Scientific Journal Impact Factor (SJIF): 1.711
International Journal of Modern Trends in Engineering and Research
IJMTER
ISSN (ONLINE) : 2349 - 9745
ISSN (PRINT) : 2393 - 8161

A Genetic & Nelder Mead Coefficient Optimization For Low Power Realization Of FIR Filter

Neha Daniel[1], Chandrahas Sahu[2]
[1,2]*Electronics & Telecommunication Engineering Department, SSCET*

Abstract—In this paper a new meta-heuristic method called Genetic-Nelder Mead algorithm is applied to determine the best optimal impulse response coefficients of FIR low pass, trying to meet the respective ideal frequency response characteristics. These techniques reduce switching activity which is directly related to the power consumption of a circuit. So as to reduce the power dissipation of its implementation on a programmable DSP. The sources of power dissipation show that the power dissipation depends on the total hamming distance between the successive coefficient values. Experimental results on FIR filter examples show that the meta-heuristic algorithms result in up to 51% reduction in the total Hamming distance. This directly translates into reduction in the power dissipation in the coefficient memory data bus and the multiplier.

Keywords—FIR, Low pass filter, Coefficient Optimization, GA, Nelder Mead Algorithm, HD

I. INTRODUCTION

The Finite Impulse Response (FIR) filter are a type of digital filter that are much superior in the level of performance .Finite impulse response and are widely used in signal processing and communication system in applications like noise reduction, channel equalization, echo cancellation, image enhancement, speech and waveform synthesis .With the tense progress of wireless communication system and movable devices, the power Reduction has become a major issue. In applications, such as communication systems and portable storage device, power dissipation is the major problem these days, so as to provide low power consumption and hence longer battery life time is a must. With the rapid development of internet and information on demand, the wireless terminals are becoming more and more popular. With limited energy in a reasonable size battery, lowest power dissipation in digital communication devices is necessary. Many of the communication system today use digital signal processors (DSP) to resolve the spread information. The Finite impulse response (FIR) filters are very important structure blocks in many digital processing systems (DSP).

Signal switching activity is a major factor of power dissipation in CMOS circuits. The FIR algorithm is then mapped in to the architecture of MAC instruction. During the execution of FIR algorithm, the coefficient values directly influence the signal switching activity. The transition densities of multiplier input are influenced by the hamming distance between the successive filter coefficient values. The coefficients are optimized so as to minimize the signal switching activity and thus we can reduce the power dissipation.

The Nelder–Mead algorithm is a simplex local search method et al. [7]. If any point of a simplex is taken as the origin, then remaining other points will define vectored directions. Through a sequence of elementary geometric transformation (reflection, contraction, expansion and multi-contraction), the primary simplex moves will expands or contrasts. To select appropriate transformation, the method only uses the values of the function to be improved at the vertices of the simplex method considered. After each revolution, the existing worst vertex is replaced by a better one. The Nelder-mead simplex method is a very standard derivative free method for finding a local minimum of a function. The worse vertices where f(x,y) is largest, is rejected and replaced with a new vertices. A new triangle is formed and the search is continued. The process generates a sequence of triangles, which has the ability of different shapes for which the function values at the vertices get smaller and smaller .A simplex method succeeds in obtaining a good fall in the function value using a quite small number of function evaluations.

Genetic algorithms can implemented as a simulation in which a population of abstract representations (called chromosomes or genome) of aspirant solutions (called individuals, creatures, or phenotypes) to an optimization problem develops toward better results. Traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible. The growth generally starts from a population of randomly generated individuals and happens in generations. In each group, the fitness of every single in the population is calculated, multiple individuals are stochastically nominated from the recent population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a fresh population. The fresh population is then used in the next iteration of the algorithm. Commonly, the algorithm dismisses when either a maximum number of generations have been produced, or a satisfactory fitness level has been reached for the population. It can be seen that evolutionary algorithms differ expressively from more traditional search and optimization methods.

In this paper a different meta-heuristic method called Genetic-Nelder mead algorithm is applied to determine the best optimal impulse response coefficients of FIR low pass, trying to meet the respective ideal frequency response characteristics. If we use merely a Genetic Algorithm then we must face the problem of low accuracy. If we use only Nelder-Mead, then we face the problem of the possible convergence to a local (not to the global) minimum. These disadvantages will be removed in the case of our Hybrid method that combines Genetic Algorithm with Nelder-Mead method .so as to reduce the power dissipation of its implementation on a programmable DSP. The sources of power consumption show that the power consumption depends on the total hamming distance between the successive coefficients values. Moreover, if we can reduce the small percentage of switching activity between the coefficients, slight change will be there in the power consumption.

II. FIR (Finite Impulse Response)

A Finite Impulse Response (FIR) digital filter is one whose impulse response is of finite duration. The impulse response is "finite" because there is no feedback in the filter. If we put in an impulse (that is, a single "1" sample followed by many "0" samples), zeroes will eventually come out after the "1" sample has made its way in the delay line past all the coefficients.

FIR (Finite Impulse Response) filters are implemented using a finite number "n" delay taps on a delay line and "n" computation coefficients to compute the algorithm (filter) function. The above structure is non-recursive, a repetitive delay-and-add format, and is most often used to produce FIR filters. This structure depends upon each sample of new and present value data. The number of taps (delays) and values of the computation coefficients ($h_0,h_1,\ldots..h_n$) are selected to "weight" the data being shifted down the delay line to create the desired amplitude response of the filter. In this configuration, there are no feedback paths to cause instability. The calculation of coefficients is not constrained to particular values and can be used to implement filter functions that do not have a linear system equivalent. More taps increase the steepness of the filter roll-off while increasing calculation time (delay) and for high order filters, limiting bandwidth.

2.1 Advantages of FIR over IIR Filter

FIR filters have the following advantages over the IIR filters

1. FIR filters are linear phase filters, which is useful in speech processing.
2. FIR filters are always stable because all the poles are within the unit circle.

3. The designing methods are generally linear for FIR filters.
4. The start-up transitions have finite duration in FIR filters.

III. DESIGNING TECHNIQUES OF FIR FILTER

There are essentially three well-known methods for FIR filter design namely:

(1) The window method
(2) The frequency sampling technique
(3) Optimal filter design methods

3.1 Optimal Design of FIR Filter using Genetic Algorithm

The genetic algorithm loops over an iteration process to make the population evolve [10]. Each consists of the following steps:

1. SELECTION: The first step consists in selecting individuals for reproduction. This selection is done randomly with a probability depending on the relative fitness of the individuals so that best ones are often chosen for reproduction than poor ones.
2. REPRODUCTION: In the second step, offspring are bred by the selected individuals. For generating new chromosomes, the algorithm can use both recombination and mutation.
3. EVALUATION: Then the fitness of the new chromosomes is evaluated
4. REPLACEMENT: During the last step, individuals from the old population are killed and replaced by the new ones.

The algorithm is stopped when the population converges toward the optimal solution.

IV. POWER DISSIPATION FACTORS AND MEASURES

Each step in FIR filter algorithm involves getting the appropriate coefficient and data value and performing multiply-accumulate computation. Thus address and data buses of both the memories and multiplier see experiences the most signal switching activity during FIR filtering. Hence these form the main sources of power dissipation.

4.1. MEASURES OF POWER DISSIPATION IN BUSES

Signal switching activity is the major component of power dissipation in CMOS circuits. The power dissipation depends both on the frequency of switching and capacitive loading of the signal. For a typical embedded processor, address and data buses are networks with a large capacitive loading. Hence signal in these networks has a significant impact on power consumption. In addition to the capacitance of each signal, inter signal capacitance also contributes to bus power dissipation. The power dissipation due to inter signal capacitance varies depending upon the adjacent signal values. The current required for signals to switch between 5's (0101) and A's (1010) is about 25% more than the current required for the signals to switch between 0's (0000) and FFFF.

4.2 MEASURES OF POWER DISSIPATION IN MULTIPLIER

Due to high speed requirements, parallel array architecture are used for implementing dedicated multiplier in programmable DSPs .The power dissipation of the multiplier is directly proportional to the number of switching at all the internal nodes of the multiplier. The number of internal node switching depends on the multiplier input values. This dependence can be analyzed using the

Volume 02, Issue 11, [November – 2015] ISSN (Online):2349–9745 ; ISSN (Print):2393-8161

measure of circuit activity the transition density. Transition density of a signal is the average number of transition depends on the transition densities and the probabilities of the multiplier inputs. The transition density of the multiplier inputs depends upon the Hamming distance between successive input values. The input signal probability depends upon the number of 1s in the input values of the multiplier. These two thus forms the measures of multipliers power dissipation. It can also be shown that the transitions in least significant (LSBs) of the multiplier input contribute more to the power dissipation than the most significant bits (LSBs) of the multiplier input contribute more to the power dissipation than the most significant bits (MSBs). Thus while minimizing transition densities of all the input is important, larger gains are achieved by focusing on lower order bits of the input signal.

4.3 POWER REDUCTION IN DATA BUSES AND MULTIPLIERS

During FIR filtering, the coefficient and data memory data buses provide successive coefficients and data values for the weighted sum computation. The power dissipation in the coefficient memory data bus hence depends on the successive coefficient values and the power dissipation in the data memory data bus depends upon the successive data values. Since the data memory value forms the input to the FIR filter, the data memory values forms the inputs to the FIR filter, the data bus power dissipation cannot be controlled during FIR filter synthesis. The coefficient
memory data bus power however can very much be minimized by optimizing the filter coefficient so to reduce the Hamming distance between the successive coefficients values and also by reducing the total number of signal toggling in opposite directions between successive coefficients. The coefficients and the input data samples form the inputs to the multiplier during FIR filtering. The multiplier power dissipation thus depends upon the number of toggles and also on the number of 1s in these inputs. The coefficient optimization for reducing the coefficient memory data bus power thus also reduces the multiplier power. Higher power reduction can be achieved by focusing on the lower significant bits of the coefficients during minimization.

4.4 ALGORITHM FOR HAMMING DISTANCE MINIMIZATION

Step1:- For a given FIR filter coefficients A[i] (i = 1, N-1) and given pass band ripples (*Pdb_ req)* and stop band attenuation (*S*. Calculate the Hamming Distance between(S *db _ req*)
A[i], A[i-1] and A[i], A[i+1]

Step 2:- Now perturb each coefficient (increase the value of each coefficient one by one by 1) and calculate new hamming distance between the coefficients. A[i+], A[i-1] and A[i+], A[i+1] Such that HD(A[i], A[i-1]) +HD (A[i], A[i+1])> HD(A[i+], A[i-1])+HD(A[i+], A[i+1]) And Euclidian distance (A[i+], A[i+1]) is minimum.

Step 3:- Replace A[i+],and A[i+1] to get a new set of coefficients .

Step 4:- Compute pass band ripples (*Pdbi+)* and stop band attenuation (*Sdbi+)*from a new set of coefficient A[i+]

Step 5:- If pass band ripples (*Pdbi*+)< $P_{db\text{-}req}$ and stop band attenuation (*sdbi+)*> *sdb_req* calculate tolerance

$T = (pdb\_req - pdbi)/pdb\_req + (sdb\_req - sdbi)/sdb\_req$

Else

$T = 0$

Step 6:- Now again perturb each coefficient (decrease the value of each coefficient one by one 1) and calculate new hamming distance between the coefficients.

A[i-], A[i-1] and A[i-], A[i+1] Such that

HD(A[i], A[i-1]) +HD (A[i], A[i+1])> HD(A[i-], A[i-1])+HD(A[i-], A[i+1]) And
Euclidian distance (A[i-], A[i+1]) is minimum

Step7:- Replace A[i] with A[i-] to get a new set of coefficients.

Step8:- Compute pass band ripple (*Pdbi* and stop band attenuation (*Sdbi* from a new set of(*dbi-*) coefficient A[i].

Step9:- If pass band ripples pdbi-< *Pdb_req* and stop band attenuation *Sdbi*-> *Sdb_req* Calculate tolerance T =(p*dbreq-p dbi-*)/ *dpbreq* (s *dbi-* sdbreq)/s *dbreq*

Else

Toli- = 0

Step10:- Calculate gain function g for new coefficient values *A[i+] and [i-]*

g = (Tolerance * HD reduction) is maximum

for g >0

Replace original coefficients with new value

4.5 ALGORITHM FOR HAMMING DISTANCE MINIMIZATION USING GA & NELDER

Step1 Compute filter coefficients *h1(n)* and freq. response *H* (ω) of ideal FIR filter for

0 < =*n* <= *N* -1.

Step 2 Calculate the Hamming distance (*HD)* between the FIR filter coefficients *h*1 (*n*)

Step 3 Set the Number of chromosomes *(k)*, mutation rate *(m)*, Cross over rate *(c)*, Stopping criteria.

Step 4 Populate *k* sets of possible designed solutions, to produce symmetric coefficients *H*

and 0 <= *n* <= *N* -1

H*D* (n)0 <= *i* <= *k* - 1

Step 5 Compute the frequency response of the coefficients chromosomes for *H* (ω i*D*)population.

Step 6 Calculate the Hamming distance in each of the coefficient chromosome.

Step 7 Evaluate the fitness of the chromosomes

f (*i*) = *f* + *f*

Step 8 Apply Roulette wheel selection.

Step 9 Apply crossover operators at a desired rate.

Step 10 Mutate at a desired rate

Step 11 Evaluate again the fitness of the chromosomes.

Step 12 If the Stopping criterion is met store the chromosomes according to the fitness, else go to stop 8.Flow chart for hamming distance minimization using G.A. is given in the next section

International Journal of Modern Trends in Engineering and Research (IJMTER)
Volume 02, Issue 11, [November – 2015] **ISSN (Online):2349–9745 ; ISSN (Print):2393-8161**

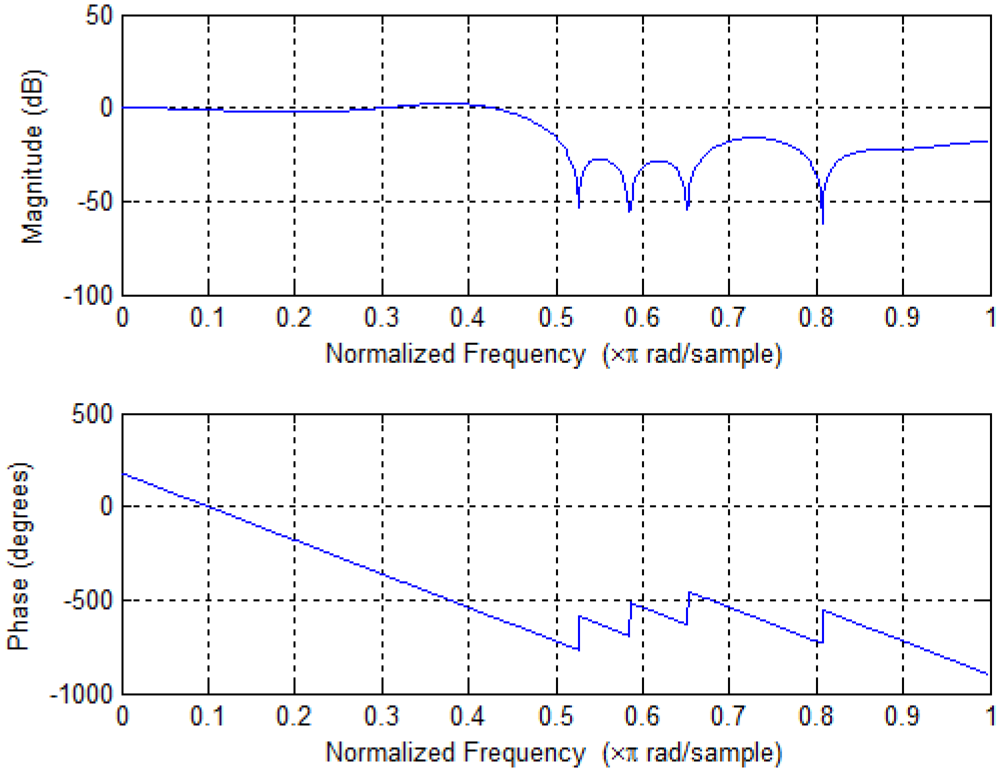
V. RESULTS

- The low pass filter of oreder 24 , there will be 36%reduction in total hamming distance

FIR Filter	Initial HD	Steepest Descent HD	% Reduction HD
Lp_16K_3K_4K_.1_42_24	180	118	36.71%

Example 1

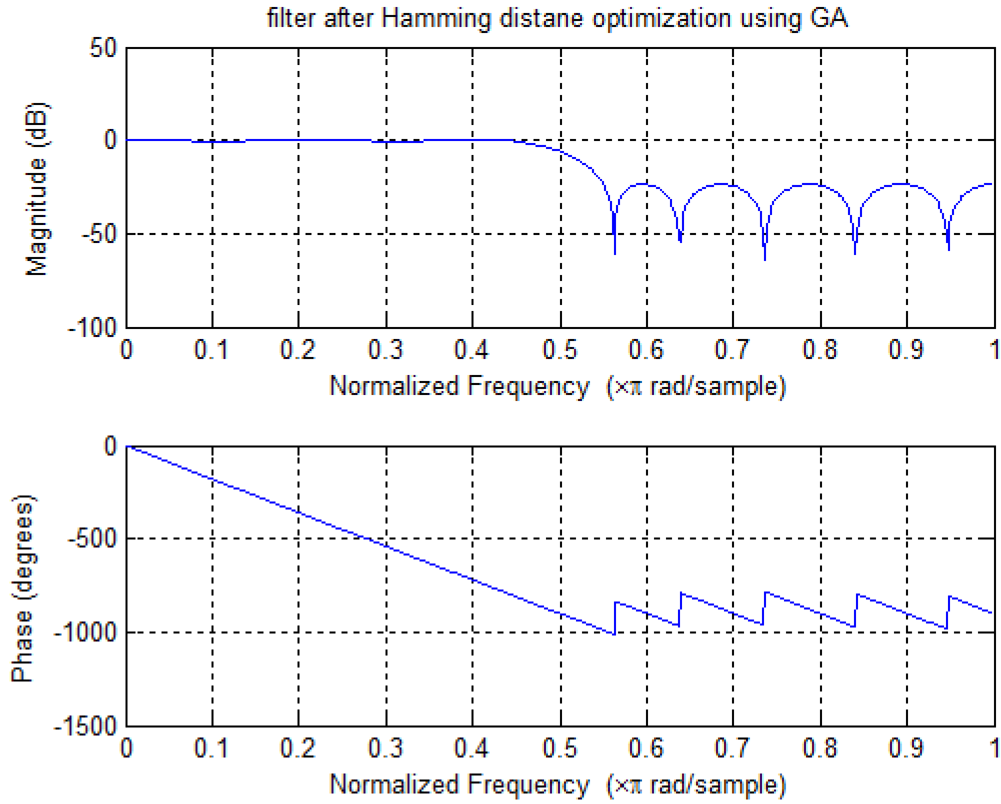
- The magnitude and frequency of low pass FIR filter without optimization for example 1



- Result of FIR filters in terms of percentage Hamming distance using Genetic Algorithm Approach and Nelder Mead algorithm are summarized in a table given below

FIR Filters	Initial HD	GA & NM HD	% Reduction Hamming Distance
Lp_16K_3K_4K_.1_62_20	252	94.5	51.02%

The magnitude and frequency response



REFRENCES

[1] Khaled Grati,1 Nadia Khouja "Power Consumption Models for Decimation FIR Filters in Multistandard Receivers" Department and graduate institute of electronics engineering , Hindawi Publishing Corporation VLSI Design Volume 2, 15 March 2012

[2] Mahesh Mehendale S.D. Sherlekar, Venkatesh "COEFFICIENT OPTIMIZATION FOR LOWPOWER REALIZATION OF FIR FILTERS" 1995 IEEE Indian Institute of Technology, Bombay

[3] N. Sankarayya, Student Member, IEEE, Kaushik Roy, Senior Member, IEEE, and Debashis Bhattacharya, Member, IEEE "Algorithms for Low Power and High Speed FIR Filter Realization Using Differential Coefficients AND SYSTEMS" IEEE TRANSACTIONS ON CIRCUITS —II: ANALOG AND DIGITAL SIGNAL PROCESSING, VOL. 44, NO. 6, JUNE 1997

[4] Subhankar Bhattacharjeea, SanjibSilb, AmlanChakrabartic 'Evaluation of Power Efficient FIR Filter for FPGA Based DSP" Applications International Conference on Computational Intelligence: Modeling Techniques and Application CIMTA) 2013

[5] Jeffrey T. Ludwig, S . Hamid Nawab, and Anantha P. Chandrakasan "Low-Power Digital Filtering Using Approximate Processing" IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. 31, NO. 3, MARCH 1996

[6] Shahnam Mirzaei, Anup Hosangadi, Ryan Kastner "FPGA Implementation of High Speed FIR Filters Using Add and Shift Method" University Of California

[7] D. Kodek and K. Steiglitz, "Comparison of Optimal and Local Search Methods for Designing Finite Word length FIR Digital Filters", IEEE Transactions on Circuits and Systems, January 1981