International Journal of Electrical and Computer Engineering (IJECE)
Vol. 7, No. 4, August 2017, pp. 1874~1881
ISSN: 2088-8708, DOI: 10.11591/ijece.v7i4.pp1874-1881

FPGA Implementation of Higher Order FIR Filter

Gurpadam Singh[1], Neelam R. Prakash[2]
[1]Departement of Electronics and Communication Engineering, Beant College of Engineering and Technology (BCET) Gurdaspur, India
[2]Electronics and Communication Engineering at PEC University of Technology, Chandigarh, India

Article Info

Article history:

Received Aug 29, 2016
Revised Apr 4, 2017
Accepted Apr 21, 2017

Keyword:

Fir filter
FPGA
Higher order fir filter
Xilinx

ABSTRACT

The digital Finite-Impulse-Response (FIR) filters are mainly employed in digital signal processing applications. The main components of digital FIR filters designed on FPGAs are the register bank to save the samples of signals, adder to implement sum operations and multiplier for multiplication of filter coefficients to signal samples. Although, design and implementation of digital FIR filters seem simple but the design bottleneck is multiplier block for speed, power consumption and FPGA chip area occupation. The multipliers are an integral part in FIR structures and these use a large part of the chip area. This limits the number of processing elements (PE) available on the chip to realize a higher order of filter. A model is developed in the Matlab/Simulink environment to investigate the performance of the desired higher order FIR filter. An equivalent FIR filter representation is designed by the Xilinx FIR Compiler by using the exported FIR filter coefficients. The Xilinx implementation flow is completed with the help of Xilinx ISE 14.5. It is observed how the use of higher order FIR filter impacts the resource utilization of the FPGA and it's the maximum operating frequency.

Corresponding Author:

Gurpadam Singh,
Departement of Electronics and Communication Engineering,
Beant College of Engineering and Technology,
Gurdaspur, Punjab, India, PIN 143521.
Email: gurpadam@yahoo.com

1. INTRODUCTION

There are many problems in the engineering which are computational intensive. Considerable attention has been given on the implementation of FIR filters ranging from full custom to general purpose processors. The digital Finite-Impulse-Response (FIR) filters are mainly employed in digital signal processing applications. To attain high performance, various design and implementation strategies of digital FIR filters have been reported in the literature. To utilize the advantages of systolic processing, several algorithms and architectures have been suggested for systolization of FIR filters by Mohanty & Meher [1] and Parhi [2].

Razak et al. [3] described the design of transposed form FIR filter on FPGAs using VHDL. Ab-Rahman et al. [4] had presented ASIC design flow for the implementation of adaptive FIR filter using MATLAB and Mentor Graphics IC Design tools. Daitx et al. [5] proposed a VHDL approach for designing of optimized FIR filters where the general coefficient representation for the optimal sharing of partial products in multiple constants multiplications is used. Meher et al. [6] reported the design optimization of one and two dimensional fully pipelined computing structures for area delay-power-efficient implementation of FIR filter by using the systolic decomposition of distributed arithmetic based inner product computation. The main components of digital FIR filters design on FPGAs are the register bank to save the samples of signals, adder to implement sum operations and multiplier for multiplication of filter coefficients to signal

samples. Jadhav & Mane have suggested a parallel FIR digital filter structure which is suitable for efficient implementation of filters whose coefficient values are sums of power of two terms and by exploiting this architecture it is shown that merged multipliers provide the best trade-off between speed and resource requirements [7].

Although, design and implementation of digital FIR filters seem simple but the design bottleneck is multiplier block for speed, power consumption and FPGA chip area occupation. A common interesting idea in digital circuits design is replacing multiplier block by some shift and add operations and sharing common terms.

Application specific integrated circuits (ASICs) are considered a good choice where there is a need to achieve a higher speed of operation however field programmable gate arrays (FPGAs) also provide a suitable alternative to ASIC due to the advent of new technologies that have reduced delay times of FPGAs and the availability of powerful synthesis tools.

There are two main designing environments for implementation of digital FIR filters; digital signal processor (DSP)-based and FPGA-based solutions. Traditional solutions are based on microprocessors, microcontrollers and DSP devices, whereas FPGAs provide design flexibility and hardware parallelism by breaking the paradigm of sequential execution for high speed applications [8]. For effective implementation of computationally intensive applications having non standard algorithms, the designer may find that mapping the entire application on FPGAs is the only option. Moreover FPGAs offer design reuse, and better performance than a software solution mapped on a DSP [9]. In reconfigured systems, FPGAs provide custom hardware solutions in many high bandwidth signal processing applications.

The multipliers are an integral part in FIR structures and these use a large part of the chip area. This limits the number of processing elements (PE) available on the chip to realize a highest order of filter. Similarly, for DA-based (multiplierless) implementation of FIR filters, the memory requirement increases exponentially with the filter order. Hence the complexity of FIR implementation grows with the filter order and the precision of computation; therefore the real-time realization of these filters with desired level of accuracy is a challenging task [6].

2. DIGITAL FILTERS

2.1. Types of Digital Filters

Rabiner [10] has distinguished the digital filters types by the way these are realized from their filter characteristics.

Finite-duration impulse-response (FIR):

In case of FIR filters the duration of the filter impulse response $h(n)$, is finite i.e.

$$h(n) = 0,\ n > Ni < \infty \tag{1}$$

$$h(n) = 0,\ n < Nj > -\infty, \tag{2}$$

$$Ni > Nj \tag{3}$$

Infinite-duration impulse-response (IIR):

In case of IIR filters the duration of the filter impulse response $h(n)$, is infinite; i.e., there exists no finite values of either Ni or Nj.

2.1.1. Recursive Realization

Recursive realization describes the way how an IIR or FIR is realized. The filter output $y(n)$, is obtained in terms of past filter outputs $y(n-1)$ ... as well as in terms of past and present filter inputs $(n), x(n-1), \dots$. Thus the output of a recursive realization can be written as

$$y(n) = F(y(n-1), y(n-2), \dots., x(n), x(n-1), \dots) \tag{4}$$

2.1.2. Nonrecursive Realization

Nonrecursive realization means that only the past and present inputs are used to get the current filter output *y(n)*; i.e., previous outputs are not used to generate the current filter output $y(n)$. This realization can be written as

$$y(n) = F(x(n), x(n-1), \dots) \tag{5}$$

The realization of both FIR filters and IIR filters can be performed using nonrecursive and recursive methods [11], [12]. However it is worth to mention here that, usually nonrecursive realizations of FIR filters and recursive realizations of IIR filters are most efficient and are in generally used.

2.2. Need for FPGA Implementation

In the present scenario there is need of high speed, low overhead and low cost for the implementation of any digital circuit. Keeping in view of DSP arithmetic related support, many FPGA families provide fast-carry chains (Xilinx Virtex, Altera FLEX) that allow large bit width fast adders, without the use of carry look-ahead decoders. This DSP arithmetic support helps to implement multiply-accumulates (MACs) related operations at a very high speed.

In the DSP hardware design, hardware description languages (HDL) are more preferably used than graphical design entries as HDL based entries helps "code reuse" in a much better way than graphical design entries. The textual environment is preferred due to better algorithmic control of design and the availability of a number of design styles.

Moreover, now a days devices are based on a different structure (depending on the specific application) for each generation of devices further subdivided into families with each family having different features. In Xilinx there are higher performance Virtex device and lower-cost Spartan devices. Similarly Altera has high performance Stratix device and lower-cost Cyclone devices. Virtex and Stratix both have specialized offerings in the embedded, digital signal processing, and high bandwidth processing areas. So, many FPGA families provide the best DSP complexity and maximum speed.

2.3. Systolic FIR Applications

Kung and Leiserson [13] have reported systolic system consisting of an array of processing elements (typically multiplier-accumulator chips) in a pipeline structure that is used for applications such as image and signal processing. The systolic array introduced by H. T. Kung of Carnegie-Mellon in 1978, referred to the rhythmic transfer of data through the pipeline, like blood flowing through the vascular system.

A systolic array can achieve higher computation throughput without the need of increase of memory bandwidth as shown in Figure 1.

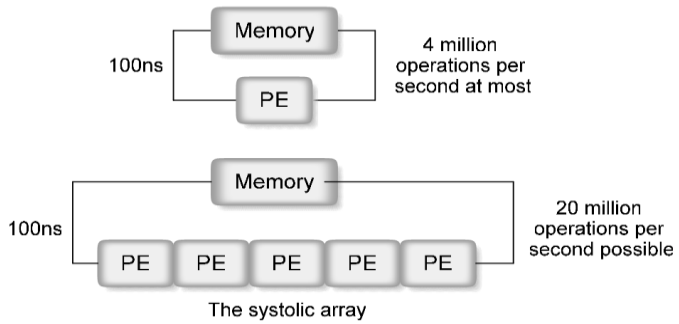

Figure1. Systolic array throughput

Systolic array architecture improves timing by reducing critical path with the use of same amount of hardware. By involving the concept of pipelining and parallel processing into systolic architecture highly reduce adaption delay, chip area and power consumption [14]. With increase of order of filter the hardware resource optimization becomes a real challenge.

3. LITERATURE REVIEW ON FIR FILTERS AND THEIR IMPLEMENTATION

The interest in FIR filter design techniques is renewed because of the availability of powerful optimization algorithms for the design of problems in the past many years. Various optimization techniques have been proposed by Gold & Jordan [15], Rabiner et al. [16], [17], Herrmann & Schuessler [18] and Herrmann [19] which, along with the classical window design methods reported by Kaiser [20] and Helms [21] provide the user with several possibilities for approximating filters with arbitrary frequency-response characteristics.

Rabiner [10] has discussed the general theory behind windowing and two optimization techniques-frequency-sampling and equiripple designs, and has compared these methods with respect to several practical and theoretical criteria. Optimum window designs have been proposed by Kaiser and Helms. Rabiner et al. and Rabiner & Steiglitz [16] have reported an extensive catalog of frequency-sampling designs on several standard filters. Herrmann & Schuessler [18] and Herrmann [19] reported a third technique for designing FIR filters which solves a system of nonlinear equations to generate a filter with an equiripple approximation error**.**

Ramstad and Saramaki [22] have reported that narrow transition-band FIR filters often require forbidding filter lengths for practical implementations. IIR filters offer significantly lower order but they suffer from shortcomings like nonlinear phase, instabilities and very large word-length requirements for the same filter specifications, and it is therefore desirable to find alternative FIR structures that lower the processing load.

Jiang and Bao [23] have designed a 16-order FIR filter response coefficients by Matlab toolbox FDA Tool and analyzed the feasibility for the filter performance on FPGA chip EPIK30TCI44-3.

4. MODEL DEVELOPMENT

This paper is aimed to consider the design and implementation of a higher order FIR filter on a FPGA. A model is developed in the Matlab/Simulink environment to investigate the performance of the higher order FIR filter. The design flow is simulated and tested with System Generator, a system level modeling tool available from Xilinx. System Generator can be used for designing and testing of DSP systems on FPGAs in a Matlab/ Simulink environment. The FIR filter is designed in the FDATool by setting filter performance specifications as per Table 1 and the frequency response of the same is checked to measure how closely our design meets the filter specifications. Once the desired filter response has been achieved, the filter coefficients are exported to the Matlab workspace. An equivalent FIR filter representation is designed by the Xilinx FIR Compiler by using the exported FIR filter coefficients. The Xilinx implementation flow is completed with the help of Xilinx ISE 14.5

Table 1. Bandpass FIR Filter Parameters

Design Method	Equiriple
Order of FIR filter	182
Sampling Frequency Fs	3MHz
Stopband Frequency Fstop	560KHz
Passband Frequency Fpass	600KHz
Passband Frequency Fpass	900KHz
Stopband Frequency Fstop	940KHz
Passband Weight Wpass	0.1
Stopband Weight Wstop	0.001

5. SIMULATION AND IMPLEMENTATION

Simulation and implementation of the digital FIR filter on single FPGA chip are discussed with different higher order of filters. The FIR filter is implemented on Virtex6; chip number xc6vlx240t-ff1156, from Xilinx Inc. The different higher order FIR filters are implemented using Xilinx System Generator in Matlab/Simulink environment. System Generator uses the MathWorks model-based Simulink design environment for FPGA design. FIR filter is designed in the Simulink modeling environment using a Xilinx specific blocksets as shown in Figures 2 and Figure 3. System Generator uses a FIR Compiler block which helps to target the dedicated DSP48 hardware resources available in the Virtex devices to create optimized FPGA implementations of the designed model. MathWorks FDATool helps to create the coefficients for Xilinx FIR Compiler. We have designed and simulated FIR filters using (a) FDATool and (b) Xilinx FIR Compiler.

The frequency response of the designed filter having order 182 is interpreted in figure 4. The simulated outputs of FDATool block and Xilinx FIR compiler blocks are presented in Figures 6 and Figure 7. FIR filter designs are synthesized using Xilinx ISE 14.5 edition.

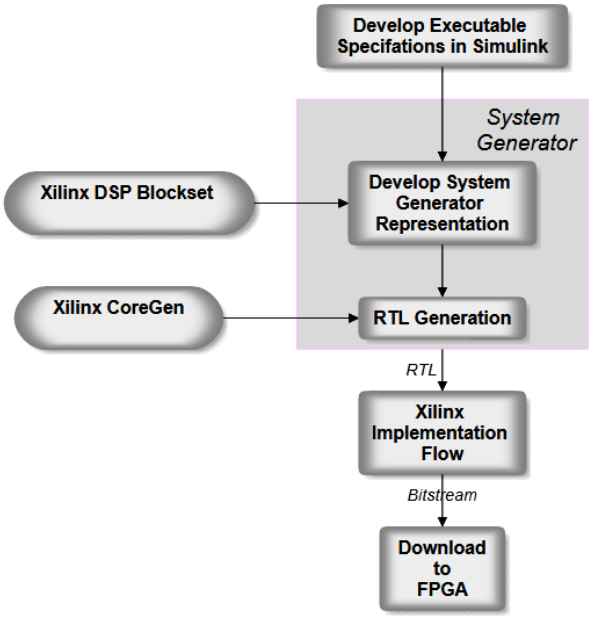

Figure 2. System Development Flow

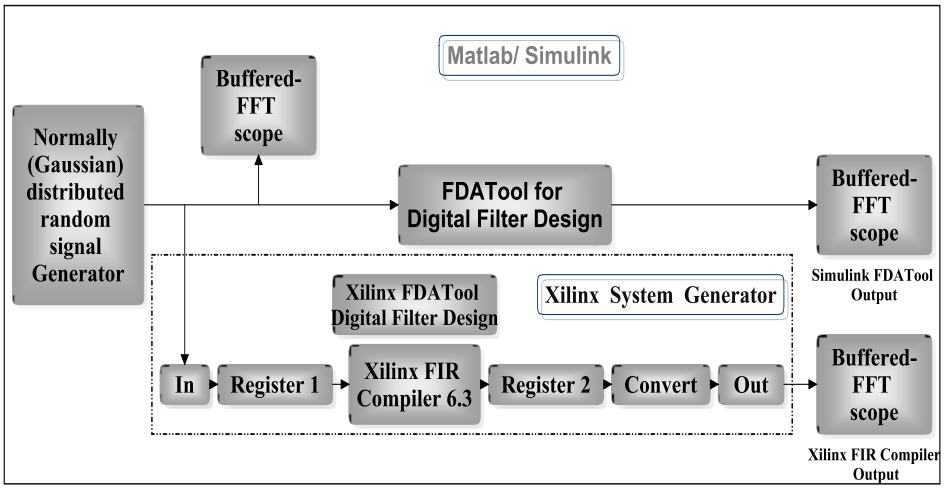

Figure 3. System model of FIR filter in Matlab/Simulink environment using Xilinx System Generator

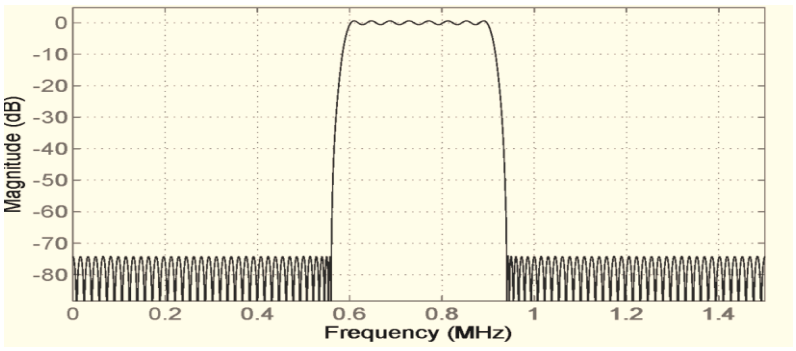

Figure 4. Frequency response of FIR filter (N=182) designed by FDATool in Matlab/Simulink environment

The response of the above designed filter is observed by applying a normally (Gaussian) distributed random signal (Figure 5) on a buffered FFT scope. Similarly FIR filter designed with the help of Xilinx System Generator is also applied with the same input signal. The response of the filters to the input signal is represented in Figures 6 and Figure 7.

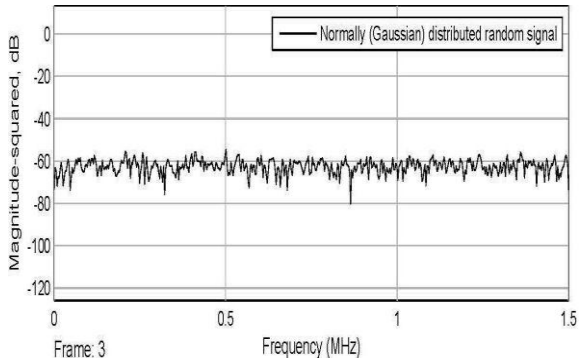

Figure 5. Normally (Gaussian) distributed random input signal

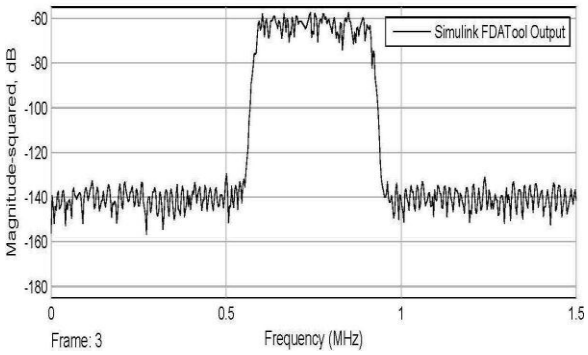

Figure 6. Simulink FDATool Output on buffered spectrum scope for FIR filter order 182

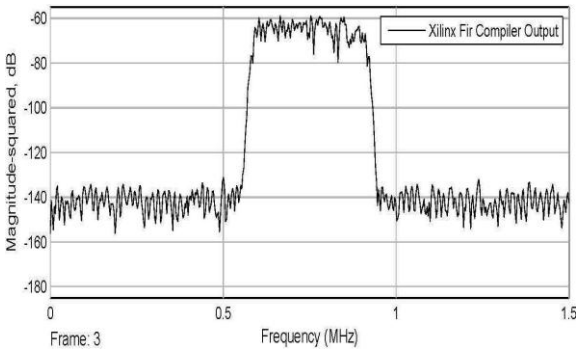

Figure 7. Xilinx Compiler Output on buffered spectrum scope for FIR filter order 182

6. RESULTS AND DISCUSSION

Frequency response of the output of both designed filters is observed on the buffered FFT(spectrum) scopes as shown in figure 6 & 7. After comparing the results from the two spectrum scopes it is observed that the results are similar and meets the specifications in the passband and stopbands though fixed point effects are seen in the FIR Compiler output.

To investigate further, higher order FIR filters (with N=273, 365 and 409) are designed and their response to input signal are observed. The designed FIR filters are synthesized and are implemented. The results for device utilization are summarized in table 2. The maximum operating frequency achieveable with synthesized architectures is given in table 3.

Table 2. Device Utilization Summary

Order of Filter	Slice Logic Utilization		Slice Logic Distribution		IO Utilization	Specific Feature Utilization
N	Slice Registers	Slice LUTs	Occupied Slices	LUT Flip Flop pairs used	Bonded IOBs	DSP48E1s
182	2064	840	387	1310	37	32
273	3021	1171	597	1960	37	47
365	3975	1435	852	2675	37	62
409	4471	1598	939	3040	37	70

Table 3. Timing Report of Designed FIR Filter

	Post-PAR Static Timing		
Order of Filter (N)	Minimum period* (ns)	Maximum operating frequency (MHz)	Maximum path delay from/to any node(ns)
182	3.349	298.597	2.378
273	3.986	250.878	2.150
365	4.030	248.139	1.999
409	5.430	184.162	2.713

* The minimum period statistic assumes all single cycle delays.

7. CONCLUSION

Higher order FIR filters are successfully designed and implemented up to an order of 409. Their output response to a normally (Gaussian) distributed random signal is observed on buffered FFT scopes. The output response of filters designed in FDATool based digital filter and Xilinx based FIR compiler are compared. It is observed from two spectrum scopes that response of filters designed in FDATool based digital filter and Xilinx based FIR compiler are similar and both meets the specifications in the passband and stopbands. From the Xilinx design summary it is observed that all signals are completely routed. The impact of higher order FIR filter on the resource utilization of the FPGA and it's the maximum operating frequency is investigated from the results presented in Table 2 and Table 3. It is observed that device utilization of slice registers, slices LUT, LUT based flip-flops and DSP48E1s pair goes on increasing with the increase of order where as the bonded IOB remains the same. From the Post PAR reports it is observed that with the increase of the order of the filter (N) the maximum operating frequency decreases as the minimum period increases. The maximum path delay from/to any node in is minimum for N= 365 in comparison to other observations.

REFERENCES

[1] Mohanty BK, Meher PK, "Novel Flexible Systolic Mesh Architecture for Parallel VLSI Implementation of Finite Digital Convolution", *IETE J. Res*, 1988;44(6):261–266.

[2] Parhi K K, "VLSI Digital Signal Processing Systems: Design and Implementation", New York, Wiley, 1999.

[3] Razak AHA, Abu Zaharin MI, Haron NZ, *"Implementing Digital Finite Impulse Response Filter using FPGA"*, Proc. of Asia-Pacific Conference on Applied Electromagnetic, Melaka, Malaysia, 2007;12: 1-5.

[4] Ab-Rahman AAH, Kamisian I, Shaameri AZ, *"ASIC Modeling and Simulation of Adaptive FIR Filter,"* in Proc. International Conference on Electronic Design, Penang, Malaysia, 2008; 12:1-4.

[5] Daitx FF, Rosa VS, Costa E, Flores P, Bampi S, *"VHDL Generation of Optimized FIR Filters"*, 2nd International Conference on Signals, Circuits and Systems (SCS 2008), 2008:1-5.

[6] Meher PK, Chandrasekaran S, Amira A, *"FPGA Realization of FIR Filters by Efficient and Flexible Systolization using Distributed Arithmetic*", IEEE Trans. Signal Processing, 2008;56(7): 3009 – 3017.

[7] Jadhav SB, Mane NN, "A Novel High Speed FPGA Architecture for FIR Filter Design", *International Journal of Reconfigurable and Embedded Systems (IJRES)*, 2012;1(1):1-10, ISSN: 2089-4864.

[8] Nekoei F, Kavian YS, Strobel O, *"Some Schemes of Realization Digital FIR Filters on FPGA for Communication Applications"*, 20th Int. Crimean Conference on Microwave & Telecommunication Technology(CriMiCo'2010), 13-17th September, Sevastopol, Crimea, Ukraine.2010.

[9] Khan SA, "Digital Design of Signal Processing Systems: A Practical Approach", John Wiley & Sons, 2011.

[10] Rabiner LR, "Techniques for Designing Finite-Duration Impulse-Response Digital Filters", *IEEE Trans. Communication Technology*, 1971;Com-19.

[11] Gold B, Jordan KL, *"A Note on Digital Filter Synthesis,"* Proc. IEEE (Lett.), 1968;56(10): 1717-1718.

[12] Voelcker HB, Hartquist EE, "Digital Filtering via Block Recursion", IEEE Trans. Audio Electroacoust, 1970; AU-18: 169-176.

[13] Kung HT, Leiserson CE, *"Systolic Array for VLSI"*, Sparse Matrix Proc., 1979; 256-282.

[14] Bailmare RH, Honale SJ, Kinge PV, "Design and Implementation of Adaptive FIR filter using Systolic Architecture", *International Journal of Reconfigurable and Embedded Systems (IJRES)*, 2014;3(2): 54-61, ISSN: 2089-4864.
[15] Gold B, Jordan KL, "A Direct Search Procedure for Designing Finite Duration Impulse Response Filters", *IEEE Trans. Audio Electroacoust*, 1969;AU-17:33-36.
[16] Rabiner LR, Steiglitz K, "The Design of Wide-band Recursive and Nonrecursive Digital Differentiators", *IEEE Trans. Audio Electroacoust*, 1970; AU-18:204-209.
[17] Rabiner LR, Gold B, and McGonegal CA. An approach to the approximation problem for nonrecursive digital filters. IEEE Trans. Audio Electroacoust. 1970;AU-18:83-106.
[18] Herrmann O, Schuessler H W, *"On the Design of Selective Nonrecursive Digital Filters"*, IEEE Arden House Workshop, Harriman, N. Y.1970.
[19] Herrmann O, "On the Design on Nonrecursive Digital Filters with Linear Phase," Electron. Lett. 1970.
[20] Kaiser J F, "Digital Filters, in System Analysis by Digital Computer", F. F. Kuo, J. F. Kaiser, Eds. New York, Wiley. 1966.
[21] Helms HD, "Nonrecursive Digital Filters: Design Methods for Achieving Specifications on Frequency Response," *IEEE Trans. Audio Electroacoust*., 1968; AU-16:336-342.
[22] Ramstad TA, Saramaki T, *"Multistage, Multirate FIR Filter Structures for Narrow Transition-band Filters"*, IEEE International Symposium on Circuits and Systems.1990.
[23] Jiang X, Bao Y, *"FIR Filter Design based on FPGA"*, International Conference on Computer Application and System Modeling (ICCASM 2010), 2010;13:621-624.

BIOGRAPHIES OF AUTHORS

Gurpadam Singh received his BE degree in Electronics and Electrical communication Engineering from Panjab University, Chandigarh, India) in 1994 and M. Tech. degree in Electronics Punjab Technical University, Jalandhar, (India) in 2003. He is pursuing his Ph.D Degree from PEC University of Technology Chandigarh, India. He has work experience of Academia of over 18 years. Presently he is working as an associate professor in the department of Electronics and Communication Engineering at Beant College of Engineering and Technology, Gurdaspur, Punjab, INDIA. He has a number of publications, He has guided many M.Tech. students in field of Electronics and Communication Engineering**.**

Dr. Neelam Rup Prakash received her BE degree in Electronics and Electrical communication Engineering from Punjab Engineering College (Panjab University, Chandigarh, India) in 1987 and her ME degree in Electronics from Punjab Engineering College (Panjab University, Chandigarh, India) in 1996 and was awarded the certificate of Merit for 1st position in University. She completed her Ph.D Degree also from Punjab University Chandigarh, India in 2002. She has work experience of both Industry and Academia of over 27 years. She has number of publications, She has guided many ME and Ph.D Students in field of Electronic System Design**.** Dr. Prakash is a member of IEEE, USA and IEEE Circuits and Systems Society.