DOI: 10.35598/mcfpga.2023.011

Building a Virtual Hardware Laboratory with FPGA and Raspberry Pi Integration

Vladyslav Baida
ORCID 0009-0004-2172-8057
Department of Radio Engineering and Embedded Systems
Chernihiv Polytechnic National University
Chernihiv, Ukraine
vvmeansvlad@gmail.com

Sergii Ivanets
ORCID 0000-0002-9587-0783
Department of Radio Engineering and Embedded Systems
Chernihiv Polytechnic National University
Chernihiv, Ukraine
Sergey.Ivanets@gmail.com

***Abstract*— Nowadays, when the country is facing the coronavirus epidemic and the war, it is very important for educational institutions to provide their students with the opportunity to continue working remotely.**

All majors that require certain physical equipment have faced problems, as teachers need to find ways to provide students with all the equipment they need to study.

This problem was faced by those specialties that study hardware development, in particular digital circuitry using programmable logic integrated circuits.

Therefore, the issue of creating an application that will allow students to perform laboratory work online, for which they only need access to the Internet, is becoming relevant.

Keywords—FPGA, Raspberry Pi, Hardware, Remote Laboratory

I. Introduction

Practical and laboratory work is an integral part of education, as it allows students to consolidate the knowledge gained in lectures, gain an understanding of the practical use of this knowledge, and motivate them to continue their work.

In many specialties, such work requires the use of certain material resources, such as laboratory stands, etc.

Given that nowadays it is important to provide students with the opportunity to continue their studies despite external factors such as the epidemic and war, there is a need to create an application that will allow students studying digital circuitry to perform laboratory work remotely [1].

This application should be aimed at improving the quality of education in general, as it will make the learning process more adaptable to unforeseen situations in which attending higher education institutions becomes impossible.

The application for remote FGPA laboratory work should preferably be implemented as a browser-based application, as this is more convenient than using special client applications for a personal computer or phone.

An important component of the system that implements a remote laboratory is a video camera, which allows the user to see in real time what exactly is happening to the hardware during the work.

Also, since the user does not have direct access to the hardware, and therefore cannot physically interact with it, the application must provide the ability to press virtual keys that simulate real buttons in the hardware.

Also, based on the fact that the user does not have direct access to the hardware, the application should provide the ability to flash the FPGA with code written in the hardware description language.

This article discusses the principle of operation of the system, which allows you to perform laboratory work using FPGA boards remotely, namely to test the operation of digital circuits implemented on FPGAs, and the user only needs a device with Internet access

II. How to Interact with a FPGA Board

The FPGA laboratory work involves the student writing code using a HDL (Hardware Description Language), loading this code into the FPGA, and testing its operation.

Testing consists of applying certain signals to the logical inputs of the FPGA, reading the signals on the logical outputs, and checking the correlation of these signals for correctness.

So, to implement a remote laboratory, you need a device that has the hardware capabilities to set signals on its outputs and read signals on its inputs.

GPIO (General-purpose input/output) devices are suitable for these purposes, since the signals are a sequence of logical levels "1" and "0".

Thus, having a device with GPIO and the ability to write software for it, it opens up the possibility of building timing diagrams that display the FPGA's response to external signals.

As mentioned in the introduction, it is important to be able to flash the FPGA board with user-written code using a hardware description language, and therefore the system that implements the remote laboratory must be able to interact with the FPGA board through special interfaces for its programming. For example, JTAG [2].

III. Choosing the Hardware

In order to make a choice that balances the above characteristics, you should first analyze the hardware platform requirements in detail.

- First, since the main way to transmit signals in these boards is through I/O ports, the hardware module responsible for transmitting signals to and from the FPGA must have such ports. A microcontroller, another FPGA, or a single-board computer are suitable for this purpose.
- Secondly, the platform must be able to connect to other components of the system using network interfaces. This is difficult to do on an FPGA, somewhat easier on a microcontroller by using libraries for network protocols such as LwIP, and much easier and faster on single-board computers that can run programs written in languages higher than C.
- Thirdly, to scale and develop the project in the future, the platform should provide the ability to update the program code or firmware. This is quite convenient and easy to do on single-board computers running the Linux operating system, which allows remote system management, such as the SSH protocol, and data transfer, such as the FTP protocol.
- Fourthly, as mentioned above, the hardware platform must be able to flash the FPGA, and therefore have interfaces that allow you to connect special programmers. For example, USB-JTAG.

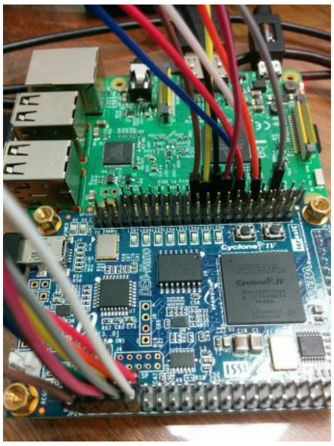

Fig. 1. Raspberry Pi connected to FGPA [3].

Thus, the use of a single-board Raspberry Pi computer as a hardware platform is suitable for this system, since it:

- allows you to write an algorithm in all popular programming languages, which, first of all, increases the speed of development, allows you to use ready-made libraries, and thus reduces the cost of development.
- runs on the Linux operating system, which makes it easy to update the code or transfer files (for example, log files) using special remote control protocols.
- has more computing power than microcontrollers.
- has USB ports that allows using a USB-JTAG FPGA programmers.

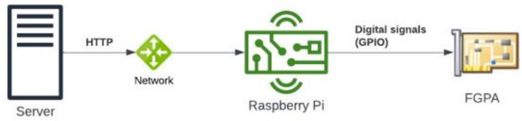

Fig. 2. Structural scheme.

IV. Structural Scheme

Fig. 2 shows the hardware components of the system, namely:

- The Server block is a computer that runs all the programs of the server part.
- The Network block, which is any variant of a computer network that implements the TCP/IP model.
- The Raspberry Pi unit, which, accordingly, is the hardware part of the system that is responsible for communication with the FPGA.
- FPGA unit, which is an FPGA.

Communication between the server and the Rasbperry Pi takes place via the Network computer network and uses the HTTP protocol.

The Rasbperry Pi communicates with the FPGA via GPIOs on both boards, creating a digital communication channel.

References

[1] S. Afanasiev, N. Poberezkyi, S. Ivanets, “Using remote hardware education kit to study electronics courses,” Theoretical and Applied Aspects of Device Development on Microcontrollers and FPGAs MC&FPGA- 2020, pp.34-35, 2020

[2] What is JTAG [Electronic resource]. - https://digilent.com/blog/what-is-jtag/

[3] Raspberry Pi to FPGA Communication [Electronic resource]. /Wikipedia. - https://catherineh.github.io/programming/2016/11/15/raspberry-pi-to-fpga-communication-example