View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Repositório Científico do Instituto Politécnico do Porto

Using FPGAs to create a reconfigurable IEEE1451.0-compliant weblab infrastructure

Ricardo Costa[12], Gustavo Alves[1] and Mário Zenha-Rela[2]
ISEP/CIETI/LABORIS[1], FCTUC/CISUC[2]
rjc@isep.ipp.pt, gca@isep.ipp.pt, mzrela@dei.uc.pt

Abstract

The reconfiguration capability provided by Field Programmable Gate Arrays (FPGA) and the current limitations of weblab infrastructures, opened a new research window. This paper focus on describing the way weblabs can be reconfigured with different Instruments & Modules (I&M) required to conduct remote experiments, without changing the entire infrastructure. For this purpose, the paper emphasizes the advantage of using FPGAs to create reconfigurable weblab infrastructures using the IEEE1451.0 Std. as a basis to develop, access and bind embedded I&Ms to an IEEE1451.0-Module.

1. Introduction

In the electronic domain reconfiguration is becoming a familiar word since the appearance of FPGAs. These provide the ability of redefining an architecture based on a set of internal modules that can be interconnected according to a set of rules described by standard Hardware Description Languages (HDL). This means reconfiguring the device, and therefore the way it runs, without replacing its main hardware. This flexibility provided by FPGAs can be viewed not only as a thematic of study in engineering courses, but also as devices able to create the so-called weblab infrastructures, by the implementation of sensors/actuators that can be the I&Ms required to use in a remote experiment [1].

Weblabs allow the remote conduction of laboratorial experiments, providing a way for teachers and students to access real equipment, provided by an infrastructure, using a simple device connected to the Internet. Since the 90's that weblabs are proliferating in education, especially in engineering and science disciplines [2][3][4][5] where laboratorial work is fundamental [6][7]. This is justified essentially by the flexibility they provide on accessing, without time and place constrains, the equipment commonly available in a laboratory, which comprehends a set of I&Ms connected to an Experiment Under Test (EUT). Noticeably, the implementation of weblabs in different institutions can be increased if improving their infrastructures, namely by: i) enabling their reconfiguration (only setting up connections of predefined I&Ms is currently allowed [8]) and, ii) adopting a standard solution for their implementation and access. Despite these two problems are being debated in the GOLC technical committee [9], currently there is not yet a solution to solve them. While the standard access to a weblab infrastructure can be overcome by the use of a common API, infrastructural and reconfiguration aspects are still unsolved. It is precisely in this scenario that the reconfigurable nature of FPGAs and the use of a standard approach, can contribute to overcome the two referred problems.

Adopting FPGAs as the main device of a weblab infrastructure allow reconfiguring, in its core, a set of embedded I&Ms that, if described through standard HDL files, can be shared by the entire educational community. At the same time, if these same I&Ms follow a specific standard, they will be easily shared, integrated and accessed, promoting more collaboration among institutions in the development and dissemination of weblabs.

Therefore, for promoting a high widespread of weblabs in education, this paper proposes joining the capabilities provided by the reconfigurable nature of FPGAs, to the large focus provided by the IEEE1451.0 Std., that allows defining and network-interfacing transducers, which can be the referred I&Ms. The paper describes the implementation of a generic and synthesizable IEEE1451-Module for FPGA devices, and a methodology to develop, access and bind I&Ms compatible with this module.

Section 2 provides an overview about the IEEE1451.0 Std., and presents the weblab infrastructure implemented at our laboratory. Section 3, presents the IEEE1451.0 Std. and the IEEE1451.0-Module, this entirely described in the standard Verilog HDL. Section 4, describes the process of creating and binding I&Ms to that IEEE1451-Module, so they can be used by the weblab infrastructure to conduct experiments. Section 5 explains the reconfiguration process of the weblab infrastructure, and section 6 concludes this paper and presents ongoing work.

2. Weblab infrastructure overview

The IEEE1451.0 Std. [10] aims to network-interface transducers through an architecture based on two modules: the Transducer Interface Module (TIM), that controls Transducer Channels (TC), and the Network Capable Application Processor (NCAP), that provides network access to the TIM and to those TCs. The behaviour and features of a TIM and TCs are described by Transducer Electronic Data Sheets (TEDS) monitored by a status register and controlled by a set of commands that may be accessed by an IEEE1451.0 HTTP API.

As illustrate in figure 1, the implemented infrastructure uses the NCAP implemented in a micro webserver, connected by a serial RS232 interface to the TIM. This is implemented in a FPGA-based board that provides a set of interfaces (digital I/O, DAQs, etc.) to access a specific Experiment Under Test (EUT). Internally, the adopted FPGA is reconfigured by a generic IEEE1451.0-Module that, by decoding a set of commands received from the NCAP, controls TCs and therefore, the embedded I&Ms bound to it. To run remote experiments, users remotely access these I&Ms, that are connected to the EUT, using the IEEE1451.0 HTTP API implemented in the NCAP.

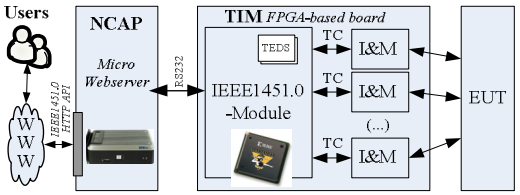

Fig. 1. Implemented weblab infrastructure.

While standardization is guaranteed by the adoption of the IEEE1451.0 Std., the use of an FPGA for implementing the TIM is fundamental, since it can be reconfigured with I&Ms required for a specific experiment, and these can run independently and in parallel, like in a traditional laboratory.

Therefore, seeking for a flexible and reconfigurable solution, the TIM was entirely described in Verilog HDL, which guarantees its portability towards any kind of FPGA. Internally the IEEE1451.0-Module implements all features described by the standard, controlling the TCs used to access the embedded I&Ms. The adoption of this architecture required the TIM description supported in two fundamental aspects: i) the IEEE1451.0-Module is able to be redefined according to the adopted I&Ms and, ii) each I&M is described through a set of files following a specific methodology.

3. IEEE1451.0-Module

Entirely described in Verilog HDL, as illustrated in figure 2, the IEEE1451.0-Module internally comprehends 4 other modules:

1- Decoder/Controller Module (DCM) - is the Central Processing Unit (CPU) that controls all the other modules, by decoding commands received from an Universal Asynchronous Receiver / Transmitter Module (UART-M) or by the reception of event signals generated by I&Ms.

2- TEDS Module (TEDS-M) - comprehends an internal controller able to access TEDSs.

3- Status/State Module (SSM) - manages the operating states and the status registers of each TC and TIM.

4- UART Module (UART-M) - interfaces the NCAP and the TIM through a RS232 interface using receiver/transmitter modules (Rx/Tx).

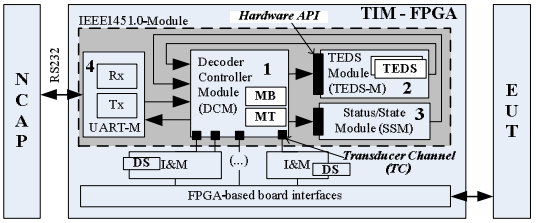

Fig. 2. Internal modules of the IEEE1451.0-Module.

The DCM controls the entire IEEE1451.0-Module by implementing the following features: i) provides IEEE1451.0 commands defined through a set of command-tasks, ii) implements error detection mechanisms, iii) controls both the SSM and the TEDS-M by reading, writing or updating their internal memories using a set of commands provided by dedicated hardware APIs, iv) controls the UART-M used to establish the NCAP-TIM interface, and iv) controls a set of embedded TC-tasks that manage the TCs, running as actuators, sensors or event sensors. The DCM provides a set buses that interfaces the TEDS-M, SSM and I&Ms, the UART-M to receive/transmit commands from/to the NCAP, and two external memories that support the operations of the DCM, named Memory Buffer (MB) and Map Table (MT). The MB gathers temporary TEDS' fields before they can be written into a TEDS's memory provided within the TEDS-M. It also acts as a data-bridge to Data Sets (DS), which are available in each I&M to hold internal data sent or received by IEEE1451.0 commands. The MT implements a table to associate each TEDS, defined in the TEDS-M, to a particular TC or TIM, according to a specific Identification Field (ID). Defined during a reconfiguration process described in section 5, it is based on this association that the DCM may understand which TEDS should be accessed after a reception of command.

The TEDS-M integrates all TEDSs adopted by the infrastructure, including those associated to a particular I&M, to the TIM and/or to TCs. This module comprehends an internal controller that provides particular commands to write, read or update each TEDS. To facilitate the access to those commands, the TEDS-M provides a hardware API, that can be used by the DCM, namely by command-tasks that implement IEEE1451.0 commands, and by TC-tasks that manage the interface between the I&Ms and the DCM.

The SSM provides access to two independent memories whose contents specify the operation states and the status of the TC/TIM. During the DCM operation, those memories will be accessed by command/TC-tasks to update the state and the status of each TC/TIM. The access to those memories is made using a set of commands provided by an internal controller, whose access can also be made by a hardware API.

The UART-M is controlled by the DCM using a handshake protocol that manage a set of signals to access two internal buffers and to control all data flow during transmissions. Structured in internal modules, the UART-M also implements a mechanism for validating and creating data structures according to the IEEE1451.0 Std..

In order to fulfill the reconfigurable requirements of the weblab infrastructure, besides using FPGA technology, the IEEE1451.0-Module was described through a set of Verilog HDL files some of them redefined according to the I&M adopted for a particular experiment. Moreover, its automatic redefinition, that is a part of the reconfiguration process, required the use of a specific architecture for developing and binding I&Ms, so they can be compatible with the IEEE1451.0-Module and therefore, able to be accessed according to the IEEE1451.0 Std..

4. Compatible Instruments & Modules

To bind I&Ms to the IEEE1451.0-Module, these should be designed in different parts. These parts include one or more modules bound through TC lines to a set of TC-tasks, which are described in Verilog HDL and embedded in the DCM. As illustrated in figure 3, these tasks allow the access to the other modules and the interface between the IEEE1451.0-Module and each I&M, enabling their control according to TEDSs' contents that should also be defined by the developer. The number of TCs depends on the I&M's architecture and the parameters to be controlled.

Therefore, the design of an I&M compatible with the IEEE1451.0-Module comprehends an architecture divided in 3 distinct parts: i) HDL modules describing the I&M itself, ii) TC-tasks to control and interface those same modules with the DCM; and iii) TEDSs to define the behaviour of the entire IEEE1451.0-Module and of each TC. An I&M is accessed by one or more TCs controlled by TC-tasks managed according to the data available within TEDSs and status/state memories. Since I&Ms' developers need to define both the TC-tasks and the HDL modules, they can adopt any type of handshake protocol to exchange data between the DCM and the I&Ms. Some TC-tasks are optional others mandatory, and they are responsible to automatically access the TEDS-M, the SSM, in some situations the UART, and the MB, when the IEEE1451.0-Module receives event signals or IEEE1451.0 commands.

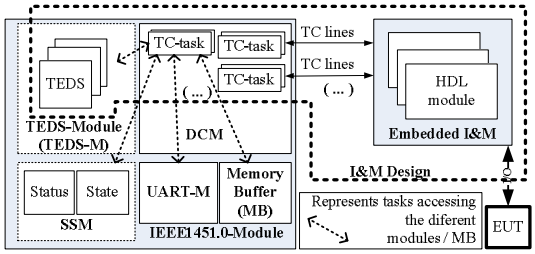

Fig. 3. Parts required for defining I&Ms compatible with the IEEE1451.0-Module.

To simplify the design of an I&M, each TC-task accesses those modules using the hardware APIs, facilitating this way their description and independence toward the specificities of the DCM implementation. They should be defined according to the adopted TC, so the DCM may automatically use them to handle received commands or events generated by I&Ms. The number of adopted TCs depends on developers' options that should take into consideration the parameters to control in an I&M, the TEDS's definitions, and the resources available in the FPGA. Therefore, the development of an I&M compatible with the IEEE1451.0-Module should follow the sequence presented in figure 4.

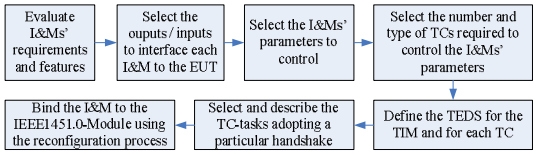

Fig. 4. Sequence for implementing an I&M compatible with the IEEE1451.0-Module.

Developers should start by evaluating the requirements and features of the I&M they want to develop, estimating its complexity to understand what modules should be described. For that purpose, the outputs and inputs connected to the EUT should be selected, namely the associated signals, which are managed by I&Ms' parameters controlled by TCs. After selecting the inputs/outputs and the parameters to be controlled, developers should define the number of TCs. This definition should be made according to the type of parameters to control and

the requirements posed to the FPGA device, since the use of several TCs may require many FPGA resources. Once selected the TCs used to access the I&M, developers should define the TEDSs to describe the TIM architecture and the TCs' behaviour that, among other definitions, specifies if a TC acts as an actuator, a sensor or as an event sensor. Current solution suggests that at least the TC-TEDS should be defined for each TC, but developers may define others TEDSs, as described by the IEEE1451.0 Std.. The way those TCs are controlled is made by a set of predefined TC-tasks described by the developer, so they can provide the interface to the other modules within the IEEE1451.0-Module. To simplify developments, the hardware APIs provided by the TEDS-M and the SSM should be used with the protocol adopted to control the data transmission/reception of the UART-M. After all these definitions, a specific I&M is available to bind to the IEEE1451.0-Module using a reconfiguration process.

5. Reconfiguration

After describing the I&Ms, these can be bound to the IEEE1451.0-Module so they can be used in a specific experiment. For this purpose, the infrastructure, namely the TIM, should be reconfigured, which means changing the internal connections of the FPGA. This reconfiguration process involves a set of steps described in figure 5, currently supported by a specific web reconfiguration tool already detailed in [11].

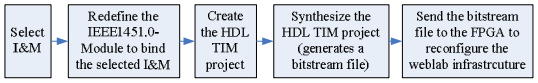

Fig. 5. Weblab infrastructure reconfiguration sequence.

This tool is available in a remote machine named Labserver that runs the entire reconfiguration process. Internally this machine integrates a set of software modules and, in particular, the IEEE1451.0-Module that will be redefined according to the selected I&Ms and to some configuration rules. For this purpose, users should start selecting two groups of files. The first group describing each I&M, and the second group describing all changes to be made in the TIM and in the IEEE1451.0-Module, so it may bind the selected I&Ms. The TIM, and in particular the IEEE1451.0-Module, is then redefined according to the rules defined in a configuration file, and a new HDL project will be created and synthesized to the selected FPGA using the tool associated to its manufacturer. A *bitstream* file is then created and sent to the FPGA, reconfiguring the weblab infrastructure to run the specified experiment.

6. Conclusions and ongoing work

The use of FPGAs is a promising solution for developing reconfigurable weblab infrastructures. This document emphasized this aspect, presenting current weblabs problems, and the way these can be solved by joining the IEEE1451.0 Std. basis with FPGA technology. The development of a reconfigurable, flexible and universal solution at low prices, is the main objective of the described work. In the next months a prototype experiment based on step-motors will be validated by some specialist in the area. The goal is to get feedback about the implemented infrastructure and the methodology for reconfiguring the weblab infrastructure. In the future, the intention is to enlarge the offer of compatible I&Ms, so other experiments can be designed. For further details, readers are invited to visit the webpage: www.dee.isep.ipp.pt/~rjc/phd.

References

[1] Antonio de la Piedra, An Braeken and Abdellah Touhafi, 'Sensor Systems Based on FPGAs and Their Applications A Survey', *Sensors*, vol. 12, no. 9, pp. 12235–12264, Sep. 2012.

[2] Javier García Zubía and Gustavo R. Alves, *Using Remote Labs in Education - Two Little Ducks in Remote Experimentation* -. Deusto Digital, 2011.

[3] Abul K.M. Azad, Michael E. Auer and V. Judson Harward, Ed., *Internet Accessible Remote Laboratories - Scalable Elearning Tools for Engineering and Science Disciplines*. Engineering Science Reference - IGI Global, 2011.

[4] Luís Gomes, 'Current Trends in Remote Laboratories', *IEEE Transactions on industrial electronics*, vol. 56, no. 12, p. 4744, Dec. 2009.

[5] G.R. Alves et al., 'Using VISIR in a large undergraduate course: Preliminary assessment results', *2011 IEEE Global Engineering Education Conference (EDUCON)*, pp. 1125 –1132, Apr. 2011.

[6] Lyle D. Feisel and George D. Peterson, 'A Colloquy on Learning Objectives For Engineering Education Laboratories', *Proceedings of the American Society for Engineering Education*, p. 12, 2002.

[7] Doru Popescu and Barry Odbert, 'The Advantages Of Remote Labs In Eng. Education', *Educator's Corner - Agilent Tech. - application note*, p. 11, Apr. 2011.

[8] Mohamed Tawfik et al., 'Virtual Instrument Systems in Reality (VISIR) for Remote Wiring and Measurement of Electronic Circuits on Breadboard', *IEEE Transactions on Learning Technologies*, vol. PP, no. 99, p. 1, 2012.

[9] 'GOLC - The Global Online Laboratory Consortium', 2012. [Online]. Available: http://www.online-lab.org/. [Accessed: 12-Nov-2012].

[10] IEEE Std. 1451.0™, 'IEEE Standard for a Smart Transducer Interface for Sensors and Actuators', *The Institute of Electrical and Electronics Engineers, Inc.*, p. 335, Sep. 2007.

[11] Ricardo Costa, Gustavo Alves and Mário Zenha-Rela, 'Reconfigurable IEEE1451-FPGA based weblab infrastructure', *9th Int. Conf. on Remote Eng. and Virtual Instrumentation (REV)*, pp. 1 –9, Jul. 2012.