote laboratory stands and 24/7 availability of the hardware received especially high grades. Relatively low variances obtained for most of the questions confirm consistency of the assessment regardless of the differences between individual courses (and exercises) realized in the remote environment.

Another important observation concerns a relatively high variance for questions related to communication with the teachers. It should be noted that a total of five teachers were involved in realization of the courses discussed in this work, hence, the increased variance reflects the differences between their attitude to contact with the students.

Finally, the results concerning setup of the laboratory stands and opinions on the local installation of CAD packages seem to be counterintuitive. On the other hand, they indicate that majority of the students is not afraid of "experimenting" with the software that can be used for realization of the tasks on their private computers. It is also worth mentioning that 15 of questioned students (8.2% of the pool) used operating systems other than Windows to connect with the remote hardware. Having in mind this group, availability of CAD software on the remote machines is still of high importance.

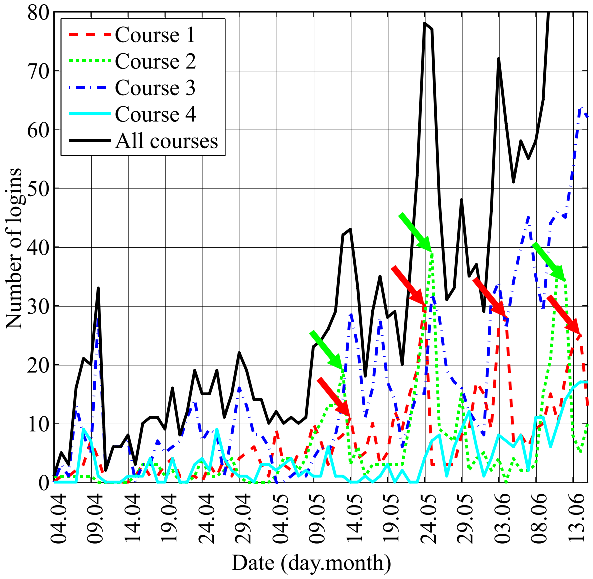


Fig. 5. Daily number of logins: green and red arrows represent the deadlines for realization of specific laboratory exercises.

B. Evaluation of the Laboratory—System Load

Fig. 5 shows the load of the remote laboratory (all stands). The data were obtained from a total of 201 undergraduate students. The number of participants assigned to particular courses was as follows: 1) Course 1 – 43; 2) Course 2 – 36; 3) Course 3 – 109; and 4) Course 4 – 13. Overall, the remote lab registered 2398 logins and a total workload of 4180 h (almost 21 h per student).

Theoretical capacity of the system (assuming the maximum possible length of the time slot that amounts to 2 h) corresponds to 120 students/day. Organization of the laboratories was divided into two cycles. In the first one (between April 4th and May 6th), the deadlines on realization of the exercises were not imposed. Then, between May 7th and June 15th, due dates on submission of the task reports were set for the Courses 1 and 2 to balance the system load.

In the first cycle, the average load of the system was only 12.3%. A significant increase, to average of 50.4%, was observed in the second time span. It should be noted that the system load increased significantly at the end of the semester and reached its peak of over 100 individual logins per day between June 12th and June 15th. Notwithstanding, for the Courses 1 and 2, the increase of the system load was observed only around the deadlines. Based on the results, one can infer that the introduction of nonoverlapping deadlines for realization of tasks on all courses would result in a more balanced use of the resources. A good indication of this observation is the (unbalanced) Course 3, where substantial increase of login counts toward the end of the semester can be observed.

TABLE II
STATISTICAL SIGNIFICANCE OF THE RESULTS

Course	Years	$\lvert x_1 \rvert$	μ_1	σ_1^2	$\lvert x_2 \rvert$	μ_2	σ_2^2	p-val*	p-val#
1	2018/2020	77	0.71	0.060	43	0.58	0.037	0.00	0.00
	2019/2020	95	0.77	0.071				0.00	0.00
2	2018/2020	56	0.53	0.026	36	0.56	0.025	0.53	0.56
	2019/2020	61	0.58	0.031				0.72	0.72
3	2018/2020	137	0.63	0.034	109	0.78	0.005	0.00	0.00
	2019/2020	111	0.68	0.022				0.00	0.00
4	2018/2020	14	0.69	0.047	13	0.65	0.158	0.02	0.01
	2019/2020	16	0.62	0.003				0.02	0.01

* Calculated using unequal variances t-test
Calculated using Wilcoxon rank-sum test

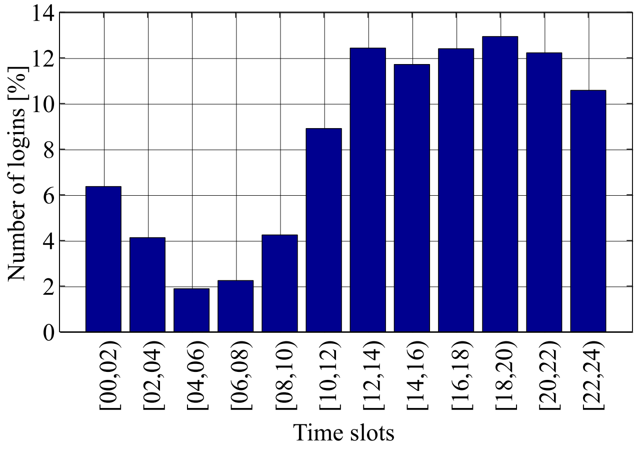


Fig. 6. Daily load of the system (all laboratory stands) over the entire period of the remote laboratory availability.

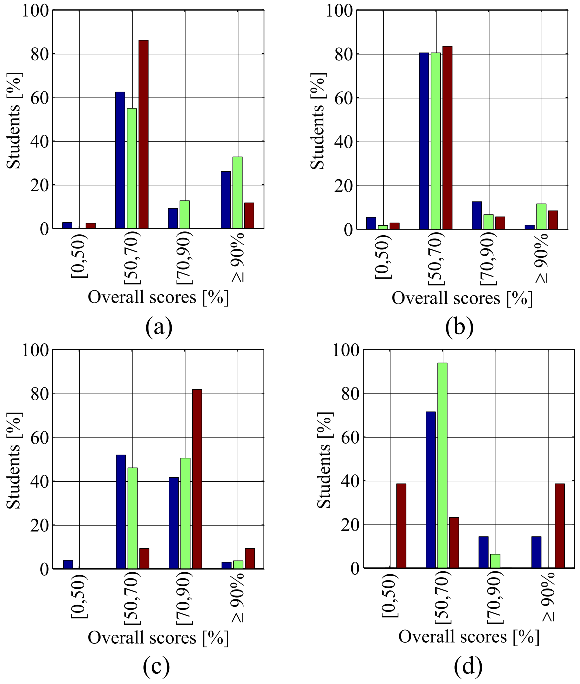


Fig. 7. Scores obtained during the stationary labs in 2018 (blue) and 2019 (green), as well as during the remote laboratories introduced in 2020 (brown). Courses: (a) 1, (b) 2, (c) 3, and (d) 4.

Daily load of the laboratory stands shown in Fig. 6 indicates that majority of the students realized their tasks between 10:00 and 02:00 (note that a 24-h clock is used here). Only 12.5% of the overall load falls into the timeframe between 02:00 and 10:00, which suggests that in this time the laboratory could be turned off (for most of the semester except the last week before its end; see Fig. 5).

The analysis of the system load indicates that—assuming balanced use of the resources (enforced by the deadlines)—the daily capacity of the laboratory, sufficient for the total number of around 200 students, should amount to about 50 unique logins (25% of the number of students). Assuming the connection time of up to 2 h per student, such capacity can be ensured using only six laboratory stands working 7 days a week between 10:00 and 02:00. One redundant stand could also be prepared to ensure uninterrupted realization of exercises in the unlike event of hardware failure.

C. Evaluation of the Laboratory—Scores

As already mentioned in Section III-A, the undergraduate FPGA laboratory introduces almost 30 practical exercises. The tasks can be roughly divided into four categories that reflect the level of difficulty and time expenditure required for their completion. Example tasks in each group involve realization of the experiments concerning:

1. [0, 50)%—implementation of a simple combinatorial and sequential logic, basic interactions with peripherals such as diodes, or 7-segment displays, HDL-based benchmarks;
2. [50, 70)%—multimodule systems that implement control logic for peripheral components, such as displays, PS/2 keyboard, or UART receivers;
3. [70, 90)%—systems comprising simple IP cores, such as buffers and memory modules and UART transceivers;
4. $\geq$ 90%—generation of curves on the VGA monitor, implementation and programming of FPGA-based microprocessors, or validation of AES ciphering cores.

Fig. 7 shows the grades obtained for realization of exercises in the stationary (years 2018 and 2019) and the remote (year 2020) environments. Statistical significance of the results has been verified using a two-sample unequal variances t-test and a Wilcoxon rank-sum test [43], [44]. The null hypothesis H_0 was that there is no difference between the scores obtained for realization of exercises in the stationary and remote labs. The analyses have been performed separately for each course and their results are gathered in Table II. The parameters $\lvert x_k \rvert$, μ_k, and $\sigma_k{}^2$ (k = 1, 2) represent the size of the data sets, their mean, and variance, respectively. The p-values obtained for the Courses 1, 3, and 4 are lower than the assumed significance rate of 5% which allows to reject the null hypothesis. Hence, the results indicate that remote realization of tasks affected the scores obtained by the students assigned to the mentioned courses. However, for the Course 2, the null hypothesis H_0 cannot be rejected.

As shown in Fig. 7, for certain courses, the overall scores obtained by the students who worked remotely are worse

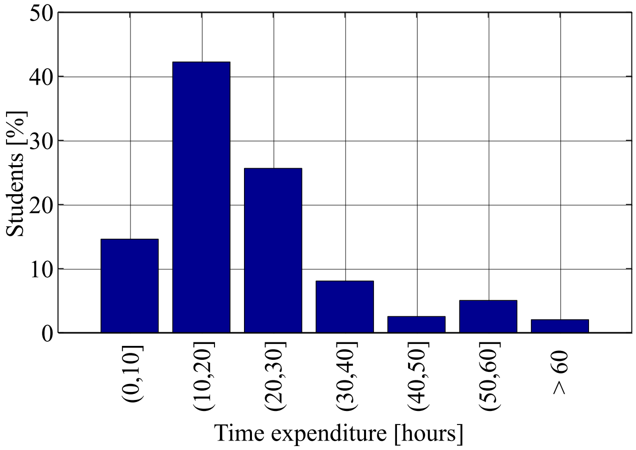


Fig. 8. Time expenditure for realization of laboratory exercises—all undergraduate courses.

compared to stationary-based results. Although the obtained outcomes are inconsistent with the conclusions drawn in the available literature [19], [26], they can be explained based on the specific features of the presented laboratories, as well as the results of the survey from Table I. First, the discussed labs are organized in such a way that the exercises that have to be realized to obtain 50% score are relatively easy, whereas complexity of tasks that have to be completed to obtain higher grades increases exponentially. Furthermore, the exercises for the entry-level course (Course 3) are significantly simpler and less time-consuming than the remaining ones. Based on that, one can infer that the students seek for a tradeoff between the final grade and time expenditure required to complete the tasks. Similar conclusion can be drawn based on the analysis of the results shown in Fig. 8. As can be implied from Table I, the worsened scores obtained for more advanced courses could also result from hindered communication with instructors, as compared to stationary laboratories.

D. Comparison Against Other Remote FPGA Labs

The presented remote laboratory has been compared against other architectures from the literature [12]–[15], [18], [19]. The results gathered in Table III indicate that the proposed solution outperforms other systems not only in terms of a short implementation time but also support for the heterogenic FPGA hardware, or flexibility of the virtual I/O interface. Other advantages include high scalability, reuse of the hardware used for stationary teaching, negligible requirements for the client hardware/software, long access time (up to 2 h per session), and high similarity between the remotely developed HDL codes and the on-site written ones. A more detailed discussion of the mentioned aspects is provided below.

TABLE III
COMPARISON OF REMOTE LABORATORY ARCHITECTURES

.

	[12]	[13]	[14]	[15]	[18]	[19]	This work
Time to implement	– –	– –	– –	+	–	–	+ +
Reuse of on-site hardware	+	+ +	– –	–	–	–	+ +
Diversity of FPGA hardware	– –	+	– –	– –	– –	– –	+ +
Flexibility of virtual I/O	– –	–	+	– –	+ +	+ +	+ +
Scalability	– –	+	+ +	+ +	+	+	+ +
Local software requirements	–	–	–	–	–	+	+
Flexibility w.r.t. experiments	+	+	+	–	+	+	+ +
HDL similarity – remote vs. local	–	–	+	+ +	+ +	+	+ +
Ease of authentication	–	+	–	+	+	+	+
Session length	– –	– –	–	N/A	+	+	+ +
User interface technology*	WI	WI	WI	RDP	WI	WI	RDP
Interactive user interface	–	+	– –	+	+ +	+	+ +
Development cost	–	+	– –	– –	–	–	+ +
Remote grading	+	N/A	N/A	N/A	+	+	+ +
Survey-based assessment	N/A	–	N/A	N/A	–	+ +	+

* WI – Web-based interface

The first important observation is that almost all of the considered architectures implement the working scheme that is based on availability of the necessary CAD on the client machine [12]–[15], [18]. Such a solution is even promoted as advantageous for reduction of the server load [14]. On the other hand, the CAD that can be installed for free on student's machine is often limited in functionality. Furthermore, local installation of the software enforces the use of PC-compatible clients, as tablets and smartphones are unsuitable for modern CAD packages (both architecture- and powerwise). From this perspective, the availability of the software on the server side not only addresses the potential license issues for the use of commercial CAD but is also important from the standpoint of providing access to all the required tools regardless of the type of client machine used for connection.

Majority of the considered laboratories enable interaction with the hardware through custom Web-based interfaces [12]–[14], [18], [19]. The reasoning behind high popularity of Web-based solutions remains unclear, because the necessity of developing custom applications that reside on the dedicated servers negatively affects the cost and the time required to start the laboratory. Furthermore, Web interfaces may hinder working with the hardware. A good example is the solution from [14], where the behavior of the FPGA system under test is not provided in the form of a live-feed. Instead, it is first registered on the server as a video file and then sent to the user. Only one of the considered laboratory architectures (except the proposed one) uses RDP for establishing a client–server connection [15]. However, lack of Webcam interfaces in [15] decreases the potential benefits resulting from the versatility of RDP.

Most of the considered architectures is characterized by relatively high scalability along with the acceptable similarity of the HDL codes validated using the local and remote environments. On the other hand, the discussed solutions are often dedicated to work with homogenic FPGA boards [12], [14], [15], [18], [19]. Moreover, virtual

TABLE IV
REMOTE FPGA LABORATORY: COST BREAKDOWN (10 STANDS)

Item	Pcs	Cost per item [USD]	Cost [USD]
Arduino PRO mini	10	3.05	30.50
USB-UART interface	10	1.91	19.10
Webcam	10	18.47	184.70
USB cables	10	0.74	7.40
PCB fabrication	10	13.99	139.90
Miscellaneous	N/A	N/A	16.00
Total cost			397.60
Cost per stand			39.76

I/O used in [12] and [15], or the entire concept of the laboratory discussed in [14], hinder their adaptation to work with heterogenic hardware.

A common feature of the discussed laboratories—which solidifies their main role as an enhancement of the stationary teaching model rather than its replacement—is the timeframe for which the access to the remote FPGA hardware is granted. In [12] and [13], the connection time is either limited to only a few minutes, or the user is disconnected after a short idle. In [14], the user is banned after each connection for an unspecified period of time, which makes a quick correction of errors identified in the validated HDL codes virtually impossible.

Another observation is that the problem of evaluating the exercises realized in the remote environment is either neglected or treated superficially. In [12] and [19], students' knowledge is assessed based on online tests and quizzes. In [18], a short information is provided that the grading process involves analysis of source codes, comments, and screenshots from the Web interface.

Analysis of the remote laboratories usefulness/efficiency is also an important aspect that is often neglected in the literature. In [13] and [18], the evaluation is performed using a small pool of students which is unsuitable for drawing unambiguous conclusions on the experience of working in the remote environment. In this regard, the work [19] deserves attention as it examines a large group of students and provides a detailed analysis of the obtained data. It should be stressed out that the assessment results presented here are competitive as they are not only drawn based on a fairly large number of questionnaires but also combine survey with analyses concerning students' involvement in the realization of exercises and utilization of system's capacity.

The development cost, although difficult to estimate [19], is also an important factor. Here, the main (and true) assumption is that conversion of the FPGA laboratory to the remote one was conducted using the already available software/hardware. The only real cost was related to the development of the virtual I/O interface. The time-expenditure of the staff for preparation of the remote environment was not accounted for to the overall cost. The reason is that mentioned tasks were performed as a part of the staff members duties. The overall cost of developing ten laboratory stands—given in Table IV—is less than $400, which is order of magnitude lower compared to, e.g., [19].

One should also consider whether a high reliance on an open-source software is necessary for rapid transition to the remote teaching model. Although open-source packages benefit from transparency, their setup is often more time-consuming as compared to, e.g., RDP. A good example is the architecture of [14], which is entirely based on the open-source solutions. It was not only expensive but also offered a limited user interface. In this context, the system based on appropriate combination of open and proprietary technologies seem to provide not only cheap but also efficient solution to the remote laboratory.

V. CONCLUSION

In this work, a cheap architecture of the remote FPGA laboratory that enables a rapid transition from the stationary to the remote teaching model has been presented. The proposed solution supports heterogenic FPGA hardware and a variety of HDL experiments oriented toward interfacing with the diverse peripheral components. Furthermore, it provides a remote access to all required software packages, which is useful for realization of the exercises not only using the PC but also tablets or smartphones. The presented architecture has been evaluated based on a survey data obtained from 184 students. The quantitative results, backed up by a positive feedback from the students, indicate usefulness of the proposed remote laboratory as a tool that provides hands-on-experience with actual hardware during pandemic.

REFERENCES

[1] X. Han *et al.*, "Novel coronavirus disease 2019 (COVID-19) pneumonia progression course in 17 discharged patients: Comparison of clinical and thin-section computed tomography features during recovery," *Clin. Infectious Diseases*, vol. 71, no. 15, pp. 723–731, 2020.

[2] *The Statistical Office of the European Union, Unit G4: Innovation and Digitalization*, Eurostat, Luxembourg City, Luxembourg, 2020.

[3] *WebEx*, Cisco Syst., Inc., San Jose, CA, USA, 2020.

[4] *Zoom Video Communications*, Almaden Boulevard, San Jose, CA, USA, 2020.

[5] *Teams*, Microsoft Corp., Redmond, WA, USA, 2020.

[6] *Skype*, Microsoft Corp., Redmond, WA, USA, 2020.

[7] E. Ivov, "Jitsi," in *The Architecture of Open Source Applications*, A. Brown and G. Wilson, Eds. Railegh, NC, USA: Lulu, 2011, pp. 139–159.

[8] *Moodle*, Moodle Pty Ltd, West Perth, WA, Australia, 2020.

[9] *Schology*, Lexington, New York, NY, USA, 2020.

[10] *Canvas*, Instructure, Salt Lake City, UT, USA, 2020.

[11] *edmodo*, Mariners Island Blvd, San Mateo, CA, USA, 2020.

[12] J. Soares and J. Lobo, "A remote FPGA laboratory for digital design students," in *Proc. Portuguese Meeting Reconf. Syst.*, 2011, pp. 95–98.

[13] F. Morgan, S. Cawley, and D. Newell, "Remote FPGA lab for enhancing learning of digital systems," *ACM Trans. Reconfig. Techn. Syst.*, vol. 5, no. 3, pp. 1–13, 2012.

[14] M. Reichenbach, M. Schmidt, B. Pfundt, and D. Fey, "A new virtual hardware laboratory for remote FPGA experiments on real hardware," in *Proc. Int. Conf. e-Learn. e-Bus. Enterprise Inf. Syst. (e-Gov.)*, 2011, pp. 1092–1097.

[15] W. M. El-Medany, "FPGA remote laboratory for hardware e-learning courses," in *Proc. IEEE Int. Conf. Comput. Techn. Elect. Electron. Eng.*, 2008, pp. 106–109.

[16] R. Hashemian and T. R. Pearson, "A low-cost server-client methodology for remote laboratory access for hardware design," in *Proc. ASEE/IEEE Front. Educ. Conf.*, 2009, pp. 240–244.

[17] F. Morgan *et al.*, "Remote FPGA lab with interactive control and visualisation interface," in *Proc. Int. Conf. Field Program. Logic Appl.*, 2011, pp. 496–499.

[18] V. Fotopoulos, A. Fanariotis, T. Orphanoudakis, and A. Skodras, "Remote FPGA laboratory course development based on an open multimodal laboratory facility," in *Proc. Panhellenic Conf. Inf.*, 2015, pp. 447–452.

[19] K. P. Ayodele, I. A. Inyang, and L. O. Kehinde, "An iLab for teaching advanced logic concepts with hardware descriptive languages," *IEEE Trans. Educ.*, vol. 58, no. 4, pp. 262–268, Nov. 2015.

[20] V. Fotopoulos, A. Spiliopoulos, and A. Fanariotis, "Preparing a remote conducted course for microcontrollers based on Arduino," in *Proc. Int. Conf. Open Dist. Learn.*, 2013, pp. 1–7.

[21] B. Barros, T. Read, and M. F. Verdejo, "Virtual collaborative experimentation: An approach combining remote and local labs," *IEEE Trans. Educ.*, vol. 51, no. 2, pp. 242–250, May 2008.

[22] Z. Aydogmus and O. Aydogmus, "A Web-based remote access laboratory using SCADA," *IEEE Trans. Educ.*, vol. 52, no. 1, pp. 126–132, Feb. 2009.

[23] N. E. Cagiltay, E. Aydin, R. Oktem, A. Kara, M. Alexandru, and B. Reiner, "Requirements for remote RF laboratory applications: An educators' perspective," *IEEE Trans. Educ.*, vol. 52, no. 1, pp. 75–81, Feb. 2009.

[24] E. Besada-Portas, J. A. Lopez-Orozco, L. de la Torre, and J. M. de la Cruz, "Remote control laboratory using EJS applets and TwinCAT programmable logic controllers," *IEEE Trans. Educ.*, vol. 56, no. 2, pp. 156–164, May 2013.

[25] C. M. Ionescu, E. Fabregas, S. M. Cristescu, S. Dormido, and R. De Keyser, "A remote laboratory as an innovative educational tool for practicing control engineering concepts," *IEEE Trans. Educ.*, vol. 56, no. 4, pp. 436–442, Nov. 2013.

[26] A. Chevalier, C. Copot, C. Ionescu, and R. De Keyser, "A three-year feedback study of a remote laboratory used in control engineering studies," *IEEE Trans. Educ.*, vol. 60, no. 2, pp. 127–133, May 2017.

[27] F. Steger, A. Nitsche, A. Arbesmeier, K. D. Brade, H. Schweiger, and I. Belski, "Teaching battery basics in laboratories: Hands-on versus simulated experiments," *IEEE Trans. Educ.*, vol. 63, no. 3, pp. 198–208, Aug. 2020.

[28] *Atmega 328P*, Microchip Technol., Chandler, AZ, USA, 2020.

[29] *CP2120*, Silicon Labs, Austin, TX, USA, 2015.

[30] *FPGA Evaluation Boards*, Digilent Inc., Pullman, WA, USA, 2019.

[31] B. C. Readler, *VHDL by Example: A Concise Introduction to FPGA Design*, San Bernardino, CA, USA: Full Arc Press, 2014.

[32] *PS/2 Device Datasheet*, Cypress Semicond., San Jose, CA, USA, 2014.

[33] K. Jamsa, *Introduction to Web Development Using HTML-5*. Burlington, MA, USA: Jones & Bartlett Learn., 2014.

[34] Q. A. Acton, Ed., *Internet Protocols—Advances in Research and Application*, Scholarly Editions, Atlanta, CA, USA, 2013.

[35] *Rdesktop—Remote Desktop Protocol*, Cendio AB, Linköping, Sweden, 2019.

[36] *Chrome Remote Desktop*, Alphabet Inc., Mountain View, CA, USA, 2020.

[37] *SPICE Remote Computing Protocol Definition V1.0, SPICE Project*, Red Hat, Inc., Raleigh, NC, USA, 2009.

[38] T. Richardson, Q. Stafford-Fraser, K.R. Wood, and A. Hopper, "Virtual network computing," *IEEE Internet Comput.*, vol. 2, no. 1, pp. 33–38, Jan./Feb. 1998.

[39] *TeamViewer*, TeamViewer Germany GmbH, Goppingen, Germany.

[40] *SystemC*, Accellera Syst. Initiat., Elk Grove, CA, USA, 2007.

[41] *TeamUP*, Teamup Solutions AG, Zurich, Switzerland, 2020.

[42] A. van Alphen, R. Halfens, A. Hasman, and T. Imbos, "Likert or rasch? Nothing is more applicable than good theory," *J. Adv. Nursing*, vol. 20, no. 1, pp. 196–201, 1994.

[43] B. L. Welch, "The generalization of 'Student's' problem when several different population variances are involved," *Biometrika*, vol. 34, nos. 1–2, pp. 28–35, 1947.

[44] J. Pearce and B. Derrick, "Preliminary testing: The devil of statistics?" *Reinvent. Int. J. Undergrad. Res.*, vol. 12, no. 2, pp. 1–18, 2019.