ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 4, October 2015

Digital Filter Design Using FPGA

Suvadip Roy, L. Srivani, D. Thirugnana Murthy
Atomic Energy Regulatory Board, Indira Gandhi Centre for Atomic Research

Abstract - Nowadays Digital Filters are replacing Analog Filters which are used widely in front end Electronics to remove the unwanted component from the signal and increase the signal to noise ratio. Their widespread popularity is basically because of a set of programmed coefficients derived from the analog filter specifications which can in whole control the filter operation and unlike analog filters the value of bulky capacitors and inductors do not affect the filter operability. Initially digital filters were implemented on PDSP and ASIC. But due to the comparatively lower cost and customized design possibilities, FPGA based system have gained much popularity.Moreover most of the FPGAsare reprogrammable hence by programming different filter coefficients the type of filter implemented can be changed as required.This work consists of designing a digital filter from the analog filter specifications and implementing the digital filter on a FPGA development board.

Index Terms **–Finite Impulse Response (FIR), Infinite Impulse Response (IIR), Field Programmable Gate Arrays (FPGA), Multiply and Accumulate (MAC)**

I. INTRODUCTION

Nowadays we live in the digital world where Digital Signal Processing is used widely in several spheres and domains. Filtering of unwanted signals from the desired signals have been used since time immemorial and nowadays filtering in the digital domain is used widely which has got several advantages over filtering in the analog domain like increased accuracy, requirement of lower filter order and availability of various digital filter types and configurations to fit for a specific use. Moreover in case of analog filters with increase in the order of the filter the number of filter components increases which increases it's complexity. Another disadvantage of analog Filters for which digital filters came in the limelight is that the cut off frequency depends on the values of the filter components like resistor and capacitor values, any change in component values which is technically referred to as drift of components due to time or physical parameters like temperature, changes the cut off frequency of the filter. These problems are overcome in digital filters whose characteristics depends on the filter coefficients which once programmed for a specific filter order and type do not change with time.Moreover digital filters are advantageous over the analog counterpart in the sense that they have faster roll off, less transition width and less overshoot in case of time domain operations [1]. For all these, digital filters come handy in several real time applications like in front end of Data Acquisition systems, Image processing applications where it can be used to recover a blurred image, in real time audio recording applications as well as in biomedical engineering fields like ECG signal processing where problems like baseline wander and removal of 50 Hz power supply noise can be handled well with these filters.Moreover analog filters experience ripples in pass band and stop band whereas responses of digital filters is much flatter compared to analog filters. In the nuclear domain also digital filters are being used nowadays in front end electronics.Implementation of the filter hardware using programmable Digital Signal Processors(PDSP) or by using Application Specific Integrated Circuits (ASIC) [2] have become increasingly popular. But due to lower cost and versatility of Field Programmable Gate Array (FPGA) technology, it is mostly preferred over ASIC / PDSP technology for filter implementation and also because of its higher package density and availability of **Multiply and Accumulate** (MAC) units which are the basic building blocks of digital filter hardware. Sometimes FPGA's with available DSP resources are preferred for ease of filter implementation. This paper contains an overview of digital filters and its types, basic design flow for designing digital filters, design and hardware implementation of a Moving Average Filter on aActel's PROSAIC3 FPGA development board and its verification.

II. DIGITAL FILTER BASICS

Digital Filters are mainly classified into two types:

i. Finite Impulse Response or Non recursive Filters
ii. Infinite Impulse Response or Recursive Filters

Finite Impulse Response (FIR) Filters are those whose output depends on present and past input only. They do not contain any feedback from output. The basic mathematical equation governing the output of such a filter is :-

$y[n] = b(0)*x[n] + b(1)*x[n-1] + \ldots\ldots b(N)*x[n-N]$ **(1)**

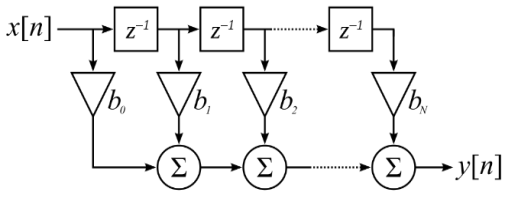

Fig.(1) - Architecture of FIR filters

Fig. 1 shows a basic Transversal realization of a FIR filter [3]. Other form includes Cascade and Lattice structure realizations. The order of a filter depends on the number of delay lines present in the filter architecture. Again b[0],b[1]… b[N] are called the Filter kernel or the Filter coefficients. Some varieties of the FIR filters include Windowed Sinc, Moving average etc. Output of Recursive or Infinite Impulse Response (IIR) Filters

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 4, October 2015

depends on present input, past inputs as well as past outputs. So these filters have feedback from the output.

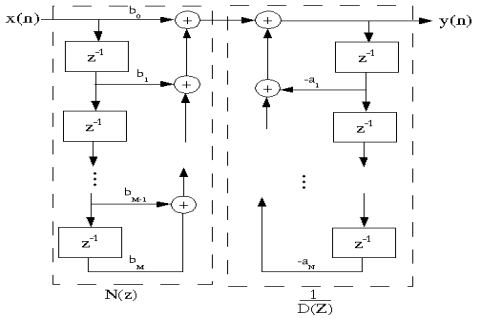

Fig.(2) - Architecture of IIR filters

Here the filter consists of two set of coefficients the numerator and denominator coefficients called the recursion coefficients. Fig.2 shows Direct-Form – I realization of the filter [3]. Other forms include Direct-Form-II, Lattice and Lattice Ladder structure. The basic equation governing the output of such a filter is:-

y[n] = 1/a(0) * {b(0)*x[n] +b(1)*x[n-1] + ……. b(P)*x[n-P] – a(1)* y[n-1] -a(2)*y[n-2] -…….a[Q]*y[n-Q] }**(2)**

Where P is the order of the Feed forward Network and Q is the order of the Feedback network The basic property of these Filters is that their impulse response last till infinite time but generally it can be truncated after some coefficients as the other coefficients seldom have major contribution. Butterworth, Chebyshev Type –I & II and Elliptic filters are a few examples. The basic Filtering operation is called '**Convolution**'. The time domain output depends on the convolution results of the input with the Filter coefficients.

y[n] = x[n] * h[n]

(3)

Where h[n] is the impulse response of the filter popularly called the filter coefficients in case of FIR filters. It can be seen from the basic filtering architecture that the basic operation consists of multiplication and addition which are the two major mathematical operations and are well performed by the **Multiply & Accumulate** units (MAC) [4].

Table (I) – Selection Criteria for Digital Filters

TYPE OF FILTER		**CRITERION FOR SELECTION**
1.	FIR	Stable response,linear phase
2.	IIR	Faster processing, economical.
a.	Butterwroth	Flat response, Higher transition width, higher order.
b.	Chebyshev	Exhibit ripples,faster roll off,lower order.

III. BASIC DESIGN FLOW

For designing any digital filter, a sequence of steps as mentioned in Fig.3 is required:

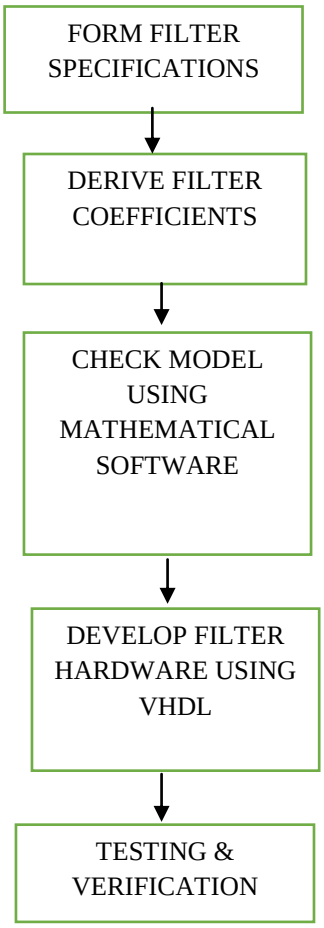

Fig. (3) - Basic Design Flow

A. Forming the Filter Specifications

The first step consists of forming the digital filter specifications from the analog filter specifications like selecting the sampling rate, the type of filter and the order of the filter. Whether a FIR or IIR filter is required depends on the specific area where the filter needs to be applied like if linear phase response is required then a FIR filter should be the ideal choice or else if speed of operation is the factor an IIR filter may be preferred. Moreover the specific category of filter whether a Butterworth or a Chebyshev filter would be required depends on the kind of response needed. Like if equiripple response is required then a higher order Butterworth is the ideal requirement. Whereas if the main requirement is to minimize the filter order then a Chebyshev or an Elliptic filter which may exhibit ripples in the pass band and in the stop band may solve the purpose [5]. The type of filter suitable for specific purpose is covered in Table. I. Once the Type and the order of the filter as well as it's cut off frequency is formulated then the next step becomes finding the filter coefficients specific to the filter to be designed.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 4, October 2015

B. Deriving Filter Coefficients& Checking response using Mathematical Software

One way of finding the filter coefficients is to derive them from the 'Z' domain transfer function. The general equation is as :

$$H(z) = \frac{a_0 + a_1z^{-1} + a_2z^{-2} + \ldots\ldots\ldots\ldots + a_nz^{-n}}{1 + b_1z^{-1} + b_2z^{-2} + \ldots\ldots\ldots + b_nz^{-n}} \quad \textbf{(4)}$$

Where a_0, $a_{1\ldots}$ are the numerator coefficients and b_0, $b_1\ldots$ are the denominator coefficients. For a FIR filter denominator is 1.

Another way of finding the coefficient is by use of software tools for faster calculations.Several mathematical software's are available which can directly find out the filter coefficients given the corresponding filter specifications. The response of filter with the derived coefficients to test inputs by simulation can also be found out to establish the filtering capability. Moreover verificationof unwanted frequency components removed while filtering can also be performedby comparing the frequency spectrum of the input and the output. The basic algorithm is as follows:-

a. Give the input specifications.
b. Derive the filter coefficients.
c. Give a test signal in time domain to the filter having the coefficients as found in the above step
d. Observe the response in time domain for the filter to the test input.
e. Observe the frequency spectrum of the input and filtered signal to check for any unwanted component.[6]

C. Designing Filter Hardware using VHDL

The design entry and simulations are carried out using Actel'sLibero IDE and the experiments are performed using the FPGA development board.The entire project is created using Actel'sLibero IDE as the FPGA used for programming is an Actel designed PROSAIC3 FPGA. The basic algorithm implemented using HDL for FIR filters is as follows:-

a. Declare an array of Integers to store Input sample values.
b. Declare the required library functions.
c. Declare the filter entity consisting of Input ports for input signal and clock signal and output port for output filtered signal.
d. In the filter architecture convert the input data from logic vector to integer for calculation flexibility.
e. Truncate the Floating point filters coefficients into fixed point like if coefficient is 0.129 then it should be like 129/1024 so accuracy of coefficients is around 97%.
f. Implement convolution operation by repeated multiplication and addition at each clock's rising edge.
g. Shift the input signal one step to the left of each cell of array at each clock instant.
h. Convert the result from integer to logic vector.
i. Give real time output to the output port.

The above algorithm gives the basic steps for writing the VHDL code. In step (e) truncation is required because VHDL only allows multiplication of integers and division by any integer in the power of 2 because in general division means left shift of bits. [7] Once the code is ready, simulation has to be performed using the test bench by giving specific input combinations and checking the Pre Synthesis, Post Synthesis and Post Layout Simulation results using the same test bench. If the results are as predictedit proves that the logic have been implemented correctly. The next step in that case is to create the programming file and burn it into the FPGA and check the results else reprogramming has to be attempted. [8]

D. Testing & Verification

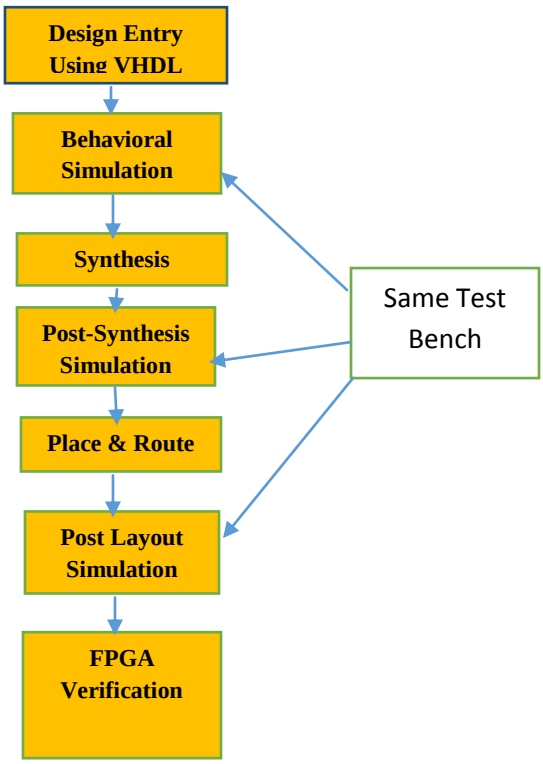

Fig. (4)- VLSI Design Flow

Fig. (4) Shows the VLSI design flow model in which the steps including Synthesis of the VHDL code into Gate level net list, Placing and routing the gates to produce the device programmable bit steam was done in the previous step, the next step included generating the programming bit file and burning it on to the FPGA. For testing a student development kit was used and verified that the hardware was functioning properly by giving proper digitalized test inputs and checking the output pattern. Ideal testing platform would be to inject an analog signal with noise riding on it, digitalizing the signal with an analog to digital converter (ADC), filtering the sequence

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 4, October 2015

by passing it through the filter hardware implemented on the FPGA, sending back the filtered signal to a digital to analog converter (DAC) for reconstruction and then comparing the initial input and the filtered output but in this case as the FPGA kit did not have an inbuilt ADC, so the input had to be digitalized and the digitalized samples were stored in specific location in the FPGA which would be fetched by the filter at appropriate clock edges and filtering operation would be implemented on the samples.

IV. DESIGN OF A MOVING AVERAGE FILTER

Out of the several filters designed by the generalized steps mentioned above one unique time domain filter that was designed was a Moving Average filter which is used to filter out the time domain disturbances sitting on the signal. It is a FIR filter and a 6^{th} order Moving Average filter was designed. The Input output relationship is given by:

$$y[n]=\{x[n]+x[n-1]+x[n-2]+x[n-3]+x[n-4]+x[n-5]+x[n-6]\} / 7$$

So the filter impulse response is given by:

$$h[n]=\{0.142,0.142,0.142,0.142,0.142,0.142,0.142\}$$

The response of the filter to test inputs were seen by using a mathematical software by generating noisy simulated inputs and checking the output after passing through the filter with the above coefficients. Fig. (5) shows the simulated input waveform, whereas Fig. (6) gives the output after filtering during simulation.

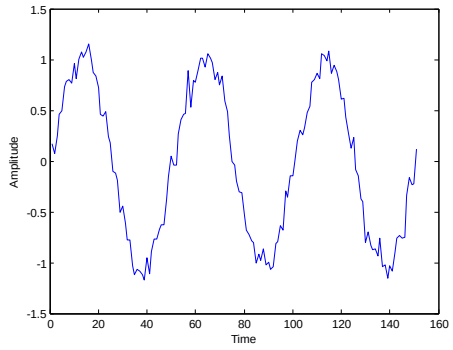

Fig. (5) –Noisy Input to the Filter

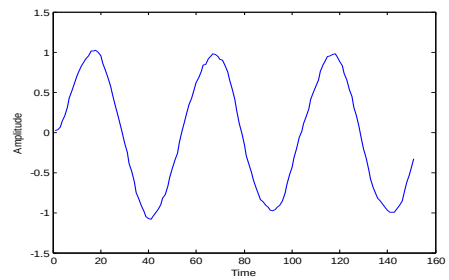

Fig. (6) –Filtered Output

The Seven point moving average filter with the above filter coefficients was designed using HDL and a test-bench was written to verify the design during Behavioral Simulation, Post-synthesis Simulation & Post-layout simulation. Finally the generated bit stream after successful Post Layout simulation step was programmed into the PROSAIC3 FPGA (Device- M1A3P1000 , Package – 484 FBGA) using a JTAG interface . Response of the filter was checked and found to be appropriate.

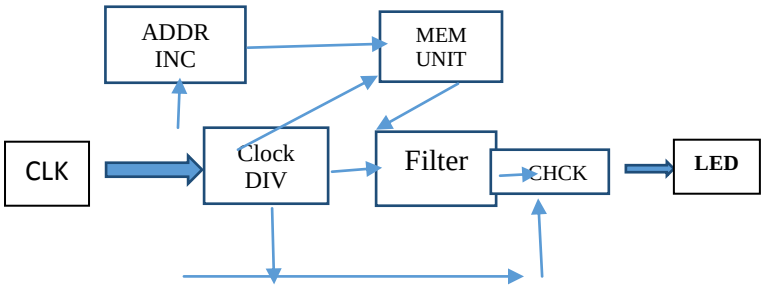

Fig. (7) –Block Diagram of FPGA Internal Configuration

Fig. (7) gives the internal modules designed within the FPGA hardware. The actual hardware implemented in the FPGA was divided into five modules namely the clock divider, Address increamenter, Filter module, Memory Unit and the Checker module. The clock divider module was used to convert the reference frequency available in the board into a lower frequency required for operation and to synchronize the operation of the other modules. As no inbuilt ADC module was present on board, samples of the input signal was stored in the Memory unit in from of 8 bit samples in 32 memory location. The Address increamenter was used as a 5 bit counter to fetch one sample at a time from a particular memory location and give it to the Filter module. The output obtained from the Filter module was checked in the checker module where the bit steam corresponding to the envisaged output was stored. If the output of the Filter and the checker module matched then it would light up a LED interfaced to the FPGA.

V. RESULTS & DISCUSSIONS

Digital Filters were designed by defining the filter specifications in the analog domain, transforming the analog filter specification into digital filter specification, defining the order of the filter, the sampling rate and type of filter to be used like a FIR or an IIR filter depending on the specifications like pass band ripple, linear phase requirement etc. The coefficients were derived for the filter based on the specifications and response checks were done using mathematical software with the same filter coefficients derived earlier to check the output for the filter designed from the given specifications. VHDL was used in the design entry stage to implement the filter hardware. Test bench written in VHDL was used to check the correctness of hardware implementation as a part of behavioral simulation (after design entry), post-synthesis

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 5, Issue 4, October 2015

simulation (after the code is converted to a gate level net list using a synthesizer) and post-layout simulation (after placement & routing). For all the three phases of simulation, the same test bench was used.Actel's PROSAIC3 FPGA kit was programmed with the generated bit stream using a JTAG interface and test inputs in real time were given and filtering capabilities of the designed filter hardware was checked and found to be appropriate.

VI. CONCLUSION

Filters such as seven point moving average filter for time domain filtering or waveform shaping, frequency selective filters like low pass, high pass filters were implemented and checked.A tunable Low pass FIR filter was also designed which can be tuned in the frequency range from 1Khz to 15Khz according to the user select inputs [9-10]. Moreover an IIR Butterworth 6^{th} order Notch filter was designed to suppress power supply interference. An all pass filter in form of a Hilbert transformer which would cause a phase change of 90 degree to all frequency components was implemented. This work concludes that in future digital filters can be implemented in flexible FPGA based architecture where the coefficients can be programmed and changed when required to implement filters of specific requirement. Concepts of tunable filters based on user selection can also be implemented using such flexible architecture.

VII. FUTURE ENHANCEMENT

Time domain Filters to deal with salt and pepper noise and shot noise which are seen in daily life and will be helpful in front end signal processing to be attempted and implemented in real time signal processing applications.

VIII. ACKNOWLEDGEMENT

I sincerely acknowledge the opportunity given to me by BARC Training School (IGCAR campus), AERB &Director, EIRSG, IGCAR for carrying out the work.

REFERENCES

[1] Vijender Saini, Balwinder Singh & Rekha Devi, "Area optimization of FIR filter and its implementation in FPGA", International Journal of Recent Trends in Engineering, Vol 1, No. 4, pp. 55-58, May 2009

[2] Kowalski J.E. &Berner, J.B, " Digital Filter ASIC For NASA deep space radio science receiver" ASIC conference & Exhibit, pp.39 – 42, 18-22 Sep 1995, Austin, TX.

[3] "Digital Signal Processing", Ramesh Babu, 4^{th} edition.

[4] A.Heubi, M.Ansorge, F.Pellandini. "Low power Architecture for Digital Signal Processing", GRETSI, 1993, 3661-3664.

[5] "The Scientists & Engineers guide to Digital Signal processing", Steven.W.Smith.

[6] www.mathworks.com

[7] "Digital Signal Processing Using Field Programmable Gate Arrays", Uwe-Meyer-Baese, Springer publication.

[8] Sara Grassi, Alexandre Heubi, Micheal Ansorge, Fausto Pellandini."Study of a VLSI implementation of a noise reduction algorithm for Digital hearing aid", EUSIPCO, 1994, pp-1661-1664.

[9] E.C Tan. "Variablelowpass wave-digital filters", Electronics letters, vol.18 No.8, pp.324-326.

[10] S.S Ahuja&.S.C Dutta Roy. "Variable Digital Filters", IEEE trans. Circuits.Systms, vol-CAS- 27, N.O-9, pp.836-838.

AUTHORS PROFILE

Shri Suvadip Roy is an Electronics Engineer from 56^{th} Batch of BARC Training School. He has done his Post Graduation in Electronics & Instrumentation Engineering. Currently he is working as a Nuclear Regulator in Atomic Energy RegulatoryBoard and is associated with review of Instrumentation & Control Aspects of Nuclear Power Plants. His Area of Interest includes Digital Electronic Systems, Digital Signal Processing, NeutronicInstrumentaion, Modern Control Theory & Use of Programmable Logic Devices (FPGA & CPLD) in Safety Critical Applications in Nuclear Power Plants.