A demo prototype of a reconfigurable IEEE1451.0-compliant and FPGA-based weblab

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

Abstract - **A reconfigurable weblab prototype was designed according to the IEEE1451.0 Std. based on FPGAs. A brief introduction about its architecture and underlying infrastructure is presented. After an overview about the main features of the weblab, namely the standard access and the reconfiguration capability, the proposed demonstration for the exhibition session is described.**

Index Terms - **Remote experimentation, Weblabs, Remote labs, IEEE1451.0 Std., reconfiguration.**

I. Introduction

Weblabs are a complementary resource to support the experimental work required in engineering courses. They provide to students and teachers the ability of remotely conducting real experiments through the Internet, enabling the interaction with real equipment available in a laboratory. Technically, the development of weblabs follows traditional client-server architectures, but without any standardize approach, both at design and access levels. Additionally, current solutions do not allow selecting and/or replacing the instrumentation required to interface the target experiments, in the same way as in traditional laboratories. Some solutions enable setting up connections between preselected instruments and the target experiments using switching matrixes [1][2], but they do not enable selecting new ones. Therefore, to overcome these two limitations, namely: i) lack of standard access and design of weblabs and; ii) the impossibility of changing/replicating the instruments, led to analyze the possibility of using the IEEE1451.0 Std. and the FPGA technology for developing reconfigurable weblabs.

Section II provides a generic overview about the IEEE1451.0 Std. and its appliance to weblabs. Section III presents the implemented architecture. Before concluding, section IV describes the proposed demonstration of the implemented weblab for the exhibition session.

II. IEEE1451.0 Std. applied to weblabs

The IEEE1451.0 Std. defines an architecture to network-interface transducers (sensor and actuators) controlled and accessed based on specifications defined in data structures named Transducer Electronic Data Sheets (TEDSs) [3][4]. The architecture is supported by two modules: TIM (Transducer Interface Module) and NCAP (Network Capable Application Processor), interfaced according to other IEEE1451.x Stds. using physical protocols, such as the RS-232. Besides defining particular functionalities for the transducers, which make them *smart*, the standard specifies commands and APIs for their local and/or remote access, namely an IEEE1451.0-HTTP API that facilitates their remote control through the Web.

Supported on the *smart* operations of the transducers and on their standard access, the IEEE1451.0 Std. was seen as a promising solution for designing weblabs. The *smart* transducers defined by the standard can be the instruments or modules, named weblab modules, required to interface and to control the target experiments. These weblab modules can be designed and accessed according to the IEEE1451.0 Std., and embedded in a weblab that is implemented according to a particular architecture.

III. Implemented weblab architecture

The weblab enables the standard access and the reconfiguration of an infrastructure with different weblab modules described through standard Verilog HDL files. These files are easily shared and able to be embedded into different types of FPGAs, making then versatile and platform independent. For this purpose, a weblab architecture was implemented, comprising a weblab server and an underlying infrastructure, as illustrated in figure 1.

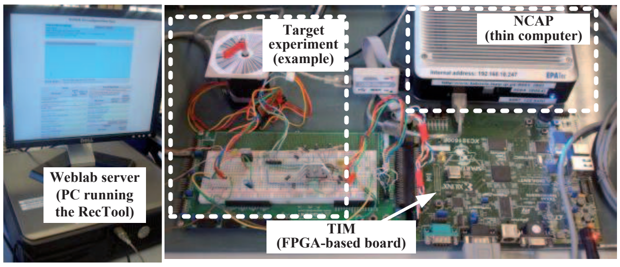
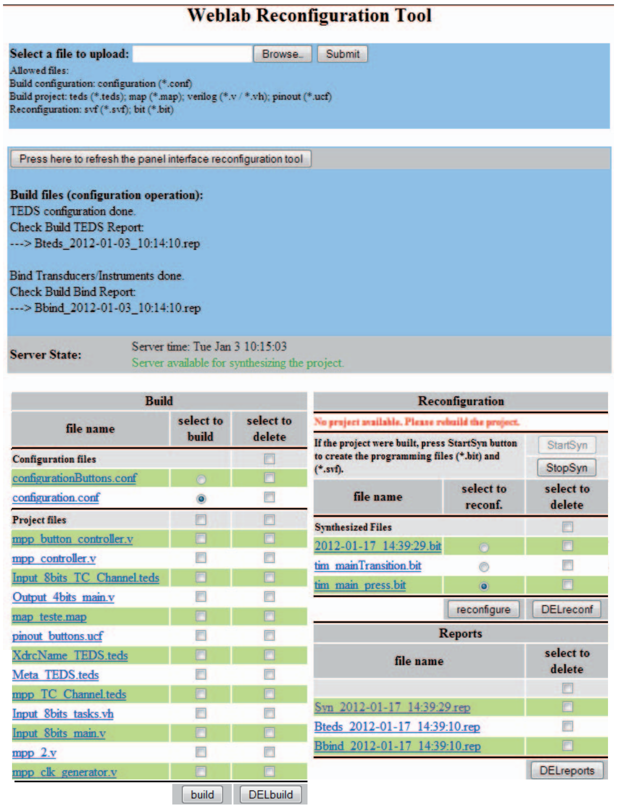

Target experiment (example)
NCAP (thin computer)
Weblab server (PC running the RecTool)
TIM (FPGA-based board)

Figure 1. Architecture of the IEEE1451.0-compliant FPGA-based weblab.

For reconfiguring the infrastructure with different weblab modules the weblab server provides a Reconfiguration Tool (RecTool), accessible through the web interface represented in figure 2. The infrastructure was designed according to a thin implementation [5] of the IEEE1451.0 Std., and is supported by an NCAP and a TIM, interfaced through a serial RS-232 connection. While the weblab server and the NCAP are respectively implemented through a PC and a thin computer, the TIM adopts an FPGA-based board. The reconfiguration capability of the FPGA and the I/O interfaces provided by the board, enable its reconfiguration with weblab modules required to interface the target experiments. By following a reconfiguration process, these modules are bond to a predefined IEEE1451.0-compliante module inside the FPGA using a set of Transducer Channels (TCs). Once reconfigured, students and teachers may access the TCs to

978-1-4799-2024-2/14/$31.00 ©2014 IEEE

control the weblab modules using IEEE1451.0-HTTP API methods.



Weblab Reconfiguration Tool

Select a file to upload:

Allowed files:
Build configuration: configuration (*.conf)
Build project: teds (*.teds); map (*.map); verilog (*.v / *.vh); pinout (*.ucf)
Reconfiguration: svf (*.svf); bit (*.bit)

Press here to refresh the panel interface reconfiguration tool

Build files (configuration operation):
TEDS configuration done.
Check Build TEDS Report:
---> Bteds_2012-01-03_10:14:10.rep

Bind Transducers/Instruments done.
Check Build Bind Report:
---> Bbind_2012-01-03_10:14:10.rep

Server State: Server time: Tue Jan 3 10:15:03
Server available for synthesizing the project.

Build		
file name	**select to build**	**select to delete**
Configuration files		□
configurationButtons.conf	○	□
configuration.conf	●	□
Project files	□	□
mpp_button_controller.v	□	□
mpp_controller.v	□	□
Input_8bits_TC_Channel.teds	□	□
Output_4bits_main.v	□	□
map_teste.map	□	□
pinout_buttons.ucf	□	□
XdrcName_TEDS.teds	□	□
Meta_TEDS.teds	□	□
mpp_TC_Channel.teds	□	□
Input_8bits_tasks.vh	□	□
Input_8bits_main.v	□	□
mpp_2.v	□	□
mpp_clk_generator.v	□	□
	build	DELbuild

Reconfiguration

No project available. Please rebuild the project.

If the project were built, press StartSyn button to create the programming files (*.bit) and (*.svf).

file name	**select to reconf.**	**select to delete**
Synthesized Files		□
2012-01-17_14:39:29.bit	○	□
tim_mainTransition.bit	○	□
tim_main_press.bit	●	□
	reconfigure	DELreconf

Reports

file name	**select to delete**
	□
Syn_2012-01-17_14:39:29.rep	□
Bteds_2012-01-17_14:39:10.rep	□
Bbind_2012-01-17_14:39:10.rep	□
	DELreports

Figure 2. Screenshot of the weblab RecTool interface.

IV. Demonstration Methodology

During the exhibition session, the presented weblab prototype will be demonstrated. Due to the specificity of the implemented solution, the audience will be guided through different stages, which involve **reconfiguring** the weblab with a set of predefined weblab modules using the RecTool, and **accessing** it using standard commands issued by IEEE1451.0-HTTP API methods.

As illustrated in figure 3, two types of experiments, using two distinct configurations, will be prepared, namely: i) a simple digital I/O hardware loop control and; ii) a step motor control.

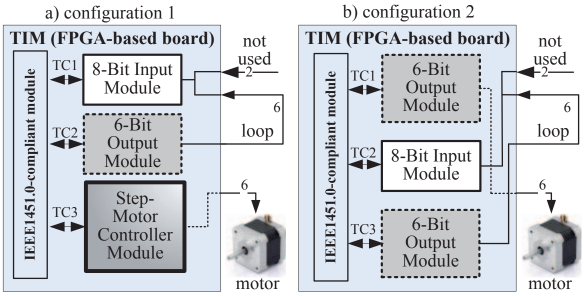


Figure 3. Weblab configurations.

The audience will have access to a supporting webpage detailing the different configurations of the weblab infrastructure, which involves the use of three weblab modules namely, two digital I/O modules and one Step Motor Controller Module (SMCM) that automatically generates step sequences for rotating the step motor. These modules will be provided through a set of HDL files that will be uploaded to the weblab (using the RecTool) to reconfigure the infrastructure according to the two configurations. The first configuration uses the 3 weblab modules. The I/O hardware loop control is made using the two digital I/O modules using TC1 and TC2, and the step motor is controlled by the SMCM using the TC3. The second configuration is adopted to control the same target experiments, but the weblab modules are changed. The I/O hardware loop control uses the same I/O digital modules but now controlled through TC2 and TC3, and the step motor is now controlled through a replication of the Output digital module using TC1, which requires users to define each step sequence for rotating the step motor. The proposed demonstration stages are illustrated in figure 4.

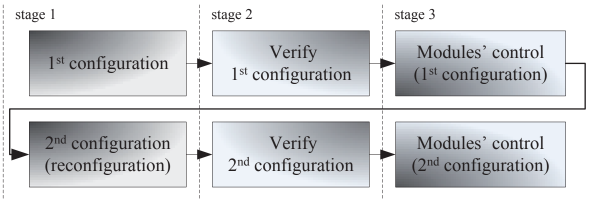

stage 1
stage 2
stage 3
1st configuration
Verify 1st configuration
Modules' control (1st configuration)
2nd configuration (reconfiguration)
Verify 2nd configuration
Modules' control (2nd configuration)

Figure 4. Demonstration stages.

V. Conclusion

The proposed demonstration intends to emphasize the relevance of using a low-cost solution for implementing a reconfigurable and standard-based weblab solution. The possibility of reconfiguring a weblab with different weblab modules able to be shared and replicated without changing the physical infrastructure will be highlighted. Although the specificity of the implemented weblab, the different demonstration stages will have a personalized assistance to demonstrate its relevance for designing weblabs. All devices required to implement the weblab architecture (the weblab server, the underlying infrastructure and the target experiments) will be placed in the exhibition room.

References

[1] Mohamed Tawfik et al., ‘Virtual Instrument Systems in Reality (VISIR) for Remote Wiring and Measurement of Electronic Circuits on Breadboard’, *IEEE Trans. Learn. Technol.*, vol. PP, no. 99, p. 1, 2012.

[2] Jan Machotka, Zorica Nedic and Özdemir Göl, ‘Collaborative Learning in the Remote Laboratory NetLab’, *J. Syst. Cybern. Informatics JSCI*, vol. 6, no. 3, pp. 22–27, 2008.

[3] IEEE Std. 1451.0^{TM}, ‘IEEE Standard for a Smart Transducer Interface for Sensors and Actuators - Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats’, *Inst. Electr. Electron. Eng. Inc*, p. 335, Sep. 2007.

[4] Eugene Y. Song and Kang Lee, ‘Understanding IEEE 1451-Networked smart transducer interface standard What is a smart transducer’, *IEEE Instrum. Meas. Mag.*, pp. 11–17, Apr. 2008.

[5] Ricardo J. Costa, Gustavo R. Alves and Mário Zenha-Rela, ‘Extending the IEEE1451.0 Std. to serve distributed weblab architectures’, *1st Exp. Int. Conf. Expat11 Calouste Gulbenkian Found. Lisb.-Port.*, p. 7, Nov. 2011.

978-1-4799-2024-2/14/$31.00 ©2014 IEEE