Volume 4, Issue 1, January 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Review on FPGA Based Pulse Processing System

Nishtha Vyas, Abhishek Shah
VLSI & Embedded System Design,
GTU PG School, India

Abstract—In process to create a more suitable and accurate pulse processing system FPGA system is developed. Analog system takes wide space for pulse processing. FPGA technology has replaced this disadvantage. It has become an extremely cost-effective means of off- loading computationally intensive digital signal processing algorithms to improve overall system performance. The digital filter implementation in FPGA, utilizing the dedicated hardware resources can effectively achieve application-specific integrated circuit (ASIC)-like performance while reducing development time cost and risks. A low-pass filter can be implementing on FPGA. MATLAB tool with FPGA system is best way to design digital system.

Keywords— FPGA, MATLAB, FDATool, LABVIEW,iMPACT

I. INTRODUCTION

For a long time, the close to optimal filtering, that is, pulse processing, was only possible to perform with analog components. For example, classical spectrometry amplifiers with Gaussian pulse shaping have been used for more than 40 years. Due to the development of fast analog to digital converters (ADCs), field programmable gate arrays (FPGAs), and digital signal processors (DSPs) during the last few decades, it has become possible to digitize pulses even after preamplifier or phototube, and process them in a real time. The FPGA can handle the readout, trigger decisions, and simple to medium levels of complexity of signal processing. Dependent on the input-output and processing requirements, the FPGA can be replaced or augmented by DSPs for calculations that are more complex. Therefore, the digital electronics, which was limited to the control of the acquisition process and data storage, has become feasible for signal processing as well. This immediately opened possibilities to design instruments with relatively high component density at a reasonable cost per channel. Such instruments could implement many analog processing functions like pulse discrimination, pulse amplitude filtering, pile-up correction, and base-line restoration.

II. HISTORY

Electrical signals are digitized earlier therefore less analog components are used which should improve noise immunity and temperature stability giving potentially better resolution. Digital filter design techniques provide a high degree of freedom, which may result in better noise suppression and better resolution (optimal or close to optimal filtering). Usually the pulses from the detector are not coming uniform in time and the time of arrival cannot be predicted. In the case of high counting rates, two or more pulses can overlap with high probability (pile-up). Using analog circuits, it is to very difficult to separate them, and one or more pulses should be rejected which decreases number of processed events, that is, system throughput. In order to keep the throughput high, one can decrease pulse width, but it will compromise energy resolution since pulse shape will be far more from optimal. Since digital pileup rejection is more efficient, it will result in a higher throughput. Higher density and lower supply voltage integrated circuits reduce size and improves portability of nuclear spectrometry systems. This may be very important for in situ applications such as space research, mining, cultural heritage, etc. In laboratory conditions, and for experiments with a large number of detectors, it may reduce the number of cables, crates, and costs. State of the art equipment for nuclear spectrometry can be as small as a desktop size instrument, or as large as an accelerator. They are usually expensive and technically complex.

Data acquisition using digital electronics differs from event-driven acquisition using analog electronics and traditional ADCs. The input signal is continuously sampled and the quantity of interest is extracted for each pulse. Event processing may be internally or externally triggered. In the first case, processing is initiated whenever the signal meets certain criteria, which can be as simple as a threshold. In the second case, many modules can be synchronized among themselves. Every packet of information can be time stamped with an internal clock value, formatted, and put into an output buffer for a delayed readout. The data buffers are then retrieved by a main data acquisition computer.

The designing of an FIR filter in VHDL with MATLAB (for the generation of coefficients of filter) and programming it onto an FPGA is good technology of digital system. Implementation of project onto an FPGA(including hardware and software parts) VHDL, MATLAB and basic digital filter concepts are used. Next isLABVIEW tool used to display output value. Fig 1 shows analog signal processing in normal DSP processing system.

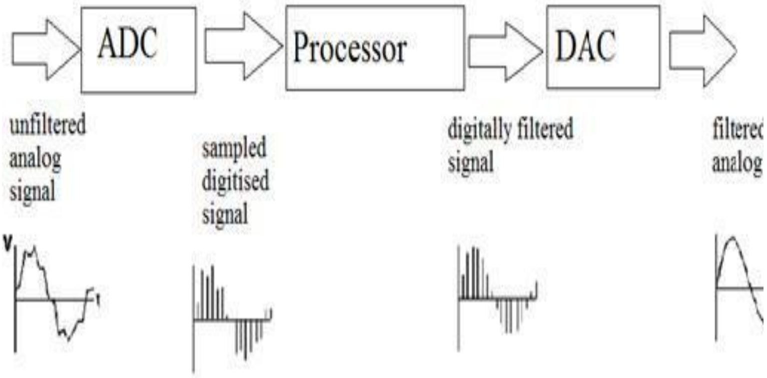

Fig 1 Analog Signal Processing

III. BLOCK DIAGRAM

The various functional blocks include Pulsar, Pre-amplifier, ADC , FPGA ,and LABVIEW module are shown in Fig 2.

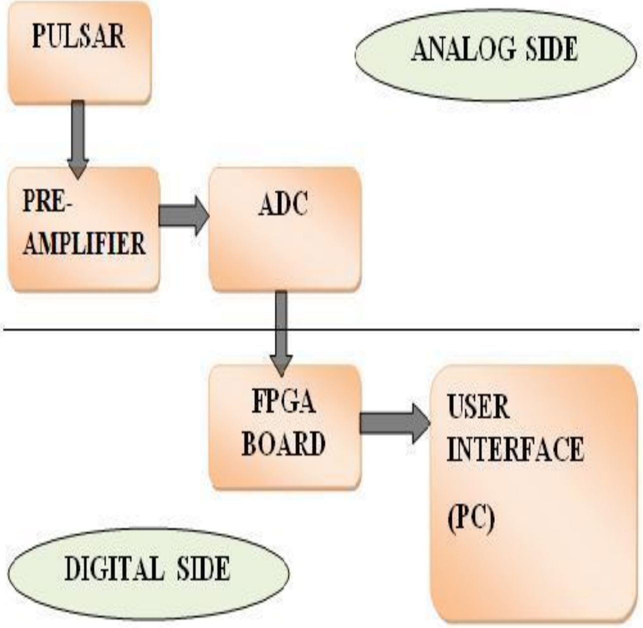

Fig 2 Block Diagram of Smart Energy Meter

A brief description of Individual blocks:

1) Pulsar: This module is used to provide signal as input to system. There are many types of pulsar available. It can provide signal with desired rise time and fall time and it can adjust signal with high frequency as per its specification.

2) Preamplifier: A preamplifiers takes a very small signal and increases it to a line level voltage. It increases the strength of the input signal.

3) ADC: The analog input coming from previous stage is converted in to digital 12 bit input by using Pmod AD1 analog to digital converter. It is connected to FPGA Spartan 6 microboard from Pmod space given on board.

4) FPGA: Field programmable Gate Arrays (FPGAs) are pre-fabricated silicon devices that can be electrically programmed in the field to become almost any kind of digital circuit or system. For low to medium volume productions, FPGAs provide cheaper solution and faster time to market as compared to Application Specific Integrated Circuits (ASIC) which normally require a lot of resources in terms of time and money to obtain first device. FPGA presents a compelling alternative for digital system implementation due to their less time to market and low volume cost. The re- configurability of an FPGA is on an underlying programming technology, which can cause a change in behavior of a pre- fabricated chip after its fabrication.

5) LABVIEW: It is a platform and development environment for a visual programming language from National Instruments. LabVIEW is commonly used for data acquisition, instrument control, and industrial automation on a variety of platforms including Microsoft Windows, various flavors of UNIX, Linux. LabVIEW is a graphical program development application developed by National Instruments in 1986 to integrate engineering tasks like:

o Interfacing computers with the instruments.
o Collecting, storing, analysing, transmitting measured data.
o Developing program in a graphical environment.

o Providing an effective user interface.

6) i*MPACT* : This software is used to interface FPGA with digital coding. Xilinx generates bitstream file (.bit) and that program file connects is being loaded to iMPACT. It will interface all hardware components are linked with digital code. As shown in figure bit file is being loaded successfully and FPGA is now connected with system.

IV. INTERFACING ADC AND FPGA

The ADC converts an analog input signal ranging from 0-3.3 volts to a 12-bit digital value in the range 0 to 4095. The ADC has two simultaneous A/D conversion channels, each with a 12-bit converter and filter. Each channel can sample a separate stream of analog signals. The ADC can also convert a single stream of analog signals using only one channel. The filters limit the analog signal bandwidth to a frequency range suitable to the sample rate of the converter. The ADC uses the serial bus standard to send converted data to the host system. The serial bus can run at up to 20 MHz. The ADC has a 6-pin header and a 6-pin connector for easy connection to any other system. The AD1 can be powered by voltage from either a Digilent system board or an outside device. Damage can result if power is supplied from both sources or if the outside device supplies more than 3V.

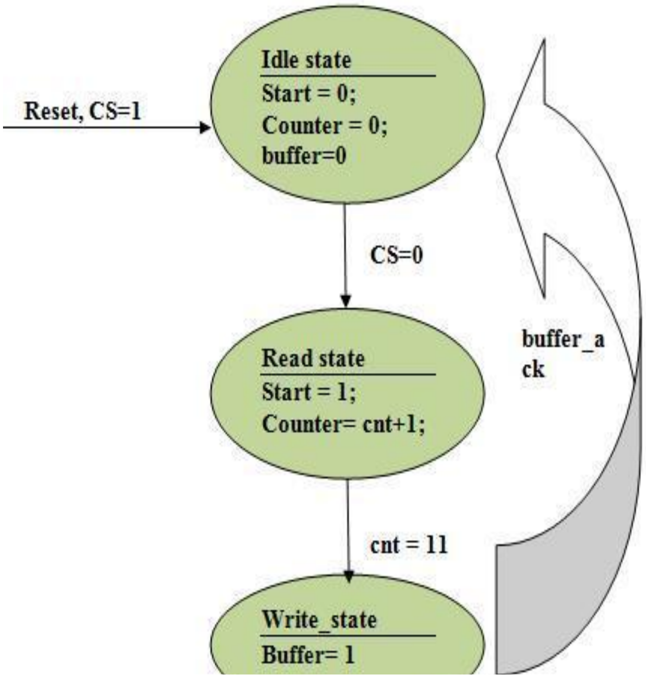

Fig 3 State Diagram Of ADC operation

To interface ADC PmodAD1 module to FPGA Spartan-6. Hardware of ADC can be configured using VHDL coding done in Xilinx13.4. Here FPGA Provides 100MHz clock whereas ADC requires only 50MHz clock to start operation. That 50MHz will be divided into 12.5MHz as it is required for serial data input. This operation is done by clock divider coding. ADC will give Output 12bit serial data, now that data is to be converted into parallel load to be fetched from FPGA to user interface.

The data coming from ADC-FPGA will be transmitted to PC from USB of FPGA. Now next task is RS232 communication from FPGA to User interface. The Spartan-6 FPGA LX9 Micro Board implements a Silicon Labs CP2102 device that provides a USB-to-UART bridge. The USB physical interface is brought out on a USB micro-B connector labeled “J3.” Power supplied by the USB host via connector J3 (+5V_USB_B) is used in conjunction with power from the other USB port, through diodes D13 and D16 to power the S6LX9 board. RS232 communication is done by connecting T7 and R7 to related pins of digital code.

V. FILTER DESIGN

FIR filters are digital filters with finite impulse response. They are also known as non- recursive digital filters as they do not have the feedback (a recursive part of a filter) [3]. Finite Impulse Response (FIR) filters are defined by scaled and time-delayed versions of the filter input signal only, as given by the following difference equation:

$y[n] = b_0\ x[n] + b_1\ x[n-1] + b_2\ x[n-2]...,\quad n=0,1,2$ ….. (1)

Where the input and output y[n] =0 for n < 0. An FIR filter can be represented by a block diagram as shown in Fig 4.

The z^{-1} terms represent unit delays while this representation for a delay element is common and widely accepted in the signal processing community, the specification of delay in terms of powers of z is a z-domain characterization while the block diagram itself is a time- domain representation.

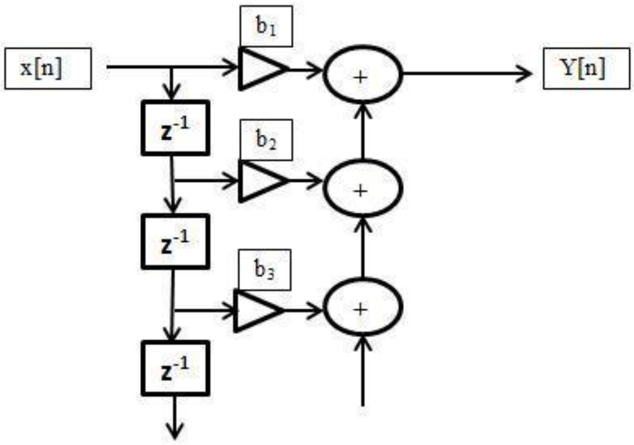

Fig 4 FIR Block Diagram

The Filter specifications includes
(1) Signal characteristics
(2) The characteristics of filter
(3) The implementation technique
Although the above specification is application dependent it will be helpful to devote some time on the characteristics of the filter. The characteristics of digital filters are often in specified in the frequency domain. For frequency selective filters, such as low-pass and band- pass filters, the specifications are often in the form of tolerance. Fig 5 shows the magnitude frequency response specifications for a low-pass filter. In the pass-band, the magnitude response has a peak deviation of Ap and in the stop-band; it has a maximum deviation of As. The width of transition band determines how sharp the filter is. The magnitude response decreases monotonically from the pass-band to stop- band in this region.
The following are key parameters of interest:
o A_p peak pass-band deviation(or ripples)
o A_s stop- band deviation.
o F_{st} stop- band edge frequency.
o F_p pass- band edge frequency.
o F_s sampling frequency.
The edge frequencies are often given n the normalized form, that is as the fraction of the sampling frequency (f/Fs). Pass-band and stop-band deviation may be expressed in decibels. When they specify the pass-band ripples and minimum stop-band attenuation respectively. Thus the minimum stop-band attenuation, As and the eak pass-band ripple,
o Ap, in decibels are given as
o As (stop-band attenuation) = -20log10 as
o Ap (pass-band ripple) = 20 log10 (1+ap)
The difference between fs and fp gives the transition width of the filter. Another important parameter is the filter length, N, which defines the number of filter.

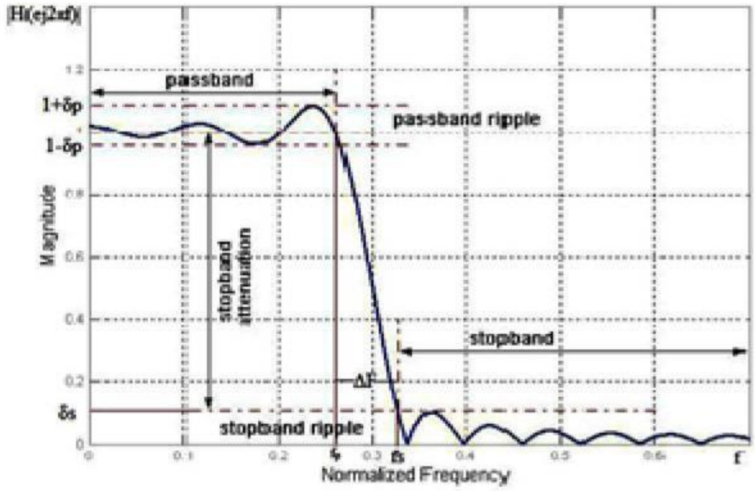

Fig 5 Magnitude frequency response specifications for a low-pass filter

The coefficient and input sinusoidal signal is generated in the MATLAB and it converted into hexadecimal values by digitization. The hexadecimal values of coefficient and signal are proceeded to obtain the desire output through the multiplication and addition is done with IP cores which is inbuilt in Xilinx 13.4 software. Fig 6 shows the design flow of the entire process of FIR filter implementation on FPGA through VHDL coding done in Xilinx ISE design suit 13.4 versions.

To design coefficient of filter the process can be done using MATLAb. Matlab provides the method to generate .COE file which decides the response of filter and order of the filter. That file can be exported to Xilinx ip core. It will design filter according to that file. The whole process is show in next figure.

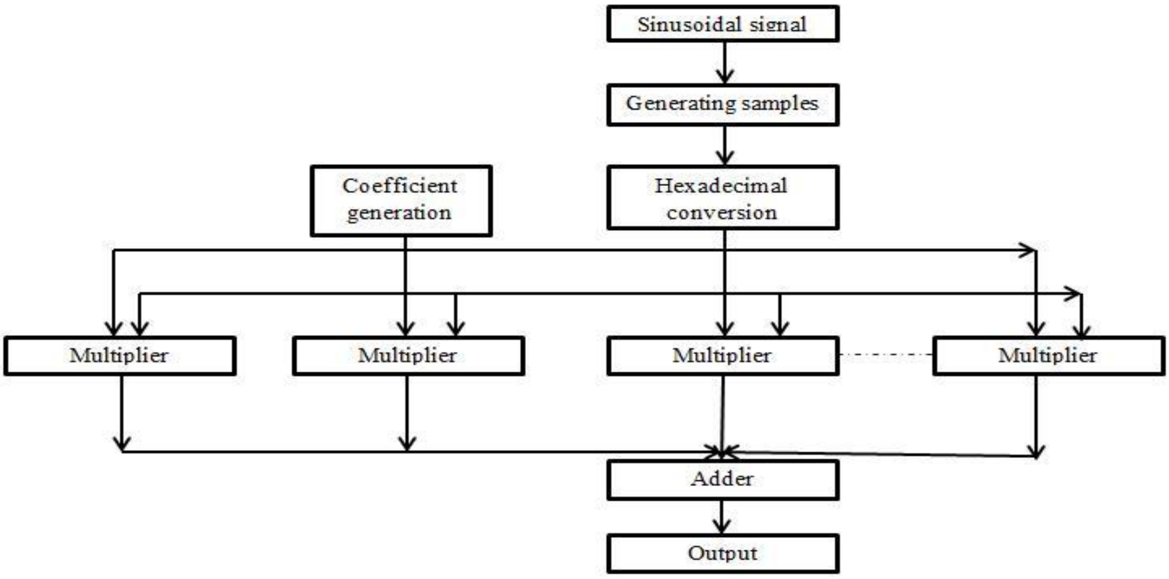

Fig 6 Flow of Filter Implementation

1. Determine a desired response or a set of desired response.
2. Select a class of filter (in this case filter is FIR) for approximating the desired response.
3. Establish a criterion for the response of a filter in the selected class compared to the desired response.
4. Synthesize the filter using a proper structure and a proper implementation form.
5. Analyse the filter performance.

VI. CONCLUSIONS

The filters are widely used in signal processing and can be implemented using programmable digital processors. Realization of large order filters the speed, cost, and flexibility is affected because of complex computations and the development of FIR filters on FPGAs enhanced speed of signal processing. This is due to the fact that the hardware implementation of a lot of multipliers can be done on FPGA which are limited in case of programmable digital processors. This paper mainly describes the design method of filter which is based on FPGA, Xilinx tools and MATLAB. By using these tools time required to get desired results has become less. VHDL has been used to enter hardware description. To test the correctness of the design the observed output is compared with the calculated output results from MATLAB implementation that confirms the effectiveness of the design. VHDL codes have been written, synthesized, mapped then successfully configured and prototyped. Filter designed fully complies with design requirements.

REFERENCES

[1] Ray Goslin, Dr. San Jose, CA 95124 "A Guide to Using Field Programmable Gate Arrays (FPGAs) for Application-Specific Digital Signal Processing Performance Gregory Digital Signal Processing" Program Manager Xilinx, Inc. 2100 Logic .

[2] WANG Jian-xing,YAO Qi-guo**.** Design of Chebyshev Low-pass Filter Based on MATLAB [J], Journal of Xinxiang University(Natural Science Edition),, 2012.2

[3] Chao Cheng and Keshab K Parhi. Low-Cost Parallel FIR Filter Structures With 2-Stage Parallelism [J].IEEE Transactions on Circuits and Systems I: Regular,2007,54(2):280~290.

[4] Hu Guang-shu. Digital signal processing-theory,algorithm and realizes[M]. 2nd ed.Beijing: Tsinghua University Press,2003:296~307.

[5] Chun Hok Ho,Chi Wai Yu and Leong P. Floating-Point FPGA: Architecture and Modeling IEEE Transactions on Very Large Scale Integration Systems, 2008,17(12): 1709~1718.

[6] D. E. Borth, I. A. Gerson, J. R. Haug, and C. D. Thompson "A flexible adaptive FIR filter VLSI IC". IEEE Journ. Select. Areas Commun. SAC-6(3):494–503

[7] Lang, "pole radius constraint," IEEE Trans. Signal Processing, vol. 48,Nov. 2011.

[8] CHEN SH J, GUO J J. Design and implementation of raised cosine roll-off FIR filter based on

FPGA[J].Telecom Power Technologies.2007, 24(2):19-21

[9] CHENG X P, QU B, LU G. An application of immune algorithm in FIR filter design[C].Nanjing: Proceedings of the 2003 International Conference on Neural Networks and Signal Processing,2003(1): 473-475.

[10] EVEN G, PAUL W J. On the design of IEEE compliant floating point units [J].IEEE Trans on Computers.2000,49(5):398-413.