International Journal of Engineering Education Vol. 33, No. 2(B), pp. 865–876, 2017
Printed in Great Britain

0949-149X/91 $3.00+0.00

Leveraging Web Services and FPGA Dynamic Partial Reconfiguration in a Virtual Hardware Design Lab*

OCTAVIAN M. MACHIDON[1], ALINA L. MACHIDON[2], PETRU A. COTFAS[1] and DANIEL T. COTFAS[1]
[1] "Transilvania" University of Brasov, Eroilor Blvd 29, 500036, Brasov, Romania.
[2] SC Vision Systems SRL, Aurel Vlaicu 61, 500188, Brasov, Romania. E-mail: octavian.machidon@unitbv.ro, fechete.alina@gmail.com, pcotfas@unitbv.ro, dtcotfas@unitbv.ro

The recent rise and development of a variety of virtual and distance learning platforms in the field of engineering has brought on common downsides for most of these solutions. Among them, the difficulty in offering access to physical equipment in fields where a practical, hands-on approach is mandatory and also a low effectiveness in resource sharing, due to the fact that traditionally, one hardware equipment can be used by only one student at a given moment. This paper describes the design and implementation of a digital hardware design remote virtual laboratory that provides solutions to these issues. Reconfigurable hardware development platforms are made available online using Web services, thus providing easy access for student practice by the means of an intuitive Web interface that offers the user the possibility to remotely configure and communicate with its designated hardware resource. Another particular feature of this implementation is that the same development board can be shared simultaneously by several users by using dynamic partial reconfiguration, thus achieving an improved resource sharing effectiveness. The developed laboratory works are oriented on digital hardware design but also on using the digital hardware for running the mathematical models of renewable energy sources. In order to evaluate the system, an analysis is being presented showing the technical and learning benefits brought by using this virtual platform.

Keywords: remote lab; distance learning; hardware design; System-on-Chip

1. Introduction

The latest technological advances of the recent years have led to the rise and development of a variety of virtual and distance learning solutions. Such online platforms cover nearly all field of education, courses and other materials being published online together with evaluation techniques that make possible granting online graduation certificates. These online learning platforms, while bringing on a series of advantages like multi-user handling, resource sharing, up-to-date content and quality learning materials, also usually have a common downside: the difficulty in offering access to physical equipment in the fields where a practical, hands-on approach is mandatory for a full learning experience.

This is the case for the field of electronic engineering, where the role of laboratory work is very important. Many challenges need to be overcome when developing a real-time electronic engineering laboratory training platform, especially regarding the management and sharing of the hardware resources, the user interface, I/O data communication and activity evaluation.

Digital hardware design and embedded system programming are present in our Faculty's study programs at both undergraduate and master levels. These courses have a substantial focus on laboratory work and hands on activities, providing students with access to hardware development platforms. Offering all students direct access to such resources can be done however only during the specific course hours and at the laboratory location, which limits the invaluable experience and knowledge they can gain from a hands-on approach. Since we consider this as being a key factor, we have made efforts in the design and development of a remote virtual laboratory platform that could overcome the downsides mentioned above.

In this paper, we present the development of a remote System-on-Chip (SoC) virtual laboratory supporting training and exercises in the fields of digital and embedded system design, which is made available online to students and faculty. Our implementation proposes a novel approach that takes advantage of Web technologies and the latest SoC technological features, especially dynamic partial reconfiguration (PR).

The main contributions of this paper are:

- A Web service-based organization of the virtual learning platform that enables an easy scaling of the number of SoC platforms available and facilitates inter-institutional sharing
- By leveraging the partial reconfiguration of the SoC devices, a dynamic and efficient access management of the hardware resources has

* Accepted 13 December 2016.

been accomplished, where the same device can be simultaneously accessed by several users, achieving a form of "hardware multitasking".
- The proposed virtual lab architecture allows for a large number of users to be offered practical experience using minimal hardware resources

Besides the educational perspective of the service-based integration of the reconfigurable hardware devices, we based our efforts on other important benefits highlighted by current research in this direction [1], like reducing the redesign complexity and improving programmability of such devices. The synergy between the two technologies, reconfigurable and distributed computing, leads to an important feature with applications in both educational and industrial field: filed upgradeability of the reconfigurable hardware resources (they can be (re)configured and/or upgraded from a remote location, without physical access to the specific hardware resource).

This approach enables high-end reconfigurable SoC platforms to become available to every student via the Internet, a very important asset given the inherent difficulties (financial, logistical) encountered while attempting to grant each student direct, physical access to such devices. The student can be thus granted access to a variety of SoC and FPGA (Field Programmable Gate Array) target boards on the cloud computing model "Hardware-as-a-Service", having the possibility to deploy their designs online.

The rest of this paper is organized as follows: section 2 reviews previous work in the design and development of such remote hardware design platforms. Section 3 describes our platform's architecture, implementation and usage. Section 4 presents the technical validation of the system and the evaluation of the learning impact, and finally section 5 concludes the paper and proposes future extensions on this subject.

2. Background and related work

With regard to remote FPGA laboratories, the subject of this paper, existing research [2, 3] has underlined the key assets brought on by such remote platforms including high scalability and flexibility, reduced implementation and maintenance costs, enhanced student experience with improved learning results and an easy inter-institutional sharing of such facilities.

The most basic implementations of such remote labs [4–6] are based on a Windows Remote Access to a local PC that provides access to local laboratory equipment and also runs dedicated development and deployment software. Nevertheless, these implementations come with considerable limitations regarding software/hardware resource sharing, while only partially reducing costs and being difficult to use and manage due to the fact that remote workstations are needed for the users to access and run the design tools.

The main technical challenge behind the development of a remote learning laboratory is the difficulty in integrating various software tools and hardware devices, and in providing remote access to these resources. Such an integration effort basically deals with developing a flexible middleware on top of these resources, able to expose their functionality and abstract away part of their implementation and low-level characteristics. Consequently, the ideal building blocks of such a middleware are Service-Oriented Architectures (SOA), which offers important assets like high flexibility, extensibility and reduced complexity of the design in operating with a variety of heterogeneous technologies. SOA accomplishes a high level abstraction of the resources due to its unified approach—based on standard protocols and technologies, like SOAP (Simple Object Access Protocol), XML (Extensible Markup Language), and so on—that makes these resources available to users as services with a standardized interface.

Given the above, we have chosen a service-oriented design and implementation approach in order to be able to make the learning resources more easily available to students by the means of online Web services. Furthermore, we have capitalized on a synergy between this SOA-based approach and an emerging and innovative technology—the latest System-on-Chip (SOA) architectures, integrating both FPGA programmable logic (PL) and single or multi-core processing systems (PS).

Over the traditional hardware design laboratories—that use the classic FPGA development boards which allow embedded software/hardware microprocessors to be instantiated in the design—this approach takes advantage of two key features: dynamic full and partial reconfiguration of the PL by the PS (allowing total or partial modifications of the circuit's functionality during its runtime), and enhanced IP (Intellectual Property) Core integration mechanisms—based on Network-on-Chip (NoC) or advanced bus interconnect structures.

Another reason for our choice of an SOA-based approach because it opens the doors for future extensions of our system, like a possible integration with an existing LMS. This is possible by using interoperability specifications based on SOA, such as IMS LTI (Learning Tools Interoperability) [7].

The service-oriented perspective applied in our setup is also part of an ongoing trend: another recent approach that fosters service-oriented archi-

tectures and Web services is described in [8]. It has the software tasks (EDA design flow: design, simulation and implementation) being executed on both the local user's PC and the online Web service, by having a hybrid architecture. Despite the fact that it improves significantly the hardware resource usage and effectiveness, its performance and reliability still require the users having high performance computers.

There have been several other attempts in developing virtual and remote laboratory access in this domain: the MIT iLab Remote FPGA & CPLD Laboratory [9], the Ilmenau University of Technology Online Lab [10], or the Hellenic Open University (HOU) implementations of an Arduino [11] and FPGA remote lab [12]. These high-end implementations of hardware design virtual laboratories, most of which Web service-based, offer an easy sharing of the resources amongst users and also institutions, favoring the creation of a network of shared online laboratories [13].

However, all of these implementations are based on time-multiplexing the user's access to the development devices, with one device being entirely reserved and accessed by a single user for a given time interval. The remote laboratory architecture described in this paper proposes a significant improvement to this approach. The key feature that brings novelty to our implementation is the "hardware multitasking" capability of each development board. This means that the same board can accommodate up to 4 user designs operating in parallel, without interfering with one another, leading to a much more efficient use of the hardware resources.

3. System description

In our implementation (depicted in Fig. 1) we have used a dedicated quad-core Intel i7 server for deploying the Web application and Web service, and a network of Avnet ZedBoard 7020 (based on the Xilinx XC7Z020 AP SoC) as reconfigurable hardware platforms to be offered for student practice.

3.1 Web service and application

Web services and service-oriented architectures, the building blocks of the emerging "Everything-as-a-Service" trend, are being credited as viable solutions for ensuring a high degree of inter-operability and easy integration of various heterogeneous devices and technologies. Consequently, we aimed at developing a service-based middleware capable of making the configuration and communication interface with reconfigurable hardware design board available to the students as online Web services. This approach has the potential of increasing the scalability and adaptability of the entire remote learning platform and also ease access and use regarding the target boards since the potential users don't need to worry about installing and managing the specific software tools needed for such operations (either in a classic laboratory setup or on personal computers).

We have implemented two Web-based components (as shown in Fig. 2): a Web Service and Web application implementing the user interface, both being developed using Java and NetBeans IDE (Integrated Development Environment) and running on a Glassfish 4.0 Server instance.

Being targeted especially for student-use, a key component of the platform is the Web-based user interface. Since this interface plays also the role of interacting with the Web service, exposing its functionality, we have implemented it as a Web application integrating JSP (Java Server Pages) web pages (the actual user interface) and a Java servlet, an intermediary agent ensuring the SOAP communication with the Web service and processing of the

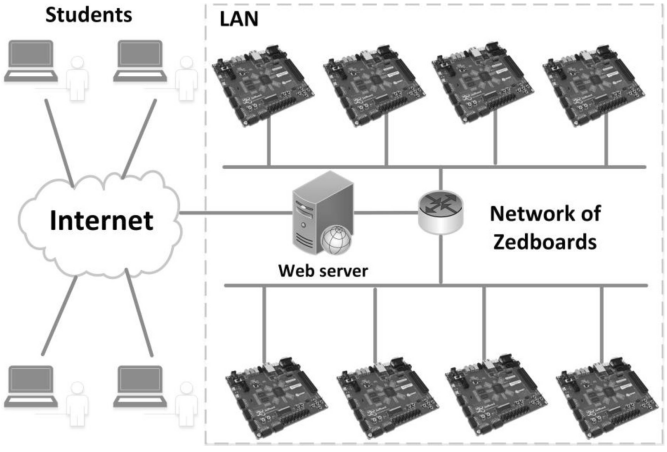

Fig. 1. Overview of the remote hardware design laboratory.

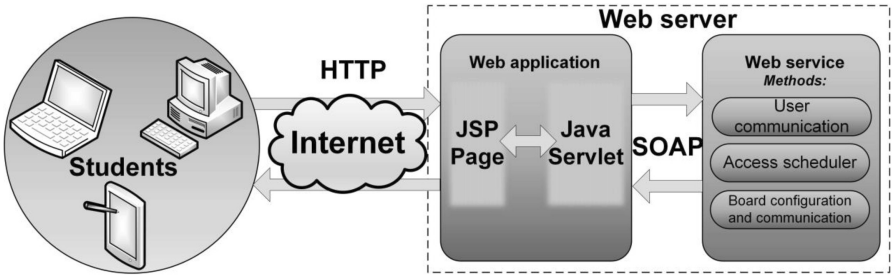

Fig. 2. Web service and application architecture.

information from/to the JSP page. We have chosen JSP for implementing the user interface since it combines HTML, XML and embedded Java code [14], being thus able to generate dynamic web content.

The Web service is the core of our platform, being responsible with communicating via the local network with the development boards and via the Web application interface with the students/users. We have implemented three methods in our Web service: one for communicating and data transfer with the user (JSP page interface), one for remote configuring a development board with a given bitstream configuration file, and the last for scheduling user access to the boards. A MySQL database is dedicated to handling the user account data, scheduling system information and other usage statistics.

Of critical importance to the entire remote learning platform is the user access management to the reconfigurable resources. This is handled, as mentioned above, by a dedicated method inside the Web service which manages a user reservation schedule for accessing the processing cores on the development boards.

3.2 The network of reconfigurable hardware platforms

Having high performance and versatility requirements, we have chosen the Avnet ZedBoard 7020 development board [15], with a Xilinx Zynq™-7000 AP (All Programmable) SoC XC7Z020 [16]. The XC7Z020 AP SoC integrates in the same device a dual-core ARM® Cortex™-A9 MPCore™ processing system (PS) and Xilinx programmable (reconfigurable) logic (PL). This unique architecture also enables the possibility of complete and partial reconfiguration of the PL by the PS, which represents an important feature for this implementation.

A network of eight such Zedboards has been set up, with each board's communication capability being based on the on-board Gigabit Ethernet controller and the TCP/IP LwIP (Lightweight IP) Stack 1.4.0 [17].

Each board's ARM-based processing system runs an embedded C application integrating the following components: network communication interface, managing a memory-based file system, and managing the complete/partial reconfiguration of the programmable logic. Two network applications are running on each SoC's processing system: a TFTP (Trivial File Transfer Protocol) server—for transferring and storing full and partial bitstreams—and a TCP server responsible with data transfer and exchanging control and status commands with the Web service. Each Zynq SoC has been set up with a 128MB file system (FS) residing in the on-board DDR3 memory which is used for storing and handling the configuration files (bitstreams). This FS has been implemented using the LibXil MFS component and has been integrated in the embedded application running on the SoC using specific C function calls and libraries.

The programmable logic of the Zynq SoCs is prepartitioned with 4 partially reconfigurable regions, all having the same size, available resources (logic cells and bRAMs) and same generic I/O ports. A generic module has been instantiated in each partially reconfigurable regions, which have been integrated as peripherals of the embedded system. These modules can host user-defined logic—IP (Intellectual Property) Cores—and can be re-configured at any time via the Web service. The module's standardization has to be taken into consideration by users developing designs to be deployed on these processing cores, since they have to constrain their top-level module to the standard I/O interface, or encapsulate it in a wrapper module. The internal architecture and communication interface of the Zynq SoC is described in Fig. 3.

The key building block of our platform, which basically enables the ''hardware multitasking'' feature that allows several users to run their designs simultaneously on the same SoC device, is dynamic partial reconfiguration. The embedded programmable logic of the Zynq is being either complete (at board start-up) or partially (when a user takes control of a processing core) reconfigured through the device configuration (DevC)/Processor Configuration Access Port (PCAP) interface by a dedicated method of the embedded C application. This

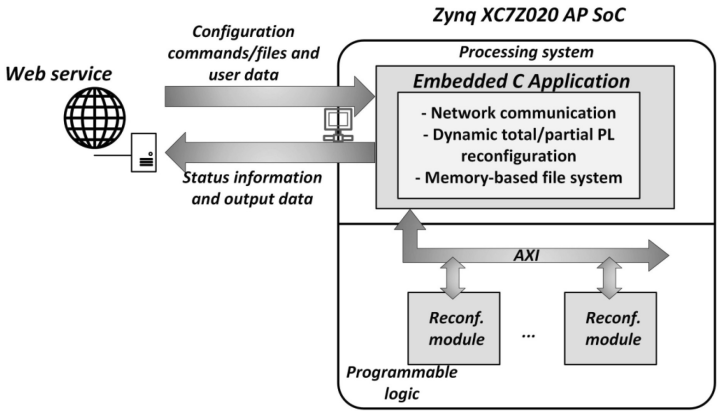

Fig. 3. Internal configuration and communication of a Zynq SoC.

means that each partially reconfigurable region can be re-configured at any time during the SoC's operation, without interfering in any way with the functionality of the rest of the chip.

3.3 User data transfer interface

After the student is being granted access to a reconfigurable processing core on one of the development boards, he can execute his design by (re)configuring the core with his own bitstream(s). He is further on provided with a basic interface for communicating with his design: eight 64-bit software registers, 4 input/4 output, can be used for data transfer to/from the IP core that is running on the board.

Basically, the student can perform two operations: "writing" the values entered in the online form to the board, and "reading" the content of the output software registers by displaying them on the webpage. This simple to use communication interface (shown in Fig. 4) provides the student with important feedback regarding the correct operation of his design, and allows him to perform experiments by sending data to be processed on the board and receive the results.

This data transfer interface has been implemented over the network by a TCP message exchange communication between the Web service and the development boards. On the Zynq SoC side, a mechanism has been designed to facilitate data transfer from the embedded C application running on the processing system (which handles the TCP communication with the Web service) and the processing cores from the programmable logic.

The application can receive either a write or read request encapsulated in a TCP message containing

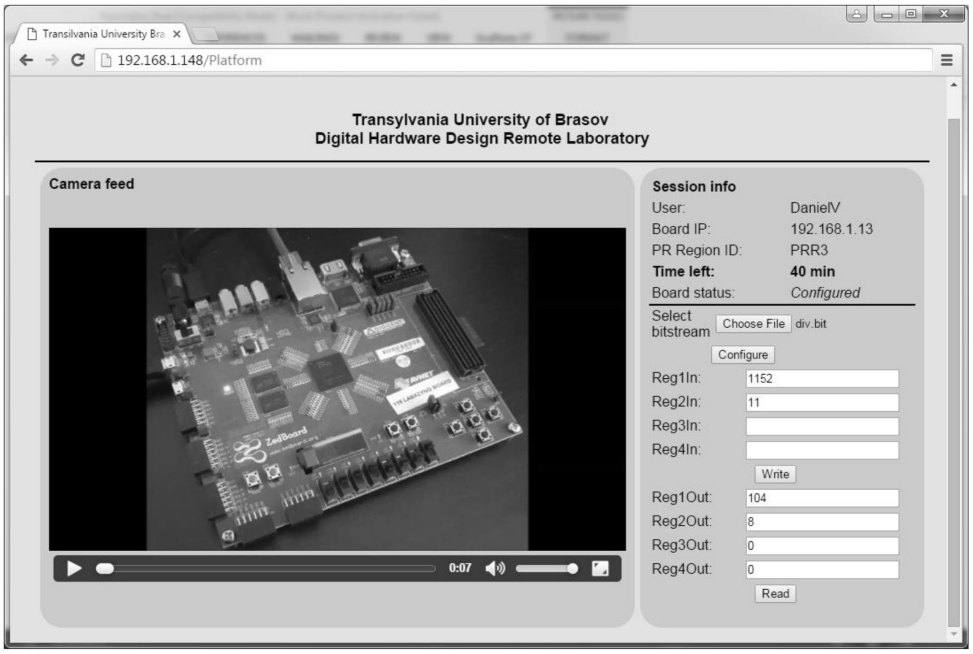

Fig. 4. Remote lab user interface

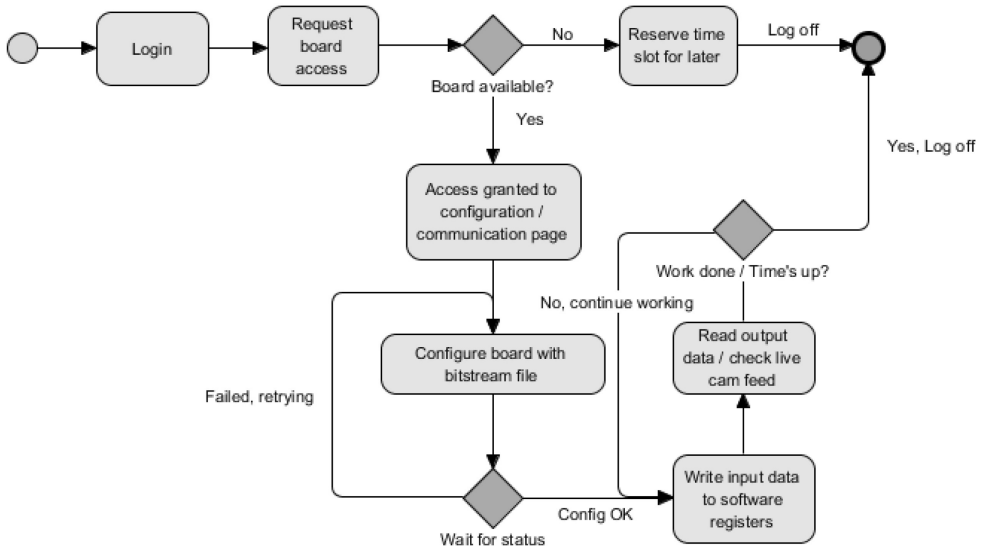

Fig. 5. User interaction scenario diagram.

the IP core's ID. A dedicated C sub-routine accesses the software registers of the indicated IP core and performs the required operation, returning either a completion status (when writing), or the register's values (when reading), information which is sent back via TCP communication to the Web service and is displayed on the Web page.

3.4 Usage and applications

The efforts in the development of this remote hardware design laboratory targeted achieving an interactive environment capable of offering a hands-on experience as realistic as possible using a reduced number of resources (by the means of the ''hardware multitasking'' feature described above) while ensuring the safety of the remote equipment.

This remote learning platform has the following modus operandi (depicted in Fig. 5): students can access the remote virtual platform online, via the JSP interface, through a login with their credentials. Each user then makes a request on the JSP page for a board to deploy his configuration file on, request that is sent to the scheduler method of the Web service. The service manages the network of Zedboards connected onto a LAN with the server by keeping track of the available and in-use boards, and chooses a vacant device to be assigned to the particular user that made the request. If there are no vacant boards at the time of the request, the user is offered the possibility to reserve a slot at a later time when there is availability and which is convenient to him.

When a board is available, its IP address and configuration status are sent as a reply message from the Web service to the JSP page in order to inform the user accordingly that he has been given access. Further on, the user is redirected to a new page that shows a live camera feed of their target board, information about it (its IP address and the ID of the reconfigurable module he has been granted access to) and allows for uploading a bitstream file to be used for configuration. After configuring the module with his design's bitstream, the I/O registers are displayed on the Web page, for sending and reading input/output data to and from the design that was loaded on the board.

The exercises and applications comprised in the hardware design course and which student experimented on using the remote platform ranged from introductory combinational logic/sequential logic circuits like gates, adders, counters, multipliers, dividers, finite state machines (Mealy and Moore) leading up to more complex mathematical applications (like encryption modules, encoders/decoders, floating point or complex adders and multipliers). These applications are compatible with the I/O interface described above and also allow for visual demonstration by outputting results on the board's LCD display or LEDs, visible on the live camera feed.

The most advanced laboratory applications involve hardware-based design of mathematical models and algorithms used in the study of photovoltaic panels. Such an example of implemented laboratory application is the photovoltaic panels (PV) study depending on the irradiance and temperature level. The PV is modelled based on the one diode model [18].

Based on an iterative process, implemented in the hardware application core running on a Zynq SoC, the current-voltage (I-V) characteristic is obtained. The students can easily vary the level of the irradiance and/or temperature level and observe and understand their effects on the PV behavior (Fig. 6).

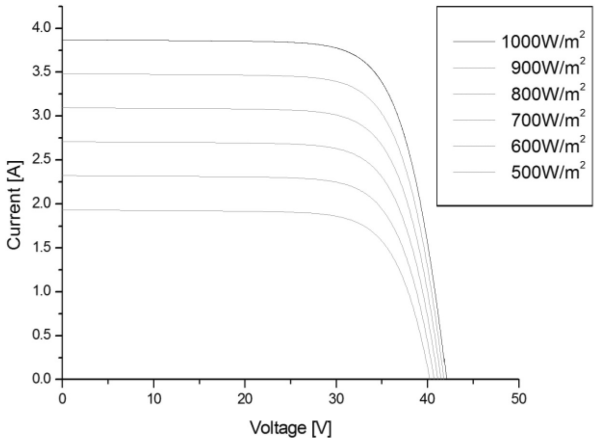

Fig. 6. PV panel I-V characteristic dependence on the irradiance level.

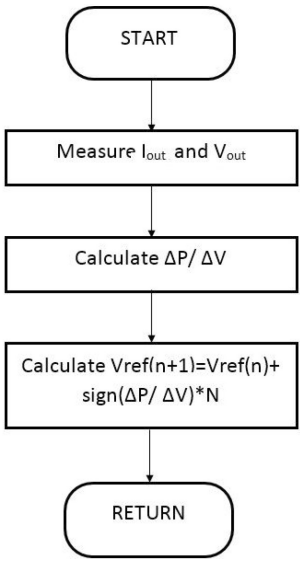

Fig. 7. Flowchart of P&O algorithm implemented.

This is simply done by entering the desired values using the Web interface, computing the numbers on the Zynq processing core, and reading the output values.

Having the PV panel model implemented, the students can successfully pass to the next step to study the PV panel as an energy source. In order to extract the maximum energy from the PV panel, the MPPT (Maximum Power Point Tracking) systems should be used. The students should implement an MPPT algorithm and apply it to the already implemented PV panel model. A common MPPT algorithm used is the Perturbation and Observation one [19].

A simplified flowchart is presented in Fig. 7. The algorithm supposes that the output current and voltage of the PV panel are known (measured). The output power P(n) and the power and voltage variations from the previous iteration are calculated

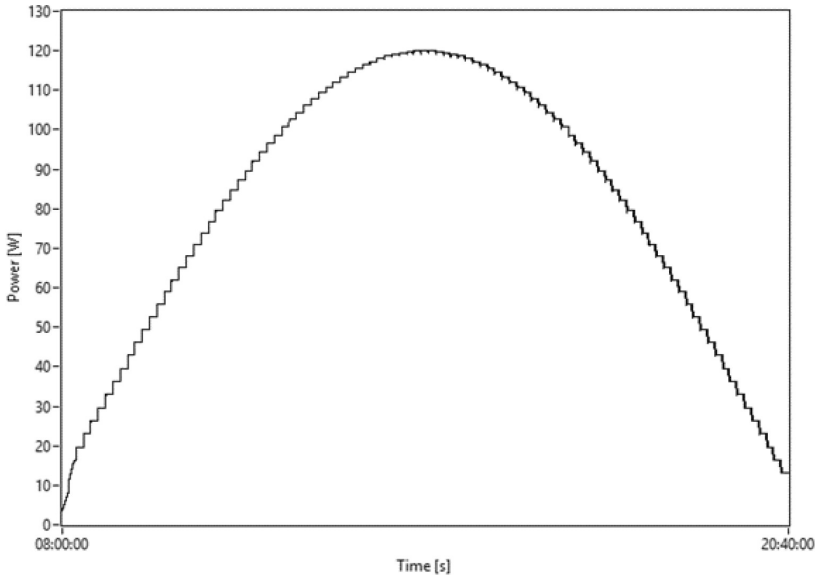

Fig. 8. PV panel response on the MPPT P&O algorithm.

$\Delta P = P(n)–P(n–1)$ respectively $\Delta V = V(n)–V(n–1)$. Depending on the sign of the ration between ΔP and ΔV, the reference voltage of MPPT system is determined and used in the next iteration.

The students implement the algorithm using Verilog HDL (Hardware Description Language), simulate their design and once validated, they deploy it on one of the available processing cores of the remote learning platform. Further on, they are supposed to run it by making several iterations (steps) over a certain time interval and write down the output values. Running this algorithm, the PV panel response is obtained and can be graphically represented based on the values obtained. Fig. 8 shows the PV panel response based on the MPPT algorithm under a sinusoidal variation of the irradiance level.

A similar approach can be applied for studying other renewable energy sources, like Peltier modules or wind turbines. This represents another important asset that the remote learning platform described brings, by allowing a large number of students to study such systems not only theoretically, but to experiment and make real measurements with the help of the hardware-implemented simulation models hosted by the Zynq SoC and the easy to use Web interface for data communication.

4. Discussion

After successfully designing and implementing the system, the next step was to test and evaluate it from two perspectives: technical and educational. In other words, the evaluation efforts targeted to check whether or not the entire system is feasible from a technical point of view and to evaluate the impact of its use on students during a digital hardware design course.

4.1 Technical assessment

The technical validation process of the entire remote learning platform consisted of performing experimental setups involving:

(a) Testing the Web service and application—Web interface, access scheduler, and board configuration and communication interface.
(b) Testing the embedded C application running on the SoC processing system with all its functionalities: PL total/partial configuration and network communication with the Web service.

In order to do the above, we have created several hardware designs and implemented them on the available boards via the online interface. The board's PL has been partitioned in two reconfigurable regions, with the available resource count displayed in Table 1. The designs implemented ranged from mathematical modules—multipliers, to encryption modules like AES (Advanced Encryption Standard) and SHA-2 (Secure Hash Algorithm-2) hash algorithms, processing cores similar to the ones most likely to be implemented by students at the hardware design course or for their individual practice.

The reconfigurable partition resource availability criterion has been chosen for the system validation process because it is vital to ensure that, after partitioning the PL, each reconfigurable partition offers sufficient resources for implementing the most common designs—included in the hardware design course curricula and other designs that students are more likely to implement for their term projects or just practice.

We have thus validated successfully the Web interface and board configuration/communication components of the Web services, and for testing the scheduler service, several accounts were created and simultaneously access requests were launched in order to make the schedule "busy".

On the SoC part, two issues were investigated:

1. The network transfer time of the total/partial configuration files and input/output data
2. The partial reconfiguration time needed for configuring the available processing core each time a student sends his bitstream via the Web interface.

The timing measurements were considered to be very important in validating this system as being operational in "real-time", in other words, the experience provided to the students accessing the virtual lab should offer response times regarding board access (for configuration and data transfer) as similar as possible to the ones in the case of direct, hands-on usage of the board. Thus, we had to ensure that bitstream transfer, partial reconfiguration of the specific region, transferring input/output data and the timings referring to the Web interface are inducing delays as small as possible, since high delays would translate in big lags being experienced by the student on the Web interface when configur-

Table 1. Resource availability for the partially reconfigurable regions of the Zynq SoC

Resource type	Available per region
Lookup Tables	4000
Flip-flops/latches	8000
L-Logic slices	650
M-Logic slices	350
DSP48E1 slices	20
FIFO18E1	10
RAMB18E1	10
RAMBFIFO36E1	10

Table 2. Timing analysis for Zynq reconfiguration and communication

Action	Average duration
Loading partial bitstream	2.25ms
Data transfer Server – SoC	16.1ms
Design execution time	8.25ms
Data transfer SoC – Server	18.4ms

ing or communicating with the board, rendering the remote learning platform inefficient.

The timing analysis results are shown in Table 2.

The partial bitstream files transferred via the network had around 215kB in size and the Gigabit Ethernet communication between the board and the Web service over the network achieved an average throughput of 80MB/s. The above figures show that the total time needed for input and output data to be transferred between the server and the SoC board was about 35ms.

Regarding the second item under evaluation, the timing results clearly show that the overhead induced by the partial reconfiguration of the module with the student bitstream file is negligible compared to the overall timing of the design operation.

The timing and performance analysis performed showed in the end that no noticeable delays are being induced by the Web service architecture, network communication, or SoC configuration, the e-learning platform being able to offer virtually instant responses to the students' requests.

Another important fact is that this approach leads to a drastic decrease in the expenditure with hardware resources since the same reconfigurable SoC board is used for the practice of several students (offering the possibility of simultaneously accommodating up to 4 users). This feature is of great importance for future developments since almost 100 students attend each year digital hardware design courses at our Faculty.

Last but not least, the architecture of this remote virtual platform, being based on Web services, favors sharing not only between students from the same institution, but also on an inter-institutional level [20], an important asset that meets the European trend in creating inter-institutional online digital design labs [21].

4.2 Evaluating the learning impact

In order to obtain feedback from the students regarding their experience with the remote hardware design laboratory two directions were followed:

(a) A survey consisting of questionnaires handed to students and submitted anonymously.
(b) Measurements of the total time the remote platform has been accessed for experiments, and the number of distinct students that requested and were granted access.

We have designed a survey mapping the most important objectives in the evaluation of the remote platform. The population for this survey included 28 students that worked effectively with the remote platform during a semester.

When developing the survey, we have validated the items based on the four validity-criteria (face, content, construct and criterion-related validity). Regarding the content validity, the items were validated with other department professors in the digital hardware design field to check that they are pertinent to the area of study. The criterion-related validity was checked by discussion with students and analyzing their feedback. We are currently working on (based on further results and survey answers) analyzing and constantly validating the survey from the construct-validity point of view – this being an iterative process.

The survey was composed of the following statements, the students being asked to evaluate them with grades designed on a Likert scale from 1 to 5 (1: strongly disagree; 2: disagree; 3: neither agree nor disagree; 4: agree; and 5: strongly agree):

1. The Web interface is intuitive and easy to use
2. This online remote lab helps gain more practical experience since the development boards can be used virtually any time for practice
3. Having the possibility of direct individual access to such a development board for experiments improves understanding of the concepts related to digital hardware design
4. The online learning and experimenting platform increases practice time since there is no need to install and manage board drivers and tools locally
5. There are enough available hardware resources for accommodating a design on each processing core
6. The I/O data transfer interface is practical and sufficient for the user's needs

The first question regards the graphical Web-based interface of the remote laboratory, a key element of the entire setup. The next three questions focus on assessing the qualitative impact that the remote lab had on the learning process with regard to the quality and quantity of the practical experience provided to students. Finally, the last questions are related to the adequacy of available hardware resources and the user interaction with the remote board (data communication interface).

The average values of the student's responses are shown in Fig. 9. These results showed that the main

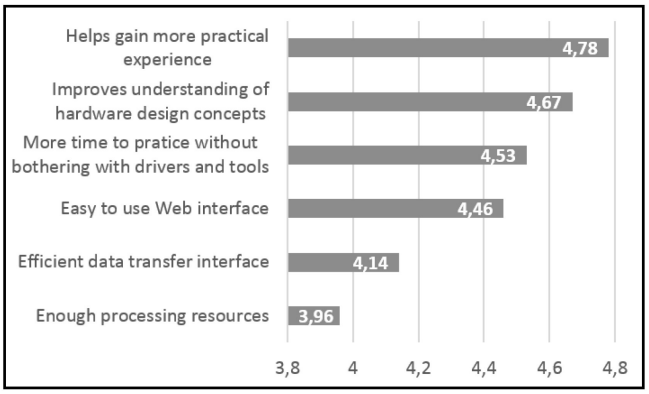

Fig. 9. Average student evaluation of the survey statements.

objectives of this remote laboratory received the highest average of students' responses. This proves on one hand that by being provided online, around the clock access to development boards for performing experiments, the students feel they have gained more practical experience, and on the other that, having direct, individual access, helped them better understand the concepts of hardware design (compared to the case where they worked on-premise in lab, but in teams).

Another asset underlined by students in their feedback was that this approach allowed them not to bother with installing the board drivers and other specific tools locally, so they were able to focus solely on the design and implementation of the hardware system. This is also an important advantage for the academic staff—professors, technicians, and other personnel—who can thus focus on improving the teaching material and not on managing the laboratory infrastructure (since only a server and network of boards are needed, instead of tens of workstations).

Furthermore, the students considered the Web interface of the remote platform as being easy to use and practical. In this way, we assessed whether the Web pages are intuitive or raise any difficulties in using them to get the tasks done (require and receive access to a board, configure it and communicate with it).

The lowest average values of student preferences were obtained by two items regarding the data communication interface and the processing core's resource availability. The first issue is due to the fact that the interface was conceived exclusively for educational purposes, so it does not allow a very fast or very high data traffic. The second is due to the particularities of the implementation, is flexible and subject to change based on the type of applications to be implemented: by reducing or increasing the number of reconfigurable modules available on each board, their resource count decreases/increases accordingly.

By analyzing the Web analytics data regarding the online usage of the remote laboratory, it was shown that all the students in the target group accessed the platform and performed experiments not only during the classes, but also at other times during the working days, and even in the weekend. This shows that having a remote, web-based access to such resources offers more possibilities for students to experiment real, practical hardware design since they can do this outside the regular courses and do not necessarily to be at the faculty premises, being provided online access to the development boards, via a Web page.

Lastly, we considered that another important validation of the lab regarding the impact on learning would be comparing the scores obtained for the digital hardware design semester projects by the students who benefited from access to the remote platform, with the ones of the students in the previous year when this platform was not available. Since the subjects of the semester projects remained the same, we were interested in finding out if having the chance to practice on the development boards outside the regular laboratory classes actually made a difference regarding the students' ability to practically design and implement a digital circuit. The results have shown that there has been an increase of 9% of the grade average (8.97 vs. 8.23) in the case of the students who accessed the online learning platform. This could be explained by the fact that, being able to practice more with their designs on the development boards, the students had the chance to better identify and correct flaws and bugs which may have went unnoticed in the simulation.

5. Conclusions

In this paper, a novel design and framework for an online remote digital hardware design laboratory has been presented. The main components of the remote lab have been described: the Web application incorporating a JSP user interface, a Web

server responsible with user interaction, access scheduling and management and communication with the hardware resources, and a network of System-on-Chip platforms able to host and execute the students' designs.

Compared to other current remote access laboratories (identified and briefly described in this work), our platform has the advantage of a superior efficiency with regard to the number of hardware resources needed, since the same development board can simultaneously be accessed for practice by up to four students. This key feature, a form of "hardware multitasking", has been implemented by taking advantage of the dynamic partial reconfiguration property of the Zynq SoC devices.

Following the technical and architectural details of the platform, the working methodology has been described, together with an application example (a mathematical model of a renewable energy source) that the students implemented using the remote lab.

Finally, an assessment was made of the results obtained by using this platform, both from a technical perspective and by analyzing the outcome on the learning process. The latter issue has been evaluated by providing a questionnaire to the students who used the remote lab during a semester. The results have shown that students consider such an approach essential in providing the opportunity of increasing their practical experience with real reconfigurable embedded platforms. They can thus practice whenever they want—outside the standard course/laboratory hours, which is a huge improvement over the traditional laboratory approach. As a result, with this virtual platform the constraints (spatial/temporal) of a traditional digital system design laboratory have been overcome and students were able to experiment in their free time, from any remote location.

Regarding future developments, we are currently focusing on improving the remote learning platform by adding new features, expanding the range of exercises available for practice, and adding new peripheral hardware. Also, based on the requests received from several students who were interested into the possibility of using this platform for implementing and testing more complex designs (like final degree projects), we are working on extending the I/O communication with the board, allowing more control to the users. This will be accomplished by adding to each board a Digilent PmodNIC100 peripheral module (featuring a Microchip ENC424J600 Stand-Alone Ethernet Controller) and it would offer the user a dedicated network communication interface for his design providing the possibility of direct, real-time high-speed network data transfer to the board.

Another important future research line we are considering is the integration of our system with an existing LMS (Learning Management System), which would enhance both student and teacher virtual experience and also allow an improved supervision of the learning process and assessment of the learning duration, quality and outcome. Such an integration would be facilitated by the existing SOA-based organization of our platform.

Last but not least, since the hardware infrastructure of the remote lab is available via a network, it is possible to easily integrate several workspaces comprised of development boards situated at different locations. This feature offers important collaboration opportunities on a worldwide, inter-institutional level, a research direction that we are also focusing on since it allows providing online access to hardware design resources for students from institutions where such devices may not be affordable or widely available.

Acknowledgments—This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS-UEFISCDI, project number PN-II-RU-TE-2014-4-1083.

References

1. C. Wang, X. Li, P. Chen, J. Zhang, X. Feng and X. Zhou Regarding Processors & Reconfigurable IP Cores as Services, In *Proceedings of IEEE 9th International Conference on Services Computing*, pp. 668–669, 2012.
2. D. Lowe, Evolving Remote Laboratory Architectures to Leverage Emerging Internet Technologies, *IEEE Transactions on Learning Technologies*, **2**(4), 2009, pp. 289–294.
3. J. Trevelyan, Lessons learned from 10 years experience with remote laboratories, In *Proceedings of the iNEER International Conference on Engineering Education and Research "Progress through Partnership"*, Olomouc and Bouzov Castle, Czech Republic, pp. 1562–3580. 2004.
4. R. Hashemian and J. Riddley, FPGA e-Lab, a technique to remote access a laboratory to design and test, In *IEEE International Conference on Microelectronic Systems Education*, 2007. MSE'07, pp. 139–140. IEEE, 2007.
5. R. Hashemian and T. Pearson, A low-cost server-client methodology for remote laboratory access for hardware design, In *39th IEEE Frontiers in Education Conference*, 2009. FIE'09, pp. 1–5. IEEE, 2009.
6. M. S. Bascil, I. Yazici and F. Temurtas. A FPGA based remote accessible digital system laboratory prototype, In *2012 International Conference on e-Learning and e-Technologies in Education (ICEEE)*, pp. 49–53, IEEE, 2012.
7. M. A. Conde, F. J. García-Peñalvo, M. Alier, E. Mayol and C. Fernández-Llamas, Implementation and design of a service-based framework to integrate personal and institutional learning environments, *Science of Computer Programming*, **88**, 2014, pp. 41–53. doi: http://dx.doi.org/10.1016/j.scico.2013.10.012
8. N. Koike, Cyber laboratory for hardware logic experiments: A seamless integration of actual laboratory and remote laboratory. In *2012 International Conference on Information Technology Based Higher Education and Training (ITHET)*, pp. 1–5. IEEE, 2012.
9. V. J. Harward, J. A. Del Alamo, S. R. Lerman, P. H. Bailey, J. Carpenter, K. DeLong and C. Felknor et al., The ilab shared architecture: A web services infrastructure to build communities of internet accessible laboratories, *Proceedings of the IEEE* **96**(6), 2008, pp. 931–950.
10. D. Pop, D. G. Zutin, M. E. Auer, K. Henke and H.D.

Wuttke. An online lab to support a master program in remote engineering, In *2011 First Global Online Laboratory Consortium Remote Laboratories Workshop (GOLC)*, pp. 1–6. IEEE, 2011.
11. V. Fotopoulos, I. A. Spiliopoulos and A. Fanariotis, Preparing a remote conducted course for microcontrollers based on Arduino, In *Proceedings of 7th International Conference in Open and Distance Learning* (Athens, Greece, 2013).
12. V. Fotopoulos, A. Fanariotis, T. Orphanoudakis and A. N. Skodras. Remote FPGA laboratory course development based on an open multimodal laboratory facility, In *Proceedings of the 19th Panhellenic Conference on Informatics*, pp. 447–452. ACM, 2015.
13. D. G. Zutin. Invited Paper—The iLab-Europe Initiative: Exploiting Possibilities Created by a Network of Shared Online Laboratories, *ASEE Annual Conference and Exposition Frankly*, American Society for Engineering Education, January 2013.
14. B. Perry, *Java Servlet & JSP Cookbook*, O'Reilly Media, Inc., 2004, pp. 119–120.
15. http://www.zedboard.org ZedBoard Development System (accessed 21 Dec 2015).
16. www.xilinx.com/zynq Zynq-7000 All Programmable SoC (accessed 8 Dec 2015).
17. A. Sarangi, S. MacMahon and U. Cherukupaly, Light-Weight IP application examples, Xilinx Corp., San Jose, CA. Application Note XAPP1026 v5.1, 2014.
18. D. Sera, R. Teodorescu and P. Rodriguez, PV panel model based on datasheet values, *International Symposium on Industrial Electronics*, pp. 2392–2396, IEEE, 2007.
19. S. Sengar, Maximum Power Point Tracking Algorithms for Photovoltaic System: A Review, *International Review of Applied Engineering Research*, **4**(2), 2014, pp. 147–154.
20. F. Morgan and S. Cawley, Enhancing learning of digital systems using a remote FPGA lab, In *2011 6th International Workshop on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, pp. 1–8. IEEE, 2011.
21. J. Vandorpe, J. Vliegen, R. Smeets, N. Mentens, M. Drutarovsky, M. Varchola and K. Lemke-Rust et al. Remote FPGA design through eDiViDe—European Digital Virtual Design Lab, In *2013 23rd International Conference on Field Programmable Logic and Applications (FPL)*, IEEE, 2013.

Octavian M. Machidon graduated as B. Eng. in Applied Electronics from the Electrical Engineering & Computer Science Faculty at Transylvania University in 2009 and as M. Eng. in Integrated Electronic & Communication Systems—specialized in Embedded Systems—from the same Faculty in 2011. He obtained the Ph.D. degree in Electronics Engineering and Telecommunications at the same university. Since 2015 he is an Assistant Lecturer at the university's Electronics and Computers Department. Before his academic career, he worked in the semiconductor industry at eASIC Corporation, as an engineer responsible with the design and implementation of integrated circuits in the eASIC technology. His current research interests cover VLSI microelectronics, embedded systems, remote engineering, reconfigurable and distributed computing, and the ethics of information and communication technology.

Alina L. Machidon received the B. Eng. degree in Information Technology in 2011, followed in 2013 by the Master's degree in Advanced Automated Systems and Information Technologies, both from Transylvania University of Brasov, Electrical Engineering & Computer Science Faculty. Since 2010 she has been working in the Software Industry on Mobile Application and Web Development. Her current research interests include mobile computing, web services, web applications, cloud/distributed computing and remote engineering.

Petru A. Cotfas received the B.E. degree in Mathematics and Physics and also in Computer Science in 1997 and 2001 respectively, and the Master's degree in Mathematics and Computer Science at Transylvania University of Brasov, in 1998. He obtained the Ph.D. degree in Material Science Engineering at Transylvania University of Brasov, in 2007. In 1999, he joined the Physics Department and from 2011 he became member of the Electronics and Computers Department. In 2015 he became an Associate Professor. Dr. Cotfas' current research interests include remote engineering, virtual instrumentation, renewable energy and nondestructive testing. He is also a member of the Institute of Electrical and Electronics Engineers, the International Association of Online Engineering, the Romanian Physical Society and NI community.

Daniel T. Cotfas received the B.E. degree in Mathematics and Physics at the Transylvania University of Brasov, Romania, in 1995 and the master degree in Mathematics: Statistics, probabilities and Systems reliability from Transylvania University of Brasov, Romania, in 2002. He obtained the Ph.D. degree in Industrial Engineering at the Transylvania University of Brasov, Romania in 2008. In 2002, he joined the Physics Department, Transylvania University of Brasov and in 2011 he joined the Electronics and Computers Department. In 2015 he became an Associate Professor. His current research interests include: renewable energy (photovoltaic energy), solar cell testing, optoelectronics, virtual instrumentation and remote engineering. He is a member of the Institute of Electrical and Electronics Engineers (IEEE), the International Association of Online Engineering (IAOE), Physics Romanian Society (SRF), European Optical Society (EOS), Leonardo Virtual Community. He won three awards at the Graphical System Design Achievement Awards at NIWeek 2013, Austin, USA with the Renewable Energy Laboratory Using NI ELVIS, NI LabVIEW, and NI myDAQ and Gold medal at Euroinvent, Iasi Romania, 2015.