A Remote FPGA Laboratory for Digital Design Students

João Soares*, Jorge Lobo
ISR - Institute of Systems and Robotics
DEEC, FCT, University of Coimbra
{*jsoares, jlobo*}*@isr.uc.pt*

Abstract

We have developed a new remote and interactive laboratory for engineering students who are learning digital design and need to test digital circuits on a FPGA board platform. The aim of this project is to make a more efficient use lab resources over the Internet, so that students can test digital circuits as if they were testing in loco and not in simulation. Students, or any one interested, are invited to interact with the remote board by viewing it on the webcam applet and control it by switching buttons virtualized on a simple control panel. To complement this, we have created a platform to support a simple database to store projects and cores, and a platform for simple quizzes and games.

1. Introduction

Although modern freely available digital design tools allow for flexible simulation vector editing and easy simulation, for first year students learning digital design the hands-on approach is much more rewarding, and less confusing. But lab class resources are limited, and without flexible working hours. The challenge is to transpose the physical interaction between the user and the equipment of the laboratory to an Internet application interface, as realistic as possible, so that students can interact with a real FPGA board at the lab from anywhere.

In our case we would like to connect to an ALTERA DE2 FPGA board [1] over the Internet and use it as if the user is *in loco*, programming, testing and looking at it. This board has got many switches, keys, LEDs, LCDs and 7-segment displays (fig. 1), and we need to look at it and use the switches and keys, remotely. This was accomplished by developing a webpage interface, with PHP dynamic functionalities that enabled the remote use of the DE2 boards and a webcam feedback.

We have also created two complementary platforms on the server. One is to support a database to store projects and cores made by students, researchers or professors who would like to contribute to the community. The other one is to support simple quizzes in which students can answer tests from o pool of questions, for testing their knowledge in digital design.

*The work of João Soares was partially supported at ISR-University of Coimbra by the FCT (Fundação para a Ciência e Tecnologia) scholarship BII/UNI/0048/EEI/2009/B02.

2. Related Work

Virtual labs provide an online visualization of some simulated experiment, the idea behind remote labs it to have an actual hardware experimental setup and allow the user to interact over the internet. The idea of creating remote labs is not novel, and there are multiple examples online of remote labs in diverse engineering fields. However specifically for FPGA based digital design there are not so many. In [2] a set of experiments are run on two FPGAs connected to a microcontroller. A full web interface allows experimentation around a fixed set of combinational and sequential circuits, with results shown alongside the schematic symbol or with timing diagrams. In [3] a FPGA board connected to a PC is used along with PC-based measurement equipment, such as logic analyzers and pattern generators, were used to develop a remote multi-user time-sharing hardware experiment system. In [4] a remote laboratory for a Xilinx board is presented. The infrastructure described gives remote users the ability to dynamically power on/off the FPGA boards, upload/download files, configure the boards online and execute synthesized designs while sending input and output through the Internet. These however are complex approaches. In our work we take a very simple but effective approach described in the following sections, suitable for the introductory nature of the courses attended by first year students, but not limited to any fixed set of experiments.

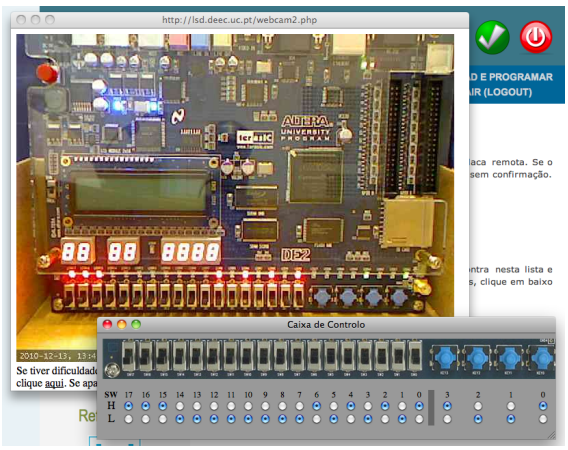

Figure 1. Screen shot of the remote laboratory interaction with the FPGA board.

3. The Remote FPGA Laboratory

The implemented remote FGPA laboratory is online at `http://lsd.deec.uc.pt`. To have access to the hardware a simple registration is required to have a user account. Figure 1 shows the remote interaction with the DE2 FPGA board.

A webcam conveys the realtime results, and the board switches and keys are virtualized. Students use a free web-edition version of the Altera Quartus II digital design IDE [5], and just upload the final programmer file (.sof). The only change they have to perform to the project is to use a component that virtualizes the board switches and keys. The DE2 Web Server is located in our Campus LAN and is accessible through the Internet, so that anyone from anywhere can remotely access the lab, as shown in figure 2. The server is connected via USB to the FPGA JTAG port on the DE2 board, and the webcam connects to the server using USB.

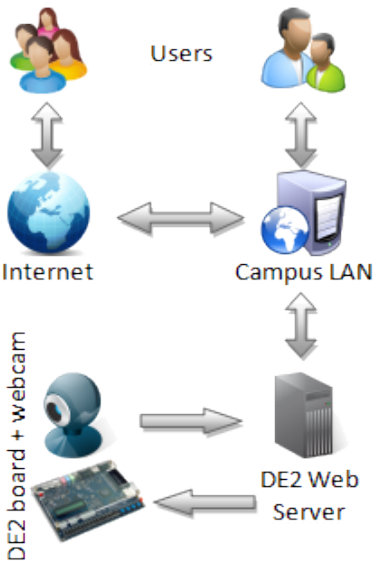

Figure 2. Network and connections for the Remote Laboratory.

3.1. Scope of Target Digital Designs

The remote laboratory is suitable for combinational and sequential circuits with a simple interface, but only limited in complexity by the Cyclone II Altera FPGA on the DE2 board. The range of switches, LEDs, 7 segment displays and LCD alone provide a rich interface, and are all readable on the webcam image.

The primary intended users of the system are first year students of digital design of the course on Electrical and Computer Engineering at Coimbra University. Some years ago the switch was made from traditional TTL logic to reconfigurable logic devices, with the focus on controllers implementing simple finite state machines, and moving up to register transfer level (RTL) design with datapaths and associated controllers implementing high level state machines. Schematic design entry is mostly used, with some components implemented in VHDL on the more advanced designs. A set of 9 lab assignments range from simple combinational circuits and a simple ALU implementation, to more complex sequential circuits for a vending machine, and a last assignment with a 3-instruction programable processor that the students have to enhance to a 6-instruction programable processor. With the exception of the VGA output used on the some of the last assignments, they can all be fully run on the remote lab, eventually with reduced clock speeds when the board clock is used.

3.2. Adapting the Design to Run on Remote Board

To test a digital circuit on the remote laboratory instead of directly on the DE2 board the project design needs to take into account the virtualized switches and keys, following the steps shown in figure 3.

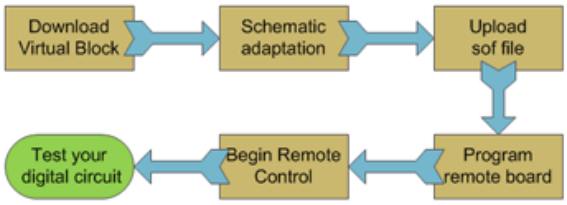

Figure 3. Diagram of the user required steps to test a digital circuit.

Before programming the remote board with the digital design compiled on Altera Quartus II, the user needs to modify the schematic in order to make the input pins liable to be changed via JTAG, instead of changed physically with the board switches and keys. In order for that, the user needs to download the virtual block and connect to it the input pins that corresponded to board switches and keys. This is shown for a simple combinational circuit on figure 4, the original circuit, and 5, the same circuit using the virtual block.

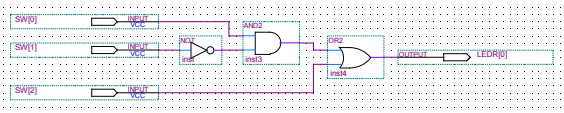

Figure 4. Simple combinational initial circuit.

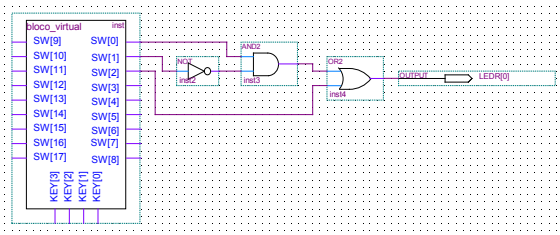

Figure 5. Simple combinational circuit using virtual block.

This is done in schematic design entry, and there is no

REC 2011 (ISBN 978-989-97151-0-3)

limit on the design complexity, as long as the slow actuation of switches and keys, and visual feedback of the board is sufficient to test the circuit. If the project is in VHDL or Verilog, the top-level file needs to be encapsulated in a block and placed into a new top-level schematic file connected with the virtual block for the virtual inputs.

After this simple adaptation, the user can login and upload the compiled .sof file to the server, reconfigure the FPGA and begin the remote control.

3.3. Circuit Upload and Interaction with Remote Board

To have access to the hardware the user must register and have the request accepted by the administrator. For students attending the course this can be done in class with a batch auto-validation mode available to the administrator. The hardware access policy is singe user time limited login, meaning that two or more users cannot access and control the remote board simultaneously. The license expires after 5 minutes, enough time to test the circuit, after which a new login is accepted. Users trying to login are informed of the remaining time when the hardware is already being used. This monitoring is an alternative to a queue system that is not yet implemented.

Once the .sof file is programmed in the remote board, the user can open the control box and the Webcam applet box. The control box has a total of twenty-two buttons, in which the eighteen switches and the four keys are simulated in the same way, since coding the keys in PHP was not practical. Figure 1 shows a simple circuit that connects the switches to the red LEDs and the keys to the green LEDs, notice the pattern on the control windows buttons and the LEDs on the webcam image. One of the limitations of the current virtual control system is that it takes one second for the change on a button to take effect on the board. This limitation is due to the simple approach taken that only requires a minimal virtual block and no additional hardware, as detailed in the following section.

4. Implementation Technical Details

Figure 6 shows the hardware used to implement the system, a standard PC, a DE2 board and a webcam. The PC is running Apache webserver and Altera Quartus II. The dynamic pages are implemented using Apache 2.2.14 with PHP 5.3.1. The applet for the Webcam is powered by Yawcam, free software available on the Internet.

4.1. Virtualizing the Board Switches and Keys

We required some way of conveying to the board the user intended state for the switches and keys. After considering several options from the available connectors on the board, such as USB, RJ-45, RS-232, and rejecting most due to the required custom development of a solution, we chose a simple way to achieve the required remote communication: In System Memory Content Editor of Quartus II. With this application, the computer could edit data stored

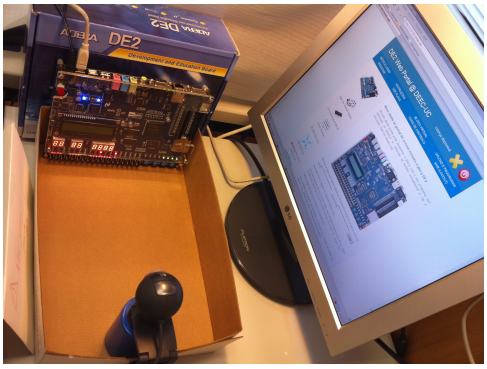

Figure 6. Server equipment: PC, DE2 and webcam

in the DE2 board while testing a circuit. Altera's In-System Modification of Memory and Constants is intended for debug purposes, and an equivalent functionality is also available for Xilinx FPGAs with the Digilent Adept software.

We thus created a virtual block for the switches and keys, made by content that is editable by this application. We set a LPM editable binary constant for each button and collected the twenty-two well-identified constants on a block. This block can be added to any top-level schematic, and connected to the input wires where the physical inputs were linked. A direct link with the PHP code was not possible, so we created a program in C++ to handle the command line console of Quartus II, and used a file based solution.

The control box is made of twenty-two pairs of radio buttons which simulate the twenty-two buttons (switches and keys). At first they are on the default position: switches on status OFF and keys on status ON. A change on a pair of radio buttons will produce a request on the server to create a file. The file contains all the data needed for a process handler that runs behind to find what was the button in question and what was the changed state, in just three numbers. The state information is needed because the server doesn't know the current state of every button, only the control box on the terminal browser knows.

The process handler, the C++ program that writes commands on the Quartus II SignalTap console, is a continuous cycle running at 50Hz that checks the existence of the mentioned data file (fig. 7).

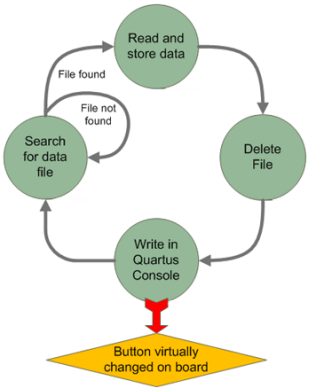

Figure 7. Cycle of the Process Handler

If the file exists, retains its data, delete it and write the proper command on the console to change the status of the desired button. After making that change, which lasts one second by limitation of the writing process (not because of the frequency), it starts over again.

On the client side, the user sees a message to wait a second until change in the button takes effect and then the page goes back to the control box, with the right current status of that button (stays in the browser memory).

4.2. Website Server and System Management

The server has a web interface backoffice to help the used account management and overall site maintenance. The Administrator can, for instance, see who's the user who is currently working on the board and may force him out if he was acting improperly, by cleaning his exclusivity license that gives him access. More than that, he can suspend temporarily his board account or even eliminate it in a simple click.

There is a special page where the Administrator can upload PHP pages, or any other file, for updating purposes, with instantaneous effect. There is also a permanent register to log all major events taken by the users and the Administrator. He can read this register to decide improvements on the system or detect users misuses or abuses. Every database of this system is a simple text file. Webpages programmed in PHP obtain all stored information, like the user account list, by reading text files and processing information to generate the webpages.

5. Additional Website Features

To complement the server functionalities two key features were added, as indicated in figure 8. A database to store projects and cores made by students, researchers or professors who would like to contribute to the community. A page to support simple quizzes in which students can answer tests from o pool of questions, for testing their knowledge in digital design, and also a java applet for digital circuit simulation [6].

To use the LSD OpenCores database the user must login, but this is distinct from the hardware access login and multiple users can use the database concurrently. When uploading the files the user is given an online form to provide some classification and description. For download the users can browse or search the database for specific types of code, keywords, etc. The quizzes page includes twenty-question multiple-choice quizzes, as well as interactive Flash quizzes. The system administrator can control these to grade student performance, since repeating the same test is allowed, but the scores are retained from the first trial.

6. Conclusions and Future Work

We have presented a simple approach for a remote laboratory using an Altera DE2 FPGA board. The system is targeted at introductory courses of digital design where a simple webcam can provide the feedback from the FPGA board LEDs and displays, and the board switches are virtualized for remote web control. The system is fully based on free software and no specific custom hardware is required other than a standard webcam and the FPGA board already used at the lab.

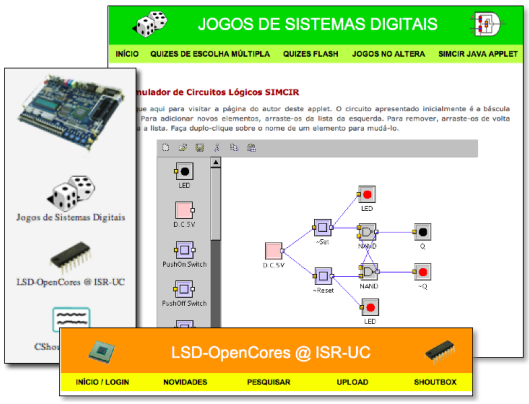

Figure 8. Partial views of some of the implemented additional website features, including the logic circuit simulation java applet Simcir [6].

The system was only implemented recently, and has only been running continuously in the last month of the semester. The systems logs show successful usage by the students, and some gave a very positive feedback, although a longer trial period is required to evaluate the student's response and impact on the learning.

Some issues need some improvement, such as the latency time on changing a switch. As future work we intend address this and to make this system a multi-user system, by configuring several DE2 boards (or other FPGA boards), revise access policies and implement a queue system. We are also considering adding a second webcam and sound, to allow for peripherals connected to the board, such as a VGA monitor, sound output or some custom experimental setup connected to the expansion connectors.

References

[1] Terasic. DE2 development and education board. *www.terasic.com*, last visited December 2010.

[2] L. Gomes, G. Patricio, R. Ferreira, and A. Costa. Remote experimentation for introductory digital logic course. In *E-Learning in Industrial Electronics, 2009. ICELIE '09. 3rd IEEE International Conference on*, pages 98 –103, 2009.

[3] Norihiro Fujii and Nobuhiko Koike. A time-sharing remote laboratory for hardware design and experiment with shared resources and service management. *ITHET 6th Annual International Conference*, July 2005.

[4] Kushal Datta and Ron Sass. RBoot: Software Infrastructure for a Remote FPGA Laboratory. *Field-Programmable Custom Computing Machines, Annual IEEE Symposium on*, 0:343–344, 2007.

[5] Altera. QUARTUS II . *www.altera.com*, last visited December 2010.

[6] Kazuhiko Arase. Simcir - logic circuit simulator. *www.d-project.com/simcir/*, last update: December 2009.