Design of FIR Filter Using Adaptive LMS Algorithm for Energy efficient applications

N Archana[1], SK Mobeen[2], K Bhargavi[3], T Pavankumar[4], P.Vali Basha[5],

[1,2,3,4]B.Tech Student, Department of ECE, N.B.K.R Institute of Science and technology, Nellore, India,

[5]Assistant Professor, Department of ECE, N.B.K.R Institute of Science and technology, Nellore, India,

Email: vali.gps@gmail.com

Abstract: The paper presents low power and area efficient LMS (Least Mean Square) adaptive filter. Least Mean Squares algorithm used in many DSP applications to process the signal and designed for generating filter coefficients based on the processed signal. The proposed method reduces the power consumption and area due to adaptation of the weights generated internally according to the corrected output and previous weights. The weights area calculated using LMS algorithm. The proposed Simulation is done using Modelsim, structure coded using Verilog HDL and synthesized by Xilinx ISE. The power is calculated using Xilinx power estimator by targeting to FPGA. The observed results for the proposed system improved the area and power compared to conventional FIR filter.

Keywords: Adaptive noise canceller, Adaptive, FIR filter, Least Mean Square

I. Introduction

Adaptive Filters are widely used in many signal processing applications such as system identification and modeling, equalization, interference and echo cancellation. The tapped delay line finite impulse response (FIR) filter whose weights are updated by the famous Widrow-Hoff least mean square algorithm is the most popularly used adaptive filter not only due to its simplicity but also due to its satisfactory convergence performance. In the last two decades, the multiplier less distributed arithmetic (DA)-based technique has increased its importance in the implementation of filters, due to its high throughput processing capability and regularity, which results in cost effective and area-time efficient computing structures. In the literature DA based design of Adaptive filter has been suggested. A fixed-coefficient filter can be easily realized using DA by storing the partial products of filter coefficients in the LUT. But in adaptive filter implementation, there is some difficulty in partial products of the filter coefficient stored in the LUTs. These are to be calculated every time before filtering.

Although it is necessary to usage fixed point arithmetic when compared with floating point arithmetic it occupies a large area in FPGA. Mostly, in adaptive filters is constructed using FIR structure instead of IIR structures. Hence, the output of FIR filters is the combination of inputs and the coefficients.

However, introduction of adaptive FIR filter requires a suitable algorithm to update the filter coefficients according to the input samples. LMS algorithm is used to update the filter coefficients, due to the low computational, simplification and better performance. The LMS algorithm used in implementing adaptive algorithms seems give better performance when compared to other algorithms. When compared to LMS, Recursive Least Squares (RLS) algorithm gives faster convergence but slows down the system performance and increases area. In fixed point arithmetic, the adaptation process must be decreased to improve the further performance of adaptive FIR filter, by considering the shorter bit length for representing the coefficients.

II. Previous Work

Finite Impulse Response (FIR) filter:

FIR filter is a type of digital filter. It's also called as non-recursive filter because it does not have the feedback. This system present output is depends upon the present and past input. The duration of the impulse response is finite which ranges from 0 to N-1. The impulse response of an Nth-order FIR filter lasts for N+1 sample, and then dies to zero. The FIR filter output is expressed as a convolution of the coefficient sequence with the input signal which is given by difference equation,

$$y(n) = h(n) * x(n) = \sum_{n=0}^{N-1} h(n)x(k-n)$$

Where x(n) is the input signal, y(n) is the output signal, h(n) is the filter coefficients (impulse response), N is the filter order (taps). FIR filters are particularly useful for applications where exact linear phase response is required. The FIR filter is generally implemented in a non-recursive way which guarantees a stable filter. Fig. shows the basic block diagram for a four-tap low pass FIR filter.

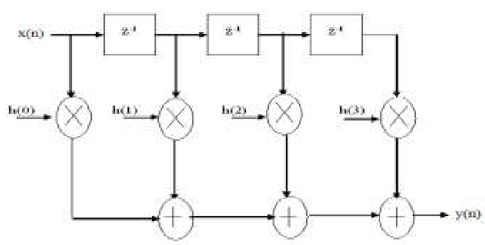


Fig.1. Direct form FIR filter structure

This direct form structure uses 4 multiplications and 3 additions per sample. In this 4 tap low pass FIR filter, the coefficients are multiplied with the corresponding delayed data samples and that are accumulated to produce the result. The FIR filter is designed by using following steps, Obtain the difference equation for filter output in terms of input and order of filter.

- Determination of filter coefficient or tap weights used to multiply against delayed sample values.
- Selection of number of taps based on the stop band attenuation, less rippling and steeper roll off.
- Selection of appropriate windowing method for calculating the window coefficient with minimal computational effort.
- Determine the value of filter coefficient and windowing function based on the frequency response and cutoff frequency.
- Design of optimized PDP n bit delay element, adder and multiplier for producing the output with low power and delay.

The filter coefficients are determined by using windowing method which is always preferred because of its relative simplicity, ease of use and minimal computational effort as compared to other methods. The windowing function is used to reduce the ringing effect which takes place near the band edge of the filter. It has the effect of making the filter impulse response as finite in duration and also has the effect of smearing or smoothing the desired frequency response. The Hamming window is preferred because of its ability to reduce its side lobes while maintaining the main lobe width as $8\pi/N$ than other windowing.

Generally, to filter out the input signal it actually requires the reference signal, whereas it is absence in the time invariant filter. In overall, the adaptive filter block diagram is shows in the Fig. 2, whereas iteration index is indicates by n, the signal input is meant by x(n),adaptive filters output signal is denoted by y(n), and reference or desired signal is denoted by d(n). The error signal e (n) is obtained from the filter output y (n) subtracted from the desired d (n). The error signal is given as a input to the adaptive algorithm to determine the updated filter coefficients. The minimization objective used to match the adaptive filter's output signal with the desired signal.

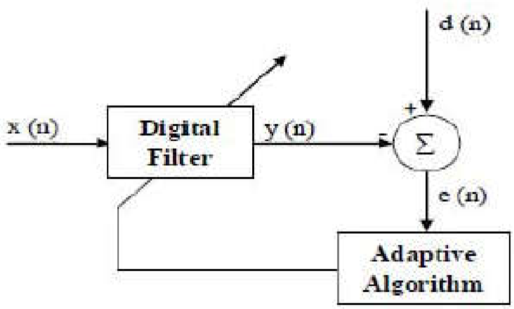


Fig. 2. Block diagram of a conventional Adaptive Filter.

III. Proposed Work

A. Adaptive noise cancellation

Mostly the adaptive filter is used in noise cancellation applications. The desired signal is combination of source signal and noise signal which is uncorrelated to the signal as shown in Fig. 3.Filter takes a noise input and correlates with the noise in desired signal to obtain the actual signal. Input of a filter is a reference noise which is correlated with the noise in the desired signal. The error term e (n) obtained from the system is then used to cancel the noise in the original signal by using the LMS algorithm.

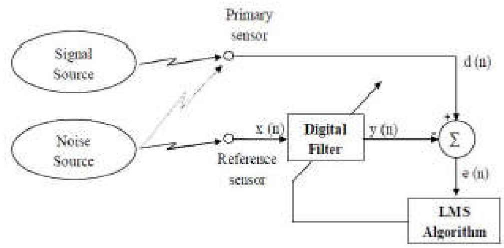


Fig. 3. Architecture block diagram of LMS fir filter with input source.

B. LMS algorithm

Most widely use algorithm in adaptive filter is an LMS algorithm due to its simplicity. It doesn't needs an extra mathematical calculation like matrix inversion nor correlation function. Mean Square Error (MSE) logic is used in LMS algorithm. It uses an input signal, step-size parameter, the subtraction of desired signal and filter output signal for calculating the updated filter coefficients.

1) LMS Equation*:* Based on the filter taps and input response the equation will be obtained for number of iterations. The equation to updated tap weight w (n) using the input signal x (n) and the desired response d (n) with step size μ is shown in equation.

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{x}(n) [d(n) - \mathbf{x}(n)\mathbf{w}(n)] \quad (1)$$

Whereas μ is the step size, hence the filter output is the sum of the product of tap weights and input signal

$$y(n) = \mathbf{x}(n) * \mathbf{w}(n) \quad (2)$$

Error signal e (n) is defined as the subtraction of the desired signal and the filter response signal

$$e(n) = d(n) - y(n) \quad (3)$$

So, Equation (1) can be further written in terms of the error signal and the tap weights:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu\mathbf{x}(n)\, e(n) \quad (4)$$

The formula for the LMS algorithm is shown in equation (4). As illustrated in the equation, each updated tap weight needs the current tap weight and the current error signal obtained from the desired response after subtraction. The algorithm doesn't require depth knowledge of the whole cross-correlation vector or autocorrelation matrix doesn't require matrix computations.

2) Convergence boundary*:* Convergence time of LMS algorithm is assigned based on step size μ. Suppose if μ is small, convergence time may take long and it may not serve the purpose of LMS filter usage. However if μ is too large, then the algorithm may not be converge forever. LMS algorithm proven to converge for values of μ which is less than the reciprocal of the largest Eigen value of the autocorrelation matrix of x(n).

3) Selection of Adaptive Parameters: The performance of LMS is decided based on step size parameters and filter order. A step size parameter μ introduced here is used to control the step width of the iteration and thus convergence speed and the stability of the algorithm.

C. Adaptive FIR filter Structural

The LMS algorithm uses a structure of an FIR filter. The structural view of FIR filter is shown in Fig. 3. From the figure 4, filter as two main components those are L delay registers and weight update blocks. The Unit Delay Registers are made of simple D Flip-Flops. And each Weight Update component consists of a multiplier, adder and a buffer to store the new updated weights of the filter coefficient. According to equation (3) error signal is obtained from the difference of the filter output and desired signal. The error signal is then multiplied with input signal and step size μ, which produces next sets of filter coefficients.

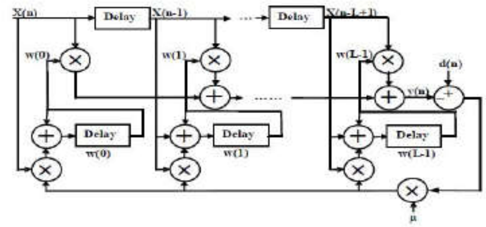


Fig. 4. Structural view of FIR filter with weights using LMS algorithm.

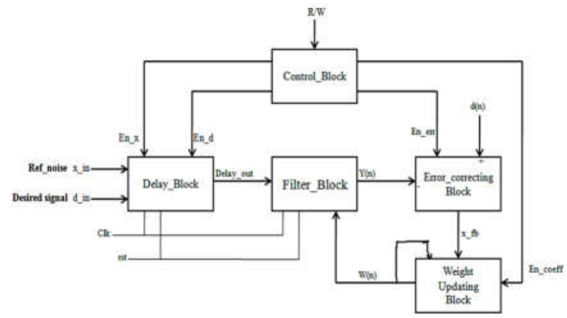


Fig. 5 External view of proposed FIR filter with LMS algorithm

The LMS architecture is divided into five blocks-

1. Controller block
2. Delay based Block
3. Filter Block
4. Error Correcting Block
5. Weight Updating Block

1) Controller block

The Controller block controls the synchronization of the complete system. It generates four control enable signals, which triggers the Delay based Block, the Weight Updating Block and the Error Correcting Block separately. When read=1 all the enable signals get high and reads current input and produces the respective output. When write=1 all the triggering signals will not trigger any block so it doesn't read any input signal but checks the output.

2) Delay based block

The Delay based Block gets the reference noise signal x_in as an input and the desired input signal d_in by triggering the signal en_x and en_d. And it produces the delay of M tap delay. When enable signals get high output follows the input otherwise it will produce delay signal.

3) Filter block

The Multiply Accumulator (MAC) Block multiplies the M tap delay signal Delay_out with the M tap weight w(n) separately, and adds to get y(n). In Filter block, if the clock is posedge and reset is enabled then filter output will be zero and when clock is negedge and reset is disabled then filter output will obtained by multiplying reference input signal and updated weight.

4) Error correcting block

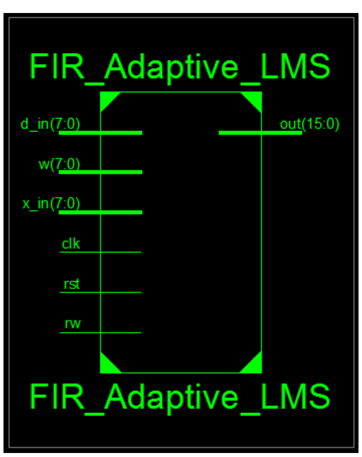
The Error Correcting Block subtracts y(n) from d(n) and obtain the approximate e_out error signal. And it multiplies x_out, e_out and u to produce signal x_fb. When en_err enable signal get enabled it will generate error signal e_out and feedback signal x_fb.

5) Weight Update Block

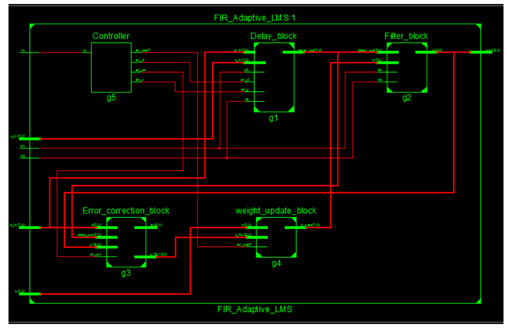
The Weight Update Block updates the vector weight w(n) to w(n+1) that will be used in the next iteration. When enable signal en_coeff get triggered next weight equal to current weight plus feedback signal otherwise next weight equal to zero.

IV. Simulation Results

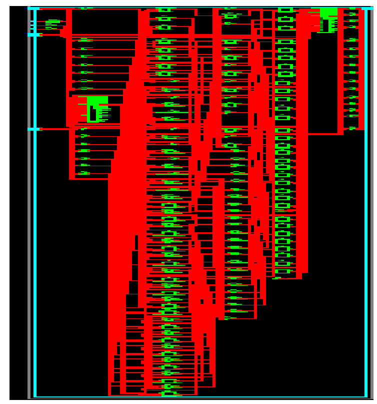
Block diagram



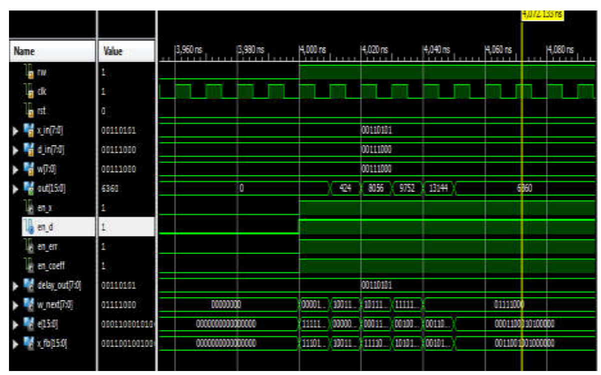
RTL Schematic:

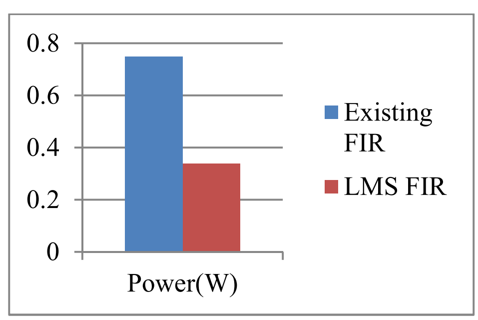


Technology Schematic:

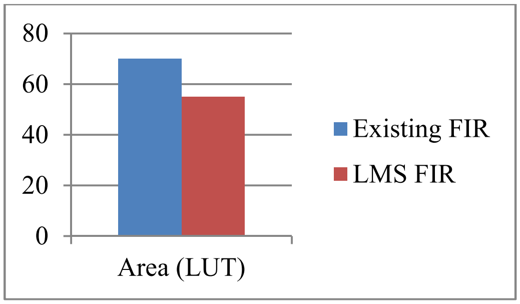


Output Simulation





Power Comparison



Area Comparison

V. CONCLUSION

The proposed architecture shows the area efficient Implementation of adaptive LMS algorithm based FIR filter. The simulation results have been observed by using Xilinx simulator. Synthesized by XILINX XST tool and targeted for SPARTAN 3E FPGA device XC3S250E. Device utilization values of both algorithms are compared. From the area results, observed that the area reduced in terms of slices and LUTs compared to Conventional FIR filter. Implemented architecture also reduced the power consumption by the Adaptive algorithm using LMS. The observed results for the proposed adaptive LMS algorithm improved area and less in power compared to conventional FIR filter.

REFERENCES

[1] Shweta B Throat. Prof. S.I.Nipanikar “Adaptive Filter using FPGA For Audio Enhancement”, IJAETE 2016.

[2] S. Haykin and B. Widrow, Least-Mean-Square Adaptive Filters. Hoboken, NJ, USA: Wiley-Interscience, 2003.

[3] L.-K.Ting, R.Woods, and C. F. N. Cowan, “Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 1, pp. 86–99, Jan. 2005.

[4] Liang Zhao, Wen-Zhan Song, Xiaojing Ye, “Fast Decentralized Gradient Descent Method and Applications to In-situ Seismic Tomography”, 2015IEEE.

[5] Bharat Naresh Bansal, Dr. Amanpreet Singh, “Study of Various Techniques for Designing of Efficient FIR Digital Filter”, IJARCSSE, Volume 4, Issue 1, January 2014

[6] Xilinx, “Spartan-3E FPGA Family Complete Data Sheet”, Data Sheet DS312, Xilinx,Inc.,2005.

[7] P. K.Meher and S. Y. Park, “Low adaptation-delay LMS adaptive filter Part-II: An optimized architecture,” in Proc. IEEE Int. Midwest Symp. Circuits Syst., Aug. 2011.

[8] HesamAriyadoost, Yousef S. Kavian, Karim Ansari As “Performance Evaluation of LMS and DLMS Digital Adaptive FIR Filters by Realization on FPGA”, International Journal of Science & Emerging Technologies, Vol-1 No. 1, 7-10, September, 2011.

[9] E. Mahfuz, C. Wang, and M. O. Ahmad, “A high throughput DLMS adaptive algorithm,” in Proc. IEEE Int. Symp. Circuits Syst.,May 2005, pp. 3753–3756.

[10] A. B. Diggikar, S.S.Ardhapurkar “Implementing FSM to control Adaptive Filter for Noise Cancelling in Speech signal”, Proc. of the International Conference on Advanced Computing and Communication Technologies, 452-457,2011.