FPGA Implementation of Higher Order FIR filter: Speed vs. Area

[1]Gurpadam Singh, [2]Dr Neelam Rup Prakash
[1]Associate Professor,[2] Professor,
[1]Department of Electronics and Communication Engineering,
[1]Beant College of Engineering and Technology, Gurdaspur, Punjab,India

***Abstract*—In this paper a higher order FIR filter is designed with a digital filter design tool in Matlab/Simulink for given specifications. An equivalent model is developed with the help of Xilinx System Generator blocks in the Matlab/Simulink environment. For simulation appropriate parameters are set in the Simulink model. The simulation of the filter is completed with Digital Filter Design tool as well as with Xilinx FIR Compiler. The implementation results are obtained by setting Optimization Goal: first as Speed and then as Area in the Xilinx ISE Design Suite. A comparative evaluation of implementation with Optimization Goal: Speed vs Area is carried out based on Place and Route Report, Post-PAR Static Timing Report and Xilinx XPower Analyzer's Power Report.**

***Index Terms*— FIR Filter, higher order FIR filter, FPGA.**

I. INTRODUCTION

Many problems in the engineering are computational intensive. Considerable attention is given for these computational intensive problems and their efficient way of implementation. The digital Finite-Impulse-Response (FIR) filters are mainly employed in digital signal processing applications. 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. described the design of transposed form FIR filter on FPGAs using VHDL [3]. Ab-Rahman et al. had presented ASIC design flow for the implementation of adaptive FIR filter using MATLAB and Mentor Graphics IC Design tools [4]. Daitx et al. proposed a VHDL approach for designing optimized FIR filters where the general coefficient representation for the optimal sharing of partial products in multiple constants multiplications is used [5]. 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. 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. 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 [6]. 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. [7]. In this paper an effect is made to evaluate the design and implementation of higher order FIR filters with optimization goal, first with speed, and then with area by comparing the Place and Route Reports, Post-PAR Static Timing Reports and Xilinx XPower Analyzer's Power Reports.

II. DIGITAL FILTERS

Types of Digital Filters [8]

Rabiner 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.,

$$h(n) = 0,\ n > N1 < \infty, \quad (1)$$

$$h(n) = 0,\ n < N2 > -\infty, \quad (2)$$

$$N1 > N2 \quad (3)$$

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 $N1$ or $N2$.

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 $x(n), x(n-1), \ldots$. Thus the output of a recursive realization can be written as

$$y(n) = F(y(n-1), y(n-2), \ldots., x(n), x(n-1), \ldots) \quad (4)$$

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

$$y(n) = F(x(n), x(n-1), \ldots) \quad (5)$$

The realization of both FIR filters and IIR filters can be performed using nonrecursive and recursive methods [9], [10]. 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.

III. 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[11], Rabiner et al. [12], [13], Herrmann & Schuessler [14] and Herrmann [15] which, along with the classical window design methods reported by Kaiser [16] and Helms [17] provide the user with several possibilities for approximating filters with arbitrary frequency-response characteristics. Rabiner 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 [8]. Optimum window designs have been proposed by Kaiser and Helms. Rabiner et al. and Rabiner & Steiglitz have reported an extensive catalog of frequency-sampling designs on several standard filters. Herrmann & Schuessler and Herrmann reported a third technique for designing FIR filters which solves a system of nonlinear equations to generate a filter with an equiripple approximation error [14], [15]**.** Ramstad and Saramaki 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 [18].

IV. MODEL DEVELOPMENT

In this paper a model is developed in the Matlab/Simulink environment with the help of Xilinx System Generator to investigate the effect of speed optimization vs area optimization for the FPGA implementation of higher order FIR filter. The design flow is simulated and tested with System Generator, a system level modeling tool available from Xilinx. The FIR filter is designed in the FDATool by setting filter performance specifications as per table 1 and the frequency response is checked to find how our design meets the given filter specifications. Then filter coefficients are exported to the Matlab workspace. An equivalent FIR filter is designed with the help of Xilinx FIR Compiler. In this paper Xilinx implementation flow is completed with Xilinx ISE 14.5.

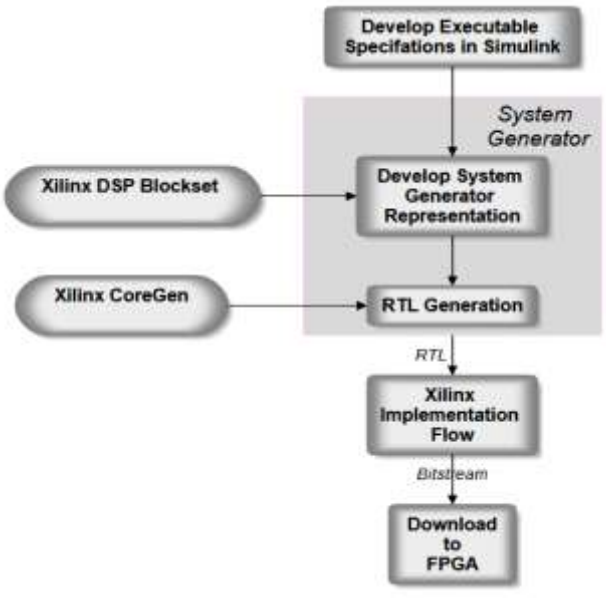

Figure 1. System Development Flow

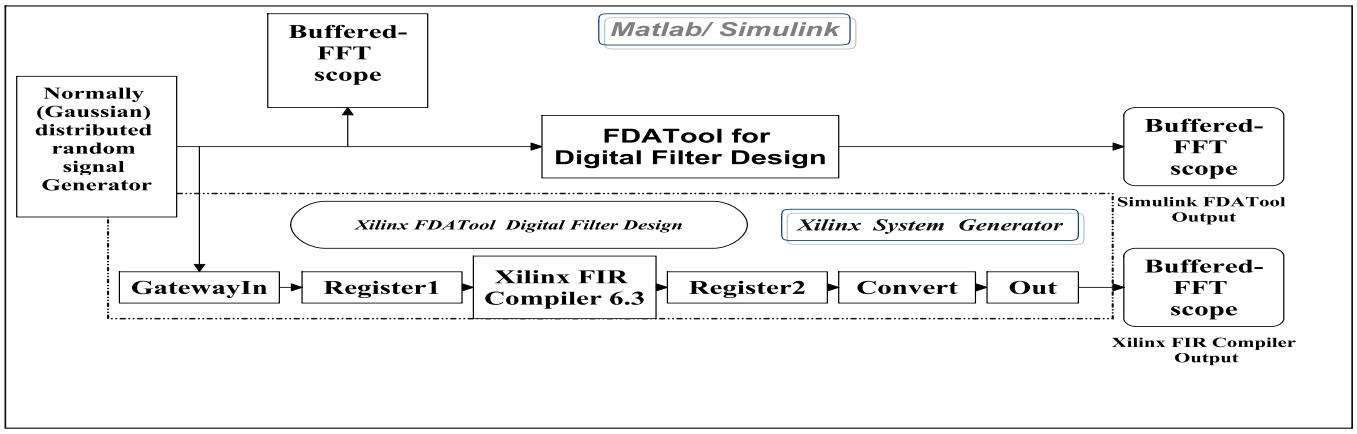

Figure 2. System model of FIR filter in Matlab/Simulink environment using Xilinx System Generator.

Table 1 Bandpass FIR Filter Parameters

Design Method	Equiripple
Order of FIR filter	409
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

V. SIMULATION AND IMPLEMENTATION

FIR filters of the order of N= 182, 273, 367 and 409 are designed. An evaluation is made on the basis of slice logic utilization, timing analysis and Xilinx's Xpower power analysis report.Simulation and implementation of the digital FIR filter on single FPGA chip are discussed with different higher order of filters. The FIR filters are implemented on Virtex6. 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 figure 2. System Generator uses a FIR Compiler block which helps to use 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 409 is interpreted in figure 3. The simulated outputs of FDATool block and Xilinx FIR compiler blocks are presented in figures 5, 6 & 7. FIR filter are synthesized using Xilinx ISE 14.5 edition.

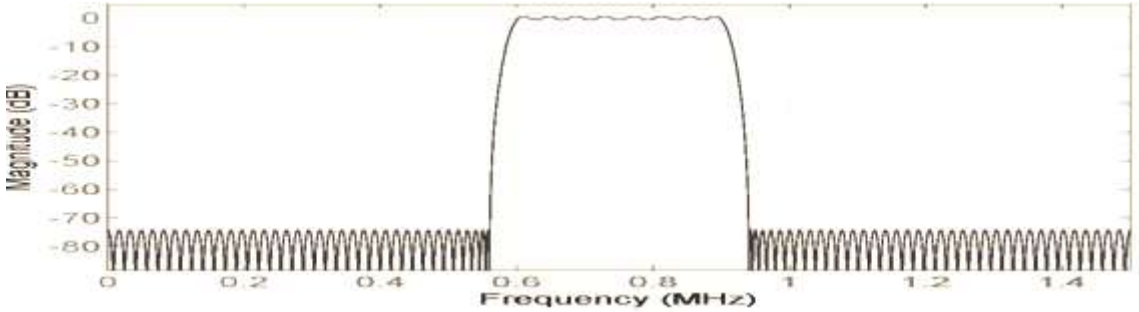

Figure 3. Frequency response of FIR filter (N=409) designed by FDATool in Matlab/Simulink environment.

The response of the designed filter to a normally (Gaussian) distributed random signal (figure 4) is observed on a buffered FFT scope. Similarly response of FIR filter designed using Xilinx System Generator to the same input signal is observed. These responses are represented in figures 5 & 6.

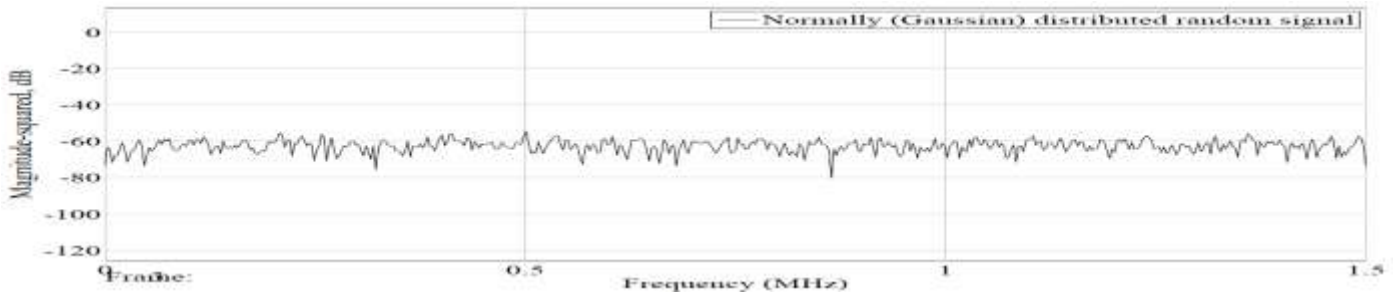

Figure 4. Normally (Gaussian) distributed random input signal

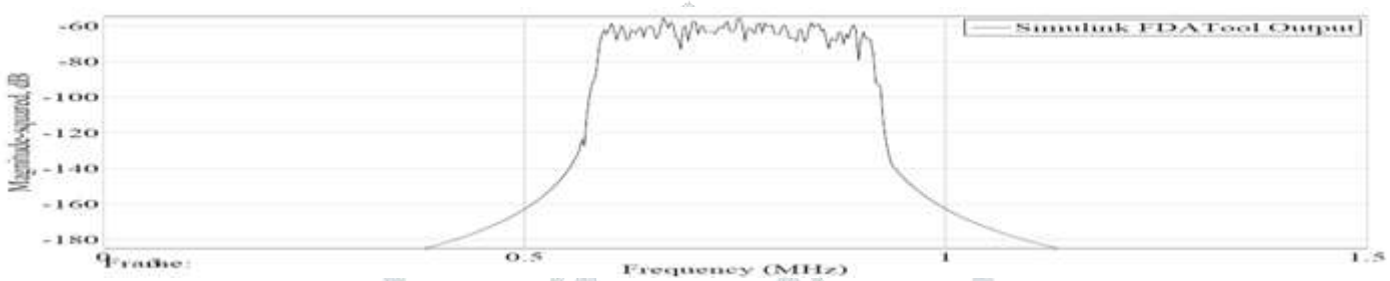

Figure 5. Simulink FDATool Output on buffered spectrum scope for FIR filter order 409.

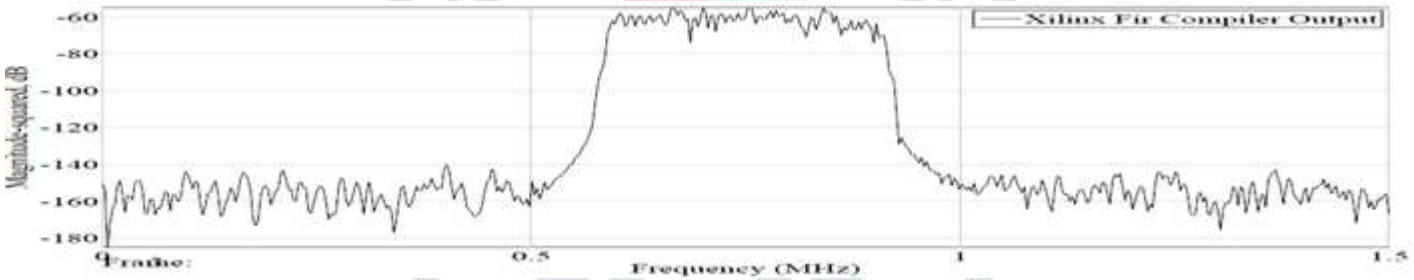

Figure 6. Xilinx Compiler Output on buffered spectrum scope for FIR filter order 409.

VI. RESULTS AND DISCUSSION

Frequency response of the output of both designed filters is observed on the buffered FFT(spectrum) scopes as shown in figures 5, 6. From the 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 in figure 6. Higher order FIR filters with filter order(N)=182, 273,367, 409 are simulated(on FDATOOL) and Implemented(by Xilinx FIR Compiler) on FPGA by setting the strategies in the tool first with optimization goal as speed and then with Design Goal: Area Reduction, Design Strategy: Area Reduction with Physical Synthesis and optimization Goal as Area. The results for device utilization, IO & specific feature utilization, timing analysis and power analysis are summarized in tables 2, 3, 4 and 5. The results are also graphically presented in figures 7,8, 9, 10, 11, 12 and 13.

Table 2 Device Utilization Summary

Order Of Filter	**Slice Logic Utilization**			
	Optimization Goal : SPEED [20]		**Optimization Goal : Area Reduction**	
N	***Slice Registers***	***Slice LUTs***	***Slice Registers***	***Slice LUTs***
182	2064	840	2004	774
273	3021	1171	2943	1138
367	4044	1419	3936	1607

409	4471	1598	4359	1845

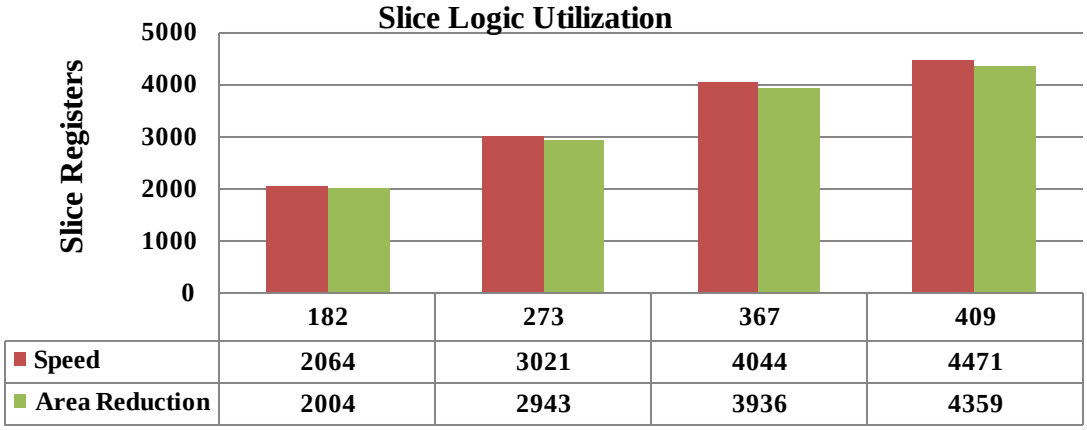

Figure 7. Slice Register utilization for FIR filter order N=182,273,367,409 designed with Speed vs Area optimization goals.

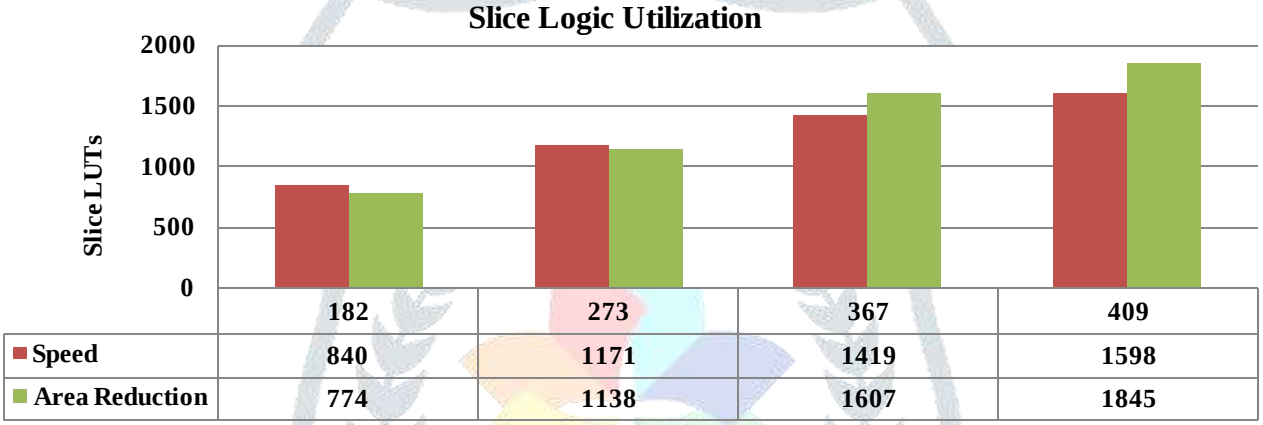

Figure 8. Slice LUT utilization for FIR filter order N=182,273,367,409 designed with Speed vs Area optimization goals.

Table 3 IO Utilization & Specific Feature Utilization

Order of Filter	IO Utilization & Specific Feature Utilization			
	Optimization Goal : SPEED [20]		Optimization Goal : Area Reduction	
N	*Bonded IOBs*	*DSP48E1s*	*Bonded IOBs*	*DSP48E1s*
182	37	32	37	32
273	37	47	37	47
367	37	63	37	63
409	37	70	37	70

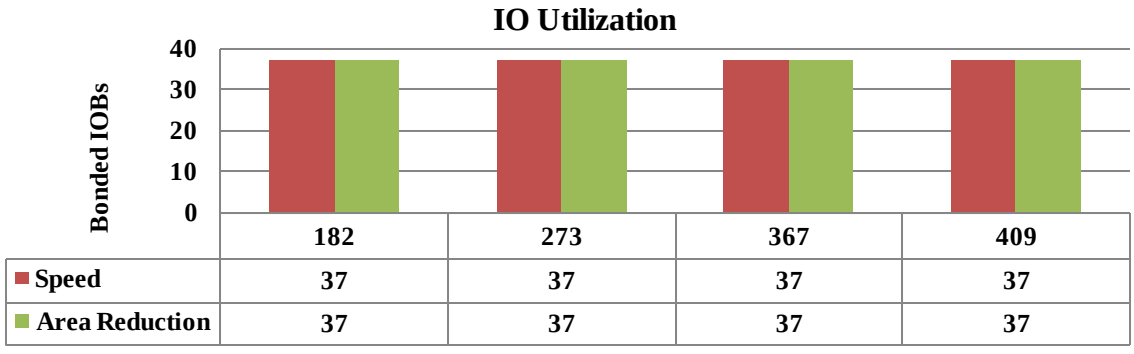

Figure 9. Bonded IOBs utilization for FIR filter order N=182,273,367,409 designed with Speed vs Area optimization goals.

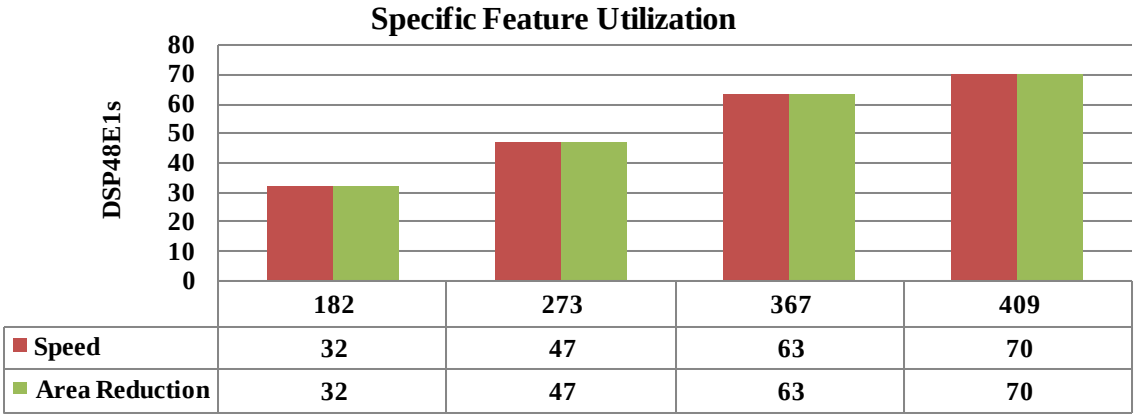

Figure 10. DSP48E1s utilization for FIR filter order N=182,273,367,409 designed with Speed vs Area optimization goals.

Table 4 Timing Analysis Report of Designed FIR Filter

	Post-PAR Static Timing					
	Optimization Goal : SPEED [20]			Optimization Goal : AREA Reduction		
Order of Filter (N)	*Minimum period*(ns)*	*Maximum operating frequency (MHz)*	*Maximum path delay from/to any node(ns)*	*Minimum period (ns)*	*Maximum operating frequency (MHz)*	*Maximum path delay from/to any node (ns)*
182	3.349	298.597	2.378	4.929	202.881	4.365
273	3.986	250.878	2.150	6.466	154.655	3.884
367	4.21	237.530	1.861	5.974	167.392	4.005
409	5.430	184.162	2.713	5.987	167.029	4.275

* The minimum period statistic assumes all single cycle delays.

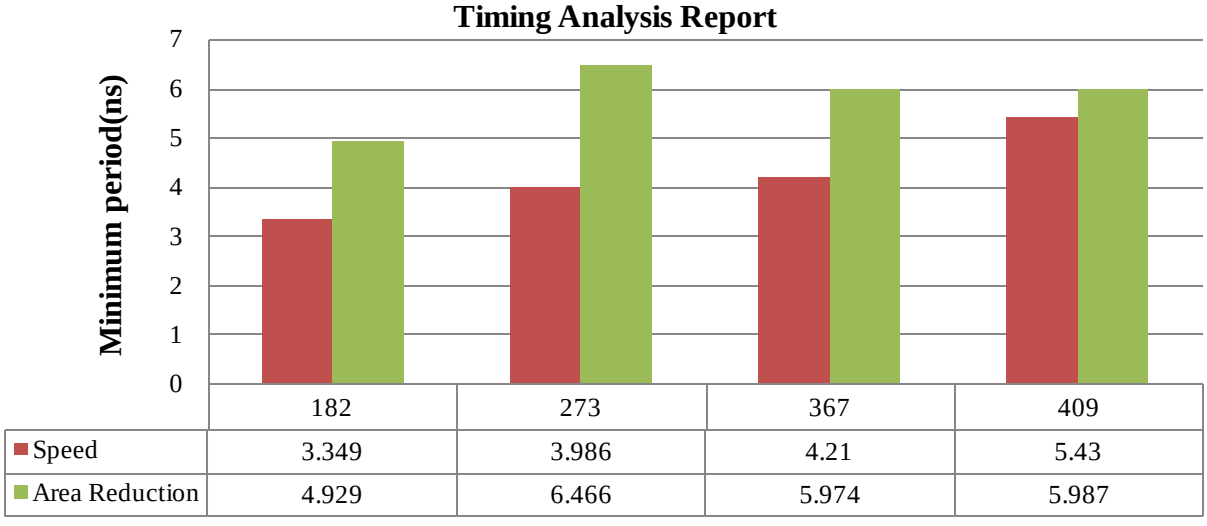

Figure 11. Minimum period for FIR filter order N=182,273,367,409 designed with Speed vs Area optimization goals.

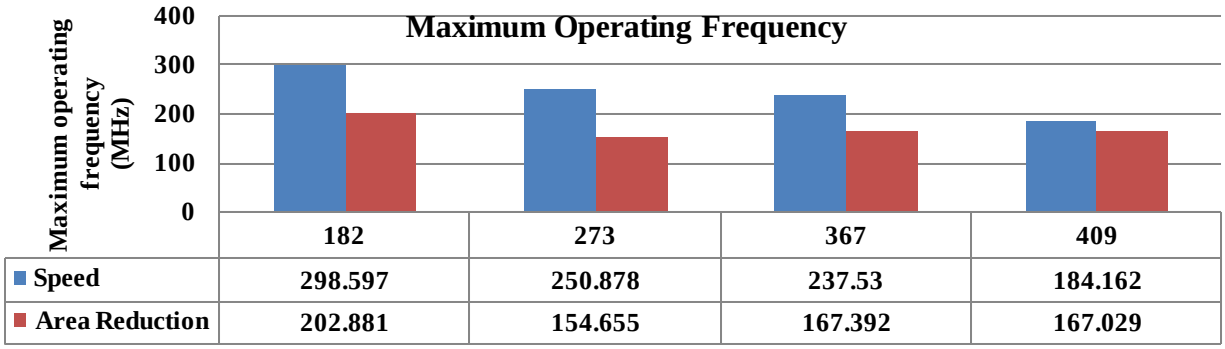

Figure 12. Maximum operating frequency for FIR filter order N=182,273, 367, 409 designed with Speed vs Area optimization goals.

Table 5 Power Analysis Report of Designed FIR Filter

	Power Supply Summary					
	Optimization Goal : SPEED[20]			Optimization Goal : AREA Reduction		
Order of Filter (N)	*Dynamic Power (mW)*	*Static Power (mW)*	*Total (mW)*	*Dynamic Power (mW)*	*Static Power (mW)*	*Total (mW)*
182	69.81	3423.71	3493.52	47.10	3423.00	3470.10
273	86.28	3424.22	3510.50	63	3423	3486
367	112.45	3425.04	3537.48	80.99	3424.06	3505.05
409	117.33	3425.19	3542.52	87	3424	3511

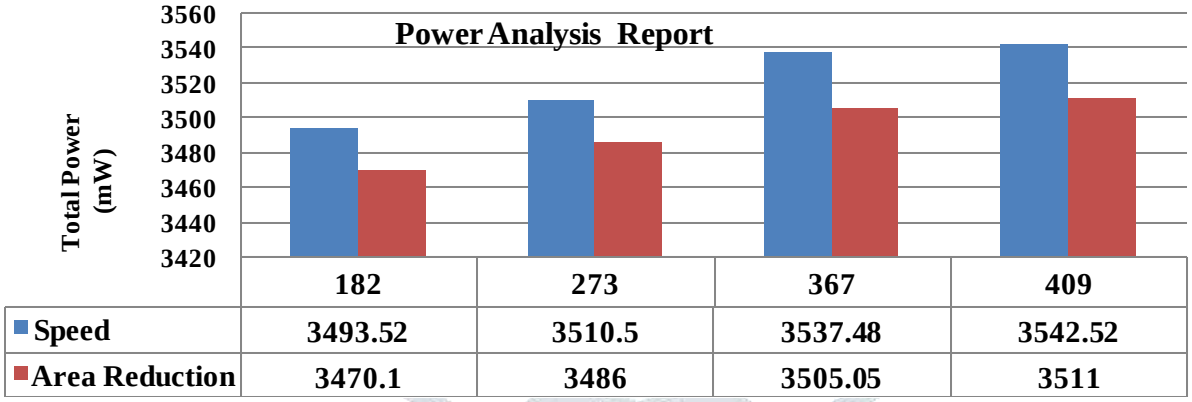

Figure 13.Total power analysis report for FIR filter order N=182,273,367, 409 designed with Speed vs Area optimization goals.

VII. CONCLUSION

Higher order FIR filters with filter order(N)=182, 273,367, 409 are successfully simulated(on FDATOOL) and implemented(by Xilinx FIR Compiler) on FPGA by setting the strategies in the tool, first with optimization goal as speed, and then with Design Goal: area reduction, design strategy: area reduction with physical synthesis and optimization goal as area. Response of filter to Normally (Gaussian) distributed random input signal is observed on a buffered FFT scopes for FDATOOL output and Xilinx FIR Compiler output. From Xilinx design summary it is observed that all signals are completely routed. A comparative evaluation of implementation with optimization goal: Speed vs Area is carried out based on Place and Route Report, Post-PAR Static Timing Report, Xilinx XPower Analyzer's Power Report. From Place and Route Report it is observed that less number of slice registers is utilized when the implementation goal is area. IO utilization & specific feature utilization summary clearly represents that number of bonded IOBs and DSP48E1s do not change when implementation goal is either speed or area. From Post-PAR static timing report for N=182,273,367 & 409 it very much indicated that less maximum operating frequency achievable when implementation goal is area. Power supply summary shows that there are less power requirements with implementation optimization goal as area compared to optimization goal as speed for the design & implementation of same higher order FIR on FPGA.

REFERENCES

[1] B. K. Mohanty and P. K. Meher, "Novel flexible systolic mesh architecture for parallel VLSI implementation of finite digital convolution," IETE J. Res., vol. 44, no. 6, pp. 261–266, Nov. 1988.

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

[3] A.H.A. Razak, M.I. Abu Zaharin and N.Z. Haron,"Implementing digital finite impulse response filter using FPGA," Proc. of Asia-Pacific Conference on Applied Electromagnetic, Melaka, Malaysia, Dec. 2007, pp. 1-5.

[4] A.A.H. Ab-Rahman, I. Kamisian and A.Z. Shaameri, "ASIC modeling and simulation of adaptive FIR filter," in Proc. International Conference on Electronic Design, Penang, Malaysia, Dec. 2008, pp. 1-4.

[5] F. F. Daitx, V.S. Rosa, E. Costa, P. Flores and S. Bampi, "VHDL generation of optimized FIR filters," 2nd International Conference on Signals, Circuits and Systems (SCS 2008), pp. 1 - 5.

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

[7] P.K. Meher, S. Chandrasekaranand and A. Amira, "FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic," IEEE Trans.Signal Processing, vol. 56 , issue 7 , Part 1, pp. 3009 – 3017, 2008

[8] L. R. Rabiner, "Techniques for designing finite-duration impulse-response digital filters,"IEEE Trans.Communication Technology, vol. Com-19, Apr. 1971.

[9] B. Gold and K. L. Jordan, "A note on digital filter synthesis," Proc. IEEE (Lett.), vol. 56, pp. 1717-1718, Oct. 1968.

[10] H. B. Voelcker and E. E. Hartquist, "Digital filtering via block recursion," IEEE Trans.Audio Electroacoust., vol. AU-18, pp. 169-176, Jun. 1970.

[11] B. Gold and K. L. Jordan, "A direct search procedure for designing finite duration impulse response filters," IEEE Trans.Audio Electroacoust., vol. AU-17, pp. 33-36, Mar. 1969.

[12] L.R. Rabiner and K. Steiglitz, "The design of wide-band recursive and nonrecursive digital differentiators,"IEEE Trans.Audio Electroacoust., vol. AU-18, pp. 204-209. Jun. 1970.

[13] L. R. Rabiner, B. Gold, and C. A. McGonegal, "An approach to the approximation problem for nonrecursive digital filters," IEEE Trans. Audio Electroacoust., vol. AU-18. pp. 83-106, Jun. 1970.

[14] O. Herrmann and H. W. Schuessler, "On the design of selective nonrecursive digital filters," presented at the IEEE Arden House Workshop, Harriman, N. Y., Jan. 1970.

[15] O. Herrmann, "On the design on nonrecursive digital filters with linear phase," Electron. Lett., 1970.

[16] J. F. Kaiser, "Digital filters," in System Analysis by Digital Computer, F. F. Kuo and J. F. Kaiser, Eds. New York:Wiley, 1966.

[17] H. D. Helms, "Nonrecursive digital filters: design methods for achieving specifications on frequency response," IEEE Trans.Audio Electroacoust., vol. AU-16, pp. 336-342, Sept. 1968.

[18] T.A. Ramstad and T. Saramaki, "Multistage, multirate FIR filter structures for narrow transition-band filters," IEEE International Symposium on Circuits and Systems, 1990.

[19] Xilinx System Generator for DSP User Guide for ISE Design Suite 14.5, www.xilinx.com.

[20] G. Singh and N. R. Prakash, "FPGA Implementation of Higher Order FIR Filter," International Journal of Electrical and Computer Engineering (IJECE) Vol. 7, No. 3, August 2017, pp. 1874~1881 ISSN: 2088-8708, DOI: 10.11591/ijece.v7i3.pp1874-1881.

[21] X. Jiang and Y. Bao, "FIR filter design based on FPGA," International Conference on Computer Application and System Modeling (ICCASM 2010), pp. V13-621 - V13-624.

[22] H.T. Kung and C.E. Leiserson, "Systolic array for VLSI", Sparse Matrix Proc., pp. 256-282, 1979.

[23] F. Nekoei, Y.S. Kavian and O. Strobel, "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.