Mem. S.A.It. Vol. 88, 186

Memorie della

Design and implementation of projects with Xilinx Zynq FPGA: a practical case

R. Travaglini[1], I. D'Antone[1], S. Meneghini[1], L. Rignanese[1,2], and M. Zuffa[1]

[1] Istituto Nazionale di Fisica Nucleare – Sezione di Bologna, Viale C. Berti Pichat 6/2, 40127 Bologna, Italy, e-mail: riccardo.travaglini@bo.infn.it

[2] Università di Bologna, Dipartimento di Fisica e Astronomia, viale C. Berti Pichat 6/2, 40127 Bologna, Italy

Abstract. The main advantage when using FPGAs with embedded processors is the availability of additional several high-performance resources in the same physical device. Moreover, the FPGA programmability allows for connect custom peripherals. Xilinx have designed a programmable device named Zynq-7000 (simply called Zynq in the following), which integrates programmable logic (identical to the other Xilinx "serie 7" devices) with a System on Chip (SOC) based on two embedded ARM®processors. Since both parts are deeply connected, the designers benefit from performance of hardware SOC and flexibility of programmability as well. In this paper a design developed by the Electronic Design Department at the Bologna Division of INFN will be presented as a practical case of project based on Zynq device. It is developed by using a commercial board called ZedBoard hosting a FMC mezzanine with a 12-bit 500 MS/s ADC. The Zynq FPGA on the ZedBoard receives digital outputs from the ADC and send them to the acquisition PC, after proper formatting, through a Gigabit Ethernet link. The major focus of the paper will be about the methodology to develop a Zynq-based design with the Xilinx Vivado software, enlightening how to configure the SOC and connect it with the programmable logic. Firmware design techniques will be presented: in particular both VHDL and IP core based strategies will be discussed. Further, the procedure to develop software for the embedded processor will be presented. Finally, some debugging tools, like the embedded Logic Analyzer, will be shown. Advantages and disadvantages with respect to adopting FPGA without embedded processors will be discussed.

Key words. System on Chip - Firmware - Embedded Software - Acquisition System

1. Introduction

The project described in this paper consists in a printed circuit board (PCB) realizing one acquisition channel running at 1 GS/s. It is built using only commercial devices. This project has been developed by the Electronic Design Department at the Bologna Division of the Italian National Institute for Nuclear Physics (INFN). This task is a feasibility study to cope with future requests from the experiments and follows a previous experience when a 500 MS/s board was built. Such a fast acquisition rate allows for better performances both in pulse shape discrimination and pile-up rejection,especially when using detectors with fast response (for instance Silicon PhotoMultiplier and fast Scintillators with rising times of the order of 10 ns). In the following, after a short presentation of the Electronic Design Department and its experience on designing with FPGA, the project is briefly introduced.

1.1. The Electronic Design Department at the Bologna Division of INFN: FPGA-related activity and experience

The Electronic Department has to provide support to all the electronics activity performed in the INFN Bologna Division. Its tasks are the design, the development and the installation of electronic devices for the front-end, trigger and acquisition systems of particle physics experiment. It is composed by 14 members with several distinct professional qualifications and skills. Concerning the FPGA, the Electronic Department has a twentyfold experience on using programmable devices of the major firms (Altera, Microsemi, Xilinx). It deals with every FPGA-related aspect : PCB design, firmware development and design of software for embedded microprocessors. Several software applications are used to design, simulate and verify the boards: for instance Orcad from Cadence and Pads, Expedition and Hyperlynk from Mentor. The firmware is designed by using several programming languages: mainly VHDL but also C, C++ and Handel-C as well. Test and control software is mainly developed with C and C++.

1.2. R&D design: a 1GS/s interleaved ADC

The Electronic Department realized a custom PCB for the acquisition system of the FAMU(A. Adamczak et al. 2016) experiment to be used for data-taking during the 2015 run. This board can sample signals at 500 MS/s and it is implemented with commercial 12-bit ADC (AD9434 from Analog Devices 2013). In order to get sampling at a double rate, it was decided to build a system which had the same device as a basic digitizer component but with an interleaved architectures: two ADCs running with clocks having opposite phases. The goal is to get experienced with new problematics like the cross-calibration of the ADCs and the enhanced frequency range of the analog circuit. An advantage of re-using the same ADC component is that some parts of the previous circuit don't need to be re-designed. Moreover, it was decided to use a different FPGA with respect to the FAMU system where an Altera device was foreseen in order to receive data from the ADC and send them to the acquisition PC through USB-3 connection. In this project we decided to acquire the ADC output with a commercial development board hosting a Xilinx Zynq FPGA, called ZedBoard (see Fig. 1). Such a way we could make experience with this device, which was looking promising for future projects. The 1GS/s sample board has been implemented as a mezzanine card, plugged to the Zedboard through a standard low-density FMC connector.

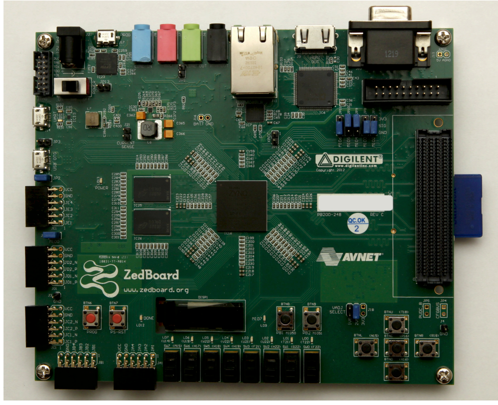

Fig. 1. Picture of the Zedboard (Avnet Inc. 2014). The board hosts a Zynq 7020 as a main processing unit, based on a dual core ARM processor with a maximum clock frequency of 866 MHz. It is equipped with both a 512 MB DDR3 RAM and a 256 Mbit flash memory. Some interfaces available are: GPIO, HDMI, Ethernet, low-density FMC connector.

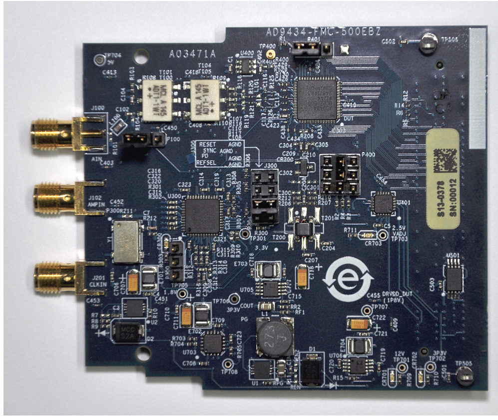

Fig. 2. Picture of the AD9434-FMC evaluation board from Analog Devices. SDA connectors for analog inputs and clock are visible on the left. The visible IC are a clock distributor and jitter cleaner (in the center-left of the board) and the main ADC in the upper region.

1.3. Hardware and firmware current status and preparatory tests

The design of the PCB with the interleaved ADCs is currently on going: an almost-final layout with preliminary routing is under simulation. The firmware for the Zynq FPGA is available in preliminary version supporting one only ADC. This initial design has been successfully tested with an Analog Devices evaluation board (see Fig 2) connected to the ZedBoard through the FMC connector. In the following of this paper, this firmware version will be presented; it will be preceded by an overview of the Zynq FPGA platform as well as a description of the strategy which has been adopted to develop both firmware and software embedded.

2. The Xilinx Zynq FPGA: architecture and design methodology

The Zynq-7000 System On Chip (Xilinx 2015a) from Xilinx is a device implemented in 28 nm technology and composed by two distinct parts (see Fig. 3):

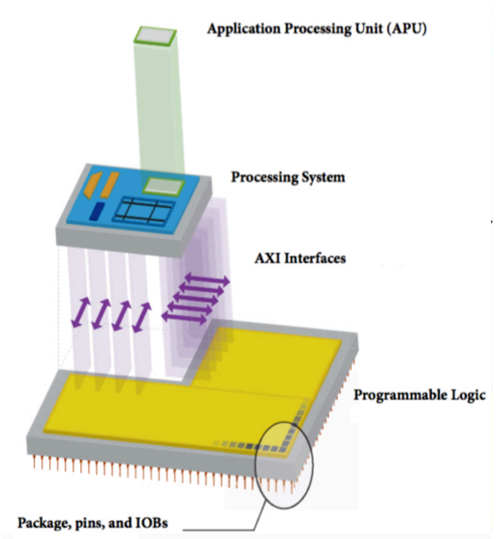

Fig. 3. Pictorial view of the Zynq architecture (courtesy of L. H. Crockett et al. 2014). About 3/4 of the device is in use by the Programmable Logic, while the Process System is located in the upper left quadrant.

- a Processing System (PS) based on high-performance and highly specialized hardware resources;
- a Programmable Logic (PL) section.

They are interconnected with several busses based on AXI protocol.

The PS encompasses a so-called Application Processing Unit formed by a Dual-core ARM®Cortex™-A9 CPU with a three level cache and additional computational engines like the Floating Point Unit (APU). Moreover, the PS includes also a set of peripheral interfaces and controllers (DDR3, USB, SPI, Gigabit Ethernet, I^2C, CAN, ...). This is an advantage with respect to others FPGA: no specific firmware is to be designed in order to implement most of the standard interfaces. Some of them could be very complex as well as require a lot of developing time indeed. Note that all the I/O of these interfaces can be routed either directly on chip pads or to the PL; this is achieved by properly configuring a Multiplexers matrix. Three kind of

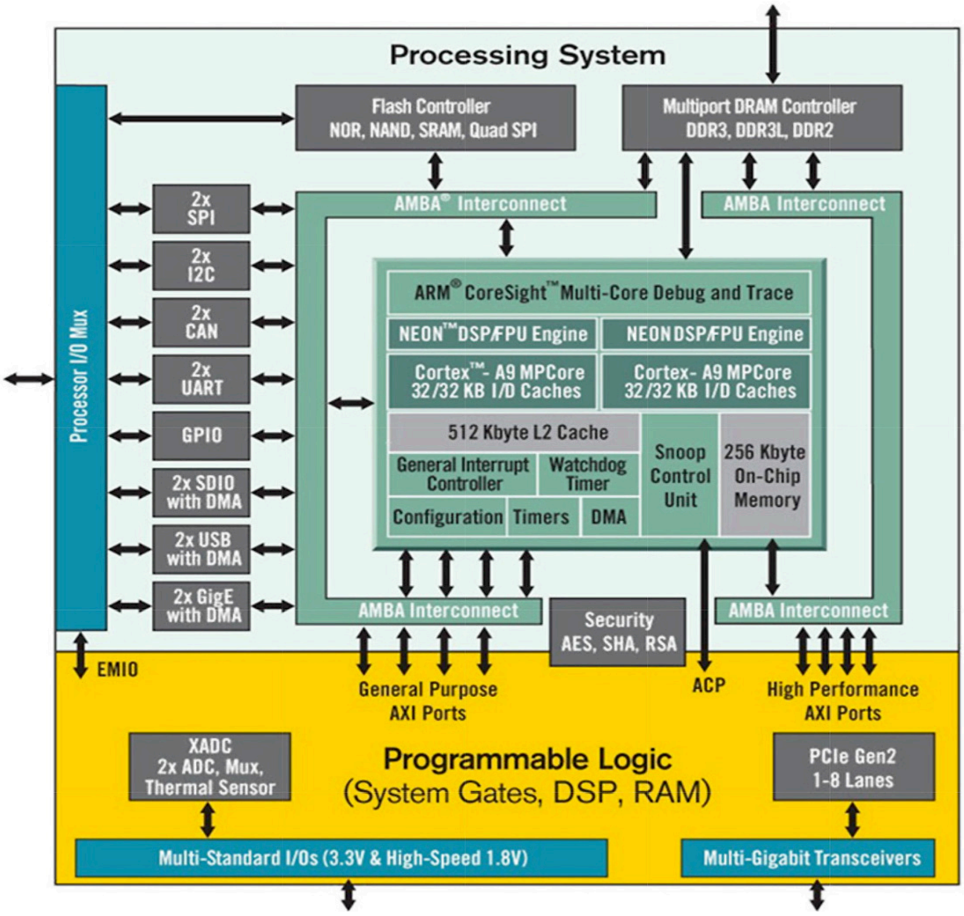

Fig. 4. Schematic representation of the Zynq Processing System (courtesy of L. H. Crockett et al. 2014). The available interfaces/controllers are also shown in grey.

interconnect are implemented towards the PL: High Performance AXI (with enhanced burst support), General Purpose AXI, Accelerator Coherency Port (dedicated to achieve coherency between APU caches and elements within the PL). A schematic view of the PS can be seen in figure 4. Description of the AXI protocol can be found in Xilinx (2015b). The PL is identical for all the devices of the Xilinx serie-7 (see Fig. 5. It is mainly composed by slices of Configurable Logic Blocks (CLB) whose interconnections are programmable. Additional resources are available: clock management tiles, Block RAMs, tranceivers, two channels 12-bit ADC sampling at 1 MS/s, DSP slices, general purpose Input/Output and a PCI Express block. The Zynq product is a family of several devices: not all the resources are available on all devices and, if any, their number is still variable.

The CLB are the dominant resources of the PL. As can be seen in Fig 6, they are composed by two slices of logic gates that slightly differ from each other, connected by a configurable switch matrix which also provides routing to all others PL resources. Dedicated fast carry paths between slices are available too.

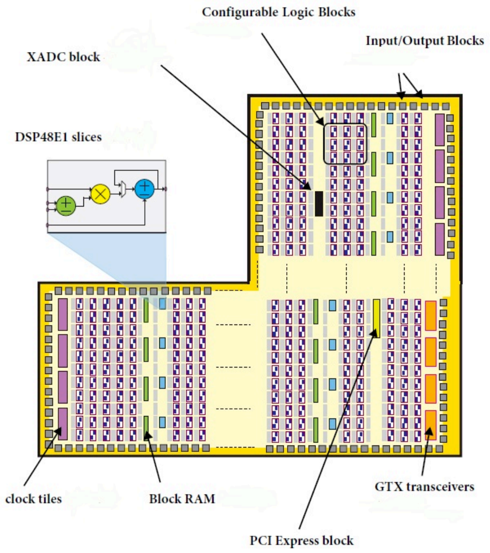

Fig. 5. Layout of the Programmable Logic section of the Zynq (courtesy of L. H. Crockett et al. 2014). Location of the different resources is shown, even though not with accurate proportion.

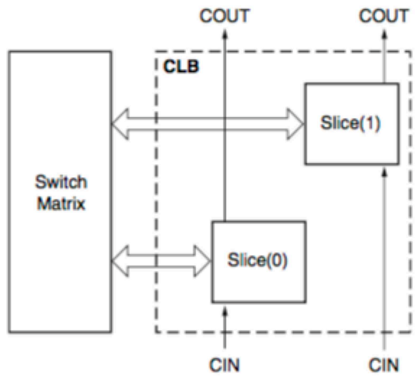

Fig. 6. Schematic view of the Configurable Logic Block (CLB) (courtesy of Xilinx 2014): two slices of programmable logic gates are available. A direct connection among them don't exist: throughout a Switch Matrix they can be connected both each other and with all the other chip's resources. Two types of CLB are implemented: they differ for being composed by slightly dissimilar kinds of slices. A fast carry logic path is available towards the slices of the adjacent CLBs.

As an example, the composition of a slice is shown in figure 7: it consist on several D-type flip-flops and 6-Input programmable look-up tables, all of them connected by configurable mux. Due to the technology size, a considerable number of logic elements are available: for instance, the component 7020 used in this project encompasses about 100,000 flip-flops and 53,000 look-up tables. Due to that, complex firmware designs can be implemented and high performances can be achieved as well. The aforementioned Zynq architecture shows how the well-known FPGA flexibility is associated with high performance processing elements. The firmware design has to benefit from it. A reasonable strategy is to implement with software any functionality which is more easily described with it or which could benefit from being implemented with an operative system. Examples are user-applications, algorithms with sequential data processing, standard protocols management (Ethernet, SPI, ...). On the other hands, the PL has to be used to implement more demanding functionalities: for instance, algorithms that require either greater data rate or higher parallelism which can be implemented as offloading coprocessors. Moreover, custom peripherals interface can be realized in firmware as well. Xilinx provides a complete suite of tools to manage the whole Zynq-based project, called Vivado (suitable for any Xilinx FPGA not older than the serie 7). The firmware can be designed both with high-level programming languages (like VHDL or C/C++) and a proprietary schematic editor (called IP Integrator). The latter is very useful because is mainly targeted on putting together AXI-based IP. Indeed, it provides a full catalog of IP cores, speeding up the development of complex design. Besides, it has powerful features, like automatic connection identification and design rule checking. The software is developed with a tool named SDK (invoked within Vivado). In addition to editing capability and code compilation engine, it provides operative systems, drivers, standard libraries as well as several useful templates. Moreover, SDK and Vivado provide some tools for debugging and monitoring, like GDB for software and an Integrated Logic Analyzer for the firmware.

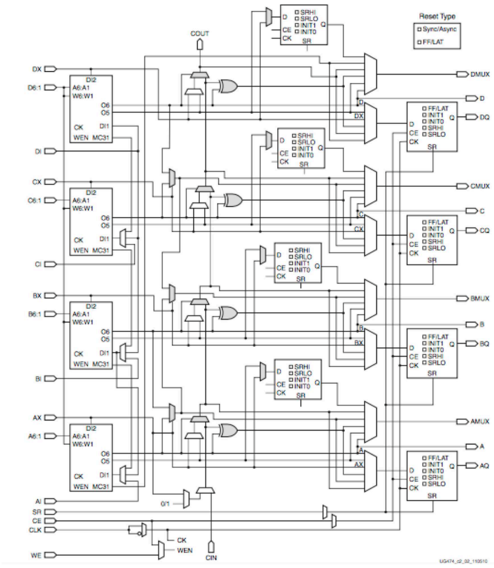

Fig. 7. Schematic view of a SliceM (one of the two available types, picture courtesy of Xilinx 2014). Every SliceM contains four logic-function generators (or look-up tables), eight storage elements, several wide-function multiplexers. These elements are used by all slices to provide logic, arithmetic, and ROM functions. In addition, some slice types support two additional functions: storing data using distributed RAM and shifting data with 32-bit registers. They are realized by properly configuring the slice resources.

3. The implemented design: overview and highlights

The design is inspired by a similar project ([1]) which was developed from Analog Devices to acquire data from the AD9434 evaluation board with a Spartan-6 based Xilinx board (called ML605). This former project was based on a IP core software microprocessor, called Microblaze. For the actual design, the overall architecture has been kept but only few VHDL modules have ben maintained (still with little changes); most of the firmware have been redesigned with IP Integrator. The former Analog's software has been mainly unchanged (it was only a self-test of the connection with the ADC) even though a new application has been developed.

The firmware has been realized in several steps:

- porting of Analog's IP core devoted to data receving from the ADC into a distinct Vivado project so that a new IP core with AXI interface have been designed;
- configuration of the Zynq PS by means of a Xilinx IP core into the IP Integrator;
- realization of the processing environment by adding IP cores (within IP Integrator) to be realized in PL (AXI interconnections, DMA engine, resets infrastructures, Integrated Logic Analyzer probes, custom AD9434 interface) as it can be seen in figure 8;
- automatic generation of VHDL wrapper of the IP integrator design (called Block Design);
- adding of a VHDL module to interface Zynq SPI with ADC SPI implementation (minor changes with respect to the one provided by Analog);
- synthesis;
- insertion of more Integrated Logic Analyzer probes (to debug some critical signals of the synthesized netlist);
- I/O pin assignment;
- timing contraints definition;
- Vivado implementation (i.e. place and route) and programming file (*bitfile*) generation.

Please note that neither Ethernet nor SPI nor DDR3 controller have had to be designed. Since they are already available in the Zynq hardware, they should only to be configured. These is a clear advantage when adopting this device.

One of the more demanding part of the firmware is the custom IP core which have been designed to acquire the ADC data. The output from the AD9434 is a 12-bit digital bus running at 500 MHz: the FPGA I/O deserializers are used to convert it to a 4x12 bit-wide bus running at 125 MHz, so made of

[1] `https://wiki.analog.com/resources/fpga/xilinx/fmc/ad9434`

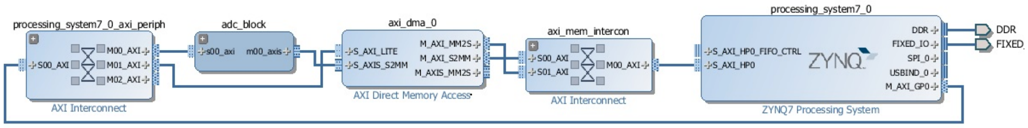

Fig. 8. Schematic of the firmware implemented with IP Integrator. Only AXI connections are shown for a better readability. Each block is a IP core which can easily configured within the GUI. This is well representative of a how a Zynq-based design should be organized and which blocks are needed. It includes the mandatory Zynq Processing System block (to configure the PS) and some general useful cores like the ILA (integrated Logic Analyzer) and the AXI Direct Memory Access (the native DMA into the PS is not use because it is limited to 32-bit data while the axi_ad9434 AXI-master in this revision is a 64-bit data). While this latter cores are optional, the AXI infrastructure relies on some mandatory cores like AXi interconnections and Reset system. The custom ADC interface, designed in VHDL and implemented in a separate Vivado project, appears as a single block, called axi_ad9434. This way of designing custom cores advantages code development and re-usability.

the parallel of 4 consecutive samples. This bus is then padded to a standard 64-bit AXI interface. The core has also AXI-master capability in order to write data directly into the RAM by means of a DMA engine. In this first version of the firmware, the main AXI bus has been configured as a 64-bit running at 100 MHz. The advantage of using the Zynq platform is due to the performance of the PS (with respect to the same functionalities as implemented on programmable logic): in principle a 250 MHz maximum speed is achievable by datasheet(Xilinx 2015c). Tests are on going to increase the performance; the main limitation comes from the IP cores implemented with programmable logic. A summary of the resource used is shown in Table 1 as reported by Vivado (only relevant percentage are listed); low utilization make us confident that there will be enough resources to implement a second ADC channel.

Concerning the software application, a new program has been developed, starting from a template provided by SDK called ”echo server”. The original template implements a TCP server which simply replies to any incoming packet by sending back the same received content. A set of custom command has been defined and the software have been modified in order to be able to decode them from the incoming Ethernet packet and to process. Some implemented commands are listed below:

Table 1. FPGA resource utilization of the implemented firmware design; only main components and primitives are listed

Resource Type	Utilization (%)
Slice LUTs	11.07
Slice Registers	8.09
Slice	19.94
LUT as Logic	10.07
LUT as Memory	3.03
LUT Flip Flop Pairs	5.80
Block RAM Tile	5.00
Bonded IOB	16.50

– read the firmware version;
– configure the custom IP core which receive ADC data;
– configure the ADC and the clock buffer into the Analog development board;
– configure acquisition (form instance, program the number of samples);
– receive sample data.

The client TCP running on a PC has been developed with the ROOT[2] software, designed at CERN. ROOT provides a set of cross-platform C++ libraries with a large amount of functions devoted to Graphical User interface development as well as data processing and visualization.

[2] `https://root.cern.ch`

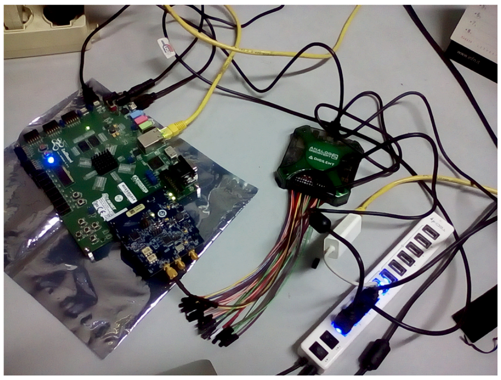

Fig. 9. Picture of the compact testbench. The ZedBoard hosts the AD9434 evaluation board. The Analog Discovery 2 signal generator is also visible.

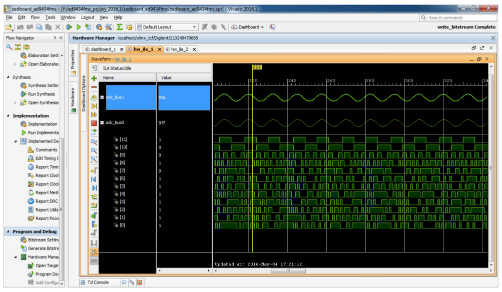

Fig. 10. Screenshot of the Internal Logic Analyzer configured to monitor the 125 MHz 48-bit bus made of 4 consecutive samples. Taking this picture an input sinusoidal signal ad 10 MHz frequency were applied. Debugging capability takes a clear advantage from both digital and analog waveform view.

4. Test, performance and measurements

The testbench used to check the project is very simple: it requires only a high-quality signal generator. Sinusoidal signals are generated at different frequencies and several parameters of the sampled waveform are measured: mean frequency, peak-to-peak amplitude and SINAD are the most interesting ones in order to certify the ADC performance. An additional compact test stand has been realized , with a bandwidth-limited but portable signal generator (see Fig. 9), for tutorial purposes. A 2 analog channels generator from Digilent (called Analog Discovery 2)[3]. It is a 100 MS/s USB oscilloscope, logic analyzer and signal generator with an analog bandwidth of 30 MHz (well below the 250MHz Nyquist frequency of the AD9434 at 500 MS/s). A 10 MHz sinusoidal signal, sampled data by the Integrated Logic Analyzer, is shown in Fig. 10. Since the internal bus is made of 48 bits joining 4 consecutive samples, every group constitutes an effective 125 MS/s time sampling . In the same Figure the waveforms of two of those group are visualized (an analog visualization format is available) after being sampled by the internal debugger. For the second group also the digital representation of each bit out of the 12 is shown. The powerful debugging capability of the Integrated Logic Analyzer is then visible, both for digital and analog signal representation. The same sinusoidal signal is reported as shown by the GUI client (see Fig. 11). Data are acquired through Ethernet and then properly combined in order to visualize the correct 500 MS/s sampled set. Compared with the screenshot from the Integrated Logic Analyzer it is clearly visible that the waveform is sampled at a fourfold rate. Just for completeness, an example of analysis performed with the MATLAB®software on the acquired waveform is shown in Fig. 12. This picture correspond to the measurement of the SINAD over a sampled 140.3 MHz sinusoidal input. About 500 samples of a 10 MHz sinusoidal input are shown.

5. Summary and conclusions

A design implemented on a Xilinx Zynq FPGA has been presented. It has been realized by the Electronic Design Department at the Bologna Division of the INFN, within a currently ongoing project. The final aim is to build a PCB with a single acquisition channel running at 1GS/s , performed by 2 interleaved ADCs. The design presented in this paper is a first version which acquires samples from a single

[3] https://reference.digilentinc.com/reference/instrumentation/analog-discovery-2/reference-manual?redirect=1id=analog_discovery_2:refmanual

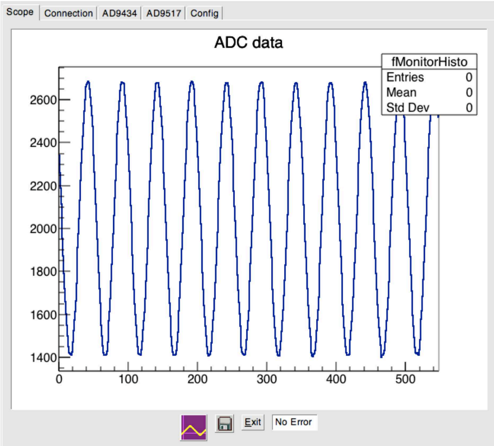

Fig. 11. Screenshot from the monitoring panel of the server GUI application running on the acquisition PC.

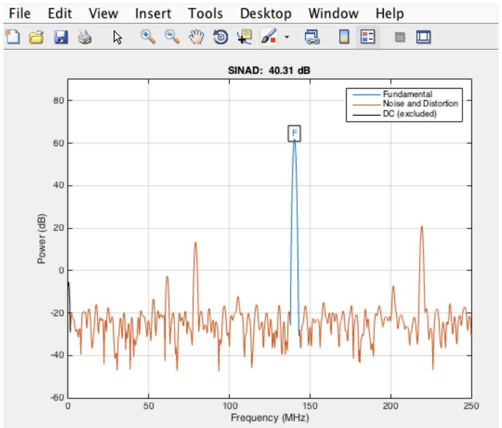

Fig. 12. SINAD calculation from acquired samples of a sinusoidal 140.3 MHz input signal. The data are exported from the client GUI application into MATLAB®.

ADC running at 500 MS/s. It has been developed and tested with commercial boards in order to make experience with the Zynq device as well as to perform preliminary measurements. The Zynq device has been described, together with the implemented firmware and software design. Some highlights are pointed out. Finally, some examples of performed measurement are shown. The experience gained with the Zynq FPGA lead us to take some tentative conclusions about advantages and disadvantages on using such a device. First, it is a very well documented platform with several tutorials and templates available, so that the learning phase is relatively fast. Several development board exists so that prototyping and testing both firmware and software is a fast process as well. The Zynq device allows to integrate standard protocols (like Ethernet) in a straightforward way. In fact, most of the hardware is already available on-chip so that most of the implementation effort is software-based; drivers and high level functionalities are already provided by the development tools. Even though the full Zynq processing performances weren't needed in thi design, we proved to be able to acquire about 2000 consecutive 12 bit-wide samples at 500 MS/s and to send them through Ethernet in a very reliable way. From our experience this device turns out to be very powerful to develop performant testbenches and acquisition systems in a short time.

References

Adamczak, A., et al. 2016, Journal of Instrumentation, 11 , P05007

Analog Devices 2013, AD9434 Datasheet (12-Bit, 370 MSPS/500 MSPS, 1.8 V Analog-to-Digital Converter), `www.analog.com`

Avnet Inc. 2014, ZedBoard (ZynqTM Evaluation and Development) Hardware User's Guide, `www.zedboard.org`

Crockett L. H., et al. 2014, The Zynq Book: Embedded Processing with the ARM Cortex-A9 on the Xilinx Zynq-7000 All Programmable SoC, 1st ed. (Strathclyde Academic Media, Glasgow, UK)

Xilinx 2014, 7 Series FPGAs Configurable Logic Block User Guide, UG474 (v1.7), November 17

Xilinx 2015a, Zynq-7000 All Programmable SoC Technical Reference Manual, UG585 (v1.10), February 23

Xilinx 2015b, Vivado Design Suite AXI Reference Guide, UG1037 (v3.0), June 24

Xilinx 2015c, Zynq-7000 All Programmable SoC (Z-7010, Z-7015, and Z-7020): DC and AC Switching Characteristics, DS187 (v1.17), November 24