Rapid FPGA Development Framework Using a Custom Simulink Library for MTCA.4 Modules

Paweł Prędki, Michael Heuer, Łukasz Butkowski and Andrzej Napieralski, *Senior Member, IEEE*

***Abstract*—The recent introduction of advanced hardware architectures such as the Micro Telecommunications Computing Architecture (MTCA) caused a change in the approach to implementation of control schemes in many fields. It required the development to move away from traditional programming languages (C/C++) to hardware description languages (VHDL, Verilog), which are used in FPGA development. With MATLAB/Simulink it is possible to describe complex systems with block diagrams and simulate their behavior. Those diagrams are then used by the HDL experts, to implement exactly the required functionality in hardware. Both the porting of existing applications and adaptation of new ones requires a lot of development time from them. To solve this, Xilinx System Generator, a toolbox for MATLAB/Simulink, allows rapid prototyping of those block diagrams using hardware modelling. It is still up to the firmware developer to merge this structure with the hardware-dependent HDL project. This prevents the application engineer from quickly verifying the proposed schemes in real hardware. The framework described in this article overcomes these challenges, offering a hardware-independent library of components that can be used in Simulink/System Generator models. The components are subsequently translated into VHDL entities and integrated with a pre-prepared VHDL project template. Furthermore, the entire implementation process is run in the background, giving the user an almost one-click path from control scheme modelling and simulation to bit-file generation. This approach allows the control theory engineers to quickly develop new schemes and test them in real hardware environment. The applications may range from simple data logging or signal generation ones to very advanced controllers. Taking advantage of the Simulink simulation capabilities and user-friendly hardware implementation routines, the framework significantly decreases the development time of FPGA-based applications.**

***Index Terms*—DESY, FLASH, European XFEL, feedback control, control system, MTCA, Simulink, hardware description languages, System Generator, Xilinx**

I. Introduction

THE Micro Telecommunications Computing Architecture (MTCA) hardware has been steadily replacing the older, VME-based infrastructure in many systems at Deutsches Elektronen-Synchrotron (DESY) [1]–[3]. Among the systems in question are the Low Level Radio Frequency (LLRF) control system as well as the optical synchronization system at the Free Electron Laser in Hamburg (FLASH) (see Fig. 1). MTCA-based installations at FLASH are considered to be test systems for the European X-Ray Free Electron Laser (XFEL). Nevertheless, improved performance and full functionality is still required. The above mentioned systems take advantage of both off-the-shelf hardware, such as the SIS8300 and SIS8300L boards from Struck, and in-house developed Advanced Mezzanine Cards (AMC). One common feature of those boards is the use of Xilinx Field Programmable Gate Array (FPGA) integrated circuits, which perform, among other tasks, data processing required in control systems.

P. Prędki and A. Napieralski are with the Department of Microelectronics and Computer Science of the Lodz University of Technology, Poland. e-mail: {ppredki, napier}@dmcs.p.lodz.pl

Łukasz Butkowski and Michael Heuer are with the Deutsches Elektronen-Synchrotron. e-mail: {lukasz.butkowski, michael.heuer}@desy.de

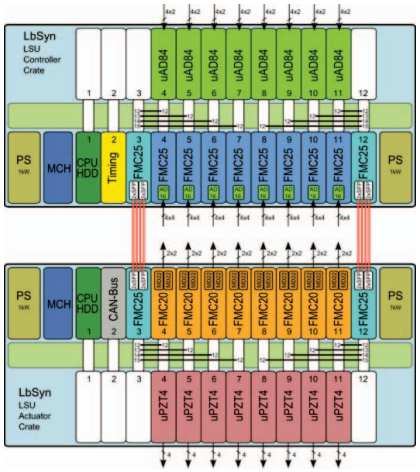

Fig. 1. Proposed MTCA-based infrastructure of the fiber link stabilization unit control - part of the optical synchronization system

The use of FPGAs requires a different development approach and different set of skills compared to the Digital Signal Processor (DSP) boards often used in VME [4]. The data processing algorithms now have to be developed using hardware description languages such as VHDL or Verilog. Even though FPGAs were also used in VME, those devices were less complex and less diverse than what MTCA brings.

This extended abstract shortly presents the motivation behind the work and the proposed solution to the indicated problem. The more detailed description of the designed toolset as well as its practical applications will be presented in the full paper.

II. Motivation

The FPGA projects that are deployed in the LLRF and optical synchronization systems are divided into two separate parts. One section of the VHDL code is responsible for controlling the hardware (e.g. external memories, ADCs, PCI Express) and is specific to a given board. The other section deals only with the application-specific algorithms and does not interact with the hardware directly (see Fig. 2).

A typical development process for new FPGA-based applications consists of the application engineer designing and testing the control scheme using mathematical formulas and/or

978-1-4799-3659-5/14/$31.00 ©2014 IEEE

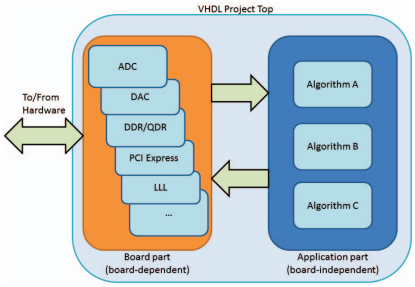

Fig. 2. Board/Application division of VHDL projects

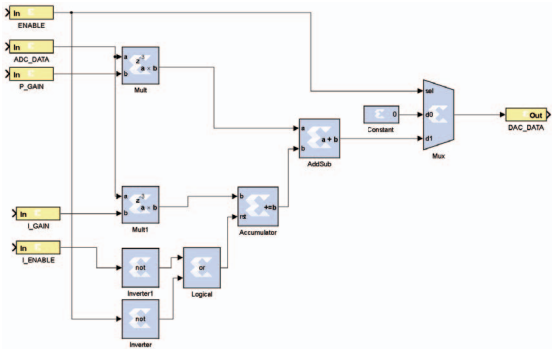

Fig. 3. Pure Simulink/System Generator design of a simple PI controller

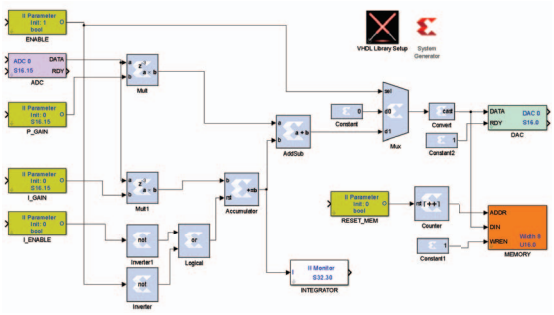

Fig. 4. PI controller design using the VHDL LIbrary for Simulink

block diagrams, often with the help of Matlab software. Then, an FPGA expert implements the algorithms using hardware description languages, and integrates it with the board-specific part in a Xilinx ISE project [5]. Whenever a change needs to be introduced, the application engineer revises the design and forwards it to the FPGA expert, who needs to rework the HDL code to apply the changes.

This approach makes it difficult for the application engineer to quickly evaluate if the proposed design is suitable for hardware implementation. Software simulation using Matlab and Simulink [6] is usually not enough. The FPGA expert always needs to be involved in the process, which leads to a longer development time.

A solution should be found that allows the application engineer to work on and test in hardware the application part of any FPGA-based project, without deep knowledge of HDLs and the underlying hardware structure. At the same time, the FPGA expert should be relieved of some of the tasks dealing with the application.

III. Custom Library Approach

The authors propose a custom Simulink library, called the VHDL Library, that overcomes those problems. It gives the user access to blocks that correspond to specific hardware elements (memories, ADCs, DACs, high-speed transceivers, etc.). This means, for example, that instead of using a generic Xilinx System Generator input blocks for both the ADC input and the P gain parameter in a controller model (Fig. 3), two different blocks should be used, encompassing the functionality of an ADC data stream and a PCI Express memory-mapped register [7], respectively (Fig. 4). Not only does this approach make the model easier to read and understand, it also allows the tools to generate instances of specific VHDL code blocks in the application part of the project and interface them directly with the board part. As a result, the application engineer is able to follow the path from design to hardware implementation without the help of the FPGA expert, which was the initial requirement. The VHDL Library toolchain also generates configuration scripts that help the application engineer monitor and control his or her design using Matlab.

IV. Conclusions And Future Plans

The proposed solution of a VHDL Library helps the application engineers to quickly design a model, simulate it and test in hardware with limited knowledge of any HDL and structure of Xilinx ISE projects.

It is also possible for users with more advanced knowledge of VHDL to look into the code because it consists of instantiated VHDL entities of modules developed by the FPGA experts. This approach differentiates the VHDL Library from other Simulink-based development frameworks [8]. It also allows the library to be expanded by additional modules that are requested by the application engineers.

The fact that the VHDL Library has been successfully used in two projects for the optical synchronization system proves its value. The details concerning the application examples and the specific structure of the components of the VHDL Library will be presented in the full paper.

References

[1] J. Branlard *et al.*, "MTCA.4 LLRF system for the european XFEL," in *Mixed Design of Integrated Circuits and Systems (MIXDES), 2013 Proceedings of the 20th International Conference*, pp. 109–112, June 2013.

[2] A. Hidvegi *et al.*, "Timing and triggering system for the European XFEL project - a double sized AMC board," in *Real Time Conference (RT), 2012 18th IEEE-NPSS*, pp. 1–3, June 2012.

[3] K. Przygoda, M. Felber, and H. Schlarb, "MTCA.4 compliant piezo driver RTM for laser synchronization," in *Mixed Design of Integrated Circuits and Systems (MIXDES), 2013 Proceedings of the 20th International Conference*, pp. 123–126, June 2013.

[4] P. Predki, "DSP-based controller for optical synchronization applications for the Free Electron Laser in Hamburg," in *Proc. OWD*, pp. 201–204, 2011.

[5] http://www.xilinx.com/products/design-tools/ise-design suite/index.htm, "Xilinx ISE [Online]."

[6] http://www.mathworks.com/products/simulink, "MathWorks Simulink [Online]."

[7] A. Piotrowski *et al.*, "Integral Interface - universal communication interface for FPGA-based projects," in *Mixed Design of Integrated Circuits and Systems, 2007. MIXDES '07. 14th International Conference on*, pp. 115–119, June 2007.

[8] B. Fernandes, P. Gessler, and C. Youngman, "High level FPGA programming framework based on Simulink," in *Proc. ICALEPCS*, pp. 782–785, 2013.