TAF Journal of Advances in Technology and Engineering Research JATER
2018, **4**(3): 111-117

PRIMARY RESEARCH

Towards designing high-performance restful multimedia web services on FPGA

Al-Canaan Amer [1*], Khoumsi Ahmed [2]

[1] Department of Electrical Engineering, Islamic University of Al-Madinah, Al-Madinah, Saudi Arabia

[2] Department of Electrical and Computer Engineering, Sherbrooke University, Quebec, Canada

Keywords

Multimedia Web services
FPGA
Quality of service
VHDL

Received: 20 April 2018
Accepted: 25 May 2018
Published: 12 June 2018

Abstract

The pertinent characteristics of Field Programmable Arrays (FPGA), such as parallelism, low-power consumption, programmability, and low cost, make FPGA convenient for real-time applications, time-consuming algorithms as well, and services (MWS). Once built on FPGA, MWS becomes much less vulnerable to software attacks. Although FPGA technology implies lower performance and higher dimensions than Application Specific Integrated Circuits (ASIC), FPGA offers the advantages of programmability, reduced design costs and time-to-market. Indeed, multimedia Web services on FPGA are becoming common due to the availability different design tools as well as open-source modules. In this article, we are particularly interested in designing multimedia Web services with high Quality of Service (QoS) that employ substantial processing and memory resources.

I. INTRODUCTION

The widespread of design on FPGA is attributed to several characteristics including parallelism, low-power consumption, ability to be reprogrammed, affordable cost, availability of design tools and invulnerability to software attacks. Parallelism and low-power consumption is compelling for embedded systems and distributed applications such as MWS. FPGAs offer numerous advantages to engineering domains such as prototyping for ASIC, academic research [1, 2, 3, 4, 5] spacecraft and high-energy physics [6]. Compared to FPGAs, ASICs offer low area on Silicon, better performance and low-power consumption. On the other hand, design on FPGA offers parallelism, short time-to-market, affordable and flexible platform for prototyping and academic applications.

For designs on FPGA, the maximum clock frequency is limited by the placement of modules and on the area they occupy within the FPGA chip.

The parallelism aspect of FPGA is relevant in accelerating various intensive processing tasks, such as compression algorithms for multimedia manipulation, for regular expression matching [7] and real-time multi-face detection [8] for video surveillance applications. The flexible hardware aspect of FPGAs makes them appealing to the design of Web services for their inherent immunity against software viruses, for instance. These advantages have motivated our research work through the utilisation of FPGA in high-quality, secure and high-performance MWS.

Several realisations of MWS focus on lightweight services with modest sizes of data, while others are based upon intermediary microprocessor module to accomplish Web services on FPGA employing a specific flavour of Linux kernel. Although the exploitation of already available libraries for communication is between the FPGA and the Web through the Ethernet port may facilitate the development on FPGA, this reduces significantly the performance and efficiency. However, different Web services were successfully designed on FPGAs without necessarily focusing on the QoS. In contrast to that, in this paper we demonstrate our design and realisation of high QoS MWS on FPGA through incorporat-

[*] Corresponding author: Amer Al-Canaan
[†] email: amerC@iu.edu.sa

ing a Physical Layer module [9, 10, 11] and different Hardware Description Language (HDL) modules targeting different layers of the Open Systems Interconnection (OSI) model. The Representational State Transfer (REST) style of Web service design is proven to be more efficient and of lighter weight compared to the Simple Object Access Protocol (SOAP) MWS . Previous research articles show the relevance of using REST style rather than the SOAP protocol for the design of MWS [12, 13, 14, 15]. Therefore, we have designed an image-retrieval RESTful MWS, which utilises FPGA development boards from Digilent [16, 17], pure HDL code, open source modules and free design tools.

The rest of this paper is organised as follows: In section II, we present an overview of related works. In section III, we present a VHDL design example of a UDP and ARP packet sender on FPGA, which can be integrated into a MWS on FPGA. Following the design example, a design of an image-retrieval MWS was accomplished utilising Xilinx Spartan 6 FPGA [18] on the Nexys3 board, which is briefly presented in section IV for the sake of brevity and further details are left for an upcoming paper. Section V contains our conclusion and future work.

II. RELATED WORK

In this section, we give an overview of related works on Web services design on FPGA. The work in [19] aims at reducing implementation time and effort to create flexible and high performance Web services. The design is simplified through the utilisation of high level programming languages and on some intermediary modules on FPGA, namely the NIOS II micro-controller. This facilitates design by taking advantage of existing libraries.

The work in [20] presents a MWS for interfacing some specialised laboratory equipment on campus with remote end-users who are equipped with PDA's in order to get visual feedback. The realisation includes a Web server and FPGA Spartan-3E board and a Web-cam. The design takes into consideration the low processing resources in PDA's. Unlike our work, the work in [20] focuses the attention on the usefulness of the features rather than the QoS and performance. The work in [21] presents an Internet communications service between PC and FPGA through a Web page, based upon the SOAP protocol. The service enables data transfer through the Ethernet port by utilising a custom transport protocol that is based upon raw sockets. The custom protocol utilised provides a simple solution for communications by avoiding the complexity of implementing a TCP/IP stack on FPGA. Unlike the work in [21], our work is based upon a TCP/IP stack, which provides a RESTFul and compatible Web service. The works in [22, 23] describe the use of an FPGA-based system to attack a cryptoprocessor deployed in a banking infrastructure, based upon Data Encryption Standard (DES). The authors demonstrate the utilisation of pipelined solution and replication processing blocks on FPGA, as two valid and efficient schemes.

The work in [24] proposes a security solution based upon FPGA for multimedia applications on wireless LAN, which reduces the processing delay. By moving the processing load of security to the end systems' CPUs and by disabling the default security schemes, the performance could be augmented considerably, by 3.7 folds.

Our work can be essentially distinguished from other works by focusing on designing and implementing high QoS and high performance MWS, where the payload data are in binary format (images, voice and video) rather than text and eXtensible Markup Language (XML). We also cover up a wider range of data sizes, which are more realistic and common to MWS.

In order to efficiently profit from the hardware resources on FPGA, our proposed architecture is completely realised with HDL without the addition of CPU modules on FPGA.

III. DESIGN EXAMPLE OF A UDP/ARP PACKET SENDER ON FPGA

This section presents a design example of a User Datagram Protocol (UDP) and Address Resolution Protocol (ARP) packet sender module at a rate of 10 Mbps, which can be utilised for test purposes or in a simple video on-demand MWS. This section aims at presenting a design example that leads to seamless creation of MWS on FPGAs. The design in this section concentrates on the lower layers of the OSI [9] model, i.e., the data link layer, including its Media Access Control (MAC) sub-layer, and the physical layer (Phy).

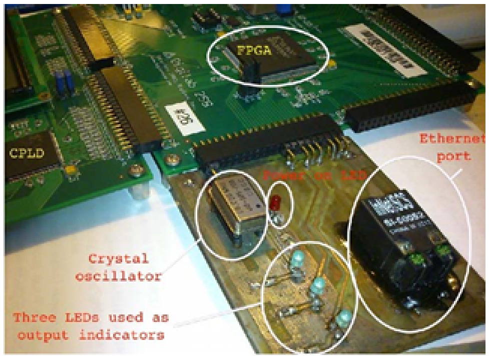

Fig. 1. Custom Ethernet board connected with the FPGA D2SB board

We have opted for the Spartan 2E FPGA on the D2SB [17] board that does not have an integrated Ethernet port.

ISSN: 2414-4592
DOI: 10.20474/jater-4.3.2

Therefore we have built a custom 10-Mbps Ethernet extension card in order to enable communications for MWS through the Ethernet port. The D2SB board and the designed Ethernet extension card are shown in Figure 1.

A complete MWS was accomplished on the Nexys3 board, which has an integrated Phy chip and one RJ-45 port that could not presented in this paper for the sake of brevity.

The features of the D2SB platform, which is con-nected to our module (as shown in Figure 1) can be summarised as follows:

- FPGA chip: Xilinx XC2S200E of the Spartan IIE family.
- Oscillator: a 50 MHz non-crystal oscillator, (an additional oscillator can be installed on the free DIP socket).
- More than a hundred I/O ports that are available at the six extension connectors, two of which can be connected with the data I/O (input/output) DIO5 board, providing 16 LEDs, a 4-digit 7 segment display, 16 push buttons, 8 switches and a two-line LCD.
- ROM used to store the generated programming files.

A. I/O Standards Phy Modes Supported by FPGA

1) I/O standards : Interfacing with FPGA development boards is enhanced through multiple I/O digital standards, which include Low Voltage Transistor-Transistor Logic (LVTTL), Low Voltage Complementary Metal Oxide Semiconductor (LVCMOS2), Low Voltage Differential Signal (LVDS) and Low Voltage Positive Emitter Coupled Logic (LVPECL). For low-speed applications with low interference and noise, LVTTL is adequate; otherwise the utilisation of a differential pair would be inevitable. Unlike the LVTTL and LVC MOS2 I/O standards, differential-signalling standards, on which depend many serial communications, such as Ethernet, necessitates one pair of wires per signal line. Ethernet transmit (Tx) and receive (Rx) differential-pair signals can be fed directly into the FPGA through selecting matched differential I/O pairs as described in the data sheet of FPGA (Spartan 2E FPGA in this context). This is particularly essential when using add-on cards that offer extended functionalities or when connecting the FPGA with other digital systems. In the context of Phy extension board, this insures simple and reliable interfacing with the FPGA.

Differential I/O pins offer high noise rejection and are carefully routed to obtain identical electrical characteristics on copper tracks, which means that the line widths, lengths and spacing should be identical. According to our experience regarding all of the differential I/O pairs of the D2SB FPGA board, almost all differential I/O pairs were properly matched except for one or two differential pairs.

2) Phy modes supported by FPGA: Independent Interface (MII) [25] connects the MAC sub-layer with the Phy layer (Phy chip) at bit rates up to 100 Mbps. The MII mode is the only supported mode on the Nexys3 board, though its Phy chip supports other modes. Although this reduces the number of components (like resistors and oscillators, for instance) and reduces fabrication cost, it restricts the choice of design of HDL modules. For Gigabit MII (GMII), the data are exchanged one byte at a time at a reference frequency of 125 MHz. For 10-Gbit MII (XGMII), 32 bits of data are exchanged at a time.

The detection of the Start of Ethernet Frame Delimiter (SFD) and the end of transmission of a frame is done through a Finite State Machine (FSM) implemented in VHDL, which is inspired by the open-source resources in [26]. Indeed, the availability of open-source HDL modules enhanced production for research and industry.

Since FPGA is programmable hardware, it is essential to have some relevant knowledge about digital hardware and logic design. Though the syntax of HDL has much in common with software and computer languages, HDL has its own particular aspects and purpose [1].

B. Designing and Building an Ethernet Extension Board

We have built several prototypes of an Ethernet extension board that enabled us to investigate the use of single-ended and differential signals into the FPGA as well as to evaluate several schematics and layouts using Computer Aided Design (CAD) software, such as geda schem, geda pcb, kicad, DipTrace and Eagle. The schematic diagram of the final prototype has been achieved with geda schem as depicted on Figure 2. The following components are shown on both Figures 1, 2:

- Two differential I/O pairs for the Tx and Rx signals.
- Single-ended external 60-MHz crystal oscillator. It provides an accurate frequency and precise timing through the Delay Locked Loop (DLL) module of the FPGA in order to obtain the 20-MHz clock for the 10-Mbps Ethernet.
- RJ-45 connector with an isolation transformer and integrated magnetics.
- Power-on LED indicator.
- Three green LEDs connected to three FPGA I/O pins, which may be used as additional signal indicators.

The fabrication of the Printed Wiring Boards (PWB) prototypes of the Ethernet extension interface (as shown on Figure 2) on double-side boards involved household resources, such as laser-printer toner transfer, non toxic copper etching and tinning using electroplating for extended protection of the copper traces.

C. Overview of Software Tools for FPGA

The design on FPGA necessitates specific software tools in order to write HDL modules, simulate HDL, synthesize code, and program the FPGA. Software tools for FPGA are hardware-dependent, such as Xilinx Webpack with isim simulator. We limit our discussion to Xilinx-related tools since we utilised only FPGA chips from Xilinx, which is supported for Windows and Linux. We therefore opted for Xilinx on Debian Linux as our development platform, which proved to be user-friendlier and very productive.

We utilised several testing tools including packet-sniffing and packet-generation tools, such as Wireshark and packeth. In addition, tools like wget, curl and SocketTest were used for checking TCP and HTTP packets.

Wireshark software is used to capture different types of packets such as UDP and ARP. In addition, some packets are selected and viewed on the LCD (on a separate card attached to the D2SB card), for which we have utilised a FIFO memory and a dedicated state machine

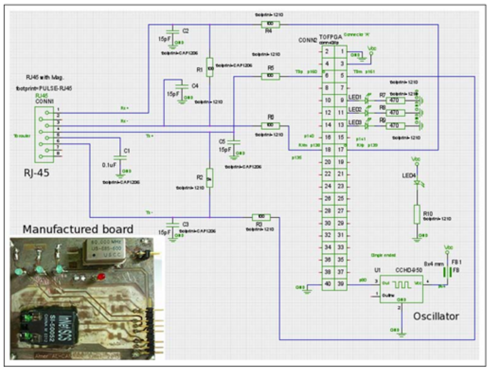

Fig. 2. Schematic diagram for the Ethernet extension module and the manufactured board on the lower left corner

IV. REALISATION OF A HIGH QOS IMAGE-RETRIEVAL MWS ON FPGA

In this section, we provide details about the design and the implementation of a high QoS image-retrieval MWS on the Nexys3 FPGA board, as depicted on Figure 3. We provide a complete design and realisation of a MWS on FPGA that concentrates mainly on the higher layers of the OSI model, whereas the previous design in section IV concentrates on the lower layers of the OSI. This section also concentrates on the service logic of the MWS as well as on several QoS parameters. The realisation of MWS on FPGA in this section is inspired by the design example presented in section IV. The Nexys3 board is equipped with Spartan 6 Xilinx XC6SLX16 FPGA and offers several peripherals, such as an RJ-45 port and Phy chip. Compared to the previous FPGA board used in section IV, the Nexys3 board is equipped with all necessary peripherals for the design of sophisticated Web services without the need for any additional hardware. We have investigated several open-source projects (cores), of which the Ethernet MAC 10/100 Mbps project has been adopted and tailored in order to suite our MWS on the Nexys3 board. This module was selected for two reasons:

1) It offers compatible MII standard with the Phy available on the Nexys3 board

2) It has multiple features such as:

• Preamble generation and removal.

• Automatic padding for short frames before transmission.

• Detection of too long or too short packets.

• Ability to transmit large packets.

• Full duplex support. • Supported rates include 10 and 100 Mbps.

The first step of adapting the Ethernet module was to connect the Phy (see section IV-A) input and output (I/O) signals to the FPGA through the user constraints file of the Nexys3 board. Setting up the operation mode of the Phy (i.e., 100 Mbps in full-duplex mode) on the Nexys3 board has been done through the addition of several I/O buffers.

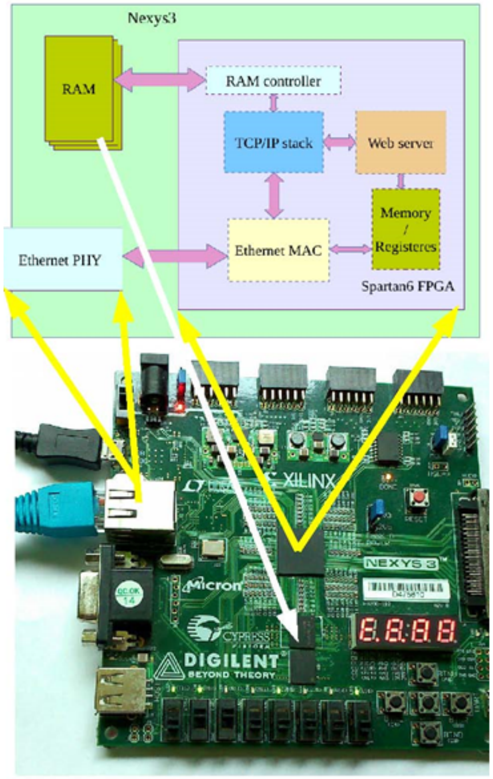

Fig. 3. Architecture of the MWS on FPGA using the Nexys3 board

This has been achieved through a careful study of the data-sheet of the Phy chip on the Nexys3 board. We have written a FSM module in VHDL to initialise the different configuration registers and provided input data for the Wishbone in order to activate the transmission module.

Besides, we have added two RAM modules to save the received and transmitted packets. The Wishbone is a data communications standard for exchanging the frames to be sent and received between the memory and the Ethernet core. For test purposes, we have added new modules in order to generate fixed-width UDP and ARP packets that are transmitted periodically every one second. This phase has been successfully achieved throughout the FSM for the Wishbone interface module that we have developed using tools like StateCad [27]. Although the Ethernet MAC 10/100 Mbps project offers a rich set of options and features, its documentation lacks some critical information about the operation of the Wishbone interface, which has caused extra delays for the design of the receiver module. In order to avoid design-related delays, we opted for the adaptation of a second HDL project core, namely, the TCP/IP socket, of which the characteristics are compared to other technologies. The TCP/IP socket project is compelling since it offers a TCP/IP stack and a working example of a simple HTTP server, which offers an interface page in HTML. Several improvements have been done to meet our requirements. For example, we worked closely with the original author of the module in order to provide support for the MII Phy standard of the Nexys3 board. We have added more features to the HTTP stack including binary file transfer and carried out some optimisations to reduce FPGA resource consumption.

A simple HTML interface was designed of the MWS, not shown here for the sake of brevity, displays an image, allows control of 4 LEDS, displays the state of the 8 switches and provides an estimation of the response time on the Nexys3 board. Though the comparison between FPGA and computer implementations indicates that for small sized images both technologies show similar performances, they differ enormously with respect to power consumption and clock speed.

The power consumed by our image-retrieval MWS realised on FPGA (using the Nexys3 board with USB2 port) is well below 10 Watts, whereas the power consumed by a typical 13-inch MacBook Pro laptop computer (running Mac OS X) is around 64 Watts. The clock frequencies of the CPUs in our tests are 2400 MHz and 2612 MHz compared to 100 MHz for the MWS on FPGA, which shows an obvious advantage of designing on FPGAs.

V. CONCLUSION

Parallelism, programmable in-memory design, af-fordable cost, availability of design tools and low-power consumption make FPGA a highly compelling alternative for computers in several domains, including MWS. FPGAs provide excellent platforms for ASIC prototyping and for secure multimedia Web services. The widespread of parallel processing and pipelined solutions on FPGAs have opened the way for time-consuming and processing-intensive applications.

Our contributions in this article can be summarised as follows:

1) We present a design example of a UDP/ARP packet sender on FPGA, which is intended to inspire new designs of MWS on Spartan 2E FPGA and is focused mainly on the lower layers of the OSI model.

2) We elaborate on a more sophisticated design of high QoS MWS on Spartan 6 FPGA based upon pure HDL and open-source VHDL modules.

3) We elaborate on our design, implementation tools and environments. The second design represents a thorough approach to the realisation of a low-power, and secure RESTful image-retrieval MWS on FPGA, which concentrates on the upper layers of the OSI model.

For the sake of brevity, some details about test cases and results could not be included in this article. For larger payloads, the MWS on FPGA is to out-perform their computer counterparts by several folds, since the lager image sizes mean longer processing times, i.e., the MWS threads would be served more frequently by the CPU.

A. *Future Work*

We plan to extend our approach to include more QoS parameters such as response time, download time, processing time, rapid compression, availability, throughput and security. Our future work would concentrate on rapid compression, availability, throughput and security. After discussing the use of text and image payloads, we would include more multimedia types, such as video and voice.

The work in this paper can be extended to include specialised QoS management modules for Web service workflow interactions with minimum overhead cost in order to enhance the design and realisation of high performance and high fidelity MWS. The role of the QoS management module is to perform negotiation, agree-ment and to monitor different QoS parameters and service interactions. The Web service interactions workflow module is to prevent, detect and resolve Web service interactions that may occur during operation. We plan to add more features to our MWS, such as enabling the transfer of large files, by integrating new HDL modules to regulate data transfer with recovery and resume options utilising the chunked-transfer encoding of the HTTP/1.1 protocol.

VI. ACKNOWLEDGMENT

We thank Mr. Jonathan P. Dawson for his valuable support and collaboration regarding the adaptation of the MII interface of the TCP/IP socket to the Nexys3 board.

REFERENCES

[1] D. Koch, F. Hannig, and D. Ziener, *FPGAs for Software Programmers*. New York, NY: Springer, 2016.

[2] B. S. C. Varma, K. Paul, and M. Balakrishnan, *Architecture Exploration of FPGA Based Accelerators for BioInformatics Applications*. New York, NY: Springer, 2016.

[3] R. Razzaghi, M. Mitjans, F. Rachidi, and M. Paolone, ``An automated FPGA real-time simulator for power electronics and power systems electromagnetic transient applications,'' *Electric Power Systems Research*, vol. 141, pp. 147-156, 2016. doi: https://doi.org/10.1016/j.epsr.2016.07.022

[4] N. Lefebvre, M. Khelifi, and Y. De Ladurantaye, ``Design of a Prototype Presss for in-field pressing of sweet sorghum and sweet pearl millet biomass,'' *Journal of Applied and Physical Sciences*, vol. 1, no. 1, pp. 29-32, 2015. doi: https://doi.org/10.20474/-japs1.1.5

[5] T. H. Cho and G. M. Jeon, ``A method for detecting man-in-the-middle attacks using time synchronization one time password in interlock protocol based internet of things,'' *Journal of Applied and Physical Sciences*, vol. 2, no. 2, pp. 37-41, 2016. doi: https://doi.org/10.20474/japs-2.2.2

[6] M. Wirthlin, ``High-reliability fpga-based systems: Space, high-energy physics, and beyond,'' *Proceedings of the IEEE*, vol. 103, no. 3, pp. 379-389, 2015. doi: https://doi.org/10.1109/jproc.2015.2404212

[7] P. Russek and K. Wiatr, ``FPGA-accelerated algorithm for the regular expression matching system,'' *International Journal of Electronics*, vol. 102, no. 1, pp. 71-88, 2015. doi: http://dx.doi.org/10.1080/00207217.2014.938313

[8] N.-J. Wang, S.-C. Chang, and P.-J. Chou, ``Real-time multi-face detection on FPGA for video surveillance applications,'' *Journal of the Chinese Institute of Engineers*, vol. 38, no. 1, pp. 120-127, 2015. doi: https://doi.org/10.1080/02533839.2014.929771

[9] W. A. Shay, *Understanding Data Communications and Networks*. Boston, MA: International Thomson Publishing, 1998.

[10] A. Dutta, J. Fifield, G. Schelle, D. Grunwald, and D. Sicker, ``An intelligent physical layer for cognitive radio networks,'' in *Proceedings of the 4th Annual International Conference on Wireless Internet,* Maui, HI, 2008.

[11] M. M. Yaacob, A. R. Hussein, and M. F. Othman, ``A computer program for fault diagnosis and assessment of quality insulation oil in power transformer based dissolved gas analysis,'' *International Journal of Technology and Engineering Studies*, vol. 1, no. 3, pp. 98-106, 2015. doi: https://doi.org/10.20469/ijtes.40005-3

[12] G. Goth, ``Critics say web services need a rest,'' *IEEE Distributed Systems Online*, vol. 5, no. 12, pp. 1--1, 2004. doi: https://doi.org/10.1109/mdso.2004.35

[13] F. Belqasmi, J. Singh, S. Y. B. Melhem, and R. H. Glitho, ``Soap-based vs restful web services: A case study for multimedia conferencing,'' *IEEE Internet Computing*, vol. 16, no. 4, pp. 54-63, 2012. doi: https://doi.org/10.1109/mic.2012.62

[14] A. Al-Canaan and A. Khoumsi, ``Multimedia web services performance: Analysis and quantification of binary data compression,'' *Journal of Multimedia*, vol. 6, no. 5, pp. 45-50, 2011. doi: https://doi.org/10.4304/jmm.6.5.447-457

[15] N. Onose, R. Khalaf, K. Rose, and J. Siméon, ``A restful workflow implementation on top of distributed xquery,'' in *5th International Workshop on XQuery Implementation,* Scottsdale, AZ, 2008.

[16] Digilent, ``Electrical engineering store,'' 2014. [Online]. Available: https://bit.ly/2u9dRJR

[17] D. Product, ``Product details,'' 2015. [Online]. Available: https://bit.ly/2u9dRJR

[18] Spartan, ``Spartan-6 product advantage,'' 2014. [Online]. Available: https://bit.ly/2UCIuTG

[19] A. Ruta, R. Brzoza-Woch, and K. Zielinski, ``On fast development of FPGA-based SOA services—machine vision case study,'' *Design Automation for Embedded Systems*, vol. 16, no. 1, pp. 45-69, 2012. doi: https://doi.org/10.1007/s10617-012-9084-z

[20] R. J. Costa, G. R. Alves, M. Zenha-Rela, M. Zenha-Rela, R. Poley, and C. Wishart, ``FPGA-based weblab infrastructures guidelines and a prototype implementation example,'' in *2009 3rd IEEE International Conference on E-Learning in Industrial Electronics,* New York, NY, 2009.

[21] O. M. Machidon and F. Sandu, ``Enhancing fpga i/o communication using web services,'' *Bulletin of the Transilvania University of Brasov*, vol. 7, no. 2, pp. 89-94, 2014.

[22] R. Clayton and M. Bond, ``Experience using a low-cost FPGA design to crack DES keys,'' in *International Workshop on Cryptographic Hardware and Embedded Systems,* Berlin, Heidelberg, 2002.

[23] M. Bond, ``Attacks on cryptoprocessor transaction sets,'' in *International Workshop on Cryptographic Hardware and Embedded Systems,* Berlin, Heidelberg, 2001.

[24] T. Hayajneh, S. Ullah, B. J. Mohd, and K. S. Balagani, ``An enhanced WLAN security system with FPGA implementation for multimedia applications,'' *IEEE Systems Journal*, vol. 11, no. 4, pp. 2536-2545, 2017. doi: https://doi.org/10.1109/jsyst.2015.2424702

[25] Microchip, ``Microchip technology,'' 2016. [Online]. Available: https://bit.ly/2xr3rcV

[26] FPGA, ``FPGA projects: 26 projects to build using an FPGA board,'' 2015. [Online]. Available: https://bit.ly/2JdubUe

[27] Xilinx, ``Xilinx and samsung jointly enable the world's first 5g nr commercial deployment,'' 2015. [Online]. Available: https://bit.ly/2nca7Fr