FPGA-Based Remote Laboratory for Digital Electronics

Óscar Oballe-Peinado
Departamento de Electrónica
Universidad de Málaga
Málaga, España
oballe@uma.es

Julián Castellanos-Ramos
Departamento de Electrónica
Universidad de Málaga
Málaga, España
jcramos@uma.es

José Antonio Sánchez-Durán
Departamento de Electrónica
Universidad de Málaga
Málaga, España
jsd@uma.es

Rafael Navas-González
Departamento de Electrónica
Universidad de Málaga
Málaga, España
rjnavas@uma.es

Alberto Daza-Márquez
Departamento de Electrónica
Universidad de Málaga
Málaga, España
alma@uma.es

Jesús Alberto Botín-Córdoba
Departamento de Electrónica
Universidad de Málaga
Málaga, España
jesus.botin@uma.es

***Abstract* — Developing and implementing a remote lab based on FPGA can improve the teaching of Digital Electronics, giving students a web application to remotely test the designs proposed in practical sessions on real hardware. A server connected to the development system configures the programmable digital device with the proposed designs and allows the remote user to interact with the input and output peripherals. The server can manage connections and their time of use in order to avoid congestion of resources, and also allows usage statistics to be drafted.**

Keywords: e-learning, remote lab, digital electronics, FPGA

I. INTRODUCTION

Teaching Digital Electronics usually includes a practical component that is carried out in teaching labs. These practical sessions generally involve development systems based on a programmable logic device that, thanks to its versatility, allows digital circuits (both combinational and sequential) and embedded systems to be implemented. A computer application allows circuits to be designed, simulated, and implemented in the lab itself. However, students are now demanding greater availability of these labs in order to verify their designs on real hardware. This brings several problems, mainly a lack of availability of labs due to high occupation levels, the number of available boards being unable to meet peaks in demand, and the need for people to control the use of the development platform. A web application that can help to remotely verify the designs on real hardware, just like in the lab, is therefore a very interesting option.

This idea is not new [1], [2] and is used mainly by distance learning institutions, for obvious reasons. Most systems [3] currently use webcams to monitor behavior. This not only has a financial cost in terms of the camera itself, but also on bandwidth requirements between server and client systems.

It should also be noted that Spanish universities such as Universidad Nacional de Educación a Distancia (UNED) [4] and the Universidad de Deusto [5] implement remote labs using webcams.

The proposal presented in this paper is to create a remote lab that does not require webcams for feedback, using instead a virtual representation of the development system through a web application. Moreover, the peripherals available in the development kit are extended through the use of a microcontroller that simulates their behavior.

This work has been funded by Educational Innovation Project PIE19-141 of Universidad de Málaga.

The structure of this article is as follows: firstly, there is a general introduction to the work completed and the capabilities developed by the system. The architecture of the web application is then described with details of the design decisions, along with the application interface. This is followed by a discussion of the development of peripherals' virtualization firmware, and finally the conclusions reached.

II. DESCRIPTION OF THE WORK

This work aims to implement a remote lab system for Digital Electronics in which users can configure a programmable logic device remotely through their browser, verifying the behavior of their designs in the same way as they would do in the lab. This aspect is important, as the design for both remote and on-campus programming will be the same, without having to alter the circuit.

In our case, the remote lab is implemented through a server connected via USB to an FPGA development board (Digilent Inc. Nexys 3). [6] FPGAs are programmable logic devices capable of implementing any circuit or set of circuits, whether combinational or sequential. The only constraint is that the design must not use more resources than those of the device itself. This is not a problem to be considered here, given the educational nature of the designs in the practical lab sessions and their medium or low complexity.

Nexys 3 development board includes a series of basic peripherals (switches, pushbuttons) to stimulate the circuit inputs, and others (LEDs, 7-segment displays, etc.) to observe the behavior of the outputs of these circuits.

Moreover, the Nexys 3 has a series of expansion connectors (PMODs) that can be used to connect different peripherals to the FPGA. The catalog is very extensive [7] and includes all types of modules (sensors, A/D and D/A converters, etc.), bringing almost infinite possibilities in terms of developing practical training sessions.

Unlike other remote labs mentioned above, this proposal avoids the use of webcams to verify the system's behavior, thus decreasing the bandwidth needed in communications between server and client. To achieve this, a series of modules that allow excitation and monitoring of the status of the development board's peripherals must be incorporated in the design (in this case fully transparent to the student). The basic input/output peripherals in the Nexys 3 are implemented through registers included in the FPGA: read registers for output peripherals and write registers for input

978-1-7281-6732-9/20/$31.00 ©2020 IEEE

peripherals. Moreover, an Arduino board [8] connected to the Nexys 3 board's expansion ports simulates the behavior of the different expansion modules that could be used.

A web application allows users to access the server where access control is managed, establishing a time for use. Once logged in, the FPGA configuration file must be transferred before proceeding with digital design verification. Graphic components that act as virtual input (switches, pushbuttons, etc.) and output (LEDs, 7-segment displays) peripherals can be used to this end, as well as others connected to the expansion ports.

The following is a detailed description of the system's characteristics, for correct operation as a remote lab:

- Users can access the development platform remotely wherever they may be, this being the main benefit of this type of system. The web application must also be compatible with most browsers and platforms (PC, tablet, smartphone).
- New registrations are allowed, although user validation is supervised in order to ensure all system users are part of the educational community the tool is intended for.
- Users can amend their own details (login password, contact details, etc.).
- Users must log into the system, therefore allowing statistics to be drafted in order to evaluate the experience.
- Web application use time is controlled, and sessions will expire after a certain duration, thus guaranteeing the availability of the remote lab for different users in periods of high demand.
- An administration interface is available for user management.
- Users upload the configuration file with the design for the FPGA, and the system reports the result of the configuration (right or wrong).
- The basic FPGA input peripherals can be acted on remotely through their on-screen representation, transferring these actions to the device in real-time.
- There is also the option to choose virtual external modules to interact with in the designs.
- The information provided by the server about the status of the development board's output peripherals is read and displayed virtually.
- The system can communicate with the Arduino board, reporting the virtual module to be emulated and also sending and receiving data to interact with it.
- The remote development board behaves just like the actual system, meaning the design does not have to be modified when performing remote or on-site verification. The same configuration file can be used in both cases.
- The system guarantees high availability, with both the server and the development board powered by an uninterruptible power supply (UPS).
- The remote lab's web application must be robust, to ensure it works correctly whatever the interaction carried out on it.

III. Web Application Architecture

A web application is generally structured as a three-layer application, but in this case a four-layer structure has been considered (Fig. 1).

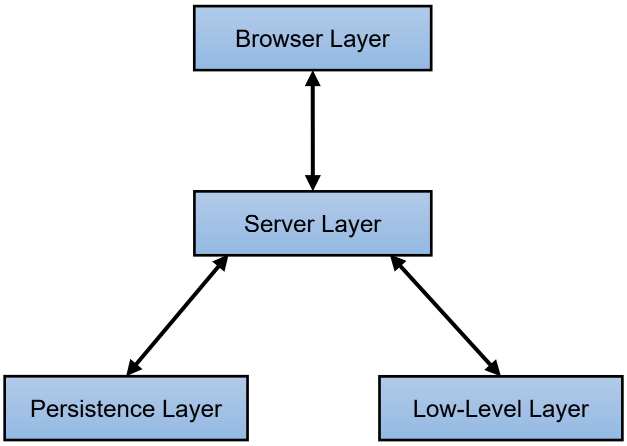

Fig. 1. Remote lab web application layers

The web browser sends requests to the server layer, that offers services using the persistence layer (database access), or to the low-level layer (FPGA access). Fig. 2 shows the remote lab system architecture.

All the technologies used in implementing the web application are open source.

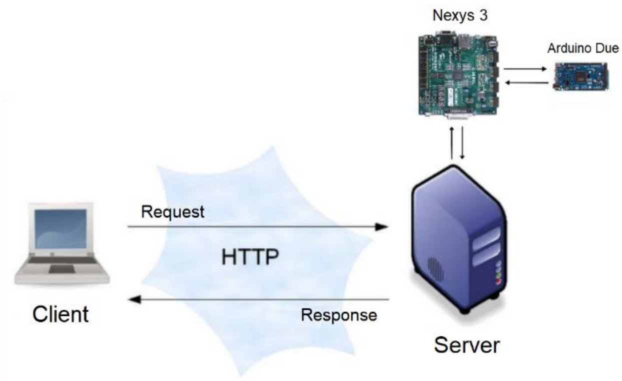

Fig. 2. Remote lab system architecture diagram.

A. Browser Layer

This is also known as the application's front-end and includes all technologies running on the client-side (i.e. all those running on the web browser), mainly in three languages: HTML, CSS, and JavaScript. These technologies are used to add certain functionalities based on interpreted languages (scripts) on the client-side, offering an interactive experience that does not require the web application's pages to be reloaded each time they are acted upon. Moreover, technologies have been developed to coordinate these languages with technologies on the server-side.

In the specific case of the remote lab, this layer resolves the requirements directly associated with the interface, such as forms and virtual controls.

The languages used in this layer are:

- HTML5 (HyperText Markup Language, version 5) and CSS3 (Cascading Style Sheets), for the rendering of the web application's pages in the different clients. Both technologies incorporate new features, providing a platform to develop complex web applications.
- JavaScript (commonly abbreviated as JS), an interpreted programming language used in its client-side form and implemented as part of the web browser, allowing improvements in the user interface and dynamic web pages.

Other related technologies have been used in addition to these languages.

For information transfer:

- AJAX (Asynchronous JavaScript And XML): a web development technique for creating interactive applications or RIAs (Rich Internet Applications). These applications run on the client, i.e. in the user's browser, while asynchronous communication with the server is maintained in the background. Changes on the pages can, therefore, be made without having to reload them, thus improving interactivity, speed, and usability in the applications.
- JSON (JavaScript Object Notation): a lightweight text format for data exchange. JSON is a subset of JavaScript's literal object notation, although it is today considered an independent language format due to its extensive use as an alternative to XML.

For fast, efficient development:

- Bootstrap: a web framework or open-source toolkit for designing web sites and applications. It contains design templates with typography, forms, buttons, boxes, navigation menus, and other design elements based on HTML and CSS, as well as optional JavaScript extensions.
- jQuery: a multi-platform JavaScript library for more straightforward interaction with HTML documents, and to manipulate the DOM tree, manage events, develop animations, and add AJAX interaction to web pages.

B. *Server Layer*

Numerous programming languages are used to develop web applications on the server-side. Of all the technologies available for this layer, it was decided to use the following:

- Python: a powerful, easy-to-learn programming language. It has efficient, high-level data structures, and a simple but effective approach to object-oriented programming. Python was used because Django was chosen for web design. Django is an open-source web development framework written in Python which respects the design pattern known as Model-View-Controller. Django focuses on reusability, connectivity, and extensibility of components, rapid development, and the principle of Don't Repeat Yourself (DRY). Python is used in all parts of the framework, including configurations, files, and data models.
- Node.js: a JavaScript runtime environment built with the Chrome V8 JavaScript engine. Node.js uses a non-blocking, event-oriented I/O operation model, making it light and efficient. In our case, Node.js was required to ensure efficient asynchronous communication between client and server sides. An optimal socket-based solution was found based on Node.js characteristics, such as non-blocking I/O and event-driven operation, achieving very acceptable response times.

C. *Persistence Layer*

This layer contains the database server required by Django. It has controllers for different database servers, although it was considered more appropriate to use MariaDB, which is a GPL (General Public License) database management system derived from MySQL. It also has high compatibility with MySQL, since it has the same commands, interfaces, APIs, and libraries.

D. *Low-Level Layer*

This layer houses the executables that allow physical access to the FPGA. Utilities provided by Digilent Inc. have been used for programming. These applications are invoked from the operating system. Moreover, one of the utilities provided by the manufacturer has been recoded to minimize access/response time, thus maximizing access speed. C language has been used to program this script, due to its low-level language characteristics.

IV. Web Application Interface

The remote lab interface is very simple and consists of only a few windows.

A welcome message appears on the application's home page, along with information about its features and the benefits of using it. There are links to the application's login and registration pages, as appropriate (Fig. 3).

Fig. 3. Remote lab home page.

The registration page includes a form to enter personal details, as well as a username, an email address, and a password (Fig. 4).

Fig. 4. Remote lab registration window.

Once inside the application, the main page shows a replica of the Nexys 3 development board, where graphic components have been incorporated to interact with its peripherals (Fig. 5). Moreover, a console has also been included to show events messages when configuring the FPGA or validating the design.

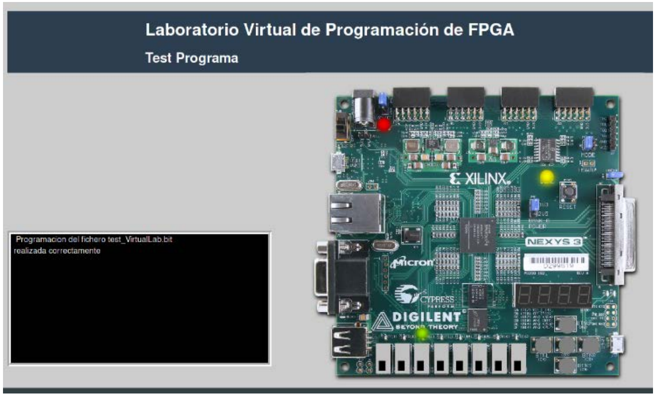

Fig. 5. Main remote lab window along with the message console.

A screenshot of the main screen is included, showing a virtualized hexadecimal keyboard (Fig. 6), as an example of using virtual peripherals connected to the FPGA's expansion modules.

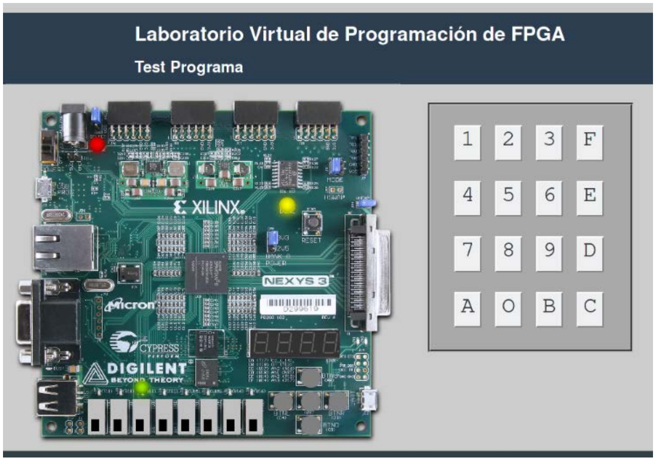

Fig. 6. Main remote lab window with virtualized external peripheral (hexadecimal keyboard).

V. Peripherals' Emulation Firmware

The various peripherals that can be connected through the Nexys 3 expansion modules (PMODs) are emulated by a microcontroller on an Arduino board. Since FPGAs are digital devices, the signals from the modules that can be connected to the PMODs are also digital and can be perfectly emulated by a microcontroller. The server oversees the programming of the Arduino board with the firmware that simulates the selected peripheral. Graphic control for the work window and the emulation firmware must, therefore, be developed every time a new peripheral is to be added to the catalog.

Fig. 7 shows the Nexys 3 development system with the Arduino UNO microcontroller connected to one of the PMOD connectors. In this case, a 5V (Arduino UNO working voltage) to 3.3V (PMOD expansion modules working voltage) voltage matching module was necessary.

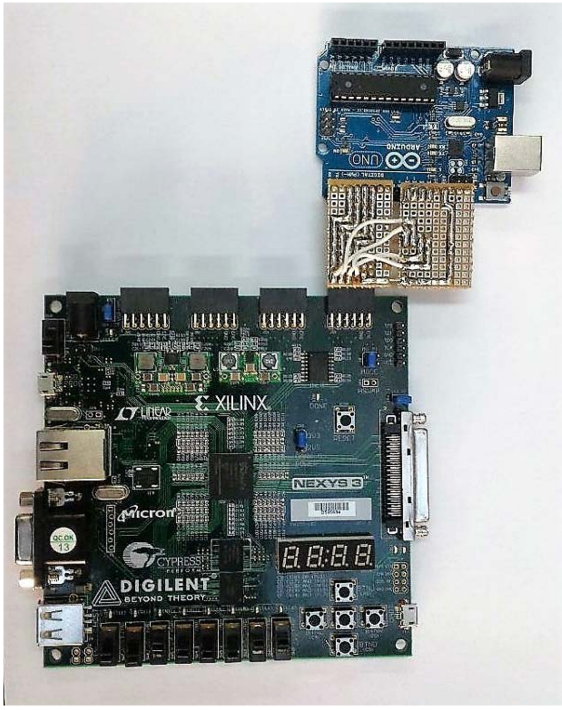

Fig. 7. Nexys3 development kit with Arduino UNO for emulation of peripherals.

VI. Conclusions

This work develops a web application as a tool for remote teaching of practical lab sessions in Digital Electronics. The application allows remote access to an FPGA-based development board (Nexys 3), as the correct operation of the designs can be checked by acting on the peripherals. Moreover, a microcontroller (Arduino UNO) emulates the behavior of a hexadecimal keyboard in order to demonstrate the feasibility of emulating external peripherals that could be connected to the Nexys 3 board's expansion connectors (PMODs). The remote lab's web application has been developed using free software.

The system offers students the benefit of being able to verify their designs remotely before being evaluated in the laboratory. Moreover, in terms of teaching, it allows use statistics to be generated in order to evaluate the impact of the experience. This work is being carried out as part of bachelor's and master's degree projects, developing both the remote lab software from the server's point of view and also new virtualized hardware components.

The system is currently in its first implementation phase, meaning it is being used by a small group of students in order to obtain information and statistics for a detailed analysis of the experience. The factors being evaluated are resource access times and adaptation to client needs, bandwidth consumed, options to scale the system according to demand, and limitations in the emulation of peripherals by the microcontroller.

References

[1] H. H. Saliah, M. Saad, H. Hassan, and I. de la Teja, "Virtually and remotely accessing and controlling laboratory real devices: a new trend in teaching and learning in engineering," in Proceedings of the International Conference on Information Technology Based Higher Education and Training, 2000, pp. 107–114.

[2] J. García Zubía, "Educational software for digital electronics: BOOLE-DEUSTO," in Proceedings - 2003 IEEE International Conference on Microelectronic Systems Education: Educating Tomorrow's Microsystems Designers, MSE 2003, 2003, pp. 20–22.

[3] M. Reichenbach, M. Schmidt, B. Pfundt, and D. Fey, "A New Virtual Hardware Laboratory for Remote FPGA Experiments on Real Hardware," in Proceedings of the 2011 International Conference on E-learning, E-business, Enterprise Information System E-Government (EEE), 2011, pp. 17–23.

[4] UNED, "Entrada al laboratorio virtual." [Online]. Available: http://ohm.ieec.uned.es/fpga_lab/. [Accessed: 16-Mar-2020].

[5] Universidad de Deusto, "WebLab-Deusto - Labs." [Online]. Available: http://weblab.deusto.es/website/labs.html. [Accessed: 16-Mar-2020].

[6] "Nexys 3 Spartan-6 FPGA Trainer Board - Digilent." [Online]. Available: https://store.digilentinc.com/nexys-3-spartan-6-fpga-trainer-board-limited-time-see-nexys4-ddr/. [Accessed: 19-Jan-2020].

[7] "Digilent Pmod Modules and Connectors - Interface with Development Boards." [Online]. Available: https://store.digilentinc.com/pmod-modules-connectors/. [Accessed: 21-Apr-2020].

[8] "Arduino Due | Arduino Official Store." [Online]. Available: https://store.arduino.cc/arduino-due. [Accessed: 19-Jan-2020].