

# Novel Approach of Designing Multiplier-less Finite Impulse Response Filter using Differential Evolution Algorithm

Abhijit Chandra

Department of Electronics and Telecommunication Engineering  
Bengal Engineering and Science University, Shibpur, India  
Email: abhijit922@yahoo.co.in

Sudipta Chattopadhyay

Department of Electronics and Telecommunication Engineering  
Jadavpur University, Kolkata, India  
Email: sudiptachat@yahoo.com

*Abstract—Reduction of computational complexity of digital hardware has drawn the special attention of researchers in recent past. Proper emphasis is needed in this regard towards the settlement of computationally efficient as well as functionally competent design of digital systems. In this communication, we have made one novel attempt for designing multiplier-free Finite duration Impulse Response (FIR) digital filter using one robust evolutionary optimization technique, called Differential Evolution (DE). The search has been directed through two sequentially opposite paths which include quantization and optimization as fundamental operations. Besides performing a detailed comparative analysis between these two proposed approaches; the performance evaluation of the designed filter with other existing discrete coefficient FIR models has also been carried out. Finally, the optimum search method for realizing the required set of specifications has been suggested.*

**Index Terms—** Differential Evolution (DE), Finite duration Impulse Response (FIR) filter, Multiplier-less architecture, Sum of power of two (SPT) terms, Total power of two (TPT) terms, Zero-valued filter coefficient (ZFC)

## I. INTRODUCTION

Finite duration Impulse Response (FIR) filter has found enormous applications in the field of communication and signal processing for a number of encouraging features associated with it like phase-linearity, guaranteed stability [1,2] etc. However, the applicability of this type of filter is heavily challenged by its equivalent Infinite duration Impulse Response (IIR) structure because of the hardware efficient construction of the latter [3,4]. Therefore in order to make FIR filter hardware friendly, a number of research articles have been published in the past [5-9] and a good number of works are still being carried out.

Different approaches have been proposed in the literature for reducing the computational complexity of FIR filter. However, major emphasis has been put forward by quantizing each of its impulse response

coefficient by means of sum of power of two (SPT) terms, as far as the survey result of hardware efficient FIR filter design is concerned [10-12]. This special architecture enables the FIR filter to carry out the operation of multiplication only through the use of delay elements and adders and thus reduces the hardware complexity significantly.

The area of signal processing and mostly the design of digital filter is being currently influenced by a number of evolutionary computational mechanisms. The success of these schemes over orthodox optimization technique has already been firmly established in [13] which has inspired the researchers to think about intelligent optimization to be deployed in wide area of science and engineering. As a matter of fact, the techniques like Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Differential Evolution (DE) have been popularly used in designing FIR and IIR filter over the last few years or so [14-17]. Amongst them, DE has particularly developed unique interest amongst researchers because of its inherent simple yet effective construction along with its limited number of control parameters [18-22]. This has triggered the applicability of DE even if in most challenging scenarios.

Proper selection of filter hardware is necessary for building the designed system robust against any impairment. Therefore the optimization between the system complexity and achievable response is a matter of great concern to the system designers and thus demands for considerable attention. DE, being a powerful evolutionary optimization tool, combined with the concept of coefficient quantization has been jointly exploited in this paper to minimize the required hardware of FIR filter. The impact of filter order and total number of quantization levels on its response have been examined thoroughly by considering different values for those parameters. Finally, the proposed architecture has been designed using VHDL code and subsequently been compared with other discrete coefficient existing FIR models to prove the superiority of our algorithm.

The entire paper has been organized as follows: Section II briefly describes the theory behind DE with a special emphasis to each of its different steps. Section III demonstrates the proposed novel algorithm with significant mathematical explanations. Simulation results and critical analysis with other state-of-the-art techniques have been illustrated in Section IV, followed by a conclusion in Section V.

## II. THEORETICAL BACKGROUND OF DIFFERENTIAL EVOLUTION ALGORITHM

Differential Evolution (DE) algorithm has emerged as a very competitive form of evolutionary computing for more than a decade ago. The first written article on DE appeared as a technical report by R. Storn and K. V. Price in the year 1995 [19]. In quick times, DE has turned out to be one of the best evolutionary optimization algorithms for solving real valued test function. Afterwards, researchers have carried out a number of modifications on classical DE to come up with techniques like Fuzzy Adaptive DE (FADE) [23], Self Adaptive DE (SADE) [24], Opposition-based DE (ODE) [25], DE with global and local neighborhoods (DEGL) [26], JADE [27] and so on which prove themselves to be superior to conventional one in various respects.

DE is very simple yet robust, powerful, stochastic, population based and easy to use optimization algorithm, which has been developed to optimize real parameter and real valued functions. General problem formulation for DE is for an objective function  $f: R^D \rightarrow R$ ; the proper selection of  $x^* \in X$  such that  $f(x^*) \leq f(x)$  for all  $x \in X$ . The distinct idea behind DE is a new scheme for generating trial parameter vectors  $x$ . Therefore this algorithm can optimize any function with D real parameters for any positive integer number D. Before the execution of the algorithm, size of the population should be selected properly. Population size (P) does not change during the minimization process.

The name of the algorithm is chosen as Differential Evolution (DE) to signify a special type of differential operator in it that has been utilized to create new offspring from the parent chromosomes without adopting classical crossover or mutation. By performing an extensive test bed it has been found that this optimization technique converges at a faster rate than any other globally acclaimed optimization techniques [20-21]. DE can be implemented very easily using very few parameter tuning which makes the algorithm reasonably popular very soon.

Differential Evolution algorithms can be divided into four steps, namely Initialization, Mutation, Recombination, and Selection.

### A. Initialization

This step indicates the beginning of the search algorithm. Initially, the values of the parameter vector are chosen randomly in such a way that it can cover the entire parameter space. In case the distribution of random variable is unknown, uniform probability distribution for random variables is normally assumed without the loss of

any generality. For  $x_{i,j,1}$  denoting the  $i^{th}$  element of the  $j^{th}$  member of population at first iteration, its value must be within the upper and lower bound of the random variable and can be written as [18, 22]:

$$x^L < x_{i,j,1} < x^U \quad (1)$$

### B. Mutation

The step of mutation actually expands the search space by evolving offspring from the parents. However, unlike traditional Evolutionary Algorithms (EA), the DE variants perturb the population members of current generation with the scaled differences of randomly selected and distinct population members from the previous generation. Therefore no separate probability density function has to be used for generating the offspring [18].

Specifically, from the  $i^{th}$  member of the current population  $x_{i,G}$ , a mutant or donor vector  $v_{i,G+1}$  is created in the process of mutation. Depending upon the generation of the donor vector from the parameter (or target) vector, different variants of DE has been developed.

The general convention used in the literature is DE/p/q/r, where p represents a string denoting the vector to be perturbed, q is the number of difference vectors considered for perturbation of r stands for the types of crossover being used i.e. either exponential or binomial.

In the DE/rand/1 scheme the mutant vector of the next generation  $v_{i,G+1}$  is generated according to the following equation [18-20]:

$$v_i(G+1)_{rand/1} = x_{p_1}(G) + F \cdot [x_{p_2}(G) - x_{p_3}(G)] \quad (2)$$

Here F is called the Weighting Factor and running index 'i' is different from other three indices like  $p_1$ ,  $p_2$  and  $p_3$ . Consequently, DE scheme requires minimum 4 no of population for every iteration.

DE/rand to best/1 follows the same procedure as that of DE/rand/1. The only difference is that the donor vector  $v_{i,G+1}$  is created using any two randomly selected parameter vectors and the best member of the current generation. Mathematically, this can be outlined as follows [22, 25]:

$$\begin{aligned} v_i(G+1)_{randto\ best/1} &= x_i(G) + \lambda \cdot [x_{best}(G) - x_i(G)] \\ &+ F \cdot [x_{p_1}(G) - x_{p_2}(G)] \end{aligned} \quad (3)$$

For the scheme of DE/best/1 and DE/best/2, donor vector of any generation can be generated without using the corresponding parameter vector. The mutant vectors for these two schemes can be written respectively as [23-26]:

$$v_i(G+1)_{best/1} = x_{best}(G) + F \cdot [x_{p1}(G) - x_{p2}(G)] \quad (4)$$

$$v_i(G+1)_{best/2} = x_{best}(G) + F \cdot [x_{p1}(G) + x_{p2}(G) - x_{p3}(G) - x_{p4}(G)] \quad (5)$$

In (4) and (5),  $x_{best}(G)$  identifies the member from the  $G^{th}$  generation which shows the maximum fitness. The concerned population member 'i' and the four other members from the same generation i.e.  $p_1, p_2, p_3$  and  $p_4$  are entirely different from each other.

For the scheme of DE/rand/2, in order to construct donor vector, five such distinct parameter vectors and two Weighing Factors ( $F_1, F_2$ ) are needed. The equation governing the generation of mutant vector is having the form [18, 22]:

$$v_i(G+1)_{rand/2} = x_{p1}(G) + F_1 \cdot [x_{p2}(G) - x_{p3}(G)] + F_2 \cdot [x_{p4}(G) - x_{p5}(G)] \quad (6)$$

### C. Recombination

Recombination incorporates successful solutions from the previous generation. It plays a significant role to increase the potential diversity of the population member. At the end of this step, the trial vector  $u_{i,G+1}$  is developed from the elements of target vector  $x_{i,G}$  and the elements of donor vector  $v_{i,G+1}$ . Elements of donor vector enter the trial vector with probability CR. The trial vector can be constructed according to the following equation [19-21]:

$$u_{i,j}(G+1) = \begin{cases} v_{i,j}(G+1); & rand_{j,i} < CR \text{ or } j = I_{rand} \\ x_{i,j}(G); & rand_{j,i} > CR \text{ and } j \neq I_{rand} \end{cases} \quad (7)$$

In the above equation,  $rand_{j,i}$  is a random value in the range [0, 1] and  $I_{rand}$  is a random integer from [1, 2, ..., D]. CR denotes recombination or cross-over probability and certainly is in the range [0, 1]. This, in turn, makes sure that the trial vector  $u_{i,G+1}$  gets at least one parameter from the mutant vector  $v_{i,G+1}$ .

### D. Selection

This is the final step of the evolutionary algorithm. Comparing the values of trial vector  $u_{i,G}$  with the target vector  $x_{i,G}$  it can be decided whether or not the trial vector will be able to be a member of the next generation. This can be mathematically formulated as follows [25,26]:

$$x_i(G+1) = \begin{cases} u_i(G+1); & \mathfrak{Z}(u_i(G+1)) \leq \mathfrak{Z}(x_i(G+1)) \\ x_i(G); & \mathfrak{Z}(x_i(G)) > \mathfrak{Z}(x_i(G+1)) \end{cases} \quad (8)$$

Where  $\mathfrak{Z}(Y_i(G))$  identifies a cost function related to the problem of interest associated with the  $i^{th}$  vector  $Y_i(G)$  at iteration 'G'.

If the trial vector  $u_{i,G+1}$  yields smaller cost function than target vector  $x_{i,G}$  then the value of trial vector is assigned to  $x_{i,G+1}$ ; otherwise the old value of the target vector will be sustained. Mutation, Recombination and Selection process will continue until any termination criteria is reached. If the termination criteria can not be achieved even after the execution of maximum number of iterations, the member of the population having the least functional value is chosen as the optimized solution to the concerned problem.

## III. MATHEMATICAL FORMULATION OF THE PROPOSED APPROACH

This work actually deals with designing a discrete coefficient FIR filter under several constraints like maintaining the required specification of filter response as specified by user, using minimum number of hardware elements, minimizing the number of non-zero coefficient in the impulse response and so on. In connection to multiplier-less filter design, an L-length FIR filter with power of two coefficients may be mathematically represented as:

$$h^L(n) = \sum_{k=0}^{\Delta-1} c_{kn} 2^{-k}, \quad n = 0, 1, 2, \dots, L-1 \quad (9)$$

In the above equation, the term ' $\Delta$ ', also called word length (WL) of filter coefficient, gives a measure of the total number of quantization levels used during the process of quantization and  $c_{kn}$  is nothing but a binary mask for generating the discrete coefficients from a limited set of values available for construction. For a particular sample  $n_0$ , this may have the form as:

$$C_{n_0} = [c_{0,n_0} \ c_{1,n_0} \ c_{2,n_0} \ \dots \ c_{\Delta,n_0}]; \quad c_{k,n_0} \in \mathbb{B} = \{0, 1\}, \\ k \in \{0, 1, \dots, \Delta-1\} \text{ and } n_0 \in \{0, 1, \dots, L-1\} \quad (10)$$

Hence the proper assignment of individual  $c_{kn_0}$  is the key that governs the frequency characteristics as well as the hardware requirement of the FIR filter. More interestingly, these two issues are contradictory to each other and thus require thoughtful and logical considerations. For a fixed  $\Delta$ , there are  $2^\Delta$  numbers of all possible combinations of mask vectors in a set  $\mathbb{M}$ , all of which may or may not be required for constituting the final response. This necessitates a search for obtaining the most favorable collection of mask vectors  $\{\widehat{C}_{k,n}\} \forall k \in \{0, 1, 2, \dots, \Delta-1\}$  and  $n \in \{0, 1, \dots, L-1\}$  from the set  $\mathbb{M}$ .

In recent years, the approach for searching the most suitable vectors from a collection of potential vectors has been modified tremendously through the innovation of a number of artificially powerful mechanisms. Amongst them evolutionary computation demands for particular concentration because of its several attractive attributes. This study involves one of the most popularly employed

evolutionary techniques, named as Differential Evolution (DE), for optimizing the discrete FIR coefficients. By using general notation, this may simply be outlined as:

$$h_{opt}^L(n) = \mathcal{O}[\widetilde{C_{k,n}}] = \mathcal{O}[Q_k\{h^L(n)\}];$$

$$\forall k \in \{0,1,2, \dots, \Delta - 1\} \text{ and } n \in \{0,1, \dots, L - 1\}$$
(11)

where the symbols  $\mathcal{O}$  and  $Q$  signify the optimization and quantization operations respectively.

Interchanging the order of operations in (11) can sometimes be regarded as an alternative approach of having optimally selected discrete coefficient FIR filter. The necessary design rule has been outlined as:

$$h_{opt}^L(n) = Q_k[\mathcal{O}\{h^L(n)\}];$$

$$\forall k \in \{0,1,2, \dots, \Delta - 1\} \text{ and } n \in \{0,1, \dots, L - 1\}$$
(12)

One prominent difference can be pointed out between the two optimization approaches as described in the above equations, namely Approach 1 (Quantization prior to optimization) and Approach 2 (Optimization prior to quantization). In (11), optimization is performed on already quantized coefficients and thus through the process of optimization it has the effect of reducing the initial quantization error incurred at the very beginning. Therefore, the optimum outcome is expected to approximate the ideal or near-ideal solution in an accurate sense. However, approach 2 begins with optimizing the filter coefficient and is incorporating the quantization process at the final stage. Therefore it suffers from serious quantization error which has almost no chance to get minimized. On the other hand, approach 2 may provide some coefficient value less than  $2^{-(\Delta-1)}$  after carrying out the search mechanisms. After passing through a uniform quantizer, they would finally result in zero-valued coefficients and therefore reduce the hardware cost noticeably.

Nevertheless, both of these two procedures call for applying an optimization over a  $\Delta$  (for approach 1) or  $L$  (for approach 2) dimensional search space. The use of Differential Evolution (DE) optimization for this specific purpose additionally includes some fundamental steps, more commonly observed in any evolutionary programming, like initialization, mutation, cross-over or recombination and selection. The potential solution vector  $X_i^0 \forall i \in \mathbb{P}$  of the concerned problem is accepted in the process of initialization from a  $D$ -dimensional search space  $\mathfrak{R}^D$  where  $\mathbb{P}$  is the set of population members and  $D$  is either equal to  $\Delta$  or  $L$ , depending upon the selected approach.

The immediate next step of initialization is called mutation in which the entire search region is explored intensively. The mutant vector for the next generation  $Y_i^{G+1}$  is generated from the parameter vector  $X_i^G$  of the current generation. The mutant vector further undergoes through the process of cross-over (recombination) and selection to finally end up with the potential parameter vector for the next iteration. However, the creation of parameter vector for the next iteration from that of

present generation is largely monitored by the rules adopted during mutation, cross-over and selection. Thus for  $k^{th}$  mutation rule,  $l^{th}$  cross-over rule and  $m^{th}$  selection scheme; the formal mathematical notation of evolutionary optimization mechanisms can be described  $\forall i \in \mathbb{P}$  and  $\forall g \in \mathbb{G}$  as:

$$\mathfrak{M}_k[X_i^g] = Y_i^{g+1}$$
(13)

$$\mathfrak{C}_l[Y_i^{g+1}] = Z_i^{g+1}$$
(14)

$$\mathfrak{S}_m[X_i^g, Z_i^{g+1}] = X_i^{g+1}$$
(15)

Use of DE under scheme 1 would result in an optimally selected binary matrix  $\mathbb{C}_{opt}$  whose each row identifies the mask coefficient for a single coefficient of impulse response, as shown in the following equation:

$$\mathbb{C}_{opt} = [\mathbb{C}_0^* \mathbb{C}_1^* \mathbb{C}_2^* \dots \mathbb{C}_{L-1}^*]^T$$
(16)

,where the asterisks stands for optimization.

In (13), each of  $\mathbb{C}_i^* \forall i \in \{0,1,2, \dots, L - 1\}$  is a row vector having  $\Delta$  elements. However, optimization under scheme 2 only considers  $L$ -dimensional vector as its agents and therefore would result in an optimum solution as:

$$\mathbb{H}_{opt} = [h_0^* \ h_1^* \ h_2^* \ \dots \ h_{L-1}^*]$$
(17)

#### IV. RESULTS

In this work, a recent population based optimization algorithm has been used for the coefficient quantization of low-pass FIR filter. As any population-based strategy consumes considerable amount of time in locating the optimum solution, the researchers have placed their doubts in incorporating such methods in practical signal processing applications. This has been one of the biggest challenges for us particularly when it is applied for the purpose of reducing the computational complexity. The inclusion of DE in this specific problem of signal processing has been justified by showing the variation of averaged cost function with the number of iterations for two different approaches in Fig. 1, keeping the size of population fixed at 100.

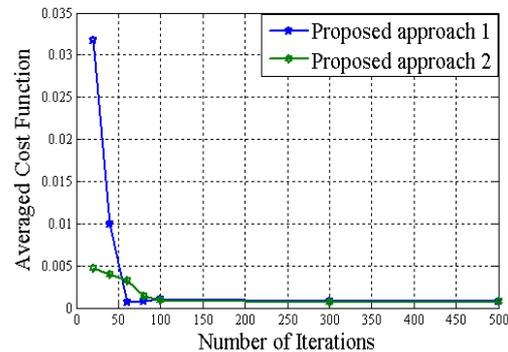


Figure 1. Convergence behavior of proposed approaches

From the plot of Figure 1, it can be clearly inferred that our proposed method does not necessitate the use for a

significantly large number of iterations to converge. More specifically, around 60 (for approach 1) and 100 (for approach 2) iterations are needed for locating the most favorable solution. Although, the initial cost function is comparatively large for approach 1, it diminishes abruptly beyond a certain number of generations. Therefore, the possibility of not finding the optimal solution through our proposed approaches does not arise at all.

After comparing the two approaches of discrete-coefficient FIR filter design by means of their computational time; the quality of the solution obtained by them is another important issue to be addressed. The resulting frequency response of the proposed multiplier-less FIR filter, achieved through two different approaches, has been plotted in Figure 2 for three distinct lengths of the filter. The word length of the impulse response coefficient has been kept at 8 to show the competency of our proposed algorithm to exhibit the required specification even with a smaller WL. The pass-band and stop-band edge frequency are located at 0.25 rad/pi and 0.5 rad/pi respectively and the allowable pass-band ripple and stop-band attenuation has been selected as 1 dB and 50 dB respectively in the entire design.

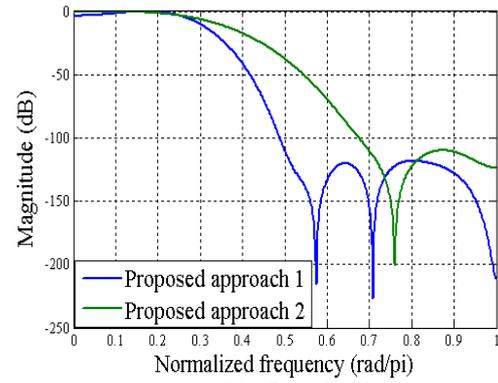
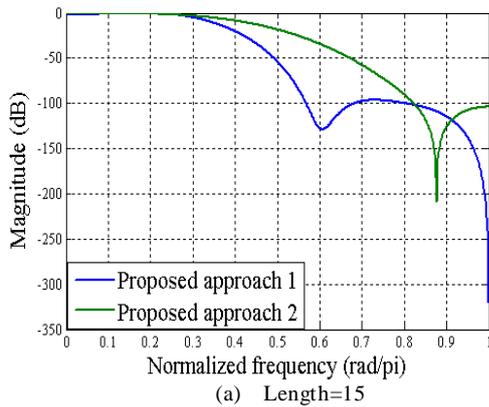


Figure 2. Frequency response of the designed filter (WL=8)

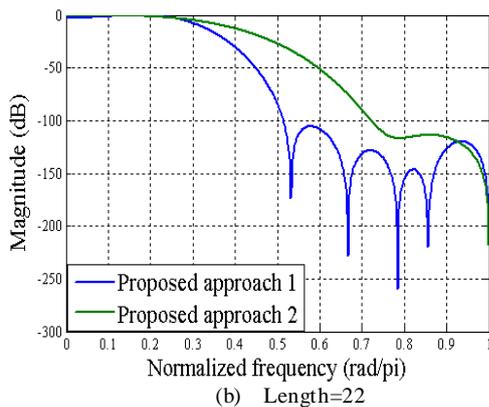
It can be unambiguously noticed that irrespective of the filter order, the proposed approach 1 which involves quantization prior to optimization would result in better frequency response, particularly in the transition and stop-band region, as compared to approach 2. Additionally, the pass-band responses of designed filter remain almost identical regardless of the sequence of selecting optimization and quantization. The numerical values of transition-band and stop-band attenuation has been listed in TABLE 1 at some distinct frequency points for three different length of multiplier-less low-pass FIR filters, considered in this paper.

TABLE 1. NUMERICAL VALUES OF ATTENUATION IN DB

Length		15		22		29	
Approach		1	2	1	2	1	2
Frequency points (rad/pi)	0.3	5.092	3.349	7.723	4.934	10.3	6.471
	0.4	20.19	8.633	30.24	12.72	40.93	17.32
	0.5	53.59	18.46	83.35	27.07	109.4	37.31
	0.6	127.4	34.32	108.2	51.09	133.3	69.58
	0.7	97.59	57.67	131.2	90.1	154	110.3
	0.8	99.96	90.38	155.1	116.5	118.3	122.3
	0.9	114.7	124.8	126.1	116	128	111.2
	1.0	319.7	103.9	207.6	217.8	210.7	123.6



(a) Length=15



(b) Length=22

The numerical entries of TABLE 1 firmly establish the superiority of approach 1 than approach 2 in designing low-pass FIR filter, as far as their transition and stop-band attenuation values are concerned. However, the selection of suitable filter order seems to be an essential issue since the choice of higher order filter always costs more hardware usage. In addition to this, for a particular filter order, the choice of word length plays an essential role in determining associated hardware cost. Higher the word length, more is the accuracy and thus demands more hardware elements. The impact of these two factors on the hardware complicity has been thoroughly studied in this work by considering three different lengths and four distinct word lengths for each of the three lengths examined in this paper.

Since the proposed multiplier-less FIR filter has been implemented by means of sum of power of two (SPT) terms; a number of performance parameter has been taken into account for calculating the associated hardware cost of the design. This includes total number of power of two terms (TPT), multiplier delay flip-flop (MDF),

multiplier adder (MA) and zero-valued filter coefficients (ZFC). The variation of these parameters with the word length of the FIR filter has been presented pictorially in Figure 3 through 6 for length 15, 22 and 29 of the designed filter respectively.

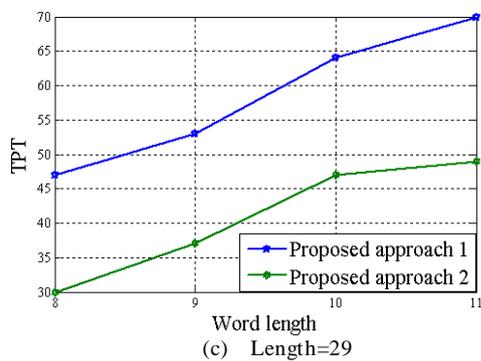
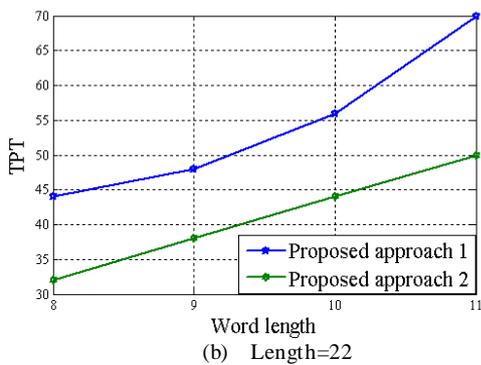
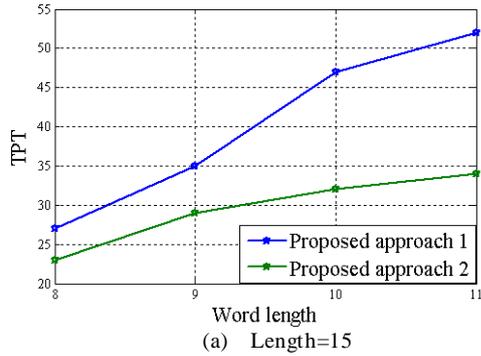


Figure 3. Variation of TPT with Word length of the proposed multiplier-less FIR filter

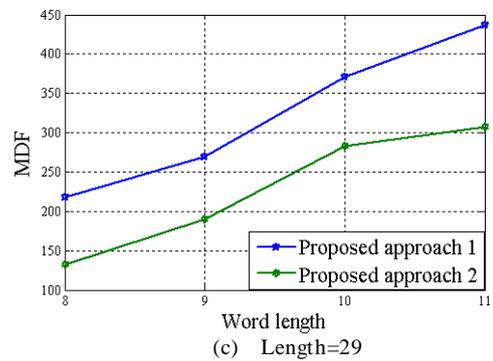
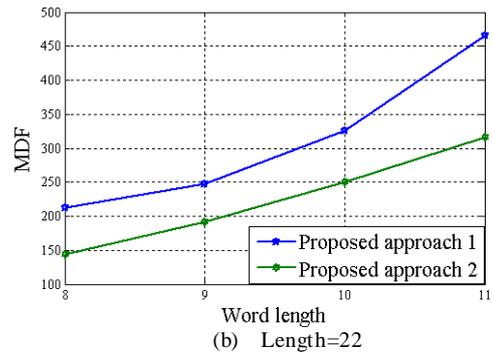
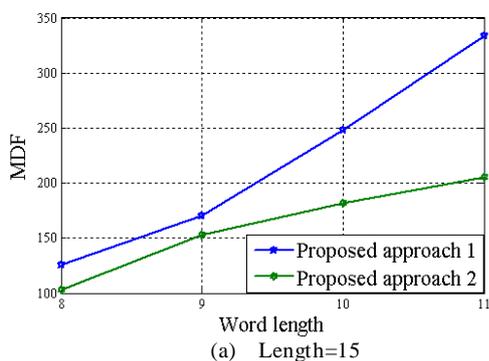
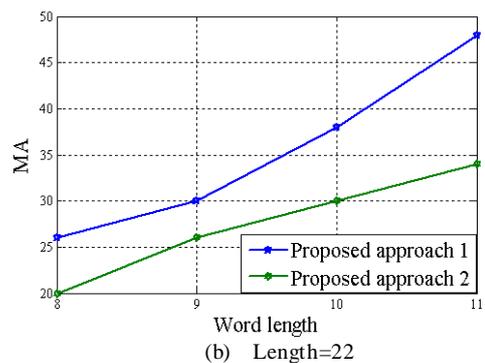
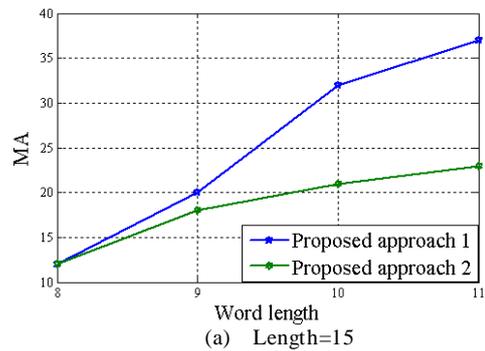


Figure 4. Variation of MDF with Word length of the proposed multiplier-less FIR filter



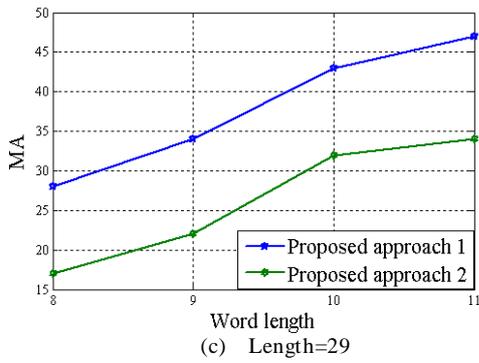


Figure 5. Variation of MA with Word length of the proposed multiplier-less FIR filter

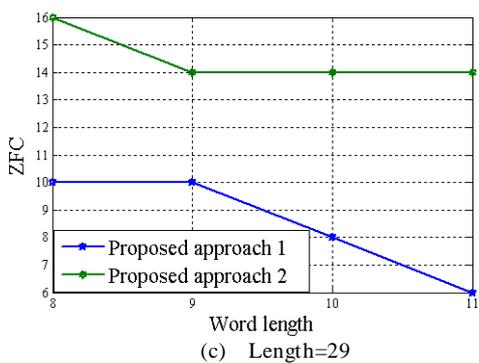
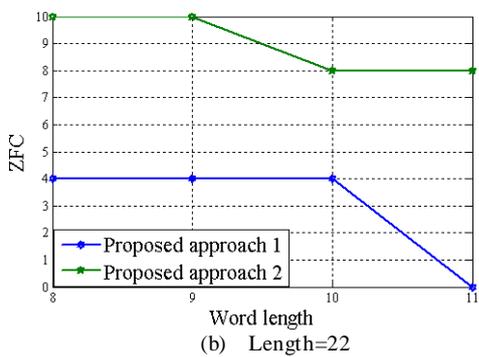
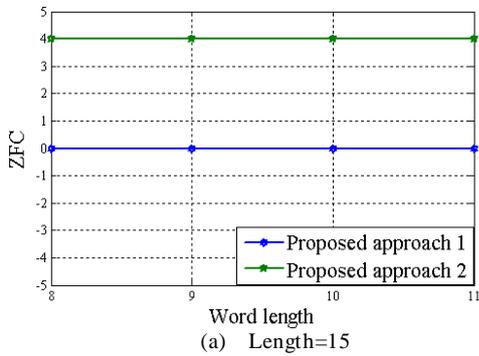


Figure 6. Variation of ZFC with Word length of the proposed multiplier-less FIR filter

Looking at the above diagrams, the supremacy of approach 2 can well be revealed as far as their design complexity is concerned. Figure 3 to 5 clearly indicate that approach 2 is always in need of fewer TPT, MDF

and MA when compared with approach 1 for any specific value of word length. Total number of zero-valued filter coefficient (ZFC) in the impulse response also serves as a useful measure in evaluating its performance for high speed application. As obvious, more ZFC implies less mathematical computations and thus supports the designed architecture to be employed in complicated circuits. Approach 2 continues to outperform approach 1 in connection to this parameter also; as it always incorporates more ZFC than our first approach by a considerably large margin.

While dealing with the design of hardware efficient multiplier-less FIR filter structure in this work, the requirement of different types of hardware blocks has been treated as a useful measure for comparing their architectures. Since the conventional multipliers of filter model have been eliminated by simpler structures in this particular design; the requirement of alternative hardware for multipliers like delay elements, full-adders and subtractors may become helpful in evaluating the hardware complicity of the designed filter. In order to critically analyze the system complexity from this viewpoint, the proposed filter along with the other existing multiplier-free FIR filters have been developed by means of hardware description language. The VHDL code has been synthesized using Xilinx Synthesis Technology (XST) with a product version of ISE 12.3 and executed in Intel(R) Core(TM) i3 processor with 2 GB RAM and 32-bit operating system. To be specific, the hardware cost of the proposed multiplier-less filter of length 22 and 29 have been subsequently been compared with that of other existing multiplier-free FIR models of same or approximately same length. The summary of the entire synthesis report has been listed in TABLE 2 and TABLE 3 respectively.

TABLE 2. COMPARATIVE ANALYSIS AMONGST VARIOUS MULTIPLIER-FREE FIR FILTERS IN TERMS OF HARDWARE COST AFTER REALIZATION THROUGH XST (LENGTH OF THE PROPOSED FILTER=22)

Method	Word length	I/O Buffer	Full adder	Subtractor	Delay
Kaakinen [8]	10	3	17	1	25
	9	3	14	1	27
Yu [9]	10	3	17	1	25
	11	3	20	1	26
Proposed approach 1	8	3	9	1	26
Proposed approach 2	8	3	9	1	23

TABLE 3. COMPARATIVE ANALYSIS AMONGST VARIOUS MULTIPLIER-FREE FIR FILTERS IN TERMS OF HARDWARE COST AFTER REALIZATION THROUGH XST (LENGTH OF THE PROPOSED FILTER=29)

Method	Word length	I/O Buffer	Full adder	Subtractor	Delay
Saramaki [5]	11	3	14	1	39
Yao [6]	13	3	18	1	38
Jheng [7]	12	3	17	1	39
Proposed approach 1	8	3	14	1	30
Proposed approach 2	8	3	11	1	27

Looking at the numerical entries in the above tables, a number of promising outcomes can be drawn. Critical analysis of these experimental findings reveal the hardware efficiency of the proposed design in a better way. It can be well observed that blocks like I/O buffers and subtractors remain fairly insensitive to the choice of the adopted method of designing hardware friendly FIR filter along with their specifications. However, the usage of full adder and delay elements seems to be a determining factor in selecting the suitable filter architecture. In connection to this, our approach of favoring optimization techniques like Differential Evolution (DE) in finding the optimal filter coefficients proves itself to be very much proficient than other state-of-the-art conventional approaches. Irrespective of the filter length, both of our approaches require less number of full adders and delay blocks than other promising findings available in literature. It has already been demonstrated previously in this article that our second approach proves to be more hardware efficient than the first one while the later exhibits better low-pass characteristics than the former. This inspection proves to be accurate after the realization of the designed filters through XST. Additionally, the significant contributions made by the above two tables lies in the fact that the process of quantization prior to optimization, i.e. approach 1, apart from showing better low-pass nature results in a FIR filter design which outperforms the other existing models in terms of hardware complexity as well. If further reduction in hardware seems to be essential, then one can think about incorporating approach 2 in this regard.

#### V. CONCLUSION

System designers often face problems in realizing complicated digital circuits as trade off between system complexity and achievable performance becomes inevitable. In this paper, two flexible approaches have been adopted by incorporating Differential Evolution algorithm for designing multiplier-less low-pass FIR filter. Experimental results reveal the pre-eminence of the proposed design strategies over the existing techniques from two different angles. Both of the proposed methods for designing multiplier-free FIR filter proves themselves to be superior than many existing well-recognized

approach of such design. In addition to this, the proposed technique imposes a flexibility between obtainable response and necessary hardware requirement and therefore emerges as a challenging solution in the area of multiplier-less FIR filter design.

#### REFERENCES

- [1] Mitra S. K. Digital Signal Processing: A Computer-based Approach. McGraw-Hill, 2001.
- [2] Somanathan Nair, B. Digital Signal Processing: Theory, Analysis and Digital-filter Design. Prentice-Hall, 2004.
- [3] Antoniou, A. Digital Filters: Analysis, Design and Applications. McGraw-Hill, 2001.
- [4] Tan, L. Digital Signal Processing: Fundamentals and Applications. Academic Press, 2008.
- [5] Kaakinen, J. Y., Saramaki, T. A Systematic Algorithm for the Design of Multiplierless FIR Filters. Proceedings of the 2001 IEEE International Symposium on Circuits and Systems, 2001, 2: 185-188.
- [6] Yao, C. Y. A Study of SPT-term Distribution of CSD Numbers and Its Application for Designing Fixed-point Linear Phase FIR Filters. Proceedings of the 2001 IEEE Symposium on Circuits and Systems, Australia: Sydney, 2001, 2: 301-304.
- [7] Jheng, K., Jou, S., Wu, A. A Design Flow for Multiplierless Linear-Phase FIR Filters: from System Specification to Verilog Code. Proceedings of the 2004 IEEE International Symposium on Circuits and Systems, 2004, 5: 293-296.
- [8] Kaakinen, J., Saramaki, T. Bregovic, R. An Algorithm for the Design of Multiplier-less Two-channel Perfect Reconstruction Orthogonal Lattice Filter Banks. Proceedings of the first International Symposium on Control, Communications and Signal Processing, 2004: 415-418.
- [9] Yu, Y. J., Lim, Y. C. Signed Power-of-two Allocation Scheme for the Design of Lattice Orthogonal Filter Banks. Proceedings of the 2005 IEEE International Symposium on Circuits and Systems, Singapore: Nanyang Technological University, 2005, 2: 1819-1822.
- [10] Li, D., Song, J., Lim, Y. C. A Polynomial-Time Algorithm for Designing Digital Filters with Power-of-Two Coefficients. Proceedings of the IEEE International Symposium on Circuits and Systems, 1993, 1: 84-87.
- [11] Lim, Y. C., Parker, S. R. FIR Filter Design over a Discrete Powers-of-Two Coefficient Space. IEEE Transactions on Acoustic, Speech, Signal Processing, 1983, ASSP-31: 583-591.
- [12] Chen, C., Willson, A. N. A Trellis Search Algorithm for the Design of FIR Filters with Signed-Powers-of-Two Coefficients. IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, 1999, 46(1): 29-39.
- [13] Chattopadhyay, S., Sanyal, S. K., Chandra, A. Design of FIR Pulse-Shaping Filter: Superiority of Differential Evolution Optimization over Convex Optimization. Proceedings of the fifth European Conference on Circuits and Systems for Communication, Serbia: Belgrade, 2010: 189-192.
- [14] Suckley, D. Genetic Algorithm in the Design of FIR Filters. IEE Proceedings of Circuits, Devices and Systems, 1991, 138 (2): 234:238.
- [15] Najjarzadeh, M., Ayatollahi, A. FIR Digital Filter Design: Particle Swarm Optimization Using LMS and Minimax

- Strategies. Proceedings of the 2008 IEEE International Symposium on Signal Processing and Information Technology, Sarajevo, 2008: 129-132.
- [16] Luitel, B., Engelbrecht, A. P. Differential Evolution Particle Swarm Optimization for Digital Filter Design. Proceedings of the IEEE World Congress on Computational Intelligence, 2008: 3954-3961.
- [17] Karaboga, N. Digital Filter Design using Differential Evolution Algorithm. EURASIP Journal of Applied Signal Processing, 8, 2005: 1269-1276.
- [18] Das, S., Abraham, A., Konar, A. Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives, Studies in Computational Intelligence (SCI), 2008, 116: 1-38.
- [19] Storn, R., Price, K. Differential Evolution—A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. International Computer Science Institute, Berkeley, 1995: TR-95-012.
- [20] Storn, R., Price, K. Differential Evolution—A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces. Journal of Global Optimization, 1997, 11(4): 341-359.
- [21] Storn, R., Price, K. Lampinen, J. Differential Evolution- A Practical Approach to Global Optimization. Springer, 2005.
- [22] Das, S., Suganthan, P. N. Differential Evolution: A Survey of the State-of-the-art. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4-31.
- [23] Liu, J., Lampinen, J. A Fuzzy Adaptive Differential Evolution Algorithm. Proceedings of the 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering (TENCON'02), Beijing, 2002: 606-611.
- [24] Brest, J., Greiner, S., Boskovic, B., Mernik, M., Zumer, V. Self-Adapting Control Parameters in Differential Evolution: A Comparative Study on Numerical Benchmark Problems. IEEE Transactions on Evolutionary Computation, 2006, 10(6): 646-657.
- [25] Rahnamayan, S., Tizhoosh, H. R., Salama, M. M. A. Opposition-based Differential Evolution. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 64-79.
- [26] Das, S., Abraham, A., Chakraborty, U. K., Konar, A. Differential Evolution with a Neighborhood Based Mutation Operator. IEEE Transactions on Evolutionary Computation, 2009, 13(3): 526-553.
- [27] Zhang, J., Sanderson, A. C. JADE: Adaptive Differential Evolution with Optional External Archive. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 945-958.

**Abhijit Chandra** received his B.E. in Electronics & Telecommunication Engineering in 2008 from Bengal Engineering & Science University, Shibpur, Howrah-711 103, India and M.E.Tel.E with specialization in Communication Engineering in 2010 from Jadavpur University, Kolkata-700 032, India.

Presently, he is working as an Assistant Professor in the Department of Electronics & Telecommunication Engineering of Bengal Engineering & Science University, Shibpur. He has published a number of papers in International/National Conferences and Journals of good repute. His research area includes Application of Evolutionary Optimization in Digital Signal Processing,

Design of Hardware Efficient FIR Filter, Digital/Mobile Communication Systems, Information and Coding Theory.

Mr. Chandra has received President's Gold Medal from Bengal Engineering & Science University, Shibpur in 2010 for being First in the faculty of Engineering and Technology at B.E. examination 2008. He has also received University Medal in 2010 for standing First in order of merit at the M.E. Tel. E. examination 2010 from Jadavpur University.

**Sudipta Chattopadhyay** received her B. Tech in Instrumentation Engineering in 1994 from Calcutta University, Kolkata- 700 009, India and M.E.Tel.E in 2001 from Jadavpur University, Kolkata – 700 032, India.

She was a Lecturer in the Department of Electronics and Communication Engineering at Institute of Technical Education and Research, Bhubaneswar, India, from 1996-2001 and also worked as Lecturer, Sr. Lecturer and Asst. Professor in the Department of Electronics and Communication Engineering at Netaji Subash Engineering College, Kolkata – 700 152, India, from 2001-2006. She is now working as an Assistant Professor in the Department of Electronics and Telecommunication Engineering, Jadavpur University, Kolkata – 700 032, India since 2006. She has published a number of papers in International/National Conferences. Her current research interests include Digital/Mobile Communication, Coding Theory and Digital Signal Processing.

Mrs. Chattopadhyay is a member of IEEE since last 10 years. She is also an Executive Committee member of the Affinity Group "Women in Engineering" under IEEE Calcutta Section.

**How to cite this paper:** Abhijit Chandra, Sudipta Chattopadhyay, "Novel Approach of Designing Multiplier-less Finite Impulse Response Filter using Differential Evolution Algorithm", International Journal of Intelligent Systems and Applications (IJISA), vol.4, no.4, pp.54-62, 2012. DOI: 10.5815/ijisa.2012.04.08