International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-7, Issue-6, March 2019

High Speed Event Driven Data Acquisition system based on ZYNQ SOC Architecture

Himanshu Tyagi, Nagendra Gajjar, Suryakant Gupta

Abstract *Acquisition and streaming of signals at high sampling rates at ~100MS/s is a challenging task due to memory and network bottlenecks. At such high rates it is important that data acquisition systems be activated only for a limited duration during an external event which is dictated by the amount of on board memory available for logging of data samples. The platforms for such applications need high speed ADC modules, sufficient on board RAM to log samples and a controller. Although commercial platforms do exist for meeting such requirements most of them are quite expensive. Contemporary system on chip (SOC) based solutions such as Xilinx ZYNQ provides an option of tightly coupled FPGA and ARM processor. The advantage of such architecture is that the FPGA can be available for handling critical time bound tasks and the processor can handle configuration management.The present work focuses on development of event driven high speed acquisition system based on Zynq SOC with user configurable options. For development of the prototype, Red Pitaya board is selected which hosts 2 on board ADC modules with 512MB RAM. The data is acquired at 125 MS/s rate for user defined event durations. Other parameters such as pre and post trigger duration, buffer length are also user configurable. The performance of such systems is also based on the type of file system utilized for storing the data. Hence in the present work details of various file formats used and their effect has been studies. Open source libraries based on Python are used to develop windows based TCP client program with Qt GUI framework for transmission of configuration parameters to Zynq platform and exchange the status of program. The server program is being executed using a C program which communicates to the ARM processor. In this paper the details of the system design, architecture, software flow and analysis of results are mentioned.*

Index Terms**:** *FPGA, ZYNQ, SOC, Data Acquisition, High Speed*

I. INTRODUCTION

Event driven architectures [1] are beneficial for acquiring data when the expected sampling rates are high and critical information is available during the occurrence of event. Event DAQ systems not only provide better file management due to logging of only relevant data but also are power efficient [2]. Event driven data acquisition systems are used at multiple fields due to inherent advantages of such systems. Some of the areas are physics based diagnostics [3], nuclear fusion systems, wireless systems [4] and many other areas. The events can be synchronous and asynchronous. Asynchronous event can occur any time during the time of experiment or observation. Synchronous events have a fixed frequency of operation and acquire data at regular intervals. The data acquired during the event is divided into pre and post trigger samples. The pre trigger samples are important to understand the transient behaviour of signals before the occurrence of event. The data acquired has to be transferred from the acquisition platform to host system for data analysis which is done after the occurrence of the event.

Commercial platforms such as NI R series cards [5], digitizer board from CAEN [6] are available but offer less flexibility and insight of the underlying process and at the same time are quite expensive. Hence for developing such systems SOC based is a good option. Development using such platforms provides the option of configurability as per technical requirements. External triggering can also be integrated with the help of digital logic. The external triggering can be via a TTL signal or level based trigger where the rise or fall of analog signal triggers the acquisition logic.

Another flexibility offered is in terms of software interface. Commercial platforms are programmable by proprietary frameworks such as LabView [7] but for SoC based systems the developers can utilize open source Python [8] scripts in order to program and run the developed systems. In present work we have developed a TCP based client server model where a server program written in C is active on the ZYNQ platform and Python based client software with Qt GUI[9] requests the server.

The adoption of a System on Chip (SoC) based technology exploiting both an ARM based processing unit and a FPGA logic provides the flexibility of a configurable device for real-time operations and as well as the possibility of deploying software components directly on-board. The Red Pitaya board [10] is currently used for the development of the architecture. Overall few works have been done in this area. [11,12] has described the development of FPGA based data acquisition but have not stressed on data logging features, also the sampling rates mentioned are of order of 10 MSps. [13] Describes the development of ZYNQ based data acquisition platform using slow ADC modules. The current work strives to achieve higher and configurable sampling rates in order to have wider applications.

Revised Manuscript Received on 30 March 2019
***** Correspondence Author
Himanshu Tyagi, Dept of ECE, Nirma Unversity, Ahmedabad, India
Nagendra Gajjar, Dept of ECE, Nirma Unversity, Ahmedabad, India
Suryakant Gupta, Institute For Plasma Research Gandhinagar,India

The paper is organized as follows: section 2 introduces the reasons for the choice of the Zynq SoC and summarizes the key features. Section 3 describes the design of data acquisition system based on Red Pitaya board and the software interface details. Section 4 provides the test results of developed system with discussions and section 5 concludes the paper.

II. OVEVIEW OF RED PITAYA HARDWARE AND AXI PROTOCOL

Fig. 1 shows an overview of the RedPitaya board. RedPitaya is low-cost platform with analog signal generation/measurement electronics. The main technical specifications are shown in table I. The Red Pitaya has adual core ARM Cortex A9+ FPGA with 4GB of RAM with the circuitry to handle fast analog inputs, analog outputs, and digital GPIO. Such features make it an attractive platform for a high speed data acquisition platform. It has a Linux interface and can be controlled/accessed by USB and Ethernet via SSH protocol. The user can develop custom logic for the FPGA and interface it with the ARM controller as per application requirement. The user can take benefit of the SOC architecture and utilize the available analog front end resources for various engineering applications. The most important aspect of SOC development is utilizing AXI bus interface. The following section describes the AXI protocol in brief.

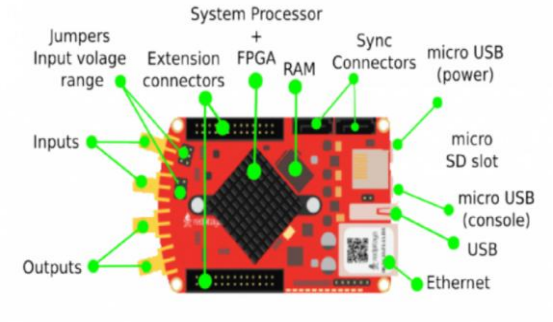

Fig 1: Red Pitaya

A AXI protocol

AXI stands for advanced eXtensible Interface, and the current version is AXI4. AXI is part of the ARM AMBA R 3.0 open standard. Utilizing this standard multiple third party developers have created IP blocks for integration and development.

AMBA standard, originally developed by ARM for microcontroller applications has been revised and extended, and it is now described by ARM as “the de facto standard for on-chip communication” [14]. The focus is now on System-on Chip (SoC), including SoCs based on FPGAs or, in the case of Zynq, a device which includes FPGA fabric.

As of now, three versions of AXI4 [15,16] are available with different properties as described below. For a particular engineering application, the choice of AXI bus protocol is important.

- AXI4: Generally used for memory-mapped links, providing the highest
- AXI4-Lite: A simplified link supporting only one data transfer per connection without bursts.
- AXI4-Stream: Used for high-speed streaming data since it supports burst transfers of unrestricted size. It works without any address mechanism; this bus type is best suited to direct data flow between source and destination.

For present case, the best choice is to use the AXI4- Stream protocol for data acquisition system development due to its inherent capacity for high-speed data transmission.

III. DESIGN OF DATA ACQUISITION SYSTEM

The most critical components of any data acquisition system are the ADC, memory available and the instrumentation bus connecting the ADC, RAM via a controller. The ADC available is LTC2145-14 [17] for analog to digital conversion. The IP blocks are written to interface these with the FPGA and they read data via a 14-bit bus. In present case the controller is the Zynq SOC device where the ADC is interfaced to the PL fabric using on AXI standard. The data from ADC is connected to the memory of the board access to which is controlled via an ARM based PS of ZYNQ.

The top level design can be created using the IP integrator [18] feature newly introduced in Xilinx Vivado [19] platform. This design strategy is high level in the sense that various blocks in the design are integrated on the basis of the interfaces between them. The data stream is transferred from one block to another based on master-slave interface. Finally all the blocks are compiled to create a top level system_wrapper.vhd file which contains the interface information.

Fig 2 describes the top level block diagram of the project where the PL and PS logics are described. The data stream from ADC is connected to a CIC filter which decimates the samples. This is done to reduce the sampling rates from original 125 MS/s by a factor of 5, 10 or 20. This helps in increasing the time duration for acquisition as the overall sampling rate is decreased.

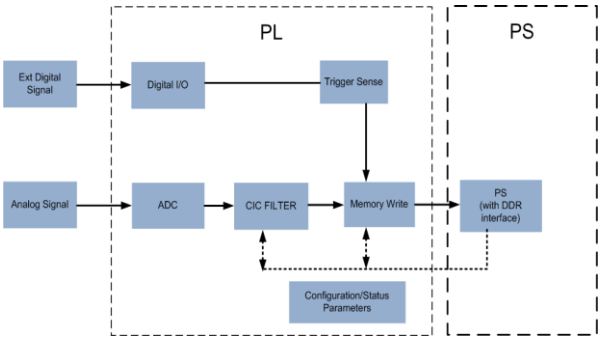

Fig 2: Top level design in ZYNQ

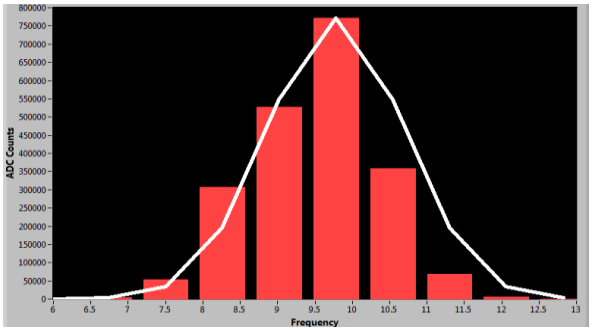

Figure 3: Histogram analysis of ADC

The trigger sense block takes input from GPIO block and raises a flag is the TTL signal is available on the digital connector of RP board. The trigger sense block activates the data transfer from CIC filter to the RAM. This data is available on the RAM until the next trigger signal on digital connector. The decimation factor 0 can be selected for bypassing the CIC filters. This dumps the data at highest sampling rates in the AXI stream interface.

A. IP block design

The IP block design is similar to the block diagram as shown in Fig 2. Some of the IP blocks have been derived from open source IPs available for Red Pitaya board [20,21] and have been modified as per application requirement.

B. Testing of the ADC

One of the most critical aspects of the acquisition system is the ADC which is the heart of the system and governs the limits of the developed system. The ADC used in the system is LTC 2145 supporting 14 bit. In order to understand the functioning of the ADC some benchmarking was done based on standard available methods. The most used method for characterizing the ADC is histogram technique [22]. Here a number of samples are captured and analyzed for accuracy and identifying the noise properties. The inputs of ADC were grounded to the PCB reference and data was acquired at the highest rate of 125 MSps. In order to obtain proper characteristics 2M samples were acquired and plotted for histogram analysis. Fig 3 shows that the mean is ~ 10 counts only which shows that the ADC is fairly accurate for small voltage signals given that the maximum counts is 16383 due to 14 bit resolution. The data considered is average of 10 such measurements.

C. Software Design

The data from ADC is available at the RAM of the ZYNQ board in a circular buffer fashion. The data is written via the PL-PS interface in memory. The data at these memory addresses is accessed by using a C program based server program. The server program has following functions:

- the configuration parameters from client software
- Provide the status to client program
- Read and acquire the data from RAM area

Once the bit file is loaded the server can be started. The server detects the client connection and waits for the user inputs. After the required parameters the data is logged in a file after a valid trigger is received which is then transferred to host system and displayed at client side.

In order to develop the client side program the language selected was Python v2.7. The main reasons for not developing a C based client program was the unavailability of Qt libraries with C. Python being an open source platform is rich in multiple computational libraries and has a wide user support. The environment selected for developing the application is Python(X, Y) [23] since it has integrated advanced libraries such as numpy, plotlib and Qt. Qt was used to develop GUI for the application with PyQT interfacing libraries. The development of a GUI is important since it gives the user an understanding of the run time status of the application. Also it gives the flexibility to transfer configuration parameter for the triggered acquisition such as total buffer size, decimation factor for CIC filters and percentage of pre and post trigger size. After obtaining the configuration parameters from client the server will be active and wait for trigger from digital extension. After the trigger, data is logged into the file. The file name is incremented with every new trigger arrival. The file format selected for logging the data was in both the text mode and binary mode. The result on file formats is described in next section.

IV. RESULTS AND DISCUSSION

In order to have a client interface, Qt based GUI was created with Python 2.7. The fig 4 describes the details where the user can enter the IP address of the server with details of sampling rate and Post trigger percentage. The maximum available buffer is 8 MS per channel. Depending upon the decimation factor used, the time duration upto which the data can be acquired is determined. For instance at sampling rate of 125 MSps , the event duration is 64 ms and at 10 MSps the event duration will be 800 ms . The use of CIC filters provide user with an option of configurable sampling rates thus increasing the pulse lengths of data acquisition. Fig 4 provides a sample plot of the data which is acquired and transferred to the host system. The signal is sine wave of 10 MHz which can be reconstructed at 125 MSps and the trigger signal is in between of the acquired signal since the post trigger percentage was kept as 50%.

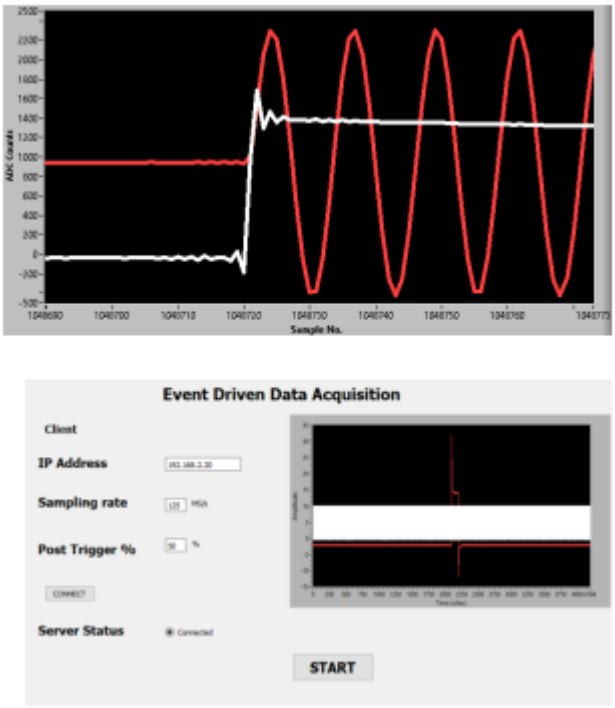

Figure 4: User Interface in Python

In order to test the system accuracy the signal and trigger were fed into the board in triggered mode though a function generator. Fig 5 shows that there is no delay between signal and trigger after post trigger measurements. This helps in benchmarking the system.

A. File system

The file format is important since it determines the process of logging the data and viewing it for analysis. There are 2 options for: binary and text files. Both have their own advantages and disadvantages in terms of footprint and readability. Although binary files have smaller size as compared to other file formats, they are not in human readable format. Text files on other hand are readable but the file size is comparatively large. In C language there are 2 dominant methods for file logging, fprintf() and fwrite(). The later supports writing entire chunk of data together to a file location and the former supports only single data logging to file.

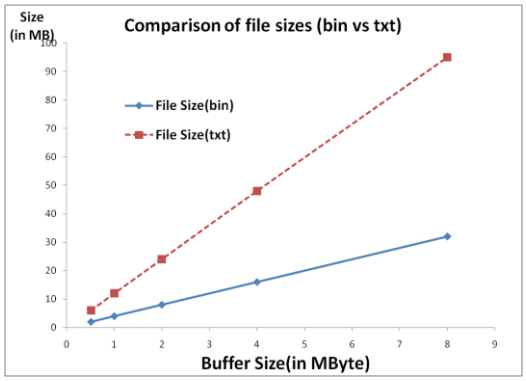

Figure 5: Testing of event DAQ

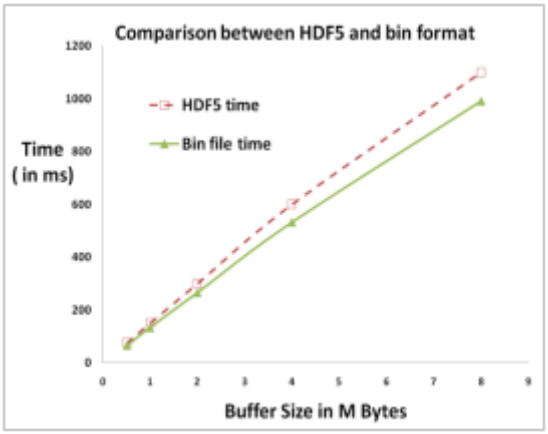

Fig 6: File size comparison

In order to assess the performance both the file formats and programming methods were tried based on variable buffer sizes at 125 MSps sampling rate. Fig 6 and 7 depicts the comparison of the time taken to log the file at server side and the foot print of the file
for both text and binary files. This clearly shows the benefit of binary file format.

B. HDF5 format

HDF5 (Hierarchical Description file v5) [24] is a widely used format for storing scientific data used through labs. In present work the data was also logged using HDF5 file format. One the benefits is the grouping of data based on hierarchical nature. Multiple data files can be grouped in a single file which can be analyzed by user directly in HDF5 viewer. Fig 8 shows the comparison between bin and HDF5 file formats in terms of time required to log the data which shows that is almost linear. This also shows that HDF5 is easy to use and also comparative to binary file format in terms of performance.

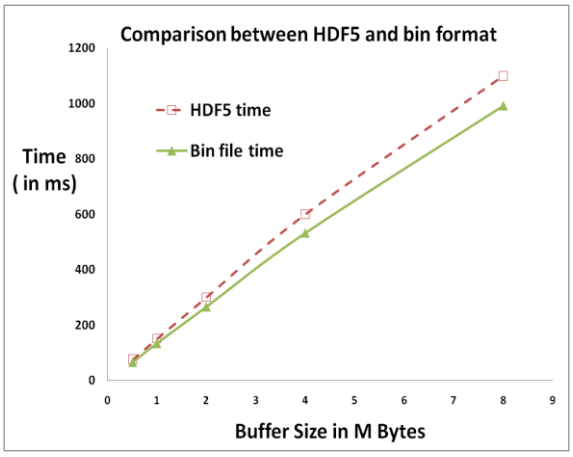

Fig 7: File logging time comparison

V. CONCLUSION

The present work has been carried out in order to understand and benchmark the applicability of ZYNQ platform for high speed data acquisition requirements.

The developed system is not only user configurable but also tries to match the performance of commercially available platforms at a fractional cost. Development of open source based client interface has increased the configurability to the developed application. The current work demonstrates the capabilities of ZYNQ platform which has brought new paradigms in field of programmable hardware technology.
The current work also explores HDF5 file format for usage in data logging

REFERENCES

1. YannisTsividis, Event-Driven Data Acquisition and Digital Signal Processing—A Tutorial, in IEEE Transactions on Circuits and Systems II, Volume 57, Issue 8,p-p 577-581, Aug 2010.
2. Tsividis Y. Event-driven, continuous-time ADCs and DSPs for adapting power dissipation to signal activity Iscas 2010 - 2010 Ieee International Symposium On Circuits and Systems: Nano-Bio Circuit Fabrics and Systems. 3581-3584
3. Wojenski1a, K. Pozniaka, G. Kasprowicza, W. Zabolotnya, A. Byszuka,P. Zienkiewicza, M. Chernyshovab, T. Czarskib, Fast data acquisition measurement system for plasma diagnostics using GEM detectors, presented at First EPs Conference on Plasma Diagnostics - 1st ECPD, April 2015
4. Alan Richard Wilson, Event Triggered Analog Data Acquisition Using the Exponential Moving Average, in IEEE Sensors journal, Vol 14, Issue 6,p-p 2048-2055, June 2014
5. NI R series cards.[Online]. Available: http://sine.ni.com/nips/cds/view/p/lang/en/nid/13897
6. CAEN Digitizers. [Online]. Available: http://www.caen.it/csite/CaenProfList.jsp?parent=95&Type=WOCateg&prodsupp=home
7. Labview.[Online]. Available: http://www.ni.com/en-in/shop/labview/labview-details.html
8. Python 2.7. [Online]. Available: https://www.python.org/download/releases/2.7/
9. QT [Online]. Available: https://www.qt.io/
10. RedPitaya, RedPitaya Open Source Instrument.[Online]. Available:http://www.redpitaya.com/.
11. Aboli Audumbar Khedkar, R.H.Khade, High speed FPGA-based data acquisition system, Microprocessors and Microsystems, Volume 49,p-p 87-94, 2017
12. Venkatraman Kandadai, Moorthi Sridharan, Selvan Manickavasagam Parvathy,Raja Pitchaimuthu,A comprehensive embedded solution for data acquisition and communication using FPGA, Journal of Applied Research and Technology, Vol 15,p-p 45-53, 2017
13. Jiawu Fan, Design of Extensible Data Acquisition System Based on the Zynq Platform , presented at 2nd International Workshop on Materials Engineering and Computer Sciences (IWMECS 2015)
14. ARM, AMBA Open Specification. [Online]. Available: http://www.arm.com/products/system-ip/amba/amba-open-specifications.php
15. ARM, "AMBA AXI and ACE Protocol Specification: AXI3, AXI4,and AXI-Lite, ACE and ACE-Lite." [Online]. Available: http: //www.arm.com/products/system-ip/amba/
16. AMBA 4 AXI4-Stream Protocol Specification. [Online].Available: http://www.arm.com/products/system-ip/amba/
17. ADC LTC, http://www.analog.com/en/products/ltc2145-14.html
18. IP integrator [Online].Available: https://www.xilinx.com/products/design-tools/vivado/quicktake-videos/using-ip-integrator.html
19. Vivado[Online]. Available: https://www.xilinx.com/products/design-tools/vivado.html
20. Anton Potonik. Red pitaya fpga project 4. http://antonpotocnik.com/?p=519284
21. http://pavel-demin.github.io/red-pitaya-notes/
22. M.F Wagdy , "Determining ADC effective number of bits via histogram testing", IEEE Transactions on Instrumentation and measurement (Vol 40 ,Issue 4, 1991, Page 770-772
23. Python x,y[Online]. Available: https://python-xy.github.io/
24. HDF5 [Online], Available https://www.hdfgroup.org/solutions/hdf5/