

# Rapid Prototyping FPGA Environment for Mixed Signal Design with Built-In Web-Interface

Manfred Ley, Dietmar Scharfer, Stefan Zupanc

**Abstract**—A low cost Field Programmable Gate Array based rapid prototyping environment for mixed signal ASIC development and concept validation is presented. The system is built up using a commercial general purpose FPGA evaluation board enhanced by a plug-in board featuring required analog functionality. A webserver embedded into the prototyping FPGA enables data collection and control of the design under investigation via Ethernet and allows seamless integration of the system into a networked laboratory instrument setup. Based on the specific requirements of the ASIC design targeted by this work, design considerations for analog board, FPGA hardware design and embedded control software design are discussed. Results in terms of digital FPGA resources and speed limitations as well as general restrictions of the concept are given. The prototyping environment presented showed its usefulness for concept engineering, verification and customer system evaluation at a considerably lower cost factor than most industrial emulation and prototyping systems.

**Index Terms**— FPGA, embedded webserver, mixed-signal ASIC, rapid prototyping, Ethernet

## I. INTRODUCTION

Rapid Prototyping has gained increasing importance within the semiconductor industry in recent years for several reasons. First of all implementing the desired functionality of a complex integrated circuit on a more flexible and cost saving platform than the final ASIC technology gives a strong enhancement regarding verification confidence. With the broad availability of reconfigurable digital Field Programmable Gate Arrays (FPGA) and also mixed-signal Field Programmable Systems on Chip (FPSoC) a real world comparison of different system solutions using the actual application environment became possible. This improves accuracy of results compared to model based simulations (as each model by nature is an

abstraction of reality) as well as verification time by executing at same order of processing speed than the final Application Specific Integrated Circuit (ASIC) design. An important second aspect related to concept engineering is the opportunity to develop and prove various solution ideas together with possible chip customers in their application environment [1], [2]. With ASIC design and manufacturing costs rising rapidly, potential chip customers even do not sign a design contract before clear proof of concept and feasibility. The work presented here shows a research level design for a cost effective FPGA based framework for all the tasks described before. The prototype ASIC design under investigation is not part of this paper except for giving the requirements for the system.

We use a standard digital FPGA evaluation board from company Xilinx (XCSP3ADSP1800) enhanced by a custom designed digital-analog and analog-digital interface board as well as standard data interfaces to PC's and laboratory measurement equipment. A special requirement is control of the FPGA over ethernet via embedded webpage, so a microcontroller executing webpage and design control software needs to be part of the prototyping FPGA.

A complete prototyping laboratory setup in Fig. 1 consists of the FPGA and analog prototyping boards, the application environment for the ASIC design under investigation (DUT), a debug PC for configuring and monitoring the FPGA internal digital circuit content via serial and JTAG interface, various additional laboratory equipment from power supply to waveform analyzer and a remote control PC for measurement mode setup and data visualisation

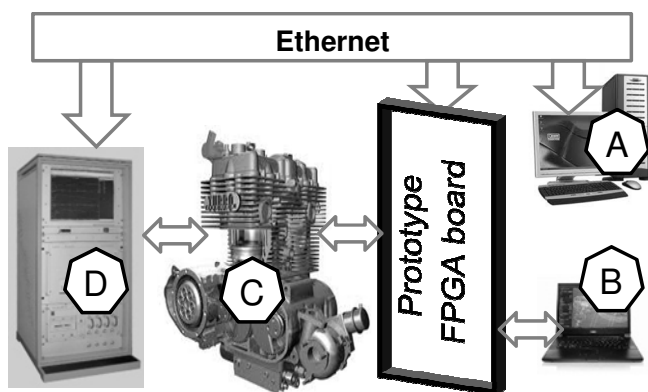


Fig. 1. Complete prototyping laboratory environment. (A) Measurement control PC, (B) FPGA debug PC, (C) Application environment, (D) Laboratory measurement equipment

Manuscript received November 26, 2012. This work was supported in part by adhoc Hard- und Software GmbH Nfg KG, Klagenfurt, Austria and PIU-PRINTEX – Leiterplatten GmbH, Vienna, Austria.

M. Ley is with the Department of Systems Engineering, Carinthia University of Applied Science, Villach, Austria (phone: +43-(0)4242-90500-2119; fax: +43-(0)4242-90500-2010; e-mail: m.ley@fh-kaernten.at).

D. Scharfer is graduate student with the Department of Systems Engineering, Carinthia University of Applied Science, Villach, Austria (e-mail: Dietmar.Scharfer@edu.fh-kaernten.ac.at).

S. Zupanc is with adhoc Hard- und Software GmbH Nfg KG, Klagenfurt, Austria, and graduate student with Carinthia University of Applied Science, Villach, Austria (e-mail: Stefan.Zupanc@edu.fh-kaernten.ac.at).

controlling the FPGA circuit, as well as all additional lab equipment via Ethernet network.

The following chapters give more details on system requirements, interface board design, FPGA design regarding webserver and digital signal processing (DSP), web interface software design and finally results and conclusion.

## II. SYSTEM REQUIREMENTS

Fig. 2 gives a block level overview on analog and digital parts of the system according to requirements given below.

### A. Analog Interface Stage

Main purpose of this circuit block is the generation of an arbitrary analog signal feed to the customer application environment and sampling of the analog response signal into the digital signal processing domain. For this first system investigation described two independent analog signal paths - means two DAC's and two ADC's working independently - are required. Data conversions need 16 bit resolution at conversion rates up to one Msamples/second. Analog inputs need to be buffered with the option of adding or compensating a DC offset to the incoming signal. The analog output signal should be feed out directly from the DAC but for demonstration purposes also over a second path via an audio range amplifier.

### B. FPGA Data handling Stage

Conversion data for DAC's and sampling data from ADC's are generally flowing via FPGA block RAM which is accessible in two different ways. First, dedicated hardware blocks for each DAC/ADC read and write data from/into a set memory space according to set conversion rates with similar interfaces handling data transfer to the DUT. Second, the embedded microcontroller should be able to supply output data to and take sample data from the block RAM for further analysis and storage. A similar data path via block RAM may sample DUT's internal test data if necessary for DUT debugging. Additional fast datapathes can be multiplexed as bypass of block RAM to avoid the input/output delay seen by the DUT.

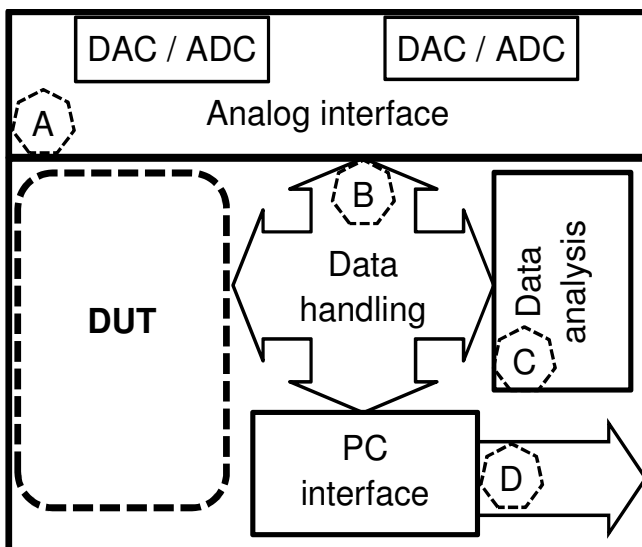


Fig. 2. Block diagram of the mixed signal prototyping system. Block names A to D refer to chapter II System Requirements.

### C. FPGA Data Analysis Stage

Sampled ADC data should be permanently analyzed regarding spectral content with live display and data delivery over Ethernet connection to the control PC.

### D. FPGA – PC Interfaces

The whole FPGA system has to fit into a software controlled laboratory setup where equipment is connected via ethernet. Furthermore, viewing specific data and function mode setup of the FPGA should be possible from remote location outside the lab via loading a webpage from any standard office PC with a web browser. Detailed DUT debugging and FPGA design needs JTAG programming interface and a serial port in addition.

## III. ANALOG INTERFACE BOARD DESIGN

This plug-in printed circuit board assemblies all additional analog parts which are not available on the XCSP3ADSP1800 FPGA main board. The analog board is designed as four-layer controlled impedance stack to minimize crosstalk from high speed digital interface lines to analog signals. All relevant signals are terminated and laid out with respect to line impedance and signal delay. Two slot high speed expansion connectors provide a direct plug-in 168-pin electrical interface to the FPGA base board as well as mechanical fixture of the board. To reduce parasitic elements and crosstalk between signal lines all parts are SMD-mounted in small form factor, so circuits for ADC's, DAC's, power supply and support features can easily be spatially separated in the board layout. Fig. 3 shows a photograph of the complete board.

### ADC/DAC Channels

Two 16bit 1Msample/sec ADC/DAC channels are implemented by Analog Devices AD7653 and AD5542 parts. Input lines are terminated for 50 ohm and buffered including the option to work bipolar and apply DC level shifting. Output lines are buffered and terminated to support

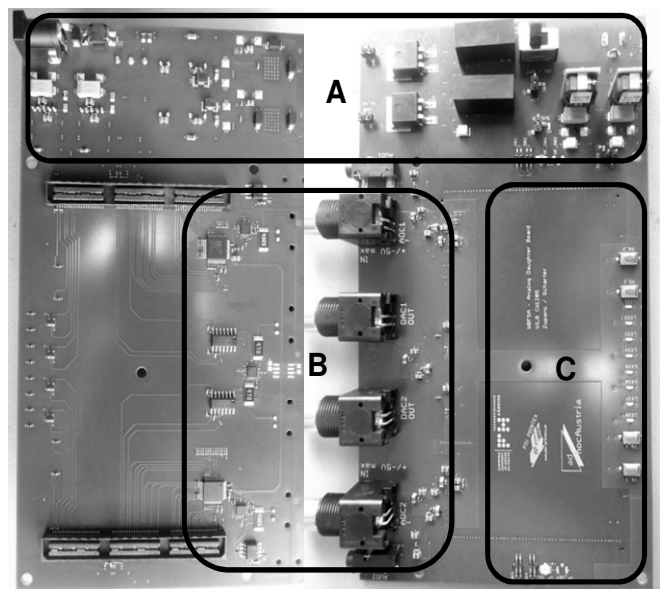


Fig. 3. Analog interface board, right side top view, left side bottom view, (A) Power supplies, (B) ADC / DAC channels, (C) Support Features, LED's, buttons, miniature probe connectors.

sufficient signal quality and driving strength to drive the application interface in 50 ohm systems. Digital data interface to the FPGA is 16bit parallel for ADC's and 3-wire serial for DAC's. Selection of discrete converter parts was constrained by the target specification of later used analog IP blocks for the planned ASIC design in terms of digital interface, conversion rate and resolution.

*Power Supply*

Supply voltages are generated on-board from an external 12V DC switching converter source. Digital supply at +5V and +3.3V, analog +/- 5V and 2.5V reference voltages are generated and distributed via separated layout planes including filtering and decoupling elements.

*Support Features*

To support debugging and control of circuit operation in the lab, several spare FPGA digital IO pins are available on the interface board and routed to LED's, push buttons and a logic analyzer probe connector. On the analog side miniature oscilloscope probe connectors are available as well as audio amplified outputs for DAC signals.

IV. FPGA HARDWARE DESIGN

The FPGA board used for this investigation features a Xilinx Spartan3ADSP1800 FPGA [3], 128MB DRAM, Flash memory, Ethernet PHY chip, JTAG and serial interfaces. The board system clock is 125MHz.

To support verification and evaluation of a DUT according to requirements, at least three main hardware blocks are needed (Fig. 4): first, a microcontroller for web interface, second, data transfer between microcontroller, DUT and analog IO's and third, a signal processing block. All support hardware should consume a minimum of FPGA resources to leave enough space for implementation of the DUT prototype. Design is done at VHDL level and use of Xilinx Core Generator IP, where the Xilinx IP usage is restricted to blocks which are available also in new generation FPGA's for easy portability to faster and bigger FPGA's than the mature SP3ADSP1800 type. Xilinx design tools ISE System Edition together with Mentor Graphics ModelSim are used for design, verification and implementation.

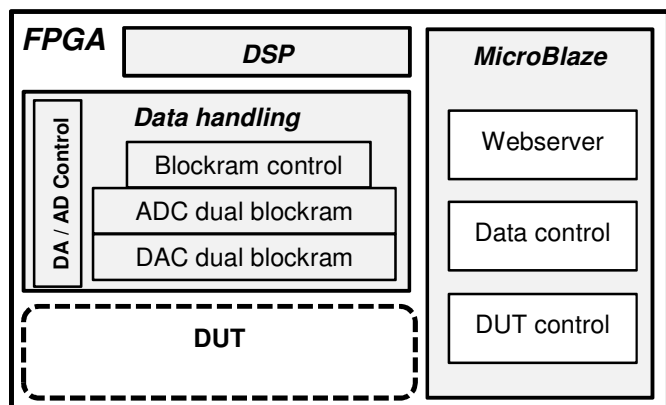


Fig. 4. FPGA hardware block diagram.

A. Microcontroller System

A microcontroller (uC) handling the protocol is necessary to fulfill the Ethernet connection requirement. As many available FPGA boards already feature an Ethernet PHY and external memory resources we decided to implement the uC into the FPGA instead of attaching an external uC system to FPGA IO pins. So in the desired remote control operation the microcontroller is able to read and write FPGA internal memory and registers directly in its address space without an additional external interface.

A 32bit Xilinx Microblaze core [4] with etherlite, uart, debug interface, DRAM controller and block RAM controller as local bus peripherals is used (Fig. 5). All FPGA internal memories and hardware control registers reside in the uC block RAM interface address space for direct software access. With this architecture all additional hardware blocks can be changed without touching the MicroBlaze system itself. This makes hardware changes faster and control software development easier than having all add-on hardware blocks individually attached as MicroBlaze bus peripherals. As the microcontroller system is smaller with less bus load it is also easier to place & route for a desired speed.

The system operates at 62.5MHz clock rate, which turns out to be at the limit for the FPGA in use, especially regarding DDR2 memory interface timing. As block RAM is primarily used for hardware data interfaces the DRAM is used for software, data storage and file system (see software chapter).

B. Data Handling

All circuit blocks are instantiated twice for the two parallel data paths from DUT/microcontroller to DAC output and ADC input to DUT/microcontroller. The memory write controller supports two separate physical RAM blocks to avoid any conflict between streaming data for the DAC

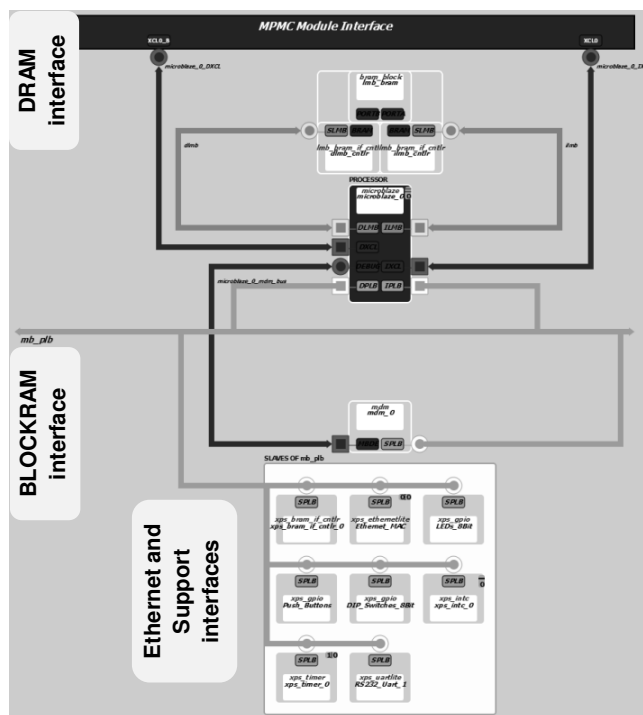


Fig. 5. MicroBlaze controller system block diagram.

from DUT and waveform write operation from the microcontroller. Due to slow write operation from microcontroller and unpredictable timing of new data sent via Ethernet, two equal uC-memories are implemented where one is delivering data to the DAC and the other one can be filled with new data independently. As for above mentioned timing reasons the microcontroller writes only sample data for 2 periods of the analog signal, the memory read controller is responsible for periodically reading the correct uC-RAM, switching write operation accordingly and reading streaming data from DUT memory. Data from the selected memory is sent to the DAC via parallel/serial conversion block and SPI interface.

ADC input data is handled in a similar way including the parallel uC-memory structure to enable high conversion rates to the DUT but slower read of complete data sections for PC storage and visualization.

### C. Digital Signal Processing

Within the given application a key characteristic response of the DUT can be seen in the spectral content of analog input data. Therefore a 1024-point FFT algorithm was implemented for hardware processing ADC-1 data. In the design presented here a Xilinx Core Generator streaming FFT block [5] configured utilizing XtremeDSP slices of the FPGA was used (X-FFT 2), but also an available Winograd-FFT block [6], [7] and a Xilinx IP core in standard configuration was examined for resource and speed comparison.

As summarized in Table I the Xilinx IP configuration X-FFT 2 delivers best resource usage and highest possible clock rate, but at the price of dependence on Xilinx tools and little insight into actual operation of the encrypted core.

Test implementation was done for 256-point FFT on three different FPGA devices to explore mainly possible speed improvements and compare resource consumption.

The FFT calculation results are stored within two alternating active memory blocks for having available a complete valid data set to be read by the microcontroller on request via web-page.

TABLE I  
FFT RESOURCE / SPEED COMPARISON

FPGA Device Logic Resource	Winograd <sup>a</sup>	X-FFT 1 <sup>b</sup>	X-FFT 2 <sup>c</sup>
Spartan 3 E 1600			
Flip Flops	6069	7031	3896
LUTs	10267	7561	3882
DSP blocks	4	0	12
max. Clock (MHz)	90	180	190
Spartan 3A DSP 1800			
Flip Flops	6024	6886	2911
LUTs	10241	7564	3269
DSP blocks	4	0	12
max. Clock (MHz)	85	165	195
Virtex 5 LX 50			
Flip Flops	6026	7114	3452
LUTs	7859	6961	3243
DSP blocks	4	0	9
max. Clock (MHz)	185	330	370

<sup>a</sup> VHDL standard ISE implementation

<sup>b</sup> Xilinx LogiCore configuration CLB logic

<sup>c</sup> Xilinx LogiCore configuration resource optimization

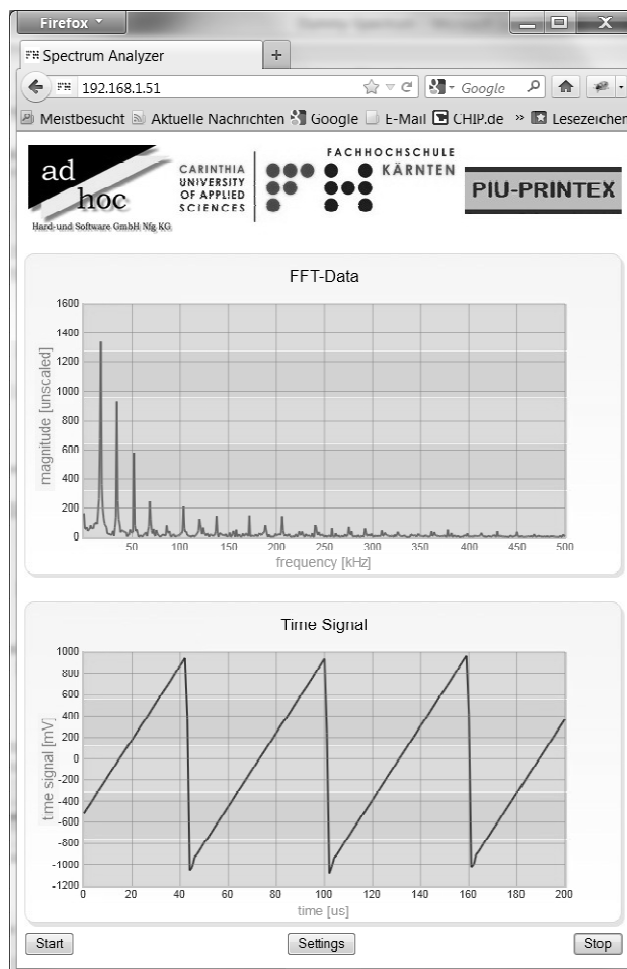


Fig. 6. Visualization and control webpage example. Sampled analog signal and spectral view for one ADC channel. Control buttons for start/stop DAC waveform generation and setting of waveform display.

### V. MICROBLAZE SOFTWARE DESIGN

The software executed by FPGA's MicroBlaze has to support Ethernet connection, deliver a webpage to a connected PC's web-browser, deliver data to the webpage and transfer commands and data sent from the PC into bit-level data in block RAM and registers. Xilinx small footprint XilKernel operating system is utilized for multithreading and basic file system operations. Ethernet and web functionality is supported by the included lightweight-IP-stack.

General software flow after start of Microblaze operation with a bootloop is loading the XilKernel and an initial software thread. Within this thread default configuration settings for DUT, ADC's and DAC's are given to hardware registers and a network initialization thread is opened. The Ethernet PHY is configured and variables like IP address and speed are set. Afterwards the TCP/IP stack is executed and puts the system into network listen mode.

When the FPGA is called by its assigned IP address from a PC's web-browser, a response-thread is opened and an Ethernet connection is established. According to the received request, an http-response is generated and sent back. The first response usually includes a file request for index.html or equivalent file, which can be displayed on the client's web-browser (Fig. 6). Other http-responses are delivering data frames for display within java scripts or

receiving commands to change control register values in the hardware [8]. All required files (html code, pictures, java script code) are supplied from XilKernel's file system residing in the DRAM. The web-page design can be done using any suitable web design tool on a PC. Files are packed into the file system offline and attached to the software executable for download into memory.

Startup of the system within a laboratory environment usually is done by first configuring the FPGA hardware and second downloading binary software files into memory via JTAG programming interface.

For self-contained operation all binaries may be stored in on-board flash memory, from which the FPGA hardware is configured at power up. Then all software binaries are copied by MicroBlaze boot code into DRAM.

## VI. RESULTS

Most important aspect of the design is FPGA device utilization to have enough resources free for implementation of the prototype design under investigation. As shown in Table II for the Spartan3ADSP1800 device about half of the slices and one third of LUT, RAM and Flip Flop resources are consumed by the prototyping environment presented. This leaves sufficient resources for the DUT in our first setup.

From FPGA performance point of view two different aspects are important. First, data throughput in the signal path DUT – DAC – ADC – DUT determines if the system is useable at all. We can achieve full DAC/ADC conversion rates of 1 Msamples/sec for a small DUT circuit. With FPGA resource utilization going up to 70% speed becomes limited due to speed grade and hardware architecture of the FPGA type used on our board. Trial implementation with other device types shows considerable improvement. Second bottleneck visible in timing reports for current implementation is DDR2-DRAM interface to the MicroBlaze controller. Practically this is not that critical as software response and user experience at the web interface is not degraded even by slowing down the processor clock.

The analog interface board shows expected performance of DAC and ADC channels with no additional noise from either power supply or FPGA board visible.

TABLE II  
FPGA RESOURCES USED (without DUT)

Logic Resource	Used	Utilization <sup>a</sup>
<i>Slice Flip Flops</i>	9,918	29 %
<i>Total 4-Input LUTs</i>	10,034	30 %
<i>Used as logic</i>	7,898	
<i>Used as 16x1 RAMs</i>	4	
<i>Used as 32x1 RAMs</i>	122	
<i>Used as Dual Port Rams</i>	538	
<i>Used as shift Registers</i>	1,472	
<i>Occupied Slices</i>	8,925	53 %
<i>DSP48As</i>	43	51 %
<i>RAM16BWERs</i>	31	36 %
<i>bonded IOBs</i>	178	34 %
<i>BUFGMUXs</i>	5	20 %
<i>DCMs</i>	1	12 %

<sup>a</sup> FPGA device XC3SD1800A-FG676-4

Required resources for software storage are quite small. Webserver and data handling executable need 579 Kbyte, the file system for storing the webpage has a size of 572 KB. Compared to available 128 MB DRAM, 16MB parallel Flash and 8 MB serial flash on the XCSP3ADSP1800 evaluation board there is enough room for additional DUT control software.

## VII. CONCLUSION

The prototyping environment presented here enables real world evaluation of mixed-signal ASIC designs by providing DAC and ADC functionality together with reconfigurable digital FPGA resources. Furthermore, operation mode of the system is controlled and data from the prototype design under test is visualized on a standard office PC with web browser without mandatory use of additional laboratory equipment.

Within the FPGA the DUT hardware is clearly separated from support hardware like MicroBlaze controller and memory blocks, so incorporating a design under investigation into the system at VHDL level is easy to do.

As FPGA hardware and controller software may reside in nonvolatile memory with encryption, the whole prototyping system including DUT is self-sufficient and may be given to customers for evaluation.

Utilization of a FPGA vendor (in this case Xilinx) supported off the shelf FPGA evaluation board makes design of the whole prototyping system much easier and cheaper than development of an own FPGA platform board. A possible drawback of the presented concept is of course the discrete non-configurable implementation of DAC and ADC, which means design of a new analog interface board for a new class of analog features required. Even taking into account design and manufacturing cost of the interface board as well as cost of FPGA evaluation board and FPGA design tools, this is a very cost effective and flexible prototyping solution compared to other commercially offered prototyping environments.

As the system presented here delivered valuable experience and proved to be useful, next implementation of a similar system will utilize state of the art FPGA technology and thus improve speed and available hardware resources by factors of three and ten respectively. On the software side a standardized web-interface like LXI (LAN eXtension for Instrumentation) should be implemented.

## ACKNOWLEDGMENT

The authors thank the colleagues from sponsoring companies for valuable discussions and help to finalize and successfully implement this work.

## REFERENCES

- [1] D. Amos, A. Lesea, R. Richter, *FPGA-Based Prototyping Methodology Manual*. Mountain View, CA: Synopsys Press, 2011.
- [2] Pong P. Chu, *FPGA Prototyping by VHDL*. John Wiley & Sons, 2008.
- [3] Xilinx Inc., Spartan-3A DSP FPGA Family Data Sheet, DS610, (4 October 2010). [Online]. Available: [http://www.xilinx.com/support/documentation/data\\_sheets/ds610.pdf](http://www.xilinx.com/support/documentation/data_sheets/ds610.pdf)
- [4] Xilinx Inc., MicroBlaze Processor Reference Guide, Version 12.0 for Embedded Development Kit 13.1, (January 2011). [Online].

Available:

[http://www.xilinx.com/support/documentation/sw\\_manuals/xilinx13\\_1/mb\\_ref\\_guide.pdf](http://www.xilinx.com/support/documentation/sw_manuals/xilinx13_1/mb_ref_guide.pdf)

- [5] Xilinx Inc., LogiCORE IP Fast Fourier Transform v7.1, (1 March 2011). [Online]. Available: [http://www.xilinx.com/support/documentaton/ip\\_documentation/xfft\\_ds260.pdf](http://www.xilinx.com/support/documentaton/ip_documentation/xfft_ds260.pdf)
- [6] G. A. Sathishkumar and K. Dr. Boopathy Bagan, "A Hardware implementation of Winograd Fourier Transform algorithm for Cryptography," Ubiquitous Computing and Communication Journal, vol. 4, no. 3, pp. 38-44.
- [7] Unicore Systems, Pipelined FFT/IFFT 256 points processor, (21 August 2011). [Online]. Available: <http://unicore.co.ua/index.php?page=home&hl=en>
- [8] JS-Charts – JavaScript Chart Generator [Online]. Available: <http://www.jscharts.com/>