Efficient Adaptive FIR Filter Design Based On CSA

Sachin Kumar  
M. Tech, Student,  
Department of ECE, RNSIT,  
Bengaluru, Karnataka

B.S Usha  
Assistant Professor,  
Department of ECE, RNSIT,  
Bengaluru, Karnataka

Abstract: A novel approach for adaptive filter based on distributed arithmetic (DA) to get high throughput using low area implementation is done. The achieved throughput rate of the proposed design is increased by the use of lookup table (LUT) update, weight update operations and Filtering implementation. For the reduction of the sampling period and area complexity the conventional adder based shift accumulator is replaced by the conventional adder based shift accumulator. Using fast bit-clock for carry-save accumulation with slower clock for all other operations reduction of power consumption is done in the proposed system. The design made use of half the number of multiplexers, smaller LUT and adders when compared to the existing system.

Keywords: Adaptive FIR filter, Distributed Arithmetic (DC), CSA (Carry Sum Adder)

I. INTRODUCTION

Most of all the algorithms in signal processing involve DA for computing the inner product of two vectors consisting of most of the computational workload. Because of which the DA is widely used in the signal processing algorithms. Using the adders and multipliers the evaluation of the inner product is done. Using DA these multipliers can be replaced for obtaining better performance by reducing the computational workload.

Most of all the algorithms in signal processing involve DA for computing the inner product of two vectors consisting of most of the computational workload. Because of which the DA is widely used in the signal processing algorithms. Using the adders and multipliers the evaluation of the inner product is done. Using DA these multipliers can be replaced for obtaining better performance by reducing the computational workload. The main cause for reduction is done due to the pre computation of the partial sum of the coefficients of filter in LUT. Due to which the DA will have fewer arithmetic resources without multipliers. DA based Adaptive filters are widely used in several digital signal processing (DSP) applications.

Kairoju Ramachary et.al [03] proposed an efficient method called Least Mean Square for cancellation of noise in electrocardiographic signals. The proposed methodology uses block based error non linear signed regressive LMS algorithm for increasing the convergence speed of the stationary signals for good tracking capability in non stationary signals. The implementation is done in Xilinx tool. Wasim Maroofi et.al [04] presented different pipeline architecture for low power implementation of Adaptive filter using DA.

The design swapped by conditional signed carry save accumulator instead of the traditional adder based shift accumulator for DA based Computation of the inner product for obtaining good efficiency with reduced power consumption. Jyothirmayi Alahari et.al [10] delegated a novel approach of pipelined architecture implementation of the Adaptive filter using DA for to get high throughput with less power and area. By updating the LUT the increase in throughput is done. For reduction of area complexity and sampling period conditional signed carry save accumulation is used. The proposed approach used efficient method for Adaptive filter design using
II. PROPOSED SYSTEM

A. TRADITIONAL ADAPTIVE FILTER

An Adaptive filter is as shown in the Figure 1. It is seen that when the input signal \( x(n) \) is fed into a adaptive filter, the corresponding output signal sample \( y(n) \) is generated at time \( n \). Comparison of this signal is done with the second signal \( d(n) \) called as the desired response signal by finding difference of these signals at time \( n \). The difference signal is called as \( e(n) \) which is called as the error signal. This signal is then fed into a block where the update of the filter coefficients are done by changing the parameter of the filter from time \( n \) to time \( (n+1) \) in a proper manner. When the increment in the time index \( n \) is done it is expected that the output of the adaptive filter matches properly to the desired response signal with the help of this adaptive process which causes the magnitude of \( e(n) \) decrease over time.

![Figure 1: Block Diagram of Proposed Work](image)

A. LEAST MEAN SQUARE (LMS) ALGORITHM

LMS approach uses a gradient based technique for steepest decent. It mainly consists of two basic steps: filtering process for computing the output of a linear filter with respect to the input signal and generation of the error estimation is also done by comparing this output with the desired response. Second step is to perform an adaptive process for adjusting the parameter of the filter in accordance with the error estimation. With each iteration of the LMS algorithm, the weights of the filter tap are updated according to the following formula given by,

\[
W(n + 1) = W(n) + \mu \cdot e(n) \cdot x(n)
\]  

(1)

\[
e(n) = d(n) - y(n)
\]  

(2)

\[
y(n) = W^T(n) \cdot x(n)
\]  

(3)

Where \( x(n) \) denotes the input vector, \( \mu \) denotes the learning rate parameter, \( y(n) \) is the filter output, \( w(n) \) denotes the weight vector for \( n \) no of iterations [01], [02].

B. DISTRIBUTED ARITHMETIC (DA)

For computing the inner dot product of a constant coefficient vector we make use of DA [05], [06], [07] and a visible input vector in a single step given by,

\[
y = \sum_{k=0}^{N-1} W_k \cdot x_k
\]  

(4)

Where \( W_k \) depicts the fixed coefficients and the input is denoted by \( x_k \). If each of these inputs \( W_k \) is a 2’s complement binary number scaled in a way that if \( |W_k| < 1 \) then each \( W_k \) can be presented as

\[
W_k = -W_{2^0} + \sum_{i=1}^{L-1} W_{2^i} \cdot 2^{-i}
\]  

(5)

Where \( W_k \) represents the Lth bit of \( W_k \).

Equations (4) and (5) are combined to get the distributed arithmetic computation as,

\[
y = \sum_{i=2^{L-1}}^{L-1} \left[ \sum_{k=0}^{N-1} x_k \cdot W_k \right] - \sum_{k=0}^{N-1} W_k \cdot x_k
\]  

(6)

Traditional implementation based on DA is as shown in the Figure 2.

![Figure 2: Traditional DA based Implementation](image)
Where \( k_j \) denotes the \((j + 1)\)th bit of \(N\)-bit binary illustration of integer \( k \) for \(0 \leq k \leq 2^N - 1\). Word that \( C_k \) for \(0 \leq k \leq 2^N - 1\) can also be pre-computed and stored in RAM-situated LUT of \(2^N\) phrases. As an alternative of storing \(2^N\) phrases in LUT, we store \(2^N - 1\) words in a DA table of \(2^N - 1\) registers. An instance of this sort of DA \([08], [09]\) table for \(N = 4\). It includes most effective 15 registers for storing the pre-computed sums of input words. Seven adders in parallel compute the brand new values of \( C_k \). Brought with an input carry “1” to generate filter output which is therefore subtracted from the preferred output \(d(n)\) to acquire the error \(e(n)\) which may require large LUT. Hence this proposed method overcome this problem using less no of LUT the overall Design is as shown in the Figure 4 with only 4 delay i.e. \(N = 4\). It contains only 4 inner product blocks and a necessary weight increment block along with the additional circuits for computing the error.

III. RESULTS AND DISCUSSION

Figure 5 depicts the Chip level diagram of the Proposed Adaptive LMS filter. Figure 6 gives the Schematic design obtained for LMS design using CSA adder using Xilinx tool. Figure 7 gives the Building Block of the CSA adder. Figure 8 depicts the overall Design Summary obtained for the Proposed System. Figure 9 depicts the obtained analog waveform for the give input signal.
IV. CONCLUSION

This paper offered the implementation of carry save adder scheme of based internal products for the computation of filter output. It is well carried out for Adaptive Filtering design. From the synthesis results, it was discovered that the proposed design consumes less power compared to our earlier DA based FIR adaptive filter.

REFERENCES