IOSR Journal of Engineering (IOSRJEN)
ISSN: 2250-3021 Volume 2, Issue 8 (August 2012), PP 12-18
www.iosrjen.org

Low Power Realization of FIR Filters Using Optimization Techniques

Er. Neha Goyal[1], Er. Jyotisana Rishi[2]
[1](*Electronics & Communication Engineering Department, B.G.I.E.T,Sangrur, India*)
[2](*Electronics & Communication Engineering Department, B.G.I.E.T, Sangrur, India*)

ABSTRACT: - *In this paper an algorithm for optimizing coefficients of a Finite Impulse Response (FIR) filter, so as to reduce power dissipation of its implementation on a programmable Digital Signal Processor is presented. We then present an algorithm that optimizes coefficients so as to minimize the Hamming distance and signal toggling. Two such techniques 'Steepest descent' and 'Genetic Algorithm' are presented to minimize these measures of power dissipation .Experimental results on a FIR filter example show that the Genetic coefficient optimization algorithm results is best in reduction of the total Hamming distance and total number of signal toggles than Steepest descent.*
Keywords: *Finite Impulse Response (FIR), Digital Signal Processor (DSP)*

I. INTRODUCTION

With the recent trend towards portable computing and wireless communication systems, power dissipation has become an important design consideration. These systems require high speed computation, complex functionalities, real-time processing capabilities and are often built around embedded processors (such as DSPs). FIR filters are one of the most common components of Digital Signal Processing applications. FIR filtering is achieved by convolving the input data samples with the desired unit impulse response of the filter. Output Y of an N-tap FIR filter is given by the weighted sum of latest N input data samples.

$$\mathbf{Y} = \sum_{i=0}^{n-1} (A_i * X_{n,-i})$$

where X's are the input data samples.

Most programmable DSP architectures provide features for efficient computation of weighted sums. These include a dedicated hardware multiplier and two (or more) separate memory spaces that can be accessed simultaneously [1]. This enables single-cycle execution of the multiply-accumulate (MAC) operation. A generic representation of such an architecture in shown in figure 1 .The two memory spaces can be used to store the Coefficients and the input data samples. The FIR algorithm can then be mapped onto this architecture as a series of MAC instructions. It can be noted that during the execution of the FIR algorithm, the coefficient values directly impact the signal switching activity, especially in the coefficient memory data bus and the multiplier. The coefficients can be optimized so as to reduce this signal switching activity and thus reduce power dissipation.

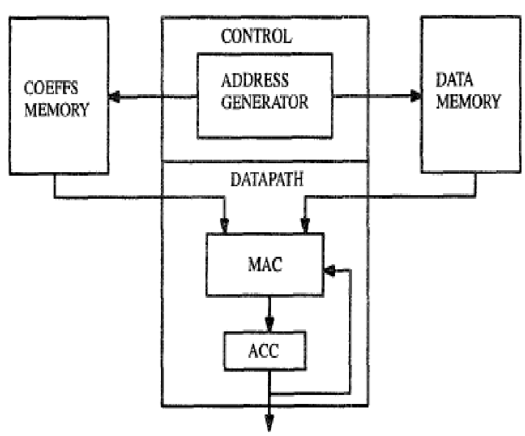


Figure1.GenericDSPArchitecture

II. FIR POWER DISSIPATION—SOURCES AND MEASURES

2.1. Components Contributing to Power Dissipation

Each step in the FIR filtering algorithm involves getting the appropriate coefficient and data values and performing a multiply-accumulate computation. Thus address and data buses of both the memories and the multiplier-adder datapath experience the highest signal activity during FIR filtering. These hardware components therefore form the main sources of power dissipation.

In addition to weighted sum computation, FIR filtering also involves updating the input data samples. For an N-tap filter, the latest N data samples are required. Hence, the latest N samples need to be stored in the data memory. After every output computation, a new data sample is read and stored in the data memory and the oldest data sample is removed. A data sample X[k] for the current computation becomes data sample X[k –1] for the next FIR computation. Thus in addition to accepting the new data sample, the existing data samples need to be shifted by one position, for every output. The power dissipated due to this data movement Can be minimized by configuring the data memory as a circular buffer [2] where instead of moving the data, the pointer to the data is moved

2.2.Measures of Power Dissipation in Buses

For a typical embedded processor, address and data buses are networks with a large capacitive loading [3]. Hence,signal switching in these networks has a significant impact on power consumption. In addition to the net capacitance of each signal(bit) of the bus, intersignal cross-coupling capacitance also contributes to the bus power dissipation. The power dissipation due to intersignal capacitance varies depending on the adjacent signal values. The current required for signals to switch between “5’s” (0101b) and A’s (1010b) is about 25% more than the current required for the signals to switch between “0’s” (0000b) and F’s (1111b) .

The Hamming distance between consecutive signal values and the number of adjacent signals toggling in opposite direction thus form the measures of power dissipation in the buses.

2.2. Measures of Power Dissipation in the Multiplier

Due to high speed requirements, parallel array architectures are used for implementing dedicated multipliers in programmable DSP’s [4]. The power dissipation of a multiplier depends on the multiplier input values. This dependence can be analyzed using the “transition density” [5] measure of circuit activity. For an array multiplier, it can be shown that the power is directly dependent on the transition densities and the probabilities of the multiplier inputs. The transition densities of the multiplier inputs depend on the Hamming distance between successive input values. The input signal probabilities depend on the number of “1’s” in the input signal values of the multiplier. These two thus form the measures of multiplier power dissipation.

III. Hamming Distance Minimization problem

3.1Problem definition

For a Given N-tap FIR filter with coefficients A_i, $i = 0, N\text{-}1$ that satisfy the filter response in terms of passband ripples, stopband attenuation and linear phase, find a new set of coefficients A_i, $i = 0, N\text{-}1$ such that the total Hamming distance between successive coefficients is minimized while still satisfying the desired filter characteristics in terms of passband ripple and stopband attenuation. Also retain the linear phase characteristics if such this constraint is specified.

3.2Problem Formulation

The Hamming Distance minimization problem is formulated as a local search problem, where the optimum coefficient values are searched in their neighborhood. This is done by using an iterative improvement process. During each iteration one or more coefficients are suitably modified so as to reduce the total Hamming distance while still satisfying the desired filter characteristics. The optimization process continues till no further reduction is possible.

The coefficient optimization is done in two phases. In the first phase, all the coefficients are scaled uniformly. The advantage of such an approach is that it does not affect the filter characteristics in terms of passband ripples and stopband attenuation and phase response. The scaling results in the same gain/attenuation ratio. In the second phase of optimization one coefficient is perturbed in each iteration. In case of requirement to retain the linear phase characteristics, the coefficients are perturbed in pairs (A_i and A_{N-1-i}) so as to preserve coefficient symmetry. The selection of coefficient for perturbation and the amount of perturbation has the direct impact on overall optimization quality. Various strategies can be adopted for coefficients perturbation [8] . The strategies adopted here include *'Steepest Descent'* and *'Genetic Algorithms'*. In the Steepest Descent approach the best coefficient perturbation is selected at every stage. The new value in the neighborhood of the coefficient value is searched for which the Hamming distance is minimum from the previous value. The Genetic Algorithms are the evolutionary algorithm which generates the random numbers and selects the best fit value according to the fitness function and search the whole space to find the global value.

IV. Algorithm for Hamming Distance minimization of FIR filters using Steepest Descent technique.

Step 1:- For a given FIR filter coefficients A[i] (i =1,N-1) and given pass band ripples (P_{db_req}) and stop band attenuation(S_{db_req}). Calculate the Hamming Distance between 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])* is minimum

Step 3 :- Replace *A[i]* with *A[i+]* to get a new set of coefficients.

Step 4 :- Compute pass band ripples(P_{dbi+}) and stop band attenuation(S_{dbi+}) from a new set of coefficients *A[i+]*

Step 5 :- If pass band ripples $P_{dbi+} < P_{db\_req}$ and stop band attenuation $S_{dbi+} > S_{db\_req}$ calculate tolerance

$T_{oli+} =(P_{db\_req} – P_{dbi+})/P_{db\_req} +(S_{dbi+} - S_{db\_req})/S_{db\_req}$

Else

$Tol_{i+} = 0$

Step 6 :- Now again perturb each coefficient (decrease 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])* is minimum

Step 7 :- :- Replace *A[i]* with *A[i-]* to get a new set of coefficients.

Step 8 :- Compute pass band ripples (P_{dbi-}) and stop band attenuation (S_{dbi-}) from a new set of coefficients *A[i-]*.

Step 9 :- If pass band ripples $P_{dbi-} < P_{db\_req}$ and stop band attenuation $S_{dbi-} > S_{db\_req}$ calculate tolerance

$T_{oli-} =(P_{dbreq} – P_{dbi-})/P_{dbreq} +(S_{dbi-} - S_{dbreq})/S_{dbreq}$

Else

$Tol_{i-} = 0$

Step 10 :- Calculate gain function **γ** for new coefficient values *A[i+]* and *A[i-]*
$\gamma = (Tolerance * HD\ reduction)$ is maximum
for $\gamma > 0$

Replace original coefficients with new value

V. GENETIC ALGORITHM

Genetic algorithm (GA) is a stochastic search method inspired by evolution and adaptation in biological systems and was first presented in 1975 by John Holland[9] . The search is conducted directly in the solution space and each solution is encoded in a certain way and is called an individual. The search is parallel in the sense that a population of individuals is maintained and the quality of the individuals is calculated by a fitness function. The population is improved by crossover, recombination of genetic material from different individuals. This is based on a hypothesis that a good solution can be built up from shorter partial solutions. Genetic diversity is maintained by a mutation operation, making random changes in the individuals. To summarize, genetic algorithm consists of five components:

5.1. A chromosomal representation of solutions.

5.2. Away to create an initial population of solutions.

5.3. A fitness function.

5.4. Genetic operators (selection, crossover, mutation).

5.5. Parameter values for the genetic algorithm (population size, probabilities for applying genetic operators etc).

5.1. Working Principle

To illustrate the working principle of GA consider a unconstrained optimization problem

Maximize *f(X)*

$$X_i^L \le X_i \le X_i^U \qquad \text{for } i = 1,2,\ldots\ldots N$$

If *f(X),* for $f(X) > 0$ is to be minimized, then the objective function is written as

$$\max imize \frac{1}{1+f(X)}$$

If $f(X) < 0$ instead of minimizing *f(X),* maximize *[-f(X)]*. Hence both maximization and minimization problems can be handled by GA.

5.2. Encoding

Since genetic algorithms search directly in the solution space, it needs a way to encode solutions in a way that can be manipulated by the genetic algorithm. This representation of a solution is called a genetic or chromosomal representation of the solution.

5.3.Population

The population is usually set up by randomizing an initial set of solutions. The population size can be variable but is usually fixed to a certain size. It is by far most common that the population is purely generational. This means that the entire population is superceded by their offspring which makes up the next generation, except individuals preserved if an elitism operator is used.

5.4. The Objective and Fitness Function

The objective function is used to provide a measure of how individuals have performed in the problem domain. In the case of a minimization problem, the most fit individuals will have the lowest numerical value of the associated objective function. This raw measure of fitness is usually only used as an intermediate stage in determining the relative performance of individuals in a GA. Another function, the fitness function, is normally used to transform the objective function value into a measure of relative fitness, thus:

$F(x) = g(f(x))$

Where *f* is the objective function, *g* transforms the value of the objective function to a non negative number and *f* is the resulting relative fitness. This mapping is always necessary when the objective function is to be minimized as the lower objective function values correspond to fitter individuals.

5.5. Genetic Operators

The basic functionality of the genetic algorithm is provided by the genetic operators. These are the functions that make up the algorithm itself, the population and fitness function can be viewed as external entities that can be plugged in and changed. Even the operators and encoding can be allowed to adapt.

VI. Algorithm for Hamming Distance Minimization using GA

Step 1 Compute filter coefficients $h_I(n)$ and freq. response $H_I(\omega)$ of ideal FIR filter for $0 \leq n \leq N\text{-}1$.
Step 2 Calculate the Hamming distance (HD_I) between the FIR filter coefficients $h_I(n)$.

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 $H_D(n)$ $0 \leq i \leq k\text{-}1$ *and* $0 \leq n \leq N\text{-}1$

Step 5 Compute the frequency response of the coefficients chromosomes for $H_D(\omega)$ in population.

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

Step 7 Evaluate the fitness of the chromosomes $f(i) = f_M + f_H$

Step 8 Apply Roulette wheel selection.

Step 9 Apply crossover operator at a desired rate.

Step 10 Mutate at a desired rate.

Step 11 Evaluate again the fitness of the chromosomes.

Step12 If the Stopping criterion is met store the chromosomes according to the fitness, Else go toStep8

VII. RESULT

Hamming Distance minimization using Steepest Descent method
Lp_16K_3K_4K_.1_62_30

Sampling Frequency =16 KHz

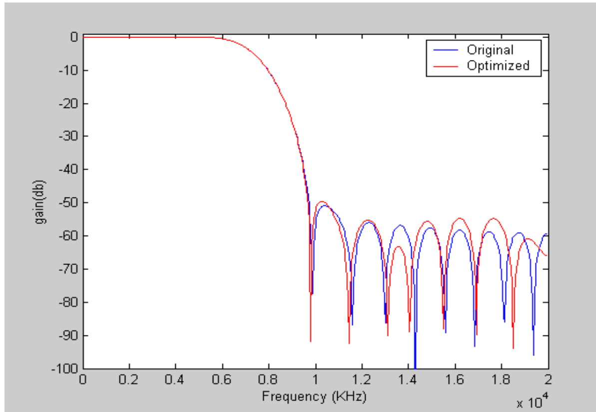
Passband Frequency =3 KHz

No .of Coefficients = 30

Initial hamming distance = 224

Final hamming distance = 170

Hamming distance reduction = 54



7.1 Result of FIR filters in terms of percentage Hamming distance and number of signal toggling reduction using Steepest descent Approach are summarized in a table given below.

Original and Optimized Response of Lp_16K_3K_4K_.1_62_30 FIR Filter

FIR Filter	Initial		Steepest Descent		% Reduction	
	HD	Togs	HD	Togs	HD	Togs
Lp_16K_3K_4K_.1_62_30	224	43	169	29	24.91%	30.95%

Result for FIR filter using Steepest Descent Approach

7.2 Hamming distance minimization using Genetic Algorithms

Lp_12K_2K_3K_.12_45_28

Sampling Frequency =12 KHz

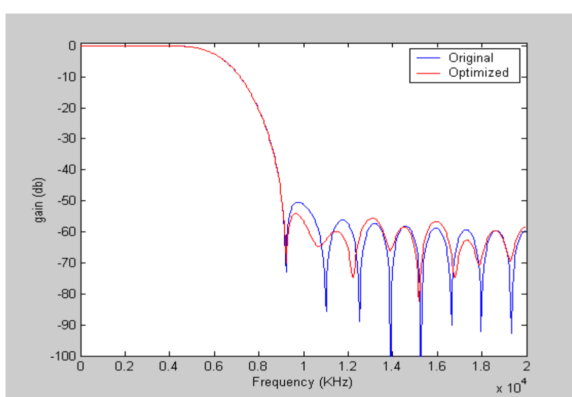
Passband Frequency =2 KHz

No. of Coefficients = 28

Initial hamming distance = 216

Final hamming distance =144

Hamming distance reduction = 72



Original and Optimized Response of Lp_12K_2K_3K_.12_45_28 FIR Filter

Result of FIR filters in terms of percentage Hamming distance and number of signal toggling reduction using Genetic Algorithm Approach are summarized in a table given below

FIR Filters	Initial		Genetic Algorithm		% Reduction	
	HD	Togs	HD	Togs	HD	Togs
Lp_16K_3K_4K_.1_62_30	216	44	144	17	33.33%	61.36%

Result for FIR filter using Genetic Algorithm

VIII. CONCLUSION

The Hamming distance minimization results for low pass FIR filter show that the total Hamming distance can be reduced upto 25% and total number of signal toggles can be reduced up to 45% obtained by Steepest Descent technique. For the same FIR filter Hamming distance can be reduced upto 34% and total number of signal toggles can be reduced upto 61% by Genetic Algorithm. So, best optimization results are Genetic Algorithm. This Hamming distance reduction directly translates into power saving in multipliers while implementing FIR filter on Digital Signal Processors (DSPs).

REFERENCES

[1] Edward A.Lee, "Programmable DSP Architectures : Part I" , IEEE ASSP Magazine ,October 1998,pp 4-19
[2] TMS320C2x/C5x User's Guides, Texas Instruments, 1993.
[3] C.-L. Su, C.-Y. Tsui, and A. M. Despain, "Saving power in the control path of embedded processors," *IEEE Design Test Comput.*, pp. 24–30,Winter 1994
[4] G.-K. Ma and F. J. Taylor, "Multiplier policies for digital signal processing," *IEEE ASSP Mag.*, Jan. 1990, pp. 6–19.
[5] F. Najm, "Transition density: A new measure of activity in digital circuits," *IEEE Trans. Computer-Aided Design*, pp. 310–323, Feb. 1993.
[6] K. K. Parhi, "Algorithms and architectures for high-speed and lowpower digital signal processing," in *Proc. 4th Int. Conf. Advances in Commun. Contr.*, Rhodes, Greece, June 1993.
[7] M. T.-C. Lee, V. Tiwari, S. Malik, and M. Fujita, "Power analysis and minimization techniques for embedded DSP software," *IEEE Trans.VLSI Syst.*, pp. 123–135, Mar. 1997.
[8] Dusan Kodek and Kenneth Steiglitz, *"Comparison of Optimal and Local Search methods for Designing Finite Wordlength FIR Digital Filters",* IEEE transactions circuits and systems vol. CAS-28, no.1, January 1981.
[9] Goldberg David E., *"Genetic Algorithms in Search, Optimization and Machine Learning*", Pearson Education Asia, 2002.